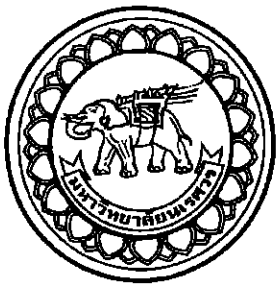


การรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกัน จากการถ่ายภาพแบบด้านข้าง

Recognizing Character with Difference Depth Perspectives from a Side shot Picture.

นายสิระ เสาวลักษณ์ รหัส 48362513

ห้องสมุดคณะวิศวกรรมศาสตร์
5 เม.ย. 2553
เลขหมู่ ส729 ก 2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2551

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน การรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้าง

ผู้ดำเนินโครงงาน นายสิระ เสาวลักษณ์ รหัส 48362513

อาจารย์ที่ปรึกษา ดร.สุรเดช จิตประไพกุลศาล

อาจารย์ที่ปรึกษาร่วม ดร.พนมขวัญ ริยะมงคล

ดร.ไพศาล มุณีสว่าง

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2551

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ

(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ

(ดร.ไพศาล มุณีสว่าง)

หัวข้อโครงงาน	การรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้าง
ผู้ดำเนินโครงงาน	นายสิระ เสาวลักษณ์ รหัส 48362513
อาจารย์ที่ปรึกษา	ดร. สุรเดช จิตประไพกุลศาล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2551

บทคัดย่อ

โครงงานนี้ได้พัฒนาโปรแกรมเพื่อทำการรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้าง โดยในขั้นต้นโปรแกรมจะทำการค้นหาบรรทัดตัวอักษร แล้วจึงนำภาพบรรทัดตัวอักษรที่ได้ไปทำการปรับปรุงรูปภาพก่อนการรู้จำ จากนั้นทำการรู้จำภาพบรรทัดตัวอักษรด้วยโปรแกรม Tesseract OCR จากผลการทดลองพบว่า โปรแกรมสามารถทำการรู้จำตัวอักษรได้ถูกต้องกว่าการใช้โปรแกรมทำการรู้จำตัวอักษรโดยตรง โดยเฉพาะเมื่อมุมในการถ่ายภาพเกินกว่า 10 องศา

Project Title	Recognizing Character with Difference Depth Perspectives from a Side shot Picture.
Name	Mr.Sira Saowaluk ID. 48362513
Project Advisor	Suradet Jitprapaikulsarn, Ph.D.
Major	Computer Engineering.
Department	Electrical and Computer Engineering.
Academic Year	2008

ABSTRACT

This project develops a program for recognizing characters from a side shot picture. First, the program searches for the line containing alphabets. Next, it manipulates the image so that all characters have the same scale. Finally, it uses Tesseract OCR to recognize characters from the modified image. Based on our experiments, this program performs significantly better recognition than using only Tesseract OCR, especially, when the camera angle is more than 10 degrees.

กิตติกรรมประกาศ

ขอขอบคุณ ดร. สุรเดช จิตประไพกุลศาล ที่คอยให้คำปรึกษา ความช่วยเหลือตลอดจนคำแนะนำและแนวทางต่างๆ ในการทำโครงงานนี้ ขอขอบคุณอาจารย์ทุกท่านและเพื่อนๆ ที่คอยให้การสนับสนุนผู้ดำเนินโครงงาน ให้สามารถทำโครงงานนี้จนสำเร็จลุล่วงไปได้ด้วยดี

สิระ เสาวลักษณ์

สารบัญ

หน้า

บทคัดย่อ ..... ก

ABSTRACT ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูปภาพ ..... ช

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบข่ายของโครงงาน 2

1.4 ขั้นตอนของการดำเนินงาน 2

1.5 แผนผังการดำเนินโครงงาน 3

1.6 ผลที่คาดว่าจะได้รับ 3

1.7 งบประมาณของโครงการ 4

บทที่ 2 ทฤษฎีพื้นฐาน 5

2.1 การประมวลรูปดิจิตอล (Digital Image Processing) 5

2.2 โอเพ่นซีวี ไลบรารี (OpenCV Library) 9

2.3 การถดถอยเชิงเส้น (Linear Regression) 10

2.4 หลักการของการรู้จำตัวอักษร 11

2.5 ปัญหาที่เกิดจากการถ่ายภาพแบบด้านข้าง 13

บทที่ 3 การออกแบบและพัฒนาระบบ 14

3.1 การค้นหาบรรทัดตัวอักษร 14

3.2 การปรับปรุงรูปภาพก่อนทำการรู้จำ 20

3.3 การรู้จำตัวอักษร 22

สารบัญ (ต่อ)

หน้า

บทที่ 4 การออกแบบซอฟต์แวร์ 23

4.1 ความต้องการของระบบ (Requirement Specification) 23

4.2 ขอบเขตของระบบ 23

4.3 การออกแบบซอฟต์แวร์ 24

บทที่ 5 ผลการทดลอง 30

5.1 การทดสอบการรู้จำด้วยภาพถ่ายเอกสาร 30

5.2 การทดสอบการรู้จำด้วยภาพทะเบียนรถยนต์ 65

5.3 วิเคราะห์ผลการทดลอง 98

บทที่ 6 บทสรุป 100

6.1 หน้าที่การทำงานของโปรแกรม 100

6.2 วิเคราะห์ผลการทดลอง 100

6.3 ปัญหาและแนวทางการแก้ไข 101

6.4 สรุปผลการทดลอง 102

6.5 ข้อเสนอแนะ 102

เอกสารอ้างอิง 103

ภาคผนวก ก. 104

ภาคผนวก ข. 110

สารบัญตาราง

ตารางที่ หน้า

5.1 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 10 เซนติเมตร 32
5.2 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 20 เซนติเมตร 34
5.3 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 30 เซนติเมตร 36
5.4 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 40 เซนติเมตร 38
5.5 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 50 เซนติเมตร 40
5.6 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 20 เซนติเมตร 42
5.7 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 30 เซนติเมตร 44
5.8 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 40 เซนติเมตร 46
5.9 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 50 เซนติเมตร 48
5.10 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 60 เซนติเมตร 50
5.11 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 20 เซนติเมตร 52
5.12 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 30 เซนติเมตร 54
5.13 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 40 เซนติเมตร 56
5.14 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 50 เซนติเมตร 58
5.15 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 60 เซนติเมตร 60
5.16 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ 62
5.17 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ 63
5.18 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับขนาดของตัวอักษรในการถ่ายภาพ 64
5.19 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 50 เซนติเมตร 67
5.20 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 75 เซนติเมตร 69
5.21 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 100 เซนติเมตร 71
5.22 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 125 เซนติเมตร 73
5.23 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 150 เซนติเมตร 75
5.24 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 175 เซนติเมตร 77

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

5.25 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 200 เซนติเมตร 79
5.26 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 50 เซนติเมตร 81
5.27 การรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 75 เซนติเมตร 83
5.28 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 100 เซนติเมตร 85
5.29 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 125 เซนติเมตร 87
5.30 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 150 เซนติเมตร 89
5.31 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 175 เซนติเมตร 91
5.32 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 200 เซนติเมตร 93
5.33 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ 95
5.34 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ 96

สารบัญรูปภาพ

รูปที่ หน้า

2.1 จุดเริ่มต้นในการ Flood Fill 5
2.2 ภาพที่ได้จากการทำ Flood Fill 4 ทิศทาง 6
2.3 ภาพที่ได้จากการทำ Flood Fill 8 ทิศทาง 6
2.4 ภาพที่ได้จากการลดสิ่งรบกวนโดยพิจารณาจากจุดรอบข้าง 7
2.5 ภาพที่ได้จากการลดสิ่งรบกวนโดยใช้ค่ากลางของข้อมูล 7
2.6 ผลลัพธ์ที่ได้จากการทำ Adaptive Threshold 8
2.7 ผลลัพธ์ที่ได้จากการทำ Scaling Transformation 9
2.8 ลักษณะของสมการที่เกิดจากการใช้ Linear Regression 10
2.9 การไม่เท่ากันของขนาดตัวอักษรในบรรทัดเนื่องจากการถ่ายภาพเอียงในด้านขวา 13
2.10 ระดับบรรทัดของตัวอักษรที่เกิดจากการถ่ายภาพเอียง 13
3.1 รูปภาพก่อน-หลัง ทำ Adaptive Threshold 14
3.2 รูปภาพก่อน-หลัง ทำการลดสิ่งรบกวน 15
3.3 ลักษณะสมการเส้นตรงของบรรทัด 15
3.4 แผนผังการทำงานของการหาสมการเส้นตรงของบรรทัด 17
3.5 สมการที่ได้จากการปรับสมการเส้นระดับบนเมื่อพบจุดของตัวอักษร 18
3.6 สมการเส้นตรงของบรรทัดตัวอักษรที่ได้จากการทำ Linear Regression 18
3.7 สมการที่ได้จากการปรับเส้นสมการให้อยู่ที่ตำแหน่งบนสุดและล่างสุดของบรรทัด 19
3.8 ตำแหน่งของค่าเริ่มต้นและสิ้นสุดของบรรทัดในแกน X 19
3.9 การหาค่าระยะห่างระหว่างสมการเส้นตรงทั้ง 2 เพื่อทำการเปรียบเทียบ 20
3.10 วิธีในการขยายบรรทัดของตัวอักษร โดยที่ YGap1 > YGab2 20
3.11 วิธีในการย่อบรรทัดของตัวอักษร โดยที่ YGap1 < YGab2 21
3.12 ผลที่ได้จากการย่อและขยายบรรทัดตัวอักษร 21
3.13 วิธีการปรับภาพบรรทัดตัวอักษรเข้าสู่แนวตรง 22
4.1 Context Diagram ของระบบ 23

สารบัญรูปภาพ (ต่อ)

รูปที่ หน้า

4.2 มุมมองในการออกแบบซอฟต์แวร์ 24
4.3 Use Cases Diagram 25
4.4 Sequence Diagram 26
4.5 Component Diagram 27
4.6 Activity Diagram 28
4.7 Class Diagram 29
5.1 ภาพถ่ายเอกสารที่ใช้ในการทดสอบ 31
5.2 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 10 เซนติเมตร 33
5.3 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 20 เซนติเมตร 35
5.4 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 30 เซนติเมตร 37
5.5 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 40 เซนติเมตร 39
5.6 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 50 เซนติเมตร 41
5.7 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 20 เซนติเมตร 43
5.8 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 30 เซนติเมตร 45
5.9 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 40 เซนติเมตร 47
5.10 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 50 เซนติเมตร 49
5.11 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 60 เซนติเมตร 51
5.12 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 20 เซนติเมตร 53
5.13 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 30 เซนติเมตร 55
5.14 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 40 เซนติเมตร 57
5.15 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 50 เซนติเมตร 59
5.16 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 60 เซนติเมตร 61
5.17 กราฟค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ 62
5.18 กราฟค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ 63

สารบัญรูปภาพ (ต่อ)

รูปที่ หน้า

5.19 กราฟค่าเฉลี่ยอัตราความถูกต้องเทียบกับขนาดของตัวอักษรในการถ่ายภาพ 64
5.20 ภาพป้ายทะเบียนที่ใช้ในการทดสอบ 65
5.21 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 50 เซนติเมตร 68
5.22 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 75 เซนติเมตร 70
5.23 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 100 เซนติเมตร 72
5.24 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 125 เซนติเมตร 74
5.25 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 150 เซนติเมตร 76
5.26 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 175 เซนติเมตร 78
5.27 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 200 เซนติเมตร 80
5.28 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 50 เซนติเมตร 82
5.29 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 75 เซนติเมตร 84
5.30 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 100 เซนติเมตร ... 86
5.31 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 125 เซนติเมตร ... 88
5.32 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 150 เซนติเมตร ... 90
5.33 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 175 เซนติเมตร ... 92
5.34 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 200 เซนติเมตร ... 94
5.35 กราฟค่าเฉลี่ยความถูกต้องเทียบกับระยะห่างในการถ่ายภาพจากกล้องดิจิตอล 95
5.36 กราฟค่าเฉลี่ยความถูกต้องเทียบกับระยะห่างในการถ่ายภาพจากกล้องโทรศัพท์มือถือ 96
5.37 กราฟค่าเฉลี่ยความถูกต้องเทียบกับมุมในการถ่ายภาพจากกล้องดิจิตอล 97
5.38 กราฟค่าเฉลี่ยความถูกต้องเทียบกับมุมในการถ่ายภาพจากกล้องโทรศัพท์มือถือ 97
6.1 ผลที่ได้จากการรู้จำด้วยโปรแกรม Tesseract OCR 101

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันการรู้จำตัวอักษรเข้ามามีบทบาทกับงานหลายๆด้าน เช่น การคัดแยกจดหมายจากรหัสไปรษณีย์, การแปลงเอกสารให้กลายเป็นข้อมูลที่คอมพิวเตอร์อ่านได้ เป็นต้น จึงเป็นโปรแกรมอำนวยความสะดวกแก่ผู้ใช้เป็นอย่างมาก

แต่ในทุกวันนี้การรู้จำตัวอักษรยังมีข้อจำกัดอยู่หลายๆด้าน เช่น ขนาดของตัวอักษรในภาพ, ความละเอียดของภาพ, ความคมชัดของภาพ, สิ่งรบกวนในภาพ และระดับความลึกของตัวอักษรในภาพ เป็นต้น ทำให้ความสามารถในการใช้งานการรู้จำตัวอักษรไม่มีประสิทธิภาพสูง โดยเฉพาะอย่างยิ่งปัญหาระดับความลึกของตัวอักษรในภาพต่างกันเป็นปัญหาใหญ่ของการรู้จำในปัจจุบัน เพราะเมื่อเกิดปัญหาดังกล่าวจะส่งผลกระทบต่อรูปภาพหลายๆด้าน เช่น ขนาดความกว้างและความสูงของตัวอักษรในบรรทัดเดียวกัน, ระยะห่างระหว่างแต่ละตัวอักษรในบรรทัดเดียว, ระยะห่างระหว่างบรรทัด และระดับเส้นบรรทัดของแต่ละตัวอักษรในบรรทัดเดียวกัน เป็นต้น ทำให้ความถูกต้องของการรู้จำต่ำลงอย่างมาก แต่ปัญหาลักษณะนี้สามารถเกิดขึ้นได้ง่าย เพราะในโลกแห่งความเป็นจริงการถ่ายรูปภาพเราไม่สามารถควบคุมให้ภาพที่ถ่ายได้มีระดับความลึกของตัวอักษรเท่ากันเสมอไป

ผู้พัฒนาเล็งเห็นว่าถ้าทำการพัฒนาการรู้จำให้สามารถรู้จำภาพที่มีระดับความลึกของตัวอักษรในภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้างได้ จะทำให้ประสิทธิภาพของการรู้จำตัวอักษรสูงขึ้น สามารถนำไปใช้ประโยชน์ในด้านเศรษฐกิจ และยังสามารถนำไปพัฒนาความสามารถของการรู้จำในด้านอื่นๆได้อีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 ศึกษาการทำงานของการรู้จำตัวอักษร และความรู้ต่างๆที่เกี่ยวข้องกับการรู้จำตัวอักษร

1.2.2 เพื่อพัฒนาประสิทธิภาพในการรู้จำตัวอักษร ให้มีความสามารถประมวลผลภาพที่มีระดับความลึกของตัวอักษรในภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้าง

1.2.3 เพื่อเป็นแนวทางสำหรับการพัฒนาความสามารถของการรู้จำตัวอักษรในลักษณะอื่นๆ

1.3 ขอบข่ายของโครงงาน

1.3.1 พัฒนาโปรแกรมสำหรับการรู้จำตัวอักษรจากภาพที่มีระดับความลึกของภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้าง

1.3.2 ตัวอักษรที่นำมารู้จำ มีจำนวน 62 ตัว ประกอบด้วย

ตัวอักษรภาษาอังกฤษพิมพ์ใหญ่	จำนวน 26 ตัว
ตัวอักษรภาษาอังกฤษพิมพ์เล็ก	จำนวน 26 ตัว
ตัวเลขฮินดูอารบิก	จำนวน 10 ตัว

1.3.3 ภาพที่ใช้ในการรู้จำ ตัวอักษรจะต้องเป็นสีดำเท่านั้น

1.3.4 อัตราความถูกต้องของผลลัพธ์จากการรู้จำโดยเฉลี่ยสูงกว่าอัตราความถูกต้องของผลลัพธ์จากการรู้จำโดยเฉลี่ยของโปรแกรม Tesseract OCR

1.4 ขั้นตอนของการดำเนินงาน

1.4.1 ศึกษาลักษณะการทำงานของการรู้จำตัวอักษร

1.4.2 ศึกษาการประมวลผลภาพดิจิตอลสำหรับทำการรู้จำ

1.4.3 ศึกษาและทดลองวิธีการที่ใช้ในการรู้จำตัวอักษร

1.4.4 ปรับปรุงวิธีการให้มีประสิทธิภาพดีและอยู่ในระดับที่ยอมรับได้

1.4.5 แก้ไขข้อบกพร่อง

1.4.6 สรุปและจัดทำเอกสาร

1.5 แผนผังการดำเนินโครงงาน

เวลา / การดำเนินงาน	พ.ศ. 2551										
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.
1. ศึกษาข้อมูลในการทำโครงงาน											
2. ปรับภาพให้เหมาะสมสำหรับการรู้จำ											
3. วิเคราะห์ความเป็นไปได้ของวิธีการที่จะใช้ในการรู้จำ											
4. ทดลองอัลกอริทึมต่างๆ ที่สามารถเป็นไปได้											
5. ปรับปรุงและแก้ไขอัลกอริทึม											
6. วิเคราะห์และสรุปผล											

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้รับความรู้เรื่องการทำงานของการรู้จำตัวอักษรและความรู้ต่างๆที่เกี่ยวข้องกับการรู้จำตัวอักษร

1.6.2 พัฒนาความสามารถของการรู้จำตัวอักษรให้มีความสามารถในการรู้จำภาพที่มีระดับความลึกของภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้างได้อย่างมีประสิทธิภาพ

1.6.3 เป็นแนวทางในการพัฒนาความสามารถของการรู้จำในด้านอื่นๆ

1.7 งบประมาณของโครงการ

1.7.1 ค่าจัดทำรายงาน	เป็นจำนวนเงิน	500	บาท
1.7.2 ค่าถ่ายเอกสาร	เป็นจำนวนเงิน	500	บาท
	รวมเป็นเงินทั้งสิ้น	1,000	บาท

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่ 2
ทฤษฎีพื้นฐาน

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานหรือความรู้ต่างๆที่จะนำมาใช้ในการรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกัน โดยในที่นี้จะกล่าวถึงเรื่องต่างๆ ดังนี้

1. การประมวลผลรูปดิจิตอล (Digital Image Processing)
2. โอเพ่นซีวี ไลบรารี (OpenCV Library)
3. การถอดถอยเชิงเส้น (Linear Regression)
4. หลักการของการรู้จำตัวอักษร
5. ปัญหาที่เกิดจากการถ่ายภาพเอียงในด้านซ้ายหรือด้านขวา

2.1 การประมวลรูปดิจิตอล (Digital Image Processing)

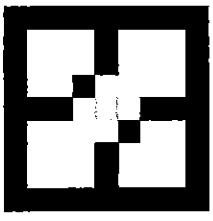
การประมวลผลรูปดิจิตอลเป็นการประยุกต์โดยใช้การประมวลผลสัญญาณดิจิตอล เพื่อประมวลผลภาพให้ได้ตามที่ต้องการ อาจจะใช้การเทคนิคหรืออัลกอริทึมต่างๆในการประมวลผล โดยในที่นี้จะกล่าวถึง 4 เทคนิคที่ใช้ คือ

2.1.1 ฟลัดฟิล (Flood Fill)

เป็นการระบายสี (Filling) รูปทรงที่ต้องการ โดยมีหลักการทำงานดังนี้

1. กำหนดจุดเริ่มต้น

สำหรับจุดเริ่มต้นในการ Flood Fill จะถูกเรียกว่า Seed Point เมื่อได้จุดเริ่มต้นแล้วจากนั้นทำการตรวจสอบว่าจุดนั้นยังไม่ถูกระบายสีและไม่ใช่ขอบของรูปทรง ถ้าตรงตามเงื่อนไขจะทำการระบายสีที่จุดนั้น



รูปที่ 2.1 จุดเริ่มต้นในการ Flood Fill [7]

2. ตรวจสอบจุดรอบข้างจุดเริ่มต้น

สำหรับการทำงานส่วนนี้มี 2 แบบคือ

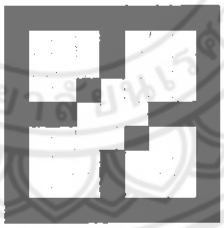
- การ Flood Fill แบบ 4 ทิศทาง

ทำการตรวจสอบจุดรอบข้างจุดเริ่มต้นทั้ง 4 ทิศทาง ถ้าพบว่าจุดรอบข้างไม่ใช่ขอบของรูปทรง จะทำการกำหนดจุดนั้นให้เป็นจุดเริ่มต้นใหม่ของการทำ Flood Fill ต่อไป

รูปที่ 2.2 ภาพที่ได้จากการทำ Flood Fill 4 ทิศทาง[7]

- การ Flood Fill แบบ 8 ทิศทาง

ทำการตรวจสอบจุดรอบข้างจุดเริ่มต้นทั้ง 8 ทิศทาง ถ้าพบว่าจุดรอบข้างไม่ใช่ขอบของรูปทรง จะทำการกำหนดจุดนั้นให้เป็นจุดเริ่มต้นใหม่ของการทำ Flood Fill ต่อไป



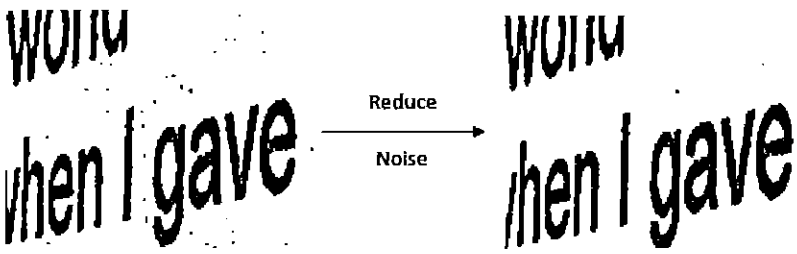
รูปที่ 2.3 ภาพที่ได้จากการทำ Flood Fill 8 ทิศทาง[7]

2.1.2 การลดสิ่งรบกวน

สำหรับการแปลงภาพจาก Gray-Scale ไปเป็นภาพแบบ Binary สิ่งที่อาจจะเกิดขึ้นได้คือสิ่งรบกวนในภาพ (Noise) ซึ่งถ้าหากเราไม่กำจัดสิ่งรบกวนออก อาจจะทำให้การประมวลผลภาพที่ได้ไม่มีประสิทธิภาพ ในการลดสิ่งรบกวนในโครงงานนี้จะใช้ 2 วิธีคือ

2.1.2.1 การลดสิ่งรบกวนโดยพิจารณาจากจุดรอบข้าง

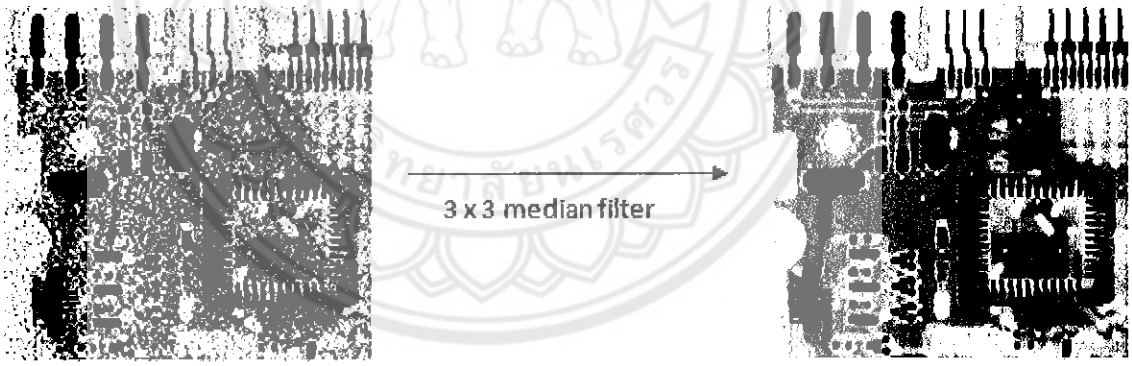
สำหรับการลดสิ่งรบกวนโดยพิจารณาจากจุดรอบข้าง จะใช้การนับจำนวนจุดสีดำรอบข้างจุดที่เราสนใจ ถ้าค่าที่ได้น้อยกว่า 3 จุด จะทำการเปลี่ยนจุดนั้นให้เป็นสีขาว



