การรู้จำตัวอักษรไทยโดยใช้ซัพพอร์ตเวกเตอร์แมชชีนและเคอร์เนล

นาย พัฒนชัย เบศรภิญโญวงศ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2545

ISBN 974-171-621-4

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

THAI CHARACTER RECOGNITION USING SUPPORT VECTOR MACHINES AND KERNELS

Mr. Patanachai Bespinyowong

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science

Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University

Academic Year 2002
ISBN 974-171-621-4

หัวข้อวิทยานิพนธ์	การรู้จำตัวอักษรไทยโดยใช้ซัพพอร์ตเวกเตอร์แมชชีนและเคอร์เนล
โดย	นาย พัฒนชัย เบศรภิญโญวงศ์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.วันชัย ริ้วไพบูลย์)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... กรรมการ
(อาจารย์ ดร.อรรถวิทย์ สุดแสง)

..... กรรมการ
(อาจารย์ ดร.วีระ เหมืองสิน)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

พัฒนชัย เบศรภิญโญวงศ์ : การรู้จำตัวอักษรไทยโดยใช้ซัพพอร์ตเวกเตอร์แมชชีนและเคอร์เนล. (THAI CHARACTER RECOGNITION USING SUPPORT VECTOR MACHINE AND KERNEL) อ. ที่ปรึกษา : ผศ.ดร.บุญเสริม กิจศิริกุล, 136 หน้า. ISBN 974-171-621-4.

วิทยานิพนธ์ฉบับนี้มุ่งเน้นที่จะปรับปรุงความถูกต้องในการรู้จำของโปรแกรมโอซีอาร์ภาษาไทย โดยได้นำเอาเทคนิคของซัพพอร์ตเวกเตอร์แมชชีน (เอสวีเอ็ม) และเคอร์เนลเข้ามาประยุกต์ใช้ในส่วนของการวิเคราะห์องค์ประกอบสำคัญของข้อมูล ซึ่งเป็นกระบวนการที่สำคัญในการดึงเอาลักษณะสำคัญของข้อมูลรูปภาพตัวอักษร ก่อนที่จะส่งข้อมูลที่ได้ไปยังส่วนรู้จำของโปรแกรมโอซีอาร์เพื่อทำการแยกแยะว่าเป็นตัวอักษรชนิดใดต่อไป โดยเรียกเทคนิคการวิเคราะห์องค์ประกอบสำคัญของข้อมูลแบบใหม่นี้เรียกว่าการวิเคราะห์องค์ประกอบสำคัญของข้อมูลแบบเคอร์เนล

ในวิทยานิพนธ์ฉบับนี้ ได้แบ่งรูปภาพที่ใช้ทดสอบออกเป็นสองกลุ่ม คือรูปภาพชุดเรียนรู้จำนวน 8,544 ตัว และรูปภาพชุดทดสอบจำนวน 1,424 ตัว ประกอบด้วยตัวอักษรแบบ AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC, EucrosiaUPC และ FreesiaUPC แต่ละแบบประกอบด้วยตัวอักษรขนาด 14, 16, 18, 20, 22, 24, 28 และ 36 จุด ผลของการทดสอบพบว่าผลของการรู้จำของโปรแกรมโอซีอาร์ภาษาไทยที่ใช้เทคนิคของการวิเคราะห์องค์ประกอบสำคัญของข้อมูลแบบเคอร์เนลให้ผลการรู้จำที่ดีขึ้นจากโปรแกรมโอซีอาร์ภาษาไทยตัวเดิม อย่างไรก็ตาม วิธีใหม่นี้กลับใช้หน่วยความจำและเวลาที่เพิ่มขึ้นจากเดิม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2545

4270453421 : MAJOR COMPUTER SCIENCE

KEY WORD: THAI-OCR / SUPPORT VECTOR MACHINE / PRINCIPAL COMPONENT ANALYSIS / CHARACTER RECOGNITION PROCESSING / FEATURE EXTRACTION / KERNEL PRINCIPAL COMPONENT ANALYSIS

PATANACHAI BESPINYOWONG : THAI CHARACTER RECOGNITION USING SUPPORT VECTOR MACHINE AND KERNEL. THESIS ADVISOR : BOONSERM KIJSIRIKUL, Ph.D., 136 pp. ISBN 974-171-621-4.

The objective of this thesis is to improve the accuracy of a Thai Optical Character Recognition (Thai-OCR) program. We extend the Principal Component Analysis method, which is used to extract features from character images, to a new method called Kernel Principal Component Analysis by using Support Vector Machines and Kernels.

In this thesis, we divided the data into 2 groups: the training set of 8,544 character images and the test set of 1,424 character images. In our experiment, the data set consists of character images from 6 fonts: AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC, EucrosiaUPC and FreesiaUPC each font composed of size 14, 16, 18, 20, 22, 24, 28 and 36 points. The experimental results show that Thai-OCR which uses Kernel Principal Component Analysis gives better results than the previous one using the original Principal Component Analysis. However, the new method consumes more memory space and processing time.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

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

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

Academic year 2002

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ ผู้ช่วยศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้ความช่วยเหลือ คำแนะนำ และแนวทางต่าง ๆ อันเป็นประโยชน์ในการทำวิจัยตลอดมา ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่านที่สละเวลาให้ข้อเสนอแนะ และคอยช่วยตรวจแก้วิทยานิพนธ์ให้มีความสมบูรณ์มากที่สุด ขอขอบคุณพี่ ๆ เพื่อน ๆ ในห้องปฏิบัติการอัจฉริยภาพเครื่องกล และการค้นพบความรู้ (MIND LAB) สำหรับทุกความช่วยเหลือในหลาย ๆ ด้าน ตลอดจนขอขอบคุณพี่ ๆ และเพื่อน ๆ รอบข้างคนอื่น ๆ ที่คอยให้กำลังใจ อันเป็นพลังในการทำงานจนสำเร็จลุล่วงในท้ายที่สุด

สุดท้ายนี้ ผู้วิจัยใคร่กราบขอบพระคุณ บิดา มารดา ที่ให้การสนับสนุน มอบความเข้าใจ และเป็นกำลังใจที่ดีให้แก่ผู้วิจัยเสมอมา ขอขอบพระคุณครับ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฌ
สารบัญภาพ ..... ญ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 งานวิจัย และทฤษฎีที่เกี่ยวข้อง 2
1.3 วัตถุประสงค์ของการวิจัย 3
1.4 ขอบเขตของการวิจัย 3
1.5 ขั้นตอนการดำเนินการวิจัย 4
1.6 ประโยชน์ที่คาดว่าจะได้รับ 4
บทที่ 2 แนวคิด และทฤษฎีที่เกี่ยวข้อง 5
2.1 หลักการทำงานของโปรแกรมโอซีอาร์ 5
2.2 ข้อมูลรูปภาพตัวอักษร 7
2.3 ซัพพอร์ตเวกเตอร์แมชชีน และเคอร์เนล 8
2.3.1 การลดความเสี่ยงเชิงโครงสร้างให้ต่ำสุด (Structural Risk Minimization) 9
2.3.2 ปริภูมิแสดงลักษณะสำคัญ กับเคอร์เนล (Feature Space and Kernel) 11
2.3.3 ระนาบหลายมิติที่ใช้แยกข้อมูลได้ดีที่สุด (Optimal Margin Hyperplane) 12
2.4 การวิเคราะห์องค์ประกอบสำคัญของข้อมูล 16
2.4.1 การวิเคราะห์องค์ประกอบสำคัญ (Principal Component Analysis) 16
2.4.2 การวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล (Kernel Principal Component Analysis) 18
2.5 นิวรอลเน็ตเวิร์กแบบแบคพรอพาเกชัน 21
2.5.1 การทำงานของนิวรอลเน็ตเวิร์กแบบแบคพรอพาเกชันในขั้นตอนการเรียนรู้ 22
2.5.2 การทำงานของนิวรอลเน็ตเวิร์กแบบแบคพรอพาเกชันในขั้นตอนการรู้จำ 24

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนา 25
3.1 การแปลงข้อมูลรูปภาพตัวอักษร 25
3.2 การแยกลักษณะสำคัญของรูปภาพตัวอักษร 27
3.3 การแยกแยะตัวอักษร 28
บทที่ 4 ผลการวิจัย 31
4.1 วิธีการทดสอบ 31
4.2 ผลการทดสอบ 32
4.3 ปัญหา และข้อจำกัด 35
บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ 37
5.1 สรุปผลการวิจัย 37
5.2 ข้อเสนอแนะ 37
รายการอ้างอิง 39
ภาคผนวก 41
ภาคผนวก ก การทดลองเลือกค่าสัมประสิทธิ์ฟังก์ชันเคอร์เนล 42
ภาคผนวก ข การใช้งานโปรแกรมไทยโอซีอาร์ 49
ภาคผนวก ค การใช้งานโปรแกรมต่าง ๆ ในขั้นตอนการเรียนรู้ 60
ภาคผนวก ง ตัวอักษรที่ใช้ในการเรียนรู้ 64
ประวัติผู้เขียนวิทยานิพนธ์ 89

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้าที่

ตารางที่ 1 ขนาดของเวกเตอร์รูปแบบที่ใช้กับการวิเคราะห์องค์ประกอบสำคัญแต่ละแบบ 28
ตารางที่ 2 ค่าโนดนำเข้า โนดซ่อน และโนดทางออกที่ใช้ในงานวิจัย 29
ตารางที่ 3 ผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดของการวิเคราะห์องค์ประกอบสำคัญแต่ละแบบ 32
ตารางที่ 4 ผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดของการวิเคราะห์องค์ประกอบสำคัญ แบบเคอร์เนลที่ค่าอัตราเรียนรู้ (learning rate) แตกต่างกันในแต่ละค่า 34
ตารางที่ 5 ผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดที่ได้จากโปรแกรมโอซีอาร์ภาษาไทยแต่ละชนิด 34
ตารางที่ 6 ผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดเมื่อใช้ วิธีการวิเคราะห์องค์ประกอบสำคัญแบบเดิม 42
ตารางที่ 7 ผลของการรู้จำรูปภาพตัวอักษรที่ผิดพลาดในฟังก์ชันเคอร์เนลแต่ละชนิด เมื่อทำการเรียนรู้ด้วยชุดรูปภาพขนาด 14 และ 16 44
ตารางที่ 8 ผลของการรู้จำรูปภาพตัวอักษรที่ผิดพลาดในฟังก์ชันเคอร์เนลแต่ละชนิด เมื่อทำการเรียนรู้ด้วยชุดรูปภาพขนาด 18 และ 20 45
ตารางที่ 9 ผลของการรู้จำตัวอักษรผิดพลาดน้อยที่สุดในฟังก์ชันเคอร์เนล 4 แบบที่เลือกไว้ เมื่อทำการปรับค่าการทำงานเพิ่มเติม 47

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้าที่

รูปที่ 1 ขั้นตอนการทำงานโดยหลักของโปรแกรมโอซีอาร์ 5
รูปที่ 2 แบบจำลองขั้นตอนการรู้จำตัวอักษร 6
รูปที่ 3 ลักษณะการสแกนเพื่อแปลงจุดภาพเป็นค่าเวกเตอร์ 8
รูปที่ 4 แผนภาพแสดงการเปลี่ยนแปลงของข้อมูลรูปภาพตัวอักษร 8
รูปที่ 5 ตำแหน่งการเลือกค่าฟังก์ชันการตัดสินใจที่ให้ค่าความเสี่ยงต่ำที่สุด 11
รูปที่ 6 ระนาบที่สอดคล้องตามสมการ 13
รูปที่ 7 ผลที่เกิดขึ้นกับระนาบในปริภูมิของข้อมูลนำเข้า 15
รูปที่ 8 สถาปัตยกรรมการทำงานของซัพพอร์ตเวกเตอร์แมชชีน 15
รูปที่ 9 ลักษณะของเมตริกซ์ของเวกเตอร์เจาะจง 17
รูปที่ 10 เปรียบเทียบ PCA ธรรมดาที่เป็นลักษณะเชิงเส้นกับ Kernel PCA 20
รูปที่ 11 ตัวอย่างของนิวรอลเน็ตเวิร์ก 21
รูปที่ 12 แผนภาพขั้นตอนต่าง ๆ ที่ใช้ในการพัฒนาโปรแกรม 25
รูปที่ 13 หน้าจอหลักของโปรแกรมไทยโอซีอาร์ 49
รูปที่ 14 คำสั่งย่อยภายใต้รายการคำสั่ง File 50
รูปที่ 15 คำสั่งย่อยภายใต้รายการคำสั่ง Image 51
รูปที่ 16 คำสั่งย่อยภายใต้รายการคำสั่ง PCA 52
รูปที่ 17 คำสั่งย่อยภายใต้รายการคำสั่ง OCR 52
รูปที่ 18 หน้าจอ Threshold 54
รูปที่ 19 หน้าจอ Rotate 55
รูปที่ 20 หน้าจอ Recognition Result โดยที่ข้อความด้านบนแสดงผลลัพธ์ของการรู้จำ 56
รูปที่ 21 หน้าจอเลือกรูปภาพที่ต้องการรู้จำทีละหลายภาพเอกสาร 57
รูปที่ 22 หน้าจอกำหนดค่าที่เกี่ยวข้องในการรู้จำทีละหลายแฟ้มรูปภาพ 58

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันคอมพิวเตอร์เข้ามามีบทบาทกับมนุษย์ในชีวิตประจำวันมากมาย มนุษย์มีการนำคอมพิวเตอร์ไปช่วยในแก้ปัญหาในงานหลากหลายด้าน โดยเฉพาะในงานด้านเอกสารที่มีการนำคอมพิวเตอร์ไปใช้งานกันอย่างกว้างขวาง โดยงานทางด้านเอกสารที่มีการนำคอมพิวเตอร์เข้าไปเป็นส่วนร่วมที่สำคัญงานหนึ่งนั้นก็คือการนำคอมพิวเตอร์ไปใช้ในการแปลงตัวอักษรในเอกสารข้อมูลที่เป็นรูปภาพให้กลายเป็นข้อมูลในรูปแบบตัวอักษรที่แก้ไขได้จากโปรแกรมประมวลผลคำโดยตรง หรือที่เราเรียกโปรแกรมที่ทำหน้าที่นี้ว่าโปรแกรมโอซีอาร์ (Optical Character Recognition : OCR) นั่นเอง

โปรแกรมโอซีอาร์มีส่วนช่วยในงานทางด้านเอกสารอย่างมาก เพราะช่วยในการแปลงข้อมูลตัวอักษรของเอกสารที่อยู่ในลักษณะที่เป็นรูปภาพที่ผู้ใช้ไม่สามารถทำการแก้ไข ดัดแปลง หรือค้นหาคำโดยการใช้โปรแกรมการประมวลผลคำทั่วไปให้กลายเป็นข้อมูลตัวอักษรที่โปรแกรมประมวลผลคำสามารถนำไปแก้ไขได้ ทำให้ช่วยย่นระยะเวลาในการที่ผู้ใช้อาจจะต้องทำการพิมพ์ตัวอักษรในเอกสารนั้นขึ้นมาใหม่จากเอกสารที่เป็นรูปภาพที่มีอยู่ ช่วยให้ประหยัดเวลาในการทำงานทางด้านเอกสารอย่างสูง อีกทั้งยังช่วยให้ประหยัดเนื้อที่ในการเก็บข้อมูลเอกสารเหล่านั้นจากการที่ต้องเก็บข้อมูลในลักษณะรูปภาพเอกสารขนาดใหญ่ก็เปลี่ยนเป็นการเก็บข้อมูลตัวอักษรแต่ละตัวตามจริงแทน

อย่างไรก็ตามโปรแกรมโอซีอาร์ในปัจจุบันก็ยังมีข้อจำกัดอยู่บ้างในหลายประการ ทั้งในแง่ความถูกต้องในการแปลงตัวอักษรที่ยังไม่สามารถทำให้ถูกต้องทั้งหมดได้ รวมไปถึงความเร็วในการทำงานของโปรแกรม จึงมีงานวิจัยมากมายที่มุ่งแก้ไขข้อผิดพลาด และเสริมประสิทธิภาพของโปรแกรมโอซีอาร์ให้ดียิ่งขึ้น หนึ่งในจำนวนนั้นก็คือวิธีการของซัพพอร์ตเวกเตอร์แมชชีน (เอสวีเอ็ม) และเคอร์เนล ที่สามารถนำมาประยุกต์ใช้ในงานด้านโอซีอาร์ในส่วนที่เกี่ยวกับการแยกลักษณะสำคัญของรูปภาพตัวอักษรได้ โดยการนำหลักการของซัพพอร์ตเวกเตอร์แมชชีน และเคอร์เนลมาประยุกต์ใช้ในกับวิธีการวิเคราะห์องค์ประกอบสำคัญ (Principal Component Analysis : PCA) ของรูปภาพตัวอักษรที่อยู่ภายในกระบวนการแยกลักษณะสำคัญของรูปภาพตัวอักษร เพื่อดึงเอาข้อมูลค่าลักษณะสำคัญที่แตกต่างกันของแต่ละรูปภาพตัวอักษรนั้นออกมา แล้วส่งข้อมูลที่ได้นั้นไปให้ยังกระบวนการแยกแยะตัวอักษรให้ทำการพิจารณาข้อมูลค่าลักษณะสำคัญที่ได้แล้วแยกแยะว่ารูปภาพตัวอักษรนั้นควรจะเป็นตัวอักษรชนิดใดต่อไป

ในงานวิจัยนี้จะมุ่งเน้นที่จะปรับปรุงวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเดิมที่อยู่ภายในกระบวนการแยกลักษณะสำคัญของรูปภาพตัวอักษรในโปรแกรมโอซีอาร์ภาษาไทยของชาญฤทธิ์ [1] โดยนำทฤษฎีของของซัพพอร์ตเวกเตอร์แมชชีน (เอสวีเอ็ม) และเคอร์เนลเข้ามาประยุกต์ ได้เป็นวิธีใหม่ที่เรียกว่าการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล (Kernel principal component analysis) พร้อมทั้งทำการพัฒนาโปรแกรมโอซีอาร์ภาษาไทยที่ใช้เทคนิคใหม่นี้เพื่อให้ได้โปรแกรมโอซีอาร์ภาษาไทยที่มีประสิทธิภาพดียิ่งขึ้นต่อไป

1.2 งานวิจัย และทฤษฎีที่เกี่ยวข้อง

ชาญฤทธิ์ [1] ได้ทำการออกแบบ และพัฒนาโปรแกรมโอซีอาร์ภาษาไทย โดยนำเอาวิธีการต่าง ๆ ที่เกี่ยวข้องในกระบวนการโอซีอาร์มาทำงานเข้าไว้ร่วมกัน อันได้แก่ขั้นตอนการนำเอกสารเข้าสู่โปรแกรม ขั้นตอนการประมวลผลภาพ ขั้นตอนการตัดแยกบรรทัดจากรูปภาพ ขั้นตอนการตัดแยกตัวอักษรจากรูปภาพ ขั้นตอนการรู้จำตัวอักษร ขั้นตอนการแก้ไขผลลัพธ์ที่ได้จากขั้นตอนการรู้จำ ขั้นตอนการสร้างบรรทัด และขั้นตอนการแก้ไขคำผิด งานวิจัยนี้ได้แสดงให้เห็นภาพการทำงานที่เกิดขึ้นของขั้นตอนต่าง ๆ ภายในโปรแกรมโอซีอาร์ภาษาไทย แสดงให้เห็นถึงข้อมูลเข้า และผลลัพธ์ที่เกิดในแต่ละขั้นตอนการทำงาน ตลอดจนทฤษฎีที่เกี่ยวข้องในแต่ละขั้นตอนการทำงาน ไม่ว่าจะเป็นการปรับปรุงคุณภาพของรูปภาพเอกสาร (image processing) การตัดรูปภาพเอกสารใหญ่ให้เป็นรูปภาพของตัวอักษรแต่ละตัว การแยกลักษณะสำคัญของข้อมูลรูปภาพตัวอักษร (feature extraction) การแยกแยะตัวอักษร (classification) ตลอดจนการแก้ไขคำผิดหลังจากได้ลำดับของรหัสตัวอักษรออกมาจนได้เป็นข้อมูลลำดับของรหัสตัวอักษรที่มีความถูกต้องมากที่สุด ที่สามารถนำไปใช้ในการทำงานกับโปรแกรมประมวลผลคำทั่วไปได้ โดยในขั้นตอนของการแยกลักษณะสำคัญของข้อมูลรูปภาพตัวอักษรนั้นได้นำวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเชิงเส้นมาใช้ควบคู่กับการแปลงแบบเค-แอล (Karhunen-Loève Transform) ที่จะทำการแปลงข้อมูลภาพตัวอักษรให้กลายเป็นข้อมูลเวกเตอร์ส่งให้กับนิวรอลเน็ตเวิร์กนำไปแยกแยะชนิดตัวอักษรต่อไป

Schölkopf [2] เสนอแนวคิดการเรียนรู้ของเครื่อง (machine learning) ที่เรียกว่าซัพพอร์ตเวกเตอร์แมชชีน (เอสวีเอ็ม) และเคอร์เนล ที่สามารถนำไปประยุกต์เข้ากับวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเชิงเส้นเดิมได้เป็นวิธีการใหม่ที่เรียกว่า การวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล ที่มีความยืดหยุ่น และมีประสิทธิภาพมากขึ้นในการดึงค่าลักษณะสำคัญของข้อมูลที่สามารถนำไปใช้ในการแยกแยะแบ่งกลุ่มของสิ่งที่เราสนใจ ที่มีประโยชน์กับงานในหลาย ๆ ด้าน

ไม่ว่าจะเป็นการรู้จำรูปแบบ (pattern recognition) สิ่งของต่าง ๆ ตลอดจนนำมาประยุกต์ใช้ในการรู้จำตัวอักษรของโปรแกรมโอซีอาร์ภาษาไทย

1.3 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาโปรแกรมโอซีอาร์ภาษาไทย ที่นำเอาวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลมาใช้ในกระบวนการแยกลักษณะสำคัญของตัวอักษรเพื่อให้ได้โปรแกรมที่มีประสิทธิภาพและความถูกต้องดียิ่งขึ้น

1.4 ขอบเขตของการวิจัย

1. ทำการพัฒนาโปรแกรมโอซีอาร์ภาษาไทยที่นำเอาวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลมาใช้หาค่าลักษณะสำคัญของข้อมูลรูปภาพตัวอักษร
2. โปรแกรมจะทำงานภายใต้ระบบปฏิบัติการ Microsoft Windows รุ่น 98 ขึ้นไป ใช้หน่วยประมวลผลกลางตั้งแต่รุ่น Pentium II มีหน่วยความจำ 64 เมกะไบต์ขึ้นไป และมีเนื้อที่ฮาร์ดดิสก์เพียงพอต่อการทำงานของโปรแกรม
3. โปรแกรมสามารถทำงานทั่วไปที่ควรมีในโปรแกรมโอซีอาร์ได้ ตั้งแต่การรับข้อมูลรูปภาพที่เป็นเอกสาร ประมวลผลจนได้ข้อมูลตัวอักษรที่สามารถนำไปให้โปรแกรมประมวลผลคำทั่วไปแก้ไขได้
4. แฟ้มข้อมูลรูปภาพเอกสารที่ใช้ในการทดสอบต้องมีลักษณะดังนี้
   - แฟ้มข้อมูลรูปภาพเอกสารต้องเป็นสีขาว – ดำ จัดเก็บลักษณะบิตแมป (bitmap)
   - ตัวอักษรที่ปรากฏในเอกสารคือ ตัวอักษรพิมพ์ภาษาไทยและภาษาอังกฤษ ตัวเลขพิมพ์ไทยและอาราบิค แบบ AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC, EucrosiaUPC และ FreesiaUPC
   - ขนาดของตัวอักษรที่จะใช้ทดสอบคือขนาด 14, 16, 18, 20, 22, 24, 28 และ 36 โดยที่ขนาดของตัวอักษรในเอกสารจะต้องมีขนาดเท่ากันทั้งเอกสาร
   - รูปแบบของตัวอักษรจะต้องไม่มีรูปแบบเอียง หนา หรือขีดเส้นใต้
   - แฟ้มข้อมูลรูปภาพเอกสารต้องเป็นรูปภาพเอกสารที่มีเฉพาะตัวอักษรอยู่ภายใน (plain text document) ไม่มีรูปภาพ หรือสัญลักษณ์ (logo) อื่น ๆ ปะปน

5. การทดสอบเปรียบเทียบจะทำการเปรียบเทียบความถูกต้องของผลการแยกแยะตัวอักษระหว่างแบบที่ใช้วิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล และแบบเดิมที่เป็นลักษณะเชิงเส้น

1.5 ขั้นตอนการดำเนินการวิจัย

1. ศึกษาโปรแกรมโอซีอาร์ภาษาไทย และงานวิจัยที่เกี่ยวข้องกับการทำงานของโปรแกรมโอซีอาร์ภาษาไทย
2. ศึกษางานวิจัยที่เกี่ยวข้องกับส่วนการรู้จำตัวอักษรในโปรแกรมโอซีอาร์ อันได้แก่งานวิจัยที่เกี่ยวกับการวิเคราะห์องค์ประกอบสำคัญที่อยู่ภายในกระบวนการแยกลักษณะสำคัญของข้อมูลของโปรแกรมโอซีอาร์ และงานวิจัยที่เกี่ยวกับกระบวนการแยกแยะตัวอักษร
3. ศึกษาแนวคิด และทฤษฎีของซัพพอร์ตเวกเตอร์แมชชีน และการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล
4. ออกแบบ และพัฒนาส่วนการแยกลักษณะสำคัญของข้อมูลของโปรแกรมโอซีอาร์ภาษาไทยที่นำหลักการของซัพพอร์ตเวกเตอร์แมชชีน และเทคนิคการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลมาใช้ในการรู้จำตัวอักษร
5. ทำการทดสอบวิเคราะห์ผลของโปรแกรมโอซีอาร์ภาษาไทยที่ได้ด้วยวิธีการใหม่เปรียบเทียบกับของเดิม
6. สรุปผลการวิจัย ข้อเสนอแนะ และจัดทำรายงานวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

