

การรู้จักของจังหวัดบนแผ่นป้ายทะเบียน
RECOGNITION OF PROVINCE NAMES ON LICENSE PLATES



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมกรรวิคและควบบคูน

คณระวิศวกรรมศาสตร

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียน

RECOGNITION OF PROVINCE NAMES ON LICENSE PLATES



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดและควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RECOGNITION OF PROVINCE NAMES ON LICENSE PLATES



WORAVIT

PANAPITAKKHUN

SIRICHAI

PHAOPONGSIL

SUKIT

POMLAKTHONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION

ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY

LADKRABANG

2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

หัวข้อปริญญาโท

การรู้จักชื่อจังหวัดบนแผ่นป้ายทะเบียน

Recognition of Province Names on License Plates

นักศึกษาผู้จัดทำ

นายวรวิษ หนาพิทักษ์กุล รหัสนักศึกษา 53011412

นายสิริชัย เผ่าพงษ์ศิลป์ รหัสนักศึกษา 53011701

นายสุกต ป้อมหลักทอง รหัสนักศึกษา 53011712

ปริญญา


วิศวกรรมศาสตรบัณฑิต

สาขาวิชา

วิศวกรรมการวัดคุม

ปีการศึกษา

2556

อาจารย์ผู้ควบคุมปริญญา	ลายมือชื่อ
รศ. ดร. เกษตร์ ศิริสันติสัมฤทธิ์	

หัวข้อวิทยานิพนธ์ การรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียน

RECOGNITION OF PROVINCE NAMES ON LICENSE PLATES

นักศึกษาผู้จัดทำ	นายวรวิษ	พนาพิทักษ์กุล
	นายสิริชัย	เผ่าพงษ์ศิลป์
	นายสุกฤต	ป้อมหลักทอง
อาจารย์ที่ปรึกษา	รศ. ดร. เกษตร์	ศิริสันติสัมฤทธิ์
ปีการศึกษา	2556	

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ได้นำเสนอทฤษฎีและหลักการพื้นฐานในการรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียน โดยหลักสำคัญคือการแยกส่วนของแผ่นป้ายทะเบียนออกจากภาพ และการแยกในส่วนของตัวอักษร ตัวเลข และชื่อจังหวัดออกจากแผ่นป้ายทะเบียน จากนั้นทำการหาขอบของตัวอักษรโดยวิธีโซเบล การหาแกนกลางของภาพโดยวิธี Skeleton การเก็บพิกัดจุดบนขอบของตัวอักษร การหมุนตัวอักษรและตัวเลขให้อยู่บนแกนเดียวกัน โดยการหา Eigen vector และ Eigen value การลดจำนวนของข้อมูลโดยวิธีการอินเตอร์โพลชัน การหาค่าสัมประสิทธิ์ของ DCT และการหาค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์ เพื่อแสดงความเป็นคุณลักษณะและเอกลักษณ์ของตัวอักษรและตัวเลขแต่ละตัว

จากผลการทดลองความถูกต้องของการแยกแผ่นป้ายทะเบียนออกจากภาพได้อย่างสมบูรณ์ คิดเป็นร้อยละ 86.36 ความถูกต้องของการรู้จำตัวอักษรและตัวเลขบนแผ่นป้ายทะเบียน คิดเป็นร้อยละ 92.41 และความถูกต้องของการรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียนคิดเป็นร้อยละ 92.36

Thesis Title	Recognition of Province Names on License Plates	
Authors	Woravit	Panapitakkhun
	Sirichai	Phaopongsil
	Sukit	Pomlakhong
Thesis Advisor	Assoc. Prof. Dr. Kaset Sirisantisamrid	
Year	2013	

ABSTRACT

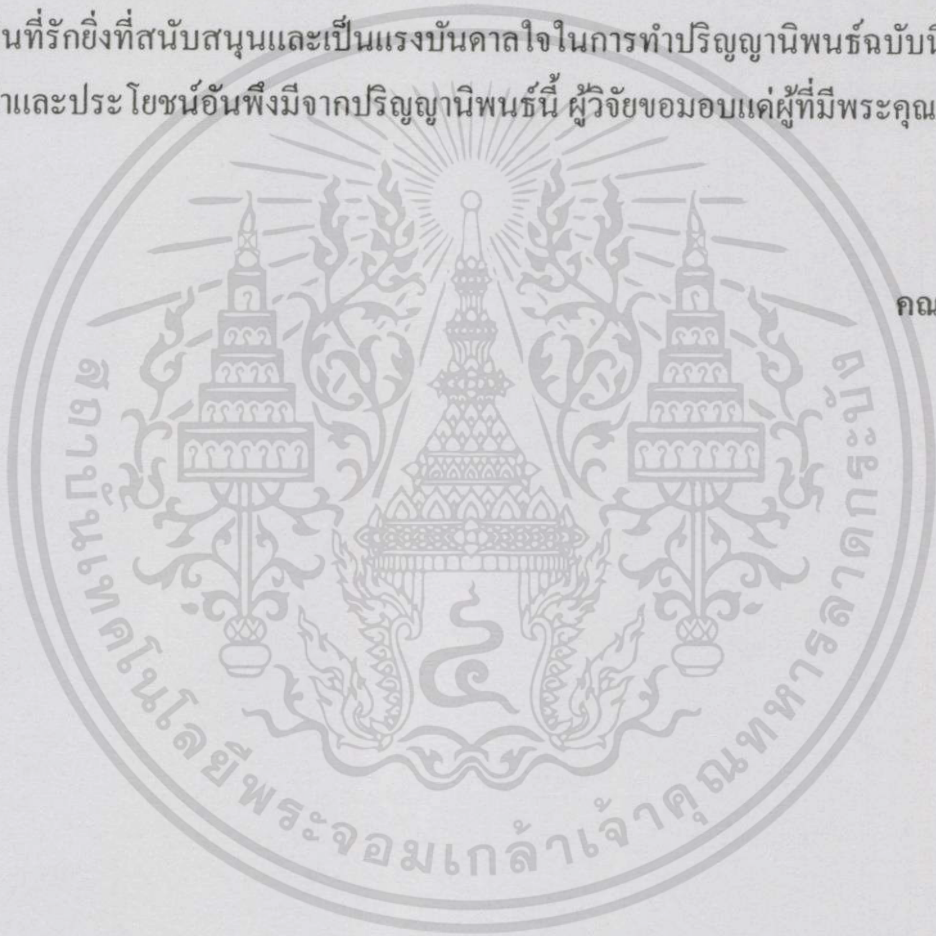
This thesis presents the basic theories and principles related to recognition of province names on license plates. The mainstream of this research consists of separating a license plate out from the images and segmentation the alphabets, numbers and province name from the license plate, edge detection by Sobel's operators, thinning of each alphabet and number, pick up the coordinate points on edge line to normalize each alphabet and number by computing Eigen vector and Eigen value, compute DCT's coefficients and finite impulse response $h(m)$ from DCT's coefficients for representation of each alphabet and number. The experimental results are shown that the proposed technique gives 86.36% of accuracy for recognition of license plate, 92.41% of accuracy for alphabet and number recognition and 92.36% of accuracy for province name's recognition.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะ รศ. ดร. เกษตร์ ศิริสันติสัมฤทธิ์ ที่ได้ให้ความเมตตาและคำแนะนำแก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเพื่ออุปกรณ์และเครื่องมือต่างๆ ในการทำปริญญาานิพนธ์นี้ ผู้วิจัยรู้สึกซาบซึ้ง และขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์สาขาวิศวกรรมการวัดคุมทุกท่านที่ได้ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญาานิพนธ์ฉบับนี้ และที่ลืมไม่ได้ คือ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ อันเป็นที่รักยิ่งที่สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญาานิพนธ์ฉบับนี้ คุณค่าและประโยชน์อันพึงมีจากปริญญาานิพนธ์นี้ ผู้วิจัยขอบอแด่ผู้ที่มีพระคุณทุกท่าน

คณะผู้จัดทำ



สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของปริญญาโท	3
1.3 ขอบเขตของปริญญาโท	3
1.4 ขั้นตอนการศึกษา	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีและหลักการ	5
2.1 โครงสร้างของระบบการรู้จำแผ่นป้ายทะเบียน	5
2.2 การประมวลผลภาพดิจิทัล	6
2.2.1 ภาพสี RGB	6
2.2.2 ภาพระดับสีเทา	7
2.2.3 ภาพสองระดับ	8
2.3 การหาขอบภาพ	10
2.3.1 การหาขอบภาพด้วยวิธี โซเบล	11
2.4 การหาแกนกลางของภาพ	12
2.5 การหาค่าแฉกพิกัด	14
2.6 การหา Eigen vector และ Eigen value	15
2.7 การอินเตอร์โพลชัน	16
2.8 การหาค่าสัมประสิทธิ์ในเทอมของโคไซน์	18
2.9 ระบบผลตอบสนองอิมพัลส์ที่จำกัดจำนวน	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 3 การแยกส่วนแผ่นป้ายทะเบียนออกจากภาพ	21
3.1 การค้นหาส่วนที่เป็นแผ่นป้ายทะเบียนในรูปภาพ	22
3.2 การแยกส่วนแผ่นป้ายทะเบียนออกจากรูปภาพ	25
บทที่ 4 การแยกและการเก็บตัวอักษรจากแผ่นป้ายทะเบียนรถยนต์	28
4.1 การแยกส่วนที่เป็นตัวอักษรออกจากแผ่นป้ายทะเบียนรถยนต์	28
4.2 การรวบรวมตัวอักษรเพื่อนำมาทำเป็นแม่แบบ จำนวน 49 ตัว	29
4.3 การวิเคราะห์ตัวอักษรแม่แบบโดยใช้วิธี FIR System	31
4.3.1 การหาค่า $h(m)$ โดยใช้วิธีการหาขอบภาพของตัวอักษร	32
4.3.2 การแสดงค่า $h(m)$ ในรูปแบบของกราฟ	39
4.4 การเก็บฐานข้อมูลตัวอักษรและตัวเลข	47
บทที่ 5 การรู้จำแผ่นป้ายทะเบียน	49
5.1 วิธีการเปรียบเทียบข้อมูลภาพจากแผ่นป้ายทะเบียนกับฐานข้อมูล	49
5.2 การรู้จำหมวดตัวอักษรและหมายเลขทะเบียน	49
5.3 การรู้จำชื่อจังหวัด	53
บทที่ 6 วิธีการเมนูของโปรแกรมโดยกราฟฟิคยูเซอร์อินเตอร์เฟส	57
6.1 GUI คืออะไร	57
6.2 การใช้ GUI ในการแสดงผลขั้นตอนและการรู้จำตัวอักษรจากแผ่นป้ายทะเบียน	63
บทที่ 7 ผลการทดลอง	69
7.1 ผลการทดลองการแยกแผ่นป้ายทะเบียนออกจากภาพ	69
7.2 ผลการทดลองการรู้จำตัวอักษรบนแผ่นป้ายทะเบียนโดยวิธีการเปรียบเทียบ กับแม่แบบ	69
บทที่ 8 ปัญหาที่เกิดขึ้นและข้อเสนอแนะ	74
8.1 ปัญหาที่เกิดขึ้นในระหว่างการทำโครงงาน	74
8.2 ข้อเสนอแนะ	74
บรรณานุกรม	75
ภาคผนวก	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
7.1 ผลการทดลองในการแยกแผ่นป้ายทะเบียนทั้งหมด 154 คัน	69
7.2 แสดงผลการรู้จำตัวอักษรที่อยู่บนแผ่นป้ายทะเบียนทั้ง 154 แผ่นป้าย	70
7.3 แสดงผลการรู้ชื่อจังหวัดที่อยู่บนแผ่นป้ายทะเบียนทั้ง 144 แผ่นป้าย	71



สารบัญภาพ

ภาพที่	หน้า
1.1 ประเภทของป้ายทะเบียนรถยนต์	1
2.1 โครงสร้างของระบบการรู้จำแผ่นป้ายทะเบียน	5
2.2 ภาพสี RGB	7
2.3 ภาพแสดงค่าระดับสีเทา	8
2.4 ภาพระดับสีเทา	8
2.5 ภาพการแสดงผลของข้อมูลที่มีค่า "0" และ "1"	9
2.6 ภาพแสดงการตัดเทรซโซลที่เหมาะสม	10
2.7 ภาพแสดงการตัดเทรซโซลที่ไม่เหมาะสม	10
2.8 หน้าต่างการหาขอบภาพด้วยวิธี Sobel ในแนวนอนและแนวตั้ง	11
2.9 หน้าต่างขนาด 3x3 เมื่อ p1 คือขอบของวัตถุ	12
2.10 ภาพแสดงการลบจุดภาพตามเงื่อนไขทั้ง 2 ขั้นตอนและแสดงภาพผลลัพธ์ของแกนกลาง	13
2.11 พิกัดของตัวเลข "5" ที่แสดงในรูป จุด (x_0, y_0) และ (x_1, y_1)	14
2.12 ทิศทาง 8 ทิศทางที่ตามเข็มนาฬิกา	14
2.13 การอินเตอร์โพลชันของข้อมูล	17
3.1 ภาพการจัดเก็บป้ายทะเบียน โดยใช้ Memory card เชื่อมต่อกับ Notebook	21
3.2 ภาพสี RGB ด้านหน้าของรถยนต์ที่มีส่วนแผ่นป้ายทะเบียนอยู่ด้วย	22
3.3 ภาพด้านหน้ารถยนต์ที่ถูกแปลงจากภาพสี RGB เป็นภาพระดับสีเทา	23
3.4 ภาพที่ได้หลังจากทำการปรับค่าเทรซโซลที่เหมาะสมแล้ว	23
3.5 ภาพที่ผ่านการแปลงจากภาพระดับสีเทาเป็นภาพสองระดับ	24
3.6 ภาพสองระดับที่ผ่านการลบกลุ่มพิกเซลสีขาวที่ติดกับขอบภาพออกไป	24
3.7 ภาพแสดงการ Label กลุ่มพิกเซลสีขาว	25
3.8 ภาพแสดงขอบเขตในการตัดส่วนที่เป็นแผ่นป้ายทะเบียนออกจากภาพ	25
3.9 ภาพแสดงส่วนที่เป็นแผ่นป้ายทะเบียนที่ได้จากภาพสี RGB	26
3.10 ภาพสองระดับของส่วนที่เป็นแผ่นป้ายทะเบียน	26
3.11 ภาพที่ได้หลังจากทำการสลับสีของพิกเซล (Inverse Binary)	26
3.12 ภาพการกำจัดกลุ่มพิกเซลสีขาวที่ติดกับขอบภาพ	27
3.13 ภาพแสดงตัวอักษรบนแผ่นป้ายทะเบียนที่มีความสมบูรณ์หลังจากผ่านคำสั่ง 'medfilt2'	27

สารบัญภาพ (ต่อ)

ภาพที่	หน้า	
4.1	ภาพแสดงส่วนที่เป็นแผ่นป้ายทะเบียนที่ปรับปรุงคุณภาพแล้ว	28
4.2	ภาพแสดงตัวอย่างในการกำหนดขอบเขตในการตัดตัวอักษรที่เพิ่มความห่างให้กับขอบเขตของตัวอักษร	29
4.3	ภาพแสดงตัวอักษรแต่ละตัวหลังผ่านคำสั่ง 'imcrop'	29
4.4	ภาพแสดงตัวอย่างของแผ่นป้ายทะเบียนที่มีความเหมาะสมในการนำตัวอักษรมาทำเป็นแม่แบบ	30
4.5	ชุดแม่แบบตัวอักษรทั้ง 49 ตัว ซึ่งประกอบไปด้วยพยัญชนะไทยจำนวน 39 ตัว และตัวเลขอารบิกจำนวน 10 ตัว	31
4.6	ตัวอย่างแม่แบบของตัวเลข "5" ในระดับสี่เทา	32
4.7	ภาพเลข "5" ที่ผ่านการหาขอบภาพด้วยวิธี โซเบล (Sobel)	32
4.8	ภาพผลลัพธ์ที่ได้จากการหาแกนกลาง (Skeleton)	33
4.9	ภาพแสดงการพล็อตจุดที่เก็บค่าตำแหน่งพิกัดที่ได้มาทับลงไปบน เส้นขอบ (Edge) เพื่อเช็คความถูกต้องของตำแหน่งพิกัดที่เก็บมาได้	33
4.10	ผลลัพธ์แสดงการ Interpolation ด้วยวิธี 'nearest'	35
4.11	ผลลัพธ์แสดงการ Interpolation ด้วยวิธี 'linear'	35
4.12	ผลลัพธ์แสดงการ Interpolation ด้วยวิธี 'spline'	35
4.13	ผลลัพธ์แสดงการ interpolation ด้วยวิธี 'cubic'	36
4.14	กราฟที่ผ่าน Discrete Cosine Transform (DCT)	37
4.15	กราฟผลลัพธ์ $h(m)$ ของเลข "5" ที่ได้จาก FIR System	38
4.16	ค่าสัมประสิทธิ์ของ FIR ($h(m)$) ของตัวเลขและตัวอักษรทั้งหมด	47
4.17	ตำแหน่งของข้อมูลของแต่ละตัวอักษรและตัวเลข	41
5.1	ตัวอย่างของกลุ่ม 7 ตัวอักษร	50
5.2	ตัวอย่างของกลุ่ม 6 ตัวอักษร	50
5.3	ตัวอย่างของกลุ่ม 5 ตัวอักษร	50
5.4	ตัวอย่างของกลุ่ม 4 ตัวอักษร	51
5.5	การแบ่งในการพิจารณาของกลุ่ม 7 ตัวอักษร	51
5.6	การแบ่งในการพิจารณาของกลุ่ม 6 ตัวอักษร	52
5.7	การแบ่งในการพิจารณาของกลุ่ม 5 ตัวอักษร	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
5.8 การแบ่งในการพิจารณาของกลุ่ม 4 ตัวอักษร	52
5.9 ตัวอย่างการกำหนดกลุ่มตัวอักษรที่ต้องการพิจารณา	55
5.10 การแสดงผลการรู้จำแผ่นป้ายทะเบียน	55
6.1 หน้าต่าง GUI	57
6.2 หน้าต่าง Push Button	58
6.3 หน้าต่าง Slider	58
6.4 หน้าต่าง Radio Button และ Check Box	59
6.5 หน้าต่าง Edit Text	59
6.6 หน้าต่าง Static Text	59
6.7 หน้าต่าง Pop-Up Menu	60
6.8 หน้าต่าง List Box	61
6.9 หน้าต่าง Toggle Button	62
6.10 หน้าต่าง Panel	62
6.11 หน้าต่าง Axes	62
6.12 รูปตัวอย่างการใช้ GUI	63
6.13 หน้าต่างแสดงการโหลดข้อมูลภาพ	63
6.14 หน้าต่างแสดงค่า error หรือ ไม่สามารถโหลด ไฟล์ภาพได้	64
6.15 หน้าต่างแสดงภาพเริ่มต้นหลังจากเรียกไฟล์ภาพจากคอมพิวเตอร์	64
6.16 ตัวอย่างหน้าต่าง GUI ในส่วนของหมายเลขทะเบียน	65
6.17 ตัวอย่างหน้าต่าง GUI ในส่วนของชื่อจังหวัด	66
6.18 หน้าต่างการแสดงผลในส่วนของชื่อจังหวัด	66
6.19 หน้าต่าง GUI หลังจากกดปุ่ม Reset	68
7.1 ตัวอย่างภาพถ่ายรถยนต์ที่สามารถค้นหาแผ่นป้ายทะเบียนได้	72
7.2 ภาพถ่ายรถยนต์ที่ไม่สามารถค้นหาแผ่นป้ายทะเบียนได้	73

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญาบัตร

แผ่นป้ายทะเบียนรถยนต์สามารถแบ่งประเภทการใช้งานได้จากสีของแผ่นป้ายทะเบียน และสีของตัวอักษร ดังเช่น

- ป้ายขาวอักษรดำ คือ ป้ายทะเบียนรถยนต์นั่งส่วนบุคคลไม่เกิน 7 ที่นั่ง
- ป้ายขาวอักษรฟ้า คือ ป้ายทะเบียนรถยนต์นั่งส่วนบุคคลเกิน 7 ที่นั่ง
- ป้ายขาวอักษรเขียว คือ ป้ายทะเบียนรถกระบะ
- ป้ายเหลืองอักษรเขียว คือ ป้ายทะเบียนรถตุ๊กตุ๊ก
- ป้ายขาวอักษรดำ คือ ป้ายทะเบียนรถยนต์ของคณะผู้แทนทางการทูต

คน 9999 กรุงเทพมหานคร	รถยนต์รับจ้างระหว่างจังหวัด
คน 9999 กรุงเทพมหานคร	รถยนต์รับจ้างบรรทุกคนโดยสารไม่เกิน 7 คน
คน 9999 กรุงเทพมหานคร	รถยนต์รับจ้างสามล้อ
คน 9999 กรุงเทพมหานคร	รถยนต์สี่ล้อเล็กรับจ้าง
คน 9999 กรุงเทพมหานคร	รถยนต์บริหารธุรกิจ รถยนต์บริการทัศนอาจร และ รถยนต์บริการให้เช่า
คน 9999 กรุงเทพมหานคร	รถยนต์ส่วนบุคคลเกิน 7 คน
คน 9999 กรุงเทพมหานคร	รถยนต์บรรทุกส่วนบุคคล
คน 9999 กรุงเทพมหานคร	รถยนต์สี่ล้อส่วนบุคคล
คน 9999 กรุงเทพมหานคร	รถยนต์สามล้อส่วนบุคคล
คน 9999 กรุงเทพมหานคร	รถพ่วง รถบดถนน รถแทรกเตอร์ และรถใช้งานเกษตรกรรม
US 0007	รถยนต์ของบุคคลในคณะผู้แทนทางการทูต และ รถยนต์ของบุคคล ในหน่วยงานพิเศษของสถานทูตคณะผู้แทนของกงสุลและองค์กรณ์ ระหว่างประเทศ
US 0007	

รูปที่ 1.1 ประเภทของป้ายทะเบียนรถยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปัจจุบันนี้คนไทยใช้รถยนต์เป็นจำนวนมาก และคาดว่าในอนาคตจะมีจำนวนการใช้รถยนต์เพิ่มขึ้น ถ้าหากเกิดเหตุการณ์ที่ไม่พึงประสงค์ที่เกี่ยวข้องกับรถยนต์ (อาทิเช่น การทำผิดกฎหมายจราจร, การเกิดอุบัติเหตุแล้วหลบหนี, การโจรกรรมรถยนต์ และการใช้รถยนต์ก่อเหตุร้ายใน 3 จังหวัดชายแดนภาคใต้) ทำให้การติดตามตัวผู้กระทำความผิดและรถยนต์ที่ใช้เป็นพาหนะยุ่งยากขึ้น ซึ่งการติดตามตัวผู้กระทำความผิดและรถยนต์ที่เป็นพาหนะนั้น ทำได้จากการตรวจสอบแผ่นป้ายทะเบียนรถยนต์ว่ามีหมวดตัวอักษรอะไร หมายเลขทะเบียนอะไร และเป็นจังหวัดใด

จากข้อความที่กล่าวไปข้างต้นทำให้รู้ว่า แผ่นป้ายทะเบียนรถยนต์มีความสำคัญในการใช้ติดตามผู้กระทำความผิดและรถยนต์ที่ใช้เป็นพาหนะ ด้วยเหตุนี้จึงเป็นที่มาของโครงการวิจัยในการเขียนโปรแกรมการรู้จำแผ่นป้ายทะเบียน

