บทที่ 2

ระบบการรู้จำลายตัวอักษรไทย

2.1. ระบบการรู้จำลายตัวอักษรไทย

การรู้จำลายตัวอักษรไทย เป็นการวิเคราะห์ภาพที่ได้จากการ scan ซึ่งเป็นข้อมูลภาพของตัวอักษรมาเรียงต่อกันเป็นกลุ่มคำหรือประโยค เพื่อให้สามารถวิเคราะห์ได้ว่าภาพนั้น ๆ ประกอบด้วยตัวอักษรอะไรบ้าง โดยข้อมูลภาพที่รับเข้ามาจะอยู่ในรูปของจุดภาพ จะถูกนำมาทำการแก้ไขปรับปรุงข้อมูลภาพให้อยู่ในรูปแบบที่สะดวกต่อการนำไปใช้งาน คือทำการเก็บข้อมูลภาพแต่ละจุดภาพโดยใช้ข้อมูล 1 จุดภาพเก็บเป็นข้อมูล 1ไบต์(byte) จากนั้นทำการกำจัดสัญญาณรบกวนและกำหนดขนาดของภาพให้เหลือพื้นที่ขอบของภาพเพียง 1 จุดภาพ เมื่อได้ภาพที่ผ่านขั้นตอนการปรับปรุงภาพแล้ว ขั้นตอนต่อไปเป็นขั้นตอนการแยกข้อมูลภาพเหล่านั้น ออกเป็นตัวอักษรทีละตัว โดยใช้วิธีการหาขอบของตัวอักษร ผลลัพธ์ที่ได้จากการหาขอบของตัวอักษร เป็นตำแหน่งของตัวอักษรแต่ละตัวว่าอยู่ที่ตำแหน่งใด พร้อมทั้งมีเลขดัชนีบอกว่าตัวอักษรตัวนั้นเป็นตัวอักษรในระดับใด และตัวเลขบอกถึงจำนวนของกลุ่มรบกวนที่เกิดจากตัวอักษรข้างเคียงและตำแหน่งของกลุ่มรบกวนว่าอยู่ที่ตำแหน่งใด แต่ผลลัพธ์ที่ได้จากการหาขอบของตัวอักษร จะมีการจัดเรียงที่ไม่ถูกต้อง เพราะการหาขอบจะพิจารณาจุดภาพจากบนลงล่าง ทำให้ตัวอักษรที่อยู่ในระดับบนถูกพิจารณาก่อน ตามด้วยตัวอักษรในระดับกลางและระดับล่างตามลำดับ ซึ่งหากนำเอาผลลัพธ์ที่ได้จากขั้นตอนนี้ไปใช้ในขั้นตอนการรู้จำตัวอักษร จะทำให้ผลลัพธ์ที่ได้ไม่ถูกตรงตามมาตรฐานการพิมพ์ในภาษาไทย จึงมีขั้นตอนการจัดเรียงตัวอักษร เพื่อทำการจัดผลลัพธ์ที่ได้จากการหาขอบให้มีความถูกต้อง โดยพิจารณาจากตำแหน่งของตัวอักษรแต่ละตัวที่ได้จากการหาขอบ เมื่อทำการจัดเรียงตัวอักษรเรียบร้อยแล้ว จึงนำผลลัพธ์ที่ได้จากการจัดเรียงตัวอักษรป้อนเข้าสู่การรู้จำตัวอักษรไทย มีผู้ทำการวิจัยและศึกษากันมามากมายหลายวิธี และแต่ละวิธีมีข้อดีข้อเสียแตกต่างกันออกไป โดยการศึกษาเรื่องการรู้จำตัวอักษรมุ่งเน้นที่ความถูกต้องและรวดเร็ว ซึ่งวิธีการที่นำมาใช้ในงานวิจัยครั้งนี้คือวิธีซินแทกติก(Syntactic Method) ซึ่งเป็นวิธีที่ค่อนข้างจะซับซ้อน แต่มีความถูกต้องแม่นยำสูง โดยนำเอาตัวอักษรที่ต้องการรู้จำไปเปรียบเทียบกับตัวอักษรต้นแบบที่ได้เก็บไว้ สำหรับผลลัพธ์ที่ได้จากการรู้จำ หากเป็นตัวอักษรที่สามารถรู้จำได้จะถูกแทนด้วยค่าที่ถูกกำหนดไว้แล้ว ในตัวอักษรต้นแบบของตัวอักษรนั้น ๆ แต่ถ้าเป็นตัวอักษรที่ไม่สามารถวิเคราะห์ได้ว่าเป็นตัวอักษรอะไร จะถูกแทนด้วยเครื่องหมาย "-" เพื่อให้รู้ว่าเป็นตัวอักษรที่ไม่สามารถวิเคราะห์ได้ ซึ่งอาจเกิดจากขั้นตอนการหาขอบของตัวอักษรหรือการรู้จำตัวอักษรไม่สามารถหาค่าได้ หลังจากที่ได้ผ่านขั้นตอนการเปรียบเทียบตัวอักษร

ต้นแบบแล้ว จะมีการแก้ไขตัวอักษรบางตัวเพื่อที่จะได้เป็นการลดขนาดของผลลัพธ์ที่ได้ และเพื่อให้มีความถูกต้องตามรหัสภาษาไทยที่ได้กำหนดไว้

รูปที่ 2.1. ระบบการรู้จำสายตัวอักษรไทย

2.2. การปรับปรุงภาพ

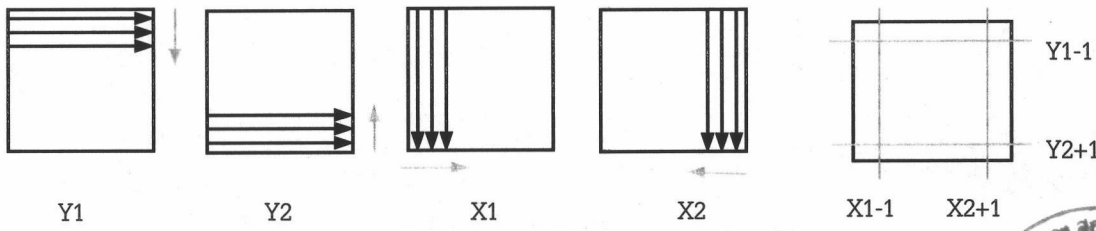
ข้อมูลภาพที่ได้จากการใช้เครื่องสแกนเนอร์(Scanner) จะถูกเก็บเป็นแฟ้มข้อมูล ซึ่งมีได้หลายรูปแบบ เช่น รูปแบบ BMP, รูปแบบ PCX เป็นต้น โดยระดับความเข้มของภาพมีอยู่หลายระดับ เช่น ระดับความเข้ม 256 ระดับ , 16 ระดับ และ 2 ระดับ ในงานวิจัยนี้มุ่งเน้นที่จะใช้ข้อมูลภาพที่มีระดับความเข้มเพียง 2 ระดับ จึงต้องมีการปรับระดับให้เหลือเพียง 2 ระดับโดยการกำหนดค่า Threshold[3] ให้กับภาพ ซึ่งมีค่าเท่ากับครึ่งหนึ่งของระดับความเข้มสูงสุด โดยค่าความเข้มของจุดภาพใด ๆ เป็นดังนี้

ความเข้มของจุดภาพ = { "1" เมื่อความเข้มของจุดภาพมากกว่าค่า Threshold ; "0" เมื่อความเข้มของจุดภาพน้อยกว่าค่า Threshold }

การเก็บข้อมูลภาพที่ได้ใหม่นี้จะเก็บโดยใช้ข้อมูลภาพ 1 จุดภาพเก็บเป็นข้อมูล 1 ไบต์(byte) ทำให้มีความสะดวกในการนำไปใช้งานมากกว่าเพราะการเก็บข้อมูลภาพ 1 จุดภาพเป็นข้อมูล 1 บิต(bit) ทำให้เสียเวลาในการประมวลผลมากกว่า

ข้อมูลภาพของตัวอักษรที่ผ่านการปรับระดับความเข้มให้เหลือเพียง 2 ระดับ มักจะมีจุดภาพที่ไม่ต้องการปะปนเข้ามา หรืออาจมีบางจุดภาพที่ขาดหายไป เป็นผลมาจากการที่อุปกรณ์นั้นมีประสิทธิภาพไม่ดีพอ หรือเกิดจากภาพต้นแบบที่ใช้ไม่สมบูรณ์ จึงต้องมีการปรับปรุงคุณภาพของข้อมูลภาพโดยใช้วิธีการกำจัดสัญญาณรบกวน[2] แก้ไขภาพที่เกิดช่องว่างขึ้นมาเป็นรู(Hold) และที่เกิดเป็นจุดเดี่ยว (Isolated point)

การกำหนดขนาดภาพเป็นการลดขนาดพื้นที่บริเวณขอบของภาพที่ไม่มีข้อมูลที่เป็นจุดภาพอยู่เพื่อให้ภาพมีขนาดเล็กลง ทำให้การประมวลผลเร็วขึ้น และเนื้อที่ที่ใช้เก็บข้อมูลน้อยลงด้วย ในการตรวจสอบจะทำการหาจุดภาพในขอบภาพด้านนั้น ๆ ทั้ง 4 ด้าน เมื่อพบจุดภาพก็จะเก็บตำแหน่งจุดนั้นเป็นขนาดของภาพ ดังรูปที่ 2.2



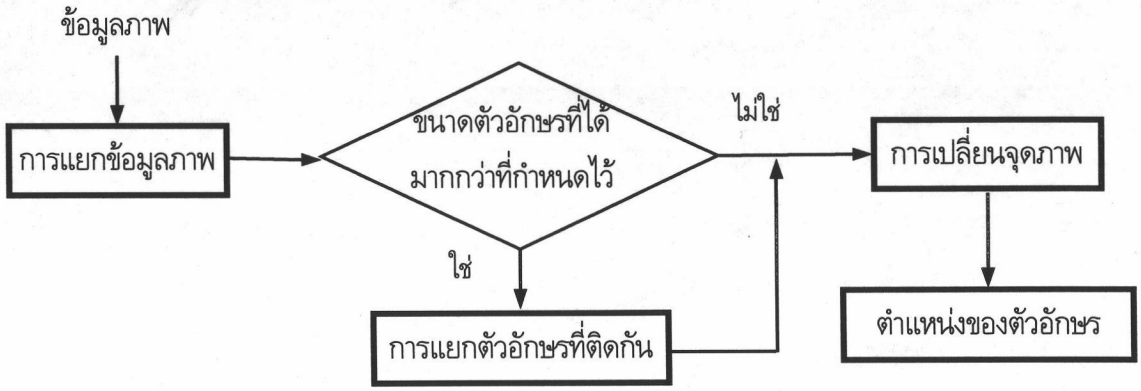
รูปที่ 2.2 ทิศทางในการตรวจสอบเพื่อปรับขนาดของภาพ

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

2.3. การแยกข้อมูลภาพออกเป็นข้อมูลของตัวอักษรเดี่ยว ๆ

เนื่องจากข้อมูลภาพที่ได้มายังไม่สามารถนำไปใช้ในการรู้จำได้ เพราะว่าไม่มีการบอกถึงตำแหน่งของตัวอักษรแต่ละตัวว่าอยู่ที่ใด จึงต้องมีการหาว่าข้อมูลภาพเหล่านั้นมีตัวอักษรอยู่ที่ตำแหน่งใดบ้าง และนำตำแหน่งเหล่านี้ไปใช้สำหรับการรู้จำตัวอักษร โดยการดึงเอาข้อมูลของตัวอักษรแต่ละตัวออกมาตามตำแหน่งที่หาได้ เพื่อที่มาทำการรู้จำตัวอักษร