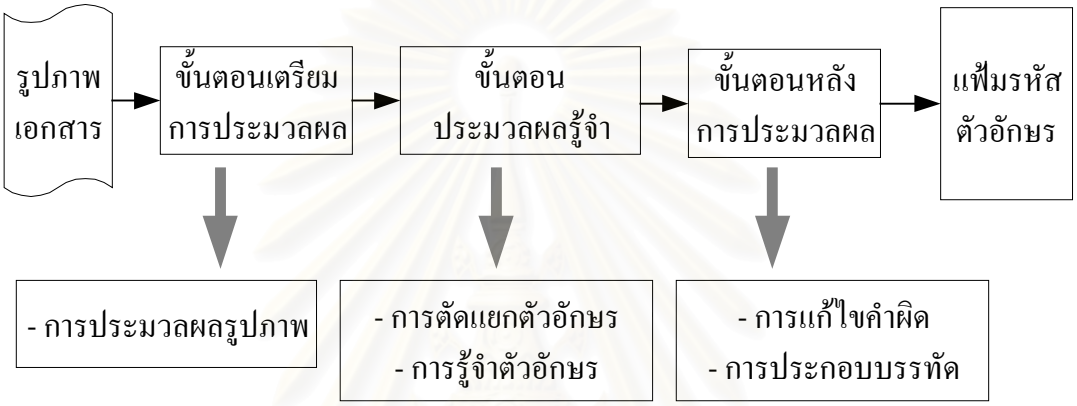
1. ได้โปรแกรมโอซีอาร์ภาษาไทยที่มีประสิทธิภาพ และความถูกต้องที่ดียิ่งขึ้น
2. สามารถนำโปรแกรมโอซีอาร์ภาษาไทยที่ได้ไปช่วยในงานด้านเอกสารได้จริง
3. สามารถนำวิธีการที่ได้จากวิทยานิพนธ์นี้ไปประยุกต์เข้ากับวิธีการใหม่อื่น ๆ เพื่อให้ได้โปรแกรมโอซีอาร์ภาษาไทยที่มีประสิทธิภาพมากยิ่งขึ้นได้

บทที่ 2

แนวคิด และทฤษฎีที่เกี่ยวข้อง

2.1 หลักการทำงานของโปรแกรมโอซีอาร์

ในการทำงานของโปรแกรมโอซีอาร์ เราสามารถแบ่งส่วนการทำงานได้เป็น 3 ส่วนหลัก ๆ ด้วยกัน (จุฬารัตน์ [3]) ดังรูปที่ 1



รูปที่ 1 ขั้นตอนการทำงานโดยหลักของโปรแกรมโอซีอาร์

โดยในแต่ละขั้นตอนมีการทำงาน ข้อมูลนำเข้า และข้อมูลส่งออกดังนี้

1. **ขั้นตอนเตรียมการประมวลผล** (Pre-Processing) ที่ขั้นตอนนี้รูปภาพเอกสารเริ่มแรกจะถูกทำการปรับปรุงลักษณะ และคุณสมบัติต่าง ๆ ของรูปภาพเอกสารโดยกระบวนการทางการประมวลผลรูปภาพ (image processing) เพื่อให้รูปภาพเอกสารที่เหมาะสมในการนำไปประมวลผลรู้จำต่อไป
   - ข้อมูลนำเข้า : รูปภาพของเอกสาร (document image) ที่ได้จากการสแกนหรือแฟ้มข้อมูลรูปภาพที่ได้ทำการสแกนเก็บไว้แล้ว
   - ข้อมูลส่งออก : รูปภาพของเอกสารที่เหมาะสมจะนำไปใช้ในขั้นตอนประมวลผลการรู้จำ
2. **ขั้นตอนประมวลผลการรู้จำ** (OCR Precessing) เป็นขั้นตอนหลักของโปรแกรมโอซีอาร์ ที่จะทำการรู้จำตัวอักษรภายในรูปภาพเอกสาร ประกอบด้วยการทำงานสองส่วนหลักด้วยกัน คือ

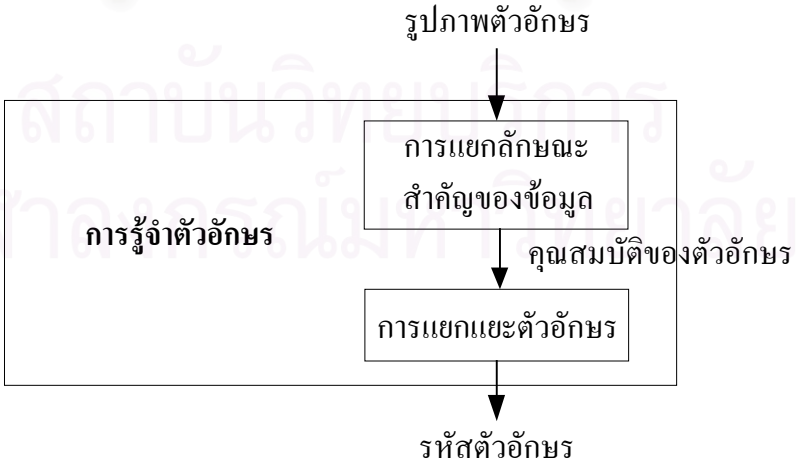
1. การตัดแยกตัวอักษร (Character Segmentation) ที่ขั้นตอนนี้รูปภาพเอกสารที่ได้จากขั้นตอนเตรียมการประมวลผลจะถูกนำมาหาส่วนที่เป็นตัวอักษร แล้วทำการตัดรูปภาพนั้นให้เป็นรูปภาพตัวอักษรแต่ละตัวออกมา
2. การรู้จำตัวอักษร (Character Recognition) ที่ขั้นตอนนี้จะนำรูปภาพตัวอักษรแต่ละตัวที่ได้จากขั้นตอนตัดแยกตัวอักษรมาทำการรู้จำได้เป็นรหัสตัวอักษร (character code) ที่สามารถนำไปใช้ในการโปรแกรมประมวลผลคำทั่วไป

- ข้อมูลนำเข้า : รูปภาพของเอกสารที่ได้จากขั้นตอนเตรียมประมวลผล
- ข้อมูลส่งออก : ลำดับของรหัสตัวอักษรที่ไดจากการรู้จำ

3. **ขั้นตอนหลังการประมวลผล (Post-Processing)** เป็นขั้นตอนที่ช่วยในการปรับปรุงผลลัพธ์ที่ได้จากขั้นตอนประมวลผลรู้จำตัวอักษร โดยทำการแก้คำที่สะกดผิด (spelling corrector) เพื่อให้ข้อมูลลำดับของรหัสตัวอักษรมีความถูกต้องมากยิ่งขึ้น

- ข้อมูลนำเข้า : ลำดับของรหัสตัวอักษร หรือแฟ้มข้อมูลที่เก็บลำดับของรหัสตัวอักษรที่ได้จากขั้นตอนประมวลผลรู้จำ
- ข้อมูลส่งออก : แฟ้มข้อมูลที่เก็บลำดับของรหัสตัวอักษรที่ได้รับการแก้ไขคำที่สะกดให้ถูกต้องแล้ว

งานวิจัยนี้มุ่งเน้นที่จะปรับปรุงการทำงานของโปรแกรมโอซีอาร์ในส่วนของการรู้จำตัวอักษรที่อยู่ในขั้นตอนประมวลผลรู้จำ โดยมีขั้นตอนการทำงานที่เกี่ยวข้องอยู่ด้วยกันสองส่วน (Casey and Lecolinet [4]) ดังรูปที่ 2



รูปที่ 2 แบบจำลองขั้นตอนการรู้จำตัวอักษร

จากรูปที่ 2 ข้อมูลที่เข้ามาในขั้นตอนการรู้จำตัวอักษรก็คือรูปภาพตัวอักษร (character image) ที่ได้รับการตัดจากรูปภาพเอกสารดั้งเดิมมาจากขั้นตอนการตัดแยกตัวอักษร และจะผ่านขั้นตอนย่อยอีก 2 ขั้นตอน นั่นคือ

1. การแยกลักษณะสำคัญของข้อมูล (Feature Extraction) เป็นขั้นตอนที่จะทำการแยกลักษณะสำคัญของรูปภาพตัวอักษร เพื่อหาคุณสมบัติเฉพาะของตัวอักษร (character properties) ในรูปภาพนั้น
2. การแยกแยะตัวอักษร (Classification) ที่ขั้นตอนนี้ คุณสมบัติเฉพาะของตัวอักษรที่ได้จะถูกนำมาพิจารณา เพื่อทำการแยกแยะว่ารูปภาพตัวอักษรนั้นเป็นตัวอักษรตัวใด และให้ผลลัพธ์ออกมาเป็นรหัสของตัวอักษร (character code) ตัวนั้นออกมา

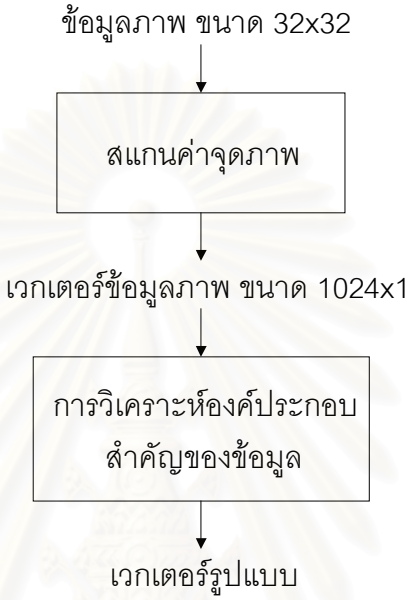
2.2 ข้อมูลรูปภาพตัวอักษร

ข้อมูลรูปภาพตัวอักษรในงานวิจัยฉบับนี้นำแนวคิด และทฤษฎีจากงานวิจัยของชาญฤทธิ์ [1] โดยใช้รูปภาพเอกสารที่มีจำนวนระดับความเข้มของภาพ 2 ระดับ (binary image) คือสีขาวและสีดำ และจะทำการแทนค่าของจุดภาพแต่ละสีด้วยค่าตัวเลข คือแทนค่าของจุดสีดำด้วย 0 และแทนค่าของจุดสีขาวด้วย 1

ค่าของจุดภาพที่ได้จะถูกนำไปใช้ประมวลผลในขั้นตอนเตรียมการประมวลผลของโปรแกรมโอซีอาร์ ไปจนถึงขั้นตอนการตัดแยกตัวอักษรเพื่อให้ได้รูปภาพตัวอักษรแต่ละตัว หลังจากนั้นรูปภาพตัวอักษรที่ได้จากขั้นตอนการตัดแยกตัวอักษร จะถูกเปลี่ยนขนาดให้อยู่ในขนาด 32x32 เท่ากันทุกภาพตัวอักษร เพื่อนำไปใช้ในขั้นตอนการแยกลักษณะสำคัญของข้อมูลรูปภาพตัวอักษร (feature extraction) ต่อไป

โดยที่ในขั้นตอนการแยกลักษณะสำคัญของข้อมูลตัวอักษร รูปภาพตัวอักษรจะถูกนำมาสแกนตั้งแต่จุดภาพมุมซ้ายบนของภาพ ไล่สแกนจากซ้ายไปขวา และบนลงล่างจนถึงจุดภาพมุมขวาล่าง ดังรูปที่ 3 โดยจะทำการแทนค่าของข้อมูลในแต่ละจุดภาพใหม่ คือให้ค่าเป็น 1 หากจุดภาพนั้นเป็นสีขาว และให้ค่าเป็น –1 หากจุดภาพนั้นเป็นสีดำ (เปลี่ยนจากค่า 0 ที่ใช้ในขั้นตอนการประมวลผลภาพในตอนแรก) ได้เป็นค่าเวกเตอร์ขนาด 1024x1 ออกมา (ซึ่งอาจมองได้เป็นเมตริกซ์ขนาด 1024x1) เพื่อนำไปใช้ในการประมวลผลด้านการวิเคราะห์องค์ประกอบสำคัญ (principal component analysis) ต่อไปตามรูปที่ 4

รูปที่ 3 ลักษณะการสแกนเพื่อแปลงจุดภาพเป็นค่าเวกเตอร์



รูปที่ 4 แผนภาพแสดงการเปลี่ยนแปลงของข้อมูลรูปภาพตัวอักษร

2.3 ซัพพอร์ตเวกเตอร์แมชชีน และเคอร์เนล

ซัพพอร์ตเวกเตอร์แมชชีน หรือเอสวีเอ็ม เป็นวิธีการเรียนรู้เชิงสถิติที่คิดค้นโดย Vapnik [5] สามารถใช้ในการเรียนรู้เพื่อแยกกลุ่มของข้อมูลสองกลุ่มออกจากกัน มีประโยชน์อย่างมากในการนำไปใช้แยกแยะข้อมูลที่เราสนใจ หรือการรู้จำรูปแบบของข้อมูล เช่นการแยกแยะว่าใช่หรือไม่ใช่วัตถุที่เราสนใจ เป็นต้น โดยเอสวีเอ็มประกอบด้วยแนวคิดพื้นฐานที่สำคัญ 3 เรื่องคือ

1. การลดความเสี่ยงเชิงโครงสร้างให้ต่ำสุด (Structural Risk Minimization) เป็นแนวคิดที่แสดงถึงขอบเขตของความเสี่ยง หรือความน่าจะเป็นที่จะเกิดความผิดพลาดของการเรียนรู้ มีประโยชน์ที่ช่วยในการหาฟังก์ชันตัดสินใจในกระบวนการเรียนรู้ที่มีความผิดพลาดน้อยที่สุดได้
2. ปริภูมิแสดงลักษณะสำคัญ กับเคอร์เนล (Feature Space and Kernel) เป็นแนวคิดที่มีความสำคัญทั้งกับเอสวีเอ็ม และการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล ที่จะทำการแมปข้อมูลจากปริภูมินำเข้า (input space) ไปสู่ปริภูมิแสดงลักษณะ

สำคัญ (feature space) เพื่อประโยชน์ในการสร้างฟังก์ชันตัดสินใจที่มีลักษณะไม่เชิงเส้นกับข้อมูลในปริภูมินำเข้า ทำให้การแบ่งกลุ่มข้อมูลมีความยืดหยุ่นมากขึ้น

3. ระนาบหลายมิติที่ใช้แยกข้อมูลได้ดีที่สุด (Optimal Margin Hyperplane) เป็นแนวคิดหลักของเอสวีเอ็ม ที่จะทำการหาระนาบที่มีระยะห่างจากข้อมูลที่ทำการแบ่งมากที่สุดในการแบ่งข้อมูลสองกลุ่มออกจากกัน

2.3.1 **การลดความเสี่ยงเชิงโครงสร้างให้ต่ำสุด** (Structural Risk Minimization)

การลดความเสี่ยงเชิงโครงสร้างให้ต่ำสุด (Structural Risk Minimization) หรือ เอสอาร์เอ็ม (SRM) เป็นวิธีการที่ช่วยในการหาฟังก์ชันตัดสินใจในกระบวนการเรียนรู้ที่ให้ความผิดพลาดน้อยที่สุด โดยการประมาณหาค่าความผิดพลาดที่ต่ำที่สุดด้วยการพิจารณาจากข้อมูลในชุดทดลอง

โดยในการแบ่งข้อมูลสองกลุ่มใด ๆ ออกจากกัน เรากำหนดให้เซตของฟังก์ชันตัดสินใจที่ใช้ในการเรียนรู้ f_α คือ

$$\{f_\alpha : \alpha \in \Lambda\}, f_\alpha : R^N \rightarrow \{\pm 1\}$$

ในกรณีนี้เราแทนผลลัพธ์ของข้อมูลในแต่ละกลุ่มด้วย 1 และ –1 และกำหนดข้อมูลตัวอย่างที่จะใช้ทำการเรียนรู้ (training data) จำนวน ℓ ตัว อยู่ในลักษณะคู่ลำดับประกอบด้วยเวกเตอร์ $\mathbf{x}$ และชื่อกลุ่มที่ใช้เรียนรู้ y กระจายตัวอยู่ด้วยความน่าจะเป็น $P(\mathbf{x},y)$ ดังนี้

$$(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_\ell,y_\ell) \in R^N \times \{\pm 1\}$$

เราต้องการที่จะหาฟังก์ชัน f_α ที่ให้ผลต่างของค่าความผิดพลาดระหว่างค่าที่เรียนรู้ได้กับค่าผลลัพธ์จริง หรือที่เรียกว่าค่าความเสี่ยง (risk) นี้มีค่าน้อยที่สุด ซึ่งค่าความเสี่ยงนี้สามารถหาได้ตามสมการที่ (1)

$$R(\alpha) = \int \frac{1}{2} \left| f_\alpha(\mathbf{x}) - y \right| d\,P(\mathbf{x}, y) \qquad (1)$$

แต่เนื่องมาจากค่าการกระจาย $P(\mathbf{x},y)$ เป็นค่าที่เราไม่ทราบค่า ทำให้ไม่สามารถหาค่า $R(\alpha)$ ได้ อย่างไรก็ตามเราสามารถประมาณค่าของ $R(\alpha)$ ด้วยหลักการอุปนัยทางคณิตศาสตร์เข้าช่วย โดยการหาค่าความเสี่ยงเชิงทดลอง (empirical risk) ที่มีค่าน้อยที่สุดแทนดังสมการที่ (2)

$$R_{emp}(\alpha) = \frac{1}{\ell} \sum_{i=1}^{\ell} \frac{1}{2} \left| f_\alpha(\mathbf{x}_i) - y_i \right| \qquad (2)$$

โดยค่าความเสี่ยงเชิงทดลอง $R_{emp}(\alpha)$ นี้จะสัมพันธ์กับค่าความเสี่ยง $R(\alpha)$ ตามสมการที่ (3) (สำหรับค่า $\alpha \in \Lambda$ ใด ๆ และค่า $\ell > h$ ด้วยความน่าจะเป็น 1-η)

$$R(\alpha) \le R_{emp}(\alpha) + \varphi\left(\frac{h}{\ell}, \frac{\log(\eta)}{\ell}\right) \tag{3}$$

โดยที่ค่าความเชื่อมั่น (confidence term) ϕ ในสมการที่ (3) สามารถหาได้จาก

$$\varphi\left(\frac{h}{\ell}, \frac{\log(\eta)}{\ell}\right) = \sqrt{\frac{h\left(\log\frac{2\ell}{h}+1\right) - \log(\eta/4)}{\ell}} \tag{4}$$

เราเรียกค่า h ในสมการที่ (4) นี้เรียกว่าค่ามิติ VC (Vapnik-Chervonenkis dimension) ของเซตของฟังก์ชันตัดสินใจ ซึ่งเป็นค่าที่แสดงถึงจำนวนข้อมูลสูงสุดที่เซตของฟังก์ชันตัดสินใจนั้นสามารถแบ่งแยกได้ถูกต้อง

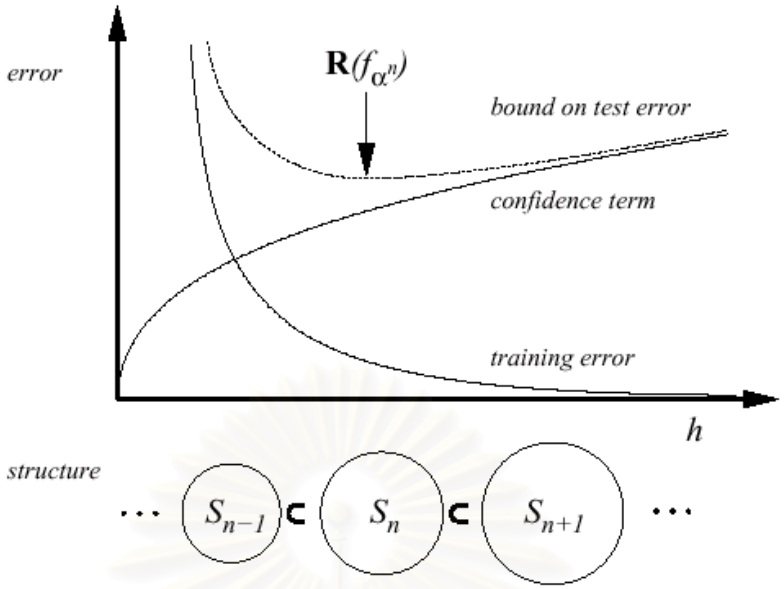
จากสมการที่ (3) เพื่อที่จะให้ค่าความผิดพลาดหรือค่าความเสี่ยง $R(\alpha)$ มีค่าน้อยที่สุด เราจะทำการควบคุมค่า $R_{emp}(\alpha)$ และค่า h ($\{f_\alpha : \alpha \in \Lambda'\}$), $\Lambda' \subset \Lambda$ เพื่อให้ค่าผลบวกทางฝั่งขวาของสมการที่ (3) ซึ่งเป็นขอบเขตบนของค่าความเสี่ยง $R(\alpha)$ มีค่าน้อยที่สุด และได้ค่า α ที่เป็นของฟังก์ชันที่ให้ค่าความเสี่ยงต่ำที่สุดออกมา และเพื่อที่จะทำการควบคุมค่า h นี้เราจึงทำการสร้างซับเซตของฟังก์ชันตัดสินใจ $S_n := \{f_\alpha : \alpha \in \Lambda'\} \subset \{f_\alpha : \alpha \in \Lambda\}$ เป็นซับเซตซ้อนกันเรื่อยไปดังนี้

$$S_1 \subset S_2 \subset \ldots \subset S_n \subset \ldots$$

ซึ่งโดยวิธีนี้จะทำให้ได้ค่า h ของแต่ละซับเซตมีค่าลดหลั่นกันไป ที่เราอาจมองได้ว่ายิ่งเซตใดที่มีจำนวนฟังก์ชันการตัดสินใจสูง (เครื่องมือการเรียนรู้ที่ซับซ้อน) ย่อมสามารถทำการแบ่งจำนวนกลุ่มข้อมูลได้สูง หรือมีค่า h ที่สูงนั่นเอง โดยค่า h ในแต่ละซับเซตจะได้ค่าไล่เรียงกันไปดังนี้

$$h_1 \le h_2 \le \ldots \le h_n \le \ldots$$

โดยท้ายที่สุดแล้วเอสอาร์เอ็มก็จะทำการเลือกฟังก์ชันการตัดสินใจ f_α ในซับเซต $\{f_\alpha : \alpha \in \Lambda\}$ ที่ทำให้ผลบวกระหว่างค่าความเสี่ยงเชิงทดสอบ $R_{emp}(\alpha)$ และค่าความเชื่อมั่นมีค่าน้อยที่สุด ที่ทำให้มั่นใจได้ว่าค่าความเสี่ยง $R(\alpha)$ จะมีค่าต่ำที่สุดตามต้องการ ดังรูปที่ 5



รูปที่ 5 ตำแหน่งการเลือกค่าฟังก์ชันการตัดสินใจที่ให้ค่าความเสี่ยงต่ำที่สุด

2.3.2 **ปริภูมิแสดงลักษณะสำคัญ กับเคอร์เนล** (Feature Space and Kernel)

ปริภูมิแสดงลักษณะสำคัญ (feature space) เป็นชื่อที่ใช้แทนปริภูมิที่มีมิติที่สูงกว่าปริภูมิของข้อมูลนำเข้า (input space) ที่เข้ามา ยกตัวอย่างเช่นในกรณีของข้อมูลเวกเตอร์ $\mathbf{x} \in R^2$ ที่เป็นปริภูมินำเข้า เราสามารถทำการแมปไปสู่ปริภูมิแสดงลักษณะสำคัญที่เกิดจากผลคูณพหุนามของทุกสมาชิกในเวกเตอร์ $\mathbf{x}$ ได้ดังนี้คือ (ให้ Φ แทนฟังก์ชันการแมป)

$$\Phi : R^2 \rightarrow R^4$$

$$(x_1, x_2) \rightarrow (x_1^2, x_2^2, x_1x_2, x_2x_1)$$

ในกรณีนี้ทำให้ได้ว่า R^4 เป็นปริภูมิแสดงลักษณะสำคัญ ส่วนเคอร์เนล (Kernel) นั้นเป็นเทอมที่ใช้แทนค่าผลคูณภายใน (dot product) ของข้อมูลบนปริภูมิแสดงลักษณะสำคัญดังสมการ

$$\mathbf{k(x,y) = (\Phi(x) \bullet \Phi(y))} \qquad (5)$$

ซึ่งเราสามารถนำเคอร์เนลมาใช้ประโยชน์ในการคำนวณหาค่าผลคูณภายในของข้อมูลบนปริภูมิแสดงลักษณะสำคัญโดยที่ไม่จำเป็นต้องทำการคำนวณค่าแมปฟังก์ชัน Φ ของข้อมูลนำเข้าแต่ละตัวก่อน ซึ่งจะค่อนข้างยุ่งยาก และเสียเวลาในการคำนวณสูง โดยการแทนค่าผลคูณภายในของข้อมูลบนปริภูมิแสดงลักษณะสำคัญจากการคำนวณค่าผลคูณภายในบนปริภูมินำเข้าได้โดยตรง ยกตัวอย่างเช่นในกรณีของฟังก์ชันแบบพหุนาม