จากการศึกษาโครงการวิจัยของรุ่นพี่ชื่อ นายกิตตินันท์ ทองคล้าย และคณะ เรื่องการรู้จำตัวอักษรบนแผ่นป้ายทะเบียนรถยนต์ เป็นการศึกษาวิธีการแยกส่วนที่เป็นแผ่นป้ายทะเบียนออกจากรูปภาพ การแยกตัวอักษรและหมายเลขทะเบียนออกจากแผ่นป้ายทะเบียน และการทำ FIR System เป็นการวิเคราะห์หาความเป็นเอกลักษณ์ของตัวอักษรแต่ละตัวโดยใช้ค่าสัมประสิทธิ์ที่ได้จากการหา Discrete Cosine Transform (DCT) มาหาค่าความเป็นเอกลักษณ์ของตัวอักษรแต่ละตัว h (m) โดยในการหาค่าความเป็นเอกลักษณ์ของตัวอักษร h (m) นั้น มีวิธีในการหาได้ 2 แบบ โดยแบบแรกจะเป็นการหาค่าความเป็นเอกลักษณ์ของตัวอักษรแต่ละตัวโดยใช้วิธีการหาแกนกลาง (Skeleton) ของตัวอักษร ส่วนแบบที่สองจะเป็นการหาค่าความเป็นเอกลักษณ์โดยใช้วิธีการหาขอบ (Edge-Sobel) ของตัวอักษร ซึ่งจะเป็นการนำตัวอักษรมาหาขอบเพื่อลดข้อมูลในการประมวลผล แต่ยังคงเอกลักษณ์ของตัวอักษรแต่ละตัวไว้ เฉพาะส่วนของหมวดตัวอักษรและหมายเลขทะเบียนของป้ายทะเบียนเท่านั้น โดยใช้วิธีการเก็บรวบรวมค่าสัมประสิทธิ์ความเป็นเอกลักษณ์ h (m) และทำการวิเคราะห์ตัวอักษรแม่แบบโดยใช้วิธี FIR System ของตัวอักษรและตัวเลขที่มีความสมบูรณ์ทั้ง 42 ตัว แต่เนื่องจากงานวิจัยดังกล่าวข้างต้นได้ทดสอบกับภาพแผ่นป้ายทะเบียนน้อยเกินไป ทำให้ไม่สามารถพิสูจน์ว่าวิธีการที่นำเสนอใช้ได้จริงหรือไม่ รวมทั้งไม่ได้ทำการรู้จำชื่อจังหวัดอีกด้วย อย่างไรก็ตาม ปัจจุบันนี้กรมการขนส่งทางบกได้ออกป้ายทะเบียนรุ่นใหม่ซึ่งลักษณะตัวอักษรมีการเปลี่ยนแปลงไปจากเดิม ซึ่งแตกต่างจากป้ายทะเบียนรุ่นเก่า ถ้าหากนำไปใช้อาจจะทำให้ไม่มีประสิทธิภาพเท่าที่ควร ดังนั้นปริญญาานิพนธ์ฉบับนี้ เป็นการศึกษาค้นคว้าวิจัยการรู้จำตัวอักษรบนแผ่นป้ายทะเบียนโดยหาวิธีการใหม่ที่เหมาะสมกับรู้จำตัวอักษรเพื่อให้สามารถรู้จำได้ทั้งชื่อจังหวัด หมวดตัวอักษร และตัวเลขทั้งหมดที่อยู่บนแผ่นป้ายทะเบียน รวมทั้งทดสอบกับภาพแผ่นป้ายทะเบียนที่มากพอสมควรซึ่งมีทั้งแผ่นป้ายทะเบียนรุ่นเก่าและรุ่นใหม่ และถ้างานวิจัยสามารถบรรลุผลสำเร็จตามที่ตั้งเป้าหมายไว้ ก็สามารถนำไปประยุกต์ใช้งานได้หลายประเภท ดังเช่น นำมาประยุกต์ใช้ในที่จอดรถอัตโนมัติเพื่อระบุเวลา เข้า-ออกของรถยนต์และคำนวณค่าที่จอดรถอัตโนมัติ แทนการแลกบัตร หรือสามารถนำมาประยุกต์ใช้ตามสี่แยกหรือสามแยกต่างๆ ที่มีสัญญาณไฟจราจรหรือด่านเก็บค่าทางด่วน เพื่อช่วยในการตรวจจับรถยนต์ที่ฝ่าไฟแดงหรือช่วยในการค้นหารถยนต์ที่คาดว่าถูกโจรกรรมมา เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปริญาานิพนธ์

1. ศึกษาพื้นฐานและทฤษฎีต่างๆที่เกี่ยวข้องกับการรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียน
2. นำพื้นฐานและทฤษฎีต่างๆ มาประยุกต์ใช้ในการแยกส่วนที่เป็นแผ่นป้ายทะเบียนออกจากรูปภาพ
3. เก็บรวบรวมค่าสัมประสิทธิ์ความเป็นเอกลักษณ์ h (m) และทำการวิเคราะห์ตัวอักษรแม่แบบโดยใช้วิธี FIR system ของตัวอักษรและตัวเลขที่มีความสมบูรณ์ทั้ง 49 ตัว ซึ่งเก็บไว้เป็นฐานข้อมูลที่ใช้ในการเปรียบเทียบกับตัวอักษรที่ต้องการ

1.3 ขอบเขตของปริญาานิพนธ์

1. สามารถแยกส่วนที่เป็นแผ่นป้ายทะเบียนออกจากภาพได้
2. สามารถเก็บรวบรวมแม่แบบทั้ง 49 ตัว มาวิเคราะห์แยก เพื่อหาค่าสัมประสิทธิ์ความเป็นเอกลักษณ์ h (m) ของแต่ละตัวอักษรและตัวเลข เพื่อเก็บไว้เป็นฐานข้อมูลในการเปรียบเทียบกับตัวอักษรที่ต้องการพิสูจน์
3. สามารถระบุได้ว่าแผ่นป้ายทะเบียนในภาพ ที่ต้องการพิสูจน์นั้นเป็นหมวดตัวอักษรตัวใด หมายเลขใดและเป็นชื่อจังหวัดใด โดยการเปรียบเทียบค่าสัมประสิทธิ์ความเป็นเอกลักษณ์ h (m) ที่ได้กับตัวอักษรแม่แบบที่เก็บไว้ในฐานข้อมูล

1.4 ขั้นตอนการศึกษา

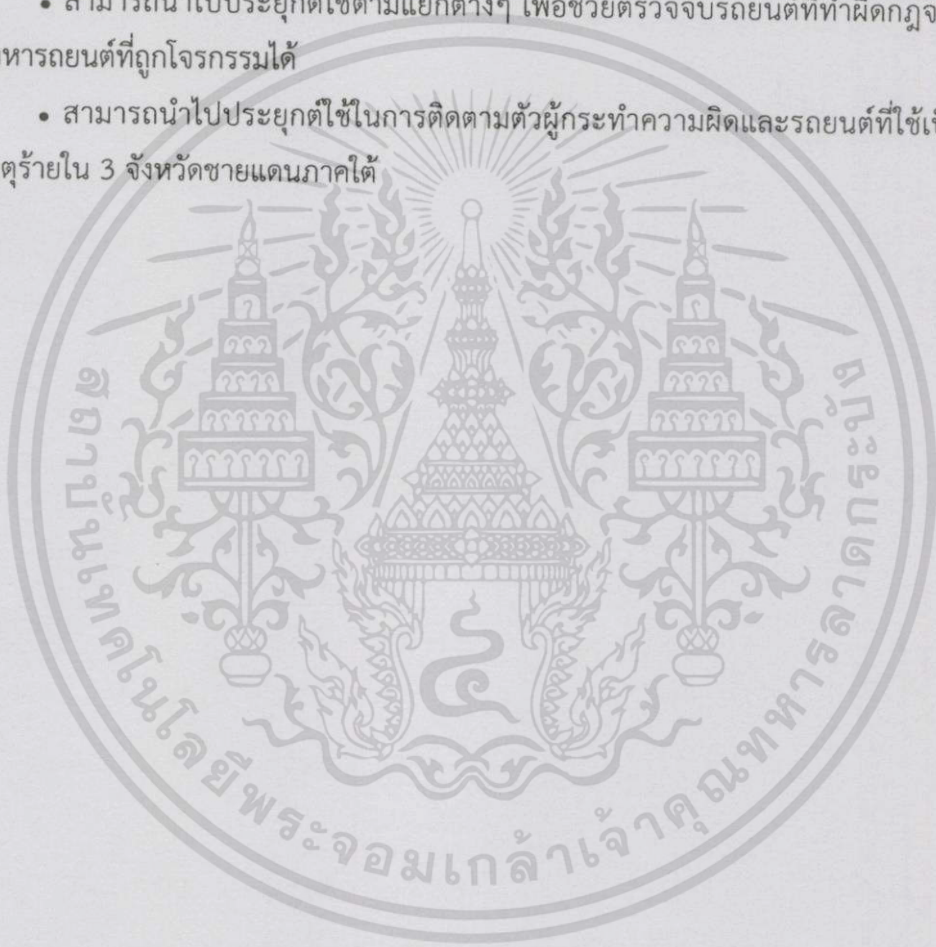
1. ศึกษาความเป็นมาและความสำคัญของปัญหา สมมติฐานของการศึกษา ทฤษฎีที่เกี่ยวข้อง และแนวคิดในการทำปริญาานิพนธ์
2. ศึกษาพื้นฐานแนวทาง และทฤษฎีที่เกี่ยวข้องกับการรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียน
3. ศึกษาหลักการและวิธีต่างๆ ที่ใช้ในการแยกส่วนที่เป็นแผ่นป้ายทะเบียนออกจากภาพสี RGB
4. ศึกษาหลักการและทฤษฎีต่างๆ ที่เกี่ยวข้องกับการวิเคราะห์และแยกตัวอักษร
5. เก็บรวบรวมข้อมูลตัวอักษรและตัวเลขให้ครบ 49 ตัวของป้ายทะเบียนรุ่นเก่าและรุ่นใหม่ เพื่อใช้เป็นแม่แบบ
6. วิเคราะห์และตรวจหาเอกลักษณ์ของตัวอักษรและตัวเลขทั้ง 49 ตัวของป้ายทะเบียนรุ่นเก่าและรุ่นใหม่เพื่อหาความแตกต่างของตัวอักษรและตัวเลขแต่ละตัว
7. ทดลองเปรียบเทียบตัวอักษรและตัวเลขที่เก็บค่าได้กับตัวอักษรแม่แบบ
8. ทดลองการรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียนเพื่อระบุว่ารถยนต์ในภาพคือจังหวัดใด
9. แสดงผลการทดลองทั้งหมดและคำนวณค่าความผิดพลาดจากผลการทดลอง
10. สรุปผลการทำงาน ปัญหาที่พบ และข้อเสนอแนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

งานวิจัยฉบับนี้เป็นเพียงแนวคิดเบื้องต้น จะทำการเก็บข้อมูลไว้ในฐานข้อมูล และนำข้อมูลดังกล่าวมาเปรียบเทียบเพื่อหาความถูกต้องว่ามีมากน้อยเพียงใด ถ้าหากข้อมูลมีความถูกต้องน้อยเกินไป ก็ให้ย้อนกลับมาดูวิธีการรู้จำชื่อจังหวัดและทำการแก้ไขวิธีการรู้จำชื่อจังหวัดใหม่ แต่ถ้าหากข้อมูลมีความถูกต้องมากพอก็จะนำไปประยุกต์ใช้ในอนาคตได้ อาทิเช่น

- สามารถนำไปประยุกต์ใช้กับที่จอดรถยนต์อัตโนมัติเพื่อตรวจสอบเวลา เข้า-ออก และคำนวณค่าจอดรถได้
- สามารถนำไปประยุกต์ใช้ตามแยกต่างๆ เพื่อช่วยตรวจจับรถยนต์ที่ทำผิดกฎจราจร หรือตรวจหารถยนต์ที่ถูกโจรกรรมได้
- สามารถนำไปประยุกต์ใช้ในการติดตามตัวผู้กระทำความผิดและรถยนต์ที่ใช้เป็นพาหนะก่อเหตุร้ายใน 3 จังหวัดชายแดนภาคใต้

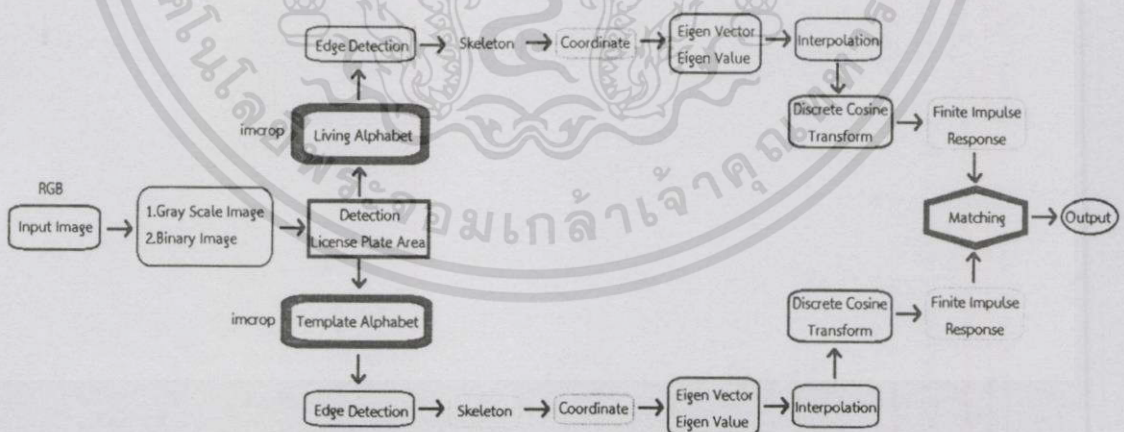


บทที่ 2 ทฤษฎีและหลักการ

ในบทนี้กล่าวถึงโครงสร้างของระบบการรู้จำแผ่นป้ายทะเบียนที่นำเสนอ รวมทั้งทฤษฎีและหลักการต่างๆ ที่เกี่ยวข้องกับการรู้จำตัวแผ่นป้ายทะเบียน ซึ่งทฤษฎีและหลักการเหล่านี้จะเริ่มจากพื้นฐานของภาพ การหาขอบตัวอักษร และอื่นๆ จนกระทั่งถึงระบบผลตอบสนองอิมพัลส์ที่จำกัดจำนวนรายละเอียดจะเป็นดังต่อไปนี้

2.1 โครงสร้างของระบบการรู้จำแผ่นป้ายทะเบียน

โครงสร้างของระบบการรู้จำแผ่นป้ายทะเบียน เริ่มต้นด้วยการถ่ายภาพด้านหน้าของรถยนต์ บริเวณที่มีแผ่นป้ายทะเบียนซึ่งเป็นภาพสี RGB จากนั้นแปลงให้เป็นภาพระดับสีเทา (Grey Scale Image) แล้วตัดค่า Threshold เพื่อแปลงให้เป็นภาพ 2 ระดับ (Binary Image) หลังจากนั้นค้นหาพื้นที่ที่คาดว่าจะ เป็นแผ่นป้ายทะเบียน ต่อมาทำการแยกตัวอักษรและตัวเลขออกจากแผ่นป้ายทะเบียนแล้วนำมาเข้าสู่ กระบวนการหาขอบของภาพ (Edge Detection) หาแกนกลางของภาพ (Skeleton) การเก็บพิกัดจุด (Coordinate Detection) การหมุนภาพให้อยู่กึ่งกลาง (Normalization) การแทรกพิกัดจุด (Interpolation) การหาค่าสัมประสิทธิ์ของ DCT (Discrete Cosine Transform Coefficients) การหาค่าสัมประสิทธิ์ FIR (Finite Impulse Response Coefficients) เพื่อใช้แทนตัวเลขและตัวอักษร หลังจากนั้นนำมาเปรียบเทียบกับฐานข้อมูล หากสามารถเปรียบเทียบเข้าคู่กันได้ (Matching) จะแสดงผลเป็น ตัวเลขหรือตัวอักษรนั้น



รูปที่ 2.1 โครงสร้างของระบบการรู้จำแผ่นป้ายทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การประมวลผลภาพดิจิทัล (Digital Image Processing)

ภาพดิจิทัล (Digital Image) หมายถึง ภาพที่เก็บอยู่ในรูปแบบดิจิทัล ซึ่งภาพที่มนุษย์มองเห็นด้วยสายตาโดยทั่วไปนั้นเป็นภาพในลักษณะของสามมิติ คือ มีมิติของความกว้าง ความยาว และความลึก ภาพดิจิทัลนี้เป็นการแสดงผลในลักษณะของฟังก์ชันสองมิติ $f(x,y)$ โดยที่ x และ y เป็นพิกัดของภาพและ f ที่พิกัด (x,y) ใดๆ ภายในภาพ คือค่า ความเข้มแสงของภาพ (Intensity) ที่ตำแหน่งนั้นๆ และเมื่อ (x,y) และแอมพลิจูดของ f เป็นค่าจำกัด (Finite Value) จึงเรียกรูปภาพนั้นๆ ว่าเป็นภาพดิจิทัล (Digital Image) และถ้ากำหนดให้ภาพ $f(x,y)$ มีขนาด M แถว และ N คอลัมน์ และพิกัดของจุดกำเนิด (Origin) ของภาพ คือ ที่ตำแหน่ง $(x,y) = (0,0)$ แล้วจะสามารถเขียนสมการให้อยู่ในรูปแมทริกซ์ ได้ดังนี้

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \dots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

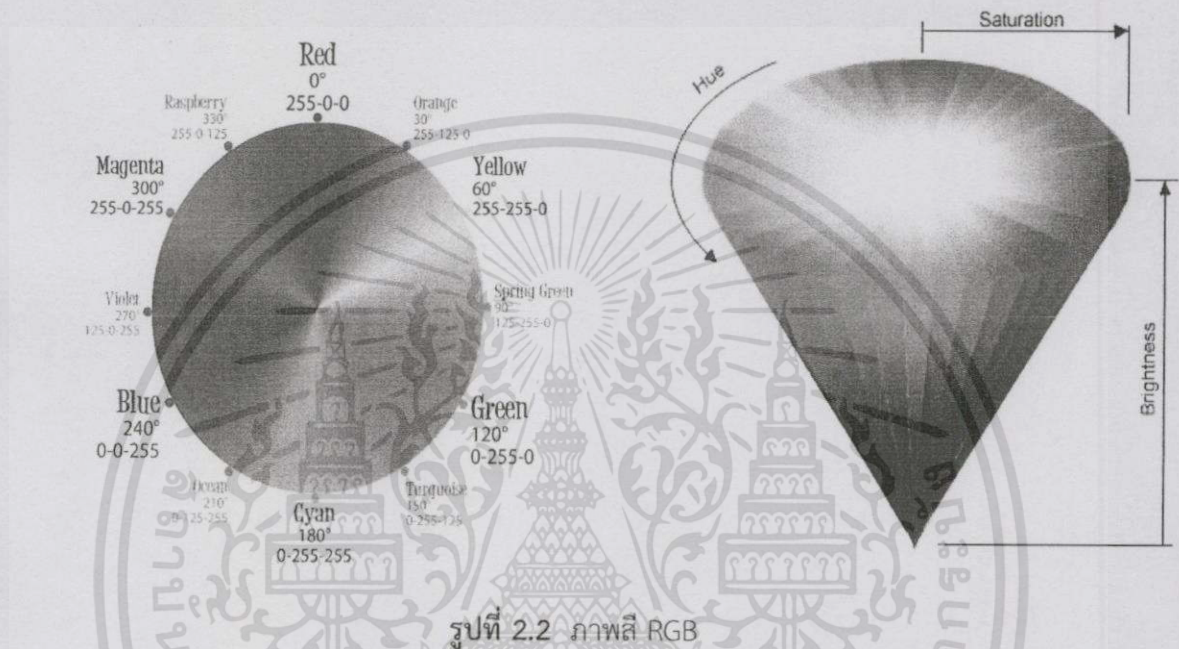
ค่าแต่ละค่าที่อยู่ในแมทริกซ์จะเรียกว่า พิกเซล (Pixel) โดยตำแหน่ง $(0,0)$ จะอยู่ทางด้านซ้ายมือสุดและด้านบนสุดของภาพ การจัดลำดับตำแหน่งของจุดภาพจะเรียงจากซ้ายไปขวาและบนลงล่าง การสร้างภาพดิจิทัลสามารถสร้างได้จากอุปกรณ์รับภาพ เช่น กล้องดิจิทัล (Digital Cameras) เครื่องแสกนภาพ (Scanners) เป็นต้น ภาพดิจิทัลยังสามารถสร้างโดยการสังเคราะห์จากสิ่งที่ไม่ใช่ข้อมูลภาพ เช่น ฟังก์ชันทางคณิตศาสตร์ หรือแบบจำลองเรขาคณิตแบบสามมิติ ซึ่งการสร้างภาพลักษณะนี้เป็นส่วนหนึ่งในงานด้านคอมพิวเตอร์กราฟิก (Computer Graphics)

การประมวลผลภาพ (Image Processing) เป็นการใช้ขั้นตอนหรือกระบวนการต่างๆ มากระทำกับภาพดิจิทัล โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณภาพของภาพให้ได้ภาพที่มีคุณสมบัติตามที่ต้องการ เพื่อการรู้จำวัตถุสิ่งของ และแผ่นป้ายทะเบียน เป็นต้น ตัวอย่างของการปรับปรุงคุณภาพของภาพเช่น การทำให้ภาพมีความคมชัดมากขึ้น มีความสว่างมากขึ้นหรือมืดลง ทำให้ภาพมีสีส้มมากยิ่งขึ้น ย่อ-ขยายภาพเพื่อประหยัดพื้นที่ในการเก็บข้อมูล เป็นต้น โดยชนิดของข้อมูลภาพที่ใช้ในการประมวลผลภาพดิจิทัลจะประกอบได้ดังนี้

2.2.1 ภาพสี RGB (RGB Image)

สีของรูปภาพต่างๆ บนจอคอมพิวเตอร์ที่ใช้กันอยู่ในปัจจุบันเกิดจากการผสมของแม่สีทางแสง 3 สี คือ แสงสีแดง (Red) แสงสีเขียว (Green) และแสงสีน้ำเงิน (Blue) เรียกกันโดยทั่วไปว่า "RGB" ซึ่งสีอื่นๆ ที่ต่างไปจากนี้เกิดจากการผสมสีของแม่สีทั้งสาม เช่น แสงสีแดงผสมกับแสงสีเหลืองได้แสง

สีส้ม แสงสีแดงผสมกับแสงสีน้ำเงินได้แสงสีม่วง หรือหากเอาแม่สีทั้งสามมาผสมกันก็จะได้แสงสีขาว โดยระบบสีทางแสงแบบ 8 บิต จะเป็นการใช้หน่วยความจำดิจิทัลขนาด 8 บิต มาแทนเฉดสีแต่ละสี ในเชิงคณิตศาสตร์ ซึ่งในแต่ละบิตมีค่าเป็น "0" กับ "1" ซึ่งแปลว่าจะมีค่าตัวเลขที่เป็นไปได้จำนวน 2 ยกกำลัง 8 (= 256) หรือตั้งแต่ "00000000" ฐาน 2" (มีค่าเท่ากับ 0 ในเลขฐาน 10) จนถึง "11111111" ฐาน 2" (มีค่าเท่ากับ 255 ในเลขฐาน 10) ซึ่งหมายความว่าจะมีเฉดสีต่างๆ กันในสีใดสีหนึ่งได้ 256 เฉดสี



รูปที่ 2.2 ภาพสี RGB

2.2.2 ภาพระดับสีเทา (Gray Scale Image)

เนื่องจากภาพสี RGB เกิดจากการผสมกันของแสงสีแดง สีเขียว และสีน้ำเงิน ในอัตราส่วนที่แตกต่างกันในแต่ละพิกเซล ทำให้เกิดเป็นภาพสี จึงเป็นเรื่องยากที่จะนำภาพสีมาทำการประมวลผล ดังนั้น เพื่อให้ง่ายต่อการคำนวณจึงควรทำภาพสีให้เป็นภาพในระดับสีเทาก่อน ซึ่งภาพระดับสีเทานั้น พิกเซลแต่ละพิกเซลจะมีค่าความเข้มสีเป็นค่าเฉลี่ยของแม่สีทั้ง 3 ในพิกเซลนั้น การเฉลี่ยค่าแม่สีทั้ง 3 ใน 1 พิกเซล เขียนเป็นสมการได้ดังนี้

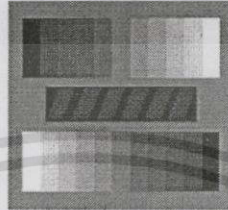
$$\text{ค่าระดับสีเทา} = [(0.299 \times R) + (0.587 \times G) + (0.114 \times B)] \quad (2.1)$$

