

รู้จำรหัสไปรษณีย์

Postal Code Recognition



โดย

นาย ธีรพงศ์ เทียนดี 41013169

นาย ทศพร ศรีสง่างามกุล 41013172

อาจารย์ที่ปรึกษา

อาจารย์ สุรเดช ตรีไตรลักษณ์

ปริญญาโท สาขาเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เลขหม.....

เลขทะเบียน.....42282

วัน, เดือน, ปี 16 พ.ค. 2545

.b.....

.i.....

ปริญญาบัตรประจำปีการศึกษา 2543

สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง รู้จำรหัสไปรษณีย์ (Postal Code Recognition)

ผู้จัดทำ

1. นาย ณัฐพงศ์ เทียนดี
2. นาย ทศพร ศรีสง่างามกุล



อาจารย์ สุรเดช ตรีไตรลักษณ์
(อาจารย์ที่ปรึกษา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รื้อจักรหัสไปรษณีย์

นาย ณัฐพงศ์ เทียนดี

นาย ทศพร ศรีสง่างามกุล

อาจารย์ สุรเดช ตรีไตรลักษณ์

(อาจารย์ที่ปรึกษา)

2544

บทคัดย่อ

ระบบรื้อจักรหัสไปรษณีย์นี้ ได้ถูกออกแบบให้สามารถทำการประมวลผลสัญญาณภาพของจดหมายเพื่อหาตำแหน่งของเลขรหัสไปรษณีย์ที่ได้ถูกเขียนลงบนซองจดหมายนั้นๆได้ และทำการตัดสินใจว่าเลขรหัสทำการแยกมาได้นั้นเป็นตัวเลขค่าใด เพื่อไปทำการประมวลผลตัดสินใจเป็นเลขรหัสไปรษณีย์ของเขตอำเภอใด โดยรายงานฉบับนี้จะนำเสนอในส่วนการศึกษาของการประมวลผลภาพ และกระบวนการสร้างโปรแกรมเพื่อใช้เป็นโปรแกรมวิเคราะห์โครงสร้างรูปแบบโดยภาษาโปรแกรมวิซวลเบสิก

Postal Code Recognition

Mr. Nattapong Tiendee

Mr. Tossaporn Srisangamkul

Mr. Suradech Tritrailuksana

(Advisor)

2544

ABSTRACT

Postal Code Recognition system was designed to have capability of Image processing on the mail's envelope to detect the position of the Postal code number which written on that envelope and recognize which numbers are from the Postal code number, then these numbers will be throught to computer and find out which district area of these number represent. This report is present about Image processing research and Pattern Recognition by Structural Analysis Method and all of this funtion is make up by Visual Basic Language.

สารบัญ

บทที่ 1 บทนำ

- 1.1 วัตถุประสงค์ของโครงการ 2
- 1.2 ขอบเขตของโครงการ 2
- 1.3 การประยุกต์ใช้งานระบบรู้จำรหัสไปรษณีย์ 3

บทที่ 2 ทฤษฎีและหลักการเบื้องต้น

- 2.1 การประมวลผลภาพเชิงตัวเลข(Digital Image Processing) 4
 - 2.1.1 การแทนภาพด้วยข้อมูลแบบดิจิทัล 4
 - 2.1.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิทัล 5
- 2.2 ไฟล์ข้อมูลภาพชนิดบิตแมป 6
 - 2.2.1 รูปแบบของไฟล์ข้อมูลภาพชนิดบิตแมป 6
 - 2.2.2 โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป 6
 - 2.2.3 การจัดเก็บไฟล์ข้อมูลชนิดบิตแมป 7
- 2.3 การสร้างภาพไบนารี 7
- 2.4 การเตรียมข้อมูลภาพ(Pre-processing) 9
 - 2.4.1 การกำจัดสัญญาณรบกวน 9
 - 2.4.2 วิธีการ วนตามหากรอบภาพ (Contour Following) 10
 - 2.4.3 วิธีการตรวจสอบหามุมภาพ (Detect Edge) 11
 - 2.4.4 กระบวนการทำเส้นบาง(Thinning) 11
 - 2.4.5 กระบวนการกำจัดส่วนเกิน 12
- 2.5 ระบบรู้จำโดยใช้การวิเคราะห์โครงสร้าง 13
 - 2.5.1 จุดแยกมีลักษณะดังยกมาเป็นตัวอย่างต่อไปนี้ 14
 - 2. 5.2 จุดปลายมีลักษณะดังต่อไปนี้ 14
 - 2.5.3 ลักษณะของจุดตัดดังแสดงต่อไปนี้ 15
 - 2.5.4 จำนวนจุดปลาย 15
 - 2. 5.5 จำนวนจุดแยกหรือจุดตัด 15
 - 2.5.6 ทิศทางการเดินทางของเส้นที่ต่อจากจุดปลาย 16
 - 2.5.7 ตำแหน่งของจุดปลาย 16
 - 2.5.8 อัตราส่วนระหว่างความสูงกับความกว้างของภาพ 16

บทที่ 3 โครงสร้างระบบและโครงสร้างโปรแกรม

- 3.1 โครงสร้างของระบบรู้จำรหัสไปรษณีย์ 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ระบบส่วนการประมวลผลภาพ	20
3.2.1 การเตรียมข้อมูลภาพ	20
3.2.2 การกำจัดสิ่งรบกวน (Noise Removing)	20
3.2.2.1 โครงสร้างโปรแกรมการกำจัดสัญญาณรบกวน	22
3.2.3 การแยกภาพด้วยวิธีวนตามหากรอบภาพ (Contour Following)	22
3.2.3.1 โครงสร้างโปรแกรมการวนตามหากรอบภาพ	24
3.2.4 การแยกข้อมูลภาพที่อยู่ในกรอบหรือขอบภาพด้วยวิธีการ ตรวจสอบมุมภาพ (Detect Edge)	25
3.2.4.1 โครงสร้างโปรแกรมการตรวจสอบมุมภาพ	26
3.2.5 กระบวนการทำเส้นบาง(Thinning)	27
3.2.5.1 โครงสร้างโปรแกรมกระบวนการทำเส้นบาง	27
3.2.6 กระบวนการกำจัดส่วนเกิน	29
3.2.6.1 โครงสร้างโปรแกรมกระบวนการกำจัดส่วนเกิน	29
3.3 ระบบส่วนการรู้จำ	29
3.3.1 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 0	31
3.3.2 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 1	31
3.3.3 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 2	32
3.3.4 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 3	34
3.3.5 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 4	35
บทที่ 5 การทดลองและผลการทดลอง	
ตอนที่ 1 การทดลองแยกข้อมูลภาพตัวเลขออกจากกรอบ	36
ตอนที่ 2 การทดลองให้โปรแกรมบอกค่าของตัวเลขจากภาพตัวเลข	38
ตอนที่ 3 การทดลองให้โปรแกรมบอกเลขรหัสไปรษณีย์จากภาพตัวเลข	42
บทที่ 6 สรุปและวิจารณ์	
6.1 ส่วนของการเตรียมข้อมูล	45
6.2 ส่วนการประมวลผลภาพ	45
6.3 ส่วนระบบการวิเคราะห์รูปแบบ	45
ภาคผนวก ก. โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป	47
ภาคผนวก ข. ตัวโปรแกรม	50
ภาคผนวก ค. รูปแบบตัวเลขที่เก็บข้อมูลได้ และ Flow Chart รวม	80
กิตติกรรมประกาศ	

สารบัญรูปภาพ

รูปที่ 1	ผังการทำงานของระบบโดยสังเขป	2
รูปที่ 2	แสดงตัวอย่างสัญญาณรบกวนที่เป็นจุดเดี่ยว	9
รูปที่ 3	แสดงภาพที่มีสัญญาณรบกวนต่ำ	10
รูปที่ 4	แสดงภาพที่มีสัญญาณรบกวนสูง	10
รูปที่ 5	แสดงการทำ Contour Following ในภาพ เป็นการหากรอบของรูปสี่เหลี่ยมผืนผ้าที่มีช่องว่างตรงกลาง	11
รูปที่ 6	แสดงรูปการแทนหาจุดมุมเพื่อตีกรอบวัตถุที่ต้องการ	11
รูปที่ 7	แสดงขั้นตอนการทำเส้นบางอย่างมีเสถียรภาพ	12
รูปที่ 8	แสดงภาพที่เกิดส่วนเกินหลังจากผ่านการทำเส้นบางมาแล้ว (ก) ภาพก่อนทำการกำจัดส่วนเกิน มีจุดแยกที่เป็นส่วนเกินอยู่หลายจุด (ข) ภาพหลังจากผ่านขั้นตอนการกำจัดส่วนเกินเรียบร้อยแล้ว	13
รูปที่ 9	แสดงตัวอย่างจุดแยกแบบต่างๆ	14
รูปที่ 10	แสดงลักษณะของจุดปลายแบบต่างๆ	14
รูปที่ 11	แสดงลักษณะของจุดตัดแบบต่างๆ	15
รูปที่ 12	แสดงตัวอย่างการนับจุดปลายของตัวเลขต่างๆ	15
รูปที่ 13	แสดงตัวอย่างการนับจุดแยกและจุดตัดของตัวเลขต่างๆ	15
รูปที่ 14	แสดงทิศทางการเดินทางของเส้นที่ต่อไปจากจุดปลาย(x0)	16
รูปที่ 15	แสดงตัวอย่างตำแหน่งของจุดปลายในควอดเรนต์ทั้งสี่	16
รูปที่ 16	แสดงตัวอย่างเปรียบเทียบอัตราความสูงต่อความกว้างของเลข 1 กับ 7	16
รูปที่ 17	โครงสร้างขั้นตอนการทำงานของระบบโดยรวม	19
รูปที่ 18	ข้อมูลภาพที่ได้แบบไบนารีที่ทำการเตรียมมา	20
รูปที่ 19	ตัวอย่างการแทนภาพที่มีสัญญาณรบกวน	21
รูปที่ 20	ตัวอย่างภาพที่ผ่านการกำจัดสัญญาณรบกวนแล้ว	21
รูปที่ 21	โครงสร้างโปรแกรมกำจัดสัญญาณรบกวน	22
รูปที่ 22	ภาพแสดงหลักการทำงานของกระบวนการ Contour Following	23
รูปที่ 23	โครงสร้างโปรแกรมการวนตามหากรอบภาพ	24
รูปที่ 24	ภาพแสดงหลักการทำงานของกระบวนการ Detect Edge	25
รูปที่ 25	ข้อมูลบล็อกเดี่ยวของภาพตัวเลขที่แยกออกมาได้	

และเตรียมทำกระบวนการทำเส้นบาง

25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 26 โครงสร้างโปรแกรมการตรวจสอบหามุมภาพ	26
รูปที่ 27 โครงสร้างโปรแกรมกระบวนการทำเส้นบาง	27
รูปที่ 28 โครงสร้างโปรแกรมกระบวนการกำจัดส่วนเกิน	29
รูปที่ 29 โครงสร้างระบบส่วนการรู้จำโดยสังเขป	30
รูปที่ 30 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 0	31
รูปที่ 31 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 1	31
รูปที่ 32 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 2	32
รูปที่ 33 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 3	34
รูปที่ 34 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 4	35
รูปที่ 35 รูป Interface โปรแกรมเมื่อทำการ Run ขึ้นมา	37
รูปที่ 36 รูป Interface โปรแกรมเมื่อทำการ Import ภาพเข้ามาแล้ว	37
รูปที่ 37 รูป Interface โปรแกรมเมื่อทำการกดปุ่มแสกนหากรอบภาพ	38
รูปที่ 38 รูปการขณะทำการประมวลผลในขั้นตอนหาค่าตัวเลขจากข้อมูลภาพ	39
รูปที่ 39 รูปการขณะทำการประมวลผลในขั้นตอนหาเลขรหัสไปรษณีย์จากข้อมูลภาพ	42
รูปที่ ก-1 โครงสร้างของ Bitmap File	47
รูปที่ ก-2 แสดงการเก็บข้อมูลของแต่ละพิกเซล	49
รูปที่ ข-1 แสดงภาพ Interface ของโปรแกรมที่สร้างขึ้น	50
รูปที่ ค-1 แสดงรูปแบบตัวเลขที่เก็บข้อมูลมาได้และเป็นคำตอบใน Flow Chart	
โดยรวม	80
รูปที่ ค-2 แสดง Flow Chart โดยรวมทั้งหมด	81

บทที่ 1

บทนำ

ในปัจจุบันนี้ เทคโนโลยีการสื่อสารก้าวหน้าไปมาก และการติดต่อก็สะดวกสบายจับไปมากขึ้นตามลำดับ ดังจะเห็นได้จากประเภทของการสื่อสารที่มีหลายหลายประเภท ไม่ว่าจะเป็นการสื่อสารโทรคมนาคมทางโทรศัพท์ ทางดาวเทียม การสื่อสารไร้สาย หรือแม้กระทั่งการสื่อสารผ่านเครือข่ายอินเทอร์เน็ตที่นับวันเริ่มจะมีแนวโน้มก้าวเข้ามาในชีวิตของพวกเรามากขึ้นทุกวันทุกวัน แต่ในหลายประเภทของการสื่อสารดังที่ได้กล่าวมาแล้วนั้น ยังมีการสื่อสารติดต่อกันที่เป็นพื้นฐานของการติดต่อสื่อสารระหว่างมนุษย์ด้วยกันมาตั้งแต่อดีตจนปัจจุบันก็มีผู้ใช้กันอยู่ทั่วไปไม่เปลี่ยนแปลง นั่นคือการติดต่อสื่อสารผ่านทางไปรษณีย์นั่นเอง ตั้งแต่อดีตจนปัจจุบันจำนวนการใช้การสื่อสารประเภทนี้ก็ยังมีมากมายไม่เปลี่ยนแปลง จากความมากมายของการติดต่อสื่อสารของผู้คนในสังคมด้วยระบบไปรษณีย์ ทำให้เกิดปัญหาในกระบวนการคัดแยกของจดหมายเป็นอย่างมาก อันเนื่องมาจากความมากมายของจำนวนผู้ใช้บริการ ความหลากหลายของผู้ส่งไปยังผู้รับในที่ต่างๆ จึงทำให้เกิดการต้องคัดแยกของจดหมายขึ้น เพื่อส่งไปยังที่หมายได้ถูกต้อง

จากอดีตการคัดแยกของจดหมายเพื่อส่งไปยังจุดหมายให้ถูกต้องนั้น มีกระบวนการคัดแยกโดยมนุษย์ และในสมัยต่อมา ได้มีการพัฒนาโดยใช้เครื่องคัดแยกของจดหมาย ซึ่งก็ได้ผลดี มีความรวดเร็วและประหยัดแรงงานขึ้นมาก แต่ถึงกระนั้น ก็ยังมีความผิดพลาดเกิดขึ้นบ้าง และในกระบวนการที่เครื่องมือไม่สามารถตัดสินใจได้ ก็ยังใช้มนุษย์ในการพิจารณาตัดสินใจในการคัดแยกอยู่ พูดได้ว่าไม่เป็นระบบที่เครื่องมือสามารถกระทำกระบวนการได้ด้วยตัวมันเอง ร้อยเปอร์เซ็นต์นั่นเอง

จากกระบวนการในการตัดสินใจของเครื่องมือที่ใช้ในการคัดแยกของจดหมาย สิ่งสำคัญที่สุดคือการทำให้คอมพิวเตอร์รู้จักลายมือมนุษย์ ซึ่งคนเราแต่ละคนลายมือไม่เหมือนกัน พูดย่างาก็คือ ถ้าให้คนสิบคนเขียนเลข 9 ด้วยลายมือ สิบคนย่อมเขียนไม่เหมือนกันแน่นอน จุดนี้เองที่เราต้องทำให้คอมพิวเตอร์ตัดสินใจให้เข้าใจว่าสิบลายมือแบบลายมือนี้คือเลข 9 เหมือนกัน เราอาจใช้ไมโครคอนโทรลเลอร์ หรือคอมพิวเตอร์ในการควบคุมกระบวนการในการตัดสินใจ แต่กระนั้นก็ยังเกิดความผิดพลาดขึ้นมา เพราะคอมพิวเตอร์ไม่ใช่มนุษย์ จากความผิดพลาดนี้ ทำให้เกิดโครงการนี้ขึ้นมา คือโครงการการศึกษาการรู้จำรหัสไปรษณีย์ อันจะทำการศึกษาเพื่อประโยชน์ในกระบวนการในส่วนการตัดสินใจของคอมพิวเตอร์ให้ถูกต้องมากยิ่งขึ้น โดยในการจะทำให้คอมพิวเตอร์จดจำรูปแบบได้นั้น ข้อมูลเบื้องต้นที่จะทำการป้อนไว้กับการจดจำของคอมพิวเตอร์จะอยู่ในลักษณะของข้อมูลรูปภาพ (Image Data) และนำไปแปลงให้อยู่ในรูปของการจัดเก็บข้อมูลแบบตัวหนังสือ (Text Data) เพื่อนำข้อมูลที่ได้ออกไปทำการประมวลผลคัดแยกรหัสไปรษณีย์ต่อไป โดยกระบวนการก่อนนี้จะทำให้คอมพิวเตอร์รู้จักกับตัวเลขรหัส เพื่อแปลงตัวเลขรหัสจากข้อมูลรูปภาพเป็นข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบตัวหนังสือได้นั้น เราจะใช้หลักการของการประมวลผลภาพ (Image Processing) และการจดจำรูปแบบ (Pattern Recognition) เป็นหลักสำคัญในการออกแบบระบบ

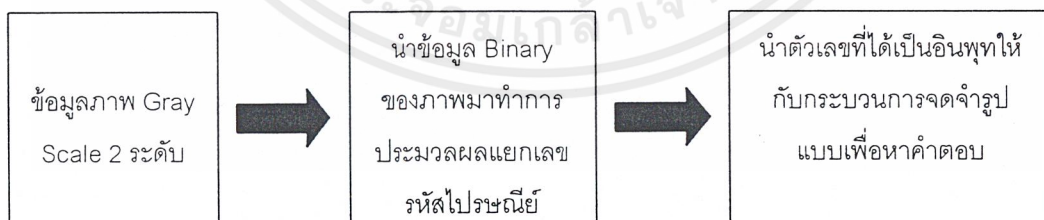
1.1 วัตถุประสงค์ของโครงการ

- เพื่อศึกษาหลักการประมวลผลภาพ
- เพื่อศึกษาหลักการจดจำรูปแบบ
- เพื่อศึกษาการเขียนโปรแกรมประยุกต์ใช้งานด้วย Visual Basic

1.2 ขอบเขตของโครงการ

ในโครงการนี้เราจะทำการศึกษาในส่วนของ การประมวลผลภาพ โดยเป้าหมายคือเพื่อให้ได้ข้อมูลที่จะนำไปเป็นอินพุตให้แก่ระบบจดจำรูปแบบ และระบบจดจำรูปแบบนั้นจะทำการวิเคราะห์ข้อมูลภาพและความสามารถของโปรแกรมที่เราจะทำคือให้โปรแกรมสามารถตอบได้ว่าข้อมูลภาพตัวเลขที่เราป้อนให้มันมีค่าเท่ากับเท่าใด

โดยขั้นต้นเราจะนำรูปภาพของรูปของจดหมายที่ได้มีการเขียนที่อยู่ผู้รับ และรหัสไปรษณีย์ด้วยลายมือคนไว้แล้ว นำไปทำการแตกจนแล้วเก็บข้อมูลมาในแบบบิตแมป(Bitmap) Gray Scale 2 ระดับ คือสีขาวกับสีดำ มาทำการประมวลผลแยกรูปภาพตัวเลขรหัสไปรษณีย์ที่ปะปนกับรูปภาพตัวหนังสือที่อยู่ในซองจดหมายเดียวกันออกมาให้ได้ โดยใช้กระบวนการประมวลผลภาพเช่น การทำไปนาไประชัน (Image Binarization) การจำแนกภาพ (Image Segmentation) โดยกระบวนการ Contour Following และ detect edge เพื่อนำผลลัพธ์ที่ได้ไปป้อนเป็นอินพุตให้กับกระบวนการจดจำรูปแบบรหัสไปรษณีย์ต่อไป



รูปที่ 1 ผังการทำงานของระบบโดยสังเขป

หลังจากผ่านขั้นตอนการแยกรูปตัวเลขรหัสไปรษณีย์ออกจากซองจดหมายแล้ว ข้อมูลภาพของรหัสไปรษณีย์จะถูกเตรียมพร้อมเพื่อเข้าสู่กระบวนการจดจำรูปแบบ เพื่อให้คอมพิวเตอร์ทำการตัดสินข้อมูลรูปภาพรหัสไปรษณีย์ออกมาเป็นข้อมูลตัวเลขที่เป็นแบบตัวอักษร ได้อย่างถูกต้อง โดยกระบวนการในส่วนของการจด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จํารูปแบบ โดยการวิธีการจดจําที่เราจะใช้คือกระบวนการรู้จําโดยการวิเคราะห์โครงสร้าง(Structural Analysis Method)

1.3 การประยุกต์ใช้งานระบบรู้จํารหัสไปรษณีย์

- ระบบคัดแยกของจดหมาย

ระบบรู้จํารหัสไปรษณีย์สามารถนำไปใช้กับระบบคัดแยกของจดหมายของกรมไปรษณีย์ได้ ซึ่งจะทำให้การคัดแยกของจดหมายไปสู่จุดหมายปลายทางสะดวกรวดเร็วยิ่งขึ้น โดยเมื่อทำการผ่านของจดหมายเข้ามาในระบบรู้จําเพื่อให้คอมพิวเตอร์ทำการวิเคราะห์แล้วตัดสินใจว่าเป็นรหัสไปรษณีย์ของอำเภอใด จังหวัดไหน แล้วนำรหัสที่วิเคราะห์ได้ไปเทียบกับฐานข้อมูล เมื่อตรวจเจอว่าจับตัวเลขรหัสจับคู่ถูกต้องกับรหัสไปรษณีย์ใด ก็จะทำการคัดแยกของจดหมายส่งต่อคัดแยก พร้อมส่งไปตามที่หมายต่างๆต่อไปได้

- พัฒนาเพื่อไปสู่ระบบรู้จําที่ซับซ้อนกว่า

จากการศึกษาระบบประมวลผลภาพและระบบรู้จําแบบต่างๆ ทำให้เราเข้าใจและสามารถนำระบบต่างๆที่ศึกษาไปประยุกต์ใช้กับระบบรู้จําที่ซับซ้อนกว่ายิ่งขึ้นไปได้เช่น ระบบรู้จําป้ายทะเบียนรถยนต์ ซึ่งต้องจดจําตัวอักษร ระบบรู้จําลายมือมนุษย์แบบออนไลน์ หรือไปจนถึงระบบ OCR แบบต่างๆต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการเบื้องต้น

2.1 การประมวลผลภาพเชิงตัวเลข(Digital Image Processing)

การประมวลผลภาพเชิงตัวเลข หมายถึง การนำภาพที่พบทั่วไปมาประมวลผลด้วยเครื่องคอมพิวเตอร์ โดยภาพที่นำมาประมวลผลด้วยเครื่องคอมพิวเตอร์นั้นจะถูกแทนที่ด้วยตัวเลขให้อยู่ในรูปของเมตริกซ์ แต่ภาพที่ได้โดยส่วนมากแล้วจะเป็นภาพที่ได้จากตัวรับสัญญาณ ซึ่งอยู่ในรูปของฟังก์ชัน $f(x,y)$ ที่ต่อเนื่องในระนาบสองมิติ (คือแกน x และแกน y) โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพ ที่ตำแหน่ง (x,y) ใดๆ ซึ่งเรียกว่าระดับสีเทา (Gray Level)

2.1.1 การแทนภาพด้วยข้อมูลแบบดิจิทัล

ภาพข้อมูลแบบดิจิทัล (Digital Image) เป็นภาพที่ถูกแปลงมาจากอนาลอก ให้อยู่ในรูปของตัวเลข โดยภาพอนาลอกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็กๆ ที่เรียกว่า พิกเซล(Pixel) ในแต่ละพิกเซล จะถูกระบุตำแหน่งด้วยคู่ออร์ดิเนต (x,y) และค่าระดับสีเทาของพิกเซลนั้นๆ โดยเราสามารถแปลงภาพเป็นข้อมูลแบบดิจิทัลได้ โดยมีขั้นตอนและวิธีการดังนี้

เมื่อเรานำสัญญาณอนาลอกที่ต้องการประมวลผลมาผ่านส่วนที่เรียกว่า ดิจิไทเซอร์ (Digitizer) ซึ่งจะมีหน้าที่ในการเปลี่ยนสัญญาณอนาลอกให้เป็นสัญญาณดิจิทัล ไม่ว่าจะเป็นแสกนเนอร์ หรือกล้องดิจิไทเซอร์ จากนั้นทำการควอนไทซ์ (Quantizing) เพื่อที่จะประมวลสัญญาณด้วยระบบคอมพิวเตอร์ ฟังก์ชันของภาพ $f(x,y)$ จะถูกทำให้เป็นสัญญาณไม่ต่อเนื่องทั้งระนาบของภาพ ซึ่งเราเรียกว่าการสุ่มภาพ (Image Sampling) ของฟังก์ชันที่ได้เรียกว่า การควอนไทซ์ระดับสีเทา (Gray Level Quantization) ก็จะได้ข้อมูลที่เป็นดิจิทัล