รูปที่ 2.4 ภาพที่ได้จากการลดสิ่งรบกวนโดยพิจารณาจากจุดรอบข้าง

2.1.2.2 การลดสิ่งรบกวนโดยใช้ค่ากลางของข้อมูล (Median Filter)

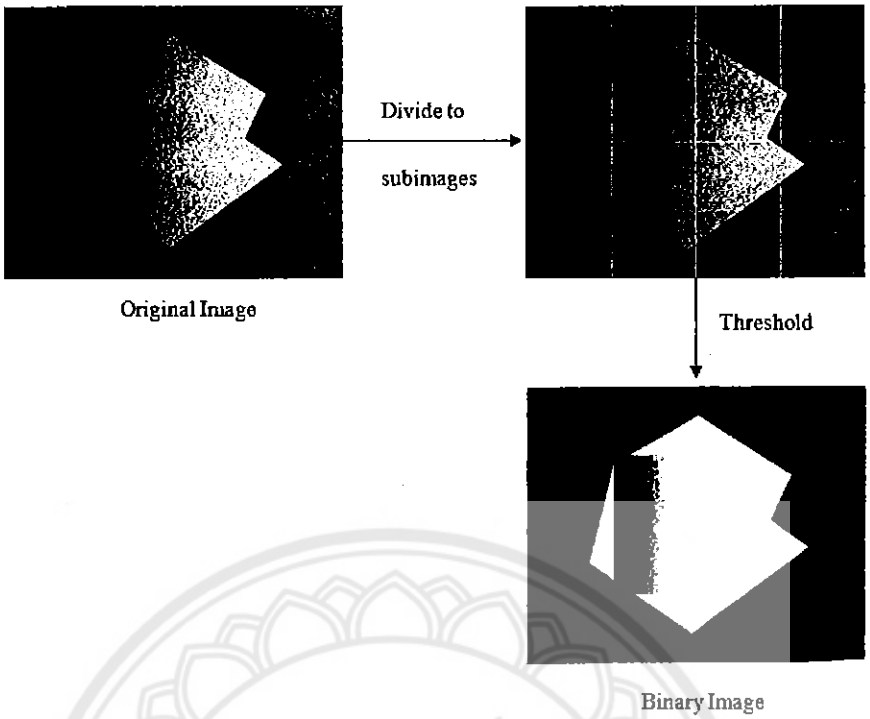
การลดสิ่งรบกวนวิธีนี้ จะเริ่มจากทำการกำหนดค่าของบล็อคที่ต้องการ เช่น 3 x 3 พิกเซลล์ เป็นต้น เมื่อได้แล้วค่าที่จุดกึ่งกลางจะมีค่าเท่ากับค่ากลางของข้อมูลในบล็อคนั้น เช่น บล็อคขนาด 3 x 3 พิกเซลล์ มีข้อมูลดังต่อไปนี้ (10, 20, 20, 20, 15, 20, 20, 25, 100) จากนั้นทำการเรียงลำดับข้อมูลในบล็อคจะได้ดังนี้ (10, 15, 20, 20, 20, 20, 20, 25, 100) จะเห็นว่าค่ากลางของข้อมูลชุดนี้คือ 20 เพราะฉะนั้นค่าที่จุดกึ่งกลางของบล็อคนี้จะมีค่าเท่ากับ 20 และทำการเลื่อนบล็อคไปจนหมดทั้งภาพ



รูปที่ 2.5 ภาพที่ได้จากการลดสิ่งรบกวนโดยใช้ค่ากลางของข้อมูล[2]

2.1.3 อแดปทีฟเทรดโฮล (Adaptive Threshold)

อแดปทีฟเทรดโฮลเป็นวิธีการสำหรับทำการเปลี่ยนรูปภาพให้อยู่ในลักษณะแบบ Binary (แต่ละพิกเซลล์มีค่าได้เพียงแค่ 0 หรือ 1 เท่านั้น) การทำ Adaptive Threshold แตกต่างจาก Global Threshold คือ จากรูปที่ 2.6 จะทำการแบ่งรูปที่ต้องการให้เป็นรูปย่อยๆ จากนั้นทำการคำนวณหาค่า Histogram ของแต่ละรูปย่อยแล้วจึงทำ Threshold แต่ละรูปย่อยนั้น ซึ่ง Global Threshold จะทำการ Threshold โดยไม่แบ่งเป็นรูปย่อยๆ



รูปที่ 2.6 ผลลัพธ์ที่ได้จากการทำ Adaptive Threshold[2]

2.1.4 สเกลลิ่งทรานฟอร์มเมชั่น (Scaling Transformation)

เป็นวิธีในการปรับขนาดของรูปภาพให้มีขนาดใหญ่ขึ้นหรือเล็กลงตามที่ต้องการ โดยใช้ลักษณะของการคูณตำแหน่งของภาพที่ต้องการด้วยอัตราการปรับขนาด ดังสมการต่อไปนี้

$$I(x,y) \rightarrow O(x',y')$$

เมื่อ

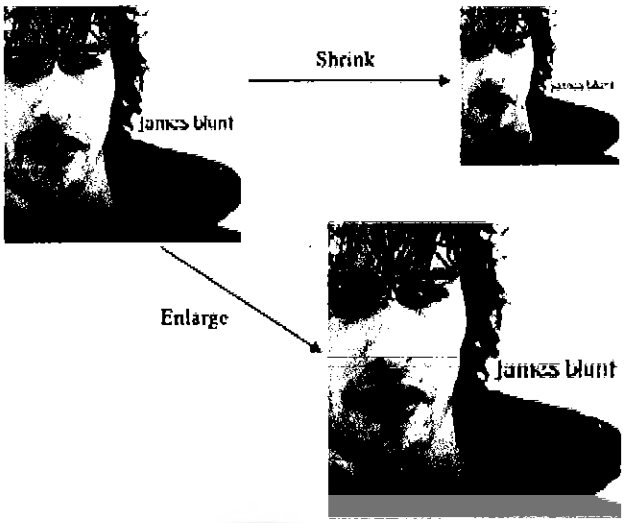
$$x' = xS_x \text{ และ } y' = yS_y$$

โดยที่ I(x,y) เป็นค่าตำแหน่งรูปเริ่มต้น

O(x',y') เป็นค่าตำแหน่งของรูปที่ต้องการ

S_x เป็นอัตราการปรับขนาดในแกน X

Sy เป็นอัตราการปรับขนาดในแกน Y



รูปที่ 2.7 ผลลัพธ์ที่ได้จากการทำ Scaling Transformation

2.2 โอเพ่นซีวี ไลบราลี (OpenCV Library)

OpenCV (Open Computer Vision) คือ ไลบราลีแบบ Open Source ที่เขียนด้วยภาษา C และ C++ ภายในประกอบด้วยฟังค์ชั่นและคลาสการทำงานต่างๆทางด้านระบบการมองเห็นด้วยคอมพิวเตอร์ (Computer Vision) พัฒนาโดยบริษัท อินเทล คอร์ปเปอเรชั่น (Intel Corporation) สามารถทำงานได้ทั้ง Windows Platform และ Linux Platform

สำหรับฟังค์ชั่นและโครงสร้างต่างๆที่ใช้ในโครงงานมีต่อไปนี้

1. IplImage โครงสร้างสำหรับจัดเก็บข้อมูลรูปภาพ
2. cvCreateImage ฟังค์ชั่นสำหรับสร้างพื้นที่สำหรับจัดเก็บรูปภาพตามขนาดที่กำหนด
3. cvLoadImage ฟังค์ชั่นสำหรับทำการโหลดข้อมูลรูปภาพจากอุปกรณ์จัดเก็บข้อมูล
4. cvReleaseImage ฟังค์ชั่นสำหรับคืนค่าหน่วยความจำที่เก็บรูปภาพสู่ระบบ
5. cvCvtColor ฟังค์ชั่นสำหรับแปลงรูปแบบสีของภาพ
6. cvAdaptiveThreshold ฟังค์ชั่นในการแปลงภาพจาก Gray-Scale เป็นภาพขาว-ดำ
7. cvSmooth ฟังค์ชั่นในการปรับปรุงภาพใช้ชัดเจนขึ้น

สำหรับรายละเอียดและวิธีการใช้งานของฟังค์ชั่นจะกล่าวอีกครั้งในส่วนของภาคผนวก ก.

2.3 การถดถอยเชิงเส้น (Linear Regression)

การถดถอยเชิงเส้น เป็นวิธีในการหาค่าของสมการเส้นตรงโดยประมาณจากจุดที่มี ซึ่งมีวิธีในการคำนวณดังนี้

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{S_{xy}}{S_{xx}}$$

โดยที่

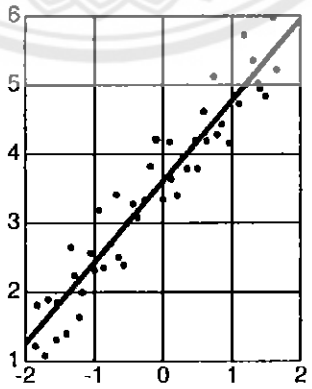
$$\bar{y} = \left(\frac{1}{n}\right)\sum_{i=1}^{n} y_i$$

และ

$$\bar{x} = \left(\frac{1}{n}\right)\sum_{i=1}^{n} x_i$$

จากการคำนวณจะได้สมการเส้นตรงโดยประมาณอยู่ในรูปของ

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$



รูปที่ 2.8 ลักษณะของสมการที่เกิดจากการใช้ Linear Regression[4]

2.4 หลักการของการรู้จำตัวอักษร

การรู้จำตัวอักษรเป็นการนำรูปภาพตัวอักษรที่ได้ มาทำการวิเคราะห์รูปภาพ เพื่อแปลงข้อมูลรูปภาพให้กลายเป็นรหัสตัวอักษร โดยจะใช้ความรู้หลายแขนงมาประกอบเช่น Pattern Recognition, Digital Image Processing, Digital Signal Processing โดยการรู้จำตัวอักษรมีโครงสร้างดังต่อไปนี้

2.4.1 กระบวนการประมวลผลขั้นต้น (Pre-Processing)

เนื่องจากรูปภาพที่จะทำการรู้จำตัวอักษรนั้น อาจจะมีสิ่งต่างๆที่มีผลทำให้การรู้จำตัวอักษรไม่มีประสิทธิภาพ จึงต้องมีกระบวนการประมวลผลขั้นต้นเพื่อปรับคุณลักษณะของรูปภาพต่างๆ โดยที่กระบวนการประมวลผลขั้นต้นนั้นมีความสำคัญต่อประสิทธิภาพของการรู้จำตัวอักษรมาก เพราะหากเกิดความผิดพลาดขึ้นในส่วนนี้ จะส่งผลกระทบต่อกระบวนการถัดไปด้วย

สำหรับขั้นตอนต่างๆในกระบวนการประมวลผลขั้นต้นที่สำคัญมีดังต่อไปนี้

- การกรองสิ่งรบกวน (Noise Filtering)

เนื่องจากรูปภาพต้นฉบับที่นำมาทำการรู้จำตัวอักษรอาจมีสิ่งรบกวนต่างๆ ซึ่งส่งผลให้คุณภาพของรูปภาพไม่ดี และเมื่อนำภาพมาทำการรู้จำตัวอักษรจะทำให้ประสิทธิภาพในการรู้จำตัวอักษรลดลง จึงต้องทำการกรองสิ่งรบกวนให้ออกไปจากรูปภาพ

- การปรับแต่งข้อมูล (Normalization)

ในขั้นตอนนี้จะทำการปรับแต่งรูปภาพตัวอักษรต่างๆ ให้อยู่ในรูปแบบที่จะนำมาใช้งาน ยกตัวอย่างเช่น การปรับขนาดรูปตัวอักษร, การแปลงรูปสีให้กลายเป็นรูปขาวดำ เป็นต้น

- การตัดแบ่งพื้นที่ใช้งาน (Cropping)

การตัดแบ่งพื้นที่ใช้งานนั้น จะทำการดึงเอาเฉพาะส่วนของรูปตัวอักษรออกจากรูปภาพ เพื่อส่งต่อให้ขั้นตอนต่อไป

2.4.3 การรู้จำ (Recognition)

การรู้จำจะเป็นการทำงานที่ระบบจะตัดสินใจว่า รูปตัวอักษรที่ส่งเข้าไปในกระบวนการนั้นเป็นรหัสตัวอักษรอะไร สำหรับวิธีการรู้จำนั้นมีหลายวิธี แต่สามารถจำแนกออกเป็น 4 กลุ่มได้ดังนี้

- วิธีการเข้าคู่รูปแบบ (Template Matching)

วิธีการนี้เป็นวิธีการแรกๆ ในการรู้จำตัวอักษร หลักการโดยทั่วไปคือ จะมีการสร้างรูปแบบ (Template) ขึ้นมาสำหรับอ่านตัวอักษรโดยจะกำหนดตำแหน่งสำคัญที่สามารถใช้แยกแยะความแตกต่างระหว่างตัวอักษรได้ ในการทำงานจะนำรูปภาพตัวอักษรที่ต้องการมาทำการเปรียบเทียบกับ Template เพื่อวัดความคล้ายคลึงกันของรูปแบบ จากนั้นจะทำการระบุว่าเป็นรหัสตัวอักษรอะไร สำหรับวิธีการนี้จะมีผลกระทบค่อนข้างง่ายต่อ สิ่งรบกวนต่างๆ ขนาด และการเอียงของตัวอักษร

- วิธีการทางสถิติ (Statistical Approach)

วิธีการทางสถิติจะใช้ทฤษฎีทางสถิติมาใช้ในการแยกแยะ โดยอาจจะใช้ค่าของความน่าจะเป็นหรือฟังค์ชั่นสำหรับการแจกแจงความน่าจะเป็นเพื่อการตัดสินใจ ในการทำงานนั้นจะนำรูปภาพตัวอักษรที่ได้มาทำการรู้จำ และวัดผลเป็นค่าของความน่าจะเป็นของแต่ละตัวอักษร เมื่อผ่านการรู้จำครบทุกตัวอักษรแล้ว จึงนำเอาผลลัพธ์ที่ได้ทั้งหมดมาเปรียบเทียบกันว่ามีความน่าจะเป็นตัวอักษรใดมากที่สุด แล้วจึงส่งผลลัพธ์ออกเป็นตัวอักษรนั้น

- วิธีการวิเคราะห์ทางโครงสร้าง (Structural Analysis)

การวิเคราะห์โครงสร้างจะถือว่าตัวอักษรทุกตัวประกอบกันด้วยองค์ประกอบพื้นฐาน เช่น เส้นตรง, วงกลม เป็นต้น ในขั้นตอนของการรู้จำนั่นระบบจะนำข้อมูลลักษณะสำคัญที่ประกอบเป็นตัวอักษรนั่น ส่งให้กับส่วนที่วิเคราะห์รูปแบบของตัวอักษร เช่น Formal Grammar Machine, โครงสร้างกราฟ หรือโครงสร้างต้นไม้ เป็นต้น โดยในการตัดสินใจว่าเป็นตัวอักษรอะไรจะดูจากรูปแบบการเชื่อมต่อกันขององค์ประกอบต่างๆ วิธีการนี้มีความได้เปรียบที่สามารถทำการรู้จำตัวอักษรได้หลากหลาย แต่ทั้งนี้ขึ้นอยู่กับการสร้างกฎต่างๆ ที่วิเคราะห์รูปแบบของตัวอักษร

- วิธีทางโครงข่ายประสาทเทียม (Neural Network)

วิธีการนี้จะทำการเลียนแบบการทำงานของสมองมนุษย์ ที่มีโครงข่ายของเส้นประสาทเชื่อมต่อกันเป็นจำนวนมาก โดยความรู้จะได้จากการฝึกสอน ในการเตรียมการจะต้องทำการฝึกสอนโครงข่ายประสาทเทียมโดยส่งรูปตัวอักษรพร้อมทั้งบอกรหัสตัวอักษร เพื่อให้โครงข่ายประสาทเทียมมีความสามารถในการแยกแยะรูปแบบต่างๆ ได้หลากหลาย

2.4.3 กระบวนการประมวลผลขั้นปลาย (Post-Processing)

เนื่องจากผลที่ได้จากการรู้จำนั้นไม่ได้หมายความว่าจะมีความถูกต้องทั้งหมด ในขั้นตอนนี้จะทำการนำผลที่ได้มาตรวจสอบกับการสะกดคำหรือไวยากรณ์ของภาษา เพื่อให้ผลที่ได้ออกมาถูกต้องมากยิ่งขึ้น ซึ่งอาจจะแก้ไขให้อัตโนมัติหรือแจ้งให้ผู้ใช้ทราบว่าคำดังกล่าวไม่ถูกต้อง

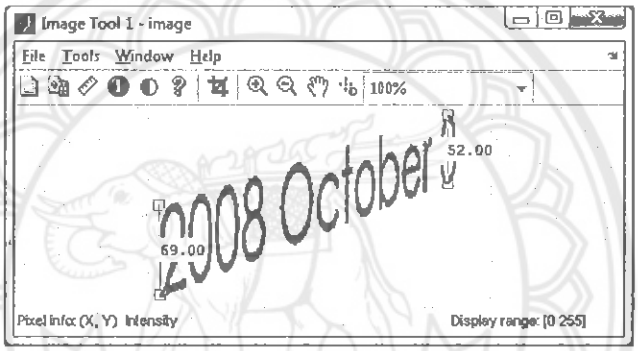
ในโครงงานนี้จะใช้โปรแกรม Tesseract OCR ซึ่งเป็นโปรแกรมประเภท Open Source ในการรู้จำตัวอักษร สำหรับรายละเอียดของโปรแกรมนี้จะกล่าวอีกครั้งในส่วนของภาคผนวก ข.

2.5 ปัญหาที่เกิดจากการถ่ายภาพแบบด้านข้าง

เนื่องจากการถ่ายภาพแบบด้านข้าง (Side shot) จะทำให้ภาพเกิดปัญหาต่างๆ ได้ดังนี้

2.5.1 ขนาดของตัวอักษรในบรรทัดเดียวกัน

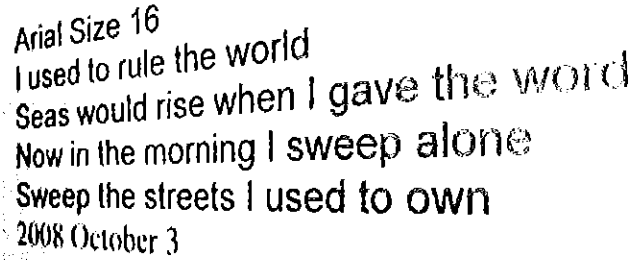
การถ่ายภาพแบบด้านข้างจะทำให้เกิดระดับความลึกของตัวอักษรในภาพไม่เท่ากัน และยังทำให้ขนาดของตัวอักษรในบรรทัดเดียวกันมีขนาดไม่เท่ากัน ดังรูปที่ 2.9



รูปที่ 2.9 การไม่เท่ากันของขนาดตัวอักษรในบรรทัดเนื่องจากการถ่ายภาพเอียงในด้านขวา

2.5.2 เส้นระดับบรรทัดของตัวอักษร

การถ่ายภาพแบบด้านข้างยังส่งผลให้เส้นระดับบรรทัดของแต่ละบรรทัดตัวอักษรในรูปไม่อยู่ในแนวระดับ ทำให้เกิดการซ้อนกันของบรรทัดจากเส้นบรรทัดในแนวตรง ซึ่งเมื่อนำภาพไปทำการรู้จำตัวอักษรจะทำให้ผลที่ได้มีความถูกต้องลดลง



รูปที่ 2.10 ระดับบรรทัดของตัวอักษรที่เกิดจากการถ่ายภาพเอียง

บทที่ 3

การออกแบบและพัฒนาระบบ

บทนี้จะกล่าวถึงวิธีในการออกแบบและพัฒนาระบบสำหรับโครงงานนี้ เพื่อความสะดวกในการอธิบายและการทำความเข้าใจ ผู้พัฒนาจึงทำการแบ่งโปรแกรมออกเป็น 3 ส่วนหลักๆ ดังนี้

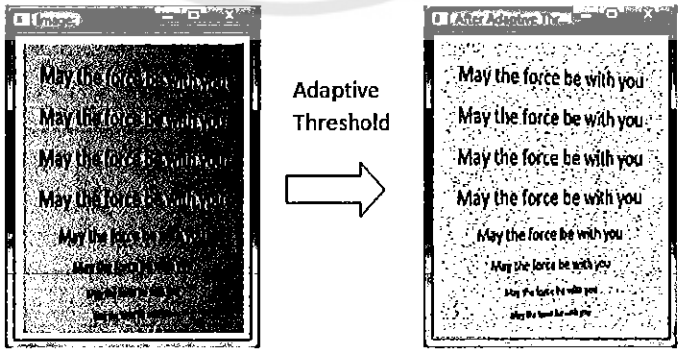
1. การค้นหาบรรทัดตัวอักษร
2. การปรับปรุงรูปภาพก่อนทำการรู้จำ
3. การรู้จำตัวอักษร

3.1 การค้นหาบรรทัดตัวอักษร

เนื่องจากการถ่ายภาพแบบด้านข้างส่งผลกระทบให้รูปของตัวอักษรแต่ละบรรทัดไม่อยู่ในระดับเดียวกัน จึงต้องทำการค้นหาบรรทัดตัวอักษร, ค่าเริ่มต้นและค่าสิ้นสุดของบรรทัดในแกน X และหาสมการเส้นตรงของบรรทัด เพื่อจะนำค่าที่ได้ไปใช้ในการปรับปรุงรูปภาพก่อนทำการรู้จำ โดยมีขั้นตอนดังนี้

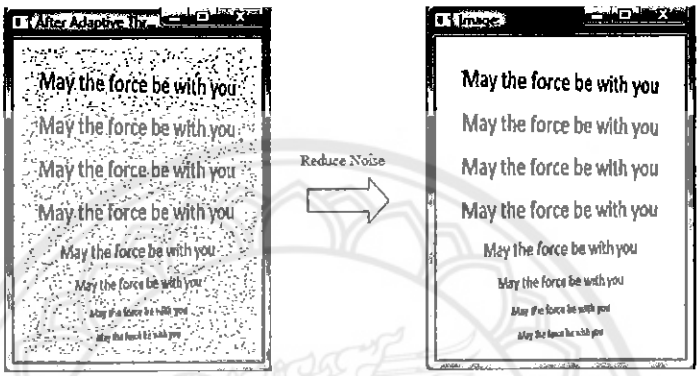
3.1.1 การแปลงภาพเป็น Binary

เพื่อความสะดวกในการประมวลผลภาพจึงทำการแปลงภาพจาก Grayscale ให้กลายเป็นภาพแบบ Binary โดยการทำ Adaptive Threshold



รูปที่ 3.1 รูปภาพก่อน-หลัง ทำ Adaptive Threshold

แต่เนื่องจากการทำ Adaptive Threshold อาจจะทำให้เกิดสิ่งรบกวน (Noise) จึงต้องทำการลดสิ่งรบกวนเพื่อหลีกเลี่ยงปัญหาที่จะตามมา เช่น การหาเส้นระดับบรรทัดผิดพลาด, การรู้จำตัวอักษรผิดพลาด เป็นต้น โดยสำหรับการรู้จำตัวอักษรตัวไปจะใช้วิธีการลดสิ่งรบกวนโดยพิจารณาจากจุดรอบข้าง และสำหรับการรู้จำตัวอักษรจากทะเบียนรถยนต์จะใช้การลดสิ่งรบกวนโดยใช้ค่ากลางของข้อมูล