R คือ ค่าอินพุตพิกเซลสีแดง

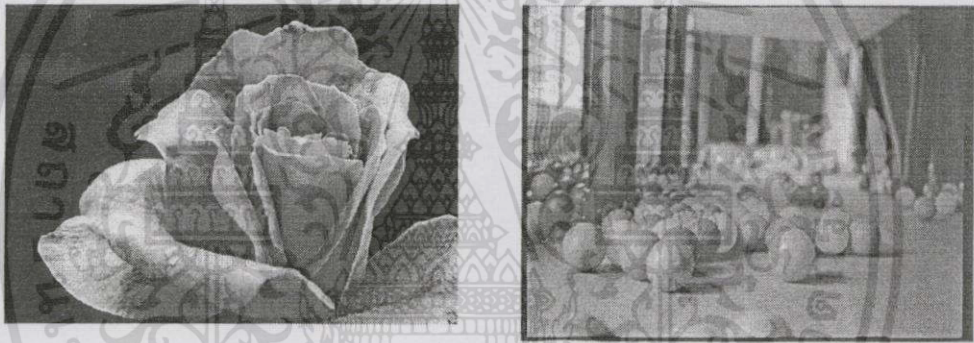
G คือ ค่าอินพุตพิกเซลสีเขียว

B คือ ค่าอินพุตพิกเซลสีน้ำเงิน

จากสมการที่ (2.1) จะเห็นได้ว่าค่าระดับสีเทาจะมีค่าตั้งแต่ 0 ซึ่งจะแสดงผลเป็นสีดำ จนกระทั่งถึง 255 ซึ่งจะแสดงผลเป็นสีขาว พิกเซลที่มีค่าระดับสีเทาอยู่ในช่วง 0 ถึง 255 จะแสดงผลตามค่าระดับสีเทา โดยจะไล่จากสีดำซึ่งมีค่าในระดับสีเทาเท่ากับ 0 และจะมีสีเทาจางลงเรื่อยๆ เมื่อค่าระดับสีเทาเพิ่มขึ้น จนกระทั่งเป็นสีขาวเมื่อค่าระดับสีเทาเท่ากับ 255 ดังแสดงในรูปที่ 2.3 และตัวอย่างภาพระดับสีเทาแสดงดังรูปที่ 2.4



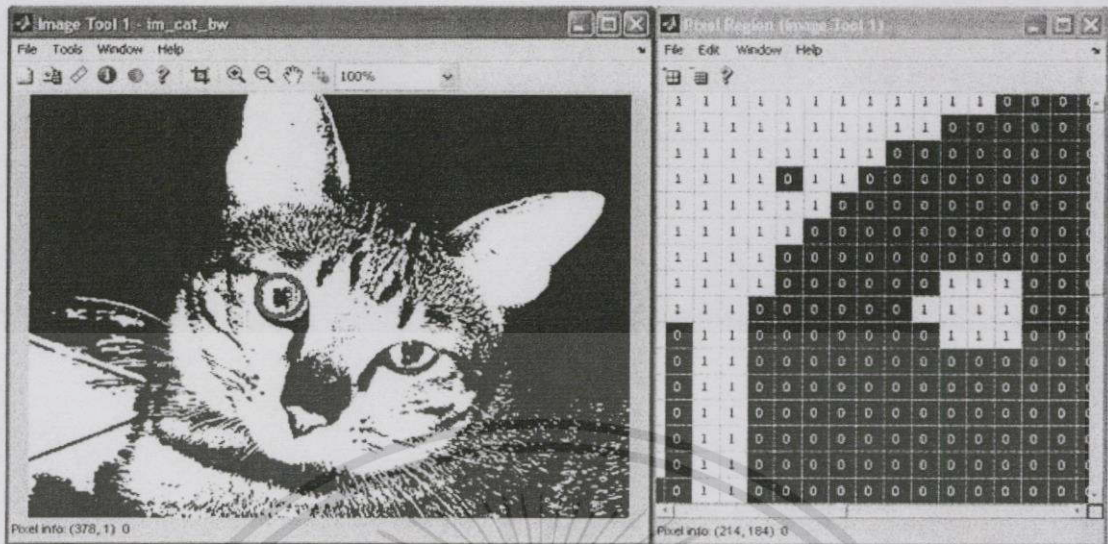
รูปที่ 2.3 ภาพแสดงค่าระดับสีเทา



รูปที่ 2.4 ภาพระดับสีเทา

2.2.3 ภาพสองระดับ (Binary Image)

ภาพสองระดับ หมายถึง ภาพที่มีระดับความเข้มของสีเพียง 2 ระดับ คือแต่ละพิกเซลจะมีค่าเพียงสองค่าคือ 0 และ 1 โดยพิกเซลที่มีค่าเป็น 0 จะแสดงผลเป็นสีดำ และพิกเซลที่มีค่าเป็น 1 จะแสดงผลเป็นสีขาวดังรูปที่ 2.5 (ก) เมื่อพิจารณาที่วงกลมสีแดง โดยทำการขยายและแสดงในรูปของข้อมูลจะแสดงดังรูปที่ 2.5 (ข) การประมวลผลภาพดิจิทัลโดยทั่วไปนั้นจะนิยมแปลงภาพสี RGB ให้เป็นภาพระดับสีเทาก่อน จากนั้นจึงแปลงข้อมูลภาพระดับสีเทาให้เป็นภาพสองระดับเพื่อลดจำนวนข้อมูลในภาพลง ทำให้ประมวลผลได้เร็วยิ่งขึ้นและยังลดเนื้อที่เก็บข้อมูลภาพลงอีกด้วย



(ก)

(ข)

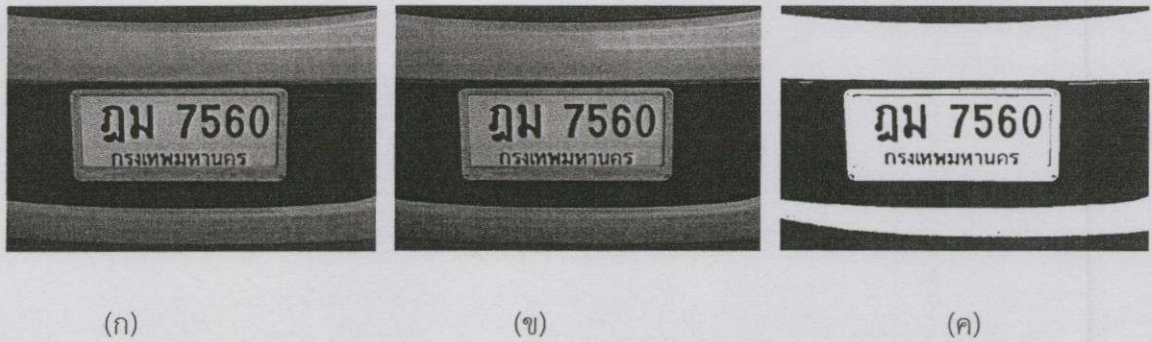
รูปที่ 2.5 ภาพการแสดงผลของข้อมูลที่มีค่า “0” และ “1”

(ก) ภาพสองระดับ

(ข) ภาพทำการแสดงในส่วนข้อมูล

การสร้างภาพสองระดับนั้นสามารถทำได้โดยใช้เทคนิคการตัดค่าเทรชโฮล (Threshold Techniques) จากภาพระดับสีเทา โดยค่าเทรชโฮลจะเป็นค่าใดก็ได้ที่อยู่ในช่วง 0 ถึง 255 เช่น การกำหนดค่าเทรชโฮลเท่ากับ 200 ซึ่งเป็นค่าๆ หนึ่งของภาพระดับสีเทา พิกเซลที่มีค่าระดับสีเทามากกว่า 200 จะเปลี่ยนจากสีเทาเป็นสีขาว ส่วนพิกเซลที่มีค่าระดับสีเทาน้อยกว่า 200 จะเปลี่ยนจากสีเทาเป็นสีดำ ซึ่งในการพิจารณาว่าพิกเซลไหนของข้อมูลภาพควรจะเป็นพิกเซลสีขาวหรือสีดำนั้นขึ้นอยู่กับความต้องการว่าต้องการภาพส่วนใดมาประมวลผล รูปที่ 2.6 แสดงการตัดเทรชโฮลที่เหมาะสม เพื่อหาบริเวณหรือพื้นที่ๆ เป็นแผ่นป้ายทะเบียนรถยนต์ในภาพ จึงต้องทำให้ส่วนที่เป็นแผ่นป้ายทะเบียนเป็นสีขาวหรือมีค่าเป็น 1 และพยายามให้ส่วนอื่นๆ ไม่ใช่แผ่นป้ายทะเบียนเป็นสีดำหรือมีค่าเป็น 0 ให้มากที่สุด รูปที่ 2.7 แสดงผลของการเลือกใช้ค่าเทรชโฮลที่ไม่เหมาะสม ซึ่งจะทำให้เห็นบริเวณหรือพื้นที่ที่เป็นแผ่นป้ายทะเบียนไม่ชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ภาพแสดงการตัดเทรซโฮลที่เหมาะสม

(ก) ภาพสี RGB

(ข) ภาพระดับสีเทา

(ค) ภาพสองระดับที่ได้จากค่าเทรซโฮลเท่ากับ 80



(ก)

(ข)

รูปที่ 2.7 ภาพแสดงการตัดเทรซโฮลที่ไม่เหมาะสม

(ก) ภาพสองระดับที่ได้จากค่าเทรซโฮล (เท่ากับ 25) น้อยเกินไป

(ข) ภาพสองระดับที่ได้จากค่าเทรซโฮล (เท่ากับ 160) มากเกินไป

2.3 การหาขอบภาพ (Edge Detection)

การหาขอบภาพเป็นการหาขอบของวัตถุหรือตัวอักษรและตัวเลขที่อยู่ในภาพ ซึ่งจะเป็นการลดจำนวนของข้อมูลลงแต่จะยังคงความเป็นเอกลักษณ์ของวัตถุหรือตัวอักษรเอาไว้ การหาขอบภาพคือการตรวจสอบว่าเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใดในภาพ โดยวัดจากการเปลี่ยนแปลงความเข้มในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว การหาขอบภาพมีหลายวิธี เช่น Spatial Filter หรือ Morphological อย่างไรก็ตาม ในการหาขอบภาพจะกระทำกับภาพสีเทา (Gray Level Images) ที่มีค่าอยู่ระหว่าง 0-255 ซึ่งวิธีที่โครงการนี้เลือกใช้ คือ วิธีการหาขอบภาพโดย Sobel's Operators

2.3.1 การหาขอบภาพด้วยวิธีโซเบล (Sobel's Operators)

การหาขอบภาพโดย Sobel's Operators เป็นการหาเส้นขอบของวัตถุหรือตัวอักษรโดยใช้หน้าต่างขนาด 3×3 วางทับกับข้อมูลภาพ 2 มิติ $f(x, y)$ ที่ตำแหน่ง (x, y) แล้วทำการ Convolution คำนวณในแต่ละพิกเซลของหน้าต่างเข้ากับข้อมูลภาพ อย่างไรก็ตาม การหาขอบภาพโดย Sobel's Operators จะใช้หน้าต่างขนาด 3×3 จำนวน 2 หน้าต่าง ดังแสดงในรูปที่ 2.8

-1	-2	-1
0	0	0
+1	+2	+1

+1	0	-1
+2	0	-2
+1	0	-1

(ก) (ข)

รูปที่ 2.8 หน้าต่างการหาขอบภาพด้วยวิธี Sobel ในแนวนอนและแนวตั้ง

(ก) หน้าต่างสำหรับหาขอบภาพแนวนอน

(ข) หน้าต่างสำหรับหาขอบภาพแนวตั้ง

เนื่องจากการหาขอบภาพโดย Sobel's Operators เป็นการหาขอบภาพทั้งแนวตั้งและแนวนอน ดังนั้นในตอนแรก เราจะวางหน้าต่างในแนวนอนที่ด้านบนซ้ายของข้อมูลภาพ $f(x, y)$ จากนั้นทำการ Convolution หน้าหนักที่อยู่ในหน้าต่างกับข้อมูลภาพ 2 มิติ ผลลัพธ์ที่ได้จะเก็บไว้ในภาพเอ๊าท์พุท $G_x(x, y)$ เมื่อได้ผลลัพธ์แล้วหน้าต่างจะเลื่อนไปทางขวาหนึ่งพิกเซลและทำการ Convolution อีกเพื่อให้ได้ผลลัพธ์และเก็บไว้ใน $G_x(x, y)$ โปรแกรมจะดำเนินการเช่นนี้ไปเรื่อยๆ จากซ้ายไปขวาและจากบนลงล่างจนครบทั้งภาพ ซึ่งเราจะได้ขอบภาพทั้งแนวนอน ในทำนองเดียวกัน เราใช้หน้าต่างแนวตั้งทำการ Convolution กับข้อมูลภาพ $f(x, y)$ จนครบทั้งภาพก็จะได้ขอบภาพแนวตั้งทั้งหมดที่เก็บไว้ในตัวแปร $G_y(x, y)$

เมื่อได้ขอบภาพในแนวนอน $G_x(x, y)$ และแนวตั้ง $G_y(x, y)$ แล้ว จากนั้นทำการหาขอบภาพทั้งหมดด้วยการคำนวณหาขนาดของ Gradient (∇f) โดยใช้สมการที่ (2.2) หรือ (2.3)

$$\nabla f = \sqrt{G_x^2 + G_y^2} \quad (2.2)$$

$$\nabla f \approx |G_x| + |G_y| \quad (2.3)$$

อย่างไรก็ตาม เนื่องจากขนาดของ Gradient (∇f) ที่คำนวณได้อาจมีค่ามากกว่า 255 ดังนั้นเราจำเป็นต้องทำการสเกลลิง (Scaling) เพื่อให้ขนาดของ Gradient อยู่ในช่วง 0-255 จากนั้นจึงเลือกค่าเทรซโฮลเพื่อแสดงขอบภาพเป็นสีขาวและดำหรือเรียกว่าภาพสองระดับ (Binary Images)

การหาขอบภาพดังที่ได้อธิบายมาแล้วข้างต้น เมื่อนำมาใช้หาขอบของตัวอักษร จะช่วยลดข้อมูลในการประมวลผลแต่ยังคงไว้ซึ่งลักษณะเด่น (Features) ของตัวอักษรแต่ละตัวไว้

2.4 การหาแกนกลางของภาพ (Skeleton)

การหาแกนกลางของภาพเป็นการหาแกนกลางของวัตถุหรือการกำจัดพิกเซลที่เส้นขอบให้เหลือเพียง 1 พิกเซล การหาแกนกลางของภาพบางครั้งเรียกว่า “การทำให้บาง (Thinning)” วิธีที่การหาแกนกลางนั้นประกอบด้วย 2 ขั้นตอน ดังที่เขียนไว้ข้างล่าง ซึ่งในแต่ละขั้นตอนจะทำการลบพิกเซลแต่ละพิกเซลในตัวอักษรออกทีละจุด โดยใช้หน้าต่างขนาด 3×3 ดังแสดงในรูปที่ 2.9 วางลงบนมุมซ้ายบนของภาพโดยหน้าต่างจะเลื่อนจากซ้ายไปขวาและจากบนลงล่าง เพื่อตรวจหาพิกเซลที่มีค่าเป็น “1” เมื่อตรวจพบแล้วจึงเริ่มขั้นตอนที่ 1 และ 2

อย่างไรก็ตาม ก่อนการดำเนินการตามขั้นตอนที่ 1 และ 2 ภาพที่ต้องการหาต้องเป็นภาพสองระดับเท่านั้น คือ มีค่าเป็น “0” และ “1” โดยค่าที่เป็น “1” หมายถึงวัตถุ และค่าที่เป็น “0” หมายถึง ฉากหลังของภาพ

p9	p2	p3
p8	p1	p4
p7	p6	p5

รูปที่ 2.9 หน้าต่างขนาด 3×3 เมื่อ p1 คือ ขอบของวัตถุ

ขั้นตอนที่ 1 เป็นการลบจุดภาพบริเวณขอบวัตถุทางด้านขวามือและด้านล่าง โดยวางหน้าต่าง 3×3 ทาบไปบนจุดภาพที่มีค่าเป็น “1” และกำหนดให้จุดดังกล่าวเป็น p1 จากนั้นตรวจสอบเงื่อนไข ต่อไปนี้

1) หาผลรวมของจุดภาพมีค่าเป็น “1” ที่อยู่ล้อมรอบจุด p1 จะต้องมีย่านตั้งแต่ 2-6 จุดภาพ

2) นับจำนวนครั้งที่มีการเปลี่ยนแปลงจาก “0” ไปเป็น “1” รอบจุด p1 ในลักษณะตามเข็มนาฬิกาซึ่งจะต้องมีจำนวนครั้งของการเปลี่ยนแปลงเกิดขึ้น 1 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การ AND กันทางโลจิกของจุด p2, p4, p6 แล้วได้ผลลัพธ์เท่ากับ “0”

4) การ AND กันทางโลจิกของจุด p4, p6, p8 แล้วได้ผลลัพธ์เท่ากับ “0”

ขั้นตอนที่ 2 เป็นการลบจุดภาพบริเวณขอบของวัตถุด้านซ้ายมือและด้านบนของขอบภาพ โดยวิธีการเหมือนขั้นตอนที่ 1 เพียงแต่เปลี่ยนเงื่อนไข ดังนี้

3) การ AND กันทางโลจิกของจุด p2, p4, p8 แล้วได้ผลลัพธ์เท่ากับ “0”

4) การ AND กันทางโลจิกของจุด p2, p6, p8 แล้วได้ผลลัพธ์เท่ากับ “0”

การหาแกนกลางของภาพเป็นการลบจุดภาพออกไปโดยจะวางหน้าต่างขนาด 3x3 ทาบไปบนจุดภาพที่มีค่าเป็น “1” และกำหนดให้เป็น p1 จากนั้นตรวจสอบว่าเป็นไปตามเงื่อนไขในขั้นตอนที่ 1 หรือไม่ ถ้า p1 มีเงื่อนไขตรงกับเงื่อนไข 4 ข้อข้างต้นพิกัดของจุด p1 จะถูกจัดเก็บไว้ จากนั้นหน้าต่าง 3x3 จะเลื่อนไปทางขวามือ 1 พิกเซล และดำเนินการเหมือนกับที่กล่าวไปแล้วข้างต้นจนหมดทั้งภาพ เมื่อดำเนินการครบทุกจุดในภาพแล้ว พิกัดที่ถูกจัดเก็บไว้ทั้งหมดจะถูกเปลี่ยนจาก “1” เป็น “0” หรือเป็นการลบจุดภาพนั้น โดยการลบพิกเซลนั้นจะลบจากซ้ายไปขวาและจากบนลงล่าง จากนั้นเมื่อเสร็จในขั้นตอนที่ 1 แล้ว จึงพิจารณาตรวจสอบเงื่อนไขในขั้นตอนที่ 2 ต่อ เมื่อตรงตามเงื่อนไขก็จะดำเนินการเหมือนกับขั้นตอนที่ 1 วนอย่างนี้ไปเรื่อยๆจนเหลือแต่พิกเซลสุดท้ายที่เป็นแกนกลางของภาพ แสดงดังรูปที่ 2.10



รูปที่ 2.10 ภาพแสดงการลบจุดภาพตามเงื่อนไขทั้ง 2 ขั้นตอน และแสดงภาพผลลัพธ์ของแกนกลาง

(ก) รูปภาพตัวอย่างก่อนการดำเนินการตามเงื่อนไขทั้ง 2

(ข) เงื่อนไขตามขั้นตอนที่ 1 ลบจุดภาพทางด้านขวามือและด้านล่าง

(ค) เงื่อนไขตามขั้นตอนที่ 2 ลบจุดภาพทางด้านซ้ายมือและด้านบน

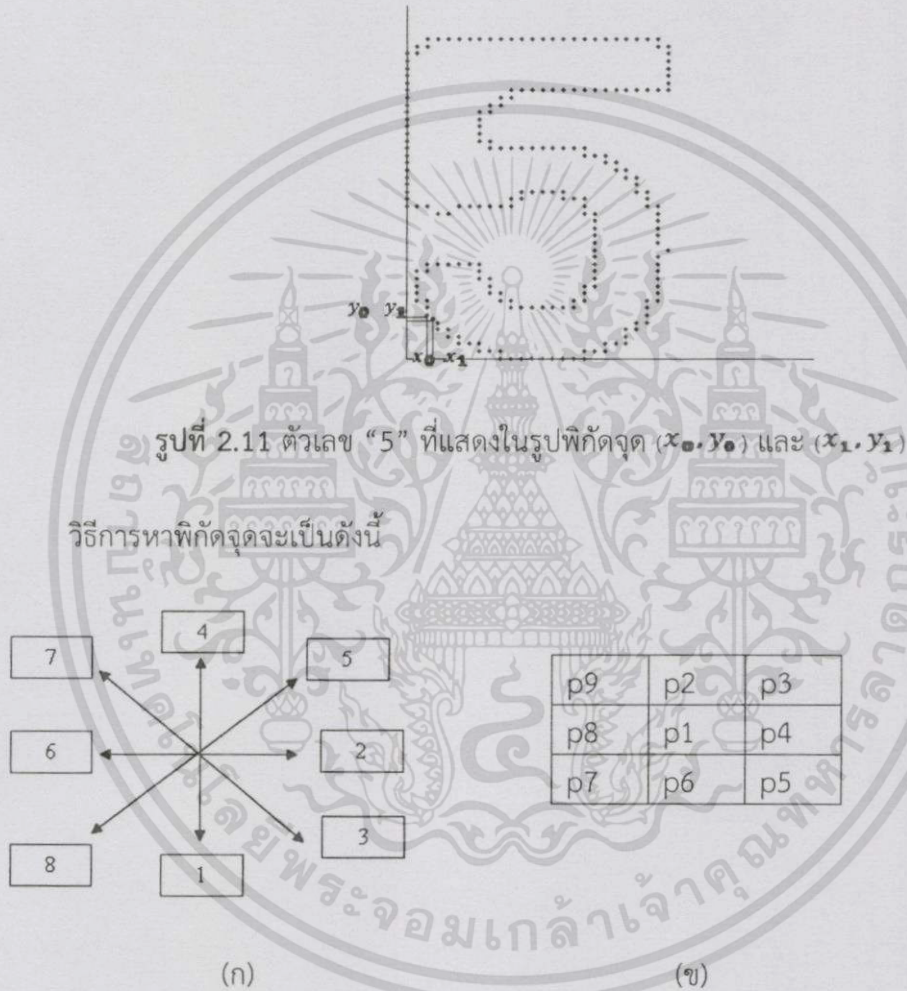
(ง) ผลลัพธ์ที่ได้จากการลบจุดภาพออกไปจนเหลือแต่แกนกลางของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การหาตำแหน่งพิกัด (Coordinate Detection)

การหาตำแหน่งพิกัดเป็นการค้นหาและเก็บพิกัดของจุดภาพที่อยู่บนขอบของตัวอักษรในแนวแกน x และ y แล้วจัดเก็บในรูปแบบ x_i และ y_i เมื่อ $i = 0, 1, 2, \dots, N-1$

รูปที่ 2.11 เป็นการแสดงพิกัดจุดของเส้นขอบของตัวเลข "5" ที่มีจำนวน N จุด ใน XY -plane โดยกำหนดให้จุดเริ่มต้นอยู่ที่จุด (x_0, y_0) พิกัดของเส้นขอบของตัวอักษรที่จุดถัดไป คือ $(x_1, y_1), (x_2, y_2), \dots, (x_{N-1}, y_{N-1})$



รูปที่ 2.12 (ก) ทิศทาง 8 ทิศทางที่ตามเข็มนาฬิกา

(ข) หน้าตาขนาด 3×3 เมื่อ p_1 คือขอบของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการหาพิกัดจุดของเส้นขอบของตัวเลข “5” ที่แสดงดังรูปที่ 2.11 ชั้นแรกโปรแกรมจะทำการค้นหาขอบภาพที่มีค่าเป็น “1” โดยโปรแกรมจะเคลื่อนที่จากซ้ายไปขวาและจากบนลงล่าง ต่อมาเมื่อพบขอบภาพที่มีค่าเท่ากับ “1” โปรแกรมจะทำการวางหน้าต่างขนาด 3x3 ดังรูป 2.12 (ข) ลงที่จุดนั้นโดย p_1 คือขอบของวัตถุ จากนั้นทำการเก็บพิกัดในแนวแกน x และ y ต่อมาโปรแกรมจะพิจารณาจุดข้างเคียงที่อยู่ล้อมรอบ จุด p_1 ในลักษณะ 8 ทิศทางตามเข็มนาฬิกาดังรูปที่ 2.12 (ก) ถ้าตรวจพบจุดภาพถัดไปมีค่าเป็น “1” จะทำการย้ายจุด p_1 ไปที่จุดต่อไปและทำการเก็บพิกัด การดำเนินการจะกระทำไปเรื่อยๆ จนกระทั่งสามารถเก็บพิกัดของจุดภาพที่อยู่บนขอบของตัวเลข ได้ทั้งหมด สุดท้ายนำพิกัดในแนวแกน x และ y มาจัดเก็บในรูปของ x_i และ y_i เมื่อ $i = 0, 1, 2, \dots, N-1$