สมมติว่าสัญญาณภาพต่อเนื่อง $f(x,y)$ ถูกดิจิไทซ์ในระนาบ x และ y เป็นช่วงเท่าๆกัน เราสามารถจัด $f(x,y)$ ให้อยู่ในรูปของเมตริกซ์ ขนาด $N \times N$ ได้ดังสมการที่ 2.1

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,2) \dots f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) \dots f(1,N-1) \\ \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & f(N-1,2) \dots f(N-1,N-1) \end{matrix} \quad \text{สมการที่ 2.1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งทางขวาของสมการ จะเรียกได้ว่า ข้อมูลภาพดิจิทัล และทุกๆ สมาชิกของเมตริกซ์ จะเรียกว่า พิกเซล จากขบวนการสร้างภาพดิจิทัลข้างต้น จะเห็นได้ว่าเราสามารถทราบขนาด ของความละเอียดของ ภาพ $N \times N$ พิกเซล และจำนวนระดับของเกรย์สเกล ในทางปฏิบัติการทำคอนโตนเซชันในระบบภาพดิจิทัล จะมีค่าดังสมการที่ 2.2

$$B = N \times N \times M$$

สมการที่ 2.2

เมื่อ $B =$ ขนาดของข้อมูลภาพที่เป็นดิจิทัล
 $G =$ จำนวนของเกรย์สเกล ที่ต้องการใช้ในการเก็บข้อมูลภาพ
 $M =$ จำนวนบิตที่ใช้ในการแทนข้อมูลภาพ 1 พิกเซล
 โดย M สามารถหาได้จาก
 $G = 2^M$

2.1.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิทัล

โดยทั่วไปแล้ว ข้อมูลภาพจะมีค่าความเข้มตั้งแต่ 2 ระดับขึ้นไป แต่ที่ใช้กันมากจะใช้กันที่ค่าระดับ ความเข้มของจุดภาพเท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของจุดภาพอยู่ในช่วง (0-255) โดยใช้เนื้อที่การเก็บข้อมูลภาพขนาด 1 ไบต์ หรือ 8 บิต ถ้าสำหรับข้อมูล i จุดภาพ ($2^8 = 256$) ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มสูงๆ อาจจะต้องการจำนวนบิตสำหรับการเก็บข้อมูลมากกว่า 8 บิต คืออาจจะเป็น 16 หรือ 24 บิต โดยค่าความเข้มของจุดภาพจะเท่ากับ 2^{16} และ 2^{24} ตามลำดับ โดยจะแยกความแตกต่างของแต่ละระดับให้เห็นอย่างชัดเจนได้ดังนี้

1. ภาพ 2 ระดับ คือมีเพียงแค่จุดขาวกับจุดดำเท่านั้น โดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต โดยลักษณะการเก็บข้อมูลภาพแบบนี้เราได้เลือกใช้ในโครงงานนี้นั่นเอง
2. ภาพ 16 ระดับ คือในแต่ละจุดภาพจะมีขนาดของข้อมูล 4 บิต ซึ่งทำให้สามารถแสดงได้ถึง 16 ระดับสี หรือ Gray Scale 16 Levels ขึ้นอยู่กับภาพนั้นว่าเป็นสีหรือขาวดำ
3. ภาพ 256 ระดับ คือในแต่ละจุดภาพจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงภาพได้ ความเข้มถึง 256 ระดับสี หรือ Gray Scale 256 Levels ขึ้นอยู่กับภาพนั้นว่าเป็นสีหรือขาวดำ
4. ภาพทิวทัศน์ (True Color) คือในแต่ละจุดภาพจะมีขนาดของข้อมูล 24 บิต ทำให้แสดงภาพออกมาได้เหมือนจริงมากที่สุด เพราะสามารถแสดงสีได้ถึง 16,777,216 สี ภาพทิวทัศน์สามารถแสดงได้เฉพาะภาพสีเท่านั้น ไม่สามารถแสดงผลเป็นภาพขาวดำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลภาพนี้ใช้วิธีตั้งค่าของแม่สีในตารางสี โดยอาจเลือกสีเป็นแบบ 16 สีจาก 64 สี หรือ 16 สี จาก 262,144 สี หรือ 256 สีจาก 262,144 สี ขึ้นอยู่กับโหมดการแสดงผล สำหรับทรูคัลเลอร์ จะไม่มีการเลือกสีแสดงผล โดยการส่งค่าสี RGB ผ่าน D/A ทีละ 8 บิต ออกไป ความแตกต่างของการแสดงผลและขาวดำ คือ ภาพขาวดำจะต้องตั้งให้แม่สีทั้งสามสี มีค่าเท่ากัน เนื่องจาก VGA กำหนดให้แม่สีแต่ละสีใช้ได้เพียง 64 ระดับเท่านั้น หากต้องการให้เห็นจริงทั้ง 256 ระดับนั้น ต้องแสดงในโหมดทรูคัลเลอร์ แล้วให้ RGB มีค่าเท่ากัน ซึ่งในโหมดนี้ จะสามารถใช้วีจีทีเออร์ได้ 8 บิต สำหรับแต่ละแม่สี รวมเป็น ทรูคัลเลอร์ 24 บิตนั่นเอง

โดยทั่วไปวิธีการประมวลผลภาพเชิงตัวเลขที่ทำให้คอมพิวเตอร์สามารถรู้จักวัตถุภายในภาพได้นั้นพอจะแบ่งได้สองระดับด้วยกันคือ การประมวลผลภาพในระดับต่ำ (Low-Level Image Processing) และการประมวลผลภาพในระดับสูง (High-Level Image Processing) การประมวลผลภาพในระดับต่ำจะเป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมด เพื่อหาตัวแปรต่างๆ มาอธิบายข้อมูลภาพ โดยมีจุดประสงค์เพื่อนำตัวแปรเหล่านั้นไปใช้ในการประมวลผลภาพระดับสูงต่อไป โดยทั่วไปแล้วการประมวลผลภาพระดับต่ำจะประกอบไปด้วย การประมวลผลภาพก่อน (PreProcessing) การกำจัดสัญญาณรบกวนหรือการทำให้ภาพคมชัด การหาขอบภาพ เป็นต้น

การประมวลผลระดับสูง เป็นการนำผลลัพธ์หรือสัญลักษณ์ที่ได้จากการประมวลผลระดับต่ำมาตีความหรือประมวลผลเพื่อนำให้คอมพิวเตอร์สามารถรู้จักและเข้าใจภาพได้สำหรับความแตกต่างของการประมวลผลภาพระดับต่ำและการประมวลผลภาพระดับสูงนั้นคือ ข้อมูลที่นำมาใช้ในการประมวลผลภาพ โดยที่การประมวลผลภาพระดับต่ำจะใช้ค่าความสว่างของจุดโดยตรง ส่วนการประมวลผลภาพระดับสูงนั้น ข้อมูลภาพที่นำมาประมวลผลจะถูกแสดงในรูปของสัญลักษณ์ ซึ่งสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่างๆ ที่มีอยู่ในภาพเช่น ขนาดของวัตถุ รูปร่าง และความสัมพันธ์กันระหว่างวัตถุภายในภาพ

2.2 ไฟล์ข้อมูลภาพชนิดบิตแมป

2.2.1 รูปแบบของไฟล์ข้อมูลภาพชนิดบิตแมป

รูปแบบของไฟล์ข้อมูลภาพชนิดบิตแมป เป็นฟอร์แมตของวินโดวส์บิตแมป (Bitmap) ซึ่งเป็นมาตรฐานสำหรับไฟล์กราฟิกบนระบบปฏิบัติการวินโดวส์ ซึ่งจะใช้ในการตัดต่อภาพ หรือสำเนาต่างๆ ลงบนคลิปบอร์ด(Clipboard) เมื่อเวลาจัดเก็บไฟล์ที่มีสกุล BMP ซึ่งเป็นฟอร์แมตนี้ยังสามารถใช้เป็นวอลเปเปอร์ (Wallpaper) ของวินโดวส์ได้อีกด้วย

2.2.2 โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป

โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป จะประกอบด้วย 3 ส่วนคือ

1. ข้อมูลเฮดเดอร์(Header)
2. ข้อมูลจานสี(Palette)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ข้อมูลภาพ(Data)

1. ข้อมูลเฮกเตอร์ คือ ข้อมูลที่อยู่บริเวณส่วนหัวของไฟล์ ซึ่งประกอบด้วยข้อมูลที่บอกรายละเอียดต่างๆของภาพ เช่น ความกว้าง ความยาวของภาพ จำนวนสี จำนวนบิต ความละเอียด เป็นต้น ซึ่งโครงสร้างของเฮกเตอร์จะแสดงในภาคผนวก

2. ข้อมูลงานสี คือ ข้อมูลที่บอกถึงชุดของงานสี ที่เกิดจากการผสมแม่สีทั้งสาม คือ แดง เขียว และน้ำเงิน มาผสมกันได้เป็นสีต่างๆ ตามจำนวนสีของภาพ เช่น รูปขนาด 4 บิต จะมี 16 ระดับสี รูปขนาด 8 บิต จะมีขนาด 256 ระดับสี เป็นต้น ซึ่งถ้ามีจำนวนสีน้อยๆ ก็จะมีการเก็บค่างานสีนี้ลงไปในไฟล์ด้วย แต่ถ้าเป็นรูปประเภท 24 บิตจะไม่มีค่างานสี แต่จะใช้วิธีการเก็บค่าแม่สีทั้งสามลงไปเป็นข้อมูลแทนเพราะถ้าเก็บค่างานสี ที่มีถึง 16.7 ล้านสี ลงไปด้วยนั้นจะเปลืองเนื้อที่มาก ข้อแตกต่างที่สำคัญของบิตแมปขนาดนี้คือ 1 พัลบิตแมป จะเก็บค่าของงานสี ชุดละ 4 ไบต์ แต่ถ้าใช้แค่ 3 ไบต์ เช่นกัน คือ แดง เขียว และน้ำเงิน อย่างละ 1 ไบต์

3. ข้อมูลภาพ คือ ข้อมูลสีของภาพแต่ละจุดที่มาประกอบกันเป็นรูปภาพ ซึ่งค่าที่เก็บนี้จะเป็นค่าที่ใช้ในการชี้ตาราง Palette หมายเลขอะไร เช่น จุดแรกมีค่าเป็น 10 ก็ให้ไปเปิดตาราง Palette หมายเลข 10 สมมติว่าในตารางหมายเลข 10 มีค่าของแม่สีดังนี้ $R = 0$ $G = 0$ และ $B = 100$ ก็จะได้จุดนี้เป็นสีน้ำเงิน ซึ่งถ้าเป็นในกรณีของรูป 24 บิต จะเป็นการอ่านข้อมูลขึ้นมา 3 ค่า เป็นค่าของแม่สี RGB แล้วนำไปผสมกันบนจอภาพแทน

2.2.3 การจัดเก็บไฟล์ข้อมูลชนิดบิตแมป

การเก็บไฟล์ข้อมูลภาพชนิดบิตแมป มีการเก็บอยู่ 2 แบบคือ

- แบบบีบอัดข้อมูล
 - RLE 4 เป็นการบีบอัดข้อมูลแบบ Run-Length Encoder แบบ 4 บิต
 - RLE 8 เป็นการบีบอัดข้อมูลแบบ Run-Length Encoder แบบ 8 บิต
- แบบไม่ได้บีบอัดข้อมูล

เป็นการเก็บข้อมูลจริงของสีของพิกเซล ซึ่งทำให้ขนาดของไฟล์ค่อนข้างใหญ่ แต่จะทำการแสดงภาพได้เร็วกว่า เพราะไม่ต้องเสียเวลาในการคลายข้อมูล

2.3 การสร้างภาพไปนารี

อุปกรณ์ที่มีความสามารถในการแสดงผลได้แค่ 2 ระดับ หรือ 2 สี คือสีขาวกับสีดำยังมีการใช้งานกันอยู่อย่างแพร่หลาย เช่น เครื่องพิมพ์ (Printer) เครื่องโทรสาร (Fax) จอภาพแสดงผลแบบโมโนโครม (Monochrome Monitor) เป็นต้น เนื่องจากอุปกรณ์เหล่านี้เป็นอุปกรณ์ ที่มีราคาถูก ดังนั้นการที่จะแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือพิมพ์รูปภาพที่มีระดับความเข้มของภาพหลายระดับซึ่งมีมากกว่าความสามารถในการแสดงผลของอุปกรณ์เหล่านั้นที่มีเพียงแค่ 2 ระดับ จำเป็นต้องมีกระบวนการหนึ่งเข้ามาช่วยในการแปลงข้อมูลภาพจากความเข้มหลายระดับแปลงสู่ความเข้ม 2 ระดับเพื่อการแสดงผล

จะเห็นได้ว่าการที่จะแก้ปัญหาการแสดงผลภาพที่มีความเข้มหลายระดับบนอุปกรณ์ที่สามารถแสดงผลได้ 2 ระดับนั้น จะต้องทำการแปลงข้อมูลภาพให้เป็นภาพไบนารี (Binary Image) ก่อนซึ่งการสร้างภาพไบนารี ก็หมายถึงการแปลงข้อมูลภาพที่มีระดับความเข้มหลายระดับ (Multi Level Image) ให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ นั่นคือ 1 จุดภาพสามารถมีได้แค่ 2 ค่าเท่านั้นคือ 1 หรือ 0 โดยจุดภาพที่แทนด้วย 1 จะหมายถึงจุดภาพที่มีสีดำ และจุดภาพที่แทนด้วย 0 คือจุดภาพที่มีสีขาว เมื่อทำการแปลงเป็นภาพไบนารีแล้วจึงนำภาพนั้นไปแสดงผลบนอุปกรณ์เหล่านั้น จะเห็นได้ว่าการแปลงข้อมูลภาพหลายระดับเป็นภาพไบนารีจึงมีความจำเป็นและมีประโยชน์มากในการแสดงผล ภาพที่มีระดับความเข้มของภาพหลายระดับบนอุปกรณ์ที่มีความสามารถในการแสดงผลได้ 2 ระดับ สำหรับประโยชน์อีกประการหนึ่งในการแปลงข้อมูลภาพนั้นเป็นข้อมูลภาพไบนารีคือการลดเนื้อที่การเก็บข้อมูลภาพจะใช้เนื้อที่ในการเก็บ 8 บิต เมื่อสร้างภาพเป็นภาพไบนารีแล้วสามารถลดลงได้ถึง 8 เท่า นั่นคือ 1 จุดภาพจะใช้เนื้อที่ในการเก็บเพียง 1 บิตเท่านั้น อีกทั้งยังสามารถนำไปประยุกต์ใช้งานได้อย่างแพร่หลาย เช่นนำไปประยุกต์ใช้ในการวิเคราะห์เอกสารในขั้นตอนที่เรียกว่าการประมวลผลขั้นต้น (PreProcessing) เป็นต้น

ในการสร้างภาพไบนารี สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล (Thresholding Technique) โดยพิจารณาว่าจุดภาพใดควรจะเป็นจุดขาวหรือจุดดำ จะกระทำโดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งซึ่งเรียกว่า “ค่าเทรชโฮล” (Threshold Value) เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุ (Object) และพื้นหลัง (BackGround) โดยค่าของจุดภาพใดๆ ที่มีค่าไยอกกว่าค่าเทรชโฮลจะถูกกำหนดให้เป็น 1 (จุดดำ) และถ้าค่าของจุดภาพใดๆ ที่มีค่ามากกว่า หรือเท่ากับค่าเทรชโฮลจะถูกเปลี่ยนให้เป็น 0 (จุดขาว)

ในการสร้างภาพไบนารีโดยใช้เทคนิคเทรชโฮลเพื่อให้ได้ผลลัพธ์ที่ได้เหมาะสมและคมชัด สิ่งสำคัญที่สุดคือ ค่าเทรชโฮล เนื่องจากถ้าเลือกค่าเทรชโฮลที่ไม่เหมาะสม(ค่าเทรชโฮลที่มีค่ามากหรือน้อยเกินไป) ภาพที่ได้อาจจะไม่เหมาะสม ขาดความคมชัดและรายละเอียดบางส่วนขาดหายไป กล่าวคือภาพที่ได้อาจจะมืดเกินไป (จุดดำมากเกินไป) หรือสว่างเกินไป (จุดขาวมากเกินไป) อันเป็นผลทำให้ภาพที่ได้ไม่สวยงามเท่าที่ควร ดังนั้นปัญหาของการสร้างภาพไบนารีโดยวิธีการเทรชโฮลนี้คือ ทำอย่างไรจึงจะสามารถคำนวณหาค่าเทรชโฮลที่เหมาะสมสำหรับภาพแต่ละภาพที่จะนำมาทำการสร้างภาพไบนารี

แต่ในรายงานฉบับนี้จะไม่กล่าวถึงถึงรายละเอียดถึงกระบวนการทำเทรชโฮล เนื่องมาจากกระบวนการของโครงการนี้ไม่มีการทำในส่วนนี้ เพราะภาพที่นำมาทำการประมวลผลอยู่ในรูปของภาพขาวดำ ความเข้ม 2 ระดับอยู่แล้ว จากที่ได้แสดกนภาพมาจากซองจดหมาย ไม่มีความจำเป็นที่จะต้องเก็บข้อมูลภาพมาในแบบความเข้มหลายระดับ ดังนั้นในโครงการนี้จึงไม่มีส่วนกระบวนการทำเทรชโฮล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การเตรียมข้อมูลภาพ(Pre-processing)

กระบวนการสำคัญอีกขั้นตอนหนึ่งในการประมวลผลภาพเบื้องต้นก่อนที่จะไปสู่ขั้นตอนการจดจำรูปแบบ ก็คือกระบวนการเตรียมข้อมูลภาพ โดยในขั้นแรกภาพไบนารีที่ถูกสร้างขึ้นมาจากการแสกนนั้น ย่อมต้องมีสัญญาณรบกวนเกิดขึ้นแน่นอน ดังนั้นในขั้นตอนแรกคือการทำกำจัดสัญญาณรบกวน(Noise Removing) เมื่อกำจัดสัญญาณรบกวนเรียบร้อยแล้วต่อมาเราจะเริ่มทำการแยกวัตถุที่ต้องการออกจากพื้นหลังด้วยวิธีการตามหาขอบภาพ (Contour Following) เมื่อทราบขอบเขตของวัตถุที่ต้องการแล้วเราก็จะทราบกรอบของวัตถุด้วยวิธีการตรวจสอบหามุมภาพ (Detect Edge) เมื่อเราทราบมุมภาพแล้วเราก็จะแยกวัตถุที่เราต้องการออกมาจากพื้นหลังได้อย่างสมบูรณ์ โดยวัตถุที่เราต้องการในโครงการนี้คือข้อมูลภาพตัวเลขที่เป็นรหัสไปรษณีย์ซึ่งจะบรรจุอยู่ในช่องสี่เหลี่ยม เมื่อเราแยกตัวเลขออกมาจากช่องสี่เหลี่ยมได้แล้วนั้นเราจะต้องทำกระบวนการทำเส้นบาง(Thinning) ซึ่งเป็นขั้นตอนที่สำคัญมากในการเตรียมอินพุทของระบบรู้จำ แต่ส่วนใหญ่ข้อมูลภาพที่ผ่านกระบวนการทำเส้นบางมาแล้วอาจจะเกิดเส้นที่เป็นส่วนเกินขึ้นมาได้ดังนั้นเราจึงจำเป็นต้องมีกระบวนการกำจัดส่วนเกินเข้ามาอีกเพื่อช่วยให้ประสิทธิภาพของกระบวนการเตรียมข้อมูลภาพเป็นไปอย่างถูกต้องและสมบูรณ์ที่สุด และนำเข้าสู่กระบวนการจดจำรูปแบบต่อไปซึ่งสามารถประมวลผลได้ที่ละหนึ่งตัวเลขเท่านั้น

2.4.1 การกำจัดสัญญาณรบกวน

การกำจัดสัญญาณรบกวนอาศัยหลักการการแสกนหาจุดดำจุดเดี่ยวที่อยู่บนรูปภาพซึ่งเราตัดสินใจว่ามันคือสัญญาณรบกวน แต่ในบางกรณีผู้ใช้หลักการนี้อาจจะกำหนดค่าค่าหนึ่งขึ้นมาก็ได้ว่าถ้ามีจำนวนจุดติดกันได้เท่าไรให้ยังถือว่าจุดเหล่านั้นเป็นสัญญาณรบกวนอยู่โดยไม่จำเป็นต้องเป็นจุดเดี่ยวเสมอไป อาจจะเป็นจุดคู่ หรือจุดสามจุดติดกันก็ได้แล้วแต่จะกำหนดไป โดยจุดเดี่ยวบนรูปภาพหมายถึงจุดที่ไม่มีจุดดำจุดอื่นอยู่ติดอยู่เลยในรอบด้านแปดทิศของตัวมัน ตัวอย่างจุดเดี่ยวดังแสดงต่อไปนี้

x1	x2	x3
x8	x0	x4
x7	x6	x5

รูปที่ 2 แสดงตัวอย่างสัญญาณรบกวนที่เป็นจุดเดี่ยว

พิกเซลที่เราตัดสินใจว่าเป็นสัญญาณรบกวนในระบบนั้นๆจะถูกกำจัดออกไปเพื่อให้ข้อมูลภาพมีความสมบูรณ์มากขึ้น โดยอาศัยหลักการตรวจสอบจุดที่ล้อมรอบจุดที่เรากำลังทำการวิเคราะห์อยู่ว่ามีจุดสีดำติดกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่หรือไม่ ถ้ามีจุดค่าปรากฏอยู่โดยรอบเราจะตัดสินใจว่าจุดนั้นไม่ใช่สัญญาณรบกวน และจะไม่ทำการลบมันออก แต่ถ้าหากว่าจุดที่เราทำการวิเคราะห์อยู่นั้นไม่มีจุดค่าล้อมรอบอยู่เลยไม่แต่จุดเดียวเราจะตัดสินใจว่าจุดนั้นเป็นสัญญาณรบกวน เราจะทำการลบจุดนั้นออกไปเพื่อเป็นการกำจัดสัญญาณรบกวนในข้อมูลภาพนั่นเอง รูปต่อไปนี้จะแสดงถึงข้อมูลภาพขาวดำที่มีสัญญาณรบกวนเกิดขึ้น



รูปที่ 3

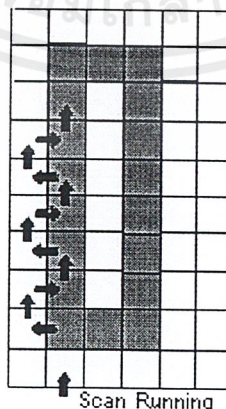
แสดงภาพที่มีสัญญาณรบกวนต่ำ

รูปที่ 4

แสดงภาพที่มีสัญญาณรบกวนสูง

2.4.2 วิธีการ วนตามหากรอบภาพ (Contour Following)

วิธีการวนตามหากรอบภาพนี้ มีหลักการพื้นฐานคือเมื่อเริ่มการแสกนเราจะแสกนมาเรื่อยๆ ถ้าหากเจอจุดขาว (แทนด้วยค่า 0) จะให้ตัวแปรที่เป็นตัวชี้จุดทำการเดินขวา และถ้าจุดต่อไปเป็นจุดสีขาวอีกก็จะทำการเดินขวาอีกไปเรื่อยๆ จนกว่าจะพบจุดสีดำ (แทนด้วยค่า 1) ตัวชี้จุดจะทำการเดินซ้าย โดยทิศทางเดินข้างอิงจากทิศที่เดินทางมาถึงจุดสีดำก่อนหน้าหนึ่งจุด แล้วทำการวนตามหาขอบภาพไปเรื่อยๆ ดังภาพตัวอย่างต่อไปนี้ จะทำให้สามารถเข้าใจได้ง่ายมากขึ้น สังเกตได้ว่าเมื่อทำการพบจุดดำที่เป็นกรอบแล้วตัวชี้จะทำการวนหากรอบไปเรื่อยๆ จนมาบรรจบพบกันเป็นกรอบซึ่งเป็นส่วนรอบนอกสุดของรูปภาพนั่นเอง