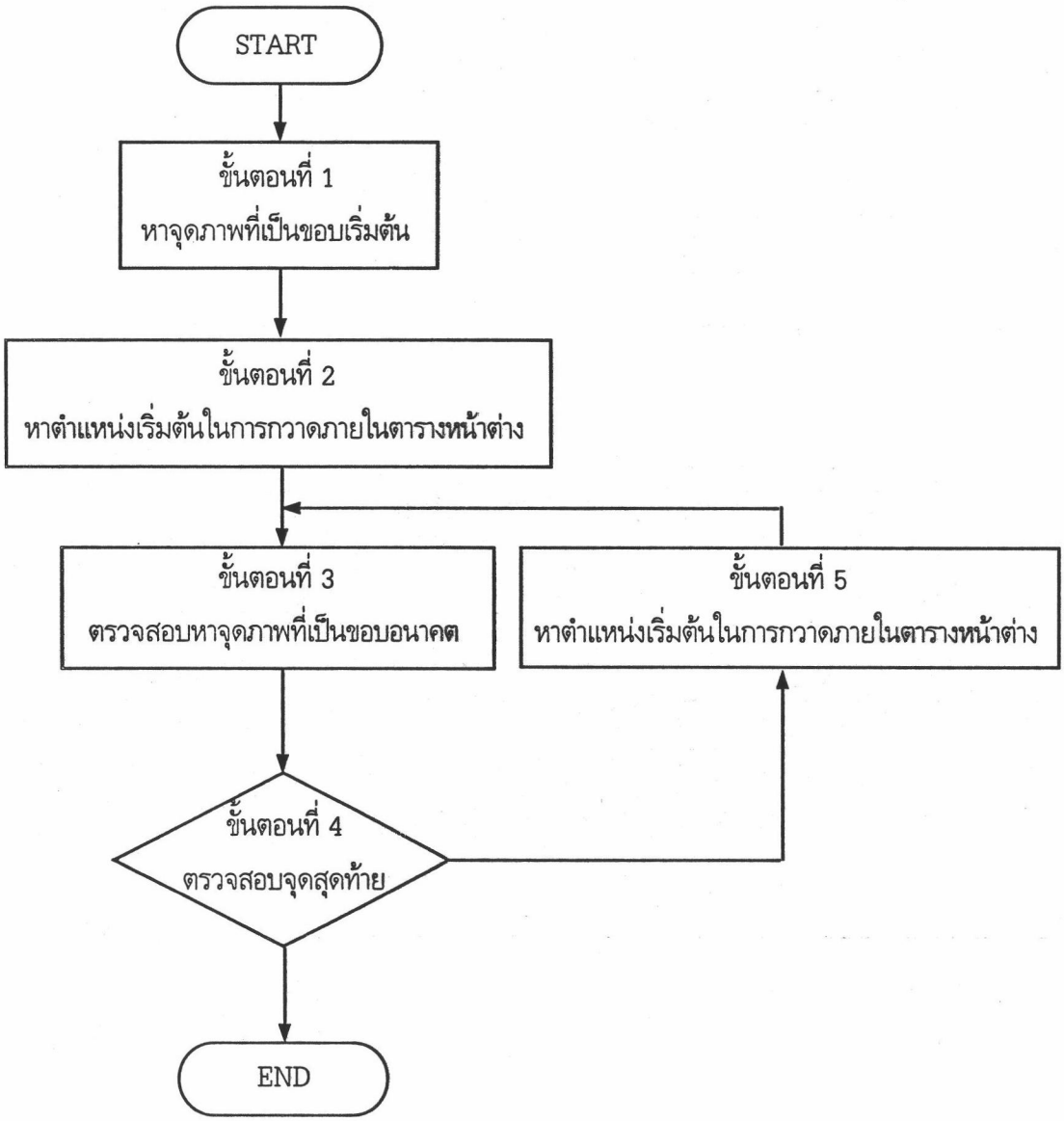
สำหรับขั้นตอนในการแยกข้อมูลภาพออกเป็นข้อมูลของตัวอักษรเดี่ยว ๆ นั้น พิจารณาจากโครงสร้างในรูปที่ 2.3 ข้อมูลภาพที่รับเข้ามาจะถูกทำการแยกออกเป็นตัวอักษรแต่ละตัว โดยวิธีการหาขอบของตัวอักษร จากนั้นก็จะตรวจสอบว่าตัวอักษรที่แยกออกมานั้นเป็นตัวอักษรที่ติดกันหรือไม่ ถ้าติดกันก็จะถูกนำไปเข้าขั้นตอนการแยกตัวอักษรที่ติดกัน โดยผลลัพธ์ที่ได้จะเป็นตำแหน่งของตัวอักษรแต่ละตัว จากนั้นจึงนำไปตรวจสอบหาจุดรบกวนจากจุดภาพข้างเคียงว่ามีหรือไม่ และทำการเปลี่ยนจุดภาพที่ได้นี้ให้เป็นค่าอื่นเพื่อที่จะได้หาตำแหน่งของตัวอักษรตัวต่อไป



รูปที่ 2.3 แสดงโครงสร้างของการแยกข้อมูลภาพออกเป็นตัวอักษรเดี่ยว ๆ

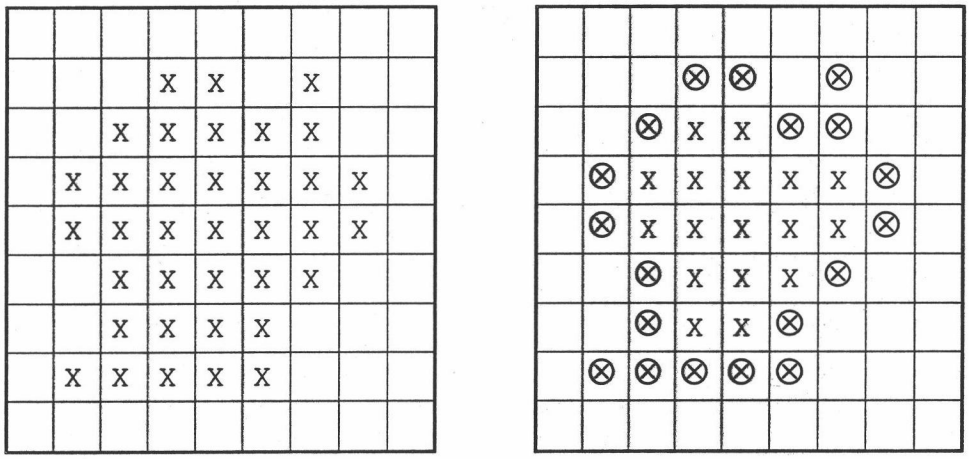
2.3.1. การแยกข้อมูลภาพ ข้อมูลภาพที่รับเข้ามา จะถูกทำการแยกออกให้เป็นข้อมูลภาพของตัวอักษรแต่ละตัว โดยผลลัพธ์ที่ได้จากการแยกข้อมูลภาพเป็นตำแหน่งของตัวอักษรแต่ละตัว ซึ่งการแยกข้อมูลภาพนั้นจะใช้วิธีการหาขอบของตัวอักษร

2.3.1.1. การหาขอบของตัวอักษร ทำให้ได้ตำแหน่งของตัวอักษรออกมา ซึ่งตำแหน่งที่ได้ประกอบด้วยขอบทั้ง 4 ด้านของตัวอักษรตัวนั้น วิธีการหาขอบของตัวอักษรมีอยู่หลายวิธี ในงานวิจัยนี้ใช้วิธีการหาขอบของตัวอักษรโดยใช้ตารางหน้าต่างขนาด 3x3 [4] ซึ่งสามารถอธิบายและเขียนเป็นแผนผัง (Flow Chart) ได้ดังรูปที่ 2.4



รูปที่ 2.4 ขั้นตอนการหาขอบของภาพโดยการใช้ตารางหน้าต่าง

จากภาพวัตถุที่ได้จำลองขึ้นมาดังรูปที่ 2.5(ก) จะได้ผลลัพธ์ดังแสดงในรูปที่ 2.5(ข) โดย X แทนจุดภาพที่เป็นวัตถุ และ ⊗ แทนจุดขอบภาพที่หาได้จากวิธีการหาขอบของตัวอักษร



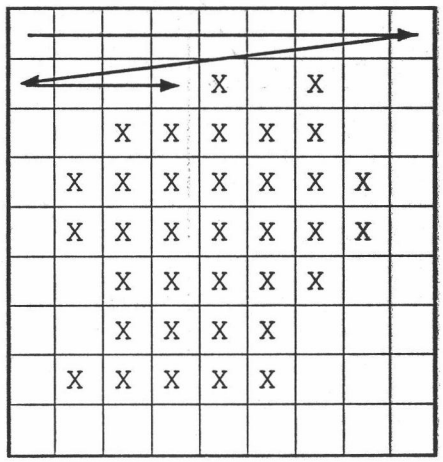
(ก) (ข)

รูปที่ 2.5 แสดงภาพตัวอย่างและผลลัพธ์ที่ได้จากการใช้วิธีการหาขอบภาพของตัวอักษร

(ก) ภาพตัวอย่างที่ใช้ในการหาขอบของวัตถุ

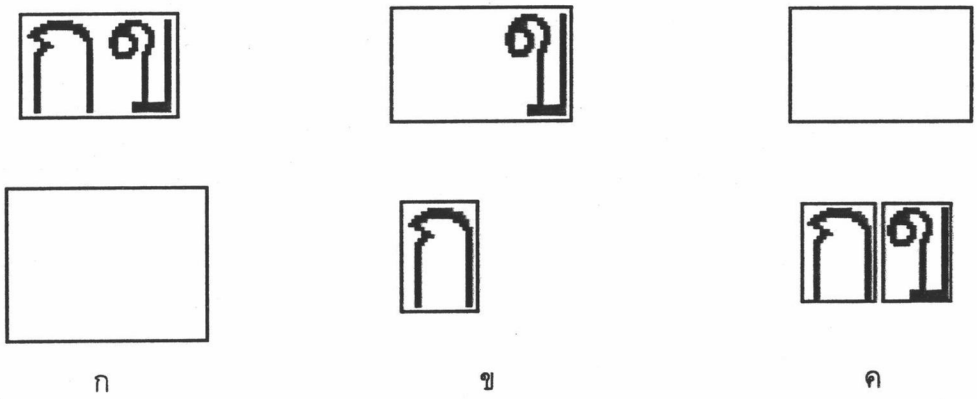
(ข) ผลลัพธ์ที่ได้จากการหาขอบของวัตถุ

2.3.1.2. ทิศทางในการแยกข้อมูลภาพ เพื่อให้สามารถตรวจสอบได้ว่าข้อมูลภาพที่รับเข้ามานั้น ได้ผ่านการแยกข้อมูลภาพออกเป็นตัวอักษรแล้ว จึงได้กำหนดทิศทางในการแยกข้อมูลภาพโดยเริ่มตรวจสอบจากบนลงล่างและซ้ายไปขวา



รูปที่ 2.6 แสดงทิศทางในการแยกข้อมูลภาพ

2.3.1.3. ผลลัพธ์ที่ได้จากการแยกข้อมูลภาพ สามารถแสดงได้ดังรูปที่ 2.7 โดยผลลัพธ์ที่ได้สามารถแบ่งได้ 2 ประเภท ดังนี้



รูปที่ 2.7 การแยกข้อมูลภาพของกลุ่มคำออกทีละตัวอักษร

(ก)บน ภาพต้นแบบที่ยังไม่ได้ผ่านขั้นตอนการแยกข้อมูลภาพ

(ก)ล่าง ผลลัพธ์ที่ได้จากการผ่านขั้นตอนการแยกข้อมูลภาพ

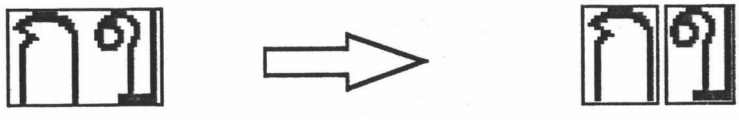
(ข)บน ภาพต้นแบบที่ผ่านขั้นตอนการแยกข้อมูลภาพครั้งที่ 1

(ข)ล่าง ผลลัพธ์ที่ได้จากการผ่านขั้นตอนการแยกข้อมูลภาพ ได้ข้อมูลภาพมา 1 ตัวอักษร

(ค)บน ภาพต้นแบบที่ผ่านขั้นตอนการแยกข้อมูลภาพครั้งที่ 2 จะไม่มีข้อมูลภาพเหลืออยู่แล้ว

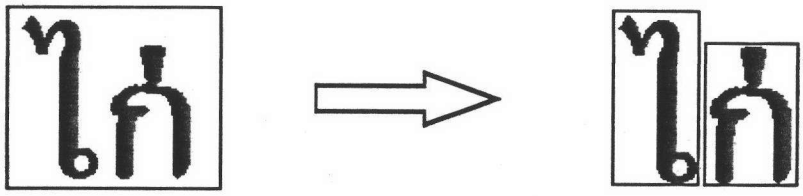
(ค)ล่าง ผลลัพธ์ที่ได้จากการผ่านขั้นตอนการแยกข้อมูลภาพ ได้ข้อมูลภาพมาอีก 1 ตัวอักษร

2.3.1.3.1. ตัวอักษรปกติ คือข้อมูลภาพที่แยกออกมาเป็นตัวอักษร สามารถนำไปทำการรู้จำได้ และไม่มีจุดภาพอยู่บริเวณขอบของตัวอักษร



รูปที่ 2.8 แสดงการแยกข้อมูลภาพที่ได้ตัวอักษรปกติ

2.3.1.3.2. ตัวอักษรติดกัน คือข้อมูลภาพที่แยกออกมาเป็นตัวอักษร แต่ตัวอักษรที่แยกออกมาไม่สามารถนำไปรู้จำได้ เพราะว่ามีการต่อเนื่องกันของตัวอักษร ทำให้ตัวอักษรมีขนาดที่กว้างหรือสูงกว่าตัวอักษรปกติ ต้องนำไปแยกตัวอักษรที่ติดกันเสียก่อน