2.6 การหาค่า Eigen vector และ Eigen value

Eigen vector และ Eigen value เข้ามามีส่วนช่วยในการ Normalize ค่าตำแหน่งพิกัดจุดของข้อมูล เพื่อหมุนข้อมูลให้อยู่ในแนวแกนเดียวกัน ดังนั้นไม่ว่าภาพที่ถูกถ่ายจะเอียงไปหรือไม่ก็ตาม ข้อมูลจะถูกหมุนให้อยู่ในแนวแกนเดียวกัน โดยค่า Eigen vector และ Eigen value นั้นสามารถหาได้จากการนำพิกัดจุดของตัวอักษรหรือตัวเลขมาหาค่าเฉลี่ยดังสมการที่ (2.4)

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (2.4)$$

โดยที่ x_i แทนค่าพิกัดในแนวแกน x และ y_i แทนค่าพิกัดในแนวแกน y ต่อมาลบค่าพิกัด x_i และ y_i แต่ละจุดออกด้วยค่าเฉลี่ย \bar{x} และ \bar{y} จะได้ \bar{X}_i และ \bar{Y}_i ดังสมการที่ (2.5)

$$\bar{X}_i = x_i - \bar{x}, \quad \bar{Y}_i = y_i - \bar{y} \quad (2.5)$$

จากนั้นทำการหาค่าของ Covariant Matrix (C) ด้วยสมการที่ (2.6)

$$C = \frac{1}{N} \sum_{i=1}^N \begin{pmatrix} (x_i - \bar{x})(x_i - \bar{x}) & (x_i - \bar{x})(y_i - \bar{y}) \\ (x_i - \bar{x})(y_i - \bar{y}) & (y_i - \bar{y})(y_i - \bar{y}) \end{pmatrix} \quad (2.6)$$

เมื่อ

$$Cov = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะได้ค่าของ Covariant Matrix (C) ดังสมการที่ (2.7)

$$C = \begin{bmatrix} \text{Cov}(X_i \cdot X_i) & \text{Cov}(X_i \cdot Y_i) \\ \text{Cov}(X_i \cdot X_i) & \text{Cov}(X_i \cdot Y_i) \end{bmatrix} \quad (2.7)$$

จากนั้นทำการหาค่าของ Eigenvector และ Eigenvalue ด้วยคำสั่ง eig ซึ่งเป็นคำสั่งที่อยู่ใน tool ของ MATLAB

$$[X \lambda] = \text{eig}(C)$$

เมื่อ λ คือ Eigen value

X คือ Eigen vector

นำค่า X_i กับ Y_i มาวางในรูป matrix

$$A = \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \quad (2.8)$$

หลังจากนั้นนำเวกเตอร์ A มาคูณกับ Eigen vector เราจะได้พิกัดจุดภาพที่ Normalized แล้ว นั่นคือ

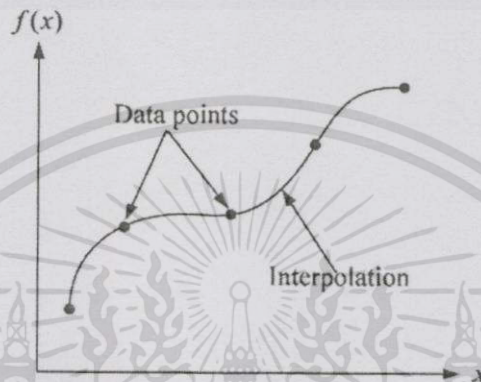
$$\text{Normalized} = A \times X \quad (2.9)$$

จากนั้นนำชุดข้อมูลพิกัดจุดที่ผ่านการ Normalized แล้วไปประมวลผลในขั้นตอนต่อไป

2.7 การอินเตอร์โพลเซชัน (Interpolation)

เมื่อมีการถ่ายภาพจากกล้องดิจิตอลหรือกล้องโทรศัพท์มือถือ อาจทำให้ภาพบิดเบือนหรือรายละเอียดขาดหายไป ซึ่งสิ่งเหล่านี้จะทำให้เกิดการสูญเสียหรือขาดหายของข้อมูลในระดับพิกเซลอยู่ตลอดเวลา จึงจะต้องมีกระบวนการเพื่อเติมเต็มพิกเซลที่หายไปหรือปรับแก้พิกเซลที่สร้างขึ้นใหม่ อาจเป็นการแทรกตำแหน่งพิกัดจุดเข้าไประหว่างตำแหน่งพิกัดจุดข้อมูลสองจุด ดังรูปที่ 2.13 เพื่อให้ภาพที่ดูถูกแก้ไขมีความสวยงามมากที่สุด เราเรียกวิธีการนี้ว่า “การอินเตอร์โพลเซชัน”

ถึงแม้จะเป็นข้อมูลชุดเดียวกัน เนื่องจากภาพถ่ายแต่ละภาพถูกถ่ายใกล้ไกลไม่เท่ากันทำให้ข้อมูลที่ได้เล็กใหญ่ไม่เท่ากัน สภาพแสงในตอนถ่ายไม่เหมือนกัน อีกทั้งความสมบูรณ์ของข้อมูลก็ไม่เท่ากัน ทำให้ตำแหน่งพิกัดจุดที่ได้นั้นไม่เท่ากันและหากนำข้อมูลที่มีตำแหน่งพิกัดจุดไม่เท่ากันมาประมวลผลก็จะทำให้ได้ผลลัพธ์ที่ผิดเพี้ยนไป ด้วยเหตุนี้เองจึงต้องนำข้อมูลมาทำอินเตอร์โพลชันเพื่อปรับจำนวนพิกัดจุดของข้อมูลให้เท่ากันก่อนนำไปประมวลผล



รูปที่ 2.13 การอินเตอร์โพลชันของข้อมูล

โดยการลดหรือเพิ่มจำนวนพิกัดจุดจะใช้คำสั่ง “interp1” ซึ่งเป็นคำสั่งที่อยู่ใน tool ของ MATLAB ซึ่งหมายถึง 1-D Interpolation คือ จะเป็นการประมาณค่าในช่วงของตำแหน่งพิกัดจุดโดยรับอินพุตเพียงตัวแปรเดียว ซึ่งในการรู้จำตัวอักษรบนแผ่นป้ายทะเบียนนี้จะใช้การอินเตอร์โพลชันในการลดตำแหน่งพิกัดจุดที่ได้จากชุดข้อมูลผ่านการ Normalized แล้วให้มีจำนวนพิกัดจุดของตัวอักษรแต่ละตัวให้เท่ากันก่อนนำไปประมวลผลต่อไป โดยการใช้งาน “interp1” มีรูปแบบคำสั่ง ดังนี้

$$UVIx = \text{interp1}(t, Xd, xi, 'OPTION');$$

โดย t แทนช่วงของจำนวนตัวแปร Xd เช่น 1:3:300 ซึ่งจะมีจำนวนตำแหน่งพิกัดจุดของตัวแปร Xd อยู่ในช่วงเท่ากับ $300/3 = 100$ จุด

Xd แทนด้วยตัวแปรที่ต้องการแทรกตำแหน่งพิกัดจุดลงไป เช่น ตำแหน่งพิกัดจุดในแนวแกนตั้ง y หรือตำแหน่งพิกัดจุดในแนวแกนนอน x

xi แทนจำนวนจุดที่ต้องการเพิ่มเข้าไปในช่วง โดยจะเป็นข้อมูลตัวเลขที่อยู่ในช่วงของ t เช่น

1:2:300 จะได้ตำแหน่งพิกัดจุดที่เพิ่มขึ้นจากเดิม 100 จุด เป็น $300/2 = 150$ จุด หรือในทางกลับกันถ้ากำหนดให้เป็น 1:5:300 จำนวนจุดที่ได้ก็จะลดลงจากเดิม 100 จุดเหลือเพียง $300/5 = 60$ จุด เป็นต้น

UVIx แทนด้วยตัวแปรที่ใช้เก็บค่าตำแหน่งพิกัดจุดที่ได้ทำการลดหรือเพิ่มตำแหน่งพิกัดจุด 'OPTION' เป็นตัวเลือกของวิธีการอินเตอร์โพลชัน ซึ่งใน MATLAB เวอร์ชัน R2011a จะมีให้เลือก 4 วิธี ซึ่งประกอบไปด้วย

1. วิธี 'nearest' คือวิธีการแทรกแบบแทรกจุดข้างเคียง
2. วิธี 'linear' คือวิธีการแทรกแบบเส้นตรง
3. วิธี 'spline' คือวิธีการแทรกแบบลูกบาศก์อิสระ
4. วิธี 'cubic' คือวิธีการแทรกแบบต่อเนื่องเป็นช่วง

โดยวิธีที่นำมาใช้ในโครงงานนี้คือ "nearest" เนื่องจากวิธี nearest ให้ค่าที่มีความราบเรียบที่สุด เกิดค่า error น้อยที่สุด จึงเหมาะแก่การอินเตอร์โพลชัน

2.8 ทฤษฎีการหาค่าสัมประสิทธิ์ในเทอมของโคไซน์ Discrete Cosine Transform

Discrete Cosine Transform (DCT) เป็นสมการทางคณิตศาสตร์ ถูกนำไปใช้ในการลดข้อมูลภาพ และจะถูกใช้กับข้อมูลภาพที่เป็นจำนวนจริง ค่าสัมประสิทธิ์ของตำแหน่งที่ได้ของตัวอักษรและตัวเลขแต่ละตัวนั้นจะไม่เหมือนกันจึงเป็นการแสดงถึงความเป็นเอกลักษณ์ของตัวอักษรแต่ละตัว โดยการหาค่าสัมประสิทธิ์ของตำแหน่งนั้นจะใช้สมการของ Discrete Cosine Transform ดังสมการที่ (2.10)

$$f(k) = w(k) \sum_{n=1}^N x(n) \cos \frac{\pi(2n-1)(k-1)}{2N}$$

$$g(k) = w(k) \sum_{n=1}^N y(n) \cos \frac{\pi(2n-1)(k-1)}{2N} \quad (2.10)$$

$$\text{เมื่อ } w(k) = \begin{cases} \frac{1}{\sqrt{N}} & \text{โดยที่ } k = 1 \\ \sqrt{\frac{2}{N}} & \text{โดยที่ } 2 \leq k \leq N \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งตำแหน่งพิกัดจุดที่ผ่านการอินเตอร์โพลชันแล้วจะนำมาทำการหาค่าสัมประสิทธิ์ DCT ซึ่งค่าที่ได้ก็คือค่าตำแหน่งพิกัดจุดที่อยู่ในแนวแกนอน $f(k)$ และค่าตำแหน่งพิกัดจุดที่อยู่ในแนวแกนตั้ง $g(k)$ หลังจากนั้นจะนำค่าสัมประสิทธิ์ของตำแหน่งที่ได้นี้มาเข้ากระบวนการ FIR System เพื่อหาค่าความเป็นเอกลักษณ์ของตัวอักษรแต่ละตัว

2.9 ระบบผลตอบสนองอิมพัลส์ที่จำกัดจำนวน

(Finite Impulse Response: FIR System)

ระบบผลตอบสนองอิมพัลส์จำกัดจำนวน (FIR System) มีลักษณะเด่น คือ มีผลตอบสนองทางเฟสแบบเชิงเส้น (linear phase) และมีความสัมพันธ์ระหว่างค่าอินพุตกับเอาต์พุตของข้อมูลในแง่ของขนาดและตำแหน่ง โดยมีค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์ $h(m)$ เป็นตัวบอกคุณลักษณะหรือเอกลักษณ์ของข้อมูล โดยจะแสดงไว้ในสมการที่ (2.11) ซึ่งระบบผลตอบสนองอิมพัลส์จำกัดจำนวนนี้เหมาะสำหรับการพัฒนาเครื่องมือวัดทางด้านกรแพทย์และในงานที่เกี่ยวข้องกับการประมวลผลภาพ

$$g_n = \sum_{m=0}^L h_m * f_{n-m} \quad (2.11)$$

เมื่อ f_{n-m} คือ เวกเตอร์ค่าอินพุตของระบบ FIR $f(k)$
 h_m คือ เวกเตอร์ค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์จำกัดจำนวน ซึ่งเป็นตัวบอกคุณลักษณะและเอกลักษณ์ของข้อมูล
 g_n คือ เวกเตอร์ค่าเอาต์พุตของระบบ FIR $g(k)$
 L คือ จำนวนลำดับของระบบ FIR

เนื่องจากอินพุต (f_{n-m}) ของระบบ FIR เป็นเวกเตอร์จึงไม่สามารถคูณกันได้ด้วยเวกเตอร์เอกลักษณ์ของข้อมูล (h_m) ดังนั้นเพื่อให้มีการคูณกันได้ของระบบ FIR เราจึงจัดเรียงค่าอินพุต (f_{n-m}) ใหม่ให้อยู่ในรูปเมทริกซ์สามเหลี่ยมล่าง (Lower Triangular Matrix, Q) ดังสมการด้านล่าง

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_n \end{bmatrix} = \begin{bmatrix} f_0 & 0 & \dots & 0 \\ f_1 & f_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_m & f_{m-1} & \dots & f_{m-L} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{bmatrix} \quad (2.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โดยที่ f คือ ค่าสัมประสิทธิ์ DCT ของพิกัดจุดในแนวแกน X
 g คือ ค่าสัมประสิทธิ์ DCT ของพิกัดจุดในแนวแกน Y
 h คือ เวกเตอร์ค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์จำกัดจำนวน

เนื่องจากเวกเตอร์ค่าอินพุต (f_{n-m}) และเวกเตอร์ค่าเอาต์พุต (g_n) เป็นสิ่งที่รู้ค่า เราต้องการหาเวกเตอร์ค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์จำกัดจำนวน (h_m) ของระบบ FIR ซึ่งเป็นตัวบอกคุณลักษณะและเอกลักษณ์ของข้อมูล เราสามารถหาค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์ (h_m) จากสมการที่ (2.13)

$$h_m = [(Q)^T * Q]^{-1} * Q^T * g_n \quad (2.13)$$



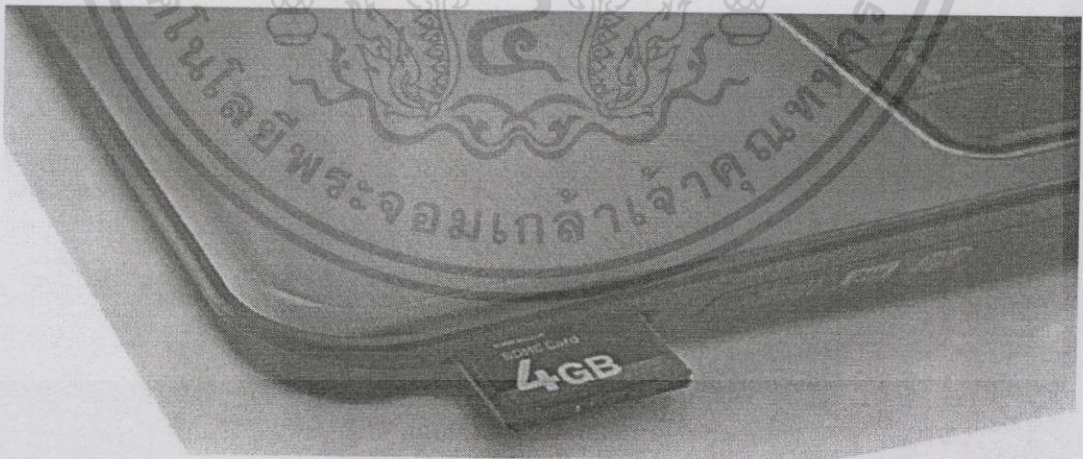
บทที่ 3

การแยกส่วนแผ่นป้ายทะเบียนออกจากภาพ

ในบทนี้จะกล่าวถึงการแยกส่วนแผ่นป้ายทะเบียนออกจากภาพ ซึ่งจะอธิบายถึงวิธีการและขั้นตอนในการค้นหาส่วนที่เป็นแผ่นป้ายทะเบียนและแยกส่วนที่เป็นแผ่นป้ายทะเบียนรถยนต์ออกจากภาพ โดยใช้โปรแกรม MATLAB ในการสร้างคำสั่งหรือสร้างขั้นตอนในการประมวลผลรูปภาพ โดยจะกล่าวถึง การนำภาพถ่ายที่ได้มาวิเคราะห์หาส่วนที่เป็นแผ่นป้ายทะเบียนและแยกส่วนที่เป็นแผ่นป้ายทะเบียนออกจากรูปภาพ รวมไปถึงวิธีการถ่ายภาพ

ข้อกำหนดในการถ่ายแผ่นป้ายทะเบียนและขั้นตอนในการจัดเก็บภาพถ่าย มีดังนี้

1. ควรใช้กล้องดิจิตอลหรือกล้องมือถือ ความละเอียดไม่ต่ำกว่า 1 ล้านพิกเซล
2. สามารถเลือกถ่ายภาพแผ่นป้ายทะเบียนได้ทั้งด้านหน้าและด้านหลังของตัวรถยนต์ เนื่องจากผลลัพธ์ที่ได้นั้นมีความใกล้เคียงกัน และควรเว้นระยะห่างในการถ่ายภาพแผ่นป้ายทะเบียนประมาณ 1-2 เมตร โดยถ่ายภาพในลักษณะเป็นภาพหน้าตรง
3. จากนั้นทำการถ่ายโอนไฟล์ข้อมูลภาพทั้งหมดไว้ในคอมพิวเตอร์ เพื่อการเรียกใช้ในขั้นตอนต่อไป โดยควรปรับขนาดของภาพที่ถ่ายมาให้มีขนาด 640x420 พิกเซล เพื่อลดขนาดของข้อมูลและง่ายต่อการประมวลผลในโปรแกรม MATLAB



รูปที่ 3.1 ภาพการจัดเก็บป้ายทะเบียนโดยใช้ Memory card เชื่อมต่อกับ Notebook

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การค้นหาส่วนที่เป็นแผ่นป้ายทะเบียนในรูปภาพ

การวิเคราะห์หาส่วนที่เป็นแผ่นป้ายทะเบียนในรูปภาพคือ การนำภาพที่มีส่วนที่เป็นแผ่นป้ายทะเบียนรถยนต์มาทำการวิเคราะห์เพื่อหาส่วนที่เป็นแผ่นป้ายทะเบียนรถยนต์ ซึ่งจะมีขั้นตอนดังต่อไปนี้

1. โหลดข้อมูลภาพถ่ายด้านหน้ารถยนต์ที่เก็บไว้ในฐานข้อมูลของคอมพิวเตอร์เข้ามา โดยใช้คำสั่ง 'imread' ซึ่งเป็น Tool ใน MATLAB ในการโหลดไฟล์ภาพขึ้นมา ภาพที่ได้จะเป็นภาพสี RGB โดยตัวโปรแกรมจะกำหนดให้นามสกุลของไฟล์ภาพเป็น JPEG หรือ .jpg แสดงดังรูปที่ 3.2



รูปที่ 3.2 ภาพสี RGB ด้านหน้าของรถยนต์ที่มีส่วนแผ่นป้ายทะเบียนอยู่ด้วย

2. แปลงภาพสี RGB จากขั้นตอนที่ 1 ให้เป็นภาพระดับสีเทา (Gray Scale Image) โดยใช้คำสั่ง 'rgb2gray' ซึ่งเป็น Tool คำสั่งใน MATLAB ซึ่งเป็นการเฉลี่ยกันของแม่สีทั้ง 3 ใน 1 พิกเซล ได้แก่ สีแดง สีเขียว สีน้ำเงิน ให้กลายเป็นพิกเซลสีเทาก่อน ดังสมการที่ (3.1) เพื่อทำให้ง่ายต่อการนำภาพไปประมวลผลในขั้นตอนต่อไป โดยรูปที่ 3.3 จะแสดงภาพถ่ายด้านหน้ารถยนต์ที่ถูกแปลงจากภาพสี RGB ให้เป็นภาพระดับสีเทา

$$\text{ค่าระดับสีเทา} = ((0.299 \times R) + (0.587 \times G) + (0.114 \times B)) \quad (3.1)$$

โดยที่ R คือ ค่าอินพุตพิกเซลสีแดง

G คือ ค่าอินพุตพิกเซลสีเขียว

B คือ ค่าอินพุตพิกเซลสีน้ำเงิน



รูปที่ 3.3 ภาพด้านหน้ารถยนต์ที่ถูกแปลงจากภาพสี RGB เป็นภาพระดับสีเทา

3. ทำการตรวจเช็คค่าเทรชโฮล ของภาพระดับสีเทาที่ได้โดยใช้คำสั่ง 'graythresh' ซึ่งเป็น Tool ใน MATLAB โดยคำสั่งนี้จะระบุค่าเทรชโฮล ของภาพระดับสีเทามาให้ ซึ่งในบางครั้งหากนำภาพระดับสีเทาที่ได้นี้มาทำเป็นภาพสองระดับเลย จะทำให้ภาพนั้นเห็นส่วนที่เป็นแผ่นป้ายทะเบียนไม่ชัดเจน เนื่องจากค่าเทรชโฮลมีค่าไม่เหมาะสม จึงต้องทำการปรับค่าเทรชโฮลแบบ Manual ช่วยในการปรับค่าเทรชโฮลจะทำให้ภาพที่ได้มีพิกเซลสีขาวเพิ่มมากขึ้นหรือน้อยลงได้ตามต้องการ หลังจากทำการปรับค่าเทรชโฮลแล้วจะได้ดังรูปที่ 3.4



รูปที่ 3.4 ภาพที่ได้หลังจากทำการปรับค่าเทรชโฮลที่เหมาะสมแล้ว

5. แปลงภาพระดับสีเทาจากขั้นตอนที่ 3 ให้เป็นภาพสองระดับ (Binary Image) โดยใช้คำสั่ง 'im2bw' ซึ่งเป็น Tool คำสั่งใน MATLAB โดยใช้ค่าเทรชโฮลที่เหมาะสมระบุลงไปคำสั่ง ซึ่งจะได้ภาพที่มีความเข้มของสีในภาพเพียง 2 ระดับ คือ พิกเซลสีขาวจะแทนด้วย "1" และพิกเซลสีดำจะแทนด้วย "0" ซึ่งการแปลงภาพจากภาพระดับสีเทาเป็นภาพสองระดับจะช่วยให้การประมวลผลของโปรแกรม MATLAB เร็วขึ้น โดยลดเนื้อที่ในการเก็บข้อมูลภาพให้มีขนาดเล็กลงเหลือเพียง 1 บิต ดังรูปที่ 3.5



รูปที่ 3.5 ภาพที่ผ่านการแปลงจากภาพระดับสีเทาเป็นภาพสองระดับ

6. นำภาพที่ได้จากขั้นตอนที่ 5 มาทำการลบพื้นที่หรือกลุ่มพิกเซลสีขาวที่ติดอยู่กับขอบภาพออกไป โดยใช้คำสั่ง 'imclearborder' ซึ่งเป็น Tool คำสั่งใน MATLAB โดยคำสั่งนี้จะช่วยในการตัดพิกเซลสีขาวที่ติดกับขอบภาพซึ่งไม่ใช่ส่วนที่เป็นแผ่นป้ายทะเบียนรถยนต์ออกไป ดังรูปที่ 3.6



รูปที่ 3.6 ภาพสองระดับที่ผ่านการลบกลุ่มพิกเซลสีขาวที่ติดกับขอบภาพออกไป

7. นำภาพที่ได้จากขั้นตอนที่ 6 มาทำการกำหนดหมายเลขให้กับกลุ่มพิกเซลสีขาวแต่ละกลุ่ม หรือที่เรียกว่าการทำ Label โดยใช้คำสั่ง 'bwlabel' ซึ่งเป็น Tool คำสั่งใน MATLAB โดยกำหนดหมายเลขให้กับกลุ่มพิกเซลสีขาวแต่ละกลุ่ม ซึ่งถ้าหากพิกเซลสีขาวที่อยู่ติดกันเป็นกลุ่มในบริเวณเดียวกันโดยไม่ถูกแยกออกด้วยพิกเซลสีดำ กลุ่มพิกเซลสีขาวเหล่านั้นก็จะถูกทำการกำหนดให้มีหมายเลข (Label) เดียวกัน ดังรูปที่ 3.7 เป็นการยกตัวอย่างการ Label กลุ่มพิกเซลสีขาวแต่ละกลุ่มอย่างคร่าวๆ และจะพบว่ากลุ่มพิกเซลสีขาวที่มีขนาดใหญ่ที่สุด คือ กลุ่มพิกเซลสีขาวหมายเลข 1 ซึ่งเป็นส่วนของแผ่นป้ายทะเบียนรถยนต์



รูปที่ 3.7 ภาพแสดงการ Label กลุ่มพิกเซลสีขาว

3.2 การแยกส่วนแผ่นป้ายทะเบียนออกจากรูปภาพ

1. หลังจากทำการ Label กลุ่มพิกเซลสีขาวแล้ว ต่อมาทำการหากลุ่มพิกเซลสีขาวที่มีขนาดใหญ่ที่สุดโดยใช้คำสั่ง 'find' ซึ่งเป็น Tool คำสั่งใน MATLAB โดยจะทำการค้นหากลุ่มพิกเซลที่มีจำนวนพิกเซลสีขาวมากที่สุด ซึ่งในรูปที่ 3.7 จะพบว่าเป็นกลุ่มพิกเซลสีขาวหมายเลข 1 จากนั้น เราจะได้กลุ่มพิกเซลสีขาวที่มีอัตราส่วนความกว้างและความยาวตามที่กำหนด ซึ่งในภาพจะเป็นกลุ่มพิกเซลสีขาวที่เป็นส่วนของแผ่นป้ายทะเบียน จากนั้นจึงทำการเก็บขอบเขตที่จะใช้ในการตัดส่วนที่เป็นแผ่นป้ายทะเบียนออกจากภาพ โดยแสดงขอบเขตในการตัดได้ดังรูปที่ 3.8 ซึ่ง $\min x1$ จะแทนตำแหน่งพิกเซลสีขาวที่อยู่ซ้ายสุด $\max x1$ แทนตำแหน่งพิกเซลสีขาวที่อยู่ขวาสุด $\min y1$ แทนพิกเซลสีขาวที่อยู่บนสุด และ $\max y1$ แทนพิกเซลสีขาวที่อยู่ล่างสุด



รูปที่ 3.8 ภาพแสดงขอบเขตในการตัดส่วนที่เป็นแผ่นป้ายทะเบียนออกจากภาพ

2. ตัดส่วนที่เป็นแผ่นป้ายทะเบียนออกจากภาพสี RGB ในช่วงของ $\min x1$ จนถึง $\max x1$ และ $\min y1$ จนถึง $\max y1$ ที่ได้จากการเก็บขอบเขตในขั้นตอนที่ 1 ก็จะได้ภาพส่วนที่เป็นแผ่นป้ายทะเบียน ดังรูปที่ 3.9

2กข 525

กรุงเทพมหานคร

รูปที่ 3.9 ภาพแสดงส่วนที่เป็นแผ่นป้ายทะเบียนที่ได้จากภาพสี RGB

3. นำภาพแผ่นป้ายทะเบียนที่เป็นภาพสี RGB มาทำการแปลงให้เป็นภาพระดับสีเทาโดยใช้คำสั่ง 'rgb2gray' ซึ่งจะสามารถระบุค่าเทรชโวลของภาพระดับสีเทาได้โดยใช้คำสั่ง 'graythresh' ซึ่งจะนำค่าเทรชโวลที่ได้นี้มาใช้ในการแปลงภาพระดับสีเทาให้เป็นภาพสองระดับด้วยคำสั่ง 'im2bw' ก็จะได้ภาพแผ่นป้ายทะเบียนที่เป็นภาพสองระดับ ดังรูปที่ 3.10

2กข 525

กรุงเทพมหานคร

รูปที่ 3.10 ภาพสองระดับของส่วนที่เป็นแผ่นป้ายทะเบียน

4. นำภาพแผ่นป้ายทะเบียนที่ผ่านการแปลงให้เป็นภาพสองระดับมาทำการสลับสีของพิกเซล (Inverse Binary) โดยใช้คำสั่ง 'imcomplement' ที่เป็น Tool ในโปรแกรม MATLAB ซึ่งจะทำให้พิกเซลสีขาวจะถูกเปลี่ยนให้เป็นพิกเซลสีดำ ในทำนองเดียวกันพิกเซลที่เป็นสีดำ จะถูกเปลี่ยนให้เป็นพิกเซลสีขาว ดังรูปที่ 3.11 เพื่อทำให้ตัวอักษรที่อยู่บนแผ่นป้ายทะเบียนมาทำการประมวลผลต่อไปได้

2กข 525

กรุงเทพมหานคร

รูปที่ 3.11 ภาพที่ได้หลังจากทำการสลับสีของพิกเซล (Inverse Binary)

5. ทำการเคลียร์กลุ่มพิกเซลสีขาวที่ติดอยู่กับขอบของแผ่นป้ายทะเบียนด้วยคำสั่ง 'imclearborder' ซึ่งจะช่วยกำจัดกลุ่มของพิกเซลสีขาวที่ติดกับขอบของแผ่นป้ายทะเบียนดังรูปที่ 3.12 (ก) ออกไป และทำการเคลียร์พิกเซลที่ไม่ใช่ตัวอักษรออกจะได้ดังรูปที่ 3.12 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

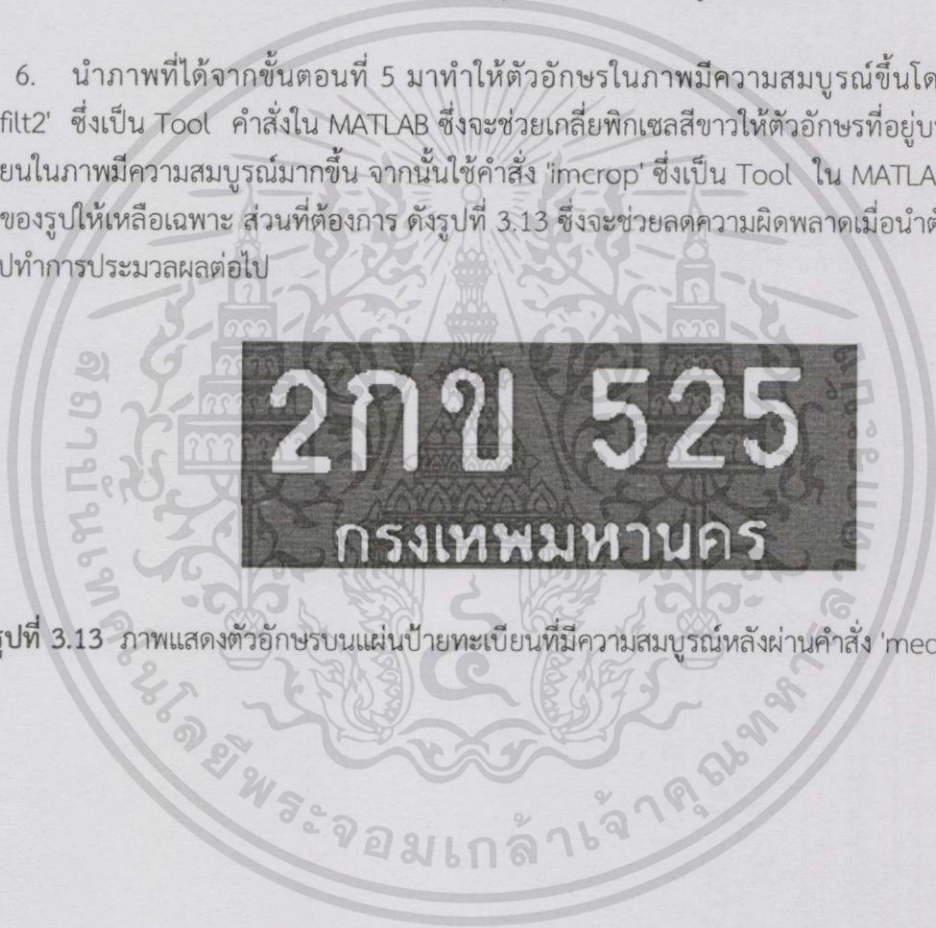


รูปที่ 3.12 ภาพการกำจัดกลุ่มพิกเซลสีขาวที่ติดกับขอบภาพ

(ก) ภาพก่อนทำการลบกลุ่มพิกเซลสีขาวที่อยู่ติดกับขอบภาพ

(ข) ภาพหลังทำการลบกลุ่มพิกเซลสีขาวที่อยู่ติดกับขอบภาพ

6. นำภาพที่ได้จากขั้นตอนที่ 5 มาทำให้ตัวอักษรในภาพมีความสมบูรณ์ขึ้นโดยใช้คำสั่ง 'medfilt2' ซึ่งเป็น Tool คำสั่งใน MATLAB ซึ่งจะช่วยให้พิกเซลสีขาวในตัวอักษรที่อยู่บนแผ่นป้ายทะเบียนในภาพมีความสมบูรณ์มากขึ้น จากนั้นใช้คำสั่ง 'imcrop' ซึ่งเป็น Tool ใน MATLAB เพื่อลดขนาดของรูปให้เหลือเฉพาะ ส่วนที่ต้องการ ดังรูปที่ 3.13 ซึ่งจะช่วยลดความผิดพลาดเมื่อนำตัวอักษรในภาพไปทำการประมวลผลต่อไป



รูปที่ 3.13 ภาพแสดงตัวอักษรบนแผ่นป้ายทะเบียนที่มีความสมบูรณ์หลังผ่านคำสั่ง 'medfilt2'

บทที่ 4

การแยกตัวอักษรและตัวเลขออกจากแผ่นป้ายทะเบียนรถยนต์ และการจัดเก็บในฐานข้อมูล

ในบทนี้จะกล่าวถึงการแยกส่วนที่เป็นตัวอักษรและตัวเลขออกจากแผ่นป้ายทะเบียน เพื่อนำตัวอักษรและตัวเลขที่ได้นี้มาทำการวิเคราะห์เพื่อเก็บไว้เป็นฐานข้อมูลและใช้ในการเปรียบเทียบกับตัวอักษรและตัวเลขที่ต้องการ

4.1 การแยกส่วนที่เป็นตัวอักษรและตัวเลขออกจากแผ่นป้ายทะเบียนรถยนต์

จากบทที่ 3 จะได้ส่วนที่เป็นแผ่นป้ายทะเบียนที่ผ่านการปรับปรุงคุณภาพให้มีความสมบูรณ์มากยิ่งขึ้น โดยในบทนี้จะกล่าวถึง การแยกตัวอักษรและตัวเลขออกจากแผ่นป้ายทะเบียน ซึ่งมีขั้นตอนดังต่อไปนี้

1. เมื่อได้ส่วนที่เป็นแผ่นป้ายทะเบียนที่ผ่านการปรับปรุงคุณภาพให้มีความสมบูรณ์มากขึ้น ดังรูปที่ 4.1 ทำการลาเบลกลุ่มพิกเซลสีขาวที่อยู่บนแผ่นป้ายทะเบียนโดยใช้คำสั่ง 'bwlabel' ซึ่งจะเป็นการกำหนดหมายเลขให้กับกลุ่มพิกเซลสีขาวที่อยู่บนแผ่นป้ายทะเบียน ซึ่งกลุ่มของพิกเซลสีขาวนี้ก็คือตัวอักษรและตัวเลข จากนั้นจะทำการกำหนดขอบเขตในการตัดตัวอักษรและตัวเลขเหล่านี้ออกจากแผ่นป้ายทะเบียน โดยการหาจุดพิกเซลสีขาวที่อยู่บนสุด ล่างสุด ซ้ายสุด และขวาสุดของกลุ่มพิกเซลสีขาวแต่ละกลุ่ม แล้วทำการเพิ่มความห่าง 15 พิกเซล ให้กับขอบเขตของตัวอักษรแต่ละตัว ดังรูปที่ 4.2 เพื่อให้โปรแกรมไม่ตัดภาพขีดขอบพิกเซลสีขาวของตัวอักษรมากเกินไป



รูปที่ 4.1 ภาพแสดงส่วนที่เป็นแผ่นป้ายทะเบียนที่ปรับปรุงคุณภาพแล้ว



รูปที่ 4.2 ภาพแสดงตัวอย่างในการกำหนดขอบเขตในการตัดตัวอักษรและตัวเลข
ที่เพิ่มความห่างให้กับขอบเขตของตัวอักษรและตัวเลขแล้ว

2. หลังจากได้ขอบเขตในการตัดตัวอักษรแต่ละตัวจากขั้นตอนที่ 1 แล้วก็ทำการตัดตัวอักษร
และตัวเลขแต่ละตัวออกมาโดยใช้คำสั่ง 'imcrop' ซึ่งเป็น Tool คำสั่งใน MATLAB ก็จะได้รูป
ตัวอักษรที่ถูกตัดแยกออกจากกัน ดังรูปที่ 4.3



รูปที่ 4.3 ภาพแสดงตัวอักษรแต่ละตัวหลังผ่านคำสั่ง 'imcrop'

4.2 การรวบรวมตัวอักษรเพื่อนำมาทำเป็นแม่แบบ จำนวน 49 ตัว

หลังจากที่สามารถแยกส่วนที่เป็นตัวอักษรออกจากแผ่นป้ายทะเบียนได้แล้ว ก็จะเข้าสู่ขั้นตอน
ในการเก็บรวบรวมตัวอักษรเพื่อใช้เป็นแม่แบบ ซึ่งตัวอักษรที่ถูกใช้เป็นชื่อจังหวัด บนแผ่นป้าย
ทะเบียนนี้มีทั้งหมด 33 ตัว จากพยัญชนะไทยทั้งหมด 49 ตัว ซึ่งจะประกอบไปด้วย ตัวพยัญชนะ
ไทยทั้งหมด 39 ตัว จากพยัญชนะไทยทั้งหมด 44 ตัวและตัวเลขอารบิกอีกจำนวน 10 ตัว โดย
พยัญชนะไทยที่กรมขนส่งทางบกไม่นำมาใช้บนแผ่นป้ายทะเบียน มีทั้งหมด 5 ตัว ซึ่งจะประกอบไป
ด้วย “จ” “ค” “ช” “ต” “ฎ” รวมเป็นตัวอักษรทั้งหมด 49 ตัวอักษร ซึ่งจะมีขั้นตอนในการเก็บ
รวบรวมตัวอักษรดังนี้

1. ทำการเลือกรูปภาพที่ตัวอักษรบนแผ่นป้ายทะเบียนมีความสมบูรณ์ชัดเจน ไม่มีส่วนที่เป็น
หัวน็อตหรือตะปูติดอยู่ ซึ่งเมื่อนำภาพมาทำการแยกเอาส่วนที่เป็นตัวอักษรออกจากแผ่นป้ายทะเบียน
แล้วจะได้ตัวอักษรที่มีความสมบูรณ์และเด่นชัด ดังเช่นรูปที่ 4.4 ซึ่งจะแสดงตัวอย่างของรูปแผ่นป้าย
ทะเบียนที่เห็นตัวอักษรชัดเจน เหมาะแก่การนำมาแยกแยะตัวอักษรเพื่อใช้เป็นแม่แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ภาพแสดงตัวอย่างของแผ่นป้ายทะเบียนที่มีความเหมาะสม
ในการนำตัวอักษรมาทำเป็นแม่แบบ

2. นำภาพในขั้นตอนที่ 1 มาเข้ากระบวนการแยกแยะส่วนที่เป็นแผ่นป้ายทะเบียนออกจากภาพซึ่งขั้นตอนและวิธีการที่ใช้ได้กล่าวไปแล้วในบทที่ 3
3. นำภาพที่ได้จากขั้นตอนที่ 2 มาทำการแยกส่วนที่เป็นตัวอักษรออกจากแผ่นป้ายทะเบียน ซึ่งหากตัวอักษรที่ได้มีความผิดเพี้ยนไม่มากนักก็นำตัวอักษรที่ได้เหล่านั้นมาเข้าโปรแกรม paint เพื่อปรับปรุงแก้ไขตัวอักษรให้มีความสมบูรณ์มากที่สุด เพื่อใช้เป็นแม่แบบหรือฐานข้อมูลที่ใช้ในการเปรียบเทียบกับตัวอักษรที่ต้องการ
4. นำภาพจากขั้นตอนที่ 3 มาทำการปรับขนาดของตัวอักษรบนหมายเลขทะเบียนและในส่วน
ของชื่อจังหวัดให้มีขนาดเท่ากัน โดยขนาดของตัวอักษร เท่ากับ 68x98 พิกเซล โดยใช้คำสั่ง
imresize ซึ่งเป็น Tool ของ MATLAB เนื่องจากขนาดของแม่แบบของตัวอักษรและตัวเลขที่ได้จาก
รุ่นที่ มีขนาด 68x98 พิกเซล จำนวน 49 ตัว ดังรูปที่ 4.5



รูปที่ 4.5 ชุดแม่แบบตัวอักษรทั้ง 49 ตัว ซึ่งประกอบไปด้วยพยัญชนะไทยจำนวน 39 ตัว และตัวเลขอารบิกจำนวน 10 ตัว

4.3 การวิเคราะห์ตัวอักษรแม่แบบโดยใช้วิธี FIR System

การวิเคราะห์ตัวอักษรแม่แบบโดยใช้วิธี FIR System เป็นวิธีหนึ่งที่ใช้ในการวิเคราะห์หาความเป็นเอกลักษณ์ของตัวอักษรแต่ละตัว โดยใช้ค่าสัมประสิทธิ์ที่ได้จากการหา Discrete Cosine Transform (DCT) มาหาค่าความเป็นเอกลักษณ์ของตัวอักษรแต่ละตัว $h(m)$ โดยในการหาค่าความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเอกลักษณ์ของตัวอักษร h (m) นั้น มีวิธีในการหาค่าความเป็นเอกลักษณ์โดยใช้วิธีการหาขอบ (Edge Detection) ของตัวอักษร

4.3.1 การหาค่า h (m) โดยใช้วิธีการหาขอบ (Edge Detection) ของตัวอักษร

1. นำภาพแม่แบบของตัวอักษรต่างๆที่ได้จากโปรแกรมการแยกส่วนตัวอักษรออกจากแผ่นป้ายทะเบียนรถยนต์ ซึ่งภาพแม่แบบที่ได้จะเป็นภาพระดับสีเทา ดังรูปที่ 4.6



รูปที่ 4.6 ตัวอย่างแม่แบบของตัวเลข “5” ในระดับสีเทา

2. โหลดภาพแม่แบบที่ได้จัดเก็บไว้ในฐานข้อมูล เข้าสู่โปรแกรม MATLAB ด้วยคำสั่ง `imread` จากนั้นทำการแปลงภาพเป็นภาพสองระดับ (Binary Image) ด้วยคำสั่ง `im2bw` แล้วนำภาพสองระดับที่ได้มาเข้ากระบวนการหาขอบภาพแบบโซเบล จะได้ผลลัพธ์ดังรูปที่ 4.7



รูปที่ 4.7 ภาพเลข “5” ที่ผ่านการหาขอบภาพด้วยวิธีโซเบล (Sobel's Operators)

3. นำภาพที่ผ่านการหาขอบภาพด้วยวิธีโซเบลมาเข้ากระบวนการหาแกนกลาง (Skeleton) ซึ่งจะได้ผลลัพธ์ดังรูปที่ 4.8



รูปที่ 4.8 ภาพผลลัพธ์ที่ได้จากการหาแกนกลาง (Skeleton)

4. นำภาพที่ผ่านการหาแกนกลางของ (Skeleton) จากขั้นตอนที่ 3 มาทำการหาตำแหน่งพิกัด (Coordinate) ของเส้นขอบ ซึ่งเป็นการเก็บตำแหน่งพิกัดในแนวแกนอน x และแนวแกนตั้ง y ดังรูปที่ 4.9 (ซึ่งส่วนที่เป็นสีแดงคือตำแหน่งพิกัดที่ได้จากการหา Coordinate โดยใช้คำสั่ง plot ซึ่งเป็น Tool คำสั่งใน MATLAB เพื่อทำการพล็อตจุดที่เก็บค่าตำแหน่งพิกัดที่ได้มาทับลงไปบนเส้นขอบ)



รูปที่ 4.9 ภาพแสดงการพล็อตจุดที่เก็บค่าตำแหน่งพิกัดที่ได้มาทับลงไปบนเส้นขอบ (Edge) เพื่อเช็คความถูกต้องของตำแหน่งพิกัดที่เก็บมาได้

5. ทำการหาค่า Eigen Vector และ Eigen Value โดยจะเริ่มจากการนำตำแหน่งพิกัด (Coordinate) ซึ่งเป็นพิกัดในแนวแกนอน x และพิกัดในแนวแกนตั้ง y ที่ได้มาจากขั้นตอนที่ 3 มาหาค่าเฉลี่ยของแต่ละแกนด้วยสมการที่ (4.1)

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad \text{และ} \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (4.1)$$

โดยที่ x_i แทนค่าตำแหน่งพิกัดในแนวแกนอน x และ y_i แทนค่าตำแหน่งพิกัดในแนวแกนตั้ง y

ต่อมานำตำแหน่ง x_i และ y_i แต่ละจุด ลบด้วยค่าเฉลี่ย \bar{x} และ \bar{y} ที่ได้ จะได้สมการที่ (4.2)

$$\text{และ } \tilde{Y}_i = y_i - \bar{y}_i \quad (4.2)$$

จากนั้นทำการหาค่าของ Covariant Matrix (C) ด้วยสมการที่ (4.3)

$$C = \frac{1}{N} \sum_{i=1}^N \begin{bmatrix} (X_i - \bar{X}_i) & (X_i - \bar{X}_i)(Y_i - \bar{Y}_i) \\ (Y_i - \bar{Y}_i) & (Y_i - \bar{Y}_i)(X_i - \bar{X}_i) \end{bmatrix} \quad (4.3)$$

เมื่อ

$$\text{Cov} = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X}_i)(Y_i - \bar{Y}_i)$$

จะได้ค่าของ Covariant Matrix (C) ดังสมการที่ (4.4)

$$C = \begin{bmatrix} \text{Cov}(X_i - \bar{X}_i) & \text{Cov}(X_i - \bar{X}_i)(Y_i - \bar{Y}_i) \\ \text{Cov}(Y_i - \bar{Y}_i) & \text{Cov}(Y_i - \bar{Y}_i)(X_i - \bar{X}_i) \end{bmatrix} \quad (4.4)$$

จากนั้นทำการหาค่าของ Eigen vector และ Eigen value ด้วยคำสั่ง eig ซึ่งเป็น Tool คำสั่งที่อยู่ในโปรแกรม MATLAB โดยจะแสดงได้ดังสมการที่ (4.6)

$$[u, v] = \text{eig}(C) \quad (4.5)$$

เมื่อ u คือ Eigen vector

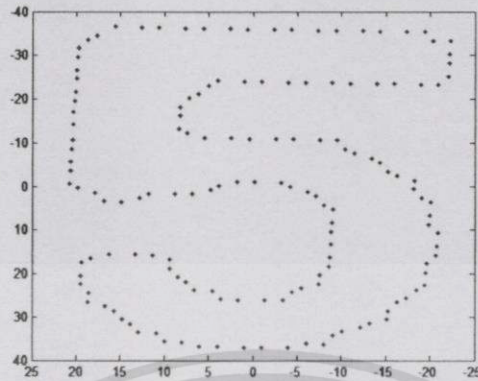
v คือ Eigen value

จากนั้นจึงนำค่า Eigen Vector ที่ได้มาคูณกับ x_i และ \tilde{Y}_i ของแกนกลาง (Skeleton) เพื่อทำการ Normalize ค่าหรือเป็นการหมุนตัวอักษรแต่ละตัวให้มาอยู่ในระนาบแกนเดียวกันก่อนนำค่าที่ได้มาทำการ Interpolation ในขั้นตอนต่อไป

6. เนื่องจากภาพถ่ายแต่ละภาพนั้นมีจำนวนของพิกเซลที่ไม่เท่ากันทำให้ตัวอักษรหรือตัวเลขที่ได้มีขนาดที่แตกต่างกัน จึงจำเป็นต้องมีการทำให้ค่าตำแหน่งพิกัดจุดที่เก็บค่ามาได้นั้นมีจำนวนที่เท่ากัน ด้วยวิธีการ Interpolation เมื่อนำไปหาค่าสัมประสิทธิ์ก็จะทำให้มีจำนวนที่เท่ากัน ซึ่งจะทำได้ง่ายและมีความถูกต้องมากขึ้นในการนำค่าสัมประสิทธิ์ที่ได้ไปประมวลผลในขั้นตอนต่อไป โดยการ Interpolation นี้จะใช้คำสั่ง interp1 ซึ่งเป็น Tool คำสั่งใน MATLAB โดยวิธีในการสอดแทรกนั้นมีด้วยกันหลายวิธี ซึ่งวิธีที่เลือกใช้มี 4 วิธีด้วยกัน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) 'nearest' คือ วิธีการสอดแทรกแบบแทรกจุดข้างเคียง จะได้ผลลัพธ์ดังรูปที่ 4.10



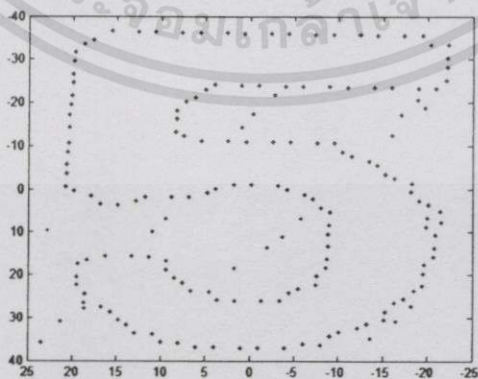
รูปที่ 4.10 ผลลัพธ์แสดงการ Interpolation ด้วยวิธี 'nearest'

2) 'linear' คือ วิธีการสอดแทรกแบบเส้นตรง จะได้ผลลัพธ์ดังรูปที่ 4.11



รูปที่ 4.11 ผลลัพธ์แสดงการ Interpolation ด้วยวิธี 'linear'

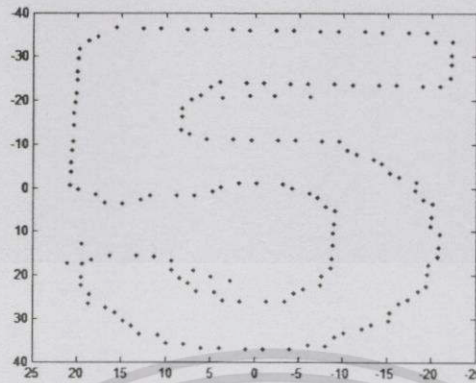
3) 'spline' คือ วิธีการสอดแทรกแบบลูกบาศก์อิสระ จะได้ผลลัพธ์ดังรูปที่ 4.12



รูปที่ 4.12 ผลลัพธ์แสดงการ Interpolation ด้วยวิธี 'spline'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) 'cubic' คือ วิธีการสอดแทรกแบบต่อเนื่องเป็นช่วง จะได้ผลลัพธ์ดังรูปที่ 4.13



รูปที่ 4.13 ผลลัพธ์แสดงการ interpolation ด้วยวิธี 'cubic'

จากการ Interpolation ด้วยวิธีทั้ง 4 นั้นจะเห็นได้ว่าการ Interpolation วิธีแบบ 'nearest' นั้นจะให้ผลลัพธ์ที่ดีที่สุด เพราะไม่มีจุดรบกวน (Noise)

7. นำค่าตำแหน่งที่ผ่านการ Interpolation แล้วมาเข้าสมการ Discrete Cosine Transform (DCT) เพื่อหาค่าของสัมประสิทธิ์ของตำแหน่งทั้งแนวแกนนอน x และแนวแกนตั้ง y ตามสมการของ Discrete Cosine Transform ดังสมการที่ (4.6)

$$f(k) = w(k) \sum_{n=1}^N \frac{x(n) \cos \pi(2n-1)(k-1)}{2N} \quad (4.6)$$

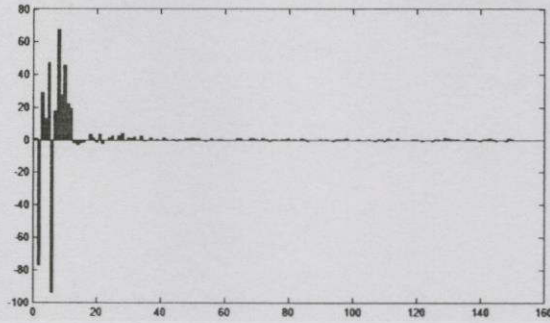
$$g(k) = w(k) \sum_{n=1}^N \frac{y(n) \cos \pi(2n-1)(k-1)}{2N}$$

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & \text{โดยที่ } k=1 \\ \frac{2}{\sqrt{N}} & \text{โดยที่ } 2 \leq k \leq N \end{cases}$$

เมื่อ k คือ จำนวนพิกัดจุดทั้งหมดของภาพที่ผ่านการ interpolation แล้ว

จากสมการข้างต้นเมื่อนำค่าตำแหน่งพิกัดจุดของแกนกลาง (Skeleton) มาเข้าสมการ DCT จะได้ค่าสัมประสิทธิ์ทั้งสองแนวแกน คือ สัมประสิทธิ์ในแนวแกนนอน $f(k)$ และสัมประสิทธิ์ในแนวแกนตั้ง $g(k)$ แล้วนำค่าสัมประสิทธิ์ที่ได้ไปพล็อตกราฟ จะได้ผลลัพธ์ ดังรูปที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 4.14 กราฟที่ผ่าน Discrete Cosine Transform (DCT)

(ก) สัมประสิทธิ์ในแนวแกนนอน $f(k)$

(ข) สัมประสิทธิ์ในแนวแกนตั้ง $g(k)$

8. จากขั้นตอนที่ 7 จะได้ค่าสัมประสิทธิ์ในแนวแกนตั้งและแนวแกนนอน หลังจากนั้นจะทำการนำค่าสัมประสิทธิ์ทั้งสองแนวแกนมาเข้าระบบ FIR System เพื่อหาค่าสัมประสิทธิ์ความเป็นเอกลักษณ์ $h(m)$ ของตัวอักษร จะได้ผลลัพธ์เป็นค่า $h(m)$ ออกมา แล้วนำค่า $h(m)$ ที่ได้นี้ไปพล็อตกราฟ เพื่อดูคุณลักษณะและเอกลักษณ์ของตัวอักษรผ่านกราฟ ซึ่งเราจะใช้คุณลักษณะของกราฟ $h(m)$ นี้ ในการตรวจเช็คความแตกต่างหรือเอกลักษณ์ของตัวอักษรแต่ละตัวใน FIR System โดยจะสามารถอธิบายลักษณะการทำงานของระบบ FIR ได้จากสมการที่ (4.7)

$$g_n = \sum_{m=0}^L h_m * f_{n-m} \quad (4.7)$$

เมื่อ f_{n-m} คือ เวกเตอร์ค่าอินพุตของระบบ FIR $f(k)$

$h(m)$ คือ เวกเตอร์ค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์ ซึ่งเป็นตัวบอกคุณลักษณะและเอกลักษณ์ ของข้อมูล

g_n คือ เวกเตอร์ค่าเอาต์พุตของระบบ FIR $g(k)$

L คือ จำนวนลำดับของระบบ FIR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากอินพุต (f_{n-m}) ของระบบ FIR เป็นเวกเตอร์จึงไม่สามารถคูณกันได้ด้วยเวกเตอร์เอกลักษณ์ของข้อมูล $h(m)$ ดังนั้นเพื่อให้มีการคูณกันได้ของระบบ FIR เราจึงจัดเรียงค่าอินพุต (f_{n-m}) ใหม่ให้อยู่ในรูปเมทริกซ์สามเหลี่ยมล่าง(Lower Triangular Matrix ,Q) ดังสมการด้านล่าง

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_n \end{bmatrix} = \begin{bmatrix} f_0 & 0 & \dots & 0 \\ f_1 & f_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_m & f_{m-1} & \dots & f_{m-L} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{bmatrix}$$

f คือ DCT ค่าสัมประสิทธิ์ DCT ของพิกัดจุดในแนวแกน X

g คือ ค่าสัมประสิทธิ์ DCT ของพิกัดจุดในแนวแกน Y

h คือ เวกเตอร์ค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์

เราสามารถหาค่าสัมประสิทธิ์ผลตอบสนองของอิมพัลส์ $h(m)$ จากสมการ (4.8)

$$h_m = [(Q)^T * Q]^{-1} * Q^T * g_n \quad (4.8)$$

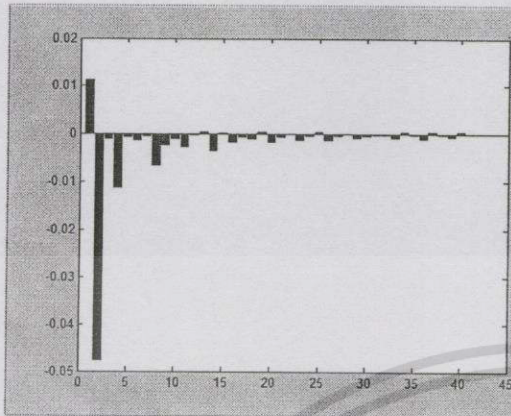
เมื่อนำผลลัพธ์ $h(m)$ มาพล็อตกราฟเพื่อที่วิเคราะห์หาความเป็นเอกลักษณ์ของตัวอักษร แต่ละตัวว่ามีความเหมือนหรือแตกต่างกันอย่างไร โดยจะแสดงผลดังรูปที่ 4.15



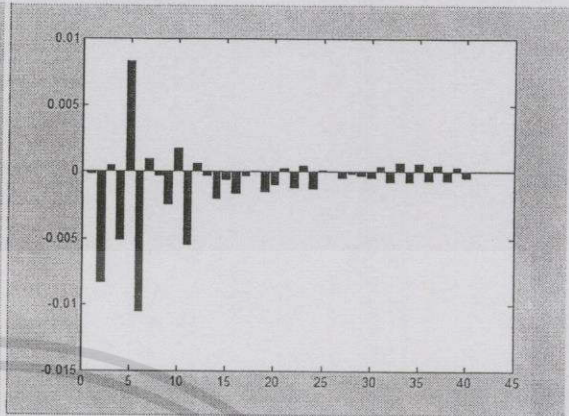
รูปที่ 4.15 กราฟผลลัพธ์ $h(m)$ ของเลข "5" ที่ได้จาก FIR System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

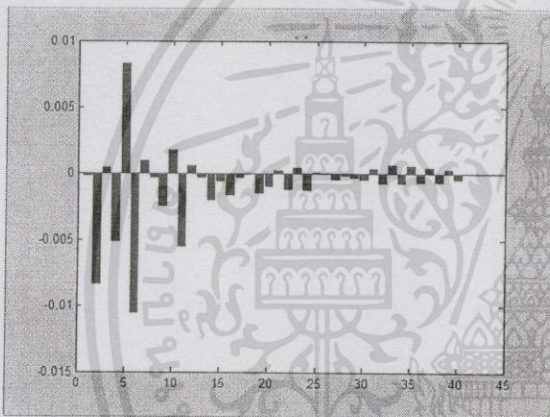
4.3.2 การแสดงค่าสัมประสิทธิ์ของ FIR ($h(m)$) ในรูปแบบของกราฟ



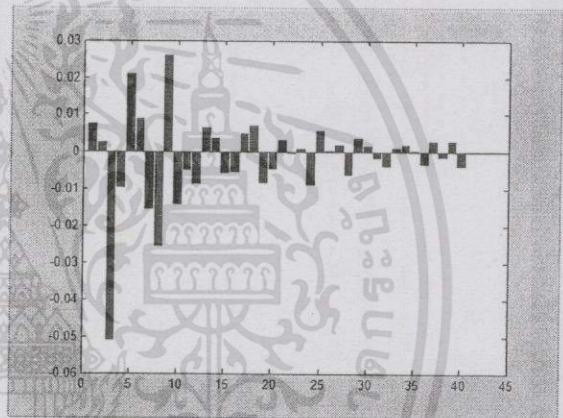
ตัวเลข 0



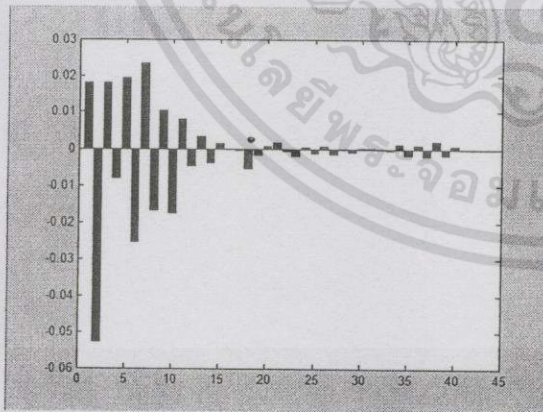
ตัวเลข 1



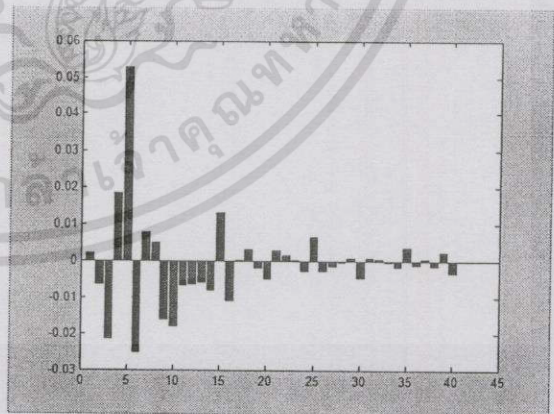
ตัวเลข 2



ตัวเลข 3

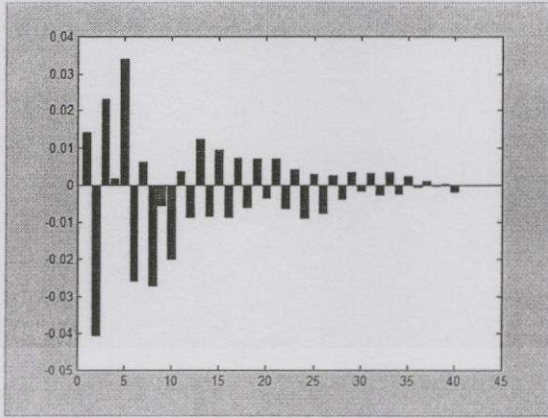


ตัวเลข 4

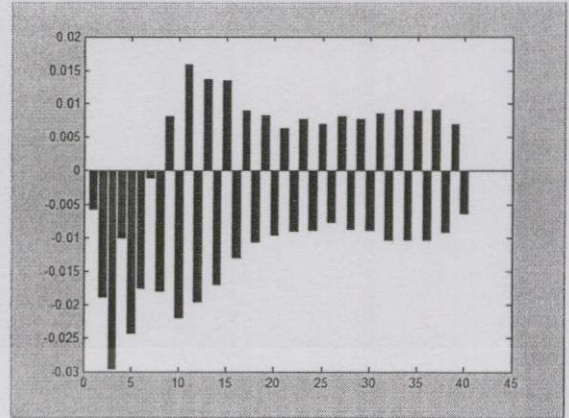


ตัวเลข 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



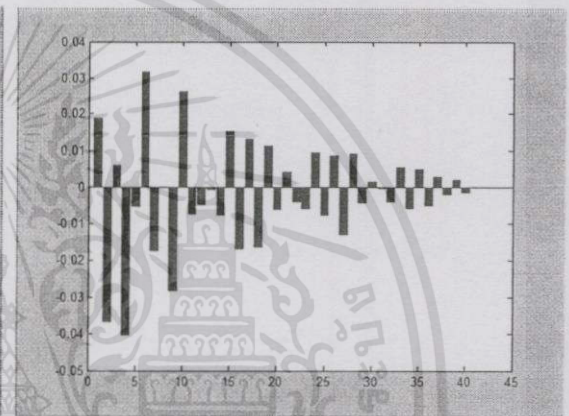
ตัวเลข 6



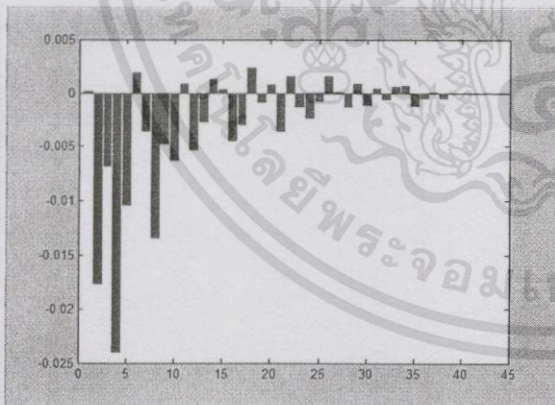
ตัวเลข 7



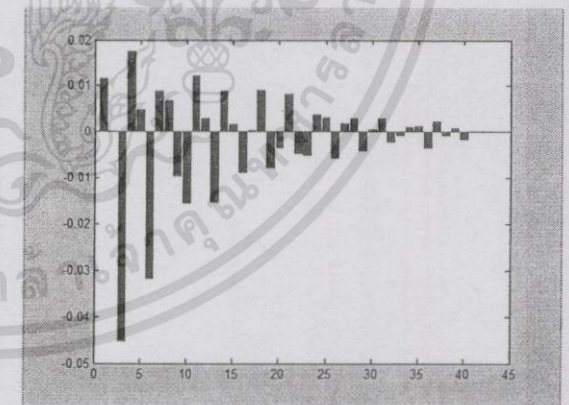
ตัวเลข 8



ตัวเลข 9

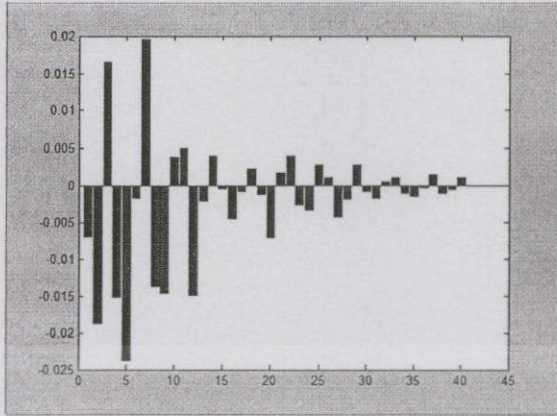


พัญชนะ ก

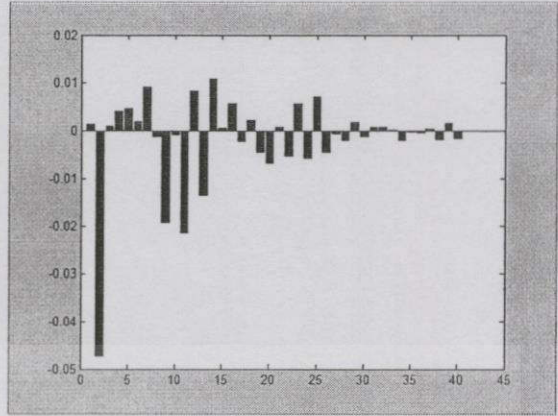


พัญชนะ ข

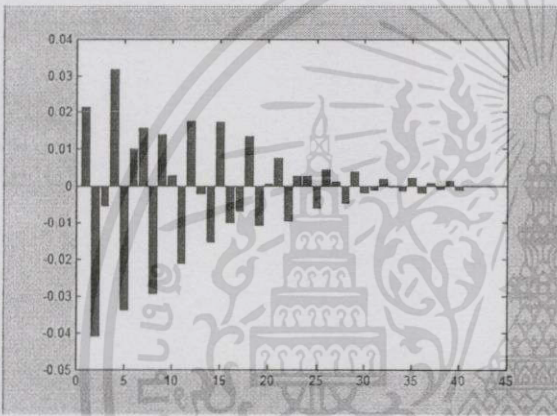
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



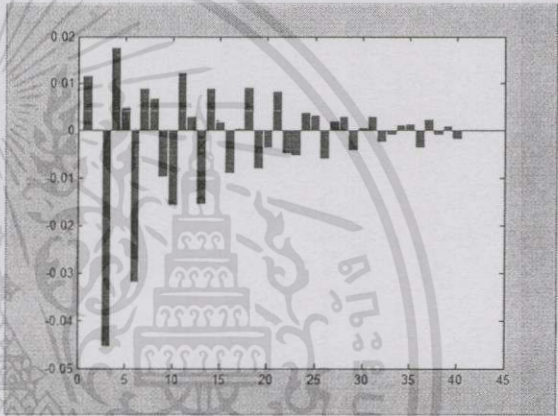
พัญชนะ ค



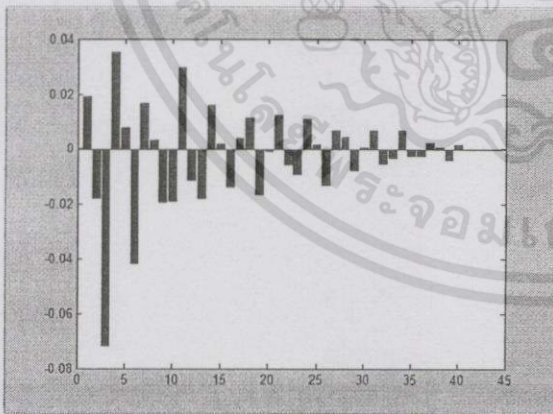
พัญชนะ ฉ



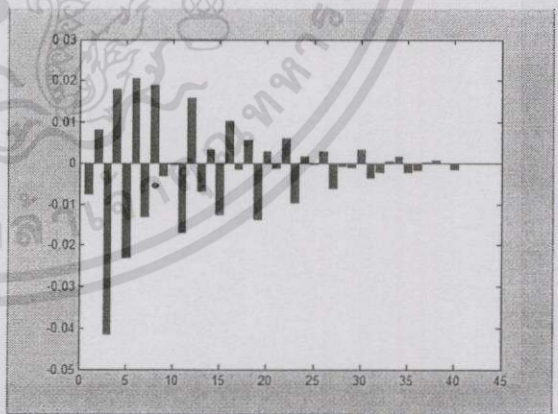
พัญชนะ ง



พัญชนะ จ

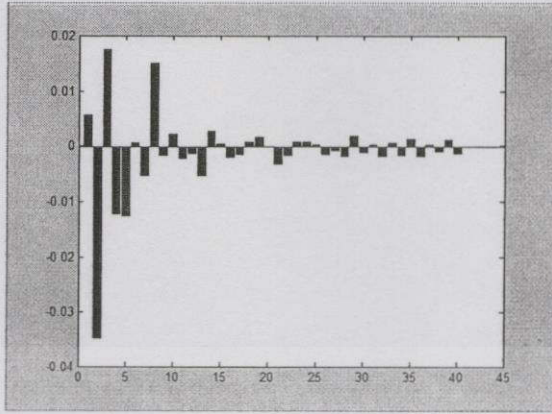


พัญชนะ ฉ

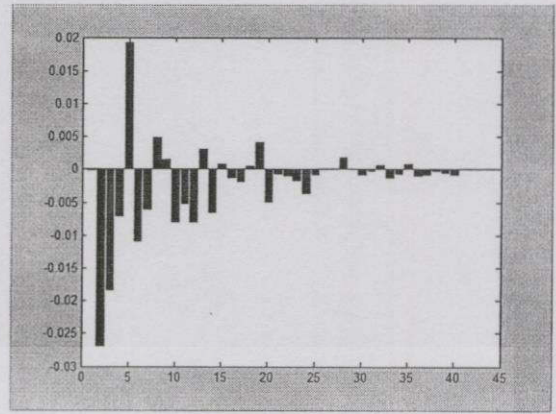


พัญชนะ ช

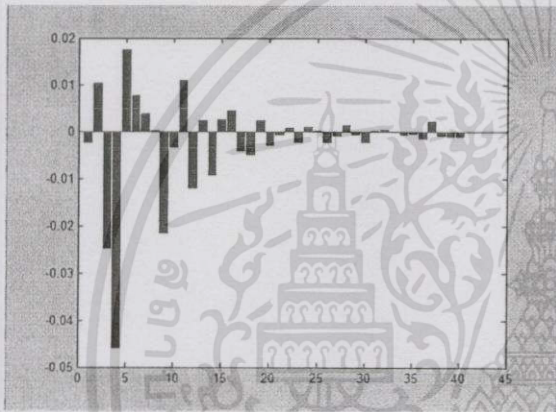
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



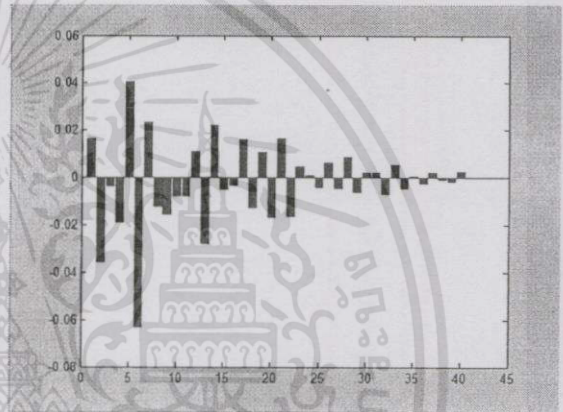
พฤษภาคม



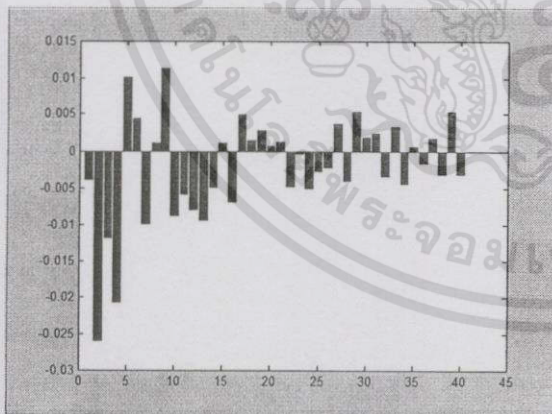
พฤษภาคม



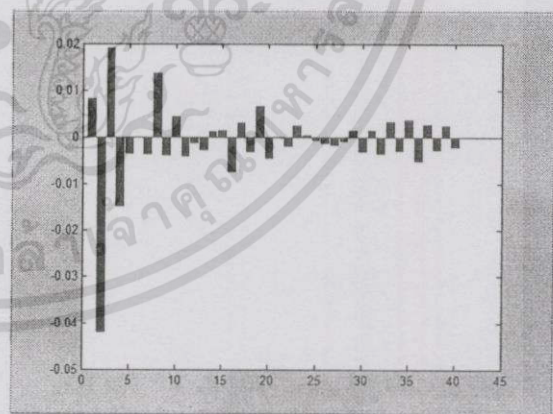
พฤษภาคม



พฤษภาคม

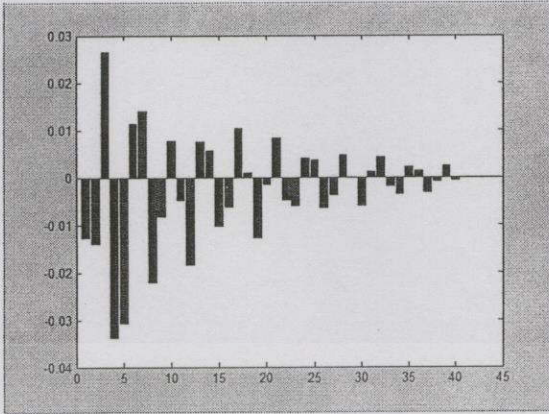


พฤษภาคม

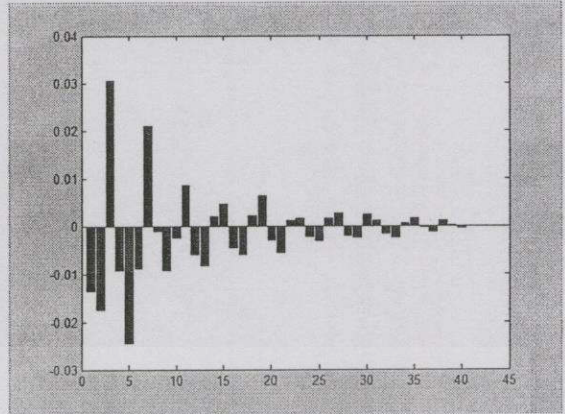


พฤษภาคม

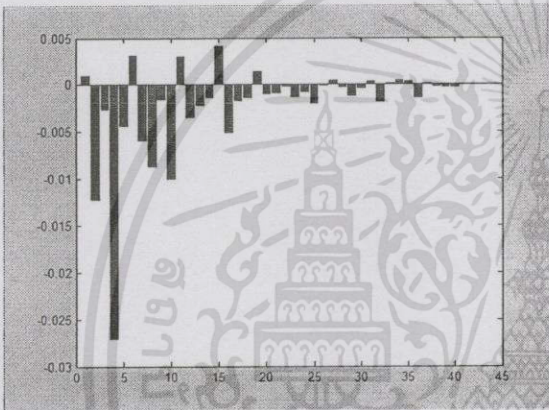
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



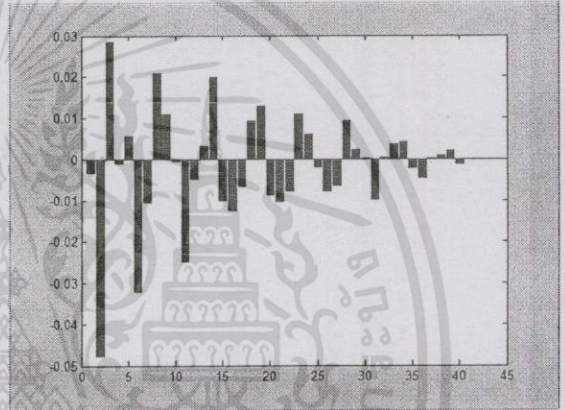
พัญชนะ ด



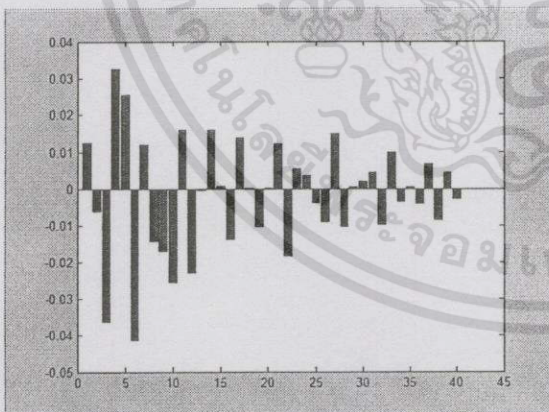
พัญชนะ ต



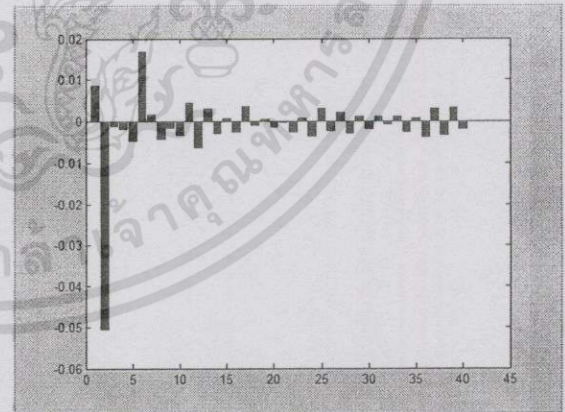
พัญชนะ ถ



พัญชนะ ท

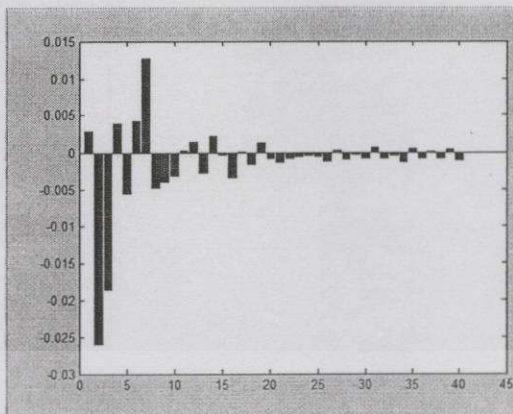


พัญชนะ ฐ

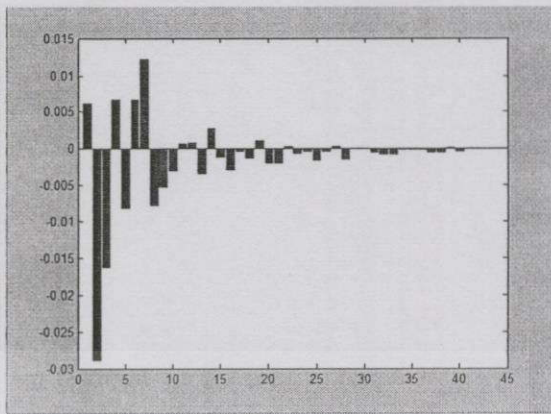


พัญชนะ ฑ

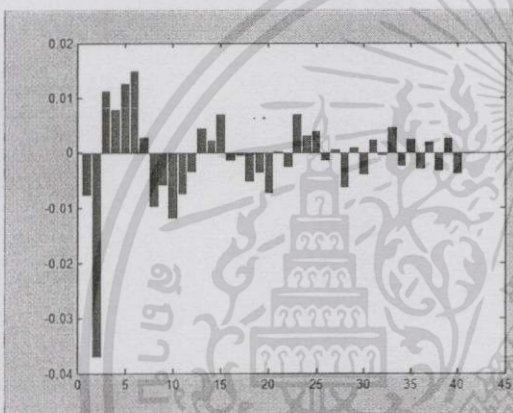
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



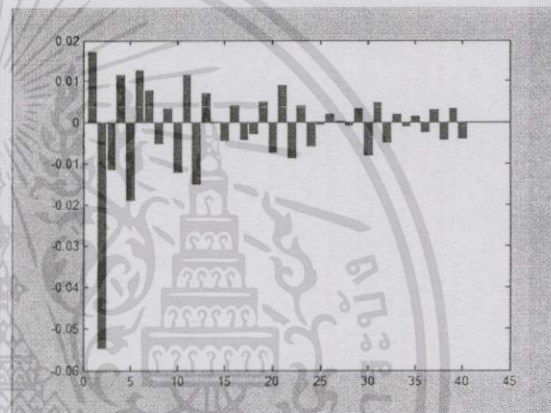
พัญชนะ บ



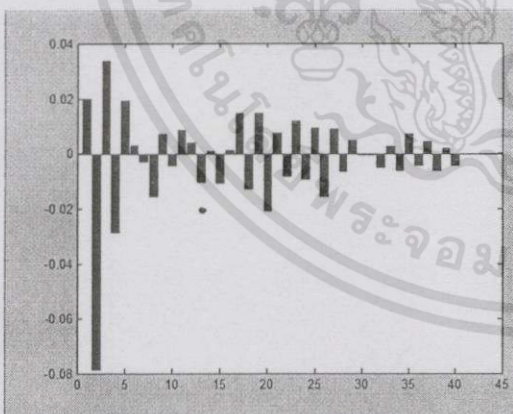
พัญชนะ ป



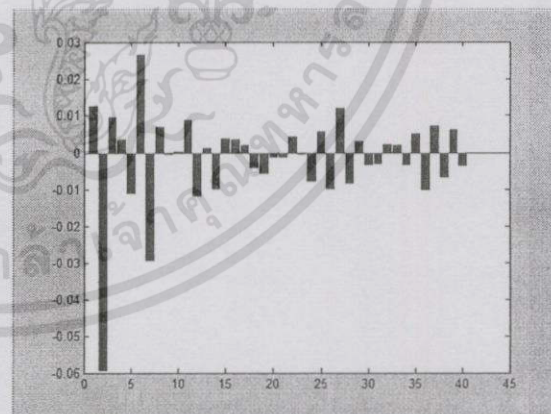
พัญชนะ ฝ



พัญชนะ ฟ

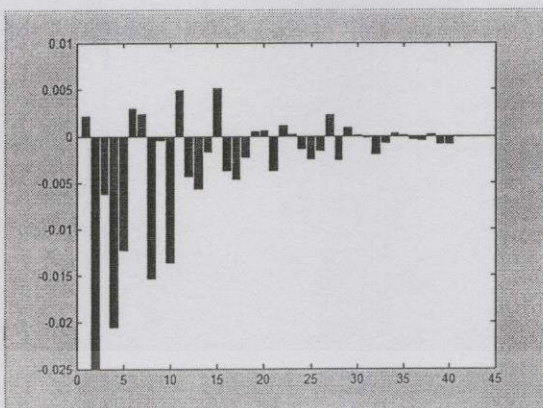


พัญชนะ พ

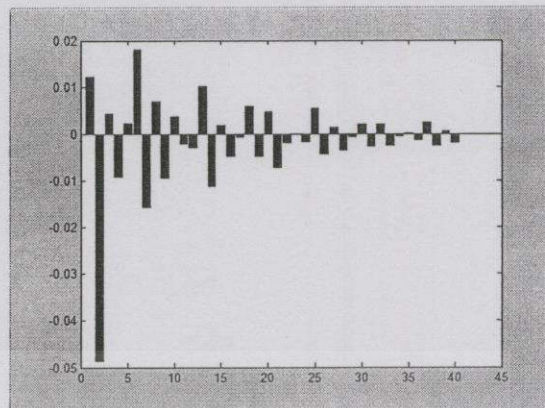


พัญชนะ ฬ

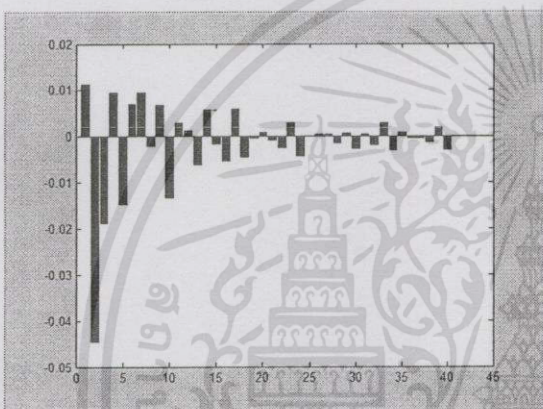
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



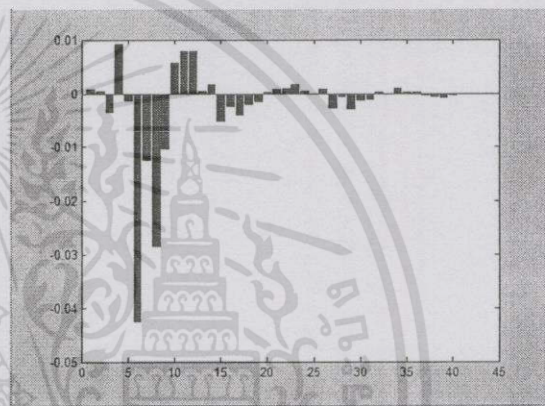
พฤษภาคม



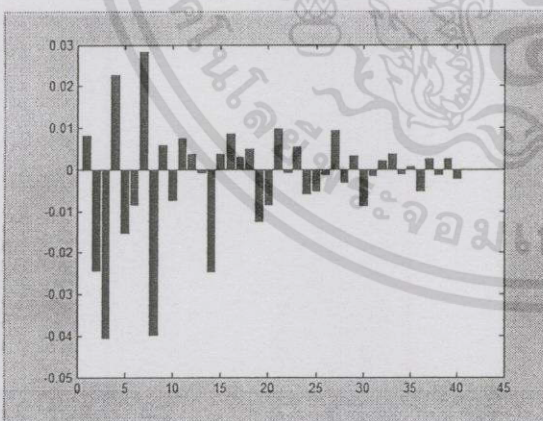
พฤษภาคม



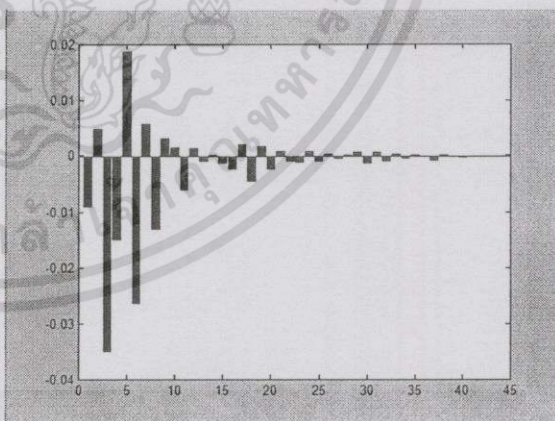
พฤษภาคม



พฤษภาคม

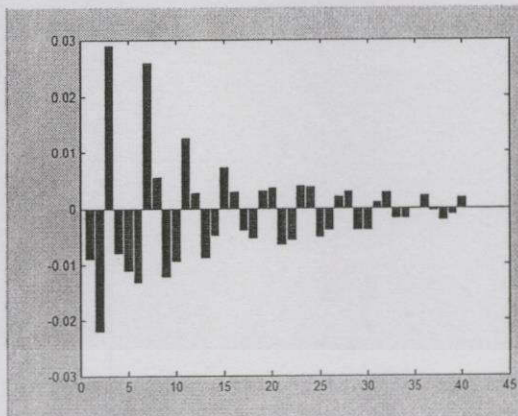


พฤษภาคม

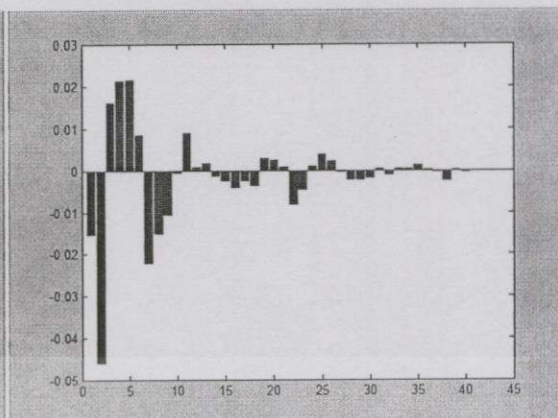


พฤษภาคม

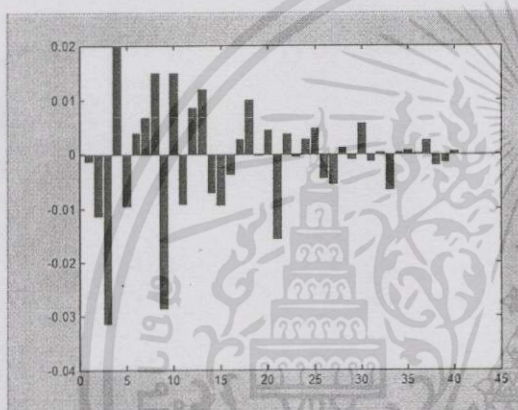
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



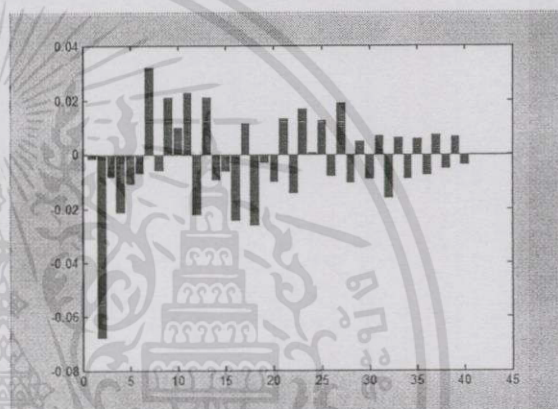
พัญชนะ ค



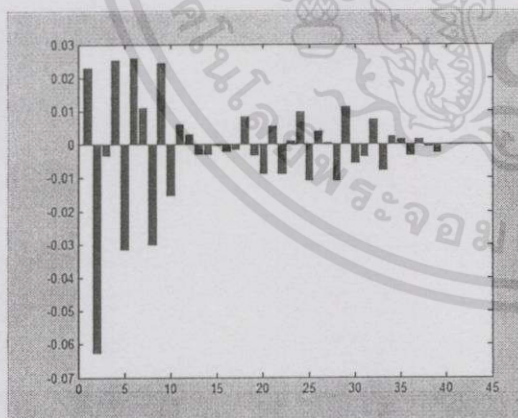
พัญชนะ ซ



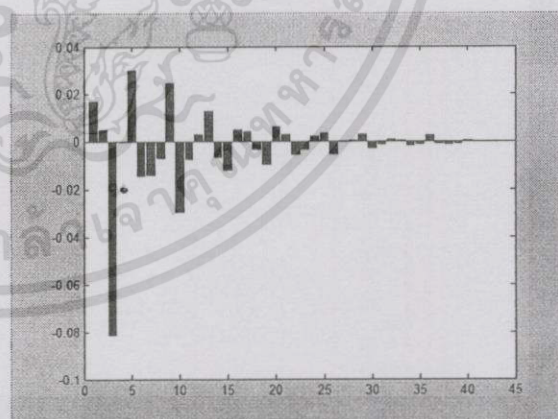
พัญชนะ ส



พัญชนะ ฮ

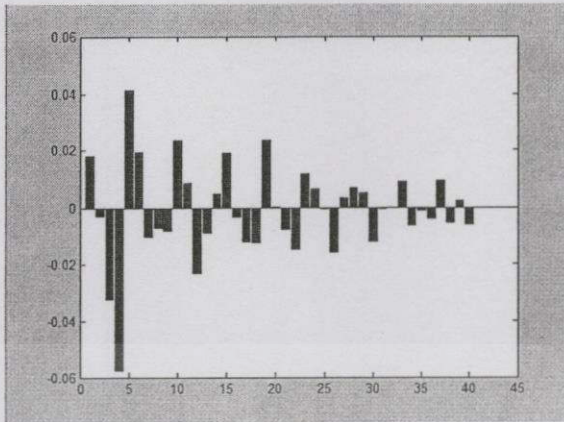


พัญชนะ ฟ



พัญชนะ อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



พยัญชนะ ฮ

รูปที่ 4.16 ค่าสัมประสิทธิ์ของ FIR ($h(m)$) ของตัวเลขและตัวอักษรทั้งหมด

4.4 การเก็บฐานข้อมูลของตัวอักษรและตัวเลข

การเก็บฐานข้อมูล

เราสามารถเก็บฐานข้อมูลของตัวอักษรและตัวเลขโดยใช้คำสั่ง Save ในโปรแกรม MATLAB save filename obj1 obj2 ...

(ชื่อไฟล์ที่เราตั้ง)(ค่า h (ค่า Q)(ค่า...)

ตัวอย่าง save SD1 h1

(ชื่อไฟล์ที่เราตั้ง (SD1)(ค่า $h1$ ตัวแปรที่เราเขียนใน m-file)

ไฟล์ที่เซฟจะมีนามสกุล .mat4

การเรียกฐานข้อมูล

เราสามารถเก็บฐานข้อมูลของตัวอักษรและตัวเลขโดยใช้คำสั่ง Load ในโปรแกรม MATLAB

ใช้คำสั่ง load filename

(ชื่อไฟล์ที่เราตั้ง) (ชื่อไฟล์ที่เราต้องการจะโหลด)

ตัวอย่าง load SD1

(ชื่อไฟล์ที่เราตั้ง) (ไม่จำเป็นต้องพิมพ์ $h1$ เพราะข้อมูลได้อยู่ในชื่อไฟล์แล้ว)

โดยเราจะเก็บค่า h ของข้อมูลแต่ละตัวอักษรตั้งแต่ W1-W49 และเก็บลงในชื่อไฟล์ตั้งแต่ Q1-Q49

ยกตัวอย่างการเก็บค่า h ของฐานข้อมูลตัวเลข 5

Save Q6 W6;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งโปรแกรมจะเก็บค่าของ W6 ลงในไฟล์ Q6

หลังจากนั้นหากเราต้องการเรียกค่า h จากฐานข้อมูลตัวเลข 5 ออกมา เราจะใช้คำสั่ง ดังนี้
load Q6;

โดยเราจะได้ค่า h ของตัวเลข 5 ซึ่งเก็บไว้อยู่ในตัวแปร W6

ภาพที่ 4.17 จะเป็นการแสดงตำแหน่งการเก็บฐานข้อมูลของรู้อำเภอปายทะเบียนทั้งหมด โดยจะกำหนดตำแหน่งของฐานข้อมูลทั้งตัวอักษรและตัวเลขเป็นเลขสีแดงด้านมุมบนซ้าย



รูปที่ 4.17 ตำแหน่งของข้อมูลของแต่ละตัวอักษรและตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การรู้จำแผ่นป้ายทะเบียน

ในบทนี้จะกล่าวถึงวิธีการที่ใช้ในการเปรียบเทียบข้อมูลภาพจากแผ่นป้ายทะเบียนกับฐานข้อมูล เพื่อที่จะแสดงผลการรู้จำของแผ่นป้ายทะเบียนว่าทำได้อย่างไร

5.1 วิธีการเปรียบเทียบข้อมูลภาพจากแผ่นป้ายทะเบียนกับฐานข้อมูล

หลังจากที่เราได้ค่าความเป็นเอกลักษณ์ ($h(m)$) ของตัวเลข “5” ในบทที่ 4 แล้ว นำมา เปรียบเทียบค่าความเป็นเอกลักษณ์ของตัวอักษรและตัวเลขในฐานข้อมูล $h_{ref}^k(m)$ ทีละตัว และ คำนวณหาค่าความผิดพลาดของการเปรียบเทียบโดยใช้สมการที่ (5.1)

$$e^k(m) = h_{ref}^k(m) - h(m) \quad (5.1)$$

เมื่อ k คือตัวอักษรและตัวเลขในฐานข้อมูล ($k=1, \dots, 49$)

m คือจำนวนข้อมูลของค่าความเป็นเอกลักษณ์ของตัวอักษรและตัวเลข

และเมื่อเขียนในรูปของนอม (Norm) ของความผิดพลาดกำลังสองจะได้สมการที่ (5.2)

$$E^k = \sqrt{e^k(1)^2 + e^k(2)^2 + \dots + e^k(m)^2} \quad (5.2)$$

การหาค่าความผิดพลาดกำลังสองในสมการที่ (5.2) เป็นการเปรียบเทียบค่าสัมประสิทธิ์ของ FIR ของตัวอักษรหรือตัวเลขที่กำลังพิจารณากับค่าสัมประสิทธิ์ของ FIR ของตัวอักษรหรือตัวเลขทุกๆตัวที่อยู่ในฐานข้อมูล จากนั้นใช้คำสั่ง “min” ซึ่งเป็น Tool ใน MATLAB เพื่อหาค่า E^k ที่น้อยที่สุด ถ้าค่าความผิดพลาดที่น้อยที่สุดตรงกับตัวอักษรหรือตัวเลขใดในฐานข้อมูล โปรแกรมจะแสดงผลการรู้จำเป็นตัวอักษรหรือตัวเลขนั้น