ให้ $\Phi : R^2 \rightarrow R^4$ แทนฟังก์ชันการแมปแบบพหุนาม

$\Phi : (x_1, x_2) \rightarrow (x_1^2, x_2^2, x_1x_2, x_2x_1)$

$\Phi : (y_1, y_2) \rightarrow (y_1^2, y_2^2, y_1y_2, y_2y_1)$

เราสามารถหาค่าผลคูณภายในบนปริภูมิแสดงลักษณะสำคัญได้ดังนี้

$$\begin{aligned} k(\mathbf{x},\mathbf{y}) = (\Phi(\mathbf{x}) \bullet \Phi(\mathbf{y})) &= (\Phi(x_1, x_2) \bullet \Phi(y_1, y_2)) \\ &= (x_1^2, x_2^2, x_1x_2, x_2x_1) \bullet (y_1^2, y_2^2, y_1y_2, y_2y_1) \\ &= (x_1^2 y_1^2 + x_2^2 y_2^2 + x_1x_2 y_1y_2 + x_2x_1y_2y_1) \\ &= (x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1x_2 y_1y_2) \\ &= (x_1 y_1 + x_2 y_2)^2 \\ &= ((x_1, x_2) \bullet (y_1, y_2))^2 \\ &= (\mathbf{x} \bullet \mathbf{y})^2 \end{aligned}$$

เราสามารถนำวิธีการในลักษณะคล้ายกันนี้ไปประยุกต์เข้ากับฟังก์ชันแบบอื่น ๆ ที่ช่วยให้เราสามารถเขียนแทนผลคูณภายในในปริภูมิแสดงลักษณะสำคัญได้ด้วยเทอมของผลคูณภายในในปริภูมิข้อมูลนำเข้า ทำให้สะดวกต่อการคำนวณอย่างมาก

ตัวอย่างของเคอร์เนลที่ใช้กันแพร่หลาย รวมถึงนำมาใช้ในงานวิจัยฉบับนี้ได้แก่

Polynomial classifiers of degree d : $k(\mathbf{x},\mathbf{y}) = (\mathbf{x} \bullet \mathbf{y})^d$

Radial Basis Function (RBF) classifier: $k(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/c)$

Neural networks : $k(\mathbf{x},\mathbf{y}) = \tanh(\kappa \bullet (\mathbf{x} \bullet \mathbf{y}) + \Theta)$

2.3.3 **ระนาบหลายมิติที่ใช้แยกข้อมูลได้ดีที่สุด** (Optimal Margin Hyperplane)

หลักการทำงานของเอสวีเอ็มจะมีพื้นฐานจากการหาระนาบเชิงเส้นที่ใช้ในการแบ่งข้อมูลออกจากกัน โดยเราจะพยายามหาระนาบที่มีระยะห่างจากข้อมูลที่ทำการแบ่งให้มากที่สุดเรียกว่าเป็นระนาบหลายมิติที่ใช้แยกข้อมูลได้ดีที่สุด (optimal margin hyperplane) และทำการหาฟังก์ชันตัดสินใจเพื่อใช้ในการแบ่งกลุ่มที่สอดคล้องกับระนาบนั้น

โดยการหาระนาบหลายมิติที่ใช้แยกข้อมูลได้ดีที่สุดนั้นจะเริ่มจากกำหนดค่าข้อมูลฝึกหัด (training data) $\mathbf{z}$ เพื่อใช้ในการเรียนรู้ เป็นข้อมูลในลักษณะเวกเตอร์จำนวน ℓ ตัวโดยที่ $((\mathbf{z}_1,y_1),\ldots,(\mathbf{z}_\ell,y_\ell) \in R^N \times \{\pm 1\})$ เราจะสามารถสร้างระนาบเชิงเส้นในปริภูมิเวกเตอร์ได้ดังสมการ

$$\{\mathbf{z} \in R^n : (\mathbf{w} \bullet \mathbf{z}) + b = 0\} \qquad (6)$$

โดยที่ค่าฟังก์ชันตัดสินใจที่ใช้ในการแบ่งกลุ่มจากระนาบที่สร้างขึ้นจะอยู่ในสมการ

$$f_{\mathbf{w},b} = \text{sgn}((\mathbf{w} \bullet \mathbf{z})+b) \tag{7}$$

โดยที่เราต้องพยายามให้ค่าผลลัพธ์ของฟังก์ชันตัดสินใจมีค่าตรงกับค่าฝึกหัดที่ให้ไปมากที่สุด นั่นคือพยายามทำให้

$$f_{\mathbf{w},b}(\mathbf{z}_i) = y_i \ , \quad i = 1,\ldots, \ell \tag{8}$$

จากสมการที่ (6) เพื่อที่ทำให้ระยะระหว่างข้อมูลสองกลุ่มที่ถูกแบ่งด้วยระนาบมีค่ามากที่สุด เราจะทำการกำหนดสมการให้ค่า $\mathbf{w}$ และ b มีค่าสอดคล้องตามสมการ

$$\mathbf{w} \bullet \mathbf{z}_i+b \geq 1 \tag{9}$$

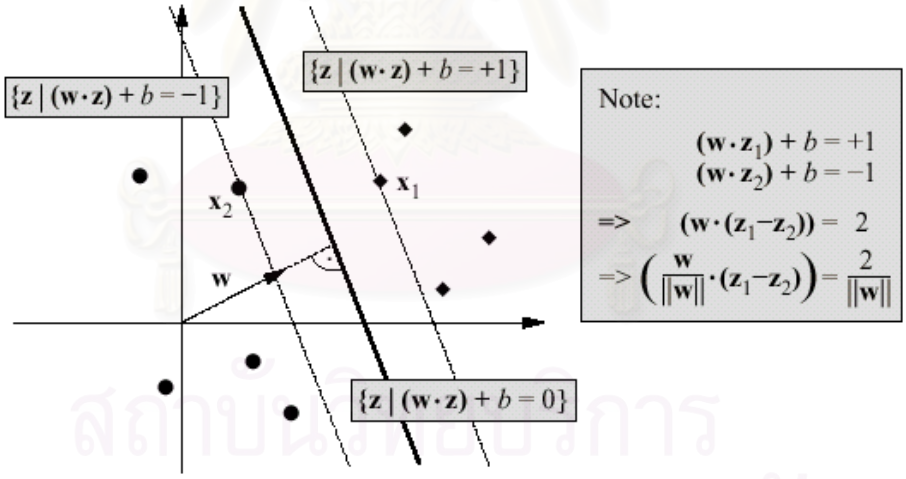
และ $$\mathbf{w} \bullet \mathbf{z}_i+b \leq -1 \tag{10}$$

หรือนั่นคือ

$$y_i \bullet ((\mathbf{w} \bullet \mathbf{z}_i)+b)-1 \geq 0 \tag{11}$$

เมื่อ i มีค่าตั้งแต่ 1 ถึง ℓ

ซึ่งสามารถแสดงระนาบที่เกิดขึ้นได้ดังรูปที่ 6



รูปที่ 6 ระนาบที่สอดคล้องตามสมการ

เพื่อให้ระยะห่างระหว่างระนาบกับข้อมูล ($2/\|\mathbf{w}\|$) มีความกว้างมากที่สุด ดังนั้นเราจึงต้องทำให้ค่า $\|\mathbf{w}\|$ มีค่าน้อยที่สุด โดยเราจะนำหลักการแก้ปัญหาค่าสุดขีดของลากรานจ์ (Lagrangian optimization theory) มาใช้ในการแก้ปัญหานี้ ดังสมการที่ (12)

$$L(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i (y_i((\mathbf{z}_i \cdot \mathbf{w})+b)-1) \tag{12}$$

โดยที่ α_i คือตัวคูณลากรานจ์ (Lagrangian multipiers)

ซึ่งหลังจากที่ทำการแก้ปัญหาสมการลากรานจ์แล้วเราจะได้ผลลัพธ์ดังสมการ

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0 \tag{13}$$

และ

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{z}_i \tag{14}$$

โดยเราสามารถแปลงสมการ (10) เข้ากับตัวคูณลากรานจ์ ได้เป็นสมการ

$$\alpha_i \bullet [y_i \bullet ((\mathbf{w} \bullet \mathbf{z}_i)+b)\text{-}1] = 0 \quad , \qquad i = 1,\ldots, \ell \tag{15}$$

โดยที่เราจะเรียกเวกเตอร์ $\mathbf{z}_i$ สำหรับทุก α_i ที่มากกว่า 0 ($\alpha_i > 0$) นี้ว่าซัพพอร์ตเวกเตอร์ (support vectors)

และจากสมการที่ (14) เมื่อนำไปแทนในสมการที่ (7) เราก็จะได้ค่าฟังก์ชันการตัดสินใจที่ใช้ในการแบ่งกลุ่มข้อมูลตามค่าซัพพอร์ตเวกเตอร์ที่ได้จากสมการที่ (15) ดังสมการที่ (16)

$$\mathrm{f}(\mathbf{z}) = \mathrm{sgn}\left(\sum_{i=1}^{\ell} \alpha_i y_i (\mathbf{z} \cdot \mathbf{z}_i) + b \right) \tag{16}$$

โดยเราสามารถที่จะประยุกต์ฟังก์ชันตัดสินใจที่ได้เข้ากับหลักการของเคอร์เนลเพื่อใช้ในการระนาบหลายมิติที่ใช้แยกข้อมูลได้ดีที่สุดในปริภูมิแสดงลักษณะสำคัญได้ดังสมการที่ (17)

$$\mathrm{f}(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{\ell} \alpha_i y_i \cdot \mathrm{k}(\mathbf{x} \cdot \mathbf{x}_i) + b \right) \tag{17}$$

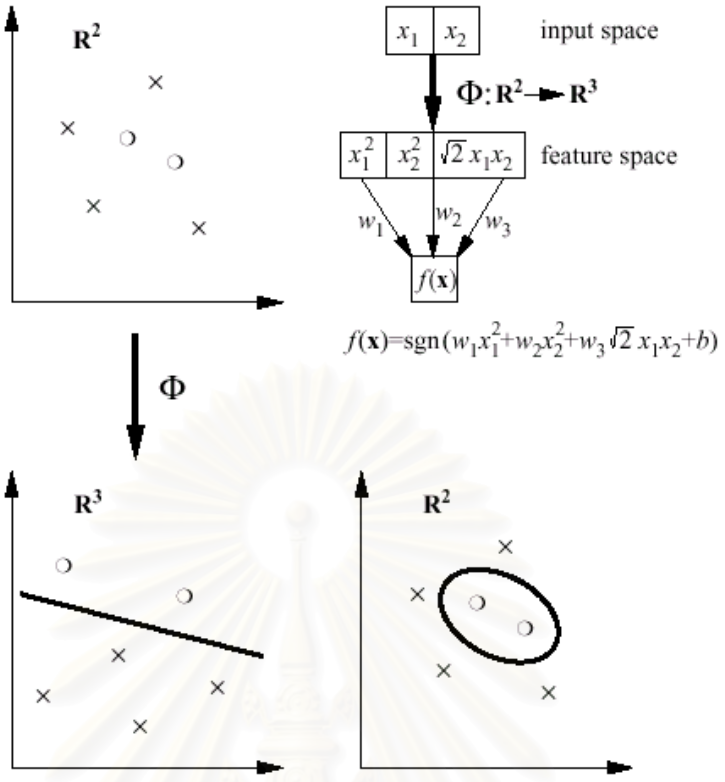
โดยที่

$$(\Phi(\mathbf{x}) \bullet \Phi(\mathbf{x}_i)) = \mathrm{k}(\mathbf{x},\mathbf{x}_i)$$

และ $$\Phi : \mathbf{x}_i \rightarrow \mathbf{z}_i$$

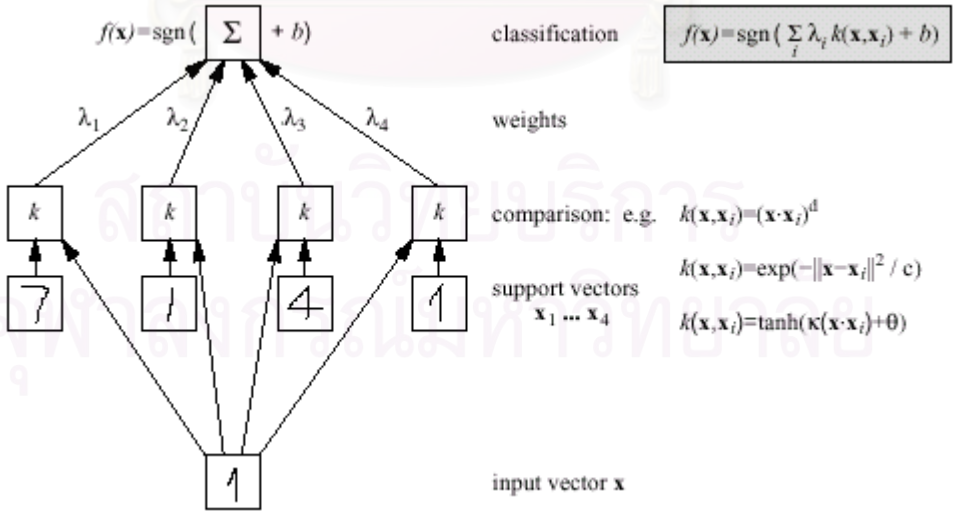
ด้วยวิธีการนี้ระนาบเชิงเส้นที่เราหาได้ในปริภูมิแสดงลักษณะสำคัญ จะกลายเป็นลักษณะระนาบไม่เชิงเส้นในปริภูมิของข้อมูลนำเข้า ทำให้เกิดความยืดหยุ่นในการแบ่งกลุ่มในปริภูมิของข้อมูลนำเข้า ดังรูปที่ 7

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 7 ผลที่เกิดขึ้นกับระนาบในปริภูมิของข้อมูลนำเข้า
เมื่อทำสร้างระนาบเชิงเส้นในปริภูมิแสดงลักษณะสำคัญ

เราสามารถสรุปการทำงานของซัพพอร์ตเวกเตอร์แมชชีนให้เห็นได้ดังรูปที่ 8



รูปที่ 8 สถาปัตยกรรมการทำงานของซัพพอร์ตเวกเตอร์แมชชีน

2.4 การวิเคราะห์องค์ประกอบสำคัญของข้อมูล

2.4.1 การวิเคราะห์องค์ประกอบสำคัญ (Principal Component Analysis)

การวิเคราะห์องค์ประกอบสำคัญเป็นวิธีทางสถิติที่สำคัญ และมีประโยชน์อย่างมากในการที่จะดึงลักษณะสำคัญ หรือคุณสมบัติจากข้อมูลที่เราสนใจ ที่มีการนำมาใช้อย่างมากในด้านการรู้จำรูปแบบต่าง ๆ โดยเฉพาะในขั้นตอนการแยกองค์ประกอบสำคัญจากรูปภาพตัวอักษรในโปรแกรมโอซีอาร์

การวิเคราะห์องค์ประกอบสำคัญนั้นอยู่บนหลักการแปลงเชิงตั้งฉากของพิกัดข้อมูล (orthogonal transformation) โดยค่าเส้นพิกัดใหม่ที่ได้นั้นจะถูกเรียกว่าองค์ประกอบสำคัญ (principal components)

ประโยชน์ขององค์ประกอบสำคัญที่ได้คือช่วยทำให้เราสามารถอธิบายข้อมูลที่อาจจะอยู่ในกลุ่มของข้อมูลที่มีมิติจำนวนมาก (multi-dimensional data) ให้มีขนาดที่เล็กลง โดยที่ข้อมูลที่ได้ยังสามารถที่อธิบายข้อมูลสำคัญที่เกี่ยวข้องภายในได้ไม่ต่างจากเดิม อีกทั้งยังช่วยให้ง่ายต่อการคำนวณทางคณิตศาสตร์มากขึ้น

Karhunen [6] ได้เสนอวิธีการหาค่าองค์ประกอบสำคัญของข้อมูลที่นิยมใช้กันโดยแพร่หลายที่เรียกว่าการแปลงแบบเค-แอล (Karhunen-Loève Transform : KLT) ที่ทำการหาค่าองค์ประกอบสำคัญของข้อมูลจากการแก้ปัญหาเวกเตอร์เจาะจง (Eigenvector) ของเมตริกซ์โคแวเรียนซ์ (Covariance matrix) ที่ได้จากข้อมูลที่เราสนใจ ซึ่งมีวิธีการดังต่อไปนี้

เริ่มจากการนำข้อมูลรูปภาพตัวอักษรตัวอย่างที่ใช้ในการเรียนรู้มาผ่านการแปลงเป็นข้อมูลเวกเตอร์ขนาด 1024x1 จากนั้นเวกเตอร์ของรูปภาพตัวอักษรทั้งหมดจะถูกนำมาหาค่าเมตริกซ์ค่าเฉลี่ย ดังสมการที่ (18)

$$\mathbf{m}_{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i \tag{18}$$

โดยที่ $\mathbf{m}_{\mathbf{x}}$ คือเมตริกซ์ค่าเฉลี่ยของเวกเตอร์รูปภาพตัวอักษรตัวอย่าง (ขนาด 1024x1)

M คือจำนวนเวกเตอร์ของรูปภาพตัวอักษรทั้งหมดที่ใช้เป็นตัวอย่างในขณะเรียนรู้

$\mathbf{x}_i$ คือเวกเตอร์รูปภาพตัวอักษรขนาด 1024x1

เมื่อหาเมตริกซ์ค่าเฉลี่ยได้แล้ว เราจะนำมาทำการหาค่าเมตริกซ์โคแวเรียนซ์ (Covariance matrix) ที่แสดงถึงความแปรผันของข้อมูลทั้งหมดโดยรวมว่าเป็นอย่างไรได้ ดังสมการที่ (19)

$$\mathbf{C_x} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} - \mathbf{m_x}\mathbf{m_x}^{\mathrm{T}} \tag{19}$$

โดยที่ $\mathbf{C_x}$ คือเมตริกซ์โคแวเรียนซ์ (ขนาด 1024x1024)

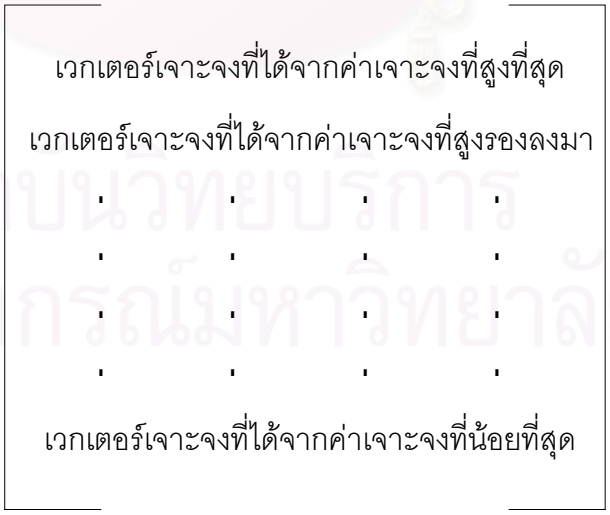
$\mathbf{x}_i^{\mathrm{T}}$ คือทรานโพสเมตริกซ์ของ $\mathbf{x}_i$

$\mathbf{m_x}^{\mathrm{T}}$ คือทรานโพสเมตริกซ์ของ $\mathbf{m_x}$

จากนั้นเราจะนำเมตริกซ์โคแวเรียนซ์ที่ได้ ไปเข้าสมการค่าเจาะจง (Eigenvalue equation) เพื่อทำการหาค่าเจาะจง (Eigen value) λ และเวกเตอร์เจาะจง (Eigen vector) $\mathbf{v}$ ตามค่าเจาะจงที่ได้ ดังสมการ

$$\lambda\mathbf{v} = \mathbf{C}\mathbf{v} \tag{20}$$

ซึ่งภายหลังจากที่ได้เวกเตอร์เจาะจงแล้ว เราจะนำเวกเตอร์เจาะจงที่ได้มาเรียงค่าเป็นเมตริกซ์ของเวกเตอร์เจาะจงตามลำดับของค่าเจาะจงจากมากไปน้อย ดังแสดงในรูปที่ 9 แล้วนำเมตริกซ์ที่ได้นี้มาใช้ในการหาเวกเตอร์รูปแบบของรูปภาพตัวอักษรใด ๆ ที่แสดงถึงค่าลักษณะสำคัญของรูปภาพตัวอักษรนั้น ๆ ได้ตามสมการที่ (21)



รูปที่ 9 ลักษณะของเมตริกซ์ของเวกเตอร์เจาะจง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$$\mathbf{y} = \mathbf{A}\,(\mathbf{x} - \mathbf{m_x}) \qquad (21)$$

โดยที่ $\mathbf{y}$ คือเวกเตอร์รูปแบบ (ขนาด 1024x1)

$\mathbf{A}$ คือเมตริกซ์ของเวกเตอร์เจาะจง (ขนาด 1024x1024)

$\mathbf{x}$ คือเวกเตอร์รูปภาพตัวอักษรที่ต้องการหาค่าเวกเตอร์รูปแบบ (ขนาด 1024x1)

$\mathbf{m_x}$ คือเมตริกซ์ค่าเฉลี่ยของเวกเตอร์รูปภาพตัวอักษรตัวอย่าง (ขนาด 1024x1)

โดยหลักการของการวิเคราะห์องค์ประกอบสำคัญนี้จะมองว่าค่าของเวกเตอร์เจาะจงที่มีค่าสูงจะมีความเกี่ยวข้องกับตัวอักษรต้นแบบมากกว่าค่าของเวกเตอร์เจาะจงที่มีค่าต่ำ ทำให้ในการคำนวณตามจริงนั้นเราจะทำการสร้างเมตริกซ์ของเวกเตอร์เจาะจงย่อย $\mathbf{A}_k$ จากเมตริกซ์ของเวกเตอร์เจาะจง $\mathbf{A}$ ในสมการที่ (21) ที่เลือกจากเวกเตอร์เจาะจงเพียง k เวกเตอร์ที่มีค่าเวกเตอร์เจาะจงมากที่สุด ที่อธิบายถึงความสำคัญที่เกี่ยวข้องกับข้อมูลมากกว่า แล้วกำหนดให้สมาชิกของเมตริกซ์ ในแถวที่ k +1 ถึง 1,024 มีค่าเป็นศูนย์ ซึ่งทำให้ช่วยประหยัดเวลาในการคำนวณลง

ภายหลังจากที่เราได้ค่าเวกเตอร์รูปแบบแล้ว เวกเตอร์รูปแบบก็จะถูกส่งไปยังส่วนแยกแยะเพื่อทำการแยกแยะกลุ่มต่อไป

2.4.2 การวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล (Kernel Principal Component Analysis)

สำหรับการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลนั้นได้ถูกเสนอโดย Schölkopf [7] ที่ได้นำหลักการของเคอร์เนลดังที่ได้อธิบายไปเข้ามาประกอบในการแก้ปัญหาเวกเตอร์เจาะจง โดยเริ่มจากทำการแมปข้อมูลเวกเตอร์รูปภาพตัวอักษรจากปริภูมินำเข้าไปยังปริภูมิแสดงลักษณะสำคัญ นั่นคือ

$$\Phi : R^N \rightarrow F,\ \mathbf{x} \rightarrow \mathbf{X}$$

โดยที่ Φ คือฟังก์ชันการแมปข้อมูลจากปริภูมินำเข้าไปยังปริภูมิแสดงลักษณะสำคัญ

F คือปริภูมิแสดงลักษณะสำคัญ

$\mathbf{x}$ คือเวกเตอร์รูปภาพในปริภูมินำเข้า (ขนาด 1024x1)

$\mathbf{X}$ คือเวกเตอร์รูปภาพในปริภูมิแสดงลักษณะสำคัญ ที่ได้จากการแมปโดยฟังก์ชัน Φ (ขนาดขึ้นกับมิติของปริภูมิแสดงลักษณะสำคัญ)

จากนั้นจะทำการหาเมตริกซ์โคแวเรียนซ์ $\mathbf{C}$ ในปริภูมิแสดงลักษณะสำคัญนี้ดังสมการ

$$\mathbf{C} = \frac{1}{M}\sum_{j=1}^{M}\Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^{\mathrm{T}} \tag{22}$$

โดยที่ M คือจำนวนเวกเตอร์ของรูปภาพตัวอักษรทั้งหมดที่ใช้เป็นตัวอย่างในขณะเรียนรู้

และทำการแก้สมการค่าเจาะจงในทำนองเดียวกันดังสมการ

$$\lambda\mathbf{V} = \mathbf{C}\mathbf{V} \tag{23}$$

จากหลักการทางพีชคณิตเชิงเส้น เราสามารถขยายสมการที่ (23) ได้เป็น

$$\lambda(\Phi(\mathbf{x}_k)\cdot\mathbf{V}) = (\Phi(\mathbf{x}_k)\cdot\mathbf{C}\mathbf{V}) \tag{24}$$

โดยค่า k มีค่าตั้งแต่ 1 ถึง M และจะได้ว่ามีสัมประสิทธิ์ α_i ที่ทำให้

$$\mathbf{V} = \sum_{i=1}^{M}\alpha_i\Phi(\mathbf{x}_i) \tag{25}$$

เมื่อแทนค่า $\mathbf{V}$ จากสมการที่ (25) ลงในสมการที่ (24) จะได้เป็น

$$\lambda\sum_{i=1}^{M}\alpha_i(\Phi(\mathbf{x}_k)\cdot\Phi(\mathbf{x}_i)) = \frac{1}{M}\sum_{i=1}^{M}\alpha_i\left(\Phi(\mathbf{x}_k)\cdot\sum_{j=1}^{M}\Phi(\mathbf{x}_k)(\Phi(\mathbf{x}_j)\cdot\Phi(\mathbf{x}_i))\right) \tag{26}$$

จากสมการที่ (26) กำหนดเมตริกซ์ $\mathbf{K}$ โดยที่ให้ค่าในแต่ละแถว และแต่ละหลักคือค่าผลคูณภายในบนปริภูมิแสดงลักษณะสำคัญของเวกเตอร์รูปภาพตัวอย่างคูณกันตามเลขแถว และหลักทุกคู่ ดังสมการ

$$\mathbf{K}_{ij} = (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)) \tag{27}$$

