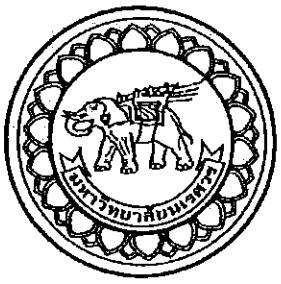


การรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกันจากภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตา

RECOGNIZING CHARACTER WITH DIFFERENCE DEPTH PERSPECTIVE FROM BIRD-EYE-VIEW AND WORM-EYE-VIEW PICTURES

นายชานนท์ วงศ์ประเสริฐ รหัส 49370937

นางสาวรุ่งนภา ยาณะเรือง รหัส 49371286

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2554

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกัน จากภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตา

ผู้ดำเนินโครงงาน นายชานนท์ วงศ์ประเสริฐ รหัส 49370937
นางสาวรุ่งนภา ยาณะเรือง รหัส 49371286

อาจารย์ที่ปรึกษา ดร.สุรเดช จิตประไพกุลศาล

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2554

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....ประธานกรรมการ
(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ
(อาจารย์สิรภพ คชรัตน์)

หัวข้อโครงงาน	การรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกัน จากภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตา
ผู้ดำเนินโครงงาน	นายชานนท์ วงศ์ประเสริฐ รหัส 49370937
	นางสาวรุ่งนภา ยาณะเรือง รหัส 49371286
ที่ปรึกษาโครงงาน	ดร.สุรเดช จิตประไพกุลศาล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

บทคัดย่อ

โครงงานนี้เป็นโปรแกรมที่ใช้ในการรู้จำตัวอักษร โดยวิธีการปรับปรุงภาพแล้วเข้าสู่โปรแกรม OCR เพื่อทำการรู้จำ

ภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตา ยังมีปัญหาในเรื่องที่ขนาดความกว้างของตัวอักษร ทำให้เกิดการมองเห็นที่คลาดเคลื่อนไปจากความเป็นจริง

โปรแกรมที่เราทำการพัฒนาช่วยให้ภาพถ่ายมีความคมชัดของตัวอักษร และปรับขนาดของตัวอักษรให้มีขนาดความกว้างเท่ากัน เมื่อนำไปทำการรู้จำก็ทำให้การรู้จำถูกต้องแม่นยำมากขึ้น

Project Title	Recognizing character With Difference Depth Perspective from Bird-eye-view and Worm-eye-view pictures.
Name	Mr. Chanont Wongprasert ID. 49370937
	Miss. Rungnapa Yanarueng ID. 49371286
Project Advisor	Dr.Suradet Jitprapaikulsarn
Major	Computer Engineering.
Department	Electrical and Computer Engineering.
Academic Year	2011

ABSTRACT

In this project, a program was developed to recognize characters from an image of a license plate that was taken from a top (bird-eyed view) or bottom (worm-eyed view) angle. Typical optical character recognition (OCR) programs usually have difficulties in recognizing these kinds of images due to different depth perspectives. Our program improves the accuracy of the OCR programs by pre-processing the images in order to eliminate the impact of different depth perspectives. According to our experiments, our pre-processing usually leads to better accuracy.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ เกิดขึ้นเนื่องจากการทำงานร่วมกันในหลายๆส่วน บุคคลแรกที่ต้องกล่าวถึงคือ ดร.สุรเดช จิตประไพกุลศาล อาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานนี้ ที่คอยให้ความเอาใจใส่แนะนำ และช่วยเหลืออยู่เสมอ ทำให้โครงงานชิ้นนี้จนสำเร็จลุล่วงไปได้ด้วยดี

ทั้งนี้ต้องขอขอบพระคุณกรรมการทั้งสองท่าน อันได้แก่ ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล และ อาจารย์สิรภพ คชรัตน์ อาจารย์ประจาภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ที่เสียสละเวลาอันมีค่าให้คำปรึกษา และแนะแนวทางในการแก้ปัญหาต่างๆ

และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้ผู้จัดทำมีวันนี้ก็คือ บิดา มารดา และพี่ชาย อันเป็นที่เคารพรักยิ่ง ที่ให้โอกาสทางการศึกษามาเป็นอย่างดี และยังให้กำลังใจเมื่อเกิดปัญหา เอาใจใส่อย่างเต็มที่ในทุกๆด้านอันหาที่เปรียบมิได้ ขอบคุณรุ่นพี่ที่ให้คำแนะนำรวมทั้งผองเพื่อนที่คอยให้กำลังใจทำให้ได้สำเร็จ ผู้จัดทำขอขอบคุณทุกๆคนที่ไม่ได้กล่าวมา ณ ที่นี้ด้วย

นายชานนท์ วงศ์ประเสริฐ

นางสาวรุ่งนภา ยาณะเรือง

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

ตาราง หน้า

สารบัญรูป

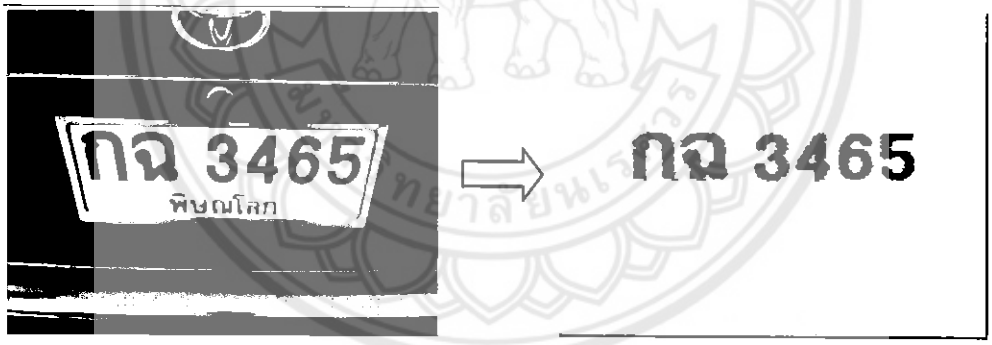
รูปที่ หน้า

สารบัญรูป(ต่อ)

บทที่1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันตัวอักษรมีมากมายหลากหลายรูปแบบ ทำให้จำเป็นต้องมีการรู้จำตัวอักษรเพื่อใช้ในการทำงานหลายๆด้าน เช่น การคัดแยกสิ่งพิมพ์ การแปลเอกสารให้คอมพิวเตอร์สามารถเก็บข้อมูลได้ แต่การรู้จำตัวอักษรยังมีข้อจำกัดอยู่หลายๆด้าน เช่น ความคมชัดของตัวอักษร ขนาดของตัวอักษร ความลึก และมุมของตัวอักษร ทำให้ความสามารถในการรู้จำตัวอักษรทำงานได้อย่างไม่เต็มประสิทธิภาพ โดยเฉพาะปัญหาที่เกิดจากการถ่ายภาพเอียงหรือปัญหาที่เกิดจากภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตา ซึ่งภาพที่ได้มีมุมของตัวอักษรที่เอียง และมีขนาดไม่เท่ากัน ซึ่งเราได้ทดลองกับป้ายทะเบียนรถที่ถ่ายในมุมที่มีขนาดต่างกันและระยะห่างในการถ่ายที่ต่างกันในการทดลองโครงงานนี้ โดยที่โปรแกรมที่เราทำการพัฒนาจะสามารถนำไปใช้ได้ไม่จำกัดเฉพาะป้ายทะเบียนรถเท่านั้น ป้ายต่างๆก็สามารถแก้ปัญหาที่เกิดขึ้นในลักษณะนี้ได้เช่นกัน



รูปที่ 1.1 ภาพที่เกิดจากการถ่ายเอียงแบบบนล่าง

ถ้าสามารถแก้ปัญหาการรู้จำของภาพที่ตัวอักษรมีขนาดความกว้างของตัวอักษรที่ถ่ายจากภาพแบบด้านบนล่างได้ ก็จะทำให้ประสิทธิภาพในการรู้จำตัวอักษรมีสูงขึ้นและยังสามารถนำไปใช้ประโยชน์ในด้านต่างๆ เช่น การรู้จำในรูปแบบอื่นๆต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 พัฒนาโปรแกรมการรู้จำตัวอักษรให้สามารถรู้จำตัวอักษรในภาพถ่ายที่ถ่ายจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่ถ่ายจากมุมต่ำกว่าระดับสายตา

1.3 ขอบข่ายของโครงงาน

1.3.1 โปรแกรมที่ทำการพัฒนาจะทดสอบเฉพาะป้ายทะเบียนรถยนต์

1.3.2 ตัวอักษรและตัวเลขที่นำมาทำการรู้จำ มีจำนวน 52 ตัว ประกอบด้วย

- ตัวอักษรภาษาไทย จำนวน 42 ตัว
- ตัวเลขอินดูอารบิก จำนวน 10 ตัว

1.3.3 ตัวอักษรในภาพต้องเป็นตัวอักษรสีดำเท่านั้น

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาการทำงานของการรู้จำตัวอักษร

1.4.2 ศึกษาการประมวลผลภาพดิจิทัล

1.4.3 ศึกษาและทดลองวิธีการรู้จำตัวอักษร

1.4.4 แก้ไขข้อบกพร่องปรับปรุงวิธีการให้มีประสิทธิภาพสูงขึ้น

1.4.5 สรุปรายงานและจัดทำรูปเล่ม

1.5 แผนผังการดำเนินงาน

ตารางที่ 1.1 แผนผังการดำเนินงาน

กิจกรรม	ก.ค. 2553	ส.ค. 2553	ก.ย. 2553	ต.ค. 2553	พ.ย. 2553	ธ.ค. 2553	ม.ค. 2554	ก.พ. 2554	มี.ค. 2553	เม.ย. 2553
1. ศึกษาข้อมูลในการทำโครงงาน	←	—	—	→						
2. ปรับภาพให้เหมาะสมกับการรู้จำ		←	—	→						
3. วิเคราะห์ความเป็นไปได้ของวิธีการที่จะใช้ในการรู้จำ			←	—	→					
4. ทดสอบอัลกอริทึมต่างๆ ที่สามารถเป็นไปได้				←	—	—	—	→		
5. ปรับปรุงและแก้ไขอัลกอริทึม								←	—	→
6. วิเคราะห์และสรุปผล									←	—
7. จัดทำรูปเล่ม										←

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้ซอฟแวร์ที่ใช้ในการรู้จำตัวอักษร ในภาพถ่ายที่เกิดจากภาพถ่ายที่ถ่ายจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่ถ่ายจากมุมต่ำกว่าระดับสายตา

1.6.2 ได้รับความรู้เรื่องการรู้จำตัวอักษร

1.6.3 ได้รับความรู้เรื่องการประมวลผลภาพดิจิทัล

1.6.4 นำความรู้ที่ได้ไปเป็นแนวทางในการพัฒนาการรู้จำต่อไป

1.7 งบประมาณของโครงงาน

1.7.1 ค่าจัดทำรายงาน	1,000 บาท
1.7.2 ค่าถ่ายเอกสาร	1,000 บาท
รวม	2,000 บาท

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่2
ทฤษฎีพื้นฐาน

ในบทนี้เราจะกล่าวถึงทฤษฎีพื้นฐานต่างๆที่เราจะนำมาใช้ในกระบวนการรู้จำตัวอักษรจากภาพที่ระดับความกว้างของตัวอักษรในภาพที่แตกต่างกัน โดยจะประกอบด้วยเนื้อหาดังนี้

1. การประมวลรูปภาพดิจิทัล (Digital Image Processing)
2. หลักการของการรู้จำตัวอักษร
3. การค้นหาความกว้างของตัวอักษร
4. การปรับปรุงภาพก่อนทำการรู้จำ
5. ปัญหาที่เกิดจากภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตา

2.1 การประมวลรูปภาพดิจิทัล (Digital Image Processing)

เป็นกระบวนการใดๆที่กระทำต่อภาพเพื่อให้ภาพที่ผ่านการประมวลผลมีความคมชัด เช่น

- การปรับภาพให้ดูดีขึ้น คมชัดขึ้น
- การแก้ไขภาพที่ไม่ชัด
- การรู้จำภาพ (Image recognition)
- การแก้ไขภาพในเชิงเรขาคณิต เช่นภาพจากดาวเทียม
- การแยกแยะบริเวณในภาพ เช่น ภาพที่ต้องการคำนวณขนาดของอวัยวะในภาพ เป็นต้น

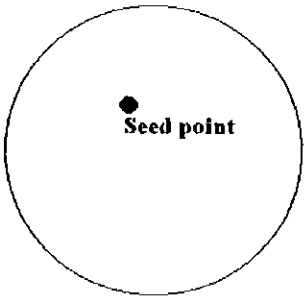
เราจะใช้เทคนิคในการประมวลผล โดยเราจะกล่าวถึงเทคนิคดังนี้

2.1.1 ฟลัดฟิล (Flood Fill) [8]

หลักการของ (Flood Fill Algorithm) มีอยู่ว่า รูปทรงที่ต้องการระบายจะต้องมีขอบที่หนาพอที่จะไม่ทำให้สีหลุดออกไปจากภาพได้ และจุดแรกที่ต้องการจะ fill เรียกว่า seed point โดย seed point นี้จะเป็นจุดใดก็ได้แต่ต้องอยู่ในภาพนั้น เท่านั้น

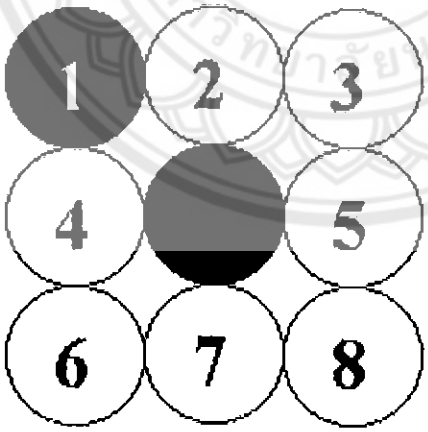
ขั้นตอนการระบายทำได้ดังนี้

1. ระบายจุดแรกหรือ seed point



2. ระบาย 4 จุดรอบ seed point (4_connected)