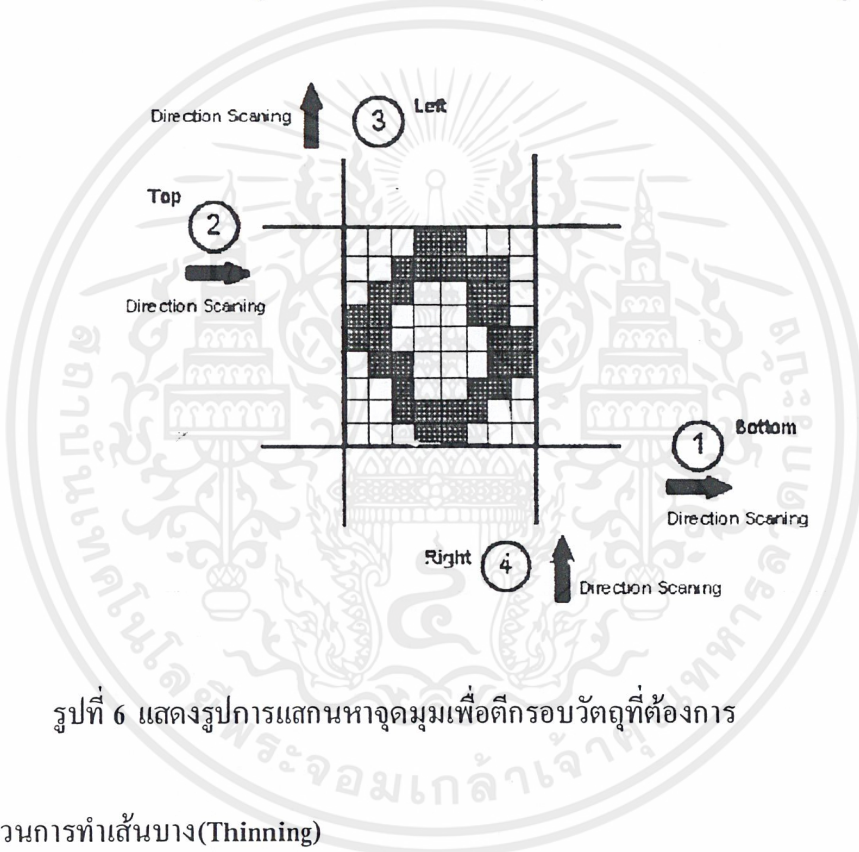
รูปที่ 5 แสดงการทำ Contour Following

ในภาพเป็นการหากรอบของรูปสี่เหลี่ยมผืนผ้าที่มีช่องว่างตรงกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 วิธีการตรวจสอบหาขอบภาพ (Detect Edge)

วิธีการตรวจสอบหาขอบภาพโดยหลักการก็คือการแสกนหาจุดดำ(แทนด้วยค่า 1) จากขอบล่างขึ้นมา ก่อนซึ่งข้อมูลด้านล่างจุดนี้จะเป็นจุดขาวทั้งหมด ต่อมาหลังจากเก็บค่าพิกัดของพิกเซลไว้แล้วเราก็จะทำการแสกนหาจุดบนสุดของข้อมูลภาพต่อไปโดยจุดสีด้านบนของจุดนี้จะเป็นสีขาวทั้งหมด แล้วก็ทำการแสกนหาจุดขอบด้านซ้ายและขวาต่อไป จนได้จุดมาทั้งสี่จุดแล้วทำการตีกรอบโดยใช้พิกัดพิกเซลที่ได้ทั้งสี่จุดนั้นเป็นหลักในการตีกรอบข้อมูลนั่นเอง เทำนี้เราก็จะได้ ข้อมูลภาพที่ต้องการแยกออกมาจากพื้นหลังโดยมีกรอบชิดพอดีกับวัตถุที่เราต้องการทำการวิเคราะห์ รูปหลักการตรวจสอบหาขอบภาพเบื้องต้นดังแสดงในรูปข้างล่าง



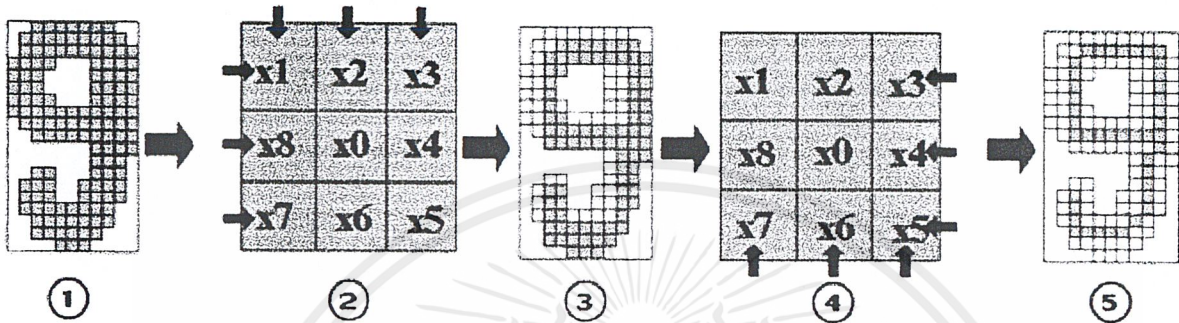
รูปที่ 6 แสดงรูปการแสกนหาจุดมุมเพื่อตีกรอบวัตถุที่ต้องการ

2.4.4 กระบวนการทำเส้นบาง(Thinning)

กระบวนการทำเส้นบางมีความสำคัญมากในขั้นตอนการเตรียมข้อมูลภาพก่อนเข้าสู่ระบบรู้จำ กระบวนการทำเส้นบาง จุดประสงค์เพื่อลดขนาดความหนาของเส้นรูปภาพให้เล็กลงจากปกติที่ขนาดใหญ่กว่า 1 พิกเซลให้ลงมาเหลือเท่ากับ 1 พิกเซลให้หมด ทำไมจึงต้องเป็น 1 พิกเซล? เพราะว่ามันเป็นเงื่อนไขที่จะใช้ได้ผลกับการตรวจสอบหาจุดแยก จุดปลาย จุดตัด และรูปแบบต่างๆในการวิเคราะห์โครงสร้างในขั้นสุดท้ายต่อไป ถ้าไม่ทำการทำเส้นบางจะทำให้ไม่สามารถวิเคราะห์ได้เลย โดยมีหลักการดังนี้ จากบล็อกพิกเซลมาตรฐาน 9 จุดก่อนแล้วเราจะแสดงให้เห็นว่าการลดขนาดความหนาของพิกเซลทำกันอย่างไร ในขั้นตอนแรกเราจะทำการแสกนหาจุดหลัก x_0 ก่อนเมื่อเจอแล้วจะทำการตรวจสอบรอบด้านทั้งแปดทิศว่ามีจุดข้างเคียงอยู่หรือไม่ เพื่อความมีประสิทธิภาพของโครงสร้างกระบวนการทำเส้นบางจึงต้องทำสองครั้งในขั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนแรกตามรูปที่ 7 ข้อ 2 เราจะทำการบีบพิกเซลทางด้านซ้ายและด้านบนของจุดหลักเข้ามาก่อน ส่วนในขั้นตอนต่อไปในรูปที่ 7 ข้อ 4 เราค่อยทำการบีบพิกเซลทางด้านขวาและด้านล่างของจุดหลักเข้ามา ดังกล่าวจะมีความสมมูลย์ในโครงสร้างถ้าหากเราใช้วิธีนี้ โดยสมการตรรกที่สำคัยในการพิจารณาการบีบลดพิกเซลในแต่ละด้านแต่ละครั้งดังที่ได้แสดงไว้



รูปที่ 7 แสดงขั้นตอนการทำเส้นบางอย่างมีเสถียรภาพ

อธิบายจากรูปที่ 7 ข้อ 1 คือรูปภาพที่เราทำการแสดกนมาแล้วมีขนาดความหนาของเส้นมากกว่าหนึ่งพิกเซล ข้อ 2 กระบวนทำเส้นบางโดยบีบทางด้านบนและด้านซ้ายก่อน โดยในขั้นตอนนี้มีสมการตรรกดังนี้

$$(x2||x4||x8 = \text{white}) \&\& (x2||x6||x8 = \text{white}) = \text{true} ? \quad \text{สมการที่ 2.3}$$

ข้อ 3 แสดงรูปภาพหลังจากทำการบีบในด้านบนและด้านซ้ายแล้ว ข้อ 4 กระบวนการทำเส้นบางโดยบีบทางด้านล่างและด้านขวาเข้ามาโดยขั้นตอนนี้มีสมการตรรกดังนี้

$$(x2||x4||x6 = \text{white}) \&\& (x4||x6||x8 = \text{white}) = \text{true} ? \quad \text{สมการที่ 2.4}$$

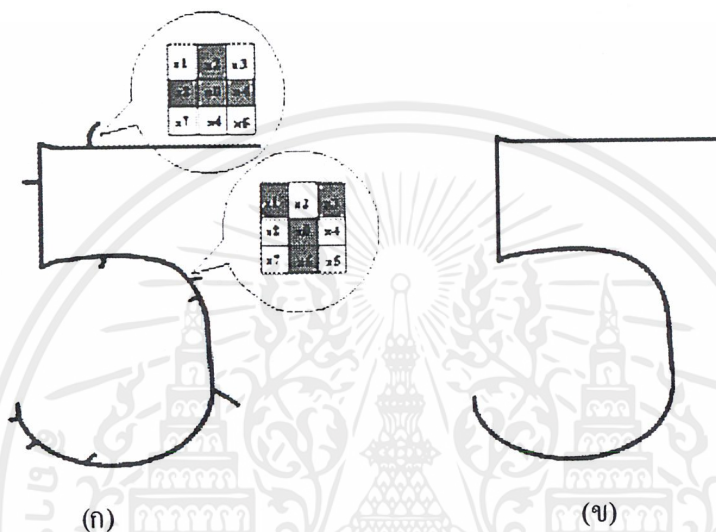
ข้อ 5 รูปภาพที่ผ่านการทำเส้นบางแล้ว อย่างไรก็ตามก็ตีกระบวนการทำเส้นบางไว้ว่าจะไม่เกิดปัญหาเลยในหัวข้อต่อไปคือกระบวนการกำจัดส่วนเกิน ซึ่งเป็นหัวข้อในการกำจัดส่วนเกินซึ่งส่วนเกินเป็นผลที่ได้จากการทำเส้นบางซึ่งเราไม่ต้องการนั่นเอง

2.4.5 กระบวนการกำจัดส่วนเกิน

ส่วนเกินเป็นผลที่ไม่ต้องการหลังจากกระบวนการทำเส้นบางดังนั้นเราจำเป็นต้องกำจัดออก เพราะเราถือว่าส่วนเกินเป็นสัญญาณรบกวนอย่างหนึ่งที่เราไม่ต้องการเช่นกัน ในระบบใดๆจะมีการกำหนดค่าสูงสุดของสัญญาณรบกวนไม่เท่ากัน ส่วนเกินก็เช่นเดียวกัน จะได้รับการกำหนดและตัดสินใจพิจารณาว่าส่วนใดเป็นส่วนเกิน ส่วนใดเป็นองค์ประกอบภาพ แล้วทำการกำจัดทิ้งเช่นเดียวกับสัญญาณรบกวน โดยวิธีการกำจัดส่วนเกิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการของกระบวนการกำจัดส่วนเกินไม่ซับซ้อนนัก โดยหลักการกล่าวคือ จะทำการแสกนหาจุดแยกในองค์ประกอบของภาพ แล้วทำการตรวจเช็คว่าส่วนเกินที่แยกออกมานั้นมีความยาวเท่าใด ถ้ามีความยาวต่ำกว่าค่าสูงสุดที่เรากำหนดไว้ให้เป็นสัญญาณครบถ้วนแล้ว เราจะทำการลบส่วนเกินนั้นออกไปทันที และทำการแสกนหาจุดแยกในองค์ประกอบภาพจนหมดและทำการวนการนี้ไปเรื่อยๆจนเป็นสิ้นสุดการกำจัดส่วนเกิน รูปแสดงภาพจุดแยกและภาพที่มีส่วนเกินเกิดขึ้นหลังจากผ่านกระบวนการทำเส้นบางดังแสดงในรูปต่อไป



- รูปที่ 8 แสดงภาพที่เกิดส่วนเกินหลังจากผ่านการทำเส้นบางมาแล้ว
- (ก) ภาพก่อนทำการกำจัดส่วนเกิน มีจุดแยกที่เป็นส่วนเกินอยู่หลายจุด
- (ข) ภาพหลังจากผ่านขั้นตอนการกำจัดส่วนเกินเรียบร้อยแล้ว

หลังจากผ่านกระบวนการกำจัดส่วนเกินแล้วข้อมูลก็พร้อมที่จะเข้าสู่ระบบรู้จำต่อไปโดยในโครงการนี้เราเลือกใช้วิธีวิเคราะห์โครงสร้าง ดังจะได้อีกต่อไป

2.5 ระบบรู้จำโดยใช้การวิเคราะห์โครงสร้าง

หัวข้อนี้จะได้อีกถึงระบบจดจำรูปแบบที่เป็นส่วนสำคัญมากที่สุดในโครงการ เพราะเป็นส่วนที่จะทำการวิเคราะห์ พิจารณาและตัดสินใจคำตอบ ซึ่งคำตอบที่ได้จะมีผลกับความถูกต้องของการทดลอง ซึ่งมีความจำเป็นที่จะต้องทำให้ครอบคลุม รัดกุม และถูกต้องมากที่สุด

ในปัจจุบันเทคโนโลยีของกระบวนการจดจำรูปแบบมีหลายแบบด้วยกัน ที่เห็นจะได้ยินกันอยู่เป็นประจำก็คือกระบวนการจดจำรูปแบบโดยวิธีใช้โครงข่ายประสาทเทียม(Neural Network) หรือกระบวนการจดจำรูปแบบโดยใช้กระบวนการ Fuzzy Logic เป็นต้น แต่ในโครงการนี้เราได้ใช้วิธีการจดจำรูปแบบซึ่งใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวิเคราะห์โครงสร้างมาใช้งาน เพราะมีการทำงานที่ไม่สลับซับซ้อน เข้าใจง่ายและตรงไปตรงมา การทำงานอาศัยหลักการพิจารณาโครงสร้างของรูปภาพแล้วทำการตัดสินใจเป็นกรณีไปเมื่อผ่านแต่ละจุดการพิจารณาไปถึงคำตอบสุดท้ายให้ถือว่านั่นคือคำตอบ โดยการทำงานจะเป็นไปตามตารางแผนผังการพิจารณาที่เราได้ทำการเก็บข้อมูลและรวบรวมความน่าจะเป็นของโครงสร้างไว้ก่อนหน้าแล้วด้วยการสุ่มตัวอย่าง

ก่อนอื่นเราต้องทำความรู้จักกับรูปแบบของจุดแยก จุดปลาย และจุดตัด ในรูปแบบของบล็อกมาตรฐานเสียก่อน เพราะเป็นพื้นฐานที่ต้องเข้าใจและนำไปใช้ในการสร้างโปรแกรมเพื่อทำการวิเคราะห์ต่อไป โดยรูปดังได้แสดงต่อไป

2.5.1 จุดแยกมีลักษณะดังยกมาเป็นตัวอย่างต่อไปนี้

รูปที่ 9 แสดงตัวอย่างจุดแยกแบบต่างๆ

2.5.2 จุดปลายมีลักษณะดังต่อไปนี้

รูปที่ 10 แสดงลักษณะของจุดปลายแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 ลักษณะของจุดตัดดังแสดงต่อไปนี้

x1	x2	x3
x8	x0	x4
x7	x6	x5

x1	x2	x3
x8	x0	x4
x7	x6	x5

รูปที่ 11 แสดงลักษณะของจุดตัดแบบต่างๆ

เมื่อเรารู้จักถึงโครงสร้างลักษณะพื้นฐานของรูปแบบจุดแบบต่างๆแล้วต่อไปเราต้องทำความเข้าใจกับองค์ประกอบในการพิจารณาหาคำตอบของระบบรู้อำแบบวิเคราะห์โครงสร้างด้วยเพื่อจะได้นำความเข้าใจในส่วนนี้ไปทำการเขียนโปรแกรมแล้วทำการวิเคราะห์ต่อไป โดยองค์ประกอบหลักๆที่เราใช้ในการพิจารณาวิเคราะห์โครงสร้างมีดังต่อไปนี้

2.5.4 จำนวนจุดปลาย

- ไม่มีจุดปลาย เช่น เลข 0
- มี 1 จุดปลาย เช่น เลข 9
- มี 2 จุดปลาย 7
- มี 3 จุดปลาย 3
- มี 4 จุดปลาย 4

ดังแสดงในรูปตัวอย่างต่อไปนี้

0 9 7 3 4

รูปที่ 12 แสดงตัวอย่างการนับจุดปลายของตัวเลขต่างๆ

2.5.5 จำนวนจุดแยกหรือจุดตัด

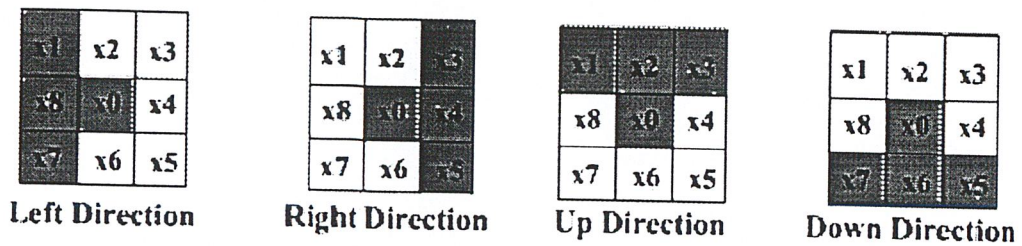
4 9

1 จุดตัด 1 จุดแยก

รูปที่ 13 แสดงตัวอย่างการนับจุดแยกและจุดตัดของตัวเลขต่างๆ

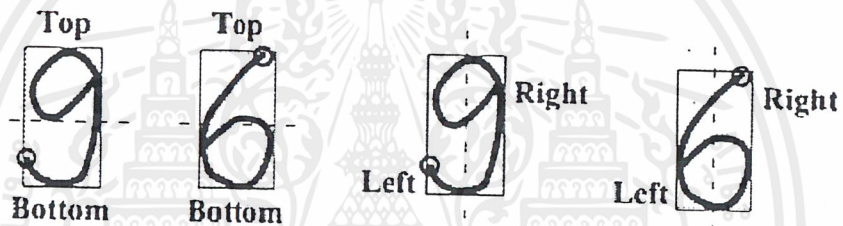
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.6 ทิศทางการเดินทางของเส้นที่ต่อจากจุดปลาย



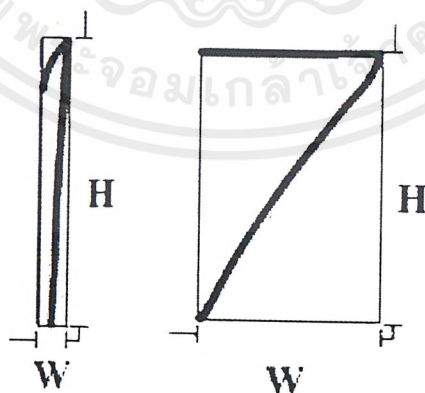
รูปที่ 14 แสดงทิศทางการเดินทางของเส้นที่ต่อไปจากจุดปลาย(x0)

2.5.7 ตำแหน่งของจุดปลาย



รูปที่ 15 แสดงตัวอย่างตำแหน่งของจุดปลายในควอดเรนต์ทั้งสิ้น

2.5.8 อัตราส่วนระหว่างความสูงกับความกว้างของภาพ



รูปที่ 16 แสดงตัวอย่างเปรียบเทียบอัตราความสูงต่อความกว้างของเลข 1 กับ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากองค์ประกอบที่แสดงให้ดูนั้นมีความสำคัญมากในกระบวนการรู้จำแบบวิเคราะห์โครงสร้าง เราจำเป็นต้องหาลักษณะพิเศษของแต่ละตัว เก็บและรวบรวมข้อมูลเพื่อมาทำเป็นโปรแกรมใช้วิเคราะห์ โดยจะเห็นได้ว่าการวิเคราะห์โครงสร้างนี้จะคุณสมบัติทางโทโพโลยีมาใช้แบ่งความแตกต่างของตัวเลข อาศัยการคำนวณหาความสัมพันธ์ระหว่างจุดต่างๆ บนโครงร่างของตัวเลข โดยจะมีองค์ประกอบที่ใช้ในการพิจารณาวิเคราะห์ดังที่ได้กล่าวไปแล้ว ในบทต่อไปเราจะกล่าวถึงหลักการสร้างและโครงสร้างโปรแกรมกันต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

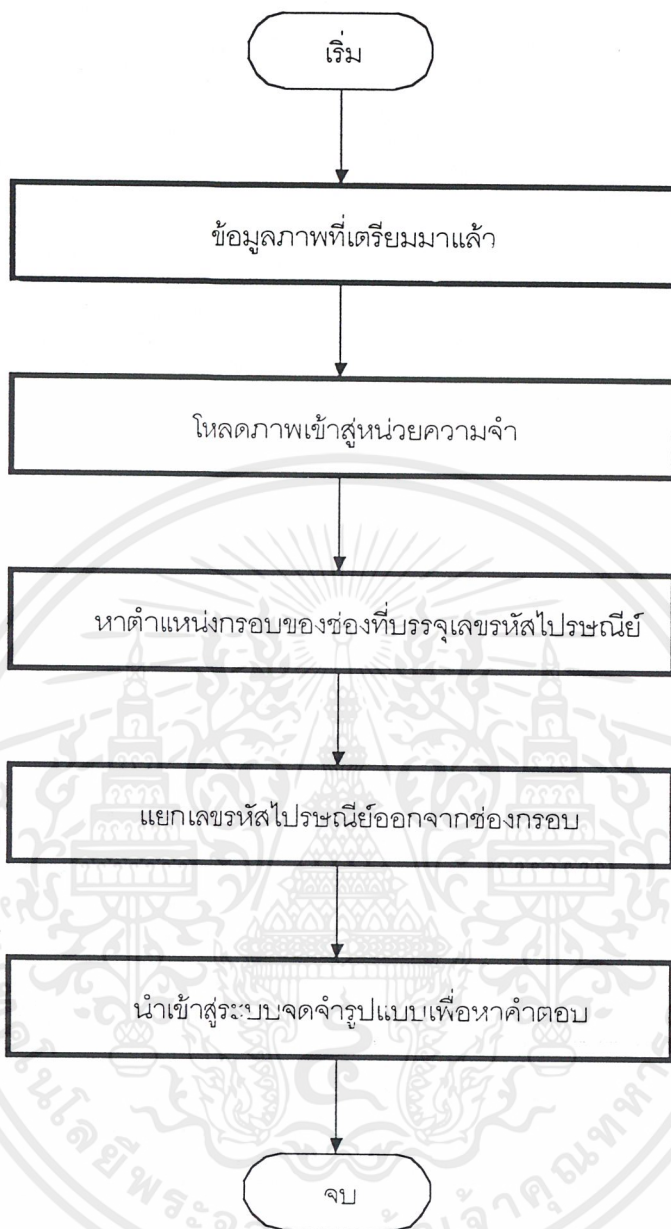
โครงสร้างระบบและโครงสร้างโปรแกรม

3.1 โครงสร้างของระบบรู้จำรหัสไปรษณีย์

ระบบรู้จำรหัสไปรษณีย์ ประกอบด้วยส่วนหลักสำคัญ 2 ส่วนคือ ส่วนการประมวลผลภาพ และ ส่วนการรู้จำรหัสไปรษณีย์ โดยในส่วนที่ทำการรับภาพเราจะทำการจำลองด้วยการแสดงภาพของจดหมายที่ได้ทำการเขียนชื่อที่อยู่และเลขรหัสไปรษณีย์มาจากเครื่องแสกนเนอร์ซึ่งภาพที่ได้อยู่ในระดับความเข้มเท่ากับ 2 คือมีแค่สีขาวกับสีดำเท่านั้น แล้วนำมาทำการแยกข้อมูลภาพเอาเลขรหัสไปรษณีย์ออกจากภาพโดยรวมของจดหมาย เพื่อเตรียมเป็นอินพุทให้แก่ระบบส่วนการรู้จำรหัสไปรษณีย์ต่อไป โดยในการทำงานของระบบในส่วนการประมวลผลภาพจะอาศัยการโปรแกรมคอมพิวเตอร์เป็นหลัก

ในส่วนของโปรแกรมการประมวลผลภาพนั้นมีขั้นตอนในการแยกหมายเลขรหัสไปรษณีย์ออกจากข้อมูลภาพโดยรวมดังแสดงในรูปที่ 17

โดยมีขั้นตอนของการเตรียมภาพโดยทำการแสกนมาจากเครื่องแสกนเนอร์โดยภาพเกรย์สเกล 2 ระดับ ซึ่งแปลงเป็นข้อมูลภาพแบบไบนารี เรียบร้อยแล้ว ส่วนกระบวนการย่อยคือการกำจัดสัญญาณรบกวนที่อาจเกิดขึ้นจากการแสกน แล้วจึงแยกกรอบช่องที่บรรจุเลขรหัสไปรษณีย์ออกมาก่อนเป็นอันดับแรก แยกออกจากภาพของจดหมายโดยรวม ในขั้นต่อมาเราจะทำการแยกเลขรหัสไปรษณีย์ออกจากกรอบที่แยกออกมาได้ในขั้นตอนก่อนหน้านี้อีกที ก็จะได้เลขรหัสไปรษณีย์ออกมาตามต้องการ โดยเลขรหัสไปรษณีย์ที่ได้จะอยู่ในลักษณะบล็อกเดี่ยว โดยแต่ละบล็อกเดี่ยวของอินพุทไม่จำเป็นต้องมีขนาดเท่ากัน เพราะเราใช้วิธีการจดจำแบบวิเคราะห์โครงสร้าง เราจะดูลักษณะพิเศษของข้อมูลในบล็อกเท่านั้น ข้อมูลที่ได้เป็นบล็อกเดี่ยวจะนำไปเข้าสู่กระบวนการรู้จำเพื่อหาคำตอบต่อไป