เมตริกซ์ $\mathbf{K}$ ที่กำหนดขึ้น ช่วยทำให้สมการที่ (26) สามารถลดรูปได้เป็น

$$M\lambda\mathbf{K}\boldsymbol{\alpha} = \mathbf{K}^2\boldsymbol{\alpha} \tag{28}$$

หรือ

$$M\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha} \tag{29}$$

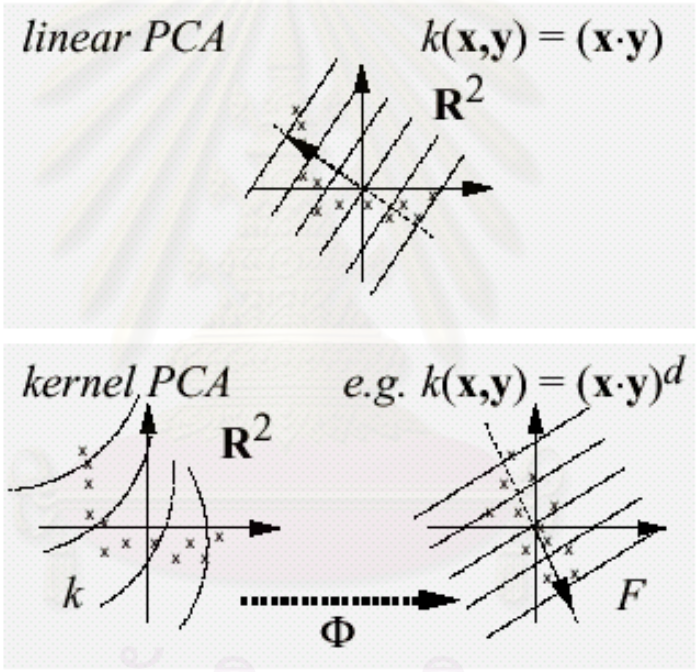
จากสมการที่ (29) สามารถมองได้เป็นสมการค่าเจาะจงสมการหนึ่ง เราสามารถหาค่าเจาะจง ของเมตริกซ์ $\mathbf{K}$ ที่นำมาใช้หาเวกเตอร์เจาะจงที่สัมพันธ์กับค่าเจาะจงแต่ละตัว และจากเวกเตอร์เจาะจงที่ได้เราสามารถนำมาหาค่าลักษณะสำคัญของรูปภาพตัวอักษรใด ๆ ได้ตามสมการ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

$$(\mathbf{V}\cdot\Phi(\mathbf{x})) = \sum_{i=1}^{M} \alpha_i \left(\Phi(\mathbf{x}_i)\Phi(\mathbf{x})\right) \tag{30}$$

ดังนั้นในการทำงานของการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล เราจะทำการสร้างเมตริกซ์ **K** จากข้อมูลรูปภาพตัวอย่างก่อน เพื่อทำการหาค่าเวกเตอร์เจาะจงของเมตริกซ์ **K** มาใช้ในหาเวกเตอร์รูปแบบของรูปภาพตัวอักษรใด ๆ ตามสมการที่ (30)

ข้อดีของการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลที่เหนือกว่าวิธีธรรมดาทั่วไปนั่นก็คือเราสามารถทำการหาค่าของเวกเตอร์เจาะจงในปริภูมิแสดงลักษณะสำคัญที่เป็นแบบเชิงเส้น แต่เมื่อมองในปริภูมิเวกเตอร์ของข้อมูลนำเข้าแล้วเราจะได้ค่าเวกเตอร์เจาะจงเป็นลักษณะไม่เชิงเส้นกับข้อมูลนั่นเอง ดังรูปที่ 10

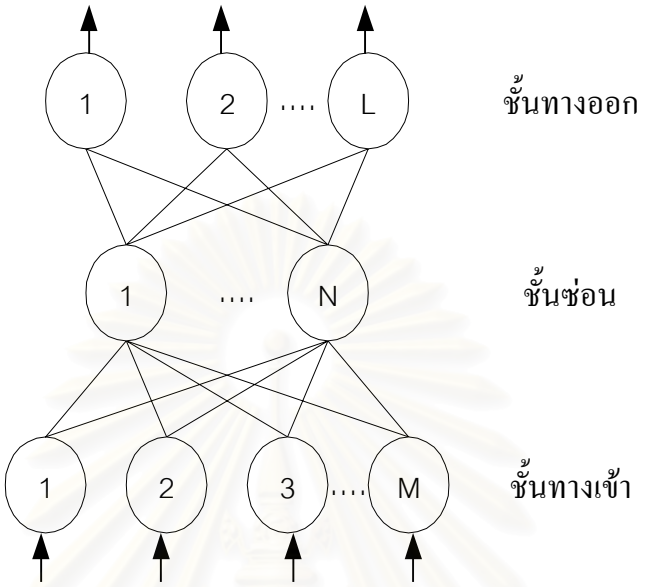


รูปที่ 10 เปรียบเทียบ PCA ธรรมดาที่เป็นลักษณะเชิงเส้นกับ Kernel PCA

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.5 นิวรอลเน็ตเวิร์กแบบแบคพรอพาเกชัน

เป็นขั้นตอนที่นำเอาเวกเตอร์รูปแบบของตัวอักษรที่ได้จากขั้นตอนการวิเคราะห์องค์ประกอบสำคัญมาทำการพิจารณาและแยกแยะตัวอักษร



รูปที่ 11 ตัวอย่างของนิวรอลเน็ตเวิร์ก

นิวรอลเน็ตเวิร์กทำหน้าที่เป็นตัวเชื่อมความสัมพันธ์ ซึ่งสามารถคำนวณหาลักษณะบางอย่างที่ก่อให้เกิดความสัมพันธ์ระหว่างข้อมูลเข้า และข้อมูลออกได้ โดยที่นิวรอลเน็ตเวิร์กแบบแบคพรอพาเกชัน เป็นเน็ตเวิร์กแบบป้อนไปข้างหน้า (feed forward network) คือจะไม่มีการป้อนผลลัพธ์ที่ได้ในแต่ละโนดย้อนกลับไปยังโนดที่ส่งข้อมูลมาให้ (แต่ในขั้นตอนการเรียนรู้จะมีการปรับปรุงค่าน้ำหนักของแต่ละโนดเมื่อตอบคำตอบผิด โดยส่งข้อมูลย้อนกลับไปจึงเรียกว่า backpropagation) โดยมีการเชื่อมโยงแบบเชื่อมถึงกันหมด (fully connected) จากรูปที่ 11 ตัวอย่างของนิวรอลเน็ตเวิร์ก แสดงให้เห็นเน็ตเวิร์กที่มี 3 ชั้น คือ ชั้นทางเข้า (input layer) ชั้นซ่อน (hidden layer) และชั้นทางออก (output layer) โดยชั้นซ่อน สามารถมีได้มากกว่า 1 ชั้น

ดังนั้นข้อมูลเข้าของ นิวรอลเน็ตเวิร์กก็คือเวกเตอร์ผลลัพธ์ (เวกเตอร์รูปแบบ) ที่ได้รับจากขั้นตอนการวิเคราะห์องค์ประกอบสำคัญ ซึ่งหากเราเลือกใช้สมาชิกของเวกเตอร์ผลลัพธ์มาจำนวน 1,024 ตัว ก็จะต้องใช้โนดนำเข้า (input node) จำนวน 1,024 โนด หรือถ้าใช้สมาชิกของเวกเตอร์ผลลัพธ์จำนวน k ตัว ก็จะต้องใช้โนดนำเข้าจำนวน k โนด

2.5.1 การทำงานของนิวรอลเน็ตเวิร์กแบบแบคพรอพาเกชันในขั้นตอนการเรียนรู้

ก่อนที่จะนำนิวรอลเน็ตเวิร์กไปใช้งานในการรู้จำนั้น จะต้องทำการสอนนิวรอลเน็ตเวิร์ก (train) ให้สามารถหาความสัมพันธ์ของข้อมูลเข้า และข้อมูลออกให้ได้ถูกต้องหรือผิดพลาดน้อยที่สุดเสียก่อน (network convergence) ซึ่งเรียกว่าขั้นตอนการเรียนรู้ (learning mode, training mode) และมีวิธีการทำงานดังนี้ (Fu [8])

- การกำหนดค่าเริ่มต้นแก่ค่าน้ำหนัก (weight) และ ค่าผันแปร (bias) โดยการสุ่มตัวเลขที่มีค่าน้อยๆ เช่นระหว่าง –0.05 ถึง 0.05
- ส่งข้อมูลที่จะทำการสอนนิวรอลเน็ตเวิร์กเข้าสู่โนดนำเข้าแล้วดำเนินการคำนวณค่าผลลัพธ์ของแต่ละโนด (ยกเว้นโนดนำเข้า) ตามสมการ

$$O_j = \mathrm{F}(\theta_j + \sum_{i=1}^{n} w_{ji}\, o_i) \qquad (31)$$

$$\mathrm{F}(a) = \frac{1}{1+e^{-a}}$$

เมื่อ O_j คือผลลัพธ์ของโนดที่ j

θ คือค่าผันแปรของโนดที่ j

w_{ji} คือค่าน้ำหนักของโนดที่ i ซึ่งส่งข้อมูล o_i มาให้โนดที่ j

o_i คือผลลัพธ์ของโนดที่ i

$\mathrm{F}(a)$ คือฟังก์ชันกระตุ้น (activation function) ซึ่งใช้ฟังก์ชันซิกมอยด์ (Sigmoid function)

- ปรับค่าน้ำหนัก

เริ่มต้นที่โนดทางออก (output node) แล้วค่อยทำย้อนกลับไปยังโนดซ่อน (hidden node) ในชั้นซ่อนขึ้นไป โดยคำนวณค่าน้ำหนักใหม่ $(W_{ji}(t+1))$ ได้ตามสมการ

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji} \qquad (32)$$

เมื่อ $W_{ji}(\mathrm{t})$ คือค่าน้ำหนักของโนดที่ i ซึ่งเชื่อมกับโนดที่ j เมื่อเวลาที่ t (หรือรอบการวนซ้ำ (iteration) ที่ t)

ΔW_{ji} คือค่าน้ำหนักที่ต้องการปรับ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ค่าน้ำหนักที่ต้องการปรับคำนวณได้จากสมการ

$$\Delta W_{ji} = \eta \delta_j O_i \quad (33)$$

เมื่อ η คือค่าอัตราการเรียนรู้ (learning rate) ($0 < \eta < 1$)

δ_j คือค่าความผิดพลาดของผลลัพธ์ของโนดที่ j

แต่ในบางครั้งต้องการที่จะให้นิวรอลเน็ตเวิร์กนั้นลู่เข้า (convergence) เร็วขึ้น ก็สามารถทำได้โดยเพิ่มค่าอินเนอร์เชีย หรือโมเมนตัมตามสมการ

$$W_{ji}(t+1) = W_{ji}(t) + \eta \delta_j O_i + \alpha(W_{ji}(t) - W_{ji}(t-1)) \quad (34)$$

เมื่อ α คือค่าอินเนอร์เชีย หรือโมเมนตัม ($0 < \alpha < 1$)

ค่าความผิดพลาดของผลลัพธ์หาได้ดังนี้

หากเป็นโนดทางออกใช้สมการ

$$\delta_j = O_j(1-O_j)(T_j - O_j) \quad (35)$$

เมื่อ T_j คือผลลัพธ์ที่ต้องการได้ (target) จากโนดทางออกที่ j

O_j คือผลลัพธ์ที่คำนวณได้ (actual) จากโนดทางออกที่ j

หากเป็นโนดซ่อนใช้สมการ

$$\delta_j = O_j(1-O_j)\sum_k \delta_k W_{kj} \quad (36)$$

เมื่อ δ_k คือค่าความผิดพลาดของผลลัพธ์ของโนดที่ k (ซึ่งเชื่อมต่อจากโนดที่ j)

W_{kj} คือค่าน้ำหนักของโนดที่ j ซึ่งเชื่อมกับโนดที่ k

ทำการปรับค่าน้ำหนักในโนดอื่นจนครบ แล้วทำการสอนนิวรอลเน็ตเวิร์กในตัวอย่างถัดไป (iteration ถัดไป) จนกระทั่งการเรียนรู้ลู่เข้า (network convergence)

เมื่อทำการเรียนรู้ของนิวรอลเน็ตเวิร์กเสร็จสิ้นลง จะทำการเก็บค่าน้ำหนัก (weight) และค่าผันแปร (bias) ของทุก ๆ โนดเพื่อนำไปใช้ในการรู้จำต่อไป

2.5.2 การทำงานของนิวรอลเน็ตเวิร์กแบบแบคพรอพาเกชันในขั้นตอนการรู้จำ

ในขั้นตอนการรู้จำ (recognition mode) จะมีการรับข้อมูล (สมาชิกของเวกเตอร์ผลลัพธ์) เข้าทางโนดนำเข้า (ในชั้นทางเข้า) จากนั้นแต่ละโนดนำเข้าจะทำการส่งข้อมูลไปยังทุก ๆ โนดซ่อน (ในชั้นซ่อน) จากนั้นแต่ละโนดซ่อนจะทำการคำนวณค่าที่ได้รับมาตามสมการ

$$O_j = \mathrm{F}(\theta_j + \sum_{i=1}^{n} w_{ji}\, o_i) \qquad (37)$$
$$\mathrm{F}(a) = \frac{1}{1+e^{-a}}$$

เมื่อ O_j คือผลลัพธ์ของโนดที่ j

θ คือค่าผันแปรของโนดที่ j

w_{ji} คือค่าน้ำหนักของโนดที่ i ซึ่งส่งข้อมูล o_i มาให้โนดที่ j

o_i คือผลลัพธ์ของโนดที่ i

$\mathrm{F}(a)$ คือฟังก์ชันกระตุ้น (activation function) ซึ่งใช้ฟังก์ชันซิกมอยด์ (Sigmoid function)

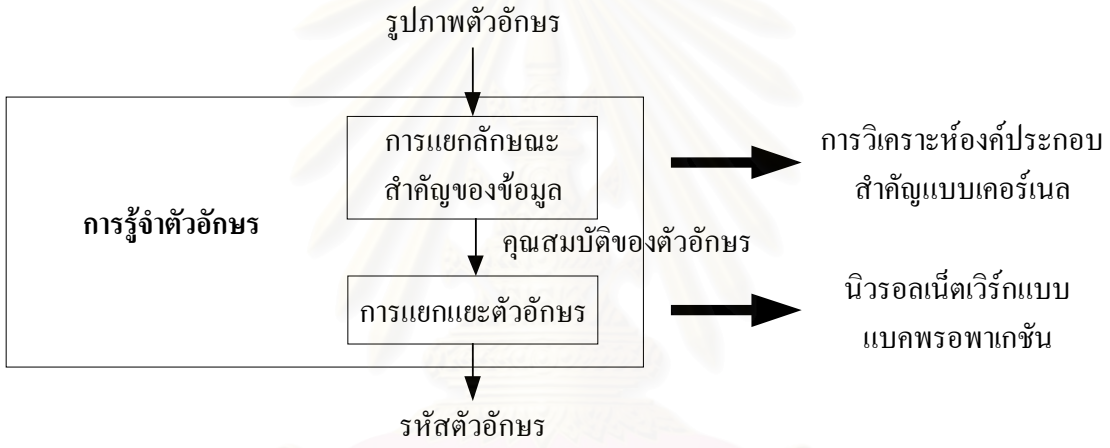
จากนั้นโนดซ่อนจะส่งผลลัพธ์ของตัวเอง ต่อไปยังทุก ๆ โนดทางออก (ในชั้นทางออก) ซึ่งโนดทางออกแต่ละโนดก็จะคำนวณค่าที่ได้รับมาตามสมการข้างบน แล้วตอบผลลัพธ์ของโนดตัวเอง ซึ่งเราสามารถพิจารณาผลลัพธ์ของทุก ๆ โนดทางออก แล้วหาผลลัพธ์สุดท้ายได้ เช่นถ้าผลลัพธ์ของโนดทางออก ที่ 1 มีค่าเท่ากับ 1 และผลลัพธ์ของโนดทางออกตัวที่เหลือเท่ากับ 0 ทั้งหมด แสดงว่าข้อมูลเข้าที่ได้รับคือข้อมูลของตัวอักษร ก เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การออกแบบและพัฒนา

ในการพัฒนาโปรแกรมโอซีอาร์ภาษาไทยของงานวิจัยฉบับนี้ มุ่งเน้นที่จะพัฒนากระบวนการในส่วนการแยกลักษณะสำคัญของรูปภาพตัวอักษรในส่วนของการรู้จำของโปรแกรมโอซีอาร์ภาษาไทย โดยนำเอาการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลมาใช้แทนการวิเคราะห์องค์ประกอบสำคัญแบบเดิม และในส่วนของการแยกแยะตัวอักษรนั้นจะนำนิวรอลเน็ตเวิร์กแบบแบคพรอพาเกชันมาใช้ในการพิจารณาว่ารูปภาพตัวอักษรนั้นเป็นตัวอักษรชนิดใด แสดงได้ดังแผนภาพในรูปที่ 12



รูปที่ 12 แผนภาพขั้นตอนต่าง ๆ ที่ใช้ในการพัฒนาโปรแกรม

3.1 การแปลงข้อมูลรูปภาพตัวอักษร

ที่ขั้นตอนนี้ข้อมูลรูปภาพตัวอักษรจะถูกนำมาทำการแปลงเป็นข้อมูลเวกเตอร์ขนาด 1024x1 โดยการไล่สแกนค่าจุดสีในรูปภาพ ซึ่งจะแทนค่าจุดสีดำด้วย –1 และแทนค่าจุดสีขาวด้วย 1 ดังที่ได้อธิบายไว้ในบทที่ 2 จากนั้นข้อมูลเวกเตอร์รูปภาพที่ได้จะถูกส่งให้กระบวนการแยกลักษณะสำคัญของรูปภาพตัวอักษรทำการหาค่าลักษณะสำคัญของตัวอักษรแต่ละตัวต่อไป

โดยในงานวิจัยนี้ได้แบ่งข้อมูลรูปภาพตัวอักษรที่ใช้เป็น 2 ชุดด้วยกัน คือ

1. รูปภาพในชุดเรียนรู้ เป็นรูปภาพของตัวอักษรที่เก็บไว้ในแฟ้มข้อมูล 1 แฟ้มต่อ 1 ตัวอักษร โดยจัดเก็บรูปภาพไว้ในลักษณะบิตแมป (bitmap) สีขาว-ดำ ประกอบด้วยรูปแบบตัวอักษร 6 รูปแบบที่ใช้กันในระบบปฏิบัติการวินโดว์ คือ AngsanaUPC, BrowalliaUPC, CordiaUPC, DilleniaUPC, EucrosiaUPC และ FreesiaUPC

ซึ่งแต่ละรูปแบบจะประกอบด้วยตัวอักษรขนาด 14, 16, 18, 20, 22, 24, 28 และ 36 จุด และมีจำนวนตัวอักษร 178 ตัว คือ

- **ตัวอักษรไทย สระ และวรรณยุกต์ จำนวน 68 ตัว**

ก ข ฃ ค ฅ ฆ ง จ ฉ ช 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 โ ใ ไ ๐ ฤ ฦ ๆ ฯ

- **ตัวอักษรภาษาอังกฤษตัวเล็ก และตัวใหญ่ จำนวน 52 ตัว**

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

- **ตัวเลข และสัญลักษณ์พิเศษอื่น ๆ จำนวน 58 ตัว**

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; ' ' < , > . ? /

และ เพิ่ม ตัวอักษร '!' โดยไม่มี '.' อยู่ด้านล่าง

เพิ่ม ตัวอักษร '?' โดยไม่มี '.' อยู่ด้านล่าง

เพิ่ม ตัวอักษร 'i' โดยไม่มี '.' อยู่ด้านบน

เพิ่ม ตัวอักษร 'j' โดยไม่มี '.' อยู่ด้านบน

เพิ่ม ตัวอักษร 'ญ' โดยไม่มี สัญลักษณ์พิเศษที่อยู่ด้านล่าง

เพิ่ม ตัวอักษร 'ฐ' โดยไม่มี สัญลักษณ์พิเศษที่อยู่ด้านล่าง

เพิ่ม ตัวอักษร สัญลักษณ์พิเศษที่อยู่ด้านล่างของ 'ญ'

เพิ่ม ตัวอักษร สัญลักษณ์พิเศษที่อยู่ด้านล่างของ 'ฐ'

โดยรูปภาพตัวอักษรในชุดเรียนรู้นี้จะใช้เป็นข้อมูลรูปภาพตัวอย่างในส่วนของการวิเคราะห์องค์ประกอบสำคัญ รวมจำนวนแฟ้มรูปภาพตัวอักษรในชุดเรียนรู้นี้เป็นจำนวนทั้งสิ้น 8,544 รูป (ตัวอย่างตัวอักษรทั้งหมดสามารถดูได้ในภาคผนวก ง.)

2. รูปภาพในชุดทดสอบ มีลักษณะรูปแบบ ขนาด และชนิดตัวอักษรเช่นเดียวกันกับรูปภาพในชุดเรียนรู้ แต่จะมีการแบ่งปันรูปแบบตัวอักษรกันไปในแต่ละชนิดตัวอักษรและขนาด เช่นในตัวอักษร ก จะประกอบด้วย รูปภาพ 8 รูป แต่ละรูปคือแต่ละขนาดตัวอักษร โดยที่ในแต่ละรูปนั้นจะมีรูปแบบตัวอักษรที่อาจจะเหมือนหรือแตกต่างกันไป รูปภาพในชุดทดสอบจะนำมาใช้ทดสอบเพื่อวัดผลภายหลังจากมีการเรียนรู้ด้วยรูปภาพในชุดเรียนรู้แล้ว รวมจำนวนแฟ้มรูปภาพตัวอักษรในชุดทดสอบเป็นจำนวนทั้งสิ้น 1,424 รูป

3.2 การแยกลักษณะสำคัญของรูปภาพตัวอักษร

ในขั้นตอนนี้ ข้อมูลเวกเตอร์รูปภาพขนาด 1024x1 ที่ได้ จะถูกนำมาแยกลักษณะสำคัญ โดยการวิเคราะห์องค์ประกอบสำคัญทั้งในแบบเดิม และแบบเคอร์เนล ผลลัพธ์ที่ได้ภายหลังจากที่ผ่านขั้นตอนนี้แล้วจะได้เป็นเวกเตอร์รูปแบบ ที่จะส่งให้กระบวนการแยกแยะตัวอักษรทำการแยกแยะต่อไป

ในกรณีของการวิเคราะห์องค์ประกอบสำคัญแบบเดิมนั้น ข้อมูลเวกเตอร์รูปภาพตัวอย่างจากชุดเรียนรู้จะถูกนำมาใช้ในการสร้างเมตริกซ์ค่าเฉลี่ย และเมตริกซ์โคแวเรียนซ์ ตามวิธีการของการแปลงแบบเค-แอล (Karhunen-Loève Transform : KLT) เพื่อใช้ในการหาเมตริกซ์เวกเตอร์เจาะจงดังสมการในบทที่ 2 ก่อนสร้างเป็นเวกเตอร์รูปแบบต่อไป โดยในการแก้ปัญหาเวกเตอร์เจาะจงนี้จะทำการคำนวณผ่านทางฟังก์ชันการทำงานที่เรียกว่า sgeev_ ที่อยู่ภายในไลบรารี (library) ที่เกี่ยวกับการแก้ปัญหาทางพีชคณิตเชิงเส้นต่าง ๆ ที่ชื่อว่า LAPACK (Linear Algebra Package) [9] โดยฟังก์ชัน sgeev_ นี้จะทำการคำนวณเพื่อหาค่าเจาะจง และเมตริกซ์ของค่าเจาะจงแต่ละตัวที่ได้จากเมตริกซ์ที่เราต้องการ ซึ่งในกรณีนี้ก็คือเมตริกซ์โคแวเรียซ์ที่ได้นั่นเอง

ส่วนข้อมูลเวกเตอร์รูปภาพจากชุดทดสอบนั้นจะถูกนำมาหาค่าเวกเตอร์รูปแบบโดยอาศัยตัวเมตริกซ์ค่าเฉลี่ย และเมตริกซ์เวกเตอร์เจาะจงที่ได้จากข้อมูลเวกเตอร์รูปภาพในชุดเรียนรู้

สำหรับกรณีของการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล ข้อมูลเวกเตอร์รูปภาพตัวอย่างจากชุดเรียนรู้จะถูกนำมาใช้ในการสร้างเมตริกซ์ $\mathbf{K}$ ที่ได้จากค่าผลคูณภายในของเวกเตอร์รูปภาพชุดเรียนรู้ที่ถูกแมปไปยังปริภูมิแสดงลักษณะสำคัญแล้วทุกตัวคูณกัน ซึ่งจะสามารถคำนวณได้โดยอาศัยฟังก์ชันเคอร์เนล ในงานวิจัยฉบับนี้เลือกทดลองฟังก์ชันเคอร์เนลไว้ 3 แบบด้วยกัน คือ

Polynomial classifiers of degree d : $k(\mathbf{x},\mathbf{y}) = (\mathbf{x} \bullet \mathbf{y})^d$

Radial Basis Function (RBF) classifier : $k(\mathbf{x},\mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/c)$

Neural networks : $k(\mathbf{x},\mathbf{y}) = \tanh(\kappa \bullet (\mathbf{x} \bullet \mathbf{y}) + \Theta)$

ซึ่งในแต่ละฟังก์ชันเคอร์เนลก็จะมีการปรับค่าตัวแปรสัมประสิทธิ์ในสมการเพื่อทดสอบหาค่าที่ให้ผลลัพธ์การทำงานออกมาดีที่สุด โดยมีการปรับเปลี่ยนดังนี้