หรือ 8 จุดรอบ seed point (8_connected)



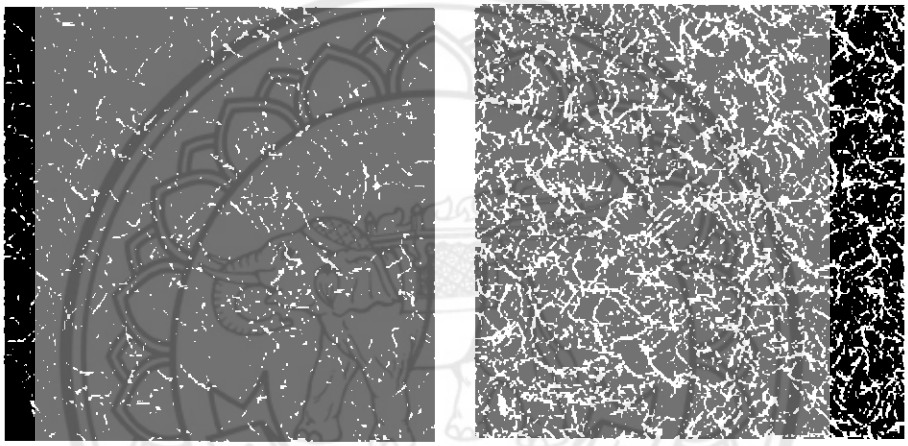
3. จุดที่ 1 จากข้อ 2 จะกลายเป็น seed point ของรอบใหม่ โดยมีเงื่อนไขอยู่ว่าจะไม่ระบายจุดที่ระบายแล้วและจะไม่ระบายขอบภาพ
4. การระบายจะเป็นไปในลักษณะ recursive นั่นคือ จะทำการระบายจุดที่ 1 ต่อไปเรื่อยๆจนกระทั่งจนถึงขอบของวัตถุแล้วจึงจะกลับมาระบายจุดที่เหลือซึ่งยังรออยู่เป็น seed point ต่อไป

2.1.2 การลดสิ่งรบกวน

การแปลงภาพสี RGB ให้เป็นภาพระดับสีเทา การแปลงภาพสีให้เป็นภาพระดับสีเทาเป็นการทำเพื่อทำให้การประมวลภาพมีความรวดเร็วและง่ายขึ้น จึงมีการเปลี่ยนภาพสีให้อยู่ในรูปของภาพระดับสีเทาที่มีค่าอยู่ในช่วง 0 ถึง 255

2.1.3 อแดปทีฟเทรดโฮล (Adaptive Thresholding)

ปกติจะใช้เวลาปรับระดับสีเทาหรือสีภาพเป็นค่านำเข้าและในการดำเนินงานที่ง่ายที่สุดจะใช้ผลภาพไบนารี่แทนการแบ่งส่วน สำหรับพิกเซลในภาพแต่ละเกณฑ์จะต้องมีการคำนวณ ถ้าค่าต่ำกว่าเกณฑ์พิกเซลมันกำหนดค่าเป็นพื้นหลังมิฉะนั้นจะถือว่าค่าเบื้องหน้า



รูปที่ 2.1 ภาพที่เกิดจากการใช้วิธี อแดปทีฟเทรดโฮล (Adaptive Thresholding)

2.1.4 การย่อและขยายภาพ (Scaling)

การย่อและการขยายภาพสามารถทำได้โดยการใช้ Scaling factor ได้แก่ Sx และ Sy ซึ่งใช้สำหรับการย่อและการขยายภาพในทางแกน x และ y ตามลำดับ โดยถ้า

$0 < S_x, S_y < 1$ แสดงว่าเป็นการย่อภาพ

$S_x, S_y > 1$ แสดงว่าเป็นการขยายภาพ

$S_x = S_r$ แสดงว่าย่อและขยายจะเป็นไปตามสัดส่วน

$S_x \neq S_r$ แสดงว่าย่อและขยายจะไม่เป็นอัตราส่วน

สมการของการ Scaling จะมีลักษณะดังนี้

$x' = x.S_x$

$y' = y.S_y$

ดังนั้นย่อและขยายภาพโดยใช้เมตริกซ์จะมีลักษณะดังนี้คือ $P' = S.P$ เมื่อ

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad P = \begin{bmatrix} x \\ y \end{bmatrix} \text{ และ } S = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

การย่อและขยายภาพเมื่อจุด Fixed ไม่ได้อยู่ที่จุด Origin

วิธีการในการย่อและขยายภาพเมื่อจุด Fixed ของการย่อและขยายไม่ได้อยู่ที่จุด Origin สามารถทำได้ดังนี้คือ

1. ให้ย้ายตำแหน่งไปยังจุด Origin
2. ทำการย่อและขยายรอบจุด Origin
3. ย้ายไปยังจุด Fixed Point เหมือนเดิม

ซึ่งจะได้สมการการย่อและขยายภาพดังนี้คือ

$$x' = (x - x_f)S_x + x_f$$

$$y' = (y - y_f)S_x + y_f$$

จะแปลงได้เป็นดังนี้คือ

$$x' = xS_x + x_f(1 - S_x)$$

$$y' = yS_y + y_f(1 - S_y)$$

ดังนั้นการย่อและขยายภาพโดยใช้เมตริกจะมีลักษณะดังนี้คือ

$$P' = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_f(1 - S_x) \\ y_f(1 - S_y) \end{bmatrix}$$

2.2 หลักการของการรู้จำตัวอักษร[2]

การรู้จำตัวอักษรเป็นการนำรูปภาพที่ได้มาทำการวิเคราะห์ เพื่อทำการแปลงรูปภาพที่ได้ให้เป็นรหัสตัวอักษร โดยการนำความรู้หลายส่วนมาประกอบเข้าด้วยกัน เช่น Digital Image Processing, Pattern Recognition การรู้จำตัวอักษรมีโครงสร้างดังนี้

2.2.1 การประมวลผลขั้นตอน

เนื่องจากรูปภาพที่เราจะนำมาทำการรู้จำตัวอักษรนั้น อาจจะมีปัญหาที่ส่งผลกระทบต่อการรู้จำตัวอักษร ทำให้ต้องมีการ นำภาพที่ได้มาทำการปรับปรุงภาพให้มีประสิทธิภาพพร้อมที่จะเข้าสู่กระบวนการ การประมวลผล ดังนั้นขั้นตอนนี้จึงมีความสำคัญต่อโครงงานนี้ เพราะหากเกิดการผิดพลาดก็จะส่งผลต่อการทำงานในขั้นตอนต่อไป

ขั้นตอนต่างๆประกอบด้วยวิธีการดังนี้

- การกรองสิ่งรบกวน(Noise Filtering)

เนื่องจากภาพที่เรานำมาทำการรู้จำตัวอักษรอาจจะมีความไม่สมบูรณ์ของภาพ หรืออาจมีสิ่งรบกวนต่างๆทำให้คุณภาพของภาพไม่ดี ดังนั้นเราจึงต้องทำการกำจัด

หรือลดสิ่งรบกวนออกจากภาพก่อนจะทำการรู้จำตัวอักษร

- การปรับแต่งข้อมูล (Normalization)

ขั้นตอนนี้จะเป็นการปรับขนาดของตัวอักษรหรือทำการปรับสีของภาพจากภาพสีให้กลายเป็นภาพสีขาวดำ เพื่อนำมาทำการรู้จำตัวอักษร

- การตัดแบ่งพื้นที่ใช้งาน (Cropping)

ขั้นตอนนี้จะเป็นการตัดแบ่งส่วนของภาพที่จำเป็นต้องนำมาทำการรู้จำตัวอักษรเพื่อส่งไปยังขั้นตอนต่อไป

2.2.2 การรู้จำ (Recognition)

การรู้จำนั้นระบบจะเป็นตัวตัดสินว่ารูปที่เรานำมาทำการรู้จำนั้น จำถูกเก็บไว้เป็นรหัสตัวอักษรใด เราสามารถแยกวิธีการรู้จำได้ 4 กลุ่ม ดังนี้

- วิธีการรู้จำแบบ Template Matching

วิธีการเข้าคู่รูปแบบเป็นวิธีการแรกๆ ที่มาใช้ในการรู้จำตัวอักษร หลักการโดยทั่วไปคือจะต้องมีรูปแบบ (template) ที่สร้างขึ้นมาสำหรับอ่านตัวอักษร โดยมีการกำหนดตำแหน่งสำคัญที่สามารถใช้แยกแยะความแตกต่างระหว่างตัวอักษรแต่ละตัว เวลาทำงานก็ให้นำรูปภาพที่ต้องการอ่านไปทาบบนแบบเพื่อวัดความคล้ายคลึงกันของภาพกับตัวแบบ จากนั้นก็ระบุว่าเป็นรหัสตัวอักษรอะไร โดยใช้ค่าผ่านระดับหรือวิธีการบางอย่างในการตัดสิน วิธีการนี้จะค่อนข้างอ่อนไหวต่อข้อมูลแทรกซ้อน ขนาด และการเอียงของตัวอักษร จึงจำเป็นต้องมีขั้นตอนการปรับแต่งข้อมูลที่ดี

- วิธีการวิเคราะห์ทางโครงสร้าง (Structural Analysis)

วิธีการวิเคราะห์ทางโครงสร้างคือการวิเคราะห์โครงสร้างตัวอักษรโดยถือว่าตัวอักษรทุกตัวประกอบด้วยองค์ประกอบพื้นฐาน ซึ่งได้มาจากการสกัดลักษณะสำคัญ เช่นเดียวกันกับวิธีการทางสถิติ ต่างกันตรงที่ลักษณะสำคัญ ที่ส่งมาให้กับขั้นตอนการรู้จำแบบการวิเคราะห์ทางโครงสร้างนี้ มักจะใช้เป็นชื่อหรือค่าที่บอกว่าลักษณะโครงสร้างสำคัญนั้นเป็นอะไร เช่น เส้นตรง วงกลม เป็นต้น แทนที่จะเป็นค่าจำนวนจริง ในขั้นตอนการรู้จำลักษณะสำคัญทั้งหลายที่ประกอบเป็นตัวอักษรนั้น จะถูกส่งเข้าไปให้กับส่วนที่ตรวจวิเคราะห์กฎการเขียนตัวอักษร เช่น ฟอร์มอลแกรมมาแมชชีน (formal grammar machine) โครงสร้างกราฟ หรือโครงสร้างต้นไม้เป็นต้น เพื่อระบุว่าเป็นตัวอะไร ซึ่งจะตัดสินโดยการดูที่รูปแบบการเชื่อมต่อขององค์ประกอบต่างๆเข้าเป็นตัวอักษรนั้นวิธีการนี้มีข้อดีตรงที่มีความยืดหยุ่นต่อความหลากหลายของตัวอักษรค่อนข้างมาก

- วิธีทางโครงข่ายประสาทเทียม (Neural Network)

วิธีทางโครงข่ายประสาทเทียมเป็นแนวทางใหม่ที่ได้รับความนิยมอย่างมากในช่วงหลัง

เนื่องจากประสิทธิภาพในด้านการรู้จำแบบ ซึ่งถูกนำไปใช้ในงานหลายๆ ด้าน รวมทั้งโอซีอาร์ด้วย โครงข่ายประสาทเทียมเป็นเทคนิคที่พยายามเรียนแบบการทำงานของสมองมนุษย์ ที่มีโครงข่ายเชื่อมต่อกันของหน่วยความจำย่อยๆ จำนวนมากที่สะสมความรู้เอาไว้ ความรู้เหล่านี้จะได้จากการฝึกสอนไว้ก่อน เช่นการสอนให้รู้จักตัวอักษร "ก" ถึง "ฮ" โดยการส่งภาพตัวอักษรเหล่านี้เข้าไป พร้อมกับบอกว่ามีค่าเป็นรหัสตัวอักษรอะไร โครงข่ายประสาทเทียมจะเรียนรู้ถึงรูปแบบตัวอักษรที่หลากหลายของตัวอักษรตัวนั้น เพื่อว่าเวลาทำงานจริงจะได้มีความสามารถพอที่จะรับมือกับภาพตัวอักษรในหลายๆ รูปแบบ สิ่งที่สอนให้กับโครงข่ายประสาทเทียมไม่จำเป็นต้องเป็นรูปของตัวอักษรอย่างที่เราเห็นกันก็ได้ อินพุตที่ส่งให้มักจะผ่านขั้นตอนการสกัดลักษณะสำคัญและกระบวนการประมวลผลเบื้องต้นอื่นๆก่อนเสมอ

- วิธีทางสถิติ (Statistical Approach)