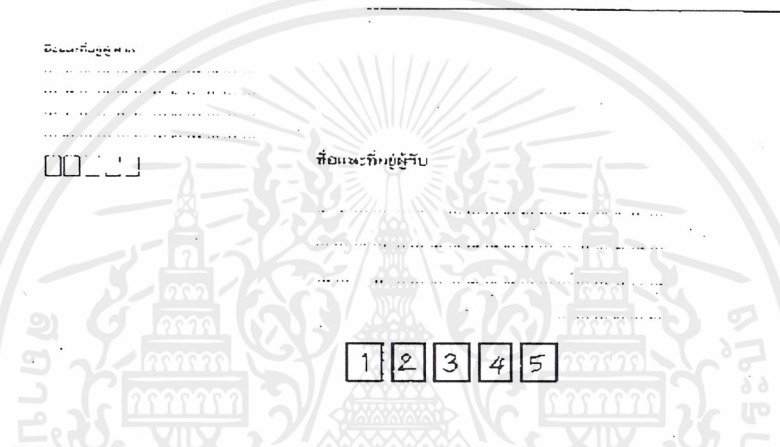
รูปที่ 17 โครงสร้างขั้นตอนการทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ระบบส่วนการประมวลผลภาพ

3.2.1 การเตรียมข้อมูลภาพ

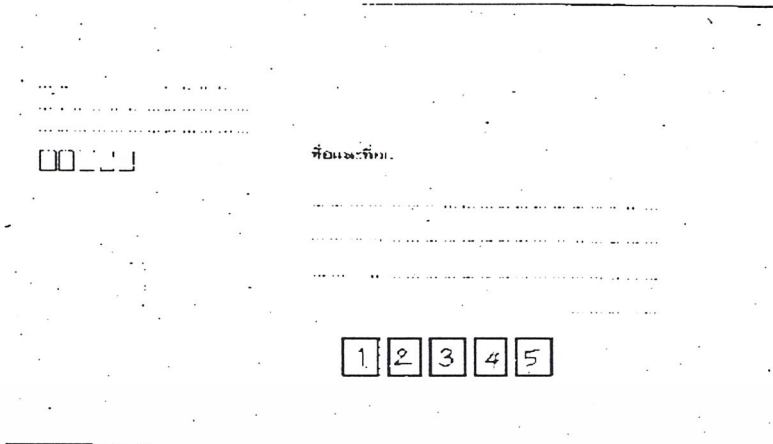
การเตรียมข้อมูลภาพในโครงการนี้ ทำโดยนำเอาซองจดหมายจริงมาทำการเขียนชื่อที่อยู่ รหัสไปรษณีย์ลงไป แล้วนำไปจำลองโดยการแสกนภาพจากเครื่องแสกนเนอร์ โดยให้ข้อมูลออกมามีความเข้มอยู่สองระดับคือสีดำกับสีขาวเท่านั้น โดยแสกนที่ระดับความละเอียด 200 dpi และภาพจะมีขนาดความกว้างยาวประมาณ 1200 x 900 pixels ดังนั้นข้อมูลภาพจะเป็นข้อมูลไบนารีอยู่แล้วพร้อมที่จะทำการประมวลผลภาพต่อไปได้ ตัวอย่างภาพที่ทำการแสกนมาดังแสดงในรูปที่ 18



รูปที่ 18 ข้อมูลภาพที่ได้แบบไบนารีที่ทำการเตรียมมา

3.2.2 การกำจัดสิ่งรบกวน (Noise Removing)

ในขั้นตอนของการแสกนภาพเป็นข้อมูลไบนารีมานั้น อาจทำให้เกิดพิกเซลสีดำเดี่ยวๆ ขึ้นในภาพบ้าง ดังที่แสดงในรูปที่ โดยสีดำเดี่ยวๆ นี้ ถ้ามามากเกินไปอาจจะมีผลกระทบต่อการประมวลผลภาพในขั้นต่อไปได้ ดังนั้นเราจึงต้องกำจัดพิกเซลที่เป็นสิ่งรบกวนพวกนี้ออกจากภาพ โดยการหาตำแหน่งของพิกเซลสีดำที่ไม่มีพิกเซลข้างเคียงเลย แล้วเปลี่ยนพิกเซลเหล่านั้นให้เป็นสีขาวแทน ตัวอย่างผลของการกำจัดพิกเซลดังแสดงในรูปที่ 19



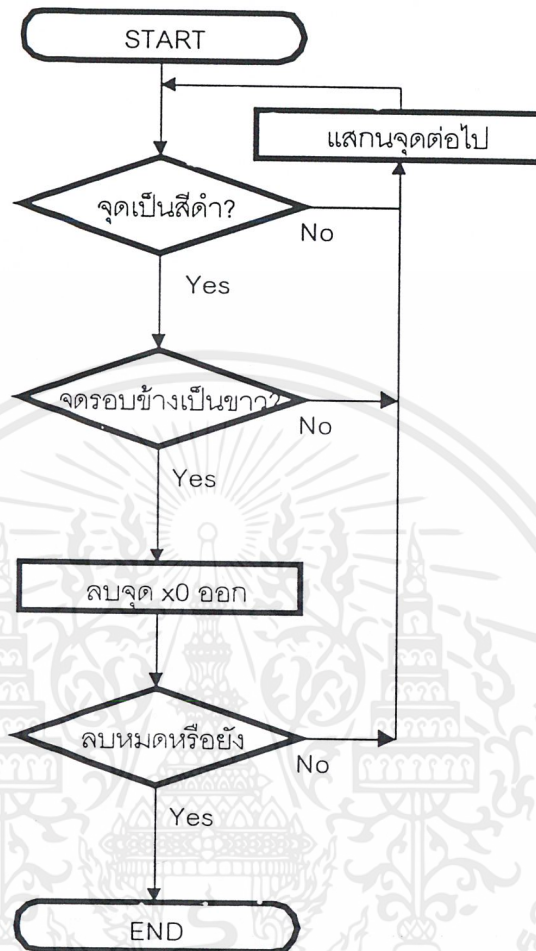
รูปที่ 19 ตัวอย่างการแสกนภาพที่มีสัญญาณรบกวน



รูปที่ 20 ตัวอย่างภาพที่ผ่านการกำจัดสัญญาณรบกวนแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1 โครงสร้างโปรแกรมการกำจัดสัญญาณรบกวน



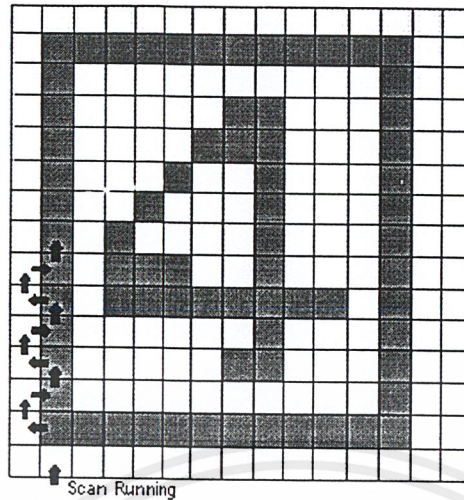
รูปที่ 21 โครงสร้างโปรแกรมกำจัดสัญญาณรบกวน

3.2.3 การแยกภาพด้วยวิธีวนตามหารอบภาพ (Contour Following)

เมื่อรับข้อมูลภาพที่ได้จากการแสดกนมาแล้วเป็นข้อมูลแบบไบนารีที่มีค่า 0 กับ 1 เรียบร้อยแล้วและทำการกำจัดสัญญาณรบกวนแล้ว ซึ่งข้อมูล 0 จะแทนส่วนที่เป็นพื้นหลัง และ 1 แทนส่วนที่เป็นตัวอักษรหรือกรอบของภาพที่บรรจุด้วยตัวเลขอยู่ภายใน โดยในโครงการนี้กรอบคือช่องที่เราใช้เขียนรหัสไปรษณีย์ใส่เข้าไปนั่นเอง โดยเราจะหากรอบของเลขรหัสไปรษณีย์ก่อนแล้วค่อยทำการแยกตัวเลขข้างในออกมามีอีกทีหนึ่ง หลักการของวิธีวนตามหารอบภาพคือ การวนหาข้อมูลภาพที่เป็นจุดดำวนครบรอบมาพบบรรจบกัน ในพื้นที่ของภาพที่คาดว่าจะเป็นที่ตั้งข้อมูลที่เป็นส่วนกรอบบรรจุเลขรหัสไปรษณีย์ที่เราต้องการ

ตัวอย่างการแยกข้อมูลภาพที่เป็นกรอบของตัวเลขอยู่ภายในกรอบ โดยวิธีนี้ไม่จำเพาะเจาะจงว่าจะเป็นกรอบสี่เหลี่ยม หรือวงกลม ใ้เป็นกรอบหรือขอบของภาพก็สามารถหาได้หมด โดยในตัวอย่างจะเป็นกรอบสี่เหลี่ยมซึ่งใช้กันมากซึ่งเป็นลักษณะมาตรฐานของกรอบเลขรหัสไปรษณีย์บนซองมาตรฐานปัจจุบัน

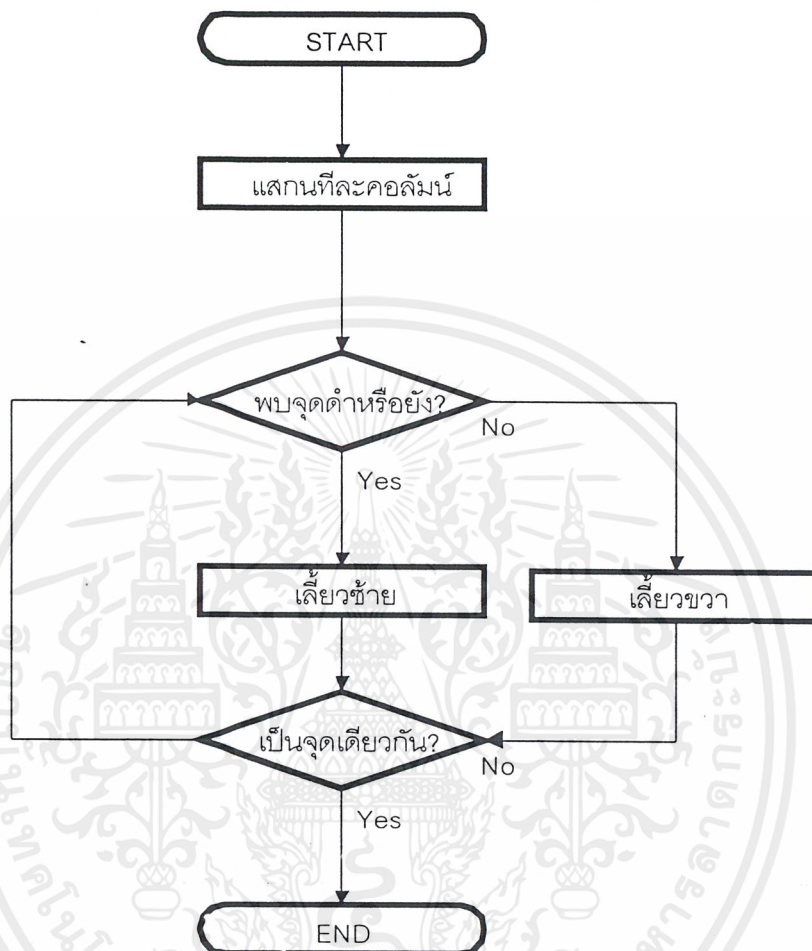
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 22 ภาพแสดงหลักการทำงานของกระบวนการ Contour Following

โดยหลักการมีดังนี้ คือเมื่อเริ่มการแสกนเราจะแสกนมาเรื่อยๆ ถ้าหากเจอจุดขาว (แทนด้วยค่า 0) จะให้ตัวแปรที่เป็นตัวชี้จุดทำการเลี้ยวขวา และถ้าจุดต่อไปเป็นจุดสีขาวอีกก็จะทำการเลี้ยวขวาอีกไปเรื่อยๆ จนกว่าจะพบจุดสีดำ (แทนด้วยค่า 1) ตัวชี้จุดจะทำการเลี้ยวซ้ายโดยทิศการเลี้ยวอ้างอิงจากทิศที่เดินทางมาถึงจุดสีดำก่อนหน้านั้นหนึ่งจุด แล้วทำการวนตามหาขอบภาพไปเรื่อยๆ ดังภาพตัวอย่างต่อไปนี้ จะทำให้สามารถเข้าใจได้ง่ายมากขึ้น สังเกตได้ว่าเมื่อทำการพบจุดดำที่เป็นกรอบแล้วตัวชี้จะทำการวนหากรอบไปเรื่อยๆ จนมาบรรจบพบกันเป็นกรอบซึ่งบรรจุด้วยตัวเลขรหัสที่อยู่ในนั้นมา เมื่อเสร็จขั้นตอนนี้แล้วเราก็จะทำการแยกตัวเลขรหัสออกมาจากกรอบกันต่อไปด้วยวิธีการตรวจสอบมุมภาพ (Detect Edge)

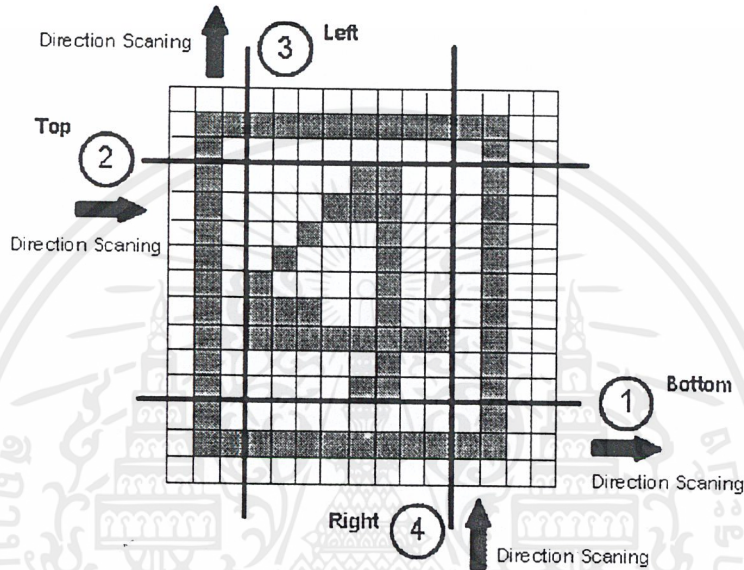
3.2.3.1 โครงสร้างโปรแกรมการวนตามหารอบภาพ



รูปที่ 23 โครงสร้างโปรแกรมการวนตามหารอบภาพ

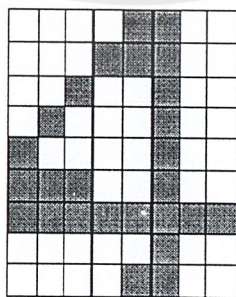
3.2.4 การแยกข้อมูลภาพที่อยู่ภายในกรอบหรือขอบภาพด้วยวิธีการตรวจสอบมุมภาพ(Detect Edge)

วิธีการนี้เป็นการแยกข้อมูลภายในกรอบหรือขอบภาพที่เราทำการแยกออกมาจากพื้นหลังได้แล้วโดยอาศัยค่ามาตรฐานที่ยอมรับได้ค่าหนึ่งในการเลื่อนกรอบให้แคบลงมาอีก ซึ่งจะทำให้เส้นกรอบที่ทำการหาด้วยวิธี Contour Following หายไปได้ แล้วหลังจากนั้นเราก็จะเริ่มทำการตรวจสอบหามุมของข้อมูลภาพที่เราต้องการจะหาถัดไป



รูปที่ 24 ภาพแสดงหลักการทำงานของกระบวนการ Detect Edge

โดยหลักการก็คือการแสกนหาจุดดำ(แทนด้วยค่า 1) จากขอบล่างขึ้นมาก่อนซึ่งข้อมูลด้านข้างจุดนี้จะเป็นจุดขาวทั้งหมด ต่อมาหลังจากเก็บค่าพิกัดของพิกเซลไว้แล้วเราก็จะทำการแสกนหาจุดบนสุดของข้อมูลภาพต่อไปโดยจุดสีด้านบนของจุดนี้จะเป็นสีขาวทั้งหมด แล้วก็ทำการแสกนหาจุดขอบด้านซ้ายและขวาต่อไป จนได้จุดมาทั้งสี่จุดแล้วทำการตีกรอบโดยใช้พิกัดพิกเซลที่ได้ทั้งสี่จุดนั้นเป็นหลักในการตีกรอบข้อมูลนั่นเอง เท่านั้นเราก็จะได้ ข้อมูลภาพที่ต้องการแยกออกมาจากพื้นหลังและแยกออกมาจากรอบเรียบร้อยแล้ว

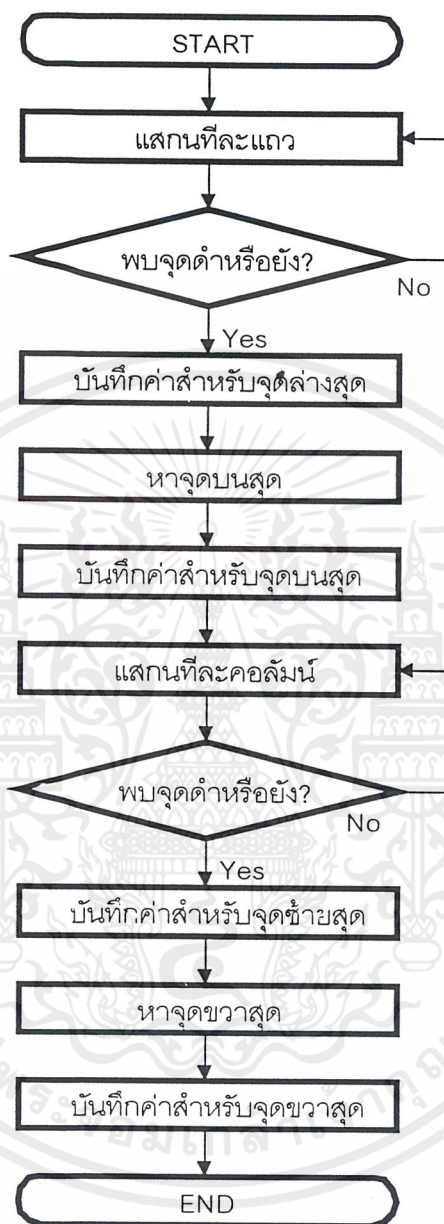


รูปที่ 25 ข้อมูลบล็อกเดี่ยวของภาพตัวเลขที่แยกออกมาได้

และเตรียมทำกระบวนการทำเส้นบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4.1 โครงสร้างโปรแกรมการตรวจสอบคุณภาพ



รูปที่ 26 โครงสร้างโปรแกรมการตรวจสอบหาคุณภาพ

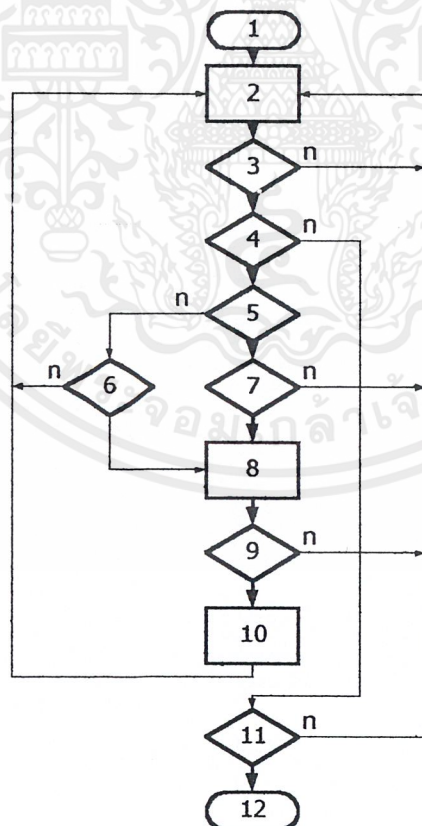
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 กระบวนการทำเส้นบาง(Thinning)

กระบวนการทำเส้นบางหรือที่เรียกกันว่า thinning นั้นเป็นขั้นตอนหนึ่งที่สำคัญมากในการทำการเตรียมข้อมูลภาพก่อนเข้าระบบการรู้จำ โดยหลักการของการทำเส้นบางคือจะหาจุดสีดำของข้อมูลภาพที่ไม่ใช่จุดสีดำเดี่ยวๆแต่เป็นจุดสีดำที่เกาะกลุ่มกัน โดยจะเริ่มการพิจารณาที่จุดสีดำเริ่มต้นตรงกลาง โดยจะพิจารณาหาจุดดำรอบข้างของจุดหลักออกไปทั้งแปดทิศทางแปดทิศรอบด้านมีจุดดำอยู่หรือไม่ ถ้ามีจุดดำอยู่รอบๆ จะทำการกำจัดตำแหน่งจุดหลักตรงกลางนั้น ไว้และจะทำการลบในภายหลังเมื่อทำการพิจารณาทุกจุดในข้อมูลภาพได้เสร็จสิ้นลงแล้ว โดยจุดดำทุกจุดที่ถูกจัดตำแหน่งไว้จะถูกลบไปพร้อมๆกันในทีเดียว เพราะถ้าหากเรากำจัดตำแหน่งไว้แล้วทำการลบเวลานั้นเลย ในการพิจารณาจุดต่อไปนั้นจุดที่ได้ลบไปก่อนหน้าอาจจะทำให้เกิดผลผิดพลาดในการทำเส้นบางได้ ดังนั้นจึงต้องกำจัดตำแหน่งที่จะลบไว้ก่อนแล้วค่อยลบพร้อมๆกันเมื่อทำการพิจารณาแล้วทุกจุดดำในข้อมูลภาพ โดยการทำเส้นบางจะสิ้นสุดลงเมื่อตรวจไม่พบจุดที่มีความหนาเกิน 1 จุด

รูปการทำงานของกระบวนการทำเส้นบางดังแสดงไว้ในรูปที่ 27 และสมการตรรกะที่ใช้ในโปรแกรมดังแสดงไว้ในสมการที่ 2.3 และ 2.4

3.2.5.1 โครงสร้างโปรแกรมกระบวนการทำเส้นบาง



รูปที่ 27 โครงสร้างโปรแกรมกระบวนการทำเส้นบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายบล็อกรูปที่ 27 โครงสร้างโปรแกรมกระบวนการทำเส้นทาง

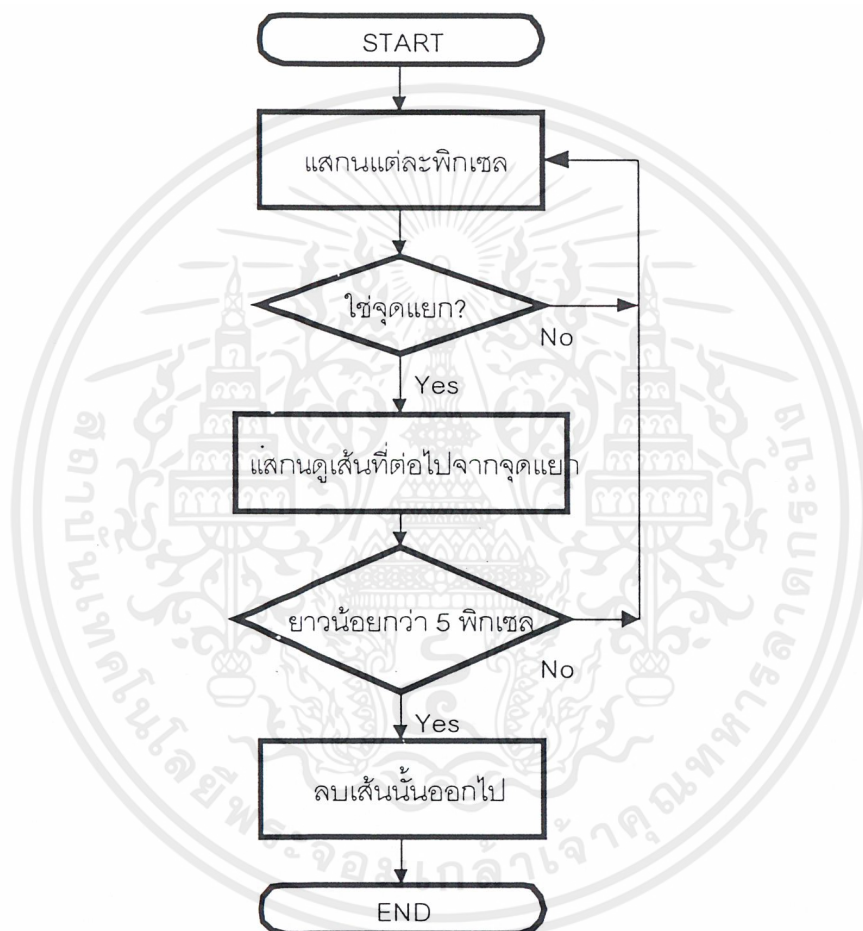
1. เริ่มการทำงาน
2. เริ่มแตกนจากซ้ายไปขวาและล่างขึ้นบนจนหมด
3. พบจุดดำหรือยัง?
4. นำสมการ $3 \leq Np \leq 7?$ และ $Sp = 1$? มาพิจารณา โดย Np คือ ผลรวมจุดดำรอบจุด $x0$ ทั้งหมด และ Np คือจำนวนครั้งของการเปลี่ยนแปลงจุดรอบข้างจากดำเป็นขาว
5. ตัวแปร $Step = 0$?
6. นำสมการตรรก $(x2 \parallel x4 \parallel x8 = white) \&\& (x2 \parallel x6 \parallel x8 = white) = true$? มาพิจารณา
7. นำสมการตรรก $(x2 \parallel x4 \parallel x6 = white) \&\& (x4 \parallel x6 \parallel x8 = white) = true$? มาพิจารณา
8. ทำการ mark จุด $x0$ ไว้
9. แสกนหมดรูปหรือยัง?
10. ทำการลบจุดที่ mark ไว้แล้ว convert ตัวแปร step จาก 0 เป็น 1 หรือจาก 1 เป็น 0
11. แสกนหมดรูปหรือยัง?
12. จบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 กระบวนการกำจัดส่วนเกิน