รูปที่ 2.9 แสดงการแยกข้อมูลภาพที่ได้ตัวอักษรติดกัน

2.3.1.4. การพิจารณาหาตัวอักษรติดกัน เนื่องจากตัวอักษรที่ติดกันนั้น มีความสูงหรือความกว้างของตัวอักษรที่ได้จากการหาขอบของตัวอักษรที่มากกว่าตัวอักษรปกติ ทำให้ต้องมีการตรวจสอบแยกตัวอักษรที่ติดกันก่อน ซึ่งวิธีการตรวจสอบตัวอักษรที่ติดกันนั้น พิจารณาได้ดังนี้

2.3.1.4.1. ความสูงของตัวอักษร ใช้สำหรับพิจารณาในกรณีที่พบว่ามีตัวอักษรที่มีความสูงผิดปกติ เพื่อตรวจสอบว่าเป็นตัวอักษรที่เกินไปในระดับบนหรือต่ำกว่าระดับล่าง หรือเป็นตัวอักษรที่ต่อเนื่องกันทางแนวตั้ง โดยจากการศึกษาตัวอักษรแบบ EucrosiaUPC ขนาด 18 point พบว่า

2.3.1.4.1.1. ค่าต่ำสุดของความสูงของตัวอักษรระดับพยัญชนะมีค่าเท่ากับ 32 จุดภาพ

2.3.1.4.1.2. ค่าสูงสุดของความสูงของตัวอักษรระดับพยัญชนะมีค่าเท่ากับ 36 จุดภาพ

2.3.1.4.1.3. ค่าความสูงของตัวอักษรที่ใช้ในการตรวจสอบว่าเป็นตัวอักษรที่ต่อเนื่องกัน คือค่าสูงสุดของความสูงของตัวอักษรระดับพยัญชนะ

2.3.1.4.1.4. จากการศึกษาวิจัยพบว่า ค่าความสูงของตัวอักษรระดับพยัญชนะสูงสุดเท่ากับ 36 จุดภาพสำหรับตัวอักษรแบบ EucrosiaUPC ขนาด 18 point

2.3.1.4.1.5. หากความสูงของตัวอักษรที่ได้จากการหาขอบของตัวอักษรมากกว่า 36 จุดภาพ แสดงว่าเป็นตัวอักษรที่ต้องทำการตรวจสอบว่าเป็นตัวอักษรที่ติดกันทางแนวตั้งหรือไม่

2.3.1.4.2. ความกว้างของตัวอักษร ใช้สำหรับพิจารณาในกรณีที่พบว่ามีตัวอักษรที่มีความกว้างผิดปกติ เพื่อตรวจสอบว่าเป็นตัวอักษรเป็นตัวอักษรที่ต่อเนื่องกันทางแนวนอนหรือไม่

โดยจากการศึกษาตัวอักษรแบบ EucrosiaUPC ขนาด 18 point พบว่า

2.3.1.4.2.1. ค่าต่ำสุดของความกว้างของตัวอักษรระดับพยัญชนะมีค่าเท่ากับ 16 จุดภาพ

2.3.1.4.2.2. ค่าสูงสุดของความกว้างของตัวอักษรระดับพยัญชนะมีค่าเท่ากับ 40 จุดภาพ

2.3.1.4.2.3. ค่าความกว้างของตัวอักษรที่ใช้ในการตรวจสอบว่าเป็นตัวอักษรที่ต่อเนื่องกันทางแนวนอน คือค่าเฉลี่ยของค่าความกว้างต่ำสุดและสูงสุดของตัวอักษรในระดับพยัญชนะ

2.3.1.4.2.4. จากการศึกษาวิจัยพบว่า ค่าความกว้างของตัวอักษรในระดับพยัญชนะสำหรับพิจารณาตัวอักษรติดกันทางแนวนอนเท่ากับ 28 จุดภาพสำหรับตัวอักษรแบบ EucrosiaUPC ขนาด 18 point

2.3.1.4.2.5. หากความกว้างของตัวอักษรที่ได้จากการหาขอบของตัวอักษรมากกว่า 28 จุดภาพ แสดงว่าเป็นตัวอักษรที่ต้องทำการตรวจสอบว่าเป็นตัวอักษรที่ติดกันทางแนวนอนหรือไม่

2.3.1.4.3. กรณีที่เป็นการติดกันของสระและวรรณยุกต์ จะมีความสูงของตัวอักษรที่ติดกันมากกว่าความสูงของตัวอักษรในระดับพยัญชนะ ทำให้การพิจารณาตัวอักษรในลักษณะนี้พิจารณาเช่นเดียวกับข้อ 2.3.1.4.1.

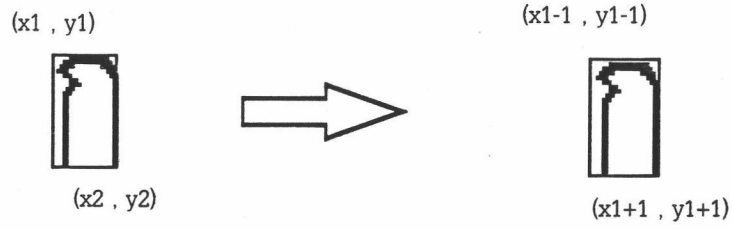
2.3.1.5. <u>การเหลือขอบของตัวอักษรไว้ 1 จุดภาพ</u> เนื่องจากผลลัพธ์ที่ได้ในขั้นตอนนี้ถูกนำไปใช้สำหรับการรู้จำตัวอักษร เพื่อให้สามารถนำไปใช้กับการรู้จำตัวอักษรซึ่งใช้วิธีซินแทกติก ที่มีตารางหน้าต่างช่วยในการทำภาพให้บาง จึงต้องมีการเหลือขอบภาพไว้ 1 จุดภาพทุก ๆ ตัวอักษรที่หาได้ดังแสดงในรูปที่ 2.10 โดยขนาดของตัวอักษรที่ได้เป็นดังนี้

$x1 = x1-1$

$y1 = y1-1$

$x2 = x2+1$

$y2 = y2+1$



(ก) (ข)

รูปที่ 2.10 แสดงการเตรียมข้อมูลสำหรับการรู้จำ
(ก) ตัวอักษรที่ได้จากการแยกข้อมูลภาพ
(ข) ตัวอักษรที่ได้เหลือขอบไว้ 1 จุดภาพ

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

2.3.1.6. แฟ้มข้อมูลของผลลัพธ์ที่ได้จากการแยกข้อมูลภาพ ใช้สำหรับการจัดเก็บข้อมูลของผลลัพธ์ที่ได้จากการแยกข้อมูลภาพ ประกอบด้วยตำแหน่งของตัวอักษรที่ได้จากการแยกข้อมูลภาพเรียงต่อกันไปตามลำดับ โดยตำแหน่งของตัวอักษร ประกอบไปด้วย

2.3.1.6.1. ตำแหน่งขอบด้านซ้ายของตัวอักษร

2.3.1.6.2. ตำแหน่งขอบด้านบนของตัวอักษร

2.3.1.6.3. ตำแหน่งขอบด้านขวาของตัวอักษร

2.3.1.6.4. ตำแหน่งขอบด้านล่างของตัวอักษร

ลำดับที่	ตำแหน่งขอบด้านซ้าย (X1)	ตำแหน่งขอบด้านบน (Y1)	ตำแหน่งขอบด้านขวา (X2)	ตำแหน่งขอบด้านล่าง (Y2)
1				
2				
3				
.	.	.	.	.

รูปที่ 2.11 โครงสร้างการเก็บข้อมูลของตำแหน่งของตัวอักษร

2.3.2. การแยกตัวอักษรที่ติดกัน เมื่อได้ผ่านขั้นตอนการแยกข้อมูลภาพ ผลลัพธ์ที่ได้มีทั้งตัวอักษรเดี่ยว ๆ ซึ่งสามารถนำไปใช้ในขั้นตอนการรู้จำได้ กับตัวอักษรที่ติดกันซึ่งเป็นข้อเสียของการหาขอบของตัวอักษรที่ไม่สามารถทำการแยกตัวอักษรที่ติดกันออกจากกันได้ จึงต้องมีการตรวจสอบตัวอักษรที่ติดกันเพื่อทำการแยกออกจากกัน ให้สามารถนำไปใช้ในการรู้จำตัวอักษรได้ โดยลักษณะของตัวอักษรที่ติดกันแสดงดังรูปที่ 2.12 มีทั้งต่อเนื่องกันในแนวนอน และต่อเนื่องกันในแนวตั้ง ซึ่งการตรวจสอบว่าเป็นตัวอักษรที่ติดกันหรือไม่นั้น พิจารณาตามข้อที่ 2.3.1.4.

(ก) (ข)

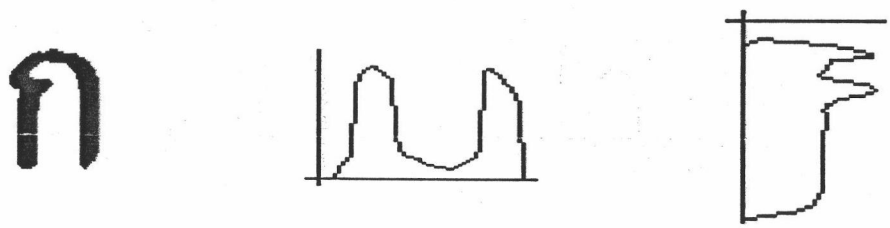
รูปที่ 2.12 แสดงกลุ่มข้อมูลของตัวอักษรที่ต่อเนื่องกัน (ก) ทางแนวนอน (ข) ทางแนวตั้ง

2.3.2.1. การเปลี่ยนข้อมูลภาพของตัวอักษรให้เป็นกราฟของตัวอักษร เนื่องจากข้อมูลภาพที่ได้มาประกอบด้วยจุดภาพที่เรียงต่อกันไป เพื่อให้ง่ายต่อการนำไปเปรียบเทียบกับตัวอักษรต้นแบบ จึงต้องมีการเปลี่ยนจากจุดภาพให้เป็นกราฟของตัวอักษร ซึ่งจะมีกราฟเกิดขึ้น 2 กราฟ คือ

2.3.2.1.1. กราฟทางแนวตั้งเป็นกราฟแสดงความสัมพันธ์ระหว่างปริมาณจุดภาพในแนวตั้งกับตำแหน่งแนวตั้งของจุดภาพกลุ่มนั้น

2.3.2.1.2. กราฟทางแนวนอนเป็นกราฟแสดงความสัมพันธ์ระหว่างปริมาณจุดภาพในแนวนอนกับตำแหน่งแนวนอนของจุดภาพกลุ่มนั้น

จากตัวอย่างในรูปที่ 2.13 (ก) จะได้กราฟของข้อ 2.3.2.1.1. ในรูปที่ 2.13(ข) และกราฟในข้อ 2.3.2.1.2. ในรูปที่ 2.13(ค)



(ก) (ข) (ค)

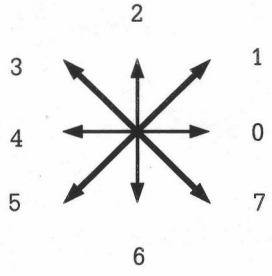
รูปที่ 2.13 แสดงการเปลี่ยนข้อมูลภาพของตัวอักษรเป็นกราฟของตัวอักษร

(ก) ตัวอักษรที่ต้องการนำมาเปลี่ยนเป็นกราฟของตัวอักษร

(ข) และ (ค) ผลของการเปลี่ยนเป็นกราฟของตัวอักษรในแนวตั้งและแนวนอน

2.3.2.2. การเปลี่ยนเป็นรูปสัญญลักษณ์ เมื่อได้กราฟของตัวอักษรมาแล้ว จะทำการเปลี่ยนให้อยู่ในรูปสัญญลักษณ์ เพื่อความสะดวกในการเปรียบเทียบข้อมูล และความรวดเร็ว

2.3.2.1.1. ลักษณะของรูปสัญญลักษณ์ รูปสัญญลักษณ์ที่ใช้ในการศึกษาครั้งนี้ได้กำหนดไว้เป็นรูปสัญญลักษณ์เส้นตรง ซึ่งมีอยู่ด้วยกัน 8 รูปแบบ โดยที่แต่ละรูปสัญญลักษณ์ใช้แสดงทิศทางของปริมาณข้อมูล ณ. จุดนั้น ๆ เช่น รูปสัญญลักษณ์เส้นตรงค่า "0" แทนปริมาณข้อมูล ณ. จุดเปรียบเทียบมีค่าเท่ากับปริมาณข้อมูล ณ. จุดก่อนจุดเปรียบเทียบ เป็นต้น ซึ่งในงานวิจัยนี้จะใช้เพียง 5 รูปสัญญลักษณ์เท่านั้น คือ "0" , "1" , "5" , "6" , "7"