วิธีทางสถิติเป็นวิธีการที่ใช้หลักการทางสถิติ โดยนำค่าความน่าจะเป็นและ/หรือฟังก์ชันการแจกแจงความน่าจะเป็นมาใช้ในการตัดสินใจ รูปภาพอินพุตที่ได้มาจากขั้นตอนการสกัดลักษณะสำคัญ จะถูกส่งเข้าไปในส่วนการรู้จำเฉพาะของแต่ละตัวอักษร ซึ่งได้ผลลัพธ์ออกมาเป็นค่าความน่าจะเป็นที่อินพุตเป็นตัวอักษรใด เมื่ออินพุตได้ผ่านส่วนการรู้จำครบทุกตัวแล้ว ก็นำเอาผลลัพธ์ที่ได้ทั้งหมดมาเปรียบเทียบกันว่าได้ค่าความน่าจะเป็นของตัวอักษรใดมากที่สุด ผลลัพธ์จะออกเป็นตัวอักษรนั้นขบวนการประมวลผลขั้นปลาย (Post-Processing)หลังจากที่ผ่านขั้นตอนการรู้จำแล้วรูปตัวอักษรที่ถูกส่งเข้าไปจะได้ผลลัพธ์ออกมาเป็นรหัสตัวอักษร ซึ่งก็ไม่ได้หมายความว่าเอาต์พุตที่ได้มาจะถูกต้องทั้งหมด ไม่มีผลิตภัณฑ์โอซีอาร์ตัวใด ไม่ว่าจะเป็นภาษาใดก็ตามที่รับรองความถูกต้อง 100 % ดังนั้นเพื่อเพิ่มความถูกต้องให้แก่โปรแกรมจึงได้มีการเสริมส่วนการตรวจสอบและแก้ไขข้อความเข้ามา โปรแกรมส่วนนี้มักจะทำงานเกี่ยวกับการตรวจสอบความถูกต้องของการสะกดคำและไวยากรณ์ภาษา โดยมักจะใช้พจนานุกรมมาช่วยในการตรวจสอบคำผิด ซึ่งอาจแก้ไขให้โดยอัตโนมัติหรือแสดงเครื่องหมายบางอย่าง เพื่อบอกให้ผู้ใช้ทราบว่าคำดังกล่าวอาจไม่ถูกต้อง ซึ่งผู้ใช้อาจแก้หรือไม่แก้ก็ขึ้นกับการตัดสินใจของผู้ใช้เอง นอกเหนือไปจากการตรวจสอบความถูกต้องระดับคำแล้ว บางโปรแกรมยังมีความสามารถตรวจสอบไวยากรณ์ในระดับประโยคได้ด้วย

2.3 การค้นหาความกว้างของตัวอักษร

ภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตาส่งผลกระทบให้รูปภาพมีขนาดของตัวอักษรไม่เท่ากัน จึงต้องทำการค้นหาความกว้างของตัวอักษร ค่า

เริ่มต้นแรกของตัวอักษรและค่าสุดท้ายของตัวอักษรตัวนั้น แล้วนำค่าที่ได้มาทำการปรับปรุงรูปภาพก่อนจะทำการรู้จำ โดยมีขั้นตอนดังนี้

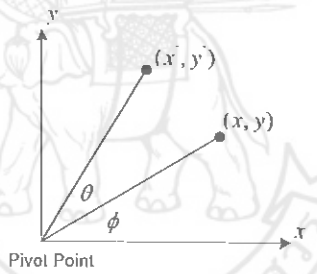
2.3.1 การแปลงภาพเป็นภาพขาวดำ (Binary)

เพื่อความสะดวกในการประมวลผลภาพจึงต้องมีการแปลงจากภาพสีให้กลายเป็นภาพ Grayscale โดยการทำ cvcolor และแปลงภาพ Grayscale ให้กลายเป็นภาพแบบ Binary โดยการทำ Adaptive Threshold

การทำอาจทำให้เกิดสิ่งรบกวน จึงต้องทำการลดสิ่งรบกวนเพื่อหลีกเลี่ยงปัญหาที่จะตามมา เช่น การหาความกว้างของตัวอักษรผิดพลาด การรู้จำตัวอักษรผิดพลาด การรู้จำตัวอักษรจากป้ายทะเบียนรถจะใช้การลดสิ่งรบกวนโดยใช้ค่ากลาง

2.3.2 การหมุนภาพ (Rotation)

เป็นการหมุนตำแหน่งของภาพในระนาบ XY รอบจุด Pivot Point (จุดหมุน)



$$x = r\cos(\phi)$$
$$y = r\sin(\phi) \qquad (1)$$

และ

$$x' = r\cos(\phi+\theta) = r(\cos\phi\cos\theta - \sin\phi\sin\theta)$$
$$y' = r\sin(\phi+\theta) = r(\sin\phi\cos\theta - \cos\phi\sin\theta) \qquad (2)$$

เพราะฉะนั้นจากสมการที่ (1) และ (2) จะได้สมการของการหมุนรอบจุด Pivot Point ดังนี้คือ

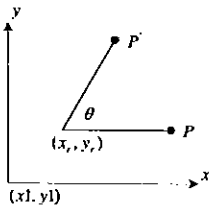
$$x' = x\cos(\theta) - y\sin\theta$$
$$y' = x\sin(\theta) - y\cos\theta \qquad (3)$$

ซึ่งสามารถเขียนให้อยู่ในรูปแบบของเมตริกได้ มีลักษณะดังนี้คือ $P' = R.P$ เมื่อ

$$P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad P = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \text{และ} \quad R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad (4)$$

การหมุนภาพเมื่อจุด Pivot ไม่ได้อยู่ที่จุด Origin พิจารณาเมื่อจุด Pivot ไม่ได้อยู่ใน

ตำแหน่ง (0,0) (ย้ายไปอยู่ที่ตำแหน่ง (x_r, y_r))



การในการหมุนภาพเมื่อจุดหมุนไม่ได้อยู่ที่จุด Origin สามารถทำได้ดังนี้คือ

1. ทำการเปลี่ยนจุด Pivot ไปยังจุด Origin
2. ทำการหมุนรอบจุด Origin
3. ย้ายกลับไปยังจุดเดิม โดยการบวกด้วย x_r และ y_r
4. สมการการหมุนรอบจุด Pivot ใด ๆ ที่ไม่ใช้จุด Origin มีลักษณะดังนี้คือ

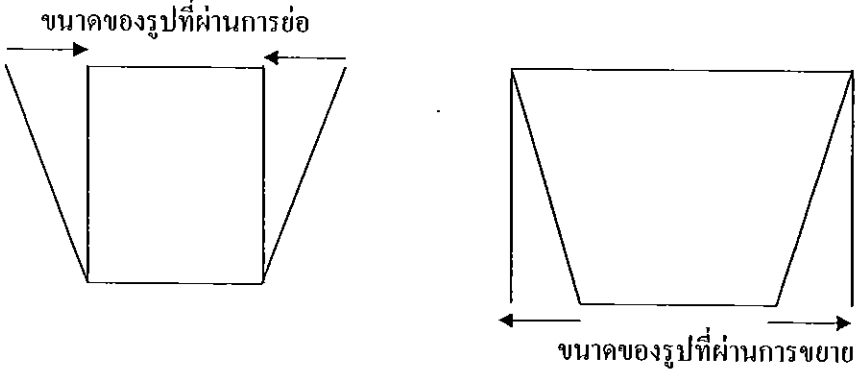
$$x' = (x - x_r)\cos\theta - (y - y_r)\sin\theta + x_r$$
$$y' = (x - x_r)\sin\theta + (y - y_r)\cos\theta + y_r \qquad (5)$$

2.4 การปรับปรุงภาพก่อนทำการรู้จำ

หลังจากที่เราได้ภาพถ่ายที่เราต้องการมาแล้ว นำมาทำการปรับปรุงภาพโดยมีวิธีการดังนี้

2.4.1 การปรับภาพบรรทัดตัวอักษรเข้าสู่แนวระดับ

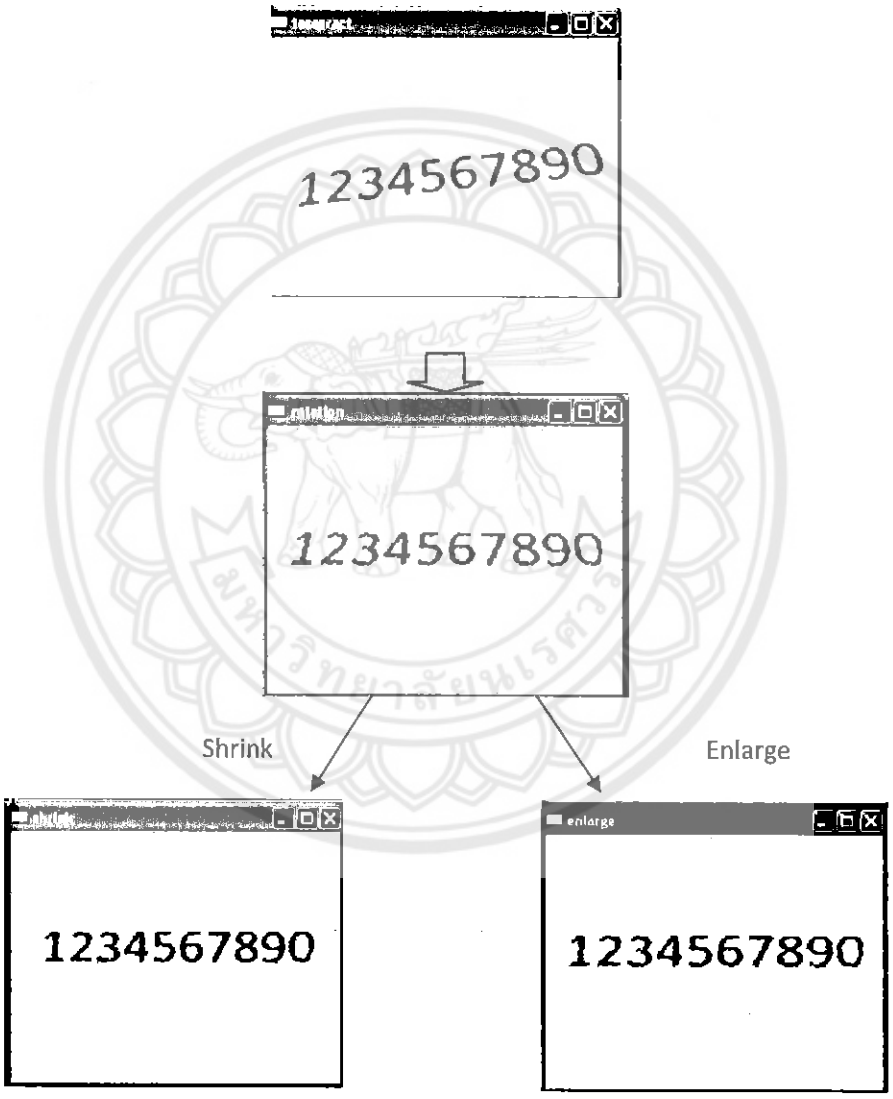
ผลจากการย่อหรือขยายบรรทัดตัวอักษรจะทำให้ระดับความสูงของตัวอักษร ในภาพมีความใกล้เคียงกับความสูงปกติ แต่ตัวอักษรยังไม่อยู่ในแนวระดับ เราจึงทำการปรับภาพตัวอักษรจากความชันให้เข้าสู่แนวระดับปกติ โดยอาศัยสมการ M = y2-y1 / x2-x1 เป็นหลัก และทำการเลื่อนค่าของจุดในรูปภาพแบบจุดต่อจุดในแนวระดับ ถ้า M $\neq$0 ให้ทำการปรับทุกแถวในบรรทัดจนกระทั่ง M = 0 เมื่อเสร็จสิ้นกระบวนการจะได้รูปภาพบรรทัดอักษรในแนวระดับปกติ



รูปที่ 2.2 ทำการย่อและขยายด้วยวิธีการย่อ (Shrink) และขยาย (Enlarge)

2.4.2 ย่อและขยายตัวเลขและตัวอักษร

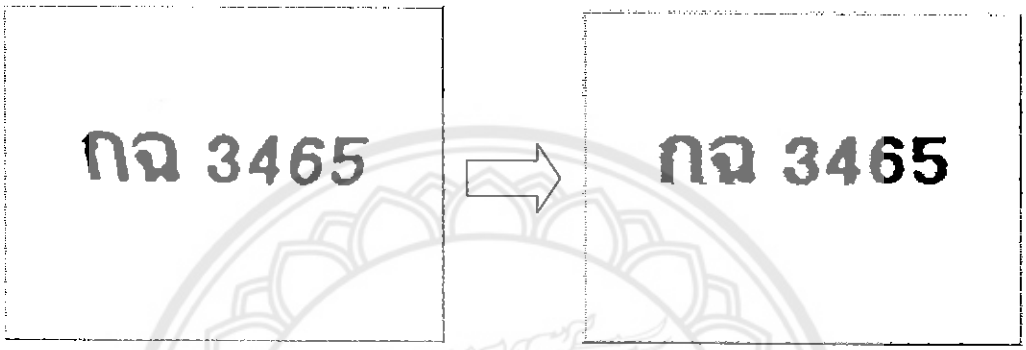
เป็นการปรับขนาดของตัวอักษรและตัวเลขให้มีขนาดเท่ากันทั้งรูปเพื่อให้ขนาดของตัวอักษรและตัวเลขมีขนาดเท่ากับขนาดปกติหรือใกล้เคียงเพื่อทำให้ชัดเจนขึ้น โดยมีขั้นตอนการทำงานดังนี้ นำสมการความชันที่ได้มาแทนค่าด้วยจุดเริ่มต้นและจุดสิ้นสุดของแกน x จากนั้นทำการขยายหรือย่อภาพในแนวระดับ เพื่อให้ระยะห่างของสมการทั้งสองเส้นสำหรับจุดทุกจุดในแกนx มีขนาดเท่ากับขนาดที่เรากำหนดไว้