เมื่อภาพตัวเลขได้ผ่านการทำเส้นบางมาแล้ว อาจทำให้เกิดส่วนเกินขึ้นได้ดังแสดงในรูปที่ 28 ดังนั้นเราจึงต้องมีโปรแกรมเพื่อกำจัดส่วนเกินไว้ด้วย

3.2.6.1 โครงสร้างโปรแกรมกระบวนการกำจัดส่วนเกิน

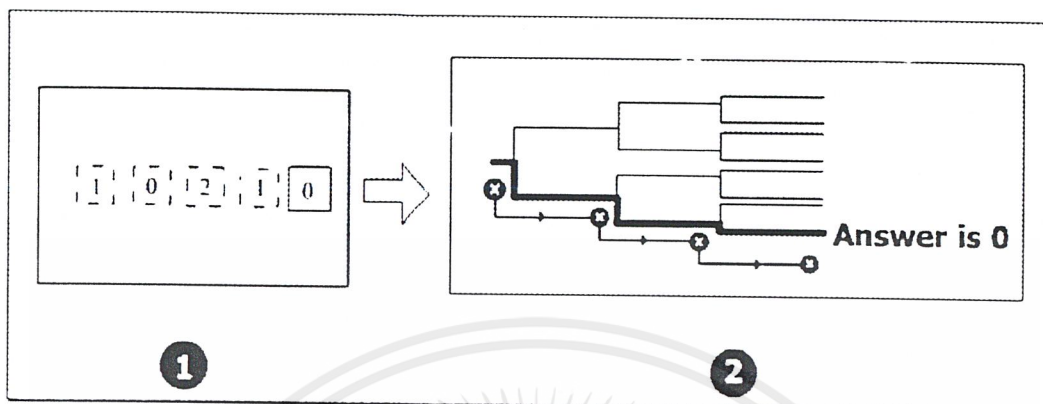


รูปที่ 28 โครงสร้างโปรแกรมกระบวนการกำจัดส่วนเกิน

3.3 ระบบส่วนการรู้จำ

ข้อมูลบล็อกเดี่ยวของภาพตัวเลขที่ได้ทำการแยกออกมาจากข้อมูลภาพรวมของซองจดหมายจะนำมาเข้าสู่กระบวนการรู้จำในส่วนนี้ ซึ่งกระบวนการรู้จำที่ใช้จะใช้หลักการวิเคราะห์โครงสร้างของข้อมูลภาพแล้วนำมาเปรียบเทียบกับเงื่อนไขที่เราได้ทำการสำรวจเก็บไว้ก่อนหน้านี้แล้ว และทำการเปรียบเทียบข้อมูลไปเรื่อยๆ จนกว่าจะพบข้อมูลสุดท้ายของเงื่อนไข คอมพิวเตอร์จะทำการตัดสินใจแล้วตอบออกมาว่าข้อมูลภาพที่ทำการนำมาวิเคราะห์นั้นคือตัวเลขอะไร โดยโปรแกรมที่ใช้เก็บข้อมูลก็นำมาเปรียบเทียบกับวิเคราะห์ข้อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มัลทภาพนี้จะใช้ ภาษาโปรแกรมคอมพิวเตอร์ Visual Basic เป็นตัวเขียนขึ้นมา โดยรูปโครงสร้างระบบส่วน การรู้จำโดยสังเขปมีดังต่อไปนี้



1. เงื่อนไขต่างๆในโปรแกรมที่นำไปสู่คำตอบ บล็อกข้อมูลแรกที่เข้าทำการวิเคราะห์
 2. บล็อกภาพตัวเลขที่แยกออกจากภาพรวมแล้ว บล็อกข้อมูลถัดมา
 3. โครงสร้างโปรแกรมย่อย ในรูปเศษวงโครงสร้าง เงื่อนไขแบบค้นไม้นักการวิเคราะห์นำไปสู่ข้อมูล ที่ถูกต้องที่สุด

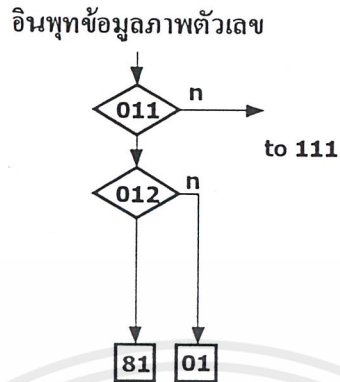
รูปที่ 29 โครงสร้างระบบส่วนการรู้จำโดยสังเขป

จากรูปข้างต้น ข้อมูลภาพที่แยกออกมาเป็นบล็อกๆแล้ว จะนำเข้าสู่โปรแกรมเพื่อทำการวิเคราะห์ ลักษณะพิเศษของแต่ละภาพแล้วนำลักษณะพิเศษนั้นไปเข้าตรวจสอบกับเงื่อนไขโดยเงื่อนไขคือข้อมูลที่เร ทำการสำรวจแล้วเก็บข้อมูลไว้แล้วให้ครอบคลุมมากที่สุด เมื่อผ่านเงื่อนไขแรกแล้วจะแยกเข้าสู่การตรวจสอบเงื่อนไขต่อไปจนสิ้นสุดทุกเงื่อนไข หลังจากนั้นแล้วคอมพิวเตอร์จะทำการตัดสินใจแล้วบอกให้เรา ทราบว่าข้อมูลภาพตัวเลขที่เรานำมาป้อนให้แก่นั้นนั้นคือตัวเลขอะไร ในรูปตัวอย่างบล็อกแรกคือเลข 0 ตาม เป้าหมายของโครงการเมื่อนำข้อมูลภาพเลข 0 เข้าไปทำการวิเคราะห์ในโปรแกรมแล้วคำตอบที่คอมพิวเตอร์ บอกแก่เราต้องเป็นเลข 0 เช่นเดียวกันดังในรูปเมื่อผ่านโปรแกรมวิเคราะห์ลักษณะพิเศษแล้วคำตอบที่ได้คือ 0 ตรงตามอินพุทที่ป้อนเข้าไป คำตอบที่ได้จึงถูกต้องตรงตามเป้าหมาย

จากบทที่ที่แล้วเรากล่าวถึงองค์ประกอบที่ใช้ในการเขียนโปรแกรมเพื่อสร้างระบบจดจำไปแล้ว มาถึง จุดนี้เราจะเริ่มทำการเขียนโปรแกรมของระบบ โดยการออกแบบโครงสร้างนั้นเราจะแยกอัลกอริทึมออกเป็น 5 ส่วนด้วยกัน โดยยึดหลักของจำนวนจุดปลาย โดยเริ่มจากภาพที่ไม่มีจุดปลาย มี 1 2 3 4 จุดปลายตามลำดับ เรียกว่า Case 0 – Case 4 ที่เป็นเช่นนี้เพราะอัลกอริทึมของเรา ใช้หลักในการพิจารณาหาจุดปลายของข้อมูล ภาพก่อนนั่นเอง และในส่วนคำตอบสุดท้ายนั้นจะเป็นตัวเลข 0-9 ที่มีลักษณะหลายๆลักษณะ มีหลายรูปแบบ แตกต่างกันไป ต่อไปเราจะแสดงให้เห็นถึงโครงสร้างโปรแกรมในส่วนของการวิเคราะห์โครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 0



รูปที่ 30 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 0

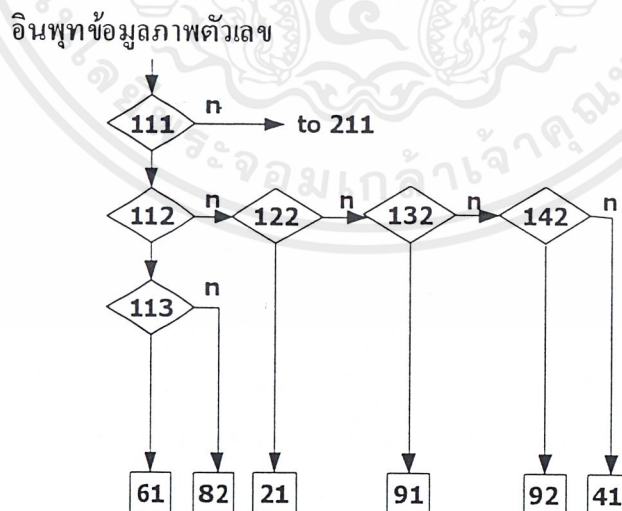
อธิบายบล็อกรูปที่ 30 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 0

011 ตรวจสอบว่ามีจุดปลายหรือไม่

012 ตรวจสอบว่ามีจุดแยกหรือจุดตัดหรือไม่

81 และ 01 เป็นคำตอบ รูปภาพของเลข 8 แบบที่ 1 และเลข 0 แบบที่ 1

3.3.2 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 1



รูปที่ 31 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายบล็อกรูปที่ 31 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 1

111 ตรวจสอบว่ามี 1 จุดปลายหรือไม่

112 ตรวจสอบว่าจุดปลายอยู่ครึ่งบนของรูปภาพหรือไม่

113 ตรวจสอบว่าถ้าทำเส้นตัดในแนวนานกับแกน x ตัดเส้นที่เป็นส่วนล่างของจุดปลายแล้วเกิดจุดตัด 1 จุดหรือไม่

122 จุดที่ต่อจากจุดปลายมีทิศวิ่งไปทางทิศซ้ายหรือไม่

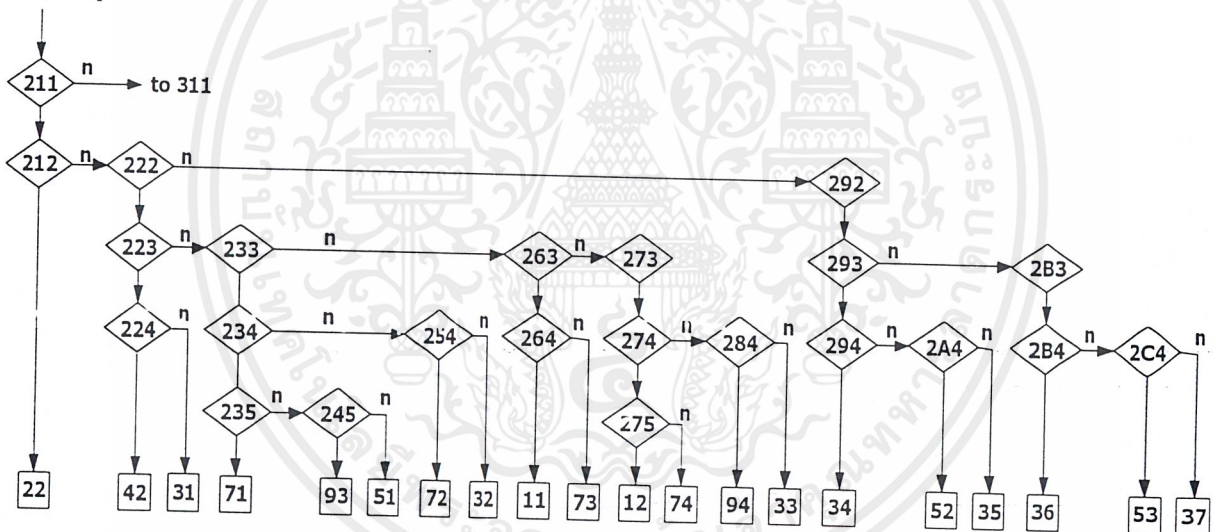
132 จุดที่ต่อจากจุดปลายมีทิศวิ่งลงด้านล่างหรือไม่

142 ตรวจสอบว่าจุดแยกที่พบมีจุดต่ำกว่าจุดกึ่งกลางรูปภาพหรือไม่

61 82 21 91 92 และ 41 คือคำตอบของรูปภาพ

3.3.3 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 2

อินพุทข้อมูลภาพตัวเลข



รูปที่ 32 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 2

อธิบายบล็อกรูปที่ 32 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 2

211 ตรวจสอบว่ามี 2 จุดปลายหรือไม่

212 จุดปลายที่อยู่ต่ำกว่ามีทิศทางวิ่งไปทางด้านซ้ายหรือไม่

222 จุดปลายที่อยู่ต่ำกว่ามีทิศทางวิ่งไปทางด้านบนหรือไม่

223 มีจุดแยกหรือไม่

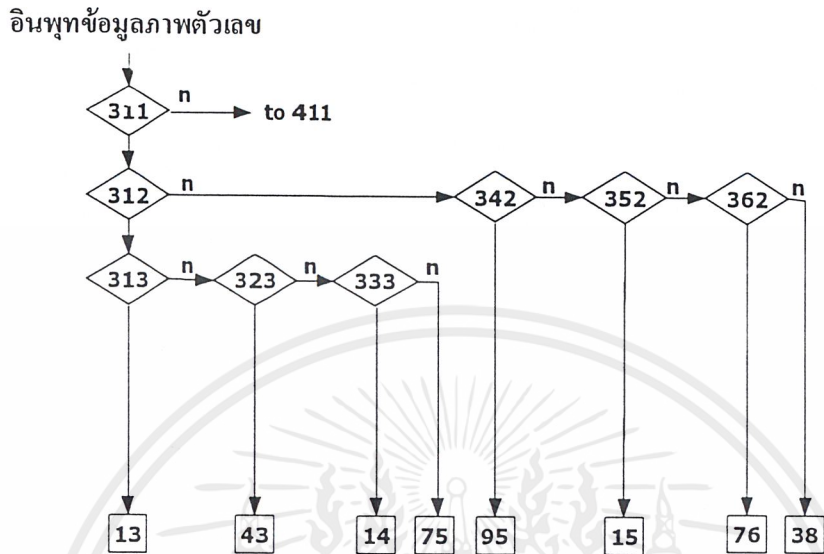
224 จุดปลายที่อยู่สูงกว่ามีทิศทางวิ่งไปทางด้านซ้ายหรือไม่

233 จุดปลายที่อยู่สูงกว่ามีทิศทางวิ่งไปทางด้านซ้ายหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 234 จุดปลายที่อยู่สูงกว่าอยู่ด้านขวาของภาพหรือไม่
- 235 จุดปลายที่อยู่สูงกว่าอยู่ต่ำกว่าจุดสูงสุดของภาพน้อยกว่า 5 พิกเซลหรือไม่
- 245 จุดปลายทั้งสูงและต่ำถ้าทำเส้นหลักผ่านจุดปลายทั้งสองขนานกับแกน y เส้นหลักทั้งสองเส้นห่างกันไม่เกิน 7 พิกเซลใช่หรือไม่
- 254 เส้นทางการเดินทางระหว่างจุดปลายด้านล่างถึงจุดปลายด้านบนมีทิศทางขึ้นด้านบน
- 263 จุดปลายที่อยู่สูงกว่ามีทิศทางวิ่งลงด้านล่าง
- 264 ตรวจสอบว่าความกว้างของภาพน้อยกว่า $1/3$ ของความสูงหรือไม่
- 273 จุดปลายที่อยู่สูงกว่ามีทิศทางวิ่งไปทางด้านบนหรือด้านขวาหรือไม่
- 274 เส้นทางการเดินทางระหว่างจุดปลายด้านล่างถึงจุดปลายด้านบนมีทิศทางขึ้นด้านบน
- 275 ตรวจสอบว่าความกว้างของภาพน้อยกว่า $1/3$ ของความสูงหรือไม่
- 284 ตรวจสอบว่าจุดปลายที่อยู่สูงกว่าอยู่ด้านขวาของภาพหรือไม่
- 292 จุดปลายที่อยู่ต่ำกว่ามีทิศทางวิ่งลงทางด้านล่างหรือด้านขวาหรือไม่
- 293 จุดปลายที่อยู่สูงกว่ามีทิศทางวิ่งไปทางซ้ายหรือไม่
- 294 จุดปลายที่อยู่สูงกว่าอยู่ต่ำกว่าจุดสูงสุดของภาพน้อยกว่า 5 พิกเซลหรือไม่
- 2A4 จุดปลายทั้งสูงและต่ำถ้าทำเส้นหลักผ่านจุดปลายทั้งสองขนานกับแกน y เส้นหลักทั้งสองเส้นห่างกันเกิน 7 พิกเซลใช่หรือไม่
- 2B3 จุดปลายที่อยู่สูงกว่ามีทิศทางวิ่งไปทางด้านบนหรือด้านขวาหรือไม่
- 2B4 จุดปลายที่อยู่สูงกว่าอยู่ต่ำกว่าจุดสูงสุดของภาพน้อยกว่า 5 พิกเซลหรือไม่
- 2C4 จุดปลายทั้งสูงและต่ำถ้าทำเส้นหลักผ่านจุดปลายทั้งสองขนานกับแกน y เส้นหลักทั้งสองเส้นห่างกันเกิน 7 พิกเซลใช่หรือไม่
- 22 32 11 73 12 74 94 33 34 52 35 36 53 และ 37 คือคำตอบของรูปภาพ

3.3.4 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 3



รูปที่ 33 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 3

อธิบายบล็อกรูปที่ 33 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 3

311 ตรวจสอบว่ามี 3 จุดปลายหรือไม่

312 ตรวจสอบว่าจุดปลายที่อยู่สูงสุดมีทิศทางวิ่งลงด้านล่างหรือด้านซ้ายหรือไม่

313 จุดปลายด้านล่างสุดกลับจุดปลายที่อยู่ตรงกลางมีความสูงห่างกันไม่เกิน 5 พิกเซล

323 ตรวจสอบว่าจุดปลายที่อยู่ตรงกลางมีทิศทางวิ่งลงด้านล่างหรือด้านซ้ายหรือไม่

333 ตรวจสอบว่าความกว้างของภาพน้อยกว่า 1/3 ของความสูงหรือไม่

342 จุดปลายที่อยู่ตรงกลางมีทิศทางวิ่งลงด้านล่างหรือไม่

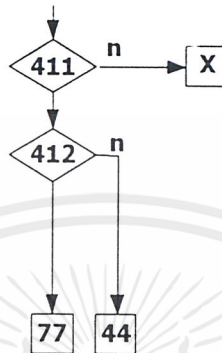
352 จุดปลายด้านล่างสุดกลับจุดปลายที่อยู่ตรงกลางมีความสูงห่างกันไม่เกิน 5 พิกเซล

362 จุดปลายที่อยู่สูงสุดกับจุดสูงสุดของภาพมีความสูงห่างกันน้อยกว่า 1/3 ของความสูง

13 43 14 75 95 15 76 และ 38 คือคำตอบของรูปภาพ

3.3.5 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 4

อินพุทข้อมูลภาพตัวเลข



รูปที่ 34 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 4

อธิบายบล็อกรูปที่ 34 โครงสร้างโปรแกรมวิเคราะห์โครงสร้าง Case 4

411 ตรวจสอบว่ามี 4 จุดปลายหรือไม่

412 จุดปลายที่อยู่สูงสุดมีทิศทางวิ่งไปทางด้านบนหรือด้านขวา

X ไม่สามารถหาคำตอบได้

77 และ 44 คือคำตอบของรูปภาพ

ที่กล่าวไปทั้งหมดเป็นโครงสร้างของโปรแกรมที่เราจะทำการเขียนขึ้นไป โดยหลังจากทำการสร้างโปรแกรมที่จะมาใช้ในระบบได้แล้ว ขั้นตอนต่อไปคือการทดลองเพื่อตรวจสอบผลว่าโปรแกรมที่เราได้ทำการสร้างขึ้นมานั้นมีความถูกต้องมากน้อยแค่ไหนและมีความผิดพลาดอย่างไรเพื่อทำการแก้ไขต่อไป

บทที่ 5

การทดลองและผลการทดลอง

เมื่อเราทำการสร้างโปรแกรมขึ้นมาแล้วตามโครงสร้างโปรแกรมในบทที่แล้ว ในบทนี้จะกล่าวถึงการทดลองใช้งานของโปรแกรม โดยจะแบ่งเป็น 3 ตอนคือ

ตอนที่ 1 การทดลองแยกข้อมูลภาพตัวเลขออกจากกรอบ

ตอนที่ 2 การทดลองให้โปรแกรมบอกค่าของตัวเลขจากภาพตัวเลข

ตอนที่ 3 การทดลองให้โปรแกรมบอกเลขรหัสไปรษณีย์จากภาพตัวเลข

โดยก่อนจะทำการทดลองขอให้เราทราบถึงรายละเอียดของข้อมูลที่จะนำมาเป็นอินพุทของระบบนั้นก็คือรูปของจดหมายไปรษณีย์กันก่อน โดยทั้งสามขั้นตอนการทดลองจะมีรายละเอียดของข้อมูลอินพุทในรูปแบบเดียวกัน แต่จะแตกต่างกันตรงตัวเลขที่เราทำการกรอกลงไปในช่วงใส่รหัสไปรษณีย์เท่านั้น เนื่องมาจากผลที่เราต้องการได้จากการทดลองในแต่ละขั้นตอนแตกต่างกันออกไป รายละเอียดโครงสร้างมาตรฐานของภาพที่จะนำมาเป็นอินพุทดังต่อไปนี้

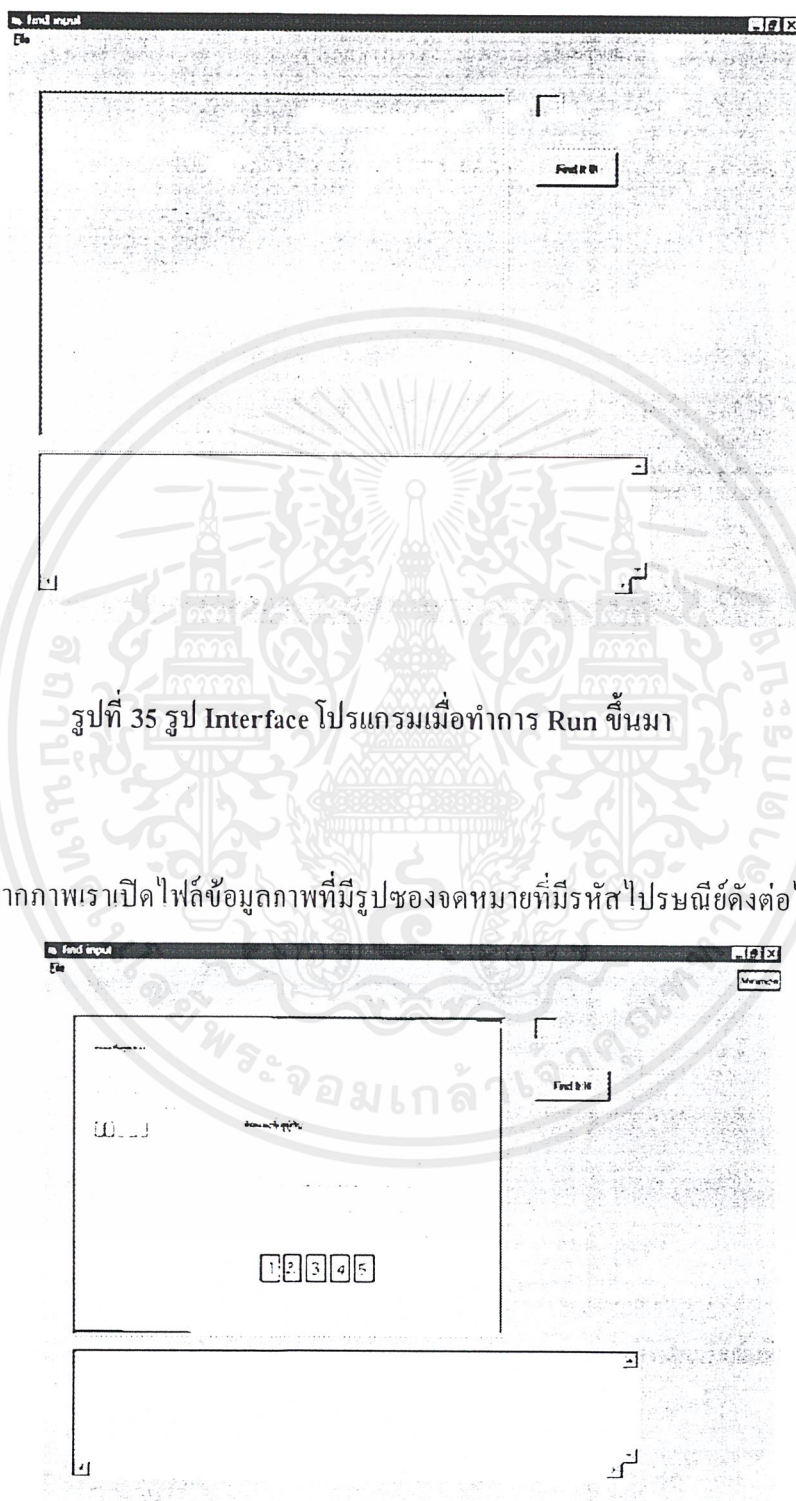
ตอนที่ 1 การทดลองแยกข้อมูลภาพตัวเลขออกจากกรอบ