รูปที่ 2.14 แสดงรูปสัญญลักษณ์ที่ใช้แทนทิศทางของปริมาณข้อมูล

2.3.2.1.2. การเปลี่ยนกราฟเป็นรูปสัญญลักษณ์ อาศัยกราฟที่ได้ซึ่งเป็นปริมาณข้อมูลที่เป็นจุดภาพ ณ. จุดที่เปรียบเทียบกับปริมาณข้อมูลที่เป็นจุดภาพ ณ. จุดก่อนจุดเปรียบเทียบ การเปลี่ยนกราฟของตัวอักษรเป็นรูปสัญญลักษณ์ จะพิจารณาทั้งข้อมูลตามแนวนอนและข้อมูลตามแนวตั้ง ผลลัพธ์ที่ได้จากการพิจารณาเป็นข้อมูลของรูปสัญญลักษณ์ที่เรียงติดต่อกัน โดยข้อมูลของรูปสัญญลักษณ์ที่ได้เท่ากับความกว้างและความสูงของตัวอักษร

2.3.2.1.2.1. การเปลี่ยนเป็นรูปสัญญลักษณ์ในแนวนอน จะกระทำกับข้อมูลภาพที่เป็นความสัมพันธ์ระหว่างปริมาณข้อมูลในแนวตั้งกับตำแหน่งของจุดภาพในแนวนอน ซึ่งจากกราฟที่ได้นั้น พบว่ามีรูปสัญญลักษณ์ที่เกิดขึ้นอยู่ด้วยกัน 3 ค่าคือ

- "0" หมายถึง ปริมาณข้อมูลที่เป็นจุดภาพในแนวตั้งของตำแหน่งในแนวนอนปัจจุบันเท่ากับ ปริมาณข้อมูลที่เป็นจุดภาพในแนวตั้งของตำแหน่งในแนวนอนก่อนนี้

- "1" หมายถึง ปริมาณข้อมูลที่เป็นจุดภาพในแนวตั้งของตำแหน่งในแนวนอนปัจจุบันมากกว่า ปริมาณข้อมูลที่เป็นจุดภาพในแนวตั้งของตำแหน่งในแนวนอนก่อนนี้

- "7" หมายถึง ปริมาณข้อมูลที่เป็นจุดภาพในแนวตั้งของตำแหน่งในแนวนอนปัจจุบันน้อยกว่า ปริมาณข้อมูลที่เป็นจุดภาพในแนวตั้งของตำแหน่งในแนวนอนก่อนนี้

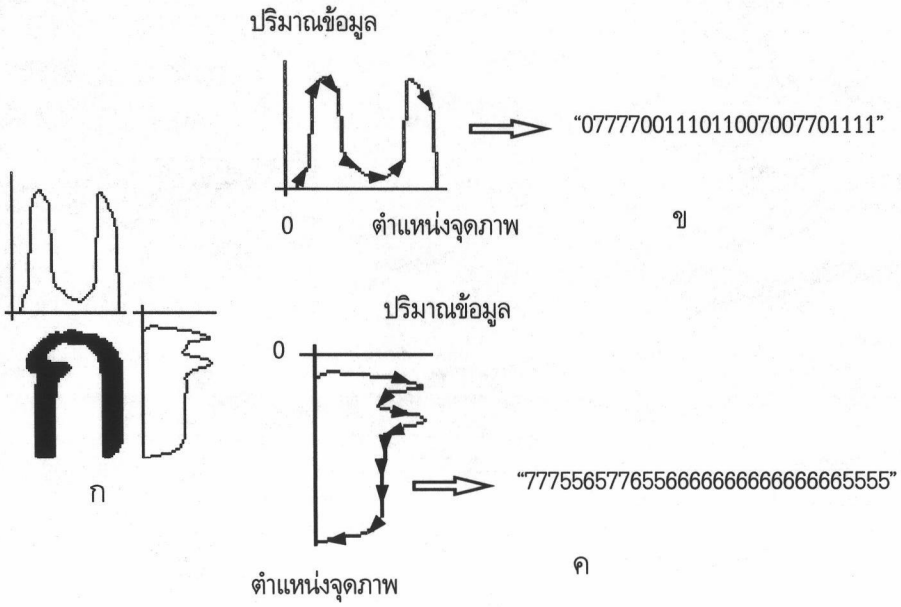
2.3.2.1.2..2. การเปลี่ยนเป็นรูปสัญญลักษณ์ในแนวตั้ง จะกระทำกับข้อมูลภาพที่เป็นความสัมพันธ์ระหว่างปริมาณข้อมูลในแนวนอนกับตำแหน่งของจุดภาพในแนวตั้ง ซึ่งจากกราฟที่ได้นั้น พบว่ามีรูปสัญญลักษณ์ที่เกิดขึ้นอยู่ด้วยกัน 3 ค่าคือ

- "5" หมายถึง ปริมาณข้อมูลที่เป็นจุดภาพในแนวแนวของตำแหน่งในแนวตั้งปัจจุบันน้อยกว่า ปริมาณข้อมูลที่เป็นจุดภาพในแนวนอนของตำแหน่งในแนวตั้งก่อนนี้

- "6" หมายถึง ปริมาณข้อมูลที่เป็นจุดภาพในแนวนอนของตำแหน่งในแนวตั้งปัจจุบันเท่ากับ ปริมาณข้อมูลที่เป็นจุดภาพในแนวนอนของตำแหน่งในแนวตั้งก่อนนี้

- "7" หมายถึง ปริมาณข้อมูลที่เป็นจุดภาพในแนวนอนของตำแหน่งในแนวตั้งปัจจุบันมากกว่า ปริมาณข้อมูลที่เป็นจุดภาพในแนวนอนของตำแหน่งในแนวตั้งก่อนนี้

ซึ่งจากลักษณะของภาพทั้ง 2 แบบพบว่าไม่มีค่าของรูปสัญญลักษณ์ "2" "3" "4" เกิดขึ้นได้เนื่องจากทิศทางของค่าดังกล่าวไม่ตรงกับทิศทางของปริมาณข้อมูลที่ใช้เปรียบเทียบกัน



รูปที่ 2.15 แสดงข้อมูลภาพและการเปลี่ยนกราฟเป็นรูปสัญญลักษณ์

(ก) ข้อมูลภาพที่จะทำการเปลี่ยนเป็นรูปสัญญลักษณ์

(ข) และ (ค) การเปลี่ยนเป็นรูปสัญญลักษณ์ในแนวนอนและแนวตั้งตามลำดับ

2.3.2.3. แฟ้มข้อมูลของตัวอักษรต้นแบบ เนื่องจากการแยกตัวอักษรติดกัน ใช้วิธีการเปรียบเทียบรูปสัญญลักษณ์ของตัวอักษรต้นแบบกับตัวอักษรที่ต้องการแยกออกจากกัน ดังนั้น ตัวอักษรต้นแบบทั้งหมดจะถูกรวบรวมเก็บไว้ในแฟ้มข้อมูลของตัวอักษรต้นแบบ เรียงกันไปตามลำดับการเข้าจัด เก็บ โดยแบ่งแฟ้มข้อมูลของตัวอักษรต้นแบบออกเป็น 2 แฟ้มข้อมูลคือ

1. แฟ้มข้อมูลของตัวอักษรต้นแบบสำหรับรูปสัญญลักษณ์ทางแนวนอน ใช้สำหรับการเปรียบเทียบเพื่อแยกตัวอักษรที่ติดกันทางแนวนอน
2. แฟ้มข้อมูลของตัวอักษรต้นแบบสำหรับรูปสัญญลักษณ์ทางแนวตั้ง ใช้สำหรับการเปรียบเทียบเพื่อแยกตัวอักษรที่ติดกันทางแนวตั้ง

โดยแฟ้มข้อมูลทั้งสองแบบประกอบด้วยตัวอักษรต้นแบบที่มีข้อมูลเพื่อใช้สำหรับการเปรียบเทียบแต่ละตัวอักษรดังนี้

1. ความกว้างของตัวอักษร เพื่อบอกถึงความกว้างของตัวอักษรต้นแบบแต่ละตัวว่ามีความกว้างเพียงใด
2. ความสูงของตัวอักษร เพื่อบอกถึงความสูงของตัวอักษรต้นแบบแต่ละตัวว่ามีความสูงเพียงใด
3. รูปสัญญลักษณ์ของตัวอักษรที่เรียงติดต่อกันไป มีปริมาณที่แตกต่างกันออกไปขึ้นอยู่กับขนาดของตัวอักษร เพื่อใช้สำหรับเป็นข้อมูลในการเปรียบเทียบกับตัวอักษรที่ต้องการนำมาแยกออกจากกัน

2.3.2.4. การเปรียบเทียบกับตัวอักษรต้นแบบ เพื่อหาความแตกต่างระหว่างตัวอักษรที่ต้องการแยกกับตัวอักษรต้นแบบ

2.3.2.4.1. เงื่อนไขการเปรียบเทียบ การเปรียบเทียบมีอยู่ 3 แบบตามลักษณะของตัวอักษรที่ติดกันคือ

2.3.2.4.1.1. การเปรียบเทียบทางแนวนอน เป็นการเปรียบเทียบสำหรับตัวอักษรที่ติดกันทางแนวนอน โดยใช้แฟ้มข้อมูลของตัวอักษรต้นแบบสำหรับรูปสัญญลักษณ์ทางแนวนอน

2.3.2.4.1.2. การเปรียบเทียบทางแนวตั้ง เป็นการเปรียบเทียบสำหรับ

ตัวอักษรที่ติดกันทางแนวตั้ง โดยใช้แฟ้มข้อมูลของตัวอักษรต้นแบบสำหรับรูปสัญญลักษณ์ทางแนวตั้ง

2.3.2.4.1.3. การเปรียบเทียบทางแนวนอนและแนวตั้ง เป็นการเปรียบเทียบสำหรับตัวอักษรที่ติดกันทั้งทางแนวนอนและแนวตั้ง โดยจะทำการเปรียบเทียบทางแนวนอนเพื่อแยกตัวอักษรที่ติดกันทางแนวนอนก่อน แล้วจึงทำการเปรียบเทียบทางแนวตั้งเพื่อแยกตัวอักษรที่ติดกันทางแนวตั้ง

2.3.2.4.2. วิธีการเปรียบเทียบ

2.3.2.4.2.1. นำข้อมูลภาพของตัวอักษรที่ติดกัน มาทำการเปลี่ยนเป็นรูปสัญญลักษณ์เช่นเดียวกับตัวอักษรต้นแบบ

2.3.2.3.4.2. อ่านข้อมูลของตัวอักษรต้นแบบมาทีละ 1 ตัวอักษร

2.3.2.4.2.3. นำรูปสัญญลักษณ์ของตัวอักษรต้นแบบกับตัวอักษรที่ติดกันมาเปรียบเทียบเพื่อหาความแตกต่างของรูปสัญญลักษณ์ ตามขนาดของตัวอักษรต้นแบบ

2.32.4.2.4. ถ้าเป็นการเปรียบเทียบตัวอักษรทางแนวนอน การเปรียบเทียบจะเริ่มเปรียบเทียบจากซ้ายมาทางขวา หากเป็นการเปรียบเทียบตัวอักษรทางแนวตั้ง จะเปรียบเทียบตัวอักษรจากบนลงล่าง

2.3.2.4.2.5. การเปรียบเทียบจะเปรียบเทียบกันแบบตัวต่อตัว โดยรูปสัญญลักษณ์ที่เหมือนกันจะให้ผลลัพธ์เท่ากับ 0 และรูปสัญญลักษณ์ที่ต่างกันจะให้ผลลัพธ์เท่ากับ 1 ซึ่งการเปรียบเทียบจะเปรียบเทียบเท่ากับความกว้างของตัวอักษรต้นแบบสำหรับในกรณีที่เป็นการแยกตัวอักษรที่ติดกันทางแนวนอน และเท่ากับความสูงของตัวอักษรต้นแบบสำหรับในกรณีที่เป็นการแยกตัวอักษรที่ติดกันทางแนวตั้ง

2.3.2.4.2.6. รวมค่าของรูปสัญญลักษณ์ที่ต่างกันทั้งหมดว่ามีค่า เท่าใด ซึ่งจะได้ค่าความแตกต่างของรูปสัญญลักษณ์ระหว่างตัวอักษรต้นแบบกับตัวอักษรที่ติดกัน

2.3.2.4.2.7. ทำการเปรียบเทียบกับตัวอักษรต้นแบบทุกตัว

2.3.2.4.2.8. **ขนาดของตัวอักษรตัวแรกของตัวอักษรที่ติดกัน** คือ ตัวอักษรต้นแบบที่มีรูปสัญญลักษณ์ต่างจากรูปสัญญลักษณ์ของตัวอักษรที่ติดกันน้อยที่สุด

2.3.2.4.2.9. **ขนาดของตัวอักษรที่ติดกันในส่วนที่เหลือ** คือ ขนาดของตัวอักษรที่ติดกันลบด้วยขนาดของตัวอักษรตัวแรกที่หาได้

2.3.2.4.2.10. ขนาดของตัวอักษรส่วนที่เหลือคือตัวอักษรตัวที่สองของตัวอักษรที่ติดกัน