- ฟังก์ชันเคอร์เนล Polynomial ทำการปรับค่าตัวแปรดีกรี d ที่เป็นจำนวนเต็ม ตั้งแต่ค่า 1 ไล่ไปเรื่อย ๆ ทีละหนึ่ง แล้วหาค่าที่ให้ค่าคำตอบที่ผิดพลาดน้อยที่สุด
- ฟังก์ชันเคอร์เนล RBF ทำการปรับค่าตัวหาร c ที่เป็นค่าจำนวนจริงที่ไม่ใช่ศูนย์ หาค่าที่อยู่ในช่วงที่ให้ผลลัพธ์ที่ดีที่สุด

- ฟังก์ชันเคอร์เนลนิวรอลเน็ตเวิร์ก ปรับค่าตัวแปร κ และ Θ ในสมการ หาค่าที่ให้ผลลัพธ์ที่ผิดพลาดน้อยที่สุด

สำหรับเมตริกซ์ $\mathbf{K}$ ที่ได้จะถูกนำมาหาเมตริกซ์เวกเตอร์เจาะจง เพื่อนำมาใช้ในการหาเวกเตอร์รูปแบบของรูปภาพตัวอักษรต่อไป ส่วนข้อมูลเวกเตอร์รูปภาพในชุดทดสอบก็จะสามารถหาเวกเตอร์รูปแบบโดยอาศัยเมตริกซ์เวกเตอร์เจาะจงของเมตริกซ์ $\mathbf{K}$ ที่ได้จากเวกเตอร์รูปภาพในชุดเรียนรู้มาใช้ในคำนวณ

3.3 การแยกแยะตัวอักษร

ข้อมูลเวกเตอร์รูปแบบที่ได้จากขั้นตอนการแยกลักษณะสำคัญของรูปภาพตัวอักษรจะถูกนำมาแยกแยะตัวอักษรโดยใช้วิธีการของนิวรอลเน็ตเวิร์กที่จะพิจารณาแยกแยะว่าเป็นตัวอักษรตัวใด โดยที่ค่าขนาดของเวกเตอร์รูปแบบที่ส่งเข้ามาให้นิวรอลเน็ตเวิร์กนั้นจะมีความสัมพันธ์กับค่าของโนดนำเข้าของนิวรอลเน็ตเวิร์กที่ใช้ในขั้นตอนการแยกแยะตัวอักษร คือต้องมีขนาดที่เท่ากัน ซึ่งในงานวิจัยนี้ขนาดของเวกเตอร์รูปแบบที่ส่งให้กับกระบวนการแยกแยะตัวอักษรจะมีขนาดแตกต่างกันไปในแต่ละกรณีของการวิเคราะห์องค์ประกอบสำคัญ สำหรับการวิเคราะห์องค์ประกอบสำคัญแบบเดิมนั้นจะคงขนาดเดิมแบบเดียวกันกับงานวิจัยของชาญฤทธิ์ [1] ไว้ เพื่อใช้ในการเทียบผลกับการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล ส่วนในกรณีของการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลได้มีการปรับเพิ่มจำนวนโนดนำเข้าให้มากขึ้นเพื่อให้สอดคล้องกับข้อมูลที่ถูกแมปไปยังปริภูมิข้อมูลที่มีมิติสูงขึ้น โดยขนาดของเวกเตอร์รูปแบบที่ใช้ในการวิเคราะห์องค์ประกอบสำคัญแต่ละแบบแสดงดังตารางที่ 1

ตารางที่ 1 ขนาดของเวกเตอร์รูปแบบที่ใช้กับการวิเคราะห์องค์ประกอบสำคัญแต่ละแบบ

ประเภทของวิธีการวิเคราะห์องค์ประกอบสำคัญ	ขนาดของเวกเตอร์รูปแบบที่ใช้
วิเคราะห์องค์ประกอบสำคัญแบบเดิม	128, 150 และ 250
วิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล	1024, 1200, 2048, 4096 และ 5120

สำหรับค่าที่กำหนดให้กับนิวรอลเน็ตเวิร์กประกอบด้วย

1. จำนวนโนดนำเข้า โนดซ่อน และโนดทางออก

ค่าโนดนำเข้า และโนดซ่อนที่ใช้ในนิวรอลเน็ตเวิร์กนั้นจะต่างกันไปในแต่ละแบบของวิธีการวิเคราะห์องค์ประกอบสำคัญ เนื่องจากขนาดของเวกเตอร์รูปแบบที่ต่างกัน ส่วนในกรณีของโนดทางออกจะมีค่าเท่ากับ 178 โนด ที่ได้มาจากจำนวนตัวอักษรทั้งหมดที่เราต้องการจำแนก

โดยในแต่ละแบบของวิธีการวิเคราะห์องค์ประกอบสำคัญในงานวิจัยนี้ กำหนดค่าโนดนำเข้า โนดซ่อน และโนดทางออกไว้ดังตารางที่ 2

ตารางที่ 2 ค่าโนดนำเข้า โนดซ่อน และโนดทางออกที่ใช้ในงานวิจัย

แบบของเน็ตเวิร์ก	การวิเคราะห์องค์ประกอบสำคัญแบบเดิม			การวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล		
	จำนวนโนดนำเข้า	จำนวนโนดซ่อน	จำนวนโนดทางออก	จำนวนโนดนำเข้า	จำนวนโนดซ่อน	จำนวนโนดทางออก
แบบที่ 1	128	128	178	1024	128	178
แบบที่ 2	128	150	178	1024	150	178
แบบที่ 3	128	250	178	1024	250	178
แบบที่ 4	150	128	178	1200	128	178
แบบที่ 5	150	150	178	1200	150	178
แบบที่ 6	150	250	178	1200	250	178
แบบที่ 7	250	128	178	2048	128	178
แบบที่ 8	250	150	178	2048	150	178
แบบที่ 9	250	250	178	2048	250	178
แบบที่ 10	-			4096	128	178
แบบที่ 11	-			4096	150	178
แบบที่ 12	-			4096	250	178
แบบที่ 13	-			5120	128	178
แบบที่ 14	-			5120	150	178
แบบที่ 15	-			5120	250	178

2. ค่าอัตราการเรียนรู้ (learning rate) เท่ากับ 0.08 ในกรณีของการวิเคราะห์องค์ประกอบสำคัญแบบเดิม ส่วนการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลนั้นจะมีการเปลี่ยนค่าไปเพื่อหาค่าที่ให้ผลลัพธ์ในการรู้จำที่ดีที่สุด
3. ค่าอินเนอร์เชีย หรือโมเมนตัม (inertia or momentum) เท่ากับ 0.95
4. ฟังก์ชันกระตุ้น (activation function) ใช้แบบฟังก์ชันซิกมอยด์ (Sigmoid function) ที่ให้ค่าของฟังก์ชันอยู่ในช่วง (0,1)

5. เงื่อนไขของการลู่เข้า (convergence) คือต้องเรียนรู้ตัวอย่างให้ถูกต้องเป็นจำนวน 1,424 ซึ่งเท่ากับจำนวนรูปภาพในชุดทดสอบ โดยมีค่าผิดพลาดไม่เกิน 0.5
6. ค่าเริ่มต้นของค่าน้ำหนัก (weight) และค่าผันแปร (bias) ได้จากการสุ่มค่าระหว่าง [-0.1,0.1]

การพิจารณาค่าผลลัพธ์ของเน็ตเวิร์กนั้นจะพิจารณาจากโนดที่ให้ค่าสูงที่สุดเพียงโนดเดียวเป็นคำตอบ เช่น กำหนดให้ว่าหากเป็นตัวอักษร ก นั้น โนดที่ 1 จะต้องมีค่าสูงที่สุด ซึ่งหากเน็ตเวิร์กใดที่ให้ค่าที่สูงที่สุดไปอยู่ที่โนดอื่น ๆ จะถือว่าเน็ตเวิร์กนั้นตอบผิด

ส่วนการพิจารณาว่าจะเลือกเน็ตเวิร์กแบบใดมาใช้งานนั้น จะพิจารณาจากความผิดพลาดของข้อมูลรูปภาพในชุดทดสอบที่ให้ค่าความผิดพลาดที่น้อยที่สุด ซึ่งหากเกิดกรณีที่มีเน็ตเวิร์กที่ให้ค่าความผิดพลาดของรูปภาพในชุดทดสอบออกมาเท่ากัน ก็จะพิจารณาค่าความผิดพลาดของข้อมูลในชุดเรียนรู้มาประกอบเพิ่มเติม แล้วเลือกเน็ตเวิร์กที่ให้ค่าความผิดพลาดที่น้อยที่สุดมาใช้งาน ซึ่งเน็ตเวิร์กที่ได้สามารถนำเข้าไปประยุกต์กับโปรแกรมโอซีอาร์ภาษาไทยของชาญฤทธิ์ [1] เพื่อสร้างเป็นโปรแกรมโอซีอาร์ภาษาไทยที่มีการรู้จำโดยใช้การวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลเพิ่มขึ้นมา ดังแสดงในภาคผนวก ข.

สำหรับโปรแกรมที่เกี่ยวข้องกับการทำงานของการแยกลักษณะสำคัญของรูปภาพตัวอักษร และการแยกแยะตัวอักษร ได้แสดงไว้แล้วในภาคผนวก ค.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

ผลการวิจัย

4.1 วิธีการทดสอบ

ในงานวิจัยนี้ได้ทำการทดสอบ เปรียบเทียบผลการรู้จำไว้ 2 กรณีด้วยกัน คือ

1. เปรียบเทียบผลการรู้จำที่ได้ระหว่างการวิเคราะห์องค์ประกอบสำคัญแบบเดิม และแบบเคอร์เนล

   การทดสอบในครั้งนี้ทำโดยการนำรูปภาพตัวอักษรในชุดเรียนรู้ทั้งหมด 8,544 รูป และรูปภาพชุดทดสอบจำนวน 1,424 รูปมาทำการรู้จำ โดยส่งรูปภาพตัวอักษรทั้งหมดเข้าสู่ขั้นตอนประมวลผลรู้จำโดยตรง โดยไม่ผ่านขั้นตอนเตรียมการประมวลผลในช่วงแรก แล้วทำการเปรียบเทียบผลลัพธ์ที่ได้จากการรู้จำที่เกิดขึ้นระหว่างการใช้การวิเคราะห์องค์ประกอบสำคัญแบบเดิม และแบบเคอร์เนล โดยผลลัพธ์ที่ได้จะเป็นจำนวนตัวอักษรที่รู้จำผิดพลาดจากจำนวนรูปภาพตัวอักษรในแต่ละชุด

2. เปรียบเทียบผลการรู้จำที่ได้ระหว่างโปรแกรมโอซีอาร์ภาษาไทยที่มีการใช้วิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล (ภาคผนวก ข.) กับโปรแกรมโอซีอาร์ภาษาไทยโปรแกรมอื่นที่มีใช้กันอยู่ในปัจจุบันเพิ่มเติม ได้แก่โปรแกรมอ่านไทย 2.0 ที่พัฒนาโดยฝ่ายกลุ่มวิจัย และพัฒนาสาขาสารสนเทศ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) [10]

   ในการทดสอบครั้งนี้รูปภาพตัวอักษรในชุดทดสอบ และชุดเรียนรู้ที่ใช้เป็นชุดเดียวกันกับการทดสอบในขั้นตอนแรก แต่ในการทดสอบครั้งนี้รูปภาพทั้งหมดจะถูกเรียกเปิดจากโปรแกรมขึ้นมารู้จำทีละรูป ซึ่งรูปภาพตัวอักษรจะถูกผ่านขั้นตอนเตรียมการประมวลผลก่อนไปสู่ขั้นตอนประมวลผลรู้จำรวมถึงมีการทำงานในขั้นตอนหลังการประมวลผลด้วย ทำให้ผลการรู้จำของวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลที่ได้ที่นำมาเปรียบเทียบในขั้นตอนนี้จะมีความแตกต่างกันกับในกรณีแรก

4.2 ผลการทดสอบ

4.2.1 ผลการทดสอบเปรียบเทียบผลการรู้จำที่ได้ระหว่างการวิเคราะห์องค์ประกอบสำคัญแบบเดิม และแบบเคอร์เนล

ผลของการทดสอบแสดงเน็ตเวิร์กที่ให้ค่ารู้จำที่ผิดพลาดน้อยที่สุด ระหว่างวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเดิม และแบบเคอร์เนลที่มีชนิดเป็นฟังก์ชันเคอร์เนล RBF ที่มีค่าสัมประสิทธิ์ตัวหาร c เท่ากับ 0.3 ที่เป็นฟังก์ชันเคอร์เนลที่ให้ผลลัพธ์ออกมาดีกว่าฟังก์ชันเคอร์เนลชนิดอื่น (โดยสามารถดูรายละเอียดการเลือกฟังก์ชันเคอร์เนล และการหาค่าสัมประสิทธิ์ที่เกี่ยวข้องกับเคอร์เนลในแต่ละแบบ เพื่อเลือกค่าฟังก์ชันเคอร์เนลที่ให้ผลลัพธ์ที่ดีที่สุดได้ในภาคผนวก ก.) สามารถแสดงได้ดังตารางที่ 3 (ผลลัพธ์ที่ดีที่สุด คือ รู้จำผิดพลาดน้อยที่สุดแสดงด้วยตัวหนา)

ตารางที่ 3 ผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดของการวิเคราะห์องค์ประกอบสำคัญแต่ละแบบ

จำนวนโนดในนิวรอลเน็ตเวิร์ก		จำนวนรูปภาพตัวอักษรที่รู้จำผิดพลาด (ตัว)	
โนดนำเข้า	โนดซ่อน	ข้อมูลชุดทดสอบ (1,424 ตัว)	ข้อมูลชุดเรียนรู้ (8,544 ตัว)
วิธีการวิเคราะห์องค์ประกอบสำคัญแบบเดิม			
128	128	32	74
128	150	32	71
128	250	28	153
150	128	31	121
150	150	33	134
150	250	31	79
250	128	29	51
250	150	34	100
250	250	32	46
วิธีการวิเคราะห์องค์ประกอบสำคัญแบบ RBF สัมประสิทธิ์ตัวหารเท่ากับ 0.3			
1024	128	30	135
1024	150	32	118
1024	250	32	126
1200	128	30	115
1200	150	34	146

1200	250	37	148
2048	128	34	118
2048	150	34	138
2048	250	33	145
4096	128	**27**	**109**
4096	150	32	112
4096	250	33	102
5120	128	32	117
5120	150	30	121
5120	250	33	94

จากตารางที่ 3 เมื่อพิจารณาผลการรู้จำที่ผิดพลาดจากรูปภาพในชุดทดสอบเป็นหลัก การวิเคราะห์องค์ประกอบสำคัญแบบเดิมให้ผลการรู้จำรูปภาพในชุดทดสอบที่ดีที่สุดโดยรู้จำผิดพลาดอยู่ที่ 29 รูปภาพตัวอักษร โดยมีผลการรู้จำผิดพลาดในรูปภาพชุดเรียนรู้อยู่ที่ 51 รูปภาพตัวอักษร ขณะที่ในการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลสามารถให้ผลการรู้จำที่ดีกว่า คือรู้จำตัวอักษรผิดพลาดน้อยกว่าอยู่ที่ 27 รูปภาพตัวอักษร แต่หากพิจารณากับรูปภาพตัวอักษรที่รู้จำผิดพลาดในข้อมูลชุดเรียนรู้ด้วยแล้ว เห็นว่ายังมีความผิดพลาดมากกว่าการวิเคราะห์องค์ประกอบสำคัญแบบเดิมอยู่สูง ดังนั้นจึงได้ทำการทดลองเพิ่มเติมกับกรณีของการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลเพิ่มขึ้นอีก โดยทำการทดลองปรับเปลี่ยนค่าที่เกี่ยวข้องกับเน็ตเวิร์กที่ให้ผลการรู้จำผิดพลาดน้อยที่สุดในตารางที่ 3 คือเน็ตเวิร์กที่มีจำนวนโนดนำเข้า และโนดซ่อนเท่ากับ 4,096 และ 128 โนด ตามลำดับ โดยเลือกทำการทดลองปรับเปลี่ยนค่าอัตราการเรียนรู้ (learning rate) จากค่าเดิมที่ 0.08 เป็นค่าอื่น ๆ แล้วเก็บผลการรู้จำตัวอักษรผิดพลาดที่ได้อีกครั้งหนึ่ง ได้ผลแสดงในตารางที่ 4

จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 4 ผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดของการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลที่ค่าอัตราเรียนรู้ (learning rate) แตกต่างกันในแต่ละค่า

ค่าอัตราการเรียนรู้	จำนวนรูปภาพตัวอักษรที่รู้จำผิดพลาด (ตัว)	
	ข้อมูลชุดทดสอบ (1,424 ตัว)	ข้อมูลชุดเรียนรู้ (8,544 ตัว)
0.01	24	29
0.02	**21**	**29**
0.04	26	49
0.06	32	106
0.08 (ค่าเดิม)	27	109

จากตารางที่ 4 เห็นได้ว่าเมื่อทำการปรับค่าอัตราการเรียนรู้ของนิวรอลเน็ตเวิร์กแล้ว สามารถให้ผลการรู้จำรูปภาพตัวอักษรผิดพลาดน้อยลงทั้งรูปภาพตัวอักษรในชุดทดสอบ และชุดเรียนรู้ โดยได้ผลการรู้จำที่ดีที่สุดอยู่ที่ค่าอัตราการเรียนรู้เท่ากับ 0.02 ซึ่งเป็นผลการรู้จำที่ดีกว่าการวิเคราะห์องค์ประกอบสำคัญแบบเดิมทั้งสองชุดด้วย ทำให้สรุปได้ว่าทฤษฎีเคอร์เนลที่นำเข้ามาประยุกต์ใช้กับการวิเคราะห์องค์ประกอบสำคัญแบบเดิมนั้นช่วยทำให้ผลการรู้จำถูกต้องยิ่งขึ้นจริง

4.2.2 **ผลการทดสอบเปรียบเทียบผลการรู้จำที่ได้ระหว่างโปรแกรมโอซีอาร์ภาษาไทยที่มีการใช้วิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล กับโปรแกรมโอซีอาร์ภาษาไทยอ่านไทย** 2.0

จากการทดสอบ พบว่าผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดระหว่างโปรแกรมโอซีอาร์ภาษาไทยที่นำเอาวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลไปใช้ให้ผลลัพธ์ที่ดีกว่าโปรแกรมโอซีอาร์ภาษาไทยอ่านไทย 2.0 ดังแสดงในตารางที่ 5

ตารางที่ 5 ผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดที่ได้จากโปรแกรมโอซีอาร์ภาษาไทยแต่ละชนิด

ชนิดของโปรแกรมโอซีอาร์ภาษาไทย	จำนวนรูปภาพตัวอักษรที่รู้จำผิดพลาด (ตัว)		เวลาที่ใช้ในการรู้จำ (นาที)
	ข้อมูลชุดทดสอบ (1,424 ตัว)	ข้อมูลชุดเรียนรู้ (8,544 ตัว)	
วิทยานิพนธ์ฉบับนี้	**276**	**1547**	838
อ่านไทย 2.0	354	2135	457

ผลการทดสอบที่ได้ ทำการทดสอบบนเครื่องไมโครคอมพิวเตอร์ ใช้หน่วยประมวลผลกลาง Pentium III 800MHz มีหน่วยความจำ 512MB ทำงานบนระบบปฏิบัติการ Microsoft Windows 2000 Professional Edition

โดยเวลาที่ใช้ในการรู้จำวัดจากเวลาที่ได้จากการรู้จำโดยวิธีการรู้จำทีละหลายแฟ้มรูปภาพ (batch recognition mode) ของโปรแกรม ซึ่งในกรณีของโปรแกรมอ่านไทย 2.0 มีข้อน่าสังเกตว่าเมื่อทำการรู้จำแฟ้มรูปภาพโดยใช้วิธีการรู้จำทีละหลายแฟ้มรูปภาพ (batch recognition mode) นั้น หากทำการรู้จำแฟ้มรูปภาพจำนวนมากรูปในคราวเดียว เวลาที่ใช้ในการรู้จำก็จะยิ่งมากขึ้นตามไปด้วย อย่างในกรณีของผลที่ได้ในตารางมาจากการรู้จำรูปภาพคราวละ 1,424 รูปภาพทั้งในส่วนของรูปภาพชุดทดสอบ และรูปภาพชุดเรียนรู้ แล้วทำการจับเวลาที่ได้ สุดท้ายจึงทำการรวมเวลาที่ใช้ในการรู้จำรูปภาพทั้งหมดมาแสดงในตาราง ซึ่งหากให้โปรแกรมอ่านไทย 2.0 เปิดแฟ้มรูปภาพขึ้นมารู้จำในจำนวนที่น้อยลงจาก 1,424 รูปในแต่ละคราวแล้ว เวลาที่ใช้ในการรู้จำจะยิ่งเร็วขึ้น ส่วนในกรณีของโปรแกรมโอซีอาร์ภาษาไทยในวิทยานิพนธ์ฉบับนี้จำนวนแฟ้มรูปภาพที่เปิดขึ้นมารู้จำในแต่ละคราวไม่มีผลต่อการเพิ่มหรือลดเวลาในการรู้จำรูปภาพทั้งหมดแต่อย่างใด

4.3 ปัญหา และข้อจำกัด

1. ปัญหาเรื่องหน่วยความจำที่ใช้ในการทำงาน

   เนื่องมาจากการที่ต้องทำการแมปข้อมูลไปยังปริภูมิที่มีมิติสูงกว่าเดิม ข้อมูลจึงมีขนาดที่ใหญ่ขึ้น ทำให้ต้องใช้หน่วยความจำที่มากขึ้นในการเก็บข้อมูลก่อนทำการประมวลผล และยิ่งข้อมูลตัวอย่างที่ใช้ในการเรียนรู้มีจำนวนมากแล้ว หน่วยความจำก็จำต้องเพิ่มมากตามไปด้วย ทำให้ในการทดลองเพื่อที่จะสร้างเวกเตอร์รูปแบบในแต่ละเคอร์เนลจำเป็นต้องใช้เครื่องประมวลผลที่มีหน่วยความจำขนาดใหญ่

2. ปัญหาเรื่องชนิดของเคอร์เนล และมิติของข้อมูลที่สูงขึ้นส่งผลต่อเวลาการทำงานที่กินระยะเวลายาวนานขึ้น

   ในการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล หากเคอร์เนลที่ใช้มีความซับซ้อนในการคำนวณสูงก็จะยิ่งเสียเวลาในการรู้จำที่เพิ่มขึ้น รวมถึงการที่วิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลต้องมีการแมปข้อมูลไปยังปริภูมิที่มีมิติที่สูงขึ้น ทำให้การหาค่าเมตริกซ์เวกเตอร์เจาะจงก็จะกินเวลานานขึ้น นอกจากนี้จำนวนโนดนำเข้าของนิวรอลเน็ตเวิร์กที่มีจำนวนเพิ่มขึ้นตามขนาดของเวกเตอร์รูปแบบที่ใหญ่ขึ้น ก็จะส่งผลให้การทำงานของนิวรอลเน็ตเวิร์กที่เป็นแบบเชื่อมต่อส่งข้อมูลถึงกันทุก ๆ โนด (fully connected) มีการระยะเวลาคำนวณที่นานยิ่งขึ้นเช่นเดียวกัน

3. ปัญหาเรื่องค่าตัวแปร และค่าสัมประสิทธิ์ต่าง ๆ ของฟังก์ชันเคอร์เนล

เนื่องจากค่าตัวแปร และสัมประสิทธิ์ที่เกี่ยวข้องในฟังก์ชันเคอร์เนลสามารถเลือกค่านำมาใช้ได้หลากหลาย ทำให้ต้องเสียเวลาในการทดลองหาช่วงของค่าที่ให้ผลที่ออกมาดีที่สุด ซึ่งจากปัญหามิติของข้อมูลที่สูงขึ้นทำให้กินเวลาการทำงานนานขึ้นก็ส่งผลให้ไม่สามารถทำการทดลองกับค่าตัวแปร หรือสัมประสิทธิ์ได้มากค่าเท่าที่ควร

4. ปัญหาเรื่องค่าที่เกี่ยวข้องกับนิวรอลเน็ตเวิร์ก

เช่นเดียวกันกับค่าตัวแปร และสัมประสิทธิ์ต่าง ๆ ของฟังก์ชันเคอร์เนล ค่าที่เกี่ยวข้องภายในนิวรอลเน็ตเวิร์กก็มีให้เลือกอยู่จำนวนมาก ซึ่งการปรับค่าในแต่ละครั้งก็ต้องเสียเวลาในการทดลองที่ค่อนข้างนาน จึงต้องอาศัยค่าจากการทดลองในงานวิจัยก่อน ๆ มาปรับใช้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากผลการวิจัยที่ได้ หากเปรียบเทียบกันระหว่างการวิเคราะห์องค์ประกอบสำคัญแบบเดิม และแบบเคอร์เนล การวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลสามารถให้ผลลัพธ์ในการรู้จำที่ดีกว่าแบบเดิมทั้งรูปภาพในชุดทดสอบและชุดเรียนรู้ และเมื่อทำการทดสอบเปรียบเทียบกับโปรแกรมโอซีอาร์ภาษาไทยอ่านไทย 2.0 ก็สามารถให้ผลการรู้จำตัวอักษรที่ดีกว่าเช่นกัน อย่างไรก็ตามในการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลนั้น จำเป็นต้องใช้หน่วยความจำในการทำงานที่มากขึ้นกว่าเดิม รวมถึงกินระยะเวลาในการรู้จำที่มากกว่าการวิเคราะห์องค์ประกอบสำคัญแบบเดิม ทั้งนี้เนื่องมาจากการที่ต้องแมปข้อมูลไปยังปริภูมิข้อมูลที่มีมิติสูงกว่าเดิม รวมกับเวลาเสียไปในการคำนวณเคอร์เนล และจำนวนโนดนำเข้าที่เพิ่มขึ้นเพื่อใช้ในการรู้จำของนิวรอลเน็ตเวิร์ก ทำให้การวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลนั้นยังจำเป็นต้องปรับปรุงการทำงานเพิ่มเติม ทั้งในแง่ของการลดทรัพยากรของระบบ ความเร็วในการทำงาน และการปรับเปลี่ยนชนิดของเคอร์เนลที่ให้ผลการรู้จำที่ถูกต้องยิ่งขึ้น เพื่อให้โปรแกรมมีประสิทธิภาพโดยรวมในการทำงานได้ดียิ่งขึ้นต่อไป

5.2 ข้อเสนอแนะ

ในงานวิจัยนี้มีข้อเสนอแนะบางประการที่น่าจะนำมาทำการปรับปรุงประสิทธิภาพของโปรแกรมโอซีอาร์ภาษาไทยให้ดียิ่งขึ้นดังนี้

1. ทดลองปรับค่าสัมประสิทธิ์ หรือค่าอื่นที่เกี่ยวข้องต่าง ๆ เพิ่มเติมทั้งในส่วนของการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล และการแยกแยะตัวอักษรโดยใช้นิวรอลเน็ตเวิร์กแบบแคพรอพาเกชัน เพื่อหาค่าที่ให้ประสิทธิภาพในการรู้จำของโปรแกรมโอซีอาร์ภาษาไทยดียิ่งขึ้น
2. ปรับเปลี่ยนการใช้เคอร์เนลชนิดอื่น ๆ ในส่วนของการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล ที่อาจจะให้ผลการเรียนรู้ที่ดียิ่งขึ้น
3. ทดลองปรับใช้วิธีการแยกแยะตัวอักษรในลักษณะอื่น ๆ นอกเหนือจากนิวรอลเน็ตเวิร์ก
4. ปรับปรุงอัลกอริทึมในขั้นตอนต่าง ๆ เพื่อให้สามารถทำงานได้เร็วยิ่งขึ้น
5. ทดลองเพิ่มเติมลักษณะแบบตัวอักษร และขนาดตัวอักษรที่โปรแกรมสามารถทำการเรียนรู้ และรู้จำ ให้มีความหลากหลายมากยิ่งขึ้น

6. ปรับปรุงให้โปรแกรมโอซีอาร์ภาษาไทยสามารถรู้จำรูปภาพเอกสารที่ไม่ใช่แค่ลักษณะเอกสารที่มีตัวอักษรเพียงอย่างเดียวทั้งเอกสาร (plain text document) แต่สามารถรู้จำเอกสารที่ประกอบด้วยรูปภาพ หรือสัญลักษณ์อื่น ๆ ที่อยู่ภายในได้
7. ขยายขีดความสามารถของโปรแกรมโอซีอาร์ภาษาไทยในส่วนข้อมูลนำเข้า (input) และส่งออก (output) ของโปรแกรมให้สามารถทำการเปิดรูปภาพเอกสารในหลายลักษณะนอกเหนือจากรูปภาพบิตแมป (bitmap) ได้ และสามารถจัดเก็บข้อมูลรหัสตัวอักษรที่ได้ในลักษณะแฟ้มข้อมูลชนิดอื่น ๆ นอกเหนือจากเท็กซ์ไฟล์ (text file) ธรรมดาได้
8. เพิ่มเติมเทคนิควิธีการต่าง ๆ ในตัวโปรแกรมโอซีอาร์ให้มากยิ่งขึ้น เช่นสามารถกำหนดส่วนในการรู้จำบนเอกสารเฉพาะส่วนได้ สามารถอ่านเอกสารที่ไม่ใช่ตัวอักษรปกติ เช่นตัวหนาหรือตัวเอียง หรือรู้จำตัวอักษรในลักษณะที่เป็นลายมือได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] ชาญฤทธิ์ สันตินานาเลิศ. การออกแบบและพัฒนาโปรแกรมโอซีอาร์ภาษาไทย. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2542.