รูปที่ 2.3 ภาพได้จากการย่อและขยาย

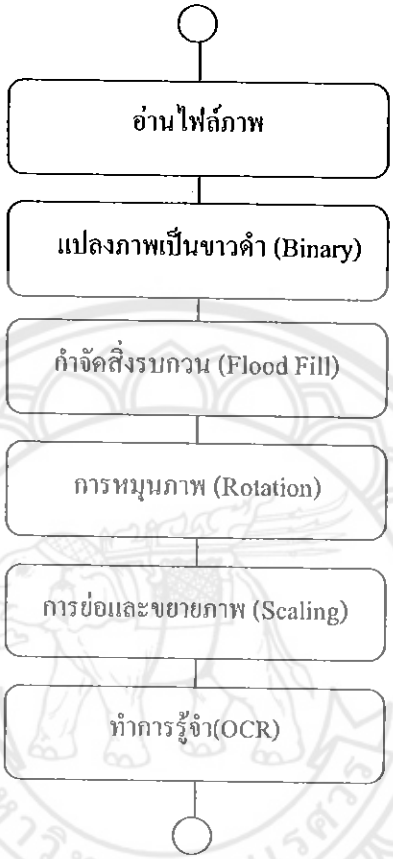
2.5 ปัญหาที่เกิดจากภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตา

ความกว้างของตัวอักษร ภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่มองจากมุมต่ำกว่าระดับสายตา ยังมีปัญหาในเรื่องที่ขนาดความกว้างของตัวอักษรด้านบนและด้านล่างมีขนาดความกว้างไม่เท่ากัน ทำให้เกิดการมองเห็นที่คลาดเคลื่อนไปจากความเป็นจริง



รูปที่ 2.4 ภาพแสดงรูปที่ผ่านการทำงานของโปรแกรมเพื่อปรับขนาด

บทที่ 3
เครื่องมือและการพัฒนาระบบ



รูปที่ 3.1 Flowchart

ในบทนี้จะกล่าวถึงเครื่องมือที่ใช้ในการพัฒนาโปรแกรม และการออกแบบการพัฒนาระบบ จะแบ่งการทำงานออกเป็นส่วนๆดังนี้

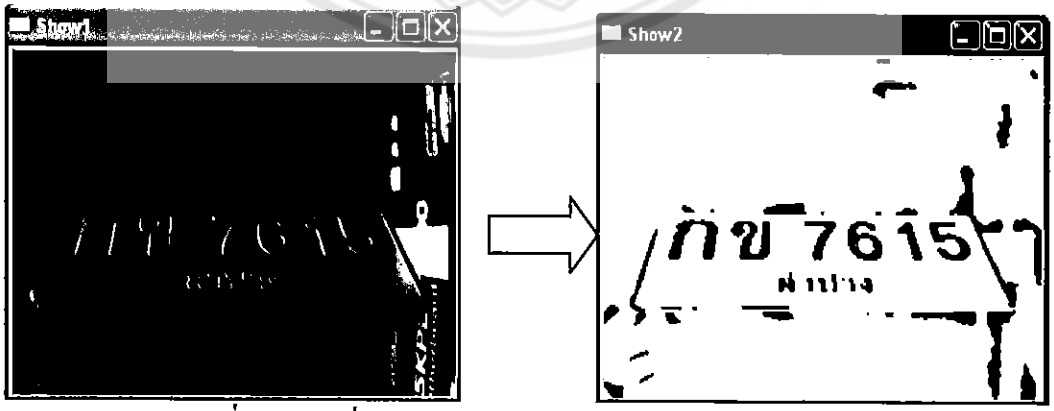
1. การแปลงภาพเป็นภาพขาวดำ (Binary)
2. การกำจัดสิ่งรบกวน (Flood Fill)
3. การหมุนภาพ (Rotation)
4. การย่อและขยายภาพ (Scaling)
5. โอเพ่นซีวี ไลบรารี่ (OpenCV Library)
6. การออกแบบซอฟแวร์

ตารางที่ 3.1 เทคนิคและเครื่องมือที่ใช้

Technique	Tools
การแปลงภาพให้เป็นขาวดำ	- OpenCV
การแยกแยะบริเวณในภาพ - กำจัดสิ่งรบกวน (Floodfill)	- พัฒนาในโปรแกรม
การแก้ไขภาพในเชิงเรขาคณิต - การหมุนภาพ (Rotation) - การย่อและขยายภาพ(Scaling)	- พัฒนาในโปรแกรม
OCR - ภาษาอังกฤษ - ภาษาไทย	- Tesseract - เทรนภาษาไทยให้กับ Tesseract

3.1 การแปลงภาพเป็นภาพขาวดำ (Binary)

ในการแปลงภาพสีให้เป็นภาพขาวดำนั้น มี 2 ขั้นตอนย่อย คือ แปลงภาพสีให้เป็นภาพสแกลสีเทา (Grayscale) แล้วแปลงภาพสแกลสีเทาเป็นภาพขาวดำ (Binary หรือ Black&Whrite)

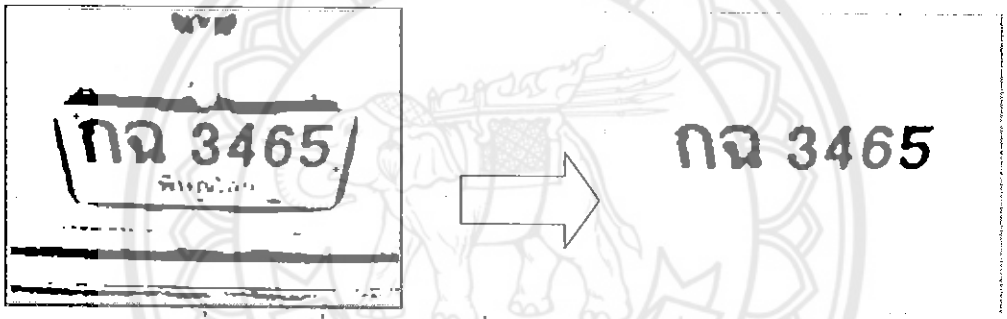


รูปที่ 3.2 ภาพที่ได้จากการแปลงเป็นภาพขาวดำ (Binaryหรือ Black&Whrite)

3.2 การกำจัดสิ่งรบกวน (Flood Fill)

หลักการ Flood Fill เป็นเทคนิคในการเลือกว่าจุดไหนที่เราต้องการกำจัดด้วยการมองด้วยสายตาแล้วทำการกำหนดจุดเพื่อทำการตัดภาพ กำจัดส่วนที่เป็น Noise ออกไป คำสั่งที่ใช้ในการกำจัด Noise คือ

```
if(ptc->row==ptr->row-1&&ptc->col==ptr->col)
else if(ptc->row==ptr->row+1&&ptc->col==ptr->col)
else if(ptc->row==ptr->row&&ptc->col==ptr->col-1)
else if(ptc->row==ptr->row&&ptc->col==ptr->col+1)
else if(ptc->row==ptr->row-1&&ptc->col==ptr->col-1)
else if(ptc->row==ptr->row-1&&ptc->col==ptr->col+1)
else if(ptc->row==ptr->row+1&&ptc->col==ptr->col-1)
else if(ptc->row==ptr->row+1&&ptc->col==ptr->col+1)
```



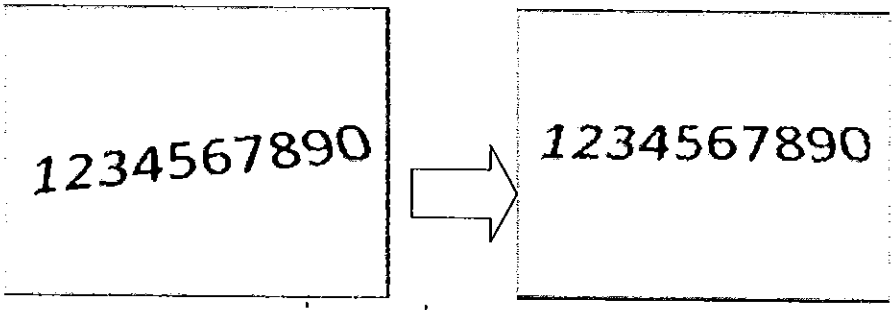
รูปที่ 3.3 ภาพที่ผ่านการกำจัดสิ่งรบกวน (Nois) ด้วยเทคนิค Flood Fill

3.3 การหมุนภาพ (Rotation)

การหมุนภาพเป็นเทคนิคที่ใช้แก้ไขภาพที่มีปัญหาจากการถ่ายภาพในมุมองศาที่ต่างกัน ทำให้เกิดความคลาดเคลื่อนทำให้เห็นตัวอักษรอยู่ในแนวระดับที่ต่างกัน ดังนั้นจึงต้องมีการปรับแนวระดับของตัวอักษรในภาพให้อยู่ในแนวระนาบเดียวกัน ซึ่งสมการที่ใช้ในการปรับแนวระดับของตัวอักษรคือ

$$x' = (x - x_r)\cos\theta - (y - y_r)\sin\theta + x_r$$
$$y' = (x - x_r)\sin\theta + (y - y_r)\cos\theta + y_r$$

และเมื่อนำภาพผ่านขั้นตอนนี้ก็จะได้



รูปที่ 3.4 ภาพที่ผ่านการหมุนภาพ (Rotation)

3.4 การย่อและขยายภาพ (Scaling)

เป็นเทคนิคที่ใช้ในการการปรับขนาดของภาพให้มีขนาดเท่ากันหรือมีขนาดใกล้เคียงกับขนาดปกติ โดยทำการปรับทั้งภาพ เมื่อเราทราบจุดอ้างอิงและรู้ขนาดความยาวของขอบด้านบนเราก็จะทำการปรับความมายาวของขอบด้านล่างเพื่อให้ได้ความยาวที่เท่ากัน ซึ่งการพัฒนาจะใช้คำสั่ง

```
numi=(m*((double)r-(double)x1))+(double)y1;
num2=(y1-1)-((y1-num1)*2);
copy=resize(copy,(y1-1)-((y2-y1)*2),num2,row,col,r);
```

3.5 โอเพ่นซีวี ไลบรารี่ (OpenCV Library)

เป็นเครื่องมือที่ใช้ในการรู้จำตัวอักษร ซึ่งการรู้จำตัวอักษรจะใช้เครื่องมือที่เรียกว่า Tesseract การรู้จำนั้นเราจะทำการรู้จำทั้งภาษาไทยและภาษาอังกฤษ และคำสั่งที่ใช้ในการรู้จำของ Tesseract คือ

```
//OCR
system("D:/tesseract-OCR/tesseract Image2.tif tesseract_eng -l eng");
system("D:/tesseract-OCR/tesseract Image2.tif tesseract_tha -l tha");
```

บทที่ 4
ผลการทดสอบ

การทดสอบความสามารถในการรู้จำภาพที่มีระดับความลึกของตัวอักษรในภาพแตกต่างกัน จากภาพถ่ายที่มองจากมุมสูงกว่าระดับสายตาและภาพที่มองจากมุมต่ำกว่าระดับสายตา จะทดสอบด้วยภาพทะเบียนรถยนต์

ในการทดสอบนี้จะใช้ภาพจากกล้องมือถือความละเอียด 2 ล้านพิกเซล ถ่ายภาพขนาด 320*240 โดยทำการถ่ายที่ระยะห่าง 50 , 75 , 100 เซนติเมตร และมุมในการถ่ายระหว่าง 30 ถึง -30 องศา การถ่ายจะเพิ่มมุมเอียงทีละ 10 องศา ในการทดสอบเราได้ทำการทดสอบกับป้ายทะเบียนรถจำนวน 4 ป้าย ตัวอย่างภาพที่ใช้ในการทดสอบแสดงในรูป

รูปที่ 4.1 ภาพถ่ายป้ายทะเบียนรถยนต์ที่ใช้ในการทดสอบ

4.1 แผนการทดสอบ

4.2 ทดสอบความสามารถในการรู้จำตัวอักษรเทียบกับระยะห่างในการถ่ายภาพ

4.3 ทดสอบความสามารถในการรู้จำตัวอักษรเทียบกับมุมในการถ่ายภาพ

ผลการทดสอบ จะแสดงเป็น 2 ส่วนคือ ตารางแสดงผลการทดสอบและกราฟแสดงผลการทดสอบในแต่ละกรณี โดยจะแสดงค่าความถูกต้องของการรู้จำออกเป็น 3 ส่วนคือ

-Original แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพก่อนทำการปรับปรุงเข้าสู่การรู้จำโดยโปรแกรม Tesseract OCR โดยตรง

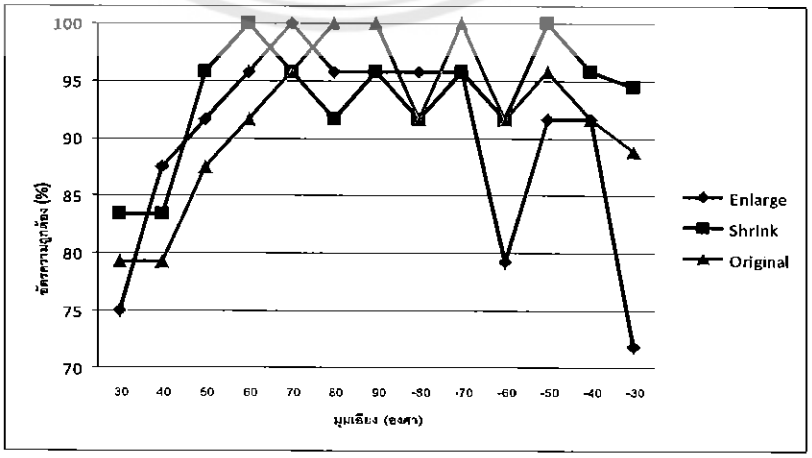
-Shrink แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพบรรทัดตัวอักษรมาย่อบรรทัดตัวอักษร แล้วจึงทำการรู้จำ

-Enlarge แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพบรรทัดตัวอักษรมาขยายบรรทัดตัวอักษร แล้วจึงทำการรู้จำ