2.3.2.4.2.11. เงื่อนไขการเปรียบเทียบคือค่าความผิดพลาดสัมบูรณ์ที่น้อยที่สุด

$$e = \min \Sigma \left| x_{ij} - y_{ij} \right|$$

โดยที่ e คือผลรวมของค่าควาผิดพลาดสัมบูรณ์

x_{ij} คือค่าของจุดภาพที่ (i,j) ของภาพต้นแบบ

y_{ij} คือค่าของจุดภาพที่ (i,,j) ของภาพที่นำมาเปรียบเทียบ

2.3.2.4.2.12. การเปรียบเทียบเพื่อแยกตัวอักษรที่ติดกัน โดยใช้ตัวอักษรต้นแบบ แสดงได้ดังนี้

-การเปรียบเทียบสำหรับตัวอักษรที่ติดกันทางแนวนอน

for j = 0 to maxchar

$$e_j = \sum_{i=0}^{width(j)} [r_i]$$

โดยที่ r_i = $\begin{cases} 0 & \text{เมื่อ } x_i = y_i \\ 1 & \text{เมื่อ } x_i <> y_i \end{cases}$

w1 = width of min [e_j]

w2 = width of test char - w1

- การเปรียบเทียบสำหรับตัวอักษรที่ติดกันทางแนวตั้ง

for j = 0 to maxchar

$$e_j = \sum_{i=0}^{height(j)} [r_i]$$

โดยที่ r_i = 0 เมื่อ $x_i = y_i$

1 เมื่อ $x_i <> y_i$

h1 = height of min [e_i]

h2 = height of test char - h2

e_i = ค่าที่ได้จากการเปรียบเทียบตัวอักษร

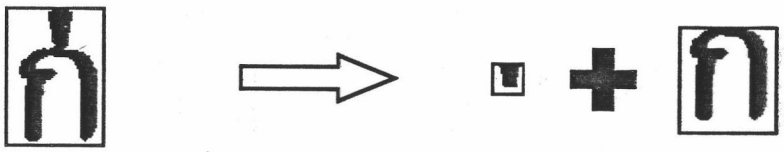
x_i = ค่าสัญญลักษณ์ที่จุดภาพตำแหน่ง i ของตัวอักษรต้นแบบ

y_i = ค่าสัญญลักษณ์ที่จุดภาพตำแหน่ง i ของตัวอักษรที่มาเปรียบเทียบ

w1,w2 = ความกว้างของตัวอักษรตัวที่ 1 และตัวที่ 2

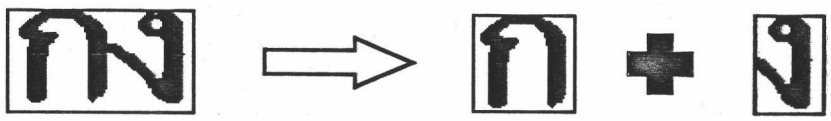
h1,h2 = ความสูงของตัวอักษรตัวที่ 1 และตัวที่2

2.3.2.4.2.13. ผลลัพธ์ที่ได้จากการแยกตัวอักษรทางแนวตั้งแสดงอยู่ในรูปที่ 2.16



รูปที่ 2.16 แสดงการแยกตัวอักษรทางแนวตั้ง

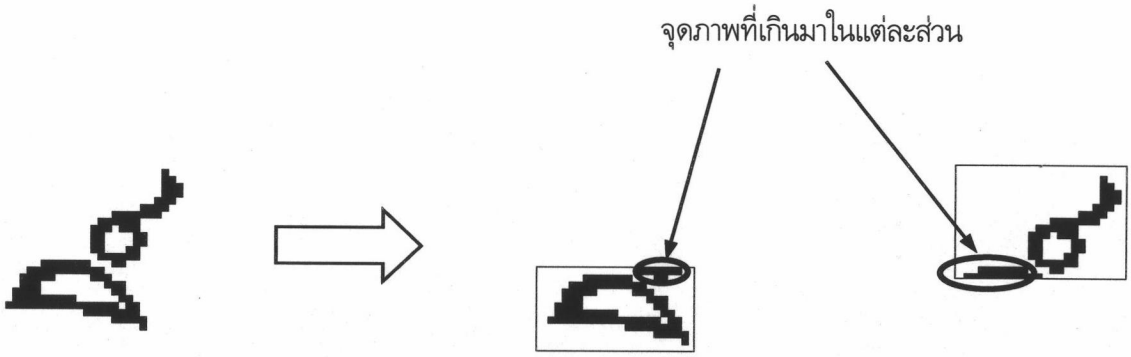
2.3.2.4.2.14. ผลลัพธ์ที่ได้จากการแยกตัวอักษรทางแนวนอนแสดงอยู่ในรูปที่ 2.17



รูปที่ 2.17 แสดงการแยกตัวอักษรทางแนวนอน

2.3.3. การตรวจสอบจุดภาพข้างเคียง เมื่อได้ทำการแบ่งข้อมูลภาพออกเป็นตัวอักษรแต่ละตัว บางครั้งพบว่าตัวอักษรที่แบ่งออกมานั้น จะมีข้อมูลของตัวอักษรข้างเคียงปรากฏอยู่ด้วย ซึ่งอาจทำให้ผลของการรู้จำผิดพลาดได้ จึงต้องมีการตรวจสอบตัวอักษรที่แบ่งออกมาว่ามีการรบกวนจากตัวอักษรข้างเคียงหรือไม่ โดยลักษณะของการรบกวนจากตัวอักษรข้างเคียงแสดงดังรูปที่ 2.18

2.3.3.1. วิธีการตรวจสอบจุดภาพข้างเคียง ตัวอักษรที่ได้จากการแยกข้อมูลภาพจะถูกทำเพิ่มขอบนอกสุดของตัวอักษรทั้ง 4 ด้านให้ว่างไว้ด้านละ 1 จุดภาพเพื่อใช้ในการตรวจสอบด้วยตารางหน้าต่างขนาด 3x3 สำหรับขั้นตอนการรู้จำ ดังนั้นการตรวจสอบจุดภาพข้างเคียงทำได้ดังนี้



รูปที่ 2.18 แสดงจุดภาพที่เกิดขึ้นจากการเหลื่อมล้ำกันของข้อมูลภาพที่ใกล้เคียงกัน

2.3.3.1.1. ตรวจสอบขอบนอกสุดทั้ง 4 ด้านว่ามีจุดภาพที่เป็นภาพอยู่หรือไม่ หากไม่พบว่ามีจุดภาพ แสดงว่าไม่มีการรบกวนจากตัวอักษรข้างเคียง แต่หากพบว่ามีจุดภาพข้างเคียง แสดงว่ามีการรบกวนจากตัวอักษรข้างเคียง

2.3.3.1.2. ใช้วิธีการหาขอบภาพเช่นเดียวกับการแยกข้อมูลภาพ เพื่อตรวจสอบว่าขนาดข้อมูลภาพของตัวอักษรข้างเคียงที่เกินมาอยู่ที่ใด

2.3.3.1.3. เก็บตำแหน่งที่ได้จากการหาขอบภาพเพื่อใช้สำหรับเป็นการบอกให้ทราบถึงตำแหน่งของข้อมูลของตัวอักษรข้างเคียง และจำนวนกลุ่มข้อมูลของตัวอักษรข้างเคียง

2.3.3.1.4. จำนวนกลุ่มข้อมูลของตัวอักษรข้างเคียง สามารถมีได้ไม่จำกัดขึ้นอยู่กับว่ามีจำนวนกลุ่มข้อมูลเท่าไร ที่ตรวจสอบในตัวอักษรตัวหนึ่ง

2.3.3.2. ผลลัพธ์ที่ได้จากการตรวจสอบจุดภาพข้างเคียง

2.3.3.2.1. ข้อมูลของผลลัพธ์ที่ได้จากการตรวจสอบ มีรายละเอียดดังรูปที่ 2.19 ประกอบไปด้วย

2.3.3.2.1.1. จำนวนของกลุ่มข้อมูลของตัวอักษรข้างเคียง

2.3.3.2.1.2. ตำแหน่งของกลุ่มข้อมูลของตัวอักษรข้างเคียง

2.3.3.2.1.3. จากรูปที่ 2.19 สามารถอธิบายได้ดังนี้

ก. จำนวนกลุ่มข้อมูล คือ จำนวนของกลุ่มข้อมูลของตัวอักษรข้างเคียง ซึ่งมีค่าได้ตั้งแต่ 0 ขึ้นไป

ข. จุดภาพของตัวอักษรข้างเคียง คือ ตำแหน่งของจุดภาพของตัวอักษรข้างเคียงที่เกินเข้ามาในแต่ละด้านของตัวอักษร ซึ่งหากจำนวนกลุ่มข้อมูลมีค่าเท่ากับ 0 ก็จะไม่มีจุดภาพของตัวอักษรข้างเคียง

จำนวนกลุ่มข้อมูล	จุดภาพของตัวอักษรข้างเคียง ชุดที่1	. ชุดที่2	. ชุดที่3	. .	. .	. ชุดที่n
1 byte	8 bytes	.	.	.	.	.

รูปที่ 2.19 แสดงโครงสร้างของการเก็บข้อมูล

2.3.3.2.2. กรณีที่ไม่มีจุดภาพข้างเคียง มีรายละเอียดดังรูปที่ 2.20 คือมีเพียงแต่จำนวนกลุ่มข้อมูลเท่านั้นโดยมีค่าเท่ากับ “0”

“0”
1 byte

รูปที่ 2.20 แสดงข้อมูลกรณีไม่มีจุดภาพข้างเคียง

2.3.3.2.3. กรณีที่มีจุดภาพข้างเคียง มีรายละเอียดดังรูปที่ 2.21 โดยขึ้นอยู่กับจำนวนกลุ่มข้อมูลว่ามีค่าเท่าไร ตำแหน่งของตัวอักษรข้างเคียงจะมีจำนวนเท่ากับจำนวนกลุ่มข้อมูล ซึ่งจากตัวอย่างในรูปที่ 2.21 แสดงให้เห็นว่ามีกลุ่มข้อมูลของตัวอักษรเพียง 2 กลุ่ม

“2”	จุดภาพตัวอักษรข้างเคียงชุด1	จุดภาพตัวอักษรข้างเคียงชุด2
1 byte	8 bytes	8 bytes

รูปที่ 2.21 ตัวอย่างกรณีมีจุดภาพข้างเคียง 2 กลุ่ม

2.3.4. การเปลี่ยนจุดภาพของตัวอักษรที่มีค่าเป็น "1" ให้มีค่าเป็นค่าอื่นที่ไม่ใช่ "0" เมื่อได้ทำการแยกตัวอักษรออกมาแล้ว ตัวอักษรที่แยกออกมานั้น ยังคงเป็นพื้นที่ส่วนหนึ่งในข้อมูลภาพนั้น จึงจำเป็นต้องทำการเปลี่ยนค่าของข้อมูลของตัวอักษรที่แยกออกมา ให้มีค่าเป็นค่าอื่น ที่มิใช่ค่า "0" เพราะว่าจุดภาพของตัวอักษรยังคงต้องเก็บไว้เพื่อนำไปใช้งานในขั้นตอนต่อไป และเป็นการบอกให้ทราบว่าข้อมูลของตัวอักษรตัวนี้ได้ทำการแยกออกไปแล้ว ไม่จำเป็นต้องมาพิจารณาข้อมูลของตัวอักษรตัวนี้อีก ซึ่งการเปลี่ยนจุดภาพนี้ยังคงให้ข้อมูลภาพมีความเหมือนเดิมอยู่ เพียงแต่จุดภาพที่มีค่าเป็น "1" ก็จะมีค่าเป็น "2" และพื้นที่บริเวณขอบด้านซ้ายของตัวอักษร เดิมมีค่าเป็น "0" ก็จะมีค่าเป็น "255"

2.3.4.1. ค่าที่ใช้แทนจุดภาพที่ได้ทำการตรวจสอบแล้ว แบ่งออกได้ 2 ค่าดังนี้คือ

2.3.4.1.1. ค่า "2" ใช้แทนจุดภาพที่มีค่าเป็น "1" เพื่อให้ยังคงรักษาข้อมูลภาพให้อยู่เหมือนเดิม และเป็นการบอกให้ทราบถึงตำแหน่งที่ได้หาขอบของตัวอักษรแล้ว

2.3.4.1.2. ค่า "255" ใช้แทนจุดภาพที่มีค่าเป็น "0" และเป็นจุดภาพที่อยู่บริเวณขอบซ้ายเท่านั้น เพื่อเป็นความรวดเร็วในการตรวจสอบข้อมูลในแต่ละแถวของจุดภาพ โดยบอกให้ทราบว่าแถวของจุดภาพนั้นได้ผ่านการตรวจสอบจุดภาพหมดแล้ว

2.3.4.2. วิธีการเปลี่ยนจุดภาพ สามารถทำได้ดังนี้

2.3.4.2.1. เริ่มต้นด้วยการหาจุดภาพที่เป็นตัวอักษร และทำการหาขอบของตัวอักษร โดยจะได้ตัวอักษรมา 1 ตัวอักษร ดังรูปที่ 2.22