[2] Schölkopf, B. Support Vector Learning. Munich : Oldenbourg Verlag, 1997.

[3] จุฬารัตน์ ตันประเสริฐ. ซอฟท์แวร์ไทยโอซีอาร์ : พนักงานพิมพ์ดีดคนใหม่. ไมโครคอมพิวเตอร์ 153 : 117-123, เมษายน 2541.

[4] Casey, R.G., and Lecolinet, E. A Survey of Methods and Strategies in Character Segmentation. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 18, No. 7 : 690-706, July 1996.

[5] Vapnik, V. The Nature of Statistical Learning Theory. 2nd ed. New York : Springer-Verlag, 1999.

[6] Hyvärinen, A., Karhunen, J., and Oja, E. Independent Component Analysis. New York : John Wiley & Sons, 2001.

[7] Schölkopf, B., Burges, J.C., and Smola, A.J. Advances in Kernel Methods : Support Vector Learning. Cambridge : The MIT Press, 1998.

[8] Fu, L. Neural Networks in Computer Intelligence. Singapore : McGraw-Hill, 1994.

[9] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., et al. LAPACK Users' Guide. 3ed ed. Philadelphia : Siam, 1999.

[10] วิทยาศาสตร์ เทคโนโลยีและสิ่งแวดล้อม, กระทรวง. สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ. ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. ฝ่ายกลุ่มวิจัยและพัฒนาสาขาสารสนเทศ. อ่านไทย 2.0[โปรแกรมคอมพิวเตอร์]. กรุงเทพมหานคร : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, 2543.

[11] ดำรง ทิพย์โยธา และ เพ็ญพรรณ ยังคง. พีชคณิตเชิงเส้น. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร : โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2540.

[12] Jolliffe, I. Principal Component Analysis. 2nd ed. New York : Springer-Verlag, 2002.

[13] Tou, J.T., and Gonzalez, R.C. Pattern Recognition Principles. London : Addison-Wesley Publishing Company, 1974.

[14] Osuna, E.E., Freund, R., and Girosi, F. Support Vector Machines : Training and Applications. A.I. Memo No. 1602, C.B.C.L. Paper No. 144, March 1997.

[15] Burges, J.C. A Tutorial on Support Vector Machines for Pattern Recognition. Boston : Kluwer Academic Publishers, 1998.

[16] Cristianini, N., and Shawe-Taylor, J. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge : Cambridge University Press, 2000.

[17] Campell, C. An Introduction to Kernel Methods. United Kingdom : Department of Engineering Mathematics, Bristol University, 2000.

[18] Smola, A.J. Learning with Kernels. Berlin, 1998.

[19] วิทยาศาสตร์ เทคโนโลยีและสิ่งแวดล้อม, กระทรวง. สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ. ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. ฝ่ายกลุ่มวิจัยและพัฒนาสาขาสารสนเทศ. โอซีอาร์:จากภาพสู่ข้อความ. กรุงเทพมหานคร : บริษัทโปรลายน์ มีเดีย จำกัด, 2543.

[20] ธเนศ ศรีวิรุฬห์ชัย. การรู้จำตัวอักษรพิมพ์ภาษาไทยโดยใช้เทคนิคด้านการวิเคราะห์องค์ประกอบสำคัญและนิวรอลเน็ตเวิร์ค. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2541.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

การทดลองเลือกค่าสัมประสิทธิ์ฟังก์ชันเคอร์เนล

ในการทดลองเพื่อหาฟังก์ชันเคอร์เนล และค่าสัมประสิทธิ์ที่เกี่ยวข้องกับฟังก์ชันเคอร์เนลที่ให้ผลลัพธ์ในการรู้จำออกมาดีที่สุดนั้น ด้วยข้อจำกัดในเรื่องหน่วยความจำในการทดลองในช่วงแรก ทำให้ไม่สามารถใช้รูปภาพตัวอย่างในชุดเรียนรู้ทั้งหมดจำนวน 8,544 รูปมาทำการทดลองในทีเดียว ทำให้ต้องทำการแบ่งรูปภาพในชุดเรียนรู้ออกเป็น 4 ชุดคือ

- ชุดที่ 1 ประกอบด้วยรูปภาพขนาด 14 และ 16
- ชุดที่ 2 ประกอบด้วยรูปภาพขนาด 18 และ 20
- ชุดที่ 3 ประกอบด้วยรูปภาพขนาด 22 และ 24
- ชุดที่ 4 ประกอบด้วยรูปภาพขนาด 28 และ 36

รวมจำนวนรูปภาพในแต่ละชุดเป็นชุดละ 2,136 รูป แล้วจึงค่อยทำการทดลองแยกไปในแต่ละชุด โดยในขั้นแรกได้ทำการทดลองกับการวิเคราะห์องค์ประกอบสำคัญแบบเดิมเพื่อใช้ในการเทียบผล ได้ผลตามตารางที่ 6 โดยทำการวนในเน็ตเวิร์กจำนวนทั้งหมด 10 ล้านรอบ และเก็บผลเน็ตเวิร์กที่ให้ค่าผลการรู้จำดีที่สุดไว้

ตารางที่ 6 ผลการรู้จำรูปภาพตัวอักษรที่ผิดพลาดเมื่อใช้
วิธีการวิเคราะห์องค์ประกอบสำคัญแบบเดิม

ขนาดตัวอักษรของชุดรูปภาพที่ใช้ในการเรียนรู้	จำนวนโนดนำเข้าและโนดซ่อนของเน็ตเวิร์กที่ให้ผลลัพธ์ที่ดีที่สุด	จำนวนรูปภาพตัวอักษรที่รู้จำผิดพลาด (ตัว)	
		ข้อมูลชุดทดสอบ (1,424 ตัว)	ข้อมูลชุดเรียนรู้ (2,136 ตัว)
14 และ 16	250 และ 150	99	6
18 และ 20	250 และ 128	80	5
22 และ 24	250 และ 150	61	13
28 และ 36	150 และ 128	90	16

สำหรับวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลจะทำการแบ่งค่าสัมประสิทธิ์ออกเป็นช่วง ๆ และหาช่วงที่ให้ผลลัพธ์ในการรู้จำที่ดีที่สุด จากนั้นจึงนำช่วงนั้นมาทำการแบ่งเป็นช่วงย่อยเพิ่มขึ้น และหาช่วงย่อยที่ให้ผลลัพธ์ที่ดีที่สุดออกมา โดยท้ายที่สุดแล้วได้ทำการทดลองปรับค่าสัมประสิทธิ์สำหรับฟังก์ชันเคอร์เนลแต่ละชนิดไว้ดังนี้

- ฟังก์ชันเคอร์เนล Polynomial ทำการปรับค่าตัวแปรดีกรี d ที่เป็นจำนวนเต็ม ตั้งแต่ค่า 1 ไล่ไปเรื่อย ๆ ทีละหนึ่งจนถึง 6 แล้วหาค่าที่ให้ค่าคำตอบที่ผิดพลาดน้อยที่สุด
- ฟังก์ชันเคอร์เนล RBF ทำการปรับค่าตัวหาร c ที่เป็นค่าจำนวนจริงที่ไม่ใช่ศูนย์ หาค่าที่อยู่ในช่วงที่ให้ผลลัพธ์ที่ดีที่สุด โดยเริ่มจาก c =-1.0 ไปยัง –0.6, -0.4, -0.2, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.6, 0.8 และ 1.0
- ฟังก์ชันเคอร์เนลนิวรอลเน็ตเวิร์ก ปรับค่าตัวแปร κ และ Θ ในสมการ หาค่าที่ให้ผลลัพธ์ที่ผิดพลาดน้อยที่สุด โดยทำการปรับค่าตัวแปร κ ไล่จากค่า 0.1 ไปยัง 1, 2, 4, 5, 8, 10, 20, 40, 100 และคงค่า Θ ไว้ที่ 0

โดยในการปรับค่าสัมประสิทธิ์เพื่อหาค่าที่ดีที่สุดในแต่ละฟังก์ชันเคอร์เนลนั้นค่อนข้างจะเสียเวลาในการทดลองมาก ทั้งระยะเวลาในการหาค่าเมตริกซ์ $\mathbf{K}$ เมตริกซ์เวกเตอร์เจาะจง และเวกเตอร์รูปแบบในขั้นตอนการแยกลักษณะสำคัญของรูปภาพตัวอักษร จึงทำให้ไม่สามารถทำการทดลองกับทั้ง 4 ชุดของรูปภาพชุดเรียนรู้ที่แบ่งไว้ทั้งหมดได้ในฟังก์ชันเคอร์เนลบางค่าสัมประสิทธิ์ที่ให้ผลการรู้จำที่มีแนวโน้มที่ค่อนข้างไม่ดี โดยการทดลองในขั้นแรกนั้น จำนวนโนดนำเข้า และโนดซ่อนที่ใช้ทำการทดลองจะใช้ค่าเดียวกันกับของการวิเคราะห์องค์ประกอบสำคัญแบบเดิมก่อน คือ 128, 150 และ 250 โนด (ตามตารางที่ 2 ในบทที่ 3) เพื่อประหยัดเวลาในการทดลอง เพราะเพียงแค่ต้องการดูแนวโน้มช่วงของค่าสัมประสิทธิ์ที่ให้ผลการรู้จำที่ออกมาดีเท่านั้น

โดยผลของการรู้จำตัวอักษรผิดพลาดในฟังก์ชันเคอร์เนลแต่ละชนิด ในแต่ละค่าสัมประสิทธิ์ ที่เปรียบเทียบกันเฉพาะชุดรูปภาพที่ใช้ในการเรียนรู้ 2 ชุด คือชุดรูปภาพขนาด 14 และ 16 และชุดรูปภาพขนาด 18 และ 20 แสดงไว้ดังตารางที่ 7 และตารางที่ 8

จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 7 ผลของการรู้จำรูปภาพตัวอักษรที่ผิดพลาดในฟังก์ชันเคอร์เนลแต่ละชนิด เมื่อทำการเรียนรู้ด้วยชุดรูปภาพขนาด 14 และ 16

วิธีการวิเคราะห์องค์ประกอบสำคัญ	จำนวนโนดนำเข้าและโนดซ่อนของเน็ตเวิร์กที่ให้ผลลัพธ์ที่ดีที่สุด	จำนวนรูปภาพตัวอักษรที่รู้จำผิดพลาด (ตัว)	
		ข้อมูลชุดทดสอบ (1,424 ตัว)	ข้อมูลชุดเรียนรู้ (2,136 ตัว)
Polynomail ดีกรี d			
d = 1	250 และ 250	100	17
d = 2	250 และ 250	110	9
d = 3	250 และ 128	126	40
d = 4	250 และ 150	395	392
d = 5	250 และ 128	825	1,161
d = 6	250 และ 250	997	1,441
RBF ที่มีค่าตัวหาร c			
c = -1.0	250 และ 150	236	247
c = -0.6	128 และ 150	193	179
c = -0.4	128 และ 250	145	73
c = -0.2	250 และ 250	167	37
c = 0.2	150 และ 150	276	50
c = 0.25	150 และ 150	171	10
c = 0.3	150 และ 128	126	15
c = 0.35	150 และ 150	98	19
c = 0.4	150 และ 250	109	35
c = 0.45	250 และ 250	112	34
c = 0.5	250 และ 128	109	45
c = 0.6	128 และ 150	130	64
c = 0.8	250 และ 150	162	123
c = 1.0	128 และ 150	226	201

นิวรอลเน็ตเวิร์กที่มีค่าตัวคูณ κ			
κ = 0.1	250 และ 150	127	29
κ = 1	250 และ 150	146	12
κ = 2	150 และ 150	141	20
κ = 4	128 และ 150	145	19
κ = 5	128 และ 150	144	14
κ = 8	250 และ 150	138	14
κ = 10	250 และ 128	135	15
κ = 20	250 และ 150	123	16
κ = 40	150 และ 128	137	23
κ = 100	150 และ 128	151	19

ตารางที่ 8 ผลของการรู้จำรูปภาพตัวอักษรที่ผิดพลาดในฟังก์ชันเคอร์เนลแต่ละชนิด เมื่อทำการเรียนรู้ด้วยชุดรูปภาพขนาด 18 และ 20

วิธีการวิเคราะห์องค์ประกอบสำคัญ	จำนวนโนดนำเข้าและโนดซ่อนของเน็ตเวิร์กที่ให้ผลลัพธ์ที่ดีที่สุด	จำนวนรูปภาพตัวอักษรที่รู้จำผิดพลาด (ตัว)	
		ข้อมูลชุดทดสอบ (1,424 ตัว)	ข้อมูลชุดเรียนรู้ (2,136 ตัว)
Polynomail ดีกรี d			
d = 1	150 และ 150	83	18
d = 2	250 และ 250	89	6
d = 3	150 และ 128	111	38
d = 4	250 และ 128	334	339
d = 5	250 และ 150	746	1,013
d = 6	250 และ 250	976	1,369
RBF ที่มีค่าตัวหาร c			
c = -1.0	250 และ 250	183	201
c = -0.6	250 และ 250	120	79
c = -0.4	128 และ 250	118	68

c = -0.2	250 และ 250	233	155
c = 0.2	250 และ 150	182	28
c = 0.25	250 และ 150	109	10
c = 0.3	128 และ 250	97	20
c = 0.35	250 และ 128	98	23
c = 0.4	250 และ 250	98	35
c = 0.45	250 และ 128	95	28
c = 0.5	150 และ 128	94	34
c = 0.6	150 และ 128	110	60
c = 0.8	250 และ 128	125	97
c = 1.0	250 และ 150	173	146
นิวรอลเน็ตเวิร์กที่มีค่าตัวคูณ κ			
κ = 0.1	250 และ 128	104	13
κ = 1	250 และ 128	105	11
κ = 2	150 และ 150	102	13
κ = 4	128 และ 128	110	9
κ = 5	250 และ 150	102	15
κ = 8	250 และ 128	107	9
κ = 10	150 และ 128	110	18
κ = 20	128 และ 250	115	13
κ = 40	250 และ 150	100	13
κ = 100	250 และ 150	93	9

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

จากตารางที่ 7 และตารางที่ 8 จะพบว่าฟังก์ชันเคอร์เนลที่ให้ให้ผลการรู้จำที่ค่อนข้างดีเมื่อเทียบกับค่าสัมประสิทธิ์อื่น หรือเมื่อเทียบกับฟังก์ชันเคอร์เนลชนิดอื่นได้แก่ฟังก์ชันเคอร์เนล 4 แบบดังนี้คือ

- ฟังก์ชันเคอร์เนลแบบ Polynomial ที่มีค่าสัมประสิทธิ์ดีกรีเท่ากับ 1
- ฟังก์ชันเคอร์เนลแบบ Polynomial ที่มีค่าสัมประสิทธิ์ดีกรีเท่ากับ 2
- ฟังก์ชันเคอร์เนลแบบ RBF ที่มีค่าสัมประสิทธิ์ตัวหารเท่ากับ 0.3
- ฟังก์ชันเคอร์เนลแบบ RBF ที่มีค่าสัมประสิทธิ์ตัวหารเท่ากับ 0.35

ในการทดลองขั้นต่อมาจึงนำฟังก์ชันเคอร์เนลในแต่ละค่าสัมประสิทธิ์ทั้ง 4 แบบนี้มาทำการทดสอบเพิ่มเติมเพื่อให้ได้ผลการรู้จำที่ดียิ่งขึ้น โดยมีการปรับเปลี่ยนค่าที่เกี่ยวข้องกับนิวรอลเน็ตเวิร์กคือจำนวนโนดนำเข้าที่ปรับเพิ่มขึ้นเป็น 1,024, 1,200 และ 2,048 ที่ทดลองแล้วว่าให้ผลการรู้จำที่ดีขึ้นกว่าการใช้โนดนำเข้าจำนวน 128, 150 และ 250 แบบเดิม นอกจากนั้นยังได้ทำการวนรอบการทำงานของนิวรอลเน็ตเวิร์กเพิ่มขึ้นจาก 1 ล้านรอบมาอยู่ที่ 3 ล้านรอบ และทำการทดลองกับชุดรูปภาพทั้ง 4 ชุดทั้งหมด โดยผลการรู้จำผิดพลาดของฟังก์ชันเคอร์เนลทั้ง 4 แบบที่ทำการปรับค่าการทำงานเพิ่มเติมแล้ว แสดงผลได้ดังตารางที่ 9

ตารางที่ 9 ผลของการรู้จำตัวอักษรผิดพลาดน้อยที่สุดในฟังก์ชันเคอร์เนล 4 แบบที่เลือกไว้เมื่อทำการปรับค่าการทำงานเพิ่มเติม

ขนาดตัวอักษรของชุดรูปภาพที่ใช้ในการเรียนรู้	จำนวนรูปภาพตัวอักษรที่รู้จำผิดพลาด (ตัว)							
	ข้อมูลชุดทดสอบ (1,424 ตัว)				ข้อมูลชุดเรียนรู้ (2,136 ตัว)			
	Polynomial		RBF		Polynomial		RBF	
	ดีกรี 1	ดีกรี 2	ตัวหาร = 0.3	ตัวหาร = 0.35	ดีกรี 1	ดีกรี 2	ตัวหาร = 0.3	ตัวหาร = 0.35
14 และ 16	89	96	99	94	16	17	12	14
18 และ 20	80	78	73	75	19	17	15	11
22 และ 24	54	62	58	62	16	10	13	8
28 และ 36	93	89	74	78	17	9	8	10

อย่างไรก็ตามผลของการรู้จำโดยใช้ฟังก์ชันเคอร์เนลในแต่ละค่าสัมประสิทธิ์ทั้ง 4 รูปแบบยังให้ผลการรู้จำที่ไม่ดีไปกว่าการวิเคราะห์องค์ประกอบสำคัญแบบเดิมในบางกรณี โดยเฉพาะจำนวนตัวอักษรที่รู้จำผิดพลาดในข้อมูลรูปภาพชุดเรียนรู้ ประกอบกับในการทดลองช่วงหลังได้มีการเปลี่ยนไปทำการทดลองในเครื่องที่มีทรัพยากรของระบบเพิ่มมากขึ้น ทำให้สามารถทำการทดลองกับรูปภาพตัวอักษรชุดเรียนรู้ 8,544 รูปได้ในคราวเดียว จึงได้ทำการนำฟังก์ชันเคอร์เนลในแต่ละค่าสัมประสิทธิ์ทั้ง 4 รูปแบบมาทำการทดลองใหม่กับรูปภาพตัวอักษรในชุดเรียนรู้ทั้งหมดและเปรียบเทียบกับการวิเคราะห์องค์ประกอบสำคัญแบบเดิมใหม่อีกครั้ง โดยในการทดลองครั้งสุดท้ายนี้ยังได้มีการปรับค่าโนดนำเข้าเพิ่มขึ้นจากเดิมอีกเป็น 4,096 และ 5,120 โนดเพื่อให้ผลการรู้จำที่ดียิ่งขึ้น ซึ่งได้ผลการรู้จำเปรียบเทียบสุดท้ายก็ได้แสดงไว้ในบทที่ 4 ตามลำดับ

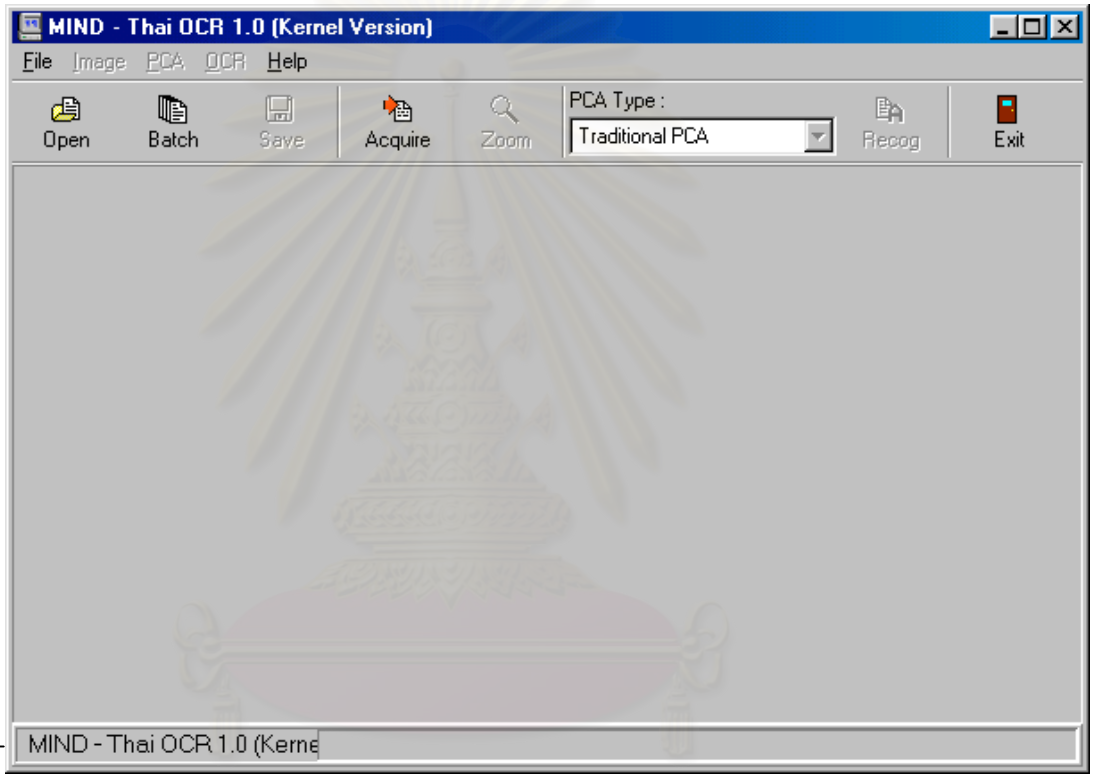
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

การใช้งานโปรแกรมไทยโอซีอาร์

ข.1 ส่วนประกอบ และคำสั่งต่าง ๆ ในโปรแกรมไทยโอซีอาร์

เมื่อเรียกโปรแกรมไทยโอซีอาร์ขึ้นมาใช้งาน โปรแกรมจะเริ่มต้นการทำงานด้วยหน้าจอหลัก ตามรูปที่ 13