สำหรับกรณีที่ ภาพทะเบียนรถยนต์ ที่ถ่ายด้วยระยะ 100 เซนติเมตรนั้น ไม่สามารถทำการสอบในมุม 30 และ- 30 องศาได้เนื่องจากความสมบูรณ์ของตัวอักษรไม่สามารถที่นำมาทำการรู้จำได้

ตารางที่ 4.1 ผลการทดสอบการรู้จำป้ายทะเบียนรถยนต์ ที่ถ่ายจากระยะห่าง 50 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	OCR		
		Original	Shrink	Enlarge
50	30	79.17	83.33	75
	40	79.17	83.33	87.5
	50	87.5	95.83	91.67
	60	91.67	100	95.83
	70	95.83	95.83	100
	80	100	91.67	95.83
	90	100	95.83	95.83
	-80	91.67	91.67	95.83
	-70	100	95.83	95.83
	-60	91.67	91.67	79.17
	-50	95.83	100	91.67
	-40	91.67	95.83	91.67
	-30	88.89	94.44	71.78



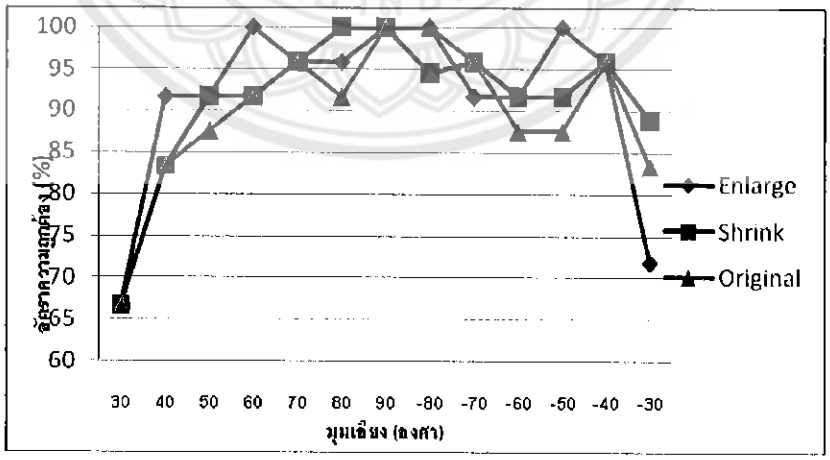
รูปที่ 4.2 กราฟผลการทดสอบการรู้จำป้ายทะเบียนรถยนต์ ที่ถ่ายจากระยะห่าง 50 เซนติเมตร

จากตารางและกราฟแสดงผลการทดสอบ จะเห็นว่าที่ระยะห่าง 50 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีเมื่อมุมในการถ่ายอยู่ในระหว่างมุม 30 ถึง 60 องศา และ -30 ถึง -60 องศา แต่จะสังเกตเห็นว่า วิธีการย่อ (Shrink) สามารถรู้จำได้ดีกว่าการขยาย (Enlarge) และภาพก่อนการปรับปรุง (Original) อย่างชัดเจน แต่มุม 70 ถึง -70 องศานั้น การรู้จำภาพหลังจากการปรังปรุงภาพทำให้มีความผิดพลาดมากยิ่งขึ้น

เมื่อพิจารณาด้านความถูกต้องโดยรวมแล้ว จะสังเกตเห็นว่าวิธีการย่อ (Shrink) สามารถปรับปรุงภาพได้ถูกต้องแม่นยำกว่าภาพก่อนการปรับปรุง (Original) ในมุมภาพที่เอียงต่ำกว่า 70 องศา

ตารางที่ 4.2 ผลการทดสอบการรู้จำป้ายทะเบียนรถยนต์ ที่ถ่ายจากระยะห่าง 75 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	OCR		
		Original	Shrink	Enlarge
75	30	66.67	66.67	66.67
	40	83.33	83.33	91.67
	50	87.5	91.67	91.67
	60	91.67	91.67	100
	70	95.83	95.83	95.83
	80	91.67	100	95.83
	90	100	100	100
	-80	100	94.44	100
	-70	95.83	95.83	91.67
	-60	87.5	91.67	91.67
	-50	87.5	91.67	100
	-40	95.83	95.83	95.83
	-30	83.33	88.89	71.78



รูปที่ 4.3 กราฟผลการทดสอบการรู้จำป้ายทะเบียนรถยนต์ ที่ถ่ายจากระยะห่าง 75เซนติเมตร

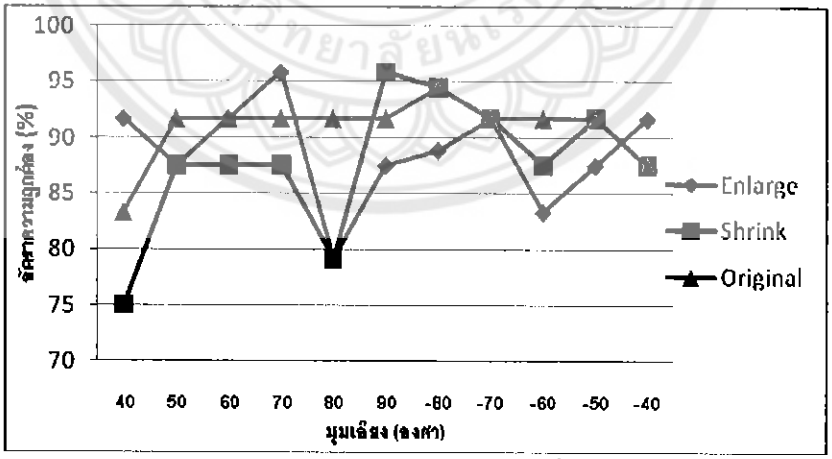
จากตารางและกราฟแสดงผลการทดสอบ จะเห็นว่าที่ระยะห่าง 75 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีเมื่อมุมในการถ่ายอยู่ในระหว่างมุม 30 ถึง 60 องศา และ -30 ถึง -60 องศา

แต่จะสังเกตเห็นว่า วิธีการย่อ (Shrink) และขยาย (Enlarge) สามารถรู้จำได้ดีกว่าภาพก่อนการปรับปรุง (Original) อย่างชัดเจน แต่มุม 70 ถึง -70 องศานั้น การรู้จำภาพหลังจากการปรังปรุงภาพทำให้มีความผิดพลาดมากยิ่งขึ้นในบางมุม

เมื่อพิจารณาด้านความถูกต้องโดยรวมแล้ว จะสังเกตเห็นว่าการย่อ (Shrink) และขยาย (Enlarge) สามารถปรับปรุงภาพได้ถูกต้องแม่นยำกว่าภาพก่อนการปรับปรุง (Original) ในมุมภาพที่เอียงต่ำกว่า 70องศา

ตารางที่ 4.3 ผลการทดสอบการรู้จำป้ายทะเบียนรถยนต์ ที่ถ่ายจากระยะห่าง 100 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	OCR		
		Original	Shrink	Enlarge
100	30	-	-	-
	40	83.33	75	91.67
	50	91.67	87.5	87.5
	60	91.67	87.5	91.67
	70	91.67	87.5	95.83
	80	91.67	79.17	79.17
	90	91.67	95.83	87.5
	-80	94.44	94.44	88.87
	-70	91.67	91.67	91.67
	-60	91.67	87.5	83.33
	-50	91.67	91.67	87.5
	-40	87.5	87.5	91.67
	-30	-	-	-



รูปที่ 4.4 กราฟผลการทดสอบการรู้จำป้ายทะเบียนรถยนต์ ที่ถ่ายจากระยะห่าง 100 เซนติเมตร

จากตารางและกราฟแสดงผลการทดสอบ จะเห็นว่าที่ระยะห่าง 100 เซนติเมตร ทั้งวิธีการย่อ (Shrink) และขยาย (Enlarge) ให้ความผิดพลาดมากยิ่งขึ้นแทบทุกมุม เมื่อเทียบกับภาพก่อนทำการปรับปรุง (Original)

เมื่อพิจารณาด้านความถูกต้องโดยรวมแล้ว จะสังเกตเห็นว่า โปรแกรมไม่สามารถปรับปรุงภาพให้มีความถูกต้องแม่นยำขึ้นเลย

4.4 วิเคราะห์ผลการทดลอง

1. เมื่อมุมในการถ่ายภาพเพิ่มขึ้น การนำภาพเข้าสู่โปรแกรมก่อนการนำไปทำการรู้จำนั้นจะให้ค่าเฉลี่ยในความถูกต้องที่ดีกว่าใช้ภาพก่อนทำการปรับปรุง (Original) โดยตรง

2. โปรแกรมสามารถทำการรู้จำตัวอักษรได้ถูกต้องแม่นยำกว่าภาพก่อนทำการปรับปรุง (Original) เมื่อตัวอักษรมีขนาดใหญ่ เนื่องจากภาพก่อนทำการปรับปรุง (Original) ตัวอักษรค่อนข้างจะมีความเอียง เมื่อปรับปรุงด้วยโปรแกรมแล้วตัวอักษรที่ได้หลังจากผ่านโปรแกรมค่อนข้างจะเข้าสู่ความสมดุล

3. เมื่อระยะทางในการถ่ายเพิ่มขึ้น จะสังเกตว่าการใช้วิธีการย่อ (Shrink) และขยาย (Enlarge) นั้น ความถูกต้องแม่นยำในการรู้จำนั้น ค่อนข้างลดลงเมื่อเทียบกับระยะทางที่ใกล้กว่า และที่ระยะ 100 เซนติเมตร จะเห็นว่า หลักการย่อ (Shrink) และขยาย (Enlarge) นั้นจะให้ความถูกต้องที่ต่ำกว่าภาพก่อนทำการปรับปรุง (Original) อย่างเห็นได้ชัด อันเนื่องมาจากการย่อและขยายภาพ (Scaling) ที่นำมาใช้ยังมีประสิทธิภาพไม่ดีเพียงพอในการทำการโครงงาน

4.5 สรุปผลการทดลอง

4.5.1 ทดสอบความสามารถในการรู้จำตัวอักษรเทียบกับระยะห่างในการถ่ายภาพ

ค่าเฉลี่ยของผลการทดสอบในตารางเป็นการทดสอบกับป้ายทะเบียนรถยนต์จำนวน 4 ป้ายและระยะห่างที่ใช้ในการทดสอบคือ 50 เซนติเมตร, 75 เซนติเมตร และ 100 เซนติเมตร เท่านั้น

ตารางที่ 4.4 ตารางแสดงค่าเฉลี่ยผลการทดสอบเทียบกับระยะห่างในการถ่ายภาพ

ระยะห่าง (เซนติเมตร)	OCR		
	Original	Shrink	Enlarge
50	91.77	93.48	89.81
75	89.74	91.27	91.27
100	90.78	87.75	88.76

4.5.2 ทดสอบความสามารถในการรู้จำตัวอักษรเทียบกับมุมในการถ่ายภาพ

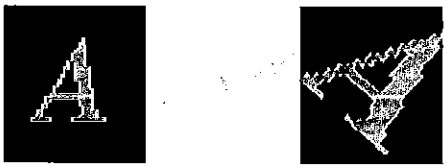
ค่าเฉลี่ยของผลการทดสอบในตารางเป็นการทดสอบกับป้ายทะเบียนรถยนต์จำนวน 4 ป้ายและมุมที่ใช้ในการทดสอบจำนวน 13 มุมคือตั้งแต่ -30 องศา ถึง 30 องศา เท่านั้น

ตารางที่ 4.5 ตารางแสดงค่าเฉลี่ยผลการทดสอบเทียบกับมุมในการถ่ายภาพ

มุมเอียง (องศา)	OCR		
	Original	Shrink	Enlarge
30	72.92	75	70.84
40	81.94	80.55	90.28
50	88.89	91.66	90.28
60	91.67	93.06	95.83
70	94.44	95.83	97.22
80	94.44	90.28	90.28
90	97.22	93.89	94.44
-80	95.37	93.51	94.90
-70	95.83	94.44	83.06
-60	90.28	90.28	84.72
-50	91.66	94.44	93.06
-40	91.66	93.05	93.06
-30	86.11	91.66	71.78

จากผลการทดสอบที่ได้ จะพบว่าในบางกรณีภาพที่ผ่านการปรับปรุงแล้ว(Shrink และ Enlarge) ก็ให้ความถูกต้องแม่นยำน้อยกว่าภาพที่ยังไม่ผ่านการปรับปรุง (Original) เพราะว่าใช้หลักการ Scaling ที่ยังมีประสิทธิภาพที่ไม่ดีพอ จากปัญหาที่เกิดขึ้น เราจึงได้เสนอวิธีการ Scaling อีกรูปแบบคือ การเปลี่ยนแปลงวัตถุโดยการหมุน บิด เคลื่อน (Affine Transformation) ที่สามารถปรับขนาดของตัวอักษรโดยการเปลี่ยนแปลงการวัดจากจุดพิกัดตามอุดมคติมาเป็นการใช้งานอย่างแท้จริงโดยการทำให้ตัวแปรต่างๆ เข้าสู่ตัวแปรใหม่ๆ เช่นจากตัวแปรค่าความเข้มของพิกเซลที่ตั้งอยู่ในตำแหน่ง (x1 y 1) ได้อยู่ในตัวแปรใหม่อย่าง (x 2 y 2)โดยมีการปรับใช้ส่วนผสมของ (translation) , (rotation) และ (scaling) หรือ (shearing) ซึ่งหลักการและการทำงานมีดังนี้

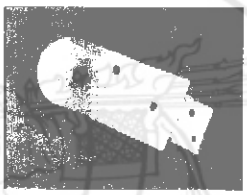
4.6 การเปลี่ยนแปลงวัตถุโดยการหมุน บิด เคลื่อน (Affine Transformation)[6]