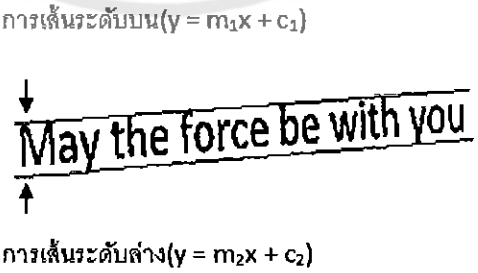


รูปที่ 3.2 รูปภาพก่อน-หลัง ทำการลดสิ่งรบกวน

3.1.2 การค้นหาสมการเส้นตรงของบรรทัดตัวอักษร

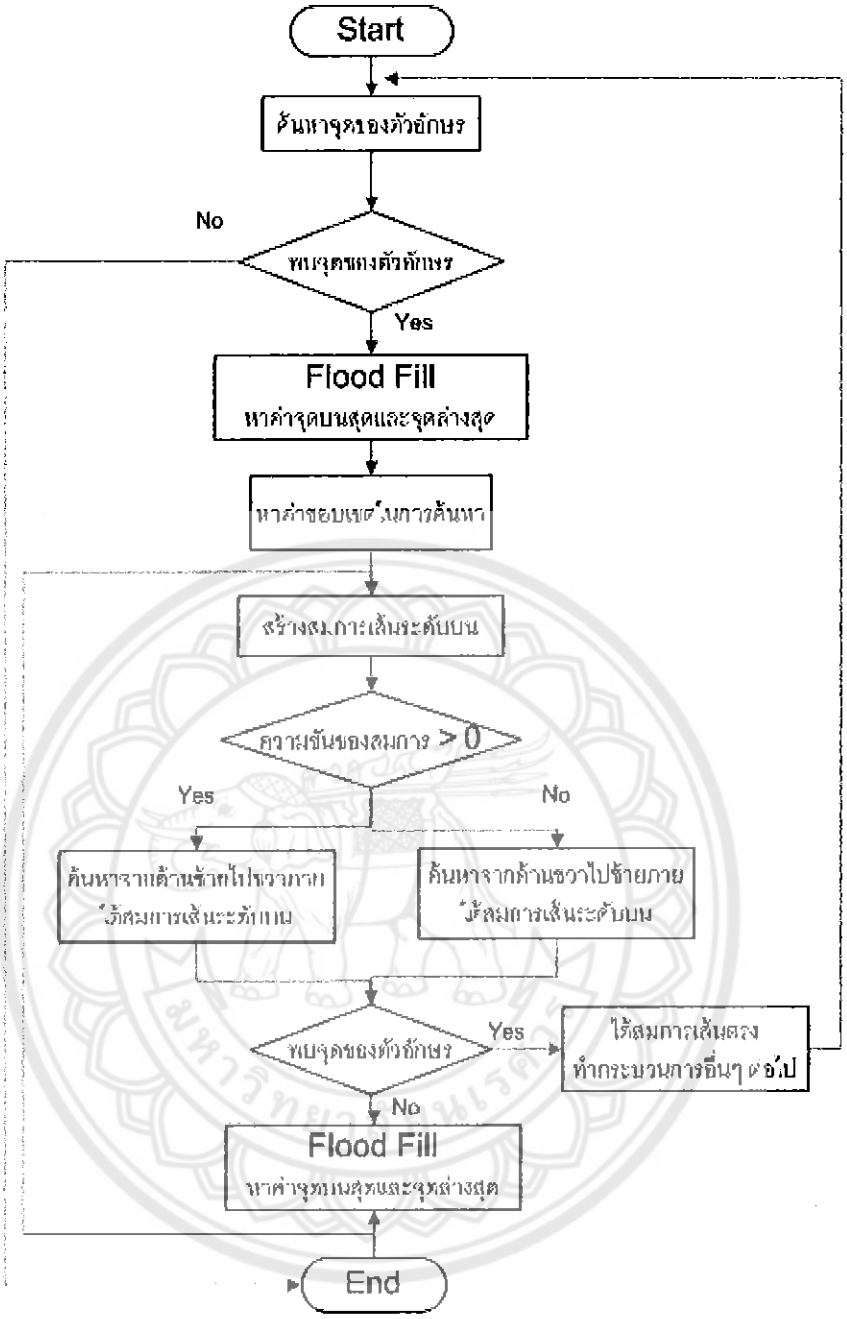
สมการเส้นตรงที่ใช้สำหรับการปรับปรุงรูปภาพมี 2 สมการคือ

- สมการเส้นระดับบนของบรรทัด
- สมการเส้นระดับล่างของบรรทัด



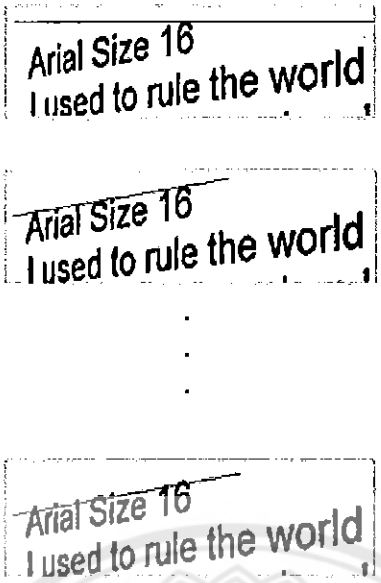
รูปที่ 3.3 ลักษณะสมการเส้นตรงของบรรทัด

สำหรับการค่าหาสมการเส้นตรงของบรรทัดตัวอักษรนั้น จะเริ่มจากการหาค่าของสมการเส้นระดับบนก่อน โดยจะทำการค้นหาจุดของตัวอักษร เมื่อพบจุดของตัวอักษรแล้วจึงทำการ Flood Fill 8 ทิศทางเพื่อหาจุดที่มีค่าของแกน Y น้อยที่สุดและจุดที่มีค่าของแกน Y มากที่สุดเพียงอย่างละหนึ่งจุดเท่านั้น จากนั้นจะนำจุดที่มีค่าของแกน Y น้อยที่สุดเก็บไว้ในอาเรย์ที่เก็บจุดของสมการเส้นระดับบน และเก็บจุดที่มีค่าของแกน Y มากที่สุดไว้ในอาเรย์ที่เก็บจุดของสมการเส้นระดับล่าง เฉพาะสำหรับจุดแรกสุดที่พบหลังจากทำการเก็บจุดที่ไว้ในอาเรย์แล้ว จะทำการหาค่าระยะห่างระหว่างจุดบนสุดและจุดล่างสุด แล้วนำค่าที่ได้ให้ลดลงเหลือแค่ 67 % แล้วจึงกำหนดให้เป็นค่าขอบเขตในการค้นหาของแกน Y เพื่อเป็นการกำหนดค่าขอบเขตสำหรับการค้นหาตัวอักษรต่างๆในบรรทัดเดียวกันและเพื่อค้นหาจุดบนของตัวอักษร โดยประมาณ หลังจากได้จุดบนและจุดล่างของตัวอักษรแล้ว จะทำการสร้างสมการเส้นระดับบนจากจุดที่มีในอาเรย์ที่เก็บจุดของสมการเส้นระดับบนโดยใช้วิธี Linear Regression และค้นหาจุดต่อไปใต้สมการเส้นระดับบนและมีค่าไม่เกินค่าขอบเขตในการค้นหา โดยมีเงื่อนไขว่าถ้าค่าความชันของสมการเส้นระดับบนมีค่าเป็นลบ จะทำการค้นหาจากทางด้านขวาของรูปภาพไปด้านซ้ายของรูปภาพ และถ้าความชันของสมการเส้นระดับบนเป็นบวกจะทำการค้นหาจากทางด้านซ้ายของรูปภาพไปทางด้านขวาของรูปภาพ เพื่อป้องกันกรณีที่การค้นหาพบตัวอักษรของบรรทัดอื่นก่อน ถ้าพบจุดของตัวอักษร จะทำตามกระบวนการที่กล่าวมาข้างต้น จนกระทั่งการค้นหาจุดภายใต้สมการเส้นระดับบนภายในขอบเขตที่กำหนดไว้ไม่พบจุดของตัวอักษร ซึ่งแผนผังการทำงานได้แสดงในรูปที่ 3.4



รูปที่ 3.4 แผนผังการทำงานของการหาสมการเส้นตรงของบรรทัด

ซึ่งการใช้วิธีการดังกล่าวจะทำให้ได้เส้นสมการที่ใกล้เคียงกับสมการเส้นตรงระดับบนของบรรทัดทุกครั้งที่พบจุดของตัวอักษร ซึ่งแสดงในรูป 3.5 หลังจากได้สมการเส้นระดับบนแล้ว จึงทำการสร้างสมการเส้นระดับล่างจากอาเรย์ที่เก็บจุดของสมการเส้นระดับล่างโดยใช้วิธี Linear Regression เช่นกัน



รูปที่ 3.5 สมการที่ได้จากการปรับสมการเส้นระดับบนเมื่อพบจุดของตัวอักษร

รูปที่ 3.6 สมการเส้นตรงของบรรทัดตัวอักษรที่ได้จากการทำ Linear Regression

แต่สมการที่ได้หลังจากการใช้ Linear Regression นั้น เส้นของสมการอาจจะไม่อยู่ที่ตำแหน่งบนสุดหรือล่างสุดของบรรทัดเสมอไปดังรูปที่ 3.6 ดังนั้นจึงต้องทำการปรับสมการที่ได้ให้อยู่ในตำแหน่งบนสุดหรือล่างสุดของบรรทัด เพื่อเตรียมสำหรับการปรับปรุงรูปภาพก่อนการรู้จำ โดยมีขั้นตอนดังต่อไปนี้

- หาจุดที่มีระยะห่างจากเส้นสมการมากที่สุด

- นำค่าของจุดที่ได้ มาทำการปรับสมการโดยคงค่าความชันไว้เช่นเดิม แล้วทำการคำนวณเพื่อหาค่าของจุดตัดแกน Y ใหม่ตามสมการ

$$c = y_{max} - mx_{max}$$

เมื่อ y_{max} เป็นค่าในแกน Y ของจุดที่มีระยะห่างจากเส้นสมการมากที่สุด

x_{max} เป็นค่าในแกน X ของจุดที่มีระยะห่างจากเส้นสมการมากที่สุด

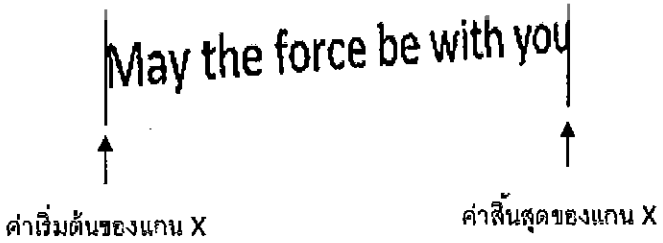
m เป็นค่าความชันของสมการ

c เป็นค่าจุดตัดแกน Y ของสมการ

รูปที่ 3.7 สมการที่ได้จากการปรับเส้นสมการให้อยู่ที่ตำแหน่งบนสุดและล่าง**สุดของบรรทัด**

3.1.3 การหาค่าเริ่มต้นและค่าสิ้นสุดของบรรทัดในแกน X

เนื่องจากตำแหน่งเริ่มต้นและสิ้นสุดของแต่ละบรรทัดตัวอักษรในภาพอาจจะไม่เท่ากัน จึงต้องทำการหาค่าเริ่มต้นและสิ้นสุดของบรรทัดตัวอักษรในแกน X เพื่อเป็นการกำหนดค่าเริ่มต้นให้กับการปรับปรุงรูปภาพก่อนทำการรู้จำ วิธีการคือจะทำการค้นหาจุดภายใต้สมการเส้นระดับบนและสมการเส้นระดับล่าง โดยให้ค่าเริ่มต้นของบรรทัดในแกน X เท่ากับจุดที่มีค่า X น้อยที่สุด และให้ค่าสิ้นสุดของบรรทัดในแกน X เท่ากับจุดที่มีค่า X มากที่สุด



รูปที่ 3.8 ตำแหน่งของค่าเริ่มต้นและสิ้นสุดของบรรทัดในแกน X

3.2 การปรับปรุงรูปภาพก่อนทำการรู้จำ

หลังจากได้ค่าต่างๆที่ต้องใช้ในการปรับปรุงรูปภาพก่อนทำการรู้จำแล้ว จึงนำค่าที่ได้มาทำการปรับปรุงรูปภาพ โดยมีขั้นตอนดังนี้

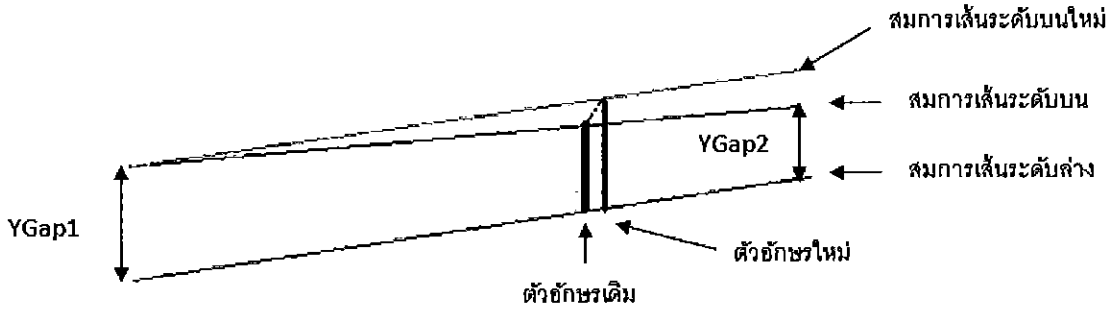
3.2.1 การย่อและขยายบรรทัดตัวอักษร

เป็นการปรับขนาดของบรรทัดตัวอักษรให้มีขนาดเท่ากันทั้งบรรทัด เพื่อให้ขนาดของตัวอักษรในภาพมีขนาดใกล้เคียงกับตัวอักษรปกติหรือทำให้ตัวอักษรชัดเจนขึ้น โดยมีขั้นตอนดังนี้

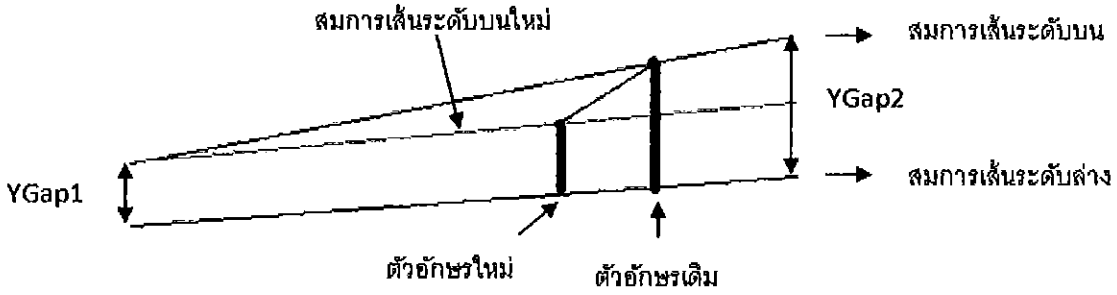
- นำสมการเส้นตรงที่ได้ทั้งสองสมการมาแทนค่าด้วยจุดเริ่มต้นและจุดสิ้นสุดของแกน X และหาระยะห่างระหว่างสมการเส้นระดับบนกับสมการเส้นระดับล่าง เปรียบเทียบระยะห่างของสมการที่ได้จากการแทนค่าของจุดเริ่มต้นและจุดสิ้นสุดของแกน X โดยสำหรับการขยายจะใช้ค่าที่มีระยะห่างมากกว่าเป็นเกณฑ์และการย่อจะใช้ค่าที่มีระยะห่างน้อยกว่าเป็นเกณฑ์

รูปที่ 3.9 การหาค่าระยะห่างระหว่างสมการเส้นตรงทั้ง 2 เพื่อทำการเปรียบเทียบ

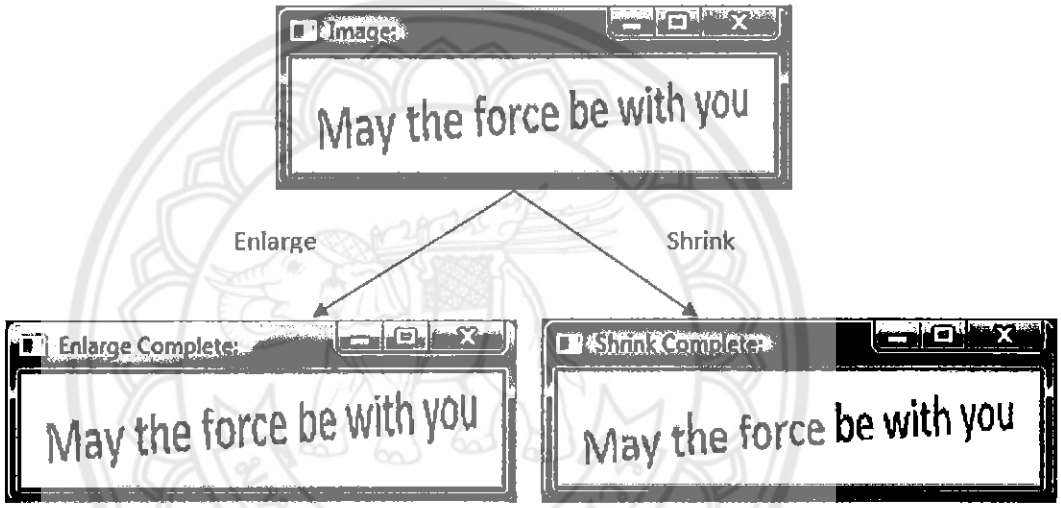
- จากนั้นทำการขยายหรือย่อภาพในแนวตั้งเพื่อให้ระยะห่างของสมการทั้งสองเส้นสำหรับทุกจุดในแกน X มีขนาดเท่ากับขนาดเกณฑ์ที่ตั้งเอาไว้ โดยจะใช้สมการเส้นระดับล่างเป็นหลัก แล้วทำการย่อหรือขยายภาพไปสู่สมการเส้นระดับบนเส้นใหม่



รูปที่ 3.10 วิธีในการขยายบรรทัดของตัวอักษร โดยที่ YGap1 > YGab2



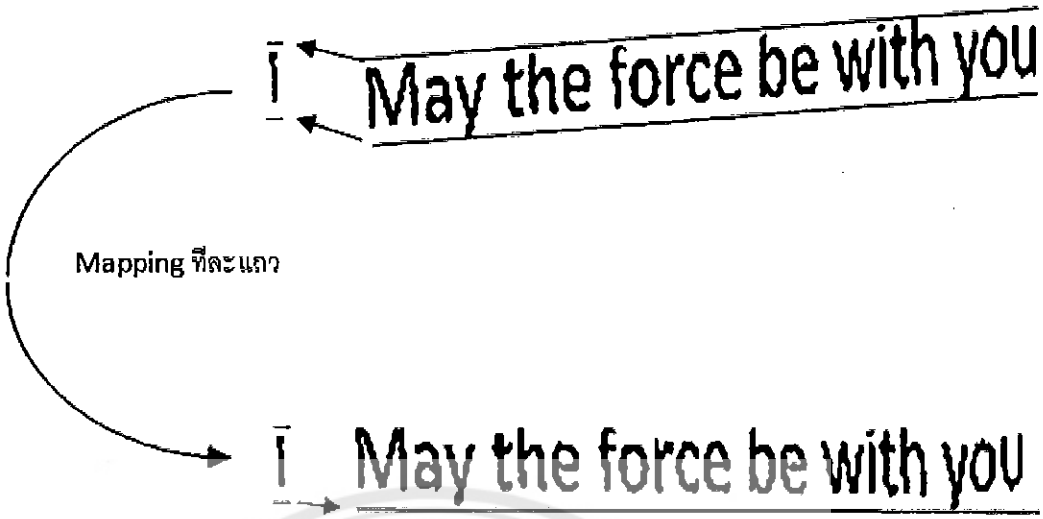
รูปที่ 3.11 วิธีในการย่อบรรทัดของตัวอักษร โดยที่ YGap1 < YGab2



รูปที่ 3.12 ผลที่ได้จากการย่อและขยายบรรทัดตัวอักษร

3.2.2 การปรับภาพบรรทัดตัวอักษรเข้าสู่แนวระดับ

ผลจากการย่อหรือขยายบรรทัดตัวอักษรจะทำให้ระดับความสูงของตัวอักษรในภาพมีความใกล้เคียงกับความสูงปกติ แต่ตัวอักษรยังไม่อยู่ในแนวระดับ เราจึงทำการปรับภาพตัวอักษรจากแนวระดับในมุมเอียงให้เข้าสู่แนวระดับปกติ โดยอาศัยสมการเส้นระดับล่างเป็นหลัก และทำการเลื่อนค่าของจุดในรูปภาพแบบจุดต่อจุดในทิศทางขึ้นสู่สมการเส้นบน เมื่อเสร็จสิ้นกระบวนการจะได้รูปภาพบรรทัดตัวอักษรในแนวระดับปกติดังรูปที่ 3.13



รูปที่ 3.13 วิธีการปรับภาพบรรทัดตัวอักษรเข้าสู่แนวตรง

3.3 การรู้จำตัวอักษร

เมื่อได้ภาพบรรทัดตัวอักษรในแนวระดับจากการปรับปรุงรูปภาพแล้ว ขั้นตอนต่อไปจะทำการรู้จำตัวอักษรเพื่อให้ได้ค่าของตัวอักษรออกมา โดยผู้พัฒนาได้ใช้โปรแกรม Tesseract OCR ในการรู้จำตัวอักษร หลังจากการรู้จำเสร็จสิ้นแล้ว จะต้องทำการอ่านค่าไฟล์ที่ได้เข้ามาสู่ไฟล์หลัก เนื่องจากผู้พัฒนากำหนดให้โปรแกรมทำการรู้จำตัวอักษรครั้งละหนึ่งบรรทัด และข้อจำกัดของโปรแกรม Tesseract OCR คือ เมื่อทำการรู้จำโดยกำหนดให้ผลลัพธ์ที่ได้ไปที่ไฟล์เดิม โปรแกรมจะทำการเขียนทับค่าของข้อมูลเก่าในไฟล์นั้น

เมื่อเสร็จสิ้นกระบวนการทั้งหมด จะทำการค้นหาบรรทัดตัวอักษรในภาพ ถ้าพบบรรทัดตัวอักษรในภาพ จะกลับไปเริ่มกระบวนการข้อ 3.1.2 ต่อไป

บทที่ 4

การออกแบบซอฟต์แวร์

เนื่องจากเพื่อความสะดวกที่ผู้อื่นจะทำการศึกษาซอฟต์แวร์ที่พัฒนาขึ้นมา จึงใช้บทนี้ในการอธิบายสิ่งต่างๆ ที่จำเป็นในการพัฒนาดังนี้

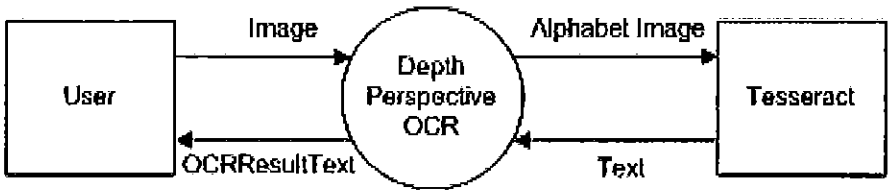
1. ความต้องการของระบบ (Requirement Specification)
2. ขอบเขตของระบบ
3. การออกแบบซอฟต์แวร์ โดยใช้ UML Diagram เป็นหลัก

4.1 ความต้องการของระบบ (Requirement Specification)

โปรแกรมสำหรับการรู้จำตัวอักษรจากภาพที่มีระดับความลึกของภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้างนั้น มีความต้องการต่างๆ ของระบบดังนี้

- สามารถเลือกภาพที่ต้องการได้
- แสดงผลที่ได้จากการนำภาพที่ถูกเลือกเข้าสู่การรู้จำโดยตรง
- แสดงผลที่ได้จากการนำภาพที่ถูกเลือกเข้าสู่การรู้จำตัวอักษรจากภาพที่มีระดับความลึกของภาพแตกต่างกัน