จุดภาพแรกที่พบ

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	1	0	1	0	0
0	1	0	0	0	1	0	0	1	0	1	0	0
0	1	0	0	0	1	0	0	1	0	1	0	0
0	1	0	0	0	1	0	0	1	0	1	0	0
0	1	0	0	0	1	0	0	1	0	1	0	0
0	1	0	0	0	1	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่ 2.22 แสดงภาพตัวอย่างในการเปลี่ยนจุดภาพ

พระบรมราชูปถัมภ์ สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

2.3.4.2.2. ทำการเปลี่ยนจุดภาพที่มีค่าเป็น "1" ของตัวอักษรที่หาได้ให้มีค่า เป็น "2" และตรวจสอบว่าในแต่ละแถวได้ผ่านการตรวจสอบหาขอบของตัวอักษรหมดหรือไม่ ซึ่งหากตรวจสอบหมดแล้ว ก็จะทำการเปลี่ยนค่าจุดภาพที่ขอบซ้าย ให้มีค่าเป็น "255" ดังแสดงในรูป 2.23

255	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	2	2	0	0	0	1	0	1	0	0
0	2	0	0	0	2	0	0	1	0	1	0	0
0	2	0	0	0	2	0	0	1	0	1	0	0
0	2	0	0	0	2	0	0	1	0	1	0	0
0	2	0	0	0	2	0	0	1	0	1	0	0
0	2	0	0	0	2	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่ 2.23 ผลลัพธ์ที่ได้จากการเปลี่ยนจุดภาพบริเวณขอบ

2.3.4.2.3. เมื่อได้ตัวอักษรมา 1 ตัวแล้ว จากนั้นกลับไปตรวจสอบหาจุดภาพและแยกข้อมูลภาพออกมาเป็นตัวอักษร และทำการเปลี่ยนค่าของจุดภาพของตัวอักษรที่ได้ให้มีค่าเป็น "2" และเปลี่ยนค่าจุดภาพที่ขอบซ้ายให้มีค่าเป็น "255" หากตรวจสอบข้อมูลภาพหมดแล้ว โดยจะได้ผลลัพธ์ออกมาดังรูป 2.24

255	0	0	0	0	0	0	0	0	0	0	0	0
255	0	2	2	2	0	0	0	2	0	2	0	0
255	2	0	0	0	2	0	0	2	0	2	0	0
255	2	0	0	0	2	0	0	2	0	2	0	0
255	2	0	0	0	2	0	0	2	0	2	0	0
255	2	0	0	0	2	0	0	2	0	2	0	0
255	2	0	0	0	2	0	0	2	2	2	0	0
255	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่ 2.24 ผลลัพธ์ที่ได้จากขั้นตอนการเปลี่ยนจุดภาพ

2.3.4.2.4. เมื่อได้ทำการตรวจสอบจุดภาพหมดแล้ว ตำแหน่งจุดขอบภาพด้านซ้ายจะมีค่าเท่ากับ "255" ทั้งหมด แสดงว่าได้ทำการตรวจสอบจุดภาพหมดแล้ว ถือเป็นการสิ้นสุดการหาขอบของตัวอักษรและการเปลี่ยนจุดภาพ

2.3.5. แฟ้มข้อมูลที่ได้จากการแยกข้อมูลภาพ เมื่อนำผลลัพธ์ที่ได้จากการแยกข้อมูลภาพโดยวิธีการหาขอบภาพและการตรวจสอบหาจุดภาพข้างเคียงมารวมกัน แฟ้มข้อมูลที่ได้ประกอบด้วยข้อมูลของตัวอักษรแต่ละตัวซึ่งมีรายละเอียดดังนี้

2.3.5.1. ตำแหน่งของตัวอักษร เพื่อบอกถึงตำแหน่งของตัวอักษรที่ได้

2.3.5.2. จำนวนของกลุ่มข้อมูลของตัวอักษรข้างเคียง

2.3.5.3. ตำแหน่งของกลุ่มข้อมูลของตัวอักษรข้างเคียง

2.3.5.4. โดยตัวอักษรแต่ละตัว จะมีลักษณะที่แตกต่างดังนี้

2.3.5.4.1. ตัวอักษรที่ไม่มีจุดภาพของตัวอักษรข้างเคียง จะมีโครงสร้างดังรูป ที่ 2.25

ตำแหน่งตัวอักษร	"0"
8 bytes	1 byte

รูปที่ 2.25 โครงสร้างของตัวอักษรที่ไม่มีจุดภาพของตัวอักษรข้างเคียง

2.3.5.4.2. ตัวอักษรที่มีจุดภาพของตัวอักษรข้างเคียง จะมีโครงสร้างดังรูปที่ 2.26

ตำแหน่งตัวอักษร	จำนวนกลุ่มข้อมูล	กลุ่มข้อมูลข้างเคียงที่ 1	กลุ่มข้อมูลข้างเคียงที่ 2
8 bytes	1 byte	8 bytes	8 bytes

รูปที่ 2.26 ตัวอักษรที่มีจุดภาพจากตัวอักษรข้างเคียง

2.4. การจัดเรียงตัวอักษร

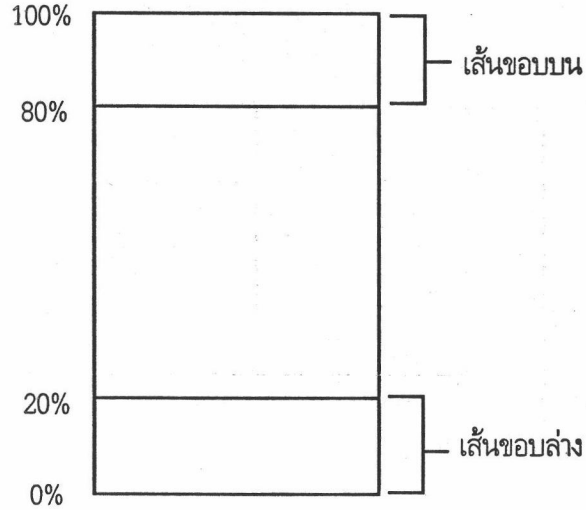
การจัดเรียงตัวอักษร เป็นวิธีการในการเรียงลำดับตัวอักษรที่ได้จากขั้นตอนการแยกข้อมูลภาพโดยนำเอาตำแหน่งของตัวอักษรที่ได้มาพิจารณา โดยจากรูปที่ 2.27 เมื่อได้ตำแหน่งของตัวอักษรแต่ละตัว

แล้ว ก็จะทำการใส่เลขดัชนีให้กับตัวอักษรทุกตัว เพื่อใช้สำหรับการบอกประเภทของตัวอักษรตัวนั้นว่าอยู่ในประเภทใด และความสะดวกในการจัดเรียงตัวอักษร โดยการพิจารณาจากขนาดของตัวอักษร ตำแหน่งของตัวอักษร และเส้นฐานที่ได้จากการแยกข้อมูลภาพ จากนั้นจึงนำไปทำการจัดเรียงตัวอักษรให้ถูกต้องตามหลักการพิมพ์ตัวอักษรภาษาไทย ซึ่งใช้กันอยู่ตามโปรแกรม word processor ภาษาไทย ทั่วไป และมีการตรวจสอบช่องว่างระหว่างตัวอักษรว่ามีการเว้นวรรคระหว่างตัวอักษรหรือไม่

รูปที่ 2.27 แสดงโครงสร้างของการจัดเรียงตัวอักษร

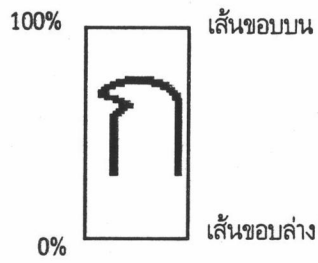
2.4.1. การหาเลขดัชนีสำหรับตัวอักษรแต่ละตัว ตัวเลขดัชนีที่ใส่เพิ่มเข้าไปนั้น ใช้สำหรับการแบ่งประเภทของตัวอักษรนั้น ๆ ว่าอยู่ในประเภทใด เพื่อใช้สำหรับกำหนดเส้นขอบล่างและเส้นขอบบนของตัวอักษรที่ต้องการรู้จำ เมื่อเวลานำไปทำการรู้จำ

2.4.1.1. รูปแบบการแบ่งประเภทของตัวอักษร ใช้รูปแบบมาตรฐานในรูปที่ 2.28 เป็นตัวแบ่งประเภทของตัวอักษร ดังนี้



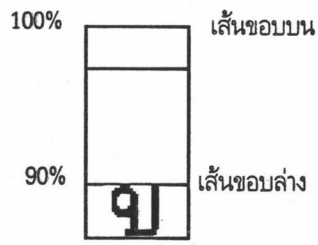
รูปที่ 2.28 แสดงรูปแบบมาตรฐานที่ใช้แบ่งลักษณะของตัวอักษร

2.4.1.1.1. ตัวอักษรระดับกลาง คือตัวอักษรที่อยู่ในช่วงของเส้นขอบบนและเส้นขอบล่าง โดยจะกำหนดให้ตัวอักษรในลักษณะนี้มีเส้นฐาน = 0% และเส้นระดับบน = 100% ตัวอักษรในลักษณะนี้ เช่น ก ข ค ง ฯลฯ กำหนดให้เลขดัชนีมีค่า = "1"



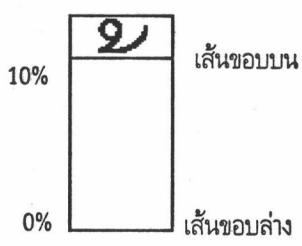
รูปที่ 2.29 ตัวอักษรระดับกลาง

2.4.1.1.2. ตัวอักษรระดับล่าง คือตัวอักษรที่อยู่ต่ำกว่าช่วงของเส้นขอบล่าง ลงมา โดยกำหนดให้ตัวอักษรในลักษณะนี้มีเส้นฐาน = 90% และเส้นระดับบน = 100% ตัวอักษร ในลักษณะนี้ เช่น ฯลฯ กำหนดให้เลขดัชนีมีค่า = "2"



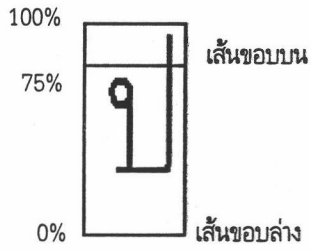
รูปที่ 2.30 ตัวอักษรระดับล่าง

2.4.1.1.3. ตัวอักษรระดับบน คือตัวอักษรที่อยู่เหนือกว่าช่วงของเส้นขอบบนขึ้นไป โดยกำหนดให้ตัวอักษรในลักษณะนี้มีเส้นฐาน = 0% และเส้นระดับบน = 10% ซึ่งส่วนใหญ่ได้แก่ สระในระดับบน วรรณยุกต์ กำหนดให้เลขดัชนีมีค่า = "3"



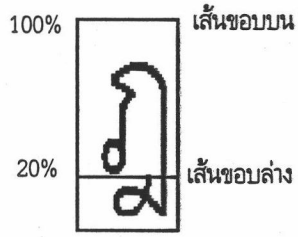
รูปที่ 2.31 ตัวอักษรระดับบน

2.4.1.1.4. ตัวอักษรที่เกินกว่าเส้นขอบบน คือตัวอักษรที่มีเส้นฐานอยู่เหนือเส้นขอบล่าง และเส้นระดับบนจะอยู่เหนือเส้นขอบบน ซึ่งลักษณะของตัวอักษรแบบนี้จะถูกกำหนดให้มีเส้นฐาน = 0% และเส้นระดับบน = 75% ตัวอักษรในลักษณะนี้เช่น ฝ ป กำหนดให้เลขดัชนีมีค่า = “4”



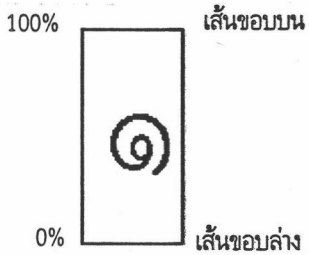
รูปที่ 2.32 ตัวอักษรระดับที่เกินกว่าเส้นขอบบน

2.4.1.1.5. ตัวอักษรที่ต่ำกว่าเส้นขอบล่าง คือ ตัวอักษรที่มีเส้นระดับบนอยู่ต่ำกว่าเส้นขอบบน และเส้นระดับล่างอยู่ต่ำกว่าเส้นขอบล่าง ซึ่งลักษณะของตัวอักษรแบบนี้จะถูกกำหนดให้มีเส้นฐาน = 20% และเส้นระดับบน = 100% ตัวอักษรในลักษณะนี้ได้แก่ ฎ ฏ กำหนดให้เลขดัชนีมีค่า = “5”