การทดลองในส่วนนี้เราจะใช้โปรแกรมที่ทำการเขียนขึ้นมาแต่จะให้เห็นผลเฉพาะในส่วนของการแยกภาพตัวเลขออกจากกรอบเท่านั้นมาทำการทดลองในขั้นตอนนี้ โดยขั้นตอนการทดลองมีดังต่อไปนี้

1. ทำการ Run โปรแกรมที่ได้เขียนขึ้นมา
2. เปิดไฟล์ข้อมูลภาพสกุล BMP ที่ได้ทำการเตรียมไว้ขึ้นมา 1 ภาพ
3. เมื่อทำการกดปุ่ม Find it 1 ครั้ง โปรแกรมจะทำการแยกตัวเลขรหัสออกมาครั้งละ 1 ตัวเลข
4. สังเกตผลที่ได้และทำการแก้ไข

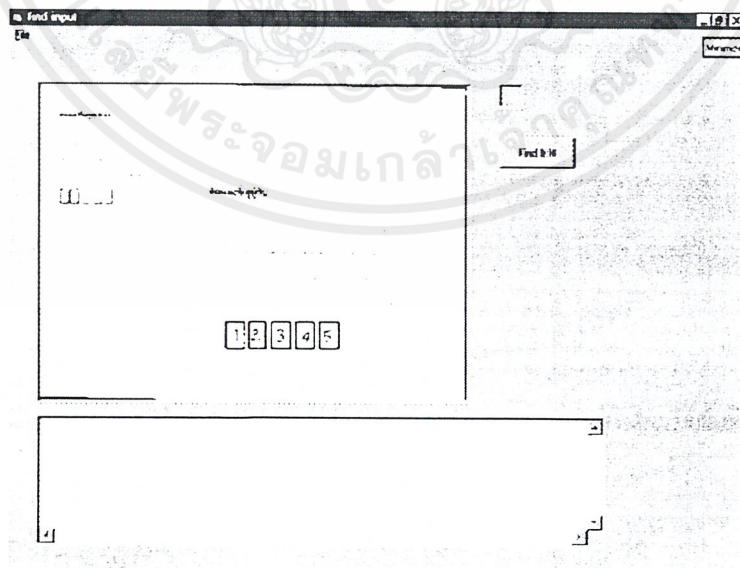
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1 ทำการ Run โปรแกรมที่ได้เขียนขึ้นมา



รูปที่ 35 รูป Interface โปรแกรมเมื่อทำการ Run ขึ้นมา

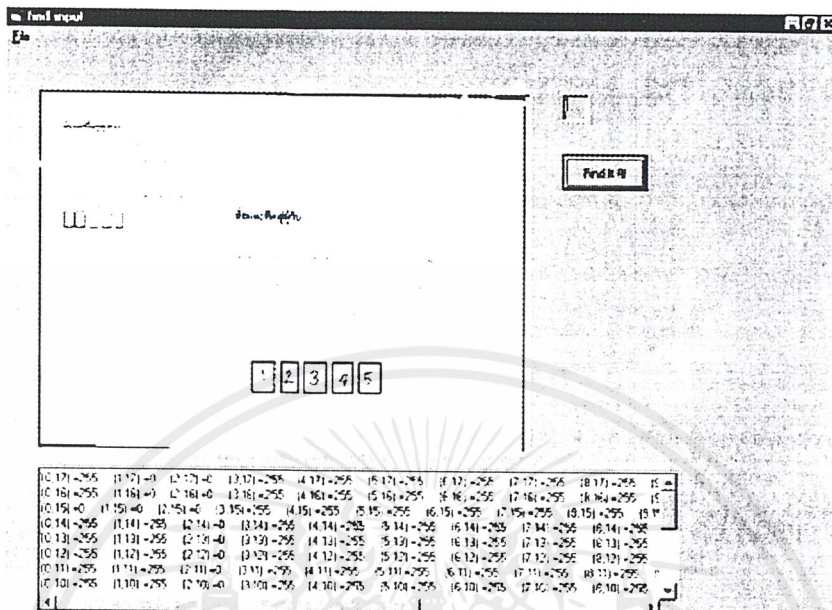
ขั้นตอนที่ 2 จากภาพเราเปิดไฟล์ข้อมูลภาพที่มีรูปของจดหมายที่มีรหัสไปรษณีย์ดังต่อไปนี้ คือ 12345



รูปที่ 36 รูป Interface โปรแกรมเมื่อทำการ Import ภาพเข้ามาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 3 ทำการกดปุ่ม Find it เพื่อทำการแยกเลขรหัสไปรษณีย์ออกมาจากรอบ



รูปที่ 37 รูป Interface โปรแกรมเมื่อทำการกดปุ่มแสกนหารอบภาพ

ในขั้นตอนนี้ให้กดจนกว่าโปรแกรมจะทำการแยกตัวเลขออกมาจากรอบครบทั้ง 5 ตัวแล้ว

โปรแกรมจะหยุดการทำงาน

ขั้นตอนที่ 4 สังเกตผลที่ได้

จากการทดลองที่ผ่านมาพบว่าการแยกคัดแยกเลขรหัสไปรษณีย์ออกมาจากรูปภาพมีความถูกต้องทุกตัวเลข โดยการประมวลผลจะทำให้ที่ละหนึ่งตัวเลขต่อการกดหนึ่งครั้งเริ่มจากตัวเลขหลักแรกไปจนถึงตัวเลขหลักสุดท้าย และจากการทดลองใช้งานโปรแกรมพบว่าเมื่อทำการคัดแยกตัวเลขครบห้าตัวแล้ว หากเรากดปุ่มเพื่อทำการประมวลผลอีก จะไม่มีอะไรเกิดขึ้นเนื่องจากได้ทำการคัดแยกตัวเลขครบทั้งห้าหลักแล้วนั่นเอง

ตอนที่ 2 การทดลองให้โปรแกรมบอกค่าของตัวเลขจากภาพตัวเลข

การทดลองในส่วนนี้เราจะใช้โปรแกรมที่ทำการเขียนขึ้นมานำมาทดลองโดยการป้อนอินพุตด้วยเลข 0-9 และให้โปรแกรมทำการตอบว่าข้อมูลภาพตัวเลขนั้นเป็นภาพของตัวเลขใด โดยข้อมูลอินพุตเราจะทำการสุ่มโดยการหาผู้เขียนที่เป็นกลุ่มตัวอย่างจำนวน 50 คน 50 ลายมือ โดยใช้ซองจดหมายจำนวน 100 ซอง และผู้เขียนที่มาจากกลุ่มจะต้องเขียนตัวเลข 0-4 ลงในจดหมายซองแรก และ 5-9 ลงในจดหมายซองที่สอง โดยลายมือบรรจงหรือไม่บรรจงก็ได้ไม่บังคับโดยขั้นตอนการทดลองมีดังต่อไปนี้

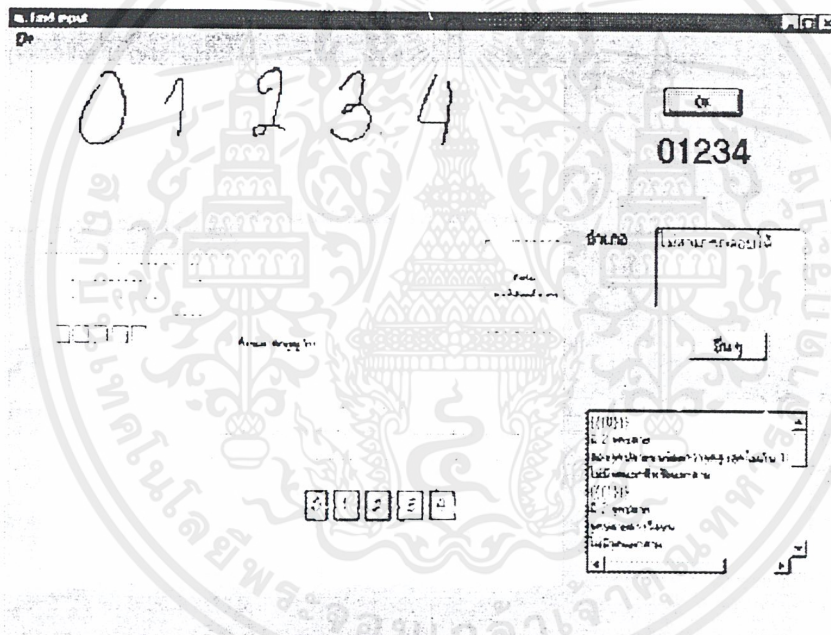
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สุ่มหากลุ่มตัวอย่างและให้เขียนตัวเลขลงในช่องจดหมายคนละ 2 ช่อง ด้วยตัวเลข 0-9 เพื่อนำมาเป็นอินพุทของระบบ
2. ทำการ Run โปรแกรมที่ได้เขียนขึ้นมา
3. เปิดไฟล์ข้อมูลภาพนามสกุล BMP ที่ได้ทำการเตรียมไว้ขึ้นมา
4. ทำการ Run โปรแกรมเพื่อประมวลผลของที่ 1
5. ทำข้อ 3 และ ข้อ 4 วนไปจนข้อมูลภาพที่เตรียมไว้หมด 100 ช่อง
6. สังเกตและบันทึกผลที่ได้

เมื่อหาตัวอย่างลายมือได้ครบแล้ว

ขั้นตอนที่ 1 ทำการ Run โปรแกรมที่ได้เขียนขึ้นมา

ขั้นตอนที่ 2 จากภาพเราเปิดไฟล์ข้อมูลภาพที่มีรูปช่องจดหมายที่มีรหัสไปรษณีย์ดังต่อไปนี้ คือ 01234



รูปที่ 38 รูปขณะทำการประมวลผลในขั้นตอนหาค่าตัวเลขจากข้อมูลภาพ

ขั้นตอนที่ 3 ทำการกดปุ่ม Run โปรแกรมเพื่อประมวลผลหาคำตอบ

ขั้นตอนที่ 4 สังเกตและบันทึกผลที่ได้ลงในตารางต่อไปนี้

ช่องที่	เลข 0	เลข 1	เลข 2	เลข 3	เลข 4	เลข 5	เลข 6	เลข 7	เลข 8	เลข 9
1	T	T	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	T	T	T

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3	T	F	T	T	T	F	T	T	T	T
4	T	T	T	T	T	T	F	T	T	T
5	F	T	T	T	T	T	T	T	T	T
6	T	F	T	T	F	T	T	T	F	T
7	T	T	T	T	F	T	T	F	T	T
8	F	F	F	T	F	T	T	T	T	T
9	F	T	F	F	T	F	T	T	F	T
10	F	T	T	T	T	T	T	F	T	T
11	T	T	T	T	T	T	T	F	T	T
12	T	T	T	T	F	T	T	F	T	T
13	F	F	T	T	T	T	F	T	T	T
14	T	F	T	T	T	T	T	T	T	T
15	T	T	T	T	T	F	T	T	T	T
16	T	T	T	T	T	T	T	T	T	T
17	T	T	T	T	T	F	T	F	T	F
18	T	F	T	T	T	T	T	F	F	F
19	T	F	T	F	T	T	F	T	F	T
20	F	T	F	T	T	T	T	F	T	T
21	T	T	T	F	T	T	T	F	T	T
22	T	T	T	T	T	T	T	T	T	T
23	F	T	T	T	T	T	F	T	T	F
24	T	T	T	T	T	F	T	F	F	F
25	F	F	T	T	F	F	T	F	T	T
26	T	T	T	T	T	T	F	T	T	T
27	T	T	T	T	F	F	F	F	T	T
28	T	T	T	F	T	T	T	F	F	T
29	F	T	T	T	F	F	T	T	T	T
30	F	T	T	T	T	T	F	T	T	T
31	F	F	T	T	T	T	T	F	T	T
32	F	F	T	T	T	F	T	T	T	T

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

33	F	F	T	T	T	F	F	T	F	T
34	F	T	T	F	T	T	F	F	F	T
35	T	T	T	F	T	T	T	T	T	F
36	T	F	F	T	T	T	T	T	T	T
37	F	F	T	T	T	T	F	T	T	T
38	T	T	T	T	F	T	T	F	T	T
39	T	T	T	T	T	T	T	T	T	T
40	T	T	T	T	T	T	T	F	T	T
41	T	T	T	T	T	T	T	F	T	T
42	T	T	T	T	T	T	F	F	T	F
43	T	T	T	T	F	T	T	T	T	T
44	T	T	T	F	T	F	T	T	T	T
45	T	T	T	T	F	F	F	T	T	T
46	F	F	F	T	T	T	T	F	T	T
47	F	T	T	T	F	F	T	T	T	T
48	T	T	T	T	T	T	T	T	T	T
49	T	T	F	T	T	T	T	T	T	T
50	T	T	F	T	T	T	T	T	T	T

ตารางที่ 1 บันทึกลำค่าความถูกต้องในขั้นตอนหาค่าตัวเลขจากข้อมูลภาพ

ผลที่ได้จากการบันทึกค่ามีดังนี้

เลข 0 ถูก 33 ผิด 17 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 66% ผิด 34%

เลข 1 ถูก 36 ผิด 14 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 72% ผิด 28%

เลข 2 ถูก 43 ผิด 7 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 86% ผิด 14%

เลข 3 ถูก 43 ผิด 7 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 86% ผิด 14%

เลข 4 ถูก 39 ผิด 11 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 78% ผิด 22%

เลข 5 ถูก 37 ผิด 13 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 74% ผิด 36%

เลข 6 ถูก 33 ผิด 17 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 66% ผิด 34%

เลข 7 ถูก 31 ผิด 19 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 62% ผิด 38%

เลข 8 ถูก 42 ผิด 8 เพราะฉะนั้น มีการตรวจสอบถูกคิดเป็น 84% ผิด 16%

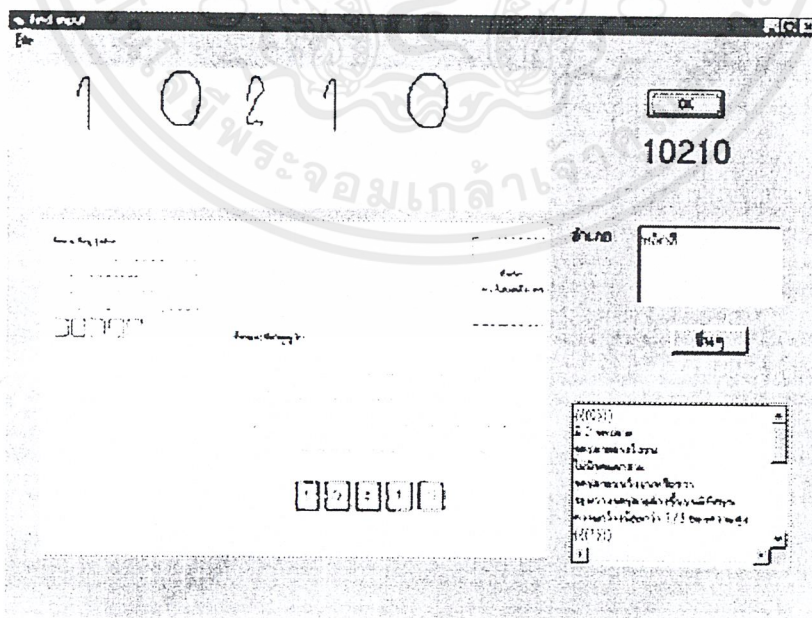
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลข 9 ถูก 44 ผิด 6 เพราะฉะนั้น มีการตรวจสอบถูกต้องเป็น 88% ผิด 12%

ตอนที่ 3 การทดลองให้โปรแกรมบอกเลขรหัสไปรษณีย์จากภาพตัวเลข

การทดลองในส่วนนี้เราจะใช้โปรแกรมที่ทำการเขียนขึ้นมา มาทดลองโดยการป้อนอินพุตด้วยเลข
เลขรหัสไปรษณีย์ และให้โปรแกรมทำการตอบว่าข้อมูลภาพตัวเลขนั้นเป็นรหัสไปรษณีย์ของอำเภอใด โดย
ข้อมูลอินพุตเราจะทำการสุ่มโดยการหาผู้เขียนที่เป็นกลุ่มตัวอย่างจำนวน 25 คน 25 ลายมือ โดยใช้ซองจด
หมายจำนวน 25 ซองและผู้เขียนที่มาจากกลุ่มจะต้องเขียนตัวเลขรหัสไปรษณีย์อะไรก็ได้ลงในซองจด
หมายโดยลายมือบรรจงขั้นตอนการทดลองมีดังต่อไปนี้

7. สุ่มหากกลุ่มตัวอย่างและให้เขียนตัวเลขรหัสไปรษณีย์อะไรก็ได้ลงในซองจดหมายคนละ 1 ซอง 25
คนเพื่อนำมาเป็นอินพุตของระบบ
 8. ทำการ Run โปรแกรมที่ได้เขียนขึ้นมา
 9. เปิดไฟล์ข้อมูลภาพนามสกุล BMP ที่ได้ทำการเตรียมไว้ขึ้นมา
 10. ทำการ Run โปรแกรมเพื่อประมวลผลของที่ 1
 11. ทำข้อ 3 และ ข้อ 4 วนไปจนข้อมูลภาพที่เตรียมไว้หมด 25 ซอง
 12. สังเกตและบันทึกผลที่ได้
- เมื่อหาตัวอย่างลายมือ ได้ครบแล้ว
- ขั้นตอนที่ 1 ทำการ Run โปรแกรมที่ได้เขียนขึ้นมา
- ขั้นตอนที่ 2 จากภาพเราเปิดไฟล์ข้อมูลภาพที่มีรูปของจดหมายที่มีรหัสไปรษณีย์ดังต่อไปนี้ คือ 10210



รูปที่ 39 รูปขณะทำการประมวลผลในขั้นตอนหาเลขรหัสไปรษณีย์จากข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 3 ทำการกดปุ่ม Run โปรแกรมเพื่อประมวลผลหาคำตอบ
ขั้นตอนที่ 4 สังเกตและบันทึกผลที่ได้ลงในตารางต่อไปนี้

ช่องที่	เลขรหัสไปรษณีย์	คำตอบที่ได้รับ	ผล
1	10210	หลักสี่, ดอนเมือง	T
2	10400	ดินแดง, ราชเทวี, พญาไท	T
3	10120	สาทร, ยานนาวา, บางคอแหลม	T
4	10120	สาทร, ยานนาวา, บางคอแหลม	T
5	10160	บางแค, หนองแขม, ภาษีเจริญ	T
6	10260	บางนา, พระโขนง, แขวงดอกไม้ หนองบอน เฉพาะหมู่ที่ 2-10	T
7	71000	อำเภอท่าลื้อ หมู่ที่ 1-2, อำเภอเมืองกาญจนบุรี	T
8	71150	อำเภอไทรโยค จังหวัดกาญจนบุรี	T
9	40000	อำเภอเมืองขอนแก่น	T
10	24150	อำเภอบางน้ำเปรี้ยว	T
11	20000	อำเภอเมืองชลบุรี	T
12	10310	บางกะปิ	F
13	20180	อำเภอสัตหีบ	T
14	57140	คอยสุเทพ	F
15	36000	ตำบลชัยสิทธิ์ทอง, อำเภอเมืองชัยภูมิ	T
16	57150	อำเภอเชียงแสน	T
17	50110	อำเภอฝาง	T
18	92000	ต.ท่าสะบ้า, ต.นาเมืองเพชร, อำเภอเมืองตรัง	T
19	73150	อำเภอแกลง	F
20	26110	อำเภอบ้านนา	T
21	73120	อำเภอนครชัยศรี	T
22	73170	อำเภอพุทธมณฑล	T
23	65170	อำเภอชาติตระการ	T
24	37000	อำเภอศรีประจัน	F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

25	67270	อำเภอเขาค้อ	T
----	-------	-------------	---

ตารางที่ 2 บันทึกค่าความถูกต้องในขั้นตอนหาหาเลขรหัสไปรษณีย์จากข้อมูลภาพ
จากผลการทดลองที่ได้ในขั้นตอนนี้พบว่ามีการวิเคราะห์ที่ถูกต้อง 21 ชอง คิดเป็น 84% ของทั้งหมด
และผิดเป็นจำนวน 4 ชอง คิดเป็น 16% ของทั้งหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและวิจารณ์

6.1 ส่วนของการเตรียมข้อมูล

ในส่วนของการเตรียมข้อมูลทางผู้ทำโครงการพบว่า การเตรียมข้อมูลนั้นเราจำเป็นต้องเตรียมข้อมูลให้เป็นมาตรฐานเดียวกัน โดยทำการแสดกภาพมาที่ขนาดเดียวกัน เพื่อทำการประมวลผลภาพจะได้ถูกต้อง ซึ่งก็เป็นไปด้วยดี เมื่อทำการตั้งค่ามาตรฐานของขนาดภาพไว้แล้วทำตามนั้น โดยขนาดที่ทำการแสดกมานั้นเหมาะสมต่อการทำการประมวลผลภาพ ไม่ใหญ่หรือเล็กจนเกินไปนัก เพราะถ้าใหญ่เกินไปจะทำให้เปลืองเนื้อที่ในการเก็บข้อมูลภาพ และทำให้การประมวลผลช้าลงอีกด้วย แต่ถ้าหากเราทำการเตรียมภาพมาในขนาดที่เล็กไปนั้นก็ทำให้ภาพที่ทำการประมวลผลออกมาได้ดูไม่รู้เรื่องเพราะเล็กเกินไป และในส่วนของสัญญาณรบกวนก็ไม่มีปัญหาใดๆ เพราะมีสัญญาณรบกวนปรากฏเพียงเล็กน้อยหลังจากที่ได้ทำการแสดกภาพ แต่ไม่มีผลแต่อย่างใด แต่อย่างไรก็ดีในส่วนของการโปรแกรมผู้ทำโครงการก็ได้ทำการเขียนในส่วนของการกำจัดสัญญาณรบกวนไว้แล้ว

6.2 ส่วนการประมวลผลภาพ

การประมวลผลภาพ อาจเกิดคำถามว่า ทำไมในกระบวนการของการทำ Detect Edge ทำไมผู้ทำโครงการไม่ใช้กระบวนการ Contour Following ไปเลยวิธีเดียว เพราะก็เป็นหาขอบของภาพเช่นเดียวกัน ทำไมต้องเพิ่มมาอีกวิธีการทำให้ยุ่งยาก

จากการทดลองผู้ทำโครงการพบว่า กระบวนการ Contour Following นั้นเหมาะกับการใช้หากรอบภาพที่ราบเรียบติดต่อกันอย่างสม่ำเสมอ เช่นกรอบสี่เหลี่ยม วงกลม เป็นต้น ถึงแม้ว่ากระบวนการ Contour Following จริงๆแล้วสามารถใช้หาขอบภาพทุกกรณีก็ตาม ดังที่ได้กล่าวมาแล้ว เมื่อใช้กระบวนการ Contour Following กับตัวอักษร ปัญหาที่พบคือ ตัวที่ตำแหน่งไม่สามารถจะวนมาบรรจบครบรอบเกิดเป็นกรอบขึ้นมาได้ อันเนื่องมาจากกรอบหรือขอบของตัวอักษรหรือตัวเลขนั้น มีความซับซ้อนและไม่แน่นอนมากนักเอง ทางผู้ทำโครงการจึงได้เลือกใช้กระบวนการ Detect Edge แทนถึงแม้จะเกิดความยุ่งยากมากกว่า เพราะต้องเพิ่มกระบวนการขึ้นมาอีกหนึ่งกระบวนการ แต่ก็ทำให้ได้ผลการตัดแยกเลขรหัสได้ดีกว่า

6.3 ส่วนระบบการวิเคราะห์รูปแบบ

ระบบในส่วนนี้ เห็นได้ชัดจากการทดลองว่าได้ผลที่เป็นที่น่าพอใจในระดับหนึ่ง เนื่องจากในส่วนของการทดลองวิเคราะห์หาค่าตัวเลขจากภาพนั้นโปรแกรมที่เราทำการสร้างขึ้นมาสามารถตอบถูกในทุกตัวเลข 0-9 มีค่าความถูกต้องในการตอบที่เฉลี่ยประมาณ 76.2% โดยการทดลองในขั้นตอนนี้เราได้ทำการสุ่มลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มือเขียนซึ่งเราไม่ได้บังคับในการเขียนลายมือโดยปล่อยให้กลุ่มที่ถูกสุมเขียนกันอย่างอิสระ เขียนบรรจงหรือไม่บรรจงก็ได้แต่ก็ได้รับผลออกมาเป็นที่น่าพอใจ