4.2 ขอบเขตของระบบ



รูปที่ 4.1 Context Diagram ของระบบ

การทำงานของระบบเริ่มจากการที่ User ทำการส่งรูปภาพที่ต้องการเข้าสู่ระบบ ระบบจะทำการประมวลผลและส่งภาพบรรทัดตัวอักษรให้ Tesseract ทำการรู้จำตัวอักษร เมื่อการรู้จำเสร็จสิ้น Tesseract จะส่งข้อความที่ได้ให้แก่ระบบ และระบบจะส่งผลสุดท้ายของการรู้จำให้ User ต่อไป

4.3 การออกแบบซอฟต์แวร์

สำหรับการออกแบบซอฟต์แวร์จะคำนึงถึง 4 มุมมองดังรูปที่ในด้านล่าง

	Dynamic	Static
External	Use cases and Sequence diagrams	Component diagrams
Internal	Activity diagrams	Class diagrams

รูปที่ 4.2 มุมมองในการออกแบบซอฟต์แวร์

โดยในแต่ละมุมมองมีความหมายดังต่อไปนี้

External-Dynamic จะแสดงการติดต่อกับภายนอก เช่น บริการต่างๆ, ข้อความ

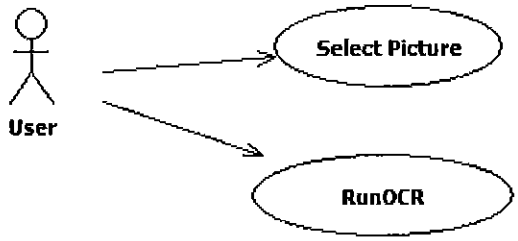
External-Static จะแสดงโครงสร้างและส่วนประกอบต่างๆ อย่างคร่าวๆ

Internal-Dynamic จะแสดงพฤติกรรมของระบบและสถานะต่างๆ ของระบบ

Internal-Static จะแสดงโครงสร้างภายในของระบบ

14 998145
วร.
ส729ก
2551

4.3.1 ยูสเคลส ไดอะแกรม (Use Cases Diagram)

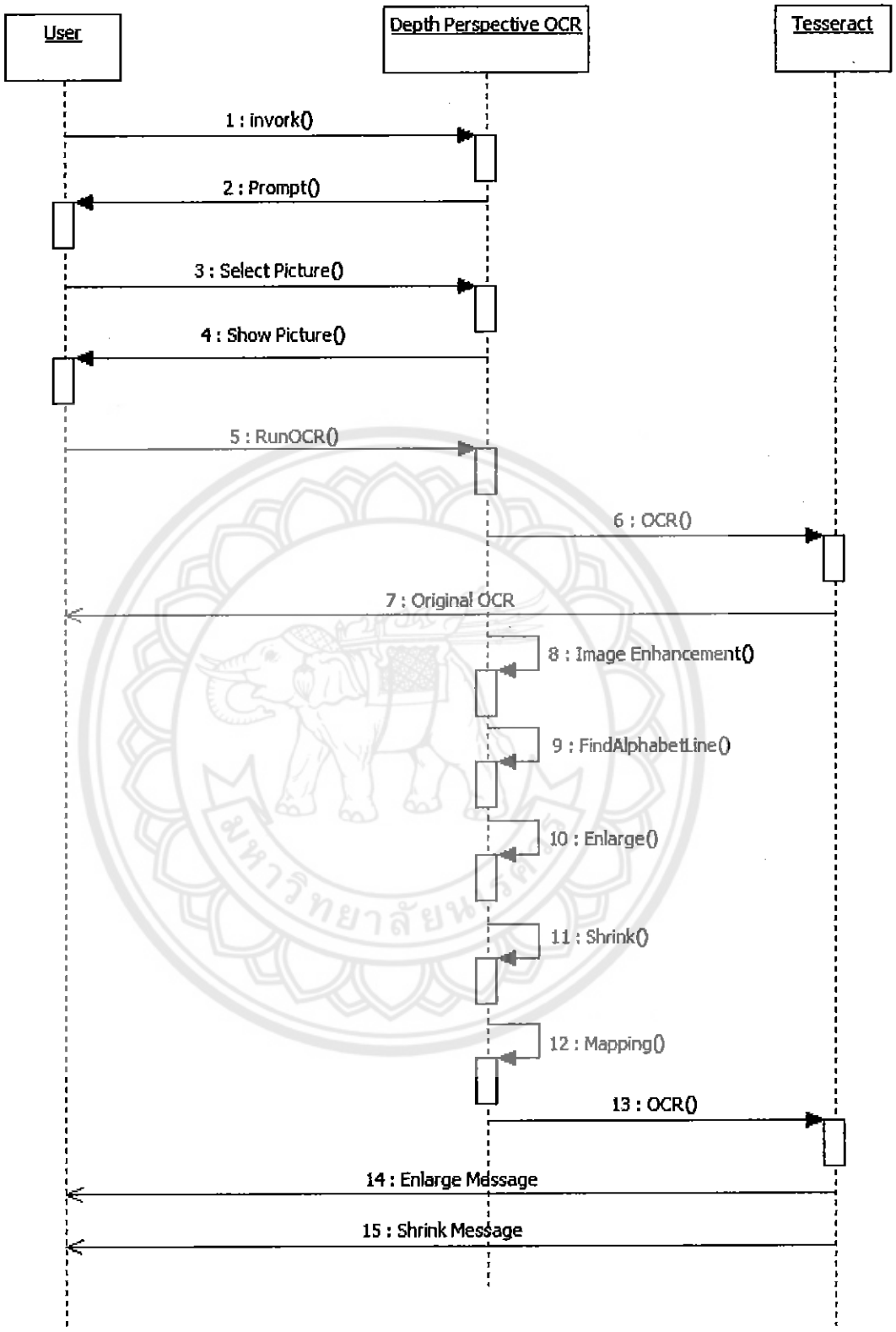


รูปที่ 4.3 Use Cases Diagram

โดยการติดต่อกับผู้ใช้จะมีเพียง 2 ขั้นตอนคือ ผู้ใช้ทำการเลือกรูปภาพที่ต้องการทำการรู้จำ และผู้ใช้สั่งให้โปรแกรมทำการประมวลผล

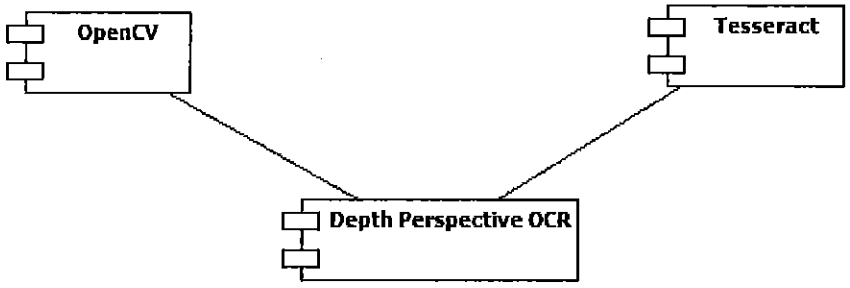
4.3.2 ซีเควน ไดอะแกรม (Sequence Diagram)

ลำดับการทำงานของโปรแกรมนี้ เริ่มจากการที่ผู้ใช้ทำการเรียกโปรแกรมขึ้นมา จากนั้นจึงทำการเลือกภาพที่ต้องการรู้จำ เมื่อได้ภาพที่ต้องการแล้วผู้ใช้จึงสั่งให้โปรแกรมทำการประมวลผล โดยโปรแกรมจะเริ่มทำการรู้จำภาพที่ต้องการก่อนเพื่อหาผลลัพธ์ในลักษณะปกติ จากนั้นโปรแกรมจะทำการปรับปรุงรูปภาพเพื่อให้ง่ายต่อการประมวลผล แล้วจึงทำการค้นหาบรรทัดของตัวอักษร เมื่อได้บรรทัดของตัวอักษรแล้วจึงทำการปรับปรุงรูปภาพก่อนทำการรู้จำโดยวิธีการย่อและขยายบรรทัดตัวอักษร จากนั้นทำการปรับบรรทัดตัวอักษรให้อยู่ในแนวระดับ แล้วจึงนำผลลัพธ์ที่ได้ไปทำการรู้จำ เมื่อทำการรู้จำเสร็จสิ้นโปรแกรมจะทำการหาบรรทัดของตัวอักษรและทำการตามกระบวนการต่อไปจนกระทั่งไม่พบบรรทัดของตัวอักษร ดังรูปที่ 4.4



รูปที่ 4.4 Sequence Diagram

4.3.3 คอมโพเนนท์ ไดอะแกรม (Component Diagram)



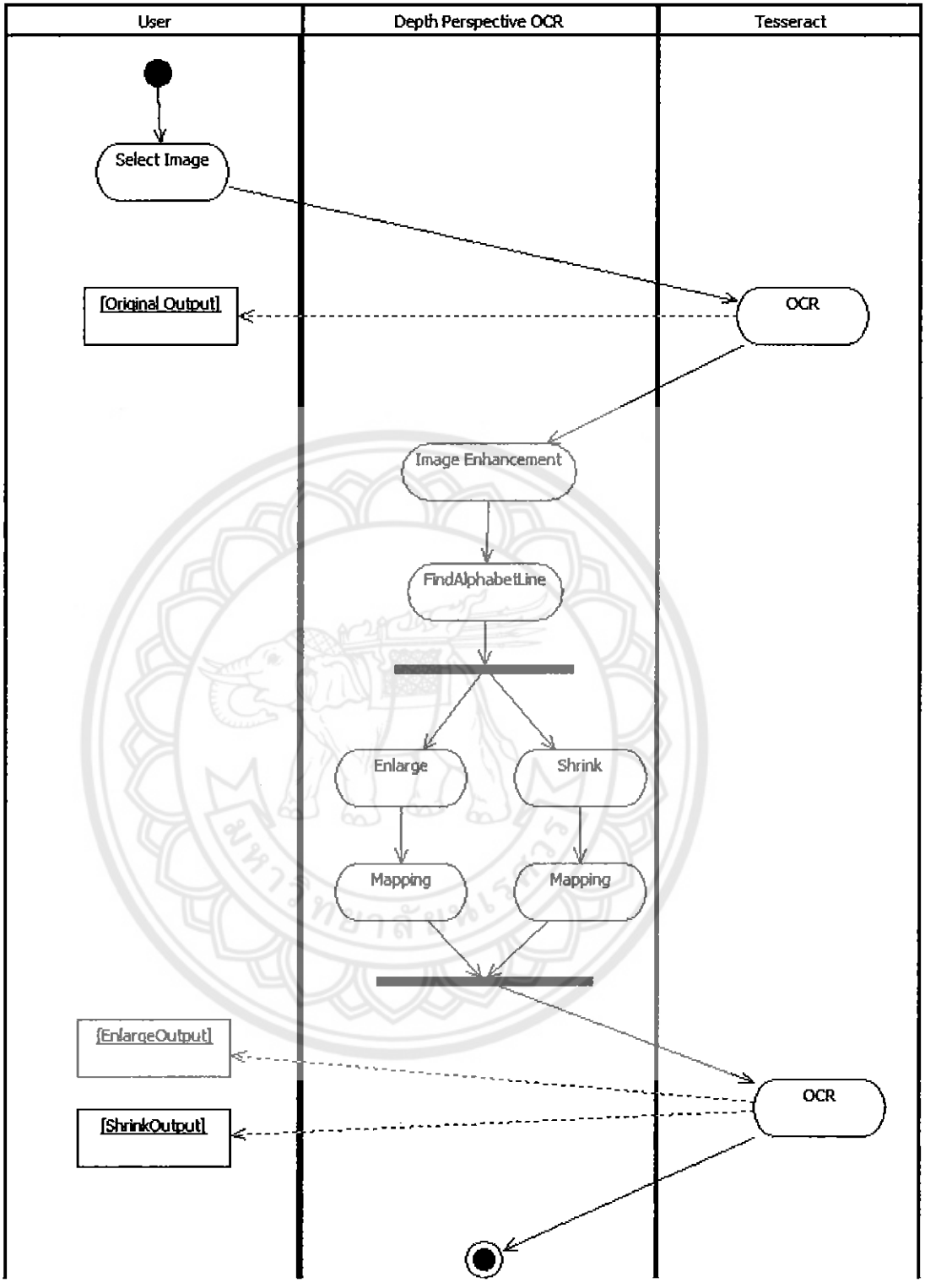
รูปที่ 4.5 Component Diagram

สำหรับ Component ในโปรแกรมจะมีเพียง 3 component คือ

- OpenCV ทำหน้าที่ด้านการประมวลผลด้านระบบการมองเห็นด้วยคอมพิวเตอร์
- Depth Perspective OCR ทำหน้าที่ด้านการประมวลผลสำหรับการรู้จำจากภาพตัวอักษรที่มีระดับความลึกของตัวอักษรในภาพแตกต่างกัน
- Tesseract ทำหน้าที่ด้านการรู้จำตัวอักษร

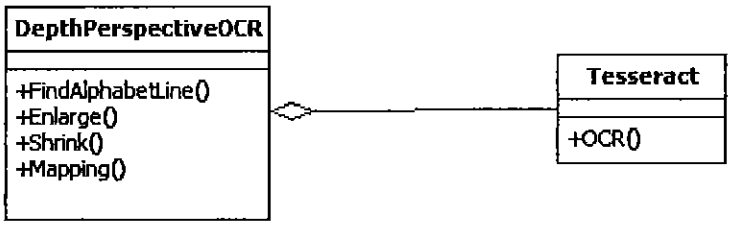
4.3.4 แอคทีวิตี ไดอะแกรม (Activity Diagram)

สำหรับกิจกรรมต่างๆ ในโปรแกรมจะเริ่มจาก User ทำการเลือกรูปภาพที่ต้องการรู้จำ จากนั้นจะนำภาพที่ต้องการไปทำการรู้จำและส่งผลที่ได้สู่ User แล้วจึงทำการปรับปรุงรูปภาพเพื่อความสะดวกต่อการประมวลผล ทำการค้นหาเส้นบรรทัดตัวอักษรในภาพ เมื่อพบเส้นบรรทัดตัวอักษรแล้วจึงทำการย่อและขยายเส้นบรรทัดนั้นแล้วทำการปรับบรรทัดตัวอักษร เพื่อให้เส้นบรรทัดตัวอักษรกลับมาในแนวระดับ จากนั้นทำการส่งภาพที่ได้เข้าสู่การรู้จำตัวอักษร แล้วจึงส่งผลที่ได้จากการรู้จำตัวอักษรคืนสู่ User ซึ่งแสดงดังรูปที่ 4.6



รูปที่ 4.6 Activity Diagram

4.3.5 คลาส ไดอะแกรม (Class Diagram)



รูปที่ 4.7 Class Diagram

ใน Class Diagram จะแบ่งออกเป็น 2 class หลักๆ คือ

- DepthPerspectiveOCR ทำหน้าที่ในการประมวลผลรูปภาพก่อนทำการรู้จำ
- Tesseract ทำหน้าที่ในการรู้จำภาพตัวอักษร

บทที่ 5

ผลการทดลอง

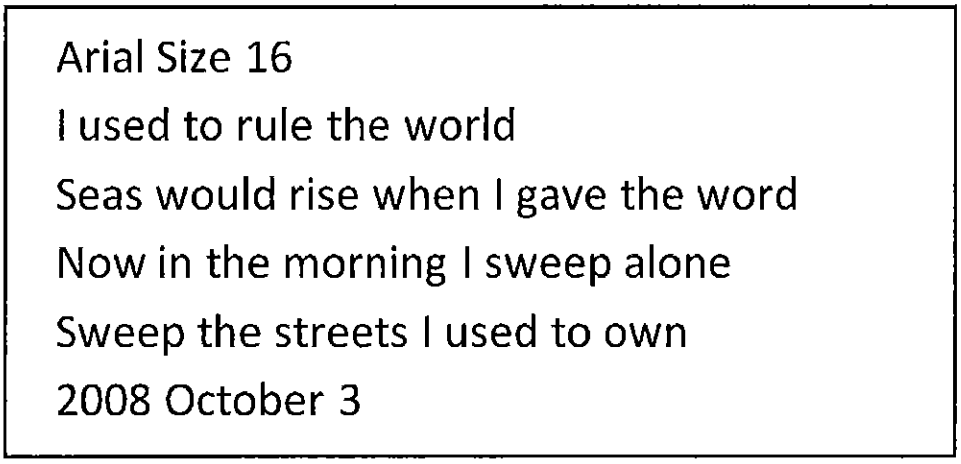
การทดลองความสามารถในการรู้จำจากภาพที่มีระดับความลึกของตัวอักษรในภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้างจะแบ่งการทดสอบออกเป็น 2 ส่วนคือ การทดสอบการรู้จำด้วยภาพถ่ายเอกสารและการทดสอบการรู้จำด้วยภาพทะเบียนรถยนต์

5.1 การทดสอบการรู้จำด้วยภาพถ่ายเอกสาร

ในการทดสอบนี้ การถ่ายภาพจะใช้กล้องดิจิตอลขนาดความละเอียด 5 ล้านพิกเซลล์ ถ่ายภาพขนาด 1600 x 1200 พิกเซลล์ และใช้ภาพตัวอักษรแบบ Calibri ขนาด 16, 20 และ 24 ในการทดสอบ โดยทำการถ่ายภาพที่ระยะห่าง 10, 20, 30, 40, 50 และ 60 เซนติเมตร และมุมในการถ่ายระหว่าง 0 – 70 องศา การถ่ายจะเพิ่มมุมเอียงทีละ 10 องศา โดยจะถ่ายภาพเอียงทั้งด้านซ้ายและด้านขวา สำหรับภาพที่ใช้ในการทดสอบแสดงในรูปที่ 5.1

จุดประสงค์ของการทดสอบการรู้จำด้วยภาพถ่ายเอกสาร มีดังต่อไปนี้

1. ทดสอบความสามารถในการรู้จำตัวอักษรเทียบกับระยะห่างในการถ่ายภาพ
2. ทดสอบความสามารถในการรู้จำตัวอักษรเทียบกับมุมในการถ่ายภาพ
3. ทดสอบความสามารถในการรู้จำตัวอักษรเทียบกับขนาดของตัวอักษรในการถ่ายภาพ
4. ทดสอบความสามารถในการรู้จำตัวอักษรระหว่างวิธีการย่อและการขยายบรรทัดตัวอักษร
5. ทดสอบความสามารถในการรู้จำตัวอักษรเทียบกับโปรแกรม Tesseract OCR



รูปที่ 5.1 ภาพถ่ายเอกสารที่ใช้ในการทดสอบ

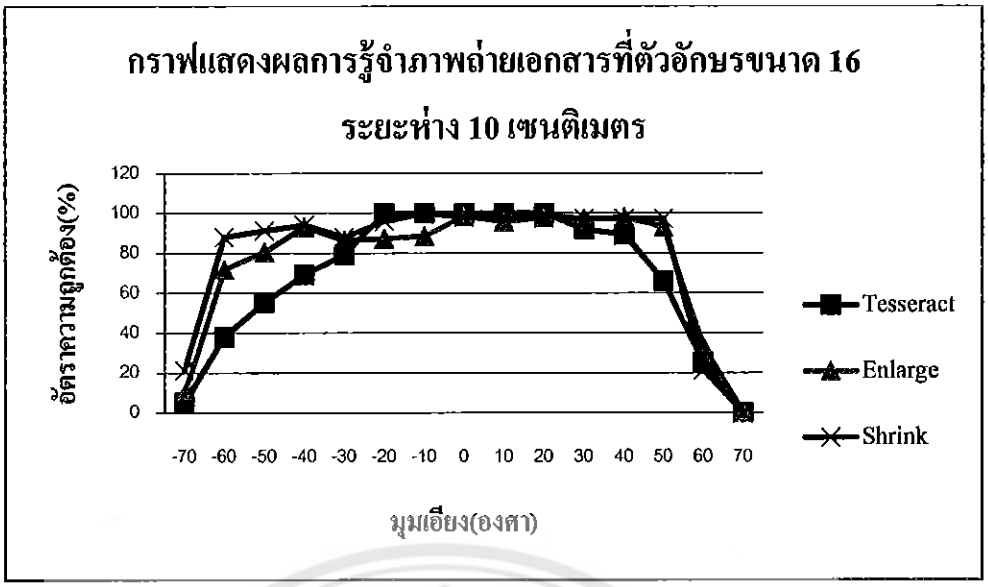
ผลการทดสอบจะแสดงเป็น 2 ส่วนคือ ตารางแสดงผลการทดสอบและกราฟแสดงผลการทดสอบสำหรับในแต่ละกรณี โดยจะแสดงค่าความถูกต้องของการรู้จำออกเป็น 3 ส่วนคือ

- Tesseract แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพเข้าสู่การรู้จำโดยโปรแกรม Tesseract OCR โดยตรง
- Enlarge แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพบรรทัดตัวอักษรมาทำการขยายบรรทัดตัวอักษร แล้วจึงทำการรู้จำ
- Shrink แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพบรรทัดตัวอักษรมาทำการย่อบรรทัดตัวอักษร แล้วจึงทำการรู้จำ

สำหรับกรณีที่ตัวอักษรขนาด 20 และ 24 ที่ระยะห่าง 10 เซนติเมตร ไม่ได้ทำการทดสอบเนื่องจากกล้องดิจิตอลไม่สามารถถ่ายภาพทั้งหมดได้ และสำหรับกรณีที่ตัวอักษรขนาด 16 ที่ระยะห่าง 60 เซนติเมตร ไม่ได้ทำการทดสอบเนื่องจากขนาดของตัวอักษรในภาพที่ได้จากการถ่ายนั้นเล็กเกินกว่าที่จะทำการรู้จำได้

ตารางที่ 5.1 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 10 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 16		
		Tesseract	Enlarge	Shrink
10	-70	5.333333	9.333333	21.33333
	-60	38	72	88
	-50	55.33333	80.66667	91.33333
	-40	69.33333	93.33333	94
	-30	79.33333	86.66667	88
	-20	100	87.33333	96
	-10	100	88.66667	100
	0	100	98.66667	98
	10	100	96	97.33333
	20	100	98	98.66667
	30	92	97.33333	97.33333
	40	89.33333	98	97.33333
	50	66	93.33333	97.33333
	60	25.33333	32.66667	21.33333
	70	0	0	0



รูปที่ 5.2 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 10 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 16 และระยะห่าง 10 เซนติเมตร โปรแกรมสามารถทำการรู้จำตัวอักษรได้ดีเมื่อมุมในการถ่ายภาพไม่เกิน 60 องศา และดีกว่าโปรแกรม Tesseract OCR อย่างชัดเจน

ตารางที่ 5.2 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 20 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 16		
		Tesseract	Enlarge	Shrink
20	-70	42	41.33333	55.33333
	-60	77.33333	80.66667	85.33333
	-50	96	94	95.33333
	-40	95.33333	97.33333	99.33333
	-30	99.33333	98.66667	100
	-20	100	98	100
	-10	100	97.33333	100
	0	99.33333	98.66667	99.33333
	10	100	98.66667	99.33333
	20	98.66667	100	100
	30	99.33333	98.66667	98.66667
	40	96.66667	100	98.66667
	50	98.66667	98	98
	60	36.66667	71.33333	79.33333
	70	40.66667	59.33333	56.66667

รูปที่ 5.3 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 20 **เซนติเมตร**

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 16 **และระยะห่าง** 20 **เซนติเมตร** โปรแกรมสามารถทำการรู้จำตัวอักษรได้ความถูกต้องใกล้เคียงกับโปรแกรม Tesseract OCR แต่ที่มุม 60 - 70 องศา โปรแกรมสามารถทำการรู้จำได้ดีกว่า

ตารางที่ 5.3 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 30 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 16		
		Tesseract	Enlarge	Shrink
30	-70	0	0	0
	-60	76	91.33333	86
	-50	98.66667	95.33333	96.66667
	-40	94	96.66667	100
	-30	98.66667	98	99.33333
	-20	99.33333	99.33333	100
	-10	98.66667	100	99.33333
	0	100	100	98.66667
	10	100	99.33333	98
	20	98	98	100
	30	99.33333	98.66667	97.33333
	40	98.66667	98.66667	99.33333
	50	92	95.33333	97.33333
	60	86.66667	79.33333	83.33333
	70	0	22.66667	0

รูปที่ 5.4 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 30 **เซนติเมตร**

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 16 **และระยะห่าง 30 เซนติเมตร** โปรแกรมสามารถทำการรู้จำตัวอักษรได้ความถูกต้องดีเหมือนกับโปรแกรม Tesseract OCR

ตารางที่ 5.4 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 40 เซนติเมตร

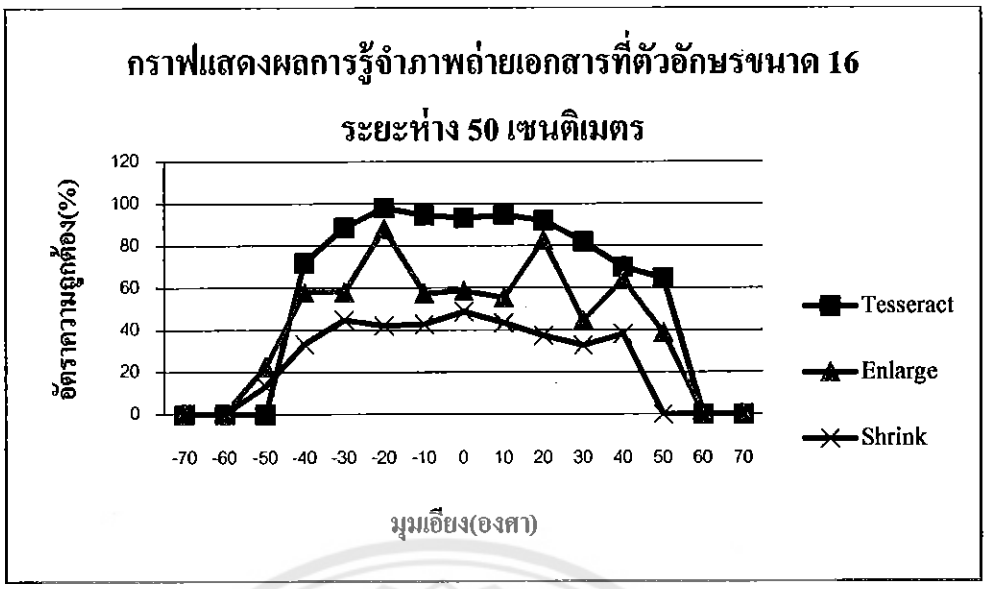
ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 16		
		Tesseract	Enlarge	Shrink
40	-70	0	0	0
	-60	39.33333	42.66667	24.66667
	-50	86.66667	84	86
	-40	98	96	91.33333
	-30	96	98.66667	94
	-20	96.66667	95.33333	94.66667
	-10	100	98.66667	98.66667
	0	100	98.66667	97.33333
	10	100	100	96.66667
	20	98.66667	97.33333	90.66667
	30	95.33333	95.33333	94.66667
	40	88.66667	92	85.33333
	50	82	83.33333	69.33333
	60	57.33333	39.33333	38
	70	0	0	0

รูปที่ 5.5 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 40 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 16 และระยะห่าง 40 เซนติเมตร โปรแกรมสามารถทำการรู้จำตัวอักษรได้ความถูกต้องใกล้เคียงกับโปรแกรม Tesseract OCR

ตารางที่ 5.5 ผลการทดสอบที่ตัวอักษรขนาด 16 และระยะห่าง 50 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 16		
		Tesseract	Enlarge	Shrink
50	-70	0	0	0
	-60	0	0	0
	-50	0	22.66667	13.33333
	-40	72	58	33.33333
	-30	88.66667	58	44.66667
	-20	98	88	42
	-10	94.66667	57.33333	42.66667
	0	93.33333	58.66667	48.66667
	10	94.66667	55.33333	43.33333
	20	92	82.66667	37.33333
	30	82	44.66667	32.66667
	40	70	64	38
	50	64.66667	38.66667	0
	60	0	0	0
	70	0	0	0



รูปที่ 5.6 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 16 และระยะห่าง 50 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 16 และระยะห่าง 50 เซนติเมตร โปรแกรมสามารถทำการรู้จำตัวอักษรไม่ดีอันเนื่องมาจากขนาดของตัวอักษรในภาพที่มีขนาดเล็กอันเนื่องมาจากการถ่ายภาพในระยะไกล

ตารางที่ 5.6 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 20 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 20		
		Tesseract	Enlarge	Shrink
20	-70	12	20.66667	13.33333
	-60	68	60.66667	81.33333
	-50	82.66667	92.66667	88.66667
	-40	82	93.33333	91.33333
	-30	75.33333	94	92
	-20	88	90.66667	92
	-10	91.33333	88.66667	92
	0	91.33333	92.66667	89.33333
	10	90.66667	88.66667	87.33333
	20	91.33333	88.66667	88.66667
	30	91.33333	94	92.66667
	40	74.66667	94	94
	50	86.66667	92.66667	91.33333
	60	65.33333	80.66667	90.66667
	70	18	50	82.66667

รูปที่ 5.7 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 20 **เซนติเมตร**

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 20 **และระยะห่าง** 20 **เซนติเมตร** โปรแกรมสามารถทำการรู้จำตัวอักษรได้ดีกว่าโปรแกรม Tesseract OCR **และสามารถทำการรู้จำ**ภาพที่มุม 70 องศาได้ดีอีกด้วย

ตารางที่ 5.7 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 30 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 20		
		Tesseract	Enlarge	Shrink
30	-70	50	58.66667	63.33333
	-60	74	78.66667	90.66667
	-50	86	91.33333	91.33333
	-40	91.33333	92.66667	92
	-30	89.33333	95.33333	92.66667
	-20	93.33333	94.66667	93.33333
	-10	87.33333	91.33333	92.66667
	0	85.33333	90.66667	92.66667
	10	96.66667	87.33333	94.66667
	20	96.66667	92	92.66667
	30	87.33333	94	92
	40	88.66667	93.33333	91.33333
	50	92	95.33333	92.66667
	60	78.66667	55.33333	80.66667
	70	0	28.66667	41.33333

รูปที่ 5.8 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 30 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 20 และระยะห่าง 30 เซนติเมตร โปรแกรมสามารถทำการรู้จำตัวอักษรได้ดีเมื่อมุมในการถ่ายภาพไม่เกิน 60 องศา และได้ความถูกต้องดีกว่าโปรแกรม Tesseract OCR

ตารางที่ 5.8 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 40 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 20		
		Tesseract	Enlarge	Shrink
40	-70	0	13.33333	16
	-60	70	64	73.33333
	-50	85.33333	92	92
	-40	88.66667	96.66667	96
	-30	94.66667	96.66667	94
	-20	93.33333	93.33333	79.33333
	-10	92.66667	95.33333	90
	0	92.66667	94.66667	92
	10	91.33333	94	93.33333
	20	91.33333	97.33333	91.33333
	30	92	94	94.66667
	40	86	90.66667	81.33333
	50	84	88.66667	88.66667
	60	73.33333	66	66
	70	8.666667	16.66667	6.666667

รูปที่ 5.9 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 40 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 20 และระยะห่าง 40 เซนติเมตร โปรแกรมสามารถทำการรู้จำตัวอักษรได้ความถูกต้องดีกว่าโปรแกรม Tesseract OCR และวิธีการขยายบรรทัดตัวอักษรจะให้ผลที่ค่อนข้างดีกว่าวิธีการย่อบรรทัดตัวอักษร

ตารางที่ 5.9 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 50 เซนติเมตร

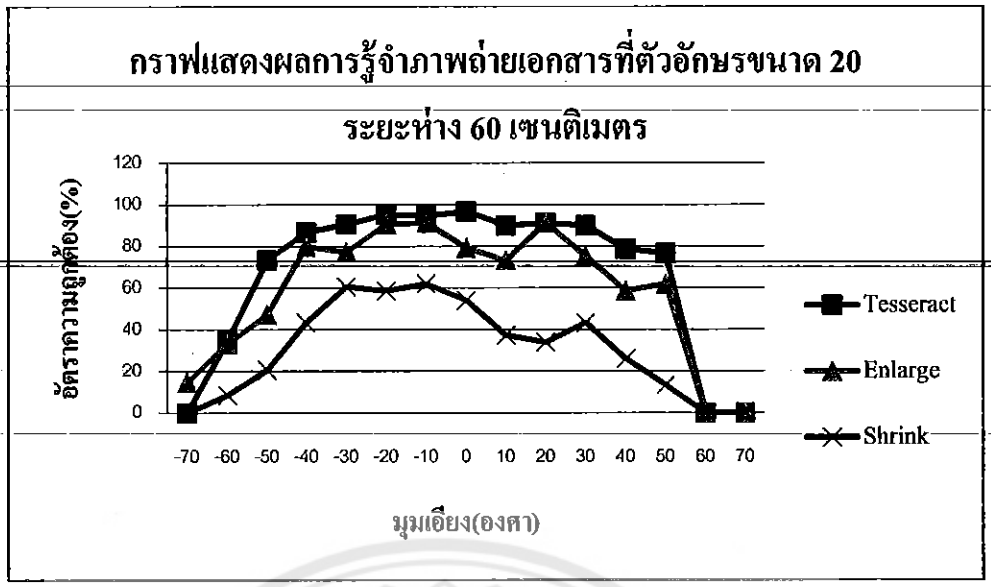
ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 20		
		Tesseract	Enlarge	Shrink
50	-70	0	0	0
	-60	42	34	24.66667
	-50	80	86	52.66667
	-40	88	88	66
	-30	91.33333	94	92
	-20	97.33333	92.66667	91.33333
	-10	98	94	88
	0	96	92	82
	10	89.33333	94.66667	70.66667
	20	96	94.66667	89.33333
	30	94.66667	90	80
	40	88	78.66667	62.66667
	50	81.33333	80.66667	52
	60	32.66667	35.33333	18.66667
	70	0	0	0

รูปที่ 5.10 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 50 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 20 และระยะห่าง 50 เซนติเมตร เมื่อใช้วิธีการขยายบรรทัดตัวอักษรจะได้ผลที่ใกล้เคียงกับโปรแกรม Tesseract OCR ในขณะที่วิธีการย่อบรรทัดตัวอักษรจะให้ผลที่แย่กว่าอย่างชัดเจน

ตารางที่ 5.10 ผลการทดสอบที่ตัวอักษรขนาด 20 และระยะห่าง 60 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 20		
		Tesseract	Enlarge	Shrink
60	-70	0	14.66667	0
	-60	34.66667	33.33333	8.666667
	-50	73.33333	47.33333	20.66667
	-40	86.66667	80	43.33333
	-30	90.66667	77.33333	60.66667
	-20	95.33333	90.66667	58.66667
	-10	94.66667	91.33333	62
	0	96.66667	79.33333	54
	10	90	73.33333	37.33333
	20	91.33333	91.33333	34
	30	90	75.33333	43.33333
	40	78.66667	58.66667	26
	50	76.66667	62	13.33333
	60	0	0	0
	70	0	0	0

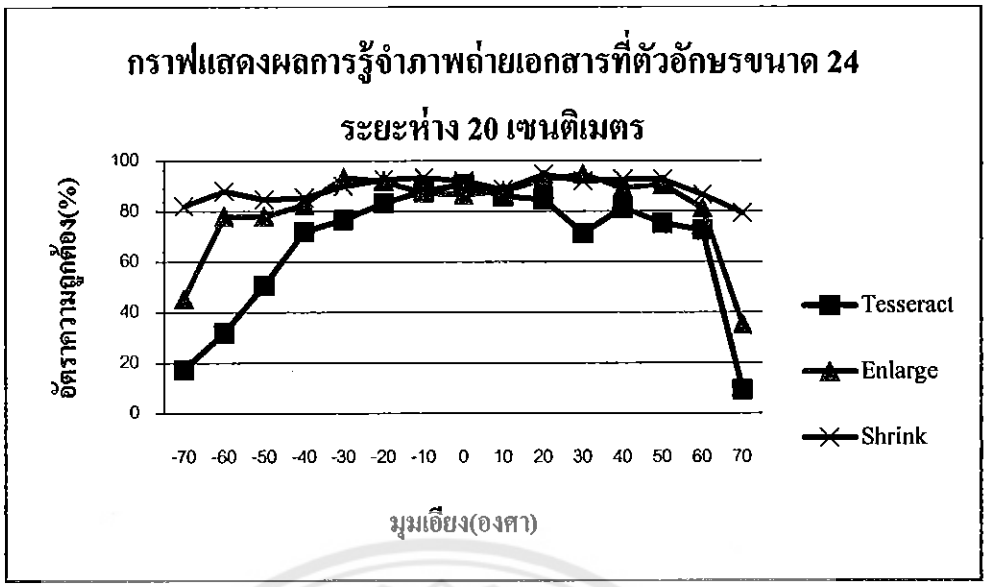


รูปที่ 5.11 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 20 และระยะห่าง 60 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 20 และระยะห่าง 60 เซนติเมตร โปรแกรมสามารถทำการรู้จำตัวอักษรไม่ดีเป็นส่วนใหญ่อันเนื่องมาจากขนาดของตัวอักษรในภาพที่มีขนาดเล็กอันเนื่องมาจากการถ่ายภาพในระยะไกล และจะเห็นได้ว่าเมื่อขนาดตัวอักษรในภาพมีขนาดเล็ก วิธีการย่อบรรทัดตัวอักษรจะให้ผลที่แย่กว่าวิธีการขยายบรรทัดตัวอักษร

ตารางที่ 5.11 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 20 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 24		
		Tesseract	Enlarge	Shrink
20	-70	17.33333	45.33333	82
	-60	32	78	88
	-50	50.66667	78	84.66667
	-40	72	82.66667	85.33333
	-30	76.66667	93.33333	90
	-20	83.33333	92	92.66667
	-10	88	87.33333	93.33333
	0	90.66667	86.66667	92
	10	86	88	88.66667
	20	84.66667	92.66667	94.66667
	30	71.33333	94.66667	92
	40	81.33333	89.33333	92.66667
	50	75.33333	90.66667	92.66667
	60	72.66667	81.33333	86.66667
	70	9.333333	35.33333	79.33333

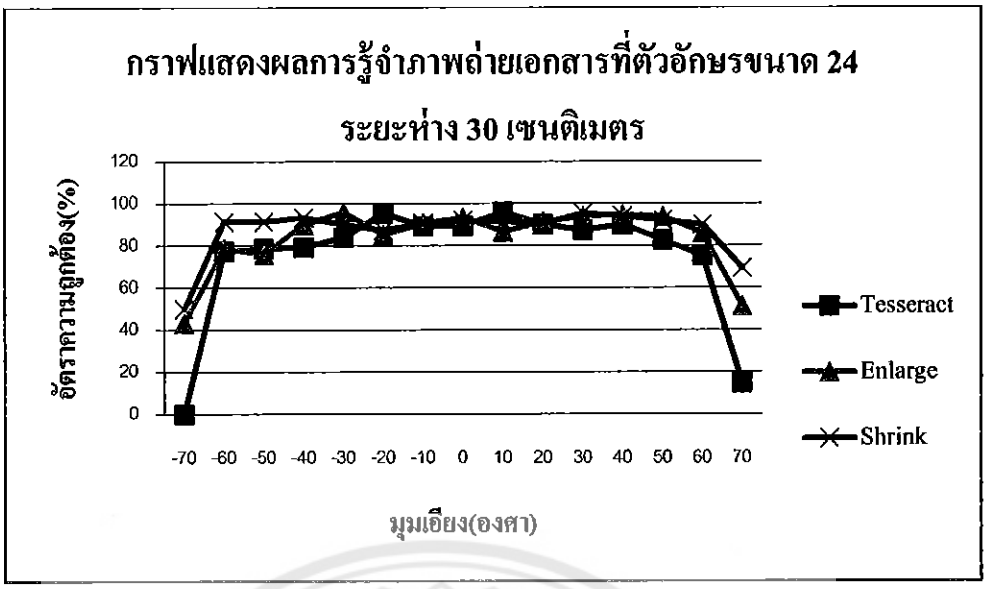


รูปที่ 5.12 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 20 **เซนติเมตร**

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 24 **และระยะห่าง** 20 **เซนติเมตร** โปรแกรมสามารถทำการรู้จำตัวอักษรได้ความถูกต้องดีกว่าโปรแกรม Tesseract OCR **และยังสามารถทำการรู้จำตัวอักษรที่มุมในการถ่ายภาพเท่ากับ** 70 **องศาได้อย่างดีอีกด้วย**

ตารางที่ 5.12 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 30 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 24		
		Tesseract	Enlarge	Shrink
30	-70	0	43.33333	50
	-60	77.33333	78.66667	91.33333
	-50	78.66667	76	91.33333
	-40	79.33333	90	93.33333
	-30	84	96	90
	-20	95.33333	85.33333	87.33333
	-10	89.33333	90.66667	90.66667
	0	89.33333	93.33333	92
	10	96	86.66667	91.33333
	20	90	91.33333	90.66667
	30	87.33333	94	95.33333
	40	90	94.66667	94
	50	82.66667	94	92
	60	75.33333	86	90
	70	14.66667	51.33333	69.33333

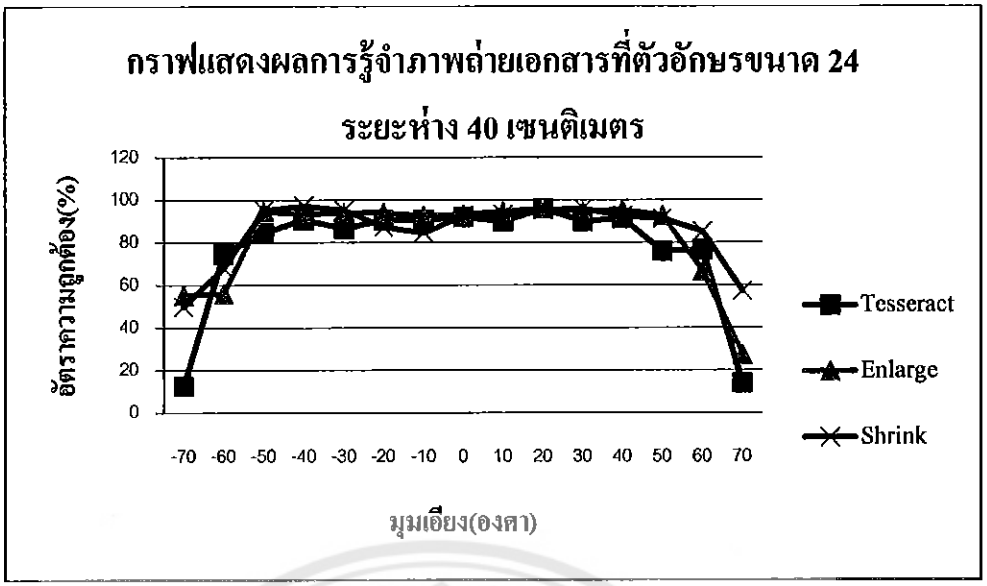


รูปที่ 5.13 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 30 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 24 และระยะห่าง 30 เซนติเมตร โปรแกรมสามารถทำการรู้จำตัวอักษรได้ความถูกต้องดีกว่าโปรแกรม Tesseract OCR แต่การรู้จำจะให้ผลอยู่ในระดับที่ดีเมื่อมุมในการถ่ายภาพไม่เกิน 60 องศา

ตารางที่ 5.13 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 40 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 24		
		Tesseract	Enlarge	Shrink
40	-70	12.66667	55.33333	50
	-60	74.66667	56	68
	-50	84.66667	94.66667	95.33333
	-40	90.66667	93.33333	97.33333
	-30	86.66667	94	95.33333
	-20	90.66667	94	87.33333
	-10	90.66667	92.66667	84.66667
	0	92	93.33333	92.66667
	10	90	94.66667	93.33333
	20	96	96	95.33333
	30	90	94.66667	96
	40	91.33333	95.33333	92.66667
	50	76	92.66667	91.33333
	60	76.66667	66.66667	85.33333
	70	14	27.33333	57.33333

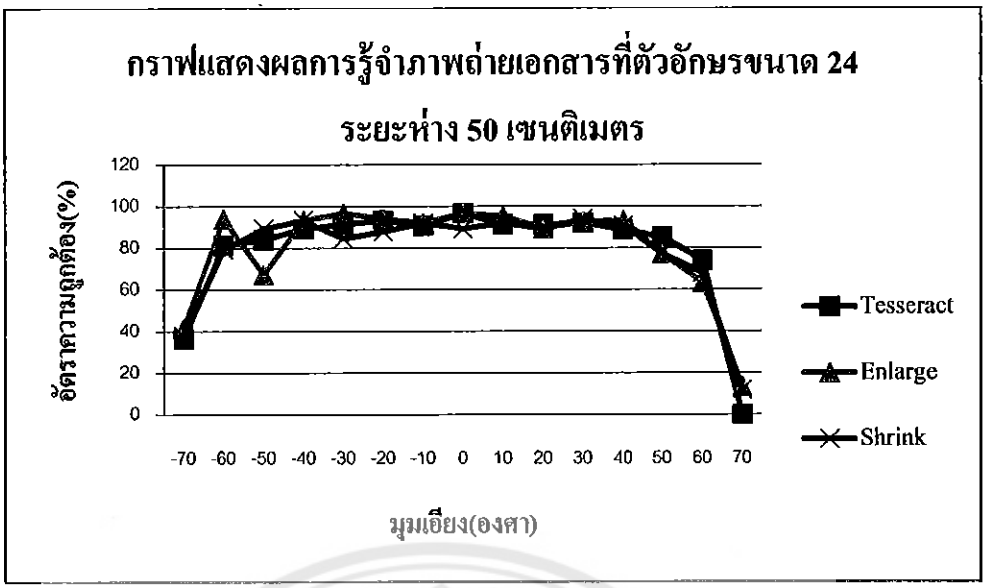


รูปที่ 5.14 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 40 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 24 และระยะห่าง 40 เซนติเมตร โปรแกรมสามารถทำการรู้จำตัวอักษรได้ความถูกต้องดีกว่าโปรแกรม Tesseract OCR

ตารางที่ 5.14 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 50 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 24		
		Tesseract	Enlarge	Shrink
50	-70	36.66667	41.33333	38
	-60	81.33333	94	79.33333
	-50	84	67.33333	89.33333
	-40	89.33333	93.33333	93.33333
	-30	91.33333	96.66667	84.66667
	-20	93.33333	94	88
	-10	90.66667	92	92
	0	96.66667	96.66667	89.33333
	10	91.33333	95.33333	92
	20	91.33333	89.33333	89.33333
	30	92	92.66667	94
	40	88.66667	93.33333	90.66667
	50	85.33333	77.33333	77.33333
	60	74	63.33333	68
	70	0	13.33333	12



รูปที่ 5.15 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 50 **เซนติเมตร**

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 24 **และระยะห่าง 50 เซนติเมตร** โปรแกรมสามารถทำการรู้จำตัวอักษรได้ความถูกต้องดีกว่าโปรแกรม Tesseract OCR

ตารางที่ 5.15 ผลการทดสอบที่ตัวอักษรขนาด 24 และระยะห่าง 60 เซนติเมตร

ระยะห่าง (เซนติเมตร)	มุมเอียง (องศา)	ตัวอักษรขนาด 24		
		Tesseract	Enlarge	Shrink
60	-70	0	0	0
	-60	53.33333	47.33333	38.66667
	-50	83.33333	85.33333	78.66667
	-40	87.33333	90	68
	-30	89.33333	96	88
	-20	94.66667	95.33333	95.33333
	-10	95.33333	88	83.33333
	0	95.33333	91.33333	88
	10	94	89.33333	83.33333
	20	88.66667	94.66667	88
	30	92	92	81.33333
	40	88	83.33333	85.33333
	50	75.33333	74.66667	44.66667
	60	52.66667	43.33333	32.66667
	70	0	0	0

รูปที่ 5.16 กราฟผลการทดสอบการรู้จำที่ตัวอักษรขนาด 24 และระยะห่าง 60 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองจะเห็นว่าที่ตัวอักษรขนาด 24 และระยะห่าง 60 เซนติเมตร จะเห็นว่าวิธีการขยายบรรทัดตัวอักษรจะให้ผลที่ค่อนข้างใกล้เคียงกับโปรแกรม Tesseract OCR

เนื่องจากข้อมูลที่ได้จากการทดสอบมีจำนวนมากเพราะมีตัวแปลที่เกี่ยวข้องกับการทดสอบเป็นจำนวนมาก จึงทำการวิเคราห์ข้อมูลและสรุปข้อมูลออกเป็น 3 ส่วนคือ

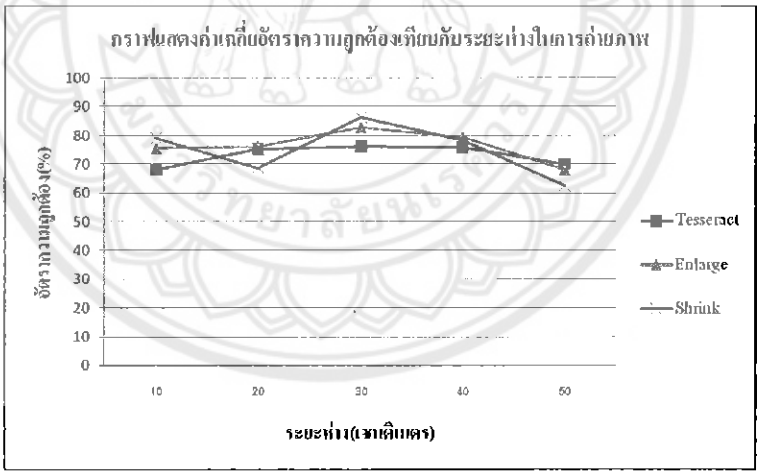
- ค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ
- ค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ
- ค่าเฉลี่ยอัตราความถูกต้องเทียบกับขนาดของตัวอักษรในการถ่ายภาพ

5.1.1 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่าย

ระยะห่างในการถ่ายมีผลกระทบต่อภาพตัวอักษรที่ได้หลายด้าน เช่น ทำให้ตัวอักษรมีขนาดเล็กลง, ความกว้างของตัวอักษรแต่ละตัวลดลง และทำให้รายละเอียดของภาพลดลง เป็นต้น จึงทำการสรุปผลได้ดังนี้

ตารางที่ 5.16 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ

ระยะห่าง (เซนติเมตร)	Tesseract	Enlarge	Shrink
10	68	75.466667	79.066667
20	75.274074	76.192593	68.77037
30	76.177778	82.82963	86.311111
40	75.822222	79.244444	78.325926
50	70.014815	68.044444	62.503704



รูปที่ 5.17 กราฟค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ

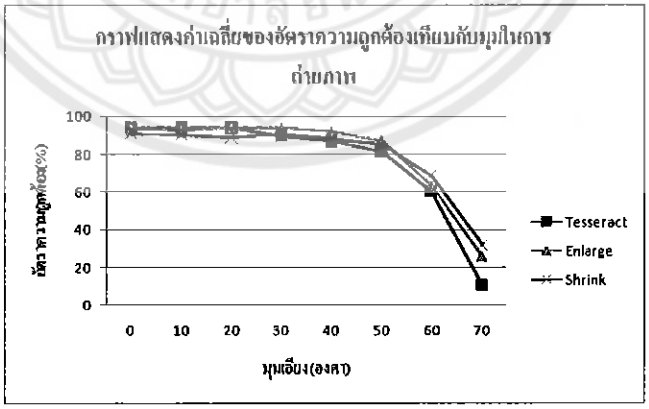
ผลที่ได้จากกราฟและตารางการแสดงผลการทดลองของค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพพบว่า โปรแกรมสามารถทำการรู้จำได้ดีเมื่อระยะในการถ่ายภาพไม่เกิน 30 เซนติเมตร

5.1.2 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ

การถ่ายภาพถ่ายเอกสารแบบด้านข้างทำให้เกิดผลกระทบต่างๆต่อรูปภาพตัวอักษร เช่น เกิดการซ้อนทับกันของบรรทัดตัวอักษร, ขนาดของตัวอักษรไม่เท่ากัน เป็นต้น จึงทำการสรุปความสามารถในการรู้จำภาพตัวอักษรตามมุมที่เปลี่ยนไปได้ดังนี้

ตารางที่ 5.17 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ

มุมเอียง (องศา)	Tesseract	Enlarge	Shrink
0	94.166667	93.444444	90.833333
10	94.055556	92.416667	90.166667
20	94.027778	93.916667	88.694444
30	89.694444	93.75	90.777778
40	86.916667	91.833333	88.055556
50	81.638889	87.194444	84.916667
60	60.527778	63.527778	68.583333
70	10.555556	25.972222	31.972222



รูปที่ 5.18 กราฟค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ

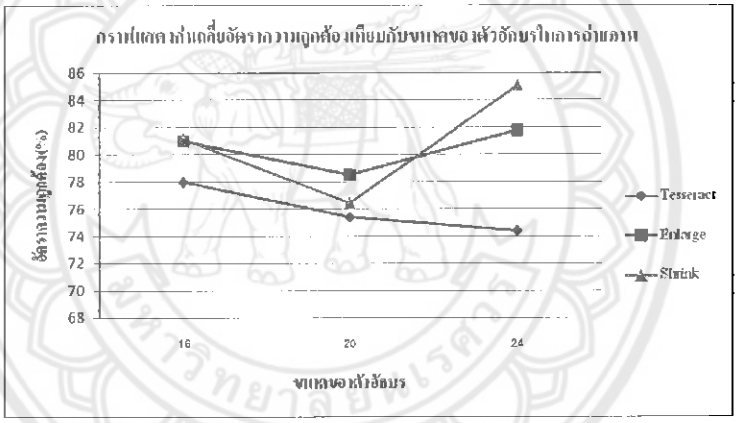
ผลที่ได้จากกราฟและตารางการแสดงผลการทดลองของค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพพบว่า โปรแกรมสามารถทำการรู้จำได้ดีเมื่อมุมในการถ่ายภาพไม่เกิน 60 องศา

5.1.3 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับขนาดของตัวอักษรในการถ่ายภาพ

ขนาดตัวอักษรที่ต่างกันทำให้เกิดผลกระทบต่อภาพตัวอักษรที่ได้เช่น ระยะห่างในการถ่ายภาพ, ขนาดของตัวอักษรในภาพถ่าย เป็นต้น จึงทำการสรุปความสามารถในการรู้จำภาพตามขนาดของตัวอักษรที่เปลี่ยนไปได้ดังนี้

ตารางที่ 5.18 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับขนาดของตัวอักษรในการถ่ายภาพ

ขนาดตัวอักษร	Tesseract	Enlarge	Shrink
16	77.9777778	80.9666667	81.1333333
20	75.4	78.4888889	76.4333333
24	74.3888889	81.7888889	85.0444444



รูปที่ 5.19 กราฟค่าเฉลี่ยอัตราความถูกต้องเทียบกับขนาดของตัวอักษรในการถ่ายภาพ

ผลที่ได้จากกราฟและตารางการแสดงผลการทดลองของค่าเฉลี่ยอัตราความถูกต้องเทียบกับขนาดของตัวอักษรในการถ่ายภาพพบว่า โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR และเมื่อขนาดของตัวอักษรเพิ่มขึ้น ความสามารถในการรู้จำจะเพิ่มขึ้นด้วย

5.2 การทดสอบการรู้จำด้วยภาพทะเบียนรถยนต์

สำหรับส่วนหนึ่งในการพัฒนาโครงงานนี้ จะนำผลที่ได้ไปทำการรู้จำภาพทะเบียนรถยนต์ แต่เนื่องจากโปรแกรมยังไม่สามารถรู้จำตัวอักษรภาษาไทยได้ จึงทำการรู้จำเฉพาะเลขทะเบียนรถยนต์เท่านั้น สำหรับการทดสอบจะใช้กล้อง 2 ชนิดคือ

- กล้องดิจิตอล ความละเอียด 5 ล้านพิกเซลล์ ถ่ายภาพขนาด 1600 x 1200 พิกเซลล์
- กล้องโทรศัพท์มือถือ ความละเอียด 1.3 ล้านพิกเซลล์ ถ่ายภาพขนาด 1280 x 960 พิกเซลล์

สำหรับการถ่ายภาพกำหนดระยะห่างที่ 50, 75, 125, … , 200 เซนติเมตร และมุมในการถ่ายระหว่าง 0 – 70 องศา การถ่ายจะเพิ่มมุมเอียงทีละ 10 องศา โดยถ่ายภาพเอียงทั้งด้านซ้ายและด้านขวา สำหรับป้ายทะเบียนที่ใช้ในการทดสอบแสดงดังรูปที่ 5.20

จุดประสงค์ของการทดสอบการรู้จำด้วยภาพทะเบียนรถยนต์ มีดังต่อไปนี้

1. ทดสอบความสามารถในการรู้จำเทียบกับระยะห่างในการถ่ายภาพ
2. ทดสอบความสามารถในการรู้จำเทียบกับมุมในการถ่ายภาพ
3. ทดสอบความสามารถในการรู้จำระหว่างวิธีการย่อและการขยายบรรทัดตัวอักษร
4. ทดสอบความสามารถในการรู้จำระหว่างภาพจากกล้องดิจิตอลและภาพจากกล้องโทรศัพท์มือถือ
5. ทดสอบความสามารถในการรู้จำเทียบกับโปรแกรม Tesseract OCR

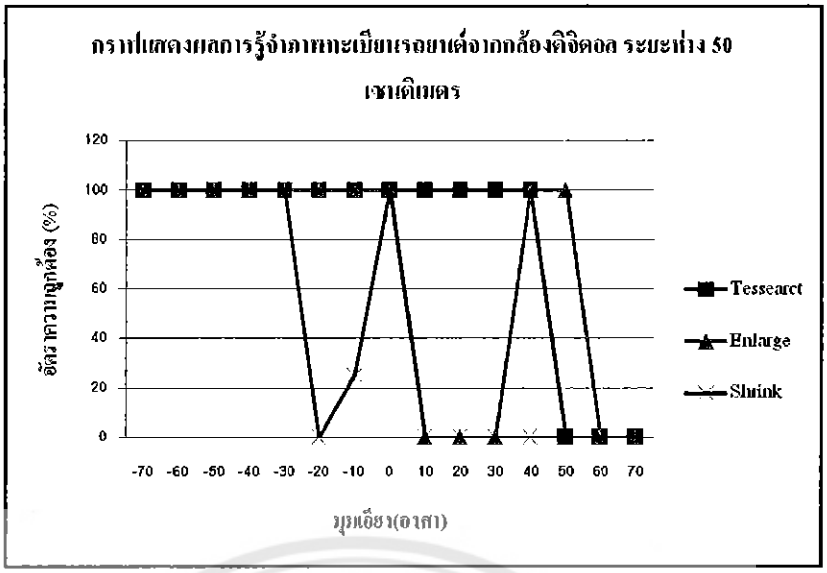
รูปที่ 5.20 ภาพป้ายทะเบียนที่ใช้ในการทดสอบ

ผลการทดสอบจะแสดงเป็น 2 ส่วนคือ ตารางแสดงผลการทดสอบและกราฟแสดงผลการทดสอบสำหรับในแต่ละกรณี โดยจะแสดงค่าความถูกต้องของการรู้จำออกเป็น 3 ส่วนคือ

- Tesseract แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพเข้าสู่การรู้จำโดยโปรแกรม Tesseract OCR โดยตรง
- Enlarge แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพบรรทัดตัวอักษรมาทำการขยายบรรทัดตัวอักษร แล้วจึงทำการรู้จำ
- Shrink แสดงค่าร้อยละของความถูกต้องที่เกิดจากการนำภาพบรรทัดตัวอักษรมาทำการย่อบรรทัดตัวอักษร แล้วจึงทำการรู้จำ

ตารางที่ 5.19 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 50 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องดิจิตอล		
		Tesseract	Enlarge	Shrink
50	-70	100	100	100
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	100
	-20	100	100	0
	-10	100	100	25
	0	100	100	100
	10	100	0	0
	20	100	0	0
	30	100	0	0
	40	100	100	0
	50	0	100	0
	60	0	0	0
	70	0	0	0

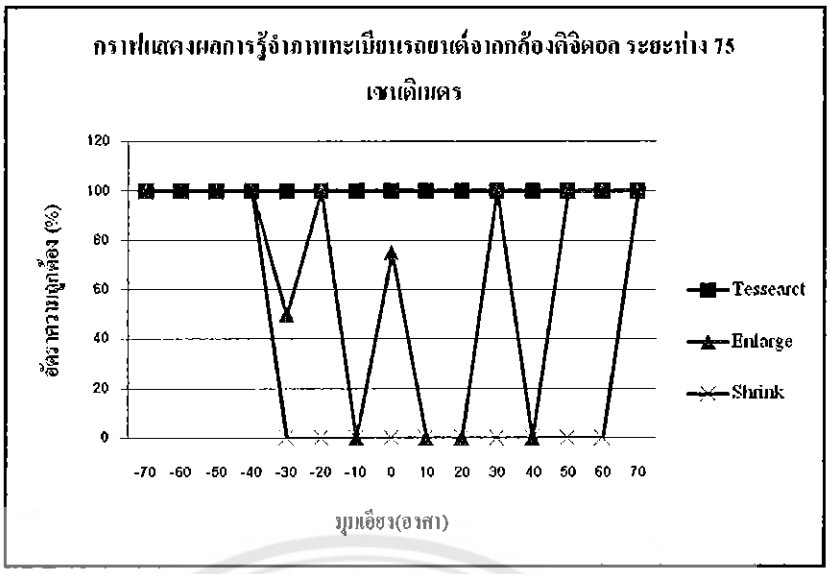


รูปที่ 5.21 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 50 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 50 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ใกล้เคียงกับโปรแกรม Tesseract OCR

ตารางที่ 5.20 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 75 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องดิจิตอล		
		Tesseract	Enlarge	Shrink
75	-70	100	100	100
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	50	0
	-20	100	100	0
	-10	100	0	0
	0	100	75	0
	10	100	0	0
	20	100	0	0
	30	100	100	0
	40	100	0	0
	50	100	100	0
	60	100	100	0
	70	100	100	100

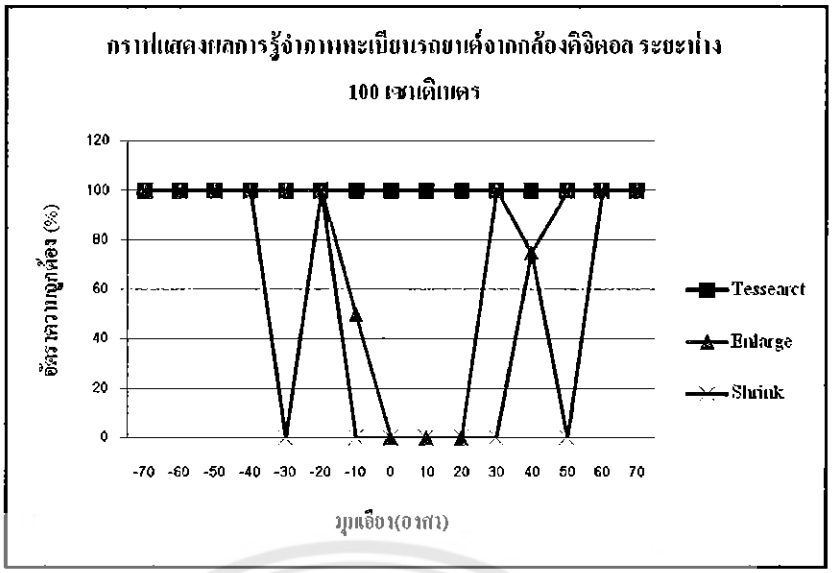


รูปที่ 5.22 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 75 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 75 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ใกล้เคียงกับโปรแกรม Tesseract OCR แต่เมื่อมุมในการถ่ายภาพอยู่ในช่วง 0 - 30 องศา พบว่าโปรแกรมให้ความถูกต้องในการรู้จำได้ไม่ดี

ตารางที่ 5.21 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 100 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องดิจิตอล		
		Tesseract	Enlarge	Shrink
100	-70	100	100	100
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	0
	-20	100	100	100
	-10	100	50	0
	0	100	0	0
	10	100	0	0
	20	100	0	0
	30	100	100	0
	40	100	75	75
	50	100	100	0
	60	100	100	100
	70	100	100	100

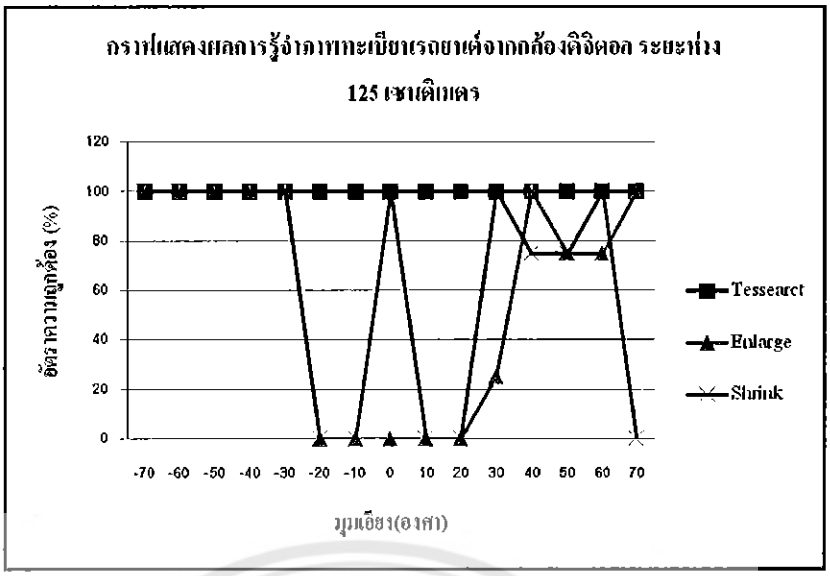


รูปที่ 5.23 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 100 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 100 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ใกล้เคียงกับโปรแกรม Tesseract OCR แต่เมื่อมุมในการถ่ายภาพอยู่ในช่วง 0 - 30 องศา พบว่าโปรแกรมให้ความถูกต้องในการรู้จำได้ไม่ดี

ตารางที่ 5.22 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 125 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องดิจิตอล		
		Tesseract	Enlarge	Shrink
125	-70	100	100	100
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	100
	-20	100	0	0
	-10	100	0	0
	0	100	0	100
	10	100	0	0
	20	100	0	0
	30	100	25	100
	40	100	100	75
	50	100	75	75
	60	100	75	100
	70	100	100	0

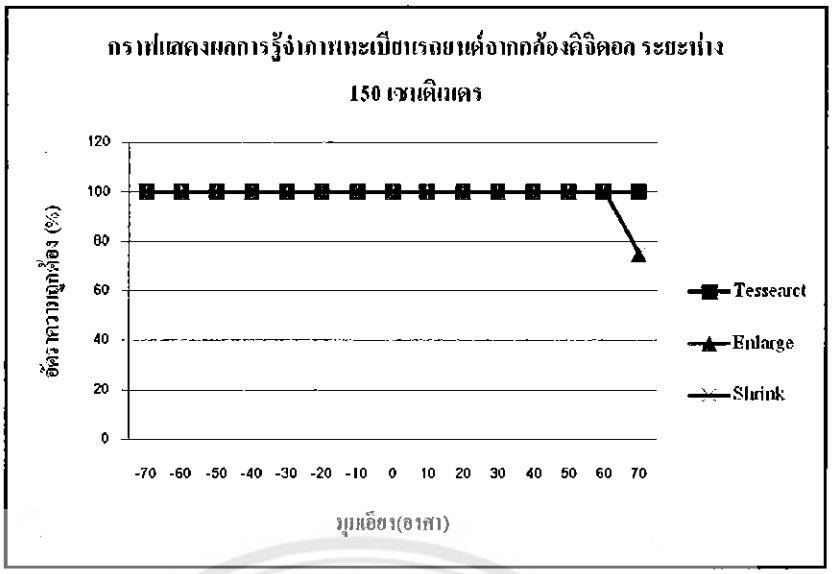


รูปที่ 5.24 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 125 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 125 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ใกล้เคียงกับโปรแกรม Tesseract OCR แต่เมื่อมุมในการถ่ายภาพอยู่ในช่วง 0 - 30 องศา พบว่าโปรแกรมให้ความถูกต้องในการรู้จำได้ไม่ดี

ตารางที่ 5.23 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 150 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องดิจิตอล		
		Tesseract	Enlarge	Shrink
150	-70	100	100	100
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	100
	-20	100	100	100
	-10	100	100	100
	0	100	100	100
	10	100	100	100
	20	100	100	100
	30	100	100	100
	40	100	100	100
	50	100	100	100
	60	100	100	100
	70	100	75	75

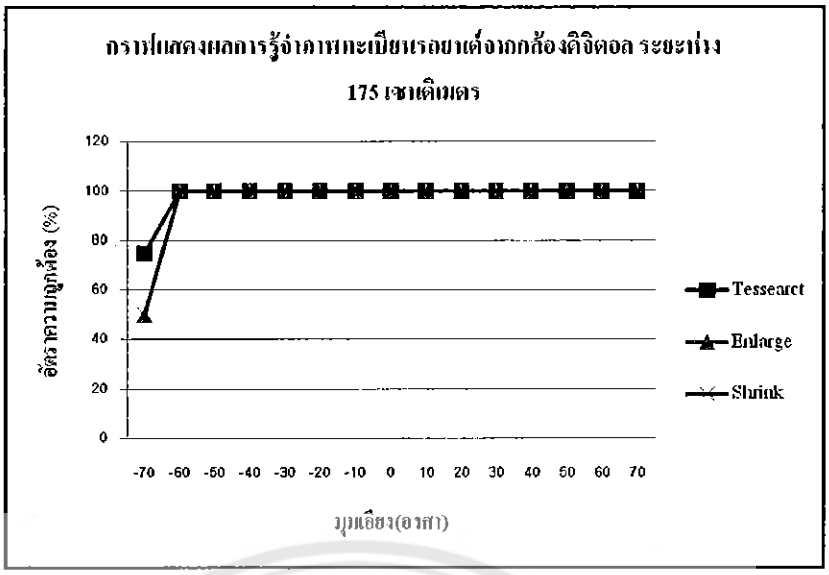


รูปที่ 5.25 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 150 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 150 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีเช่นเดียวกันกับโปรแกรม Tesseract OCR

ตารางที่ 5.24 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 175 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องดิจิตอล		
		Tesseract	Enlarge	Shrink
175	-70	75	50	50
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	100
	-20	100	100	100
	-10	100	100	100
	0	100	100	100
	10	100	100	100
	20	100	100	100
	30	100	100	100
	40	100	100	100
	50	100	100	100
	60	100	100	100
	70	100	100	100

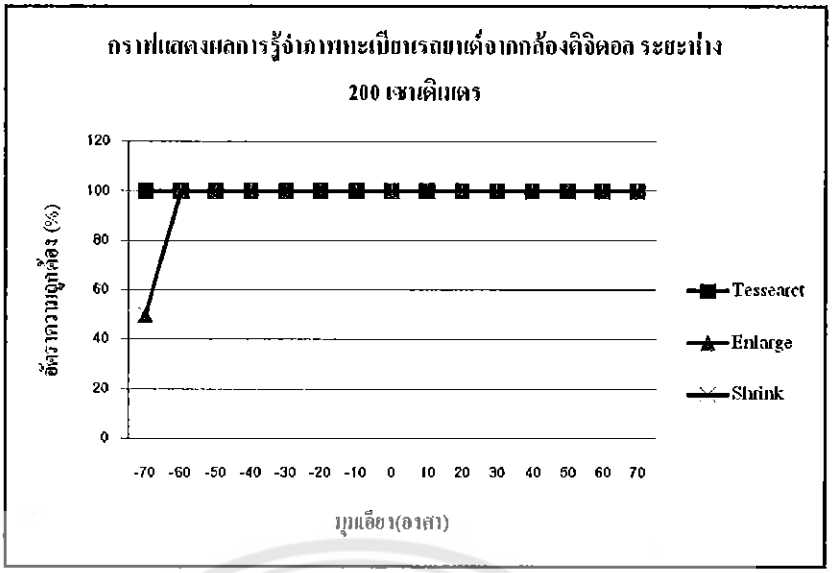


รูปที่ 5.26 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 175 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 175 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีเช่นเดียวกันกับโปรแกรม Tesseract OCR

ตารางที่ 5.25 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 200 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องดิจิตอล		
		Tesseract	Enlarge	Shrink
200	-70	100	50	50
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	100
	-20	100	100	100
	-10	100	100	100
	0	100	100	100
	10	100	100	100
	20	100	100	100
	30	100	100	100
	40	100	100	100
	50	100	100	100
	60	100	100	100
	70	100	100	100

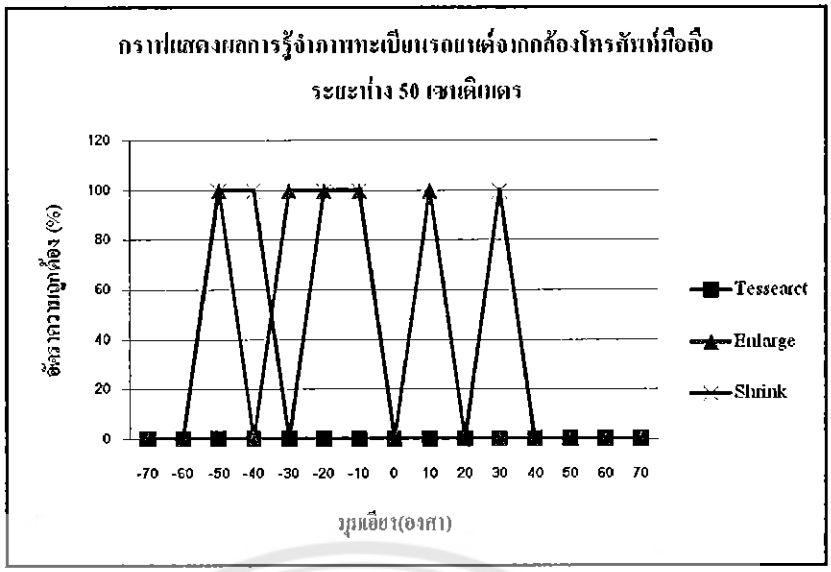


รูปที่ 5.27 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอลที่ระยะห่าง 200 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องดิจิตอล ที่ระยะห่าง 200 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีเช่นเดียวกันกับโปรแกรม Tesseract OCR

ตารางที่ 5.26 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 50 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องโทรศัพท์มือถือ		
		Tesseract	Enlarge	Shrink
50	-70	0	0	0
	-60	0	0	0
	-50	0	100	100
	-40	0	0	100
	-30	0	100	0
	-20	0	100	100
	-10	0	100	100
	0	0	0	0
	10	0	100	0
	20	0	0	0
	30	0	0	100
	40	0	0	0
	50	0	0	0
	60	0	0	0
	70	0	0	0

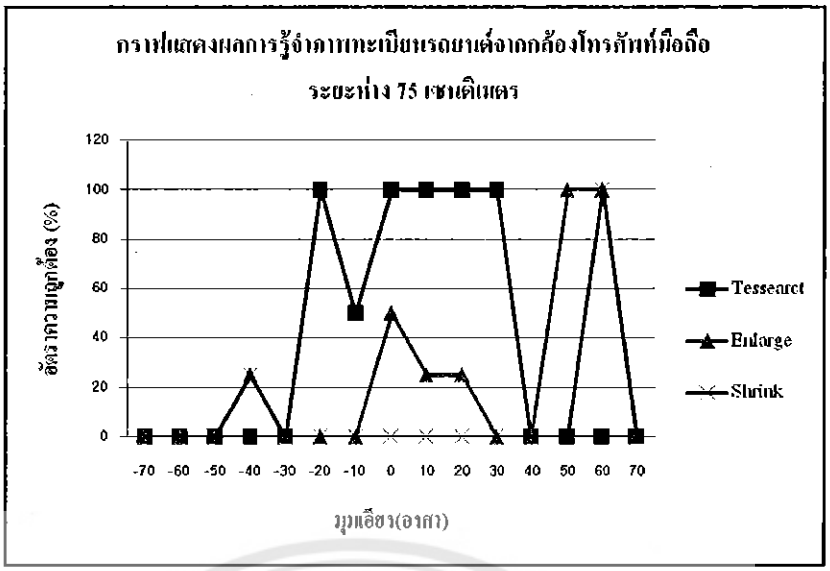


รูปที่ 5.28 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 50 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 50 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR

ตารางที่ 5.27 การรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 75 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องโทรศัพท์มือถือ		
		Tesseract	Enlarge	Shrink
75	-70	0	0	0
	-60	0	0	0
	-50	0	0	0
	-40	0	25	25
	-30	0	0	0
	-20	100	0	0
	-10	50	0	0
	0	100	50	0
	10	100	25	0
	20	100	25	0
	30	100	0	0
	40	0	0	0
	50	0	100	0
	60	0	100	100
	70	0	0	0

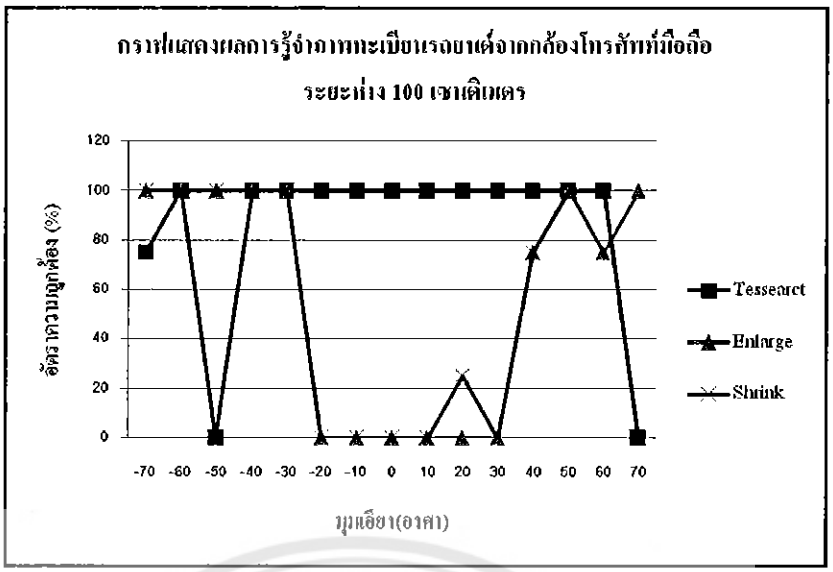


รูปที่ 5.29 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 75 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 75 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR เมื่อมุมในการถ่ายภาพอยู่ระหว่าง 30 - 60 องศา แต่โปรแกรมจะให้ผลในการรู้จำแย่กว่าโปรแกรม Tesseract OCR เมื่อมุมในการถ่ายภาพอยู่ระหว่าง 0 – 30 องศา

ตารางที่ 5.28 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 100 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องโทรศัพท์มือถือ		
		Tesseract	Enlarge	Shrink
100	-70	75	100	100
	-60	100	100	100
	-50	0	100	100
	-40	100	100	100
	-30	100	100	100
	-20	100	0	0
	-10	100	0	0
	0	100	0	0
	10	100	0	0
	20	100	0	25
	30	100	0	0
	40	100	75	75
	50	100	100	100
	60	100	75	100
	70	0	100	0

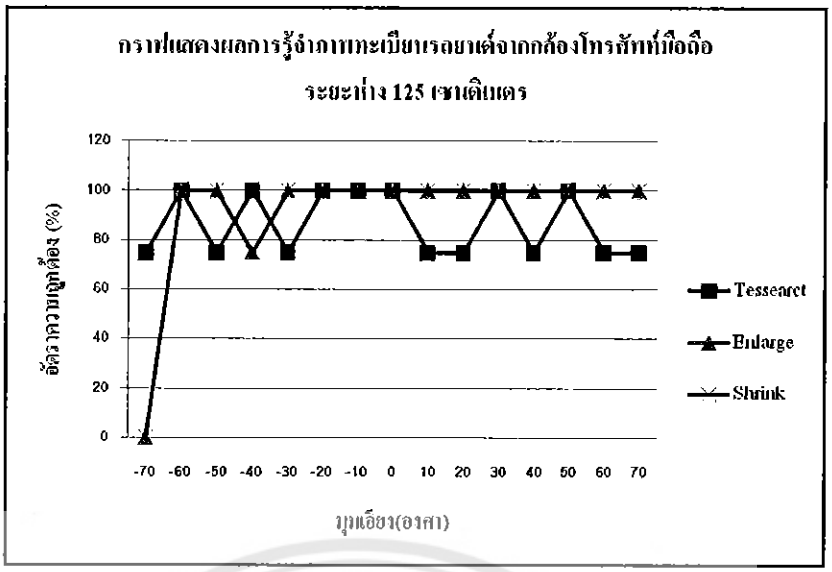


รูปที่ 5.30 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 100 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 100 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR เมื่อมุมในการถ่ายภาพอยู่ระหว่าง 30 - 60 องศา แต่โปรแกรมจะให้ผลในการรู้จำแย่กว่าโปรแกรม Tesseract OCR เมื่อมุมในการถ่ายภาพอยู่ระหว่าง 0 – 30 องศา

ตารางที่ 5.29 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 125 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องโทรศัพท์มือถือ		
		Tesseract	Enlarge	Shrink
125	-70	75	0	0
	-60	100	100	100
	-50	75	100	100
	-40	100	75	75
	-30	75	100	100
	-20	100	100	100
	-10	100	100	100
	0	100	100	100
	10	75	100	100
	20	75	100	100
	30	100	100	100
	40	75	100	100
	50	100	100	100
	60	75	100	100
	70	75	100	100

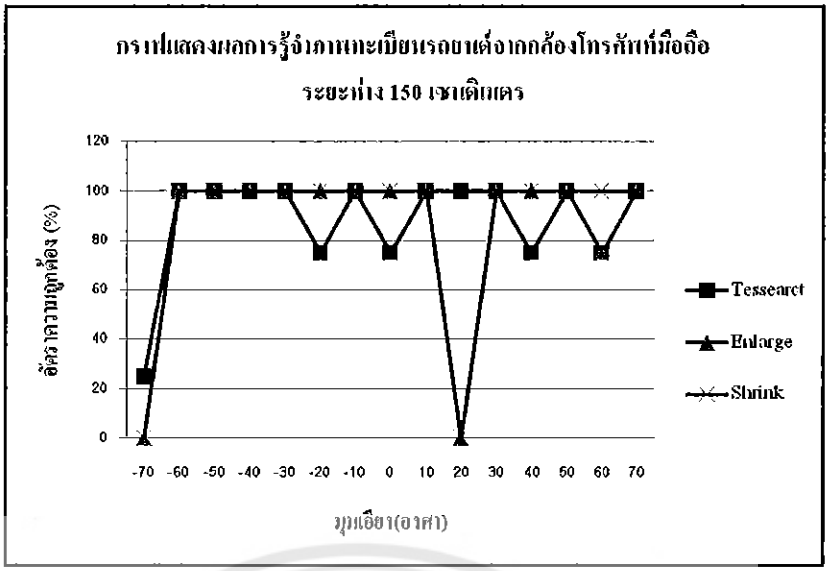


รูปที่ 5.31 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 125 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 125 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR

ตารางที่ 5.30 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 150 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องโทรศัพท์มือถือ		
		Tesseract	Enlarge	Shrink
150	-70	25	0	0
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	100
	-20	75	100	100
	-10	100	100	100
	0	75	100	100
	10	100	100	100
	20	100	0	0
	30	100	100	100
	40	75	100	100
	50	100	100	100
	60	75	75	100
	70	100	100	100

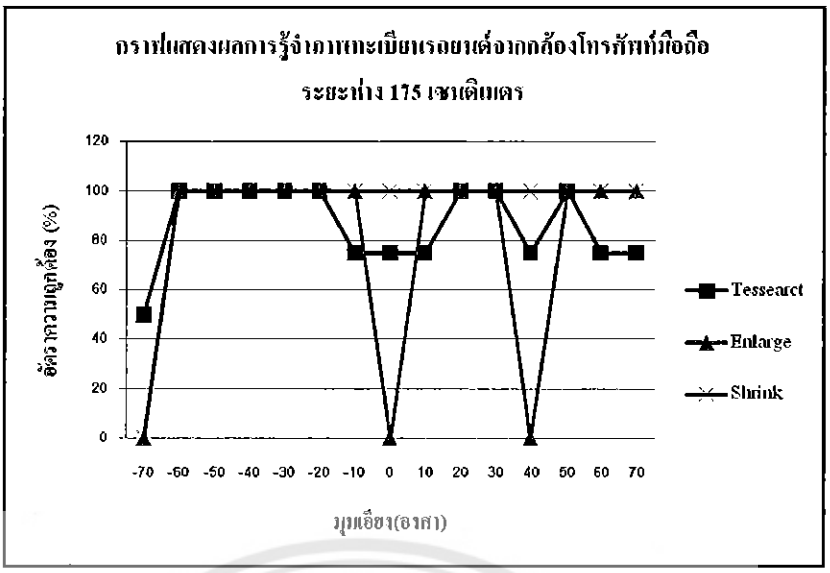


รูปที่ 5.32 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 150 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 150 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR

ตารางที่ 5.31 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 175 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องโทรศัพท์มือถือ		
		Tesseract	Enlarge	Shrink
175	-70	50	0	0
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	100
	-20	100	100	100
	-10	75	100	100
	0	75	0	100
	10	75	100	100
	20	100	100	100
	30	100	100	100
	40	75	0	100
	50	100	100	100
	60	75	100	100
	70	75	100	100

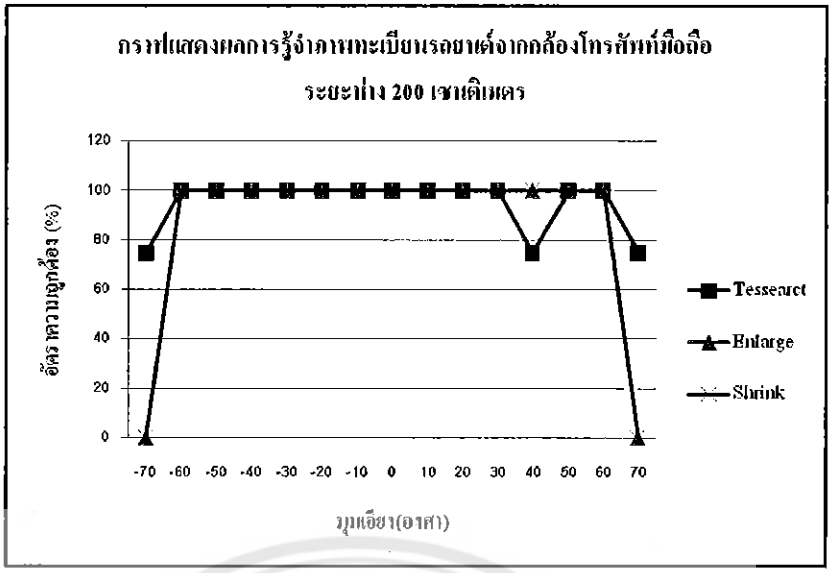


รูปที่ 5.33 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 175 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 175 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR อย่างชัดเจน

ตารางที่ 5.32 ผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 200 เซนติเมตร

ระยะทาง (เซนติเมตร)	มุมเอียง (องศา)	ภาพจากกล้องโทรศัพท์มือถือ		
		Tesseract	Enlarge	Shrink
200	-70	75	0	0
	-60	100	100	100
	-50	100	100	100
	-40	100	100	100
	-30	100	100	100
	-20	100	100	100
	-10	100	100	100
	0	100	100	100
	10	100	100	100
	20	100	100	100
	30	100	100	100
	40	75	100	100
	50	100	100	100
	60	100	100	100
	70	75	0	0



รูปที่ 5.34 กราฟผลการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 200 เซนติเมตร

จากตารางและกราฟแสดงผลการทดลองของการรู้จำภาพทะเบียนรถยนต์จากกล้องโทรศัพท์มือถือที่ระยะห่าง 200 เซนติเมตร โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR

สำหรับผลการทดสอบการรู้จำด้วยภาพทะเบียนรถยนต์ เพื่อความสะดวกในการวิเคราะห์ผลการทดลองจึงทำการสรุปข้อมูลออกเป็น 2 ส่วนดังนี้

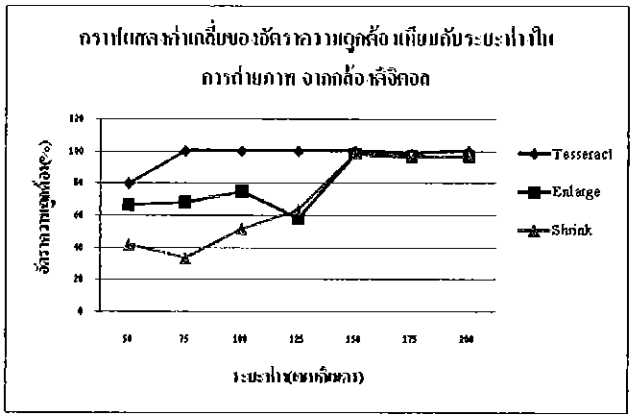
- ค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ
- ค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ

5.2.1 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ

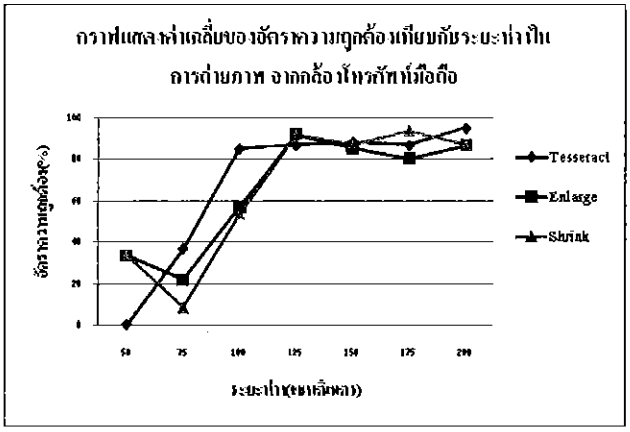
ระยะห่างในการถ่ายมีผลกระทบต่อภาพตัวอักษรหลายด้าน เช่น ทำให้ตัวอักษรมีขนาดเล็กลง, ความกว้างของตัวอักษรแต่ละตัวลดลง และทำให้รายละเอียดของภาพลดลง เป็นต้น จึงทำการสรุปผลได้ดังนี้

ตารางที่ 5.33 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับระยะห่างในการถ่ายภาพ

ระยะห่าง	ภาพจากกล้องดิจิตอล			ภาพจากกล้องโทรศัพท์มือถือ		
(เซนติเมตร)	Tesseract	Enlarge	Shrink	Tesseract	Enlarge	Shrink
50	80	66.66667	41.66667	0	33.33333	33.33333
75	100	68.33333	33.33333	36.66667	21.66667	8.333333
100	100	75	51.66667	85	56.66667	53.33333
125	100	58.33333	63.33333	86.66667	91.66667	91.66667
150	100	98.33333	98.33333	88.33333	85	86.66667
175	98.33333	96.66667	96.66667	86.66667	80	93.33333
200	100	96.66667	96.66667	95	86.66667	86.66667



รูปที่ 5.35 กราฟค่าเฉลี่ยความถูกต้องเทียบกับระยะห่างในการถ่ายภาพจากกล้องดิจิตอล



รูปที่ 5.36 กราฟค่าเฉลี่ยความถูกต้องเทียบกับระยะห่างในการถ่ายภาพจากกล้องโทรศัพท์มือถือ

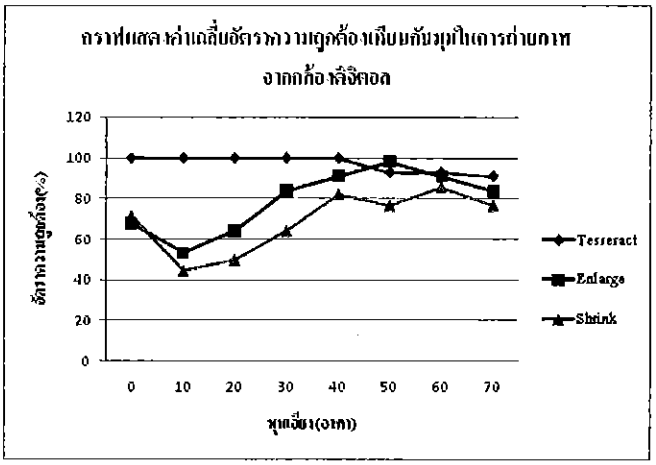
ผลที่ได้จากตารางและกราฟของค่าเฉลี่ยความถูกต้องในการรู้จำตัวอักษรเทียบกับระยะห่างในการถ่ายภาพ พบว่าโปรแกรมสามารทำการรู้จำได้ดีเมื่อระยะในการถ่ายภาพตั้งแต่ 125 เซนติเมตรเป็นต้นไป

5.2.2 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ

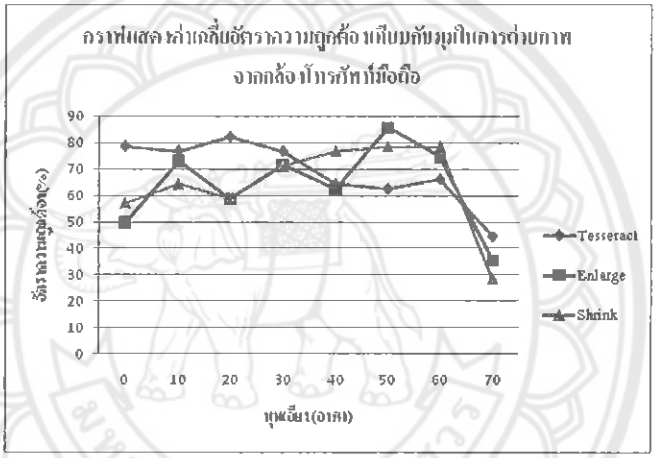
การถ่ายภาพทะเบียนรถยนต์แบบด้านข้างทำให้เกิดผลกระทบต่อรูปภาพตัวอักษรค่อนข้างมาก จึงทำการสรุปความสามารถในการรู้จำภาพทะเบียนรถยนต์ตามมุมที่เปลี่ยนไปได้ดังนี้

ตารางที่ 5.34 ค่าเฉลี่ยอัตราความถูกต้องเทียบกับมุมในการถ่ายภาพ

มุมเอียง	ภาพจากกล้องดิจิตอล			ภาพจากกล้องโทรศัพท์มือถือ		
(องศา)	Tesseract	Enlarge	Shrink	Tesseract	Enlarge	Shrink
0	100	67.85714	71.42857	78.57143	50	57.14286
10	100	53.57143	44.64286	76.78571	73.21429	64.28571
20	100	64.28571	50	82.14286	58.92857	58.92857
30	100	83.92857	64.28571	76.78571	71.42857	71.42857
40	100	91.07143	82.14286	64.28571	62.5	76.78571
50	92.85714	98.21429	76.78571	62.5	85.71429	78.57143
60	92.85714	91.07143	85.71429	66.07143	75	78.57143
70	91.07143	83.92857	76.78571	44.64286	35.71429	28.57143



รูปที่ 5.37 กราฟค่าเฉลี่ยความถูกต้องเทียบกับมุมในการถ่ายภาพจากกล้องดิจิตอล



รูปที่ 5.38 กราฟค่าเฉลี่ยความถูกต้องเทียบกับมุมในการถ่ายภาพจากกล้องโทรศัพท์มือถือ

ผลที่ได้จากตารางและกราฟของค่าเฉลี่ยความถูกต้องในการรู้จำตัวอักษรเทียบกับมุมในการถ่ายภาพ พบว่าโปรแกรมสามารถทำการรู้จำได้ดีเมื่อมุมในการถ่ายภาพอยู่ระหว่าง 30 - 60 องศา และสำหรับการรู้จำตัวอักษรด้วยภาพถ่ายจากกล้องโทรศัพท์มือถือนั้น โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR อย่างเห็นได้ชัด เมื่อมุมในการถ่ายภาพอยู่ระหว่าง 40 - 60 องศา

5.3 วิเคราะห์ผลการทดลอง

เพื่อสะดวกต่อการพิจารณาความสามารถของโปรแกรมในการรู้จำตัวอักษรตามจุดประสงค์สำหรับการทดสอบการรู้จำด้วยภาพถ่ายเอกสาร จึงทำการวิเคราะห์ผลการทดลองได้ดังต่อไปนี้

ความสามารถในการรู้จำตัวอักษรเทียบกับระยะห่างในการถ่ายภาพเอกสาร

1. โปรแกรมสามารถทำการรู้จำภาพตัวอักษรได้ดีเมื่อระยะในการถ่ายภาพไม่เกิน 30 เซนติเมตร

2. ความสามารถในการรู้จำจากระยะทางจะเพิ่มขึ้นเมื่อตัวอักษรมีขนาดใหญ่ขึ้น

ความสามารถในการรู้จำตัวอักษรเทียบกับมุมในการถ่ายภาพเอกสาร

1. โปรแกรมสามารถทำการรู้จำภาพตัวอักษรได้ดีเมื่อมุมในการถ่ายภาพไม่เกิน 60 องศา

2. โปรแกรมสามารถทำการรู้จำภาพตัวอักษรที่มุม 70 องศาได้ ถ้าตัวอักษรมีขนาดตั้งแต่ 24 เป็นต้นไป

ความสามารถในการรู้จำตัวอักษรเทียบกับขนาดของตัวอักษรในการถ่ายภาพเอกสาร

1. ความสามารถในการรู้จำตัวอักษรจะเพิ่มขึ้น ถ้าตัวอักษรมีขนาดใหญ่ขึ้น

ความสามารถในการรู้จำตัวอักษรระหว่างวิธีการย่อและการขยายบรรทัดตัวอักษร

1. การใช้วิธีการขยายบรรทัดตัวอักษรจะให้ค่าเฉลี่ยของความถูกต้องดีกว่าการใช้วิธีการย่อบรรทัดตัวอักษร

2. ถ้าภาพตัวอักษรที่ทำการรู้จำมีขนาดใหญ่เพียงพอ การใช้วิธีการย่อบรรทัดตัวอักษรจะให้ค่าความถูกต้องในการรู้จำที่ดีกว่า

3. การใช้วิธีการย่อบรรทัดตัวอักษรจะให้ความถูกต้องลดลงเมื่อภาพตัวอักษรมีขนาดเล็กลง

ความสามารถในการรู้จำตัวอักษรเทียบกับโปรแกรม Tesseract OCR

1. เมื่อมุมในการถ่ายภาพและขนาดของตัวอักษรเพิ่มขึ้น การใช้วิธีการย่อและการขยายบรรทัดจะให้ผลค่าเฉลี่ยความถูกต้องดีกว่าโปรแกรม Tesseract OCR

2. โปรแกรมสามารถทำการรู้จำตัวอักษรได้ถูกต้องกว่าเมื่อตัวอักษรมีขนาดใหญ่ขึ้น อันเนื่องมาจากโปรแกรม Tesseract OCR จะมีความสามารถในการรู้จำตัวอักษรลดลง เพราะมีโอกาสที่จะเกิดปัญหาในด้านการซ้อนทับกันของเส้นระดับบรรทัดในแนวตรงมากขึ้น

สำหรับในส่วนของการทดสอบโปรแกรมโดยใช้การรู้จำด้วยภาพทะเบียนรถยนต์ เมื่อทำการวิเคราะห์ผลการทดลองตามจุดประสงค์ของการทดสอบแล้ว ได้ผลดังนี้

ความสามารถในการรู้จำตัวอักษรเทียบกับระยะห่างในการถ่ายภาพเอกสาร

1. โปรแกรมสามารถทำการรู้จำได้ดีเมื่อระยะในการถ่ายภาพตั้งแต่ 125 เซนติเมตร เป็นต้นไป

ความสามารถในการรู้จำตัวอักษรเทียบกับมุมในการถ่ายภาพเอกสาร

1. โปรแกรมสามารถทำการรู้จำได้ดีเมื่อมุมในการถ่ายภาพไม่เกิน 60 องศา

ความสามารถในการรู้จำตัวอักษรระหว่างวิธีการย่อและการขยายบรรทัดตัวอักษร

1. การใช้วิธีการขยายบรรทัดตัวอักษรจะให้ค่าเฉลี่ยความถูกต้องในการรู้จำสูงกว่าการย่อบรรทัดตัวอักษร

2. สำหรับภาพถ่ายจากกล้องโทรศัพท์มือถือ การใช้วิธีการย่อบรรทัดตัวอักษรจะมีค่าเฉลี่ยความถูกต้องในการรู้จำสูงกว่าการขยายบรรทัดตัวอักษร

ความสามารถในการรู้จำระหว่างภาพจากกล้องดิจิตอลและภาพจากกล้องโทรศัพท์มือถือ

1. การรู้จำตัวอักษรจากภาพที่ถ่ายด้วยกล้องดิจิตอลจะมีค่าความถูกต้องโดยเฉลี่ยสูงกว่าการรู้จำตัวอักษรจากภาพที่ถ่ายด้วยกล้องโทรศัพท์มือถือ

ความสามารถในการรู้จำตัวอักษรเทียบกับโปรแกรม Tesseract OCR

1. โปรแกรมสามารถทำการรู้จำได้ดีกว่าโปรแกรม Tesseract OCR เมื่อมุมในการถ่ายภาพอยู่ระหว่าง 30 - 60 องศา และระยะห่างในการถ่ายภาพตั้งแต่ 125 เซนติเมตรเป็นต้นไป

บทที่ 6
บทสรุป

โครงงานนี้ได้พัฒนาโปรแกรมเพื่อทำการรู้จำตัวอักษรจากภาพที่ระดับความลึกของตัวอักษรในภาพแตกต่างกันจากการถ่ายภาพแบบด้านข้าง โดยใช้ทฤษฎีการประมวลผลภาพดิจิตอลในการปรับปรุงคุณสมบัติต่างๆของรูปและเพื่อเพิ่มความถูกต้องให้กับการรู้จำ ซึ่งโปรแกรมจะใช้การปรับขนาดของบรรทัดตัวอักษรในแต่ละบรรทัดและทำการปรับบรรทัดตัวอักษรให้กลับมาสู่ในแนวระดับ แล้วจึงทำการรู้จำบรรทัดตัวอักษรนั้น

ในการพัฒนาซอฟต์แวร์นั้น ผู้พัฒนาได้ใช้ภาษา C++ ร่วมกับ OpenCV ซึ่งเป็นไลบรารีสำหรับการประมวลผลภาพดิจิตอล เพื่อความสะดวกในการพัฒนาส่วนอื่นๆ แล้วจึงใช้โปรแกรม Tesseract OCR ในการรู้จำภาพตัวอักษร

6.1 หน้าที่การทำงานของโปรแกรม

สำหรับโปรแกรมที่พัฒนาในโครงงานมีหน้าที่ในการทำงานต่างๆ ดังนี้

1. โปรแกรมจะทำการค้นหาบรรทัดตัวอักษรในภาพ เพื่อกำหนดขอบเขตของในแต่ละบรรทัดตัวอักษร

2. เมื่อได้บรรทัดตัวอักษรแล้ว โปรแกรมจะทำการปรับขนาดของบรรทัดตัวอักษรให้เท่ากัน โดยใช้วิธีการย่อบรรทัดตัวอักษรและวิธีการขยายบรรทัดตัวอักษร

3. เมื่อได้ภาพที่ทำการปรับขนาดของบรรทัดตัวอักษรแล้ว โปรแกรมจะนำภาพที่ได้มาทำการรู้จำตัวอักษรด้วยโปรแกรม Tesseract OCR

6.2 วิเคราะห์ผลการทดลอง

จากผลการทดลองในบทที่ 5 สามารถทำการวิเคราะห์ผลการทดลองโดยสรุปได้ดังนี้

1. โปรแกรมสามารถทำการรู้จำตัวอักษรได้ดีเมื่อมุมในการถ่ายภาพไม่เกิน 60 องศา

2. โปรแกรมสามารถทำการรู้จำภาพตัวอักษรที่มุม 70 องศาได้ ถ้าตัวอักษรมีขนาดตั้งแต่ 24 เป็นต้นไป

3. การใช้วิธีการขยายบรรทัดตัวอักษรจะให้ค่าเฉลี่ยความถูกต้องในการรู้จำตัวอักษรสูงกว่าการใช้วิธีการย่อบรรทัดตัวอักษร

4. ถ้าตัวอักษรในภาพถ่ายมีขนาดค่อนข้างใหญ่ การใช้วิธีการย่อบรรทัดตัวอักษรจะให้ค่าเฉลี่ยของความถูกต้องในการรู้จำตัวอักษรสูงกว่าการใช้วิธีการขยายบรรทัดตัวอักษร

5. เมื่อตัวอักษรมีขนาดใหญ่ขึ้น โปรแกรมสามารถทำการรู้จำได้ถูกต้องกว่าโปรแกรม Tesseract OCR อันเนื่องมาจากมีโอกาศที่จะเกิดการซ้อนทับกันของบรรทัดตัวอักษรในแนวตรงมากขึ้น

6.3 ปัญหาและแนวทางการแก้ไข

ในระหว่างการทำโครงงานนี้ได้ประสบปัญหาต่างๆ ดังนี้

1. เนื่องจากโปรแกรม Tesseract OCR มีความสามารถในการรับรูปแบบของภาพที่ทำการรู้จำตัวอักษรได้น้อย เช่น .bmp (Bitmap) และ .tiff (Tagged Image File Format) เป็นต้น จึงต้องทำการแปลงภาพที่จะทำการรู้จำให้อยู่ในรูปแบบที่โปรแกรม Tesseract OCR สามารถอ่านได้

2. ภาพที่ทำการรู้จำในบางครั้งเกิดการเบลอจากการถ่ายภาพหรือสิ่งรบกวนต่างๆ ทำให้การค้นหาบรรทัดตัวอักษรไม่สามารถค้นหาได้ผลลัพธ์ที่ดีเท่าที่ควร โดยแนวทางแก้ไขปัญหานี้คือก่อนขั้นตอนการค้นหาบรรทัดตัวอักษรควรมีการปรับปรุงรูปภาพ เพื่อกำจัดผลที่เกิดจากการเบลอหรือสิ่งรบกวนออกให้มากที่สุด

3. สำหรับกรณีที่ขนาดตัวอักษรในภาพมีขนาดเล็กทำให้ผลการรู้จำตัวอักษรมีประสิทธิภาพต่ำลงอย่างมาก อาจจะแก้ไขด้วยการขยายภาพที่จะทำการรู้จำก่อนเพื่อขยายขนาดของตัวอักษรแล้วจึงนำมาทำการรู้จำ

4. ในบางกรณีภาพที่ทำการรู้จำมีลักษณะของภาพตัวอักษรที่ชัดเจน แต่การรู้จำให้ผลผิดพลาดเป็นจำนวนมากเนื่องจากโปรแกรม Tesseract OCR ดังรูปที่ 6.1 สำหรับวิธีการแก้ไขปัญหานี้คือ อาจจะทำการแก้ไขฟังค์ชั่นการทำงานของ Tesseract OCR ในบางส่วนที่เกี่ยวข้องเพื่อให้ได้ผลที่ดีขึ้นหรือทำการเปลี่ยนโปรแกรมสำหรับการรู้จำตัวอักษรที่มีประสิทธิภาพสูงขึ้น

รูปที่ 6.1 ผลที่ได้จากการรู้จำด้วยโปรแกรม Tesseract OCR

5. สำหรับผลการรู้จำภาพทะเบียนรถยนต์ในบางส่วนที่มีผลที่ไม่ดีนั้น ทางผู้พัฒนาได้พยายามหาสาเหตุของปัญหาโดยการลดปัญหาเกี่ยวกับสัญญาณรบกวนออกไปจนหมดแล้ว แต่ผลที่ได้ยังไม่ดีขึ้นในบางส่วน เกิดจากโปรแกรม Tesseract OCR ไม่สามารถทำการรู้จำภาพเหล่านั้นได้ สำหรับการแก้ไขอาจจะทำการเปลี่ยนโปรแกรมสำหรับการรู้จำตัวอักษร

6. เนื่องจากการทดสอบการรู้จำด้วยภาพทะเบียนรถยนต์นั้น ภาพที่ใช้ในการทดสอบมีสิ่งรบกวนค่อนข้างมากซึ่งการใช้วิธีการลดสิ่งรบกวนโดยพิจารณาจากจุดรอบข้างนั้นยังไม่เพียงพอ เพราะค่าความถูกต้องโดยเฉลี่ยที่ได้จากการรู้จำโดยวิธีดังกล่าวอยู่ที่ร้อยละ 63.8 แต่เมื่อเปลี่ยนวิธีการลดสิ่งรบกวนเป็นการลดสิ่งรบกวนโดยใช้ค่ากลางของข้อมูล (Median Filter) ทำให้ได้ค่าความถูกต้องของการรู้จำอยู่ที่ร้อยละ 74.4 ซึ่งจะเห็นได้ว่าถ้าภาพที่นำมาทำการรู้จำมีคุณภาพดี โปรแกรมก็จะสามารถทำการรู้จำได้อย่างมีประสิทธิภาพตามไปด้วย

6.4 สรุปผลการทดลอง

จากผลการทดลองโปรแกรมสามารถทำการรู้จำตัวอักษรที่เกิดจากการถ่ายภาพแบบด้านข้างซึ่งใช้การประมวลผลภาพดิจิตอลในการแก้ปัญหาได้เป็นอย่างดี โดยในการทดสอบด้วยการรู้จำภาพตัวอักษรนั้น โปรแกรมสามารถทำการรู้จำได้ดีเมื่อมุมในการถ่ายภาพไม่เกิน 60 องศา แต่สำหรับการรู้จำด้วยภาพทะเบียนรถยนต์โปรแกรมโปรแกรมสามารถทำการรู้จำได้ดีเมื่อมุมในการถ่ายภาพอยู่ระหว่าง 30 - 60 องศา และระยะห่างในการถ่ายภาพตั้งแต่ 125 เซนติเมตร เป็นต้นไป

6.5 ข้อเสนอแนะ

1. ในขั้นตอนการย่อหรือการขยายบรรทัดตัวอักษรนั้น อาจจะทำการย่อหรือขยายบรรทัดตัวอักษรตามสัดส่วนที่เหมาะสม เพื่อปรับให้ภาพบรรทัดตัวอักษรดีขึ้น

2. อาจจะทำการพัฒนาความสามารถของอัลกอริทึมในการค้นหาบรรทัดของตัวอักษรให้ดีขึ้นกว่าเดิม

3. สามารถเลือกหรือเปลี่ยนโปรแกรม Tesseract OCR ที่นำมาใช้ในการรู้จำตัวอักษรเป็นโปรแกรมอื่นได้โดยไม่กระทบการทำงานทั้งหมดของโปรแกรม

เอกสารอ้างอิง

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

[2] Rafael C. Gonzalez and Richard E. Woods, **Digital Image Processing**, Prentice Hall, 2002

[3] Stephen R. Schach, **Object-Oriented & Classical Software Engineering**, 7th Edition, McGraw-Hill, 2007

[4] **"Linear regression"** [Online]. Available: http://en.wikipedia.org/wiki/Linear_regression

[5] **"Google Code: Tesseract-OCR"** [Online]. Available: http://code.google.com/p/tesseract-ocr/

[6] **"OpenCVWiki"** [Online]. Available: http://opencv.willowgarage.com/wiki/

[7] **"Flood fill"** [Online]. Available: http://en.wikipedia.org/wiki/Flood_fill

ภาคผนวก ก.

OpenCV Library

โครงสร้างข้อมูลใน OpenCV

1. IplImage

เป็นโครงสร้างสำหรับการเก็บข้อมูลของภาพและข้อมูลพิกเซลล์ (Pixel) ของภาพ มีโครงสร้างดังนี้

```
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

จำนวนชั้นสีในรูปภาพ มีได้ตั้งแต่ 1 – 4 ชั้น

depth

ความลึกบิตของแต่ละพิกเซลล์ ประกอบด้วย

- IPL_DEPTH_8U แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 8 บิต ไม่คิดเครื่องหมาย
- IPL_DEPTH_8S แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 8 บิต คิดเครื่องหมาย
- IPL_DEPTH_16U แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 16 บิต ไม่คิดเครื่องหมาย
- IPL_DEPTH_16S แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 16 บิต คิดเครื่องหมาย
- IPL_DEPTH_32S แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 32 บิต คิดเครื่องหมาย
- IPL_DEPTH_32F แต่ละพิกเซลล์เป็นจำนวนทศนิยมขนาด 32 บิต
- IPL_DEPTH_64F แต่ละพิกเซลล์เป็นจำนวนทศนิยมขนาด 64 บิต

Origin

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

width

ความกว้างของภาพในหน่วยพิกเซลล์

height

ความสูงของรูปภาพในหน่วยพิกเซลล์

ImageSize

ขนาดของรูปภาพได้จาก width x height

imageData

เป็นตัวแปลที่เก็บค่าตำแหน่งเริ่มต้นในหน่วยความจำที่เก็บข้อมูลของภาพ

widthStep

ขนาดของข้อมูลแต่ละแถวในรูปภาพในหน่วยไบท์ (Byte)

ฟังค์ชั่นพื้นฐานใน OpenCV

1. cvCreateImage

ทำหน้าที่จองหน่วยความจำให้กับรูปภาพตามขนาดหรือข้อมูลต่างๆที่ต้องการ มีโครงสร้างดังนี้

```
IplImage *cvCreateImage(CvSize size, int depth, int channels);
```

size

ขนาดของรูปภาพที่ต้องการในหน่วยพิกเซลล์

depth

ความลึกบิตของแต่ละพิกเซลล์

- IPL_DEPTH_8U แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 8 บิต ไม่คิดเครื่องหมาย
- IPL_DEPTH_8S แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 8 บิต คิดเครื่องหมาย
- IPL_DEPTH_16U แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 16 บิต ไม่คิดเครื่องหมาย
- IPL_DEPTH_16S แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 16 บิต คิดเครื่องหมาย
- IPL_DEPTH_32S แต่ละพิกเซลล์เป็นจำนวนเต็มขนาด 32 บิต คิดเครื่องหมาย
- IPL_DEPTH_32F แต่ละพิกเซลล์เป็นจำนวนทศนิยมขนาด 32 บิต
- IPL_DEPTH_64F แต่ละพิกเซลล์เป็นจำนวนทศนิยมขนาด 64 บิต

channel

จำนวนชั้นสีของรูปภาพ

2. cvLoadImage

ทำหน้าที่ดึงข้อมูลรูปภาพจากอุปกรณ์เก็บข้อมูล (Storage) มีโครงสร้างดังนี้

cvLoadImage(string path);

path

ตำแหน่งของรูปภาพในอุปกรณ์เก็บข้อมูล

3. cvReleaseImage

ทำหน้าที่คืนค่าหน่วยความจำที่เก็บข้อมูลภาพสู่ระบบ มีโครงสร้างดังนี้

cvReleaseImage(IplImage **image);

image

ภาพที่ต้องการคืนหน่วยความจำ

4. cvCvtColor

ทำหน้าที่แปลงรูปภาพจากโหมดหนึ่งไปสู่โหมดอื่นๆ มีโครงสร้างดังนี้

void cvCvtColor(const CvArr* src, CvArr* dst, int code);

src

รูปภาพที่จะทำการแปลง โดยต้องมีความลึกบิตของชั้นสีขนาด 8 บิต (8U), 16 บิต (16U) และจำนวนทศนิยม 32 บิต (32F)

dst

รูปภาพที่ได้จากการแปลง ต้องมีชนิดของขนาดเหมือน src แต่จำนวนของแชลแนล (Channel) อาจเปลี่ยนแปลง

Code

รูปแบบคำสั่งในการแปลงโหมดภาพนั้น จะใช้ CV_<โหมดเริ่มต้น>2<โหมดที่ต้องการ> เช่น CV_RGB2GRAY คือการเปลี่ยนจากภาพแบบ RGB เป็นภาพแบบ Gray-Scale

โดยสำหรับภาพแบบ RGB นั้นใช้ 3 แชลแนลในการเก็บข้อมูลรูปแต่ละพิกเซลล์ ดังนี้

R คือ ค่าของสีแดงในพิกเซลล์

G คือ ค่าของสีเขียวในพิกเซลล์

B คือ ค่าของสีน้ำเงินในพิกเซลล์

และเนื่องจากภาพแบบ Gray-Scale นั้นใช้เพียง 1 แชลแนลในการเก็บข้อมูลรูปแต่ละพิกเซลล์ จึงต้องมีการคำนวณเพื่อเปลี่ยนค่าต่างๆดังนี้

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

เมื่อ Y คือ ค่าของในพิกเซลล์ใน Gray-Scale

5. cvAdaptiveThreshold

Adaptive Threshold เป็นฟังค์ชั่นที่ทำการเปลี่ยนภาพที่ต้องการให้อยู่ในลักษณะแบบ Binary มีโครงสร้างดังนี้

```
void cvAdaptiveThreshold(const CvArr* src, CvArr* dst, double max_value,
                         int adaptive_method, int threshold_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 จะให้ค่าของพิกเซลล์เป็นสีขาวถ้าพิกเซลล์นั้นมีค่ามากกว่า Threshold ถ้าน้อยกว่าหรือเท่ากับค่าของพิกเซลล์นั้นจะเป็นสีดำ

- CV_THRESH_BINARY_INV จะให้ค่าของพิกเซลล์เป็นสีดำถ้าพิกเซลล์นั้นมีค่ามากกว่า Threshold ถ้าน้อยกว่าหรือเท่ากับค่าของพิกเซลล์นั้นจะเป็นสีขาว

block_size

ขนาดของพิกเซลล์รอบๆที่ใช้ในการคำนวณ โดยมีค่าเริ่มตั้งแต่ 3,5,7,...

param1

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

ภาพที่ต้องการปรับปรุง

dst

ภาพที่ได้หลังจากการปรับปรุง

smoothtype

วิธีในการปรับปรุง โดยสำหรับวิธีการปรับปรุงโดยใช้ค่ากลางของข้อมูล (Median Filter) ค่าของ smoothtype จะเท่ากับ CV_MEDIAN โดยขนาดของบล็อคจะเท่ากับค่าของ param1

param1, param2, param3

ตัวแปลที่ใช้ในการคำนวณวิธีต่างๆ

ภาคผนวก ข.

Tesseract OCR

Tesseract OCR เป็นโปรแกรมสำหรับทำการรู้จำตัวอักษรแบบ OpenSource ที่พัฒนาโดย HP LAB ในปี 1985 – 1995 แต่ปัจจุบันถูกพัฒนาต่อโดยบริษัท Google โดยสามารถทำงานได้บนระบบปฏิบัติการดังนี้

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

โดยในเวอร์ชั่นปัจจุบันสามารถรองรับการรู้จำตัวอักษรได้ถึง 6 ภาษาคือ ภาษาอังกฤษ, ภาษาฝรั่งเศษ, ภาษาอิตาเลียน, ภาษาเยอรมัน, ภาษาสเปน และ ภาษาเนเธอร์แลนด์ และยังสามารถฝึกสอนให้ทำการรู้จำภาษาอื่นๆได้อีกด้วย

คำสั่งสำหรับทำการรู้จำคือ

tesseract.exe <ไฟล์รูปภาพ> <ไฟล์เอาต์พุตที่ได้จากการรู้จำ> [–l <ภาษาที่ต้องการรู้จำ>]

เช่น

tesseract.exe Test.BMP TestOCT –l eng

ขั้นตอนในการรู้จำ

1. โปรแกรมจะทำการรับรูปภาพเข้ามา แล้วทำการแปลงไฟล์รูปภาพให้อยู่ในรูปตัวแปล page_image แบบ Global

2. โปรแกรมจะทำการรู้จำโดยมีขั้นตอนดังนี้

   2.1 ทำการแปลงภาพที่ได้เป็นภาพแบบ Binary

   2.2 ทำการหาเส้นบรรทัดแล้วนำข้อมูลในบรรทัดที่ได้ใส่ในตัวแปลแบบ Block

   2.3 ทำการรู้จำภาพครั้งละ 1 Block แล้วจึงนำข้อมูลที่ได้เขียนลงในไฟล์

ประวัติผู้เขียนโครงงาน

ชื่อ นายสิระ เสาวลักษณ์

ภูมิลำเนา 77/140 ถนนศรีธรรมไตรปิฎก ตำบลในเมือง

อำเภอเมือง จังหวัดพิษณุโลก 65000

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนจ่านกร้อง
- ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail **mrdotm@gmail.com**