รูปที่ 4.5 การเปลี่ยนแปลงวัตถุ โดยการหมุน บิด เคลื่อน ที่อาจจะเรียกได้ว่าเป็นการแปลงพิกัดภาพแอฟไฟร์ (Affine Transformation)

วิธีการทำงาน

คุณสมบัติของการทำงานเกี่ยวกับการเปลี่ยนแปลงวัตถุโดยการหมุน นี้ให้พิจารณาภาพ



รูปที่4.6 ชิ้นส่วนของเครื่องจักรได้แสดงในระนาบด้านหน้าและในแนวขนาน

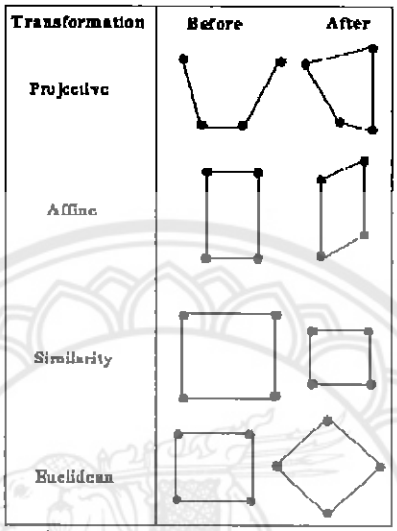
ในขณะที่ชิ้นส่วนของเครื่องจักรได้แสดงในระนาบด้านหน้าและในแนวขนาน หลุมวงกลุ่มของส่วนประกอบด้านบนนี้จะเห็นได้ว่าเป็นรูปวงกลม และเส้นที่มีการขนานและตั้งฉากของโลกนี้จะเป็นเส้นที่ใช้สำหรับการรับรู้เชิงภาพ ทั้งนี้อาจจะมีการสร้างโมเดลของชิ้นส่วนนี้ได้โดยการใช้รูปแบบรูปทรงพื้นฐานในการขึ้นโมเดล (primitive) นี้ก่อน

รูปที่4.7 ชิ้นส่วนของเครื่องจักรได้ภาพที่รูที่เห็นจะได้ว่าเป็นรูปวงรี

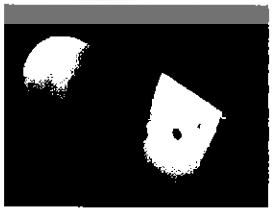
ในภาพด้านบน คือภาพที่รูที่เห็นจะได้ว่าเป็นรูปวงรี และเส้นเชิงตั้งฉากของโลกไม่ได้ถูกมองว่าเป็นเส้นเชิงตั้งฉาก

สำหรับปัญหาเกี่ยวกับมุมมอง (perspective) พบว่าจะแก้ปัญหาที่เกิดขึ้นได้ หากได้มีการอธิบายรูปร่างในฐานะที่มันจะไม่เปลี่ยนแปลงไปตามโปรเจคชั่นแบบเพอร์สเปคทีพ เรื่องราวต่างๆ ที่น่าสนใจเกี่ยวกับวิสัยทัศน์ของคอมพิวเตอร์ที่มีการใช้โมเดลนี้ คือสามารถนำมาใช้ได้โดยต้อง

อธิบายรูปร่างแบบยูคลิด (เช่นการใช้ระยะทางที่สมบูรณ์ มุมและพื้นที่) และมีการใช้การอธิบายที่เกี่ยวข้องกับการวัดเชิงสัมพัทธ์ (เช่นการใช้ความสัมพันธ์เชิงเรขาคณิตภายในเท่านั้น) แต่การใช้การวัดแบบเชิงสัมพัทธ์นี้จะมีการได้มาโดยตรงจากภาพในรูปที่ 1 ซึ่งแสดงลำดับของการเปลี่ยนแปลงเชิงระนาบที่มีความสำคัญต่อมุมมองของคอมพิวเตอร์



รูปที่ 4.8 แสดงลำดับของระนาบ ที่มีการเปลี่ยนแปลงระนาบต่อระนาบจาก Euclidean (โดยที่การหมุนและการแปลความหมายยอมให้มีได้) ต่อ projective (โดยที่สี่เหลี่ยมจัตุรัสสามารถเปลี่ยนเป็นรูปสี่เหลี่ยมโดยทั่วไปได้หากว่าจุดสามจุดไม่ต่อกันเป็นเส้นตรง) หมายเหตุว่าการเปลี่ยนแปลงรูปร่างด้านล่างในตารางจะได้มาจากค่าคงที่ (invariant) ของด้านบนนี้ และเนื่องจากมันได้แสดงกลุ่มในการอธิบาย axiom ที่มีการนิยามเอาไว้แล้ว ดังนั้นการเปลี่ยนแปลงก็จะไม่ได้เป็นความจริง การเปลี่ยนแปลงส่วนหน้าในภาพตัวอย่างด้านบนนี้ได้มีการประมาณโดยการใช้การเปลี่ยนแปลงรูปร่างแบบแอฟไฟร์เชิงระนาบ (เปรียบเทียบกับภาพด้านล่างนี้)



รูปที่ 4.9 รูปร่างแบบแอฟไฟร์เชิงระนาบ

โดยที่ระยะทางไปยังส่วนดังกล่าวนี้ไม่ใหญ่เมื่อเทียบกับกับลึก เส้นวัตถุที่ขนานได้เริ่มต้นที่จะเบนเข้าหากัน (Converge) และเนื่องจากสเกลได้มีความผันแปรไปตามความลึก การอธิบายไปยังระดับของการเปลี่ยนแปลงวัตถุเชิงภาพฉาย (projective transformation) จึงเป็นเรื่องที่จำเป็น ทั้งนี้การเปลี่ยนแปลงวัตถุการเปลี่ยนแปลงวัตถุโดยการหมุน บิด เคลื่อน (Affine Transformation) จะมี

ความสมมูลไปกับผลกระทบที่มาจากการแปลความหมาย (translation) การหมุน (rotation) และการให้สเกลแบบไอโซโทปิก และการเฉือน การการเปลี่ยนแปลงวัตถุโดยการหมุน บิด เคลื่อน (Affine Transformation) โดยทั่วไปจะมีการเขียนได้โดยการใช้จุดพิกัด ดังต่อไปนี้ คือ

$$\begin{vmatrix} x_2 \\ y_2 \end{vmatrix} = A \times \begin{vmatrix} x_1 \\ y_1 \end{vmatrix} + B$$

โดยการนิยามในเมตริกซ์ B การเปลี่ยนแปลงนี้สามารถที่จะทำให้สมบูรณ์ได้โดยการแปลความหมายอย่างแท้จริง (pure translation)

$$A = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}, B = \begin{vmatrix} b_1 \\ b_2 \end{vmatrix}$$

จากการหมุนที่แท้จริง (Pure rotation) โดยการใช้เมตริกซ์ A และได้มีการนิยามเอาไว้ (สำหรับมุมเชิงบวกได้มีการหมุนไปตามเข็มนาฬิกา)

$$A = \begin{vmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{vmatrix}, B = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

ในที่นี้ ได้มีการทำงานในพิกัดของภาพ ดังนั้นแล้วแกน y จะมีการลดลงต่ำ สูตรของการหมุนได้มีการนิยามเอาไว้เมื่อแกน y ได้มีการทำให้สูงขึ้น
ในทางเดียวกันที่การให้สเกลที่แท้จริง (pure scaling) นี้จะได้ว่าเป็น

$$A = \begin{vmatrix} a_{11} & 0 \\ 0 & a_{22} \end{vmatrix}, B = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

(ได้มีการเปลี่ยนแปลงภาพแบบแอฟไฟร์ที่มากมายแตกต่างกันที่บ่อยครั้งได้มีการนำมารวมกันเพื่อสร้างผลลัพธ์โดยรวม ทั้งนี้ค่าของการแปลงที่ได้เกิดขึ้นนี้จะมีนัยสำคัญได้เมื่อการแปลความหมายได้หลังการหมุน ที่ไม่จำเป็นต้องสมมูลไปกันไปในทางกลับกัน) เมื่อมีการเปลี่ยนแปลงรูปแบบแอฟไฟร์โดยทั่วไปแล้วจะมีการนิยามโดยการใช้ constant ทั้งหมด 6 constant ที่มีความเป็นไปได้ในการนิยามการเปลี่ยนแปลงรูปแบบนี้โดยการบ่งชี้ตำแหน่งภาพของเอาท์พุต (x_2 y_2) ของจุดพิกัดของภาพอินพุตสามคู่ใดๆ ($x_1 y_1$) ทั้งนี้แล้วในทางปฏิบัติ ได้มีจุดต่างๆมากมายได้มี

การถูกวัดและวิธีการของกำลังสองน้อยที่สุด (least squares method) จะได้มีการนำมาใช้เพื่อหาการเปลี่ยนแปลงรูปแบบที่เหมาะสมที่สุด

แนวทางปฏิบัติสำหรับการใช้งาน

มีแนวทางมากมายสำหรับการทำงานในเรื่องการบิดหมุน ที่ทำให้ผู้ใช้สามารถนิยามการเปลี่ยนแปลงรูปร่างได้โดยการบ่งชี้ คู่พิกัด 3 อันหรือน้อยกว่านั้นจากภาพอินพุตที่เข้ามา (x_1 , y_1) และมีการ re – map ในภาพเอาท์พุตที่ออกมา (x_2 , y_2) (ที่บ่อยครั้งแล้วในกรณีนี้ ผู้ใช้อาจจะมีการจำกัดให้มีการ re – mapping มุมของตำแหน่งพิกัด (corner coordinates) ของภาพอินพุตเพื่อที่จะ arbitrary new coordinates ในภาพเอาท์พุต) ทั้งนี้เมื่อการเปลี่ยนแปลงรูปแบบได้มีการนิยามเอาไว้แล้วในแนวทางนี้ กระบวนการของ re – mapping นี้ได้ดำเนินการโดยการคำนวณ สำหรับตำแหน่งพิกเซลของเอาท์พุตแต่ละอัน ($x_2 y_2$) และพิกัดที่สอดคล้องของ ($x_1 y_1$) ทั้งนี้แล้วถ้าจุดอินพุตนี้อยู่นอกของภาพแล้ว จากนั้นพิกเซลของเอาท์พุตจะถูกกำหนดให้ไปอยู่ในค่าของพื้นหลัง มิฉะนั้นแล้วค่า ของ (i) ของพิกเซลอินพุต (ii) ใกล้เคียงกับตำแหน่งของพิกเซลที่ต้องการหรือ (iii) การประมาณค่าความสัมพันธ์เชิงเส้นคู่ของพิกเซลข้างเคียง 4 อันจะถูกใช้

การแสดงการปฏิบัติการของการเปลี่ยนแปลงวัตถุโดยการหมุน บิด เคลื่อน (Affine Transformation) โดยการประยุกต์ใช้การเปลี่ยนแปลงรูปแบบกับกรณีที่พิเศษ (เช่นการใช้การแปลความหมาย การหมุนและการกำหนดสเกลที่แท้จริง) และจากนั้นการเปลี่ยนแปลงโดยทั่วไปที่เกี่ยวข้องกับส่วนประกอบของสิ่งเหล่านี้จะนำมาใช้

เริ่มต้นด้วยภาพแบบไบนารี่ (binary artificial image) ขนาด 256×256

รูปที่ 4.10 ภาพแบบไบนารี่ ขนาด 256×256

สามารถใช้การแปลความหมาย (translation) ได้โดยการใช้ปฏิบัติการด้วยเทคนิคแอฟไพร์ เพื่อให้ได้รูปดังต่อไปนี้คือ

รูปที่ 4.11 ภาพการแปลความหมาย (translation) ได้โดยการใช้ปฏิบัติการด้วยเทคนิคแอฟไพร์

เพื่อที่จะทำการแปลความหมายอย่างแท้จริง (pure translation) ได้นิยามการแปลงรูปแบบโดยการ re mapping จุดเชิงเดี่ยว (ตัวอย่างเช่น อินพุตของภาพด้านล่าง – มุมด้านซ้าย (0 , 0) ให้เป็นตำแหน่งใหม่ที่ (64 , 64)

สำหรับการหมุนที่แท้จริง (pure rotation) นี้ต้องการ การ re mapping ของตำแหน่งของสองมุมไปยังตำแหน่งใหม่ ถ้ามีการบ่งชี้ให้มุมด้านล่างซ้ายเคลื่อนย้ายไปที่ (256 , 0) และมุมด้านล่างขวาเคลื่อนย้ายไปที่ (256 , 256) จะได้ว่า

รูปที่ 4.12 ภาพแสดงการ re mapping ของตำแหน่งของสองมุม

ในทางเดียวกัน การสะท้อน (reflection) จะสำเร็จได้โดยการ swapping จุดพิกัดของมุมตรงข้ามทั้งสองซึ่งจะดังที่แสดงในรูปด้านล่างนี้

รูปที่ 4.13 ภาพการ swapping จุดพิกัดของมุมตรงข้ามทั้งสอง

การกำหนดสเกลสามารถประยุกต์ได้โดยการ re mapping เพียงแค่สองมุม , สำหรับตัวอย่างพวกเราสามารถส่งมุมซ้ายล่างไปยัง (64 , 64) ในขณะที่มีการสลักให้มุมขวาบนได้ลงมาที่ (256 , 256) และดังนั้นจะมีการหดขนาดของภาพ ดังที่ได้แสดงในรูปด้านล่าง

รูปที่ 4.14 การหดขนาดของภาพโดยการ re mapping เพียงแค่สองมุม

ที่นี้ได้มีการแปลความหมายภาพด้วย และการ Re-mapping สองจุดจะนำไปสู่การรวมกันของการแปลความหมาย การหมุนและการกำหนดสเกล