ในส่วนของ การทดลองขั้นตอนมาเป็นการทดลองให้โปรแกรมตอบจากข้อมูลภาพตัวเลขทั้ง 5 ตัวว่าเมื่อเลขทั้ง 5 ตัวประกอบกันแล้วจะเป็นเลขรหัสไปรษณีย์ของอำเภอใด เราก็ได้รับผลที่น่าพอใจที่ระดับประมาณ 84% โดยในขั้นตอนนี้กลุ่มผู้ถูกสุมจะทำการเขียนด้วยลายมือบรรจงเหมือนที่จะทำการเขียนจริงในซองจดหมายก่อนส่งจดหมายเลขที่เดียว ดังนั้นจึงแน่ใจได้ว่าการเขียนตามที่ใช้ในการทดลองจะเสมือนในการใช้งานจริง

แต่อย่างไรก็ดีในส่วนข้อผิดพลาดของการวิเคราะห์โครงสร้างก็ยังมียังเนื่องจากกระบวนการวิเคราะห์ทางคอมพิวเตอร์ไม่สามารถเป็นไปได้อย่าง 100% ไม่มีใครผู้ใดสามารถทำให้ทุกอย่างสมบูรณ์ไปได้เสียหมด ผู้ทำโครงการก็เช่นเดียวกัน ไม่ว่าจะเป็นปัญหาจากการทำเส้นบางก็ดี หรือการทำ Contour Following ก็ดี อาจจะทำให้เกิดปัญหาในการวิเคราะห์ได้เสมอ นี่เองคือสิ่งที่ผู้ทำโครงการต้องพึงพิจารณาและให้ความสนใจในการวิเคราะห์ที่แก้ไขอยู่ตลอดเวลาเพื่อให้ได้ระบบที่ใกล้ความสมบูรณ์มากที่สุดนั่นเอง

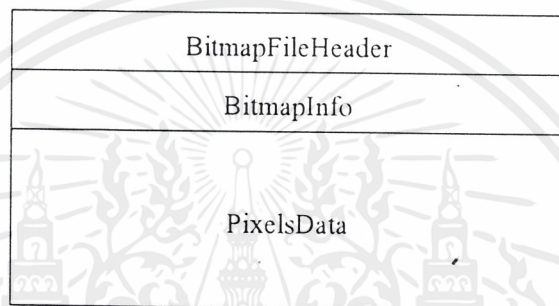


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป

ไฟล์ข้อมูลภาพชนิดบิตแมป มีโครงสร้างดังรูปก-1 โดยแบ่งออกเป็น 3 ส่วน ได้แก่ Bitmap File Header เป็นส่วนที่บอกข้อมูลของไฟล์ Bitmap Information เป็นส่วนที่แสดงขนาดและข้อมูลสีของภาพ และส่วนสุดท้ายคือ Pixel Data เป็นส่วนที่เก็บข้อมูลสีของแต่ละพิกเซล ซึ่งจะมีการเก็บข้อมูลตามขนาดของสี ถ้ามีสีมาก จะทำให้ภาพที่ต้องการเก็บมีขนาดใหญ่ไปด้วย



รูป ก-1 โครงสร้างของ Bitmap File

BitmapFileHeader

Byte	Data	Detail
1-2	File Type	Must be ASCII text "BM"
3-6	Size of File	In double word(32-bit integer)
7-10	Reserved for Future	Must be zero
11-14	Byte offset to Bitmap data	Offset from Bitmap File Header

ตาราง ก-1 แสดงข้อมูลใน Bitmap File Header

BitmapInfo

BitmapInfo ประกอบด้วยสองส่วนคือ BitmapInfoHeader เป็นส่วนที่บอกขนาดและข้อมูลสีของภาพบิตแมป และส่วนของ RGBQuad ซึ่งจะเก็บค่าตารางสีสำหรับเทียบค่าจากค่าของแต่ละพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BitmapInfoHeader

โครงสร้างของ BitmapInfoHeader ประกอบด้วยส่วนต่างๆดังนี้

biSize	จำนวน ไบต์ของ Header File
biWidth, biHeight	บอกขนาดของภาพในความกว้างและความสูงในหน่วยพิกเซล
biPlanes	เก็บเป็น 1 เสมอ
biBitCount	คือจำนวนบิตต่อ 1 พิกเซล
biCompression	แสดงการบีบอัดข้อมูล
ถ้ามีค่า BI_RGB	ไฟล์เป็นแบบ ไม่มีการอัดข้อมูล
RLE4	เป็นการบีบอัดข้อมูลแบบ Run-length Encoder แบบ 4 บิต
RLE8	เป็นการบีบอัดข้อมูลแบบ Run-length Encoder แบบ 8 บิต
BI_BITFIELDS	เป็นการบีบอัดข้อมูลแบบมีตารางสี
biSizelImage	บอกขนาดของไฟล์
biXPelsPerMeter	ความยาวแนวนอนมีหน่วยเป็นพิกเซลต่อเมตร
biYPelsPerMeter	ความยาวแนวตั้งมีหน่วยเป็นพิกเซลต่อเมตร
biClrUsed	เป็นจำนวนสีในตารางสี ที่จะถูกชี้ด้วยค่าพิกเซลในบิตแมป
biClrImportant	เป็นเลขที่แสดงว่าข้อมูลสีมีความสำคัญในการแสดงผลของบิตแมปถ้าเป็น 0 แสดงว่าทุกสีมีความสำคัญ

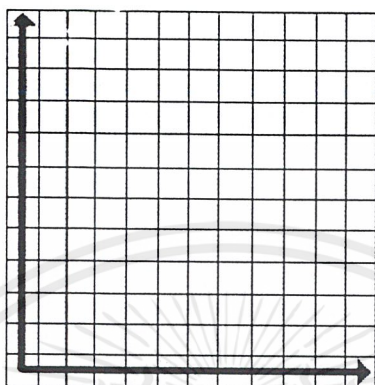
RGBQuad

RGBQuad จะเป็น โครงสร้างที่แสดงความเข้มของสีแดง สีเขียว และสีน้ำเงิน โดยมีความหมายของแต่ละฟิลด์ดังนี้

rgbBlue	แสดงความเข้มของสีน้ำเงิน
rgbGreen	แสดงความเข้มของสีเขียว
rgbRed	แสดงความเข้มของสีแดง
rgbReserved	ต้องมีค่าเป็น 0

Pixel Data

เป็นส่วนที่เก็บข้อมูลสีของแต่ละพิกเซลของภาพ โดยข้อมูลแรกจะเป็นค่าสีของพิกเซลที่อยู่แถวล่างสุดที่ตำแหน่งซ้ายสุด ข้อมูลลำดับต่อไปจะเรียงไปทางขวาจากแถวล่างจนถึงแถวบนสุด



รูปที่ ก-2 แสดงการเก็บข้อมูลของแต่ละพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ตัวโปรแกรม

ในภาคผนวก ข. นี้จะแสดงถึงตัวโปรแกรมที่ทำการเขียนมาทั้งหมด โดยแสดงส่วน Main Form ที่เป็นรูปแบบการจัดวาง Interface กับผู้ใช้ และบรรจุโปรแกรมย่อยทั้งหมด



รูปที่ ข-1 แสดงภาพ Interface ของโปรแกรมที่สร้างขึ้น

ตัวโปรแกรม

Option Explicit

```
Private Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (_
    lpvDest As Any, lpvSource As Any, ByVal cbCopy As Long)
```

```
Private Type SAFEARRAYBOUND
```

```
    cElements As Long
```

```
    lLbound As Long
```

```
End Type
```

```
Private Type SAFEARRAY2D
```

```
    cDims As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fFeatures As Integer
cbElements As Long
cLocks As Long
pvData As Long
Bounds(0 To 1) As SAFEARRAYBOUND
End Type
Private bDib() As Byte, st As Byte
Dim nowxpoint As Integer, nowypoint As Integer
Private X As Integer, Y As Integer, z As Integer, b As Integer
Private a As Byte, d As Integer, e As Integer, newx As Integer, xcenter As Integer, ycenter As Integer
Dim ybottom As Integer, ytop As Integer, xleft As Integer, xright As Integer
Dim idcase1 As Byte, ii As Integer, threepoint(8) As Integer
Dim epx(5) As Integer, epy(5) As Integer, threepoint(8) As Integer
Private Declare Function VarPtrArray Lib "msvbvm50.dll" Alias "VarPtr" (Ptr() As Any) As Long
Dim three As Byte, chd As Byte, four As Byte
Dim removex(10) As Integer, removey(10) As Integer
Dim toomin As Byte, hit As Byte, removal As Byte, maxm As Integer
Dim cccode As Byte

Private Sub Command1_Click()
    Dim cDib As New cDIBSection
    Dim cDibBuffer As New cDIBSection
    Dim tSA As SAFEARRAY2D
    Dim i As Byte
    Dim strfind As String
    For i = 1 To 5
        hit = hit + 1
        If hit = 6 Then
            hit = 1
        End If
        maxm = (hit * 40) - 20

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Debug.Print " "
Debug.Print "(""; hit; ")")"
a = 0
cDib.CreateFromPicture PicHidden
With tSA
    .cbElements = 1
    .cDims = 2
    .Bounds(0).lLbound = 0
    .Bounds(0).cElements = cDib.Height
    .Bounds(1).lLbound = 0
    .Bounds(1).cElements = cDib.BytesPerScanLine
    .pvData = cDib.DIBSectionBitsPtr
End With
CopyMemory ByVal VarPtrArray(bDib()), VarPtr(tSA), 4
Call Findedge
Call RemoveNoise
Call NewFindpoint(xcenter - 20, ycenter - 30)
Call Thinning
Call Case1
Call Draw
txt2.Text = txt2.Text & "{" & hit - 1 & "}" & vbCrLf
If idcase1 = 1 Then
    txt2.Text = txt2.Text + "มี 1 จุดปลาย" & vbCrLf
    Call Case12
ElseIf idcase1 = 2 Then
    txt2.Text = txt2.Text + "มี 2 จุดปลาย" & vbCrLf
    Call Case21
ElseIf idcase1 = 3 Then
    txt2.Text = txt2.Text + "มี 3 จุดปลาย" & vbCrLf
    Call Case31
ElseIf idcase1 = 4 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

txt2.Text = txt2.Text + "มี 4 จุดปลาย" & vbCrLf
Call Case41
ElseIf idcase1 = 0 Then
txt2.Text = txt2.Text + "ไม่มีจุดปลาย" & vbCrLf
Call Case01
Else
txt2.Text = txt2.Text + "มีมากกว่า 4 จุดปลาย" & vbCrLf
Label1.Caption = Label1.Caption + "X"
End If
Next i

On Error GoTo errfind
strfind = "postcode=" + Label1.Caption
datasearch.Recordset.FindLast strfind

Exit Sub
errfind:
MsgBox Err.Description, vbOKOnly + vbExclamation, "เงื่อนไขผิดพลาด"

End Sub

Sub Draw()
Dim m As Integer, n As Integer, o As Integer, p As Integer
Dim i As Byte, j As Byte
i = maxm: j = 0
For n = ytop To ybottom Step -1
For m = xleft To xright
i = i + 1
If bDib(m * 3, n) = 0 Then
For p = (j * 2) To (j * 2) + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        For o = i * 2 To (i * 2) + 1
            Pic2.PSet (o, p)
        Next o
    Next p
End If
Next m
i = maxm
j = j + 1
Next n

End Sub

Function ChkLeft(px As Integer, py As Integer) As Byte
    ChkLeft = 0
    If (bDib((px - 1) * 3, py + 1) = 0) Or (bDib((px - 1) * 3, py) = 0) Or (bDib((px - 1) * 3, py - 1) = 0)
Then
        ChkLeft = 1
    End If
End Function

Function ChkRight(px As Integer, py As Integer) As Byte
    ChkRight = 0
    If (bDib((px + 1) * 3, py + 1) = 0) Or (bDib((px + 1) * 3, py) = 0) Or (bDib((px + 1) * 3, py - 1) = 0)
Then
        ChkRight = 1
    End If
End Function

Function ChkUp(px As Integer, py As Integer) As Byte
    ChkUp = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If (bDib((px - 1) * 3, py + 1) = 0) Or (bDib(px * 3, py + 1) = 0) Or (bDib((px + 1) * 3, py + 1) = 0)
```

```
Then
```

```
    ChkUp = 1
```

```
End If
```

```
End Function
```

```
Function ChkDown(px As Integer, py As Integer) As Byte
```

```
    ChkDown = 0
```

```
    If (bDib((px - 1) * 3, py - 1) = 0) Or (bDib(px * 3, py - 1) = 0) Or (bDib((px + 1) * 3, py - 1) = 0)
```

```
Then
```

```
    ChkDown = 1
```

```
End If
```

```
End Function
```

```
Function FindDirection(px As Integer, py As Integer, se As Byte) As Byte
```

```
    Dim LD As Byte, m As Byte, n As Byte, OldLD As Byte, i As Byte
```

```
    Dim vleft As Integer, vup As Integer, vdown As Integer, vright As Integer
```

```
    Dim j As Integer, ccode As Byte
```

```
    m = 0: LD = 0: vleft = 0: vup = 0: vdown = 0: vright = 0: OldLD = 0
```

```
    chd = 0
```

```
    nowxpoint = px
```

```
    nowypoint = py
```

```
    Select Case se
```

```
        Case 0
```

```
            n = 5
```

```
        Case 1
```

```
            n = (ytop - ybottom) - 5
```

```
        Case 2
```

```
            n = 5
```

```
    End Select
```

```
    While (LD = 0) Or (m <> n)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OldLD = LD
If se = 2 Then
  removex(m) = nowxpoint
  removey(m) = nowypoint
  If m = 0 Then
    ccode = ccode
    If (ccode = 1) Or (ccode = 2) Or (ccode = 3) Then
      LD = 4
      GoTo L4
    ElseIf (ccode = 7) Or (ccode = 8) Or (ccode = 9) Then
      LD = 2
      GoTo L2
    ElseIf ccode = 4 Then
      LD = 1
      GoTo L1
    ElseIf ccode = 6 Then
      LD = 3
      GoTo L3
    End If
  End If
End If
If (ChkLeft(nowxpoint, nowypoint) = 1) And (ccode <> 3) And (ccode <> 9) And (LD <> 3) Then
L1:  ccode = 1
    For j = nowypoint - 1 To nowypoint + 1
      If bDib((nowxpoint - 1) * 3, j) = 0 Then
        nowxpoint = nowxpoint - 1
        nowypoint = j
      Exit For
    End If
    ccode = ccode + 3
    If (ccode = 10) Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    GoTo L5
  End If
Next j
LD = 1
vleft = vleft + 1
ElseIf (ChkUp(nowxpoint, nowypoint) = 1) And (ccode <> 1) And (ccode <> 3) And (LD <> 4) Then
L2:  ccode = 7
    For j = nowxpoint - 1 To nowxpoint + 1
      If bDib(j * 3, nowypoint + 1) = 0 Then
        nowxpoint = j
        nowypoint = nowypoint + 1
      Exit For
    End If
    ccode = ccode + 1
    If ccode = 10 Then
      If ccode = 7 Then
        GoTo L1
      ElseIf ccode = 9 Then
        GoTo L3
      End If
      GoTo L5
    End If
  Next j
  LD = 2
  vup = vup + 1
  ElseIf (ChkRight(nowxpoint, nowypoint) = 1) And (ccode <> 1) And (ccode <> 7) And (LD <> 1)
Then
L3:  ccode = 3
    For j = nowypoint - 1 To nowypoint + 1
      If bDib((nowxpoint + 1) * 3, j) = 0 Then
        nowxpoint = nowxpoint + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    nowypoint = j
    Exit For
End If
ccode = ccode + 3
If ccode = 12 Then
    GoTo L5
End If
Next j
LD = 3
vright = vright + 1
ElseIf (ChkDown(nowxpoint, nowypoint) = 1) And (ccode <> 7) And (ccode <> 9) And (LD <> 2)
Then
L4: ccode = 1
For j = nowxpoint - 1 To nowxpoint + 1
    If bDib(j * 3, nowypoint - 1) = 0 Then
        nowxpoint = j
        nowypoint = nowypoint - 1
        Exit For
    End If
    ccode = ccode + 1
    If ccode = 4 Then
        If ccode = 1 Then
            GoTo L1
        ElseIf ccode = 3 Then
            GoTo L3
        End If
        GoTo L5
    End If
Next j
LD = 4
vdown = vdown + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Else

```

L5:  If se = 2 Then
      For i = 0 To 5
          bDib((removex(i) * 3), removey(i)) = 255
      Next i
      Debug.Print removex(0), removey(0)
      removeal = 1
      Debug.Print "Remove Noise already"
      m = n - 1
  End If
End If
m = m + 1
If se <> 2 Then
  Select Case ccode
    Case 1
      If vdown <> 0 Then
        If vleft > 0 Then
          vleft = vleft - 1
        End If
        vdown = vdown + 1
        LD = 4
      End If
    Case 3
      If vdown <> 0 Then
        If vright > 0 Then
          vright = vright - 1
        End If
        vdown = vdown + 1
        LD = 4
      End If
    Case 7

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If vup <> 0 Then
  If vleft > 0 Then
    vleft = vleft - 1
  End If
  vup = vup + 1
  LD = 2
End If

```

Case 9

```

If vright <> 0 Then
  If vup > 0 Then
    vup = vup - 1
  End If
  vright = vright + 1
  LD = 3
End If

```

End Select

End If

```

If (OldLD <> LD) And (se = 1) Then

```

```

  chd = chd + 1

```

```

End If

```

Wend

```

If se = 2 Then

```

```

  For i = 0 To 5

```

```

    removex(i) = 0

```

```

    removey(i) = 0

```

```

  Next i

```

```

End If

```

```

If se = 1 Then

```

```

  'txt2.Text = txt2.Text + "left=" & vleft & vbCrLf

```

```

  'txt2.Text = txt2.Text + "up=" & vup & vbCrLf

```

```

  'txt2.Text = txt2.Text + "right=" & vright & vbCrLf

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'txt2.Text = txt2.Text + "down=" & vdown & vbCrLf
```

```
End If
```

```
If (se = 1) And (vleft > 3) Then
```

```
    FindDirection = 1
```

```
Else
```

```
If ((vleft - vup) >= 0) And ((vleft - vright) >= 0) And ((vleft - vdown) >= 0) Then
```

```
    FindDirection = 1
```

```
ElseIf ((vup - vleft) >= 0) And ((vup - vright) >= 0) And ((vup - vdown) >= 0) Then
```

```
    FindDirection = 2
```

```
ElseIf ((vright - vleft) >= 0) And ((vright - vup) >= 0) And ((vright - vdown) >= 0) Then
```

```
    FindDirection = 3
```

```
ElseIf ((vdown - vleft) >= 0) And ((vdown - vright) >= 0) And ((vdown - vup) >= 0) Then
```

```
    FindDirection = 4
```

```
End If
```

```
End If
```

```
End Function
```

```
Sub Case01()
```

```
    If (three <> 0) Or (four <> 0) Then
```

```
        Label1.Caption = Label1.Caption + "8"
```

```
    Else
```

```
        Label1.Caption = Label1.Caption + "0"
```

```
    End If
```

```
End Sub
```

```
Sub Case12()
```

```
    Dim ycenter As Integer, xcenter As Integer
```

```
    Dim numleft As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim m As Integer, n As Byte
```

```
Dim botdi As Byte
```

```
ycenter = (ytop + ybottom) / 2
```

```
xcenter = (xright + xleft) / 2
```

```
If epy(1) > ycenter Then
```

```
    n = 0
```

```
    For m = xleft To xright
```

```
        If (bDib(m * 3, epy(1) - 5) = 0) And (bDib((m + 1) * 3, epy(1) - 5) = 255) Then
```

```
            n = n + 1
```

```
        End If
```

```
    Next m
```

```
    If n = 1 Then
```

```
        Label1.Caption = Label1.Caption + "6"
```

```
        txt2.Text = txt2.Text + "ล่างจุดบนสุด มี 1 เส้น" & vbCrLf
```

```
    ElseIf (n = 2) Then
```

```
        Label1.Caption = Label1.Caption + "8"
```

```
        txt2.Text = txt2.Text + "ล่างจุดบนสุด มี 2 เส้น" & vbCrLf
```

```
    Else
```

```
        Label1.Caption = Label1.Caption + "X"
```

```
    End If
```

```
ElseIf epy(1) < ycenter Then
```

```
    botdi = FindDirection(epx(1), epy(1), 0)
```

```
    If botdi = 1 Then
```

```
        Label1.Caption = Label1.Caption + "2"
```

```
        txt2.Text = txt2.Text + "จุดปลายล่างอยู่ครึ่งล่างและวังซ้าย" & vbCrLf
```

```
    ElseIf (botdi = 2) Or (botdi = 3) Then
```

```
        txt2.Text = txt2.Text + "จุดปลายล่างอยู่ครึ่งล่างและวังบนหรือขวา" & vbCrLf
```

```
    If threepoint(1) < (ytop + ybottom) / 2 Then
```

```
        Label1.Caption = Label1.Caption + "4"
```

```
        txt2.Text = txt2.Text + "จุดแยกอยู่ต่ำกว่ากึ่งกลาง" & vbCrLf
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    Label1.Caption = Label1.Caption + "9"
    txt2.Text = txt2.Text + "จุดแยกอยู่สูงกว่ากึ่งกลาง" & vbCrLf
End If
ElseIf botdi = 4 Then
    Label1.Caption = Label1.Caption + "9"
    txt2.Text = txt2.Text + "จุดปลายล่างวิ้งล่าง" & vbCrLf
End If
End If
End Sub

Sub Case21()
    Dim botdi As Byte, topdi As Byte
    topdi = 0: botdi = 0
    botdi = FindDirection(epx(1), epy(1), 0)
    If ((ytop - epy(2)) < 10) And ((ytop - epy(1)) < 10) Or (epy(1) > ((ytop + ybottom) / 2)) Then
        txt2.Text = txt2.Text + "สองจุดปลายบนน้อยกว่าจุดสูงสุดไม่เกิน 10 และอยู่เกินครึ่ง" & vbCrLf
        If (four <> 0) Or (three <> 0) Then
            Label1.Caption = Label1.Caption + "8"
            txt2.Text = txt2.Text + "มีจุดแยกสี่หรือแยกสาม" & vbCrLf
        Else
            Label1.Caption = Label1.Caption + "0"
            txt2.Text = txt2.Text + "ไม่มีจุดแยกสี่หรือแยกสาม" & vbCrLf
        End If
    ElseIf botdi = 1 Then
        Label1.Caption = Label1.Caption + "2"
        txt2.Text = txt2.Text + "จุดปลายล่างวิ้งซ้าย" & vbCrLf
    ElseIf botdi = 2 Then
        txt2.Text = txt2.Text + "จุดปลายล่างวิ้งบน" & vbCrLf
        If three <> 0 Then
            txt2.Text = txt2.Text + "มีจุดแยกสาม" & vbCrLf
        End If
    End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

topdi = FindDirection(epx(2), epy(2), 0)
If (topdi = 1) Or ((ytop - epy(2)) < 5) Then
    Label1.Caption = Label1.Caption + "4"
    txt2.Text = txt2.Text + "จุดปลายบนวิ้งซ้ายหรือจุดปลายบนน้อยกว่าจุดสูงสุดไม่เกิน 5" &
vbCrLf
Else
    Label1.Caption = Label1.Caption + "3"
    txt2.Text = txt2.Text + "จุดปลายบนไม่วิ้งซ้ายหรือจุดปลายบนไม่น้อยกว่าจุดสูงสุดไม่เกิน 5" &
vbCrLf
End If

Else
    txt2.Text = txt2.Text + "ไม่มีจุดแยกสาม" & vbCrLf
    topdi = FindDirection(epx(2), epy(2), 0)
    If topdi = 4 Then
        If (xright - xleft) <= ((ytop - ybottom) / 3) + 3 Then
            Label1.Caption = Label1.Caption + "1"
            txt2.Text = txt2.Text + "ความกว้างน้อยกว่า 1/3 ของความสูง" & vbCrLf
        Else
            Label1.Caption = Label1.Caption + "7"
            txt2.Text = txt2.Text + "ความกว้างมากกว่า 1/3 ของความสูง" & vbCrLf
        End If
    ElseIf topdi = 1 Then
        txt2.Text = txt2.Text + "จุดปลายบนวิ้งซ้าย" & vbCrLf
        If epx(2) > (xleft + xright) / 2 Then
            txt2.Text = txt2.Text + "จุดปลายบนอยู่ครึ่งขวา" & vbCrLf
            If epy(2) < (ytop - 5) Then
                Label1.Caption = Label1.Caption + "7"
                txt2.Text = txt2.Text + "จุดปลายบนอยู่ต่ำกว่าจุดสูงสุด-5" & vbCrLf
            Else
                If Abs(epx(1) - epx(2)) <= 7 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label1.Caption = Label1.Caption + "9"
txt2.Text = txt2.Text + "จุดปลายบนและล่างหลักห่างกันไม่เกินครึ่งของความกว้าง" &
vbCrLf

ElseIf Abs(epx(1) - epx(2)) > 7 Then
Label1.Caption = Label1.Caption + "5"
txt2.Text = txt2.Text + "จุดปลายบนและล่างหลักห่างกันเกินครึ่งของความกว้าง" &
vbCrLf