รูปที่ 13 หน้าจอหลักของโปรแกรมไทยโอซีอาร์

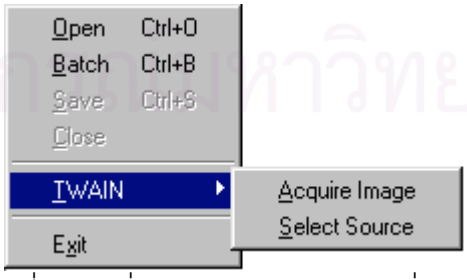
หน้าจอหลักประกอบด้วยส่วนต่างๆ 3 ส่วน คือ

1. ส่วนรายการคำสั่งและปุ่มคำสั่ง เป็นส่วนที่ใช้ในการออกคำสั่งเพื่อให้โปรแกรมไทยโอซีอาร์ทำงานตามที่ต้องการ ซึ่งอยู่ด้านบนสุดของหน้าจอหลัก
2. ส่วนแสดงภาพเอกสาร เป็นบริเวณที่แสดงภาพของเอกสารที่ผู้ใช้เปิดขึ้นมาจากแฟ้มข้อมูลภาพหรือภาพของเอกสารที่ผู้ใช้สแกนเข้ามา เพื่อเตรียมทำการรู้จำตัวอักษร
3. ส่วนแสดงสถานะ เป็นส่วนที่แสดงสถานะการทำงานของโปรแกรม

ข.1.1 รายการคำสั่ง

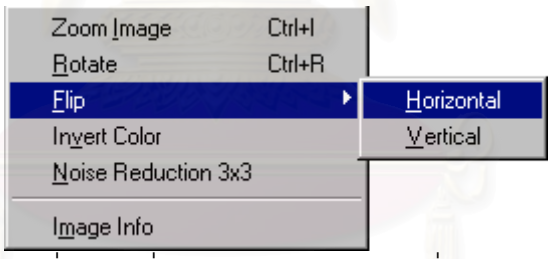
ประกอบด้วยรายการคำสั่งหลัก 5 รายการ (รายการที่มีสีเทา หมายถึงรายการคำสั่งที่ไม่สามารถใช้งานได้ในขณะนั้น จะต้องทำการเปิดแฟ้มข้อมูลภาพหรือสแกนภาพเอกสารก่อนจึงจะใช้คำสั่งเหล่านั้นได้) คือ

- รายการคำสั่ง File (รูปที่ 14) เป็นรายการคำสั่งที่เกี่ยวข้องกับการเปิดแฟ้มข้อมูลภาพและการสแกนภาพ ซึ่งมีคำสั่งย่อยดังนี้
  1. คำสั่ง Open เป็นคำสั่งที่ใช้ในการเปิดแฟ้มข้อมูลภาพที่ได้ทำการสแกน และจัดเก็บไว้ก่อนหน้านั้นแล้ว (สามารถเปิดแฟ้มข้อมูลภาพแบบ BMP ที่มีสีขาว-ดำหรือภาพโทนสีเทา 256 ระดับ เท่านั้น)
  2. คำสั่ง Batch เป็นคำสั่งที่ใช้ในการเปิดแฟ้มข้อมูลภาพที่ได้ทำการสแกน และจัดเก็บไว้แล้วขึ้นมาทำการรู้จำทีละหลาย ๆ แฟ้มข้อมูลในคราวเดียว (โปรแกรมจะทำการรู้จำแฟ้มข้อมูลไปทีละแฟ้ม)
  3. คำสั่ง Save เป็นคำสั่งที่ใช้ในการบันทึกแฟ้มข้อมูลภาพที่เปิดใช้งานอยู่ (สามารถบันทึกแฟ้มข้อมูลภาพเป็นแบบ BMP ที่มีสีขาว-ดำ เท่านั้น)
  4. คำสั่ง Close เป็นคำสั่งที่ใช้ในการปิดแฟ้มข้อมูลภาพที่เปิดใช้งานอยู่
  5. คำสั่ง TWAIN เป็นคำสั่งที่เกี่ยวข้องกับการใช้เครื่องสแกนเนอร์ ซึ่งแบ่งเป็น 2 คำสั่งย่อยคือ
     - 5.1 คำสั่ง Acquire Image เป็นคำสั่งที่ใช้ในการสแกนภาพจาก data source ที่กำหนดไว้ เป็นต้น
     - 5.2 คำสั่ง Select Source เป็นคำสั่งที่ใช้ในการกำหนด data source ที่ต้องการ (โดยมาก data source ที่กำหนดคือ driver ของสแกนเนอร์)
     - 5.3 คำสั่ง Exit เป็นคำสั่งที่ใช้เมื่อต้องการออกจากโปรแกรมไทยโอซีอาร์



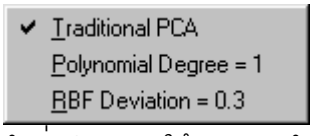
รูปที่ 14 คำสั่งย่อยภายใต้รายการคำสั่ง File

- รายการคำสั่ง Image (รูปที่ 15) เป็นรายการคำสั่งที่เกี่ยวข้องกับการประมวลผลรูปภาพ เพื่อปรับลักษณะของภาพเพื่อให้เหมาะสมกับการรู้จำ ซึ่งมีคำสั่งย่อยดังนี้
  1. คำสั่ง Zoom Image เป็นคำสั่งที่ใช้ในการแสดงภาพของเอกสารให้มีขนาดภาพตามขนาดของเอกสารจริง (actual view) หรือแสดงภาพของเอกสารให้เห็นเอกสารทั้งฉบับ (whole image view)
  2. คำสั่ง Rotate เป็นคำสั่งที่ใช้ในการหมุนภาพของเอกสาร
  3. คำสั่ง Flip เป็นคำสั่งที่ใช้ในการกลับภาพ ซึ่งมีการกลับภาพใน 2 ลักษณะ คือ กลับภาพในแนวนอน และ กลับภาพในแนวตั้ง
  4. คำสั่ง Invert Color เป็นคำสั่งที่ใช้ในการกลับสีของจุดภาพจากดำเป็นขาว และจากขาวเป็นดำ
  5. คำสั่ง Noise Reduction 3x3 เป็นคำสั่งที่ใช้ในการลดจุดภาพรบกวนที่มีขนาด 1 จุดภาพ (pixel)
  6. คำสั่ง Image Info เป็นคำสั่งที่ใช้ในการแสดงรายละเอียดของภาพเอกสาร เช่น ชื่อแฟ้มรูปภาพ, ขนาดความกว้างและความยาว (pixel), และค่าของจุดสีที่ใช้ในแฟ้มรูปภาพที่กำลังเปิดใช้งานอยู่



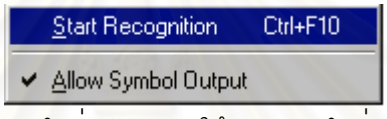
รูปที่ 15 คำสั่งย่อยภายใต้รายการคำสั่ง Image

- รายการคำสั่ง PCA (รูปที่ 16) เป็นรายการคำสั่งที่ใช้ในการเลือกวิธีของการวิเคราะห์องค์ประกอบสำคัญที่จะใช้ในการดึงค่าลักษณะสำคัญของรูปภาพเอกสาร โดยจะมีเครื่องหมาย "ถูก" อยู่หน้าชนิดของวิธีที่เลือกอยู่ ซึ่งมีให้เลือกอยู่ 3 ตัวเลือกด้วยกัน คือ
  1. Traditional PCA คือวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเดิม
  2. Polynomial Degree = 1 คือวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนลที่มีชนิดของฟังก์ชันเคอร์เนลเป็นแบบ Polynomial ดีกรีเท่ากับ 1
  3. RBF Deviation = 0.3 คือวิธีการวิเคราะห์องค์ประกอบสำคัญแบบเคอร์เนล ที่มีชนิดของฟังก์ชันเคอร์เนลเป็นแบบ Radial Basic Function ซึ่งใช้ค่าสัมประสิทธิ์ตัวหารเท่ากับ 0.3



รูปที่ 16 คำสั่งย่อยภายใต้รายการคำสั่ง PCA

- รายการคำสั่ง OCR (รูปที่ 17) เป็นรายการคำสั่งที่เกี่ยวข้องกับการรู้จำตัวอักษร มีคำสั่งย่อยดังนี้
  1. คำสั่ง Start Recognition เป็นคำสั่งที่ใช้เพื่อให้โปรแกรมเริ่มทำการรู้จำภาพเอกสารที่เปิดใช้งานอยู่
  2. คำสั่ง Allow Symbol Output เป็นตัวเลือกว่าจะอนุญาตให้การรู้จำนั้นมีผลลัพธ์ที่เป็นเครื่องหมายที่ไม่ค่อยได้ใช้หรือไม่ (ในโปรแกรมนี้คือเครื่องหมาย $? และ !) ซึ่งหากมีเครื่องหมาย "ถูก" อยู่หน้าตัวเลือกนี้จะเป็นการอนุญาตให้แสดงเครื่องหมายได้ครบทุกตัว

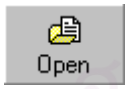


รูปที่ 17 คำสั่งย่อยภายใต้รายการคำสั่ง OCR

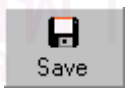
- รายการคำสั่ง Help มีคำสั่งย่อยคือคำสั่ง About ที่เป็นคำสั่งแสดงรายละเอียดของโปรแกรมนี้

ข.1.2 ปุ่มคำสั่ง และรายการตัวเลือกด่วน

เพื่อความสะดวกของผู้ใช้โปรแกรม จึงได้นำเอาคำสั่งต่างๆ จากรายการคำสั่งที่ใช้งานบ่อยมาแสดงในรูปของปุ่มคำสั่ง ประกอบด้วยปุ่มคำสั่ง 7 ปุ่ม และรายการตัวเลือกด่วน 1 รายการ ดังนี้

- 

ปุ่มคำสั่ง Open ทำหน้าที่เหมือนคำสั่ง Open ในรายการคำสั่ง File
-

ปุ่มคำสั่ง Batch ทำหน้าที่เหมือนคำสั่ง Batch ในรายการคำสั่ง File
- 

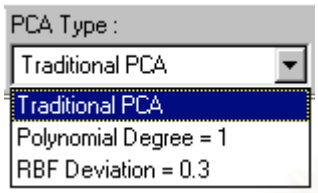
ปุ่มคำสั่ง Save ทำหน้าที่เหมือนคำสั่ง Save ในรายการคำสั่ง File
-

ปุ่มคำสั่ง Acquire ทำหน้าที่เหมือนคำสั่ง Acquire Image ในรายการคำสั่ง File – TWAIN
-

ปุ่มคำสั่ง Zoom ทำหน้าที่เหมือนคำสั่ง Zoom Image ในรายการคำสั่ง Image

-

ปุ่มคำสั่ง Recog ทำหน้าที่เหมือนคำสั่ง Start Recognition ในรายการคำสั่ง OCR
-

ปุ่มคำสั่ง EXIT ทำหน้าที่เหมือนคำสั่ง Exit ในรายการคำสั่ง File
- 

รายการตัวเลือกด่วน PCA Type ทำหน้าที่ใช้ในการเลือกวิธีของการวิเคราะห์องค์ประกอบสำคัญเช่นเดียวกับตัวเลือกในรายการคำสั่ง PCA

ข.1.3 ส่วนแสดงภาพเอกสาร

จะเป็นส่วนที่แสดงภาพของเอกสารที่เตรียมจะทำการรู้จำ หากภาพของเอกสารมีขนาดกว้างกว่าส่วนแสดงภาพ ผู้ใช้สามารถเลื่อนแถบ scroll bar เพื่อดูภาพในบริเวณอื่น หรือเลื่อน mouse pointer ไว้เหนือรูปจนเห็น mouse pointer เปลี่ยนเป็นรูป ✥ แล้วกดปุ่มซ้ายของ mouse ค้างไว้แล้วเลื่อน mouse ไปตามทิศทางที่ต้องการแทนก็ได้

หากต้องการดูภาพเอกสารทั้งภาพให้กดที่ปุ่ม Zoom และกดอีกครั้งเพื่อกลับมาแสดงภาพตามขนาดเอกสารจริง

ข.1.4 ส่วนแสดงสถานะ

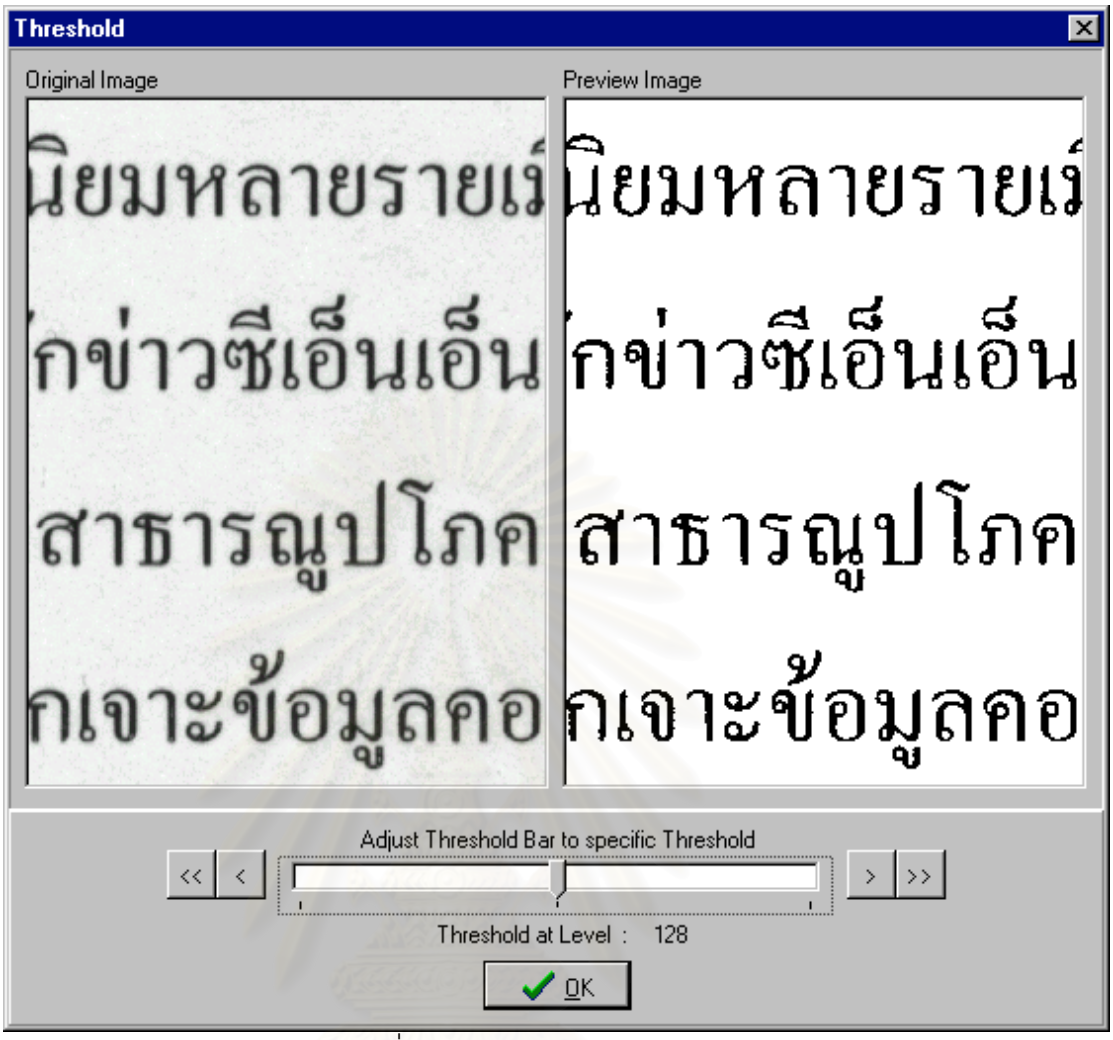
จะเป็นส่วนที่แสดงสถานะการทำงาน ขณะที่กำลังทำการรู้จำเท่านั้น

ข.2 การใช้งานโปรแกรมไทยโอซีอาร์

1. เปิดแฟ้มข้อมูลภาพหรือสแกนภาพเอกสารจากเครื่องสแกนเนอร์ หากภาพที่เปิดหรือสแกนเป็นภาพโทนสีเทา 256 ระดับ โปรแกรมจะแสดงหน้าจอ Threshold เพื่อให้ผู้ใช้กำหนดค่า Threshold ตามรูปที่ 18 ผู้ใช้สามารถกำหนดค่าของ Threshold ได้โดยเลื่อน track bar หรือ กดที่ปุ่มปรับค่า

เสร็จแล้วกดปุ่ม

(หากต้องดูภาพของเอกสารในบริเวณอื่นให้เลื่อน mouse pointer ไว้เหนือรูป จะเห็น mouse pointer เปลี่ยนเป็นรูป ✥ แล้วกดปุ่มซ้ายของ mouse ค้างไว้แล้วเลื่อน mouse ไปตามทิศทางที่ต้องการ)



รูปที่ 18 หน้าจอ Threshold

2. ภาพของเอกสารจะปรากฏขึ้นในส่วนแสดงภาพเอกสาร ซึ่งมีขนาดภาพตามขนาดของเอกสารจริง หากผู้ใช้ต้องการให้โปรแกรมทำการรู้จำ ก็สามารถกดปุ่ม Recog เพื่อเริ่มทำการรู้จำในขณะนี้ก็ได้ โดยสามารถเลือกชนิดของการวิเคราะห์องค์ประกอบสำคัญตามที่ผู้ใช้ต้องการจากรายการคำสั่ง PCA หรือเลือกจากรายการด่วนบนแถบเครื่องมือได้เช่นเดียวกัน หรือหากผู้ใช้ต้องการทำการประมวลผลภาพก็สามารถเลือกคำสั่งได้ในรายการคำสั่ง Image ซึ่งโปรแกรมจะทำการประมวลผลภาพตามคำสั่งทันที ยกเว้นการหมุนภาพ (rotate) ซึ่งจะปรากฏหน้าจอ Rotate ตามรูปที่ 19

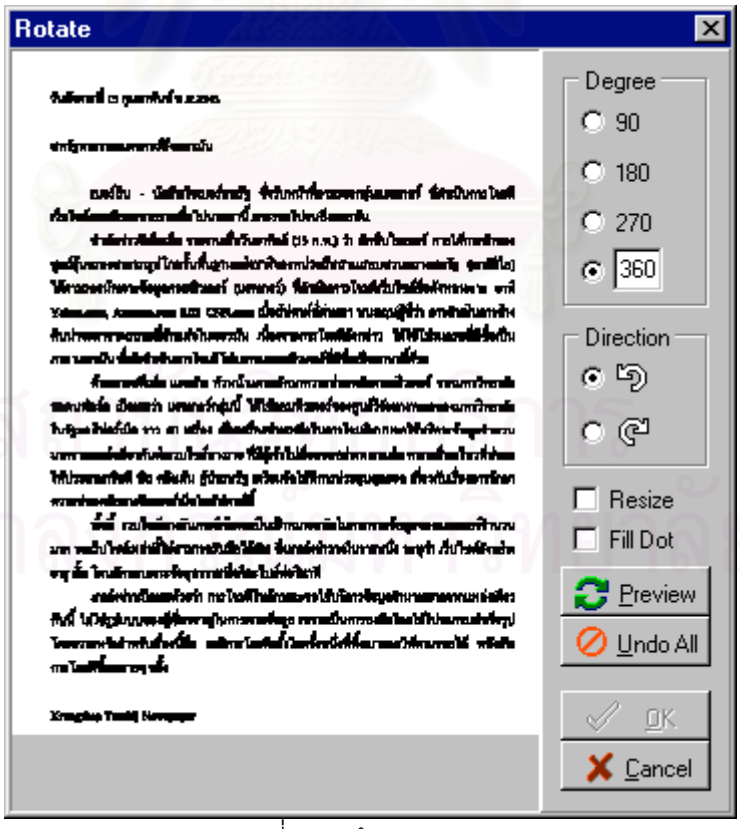
โดยผู้ใช้สามารถกำหนดองศาความเอียงและทิศทางการหมุนภาพ เมื่อกำหนดแล้วให้กดปุ่ม เพื่อดูเอกสารหลังจากทำการหมุนภาพแล้ว หากต้องการยกเลิกการหมุนให้กดปุ่ม หรือผู้ใช้สามารถกำหนดองศาความเอียงและทิศทางการหมุนภาพ โดยเลื่อน mouse pointer ไว้เหนือรูปจนเห็น mouse

pointer เปลี่ยนเป็นรูป + แล้วกดปุ่มซ้ายของ mouse ค้างไว้แล้วเลื่อน mouse ไปตามเส้นบรรทัดที่เอียง แล้วปล่อยปุ่มซ้าย โปรแกรมจะคำนวณองศาความเอียงและทิศทางการหมุนให้

การหมุนภาพของเอกสารในบางครั้งอาจทำให้ข้อความในเอกสารตกขอบ จึงต้องให้ผู้ใช้กำหนดว่าจะขยายขนาดของเอกสารหรือไม่ หากผู้ใช้ต้องการให้ขยายขนาดของเอกสารจากการหมุน ผู้ใช้จะต้องเลือกกล่องที่อยู่หน้าตัวเลือก Resize ให้มีเครื่องหมาย "ถูก" นำหน้า โปรแกรมจะทำการคำนวณขนาดของเอกสารให้ใหม่

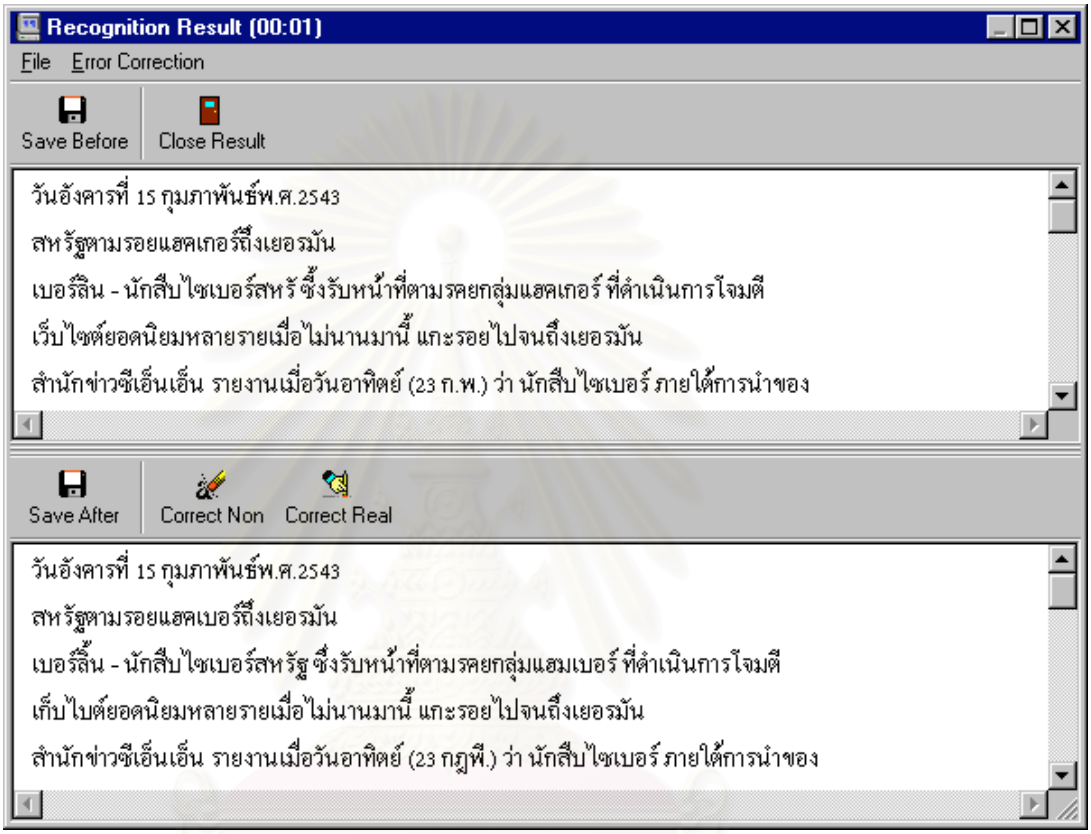
และในบางครั้งการหมุนภาพของเอกสารอาจทำให้จุดดำบางจุดหายไป (มักเกิดขึ้นในกรณีที่หมุนภาพที่มีองศาความเอียงมากๆ) จึงต้องมีการให้ค่าจุดดำคืนแก่เอกสาร โดยผู้ใช้จะต้องเลือกกล่องที่อยู่หน้าตัวเลือก Fill Dot ให้มีเครื่องหมาย "ถูก" นำหน้า โปรแกรมจะทำการคำนวณหาจุดดำที่หายไป และจะแสดงจุดดำที่คำนวณได้นั้นคืนมา

เมื่อได้ภาพที่ตรงตามความต้องการแล้วให้กดปุ่ม เพื่อยอมรับการหมุนภาพของเอกสาร หรือกดปุ่ม เพื่อยกเลิกการหมุนภาพเอกสาร



รูปที่ 19 หน้าจอ Rotate

3. เมื่อผู้ใช้ทำการประมวลผลภาพ หรือเลือกชนิดของวิธีวิเคราะห์องค์ประกอบสำคัญของข้อมูลจนเป็นที่พอใจแล้ว สามารถสั่งให้โปรแกรมเริ่มทำการรู้จำ โดยการกดปุ่ม Recog โปรแกรมจะทำการรู้จำตัวอักษรทั้งหมดในเอกสาร แล้วจะแสดงผลการรู้จำเป็นข้อความด้านบนของหน้าจอ Recognition Result ตามรูปที่ 20