รูปที่ 2.33 ตัวอักษรที่ต่ำกว่าเส้นขอบล่าง

2.4.1.1.6. ตัวเลขไทย ถูกจัดให้อยู่กลุ่มเดียวกับตัวอักษรปกติ คือเป็นตัวอักษรที่มีเส้นระดับบนอยู่ต่ำกว่าเส้นขอบบน และเส้นระดับล่างอยู่เหนือกว่าเส้นขอบล่าง ซึ่งลักษณะของตัวอักษรแบบนี้ ถูกกำหนดให้มีเส้นฐาน = 0% และเส้นระดับบน = 100% ดังนั้นค่าเลขดัชนีจึงมีค่าเท่ากับเลขดัชนี ของตัวอักษรปกติคือ “1”



รูปที่ 2.34 ตัวอักษรที่เป็นตัวเลข

2.4.1.2. วิธีการแบ่งประเภทของตัวอักษร โดยการนำเอาความสูงของตัวอักษรปกติและเส้นฐานของตัวอักษรมาใช้ในการพิจารณา ซึ่งสามารถหาได้ดังนี้

2.4.1.2.1. หาความสูงของตัวอักษรปกติ โดย

```
ให้ normal และ large =          0

DO WHILE number of caracter <> 0
    read position of character
    IF  (ความสูงของตัวอักษร > large)
        IF  ( large < ความสูงของตัวอักษร-Threshold ) and
            ( large > normal+Threshold )
            normal = large
        large = ความสูงของตัวอักษร
    ELSE
        IF (ความสูงของตัวอักษร<large-Threshold) and
           (ความสูงของตัวอักษร> normal+ Threshold )
            normal = ความสูงของตัวอักษร

IF ( normal <= Threshold*2 )
    normal = large

normal      = ขนาดความสูงของตัวอักษรปกติ
large       = ขนาดความสูงของตัวอักษรที่สูงกว่าปกติ
Threshold   = ค่าที่ใช้แยกระหว่างขนาดความสูงของตัวอักษรปกติและตัวอักษรที่สูง
              กว่าปกติ
              กำหนดไว้มีค่า = 40
```

2.4.1.2.2. หาเส้นฐาน โดย

```
normal = ความสูงของตัวอักษรปกติ

อ่านตำแหน่งตัวอักษรตัวแรก, เส้นฐาน = ขอบล่างของตัวอักษรที่อ่าน
DO อ่านตำแหน่งตัวอักษรมาทีละตัวอักษร
        IF ( ความสูงของตัวอักษรที่อ่าน < normal )
                IF ( ขอบล่างของตัวอักษรที่อ่าน > เส้นฐาน
                        เส้นฐาน = ขอบล่างของตัวอักษรที่อ่าน
                ENDIF
        ENDIF
```

2.4.1.2.3. ถ้าความสูงของตัวอักษรที่นำมาพิจารณาอยู่ในช่วงความสูงของตัวอักษรปกติ และมีขอบล่างอยู่เหนือเส้นฐาน แสดงว่าเป็น ตัวอักษรปกติ

2.4.1.2.4. ถ้าขอบล่างอยู่ต่ำกว่าเส้นฐานแต่ขอบบนอยู่เหนือเส้นฐาน แสดงว่าเป็น ตัวอักษรที่ต่ำกว่าระดับล่าง

2.4.1.2.5. ถ้าขอบล่างอยู่ต่ำกว่าเส้นฐานแต่ขอบบนไม่อยู่เหนือเส้นฐาน แสดงว่าเป็น ตัวอักษรระดับล่าง

2.4.1.2.6. ถ้าขอบบนอยู่เหนือกว่าผลบวกของเส้นฐานกับความสูงของตัวอักษรปกติ แต่มีขอบล่างอยู่ต่ำกว่าผลบวกของเส้นฐานกับความสูงของตัวอักษรปกติ แสดงว่าเป็น ตัวอักษรที่เกินไปในระดับบน

2.4.1.2.7. ถ้าขอบบนอยู่เหนือกว่าผลบวกของเส้นฐานกับความสูงของตัวอักษรปกติ แต่มีขอบล่างอยู่เหนือผลบวกของเส้นฐานกับความสูงของตัวอักษรปกติ แสดงว่าเป็น ตัวอักษรระดับบน

2.4.1.2.8. โดยที่ ขอบซ้ายมีค่าเป็น x1 ขอบบนทีค่าเป็น y1
ขอบขวามีค่าเป็น x2 ขอบล่างมีค่าเป็น y2
ความสูงของตัวอักษรมีค่าเป็น y2-y1

2.4.2. การจัดเรียงตัวอักษร เนื่องจากตำแหน่งที่ได้จากการแยกข้อมูลภาพนั้น ไม่สามารถนำไปใช้ได้ เพราะว่าตำแหน่งของตัวอักษรที่ได้มานั้น ยังไม่ถูกต้องตามมาตรฐานการพิมพ์ในภาษาไทย หากนำไปผ่านการรู้จำและนำผลลัพธ์ไปใช้กับโปรแกรมทางด้าน word processor จะให้ผลลัพธ์ที่ไม่ถูกต้อง จึงต้องมี

2.4.2.1. การจัดเรียงตัวอักษรให้ถูกต้องตามมาตรฐานของ word processor มีดังนี้

2.4.2.1.1. ตัวอักษรในระดับพยัญชนะ จะเรียงติดต่อกันไป

2.4.2.1.2. ตัวอักษรที่เป็นพยัญชนะ จะมาก่อนตัวอักษรที่เป็นสระ ไม่ว่าจะเป็นสระบน หรือสระล่าง

2.4.2.1.3. ตัวอักษรที่เป็นสระ จะมาก่อนตัวอักษรที่เป็นวรรณยุกต์

จากข้อกำหนดดังกล่าว จึงนำเอาตำแหน่งของตัวอักษรที่ได้จากการแยกข้อมูลภาพและดัชนีที่ใช้แบ่งประเภทของตัวอักษรมาใช้ในการจัดเรียงตัวอักษร

2.4.2.2. วิธีการจัดเรียงตัวอักษร ใช้วิธีการเปรียบเทียบระหว่างตัวอักษร เพื่อหาตำแหน่งที่ถูกต้องของตัวอักษรแต่ละตัว สามารถทำได้โดย

```
อ่านตำแหน่งของตัวอักษรตัวแรก
Do อ่านตำแหน่งของตัวอักษรตัวถัดไป
    IF (ขอบซ้ายของตัวอักษรที่อ่าน < ขอบซ้ายของตัวอักษร )
        IF เป็นตัวอักษรตัวแรก
            ตัวอักษรตัวที่อ่านเป็นตัวแรก, ตัวอักษรตัวที่เปรียบเทียบเป็นตัวถัดไป
        ELSE
            ตัวอักษรตัวที่อ่านต่อจากตัวอักษรที่เปรียบเทียบ
        ENDIF
    ELSE
        IF  (ขอบขวาของตัวอักษรที่เปรียบเทียบ > ขอบซ้ายของตัวอักษรที่อ่าน)
        และ (ความสูงของตัวอักษรที่เปรียบเทียบ < ความสูงของตัวอักษรที่อ่าน)
            IF เป็นตัวอักษรตัวแรก
                ตัวอักษรตัวที่อ่านเป็นตัวแรก,ตัวอักษรตัวที่เปรียบเทียบเป็นตัวถัดไป
```

```
    ENDIF
ELSE
    ตัวอักษรตัวที่อ่านต่อจากตัวอักษรที่เปรียบเทียบ
```

2.4.2.3. โดยที่ ขอบซ้ายมีค่าเป็น x1
ขอบขวามีค่าเป็น x2
ขอบบนมีค่าเป็น y1
ขอบล่างมีค่าเป็น y2
ความสูงมีค่าเป็น y2-y1
ความกว้างมีค่าเป็น x2-x1

2.4.3. การตรวจสอบการเว้นวรรค เนื่องจากในบางครั้งข้อความที่นำมาวิเคราะห์นั้น มีช่องว่างระหว่างตัวอักษรมากกว่าช่องว่างปกติของตัวอักษร ซึ่งแสดงให้เห็นว่าข้อความที่นำมาวิเคราะห์นั้น มีการเว้นช่องระหว่างตัวอักษรไว้ด้วย ระบบรู้จำก็จะทำการตรวจสอบและเติมอักขระเว้นวรรคเข้าไป เพื่อใช้แทนช่องว่างระหว่างตัวอักษร โดยตัวอักขระเว้นวรรคที่เติมเข้าไปนั้นมีค่ารหัส ASCII = "20" ในเลขฐานสิบหก หรือ = "32" ในเลขฐานสิบ หรือคือตัวอักษร " "

2.4.3.1. วิธีการตรวจสอบการเว้นวรรค

2.4.3.1.1. การตรวจสอบจะใช้ตำแหน่งของตัวอักษรแต่ละคู่ที่อยู่ติดกันมาใช้ในการพิจารณา

2.4.3.1.2. ตัวอักษรที่อยู่ข้างหน้าจะใช้ขอบขวา(x2) เป็นตัวพิจารณา

2.4.3.1.3. ตัวอักษรที่อยู่ข้างหลังจะใช้ขอบซ้าย(x1) เป็นตัวพิจารณา

2.4.3.1.4. ความกว้างเฉลี่ยของตัวอักษร = ผลรวมของความกว้างของตัวอักษรในระดับปกติทั้งหมด / จำนวนตัวอักษรในระดับปกติทั้งหมด

2.4.3.1.5. นำค่าที่ได้จากการหาความกว้างเฉลี่ยของตัวอักษรไปเปรียบเทียบกับผลต่างของตัวอักษรที่กำลังพิจารณา

2.4.3.1.6. ถ้าผลการเปรียบเทียบปรากฏว่า ผลต่างของตัวอักษรที่กำลังพิจารณามากกว่าความกว้างเฉลี่ยของตัวอักษร ให้เติมอักขระเว้นวรรคเข้าไป

2.4.4. แฟ้มข้อมูลของการจัดเรียงตัวอักษร เมื่อได้ทำการจัดเรียงตัวอักษรเรียบร้อยแล้ว ข้อมูลที่ได้จะถูกจัดเก็บเข้าไว้ในแฟ้มข้อมูลตามลำดับของตัวอักษรที่ได้ทำการจัดเรียง เพื่อนำไปใช้ในการรู้จำตัวอักษร แฟ้มข้อมูลที่ได้ประกอบด้วยข้อมูลของตัวอักษรแต่ละตัวเรียงต่อกันไปตามลำดับ โดยข้อมูลของตัวอักษรแต่ละตัวจะมีลักษณะแตกต่างกันไป ขึ้นอยู่กับลักษณะของตัวอักษรที่ได้ ดังนี้

2.4.4.1. ตัวอักษรธรรมดา คือตัวอักษรที่ไม่มีจุดภาพของตัวอักษรข้างเคียง จะมีลักษณะโครงสร้างของข้อมูลดังแสดงในรูปที่ 2.35

ตำแหน่งของตัวอักษร				ประเภทตัวอักษร	
ขอบซ้าย (x1)	ขอบบน (y1)	ขอบขวา (x2)	ขอบล่าง (y2)		"0"
2 bytes	2 bytes	2 bytes	2 bytes	1 byte	1 byte

รูปที่ 2.35 แสดงโครงสร้างของตัวอักษรปกติ

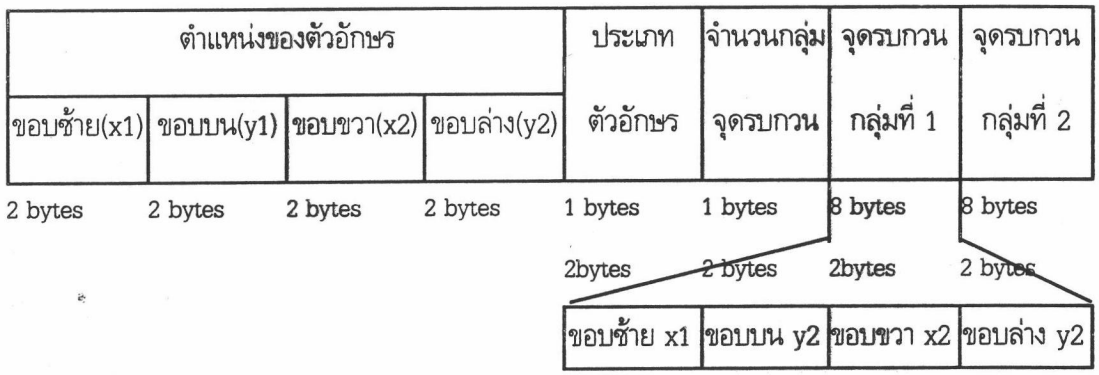
จากตัวอย่างในรูปที่ 2.36 แสดงให้เห็นถึงลักษณะของตัวอักษรธรรมดา และผลลัพธ์ที่ได้เพื่อจัดเก็บลงแฟ้มข้อมูล

(x1)	(y1)	(x2)	(y2)	ประเภท	error
1	1	26	35	1	0
27	1	45	35	1	0

(ก) (ข)