การเปลี่ยนแปลงรูปแบบแบบแอฟไฟร์โดยทั่วไปนี้มีการเฉพาะเจาะจงโดยจุดสามจุดที่มีการ re-mapping ทั้งนี้ถ้าได้มีการ re- mapping ภาพอินพุตดังนั้นแล้วจะมีการเคลื่อนย้ายมุมซ้ายล่างขึ้นไปที่ (64 , 64) และไปตามแกนแนวเฉียงที่ระดับ 45 องศา และมีการเคลื่อนย้ายมุมด้านขวาบนลงมาใน

แกนเดียวกันนี้และมีการปักให้มุมด้านทางขวาล่างอยู่ในที่ที่ต้องการ จากนั้นจะได้ภาพที่แสดงถึงผลกระทบจากการเฉือน (shearing effect) เส้นแนวขนานยังคงเป็นเส้นขนาน แต่มุมตั้งฉากจะมีการบิดเบี้ยวไป

รูปที่ 4.15 แสดงถึงผลกระทบจากการเฉือน(shearing effect) เส้นแนวขนานยังคงเป็นเส้นขนาน แต่มุมตั้งฉากจะมีการบิดเบี้ยวไป

การเปลี่ยนแปลงรูปภาพโดยวิธีการหมุน บิดนี้ได้มีการนำมาใช้บ่อยในกรณีที่ได้มีภาพที่ตรวจจับแล้วพบว่าเคยผ่านการบิดหมุนมาก่อน อย่างไรก็ตามแล้วเวอร์ชั่นที่ถูกต้องของภาพอินพุตนี้จะได้มาจากการเปลี่ยนภาพแบบแอฟไฟร์นี้โดยการ re -sampling ภาพอินพุต เช่นข้อมูล (หรือความเข้ม) ที่แต่ละจุด (x_1 , y_1) ที่มีการให้อยู่ในตำแหน่งที่ถูกต้องและสอดคล้องกับภาพเอาท์พุต ($x_2 y_2$)

สิ่งหนึ่งที่น่าสนใจมากกว่าในการใช้เทคนิคดังกล่าวนี้คือเรื่องของรีโมทเซ็นเซอร์ทางไกล อย่างไรก็ตามแล้วเนื่องจากภาพส่วนมากได้มีการเปลี่ยนแปลงไปก่อนที่จะมีการจัดเตรียมไปยังกระบวนการที่เกี่ยวกับภาพ จะแสดงให้เห็นว่าการเปลี่ยนแปลงแบบแอฟไฟร์พร้อมกับ terrestrial image

รูปที่ 4.16 การเปลี่ยนแปลงแบบแอฟไฟร์พร้อมกับ terrestrial image

ที่มี contrast-stretched (cutoff fraction = 0.9) จะได้ว่า

รูปที่ 4.17 การเปลี่ยนแปลงแบบแอฟไฟร์พร้อมกับ terrestrial image ที่มี contrast-stretched (cutoff fraction = 0.9)

อาจจะต้องการเปลี่ยนแปลงภาพนี้เพื่อที่จะให้กรอบของประตูได้เข้ามาอยู่ข้างใน กรอบสี่เหลี่ยม ทั้งนี้จะทำมันได้โดยการนิยามการเปลี่ยนแปลงรูปร่างบนพื้นฐานของการ re – mapping ของ (i) มุมด้านบนขวาไปยังตำแหน่งที่ 30 % ที่ต่ำกว่าแกน y) , (ii) มุมทางด้านขวาล่างให้อยู่ตำแหน่งที่ 10 % ที่ต่ำกว่าแกน x และ (iii) มีการปักหมุด (pinning) ลงที่มุมด้านซ้ายบน และผลที่ได้คือ

รูปที่ 4.18 การเปลี่ยนแปลงรูปร่างบนพื้นฐานของการ re – mapping

ได้มีการนิยามการเปลี่ยนแปลงรูปร่างที่ทำงานได้ดีเอาไว้สำหรับวัตถุ ตรงที่ความลึกของกรอบประตู แต่ว่าวัตถุที่ใกล้เคียงนี้ได้มีการบิดหมุนไปเนื่องจากการเปลี่ยนแปลงเชิงระนาบแบบแอฟไฟร์ ไม่สามารถตรวจสอบได้สำหรับการบิดหมุนที่ความลึกที่แตกต่างกัน

บทที่ 5
สรุปผล

โครงงานนี้เป็นการศึกษาเพื่อแก้ข้อจำกัดของการรู้จำตัวอักษรและตัวเลขในภาพถ่ายที่ถ่ายจากมุมสูงกว่าระดับสายตาและภาพถ่ายที่ถ่ายจากมุมต่ำกว่าระดับสายตา ซึ่งปัญหาที่พบ คือ ความลึกและมุมของภาพ ทำให้การรู้จำยังทำงานได้ไม่เต็มประสิทธิภาพ จึงต้องมีการพัฒนาซอฟแวร์ที่ช่วยในการรู้จำเพื่อให้เกิดการทำงานที่มีประสิทธิภาพสูงสุด

ในการพัฒนาโปรแกรมนี้ผู้เขียนได้ใช้เทคนิค การแปลงภาพให้เป็นขาวดำ การแยกแยะบริเวณในภาพ การแก้ไขภาพในเชิงเรขาคณิต และ OpenCV ซึ่งเครื่องมือที่นำมาใช้ในการรู้จำตัวอักษร อนึ่งซอฟแวร์ที่ใช้ในการพัฒนาโปรแกรมนั้น กลุ่มผู้ทำโครงงานได้ใช้ภาษา C++ ในการพัฒนาโปรแกรม เพราะเป็นภาษาที่ผู้พัฒนาคิดว่าเป็นภาษาที่ง่ายต่อการนำไปปรับเปลี่ยนเพื่อเป็นการต่อยอดโครงงาน

การทำงานของโปรแกรมจะเริ่มจากการรับภาพแล้วทำการปรับปรุงรูปภาพ โดยการเปลี่ยนภาพให้เป็นไบนารี่แล้วทำการลดสิ่งรบกวน เพื่อทำการรู้จำโดยจะใช้หลักการย่อและขยายบรรทัดตัวอักษรเพื่อวัดประสิทธิภาพการรู้จำว่าดีขึ้นกว่าภาพที่ยังไม่ได้ทำการปรับปรุงหรือไม่

เมื่อผ่านการทดสอบการทำงานของโปรแกรมภาพที่ผ่านการ Shrink ให้ค่าเฉลี่ยความถูกต้องมากกว่าภาพที่ผ่านการ Enlarge และภาพที่ยังไม่ได้ผ่านการปรับปรุงภาพอย่างเห็นได้ชัด

จากการทดลองจะได้ว่า การปรับปรุงภาพก่อนทำการรู้จำจะให้ค่าความถูกต้องที่สูงกว่า เมื่อมุมมองของการถ่ายภาพอยู่ในช่วงมุมระหว่าง 30 ถึง 60 องศา และ -30 ถึง -60 องศา แต่การย่อบรรทัดตัวอักษรจะให้ค่าความถูกต้องที่สูงกว่าการขยายบรรทัดตัวอักษร

5.1 สรุปผลการทดลอง

จากการทดสอบโปรแกรม สามารถทำการรู้จำตัวอักษร ซึ่งใช้การประมวลผลภาพดิจิทัลในการแก้ปัญหาได้เป็นอย่างดีโดยในการทดสอบด้วยการรู้จำภาพทะเบียนรถยนต์นั้น โปรแกรมสามารถทำการรู้จำได้ดีเมื่อมุมในการถ่ายภาพอยู่ระหว่าง 30 ถึง 60 องศาและ –30 ถึง -60 องศา แต่การปรับปรุงภาพก็เริ่มให้ความถูกต้องแม่นยำน้อยลง เมื่อระยะการถ่ายภาพมากยิ่งขึ้น

เมื่อพิจารณาผลการทดสอบจะพบว่าที่การทดสอบโดยใช้ระยะห่างเป็นเกณฑ์ ภาพ Original ภาพที่ผ่านหลักการ Shrink และ Enlarge ให้ผลการทดสอบออกมาที่ประสิทธิภาพเท่าๆกัน

แต่เมื่อนำมุมในการถ่ายมาพิจารณาจากกรณีจำนวน 13 กรณีที่ใช้ในการทดสอบจะได้ผลการทดสอบออกมาคือ

- ภาพ Original ให้ค่าเฉลี่ยที่ดีกว่าจำนวน 5 กรณี
- ภาพที่ผ่านหลักการ Shrink ให้ค่าเฉลี่ยที่ดีกว่าจำนวน 6 กรณี
- ภาพที่ผ่านหลักการ Enlarge ให้ค่าเฉลี่ยที่ดีกว่าจำนวน 3 กรณี

ซึ่งในมุมการทดสอบที่ -60 องศา ภาพ Original และภาพที่ผ่านหลักการ Shrink ให้ผลการทดสอบออกมาในค่าเฉลี่ยที่เท่ากัน จะเห็นได้ว่าภาพที่ผ่านการปรับปรุงภาพให้ค่าเฉลี่ยการทดสอบออกมาสูงกว่าภาพที่ยังไม่ผ่านการปรับปรุง แต่อาจจะมีบางกรณีที่ผลการทดสอบกับภาพ Original ให้ผลการทดสอบที่ดีกว่าภาพที่ได้ผ่านการปรับปรุงภาพ ซึ่งเกิดจากการปรับขนาดของตัวอักษรในภาพยังใช้การ Scaling ที่ยังมีประสิทธิภาพที่ยังไม่ดีพอ

5.2 ปัญหาและอุปสรรค

ตารางที่ 5.1 ตารางปัญหาและอุปสรรค

ปัญหาและอุปสรรค	แนวทางการแก้ไข
1. เนื่องจากข้อจำกัดในส่วนการจอง Memory ของการเก็บข้อมูลประเภท Array มีข้อจำกัดทางด้านขนาดทำให้หน่วยความจำเต็ม	-เปลี่ยนไปเก็บข้อมูลเป็นประเภท Linklist
2. สืบเนื่องจากปัญหาข้อแรก เมื่อเปลี่ยนมาใช้การเก็บข้อมูลแบบ Linklist ทำให้การประมวลผลข้อมูลช้า	-เปลี่ยนประเภทการรันโปรแกรมจาก Debug เป็น Release
3. ภาพที่ถ่ายจากป้ายทะเบียนรถ จะมีปัญหาเรื่องแสงที่สะท้อนเข้ามาทำให้เมื่อทำการเปลี่ยนภาพจากปกติเป็นภาพไบนารี่ เกิดการขาดหายของตัวอักษรทำให้ภาพไม่สมบูรณ์	-เลือกมุมที่ใช้การถ่ายภาพให้เหมาะสม โดยจะทำการถ่ายภาพจากหลายๆมุม หลายความเข้มของแสง
4. ขาดความชำนาญในการเขียนโปรแกรมด้านภาษา C++	-ศึกษาภาษา C++ เพิ่มเติม

5.3 ข้อเสนอแนะ

1. การพัฒนาโปรแกรมอาจจะใช้ภาษาอื่น ที่ไม่ใช่ C++ ก็ได้ หากผู้พัฒนามีความรู้ความเข้าใจในภาษาอื่น
2. อาจทำการพัฒนาอัลกอริทึมในการย่อขยายตัวอักษร เพื่อให้ค่าความถูกต้องในการรู้จำภาพสูงขึ้น
3. สามารถเปลี่ยนโปรแกรม Tesseract-OCR ที่นำมาใช้ในการรู้จำตัวอักษรเป็นโปรแกรมอื่นที่มีประสิทธิภาพสูงกว่า

5.4 แนวทางในการพัฒนาต่อ

1. เพิ่มขอบเขตของการรู้จำ ในส่วนของ จังหวัดในป้ายทะเบียนรถ และพัฒนา Tesseract-OCR ให้รู้จำในส่วนของสระและวรรณยุกต์ได้
2. เพิ่มขอบเขตทิศทางของภาพที่จะนำมารู้จำโดยไม่จำกัดเฉพาะทิศมุมมองที่สูงกว่าระดับสายตาและระดับต่ำกว่าสายตา

5.5 ข้อแนะนำสำหรับผู้ที่ต้องการพัฒนาต่อยอดโครงงาน

1. ผู้ที่จะทำการพัฒนาควรมีความรู้ในด้านการเขียนโปรแกรมด้วยภาษา C++
2. ทำความเข้าใจ Tool ที่จะใช้เกี่ยวกับ Image Processing และ Tesseract OCR
3. ทำการหาความรู้ในเรื่อง DataStructure และเข้าใจการออกแบบโปรแกรมในเชิงเรขาคณิต
4. ทำการศึกษาเพิ่มเติมเรื่องการแปลงวัตถุโดยการหมุน บิด เคลื่อน (Affine Transformation)

ภาคผนวก ก.