End If
End If

ElseIf epx(2) < (xleft + xright) / 2 Then
txt2.Text = txt2.Text + "จุดปลายบนอยู่ครึ่งซ้าย" & vbCrLf
topdi = FindDirection(epx(1), epy(1), 1)
If topdi = 2 Then
Label1.Caption = Label1.Caption + "7"
txt2.Text = txt2.Text + "ระหว่างจุดปลายล่างชั้นบนมีทิศบน" & vbCrLf
Else
Label1.Caption = Label1.Caption + "3"
txt2.Text = txt2.Text + "ระหว่างจุดปลายล่างชั้นบนมีทิศซ้าย" & vbCrLf
End If
End If

ElseIf (topdi = 3) Or (topdi = 2) Then
txt2.Text = txt2.Text + "จุดปลายบนวิ่งบนหรือขวา" & vbCrLf
topdi = FindDirection(epx(1), epy(1), 1)
If topdi = 2 Then
txt2.Text = txt2.Text + "ระหว่างจุดปลายล่างชั้นบนมีทิศบน" & vbCrLf
If Abs(xright - xleft) <= ((ytop - ybottom) / 3) + 3 Then    'If (xright - xleft) < 10 Then
Label1.Caption = Label1.Caption + "1"
txt2.Text = txt2.Text + "ความกว้างน้อยกว่า 1/3 ของความสูง" & vbCrLf
Else
Label1.Caption = Label1.Caption + "7"
txt2.Text = txt2.Text + "ความกว้างมากกว่า 1/3 ของความสูง" & vbCrLf

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

ElseIf topdi = 1 Then

txt2.Text = txt2.Text + "ระหว่างจุดปลายล่างขึ้นบนมีทิศซ้าย" & vbCrLf

If epx(2) > (xleft + xright) / 2 Then

Label1.Caption = Label1.Caption + "9"

txt2.Text = txt2.Text + "จุดปลายบนอยู่ครึ่งขวา" & vbCrLf

Else

Label1.Caption = Label1.Caption + "3"

txt2.Text = txt2.Text + "จุดปลายบนอยู่ครึ่งซ้าย" & vbCrLf

End If

End If

End If

End If

ElseIf (botdi = 3) Or (botdi = 4) Then

txt2.Text = txt2.Text + "จุดปลายล่างวิ่งขวาหรือล่าง" & vbCrLf

topdi = FindDirection(epx(2), epy(2), 0)

If topdi = 1 Then

txt2.Text = txt2.Text + "จุดปลายบนวิ่งซ้าย" & vbCrLf

If epy(2) < (ytop - 5) Then

Label1.Caption = Label1.Caption + "3"

txt2.Text = txt2.Text + "จุดปลายบนอยู่ต่ำกว่าจุดสูงสุดเกิน 5" & vbCrLf

ElseIf (Abs(epx(1) - epx(2)) > 7) Or (epx(2) > (xright - 5)) Then

Label1.Caption = Label1.Caption + "5"

txt2.Text = txt2.Text + "จุดปลายบนและล่างหลักห่างกันเกินครึ่งของความกว้าง" & vbCrLf

End If

ElseIf (topdi = 3) Or (topdi = 2) Then

txt2.Text = txt2.Text + "จุดปลายบนวิ่งบนหรือขวา" & vbCrLf

If epy(2) <= (ytop - 5) Then

Label1.Caption = Label1.Caption + "3"

txt2.Text = txt2.Text + "จุดปลายบนอยู่ต่ำกว่าจุดสูงสุดเกิน 5" & vbCrLf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'ElseIf Abs(epx(1) - epx(2)) > 7 Then
'   Labell.Caption = Labell.Caption + "5"
'   txt2.Text = txt2.Text + "จุดปลายบนและล่างหลักห่างกันเกินครึ่งของความกว้าง" & vbCrLf
'End If
End If
End If

```

```
End Sub
```

```
Sub Case31()
```

```

Dim topdi As Byte, topdi2 As Byte
topdi = FindDirection(epx(3), epy(3), 0)
If (topdi = 4) Or (topdi = 1) Then
txt2.Text = txt2.Text + "จุดปลายบนข้างล่างหรือซ้าย" & vbCrLf
If (Abs(epy(2) - epy(1)) < 5) And (epy(2) < ybottom + ((ytop - ybottom) / 3)) Then
Label1.Caption = Label1.Caption + "1"
txt2.Text = txt2.Text + "จุดปลายล่างกับกลาง สูงห่างกันไม่เกิน 5" & vbCrLf
Else
topdi2 = FindDirection(epx(2), epy(2), 0)
If (topdi2 = 4) Or (topdi2 = 1) Then
Label1.Caption = Label1.Caption + "4"
txt2.Text = txt2.Text + "จุดปลายกลางข้างล่างหรือซ้าย" & vbCrLf
ElseIf (topdi2 = 2) Or (topdi2 = 3) Then
txt2.Text = txt2.Text + "จุดปลายกลางข้างบนหรือขวา" & vbCrLf
If Abs(xright - xleft) < ((ytop - ybottom) / 3) Then
Label1.Caption = Label1.Caption + "1"
txt2.Text = txt2.Text + "ความกว้างน้อยกว่า 1/3 ของความสูง" & vbCrLf
Else
Label1.Caption = Label1.Caption + "7"
txt2.Text = txt2.Text + "ความกว้างมากกว่า 1/3 ของความสูง" & vbCrLf

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    End If
    End If
    End If
ElseIf (topdi = 3) Or (topdi = 2) Then
    topdi2 = FindDirection(epx(2), epy(2), 0)
    txt2.Text = txt2.Text + "จุดปลายบนวิ่งบนหรือขวา" & vbCrLf
    If topdi2 = 4 Then
        Label1.Caption = Label1.Caption + "9"
        txt2.Text = txt2.Text + "จุดปลายกลางมีทิศต่าง" & vbCrLf
    ElseIf (Abs(epy(2) - epy(1)) < 5) And (epy(2) < ybottom + ((ytop - ybottom) / 3)) Then
        Label1.Caption = Label1.Caption + "1"
        txt2.Text = txt2.Text + "จุดปลายล่างกับกลาง สูงห่างกันไม่เกิน 5" & vbCrLf
    ElseIf (ytop - epy(2)) < (ytop - ybottom) / 3 Then
        Label1.Caption = Label1.Caption + "7"
        txt2.Text = txt2.Text + "จุดปลายบนกับสูงสุด สูงน้อยกว่า 1/3 ของความสูง" & vbCrLf
    Else
        Label1.Caption = Label1.Caption + "3"
    End If
End If
End If

End Sub

```

```
Sub Case41()
```

```
Dim topdi As Byte
```

```
If three <> 0 Then
```

```
txt2.Text = txt2.Text + "มีจุดแยกสาม" & vbCrLf
```

```
topdi = FindDirection(epx(4), epy(4), 0)
```

```
If (topdi = 3) Or (topdi = 2) Then
```

```
Label1.Caption = Label1.Caption + "7"
```

```
txt2.Text = txt2.Text + "จุดปลายบนวิ่งขวาหรือบน" & vbCrLf
```

```
Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label1.Caption = Label1.Caption + "4"
txt2.Text = txt2.Text + "จุดปลายบนไม่วิ่งซ้ายหรือล่าง" & vbCrLf
End If
Else
Label1.Caption = Label1.Caption + "X"
End If
End Sub

```

```

Sub RemoveNoise()
Dim m As Integer, n As Integer, p As Integer
For Y = ybottom To ytop
For X = xleft To xright
p = 0
For m = X - 2 To X + 2
For n = Y - 2 To Y + 2
If bDib(m * 3, n) = 0 Then
p = p + 1
End If
Next n
Next m
If p <= 2 Then
For m = X - 2 To X + 2
For n = Y - 2 To Y + 2
bDib(m * 3, Y) = 255
Next n
Next m
End If
Next X
Next Y
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function ChkSp(m As Integer, n As Integer, o As Integer, p As Integer) As Integer

Dim SP As Byte

SP = 0

If bDib(m * 3, n) = 255 Then

 If bDib(o * 3, p) = 0 Then

 SP = 1

 End If

End If

ChkSp = SP

End Function

Sub Thinning()

Dim m As Integer, n As Integer, Np As Byte, SP As Byte

Dim xf(300) As Integer, yf(300) As Integer

Dim flag As Integer, step As Boolean

step = True

Do

flag = 0

For Y = ybottom To ytop

 For X = xleft To xright

 If bDib(X * 3, Y) = 0 Then

 Np = 0: SP = 0

 For m = X - 1 To X + 1

 For n = Y - 1 To Y + 1

 If bDib(m * 3, n) = 0 Then

 Np = Np + 1

 End If

 Next n

 Next m

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SP = FindSP
If (Np >= 3) And (Np <= 7) And (SP = 1) Then
  If step = True Then
    If (bDib(X * 3, Y + 1) = 255) Or (bDib((X + 1) * 3, Y) = 255) Or (bDib(X * 3, Y - 1)
= 255) Then
      If (bDib((X - 1) * 3, Y) = 255) Or (bDib((X + 1) * 3, Y) = 255) Or (bDib(X * 3, Y -
1) = 255) Then
        flag = flag + 1
        xf(flag) = X
        yf(flag) = Y
      End If
    End If
  ElseIf step = False Then
    If (bDib(X * 3, Y + 1) = 255) Or (bDib((X + 1) * 3, Y) = 255) Or (bDib((X - 1) * 3,
Y) = 255) Then
      If (bDib((X - 1) * 3, Y) = 255) Or (bDib(X * 3, Y + 1) = 255) Or (bDib(X * 3, Y -
1) = 255) Then
        flag = flag + 1
        xf(flag) = X
        yf(flag) = Y
      End If
    End If
  End If
End If

End If
Next X
Next Y
For n = 1 To flag
  bDib(xf(n) * 3, yf(n)) = 255
Next n

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

step = Not step

Loop Until (flag = 0)

End Sub

Function FindSP() As Byte

Dim SP As Byte

SP = 0

SP = ChkSp(X, Y + 1, X + 1, Y + 1) + SP

SP = ChkSp(X + 1, Y + 1, X + 1, Y) + SP

SP = ChkSp(X + 1, Y, X + 1, Y - 1) + SP

SP = ChkSp(X + 1, Y - 1, X, Y - 1) + SP

SP = ChkSp(X, Y - 1, X - 1, Y - 1) + SP

SP = ChkSp(X - 1, Y - 1, X - 1, Y) + SP

SP = ChkSp(X - 1, Y, X - 1, Y + 1) + SP

SP = ChkSp(X - 1, Y + 1, X, Y + 1) + SP

FindSP = SP

End Function

Sub Case1()

Dim m As Integer, n As Integer, Np As Byte

Dim EndPoint As Byte, SP As Byte

Dim numleft As Integer, numright As Integer

Dim line As Integer, remove As Byte

three = 0: four = 0

For Y = ybottom - 5 To ytop + 5

For X = xleft - 5 To xright + 5

Np = 0: removeal = 0: cccode = 0

If bDib(X * 3, Y) = 0 Then

For m = X - 1 To X + 1

For n = Y - 1 To Y + 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    If bDib(m * 3, n) = 0 Then
        Np = Np + 1
    End If
Next n
Next m
End If
If Np >= 4 Then
    SP = FindSP
    If (SP = 3) Or (SP = 4) Then
        cccode = 0
        For n = Y - 1 To Y + 1
            If removal = 1 Then
                Exit For
            End If
            For m = X - 1 To X + 1
                cccode = cccode + 1
                If (bDib(m * 3, n) = 0) Then
                    If ((m = X) And (n = Y)) = False Then
                        remove = FindDirection(m, n, 2)
                    End If
                    If removal = 1 Then
                        Exit For
                    End If
                End If
            End If
        Next m
    Next n
End If
ElseIf Np = 2 Then
    remove = FindDirection(X, Y, 2)
End If
Next X

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next Y
  For Y = ybottom - 5 To ytop + 5
    For X = xleft - 5 To xright + 5
      Np = 0
      If (bDib(X * 3, Y) = 0) And (bDib((X + 1) * 3, Y) = 0) And (bDib(X * 3, Y + 1) = 0) And
(bDib((X + 1) * 3, Y + 1) = 0) Then
        four = 1
      End If
      If bDib(X * 3, Y) = 0 Then
        For m = X - 1 To X + 1
          For n = Y - 1 To Y + 1
            If bDib(m * 3, n) = 0 Then
              Np = Np + 1
            End If
          Next n
        Next m
      End If
      If Np >= 4 Then
        SP = FindSP
        If (SP = 3) Or (SP = 4) Then
          three = three + 1
          threepoint(three) = X
          threepoint(three) = Y
        End If
      ElseIf Np = 2 Then
        EndPoint = EndPoint + 1
        epX(EndPoint) = X
        epy(EndPoint) = Y
        Debug.Print "endpoint"; EndPoint; " = "; X, 899 - Y
      End If
    Next X
  Next Y

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Next Y
```

```
idcase1 = EndPoint
```

```
Debug.Print "3 ๓๕๓ = "; three
```

```
End Sub
```

```
Sub Findedge()
```

```
For X = newx To 1000
```

```
  If a = 1 Then
```

```
    Exit For
```

```
  End If
```

```
    For Y = 140 To 199
```

```
      If (bDib(X * 3, Y) = 0) And (bDib(X * 3, Y - 1) = 255) And (bDib(X * 3, Y + 1) = 255) And (bDib((X - 1) * 3, Y) = 255) And (bDib((X - 1) * 3, Y - 1) = 255) And (bDib((X - 1) * 3, Y + 1) = 255) And (bDib((X + 1) * 3, Y) = 255) And (bDib((X + 1) * 3, Y - 1) = 255) And (bDib((X + 1) * 3, Y + 1) = 255)
```

```
    Then
```

```
      bDib(X * 3, Y) = 255
```

```
      Debug.Print "noise"; X, Y
```

```
    Exit For
```

```
  End If
```

```
  If bDib(X * 3, Y) = 0 Then
```

```
    Call Contour(1, 0)
```

```
    If toomin = 0 Then
```

```
      Y = 0
```

```
    Exit For
```

```
  End If
```

```
End If
```

```
Next Y
```

```
Next X
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Sub NewFindpoint(addx As Integer, addy As Integer)

a = 0

For Y = addy To addy + 60

 If a = 1 Then

 Exit For

 End If

For X = addx To addx + 40

 If (bDib(X * 3, Y) = 0) And (bDib(X * 3, Y - 1) = 255) And (bDib(X * 3, Y + 1) = 255) And
 (bDib((X - 1) * 3, Y) = 255) And (bDib((X - 1) * 3, Y - 1) = 255) And (bDib((X - 1) * 3, Y + 1) = 255)
 And (bDib((X + 1) * 3, Y) = 255) And (bDib((X + 1) * 3, Y - 1) = 255) And (bDib((X + 1) * 3, Y + 1) =
 255) Then

 bDib(X * 3, Y) = 255

 Debug.Print "noise"; X, Y

 Exit For

 End If

If bDib(X * 3, Y) = 0 Then

 Call Contour(0, 0)

 If toomin = 0 Then

 Y = 0

 Exit For

 End If

End If

Next X

Next Y

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sub Contour(se As Byte, color As Byte)

Dim x0 As Integer, y0 As Integer

Dim yold As Integer, xold As Integer

ii = 0: a = 0: x0 = X: y0 = Y: d = 0

toomin = 0: ybottom = 1000: ytop = 0: xleft = 1000: xright = 0

While (X <> x0) Or (Y <> y0) Or (a = 0)

ii = ii + 1

a = 1

yold = Y

If (ybottom > yold) And (bDib(X * 3, Y) = 0) Then

ybottom = yold

ElseIf (ytop < yold) And (bDib(X * 3, Y) = 0) Then

ytop = yold

End If

xold = X

If (xleft > xold) And (bDib(X * 3, Y) = 0) Then

xleft = xold

ElseIf (xright < xold) And (bDib(X * 3, Y) = 0) Then

xright = xold

End If

If bDib(X * 3, Y) = 0 Then

Call LLeft(X, Y, d)

Else

Call RRight(X, Y, d)

End If

Wend

If ii < 50 Then

a = 0

toomin = 1

End If

If se = 1 Then

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
xcenter = (xright + xleft) / 2
```

```
ycenter = (ytop + ybottom) / 2
```

```
newx = xright + 5
```

```
End If
```

```
End Sub
```

```
Sub LLeft(m As Integer, n As Integer, o As Integer)
```

```
  Select Case o
```

```
    Case 0
```

```
      Y = n + 1
```

```
    Case 1
```

```
      X = m - 1
```

```
    Case 2
```

```
      Y = n - 1
```

```
    Case 3
```

```
      X = m + 1
```

```
  End Select
```

```
  d = (o + 1) Mod 4
```

```
End Sub
```

```
Sub RRight(m As Integer, n As Integer, o As Integer)
```

```
  Select Case o
```

```
    Case 0
```

```
      Y = Y - 1
```

```
    Case 1
```

```
      X = X + 1
```

```
    Case 2
```

```
      Y = Y + 1
```

```
    Case 3
```

```
      X = X - 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Select
```

```
d = (o + 3) Mod 4
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Dim strfind As String
```

```
strfind = "postcode=" + Label1.Caption
```

```
datsearch.Recordset.FindPrevious strfind
```

```
End Sub
```

```
Private Sub mnuexit_Click()
```

```
End
```

```
End Sub
```

```
Private Sub mnuopen_Click()
```

```
hit = 0
```

```
Pic2.Cls
```

```
newx = 500
```

```
txt2.Text = ""
```

```
Label1.Caption = ""
```

```
cdlsample.Flags = cdIOFNHideReadOnly
```

```
cdlsample.ShowOpen
```

```
Set PicHidden = LoadPicture(cdlsample.FileName)
```

```
pic1.PaintPicture PicHidden.Picture, 0, 0, 489, 329, 0, 0, PicHidden.ScaleWidth, _
```

```
PicHidden.ScaleHeight, vbSrcCopy
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

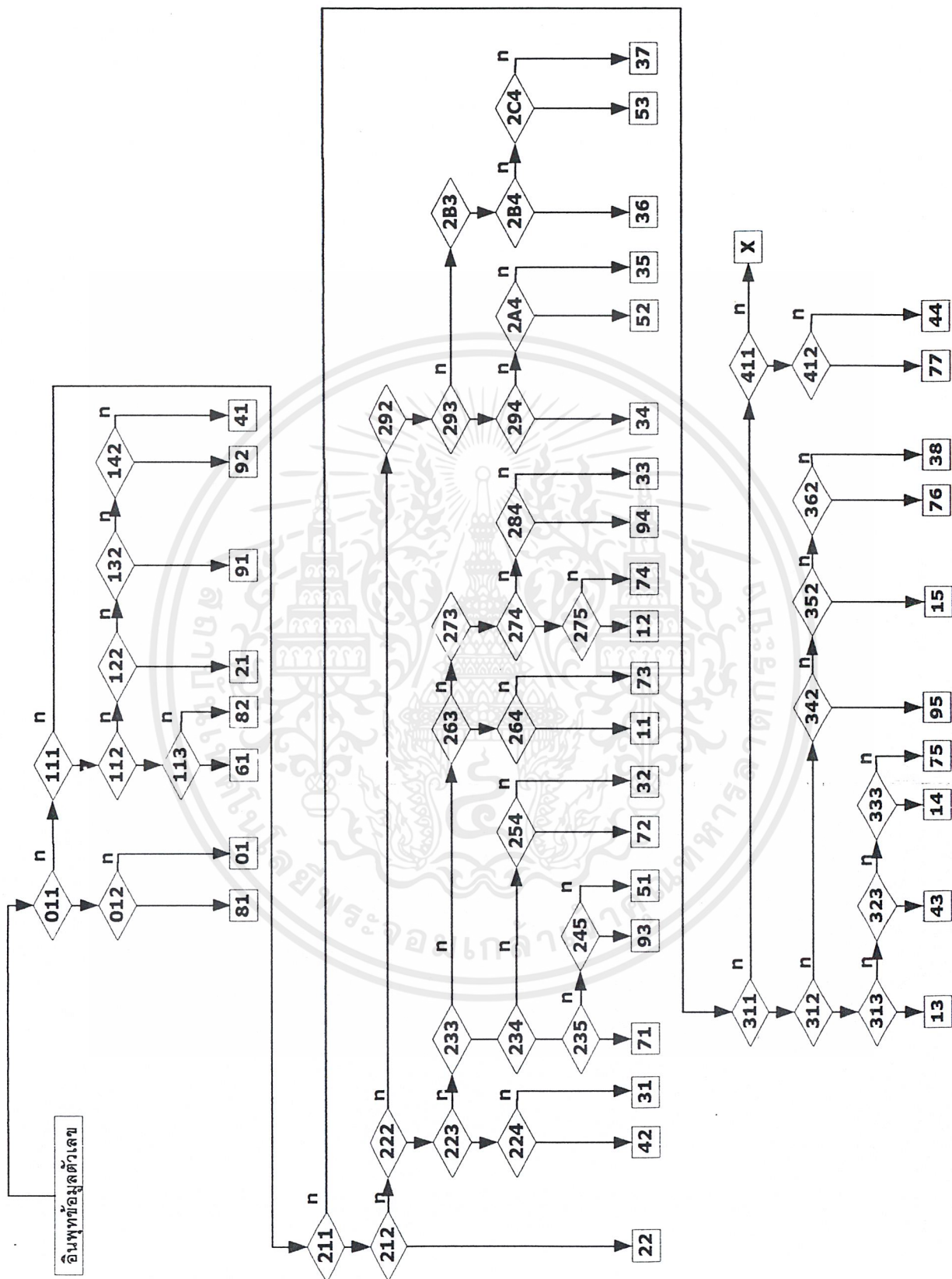
ภาคผนวก ก.

รูปแบบตัวเลขที่เก็บข้อมูลได้ และ Flow Chart รวม

ภาคผนวก ก. แสดงถึงรูปแบบตัวเลขที่เราเก็บข้อมูลได้ทั้งหมด และ Flow Chart โดยรวม โดยในส่วนของรูปแบบตัวเลขจะบอกแทนด้วยเลขรหัสที่แสดงใน Flow Chart โดยรวม อย่างเช่นรูปแบบ 81 คือรูปแบบของเลข 8 แบบที่ 1 หรือ รูปแบบ 15 ก็คือรูปแบบของเลข 1 แบบที่ 5 เป็นต้น เลขรหัสเหล่านี้แสดงใน flow Chart โดยรวมในขั้นสุดท้ายที่เป็นบล็อกรหัสของคำตอบในระบบ

8	8	0	0	0	6	2	2	2	2	2
81	82	01	01	01	61	21	21	22	22	22
9	9	9	9	9	4	4	4	4	4	4
91	92	93	94	95	41	42	42	43	44	44
5	5	5	7	7	7	7	7	7	7	7
51	52	53	71	72	73	74	75	76	76	76
3	3	3	3	3	3	3	3	1	1	1
31	32	33	34	35	36	37	37	11	12	13
1	1	1								
13	14	15								

รูปที่ ก-1 แสดงรูปแบบตัวเลขที่เก็บข้อมูลได้และเป็นคำตอบใน Flow Chart โดยรวม



รูปที่ ค-2 แสดง Flow Chart โดยรวมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ความสมบูรณ์ของรายงานฉบับนี้ที่เกิดขึ้นมาได้นั้นด้วยความกรุณาและความเอาใจใส่ของท่านอาจารย์ที่ปรึกษา ท่านอาจารย์สุรเดช ศรีไตรลักษณ์ ที่ได้เอื้อเฟื้อและให้คำแนะนำปรึกษา ให้แนวคิดที่เป็นประโยชน์ต่อโครงการมาโดยตลอด ขอขอบพระคุณอาจารย์ทุกท่านสำหรับ สำหรับทักษะวิชาความรู้อันทำให้เกิดโครงการและรายงานฉบับนี้ขึ้นมาได้ เพื่อนๆทุกคนที่ให้ กำลังใจและคำปรึกษาด้วยดีเสมอมา

และสุดท้าย ขอขอบพระคุณบิดา มารดาที่ได้ให้การสนับสนุนและส่งเสริมมาโดยตลอด

นาย ณัฐพงศ์ เทียนดี

นาย ทศพร ศรีสง่างามกุล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Keneth R. Cattleman, "Digital Image Processing," Englewood Cliffs, NJ : Prentice Hall, 1996.
2. Lawrence O'Gorman, Rangachar Kasturi, "Document Image Analysis," Los Alamitos, CA : IEEE Computer Society Press, 1995.
3. Valluru B. Rao, Hayagriva V. Rao, "C++ Neural Networks and Fuzzy Logic," New York, MIS Press, 1993.
4. Rafael C. Gonzalez and Richard E. Woods , "Digital Image Prodesing", Addison – Wesly Publishing Company, Inc., 1992, p491-495
5. Shi-Kuo Chang, Principles of Pictorial Information Systems Design, Prentice – Hall International, Inc., 1989, p47-49
6. นาย นพพร คงคำ, การรู้จำตัวอักษรคัดลายมือภาษาไทย, ปรียญานิพนธ์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2539