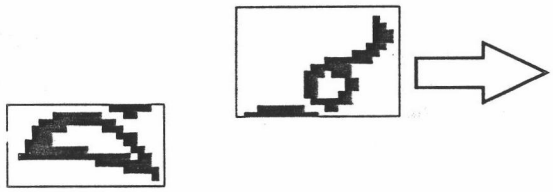
รูปที่ 2.36 ผลลัพธ์ของการจัดเรียงตัวอักษรของตัวอักษรธรรมดา
(ก) ภาพต้นแบบ (ข) ผลลัพธ์ที่ได้

2.4.4.2. ตัวอักษรที่มีจุดภาพของตัวอักษรข้างเคียง คือตัวอักษรที่มีจุดภาพของตัวอักษรข้างเคียง จะมีลักษณะโครงสร้างของข้อมูลดังแสดงในรูปที่ 2.37 โดยที่จำนวนกลุ่มของจุดรบกวนสามารถมีได้สูงสุด 256 กลุ่ม และข้อมูลที่เป็นตำแหน่งของจุดรบกวนต้องมีเท่ากับจำนวนกลุ่มของจุดรบกวนด้วย ซึ่งจากรูปที่ 2.37 แสดงให้เห็นเพียงจำนวนกลุ่มของจุดรบกวน 2 กลุ่ม



รูปที่ 2.37 แสดงโครงสร้างของตัวอักษรที่มีจุดภาพของตัวอักษรข้างเคียง

จากตัวอย่างในรูปที่ 2.38 แสดงให้เห็นถึงลักษณะของตัวอักษรที่มีจุดภาพของตัวอักษรข้างเคียง และผลลัพธ์ที่ได้เพื่อจัดเก็บลงแฟ้มข้อมูล ซึ่งจะมีจุดภาพจากตัวอักษรข้างเคียงอยู่ 1 กลุ่ม

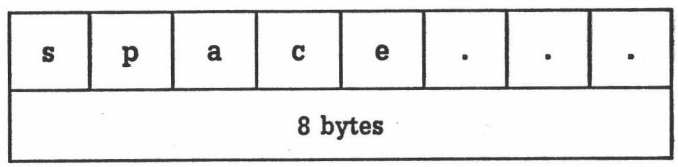


x1	y1	x2	y2	ประเภท	error	err x1	err y1	err x2	err y2
2	10	20	50	3	1	15	10	19	10
13	1	43	13	3	1	8	13	15	13

(ก) (ข)

รูปที่ 2.38 ผลลัพธ์ของการจัดเรียงตัวอักษรของตัวอักษรที่มีจุดภาพจากตัวอักษรข้างเคียง
(ก) ภาพต้นแบบ (ข) ผลลัพธ์ที่ได้

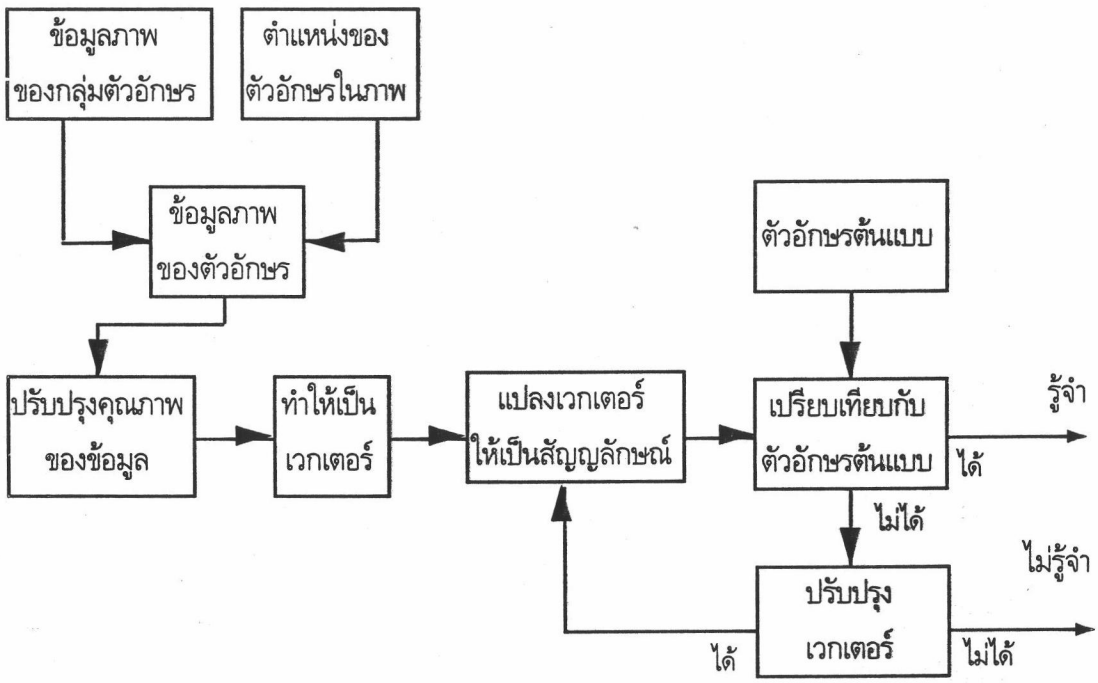
2.4.4.3. ตัวอักษรที่เป็นอักขระเว้นวรรค คือเป็นการเว้นวรรคระหว่างตัวอักษร จะมีลักษณะโครงสร้างของข้อมูลดังแสดงในรูปที่ 2.39



รูปที่ 2.39 ผลลัพธ์ของตัวอักษรที่เป็นอักขระเว้นวรรค

2.5. การรู้จำตัวอักษรไทย

เมื่อได้ผลลัพธ์จากการจัดเรียงตัวอักษรแล้ว ขั้นตอนต่อไปเป็นการรู้จำตัวอักษรไทย ซึ่งมีงานวิจัยทางด้านนี้อยู่บ้างพอสมควรในประเทศไทย โดยในงานวิจัยนี้มุ่งเน้นที่ความถูกต้องในการรู้จำ จึงได้นำเอาวิธีการรู้จำตัวอักษรไทยโดยวิธีซินแทกติก[2] มาใช้ ซึ่งเป็นวิธีที่ค่อนข้างจะซับซ้อน แต่มีความถูกต้องแม่นยำสูง โดยโครงสร้างของระบบการรู้จำแสดงดังรูปที่ 2.40



รูปที่ 2.40 โครงสร้างของระบบการรู้จำโดยวิธีซินแทกติก

ข้อมูลของตัวอักษรที่จะป้อนเข้าสู่ระบบนั้น ได้มาจากข้อมูลภาพของตัวอักษรที่ได้จากเครื่อง scanner และได้ผ่านขั้นตอนการแยกข้อมูลภาพออกเป็นกลุ่มข้อมูลของตัวอักษรแต่ละตัว โดยจะนำข้อมูลที่ได้ป้อนข้อมูลเข้าสู่ระบบทีละ 1 ตัวอักษร ตั้งแต่ตัวอักษรตัวแรก ไปตามลำดับจนถึงตัวอักษรตัวสุดท้าย

ข้อมูลภาพของตัวอักษรที่รับเข้ามาจะผ่านการปรับปรุงคุณภาพของข้อมูล โดยจะทำข้อมูลภาพที่เป็นตัวอักษรนี้ให้เหลือแต่โครงร่างของตัวอักษร จากนั้นก็จะนำข้อมูลเหล่านี้ไปเข้ารหัสเพื่อหาทิศทางของจุดภาพ แล้วแปลงจุดภาพให้เป็น Vector แต่ Vector ที่ได้จะมีรายละเอียดที่ค่อนข้างมาก จึงเปลี่ยนเวกเตอร์ให้เหลือเพียงทิศทางของเวกเตอร์เพียงอย่างเดียว เรียกว่า รูปสัญญลักษณ์ (Primitive) โดยรูปสัญญลักษณ์ที่ได้จะมีการเชื่อมโยงเป็นรูปต้นไม้ และจะมีการจัดเรียงลำดับบัพของต้นไม้ใหม่ เพื่อให้ส่วนหัวของตัวอักษรอยู่ที่จุดแรกสุดของต้นไม้เสมอ

I 16891487

แฟ้มของตัวอักษรต้นแบบ เป็นการนำต้นไม้ของเวกเตอร์และต้นไม้ของรูปสัญญลักษณ์ของตัวอักษรที่เลือกไว้เป็นต้นแบบเก็บไว้ในแฟ้ม วิธีการรู้จำก็เพียงแต่เลือกเอาเฉพาะต้นไม้ของตัวอักษรต้นแบบที่มีส่วนหัวอยู่ในเขตเดียวกันกับตัวอักษรที่ต้องการรู้จำนำมาเปรียบเทียบ ถ้าผลการเปรียบเทียบกับต้นไม้ของตัวอักษรต้นแบบตัวไหนให้ค่าความแตกต่างน้อยที่สุด และมีค่าความแตกต่างที่ต่ำกว่าค่าที่ยอมรับได้ แสดงว่าตัวอักษรที่นำมารู้จำมีความเหมือนกับตัวอักษรต้นแบบนั้นมากที่สุด

ในกรณีที่ผลการเปรียบเทียบไม่เป็นที่พอใจ ต้นไม้ของเวกเตอร์จะถูกนำไปปรับปรุงคุณภาพของเวกเตอร์ โดยการตัดเวกเตอร์ที่มีจุดปลายที่มีความยาวน้อยที่สุด หรือต่อเชื่อมเวกเตอร์ที่มีจุดปลายอยู่ใกล้กันมาก ๆ เข้าด้วยกัน ถ้าสามารถปรับปรุงเวกเตอร์ได้ ต้นไม้ของเวกเตอร์จะถูกนำไปแปลงให้เป็นต้นไม้ของรูปสัญญลักษณ์ และนำกลับไปเปรียบเทียบกับตัวอักษรต้นแบบ แต่ถ้าไม่สามารถปรับปรุงเวกเตอร์ได้ แสดงว่าไม่สามารถรู้จำตัวอักษรที่รับเข้ามาได้

สำหรับตัวอักษรไทยที่รู้จำได้นั้น จะถูกแทนค่าด้วยค่าที่ถูกกำหนดไว้แล้วในตัวอักษรต้นแบบของตัวอักษรนั้น ๆ และสำหรับตัวอักษรไทยที่ไม่สามารถจะวิเคราะห์ได้ว่าเป็นตัวอักษรอะไรนั้น จะถูกแทนค่าด้วยเครื่องหมาย "_" เพื่อให้รู้ว่าเป็นตัวอักษรที่ไม่สามารถวิเคราะห์ได้ ซึ่งอาจเกิดจากการที่การแบ่งกลุ่มของข้อมูลภาพของตัวอักษร หรือการรู้จำตัวอักษรไม่สามารถหาค่าได้

2.6. การแก้ไขรหัสของตัวอักษร

ภาษาไทยที่ใช้อยู่ในปัจจุบัน ได้มีการกำหนดเป็นตารางรหัสแอสกี(ASCII Table) โดยนำตัวอักษรไทยที่มีอยู่ทั้งหมดกำหนดใส่ในตาราง

2.6.1. ตารางรหัสแอสกีภาษาไทย[5] เนื่องจากรหัสภาษาไทยที่ใช้งานในปัจจุบันมีอยู่ด้วยกัน 2 แบบ คือ รหัสภาษาไทยมาตรฐานสมอ. และรหัสภาษาไทยมาตรฐานเกษตร ในการวิจัยนี้ใช้ มาตรฐานสมอ. เป็นที่นิยมใช้กัน โดยรหัสภาษาไทยที่ใช้นั้นจะอยู่ในตารางแอสกีตั้งแต่ค่าที่ 161 ถึง 250 (ในเลขฐานสิบ) ในตารางประกอบด้วยตัวอักษรไทยและตัวอักษรภาษาอังกฤษ ซึ่งตัวอักษรไทยนั้นประกอบด้วยพยัญชนะ สระ วรรณยุกต์ ตัวเลขไทย และตัวอักขระพิเศษ ที่ใช้ในภาษาไทยทั้งหมด สามารถดูได้ในภาคผนวก ก.

2.6.2. ตัวอักษรที่มีการแก้ไข จากตารางแอสกีในภาคผนวก ก. พบว่ามีตัวอักษรบางตัวที่ต้องทำการแก้ไข ดังนี้

2.6.2.1. ตัว ฐ ประกอบด้วย จ และ ฐ

2.6.2.2. ตัว ญ ประกอบด้วย ญ และ ◌ั

2.6.2.3. สระ แ ประกอบด้วย เ และ เ

2.6.2.4. สระ ำ ประกอบด้วย ◌ั และ า

2.6.2.5. สระ ะ ประกอบด้วย ◌ั และ ◌ะ

การแก้ไขตัวอักษรดังกล่าวจะเริ่มด้วยการตรวจสอบผลลัพธ์ที่ได้จากการรู้จำมาทีละ 1 ตัว หากพบว่าตัวอักษรที่ได้นั้นต้องประกอบด้วยตัวอักษร 2 ตัว ก็จะนำผลลัพธ์ที่ได้จากการรู้จำตัวถัดไปมาทำการแก้ไขรวมเป็นตัวอักษรเพียงตัวเดียว