รูปที่ 20 หน้าจอ Recognition Result โดยที่ข้อความด้านบนแสดงผลลัพธ์ของการรู้จำ และข้อความด้านล่างแสดงผลลัพธ์ของการแก้ไขคำผิดจากผลลัพธ์ของการรู้จำ

ผู้ใช้สามารถจัดเก็บผลการรู้จำนี้ โดยกดปุ่ม

หรือใช้คำสั่ง Save Text Before Correction ภายใต้รายการคำสั่ง File หรือหากพบว่ามีคำผิดอยู่ในผลลัพธ์ของการรู้จำ ผู้ใช้สามารถสั่งให้โปรแกรมทำการแก้ไขคำผิดจากผลลัพธ์ของการรู้จำได้ 2 วิธี คือ

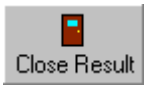
- ปุ่ม

เป็นการแก้คำผิดเฉพาะคำผิดที่ไม่เป็นคำ
- ปุ่ม

เป็นการแก้คำผิดที่ไม่เป็นคำและคำผิดที่เป็นคำ

ซึ่งสามารถจัดเก็บผลลัพธ์ของการแก้ไขคำผิดนี้ โดยกดปุ่ม

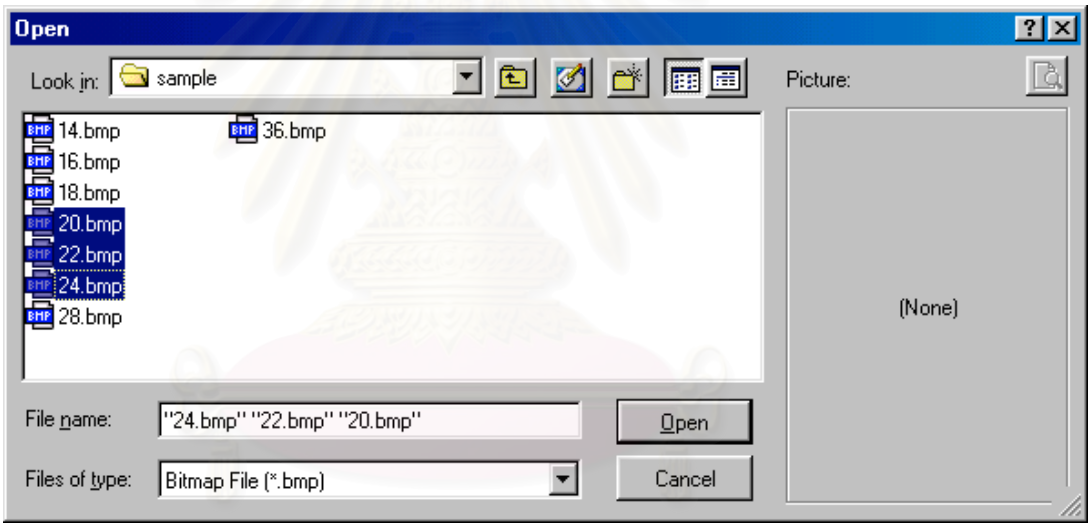
หรือใช้คำสั่ง Save Text After Correction ภายใต้รายการคำสั่ง File

และสามารถออกจากหน้าจอ Recognition Result โดยกดปุ่ม 

หรือใช้คำสั่ง Close Result ภายใต้รายการคำสั่ง File

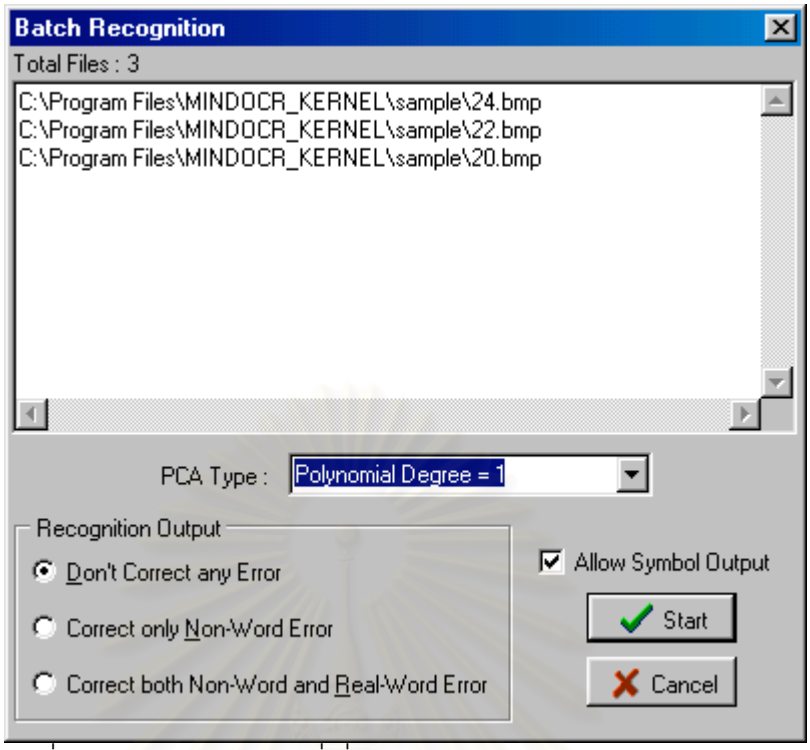
4. ในกรณีที่ผู้ใช้ต้องการทำการรู้จำเอกสารครั้งละหลายภาพ ผู้ใช้สามารถทำการสั่งให้โปรแกรมทำการรู้จำภาพเอกสารที่ต้องการทั้งหมดได้ในคราวเดียวโดยใช้คำสั่ง Batch ภายใต้รายการคำสั่ง File หรือกดปุ่ม

ที่รายการปุ่มตัวเลือกด่วน ซึ่งจะได้หน้าจอที่ให้ผู้ใช้สามารถเลือกแฟ้มรูปภาพเอกสารที่ต้องการรู้จำครั้งละหลายภาพได้ ดังรูปที่ 21



รูปที่ 21 หน้าจอเลือกรูปภาพที่ต้องการรู้จำทีละหลายภาพเอกสาร

หลังจากที่ผู้ใช้ทำการเลือกรูปภาพเอกสารที่ต้องการรู้จำจนเป็นที่พอใจแล้ว โปรแกรมจะเข้าสู่หน้าจอดังรูปที่ 22 เพื่อให้ผู้ใช้กำหนดชนิดของการวิเคราะห์องค์ประกอบสำคัญ และค่าผลลัพธ์ของการรู้จำที่ต้องการของทุก ๆ แฟ้มรูปภาพที่ทำการรู้จำ



รูปที่ 22 หน้าจอกำหนดค่าที่เกี่ยวข้องในการรู้จำทีละหลายแฟ้มรูปภาพ

เมื่อผู้ใช้กำหนดค่าที่เกี่ยวข้องในการรู้จำเรียบร้อยแล้ว ก็สามารถให้โปรแกรมเริ่มทำการรู้จำโดยการกดปุ่ม

แล้วโปรแกรมก็จะทำการรู้จำแฟ้มรูปภาพที่เลือกไว้ไปทีละรูปจนหมดทุกรูป โดยเก็บผลลัพธ์ของการรู้จำไว้ที่โฟลเดอร์เดียวกับตำแหน่งที่เก็บรูปภาพ

ข.3 แฟ้มการทำงานที่เกี่ยวข้องกับโปรแกรมไทยโอซีอาร์

- แฟ้มโปรแกรมการทำงานที่ใช้ในโปรแกรมไทยโอซีอาร์

1. mocr.exe เป็นแฟ้มโปรแกรมการทำงานของโปรแกรมไทยโอซีอาร์
2. vcl50.bpl เป็นแฟ้มไลบรารีที่ใช้ทำงานร่วมกับโปรแกรมไทยโอซีอาร์
3. vclx50.bpl เป็นแฟ้มไลบรารีที่ใช้ทำงานร่วมกับโปรแกรมไทยโอซีอาร์
4. cc3250mt.dll เป็นแฟ้มไลบรารีที่ใช้ทำงานร่วมกับโปรแกรมไทยโอซีอาร์
5. correct_tocr.dll เป็นแฟ้มไลบรารีที่ใช้ทำงานร่วมกับโปรแกรมไทยโอซีอาร์เกี่ยวกับการแก้คำผิดภาษาไทย

- แฟ้มข้อมูลที่ใช้ในโปรแกรมไทยโอซีอาร์ (อยู่ใต้ไดเรคทอรี่ย่อย Dat)

1. Weight.bpn เป็นแฟ้มข้อมูลที่เก็บค่าน้ำหนัก (weight) และค่าผันแปร (bias) ของนิวรอลเน็ตเวิร์ก
2. Eig.Finished เป็นแฟ้มข้อมูลที่เก็บเมตริกซ์เวกเตอร์เจาะจง
3. Mean.Finished เป็นแฟ้มข้อมูลที่เก็บเมตริกซ์ค่าเฉลี่ย
4. Npat.txt เป็นแฟ้มข้อมูลที่เก็บค่าของโนดทางออก (output node)
5. Confuse.prb เป็นแฟ้มข้อมูลที่เก็บค่าสถิติความผิดพลาดของโปรแกรมไทยโอซีอาร์แบบ unigram ของตัวอักษร
6. Confuse_gr.prb เป็นแฟ้มข้อมูลที่เก็บค่าสถิติความผิดพลาดของโปรแกรมไทยโอซีอาร์ แบบ trigram ของตัวอักษร (อักษรก่อนหน้า, อักษรปัจจุบัน และอักษรถัดไป)
7. Confuse_gr-2n.prb เป็นแฟ้มข้อมูลที่เก็บค่าสถิติความผิดพลาดของโปรแกรมไทยโอซีอาร์ แบบ bigram ของตัวอักษร (อักษรปัจจุบัน และ อักษรถัดไป)
8. Confuse_gr-2p.prb เป็นแฟ้มข้อมูลที่เก็บค่าสถิติความผิดพลาดของโปรแกรมไทยโอซีอาร์ แบบ bigram ของตัวอักษร (อักษรก่อนหน้า และ อักษรปัจจุบัน)
9. Pos.tri เป็นแฟ้มข้อมูลที่เก็บค่าสถิติ trigram ของประเภทของคำ (POS trigram) ใช้เป็น แบบจำลองทางภาษา (language model)
10. Trie.out เป็นแฟ้มข้อมูลพจนานุกรม และประเภทของคำ
11. Char.tri เป็นแฟ้มข้อมูลที่เก็บค่าสถิติของตัวอักษรที่เรียงต่อกัน แบบ trigram ใช้ในการหาคำที่ไม่รู้จัก
12. train.amount เป็นแฟ้มข้อมูลที่เก็บค่าจำนวนตัวอย่างที่ใช้ในการเรียนรู้
13. train.images เป็นแฟ้มข้อมูลที่เก็บค่าเวกเตอร์รูปภาพในชุดเรียนรู้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

การใช้งานโปรแกรมต่าง ๆ ในขั้นตอนการเรียนรู้

ค.1 การสร้างแฟ้มข้อมูล Npat.txt

แฟ้มข้อมูล Npat.txt เป็นแฟ้มข้อมูลที่เก็บค่าของโนดทางออก (output node) ซึ่งมีรูปแบบดังนี้

เลขที่ของโนดทางออกตามด้วย tab ตามด้วย ตัวอักษรที่เป็นค่าของโนดทางออกนั้น ตามด้วย tab ตามด้วยชื่อของตัวอักษรนั้น เช่น

1	ก	kai
2	ข	khai
3	ฃ	khud
4	ค	khay
5	ฅ	khon
6	ฆ	rakung
7	ง	ngoo
	:	
	:	

จากตัวอย่าง หมายถึงโนดทางออกที่ 1 มีค่าเท่ากับ ก และมีชื่อตัวอักษรเป็นภาษาอังกฤษว่า kai เป็นต้น

การสร้างแฟ้มข้อมูล Npat.txt นี้ สามารถสร้างได้ โดยใช้โปรแกรมประมวลผลคำทั่ว ๆ ไป แล้วจัดเก็บแบบ Text File แต่ช่องว่างระหว่างข้อมูลจะต้องคั่นด้วย tab เท่านั้น ซึ่งแฟ้มข้อมูล Npat.txt นั้น จะนำไปใช้งานทั้งในการเรียนรู้และการรู้จำด้วย

ค.2 การสร้างแฟ้มข้อมูล Npat.dbm

แฟ้มข้อมูล Npat.dbm คือแฟ้มข้อมูล Npat.txt ที่ทำการจัดเก็บให้อยู่ในรูปแบบที่โปรแกรม Trn_vec.exe จะนำไปใช้งานได้ (Trn_vec.exe เป็นโปรแกรมที่ใช้ในการเรียนรู้ เพื่อหาเมตริกซ์ค่าเฉลี่ยและเมตริกซ์เวกเตอร์เจาะจง ซึ่งจะกล่าวถึงต่อไป)

การสร้าง แฟ้มข้อมูล Npat.dbm จะใช้โปรแกรม GenNpat.exe เป็นตัวสร้าง โดยมีการใช้งานดังนี้

GenNpat Npat.txt Npat.dbm

ค.3 การหาเมตริกซ์ค่าเฉลี่ยและเมตริกซ์เวกเตอร์เจาะจง (Mean.Finished และ Eig.Finished)

การสร้าง แฟ้มข้อมูล Mean.Finished และ Eig.Finished จะใช้โปรแกรม k_trn_vec.exe เป็นตัวสร้าง โดยมีการใช้งานดังนี้

k_trnvec -I [Options]

ซึ่ง Options ต่างๆ มีดังนี้

-f ชื่อแฟ้มข้อมูล ระบุชื่อแฟ้มข้อมูลซึ่งในแต่ละบรรทัดภายในแฟ้มข้อมูลเก็บรายชื่อแฟ้มข้อมูลภาพ ชื่อแฟ้มข้อมูลภาพภายในแฟ้มข้อมูลนี้จะต้องอยู่ในรูปแบบดังนี้

xxxxxxyy.bmp

เมื่อ xxxxxx คือชื่อตัวอักษร (ต้องตรงกับชื่อตัวอักษรใน Npat.txt) เช่น KAI, KHAI เป็นต้น

และ yy คือขนาดของตัวอักษร เช่น 14, 16 เป็นต้น

ตัวอย่างชื่อแฟ้มข้อมูลภาพ เช่น

train/angsana/thai/deg/deg14.bmp

-o ชื่อแฟ้มข้อมูล ระบุชื่อแฟ้มข้อมูลซึ่งจะเป็นผลลัพธ์ของการเรียนรู้

-k จำนวน ระบุจำนวนตัวประกอบสำคัญ (default = 128)

-a ชนิดของเคอร์เนล [ค่าตัวแปรที่เกี่ยวข้อง]

ระบุชนิดของฟังก์ชันเคอร์เนลที่ต้องการใช้ในการเรียนรู้ และค่าตัวแปรที่เกี่ยวข้องตามชนิดของเคอร์เนลที่เลือก โดยชนิดของเคอร์เนลที่ระบุได้แบ่งเป็น

1 คือ ฟังก์ชัน Polynomial

2 คือ ฟังก์ชันนิวรอลเน็ตเวิร์ก

3 คือ ฟังก์ชัน Radial Basic Function

ตัวอย่างการใช้งาน เช่น แฟ้มข้อมูลที่ใช้เก็บรายชื่อแฟ้มข้อมูลภาพชื่อ train.lst และต้องการให้ผลลัพธ์เก็บในแฟ้มข้อมูลชื่อ train.out และต้องการจำนวนตัวประกอบสำคัญเท่ากับ 250 สามารถเรียกโปรแกรมให้ทำงานได้ดังนี้

k_trn_vec -I –a 3 0.3 -f train.lst -o train.out -k 1024

โปรแกรมจะทำการคำนวณ และทำการเก็บข้อมูลเมตริกซ์ค่าเฉลี่ย และข้อมูลเมตริกซ์เวกเตอร์เจาะจงในแฟ้มข้อมูลชื่อ Mean.Finished และ Eig.Finished ตามลำดับ และทำการ

วิเคราะห์องค์ประกอบสำคัญภาพที่อยู่ในรายชื่อภายในแฟ้มข้อมูล train.lst ด้วยฟังก์ชันเคอร์แนลแบบ Radial Basic Function ที่มีค่าสัมประสิทธิ์ตัวหารในสูตรเท่ากับ 0.3 แล้วเก็บผลที่ได้ในแฟ้มข้อมูล train.out และสร้างแฟ้มข้อมูล train.df ซึ่งจะถูกนำไปใช้ในขั้นตอนการหาค่าน้ำหนัก และค่าผันแปรต่อไป

ค.4 การหาค่าน้ำหนักและค่าผันแปร (Network.bpn และ Weight.bpn)

การสร้าง แฟ้มข้อมูล Network.bpn และ Weight.bpn จะใช้โปรแกรม Trn_net.exe และ Gen_wght.exe เป็นตัวสร้างตามลำดับ โดยมีการใช้งานดังนี้

trn_net -l [Options]

ซึ่ง Options ต่างๆ มีดังนี้

-a ค่าอัตราการเรียนรู้ ระบุค่าอัตราการเรียนรู้ (default = 0.2)

-i ค่าอินเนอร์เชีย ระบุค่าอินเนอร์เชีย (default = 0.95)

-t จำนวนรอบ จำนวนตัวอย่าง ค่าผิดพลาด
ระบุจำนวนรอบที่ทำการสอนก่อนทำการทดสอบการลู่เข้า (convergence) จำนวนตัวอย่างที่ต้องผ่านการทดสอบ และค่าผิดพลาดที่ยอมรับได้

-s จำนวนรอบ ระบุจำนวนรอบที่ทำการสอนก่อนทำการเก็บข้อมูลของเน็ตเวิร์กลงในแฟ้มข้อมูลชื่อ Network.save

-d ชื่อแฟ้มข้อมูล ระบุชื่อแฟ้มข้อมูลซึ่งเก็บข้อมูลชื่อแฟ้มข้อมูลที่เป็นข้อมูลสำหรับป้อนให้นิวรอลเน็ตเวิร์ก เช่น train.df

-l จำนวนรอบ ระบุจำนวนรอบที่มากที่สุดที่ทำการสอน

-AsciiDumpNoFmt > ชื่อแฟ้มข้อมูล
ระบุชื่อแฟ้มข้อมูลซึ่งเก็บผลลัพธ์ของการเรียนรู้ เช่น Network.bpn

ตัวอย่างการใช้งาน เช่น

train_net -l -d train.df -a 0.08 -t 10000 1424 0.5 -s 100000 -l 1000000 -AsciiDumpNoFmt > Network.bpn

โปรแกรมจะทำการคำนวณผลลัพธ์ค่าน้ำหนัก (weight) และค่าผันแปร (bias) จากการเรียนรู้แล้วบันทึกลงใน Network.bpn ซึ่งจะต้องนำมาแปลงรูปแบบเพื่อให้สามารถนำไปใช้ในโปรแกรมไทยโอซีอาร์ได้ โดยใช้ Gen_wght.exe เป็นตัวสร้าง โดยมีการใช้งานดังนี้

gen_wght [Options]

ซึ่ง Options ต่างๆ มีดังนี้

-i จำนวน	ระบุจำนวนโนดนำเข้าจะต้องมีค่าตรงกับจำนวนตัวประกอบสำคัญ หรือจำนวนโนดนำเข้าของ Trn_net.exe
-h จำนวน	ระบุจำนวนโนดซ่อนของ Trn_net.exe
-o จำนวน	ระบุจำนวนโนดทางออกของ Trn_net.exe
-a ชื่อแฟ้มข้อมูล	ระบุชื่อแฟ้มข้อมูลซึ่งเก็บผลลัพธ์ของการเรียนรู้ เช่น Network.bpn
-t ชื่อแฟ้มข้อมูล	ระบุชื่อแฟ้มข้อมูลซึ่งเก็บผลลัพธ์ของการแปลงรูปแบบ ที่จะนำไปใช้ในโปรแกรมไทยโอซีอาร์ เช่น Weight.bpn

ตัวอย่างการใช้งาน เช่น

gen_wght -i 250 -h 150 -o 178 -a Network.bpn -t Weight.bpn

โปรแกรมจะทำการแปลงผลลัพธ์ค่าน้ำหนักและค่าผันแปร ให้อยู่ในรูปแบบที่โปรแกรมไทยโอซีอาร์นำไปใช้ได้

ซึ่งจากการทำงานของโปรแกรมทั้ง 2 โปรแกรม สรุปได้ว่าโปรแกรมจะทำการอ่านแฟ้มข้อมูล train.df และ train.out (train.out ถูกระบุภายในข้อมูลของแฟ้มข้อมูล train.df) และทำการเรียนรู้ด้วยค่าอัตราการเรียนรู้ 0.08 ทำการทดสอบการคอนเวอร์เจนส์ทุกๆ 100,000 รอบ การทดสอบต้องผ่าน 1,424 ตัวอย่างโดยมีค่าความผิดพลาดน้อยกว่า 0.5 และทำการเก็บข้อมูลของเน็ตเวิร์กทุกๆ 100,000 รอบ และจะทำการเรียนรู้ไม่เกิน 1,000,000 รอบ

โปรแกรม Trn_net.exe นั้นสร้างขึ้นมาจากเครื่องมือทางนิวรอลเน็ตเวิร์กคือ Aspirin/MIGRAINES 6.0 โดยกำหนดให้มีโครงสร้างให้มีจำนวนโนดนำเข้า โนดซ่อน และโนดทางออกได้ตามต้องการ

ภาคผนวก ง
ตัวอักษรที่ใช้ในการเรียนรู้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ AngsanaUPC ขนาด 14 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อี อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ AngsanaUPC ขนาด 16 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อี อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ AngsanaUPC ขนาด 18 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อี อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ AngsanaUPC ขนาด 20 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ AngsanaUPC ขนาด 22 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ AngsanaUPC ขนาด 24 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ AngsanaUPC ขนาด 28 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X

Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ AngsanaUPC ขนาด 36 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ

ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S

T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x

y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’

” < , > . ? /

ตัวอักษรแบบ BrowalliaUPC ขนาด 14 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ BrowalliaUPC ขนาด 16 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ BrowalliaUPC ขนาด 18 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ BrowalliaUPC ขนาด 20 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ BrowalliaUPC ขนาด 22 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ BrowalliaUPC ขนาด 24 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ BrowalliaUPC ขนาด 28 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V

W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > .

? /

ตัวอักษรแบบ BrowalliaUPC ขนาด 36 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ

ฤ ฦ ๆ ๚

A B C D E F G H I J K L M N O P Q

R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v

w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7

8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ;

“ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ CordiaUPC ขนาด 14 จุด

ก ข ฃ ค ฅ ฆ ง จ ฉ ช 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อๆ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ CordiaUPC ขนาด 16 จุด

ก ข ฃ ค ฅ ฆ ง จ ฉ ช 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อๆ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ CordiaUPC ขนาด 18 จุด

ก ข ฃ ค ฅ ฆ ง จ ฉ ช 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อๆ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ CordiaUPC ขนาด 20 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ CordiaUPC ขนาด 22 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ CordiaUPC ขนาด 24 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ CordiaUPC ขนาด 28 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X

Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ CordiaUPC ขนาด 36 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ

ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S

T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v

w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8

9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’

” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ DilleniaUPC ขนาด 14 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ็ อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ DilleniaUPC ขนาด 16 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ็ อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ DilleniaUPC ขนาด 18 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ็ อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ DilleniaUPC ขนาด 20 จุด

ก ข ฃ ค ฅ ฆ ง จ ฉ ช 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ DilleniaUPC ขนาด 22 จุด

ก ข ฃ ค ฅ ฆ ง จ ฉ ช 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ DilleniaUPC ขนาด 24 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ DilleniaUPC ขนาด 28 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W

X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ?

/

ตัวอักษรแบบ DilleniaUPC ขนาด 36 จุด

ก ข ฃ ค ฅ ฆ ง จ ฉ ช ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ

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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ

ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R

S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v

w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “

‘ ’ ” < , > . ? /

ตัวอักษรแบบ EucrosiaUPC ขนาด 14 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ EucrosiaUPC ขนาด 16 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ EucrosiaUPC ขนาด 18 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ EucrosiaUPC ขนาด 20 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ EucrosiaUPC ขนาด 22 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ EucrosiaUPC ขนาด 24 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y
Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ − + = | \ { [}] : ; “ ‘ ’ ” < , > . ?
/

ตัวอักษรแบบ EucrosiaUPC ขนาด 28 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V
W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ − + = | \ { [}] : ; “ ‘ ’ ” <
, > . ? /

ตัวอักษรแบบ EucrosiaUPC ขนาด 36 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ
ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q
R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v
w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6
7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}]
: ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ FreesiaUPC ขนาด 14 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ FreesiaUPC ขนาด 16 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ FreesiaUPC ขนาด 18 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9
! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ FreesiaUPC ขนาด 20 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ FreesiaUPC ขนาด 22 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอักษรแบบ FreesiaUPC ขนาด 24 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < , > . ? /

ตัวอักษรแบบ FreesiaUPC ขนาด 28 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S T U V W X
Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6 7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] : ; “ ‘ ’ ” < ,
> . ? /

ตัวอักษรแบบ FreesiaUPC ขนาด 36 จุด

ก ข ฃ ค ฅ 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 อิ อี อึ อื อุ อู อ่ อ้ อ๊ อ๋ อ็ อั อ์ เ โ ใ ไ

อำ ฤ ฦ ๆ ฯ

A B C D E F G H I J K L M N O P Q R S

T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v

w x y z

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ 0 1 2 3 4 5 6

7 8 9

! @ # $ % ^ & * ( ) _ - + = | \ { [}] :

; “ ‘ ’ ” < , > . ? /

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นาย พัฒนชัย เบศรภิญโญวงศ์ เกิดเมื่อวันที่ 7 กุมภาพันธ์ พ.ศ.2521 ที่กรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต (วท.บ.) สาขาวิชาวิทยาการคอมพิวเตอร์ จากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2541 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2542

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย