โปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์

Car Registration Detect Program

นาย ภาณุเดช วศินวรรธนะ รหัส 47360151

นาย วัชรพงษ์ เจนพนัส รหัส 47360169

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..........
เลขทะเบียน 05100043
เลขเรียกหนังสือ..........
มหาวิทยาลัยนเรศวร

15093351 e.2

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน โปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์

ผู้ดำเนินโครงงาน นายภาณุเดช วศินวรรธนะ รหัส 47360151

นายวัชรพงษ์ เจนพนัส รหัส 47360169

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2550

....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติโครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

.... กรรมการ
(ดร.ไพศาล มุณีสว่าง)

.... กรรมการ
(อาจารย์ศิริพร เดชะศิลารักษ์)

หัวข้อโครงงาน	โปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์
ผู้ดำเนินโครงงาน	นายภาณุเดช วศินวรรธนะ รหัส 47360151
	นายวัชรพงษ์ เจนพนัส รหัส 47360169
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

....

บทคัดย่อ

โดยทั่วไปสังคมขนาดเล็กหรือใหญ่ เมื่อประสบปัญหาการลักขโมย ทั้งรถยนต์ และรถจักรยานยนต์ การติดตามรถที่สูญหายมีความล่าช้า คณะผู้จัดทำจึงมีแนวคิดในการพัฒนาโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์ เพื่อสืบค้นรถที่ต้องสงสัย

โปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์ที่พัฒนาขึ้น สามารถประมวลผลสัญญาณภาพของป้ายทะเบียนรถยนต์ และระบุหมายเลขทะเบียนรถยนต์ได้อัตโนมัติ โดยมีหลักการทำงานดังนี้ เริ่มต้นทำการแยกวัตถุกับพื้นหลังของภาพออกจากกันด้วยวิธีการเทรชโฮล เพื่อหาตำแหน่งของวัตถุที่เป็นป้ายทะเบียนรถ ต่อมาทำการแยกวัตถุภายในภาพออกจากกัน โดยใช้วิธีรีเจียน ลาเบลลิง จากนั้นพิจารณาบล็อกของวัตถุใดที่ควรจะเป็นป้ายทะเบียน โดยพิจารณาตามเงื่อนไขที่กำหนดให้ หลังจากนั้น ทำการแยกหมายเลขทะเบียนออกจากป้ายทะเบียนรถ ข้อมูลที่ได้รับจะเป็นบล็อก อันนำไปเข้าสู่กระบวนการเปรียบเทียบค่าความผิดพลาดกับข้อมูลต้นแบบ เพื่อระบุความถูกต้องของตัวอักษร และตัวเลขที่ได้รับ สุดท้ายนำหมายเลขทะเบียนรถยนต์ที่ได้เก็บลงในฐานข้อมูล

ผลการทำงานของโปรแกรมที่พัฒนาสามารถหาหมายเลขทะเบียนรถด้วยวิธีการรู้จำได้อย่างถูกต้อง และเวลาที่ใช้ในการประมวลผลจะขึ้นอยู่กับความสมบูรณ์ของภาพ แต่อย่างไรก็ตามโปรแกรมนี้สามารถนำไปใช้งานในการตรวจจับหมายเลขทะเบียนรถได้เสมอ นอกจากนี้โปรแกรมนี้ยังสามารถประยุกต์ใช้ในการค้นหารถที่ต้องสงสัยตามแยกถนน และสถานที่จอดรถต่างๆ เพื่อลดภาระงานให้แก่เจ้าหน้าที่ที่มีส่วนเกี่ยวข้อง

Project Title	Car Registration Detect Program.
Name	Mr. Panudet Vasinwattana ID. 47360151
	Mr. Watcharapong Jenphanus ID. 47360169
Project Advisor	Assistant Professor Suchart Yammen, Ph.D.
Major	Computer Engineering.
Department	Electrical and Computer Engineering.
Academic year	2007

Abstract

Generally both of small and big social face a problem of car and motorcycle lost. Following them that are lost has the lateness. Organizer faculty have the idea to development of the detect car registration number program for investigate the suspicious car.

Car Registration Detect program that was developed to able analyses a car image and specify car registration number has automatic, the principle works as follows. First of all, use color information to separate background and object by Thesholding method for finding position of registration plate. Second, separate each objects in an image by Region labeling method. Next, consider blocks of object that is registration plate with conditions. After that, each registration numbers were separated from registration plate. The data will that receive to is block. Brings to finding error value with original characters in the comparing process for specify character and number that receive. Finally, keep registration number to the database.

The work of a program that develops can seek registration number with recognize characters method too correctly. Time in use to processing car image will depend on the completeness of an image. However, you always receive a registration number correctly when use this program. In addition, this program can apply to decrease work responsibility for an officer who has to relate on intersection or parking.

กิตติกรรมประกาศ

โครงงานวิศวกรรมฉบับนี้ สำเร็จได้ด้วยดีก็เนื่องจากความอนุเคราะห์ของอาจารย์ที่ปรึกษาโครงงานคือ ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น รวมถึงอาจารย์ทุกท่านที่กรุณาสละเวลาอันมีค่าเพื่อชี้แนะแนวทาง ให้คำปรึกษา และให้ความช่วยเหลือ ในโอกาสนี้คณะผู้จัดทำโครงงาน จึงขอขอบพระคุณทุกๆท่าน ที่มีส่วนช่วยให้โครงงานนี้ประสบความสำเร็จได้ด้วยดี สุดท้ายขอขอบพระคุณบิดา และมารดาของคณะผู้จัดทำที่เป็นผู้สนับสนุนในทุกๆด้านโดยเฉพาะด้านการศึกษาตลอดมา คณะผู้จัดทำขอนอบระลึกในพระคุณ และขอกราบขอบพระคุณมา ณ ที่นี้

นายภาณุเดช วศินวรรธนะ

นายวัชรพงษ์ เจนพนัส

สารบัญ

หน้า

บทคัดย่อภาษาไทย .... ก
บทคัดย่อภาษาอังกฤษ .... ข
กิตติกรรมประกาศ .... ค
สารบัญ .... ง
สารบัญรูป .... ช

บทที่ 1 บทนำ
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 1
1.4 แผนการดำเนินการ 1
1.5 ระยะเวลาในการดำเนินการ 2
1.6 ผลที่คาดว่าจะได้รับ 2
1.7 รายละเอียดงบประมาณโครงงาน 3

บทที่ 2 หลักการและทฤษฎี
2.1 Digital Image Processing 4
2.1.1 ภาพระดับเทา (Gray Image) 4
2.1.2 ภาพสี (Color Image) 5
2.1.3 ภาพแบบไบนารี (Binary Image) 5
2.2 Image Segmentation 6
2.2.1 การสร้างภาพแบบไบนารีด้วยการทำเทรชโฮล 6
2.2.1.1 การการหาค่าเทรชโฮลโดยการกำหนดค่าเอง 7
2.2.1.2 การหาค่าเทรชโฮลแบบอัตโนมัติ 7
2.2.2 Region Labeling 11
2.3 Morphological Image Processing 13
2.3.1 Thinning Algorithm 13

สารบัญ (ต่อ)

หน้า

2.4 ระบบฐานข้อมูล 15

2.4.1 โครงสร้างข้อมูล 15

2.4.2 รูปแบบของระบบฐานข้อมูล 16

2.4.3 คีย์ในฐานข้อมูลเชิงสัมพันธ์ 17

2.4.4 ความสำคัญของระบบฐานข้อมูล 17

2.4.5 โปรแกรมฐานข้อมูลที่นิยมใช้ 18

2.5 การติดต่อกับฐานข้อมูล ODBC 19

2.6 Graphical User Interface (GUI) 19

2.6.1 ส่วนประกอบของ GUI ใน MATLAB 20

2.6.2 ส่วนประกอบสำคัญของ Application M-file ที่สร้างโดย GUIDE 20

2.6.3 ตัวอย่างการสร้าง GUI 21

บทที่ 3 วิธีการดำเนินงานโครงงาน

3.1 โครงสร้างของโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์จากภาพ 28

3.2 การเตรียมภาพ 30

3.3 การหาตำแหน่งป้ายทะเบียนรถยนต์ 32

3.4 การแยกหมายเลขทะเบียนออกจากแผ่นป้ายทะเบียนรถยนต์ 34

3.5 การรู้จำหมายเลขทะเบียนรถยนต์ 36

3.5.1 การเตรียมต้นแบบ 36

3.5.1.1 การเตรียมต้นแบบด้วยการแทนแต่ละพื้นที่ ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ 36

3.5.1.2 การเตรียมต้นแบบด้วยการหาจุดปลายของตัวอักษร 38

3.5.2 การเปรียบเทียบข้อมูลรับเข้ากับข้อมูลต้นแบบ........ 41

3.5.2.1 การหาค่าความผิดพลาด แบบแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ 41

3.5.2.2 การหาค่าความผิดพลาด แบบการหาจุดปลาย 41

3.6 การจัดเก็บข้อมูล 44

3.7 การติดต่อฐานข้อมูล 47

3.8 โปรแกรมตรวจจับหมายเลขทะเบียนรถที่สร้างโดย GUI บน Matlab 48

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลองและการวิเคราะห์

4.1 ผลการทดลองส่วนของการเตรียมข้อมูลภาพ 49

4.2 ผลการทดลองส่วนของการหาตำแหน่งของป้ายทะเบียนรถยนต์ 50

4.3 ผลการทดลองส่วนของการแยกหมายเลขทะเบียนออกจากป้ายทะเบียน 51

4.4 ผลการทดลองส่วนของการรู้จำ 52

4.5 ผลการทดลองส่วนของการจัดเก็บข้อมูล 53

บทที่ 5 สรุป

5.1 สรุปผลการทดลอง 54

5.2 ปัญหาและอุปสรรค 54

5.3 แนวทางการพัฒนาในอนาคต 54

เอกสารอ้างอิง 55

ภาคผนวก ก. การติดตั้งโปรแกรมและการใช้โปรแกรม 56

ภาคผนวก ข. ตัวอย่างการแปลงไฟล์ 61

ภาคผนวก ค. ตัวอย่างโค้ดโปรแกรม 63

ประวัติผู้เขียนโครงงาน 83

สารบัญรูป

รูปที่ หน้า

2.1 คุณสมบัติของภาพระดับเทา 4

2.2 ภาพสี และค่าในแต่ละพิกเซล 5

2.3 การทำเทรชโฮล 8

2.4 ทำการเทรชโฮลกับภาพระดับเทา 8

2.5 Histogram ของภาพแบคทีเรีย ด้วยค่าเทรชโฮล เท่ากับ 102 9

2.6 ภาพแบบไบนารี และข้อมูลในแต่ละพิกเซล 10

2.7 รูปแบบ 8- neighborhood และแบบ 4- neighborhood สำหรับ Region Labeling 11

2.8 ตัวอย่างกระบวนการทำงานแบบ Region Labeling 12

2.9 ตำแหน่งของพิกเซล 13

2.10 พิกเซลที่พิจารณาบริเวณขอบชัน 14

2.11 เงื่อนไขที่ 3 และ 4 ของกฎ P1 15

2.12 เงื่อนไขที่ 3 และ 4 ของกฎ P2 15

2.13 หน้าต่าง Layout Editor 21

2.14 การสร้างกล่องข้อความ 22

2.15 การกำหนดชื่อตัวแปรฟังก์ชันและข้อความที่แสดงออกมาทางกล่องข้อความ 22

2.16 การสร้างปุ่มกด 23

2.17 การกำหนด Tag และข้อความ "Calculate" ที่แสดงออกมาทางปุ่มกด 23

2.18 การเปลี่ยนคุณสมบัติ Enable จาก on เป็น inactive 24

2.19 การสร้าง M-file แบบอัตโนมัติ 25

2.20 M-file ในส่วนฟังก์ชัน cal_Callback 25

2.21 สร้างโค้ดในฟังก์ชัน cal_Callback 26

2.22 กดรันโปรแกรม 26

2.23 ผลของโปรแกรม 27

3.1 โครงสร้างของโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์จากภาพ 28

3.2 ขั้นตอนการทำงานของโปรแกรม 29

3.3 ภาพระดับเทาตัวอย่างด้านหลังของรถยนต์ 30

3.4 ภาพรับเข้าที่ทำการเทรชโฮล 30

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.5	ตัวอย่างการเลือกค่าเทรชโฮลหลายค่าเพื่อการวิเคราะห์	31
3.6	ขั้นตอนการเตรียมภาพ	31
3.7	ตัวอย่างภาพที่ทำการกำหนดบล็อกให้กับ พิกเซลที่มี label เหมือนกัน	33
3.8	ขั้นตอนการหาตำแหน่งป้ายทะเบียนรถยนต์	33
3.9	การเทรชโฮล ภาพที่ตำแหน่งป้ายทะเบียนรถยนต์	34
3.10	ผลลัพธ์จากการแยกเลขทะเบียน	35
3.11	ขั้นตอนการแยกหมายเลขทะเบียนออกจากแผ่นป้ายทะเบียนรถยนต์	35
3.12	การแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ	37
3.13	ขั้นตอนการเตรียมต้นแบบ ในแบบที่ 1	37
3.14	เทียบพิกเซลที่เป็นจุดปลายกับพิกเซลที่ไม่ใช่จุดปลาย	38
3.15	จุดปลายของตัวอักษรจากภาพต้นแบบ “ล”	39
3.16	การจุดปลายของตัวอักษรจากภาพต้นแบบ “ส”	39
3.17	จุดของหัวตัวอักษรเทียบกับแกนกลางของหัว	40
3.18	ขั้นตอนการเตรียมต้นแบบด้วยการหาจุดปลายของตัวอักษร	40
3.19	ขั้นตอนการรู้จำหมายเลขทะเบียนรถยนต์	43
3.20	E/R diagram ของข้อมูลที่จัดเก็บ	44
3.21	เริ่มสร้างฐานข้อมูลชื่อ “Car Registration Database”	45
3.22	การกำหนดคุณสมบัติต่างๆ ให้กับตาราง	45
3.23	ตัวอย่างฐานข้อมูลของโปรแกรม	46
3.24	หน้าต่างตารางว่างเปล่า เพื่อเตรียมรอรับข้อมูล	46
3.25	ส่วนต่างๆของโปรแกรม	48
4.1	เทรชโฮลด้วยค่าต่างๆ	49
4.2	เทรชโฮล ด้วยค่าต่างๆ	49
4.3	โปรแกรมในขั้นตอนตรวจจับหมายเลขทะเบียนรถที่เกิดข้อผิดพลาด	50
4.4	ภาพที่ไม่สามารถทำการตรวจสอบ	50
4.5	ผลลัพธ์จากการแยกเลขทะเบียน	51
4.6	รถยนต์ที่มีป้ายทะเบียนแบบเก่า	52

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.7 โปรแกรมในขั้นตอนตรวจจับหมายเลขทะเบียนรถที่ถูกต้องจำนวน 10 คัน 53

4.8 ข้อมูลที่เก็บลงในฐานข้อมูล 53

ก-1 ส่วนของ Data Sources (ODBC) เพื่อทำการติดตั้งให้ใช้งานกับโปรแกรม 56

ก-2 เลือกที่อยู่ของไฟล์ Car Registration Database.mdb 57

ก-3 ไปยังที่อยู่ของไฟล์ Car Registration Database.mdb 57

ก-4 หน้าต่างโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์ 58

ก-5 ทำการเลือกไฟล์ภาพรถยนต์ที่ต้องการ 58

ก-6 เมื่อเลือกไฟล์ภาพแล้วทำการตรวจจับเพื่อหาหมายเลขทะเบียน 59

ก-7 หมายเลขทะเบียนรถยนต์ที่ตรวจจับได้จากไฟล์ภาพ 59

ก-8 การเข้าสู่โปรแกรม Microsoft Access เพื่อตั้งรหัสผ่านฐานข้อมูล 60

ข-1 หลังการสร้าง GUI ชื่อไฟล์ Example บนโปรแกรม Matlab 7.0 62

ข-2 Execute-file ที่เพิ่มขึ้นมาจากใช้คำสั่ง mcc 62

มหาวิทยาลัยนเรศวร

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

โดยทั่วไปสังคมขนาดเล็กหรือใหญ่ ต่างประสบปัญหาการลักขโมย ทั้งรถยนต์ และรถจักรยานยนต์ ทำให้ประชาชนคนเราต้องการความปลอดภัยต่อทรัพย์สินของตนเอง อีกทั้งสังคมในปัจจุบันนี้ อัตราการแข่งขันสูง ทำให้การดำเนินชีวิต ต้องการความสะดวกและรวดเร็วด้วย

การนำคอมพิวเตอร์มาช่วยแบ่งเบาภาระให้กับมนุษย์ ด้วยวิธีคิดเสมือนมนุษย์ เพื่อให้ได้ผลลัพธ์ที่ถูกต้องและรวดเร็ว มีบทบาทมากในโลกอนาคต ครั้งหนึ่งเราได้ศึกษาและสังเกตเห็นสิ่งที่จะช่วยลดภาระงานให้กับเจ้าหน้าที่ได้ เมื่อเราได้เห็นเจ้าหน้าที่ตำรวจยืนส่องเครื่องตรวจจับความเร็ว ว่ามีรถทะเบียนใดบ้างที่ทำผิดกฎ ท่ามกลางแสงแดดจ้าในช่วงเวลากลางวัน นอกจากนี้การเข้า-ออกที่จอดรถตามห้างสรรพสินค้าในเวลาที่มีรถจำนวนมาก จะใช้เวลานาน ดังนั้นจึงได้มีการจัดทำโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์จากภาพนิ่งด้านหน้าหรือด้านหลังของรถ โดยใช้ความรู้ด้านการประมวลผลภาพมาวิเคราะห์สัญญาณภาพ เพื่อนำหมายเลขทะเบียนรถออกมาประยุกต์ใช้งาน

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อพัฒนาอัลกอริทึมในการตรวจจับหมายเลขทะเบียนรถยนต์จากภาพนิ่งได้

1.2.2 เพื่อพัฒนาประสิทธิภาพในการรู้จำรูปแบบตัวอักษร ให้ถูกต้อง และแม่นยำ

1.2.3 เพื่อพัฒนาโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์ให้สามารถเก็บข้อมูลลงฐานข้อมูล

1.3 ขอบเขตของโครงงาน

โปรแกรมยังไม่สามารถทำงานในเวลาจริงได้ โดยไม่มีส่วนของการถ่ายภาพ แต่นำภาพที่มีอยู่แล้วมาใช้งานแทน ซึ่งโปรแกรมจะทำงานได้อย่างไม่สมบูรณ์ ถ้าข้อมูลภาพมีลักษณะ ดังต่อไปนี้

- เลขทะเบียนมีสีซีด หรือ สีถลอก ทำให้การรู้จำเกิดความผิดพลาด
- เลขทะเบียนมีคราบสกปรก ทำให้เลขทะเบียนเชื่อมติดกัน
- สภาพแสงภายนอก มีแสงสะท้อนที่แผ่นป้ายทะเบียนสูง ทำให้ไม่สามารถหาตำแหน่งที่เป็นส่วนของป้ายทะเบียนได้
- ความเอียงของรูปภาพ หรือมุมกล้องที่ถ่าย ทำให้โปรแกรมทำงานผิดพลาด

1.4 แผนการดำเนินการ

1.4.1 ศึกษาข้อมูลในการทำโครงงาน ดังนี้

- การประมวลผลภาพโดยเฉพาะ Image Segmentation และ Morphological Image Processing
- การใช้งานโปรแกรม Matlab 7.0 โดยเฉพาะการสร้าง Graphic User Interface (GUI) และ คำสั่งที่ใช้ในการติดต่อฐานข้อมูล
- โครงสร้างฐานข้อมูล Microsoft Access

1.4.2 รวบรวมข้อมูล

1.4.3 ออกแบบอัลกอริทึมที่จะสามารถตรวจจับหมายเลขทะเบียนรถยนต์จากภาพ

1.4.4 พัฒนาโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์

1.4.5 วิเคราะห์ผลที่ได้รับและสรุปผล

1.4.6 ปรับปรุงและแก้ไขโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์

1.5 ระยะเวลาในการดำเนินการ

กิจกรรม	ระยะเวลา พ.ศ. 2550			ระยะเวลา พ.ศ. 2551	
	ม.ค. - เม.ย.	มิ.ย. - ก.ย.	ก.ย. - พ.ย.	ธ.ค. - ก.พ.	ก.พ.- พ.ค.
1. ศึกษาข้อมูลและรวบรวมข้อมูลในการทำโครงงาน	↔				
2. ออกแบบอัลกอริทึมที่จะสามารถตรวจจับหมายเลขทะเบียนรถยนต์จากภาพ		↔			
3. พัฒนาโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์			↔		
4. วิเคราะห์ผลที่ได้รับและสรุปผล				↔	
5. ปรับปรุงและแก้ไขโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์					↔

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 เพื่อเป็นแนวทางในการพัฒนาโครงงานที่เกี่ยวข้องกับการอ่านหมายเลขทะเบียนรถยนต์ ซึ่งอาจนำเอาไปใช้งานในด้านต่างๆต่อไป

1.6.2 ได้อัลกอริทึมการรู้จำตัวอักษรและตัวเลขที่มีประสิทธิภาพสูงขึ้น สามารถวิเคราะห์ตัวอักษรที่มีความคล้ายคลึงกันได้ดีขึ้น โดยผลลัพธ์ถูกต้องและแม่นยำ

1.6.3 ลดภาระงานที่เกี่ยวกับการบันทึกข้อมูลของทะเบียนรถยนต์ เช่น นำไปใช้ทางด้านระบบรักษาความปลอดภัย ในการเข้า – ออกสถานที่ การตรวจจับผู้ขับขี่ที่กระทำผิด เป็นต้น

1.6.4 นำไปใช้งานทางการสืบค้นอาชญากรรม ในกรมตำรวจ

1.6.5 สร้างเครือข่ายสำหรับผู้ที่สนใจงานทางด้านการวิเคราะห์ภาพ และรู้จำรูปแบบ

1.7 รายละเอียดงบประมาณโครงงาน

1.7.1 ค่าวัสดุสำนักงาน	400	บาท
1.7.2 ค่าหนังสือ	800	บาท
1.7.3 ค่าสำเนาเอกสารและจัดทำรูปเล่ม	800	บาท
รวม	2,000	บาท

หมายเหตุ ถัวเฉลี่ยทุกรายการ

มหาวิทยาลัยนเรศวร

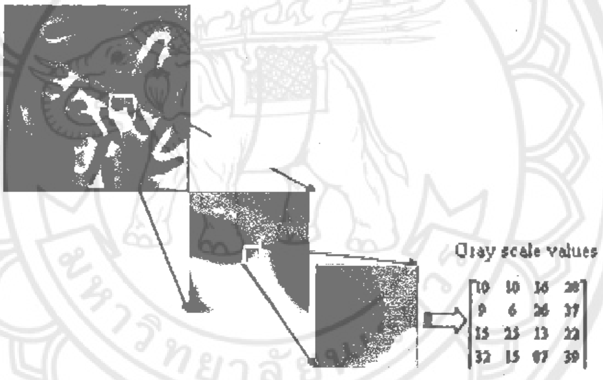
บทที่ 2
หลักการและทฤษฎีที่ใช้

2.1 Digital Image Processing [3]

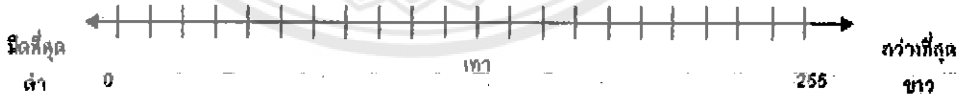
การประมวลผลรูปภาพดิจิตอล คือ การใช้อัลกอริทึมของคอมพิวเตอร์เพื่อใช้ในการประมวลผลรูปภาพ บนรูปภาพดิจิตอลที่เกี่ยวกับการประมวลผลภาพที่เกี่ยวกับการคำนวณและการจัดการสีในระดับพิกเซลหรือรูปแบบของสี มีการเปลี่ยนแปลงของสีได้ เพื่อประโยชน์ในการไปใช้ในการทำงาน

2.1.1 ภาพระดับเทา (Gray Image)

ค่าในแต่ละพิกเซลของภาพระดับเทา คือค่าความเข้มของแสง ณ แต่ละตำแหน่งของพิกเซล ซึ่งจะอยู่ในรูปของค่าระดับความเข้ม ดังรูปที่ 2.1 (ก) ค่าที่เป็นไปได้ของค่าระดับความเข้มจะขึ้นอยู่กับจำนวนบิตที่ใช้ ตัวอย่างเช่น ภาพระดับเทาจะมีค่าระดับความเข้ม 8 บิต หรือทั้งหมด 256 ระดับ ดังรูปที่ 2.1 (ข)



(ก)



(ข)

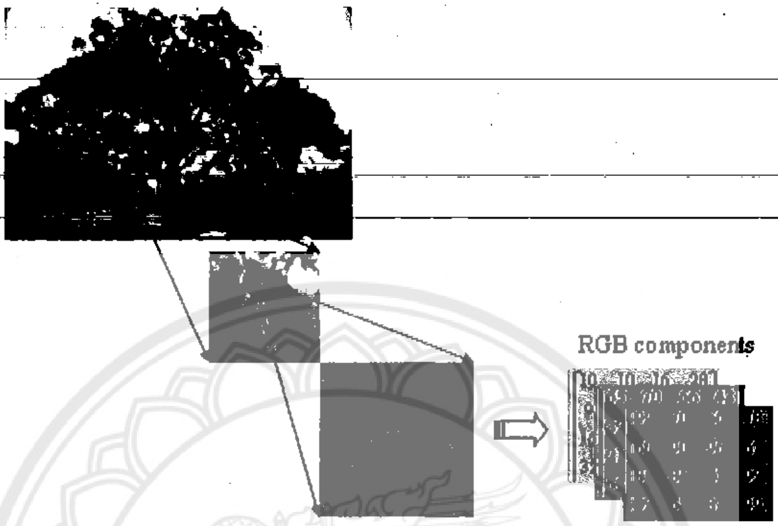
รูปที่ 2.1 คุณสมบัติของภาพระดับเทา

(ก) ภาพระดับเทา และค่าในแต่ละพิกเซล

(ข) ค่าระดับความเข้ม 8 บิต

2.1.2 ภาพสี (Color Image)

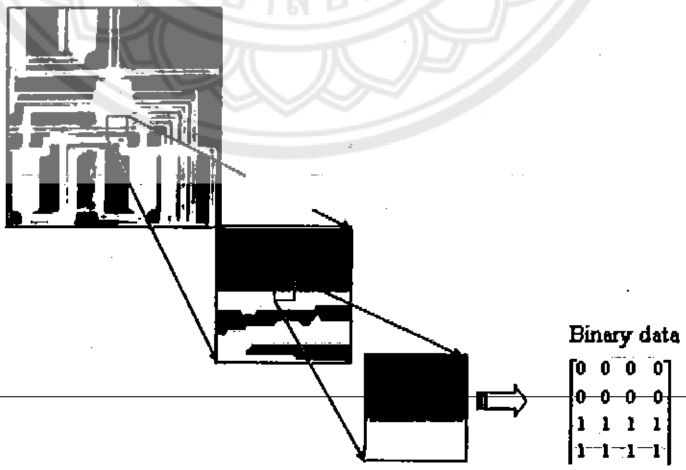
ค่าในแต่ละพิกเซลของภาพสี จะประกอบไปด้วยเวกเตอร์ที่แสดงค่าของสีแดง สีเขียว และสีน้ำเงิน อย่างละ 8 บิต ดังนั้น ภาพสี 1 พิกเซล จะประกอบไปด้วยจำนวนบิตทั้งหมด 24 บิต ทำให้ภาพสีมีจำนวนสีที่เป็นไปได้ทั้งหมด 2^{24} สี ตัวอย่างดังรูปที่ 2.2



รูปที่ 2.2 ภาพสี และค่าในแต่ละพิกเซล

2.1.3 ภาพแบบไบนารี (Binary Image)

ค่าในแต่ละ พิกเซลของภาพแบบไบนารี จะใช้แค่ 1 บิต ซึ่งจะมีค่าที่เป็น ไปได้คือ 0 (สีดำ) และ 1 (สีขาว) เท่านั้น ดังรูปที่ 2.3



รูปที่ 2.3 ภาพแบบไบนารี และค่าในแต่ละพิกเซล

2.2 Image Segmentation [1, 4] *

Image segmentation คือ กระบวนการแบ่งภาพเป็นพื้นที่ หรือวัตถุ ซึ่งมีความสำคัญมาก เพราะการวิเคราะห์เราจะไม่สามารถกระทำกับภาพทั้งภาพเพื่อให้ได้ผลลัพธ์ที่ดีได้ แต่เราจะวิเคราะห์ภาพเป็นส่วนๆไป เช่น การหาวัตถุที่ควรจะเป็นป้ายทะเบียนรถยนต์ อยู่ตำแหน่งใดภายในภาพ เป็นต้น ซึ่ง Image segmentation นี้เป็นขั้นตอนสำคัญของการวิเคราะห์ภาพ มีประโยชน์ดังนี้ คือ

1. ลดจำนวนข้อมูลในรูปภาพที่ไม่จำเป็นในการวิเคราะห์ลง เนื่องจากการทำ Segmentation เป็นการแยกแยะระหว่างส่วนที่สนใจ เช่น วัตถุในภาพ กับส่วนที่ไม่ต้องการนั่นคือ ฉากหลัง เมื่อตัดข้อมูลในส่วนที่ไม่ต้องการออกไปจำนวนข้อมูลที่เหลือที่จำเป็นในการวิเคราะห์จริงจะลดลงอย่างมาก

2. จัดระเบียบข้อมูลในรูปภาพให้เป็นกลุ่มได้ดีขึ้น ข้อมูลภาพที่ผ่านการแบ่งแยกแล้ว จะมีโครงสร้างที่ชัดเจนขึ้นและนำไปใช้งานได้สะดวกขึ้น

การทำการ Segmentation จะทำให้สามารถแยกข้อมูลภาพในส่วนที่ต้องการออกมาได้ วิธีการพื้นฐานสำหรับการ Segmentation ได้แก่การพิจารณาความสว่างของภาพสำหรับภาพระดับเทา และความแตกต่างของสีสำหรับภาพสี นอกจากนี้ขอบของภาพและลักษณะของพื้นผิว ก็เป็นองค์ประกอบหนึ่งที่จะทำให้สามารถทำการ Segmentation ได้สะดวกยิ่งขึ้น Segmentation มีหลายวิธี เช่น

- การสร้างภาพแบบไบนารีด้วยการทำเทรชโฮล (Thresholding)
- กระบวนการแบ่งกลุ่มของพิกเซล (Region Labeling)
- กระบวนการขยายกลุ่มของพิกเซล (Region Growing)
- กระบวนการแบ่งและรวมกลุ่มของพิกเซล (Region Splitting and Merging)
- กระบวนการหาขอบของวัตถุ (Boundary detection)

ในโครงงานฉบับนี้ได้เลือกใช้วิธีการเทรชโฮล (Thresholding) และกระบวนการแบ่งกลุ่มของพิกเซล (Region Labeling) โดยจะอธิบายในหัวข้อ 2.2.1 และ 2.2.2 ต่อไป

2.2.1 การสร้างภาพแบบไบนารีด้วยการทำเทรชโฮล

ภาพที่มนุษย์มองเห็นโดยทั่วไป จัดเป็นภาพสี ซึ่งคุณสมบัติภาพสีนี้จะมี 3 ระนาบ ทำให้การวิเคราะห์ภาพมีความซับซ้อน และยากต่อการพิจารณา การลดทอนความเป็นสีสามารถทำได้โดยการแปลงให้เป็นภาพระดับเทา ขั้นตอนดังกล่าวนั้นมีหลายวิธี เช่น เลือกระนาบใดระนาบหนึ่งจากภาพสี หรือ นำทั้ง 3 ระนาบมาทำการหาค่าเฉลี่ยเพื่อให้เหลือเพียงระนาบเดียว เป็นต้น ซึ่งภาพระดับเทาที่ได้จะซับซ้อนน้อยกว่าภาพสีเมื่อนำมาวิเคราะห์วัตถุต่างๆภายในภาพ

ภาพแบบไบนารี คือ ภาพที่มีระดับความเข้มเพียง 1 บิต หรือ 2 ระดับ คือ ในหนึ่งพิกเซลมีค่าเพียงสองค่าคือ 0 และ 1 เท่านั้น โดย พิกเซลที่มีค่าเท่ากับ 1 จะมีสีขาว และ พิกเซลที่มีค่าเท่ากับ 0 จะมีสีดำ ในการสร้างภาพแบบไบนารี นั้น จะใช้วิธีการทำเทรชโฮล มีหลักการพิจารณาคือ จะพิจารณาจุดของภาพว่าจุดใดควรจะเป็นจุดดำหรือจุดขาว โดยเปรียบเทียบระหว่างค่าในพิกเซลเริ่มต้นกับค่าคงที่ค่าหนึ่ง

หรือค่าเทรชโฮล นั่นเอง ซึ่งวิธีเทรชโฮล นิยมใช้มากในกรณีที่วัตถุ และพื้นหลัง ของภาพมีความแตกต่างกันอย่างชัดเจน

บางครั้งการเลือกค่าเทรชโฮลที่ไม่เหมาะสม คือ เลือกค่าเทรชโฮลที่มากหรือน้อยจนเกินไป จะทำให้ผลลัพธ์ที่ได้ อาจจะขาดความคมชัดและรายละเอียดบางส่วนขาดหายไป กล่าวคือภาพที่ได้อาจจะมืดเกินไป หรือสว่างจนเกินไป หรือบางทีอาจมีสิ่งรบกวนเกิดขึ้น ดังนั้นจะแก้ปัญหาเหล่านี้ด้วยการเลือกค่าเทรชโฮลที่เหมาะสม การเลือกค่าเทรชโฮลนั้นมีอยู่หลายวิธี โดยแต่ละวิธีเหมาะสมกับลักษณะการทำงานที่แตกต่างกันไป เช่น การหาค่าเทรชโฮลโดยการกำหนดค่าเอง และการหาค่าเทรชโฮลแบบอัตโนมัติ แต่ละวิธีอธิบายได้ดังนี้

2.2.1.1 การหาค่าเทรชโฮลโดยการกำหนดค่าเอง

วิธีนี้เป็นที่นิยมเพราะว่า เป็นวิธีที่ง่ายที่สุด โดยการสุ่มค่าเทรชโฮล การสุ่มค่าเทรชโฮลนั้นขึ้นอยู่กับประสบการณ์ของผู้ใช้เอง เลือกค่าคงที่ที่เหมาะสมค่าหนึ่ง เป็นค่าคงที่ที่อยู่ระหว่างค่าต่ำสุดและสูงสุด ของระดับความเข้มของข้อมูลภาพรับเข้า ถ้าเป็นภาพระดับเทาที่มีความลึก 8 บิต ค่าเทรชโฮลที่ได้จะอยู่ระหว่าง 0 ถึง 255 เช่น ถ้าเราเลือกค่าเทรชโฮลเท่ากับ 128 จะทำให้พิกเซลที่มีความเข้มมากกว่าหรือเท่ากับ 128 จะกลายเป็นสีขาว และ พิกเซลที่มีความเข้มน้อยกว่า 128 จะกลายเป็นสีดำ ดังสมการ 2.1

$$g(x, y) = \begin{cases} 1 & f(x, y) > T \\ 0 & f(x, y) < T \end{cases} \qquad (2.1)$$

โดยที่	f(x,y)	คือ	ข้อมูลภาพรับเข้า ที่มีระดับความเข้ม 0 ถึง L
	g(x,y)	คือ	ข้อมูลภาพผลลัพธ์ เป็นภาพแบบไบนารี
	T	คือ	ค่าเทรชโฮล
	1	คือ	จุดสีขาว
	0	คือ	จุดสีดำ

2.2.1.2 การหาค่าเทรชโฮลแบบอัตโนมัติ

วิธีการนี้เหมาะสำหรับภาพที่มีความเข้มของวัตถุ และพื้นหลังแยกกันอย่างชัดเจน วิธีนี้เป็นการหาค่าเทรชโฮลโดยไม่ต้องให้ผู้ใช้กำหนดโดยจะพิจารณาจากค่ากลางใช้การคำนวณพื้นฐานทางสถิติเข้ามาช่วย ซึ่งจะได้ค่าเทรชโฮลที่มีความเหมาะสม โดยจะมีความสัมพันธ์กับความเข้มของภาพ เช่น ภาพที่มืด ค่าเทรชโฮลจะมีค่าน้อย หรือ ภาพที่มีความสว่างสูง ค่าเทรชโฮลก็จะสูงตามไปด้วย เป็นต้น

วิธีหาค่าเทรชโฮลนี้ จะเริ่มต้นจากนำค่ากลางระหว่าง ค่าระดับความเข้มสูงสุด และค่าระดับความเข้มต่ำสุดภายในภาพเป็นค่าเทรชโฮลเริ่มต้น (T0) จากนั้นใช้ค่า T0 แบ่งพิกเซลเป็นสองส่วน คือ

ส่วนที่มีความเข้มน้อยกว่าค่า T0 และส่วนที่มีความเข้มมากกว่าหรือเท่ากับ T0 จากนั้นหาค่าเฉลี่ยของกลุ่มพิกเซลทั้งสอง ได้เป็น $\mu 1$ และ $\mu 2$ ตามลำดับ แล้วทำการหาค่าเฉลี่ยระหว่าง $\mu 1$ และ $\mu 2$ จะได้ค่าเทรชโฮลใหม่ (T) ถึงขั้นตอนนี้ จะได้ค่าเทรชโฮล 2 ค่า คือ T0 และ T ต่อมาจะทำการหาความแตกต่างระหว่าง T0 กับ T ถ้ามีค่าน้อยกว่าค่าที่กำหนดไว้ (D) ซึ่งส่วนใหญ่จะกำหนดไว้น้อยกว่า 5 ก็ให้ใช้ T เป็นค่าเทรชโฮล ถ้าค่าความแตกต่างยังมากอยู่ แสดงว่าค่าเทรชโฮลที่ได้ยังไม่เหมาะสม จะต้องกลับไปหาค่าเทรชโฮลต่อด้วยขั้นตอนเดิมอีกครั้ง โดยให้ T เท่ากับ T0 ได้แสดงวิธีการคำนวณไว้ ดังนี้

1. T0 = (Min(f(x,y)) + Max(f(x,y))) /2

   โดย

   f(x,y) คือ ภาพระดับเทา

   Min(f(x,y)) คือ ระดับความเข้มที่น้อยที่สุดของภาพ

   Max(f(x,y)) คือ ระดับความเข้มที่มากที่สุดของภาพ

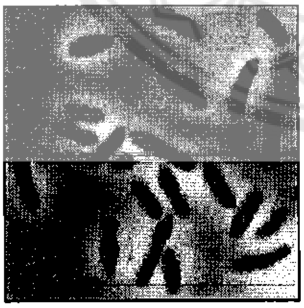
2. แบ่งพิกเซล เป็น 2 ส่วน

   g1(x, y) คือ f(x,y) ที่มีระดับความเข้ม < T0

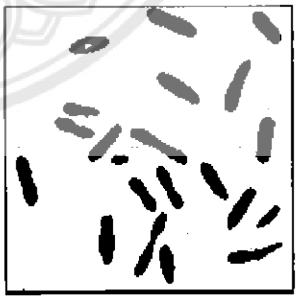
   g2(x, y) คือ f(x,y) ที่มีระดับความเข้ม > =T0

3. คำนวณค่าเฉลี่ยของ g1(x, y) = $\mu 1$ และ g2(x, y) = $\mu 2$
4. ตั้งค่า T ใหม่โดยให้ T = $(\mu 1+\mu 2)/2$
5. ย้อนกลับไปทำข้อ 2 อีกครั้ง จนกว่าค่า T-T0 จะมีค่าน้อยกว่าค่าที่กำหนดไว้ (D)

เมื่อได้ทำการเทรชโฮลกับภาพระดับเทาซึ่งเป็นภาพของแบคทีเรีย โดยใช้ค่าเทรชโฮล เท่ากับ 102 แล้ว จะได้ภาพแบบไบนารี ดังรูป 2.4 (ก) และ (ข) ตามลำดับ



(ก)



(ข)

รูปที่ 2.4 ทำการเทรชโฮลกับภาพระดับเทา

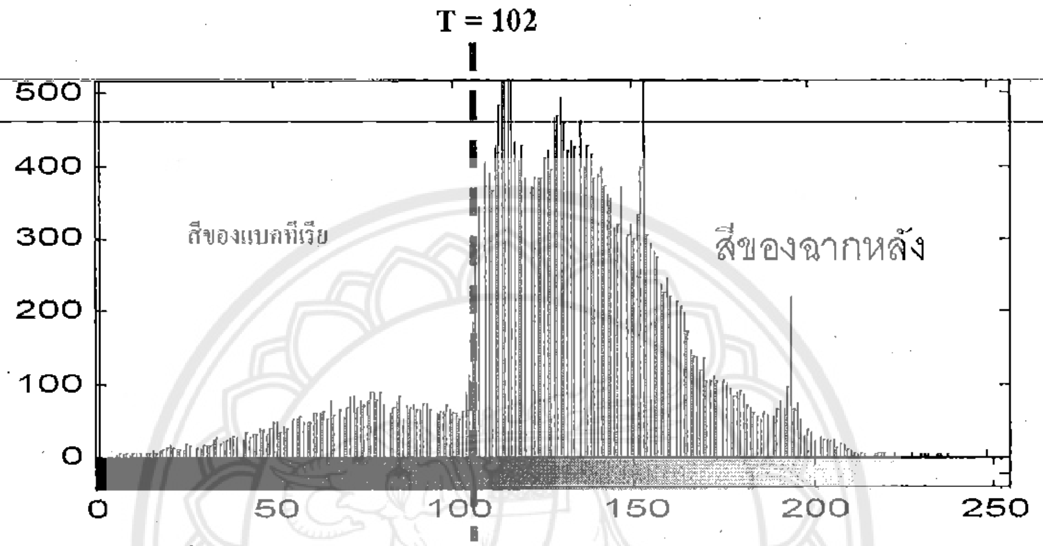
(ก) ภาพของแบคทีเรีย

(ข) ภาพของแบคทีเรียหลังการทำเทรชโฮล

เพื่อแสดงให้เห็นภาพของการเทรชโฮลได้ง่ายขึ้น การนำเสนอในรูปของ Histogram จะทำให้เข้าใจยิ่งขึ้น ซึ่ง Histogram คือ จำนวนพิกเซลที่มีในแต่ละระดับความเข้ม

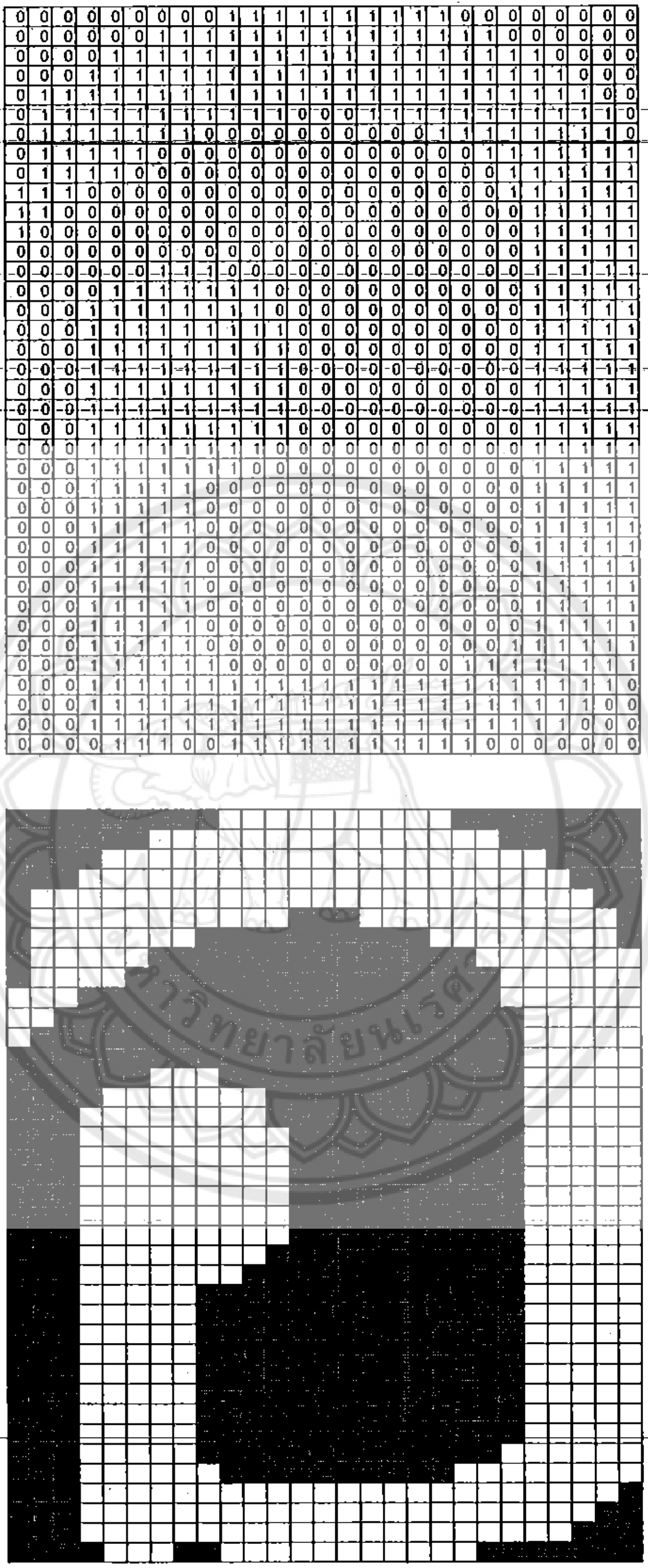
ภาพแบคทีเรียด้วยค่าเทรชโฮล เท่ากับ 102 แสดงดังรูป 2.5 โดยแกน x คือ ค่าระดับความเข้มอยู่ในช่วง 0 ถึง 255 และ แกน y คือ จำนวนพิกเซลของภาพแบคทีเรีย สัดส่วนภายในภาพมีดังนี้

- ส่วนที่เป็นแบคทีเรีย มีระดับความเข้มอยู่ในช่วง 0 ถึง 101
- ส่วนของฉากหลังจะมีจำนวนมากกว่าส่วนที่เป็นแบคทีเรียอยู่มาก มีระดับค่าความเข้มอยู่ในช่วง 102 ถึง 255



รูปที่ 2.5 Histogram ของภาพแบคทีเรีย ด้วยค่าเทรชโฮล เท่ากับ 102

ตัวอย่างภาพแบบไบนารี แสดงค่าในแต่ละพิกเซลของตัวอักษร “อ” ให้ 0 คือ สีดำ และ 1 คือ สีขาว แสดงดังรูปที่ 2.6



รูปที่ 2.6 ภาพแบบไบนารี และข้อมูลในแต่ละพิกเซล

2.2.2 Region Labeling

Region Labeling คือ กระบวนการหนึ่งในการแยกวัตถุต่างๆออกจากภาพที่เป็นภาพแบบไบนารีเท่านั้น ซึ่งจะมีความรวดเร็วสูงกว่าการแยกวัตถุด้วยกระบวนการอื่นๆ เพราะภาพที่นำมาพิจารณามีระดับความเข้มเพียงสองระดับ และสามารถแยกวัตถุออกจากกันเป็นกลุ่มๆได้อย่างชัดเจนโดยกระบวนการนี้ จะพิจารณาพิกเซลที่อยู่ติดกันและมีค่าเท่ากัน ถ้าอยู่ในเกณฑ์การรวม ก็จะนำพิกเซลนั้นมารวมกัน แต่ถ้าไม่อยู่ในเกณฑ์พิกเซลนั้นจะไม่ถูกนำมารวมเข้าไปในกลุ่มของพิกเซล

หลักการของการรวมพิกเซลนั้น จะพิจารณาเฉพาะ พิกเซล ที่มีค่า 1 เท่านั้น โดยการพิจารณาเริ่มตั้งแต่พิกเซลแรกที่อยู่ซ้ายสุดไปขวาสุด และจากบนสุดลงล่างสุดของภาพ (p) และทำการระบุตัวเลขกลุ่มว่าพิกเซลนี้อยู่ในกลุ่มไหนให้เป็น l(x,y) ซึ่งการรวมพิกเซลแบ่งเป็น 2 ลักษณะคือ แบบ 8-neighborhood นั่นคือ 8 พิกเซล รอบๆ และแบบ 4- neighborhood นั่นคือ 4 พิกเซลรอบๆ ดังรูปที่ 2.7

รูปที่ 2.7 รูปแบบ 8- neighborhood และแบบ 4- neighborhood สำหรับ Region Labeling

กำหนดให้

p(x, y) คือ พิกเซลที่พิจารณา

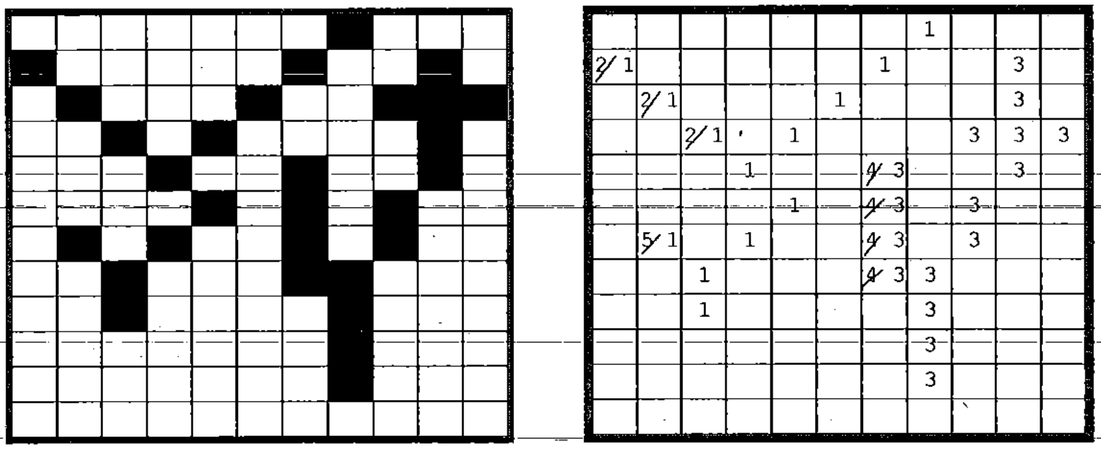
l(x, y) คือ Label ของ พิกเซล p(x, y) เริ่มต้นให้ l(x, y) แต่ละพิกเซล เป็น 0

NL คือ Neighbor ของ Label ที่จะนำมาพิจารณา

ES คือ Equivalent set ใช้อธิบายค่า Label ที่ต่างกันแต่เป็นวัตถุเดียวกัน

กระบวนการ

1. เมื่อ p(x, y) = 1
   - ถ้าไม่มีสมาชิกตัวใดใน NL > 0 ให้กำหนดค่าให้กับ l(x, y) = minimum(L)+1
   - ถ้ามีสมาชิก 1 ตัวใน NL > 0 ให้กำหนดค่าให้กับ l(x, y) = ค่าของ NL ตัวนั้น
   - ถ้ามีสมาชิก 1 ตัวขึ้นไปใน NL > 0 ให้กำหนดค่าให้กับ l(x, y) = ค่าน้อยที่สุดใน NL ที่ > 0 แล้วทำการเก็บค่าลงใน ES โดย NL ที่มากกว่า 0 คือ l(x, y)
2. ทำต่อไปอย่างนี้กับ พิกเซลถัดไปโดยทำจากซ้ายไปขวาและบนลงล่าง
3. ปรับค่า L ของภาพให้ตรงตาม ES



Equivalent Sets:

{1, 2, 5} ----> 1

{3, 4} ----> 3

รูปที่ 2.8 ตัวอย่างกระบวนการทำงานแบบ Region Labeling

จากรูป 2.8 แสดงกระบวนการทำงานแบบ Region Labeling แบบ 8- neighborhood โดยให้สีดำมีค่าเท่ากับ 1 และสีขาวเท่ากับ 0

- ถ้าพิจารณาจากซ้ายไปขวาจะพบว่าค่าของพิกเซลในแถวแรกหลักที่ 8 มีค่าเป็น 1 และไม่มีเพื่อนบ้านที่มีค่าเป็น 1 เลยจึงระบุเลขกลุ่ม หรือ ทำการ label ค่า 1 ให้พิกเซลนี้
- เมื่อไล่ไปจนสุดแถวจะทำการขึ้นแถวใหม่โดยเริ่มที่หลักแรก ตรวจสอบพบพิกเซลที่มีค่าเป็น 1 และยังไม่มีเพื่อนบ้านเช่นกัน จึงทำการ label ให้พิกเซลนี้เป็น 2
- เมื่อไล่ไปจนถึงหลักที่ 7 ก็พบพิกเซลที่มีค่าเป็น 1 และมีเพื่อนบ้านที่มี label เป็น 1 จึงจัดกลุ่มให้อยู่ในกลุ่มเดียวกับพิกเซลเพื่อนบ้านนี้ คือให้ label เป็น 1 นั่นเอง
- ทำเช่นนี้ไปจนถึงพิกเซลแถวที่ 5 หลักที่ 4 จะพบว่ามีเพื่อนบ้านอยู่ 2 พิกเซล และมี label ที่ต่างกัน ให้ทำการเลือก label ให้กับพิกเซลนี้เป็นเลข label น้อยสุด นั่นคือ 1 และเก็บเลข label ที่มากกว่า 1 ลงในตารางได้เป็น {1, 2} --> 1
- ทำไปเช่นนี้จนครบทั้งภาพ จะได้ตาราง Equivalent Sets เป็น {1, 2, 5} --> 1 และ {3, 4} --> 3
- ปรับค่า label ในแต่ละพิกเซลของภาพใหม่ ให้ตรงตามตารางคือ พิกเซลที่มี label เป็น 1, 2 หรือ 5 ให้มี label เป็น 1 ทั้งหมด และ พิกเซลที่มี label เป็น 3 หรือ 4 ให้มี label เป็น 3 ทั้งหมด

2.3 Morphological Image Processing [2, 5]

Morphological Image Processing คือ การประมวลผลภาพโดยการเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ โอเปอเรชันพื้นฐานโดยทั่วไปมักจะกระทำกับภาพแบบไบนารี แต่ที่กระทำกับภาพระดับเทานั้นก็มี ส่วนในโครงงานนี้จะกล่าวถึงแต่ส่วนที่กระทำกับภาพแบบไบนารีเท่านั้น มีหลายโอเปอเรชัน ได้แก่

- Dilation คือ การขยายภาพโดยมีสัดส่วนเท่ากันทั่วทั้งภาพ
- Erosion คือ การย่อภาพโดยมีสัดส่วนเท่ากันทั่วทั้งภาพ
- Closing คือ การย่อภาพ (Dilation) แล้วตามด้วยการทำขยายภาพ (Erosion)
- Opening คือ การทำขยายภาพ (Erosion) แล้วตามด้วยการย่อภาพ (Dilation)
- Skeleton คือ การหาโครงสร้างหลักของวัตถุ
- Thinning คือ การลบขอบของวัตถุ

ในโครงงานนี้จะกล่าวถึง Morphological เฉพาะกระบวนการ Thinning เท่านั้น โดยจะอธิบายไว้ในหัวข้อ 2.3.1

2.3.1 Thinning

Thinning คือ กระบวนการที่ทำให้วัตถุของภาพบางลงเรื่อยๆ โดยกัดกินขอบลงทีละชั้น จนเหลือแต่กระดูก ทำให้เปรียบเสมือนเป็นโครงสร้างของวัตถุนั้นๆ ซึ่งมีประโยชน์มากในการพิจารณาภาพที่มีลักษณะเป็นเส้น เช่น ลายนิ้วมือ โครงร่างเซลล์ประสาท โครงกระดูก และที่สำคัญที่สุดคือตัวอักษร ซึ่งเราจะนำกระบวนการ Thinning นี้เข้ามาช่วยในส่วนของการรู้จำตัวอักษร ดังจะกล่าวในบทถัดไป เป็นต้น

การทำ Thinning สามารถทำได้โดยอาศัยกฎ 2 ข้อ P1 และ P2 วิธีการทำ Thinning จะมี 2 ขั้นคือ ขั้นแรกจะใช้ กฎ P1 โดยการนำ Mask ขนาด 3x3 สแกนไปตามข้อมูลภาพและทำการพิจารณาพิกเซลบริเวณขอบภาพว่าสามารถลบได้หรือไม่ถ้าลบได้ให้หมายเหตุไว้แต่ยังไม่ต้องลบ หลังจากที่สแกนทั่วทั้งภาพก็ให้ทำการลบข้อมูลภาพ ขั้นที่สองใช้กฎ P2 และดำเนินการเหมือนการใช้กฎข้อที่ P1 เมื่อทำการลบข้อมูลภาพแล้ว ให้วนกลับไปทำซ้ำต่อไปเรื่อย ๆ จนไม่สามารถลบข้อมูลภาพออกได้อีก

ลักษณะของข้อมูลที่นำมาพิจารณาจะมีขนาดเท่ากับ 3x3 ดังรูปที่ 2.9 โดยที่พิกเซลปัจจุบันคือพิกเซลตรงกลาง (p_0) ดังนี้

p_8	p_1	p_2
p_7	p_0	p_3
p_6	p_5	p_4

รูปที่ 2.9 ตำแหน่งของพิกเซล

กำหนดให้

$N(p_0) = \sum_{i=1}^{8} p_i$ เป็นจำนวนของพิกเซลรอบ p_0 เมื่อ $p_0 = 0, 1$ และ i=0, 1, 2, …, 8

$T(p_0)$ คือจำนวนการเปลี่ยนแปลงของข้อมูลจาก 0 ไปเป็น 1 เมื่อพิจารณาข้อมูลใน $p_1, p_2, p_3, ...p_8, p_1$ ตามลำดับ เช่น

0	0	1
1	p_0	0
1	0	1

รูปที่ 2.10 พิกเซลที่พิจารณาบริเวณขอบชั้น

จากรูป จะได้ $N(p_0) = 4$ และ $T(p_0) = 3$

สำหรับกฎ P1 และ P2 จะมีลอจิกดังนี้คือ

$$\text{P1: } (2 <= N(p_0) <= 6) \,\&\, (T(p_0) = 1) \,\&\, (p1.p3.p5 = 0) \,\&\, (p3.p5.p7 = 0) \quad (2.2)$$

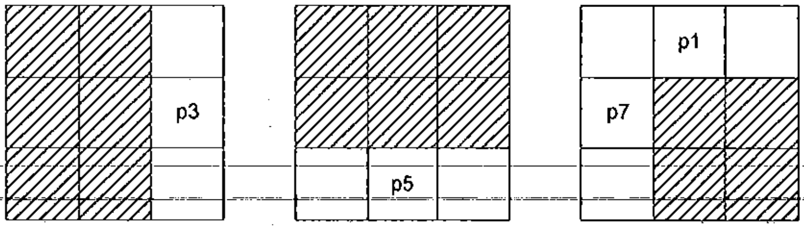
$$\text{P2: } (2 <= N(p_0) <= 6) \,\&\, (T(p_0) = 1) \,\&\, (p1.p3.p7 = 0) \,\&\, (p1.p5.p7 = 0) \quad (2.3)$$

เงื่อนไขแรกของ P1 และ P2 คือพิกเซลตรงกลาง (p_0) สามารถลบได้ก็ต่อเมื่อค่าของ *p*1 ถึง *p*8 มีค่าเท่ากับ 1(เป็นสีดำ) มากกว่า 1 ตำแหน่งและต้องไม่มากกว่า 6 ตำแหน่ง

กำหนดให้ $T(p_0) = 1$ หมายถึงต้องมีการเปลี่ยนแปลงของข้อมูลจาก 0 ไปเป็น 1 ของข้อมูล $p_1, p_2, p_3, ...p_8, p_1$ เพียงครั้งเดียว

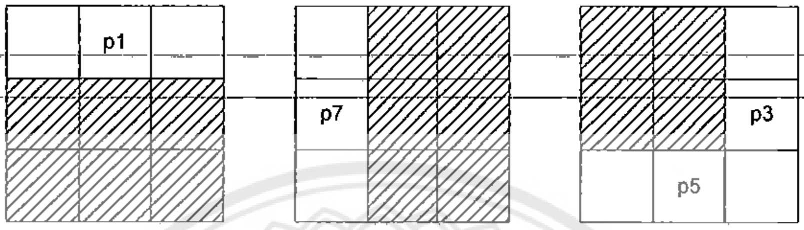
ในเงื่อนไขที่ 3 และ 4 คือ $(p1.p3.p5 = 0)$ และ $(p3.p5.p7 = 0)$ จะใช้ได้สำหรับเงื่อนไขที่ *p*3 = 0 หรือ *p*5 = 0 หรือ ถ้า *p*1 = 0 และ *p*7 = 0 ตามตัวอย่างดังกล่าวนี้แสดงไว้ดังรูปที่ 2.11 ซึ่งจะเห็นว่าเมื่อ *p*3 = 0 จะเป็นลักษณะของขอบด้านขวา เมื่อ *p*5 = 0 จะเป็นขอบด้านล่าง และเมื่อ *p*1 = 0 และ *p*7 = 0 จะเป็นมุมบนซ้าย

ใน P2 นั้นเราสามารถลบพิกเซล p_0 ได้ก็ต่อเมื่อ *p*1 = 0 หรือ *p*7 = 0 หรือ (*p*3 = 0และ *p*5 = 0) ดังแสดงในรูปที่ 2.12 ซึ่งจะเห็นว่าเมื่อ *p*1 = 0 จะเป็นขอบบน *p*7 = 0 จะเป็นขอบด้านซ้าย และเมื่อ (*p*3 = 0 และ *p*5 = 0) จะเป็นมุมล่างขวา



(ก) (ข) (ค)

รูปที่ 2.11 เงื่อนไขที่ 3 และ 4 ของกฎ P1



(ก) (ข) (ค)

รูปที่ 2.12 เงื่อนไขที่ 3 และ 4 ของกฎ P2

2.4 ระบบฐานข้อมูล [6]

ฐานข้อมูล (Database) หมายถึง กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน เพื่อลดความซ้ำซ้อนของข้อมูลและเก็บข้อมูลเหล่านี้ไว้ที่ศูนย์กลาง เพื่อที่จะนำข้อมูลเหล่านี้มาใช้ร่วมกัน

ระบบฐานข้อมูล (Database System) หมายถึง โครงสร้างสารสนเทศที่ประกอบด้วยรายละเอียดของข้อมูลที่เกี่ยวข้องกันที่จะนำมาใช้ในระบบต่าง ๆ ร่วมกัน

ระบบฐานข้อมูล จึงนับว่าเป็นการจัดเก็บข้อมูลอย่างเป็นระบบ ซึ่งผู้ใช้สามารถจัดการกับข้อมูลได้ในลักษณะต่าง ๆ ทั้งการเพิ่ม การแก้ไข การลบ ตลอดจนการเรียกดูข้อมูล ซึ่งส่วนใหญ่จะเป็นการประยุกต์นำเอาระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการฐานข้อมูล

2.4.1 โครงสร้างข้อมูล

1. บิท (Bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุด
2. ไบท์ (Byte) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำบิทมารวมกันเป็นตัวอักขระ
3. เขตข้อมูล (Field) หมายถึง หน่วยของข้อมูลที่ประกอบขึ้นจากตัวอักขระตั้งแต่หนึ่งตัวขึ้นไปมารวมกันแล้วได้ความหมายของสิ่งใดสิ่งหนึ่ง เช่น ชื่อ ที่อยู่ เป็นต้น
4. ระเบียน (Record) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำเอาเขตข้อมูลหลายๆ เขตข้อมูลมารวมกัน เพื่อเกิดเป็นข้อมูลเรื่องใดเรื่องหนึ่ง เช่น ข้อมูลของนักศึกษา 1 ระเบียน (1 คน) จะประกอบด้วย

- รหัสประจำตัวนักศึกษา 1 เขตข้อมูล

- ชื่อนักศึกษา 1 เขตข้อมูล
- ที่อยู่ 1 เขตข้อมูล

5. แฟ้มข้อมูล (File) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำเอาข้อมูลหลายๆ ระเบียนที่เป็นเรื่องเดียวกันมารวมกัน เช่น แฟ้มข้อมูลนักศึกษา แฟ้มข้อมูลลูกค้า แฟ้มข้อมูลพนักงาน

6. เอนทิตี้ (Entity) หมายถึง ชื่อของสิ่งใดสิ่งหนึ่ง ได้แก่ คน สิ่งของ การกระทำ สถานที่ ซึ่งต้องการจัดเก็บข้อมูล

7. แอททริบิวต์ (Attribute) หมายถึง รายละเอียดข้อมูลที่แสดงลักษณะและคุณสมบัติของเอนทิตี้หนึ่งๆ เช่น เอนทิตี้นักศึกษา ประกอบด้วย

- แอทริบิวต์รหัสนักศึกษา
- แอททริบิวต์ชื่อนักศึกษา
- แอททริบิวต์ที่อยู่นักศึกษา

8. ความสัมพันธ์ (Relationship) หมายถึง ความสัมพันธ์ระหว่างเอนทิตี้ เช่น ความสัมพันธ์ระหว่างเอนทิตี้นักเรียนและเอนทิตี้โรงเรียน เป็นลักษณะว่า นักเรียนแต่ละคนเรียนอยู่โรงเรียนใด ความสัมพันธ์ระหว่างเอนทิตี้ แบ่งออกเป็น 3 ประเภท ดังนี้

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-one Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตี้หนึ่งที่มีความสัมพันธ์กับข้อมูลในอีกเอนทิตี้หนึ่งในลักษณะหนึ่งต่อหนึ่ง (1:1)
- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-many Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตี้หนึ่ง ที่มีความสัมพันธ์กับข้อมูลหลายๆข้อมูลในอีกเอนทิตี้หนึ่ง ในลักษณะ (1:m)
- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-many Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลสองเอนทิตี้ในลักษณะกลุ่มต่อกลุ่ม (m:n)

จากคำศัพท์ต่างๆ ที่เกี่ยวข้องกับระบบฐานข้อมูลที่ได้กล่าวมาแล้วข้างต้น จึงอาจให้ นิยามของฐานข้อมูลในอีกลักษณะได้ว่า “ฐานข้อมูล” อาจหมายถึง โครงสร้างสารสนเทศ ที่ประกอบด้วยหลายๆ เอนทิตี้ที่มีความสัมพันธ์กัน

2.4.2 รูปแบบของระบบฐานข้อมูล

รูปแบบของระบบฐานข้อมูล มีอยู่ด้วยกัน 3 ประเภท คือ

1. ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

เป็นการเก็บข้อมูลในรูปแบบที่เป็นตาราง หรือเรียกว่า รีเลชั่น (Relation) มีลักษณะเป็น 2 มิติ คือเป็นแถว และเป็นคอลัมน์ การเชื่อมโยงข้อมูลระหว่างตาราง จะเชื่อมโยงโดยใช้แอททริบิวต์

(attribute) หรือคอลัมน์ที่เหมือนกันทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล ฐานข้อมูลเชิงสัมพันธ์นี้จะเป็นรูปแบบของฐานข้อมูลที่นิยมใช้ในปัจจุบัน

2. ฐานข้อมูลแบบเครือข่าย (Network Database)

ฐานข้อมูลแบบเครือข่ายจะเป็นการรวมระเบียนต่าง ๆ และความสัมพันธ์ระหว่างระเบียนแต่จะต่างกับฐานข้อมูลเชิงสัมพันธ์ คือ ในฐานข้อมูลเชิงสัมพันธ์จะแฝงความสัมพันธ์เอาไว้ โดยระเบียนที่มีความสัมพันธ์กันจะต้องมีค่าของข้อมูลในแอททริบิวต์ใดแอททริบิวต์หนึ่งเหมือนกัน แต่ฐานข้อมูลแบบเครือข่าย จะแสดงความสัมพันธ์อย่างชัดเจน ตัวอย่างเช่น

3. ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database)

ฐานข้อมูลแบบลำดับชั้น เป็นโครงสร้างที่จัดเก็บข้อมูลในลักษณะความสัมพันธ์แบบพ่อ-ลูก (Parent-Child Relationship Type: PCR Type) หรือเป็นโครงสร้างรูปแบบต้นไม้ (Tree) ข้อมูลที่จัดเก็บในที่นี้ คือ ระเบียน (Record) ซึ่งประกอบด้วยค่าของเขตข้อมูล (Field) ของเอนทิตี้หนึ่ง ๆ
ฐานข้อมูลแบบลำดับชั้นนี้คล้ายคลึงกับฐานข้อมูลแบบเครือข่าย แต่ต่างกันที่ฐานข้อมูลแบบลำดับชั้น มีกฎเพิ่มขึ้นมาอีกหนึ่งประการ คือ ในแต่ละกรอบจะมีลูกศรวิ่งเข้าหาได้ไม่เกิน 1 หัวลูกศร

2.4.3 คีย์ในฐานข้อมูลเชิงสัมพันธ์

ในระบบฐานข้อมูลเชิงสัมพันธ์เราจะต้องกำหนดชนิดของคีย์ เพื่อเป็นแอททริบิวต์พิเศษที่ทำหน้าที่บางอย่างเช่น เป็นตัวแทนของตาราง ซึ่งถูกกำหนดขึ้นมาเพื่อความสะดวกในการอ้างถึงทฤษฎีเกี่ยวกับการทำ Normalization ซึ่งมีดังนี้

- Primary Key (คีย์หลัก) จะเป็นฟิลด์ที่ไม่ซ้ำกันเลยในแต่ละเร็คคอร์ดในตารางนั้น และสามารถใช้คีย์นี้เป็นตัวแทนของตารางนั้นได้ทันที
- Candidate Key (คีย์คู่แข่ง) เป็นฟิลด์หนึ่งหรือหลายฟิลด์ที่พอเอามารวมกันแล้วคุณสมบัติ เป็น Primary Key และไม่ได้ถูกใช้เป็นคีย์หลัก
- Composite Key บางตารางหาฟิลด์ที่ไม่ซ้ำกันไม่ได้เลย จึงต้องใช้หลายๆฟิลด์มารวมกันเป็น Primary Key ซึ่งฟิลด์ที่ใช้รวมกันนี้เรียกว่า Primary Key
- Foreign Key เป็นฟิลด์ใดๆ ในตารางหนึ่ง (ฝั่ง Many) ที่มีความสัมพันธ์กับฟิลด์ที่เป็น Primary Key ในอีกตารางหนึ่ง (ฝั่ง One) โดยที่ทั้งสองตารางมีความสัมพันธ์แบบ One-to-many

2.4.4 ความสำคัญของระบบฐานข้อมูล

- จัดเก็บและบันทึกข้อมูล (Data Storage)
- ลดความซ้ำซ้อนของข้อมูล (Reduce Data Redundancy)
- สามารถใช้ข้อมูลร่วมกันได้ (Data Concurrency)
- ลดความขัดแย้งหรือแตกต่างกันของข้อมูล (Reduce Data Inconsistency)
- ป้องกันการแก้ไขข้อมูลต่างๆ (Protect Data Editing)

- ความถูกต้องของข้อมูลมีมากขึ้น (Data Accuracy)
- สะดวกในการสืบค้นข้อมูล (Data Retrieval or Query
- ป้องกันการสูญหายของข้อมูล หรือฐานข้อมูลถูกทำลาย (Data Security)
- เกิดการประยุกต์ใช้ระบบสารสนเทศ (Apply Information System)

2.4.5 โปรแกรมฐานข้อมูลที่นิยมใช้

โปรแกรมฐานข้อมูล เป็นโปรแกรมหรือซอฟแวร์ที่ช่วยจัดการข้อมูลหรือรายการต่าง ๆ ที่อยู่ในฐานข้อมูล ไม่ว่าจะเป็นการจัดเก็บ การเรียกใช้ การปรับปรุงข้อมูล

โปรแกรมฐานข้อมูล จะช่วยให้ผู้ใช้สามารถค้นหาข้อมูลได้อย่างรวดเร็ว ซึ่งโปรแกรมฐานข้อมูลที่นิยมใช้มีอยู่ด้วยกันหลายตัว เช่น Access, FoxPro, Clipper, dBase, FoxBase, Oracle, SQL เป็นต้น โดยแต่ละโปรแกรมจะมีความสามารถต่างกัน บางโปรแกรมใช้ง่ายแต่จะจำกัดขอบเขตการใช้งาน บ่งโปรแกรมใช้งานยากกว่า แต่จะมีความสามารถในการทำงานมากกว่า

- โปรแกรม Access นับเป็นโปรแกรมที่นิยมใช้กันมากในขณะนี้ โดยเฉพาะในระบบฐานข้อมูลขนาดใหญ่ สามารถสร้างแบบฟอร์มที่ต้องการจะเรียกดูข้อมูลในฐานข้อมูลหลังจากบันทึกข้อมูลในฐานข้อมูลเรียบร้อยแล้ว จะสามารถค้นหาหรือเรียกดูข้อมูลจากเขตข้อมูลใดก็ได้ นอกจากนี้ Access ยังมีระบบรักษาความปลอดภัยของข้อมูล โดยการกำหนดรหัสผ่านเพื่อป้องกันความปลอดภัยของข้อมูลในระบบได้ด้วย ซึ่งในโครงงานนี้เราได้เลือกใช้ MS Access เป็นฐานข้อมูลให้กับโปรแกรม
- โปรแกรม FoxPro เป็นโปรแกรมฐานข้อมูลที่มีผู้ใช้งานมากที่สุด เนื่องจากใช้ง่ายทั้งวิธีการเรียกจากเมนูของ FoxPro และประยุกต์โปรแกรมขึ้นใช้งาน โปรแกรมที่เขียนด้วย FoxPro จะสามารถใช้กลับ dBase คำสั่งและฟังก์ชั่นต่าง ๆ ใน dBase จะสามารถใช้งานบน FoxPro ได้ นอกจากนี้ใน FoxPro ยังมีเครื่องมือช่วยในการเขียนโปรแกรม เช่น การสร้างรายงาน
- โปรแกรม dBase เป็นโปรแกรมฐานข้อมูลชนิดหนึ่ง การใช้งานจะคล้ายกับโปรแกรม FoxPro ข้อมูลรายงานที่อยู่ในไฟล์บน dBase จะสามารถส่งไปประมวลผลในโปรแกรม Word Processor ได้ และแม้แต่ Excel ก็สามารถอ่านไฟล์ .DBF ที่สร้างขึ้นโดยโปรแกรม dBase ได้ด้วย
- โปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่งที่มีผู้นิยมใช้กันมาก โดยทั่วไปโปรแกรมฐานข้อมูลของบริษัทต่าง ๆ ที่มีใช้

อยู่ในปัจจุบัน เช่น Oracle, DB2 ก็มักจะมีคำสั่ง SQL ที่ต่างจากมาตรฐานไปบ้างเพื่อให้เป็นจุดเด่นของแต่ละโปรแกรมไป

2.5 การติดต่อกับฐานข้อมูล ODBC [7]

ODBC ย่อมาจาก Open Database Connectivity เป็น Application Programming Interface (API) คือ มาตรฐานในการเข้าถึงข้อมูลที่ถูกกำหนดขึ้น ที่อนุญาตให้คุณเชื่อมต่อฐานข้อมูลอื่นๆ ได้ ซึ่ง ODBC เป็นเสมือนตัวกลางระหว่างโปรแกรมที่เราใช้งาน กับฐานข้อมูลเหล่านั้น เช่น MS Access

ODBC มีพื้นฐานและการจัดตำแหน่งใกล้เคียงกับ Open Group มาตรฐานของภาษา SQL ซึ่ง ODBC เป็นระบบอินเตอร์เฟส โดยการใช้โปรแกรมนี้จะทำให้สามารถใช้คำสั่ง SQL เข้าถึงมาตรฐานข้อมูลโดยไม่จำเป็นต้องรู้จักคุณสมบัติการอินเตอร์เฟสของฐานข้อมูล ODBC จะรับรู้ภาษา SQL และแปลงเป็นคำสั่งของฐานข้อมูลแต่ละระบบ

ODBC จะใช้ในการติดต่อกับฐานข้อมูลในลักษณะ Client/Server โดยการติดต่อกับฐานข้อมูลใดๆจะไม่ขึ้นอยู่กับชนิดของโปรแกรมระบบฐานข้อมูล ซึ่งถ้ามีการประยุกต์ใช้ ODBC จะทำให้ลักษณะของการเขียนโปรแกรมเหมือนกันทั้งหมด หน้าที่หลักๆของ ODBC ก็คือ การติดต่อกับฐานข้อมูล ทำการจัดเตรียมคำสั่ง SQL เพื่อทำการส่งให้ DBMS อีกที แล้วก็จัดการประมวลผลของ Transaction ทำการรับ-ส่ง ข้อมูลที่ได้จาก ฐานข้อมูล กลับไปยังโปรแกรม พร้อมทั้งแจ้งข้อผิดพลาดถ้ารูปแบบคำสั่งนั้นผิดและความสามารถในลักษณะเช่นนี้ถือว่าเป็นคุณลักษณะเด่นที่สำคัญที่สุดของ ODBC

2.6 Graphical User Interface (GUI) [8]

Graphical User Interface (GUI) เป็น user interface ที่สร้างขึ้นด้วย graphical object แบบต่าง ๆ เช่น ปุ่มเมนู slider โดยทั่วไป objects เหล่านี้ ผู้ใช้คอมพิวเตอร์ส่วนใหญ่เข้าใจถึงความหมายและวิธีการใช้ object เหล่านี้เป็นอย่างดีอยู่แล้ว สิ่งสำคัญที่จะกล่าวถึงคือ หลังจากผู้ใช้ได้มีการกดปุ่มเมาส์ เลื่อน slider หรือ เลือกเมนู จะมีวิธีการกำหนดให้เกิดขั้นตอนต่อ ๆ ไป ตามที่ต้องการได้อย่างไร Application ต่าง ๆ ที่ออกแบบมาเป็น GUI จะพบว่าสามารถทำให้ผู้ใช้เข้าใจการใช้ application นั้นได้อย่างรวดเร็ว เพราะแทบจะไม่มีชุดคำสั่งใดให้รู้จำ การทำงานของ application จะเกิดขึ้นทันทีที่ได้ input จากผู้ใช้

สำหรับโครงงานนี้ จะอธิบายรายละเอียดต่างๆ ที่เกี่ยวข้องกับ GUIDE ซึ่งเป็นเครื่องมือหลักที่ใช้สร้าง GUI โดยจะเน้นในรายละเอียดว่า GUIDE นั้นมีเครื่องมืออะไรบ้าง และขั้นตอนในการทำงานของ GUIDE นั้นเป็นอย่างไร

การสร้าง GUI จะประกอบด้วยขั้นตอนสองขั้นตอน

- กำหนดและวางส่วนประกอบต่าง ๆ ลงบน GUI

- เขียนโปรแกรมเพื่อกำหนดการทำงานของส่วนประกอบต่าง ๆ ใน GUI

GUIDE นั้นโดยหลักใหญ่แล้วจะมีหน้าที่ในการวางส่วนประกอบที่ต้องการให้มีลงใน GUI จากนั้น GUIDE จะสร้าง M-file ที่บรรจุ handle ของวัตถุหรือ object ทั้งหมดที่สร้างขึ้นรวมทั้งคำสั่งให้ GUI ทำงาน นอกเหนือจากนั้น M-file จะให้แนวทางในการเขียนฟังก์ชัน ที่ทำงานหลังจากผู้ใช้กดเมาส์ปุ่มซ้ายหรือปรับเปลี่ยนค่าของวัตถุนั้น ซึ่งเรียกว่า callback ของวัตถุนั้น

2.6.1 ส่วนประกอบของ GUI ใน MATLAB

ดังที่ได้กล่าวมาก่อนแล้วว่า สามารถสร้าง GUI ขึ้นมาได้โดยการเขียนเป็น M-file ขึ้นมาล้วน ๆ แต่การใช้ GUIDE จะทำให้การทำงานง่ายขึ้นมากเพราะจะช่วยให้กำหนดตำแหน่งของวัตถุต่าง ๆ ได้โดยง่าย หลังจากนั้น GUIDE จะสร้างไฟล์ขึ้นมา 2 ไฟล์เพื่อเก็บและนำ GUI มาใช้ต่อไปซึ่งจะประกอบด้วย

- FIG-file ซึ่งจะบรรจุรายละเอียดของวัตถุต่างที่เป็นองค์ประกอบอยู่ในหน้าต่างรูปภาพที่เป็น GUI
- M-file ที่จะบรรจุฟังก์ชันที่กำหนดการทำงานของ GUI รวมถึง callback ทั้งหมด ซึ่ง callback เหล่านี้จะบรรจุเป็น sub function อยู่ใน M-file และ จะเรียก M-file ที่ควบคุมการทำงานของ GUI นี้ว่า Application M-file ดังนั้น Application M-file จะไม่มีข้อมูลใด ๆ เกี่ยวกับรูปแบบของส่วนประกอบที่บรรจุอยู่ใน GUI เช่นสี ขนาด ตำแหน่ง หรืออื่น ๆ เลย เพราะข้อมูลเหล่านั้นจะบรรจุอยู่ใน FIG-file

2.6.2 ส่วนประกอบสำคัญของ Application M-file ที่สร้างโดย GUIDE

GUIDE จะรวบรวมองค์ประกอบต่าง ๆ ภายใน GUI แล้วสร้าง Application M-file โดยอัตโนมัติ โดยมีรูปแบบของการสร้างที่ชัดเจน เพื่อให้ ได้โครงสร้างของ Application M-file จากนั้นสามารถนำ โครงสร้างที่สร้างโดยอัตโนมัตินั้นมาปรับแก้ เพื่อให้เกิดการควบคุม GUI ตามที่ต้องการ การกระทำดังกล่าวทำให้ได้ข้อได้เปรียบหลายประการ เช่น

- M-file จะประกอบด้วยคำสั่งที่จำเป็นในการควบคุม GUI ครบถ้วน
- M-file จะทำให้ส่งข้อมูลไปที่ส่วนต่าง ๆ ได้ง่าย สะดวก รวดเร็ว
- การใช้ M-file จะทำให้ส่งข้อมูลไปที่ส่วนต่าง ๆ ภายใต้ MATLAB ได้ง่ายขึ้น
- Application M-file จะสร้าง Sub function สำหรับ uicontrols ทุกแบบที่มีใน GUI เพื่อทำให้เขียน callback ต่าง ๆ ได้สะดวกขึ้น

แม้ว่า GUIDE จะให้ทางเลือก ว่าจะให้ GUIDE สร้างเฉพาะ fig-file เพื่อเก็บและใช้เป็นข้อมูลของ GUI ที่สร้างขึ้นเพียงอย่างเดียว แล้วเขียน M-file ขึ้นมาเอง แต่สำหรับผู้เริ่มเขียน GUI บน MATLAB คิดว่าการสร้าง GUI ด้วย GUIDE จะสะดวกกว่า หากให้ GUIDE สร้าง Application M-file ให้ด้วย ดังนั้นในการสร้าง GUI ด้วย GUIDE จะมีการกำหนดขั้นตอนดังนี้

- เลือก GUIDE Application option แล้วเลือกให้ GUIDE สร้างทั้ง FIG-file และ M-file
- การใช้ Layout Editor เพื่อวางรูปแบบของ GUI
- เรียนรู้การสร้าง Application M-file จาก GUIDE และเข้าใจถึงวิธีการทำเพื่อจะนำไปใช้ต่อ
- ปรับแก้ Application M-file ให้ทำงานตามที่กำหนด

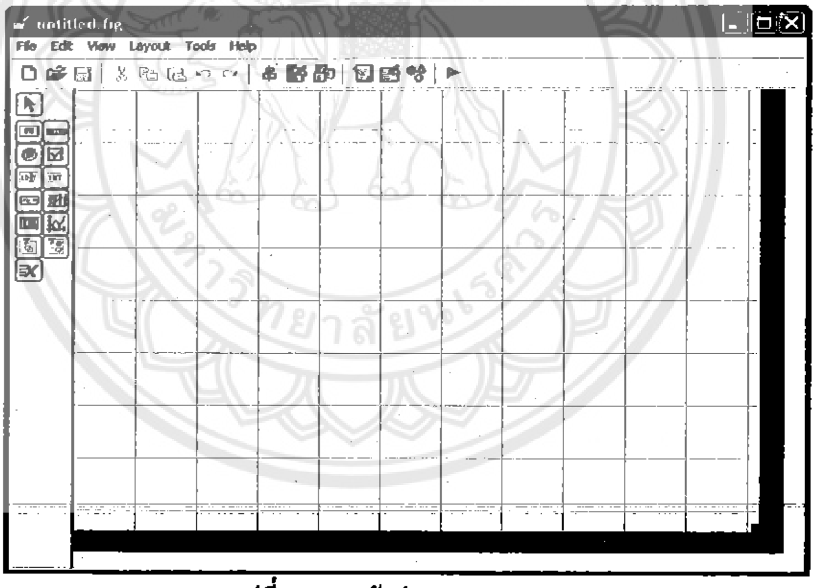
<u>หมายเหตุ</u>

GUIDE จะสร้าง Application M-file มีชื่อเดียวกับ Fig-file ที่กำหนดจาก Layout Editor เมื่อกำหนดให้ GUI นั้น Active จาก Layout Editor, GUIDE พยายามที่จะให้ Application M-file นี้ทำงานและเปิดหน้าต่าง GUI นั้นขึ้นมา

2.6.3 ตัวอย่างการสร้าง GUI

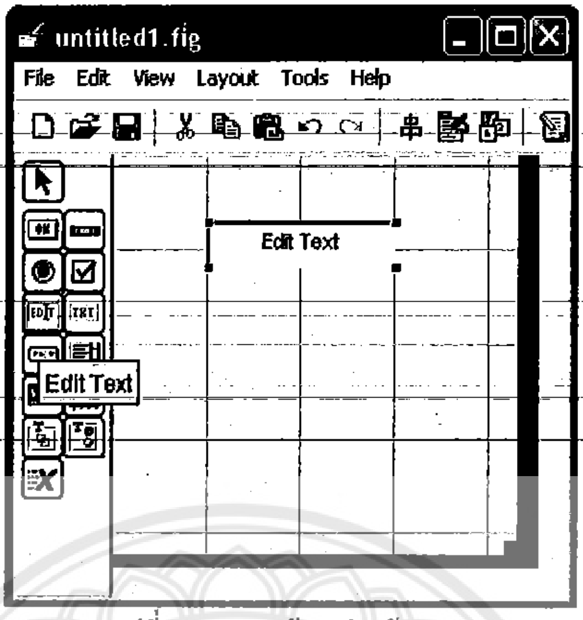
ในขั้นตอนนี้จะแสดงตัวอย่างการสร้าง GUI เบื้องต้น โดยโปรแกรมจะรับค่าตัวเลขสองค่า แล้วนำมาบวกกัน และแสดงผลลัพธ์ออกมาให้เห็นผ่าน GUI บน Matlab

เมื่อต้องการจะใช้ GUIDE นั้น ครั้งแรก บน Matlab Command Window ที่ prompt สั่ง » guide จากนั้น Layout Editor จะปรากฏขึ้น ซึ่งมีลักษณะดังรูปที่ 2.13



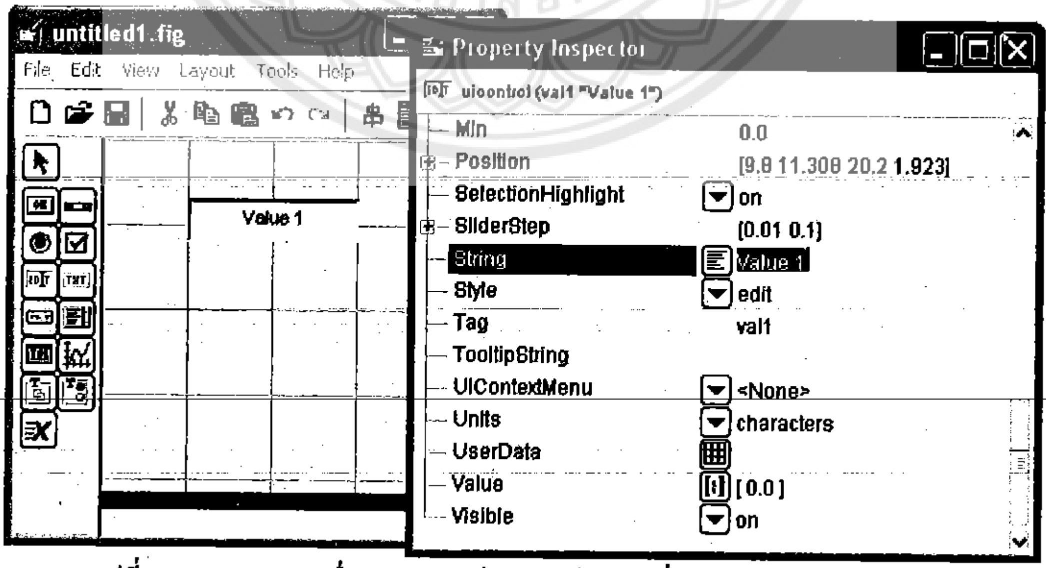
รูปที่ 2.13 หน้าต่าง Layout Editor

เราสามารถปรับขนาดหน้าต่าง GUI ได้ โดยใช้เมาส์เลือกที่มุมล่างขวาของกระดาน ในขั้นตอนนี้จะสร้างกล่องข้อความ ทำได้โดยนำเมาส์ไปคลิกเลือกตรงแถบเครื่องมือด้านซ้ายที่ชื่อว่า Edit Text แล้วลากเมาส์ค้างเพื่อกำหนดขนาดของกล่องข้อความได้ตามต้องการ จะปรากฏ กล่องข้อความสีขาวภายในมีข้อความเขียนไว้ว่า Edit Text แสดงดังรูป 2.14



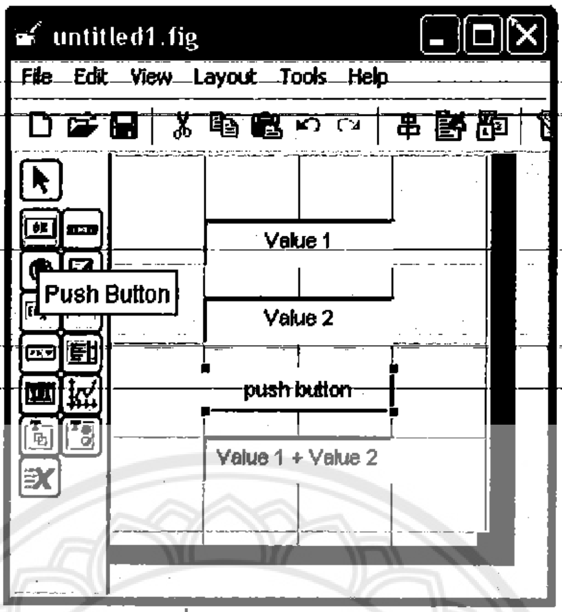
รูปที่ 2.14 การสร้างกล่องข้อความ

เราสามารถเปลี่ยนข้อความภายในกล่องข้อความได้โดยดับเบิลคลิกไปที่กล่องข้อความนั้น จะปรากฏหน้าต่าง Property Inspector ขึ้นมา หน้าต่างนี้จะบ่งบอกถึงส่วนประกอบ คุณสมบัติ และค่าต่างๆ ที่เกี่ยวข้องกับกล่องข้อความนี้ หนึ่งในนั้น เราจะสังเกตได้ว่ามีคุณสมบัติที่ชื่อว่า String เดิม ถ้าไม่ได้ไปเปลี่ยนมัน มันจะยังคงเป็น คำว่า "Edit Text" แต่ในขั้นตอนนี้เราจะเปลี่ยนข้อความให้เป็น "Value 1" และสร้างตัวแปรฟังก์ชั่นชื่อว่า val1 ซึ่งตัวแปรฟังก์ชั่นนี้ จะไปปรากฏอยู่ใน M-File ของ GUI นี้ สามารถกำหนดชื่อให้กับตัวแปรนี้ได้โดยแก้ไขในคุณสมบัติ Tag แสดงดังรูปที่ 2.15



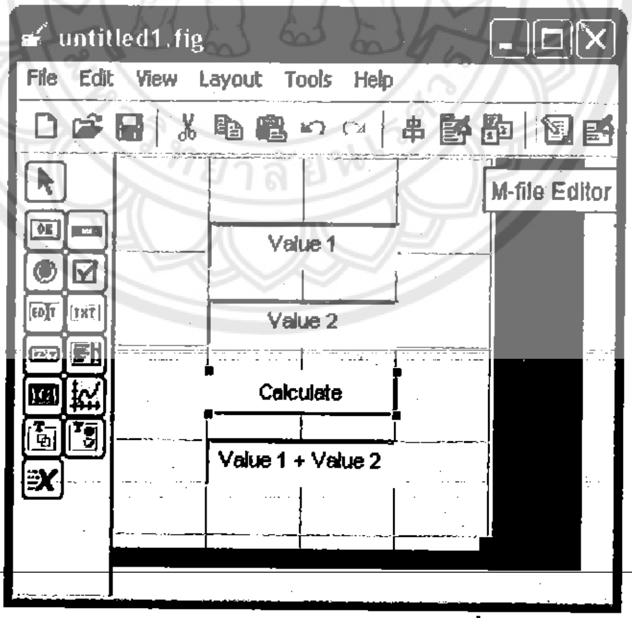
รูปที่ 2.15 การกำหนดชื่อตัวแปรฟังก์ชันและข้อความที่แสดงออกมาทางกล่องข้อความ

การสร้างปุ่มกดเพื่อทำการคำนวณ จะคลิกที่แถบเครื่องมือชื่อ Push Button แสดงดังรูปที่ 2.16



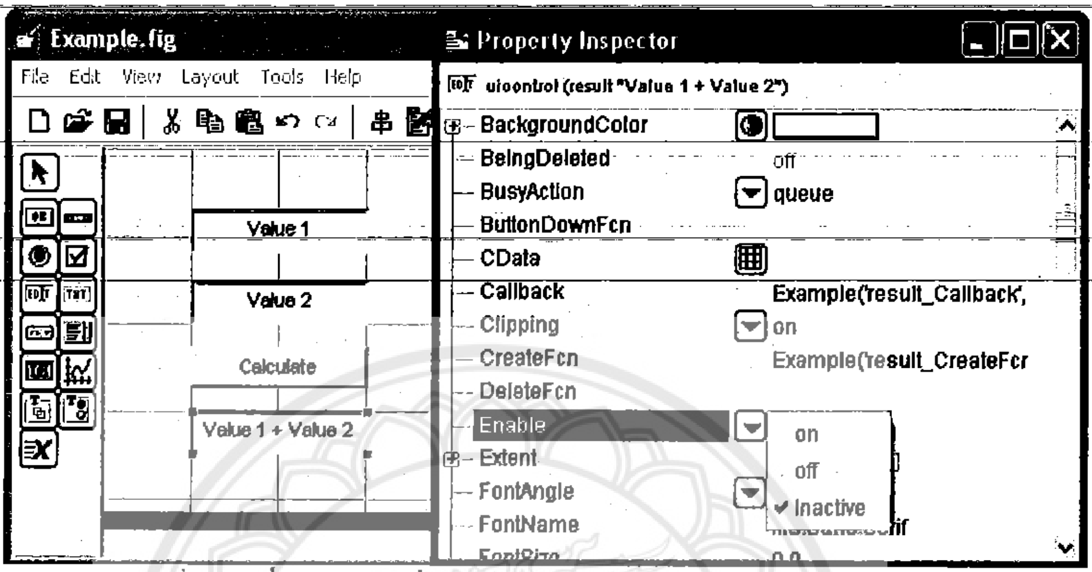
รูปที่ 2.16 การสร้างปุ่มกด

เราจะทำการเปลี่ยนชื่อปุ่มกดให้สอดคล้องกับจุดประสงค์ของโปรแกรม จาก**เดิม "Push button"** **เป็น** "Calculate" และ กำหนด Tag เป็น cal แสดงดังรูปที่ 2.17



รูปที่ 2.17 การกำหนด Tag และข้อความ "Calculate" ที่แสดงออกมาทางปุ่มกด

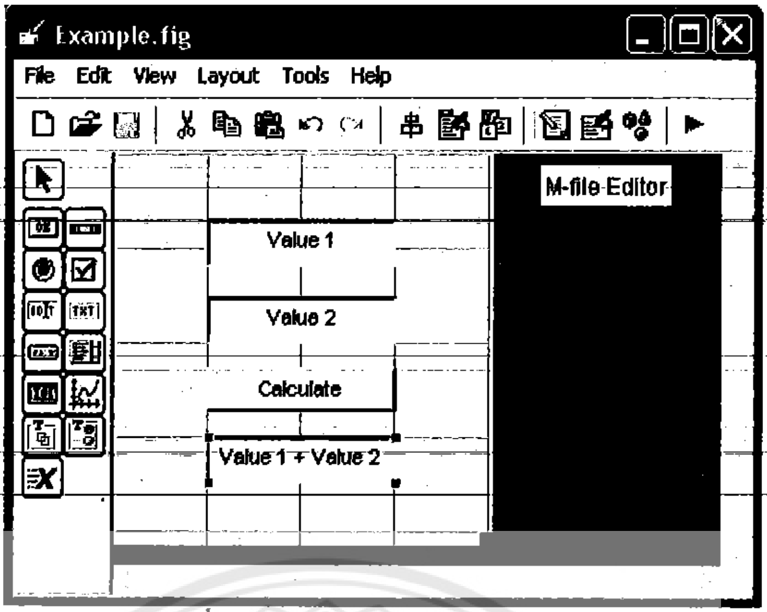
โดยปกติกล่องข้อความจะสามารถแก้ไขข้อความภายในได้ เนื่องจาก คุณสมบัติ Enable เป็น on แต่เนื่องจากกล่องข้อความในส่วนที่จะแสดงผลลัพธ์ ไม่ควรให้ถูกแก้ไขได้ เราจึงเปลี่ยนคุณสมบัติ Enable เป็น inactive แสดงดังรูปที่ 2.18



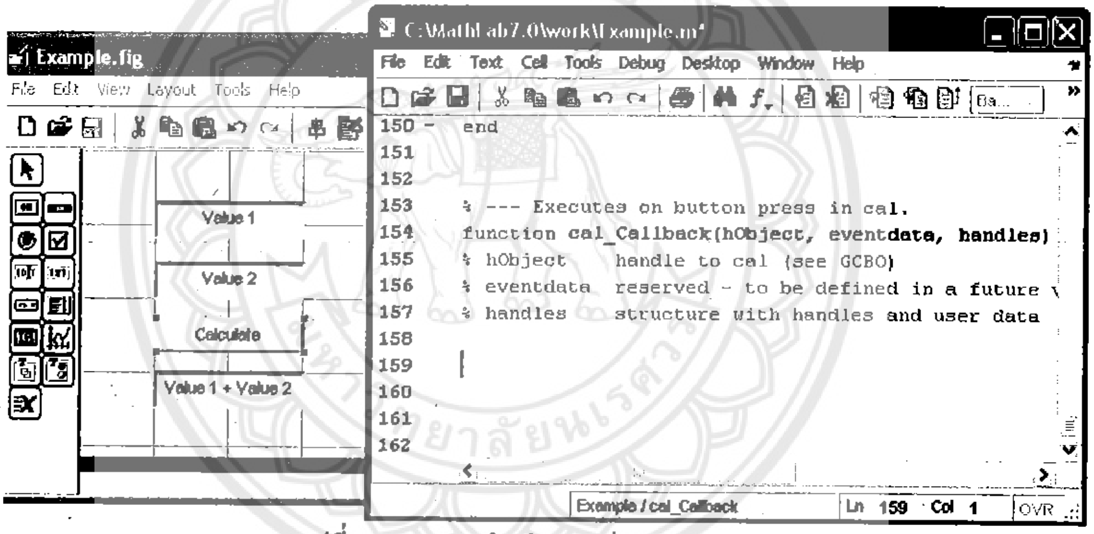
รูปที่ 2.18 การเปลี่ยนคุณสมบัติ Enable จาก on เป็น inactive

เมื่อทำการกำหนดคุณสมบัติต่างๆเรียบร้อยแล้ว ให้คลิกที่แถบเครื่องมือชื่อ M-file Editor แล้ว Matlab จะทำการสร้างโค้ดที่เป็น M-file ให้เอง โดยที่เราไม่ต้องสร้างขึ้นมา ซึ่งโค้ดที่ได้ จะมีลักษณะเป็นฟังก์ชันต่างๆ ชื่อของแต่ละฟังก์ชันจะสอดคล้องกับคุณสมบัติ Tag ที่เราได้กำหนดไว้ในตอนแรก และเมื่อเรากดปุ่ม Calculate โปรแกรมจะทำงานในส่วนที่เป็นฟังก์ชัน cal_Callback แสดงดังรูปที่ 2.19 และ 2.20

ห้องสมุดคณะวิศวกรรมศาสตร์ 05100043



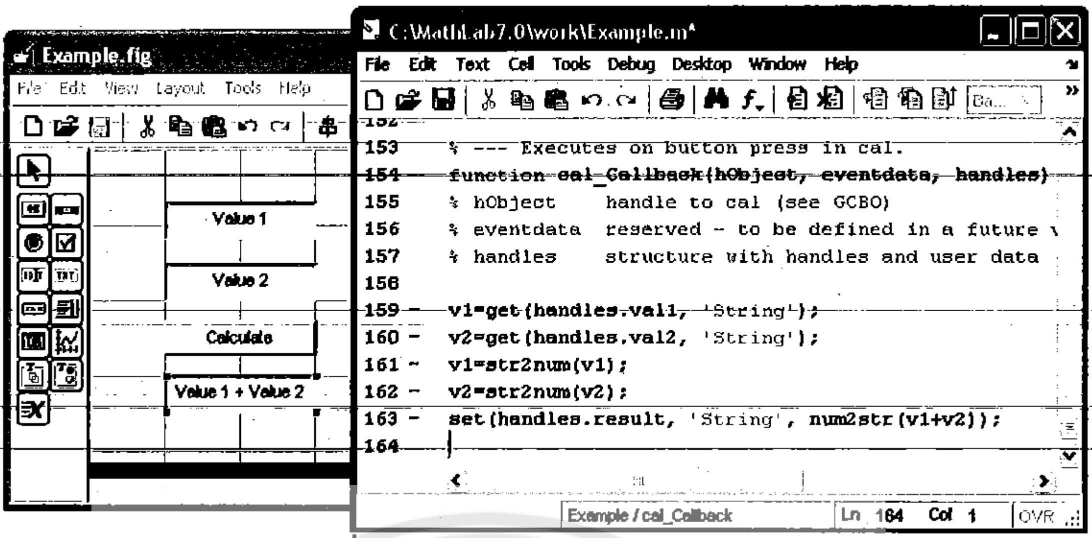
รูปที่ 2.19 การสร้าง M-file แบบอัตโนมัติ



รูปที่ 2.20 M-file ในส่วนฟังก์ชัน cal_Callback

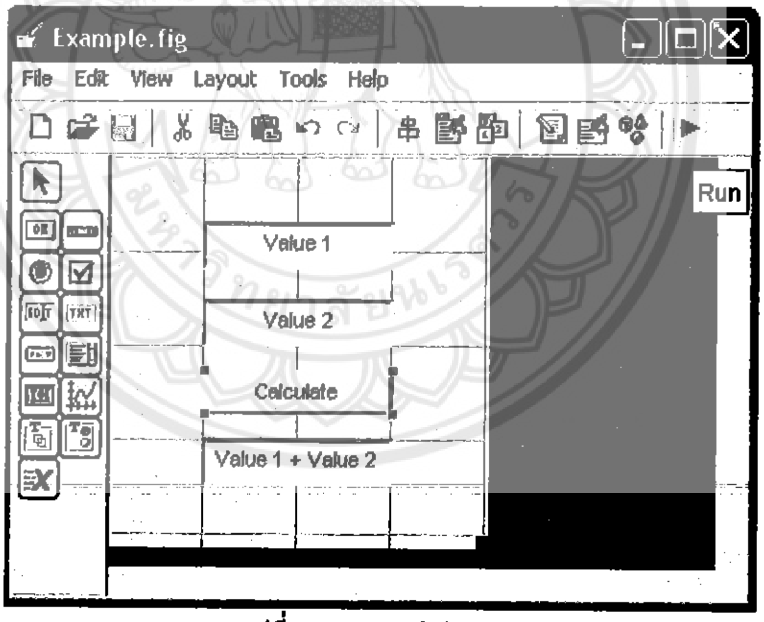
เนื่องจากภายในฟังก์ชัน cal_Callback ยังไม่มีการเขียนโค้ดใดๆลงไป เมื่อกดปุ่ม calculate ก็จะยังไม่มีการคำนวณใดๆเกิดขึ้น เราจึงต้องสร้างโค้ดลงในฟังก์ชันนี้เป็น

```
v1=get(handles.val1, 'String');     % อ่านค่าจากกล่องข้อความที่มี Tag เป็น val1 เก็บลงตัวแปร v1
v2=get(handles.val2, 'String');     % อ่านค่าจากกล่องข้อความที่มี Tag เป็น val2 เก็บลงตัวแปร v2
v1=str2num(v1);     % กำหนดให้ตัวแปร v1 เป็นตัวเลข
v2=str2num(v2);     % กำหนดให้ตัวแปร v2 เป็นตัวเลข
set(handles.result, 'String', num2str(v1+v2));     % แสดงผลบวกทางกล่องข้อความที่มี Tag เป็น result
```

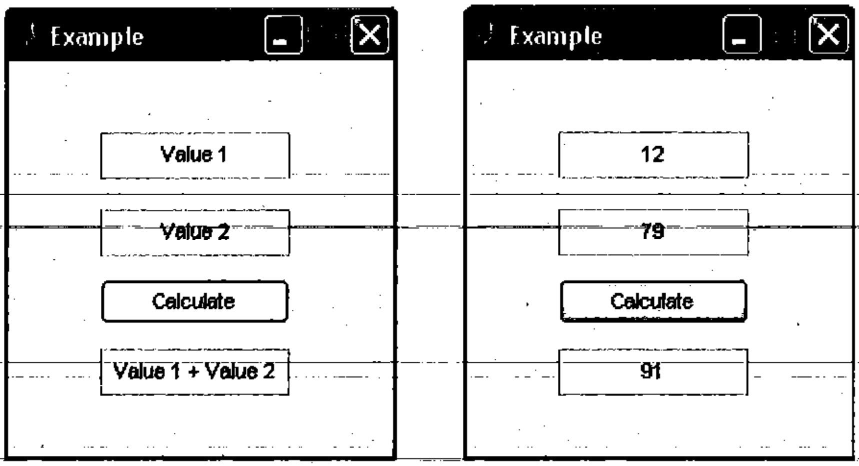


รูปที่ 2.21 สร้างโค้ดในฟังก์ชัน cal_Callback

คลิกแถบเครื่องมือชื่อ Run ตามรูป 2.22 เพื่อแสดงโปรแกรมที่สร้างขึ้น จะได้หน้าตาของโปรแกรม ดังรูปที่ 2.23



รูปที่ 2.22 กดรันโปรแกรม



(ก) (ข)

รูปที่ 2.23 ผลของโปรแกรม

(ก) เริ่มต้นโปรแกรม

(ข) ใส่ค่าตัวเลขแล้วทำการคำนวณ

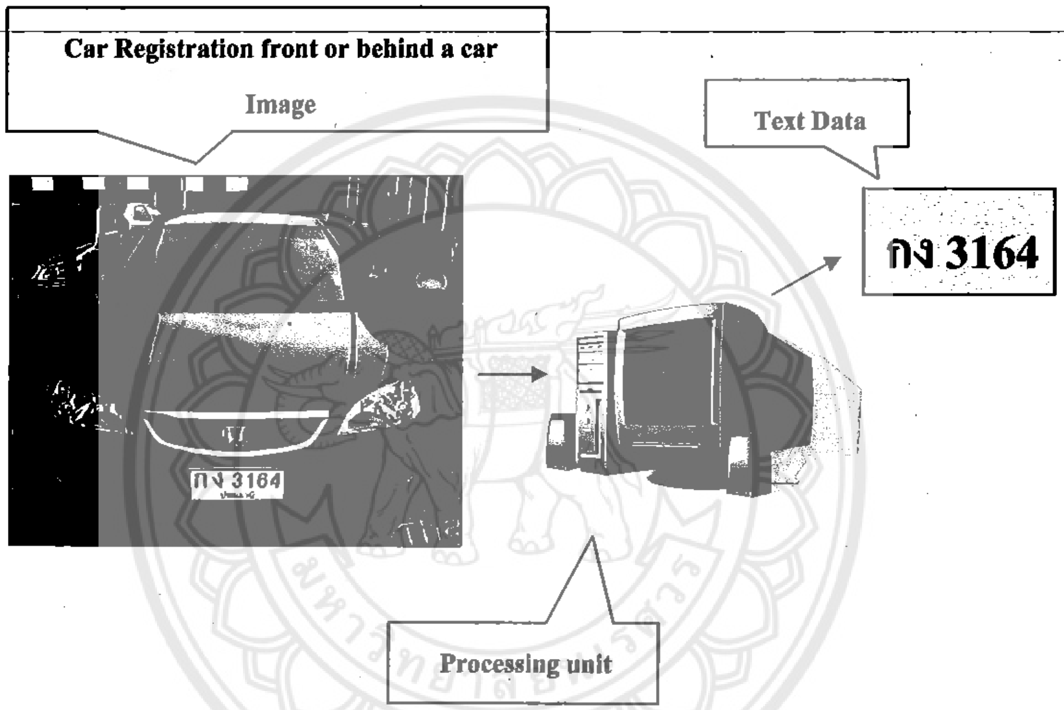
ขั้นตอนที่ได้กล่าวมา เป็นหลักการง่ายๆในการสร้าง GUI บน Matlab ซึ่งหลักการดังกล่าว เราจะได้นำมาพัฒนาต่อเพื่อให้สอดคล้องกับโปรแกรมตรวจจับหมายเลขทะเบียนรถ เพื่อให้ง่ายต่อการใช้งาน แทนที่จะต้องพิมพ์คำสั่งต่างๆ บน Matlab เราก็สามารถคลิกปุ่มต่างๆบน GUI ที่เราสร้างขึ้นมาแทนได้

บทที่ 3

วิธีการดำเนินงานโครงงาน

3.1 โครงสร้างของโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์จากภาพ

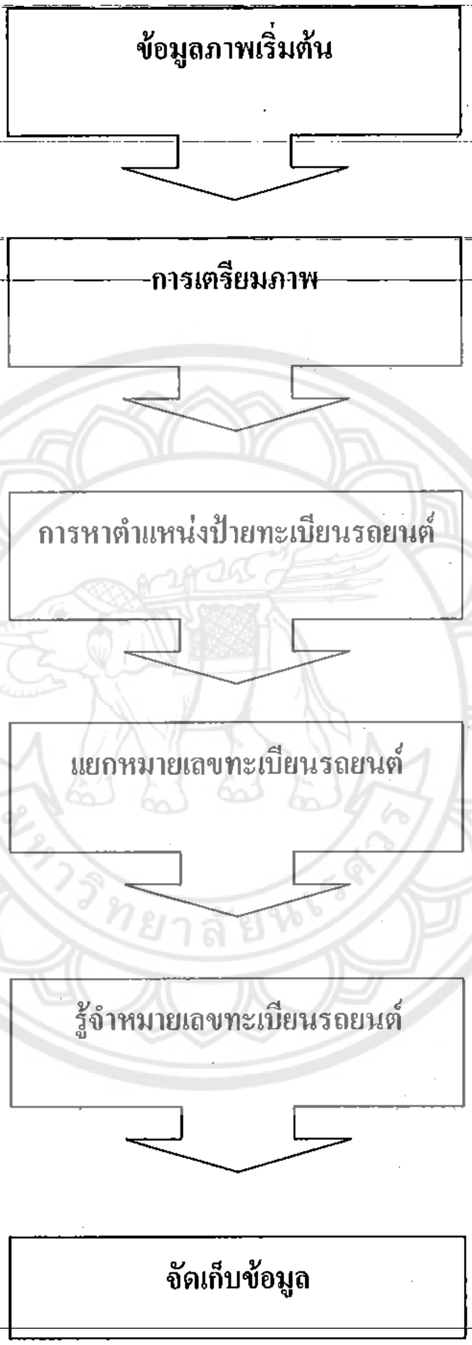
โปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์จากภาพ ประกอบด้วย ข้อมูลภาพถ่ายซึ่งอยู่ในลักษณะด้านหน้าหรือด้านหลังตรงของรถยนต์ และส่วนของโปรแกรมเพื่อทำการประมวลผลภาพ ให้ได้ผลลัพธ์เป็นข้อมูลตัวอักษร



รูปที่ 3.1 โครงสร้างของโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์จากภาพ

ในส่วนของโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์จากภาพ มีขั้นตอนการทำงานทั้งหมด ดังแสดงในรูปที่ 3.2 โดยมีขั้นตอนการทำงานของโปรแกรมดังนี้ คือ ขั้นแรกจะเป็นขั้นตอนการเตรียมภาพโดยภาพสีระดับความลึก 24 บิต จะถูกแปลงเป็นภาพระดับเทา 256 ระดับโดยจะพิจารณาเพียงระนาบสีแดง แล้วภาพจะถูกแปลงให้เป็นภาพแบบไบนารี ด้วยวิธีการเลือกค่าเทรชโฮล ซึ่งจะเหลือข้อมูลภาพเพียง 2 ระดับเท่านั้น คือ 0 และ 1 จากนั้นจะแยกวัตถุภายในภาพออกจากกัน ด้วยวิธีการ Region Labeling แล้วจึงพิจารณาแต่ละวัตถุว่าน่าจะเป็นป้ายทะเบียนหรือไม่ โดยอาศัยคุณสมบัติ ของ

ขนาด จากนั้นนำวัตถุภายในป้ายทะเบียน ซึ่งก็คือ เลขทะเบียนไปสู่กระบวนการรู้จำเพื่อระบุอักษรต่อไป

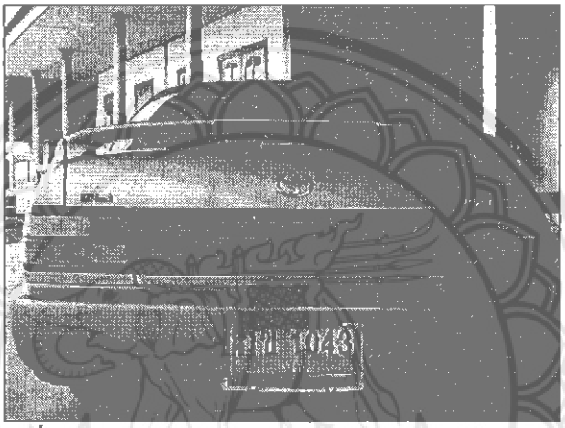


รูปที่ 3.2 ขั้นตอนการทำงานของโปรแกรม

3.2 การเตรียมภาพ

ขั้นตอนแรก จะเป็นการเตรียมข้อมูลภาพเพื่อใช้ในการประมวลผล เนื่องจากโปรแกรมยังไม่สามารถทำงานแบบเวลาจริงได้ ดังนั้นเพื่อความสะดวก จะใช้ข้อมูลภาพที่เก็บเป็นไฟล์ไว้แล้วแทนข้อมูลภาพจากกล้องดิจิตอล โดยจะใช้ภาพที่ถ่ายทางด้านหน้าหรือด้านหลังของรถยนต์ ข้อมูลภาพที่ได้จะถูกเก็บเป็นภาพระดับสี

เนื่องจากการประมวลผลภาพสีมีความซับซ้อนมาก จึงทำการแปลงภาพให้เป็นภาพระดับเทาในกรณีที่เป็นรถยนต์ ป้ายทะเบียนของรถบางคัน เป็นสีแดง ดังนั้นการแปลงภาพให้เป็นภาพระดับเทา จะพิจารณาเพียงระนาบสีแดงเท่านั้น

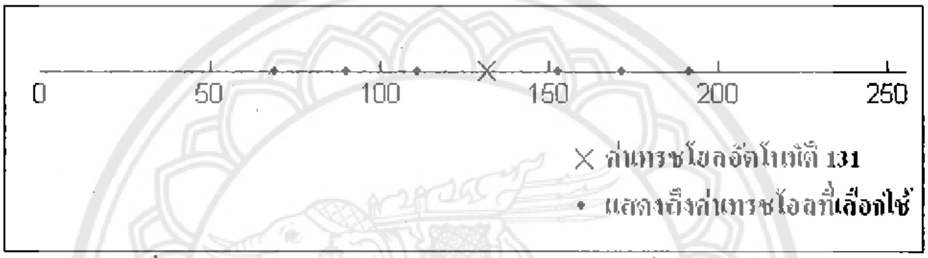


รูปที่ 3.3 ภาพระดับเทาตัวอย่างด้านหลังของรถยนต์

เมื่อได้ภาพระดับเทาแล้ว จะแปลงภาพให้เป็นภาพไบนารี ซึ่งจะเหลือข้อมูลภาพเพียงสองระดับ คือ 0 และ 1 โดย 0 จะเป็นพื้นหลัง ส่วน 1 จะเป็นวัตถุในภาพ แล้วจึงแยกวัตถุต่างๆออกจากกัน ซึ่งรวมไปถึงการแยกวัตถุที่เป็นทะเบียนรถยนต์ออกมาด้วย วิธีการแยกจะใช้การเทรชโฮล ด้วยวิธีเลือกค่าเทรชโฮลแบบอัตโนมัติผลที่ได้พื้นหลังของป้ายทะเบียนรถยนต์จะเป็นสีขาว และสีดำเป็นส่วนของหมายเลขทะเบียน

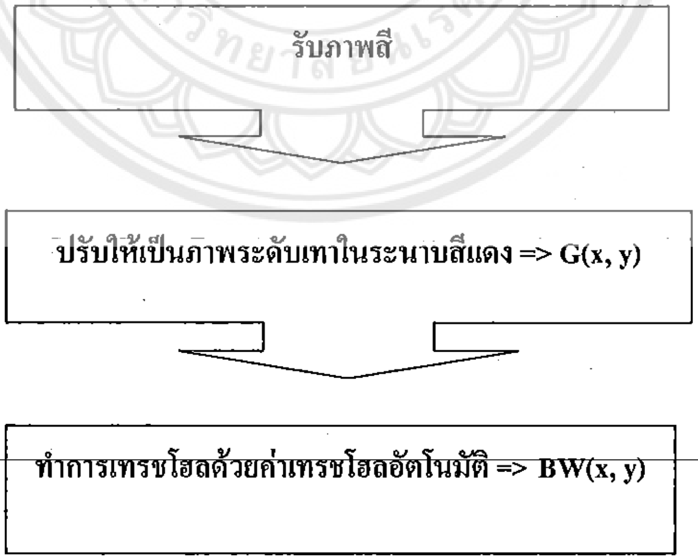
รูปที่ 3.4 ภาพรับเข้าที่ผ่านการเทรชโฮล

ในการทำงานจริงนั้น ผลที่ได้ยังมีบางภาพที่สว่างหรือมืดเกินไป ทำให้กรอบของป้ายทะเบียนหายไป หรือ กรอบของป้ายทะเบียนหนาเกินไป ทำให้กรอบและตัวอักษรติดกัน จึงไม่สามารถหาตำแหน่งของป้ายทะเบียนได้ ดังนั้นจึงได้ทำการเลือกค่าเทรชโฮลหลายๆค่า โดยเลือกค่าที่น้อยกว่าค่าที่คำนวณออกมาได้ไปเล็กน้อย จนไปถึงค่าที่มากกว่าค่าที่คำนวณออกมาไปเล็กน้อย ค่าเทรชโฮลที่ใช้แสดงดังรูป 3.5 ตัวอย่างเช่นในการทำงานของโปรแกรมจริงนั้น ถ้าเราหาค่าเทรชโฮลแบบอัตโนมัติได้เป็น 131 แล้วเราจะเลือกเผื่อไปอีก 3 ค่าที่มากกว่า และอีก 3 ค่าที่น้อยกว่าได้เป็น 193, 172, 153, 131, 112, 88, 69 โดยจะใช้ภาพที่เทรชโฮลด้วยค่า 193 ก่อนเพื่อนำเข้าสู่ขั้นตอนการหาตำแหน่งป้ายทะเบียน ถ้ายังไม่สามารถหาทะเบียนรถเจอ ก็จะเลือกใช้ค่าเทรชโฮลเท่ากับ 172 แต่ถ้ายังไม่เจออีก ก็จะลดค่าเทรชโฮลลงไปตามค่าที่ได้กำหนดไว้



รูปที่ 3.5 ตัวอย่างการเลือกค่าเทรชโฮลหลายค่าเพื่อการวิเคราะห์

ขั้นตอนการทำงานในส่วนการเตรียมภาพ แสดงแผนผังลำดับงาน ดังรูปที่ 3.6 คือ



รูปที่ 3.6 ขั้นตอนการเตรียมภาพ

3.3 การหาตำแหน่งป้ายทะเบียนรถยนต์

การหาตำแหน่งป้ายทะเบียนนี้สามารถทำได้หลายวิธี ตามที่ได้ศึกษามา ได้แก่ การหาความหนาแน่นของขอบชันตามแนวแกน x แกน y หรือ การพิจารณาวัตถุภายในภาพว่าแต่ละวัตถุมีลักษณะเป็นสี่เหลี่ยมผืนผ้าหรือไม่ ซึ่งวิธีนี้จะถูกนำมาใช้ในโครงงานนี้ เป็นต้น

การแยกวัตถุต่างๆภายในภาพออกจากกัน สามารถทำได้โดยใช้กระบวนการ Region Labeling กับภาพแบบไบนารีหรือภาพที่ได้จากการเทรชโฮล โดยพิกเซลที่เป็นวัตถุเดียวกันจะมีค่า Label เท่ากัน แล้วทำการหาตำแหน่งของแต่ละวัตถุด้วยจุดมุม 2 จุดคือ มุมซ้ายบนของวัตถุซึ่งเป็นตำแหน่งที่น้อยที่สุดของแถวและหลัก อีกมุมหนึ่งคือ มุมขวาล่างของวัตถุซึ่งเป็นตำแหน่งที่มากที่สุดของแถวและหลัก

กำหนดให้

minRow	คือ ตำแหน่งที่น้อยที่สุดของแถว
minCol	คือ ตำแหน่งที่น้อยที่สุดของหลัก
maxRow	คือ ตำแหน่งที่มากที่สุดของแถว
maxCol	คือ ตำแหน่งที่มากที่สุดของหลัก

จะได้

$$\text{height} = \text{maxRow} - \text{minRow} \quad (3.1)$$

$$\text{width} = \text{maxCol} - \text{minCol} \quad (3.2)$$

ทำการหาวัตถุที่น่าจะเป็นป้ายทะเบียนโดยประมาณ ด้วยการตรวจสอบว่ามีลักษณะคล้ายสี่เหลี่ยมผืนผ้าหรือไม่นั่นคือ ต้องมีความกว้าง(width) มากกว่าส่วนสูง(height) แต่ในการใช้งานจริงจะต้องให้มีความกว้างมากกว่าส่วนสูงอยู่พอประมาณ จึงได้กำหนดอัตราส่วนความกว้างให้มากกว่าส่วนสูงเป็น 1.5: 1 ไว้ที่ และขนาดป้ายทะเบียน (height x width) จะต้องไม่น้อยกว่า 2% ของขนาดภาพ ดังสมการ 3.3

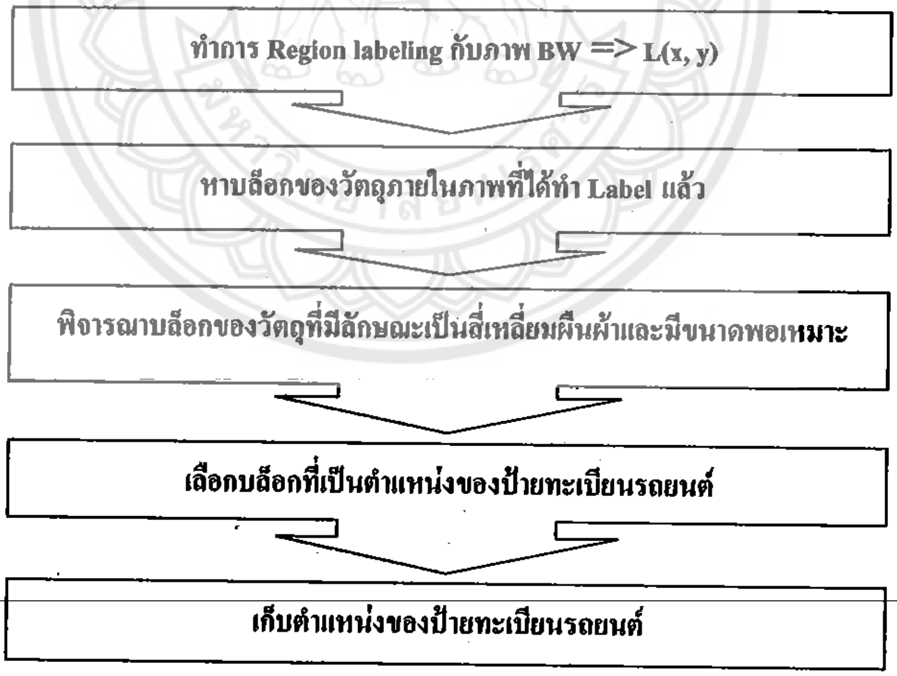
$$1.5 \times \text{height} < \text{width} \text{ และ } \text{height} \times \text{width} > 0.02 \times \text{image size} \quad (3.3)$$

วัตถุที่ผ่านเงื่อนไขดังกล่าวจะได้รับการตรวจสอบอย่างละเอียดอีกทีโดยจะเรียกวัตถุเหล่านี้ว่า บล็อก การตรวจสอบอย่างละเอียดว่าบล็อกใดเป็นป้ายทะเบียนรถอย่างแน่นอนนั้น จะพิจารณาที่บล็อก โดยจะมีวัตถุย่อยๆเรียงกันตามแนวนอน 3 วัตถุขึ้นไป ซึ่งวัตถุนี้ได้คาดการณ์ไว้ว่าน่าจะเป็นตัวอักษรที่อยู่บนป้ายทะเบียนรถ การตรวจสอบการเรียงกันของวัตถุ ทำได้โดยการคำนวณหาค่าจุดศูนย์กลางมวลของแต่ละวัตถุว่าอยู่ในแนวแกนนอนเดียวกันหรือไม่ ถ้าวัตถุนั้นจะเป็นตัวอักษรจะต้องมีความสูงมากกว่าความกว้าง และ มีขนาดมากกว่า 1% ของขนาดป้ายทะเบียน และ ไม่ใหญ่เกินกว่า 15% ของขนาดป้ายทะเบียน

รูปที่ 3.7 ตัวอย่างภาพที่ทำการกำหนดบล็อกให้กับพิกเซลที่มี label เหมือนกัน

เมื่อยืนยันได้แน่นอนแล้วว่า บล็อกใดเป็นตำแหน่งของป้ายทะเบียนรถ จะทำการเก็บตำแหน่ง minRow, minCol, maxRow และ maxCol เพื่อนำไปใช้ในการแยกเลขทะเบียนออกจากป้ายทะเบียนรถในขั้นต่อไป

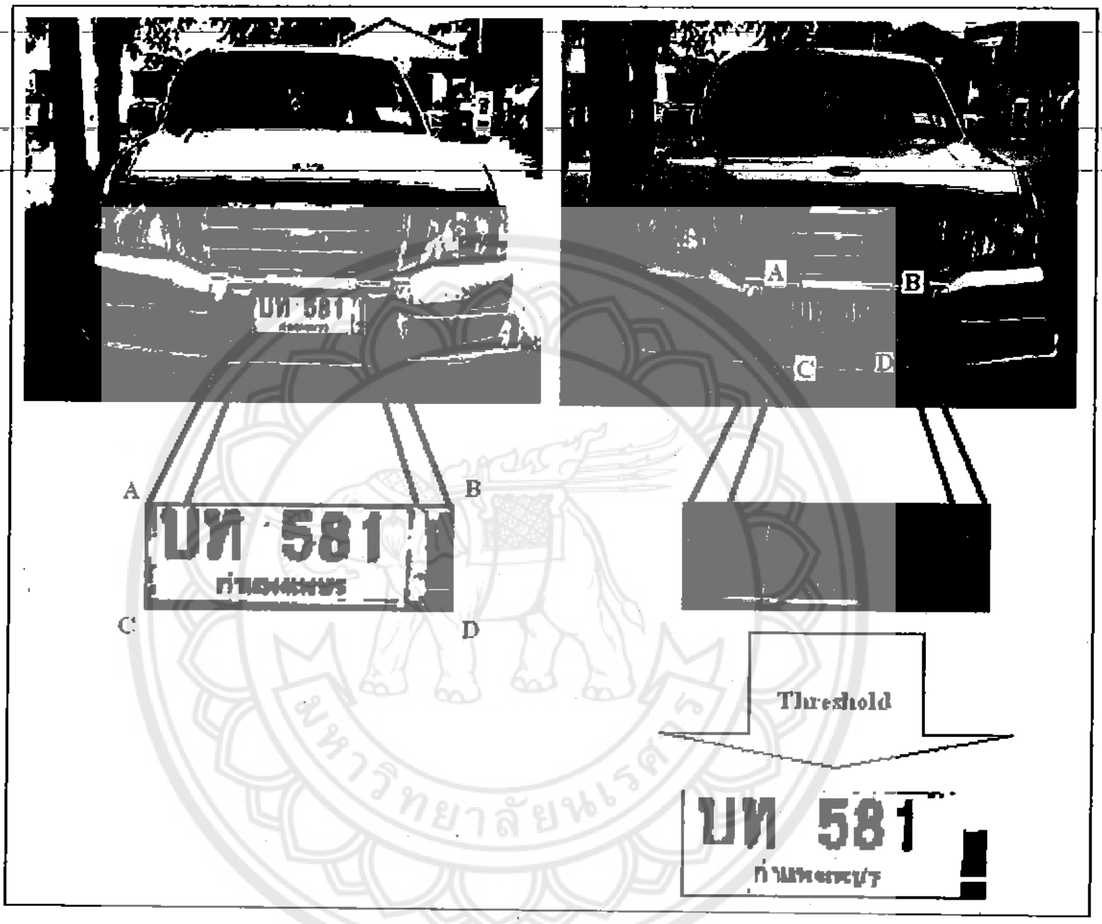
ขั้นตอนการทำงานในส่วนการหาตำแหน่งป้ายทะเบียนรถยนต์ แสดงแผนผังลำดับงาน ดังรูปที่ 3.8 คือ



รูปที่ 3.8 ขั้นตอนการหาตำแหน่งป้ายทะเบียนรถยนต์

3.4 การแยกหมายเลขทะเบียนออกจากแผ่นป้ายทะเบียนรถยนต์

เนื่องจากบางครั้งการทำเทรชโฮล กับภาพระดับเทาทั้งภาพ ผลที่ออกมายังไม่ดีมากเท่าที่ควร จะส่งผลกระทบต่อขั้นตอนการรู้จำเป็นอย่างมาก ผลของหมายเลขทะเบียนที่ผิดเพี้ยนไป เช่น ตัวอักษรยังติดกับกรอบป้ายทะเบียน ตัวอักษรที่ได้บางหรือหนาเกินไป ดังนั้นเราจึงเลือกทำเทรชโฮล ในส่วนที่เป็นตำแหน่งที่เป็นป้ายทะเบียนเท่านั้น ซึ่งตำแหน่งของป้ายทะเบียนนี้ได้จาก ผลลัพธ์ในหัวข้อ 3.3



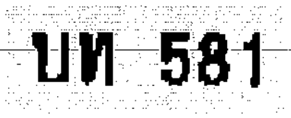
รูปที่ 3.9 การเทรชโฮล ภาพที่ตำแหน่งป้ายทะเบียนรถยนต์

หลังจากนั้น จะทำ Region labeling กับภาพทะเบียนที่ได้ เพื่อทำการแยกทะเบียนโดยตัวอักษรที่เป็นเลขทะเบียนจะต้องมีคุณสมบัติดังนี้

1. ความสูงมากกว่าความกว้าง
2. ขนาดมากกว่า 1% ของขนาดป้ายทะเบียน
3. ขนาดไม่ใหญ่เกินกว่า 15% ของขนาดป้ายทะเบียน

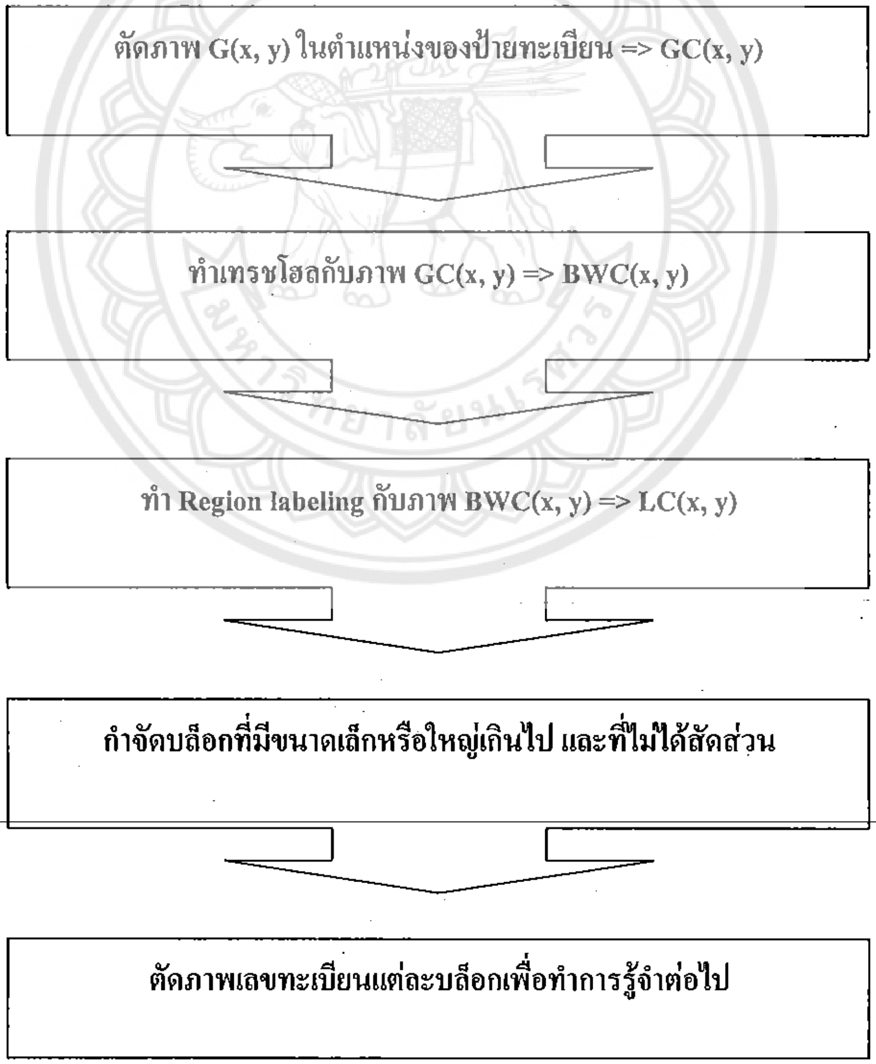
ในป้ายทะเบียนจะมีบล็อกของหมายเลขทะเบียนอยู่ภายในซึ่งระยะห่างของแต่ละบล็อกที่อยู่ติดกันมีค่าไม่เท่ากัน โดยบล็อกของหมายเลขทะเบียนจะมีอยู่ 2 ประเภท นั่นคือ บล็อกของตัวอักษร

และบล็อกของตัวเลข เนื่องจากระยะห่างระหว่างบล็อกของตัวอักษรตัวสุดท้ายกับบล็อกของตัวเลขตัวแรกนั้น มีระยะห่างมากกว่าระยะห่างของแต่ละบล็อก จะเห็นได้จากรูป 3.10 บล็อกที่ 2 และบล็อกที่ 3 จะมีระยะห่างที่มากที่สุด ทำให้โปรแกรมรู้ว่า 2 บล็อกแรกเป็นตัวอักษร และ 3 บล็อกต่อมาเป็นบล็อกของตัวเลข สุดท้ายแล้วจึงนำแต่ละบล็อกเข้าสู่กระบวนการรู้จำเพื่อระบุอักษรและตัวเลขต่อไป



รูปที่ 3.10 ผลลัพธ์จากการแยกเลขทะเบียน

ขั้นตอนการทำงานในส่วนการหาแยกหมายเลขทะเบียนออกจากป้ายทะเบียนรถยนต์ แสดงแผนผังลำดับงาน ดังรูปที่ 3.11 คือ



รูปที่ 3.11 ขั้นตอนการแยกหมายเลขทะเบียนออกจากแผ่นป้ายทะเบียนรถยนต์

3.5 การรู้จำหมายเลขทะเบียนรถยนต์

การรู้จำรูปแบบต่างๆมีบทบาทมากในปัจจุบัน สังเกตได้จาก การมีหุ่นยนต์ที่จดจำใบหน้ามนุษย์ได้ การสแกนลายนิ้วมือในที่ต่างๆ การเปรียบเทียบรูปแบบของหัวกระสุนที่ถูกยิงออกจากลำกล้องคนร้ายกับปืนที่ต้องสงสัย และ หนึ่งในนั้นก็คือ การรู้จำตัวอักษร ซึ่งจะมีความสำคัญมากกับโครงงานนี้ โดยการรู้จำนั้นก็มีหลายวิธี สามารถเลือกใช้ได้ตามความเหมาะสมของเนื้องาน เช่น การใช้โครงข่ายประสาทเทียมในการวิเคราะห์รูปแบบสภาพอากาศ หรือ การใช้ Cross-Correlation ในการเทียบรอยลึกของการบิดเกลียวบนลูกกระสุนปืน หรือ การใช้ Thinning กับลายนิ้วมือ เพื่อหาจุดปลาย หรือจุดแยกของลายเส้น เป็นต้น ซึ่งการใช้ Thinning จะนำมาช่วยในกระบวนการรู้จำของตัวอักษรด้วย

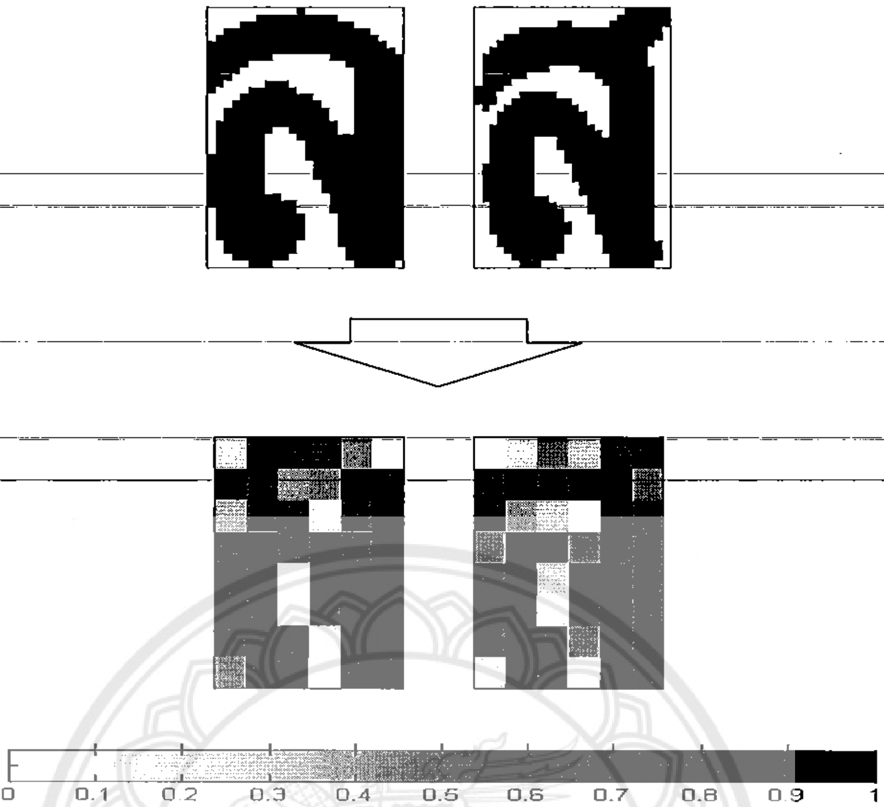
3.5.1 การเตรียมต้นแบบ

การรู้จำหมายเลขทะเบียนรถในโครงงานนี้จำเป็นที่จะต้องมีภาพต้นแบบไว้เพื่อวิเคราะห์กับภาพที่รับเข้ามา ซึ่งการจะนำภาพต้นแบบมาใช้ จะต้องมีการแปลงภาพให้เหมาะสมกับกระบวนการที่นำไปใช้งาน ในโครงงานนี้ได้ใช้วิธีเทียบต้นแบบอยู่ 2 วิธี คือ การเตรียมต้นแบบด้วยการแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ และ การเตรียมต้นแบบด้วยการหาจุดปลายของตัวอักษร

3.5.1.1 การเตรียมต้นแบบด้วยการแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ

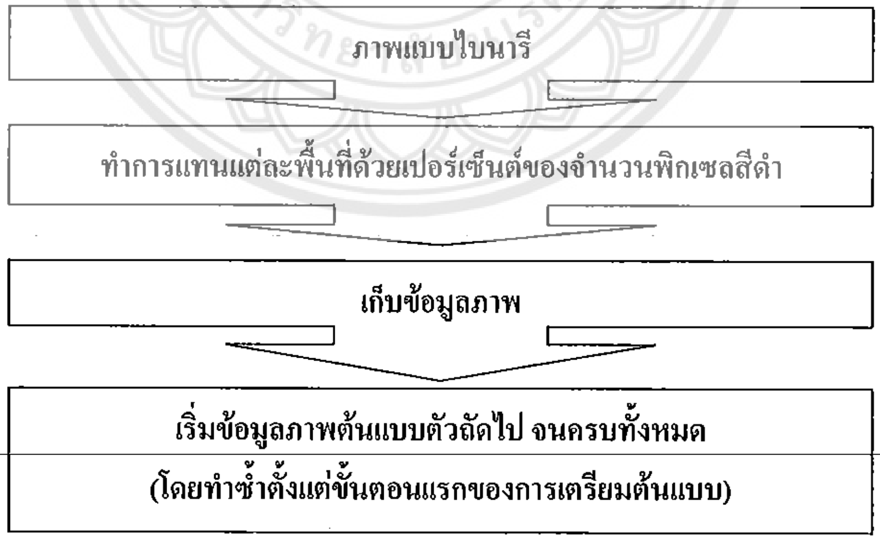
การแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำนี้ มีพื้นฐานมาจากส่วนหนึ่งของโครงข่ายประสาทเทียม เป็นเพียงส่วนของการเตรียมข้อมูลก่อนที่จะนำเข้าสู่โครงข่ายประสาทแล้วจะได้ค่าความผิดพลาดออกมา เนื่องจากมีกระบวนการหาจุดปลายของตัวอักษรเข้ามาช่วย เราจึงไม่จำเป็นต้องสร้างโครงข่ายประสาทเทียมเพื่อมาช่วยในการรู้จำนี้ แต่ใช้การหาค่าความผิดพลาดโดยประมาณโดยการเอาภาพมาลบกันเท่านั้น

การเตรียมต้นแบบนั้นจะเริ่มจากการนำข้อมูลภาพตัวอักษรต้นแบบ ซึ่งเป็นภาพแบบไบนารีมาแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ โดยบล็อกของหมายเลขทะเบียนจะถูกแบ่งให้มีขนาด m x n บล็อกโดย m คือ จำนวนแถว และ n คือ จำนวนหลัก ซึ่งในโครงงานนี้จะแบ่งขนาดเป็น 8x6 บล็อก ในบล็อกที่มีพื้นที่เป็นสีดำล้วนจะมีค่าเป็น 1 และ บล็อกที่มีพื้นที่เป็นสีขาวล้วนจะมีค่าเป็น 0 ถ้าในบล็อกที่มีพื้นที่เป็นสีดำหรือสีขาวอย่างละครึ่งจะมีค่าเป็น 0.5 แสดงดังรูป 3.12



รูปที่ 3.12 การแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ

ขั้นตอนการเตรียมต้นแบบด้วยการแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ ซึ่งอยู่ในส่วนการรู้จำหมายเลขทะเบียนรถยนต์ แสดงแผนผังลำดับงาน ดังรูปที่ 3.13 คือ



รูปที่ 3.13 ขั้นตอนการเตรียมต้นแบบด้วยการแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ

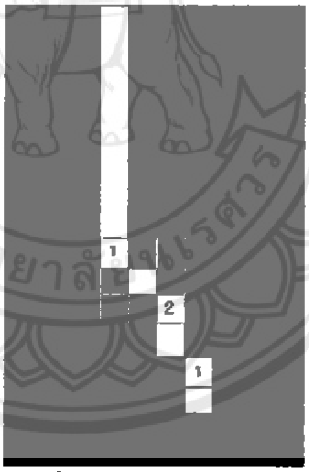
3.5.1.2 การเตรียมต้นแบบด้วยการหาจุดปลายของตัวอักษร

การหาจุดปลายของวัตถุภายในภาพ มีบทบาทสำคัญทั้งในทางการแพทย์ ด้านอาชญากรรม รวมทั้งด้านอื่นๆ เป็นอย่างมาก เช่น การวิเคราะห์การเติบโตของเซลล์ประสาท การตรวจสอบลายนิ้วมือ เป็นต้น ซึ่งกระบวนการภายในไม่จำเป็นจะต้องเป็นเพียงจุดปลายเท่านั้น อาจจะรวมไปถึง จุดที่แตกออกเป็นสองทาง หรือ สามทางขึ้นไป อาจจะนำไปใช้วิเคราะห์ ทางด้านการจราจรว่าเส้นทางไหนจะมีรถติดน้อยโดยคำนวณว่าเส้นทางนั้นผ่านแยกไหนบ้าง จำนวนแยกมีเท่าไร ดังนั้นเราจึงไม่มองข้ามความสำคัญของกระบวนการนี้ ซึ่งเราได้นำมาช่วยในการหาจุดปลายของตัวอักษรในขั้นตอนการรู้จำได้เป็นอย่างดี

การเตรียมต้นแบบนี้ เริ่มต้นด้วยการนำข้อมูลภาพตัวอักษรต้นแบบ มาผ่านกระบวนการ Thinning Morphological แบบไม่สิ้นสุด แล้วนำภาพที่ได้มาหาจุดปลาย โดยพิกเซลที่จะเป็นจุดปลายจะต้องมีคุณสมบัติดังนี้

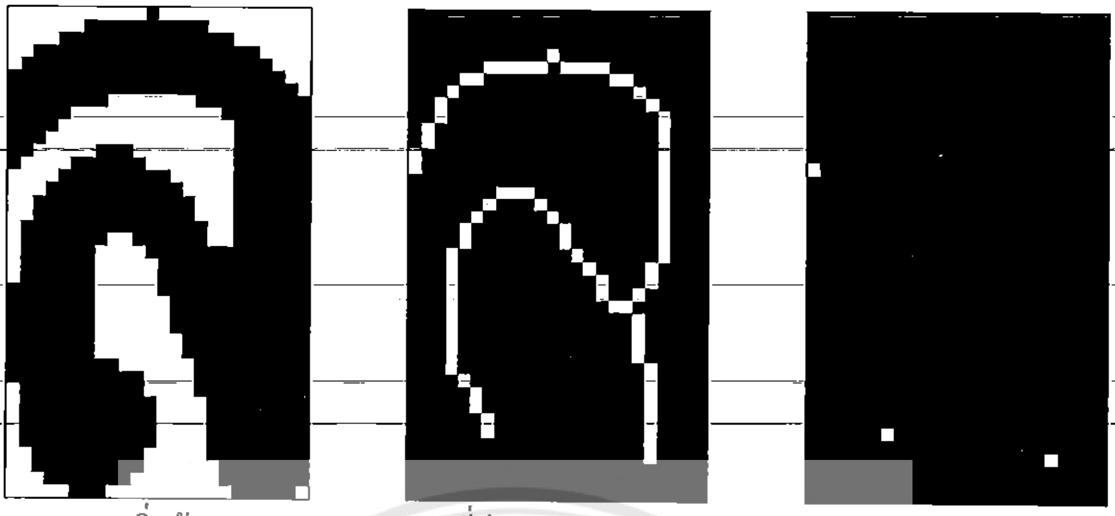
- พิกเซล นั้นจะต้องมีค่าเป็น 1 (สีขาว)
- 8- neighborhood ของพิกเซลนั้นจะมีค่าเป็น 1 เพียงพิกเซลเดียว

ดังรูปที่ 3.14 ภายในกรอบสีแดงแสดงถึงจำนวนเพื่อนบ้านเพียง 1 พิกเซล และภายในกรอบสีเหลืองแสดงถึงจำนวนเพื่อนบ้านถึง 2 พิกเซล ดังนั้นจุดปลายของเส้นอยู่ที่กึ่งกลางของกรอบสีแดง



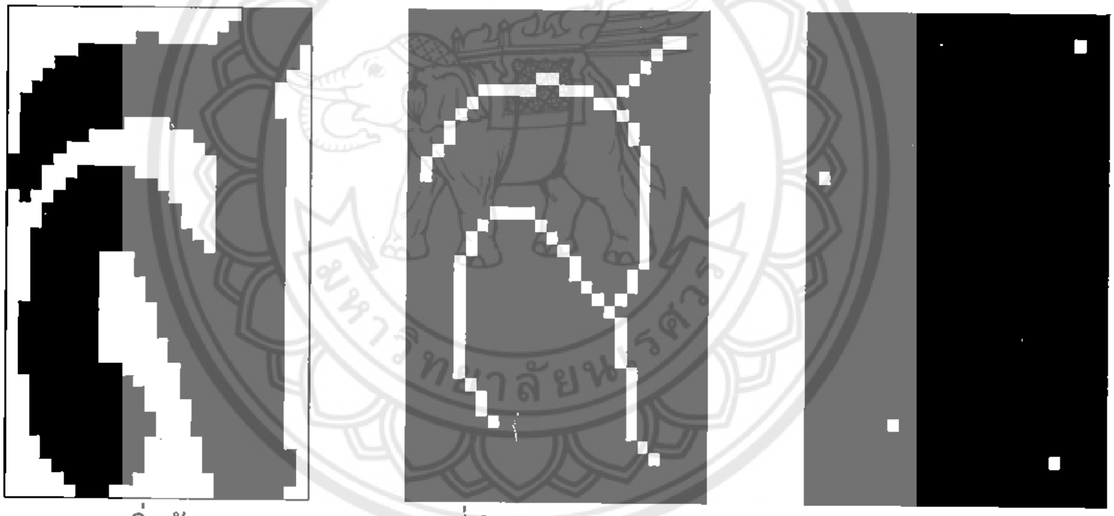
รูปที่ 3.14 เทียบพิกเซลที่เป็นจุดปลายกับพิกเซลที่ไม่ใช่จุดปลาย

เมื่อเราได้หาจุดปลายของตัวอักษรทุกจุดจนครบ จะสังเกตได้ว่า ลักษณะตัวอักษรที่มีความคล้ายคลึงกัน เช่น ล-ส , ค-ศ เป็นต้น จะมีจำนวนจุดปลายไม่เท่ากัน ทำให้สามารถแยกตัวอักษรที่มีความคล้ายคลึงกันออกจากกันได้ โดยการคำนวณค่าความผิดพลาดของแต่ละตัวอักษร จะได้กล่าวถึงในหัวข้อ 3.5.2.2



ภาพเริ่มต้น “ล” “ล” ที่ผ่านการ thinning จุดปลายของ “ล”

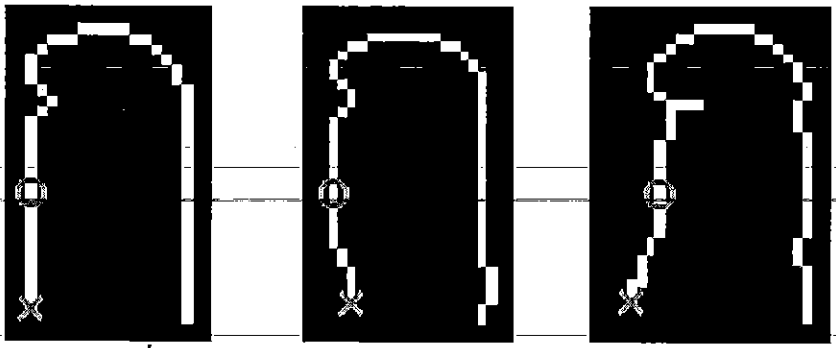
รูปที่ 3.15 จุดปลายของตัวอักษรจากภาพต้นแบบ “ล”



ภาพเริ่มต้น “ส” “ส” ที่ผ่านการ thinning จุดปลายของ “ส”

รูปที่ 3.16 จุดปลายของตัวอักษรจากภาพต้นแบบ “ส”

เนื่องจากอักษรภาษาไทยบางตัว มีความคล้ายคลึงกันมากจนบางครั้งการใช้วิธีหาจุดปลายไม่สามารถเข้ามาช่วยวิเคราะห์ได้อย่างละเอียด ดังนั้นในโครงงานนี้จะเพิ่มวิธีการตรวจสอบหัวของตัวอักษรว่าเป็นแบบหัวเข้า หัวออก หรือ ไม่มีหัว ซึ่งวิธีนี้จะใช้เฉพาะกับตัวอักษร ก, ถ, ภ เท่านั้น โดยจะเพิ่มการหาแกนกลางของหัวตัวอักษรเข้าไปช่วยในการวิเคราะห์ที่ดียิ่งขึ้น



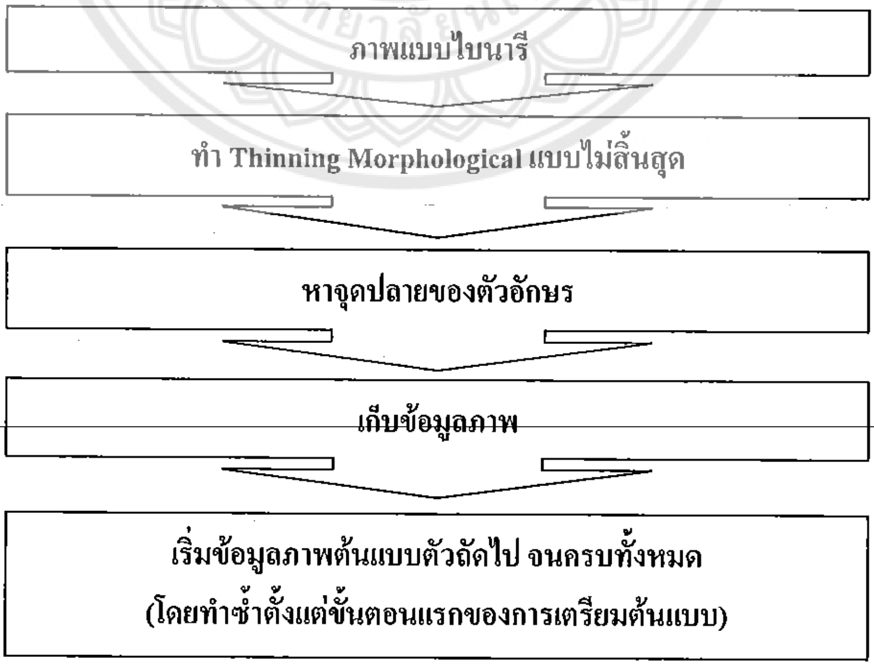
รูปที่ 3.17 จุดของหัวตัวอักษรเทียบกับแกนกลางของหัว

วิธีในการหาแกนกลางของหัวตัวอักษร ทำได้โดยแบ่งภาพเป็นครึ่งหนึ่งตามแนวแกนตั้ง และแบ่งเป็น 3 ส่วนตามแกนนอน แล้วเลือกส่วนที่อยู่ทางซ้ายตรงกลางมาพิจารณา ต่อจากนั้นทำการหาค่าเฉลี่ยของตำแหน่งแถวที่มีค่าเป็น 1 (สีขาว) จะได้ค่าที่บ่งบอกถึงแกนกลาง (จุด O ในรูป 3.17) แล้วนำไปเปรียบเทียบกับจุดที่เป็นหัวตัวอักษร (จุด X ในรูป 3.17)

เมื่อพิจารณาจากซ้ายไปขวา

- ถ้าเป็นหัวเข้า จะเจอจุด O ก่อนจุด X นั่นคือ "ถ"
- ถ้าเป็นหัวออก จะเจอจุด X ก่อนจุด O นั่นคือ "ภ"
- ถ้าเจอจุด O และจุด X พร้อมกันจะถือว่าไม่มีหัว นั่นคือ "ก"

ขั้นตอนการเตรียมต้นแบบด้วยการหาจุดปลายของตัวอักษร ซึ่งอยู่ในส่วนการรู้จำหมายเลขทะเบียนรถยนต์ แสดงแผนผังลำดับงาน ดังรูปที่ 3.18 คือ



รูปที่ 3.18 ขั้นตอนการเตรียมต้นแบบด้วยการหาจุดปลายของตัวอักษร

3.5.2 การเปรียบเทียบข้อมูลรับเข้ากับข้อมูลต้นแบบ

เมื่อได้ต้นแบบทั้ง 2 แบบแล้ว จะนำมาใช้เปรียบเทียบหาค่าความผิดพลาดกับข้อมูลภาพรับเข้า แต่เนื่องจากภาพรับเข้าที่นำมาหาจุดปลายนั้นมีขนาดไม่เท่ากันกับภาพต้นแบบ จึงต้องปรับขนาดภาพให้เท่ากันเสียก่อน แล้วจึงนำเข้าสู่กระบวนการ แปลงเช่นเดียวกับข้อมูลต้นแบบทั้ง 2 แบบ หลังจากนั้นจึงนำเข้าสู่กระบวนการเปรียบเทียบหาค่าความผิดพลาด เพื่อให้ผลลัพธ์ที่ได้นั้นมีความถูกต้องมากที่สุด การเปรียบเทียบกับต้นแบบที่ 1 และแบบที่ 2 นั้น จะมีความแตกต่างกัน จะอธิบายดังต่อไปนี้

กำหนดให้

P1(x, y) คือ ต้นแบบที่แทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำมีขนาด 6 x 8

P2(x, y) คือ ต้นแบบที่ได้จากการหาจุดปลาย

I1 คือ ข้อมูลภาพรับเข้าที่ผ่านกระบวนการหาเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ

I2 คือ ข้อมูลภาพรับเข้าที่ผ่านกระบวนการหาจุดปลาย

3.5.2.1 การหาค่าความผิดพลาด แบบแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ

การหาค่าความผิดพลาด ด้วยวิธีนี้ ก็คือการนำข้อมูลต้นแบบและข้อมูลภาพรับเข้าที่ทำการแปลงแล้วมาลบกัน ต่อจากนั้นนำค่าที่ได้มาใส่ค่าสัมบูรณ์ แล้วนำค่าในแต่ละพิกเซลมารวมกัน จะได้เป็นค่าตัวเลขทศนิยมตัวเดียว ดังจะสามารถอธิบายด้วยสมการ 3.4

$$\text{ค่าผิดพลาด (E1)} = \sum_{x=0}^{M} \sum_{y=0}^{N} |\mathrm{P1}(x,y) - \mathrm{I1}(x,y)| \qquad (3.4)$$

3.5.2.2 การหาค่าความผิดพลาด แบบการหาจุดปลาย

การหาค่าความผิดพลาด ด้วยวิธีนี้จะเริ่มด้วยการหาระยะห่างระหว่างจุดปลายจุดแรกของภาพต้นแบบ กับ ทุกจุดปลายของภาพรับเข้า แล้วเปรียบเทียบว่าค่าไหนน้อยที่สุดให้เป็นค่าความผิดพลาด ที่ 1 ต่อจากนั้นหาระยะห่างระหว่างจุดปลายจุดที่ 2 ของภาพต้นแบบกับทุกจุดปลายของภาพรับเข้า แล้วเปรียบเทียบว่าค่าไหนน้อยที่สุด ให้เป็นค่าความผิดพลาด ที่ 2 ทำเช่นนี้จนครบทุกจุดปลายของภาพต้นแบบ ถ้าทำเพียงเท่านี้จะเป็นการหาค่าความผิดพลาด เพียงมุมมองเดียว จึงต้องทำการหาค่าความผิดพลาด โดยใช้ภาพรับเข้าเทียบกับต้นแบบด้วยเพื่อความถูกต้อง ซึ่งการเทียบค่าความผิดพลาด ก็ทำได้เช่นเดียวกับการใช้ต้นแบบเทียบกับภาพรับเข้า แต่เปลี่ยนเพียง ตัวตั้งการวิเคราะห์เป็นภาพรับเข้าแทน จะทำให้ได้ค่าความผิดพลาด มาจำนวนเท่ากับผลรวมของจำนวนจุดปลายของภาพต้นแบบ กับจำนวนจุดปลายของภาพรับเข้า สุดท้ายค่าความผิดพลาดที่สมบูรณ์จะได้จากผลรวมของค่าความผิดพลาดทั้งหมดหารด้วยขนาดของภาพ

กำหนดให้

M = จำนวนจุดปลายใน P2

N = จำนวนจุดปลายใน I2

k = 1, 2, 3, …, M

l = 1, 2, 3, …, N

Xp(k) = ตำแหน่งในแนวแกนนอนของจุดปลายตัวที่ k ใน P2

Yp(k) = ตำแหน่งในแนวแกนตั้งของจุดปลายตัวที่ k ใน P2

Xi(l) = ตำแหน่งในแนวแกนนอนของจุดปลายตัวที่ l ใน I2

Yi(l) = ตำแหน่งในแนวแกนตั้งของจุดปลายตัวที่ l ใน I2

D = Matrix ขนาด M x N ที่ทำการหาค่าระยะห่างในแต่ละจุดปลาย

$$D(k, l) = \sqrt{(Xp(k) - Xi(l))^2 + (Yp(k) - Yi(l))^2} \quad (3.5)$$

โดย

Dr(k) = Matrix ที่ ทำการหาค่า minimum ของ D(k, l) ตามแนวแถว

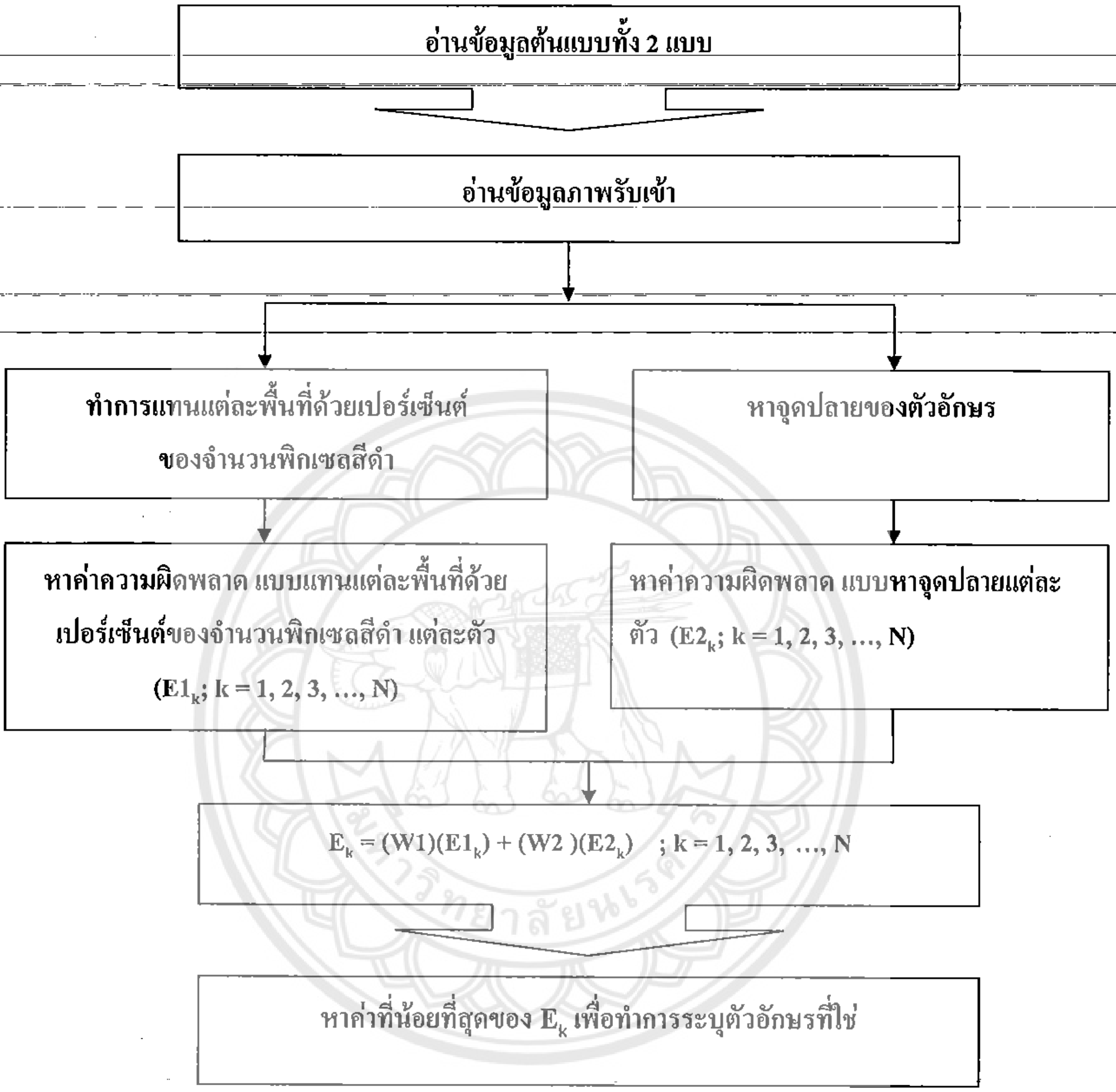
Dc(l) = Matrix ที่ ทำการหาค่า minimum ของ D(k, l) ตามแนวหลัก

$$\text{ค่าผิดพลาด (E2)} = \frac{\sum_{k=1}^{M} Dr(k) + \sum_{l=1}^{N} Dc(l)}{\text{Image size}} \quad (3.6)$$

เมื่อได้ค่าความผิดพลาด จากทั้ง 2 วิธีแล้ว เราจะนำมารวมกันเลยไม่ได้ เพราะค่าความผิดพลาด ทั้ง 2 มีค่าน้ำหนักไม่เท่ากัน จึงจำเป็นต้องทำการถ่วงน้ำหนักให้ค่าความผิดพลาด ทั้ง 2 มีค่าที่สมดุลกัน โดยในโครงงานนี้จะถ่วงน้ำหนักให้ค่าความผิดพลาด ที่ได้จากวิธีที่ 2 เป็น 20 เท่า ค่าน้ำหนักนี้ได้มาจาก ผลการทดลอง โดยหาค่าเฉลี่ยของค่าความผิดพลาดแต่ละตัวโดยประมาณ ดังสมการ 3.7

$$E = (W1)(E1) + (W2)(E2) \quad ; w1 = 1/20 , w2 = 1 \quad (3.7)$$

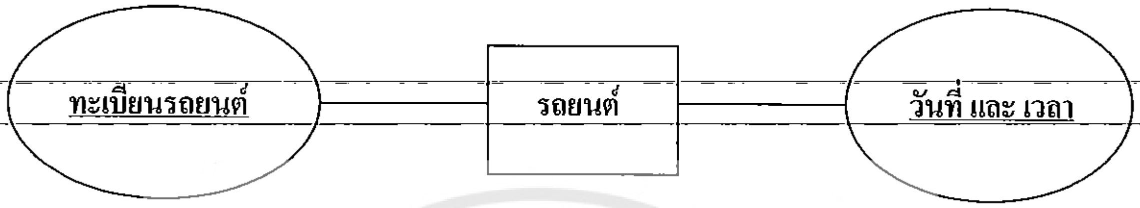
ขั้นตอนการทำงานในส่วนการรู้จำหมายเลขทะเบียนรถยนต์ แสดงแผนผังลำดับงาน ดังรูปที่ 3.19 คือ



รูปที่ 3.19 ขั้นตอนการรู้จำหมายเลขทะเบียนรถยนต์

3.6 การจัดเก็บข้อมูล

เมื่อทราบเลขทะเบียนรถแล้ว ก็จะนำข้อมูลที่ได้จัดเก็บลงฐานข้อมูล เนื่องจากข้อมูลที่ได้จากภาพนิ่งเมื่อผ่านกระบวนการรู้จำมีเพียงแค่เลขทะเบียนรถ และวันเวลาที่ถ่ายภาพ ทำให้ฐานข้อมูลไม่ซับซ้อนมากนัก ง่ายต่อการสืบค้น ดังนั้นเราจึงเขียน E/R diagram ได้ดังรูป 3.20 ซึ่งโครงงานนี้ใช้ฐานข้อมูลชนิด Microsoft Access

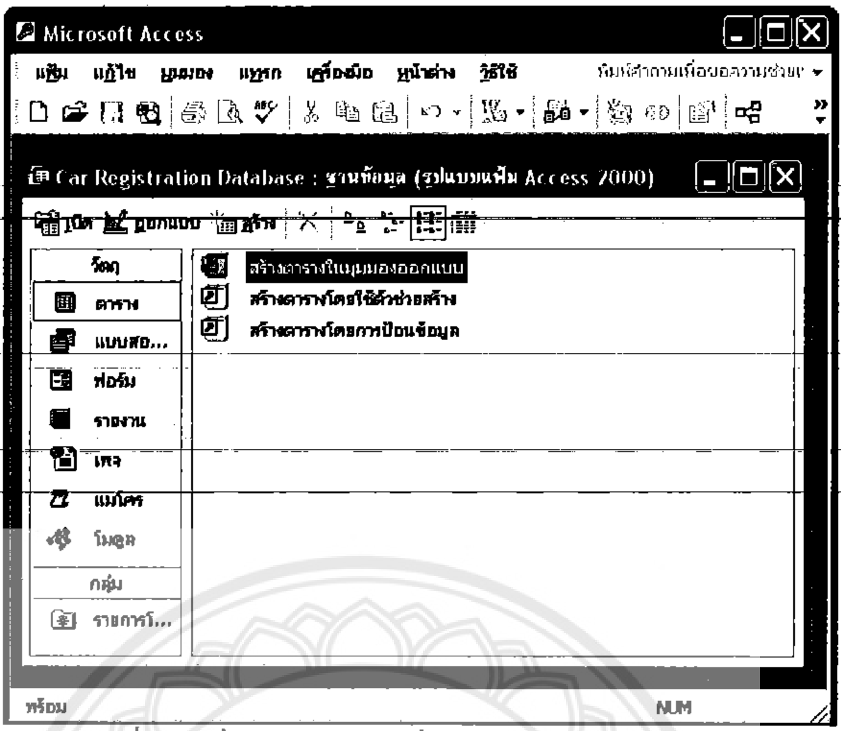


รูปที่ 3.20 E/R diagram ของข้อมูลที่จะจัดเก็บ

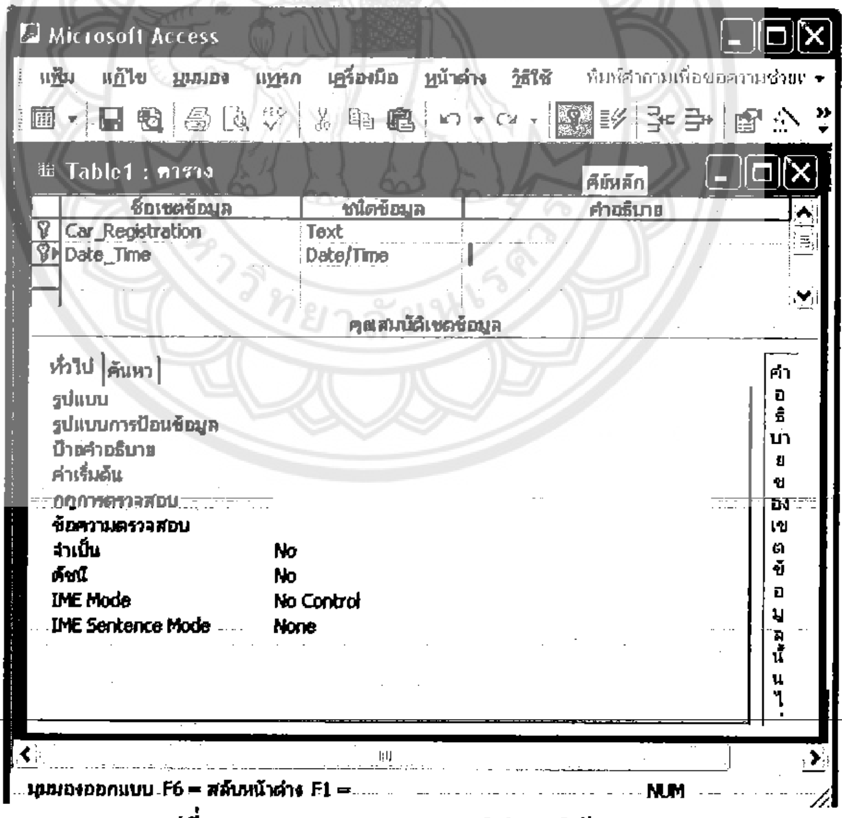
การจัดเก็บข้อมูลลงฐานข้อมูลนั้น จะใช้ Primary key 2 ตัว คือ ทะเบียนรถยนต์ กับ วันที่ และเวลา เพราะมีโอกาสเป็นไปได้น้อยมากที่จะมีรถที่มีหมายเลขทะเบียนเดียวกันถูกถ่ายในวันที่และเวลาเดียวกัน นอกจากว่าจะเป็นการถ่ายที่ใช้กล้องมากกว่า 1 ตัว

ขั้นตอนการสร้างฐานข้อมูลให้กับโปรแกรมมีดังนี้

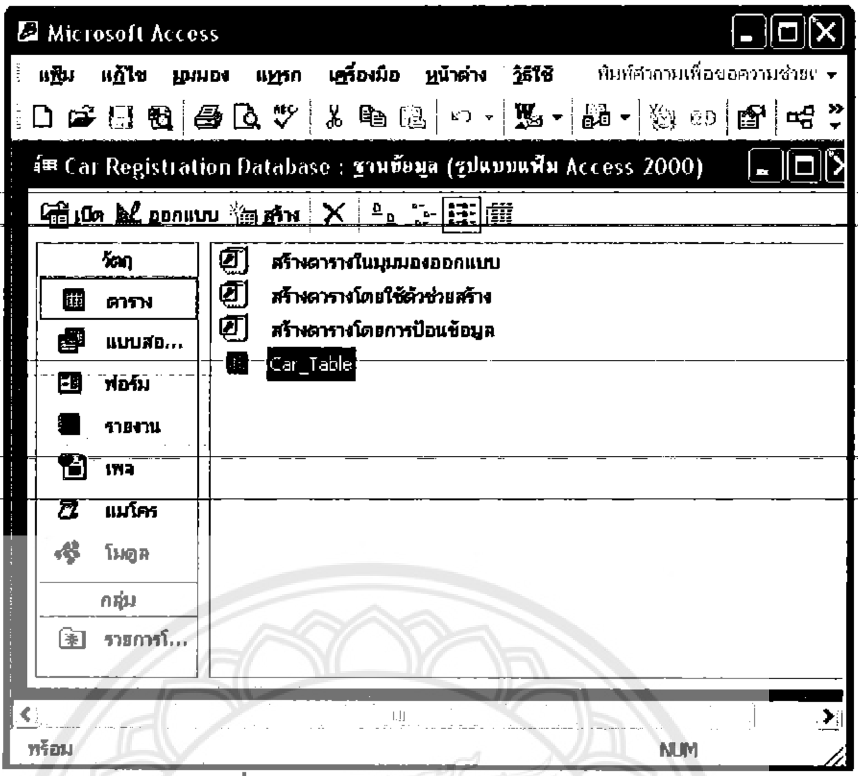
1. เปิดโปรแกรม Microsoft Access
2. ในหัวข้อสร้างทางด้านขวามือ เลือก “ฐานข้อมูลเปล่า” แล้วทำการตั้งชื่อแฟ้มฐานข้อมูลเป็น “Car Registration Database.mdb” จะปรากฏหน้าต่างดังรูปที่ 3.21
3. ดับเบิลคลิกที่สร้างตารางในมุมมองออกแบบ
4. ตั้งชื่อเขตข้อมูลเป็น “Car_Registration” ชนิดข้อมูลแบบ Text ขนาด 10 ตัวอักษร และ ชื่อเขตข้อมูล “Date_Time” ชนิดข้อมูลแบบ Date/Time
5. กำหนด primary key ให้กับเขตข้อมูลทั้งสองโดยเลือกเขตข้อมูลทั้งสองแล้วคลิกที่แถบเครื่องมือ “คีย์หลัก” ดังรูป 3.22
6. บันทึกตารางโดยตั้งชื่อว่า “Car_Table” ดังรูปที่ 3.23
7. ดับเบิลคลิกที่ Car_Table จะปรากฏฐานข้อมูลเปล่าดังรูปที่ 3.24



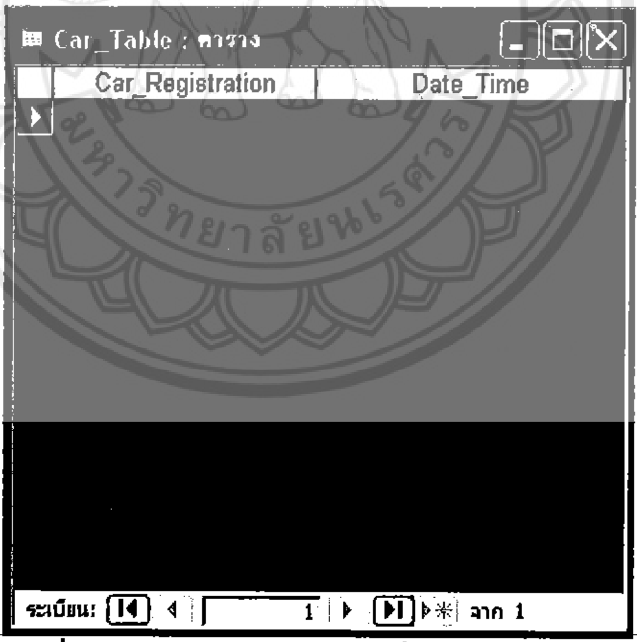
รูปที่ 3.21 เริ่มสร้างฐานข้อมูลชื่อ “Car Registration Database”



รูปที่ 3.22 การกำหนดคุณสมบัติต่างๆ ให้กับตาราง



รูปที่ 3.23 ตัวอย่างฐานข้อมูลของโปรแกรม



รูปที่ 3.24 หน้าต่างตารางว่างเปล่า เพื่อเตรียมรอรับข้อมูล

3.7 การติดต่อฐานข้อมูล

การติดต่อฐานข้อมูลในโครงงานนี้ จะเป็นการใช้ชุดคำสั่งของโปรแกรม Matlab 7.0 โดยการเรียกดูข้อมูล จะเป็นภาษา SQL ทั่วไป ทำให้สะดวกต่อการติดต่อฐานข้อมูล เพราะมีชุดคำสั่งรองรับ ไม่ต้องพึ่งพาโปรแกรมอื่นเข้ามาช่วย

ตัวอย่างการใช้งานคำสั่ง SQL ผ่านฟังก์ชันของโปรแกรม Matlab

- คำสั่งใช้เชื่อมต่อฐานข้อมูล

conn = database('datasourcename', 'username', 'password')

โดย

datasourcename คือ ชนิดของฐานข้อมูล ที่ได้ทำการตั้งค่าบน Data Sources (ODBC) ของระบบปฏิบัติการแล้ว ซึ่งสามารถศึกษาวิธีการตั้งค่าได้จากภาคผนวก

username และ password คือ ชื่อและรหัสผ่าน เพื่อเข้าถึงสิทธิต่างๆในฐานข้อมูล

conn คือ ตัวแปรที่เก็บค่าการเชื่อต่อของฐานข้อมูลไว้

ตัวอย่าง

conn = database('MS Access Database' ,'admin', '1234');

ทำการเชื่อมต่อฐานข้อมูล MS Access มีชื่อเป็น “admin” และรหัสผ่าน 1234

- คำสั่งที่ใช้เรียกดูข้อมูลภายในฐานข้อมูลตามเงื่อนไขที่ต้องการด้วยคำสั่ง SQL

curs = exec(conn, 'sqlquery')

curs = fetch(curs)

โดย

Conn คือ ตัวแปรที่เก็บค่าการเชื่อต่อของฐานข้อมูลไว้

Sqlquery คือ คำสั่ง SQL ที่ใช้เรียกดูข้อมูล

Curs คือ ข้อมูลที่ได้จากการเรียกดู

ตัวอย่าง

```
curs = exec(conn, ['SELECT Car_Registration...
            FROM Car_Table...
            WHERE Car_Registration= '''กข 1234''']);
curs = fetch(curs);
```

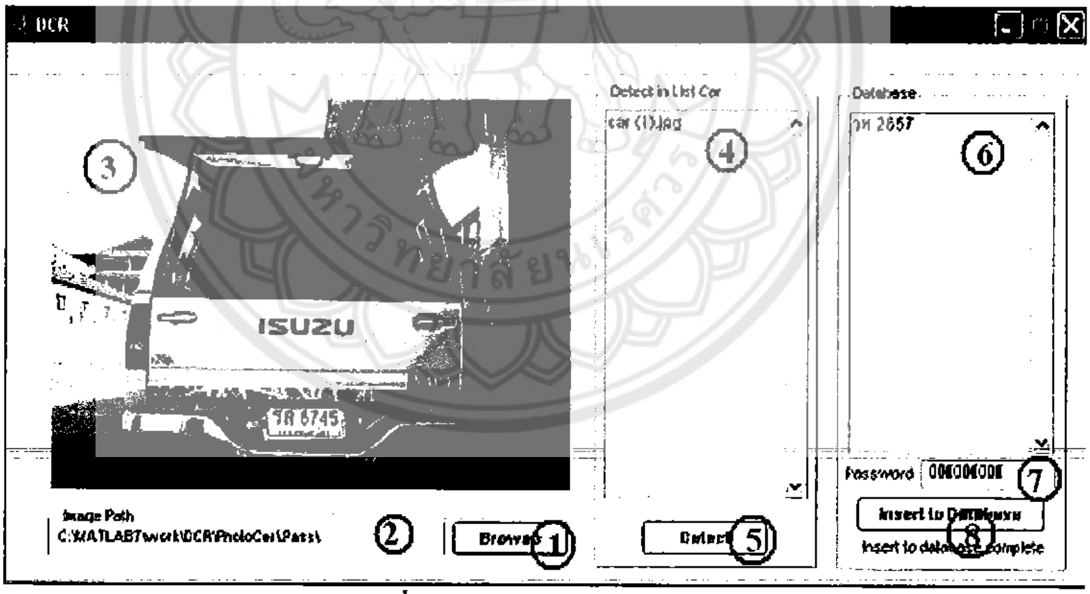
ทำการเรียกดูข้อมูลป้ายทะเบียนที่มีเลขทะเบียนเป็น “กข 1234” จากตารางรถ

3.8 โปรแกรมตรวจจับหมายเลขทะเบียนรถที่สร้างโดย GUI บน Matlab

โปรแกรมตรวจจับหมายเลขทะเบียนรถ (Detect Car Registration: DCR) คือ ผลสำเร็จของโครงงานนี้ โดยรวบรวมกระบวนการต่างๆ เริ่มตั้งแต่ การเตรียมภาพ การหาตำแหน่งป้ายทะเบียน การแยกหมายเลขทะเบียนออกจากแผ่นป้ายทะเบียน การรู้จำ และสุดท้ายเป็นการเก็บข้อมูลลงฐานข้อมูล ซึ่งการนำส่วนต่างๆมาเชื่อมโยงกัน และจัดหน้าตาของโปรแกรมให้ใช้งานได้ง่าย ดูเป็นระเบียบ จะทำได้โดยการสร้าง GUI ซึ่งวิธีสร้างวัตถุต่างๆบน GUI สามารถศึกษาได้จากบทที่ 2 หัวข้อที่ 2.5

คำอธิบายตามหมายเลขของส่วนต่างๆของโปรแกรม ดังรูป 3.25 มีดังนี้

1. ปุ่ม “Browse” ทำหน้าที่เลือกภาพที่จะนำมาหาหมายเลขทะเบียน
2. ที่อยู่ของภาพ
3. ภาพของรถที่เลือกมา
4. ชื่อของภาพที่เลือกมา
5. ปุ่ม “Detect” ทำหน้าที่ประมวลผลเพื่อหาหมายเลขทะเบียน
6. หมายเลขทะเบียนที่ได้จากภาพหลังทำการประมวลผล
7. ช่องใส่รหัสผ่านให้กับฐานข้อมูล
8. ปุ่ม “Insert to Database” ทำหน้าที่เก็บหมายเลขทะเบียนลงฐานข้อมูล



รูปที่ 3.25 ส่วนต่างๆของโปรแกรม

ขั้นตอนการใช้งานโปรแกรม และการติดตั้งโปรแกรม สามารถศึกษาเพิ่มเติมได้จากภาคผนวก ก. การติดตั้งโปรแกรม และการใช้โปรแกรม

บทที่ 4

ผลการทดลองและการวิเคราะห์

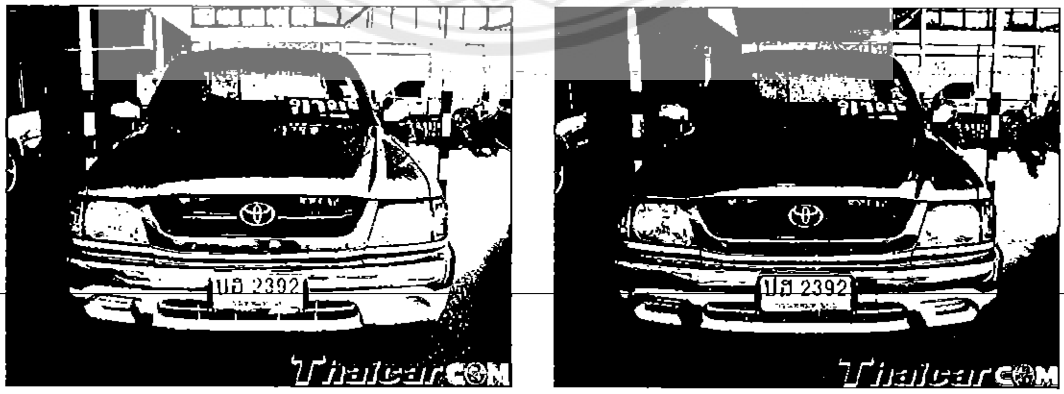
4.1 ผลการทดลองส่วนของการเตรียมข้อมูลภาพ

การทดลองจะใช้ภาพสีที่มีขนาด 640 x 480 ขึ้นไป เป็นภาพถ่ายด้านหน้าหรือด้านหลังของรถยนต์ ในการทดลองขั้นแรกเป็นส่วนของการเตรียมภาพ เพื่อรองรับภาพที่สว่างหรือมืดเกินไป ดังรูปที่ 4.1 เป็นภาพที่ถูกเทรชโฮลด้วยค่ากลาง แล้วกรอบของป้ายทะเบียนรถกลมกลืนเข้ากับตัวอักษรในป้ายทะเบียน และรูปที่ 4.2 กรอบของป้ายทะเบียนไม่ชัดเจน หรือขาดออกจากกัน ทำให้ไม่สามารถหาป้ายทะเบียนเจอได้ เราจึงเลือกค่าเทรชโฮลหลายๆค่าให้เหมาะสมต่อภาพนั้นๆ

(ก) (ข)

รูปที่ 4.1 เทรชโฮลด้วยค่าต่างๆ

(ก) ค่ากลางเท่ากับ 98 แล้วมืดเกินไป (ข) 73 แล้วสามารถเห็นกรอบทะเบียน



(ก) (ข)

รูปที่ 4.2 เทรชโฮลด้วยค่าต่างๆ

(ก) ค่ากลางเท่ากับ 126 แล้วสว่างเกินไป (ข) 151 แล้วสามารถเห็นกรอบทะเบียน

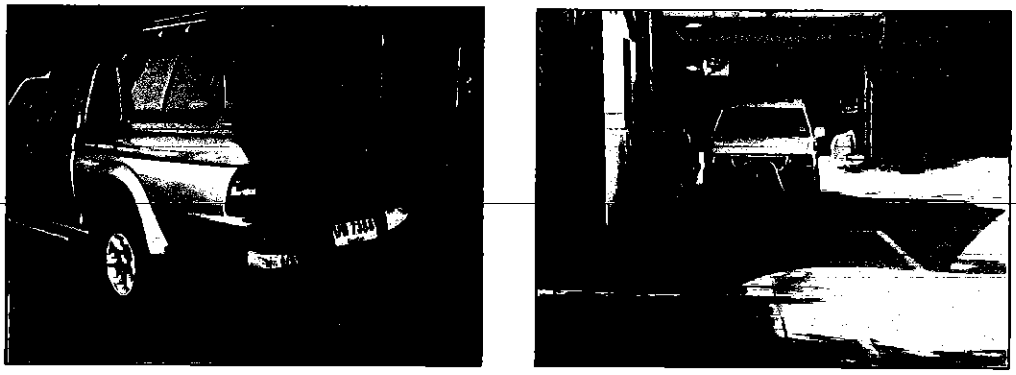
4.2 ผลการทดลองส่วนของการหาตำแหน่งของป้ายทะเบียนรถยนต์

ในการทดลองนี้ใช้ข้อมูลภาพทั้งหมด 37 ภาพ สามารถหาทะเบียนรถยนต์เจอ 35 ภาพ คิดเป็นร้อยละ 94.59 จากข้อมูลภาพทั้งหมด โดยภาพที่ไม่สามารถหาทะเบียนรถได้นั้นข้อมูลที่ได้รับจะเป็น XX XXXX ในส่วนของฐานข้อมูล แสดงดังรูปที่ 4.3

รูปที่ 4.3 โปรแกรมในขั้นตอนตรวจจับหมายเลขทะเบียนรถที่เกิดข้อผิดพลาด

การทดลองได้ทำการศึกษาปัจจัยต่างๆที่มีผลต่อการทำงานของโปรแกรมในขั้นตอนการหาตำแหน่งของป้ายทะเบียนรถยนต์ เพื่อเป็นแนวทางในการออกแบบ และปรับปรุงแก้ไขโปรแกรมต่อไปในลักษณะของภาพที่ไม่สามารถทำการหาป้ายทะเบียนได้ดังรูปที่ 4.4 มีดังนี้

1) ภาพที่ป้ายทะเบียนรถเอียงจนตัวอักษรบิดเบี้ยวไป
2) ภาพที่ถ่ายในระยะไกลจนขนาดของรถไม่ถึงครึ่งของขนาดภาพ
3) ภาพที่ถ่ายมีลักษณะเบลอ



(ก) (ข)

(ค)

รูปที่ 4.4 ภาพที่ไม่สามารถทำการตรวจสอบ

(ก) ภาพที่ป้ายทะเบียนรถเอียงจนตัวอักษรบิดเบี้ยวไป

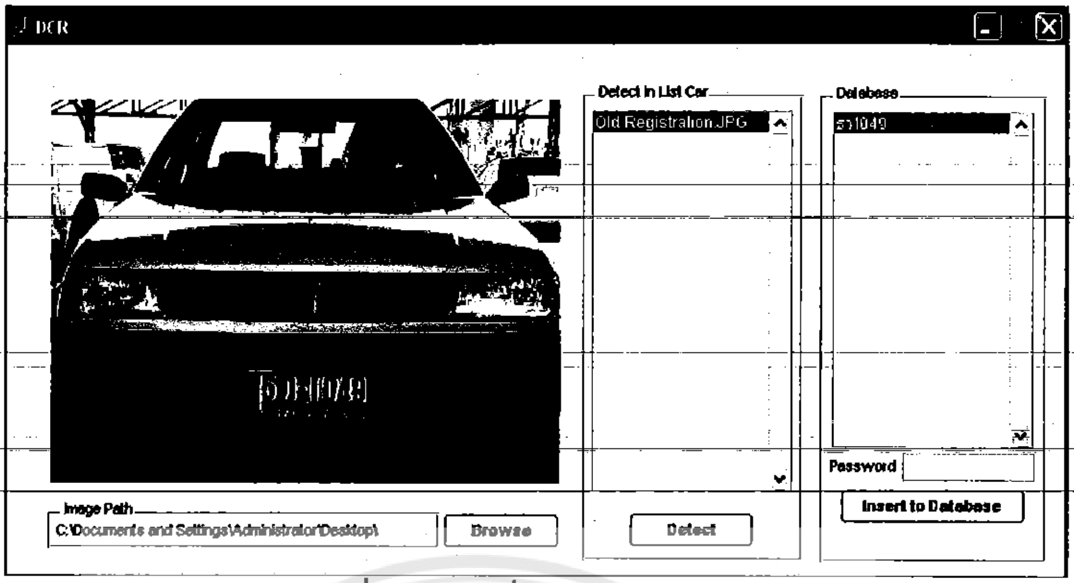
(ข) ภาพระยะไกลจนขนาดของรถไม่ถึงครึ่งของขนาดภาพ

(ค) ภาพมีลักษณะเบลอ

4.3 ผลการทดลองส่วนของการแยกหมายเลขทะเบียนออกจากป้ายทะเบียน

เนื่องจากได้ระบุให้โปรแกรมรู้อยู่ก่อนแล้วว่า 2 บล็อกหน้าเป็นบล็อกของตัวอักษร และ 3 บล็อกต่อมาเป็นบล็อกของตัวเลข ดังรูปที่ 4.5 ซึ่งเป็นรูปแบบของหมายเลขทะเบียนภายในป้ายทะเบียนรถแบบใหม่ที่ใช้กันอยู่ทุกวันนี้ ทำให้โปรแกรมทำงานได้ถูกต้อง และรวดเร็วขึ้น แต่ความผิดพลาดยังเกิดขึ้นเป็นบางครั้ง เนื่องจากรูปแบบของป้ายทะเบียนที่ผิดไปจากเดิม นั่นคือ ป้ายทะเบียนที่มีตัวเลขก่อนตัวอักษรหรือป้ายทะเบียนที่มีเพียงตัวเลขเท่านั้น เช่น รถยนต์ที่มีป้ายทะเบียนแบบเก่าดังรูปที่ 4.6 เมื่อเข้าสู่กระบวนการรู้จำจะทำให้เลข “5” เป็น “ธ” หรือ รถบรรทุกที่มีหมายเลขทะเบียน 80-5046 เป็นต้น

รูปที่ 4.5 ผลลัพธ์จากการแยกเลขทะเบียน

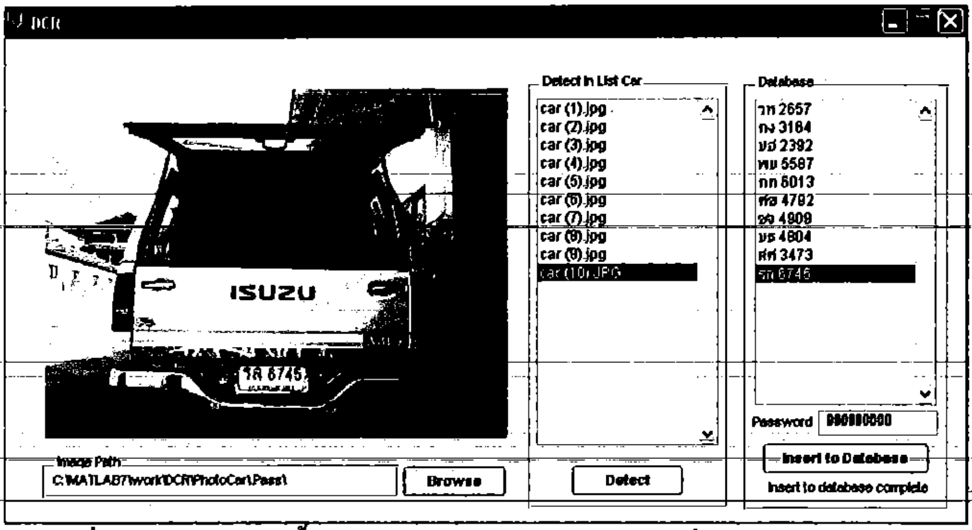


รูปที่ 4.6 รถยนต์ที่มีป้ายทะเบียนแบบเก่า

4.4 ผลการทดลองส่วนของการรู้จำ

การทดลองเพื่อศึกษาถึงผลลัพธ์ของการเปรียบเทียบข้อมูลต้นแบบกับข้อมูลรับเข้าที่เปลี่ยนแปลงไปหลายๆค่า ผลลัพธ์ที่ได้มีความยืดหยุ่นต่อสภาพแวดล้อมมากน้อยเพียงใด การนำจุดปลายของตัวอักษร สามารถเข้ามาช่วยในกระบวนการการรู้จำได้หรือไม่ รวมทั้งผลลัพธ์แต่ละตัวเมื่อเทียบกับข้อมูลรับเข้า มีร้อยละความถูกต้องสมบูรณ์ของทั้งแผ่นป้ายทะเบียนเพื่อพิจารณาความเป็นไปได้ในการใช้งาน และเป็นแนวทางในการออกแบบพัฒนา ต่อไป

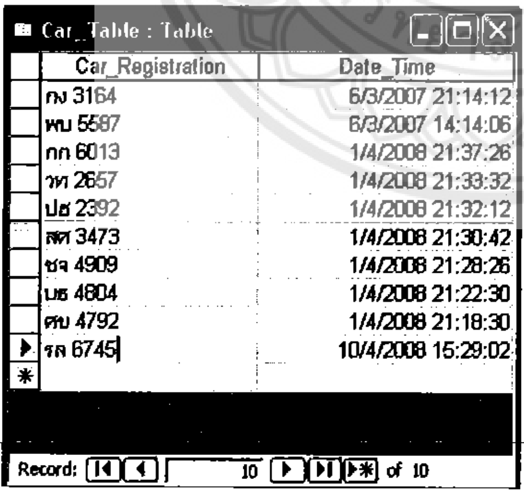
จากการทดลอง เมื่อใช้กระบวนการเทียบเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ รวมกับการนำจุดปลายของตัวอักษรเข้ามาช่วยนั้น ทำให้จำนวนของตัวอักษรที่วิเคราะห์ผิดพลาดลดน้อยลง ในขั้นตอนทดลองนี้สามารถหาตำแหน่งของป้ายทะเบียนและแยกหมายเลขทะเบียนออกมาได้เป็นจำนวน 35 ภาพ และจากจำนวนตัวอักษรที่แยกออกมาได้ สามารถรู้จำตัวอักษรที่ถูกต้องทั้งแผ่นป้ายได้จำนวน 30 ป้าย คิดเป็นร้อยละ 85.71 จากแผ่นป้ายทะเบียนรถที่หาเจอ ได้แสดงตัวอย่างโปรแกรมที่รู้จำหมายเลขทะเบียนรถยนต์จำนวน 10 คันได้อย่างถูกต้อง ดังรูปที่ 4.7



รูปที่ 4.7 โปรแกรมในขั้นตอนตรวจจับหมายเลขทะเบียนรถที่ถูกต้องจำนวน 10 คัน

4.5 ผลการทดลองส่วนของการจัดเก็บข้อมูล

หลังจากผ่านกระบวนการรู้จำแล้ว โปรแกรมจะเก็บข้อมูลทั้งที่ถูกต้องและไม่ถูกต้อง รวมทั้งภาพที่ไม่สามารถหาตำแหน่งป้ายทะเบียนรถได้เจอ (XX XXXX) ลงในฐานข้อมูล Microsoft Access โดยจะเก็บข้อมูลเพิ่มเข้าไปจากข้อมูลเดิมที่มีอยู่ ดังรูปที่ 4.8 ซึ่งเดิมมีหมายเลขทะเบียนอยู่ 10 หมายเลข และใช้โปรแกรมตรวจจับเพิ่มอีก 4 คัน แล้วจัดเก็บข้อมูลลงฐานข้อมูล ทำให้มีหมายเลขทะเบียนเพิ่มเป็น 14 หมายเลข

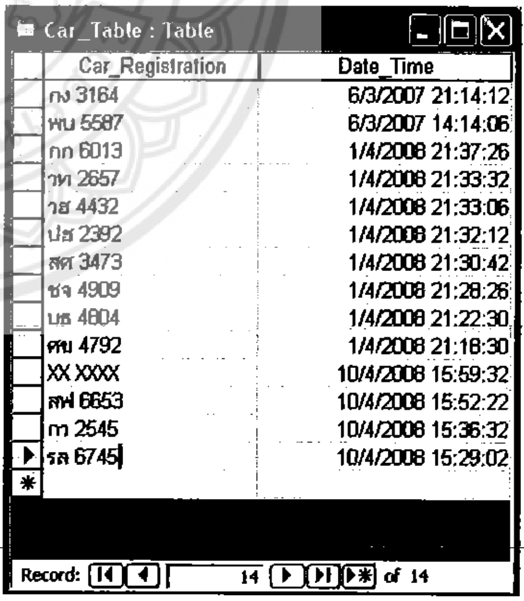


Car_Table : Table

Car_Registration	Date_Time
กง 3164	6/3/2007 21:14:12
พบ 5587	6/3/2007 14:14:06
กก 6013	1/4/2008 21:37:26
วท 2657	1/4/2008 21:33:32
ปช 2392	1/4/2008 21:32:12
สค 3473	1/4/2008 21:30:42
ชจ 4909	1/4/2008 21:28:26
บธ 4804	1/4/2008 21:22:30
ศบ 4792	1/4/2008 21:18:30
รล 6745	10/4/2008 15:29:02

Record: 10 of 10

(ก)



Car_Table : Table

Car_Registration	Date_Time
กง 3164	6/3/2007 21:14:12
พบ 5587	6/3/2007 14:14:06
กก 6013	1/4/2008 21:37:26
วท 2657	1/4/2008 21:33:32
วธ 4432	1/4/2008 21:33:06
ปช 2392	1/4/2008 21:32:12
สค 3473	1/4/2008 21:30:42
ชจ 4909	1/4/2008 21:28:26
บธ 4804	1/4/2008 21:22:30
ศบ 4792	1/4/2008 21:18:30
XX XXXX	10/4/2008 15:59:32
สห 6653	10/4/2008 15:52:22
ฑ 2545	10/4/2008 15:36:32
รล 6745	10/4/2008 15:29:02

Record: 14 of 14

(ข)

รูปที่ 4.8 ข้อมูลที่เก็บลงในฐานข้อมูล

(ก) ข้อมูลที่ถูกจัดเก็บอยู่ 10 หมายเลขทะเบียน (ข) ข้อมูลที่ถูกจัดเก็บเพิ่มอีก 4 หมายเลขทะเบียน

บทที่ 5
สรุป

5.1 สรุปผลการทดลอง

โปรแกรมที่พัฒนาขึ้น สามารถตรวจจับหมายเลขทะเบียนรถจากภาพนิ่งของรถยนต์ แม้ว่าภาพนิ่งของรถยนต์นั้นจะมีปริมาณแสงที่ไม่เหมาะสม เช่น มืดหรือสว่างจนเกินไป อีกทั้งความสมบูรณ์ของป้ายทะเบียนรถอาจจะลดลงไปบ้าง ก็ยังสามารถตรวจจับหมายเลขทะเบียนได้ถูกต้องแม่นยำ นอกจากนี้โปรแกรมนี้สามารถใช้งานในการตรวจจับหมายเลขทะเบียนรถได้เสมอ อีกทั้งยังสามารถนำไปใช้งานอื่นๆ ที่เกี่ยวข้องได้ อาทิเช่น สืบค้นคดีอาชญากรรมของกรมตำรวจ หรือลดภาระงานให้กับผู้ที่ทำหน้าที่จดบันทึกการเข้า-ออกของรถ ในสถานที่ต่างๆ

5.2 ปัญหาและอุปสรรค

จากการทดลองเพื่อหาปัจจัยที่ส่งผลกระทบต่อการทำงานของโปรแกรม ทำให้ทราบถึงสาเหตุต่างๆ ที่ทำให้เกิดข้อผิดพลาดขึ้น โดยส่งผลให้ไม่สามารถแยกส่วนของหมายเลขทะเบียนรถได้อย่างสมบูรณ์ และได้ข้อจำกัดในการใช้งาน ดังต่อไปนี้

1. เลขทะเบียนมีสีซีด หรือ สีถลอก ทำให้การรู้จำของระบบเกิดความผิดพลาด
2. เลขทะเบียนมีคราบสกปรก ทำให้เลขทะเบียนเชื่อมติดกัน
3. สภาพแสงภายนอก มีแสงสะท้อนที่แผ่นป้ายทะเบียนสูง ทำให้ไม่สามารถหาตำแหน่งที่เป็นส่วนของป้ายทะเบียนได้
4. ความเอียงของรูปภาพ หรือมุมกล้องที่ถ่าย ทำให้ระบบทำงานผิดพลาด

ดังนั้นในทางปฏิบัติสามารถนำข้อจำกัดดังกล่าวมาพัฒนาโปรแกรมเพื่อสามารถนำมาใช้งานจริงได้อย่างมีประสิทธิภาพ

5.3 แนวทางการพัฒนาในอนาคต

1. เชื่อมต่อกล้อง เพื่อถ่ายภาพรถ ให้ใช้งานได้แบบเวลาจริง
2. โปรแกรมจะสามารถใช้งานกับภาพมุมเอียงได้
3. ปรับปรุงในส่วนของการประมวลผลให้รวดเร็วขึ้น
4. นำกระบวนการโครงข่ายประสาทเทียมเข้ามาช่วยในการรู้จำ

เอกสารอ้างอิง

[1] Baxes, Gregory A. **Digital Image Processing.** Canada : John Wiley & Sons, Inc. 1994.

[2] Pearson, D. **Image Processing.** Singapore : McGraw-Hill Book Company, Inc. 1991.

[3] Gonzalez, R.C. and Richard E. Woods. **Digital Image Processing.** 2nd Ed., New Jersey : Prentice-Hall. Inc. 2001.

[4] "Image Segmentation." [online]. Available : Http://fivedots.coe.psu.ac.th/~montri/Teaching/image/segment.DOC. 2551.

[5] "Morphological Image Processing." [online]. Available : Http://fivedots.coe.psu.ac.th/~montri/Teaching/image/morph.DOC. 2551.

[6] "ระบบฐานข้อมูล." [online]. Available : Http://www.chandra.ac.th/office/ict/document/it/it04/page01.html. 2551.

[7] "ODBC." [online]. Available : Http://www.phpstreet.com/ tutorials/php/php_db_odbc.php. 2551.

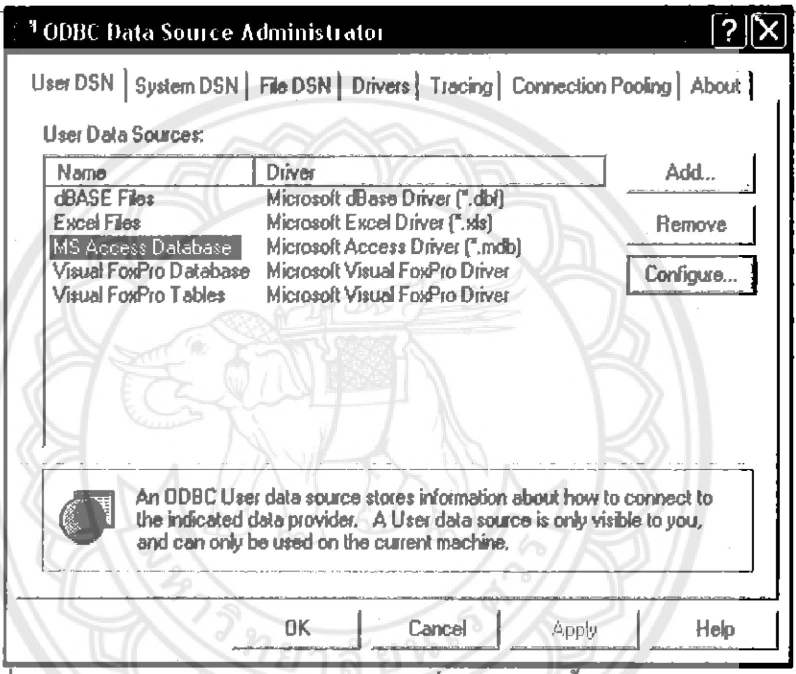
[8] "การใช้งานโปรแกรม Matlab เบื้องต้น." [online]. Available : Http://www.kmutt.ac.th/science/book/intromatlab_th.pdf

ภาคผนวก ก.

การติดตั้งโปรแกรมและการใช้โปรแกรม

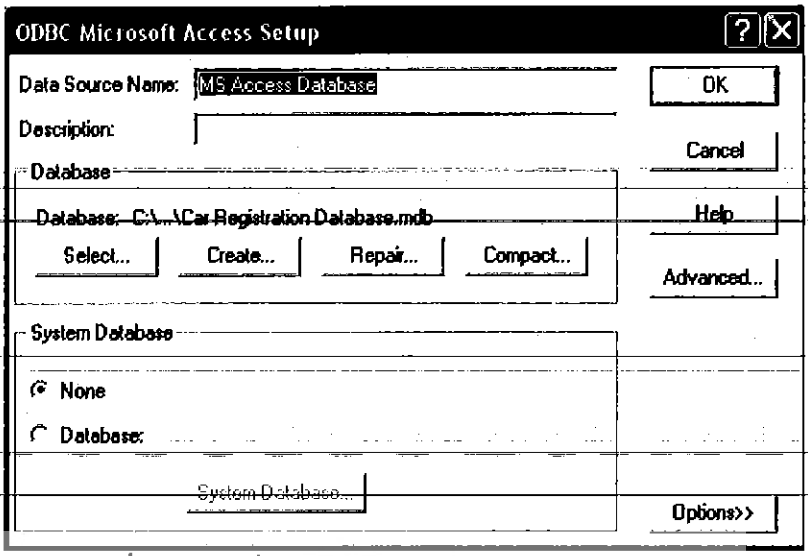
การติดตั้งโปรแกรม

1. copy folder "DCR With Database" จากแผ่นดิสก์ลงในคอมพิวเตอร์ไว้ที่ใดก็ได้ตามต้องการ
2. ไปที่ Start -> Control Panel -> Administrative Tools -> Data Sources (ODBC)



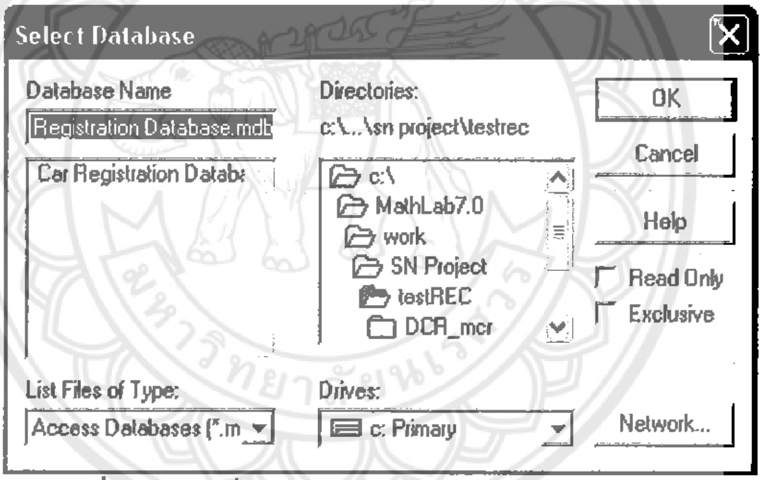
รูปที่ ก-1 ส่วนของ Data Sources (ODBC) เพื่อทำการติดตั้งให้ใช้งานกับโปรแกรม

3. เลือกที่ MS Access Database กด Configure หน้าต่าง ODBC Microsoft Access Setup จะปรากฏขึ้น



รูปที่ ก-2 เลือกที่อยู่ของไฟล์ Car Registration Database.mdb

4. เลือกที่ Select หน้าต่าง Select Database จะปรากฏขึ้น

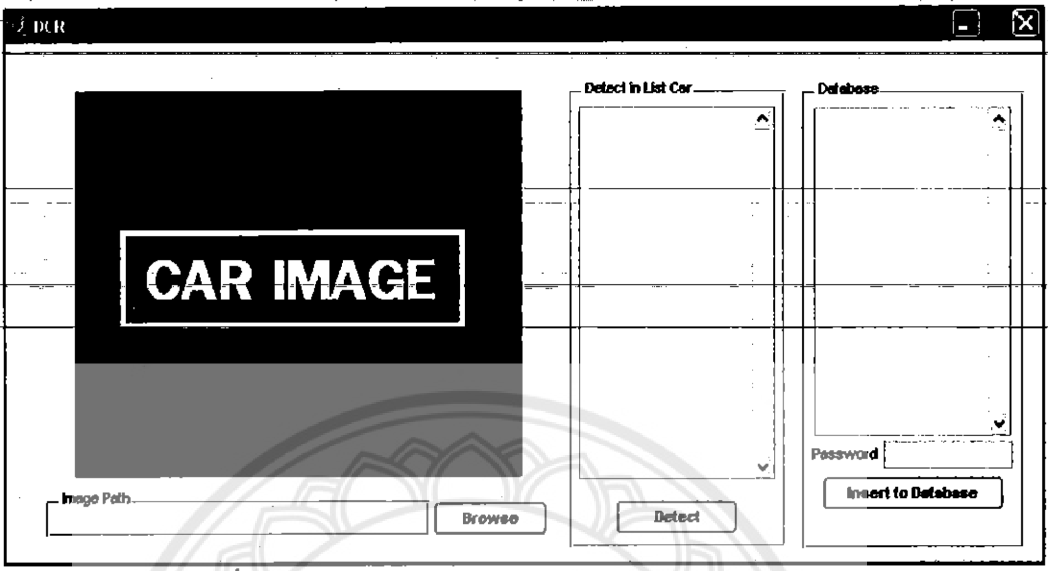


รูปที่ ก-3 ไปยังที่อยู่ของไฟล์ Car Registration Database.mdb

ให้เลือกที่อยู่ โดยไปที่ folder ที่ copy ลงมา จะปรากฏฐานข้อมูลชื่อ "Car Registration Database.mdb" คลิก OK

การใช้งานโปรแกรม

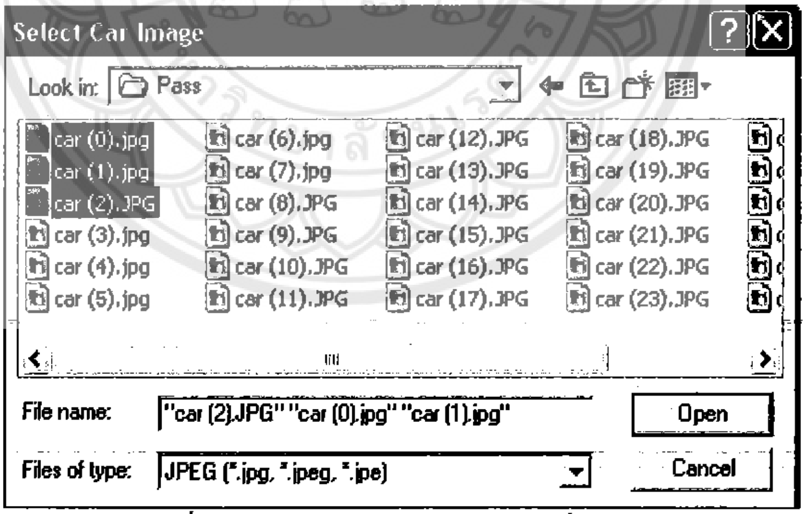
1. ไปยัง folder ที่ copy มา ดับเบิลคลิกที่ DCR.exe



รูปที่ ก-4 หน้าต่างโปรแกรมตรวจจับหมายเลขทะเบียนรถยนต์

หน้าต่างโปรแกรมจะปรากฏขึ้น

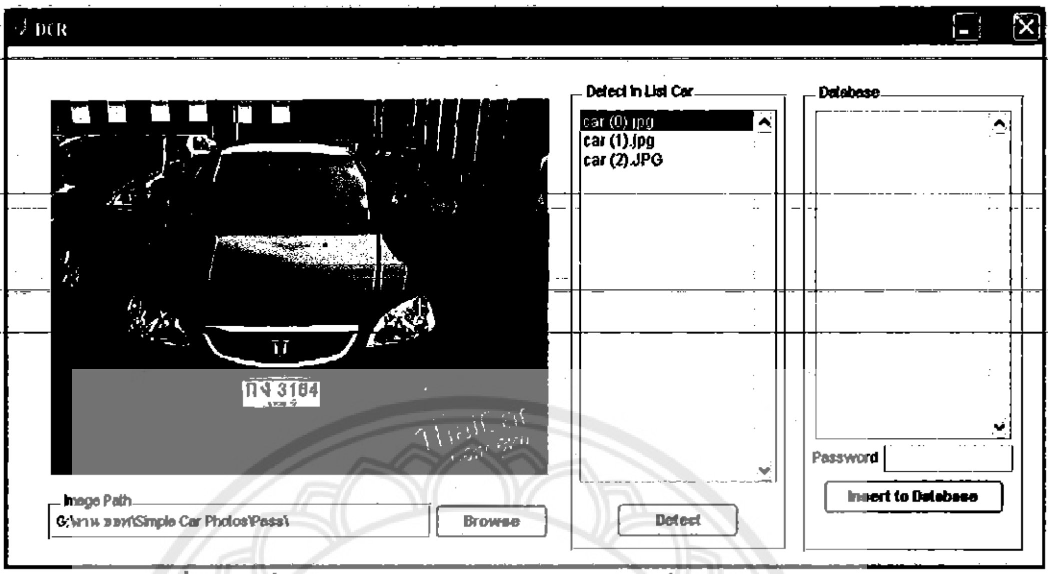
2. คลิก Browse เพื่อหาภาพรถที่ต้องการ Detect ป้ายทะเบียน