OpenCV Library

โครงสร้างของข้อมูลใน OpenCV

1. IplImage

เป็นโครงสร้างที่ใช้สำหรับการเก็บข้อมูลของภาพและข้อมูลพิกเซล(Pixel)ของภาพ จะประกอบด้วยโครงสร้างดังนี้

```
typedef struct_ IplImage
{
        Int nSize;
        Int ID;
        Int nChannels;
        Int alphaChannel;
        Int depth;
        char colorModel[4];
        char channelSeq[4];
        int dataOrder;
        int origin;
        int align;
        int width;
        int height;
        struct_IplROI *roi;
        struct_IplImage *maskROI;
        void *imageId;
        struct_IplTileInfo *tileInfo;
        int imageSize;
        char *imageData;
        int widthStep;
        int BorderMode[4];
        int BorderConst[4];
```

```
    char *imageDataOrigin;
    }
    IplImage;
```

nSize ขนาดของข้อมูล

nChannel จำนวนชั้นของสีในรูปภาพ มีตั้งแต่ 1 - 4 ชั้น

depth ความลึกบิตของแต่ละพิกเซลซึ่งจะประกอบด้วย

- IPL_DEPTH_8U แต่ละพิกเซลเป็นจำนวนเต็มขนาด 8 บิต ไม่คิดเครื่องหมาย
- IPL_DEPTH_8S แต่ละพิกเซลเป็นจำนวนเต็มขนาด 8 บิต คิดเครื่องหมาย
- IPL_DEPTH_16U แต่ละพิกเซลเป็นจำนวนเต็มขนาด 16 บิต ไม่คิดเครื่องหมาย
- IPL_DEPTH_16S แต่ละพิกเซลเป็นจำนวนเต็มขนาด 16 บิต คิดเครื่องหมาย
- IPL_DEPTH_32S แต่ละพิกเซลเป็นจำนวนเต็มขนาด 32 บิต คิดเครื่องหมาย
- IPL_DEPTH_32F แต่ละพิกเซลเป็นจำนวนทศนิยมขนาด 32 บิต
- IPL_DEPTH_64F แต่ละพิกเซลเป็นจำนวนทศนิยมขนาด 64 บิต

Origin เป็นค่าที่บอกจุดเริ่มต้นของภาพว่าจะอยู่ที่มุมบนซ้าย (Origin = 0) หรือมุมล่างซ้าย (Origin = 1)

Width ความกว้างของภาพในหน่วยพิกเซล

Height ความสูงของภาพในหน่วยพิกเซล

ImageSize ขนาดของรูปภาพที่ได้จาก width * hight

ImageData เป็นตัวแปลที่ใช้เก็บค่าตำแหน่งเริ่มต้นในหน่วยความจำที่เก็บข้อมูลรูปภาพ

widhtStep ขนาดของข้อมูลแต่ละแถวในรูปภาพในหน่วยไบท์ (byte)

ฟังก์ชันพื้นฐานใน OpenCV

1. cvCreateImage

ทำหน้าที่จองขนาดของหน่วยความจำให้รูปภาพหรือว่าข้อมูลต่างที่ต้องการมีโครงสร้าง คือ

```
IplImage *cvCreateImage(CvSize, int depth, int channel);
```

Size ขนาดของรูปภาพที่ต้องการในหน่วยของพิกเซล

depth ความลึกของบิตของแต่ละพิกเซล

- IPL_DEPTH_8U แต่ละพิกเซลเป็นจำนวนเต็มขนาด 8 บิต ไม่คิดเครื่องหมาย
- IPL_DEPTH_8S แต่ละพิกเซลเป็นจำนวนเต็มขนาด 8 บิต คิดเครื่องหมาย
- IPL_DEPTH_16U แต่ละพิกเซลเป็นจำนวนเต็มขนาด 16 บิต ไม่คิดเครื่องหมาย
- IPL_DEPTH_16S แต่ละพิกเซลเป็นจำนวนเต็มขนาด 16 บิต คิดเครื่องหมาย

- IPL_DEPTH_32S แต่ละพิกเซลเป็นจำนวนเต็มขนาด 32 บิต คิดเครื่องหมาย

- IPL_DEPTH_32F แต่ละพิกเซลเป็นจำนวนทศนิยมขนาด 32 บิต

- IPL_DEPTH_64F แต่ละพิกเซลเป็นจำนวนทศนิยมขนาด 64 บิต

Channel จำนวนสีของรูปภาพ

2. cvLoadImage

ทำหน้าที่ดึงข้อมูลรูปภาพจากอุปกรณ์เก็บข้อมูล (Storage) มีโครงสร้าง คือ

cvLoadImage(string path);

path ตำแหน่งของรูปภาพในอุปกรณ์เก็บข้อมูล

3. cvReleaseImage

ทำหน้าที่คืนค่าหน่วยความจำที่เก็บข้อมูลภาพสู่ระบบ มีโครงสร้าง คือ

cvReleaseImage(IplImage **image);

image ภาพที่ต้องการคืนหน่วยความจำ

4. cvCvtColor

ทำหน้าที่แปลงรูปภาพจากโหมดหนึ่งไปสู่โหมดอื่นๆมีโครงสร้าง คือ

Void cvCvColor(const CvArr* src, CvArr* dst, int code);

Srcรูปภาพที่จะทำการแปลงโดยต้องมีความลึกของบิตของชั้นสีขนาด 8 บิต (8U), 16 บิต (16U) และจำนวนทศนิยม 32 บิต (32F)

dst รูปภาพที่ได้จากการแปลงต้องมีชนิดของขนาดเหมือน src แต่จำนวนของแชลแนล (Channel) อาจเปลี่ยนแปลง

code รูปแบบคำสั่งในการแปลงโหมดภาพนั้น จะใช้ CV_<โหมดเริ่มต้น>2<โหมดที่ต้องการ>เช่น CV_RGB2GRAY คือการเปลี่ยนแปลงภาพจากแบบ RGB เป็นภาพแบบ Gray-Scale โดยสำหรับภาพแบบ RGB นั้นจะใช้ 3 แชลแนลในการเก็บข้อมูลรูปแต่ละพิกเซล ดังนี้

R คือ ค่าของสีแดงในพิกเซล

G คือ ค่าของสีเขียวในพิกเซล

B คือ ค่าของสีน้ำเงินในพิกเซล

ภาพแบบ Gray-Scale ใช้เพียง 1 แชลแนลในการเก็บข้อมูลรูปแต่ละพิกเซลจึงต้องมีการทำการคำนวณเพื่อเปลี่ยนค่าต่างๆ คือ

$$Y = (0.299 * R) + (0.587 * G) + (0.114 * B)$$

เมื่อค่า Y คือค่าของพิกเซลใน Gray-Scale

5. cvAdaptive Threshold

Adaptive Threshold เป็นฟังก์ชันที่ทำการเปลี่ยนภาพที่ต้องการให้อยู่ในลักษณะแบบ Binary มีโครงสร้างคือ

Void cvAdaptive Threshold (const CvArr* src, CvArr* dst, double max_value, int adaptive_method_type, int block_size, double param);

src รูปภาพที่ต้องการจะทำการแปลงเป็นภาพแบบ Binary

dst รูปภาพแบบ Binary ที่ได้

max_value ค่ามากที่สุดที่ถูกใช้กับ CV_THRESH_BINARY และ CV_THRESH_BINARY_INV

adaptive_method อัลกอริทึมที่ใช้ในการทำ Adaptive Threshold คือ

CV_ADAPTIVE_THRESH_MEAN_C หรือ CV_ADAPTIVE_THRESH_GAUSSIAN_C

threshold_type ชนิดของวิธีในการทำ Threshold ซึ่งต้องเลือกระหว่าง

- CV_THRESH_BINARY
- CV_THRESH_BINARY_INV

ซึ่งข้อแตกต่างระหว่างทั้งสองแบบคือ

- CV_THRESH_BINARY จะให้ค่าของพิกเซลเป็นสีขาวถ้าพิกเซลนั้นมีค่ามากกว่า threshold ถ้าน้อยกว่าหรือเท่ากับค่าของพิกเซลนั้นจะเป็นสีดำ
- CV_THRESH_BINARY_INV จะให้ค่าของพิกเซลนั้นเป็นสีดำถ้าพิกเซลนั้นมีค่ามากกว่า threshold ถ้าน้อยกว่าหรือเท่ากับค่าของพิกเซลนั้นจะเป็นสีขาว

Block_size ขนาดของพิกเซลรอบๆที่ใช้ในการคำนวณ โดยมีค่าตั้งแต่ 3,5,7,....

Paraml เป็นค่าที่ใช้ในการคำนวณค่าเฉลี่ยหรือค่าน้ำหนัก โดยมีค่าเริ่มต้นที่ 3,5,7,...

6. cvSmooth

ทำหน้าที่ในการปรับปรุงรูปภาพให้ชัดเจนขึ้น โดยมีโครงสร้างดังนี้

Void cvSmooth(const CvArr* src,CvArr* dst, int smoothtype, int param1, int param2, double param3);

Src ภาพที่ต้องปรับปรุง

Dst ภาพที่ได้หลังจากการปรับปรุง

Smoothtype วิธีในการปรับปรุง โดยสำหรับวิธีการปรับปรุงโดยใช้ค่ากลางของข้อมูล (Median Filter) ค่าของ smoothtype จะเท่ากับ CV_MEDIAN โดยขนาดของบล็อกจะเท่ากับค่าของ param1

Param1, param2, param3 ตัวแปลที่ใช้ในการคำนวณวิธีต่างๆ

ภาคผนวก ข.

Tesseract OCR

Tesseract OCR เป็นโปรแกรมสำหรับทำการรู้จำตัวอักษรแบบ OpenSource ที่พัฒนาขึ้นโดยบริษัท HP LAB ในปี ค.ศ. 1985-1995 และมีการนำมาพัฒนาโดย Google โดย Tesseract OCR สามารถปฏิบัติการบนระบบปฏิบัติการ ได้ ดังนี้ คือ

- Ubantu 6.06 (x86/32,x86/64)
- Ubantu 6.10 (x86/32,x86/64)
- Windows(x86/32)

โดยปัจจุบันนี้เวอร์ชั่นล่าสุด สามารถทำการรู้จำได้ ถึง 6 ภาษา คือ ภาษาอังกฤษ,ภาษาฝรั่งเศส, ภาษาอิตาเลียน, ภาษาเยอรมัน, ภาษาสเปน, ภาษาเนเธอร์แลนด์ และสามารถพัฒนาเพื่อนำไปทำการรู้จำตัวอักษรในภาษาต่างได้ต่อไป

คำสั่งที่ใช้ทำการรู้จำ คือ

Tesseract.exe <ไฟล์รูปภาพ> <ไฟล์ผลการทดลองที่ได้จากการรู้จำ> [-l <ภาษาที่ต้องการทำการรู้จำ>] ดังนี้

```
Tesseract Test.tif Test-OCR –l eng
```

ภาคผนวก ค.

การเทรนภาษา

1. สร้างไฟล์ ภาพ.tif

บันทึกไว้ใน Foder ของโปรแกรม Tessetact อยู่

2. สร้าง Box file เข้า cmd

พิมพ์ tesseract ชื่อไฟล์ภาพ.tif ชื่อไฟล์ภาพ batch.nochop makebox จะได้ไฟล์ .box

ใช้โปรแกรม cowboxer แก้ box ไฟล์(มีลิ้งให้โหลดตรง http://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract3 หัวข้อ Box file editors)

วิธีเข้าโปรแกรม เปิด box ไฟล์

-แก้ตัวอักษรให้ตรงกับภาษาไทยโดย คลิกที่ตัวอักษรกด backspace แล้วพิมพ์แก้ได้เลย เสร็จแล้ว save

-ทำให้ครบทุกตัวอักษร แล้ว save

3. run tesseract for training

พิมพ์ tesseract ชื่อไฟล์.tif ชื่อไฟล์ nobatch box.train ถ้าไม่เจอ error จะได้ไฟล์ .tr

4. ก๊อบปี้ไฟล์ .tr กับ .box นำไปไว้ในโฟล์เดอร์ training

5. compute the character set ใน cmd เข้าไปใน training พิมพ์ unicharset_extractor ชื่อไฟล์.box จะได้ไฟล์ unicharset

6. Clustering

พิมพ์ mftraining ชื่อไฟล์.tr จะได้ไฟล์ inttemp, mfunicharset, Microfeat, pffmtable พิมพ์ cntraining ชื่อไฟล์.tr จะได้ไฟล์ normproto

7. รวมไฟล์

แก้ชื่อไฟล์ 6 ไฟล์โดยเติม tha. เข้าไป ให้เป็นดังต่อไปนี้

tha.inttemp, tha.normproto, tha.pffmtable, tha.unichaset ,tha.mfunicharset, tha.Microfeat

ต่อไปพิมพ์ combine_tessdata tha. จะได้ไฟล์ tha.traineddata ก๊อบปี้ไปไว้ในโฟล์เดอร์ tessdata เรียกใช้งานได้ตามปกติ

เอกสารอ้างอิง

[1] Rafael C. Gonzalez and Richard E. Woods, **Digital image processing,** Prentice Hall, 2002

[2] [online] **"Transformation"** fivedots.coe.psu.ac.th/~montri/Teaching/image/transform.

[3] [online] **"OpenCVWiki"** http://opencv.willowgarage.com

[4] Gregory A. Baxes, **Digital image processing : principles and applications**, John Wiley & son, 1994

[5] [online] **"TrainingTesseract"** http://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract3

[6] [online] **"Affine"** http://homepage.inf.ed.ac.uk/rbf/HIPR2/affine.htm

[7] **"Starting Out with C++"** / Tony Gaddis and Barret Krupnow. - - Brief version,4th ed. Update p.cm, 2006

[8][online] **"Flood Fill"** http://www2.cs.science.cmu.ac.th

ประวัติผู้จัดทำโครงงาน

ชื่อ นายชานนท์ วงศ์ประเสริฐ

ภูมิลำเนา 87 หมู่ 5 ต.แม่สุก อ.แม่ใจ จ.พะเยา 56130

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนแม่ใจวิทยาคม จังหวัดพะเยา
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 6
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail: aaacha@hotmail.com

ชื่อ นางสาวรุ่งนภา ยาณะเรือง

ภูมิลำเนา 107 หมู่ 9 ต. ป่าแฝก อ. แม่ใจ จ. พะเยา 56130

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนแม่ใจวิทยาคม จังหวัดพะเยา
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 6
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail: legend_demon@hotmail.com