รูปที่ ก-5 ทำการเลือกไฟล์ภาพรถยนต์ที่ต้องการ

ในการจะเลือกภาพหลายๆภาพให้กด Ctrl ค้างไว้แล้วเลือกภาพทีละภาพ หรือถ้าต้องการเลือกภาพจากภาพแรกถึงภาพที่สิบ ให้กด Shift ค้างไว้แล้วคลิกที่ภาพแรกตามด้วยภาพที่สิบ เลือก Open

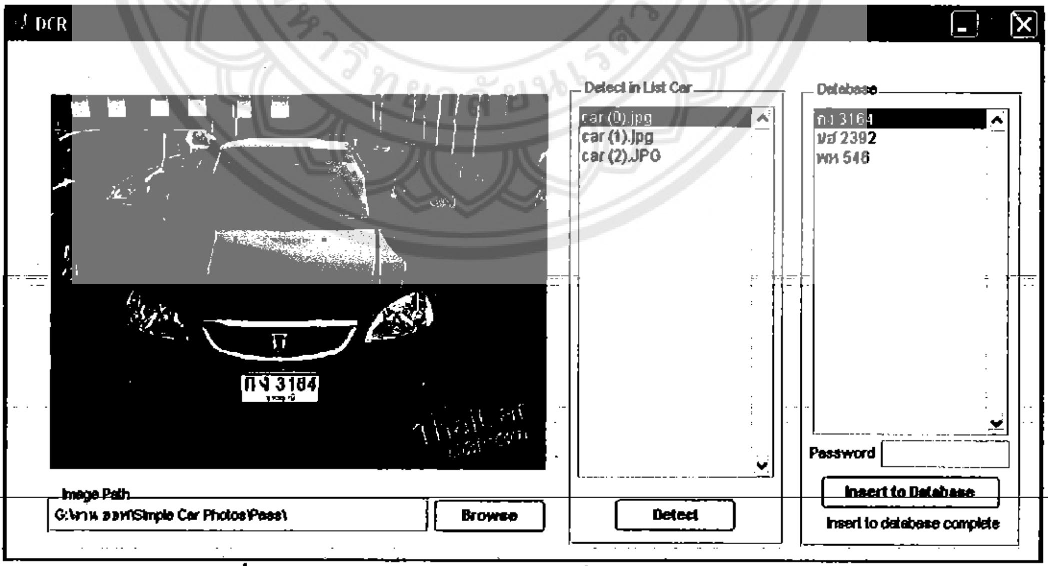
3. ภาพรถจะปรากฏ สามารถเลือกดูภาพรถได้ โดยคลิกเมาส์ตามชื่อรถที่เลือกมา



รูปที่ ก-6 เมื่อเลือกไฟล์ภาพแล้วทำการตรวจจับเพื่อหาหมายเลขทะเบียน

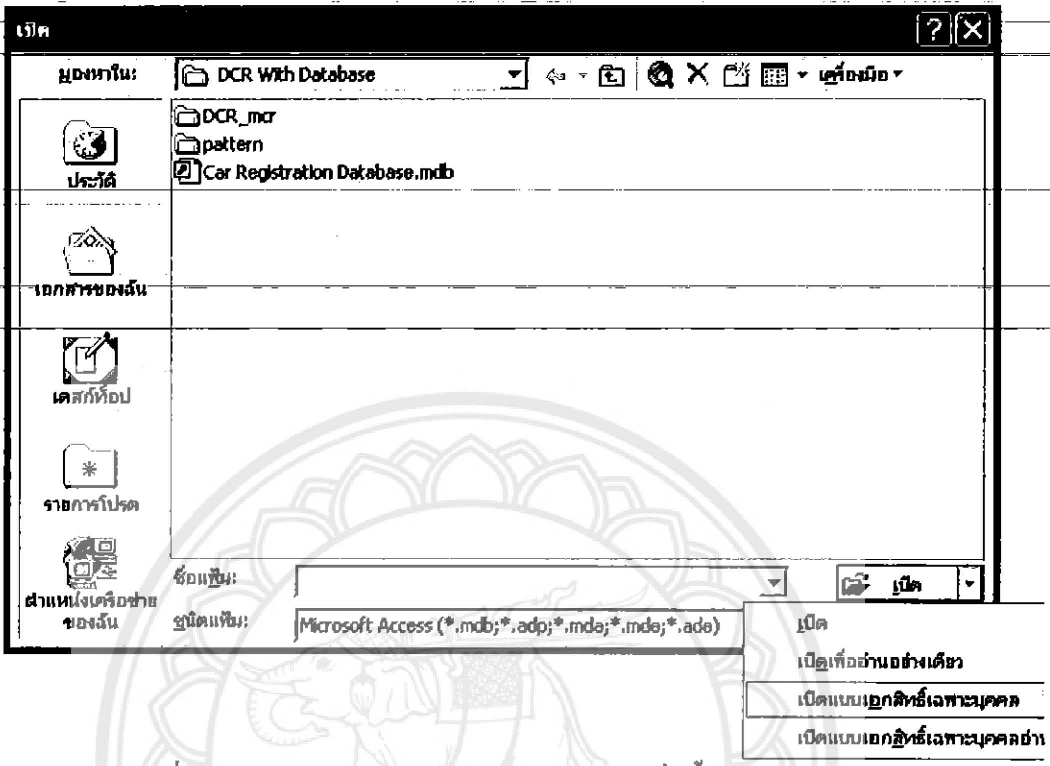
คลิกที่ "Detect" โปรแกรมจะทำการประมวลผล

4. ผลลัพธ์จะแสดงทางช่อง Database เป็นเลขทะเบียนรถของรถคันนั้นๆ คลิก "Insert to Database" เพื่อจัดเก็บลงฐานข้อมูล



รูปที่ ก-7 หมายเลขทะเบียนรถยนต์ที่ตรวจจับได้จากไฟล์ภาพ

ในเบื้องต้นยังไม่ได้มีการตั้งรหัสผ่านให้กับฐานข้อมูล ถ้าต้องการตั้งรหัสผ่านให้เข้าโปรแกรม Microsoft Access ก่อน



รูปที่ ก-8 การเข้าสู่โปรแกรม Microsoft Access เพื่อตั้งรหัสผ่านฐานข้อมูล

ตรงเมนูบาร์ เลือก File -> Open แล้วไปยัง folder "DCR With Database" ที่ copy มา เลือก "Car Registration Database.mdb" คลิกลูกศรชี้ลงข้างปุ่ม "Open" เลือก "เปิดแบบเอกสิทธิ์เฉพาะบุคคล" ดับเบิลคลิกตรงเมนูบาร์ เลือก เครื่องมือ -> ความปลอดภัย -> ตั้งรหัสผ่านฐานข้อมูล แล้วใส่รหัสตามต้องการ

เมื่อได้ทำการตั้งรหัสผ่านแล้ว ในโปรแกรม DCR ต้องใส่รหัสผ่านที่ตั้งไว้ในช่อง "Password" ก่อนคลิกปุ่ม "Insert to Database"

ภาคผนวก ข.

ตัวอย่างการแปลงไฟล์

การแปลง M - file ให้เป็น Execute - file

mcc คือ คำสั่งหนึ่งบน Matlab ที่สามารถทำให้ M-file กลายเป็น Execute-file โดยผ่านตัวคอมไพเลอร์ ซึ่งจะทำให้โปรแกรมที่เราเขียนขึ้น ไม่ต้องทำงานผ่านโปรแกรม Matlab แต่สามารถทำงานได้บนระบบปฏิบัติการทั่วไปได้เลยอย่างเพียงลำพัง โดยตัวคอมไพเลอร์จะสร้างฟังก์ชัน C หรือ C++ สำหรับ M-file ที่เราเขียนขึ้น

รูปแบบการใช้งาน compiler คำสั่ง mcc บน Matlab

```
mcc [- options] mfile
```

โดย

options คือ การเลือกใช้รูปแบบเพื่อสร้าง Execute ไฟล์

mfile คือ ชื่อของ M-file ที่ต้องการแปลง

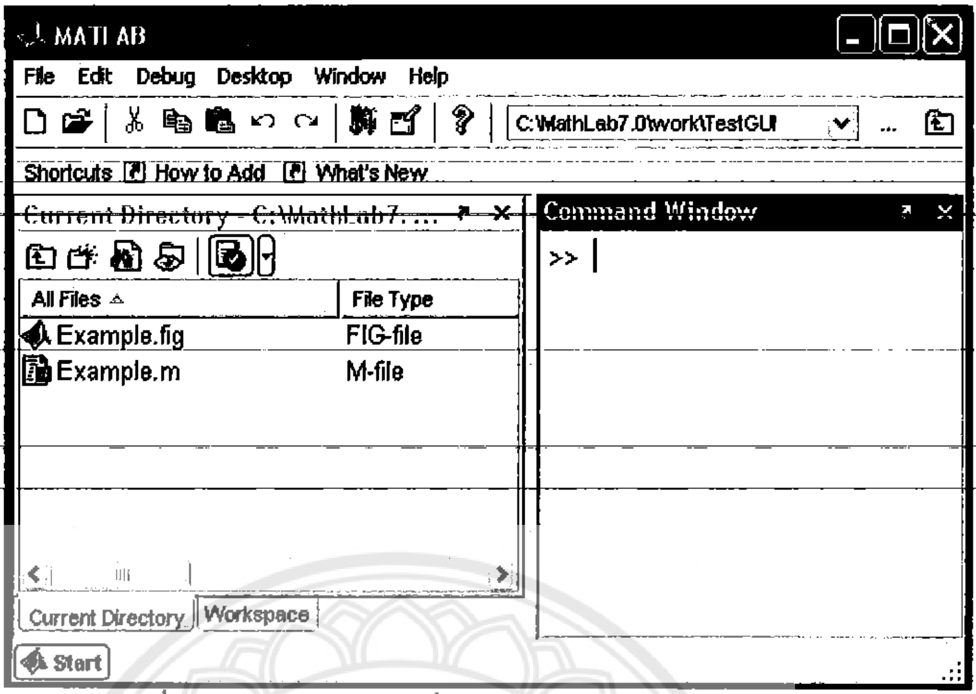
ตัวอย่าง option ของฟังก์ชัน mcc ได้แก่ – m, - v, - o, - w เป็นต้น ซึ่งสามารถศึกษารายละเอียดของ option แต่ละตัวเพิ่มเติมได้ โดยพิมพ์ help mcc ในโปรแกรม Matlab

ตัวอย่างการแปลง M-file ให้เป็น Execute-file

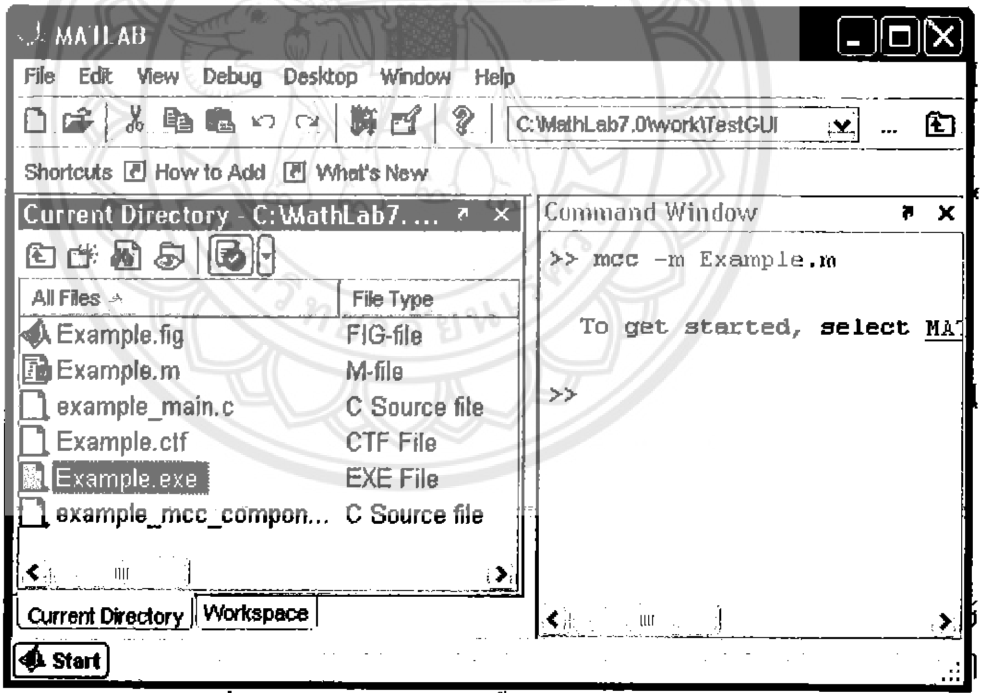
บน Matlab Command Window ที่ prompt สั่ง

```
>> mcc – m Example.m
```

เมื่อใช้คำสั่งนี้แล้วโปรแกรม Matlab จะมีการให้เลือกใช้ compiler ระหว่าง C++ และ Matlab โดยเราเองนั้นสามารถเลือกใช้ compiler ตัวใดตัวหนึ่งก็ได้ ส่วนใหญ่จะทำการเลือก compiler ที่เป็น Matlab เมื่อทำการแปลงไฟล์โดยการพิมพ์ดังกล่าวและเลือกใช้ตัว compiler แล้ว ผลลัพธ์ที่ได้จะมี Execute-file เพิ่มขึ้นมา ดังรูป ข-1 และ ข-2 ตามลำดับ



รูปที่ ข-1 หลังการสร้าง GUI ชื่อไฟล์ Example บนโปรแกรม Matlab 7.0



รูปที่ ข-2 Execute-file ที่เพิ่มขึ้นมาจากใช้คำสั่ง mcc

ภาคผนวก ค.

ตัวอย่างโค้ดโปรแกรม

โปรแกรมการรู้จำหมายเลขทะเบียนรถยนต์

```
function varargout = DCR(varargin)
% DCR M-file for DCR.fig
%      DCR, by itself, creates a new DCR or raises the existing
%      singleton*.
%
%      H = DCR returns the handle to a new DCR or the handle to
%      the existing singleton*.
%
%      DCR('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in DCR.M with the given input arguments.
%
%      DCR('Property','Value',...) creates a new DCR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before DCR_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to DCR_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help DCR

% Last Modified by GUIDE v2.5 25-Feb-2008 14:27:22
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @DCR_OpeningFcn, ...
                   'gui_OutputFcn',  @DCR_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
  gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
  [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
  gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before DCR is made visible.
function DCR_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to DCR (see VARARGIN)

if length(varargin)==0
  imshow(imread('initial.jpg'));
  handles.current_data=[];
else
```

```
    allInput=varargin{:};
    imshow(imread([char(allInput(2)) char(allInput(3))]));
    set(handles.pathFileName, 'String', allInput(2));
    set(handles.listCar, 'String', allInput(3:str2num(char(allInput(1)))+2));
    handles.current_data=allInput;
    set(handles.insertDatabaseList, 'String', allInput(str2num(char(allInput(1)))+3:end))
end

% Choose default command line output for DCR
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes DCR wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = DCR_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in browse.
function browse_Callback(hObject, eventdata, handles)
% hObject    handle to browse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
[fileName pathName] = uigetfile( ...
    {'*.jpg; *.jpeg; *.jpe','JPEG (*.jpg, *.jpeg, *.jpe)'; ...
    '*.bmp','BMP (*.bmp)'; ...
    '*.gif','Compuserve GIF (*.gif)'; ...
    '*.png','PNG (*.png)'; ...
    '*.tif; *.tiff','TIFF (*.tif, *.tiff)'; ...
    '*.*', 'All Files (*.*)'}, ...
    'Select Car Image', ...
    'MultiSelect', 'on');
if pathName~=0 | fileName~=0
  tmp=cellstr(fileName);
  tmp=[tmp(2:end) tmp(1)];
  DCR([cellstr(char(num2str(length(tmp)))) cellstr(pathName) tmp]);
  set(handles.insertDatabaseList, 'String', []);
  set(handles.insertComplete, 'String', []);
end

% --- Executes on button press in detect.
function detect_Callback(hObject, eventdata, handles)
% hObject    handle to detect (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tmp=handles.current_data;
if length(tmp)~=0 & length(get(handles.insertDatabaseList, 'String'))==0
  pathName=tmp(2);
  fileName=tmp(3:end);
  %reg=[];
  for i=1:1:length(fileName)
    reg=cellstr(detectCarReg([char(pathName) char(fileName(i))]));
    tmp=[tmp cellstr(reg)];
    DCR(tmp);
  end
```

```
end

% --- Executes on button press in insertDatabase.
function insertDatabase_Callback(hObject, eventdata, handles)
% hObject    handle to insertDatabase (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
carNamePath=get(handles.pathFileName,'String');
carNameList=get(handles.listCar,'String');
regList=get(handles.insertDatabaseList,'String');

if isempty(carNameList) | isempty(regList)
    set(handles.insertComplete, 'String', 'No data for insert to database');
else
    pass=get(handles.password, 'String');
    conn=database('MS Access Database' ,'admin', pass);
    if isconnection(conn)
        DBCarReg(conn, carNamePath, carNameList, regList);
        set(handles.insertComplete, 'String', 'Insert to database complete');
    else
        set(handles.insertComplete, 'String', 'Can'' t connect database');
    end
end

% --- Executes on selection change in listCar.
function listCar_Callback(hObject, eventdata, handles)
% hObject    handle to listCar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listCar contents as cell array
%        contents{get(hObject,'Value')} returns selected item from listCar
```

```
value=get(hObject, 'Value');
tmp=handles.current_data;
if length(tmp)~=0
   pathName=tmp(2);
   fileName=tmp(3:end);
   carImage=imread([char(pathName) char(fileName(value))]);
   imshow(carImage);
end


% --- Executes during object creation, after setting all properties.
function listCar_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listCar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in insertDatabaseList.
function insertDatabaseList_Callback(hObject, eventdata, handles)
% hObject    handle to insertDatabaseList (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns insertDatabaseList contents as cell array
%        contents{get(hObject,'Value')} returns selected item from insertDatabaseList
```

```
% --- Executes during object creation, after setting all properties.
function insertDatabaseList_CreateFcn(hObject, eventdata, handles)
% hObject    handle to insertDatabaseList (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function password_Callback(hObject, eventdata, handles)
% hObject    handle to password (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of password as text
%        str2double(get(hObject,'String')) returns contents of password as a double


% --- Executes during object creation, after setting all properties.
function password_CreateFcn(hObject, eventdata, handles)
% hObject    handle to password (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
```

```
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on key press over password with no controls selected.
function password_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to password (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

s=uint32(get(handles.figure1,'CurrentCharacter'));

% --- Executes during object deletion, before destroying properties.
function password_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to password (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

ฟังก์ชันรู้จำหมายเลขทะเบียนรถยนต์

```
function reg=detectCarReg(fileName)

% Load pattern of text and num
% ==========================================
P=struct('pattern', {});
for i=1:1:52
   pIm=imread(['pattern/pattern (' num2str(i-1) ').bmp']);
   P(i).pattern=pIm;
end

% Find car registration position
% ==========================================
```

```
[S indexText indexNum]=findReg(fileName, 0);

% Recognize text and number of car registration
% ===============================================
if length(S(1).data)==1
   [S indexText indexNum]=findReg(fileName, 1);
end
if length(S(1).data)==1
   reg='XX XXXX';
else
   reg=[];

   for i=indexText
      reg=[reg int2char(recognize(S(i).data, P, 't'))];
   end
   reg=[reg ' '];
   for i=indexNum
      reg=[reg int2char(recognize(S(i).data, P, 'n'))];
   end
end
```

ฟังก์ชันหาตำแหน่งป้ายทะเบียนรถยนต์

```
function [S indexText indexNum]=findReg(fileName, level)


RGB=imread(fileName);
GS=rgb2gray(RGB);
[sr sc]=size(GS);
GS=imresize(GS, [640 floor(640/sr*sc)]);
[sr sc]=size(GS);


if level==1
   GS=histeq(GS);
```

```
        end
imSize=sr*sc;
th=graythresh(GS);
if th>0.8
   th=0.75;
end
maxCentroid=0;
trueReg=0;
for t=th+0.2:-0.1:0.1
   BW = im2bw(GS, t);
   for nMed=1:1:20
      BW=medfilt2(BW);
   end

   L = bwlabel(BW);

   for i=1:1:max(max(L))
      [r c]=find(L==i);
      minR=min(r);
      maxR=max(r);
      minC=min(c);
      maxC=max(c);
      height=maxR-minR;
      wide=maxC-minC;
      squareSize=height*wide;
      if 1.5*height<wide & squareSize/imSize>0.01 & squareSize/imSize<0.16 & height>20
         GSES=GS([minR:maxR], [minC:maxC]);

         BWES = im2bw(GSES, graythresh(GSES(1:20,:)));
         tmpBWES=not(BWES);
         reg=[];
```

```
maxCentroid=0;
startNum=0;
for cutHead=1:1:15
    BWES=tmpBWES([cutHead:end], :);
    sres=height+2-cutHead;
    sces=wide+1;
    BWESSize=sres*sces;

    LES = bwlabel(BWES);

    centroidR=[];
    centroidC=[];
    for i=1:1:max(max(LES))
        [res ces]=find(LES==i);
        maxRES=max(res);
        textHeight=maxRES-min(res);
        textWide=max(ces)-min(ces);
        textSize=textHeight*textWide;

        if(textHeight>1*textWide & textHeight<4*textWide &
textSize/BWESSize>0.01 & textSize/BWESSize<0.15 & textHeight<sres*7/10) &
maxRES<2/3*(sres+2)
            centroidR=[centroidR (max(res)+min(res))/2];
            centroidC=[centroidC (max(ces)+min(ces))/2];
        else
            BWES(LES==i)=0;
        end
    end

    nCentroid=length(centroidC);
    far=[centroidC(2:end)-centroidC(1:end-1)];
    startNum=find(far==max(far))+1;
```

```
                if nCentroid>=3 & nCentroid<=6 & nCentroid>maxCentroid & startNum<=3
                    maxCentroid=nCentroid;
                    reg=BWES;
                    indexText=[1:startNum-1];
                    indexNum=[startNum:nCentroid];
                end
            end
        end
        if maxCentroid>=3
            break;
        end
    end
    if maxCentroid>=3
        trueReg=medfilt2(reg);

        S=struct('data', {});
        LREG = bwlabel(trueReg);
        for i=1:1:max(max(LREG))
            [rreg creg]=find(LREG==i);
            S(i).data=trueReg(min(rreg):max(rreg), min(creg):max(creg));
        end

        break;
    else
        S(1).data=[0];
        indexText=[];
        indexNum=[];
    end
end
```

ฟังก์ชันรู้จำหมายเลขทะเบียนรถยนต์

```
function ans=recognize(input, P, role)

[sr sc]=size(input);
BWProb=imProb(input, 8, 6);

[LI ELI]=findEndLine(input, 'thin');
headStatus=findHeadStatus(LI, ELI);
[reli celi]=find(ELI==1);
ans=[];

if role=='n';
   BW=[false(sr,1) input false(sr,1)];
   BW=[false(1,sc+2); BW; false(1,sc+2)];
   L = bwlabel(not(BW));
   nHole=max(max(L))-1;

    if sr>2.5*sc
       index=1;
    elseif nHole==2
       index=8;
    elseif nHole==1
       if 4.5*size(find(L==2), 1)>sr*sc
          index=0;
       else
         index=[4 6 9];
       end
    else
       index=[2 3 5 7];
    end
else
   index=[10:1:51];
```

```
end

minErr=Inf;

for k=index+1
    BWO=[];
    BWO=P(k).pattern;

    PP=imProb(BWO, 8, 6);
    error=[];
    if k==11
        if headStatus ~= 0
            error=Inf;
        end
    elseif k==12
        if sum(BWProb(1:4, 4))==0 | sum(BWProb(1, [1:3]))<0.025
            error=Inf;
        end
    elseif k==30
        if headStatus ~= 1
            error=Inf;
        end
    elseif k==31
        if sum(BWProb(4:8, 3))==0
            error=Inf;
        end
    elseif k==33
        if BWProb(8, 4)>0.01
            error=Inf;
        end
    elseif k==34
        if sum(BWProb(1:4, 4))>0
```

```
        error=Inf;
    end
elseif k==35
    if sum(BWProb(1, 1:3))>0
        error=Inf;
    end
elseif k==36
    if sum(BWProb(8, 3:4))>0.03
        error=Inf;
    end
elseif k==40
    if headStatus ~= -1
        error=Inf;
    end
elseif k==43
    if sum(sum(BWProb(6:8, 1:2)))>0
        error=Inf;
    end
elseif k==45
    if sum(BWProb(5:8, 1))>0
        error=Inf;
    end
elseif k==49
    if sum(BWProb(7:8, 4))>0
        error=Inf;
    end
elseif k==52
    if sum(BWProb(6:8, 1))==0
        error=Inf;
    end
end
```

```
        if length(error)==0
            error=sum(sum(abs(PP-BWProb)));
        end

        BWT=imresize(BWO, [sr sc]);
        [LL EL]=findEndLine(BWT, 'thin');
        [rel cel]=find(EL==1);

        err=compareError(reli, celi, rel, cel);
        err=(err+compareError(rel, cel, reli, celi))/sr/sc;
        err=20*err;

        endErr=err+error;

        if endErr<minErr
            minErr=endErr;
            ans=k-1;
            picAns=BWT;
            elAns=EL;
            errAns=err;
            errorAns=error;
        end
    end
```

ฟังก์ชันแทนแต่ละพื้นที่ด้วยเปอร์เซ็นต์ของจำนวนพิกเซลสีดำ

```
function ans=imProb(BW, m, n)
[sr sc]=size(BW);
if sr<m
    BW=imresize(BW, [m sc]);
    [sr sc]=size(BW);
end
if sc<n
```

```
  BW=imresize(BW, [sr n]);
  [sr sc]=size(BW);
end

BW=double(BW);

srRe=sr+(m-mod(sr, m));
scRe=sc+(n-mod(sc, n));
BW=imresize(BW, [srRe scRe]);
nRowPerWindow=srRe/m;
nColPerWindow=scRe/n;
probImRow=[];
probIm=[];
sumIm=sum(sum(BW));
for i=0:nRowPerWindow:srRe-nRowPerWindow
  ri=sum(BW([i+1:i+nRowPerWindow], :));
  probImRow=[probImRow; ri];
end

for j=0:nColPerWindow:scRe-nColPerWindow
  cj=sum(probImRow(:, [j+1:j+nColPerWindow]), 2);
  probIm=[probIm cj];
end
ans=probIm/sum(sum(BW));
```

ฟังก์ชันหาจุดปลายของตัวอักษร

```
function [BW2 BW3]=findEndLine(BW1, method)

BW2=bwmorph(BW1, method, Inf);

[sr sc]=size(BW2);
```

```
BW2=[false(sr, 1) BW2 false(sr, 1)];
BW2=[false(1, sc+2); BW2; false(1, sc+2)];
[sr sc]=size(BW2);
BW3=false(sr,sc);
for i=2:1:sr-1
  for j=2:1:sc-1
    if BW2(i,j)==1
      if sum(sum(BW2(i-1:i+1, j-1:j+1)))==2
        BW3(i,j)=1;
      end
    end
  end
end
BW2=BW2(2:end-1,2:end-1);
BW3=BW3(2:end-1,2:end-1);
```

ฟังก์ชันหาหัวเข้าหัวออกหรือไม่มีหัว

```
function headStatus=findHeadStatus(BW2, BW3)
[sr sc]=size(BW2);
headStatus=Inf;
[re ce]=find(BW3([floor(3/4*end):end], [1:floor(end/2)])==1);
[rm cm]=find(BW2([floor(end/2):floor(3/4*end)], [1:floor(end/2)])==1);
if length(ce)>0 & length(cm)>0
  headPoint=(ce(end)-mean(cm))/sc;
  if headPoint>1/23
    headStatus=1;
  elseif headPoint<-1/23
    headStatus=-1;
  else
    headStatus=0;
  end
end
```

ฟังก์ชันการเปรียบเทียบหาค่าความผิดพลาดของจุดปลาย

```
function err=compareError(reli, celi, rel, cel)

err=0;
for i=1:1:length(reli)
  minDist=Inf;
  for j=1:1:length(rel)
    dist=sqrt(power(reli(i)-rel(j), 2)+power(celi(i)-cel(j), 2));
    if dist<minDist
      minDist=dist;
    end
  end
  err=err+minDist;
end
```

ฟังก์ชันจัดเก็บข้อมูลเข้าสู่ฐานข้อมูล

```
function DBCarReg(conn, carNamePath, carNameList, regList)
for i=1:1:length(regList)
  carInfo=imfinfo([char(carNamePath) char(carNameList(i))]);
  dateTime=cellstr([datestr(carInfo.FileModDate, 'yyyy-mm-dd HH:MM:SS') '.0']);
  curs=exec(conn, ['SELECT Date_Time FROM Car_Table WHERE Car_Registration= '''
char(regList(i)) '''']);
  curs=fetch(curs);
  dummy=curs.data;
  isRepeated=false;
  for j=1:1:length(dummy)
    if isequal(dummy(j), dateTime)
      isRepeated=true;
      break;
    end
```

```
    end
    if ~isRepeated
        insert(conn, 'Car_Table', {'Car_Registration', 'Date_Time'}, {char(regList(i)), 
carInfo.FileModDate});
    end
end
```

ฟังก์ชันแปลงตัวเลขเป็นตัวอักษร

```
function ans=int2char(value)
if value>=0 & value<=9
    ans=char(48+value);
elseif value==10
    ans=char(3585);
elseif value==11
    ans=char(3586);
elseif value==12
    ans=char(3588);
elseif value>=13 & value<=42
    ans=char(3577+value);
elseif value==43
    ans=char(3621);
else
    ans=char(3579+value);
end
```

ประวัติผู้เขียนโครงงาน

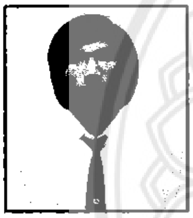
ชื่อ นายภาณุเดช วศินวรรธนะ

ภูมิลำเนา 81/27 ถ.บรมไตรโลกนารถ ต.ในเมือง อ.เมือง จ.พิษณุโลก 65000

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนพิษณุโลกพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยนเรศวร

E-mail: crosalot@hotmail.com



ชื่อ นายวัชรพงษ์ เจนพนัส

ภูมิลำเนา 448/26 ม.7 ต.หนองปลิง อ.เมือง จ.นครสวรรค์ 60000

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนโพฒิสารศึกษา
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยนเรศวร

E-mail: watcherro@hotmail.com