5.2 การรู้จำหมวดตัวอักษรและหมายเลขทะเบียน

ในการรู้จำตัวอักษรและหมายเลขทะเบียน เราจะเปรียบเทียบตัวพยัญชนะบนป้ายทะเบียนกับตัวพยัญชนะทุกตัวในฐานข้อมูลที่เป็นตัวอักษร ในทำนองเดียวกันตัวเลขบนป้ายทะเบียนจะถูกเปรียบเทียบกับตัวเลขทุกตัวในฐานข้อมูลที่ตัวเลข โดยวิธีการ คือ แบ่งลักษณะของป้ายทะเบียนที่เป็นตัวอักษร 4 กลุ่ม ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กลุ่มตัวอักษร 7 ตัวอักษร



รูปที่ 5.1 ตัวอย่างของกลุ่ม 7 ตัวอักษร

2. กลุ่มตัวอักษร 6 ตัวอักษร



รูปที่ 5.2 ตัวอย่างของกลุ่ม 6 ตัวอักษร

3. กลุ่มตัวอักษร 5 ตัวอักษร



รูปที่ 5.3 ตัวอย่างของกลุ่ม 5 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กลุ่มตัวอักษร 4 ตัวอักษร

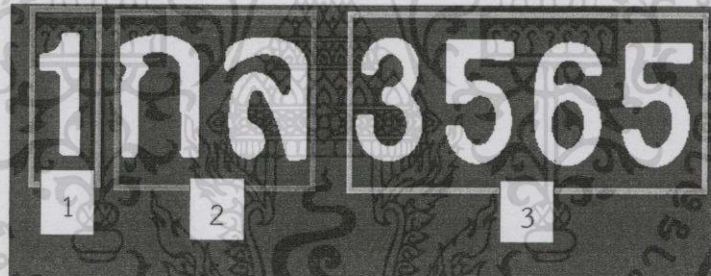


รูปที่ 5.4 ตัวอย่างของกลุ่ม 4 ตัวอักษร

หลังจากนั้นเราจะหาจำนวนตัวอักษรด้านบนของแผ่นป้ายทะเบียนโดยใช้คำสั่ง `bwlabel` และกำหนดเงื่อนไขพิจารณาการเปรียบเทียบกับฐานข้อมูลโดยให้เงื่อนไขในแต่ละกลุ่มตัวอักษรเป็นดังต่อไปนี้

1) กลุ่มตัวอักษร 7 ตัว

โดยที่ตัวอักษรกลุ่มแรกกับกลุ่มที่ 3 กำหนดให้เปรียบเทียบเฉพาะกับฐานข้อมูลของตัวเลข ส่วนกลุ่มที่ 2 กำหนดให้เปรียบเทียบเฉพาะกับฐานข้อมูลของส่วนพยัญชนะ ดังรูปที่ 5.5



รูปที่ 5.5 การแบ่งในการพิจารณาของกลุ่ม 7 ตัวอักษร

2) กลุ่มตัวอักษร 6 ตัว

โดยที่ตัวอักษรกลุ่มแรกกำหนดให้เปรียบเทียบทั้งฐานข้อมูลของตัวเลขและพยัญชนะ เนื่องจากตัวอักษรในกลุ่มแรกของทั้งสองภาพนี้ไม่สามารถแยกพิจารณาของฐานข้อมูลได้ ส่วนกลุ่มที่ 2 กำหนดให้เปรียบเทียบเฉพาะกับฐานข้อมูลของตัวเลข ดังรูปที่ 5.6



รูปที่ 5.6 การแบ่งในการพิจารณาของกลุ่ม 6 ตัวอักษร

3) กลุ่มตัวอักษร 5 ตัวอักษร

โดยที่ตัวอักษรกลุ่มแรกกำหนดให้เปรียบเทียบเฉพาะกับฐานข้อมูลของพยานะ ส่วนกลุ่มที่ 2 กำหนดให้เปรียบเทียบเฉพาะกับฐานข้อมูลของส่วนของตัวเลข ดังรูปที่ 5.7



รูปที่ 5.7 การแบ่งในการพิจารณาของกลุ่ม 5 ตัวอักษร

4) กลุ่มตัวอักษร 4 ตัวอักษร

โดยที่ตัวอักษรกลุ่มแรกกำหนดให้เปรียบเทียบเฉพาะกับฐานข้อมูลของพยานะ ส่วนกลุ่มที่ 2 กำหนดให้เปรียบเทียบเฉพาะกับฐานข้อมูลของส่วนของตัวเลข ดังรูปที่ 5.8



รูปที่ 5.8 การแบ่งในการพิจารณาของกลุ่ม 4 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การรู้จำชื่อจังหวัด

ในปัจจุบันประเทศไทยแบ่งการปกครองเป็นจังหวัด ซึ่งจะประกอบไปด้วยจังหวัดต่างๆทั้งสิ้น 77 จังหวัด

จังหวัดในประเทศไทยทั้ง 77 จังหวัดกระจายอยู่ในภาคต่างๆ ดังต่อไปนี้ ภาคเหนือ 9 จังหวัด ภาคตะวันออกเฉียงเหนือ 20 จังหวัด ภาคกลาง (รวมกรุงเทพมหานคร) 22 จังหวัด ภาคตะวันตก 5 จังหวัด ภาคตะวันออก 7 จังหวัด และภาคใต้ 14 จังหวัด

จังหวัดในประเทศไทยเรียงตามตัวอักษร มีดังนี้

1. กรุงเทพมหานคร	2. กระบี่	3. กาญจนบุรี	4. กาฬสินธุ์
5. กำแพงเพชร	6. ขอนแก่น	7. จันทบุรี	8. ฉะเชิงเทรา
9. ชลบุรี	10. ชัยนาท	11. ชัยภูมิ	12. ชุมพร
13. เชียงราย	14. เชียงใหม่	15. ตรัง	16. ตราด
17. ตาก	18. นครนายก	19. นครปฐม	20. นครพนม
21. นครราชสีมา	22. นครศรีธรรมราช	23. นครสวรรค์	24. นนทบุรี
25. นราธิวาส	26. น่าน	27. บึงกาฬ	28. บุรีรัมย์
29. ปทุมธานี	30. ประจวบคีรีขันธ์	31. ปราจีนบุรี	32. ปัตตานี
33. พระนครศรีอยุธยา	34. พังงา	35. พัทลุง	36. พิจิตร
37. พิษณุโลก	38. เพชรบุรี	39. เพชรบูรณ์	40.แพร่
41. พะเยา	42. ภูเก็ต	43. มหาสารคาม	44. มุกดาหาร
45. แม่ฮ่องสอน	46. ยะลา	47. ยโสธร	48. ร้อยเอ็ด
49. ระนอง	50. ระยอง	51. ราชบุรี	52. ลพบุรี
53. ลำปาง	54. ลำพูน	55. เลย	56. ศรีสะเกษ
57. สกลนคร	58. สงขลา	59. สตูล	60. สมุทรปราการ
61. สมุทรสงคราม	62. สมุทรสาคร	63. สระแก้ว	64. สระบุรี
65. สิงห์บุรี	66. สุโขทัย	67. สุพรรณบุรี	68. สุราษฎร์ธานี
69. สุรินทร์	70. หนองคาย	71. หนองบัวลำภู	72. อ่างทอง
73. อุตรธานี	74. อุทัยธานี	75. อุตรดิตถ์	76. อุบลราชธานี
77. อำนาจเจริญ			

โดยเราได้แบ่งกลุ่มตัวอักษรตามจำนวนตัวอักษรของแต่ละจังหวัดซึ่งแบ่งออกเป็นกลุ่มต่างๆได้ โดยไม่นับสระอะ (-ะ) ในการนับจำนวนตัวอักษรในแต่ละกลุ่มดังนี้

1.) กลุ่มตัวอักษร 12 ตัวอักษร

1. กรุงเทพมหานคร 2. พระนครศรีอยุธยา 3. นครศรีธรรมราช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.) กลุ่มตัวอักษร 11 ตัวอักษร
1. ประจวบคีรีขันธ์
- 3.) กลุ่มตัวอักษร 10 ตัวอักษร
1. สมุทรปราการ 2. สมุทรสงคราม 3. กำแพงเพชร
- 4.) กลุ่มตัวอักษร 9 ตัวอักษร
1. แม่ฮ่องสอน 2. มหาสารคาม 3. นครราชสีมา 4. สุราษฎร์ธานี
5. หนองบัวลำภู 6. อุบลราชธานี 7. อำนาจเจริญ
- 5.) กลุ่มตัวอักษร 8 ตัวอักษร
1. ฉะเชิงเทรา 2. สมุทรสาคร 3. นครสวรรค์
- 6.) กลุ่มตัวอักษร 7 ตัวอักษร
1. สุพรรณบุรี 2. ปราจีนบุรี 3. เพชรบูรณ์ 4. กาญจนบุรี
5. นครนายก 6. นราธิวาส 7. มุกดาหาร 8. ขอนแก่น
9. หนองคาย 10. เชียงราย 11. เชียงใหม่
- 7.) กลุ่มตัวอักษร 6 ตัวอักษร
1. สระแก้ว 2. อ่างทอง 3. อุดรธานี 4. อุทัยธานี
5. อุดรดิตต์ 6. กาฬสินธุ์ 7. นครปฐม 8. นครพนม
9. ปทุมธานี 10. สกลนคร 11. ร้อยเอ็ด 12. พิษณุโลก
13. เพชรบุรี
- 8.) กลุ่มตัวอักษร 5 ตัวอักษร
1. จันทบุรี 2. ชัยนาท 3. นนทบุรี 4. บึงกาฬ
5. ปัตตานี 6. พะเยา 7. ยโสธร 8. ราชบุรี
9. ลำปาง 10. สงขลา 11. สุรินทร์ 12. สิงห์บุรี
13. สุโขทัย 14. บุรีรัมย์
- 9.) ตัวอักษร 4 ตัวอักษร
1. ระนอง 2. ระยอง 3. ชลบุรี 4. สระบุรี
5. ชัยภูมิ 6. ชุมพร 7. ตราด 8. ลำพูน
9. สพบุรี 10. ภูเก็ต 11.แพร่ 12. พิจิตร
13. พัทลุง 14. พังงา
- 10.) ตัวอักษร 3 ตัวอักษร
1. ตรัง 2. ตาก 3. น่าน 4. เลย
5. สตูล 6. ยะลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นได้ทำการกำหนดตำแหน่งตัวอักษรของแต่ละกลุ่มโดยกำหนดตัวอักษร 2 ตำแหน่ง หรือ 3 ตำแหน่ง จากแผ่นป้ายทะเบียนและนำมาเปรียบเทียบกับฐานข้อมูล เพื่อทำการแสดงผลข้อมูลของภาพ

จากป้ายทะเบียน ดังรูปที่ 5.9



รูปที่ 5.9 ตัวอย่างการกำหนดกลุ่มตัวอักษรที่ต้องการพิจารณา

จากรูปที่ 5.9 เราทำการกำหนดตัวอักษร 2 ตำแหน่ง หลังจากนั้นนำมาเปรียบเทียบกับฐานข้อมูล ถ้าเปรียบเทียบกับฐานข้อมูลแล้วข้อมูลตรงกับตัวอักษร 2 ตำแหน่งนี้ ให้แสดงภาพกรุงเทพมหานคร ออกมา ดังรูปที่ 5.10



รูปที่ 5.10 การแสดงผลการรู้จำแผ่นป้ายทะเบียน

โดยเราได้กำหนดตำแหน่งตัวอักษรที่จะนำมาพิจารณาเปรียบเทียบกับฐานข้อมูลตามกลุ่มตัวอักษรจังหวัดที่ได้แบ่งไว้ก่อนหน้านี้ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.) กลุ่มตัวอักษร 12 ตัวอักษร
พิจารณา 2 ตำแหน่ง ตำแหน่งที่ 1,7
- 2.) กลุ่มตัวอักษร 11 ตัวอักษร
พิจารณา 2 ตำแหน่ง ตำแหน่งที่ 1,3
- 3.) กลุ่มตัวอักษร 10 ตัวอักษร
พิจารณา 2 ตำแหน่ง ตำแหน่งที่ 5,6
- 4.) กลุ่มตัวอักษร 9 ตัวอักษร
พิจารณา 2 ตำแหน่ง ตำแหน่งที่ 3,5
- 5.) กลุ่มตัวอักษร 8 ตัวอักษร
พิจารณา 2 ตำแหน่ง ตำแหน่งที่ 1,3
- 6.) กลุ่มตัวอักษร 7 ตัวอักษร
พิจารณา 2 ตำแหน่ง ตำแหน่งที่ 1,7
- 7.) กลุ่มตัวอักษร 6 ตัวอักษร
พิจารณา 3 ตำแหน่ง ตำแหน่งที่ 1,3,5
- 8.) กลุ่มตัวอักษร 5 ตัวอักษร
พิจารณา 3 ตำแหน่ง ตำแหน่งที่ 1,3,5
- 9.) กลุ่มตัวอักษร 4 ตัวอักษร
พิจารณา 2 ตำแหน่ง ตำแหน่งที่ 1,2
- 10.) กลุ่มตัวอักษร 3 ตัวอักษร
พิจารณา 2 ตำแหน่ง ตำแหน่งที่ 1,3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

วิธีการเมนูของโปรแกรมโดยกราฟฟิคยูเซอร์อินเตอร์เฟส

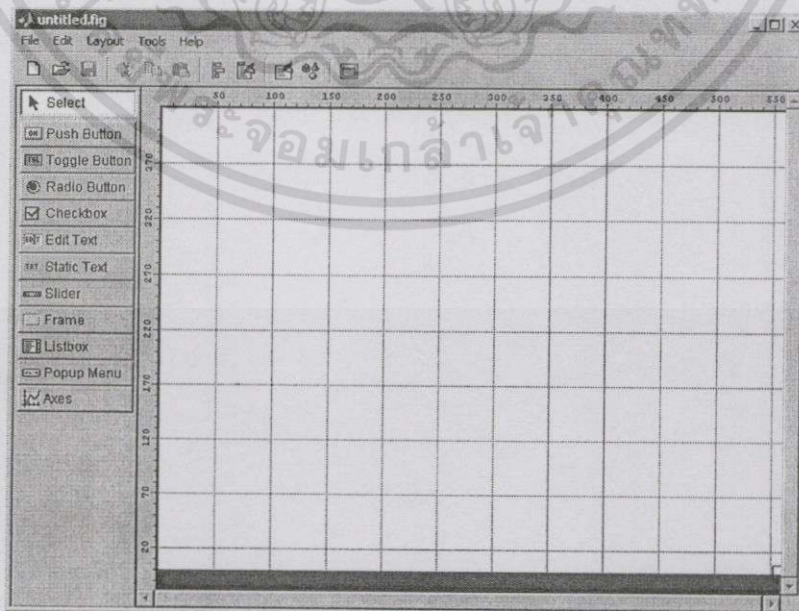
6.1 GUI คืออะไร

GUI คือ ส่วนติดต่อผู้ใช้แบบกราฟิกหรือ (GUI) เป็นจอแสดงหน้าต่างการควบคุม ซึ่งมีฟังก์ชันในการทำงานที่หลากหลาย โดยที่ผู้ใช้ไม่จำเป็นต้องเขียนคำสั่งหรือทำความเข้าใจในว่าในแต่ละฟังก์ชันถูกเขียนขึ้นมาอย่างไร

ส่วนประกอบฟังก์ชันของ GUI ประกอบด้วย

- 1.) Push Button
- 2.) Slider
- 3.) Radio Button
- 4.) Check Box
- 5.) Edit Text
- 6.) Static Text
- 7.) Pop-Up Menu
- 8.) List Box
- 9.) Toggle Button
- 10.) Panel

หน้าต่างการสร้าง GUI ในโปรแกรม MATLAB แสดงดังรูปที่ 6.1

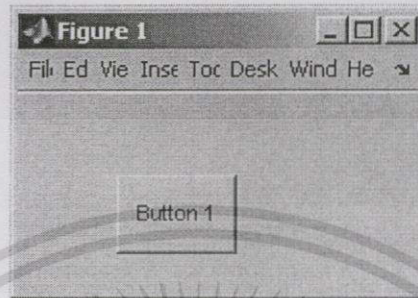


รูปที่ 6.1 หน้าต่าง GUI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Push Button

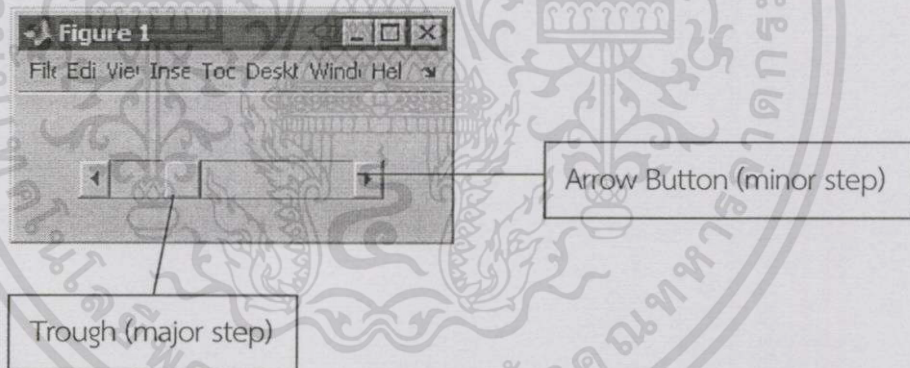
เมื่อเราสร้าง Push Button จะเป็นการสร้างปุ่มกดขึ้นมา เมื่อเรากดลงที่ช่อง button โปรแกรมจะรันตามที่เราเขียนไว้ใน Push Button นั้นๆ



รูปที่ 6.2 หน้าต่าง Push Button

Slider

เป็นหน้าต่างเลื่อนโดยที่แถบเลื่อนตรงกลางสามารถเลื่อนและสามารถกำหนดค่าได้

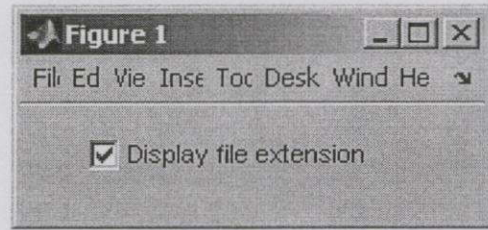
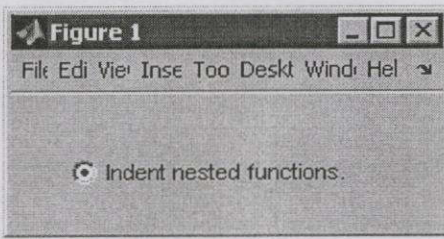


รูปที่ 6.3 หน้าต่าง slider

Radio Button, Check Box

2 ฟังก์ชันนี้จะมีการทำงานที่คล้ายคลึงกัน โดยจะเป็นการสร้างแถบข้อความซึ่งสามารถกดปุ่มเลือกหน้าข้อความได้

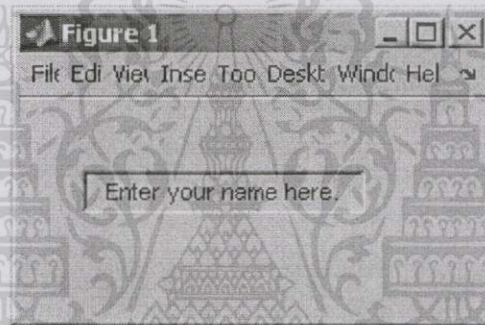
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้



รูปที่ 6.4 หน้าต่าง Radio Button และ Check Box

Edit Text

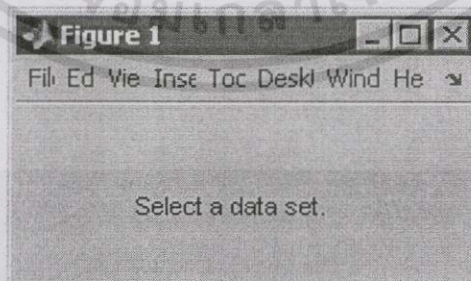
เพื่อสร้างแถบข้อความซึ่งสามารถพิมพ์ข้อความที่เราต้องการได้



รูปที่ 6.5 หน้าต่าง Edit Text

Static Text

จะคล้ายๆ กับ Edit Text พื้นหลังหลังข้อความจะไม่ยุบลงไปเหมือนกับ Edit Text

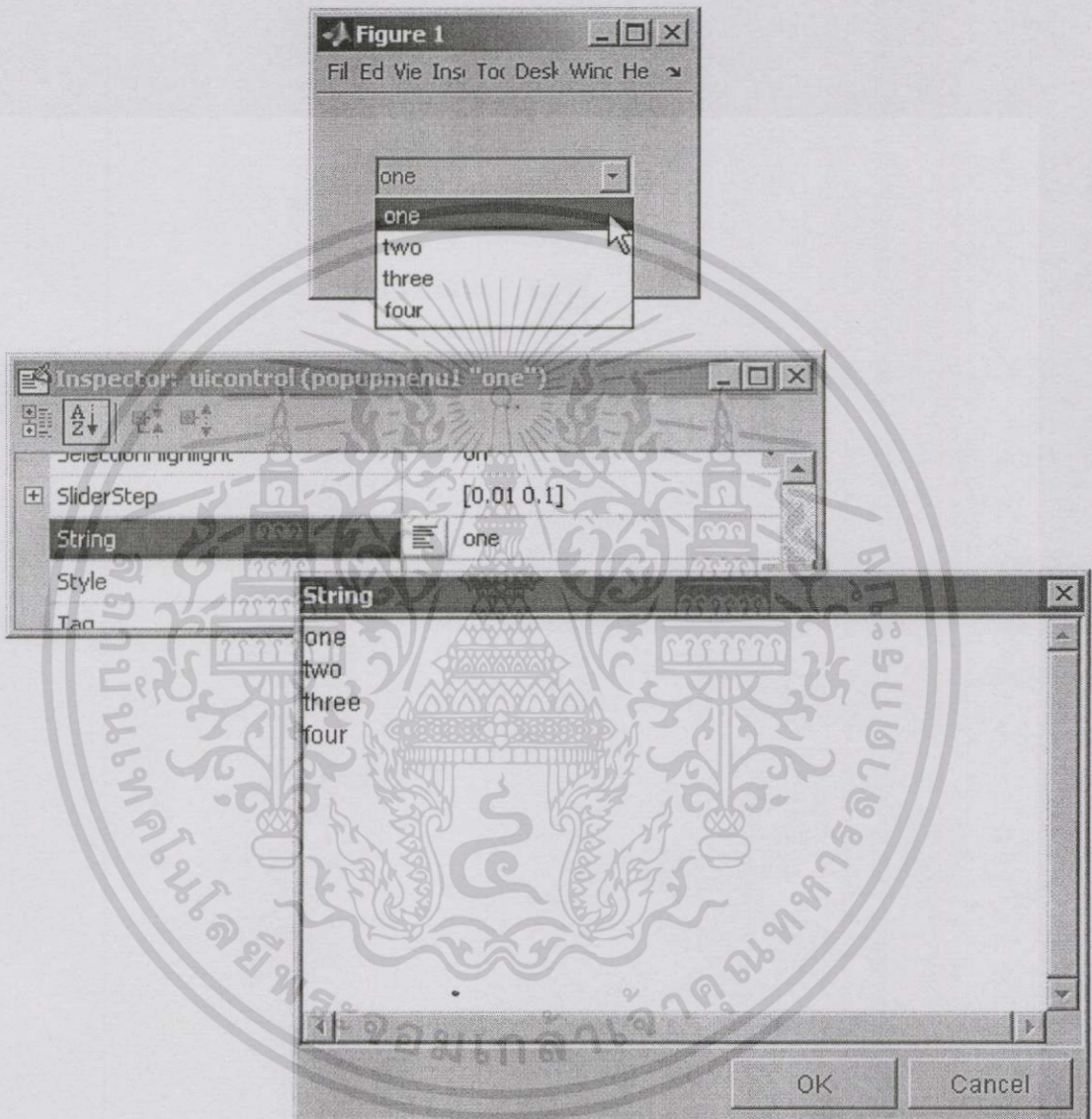


รูปที่ 6.6 หน้าต่าง Static Text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pop-Up Menu

จะเป็นการสร้าง pop-up menu ที่เราสามารถกำหนดข้อความได้มากกว่า 1 ข้อความ และเลือกได้ 1 ข้อความดังรูป

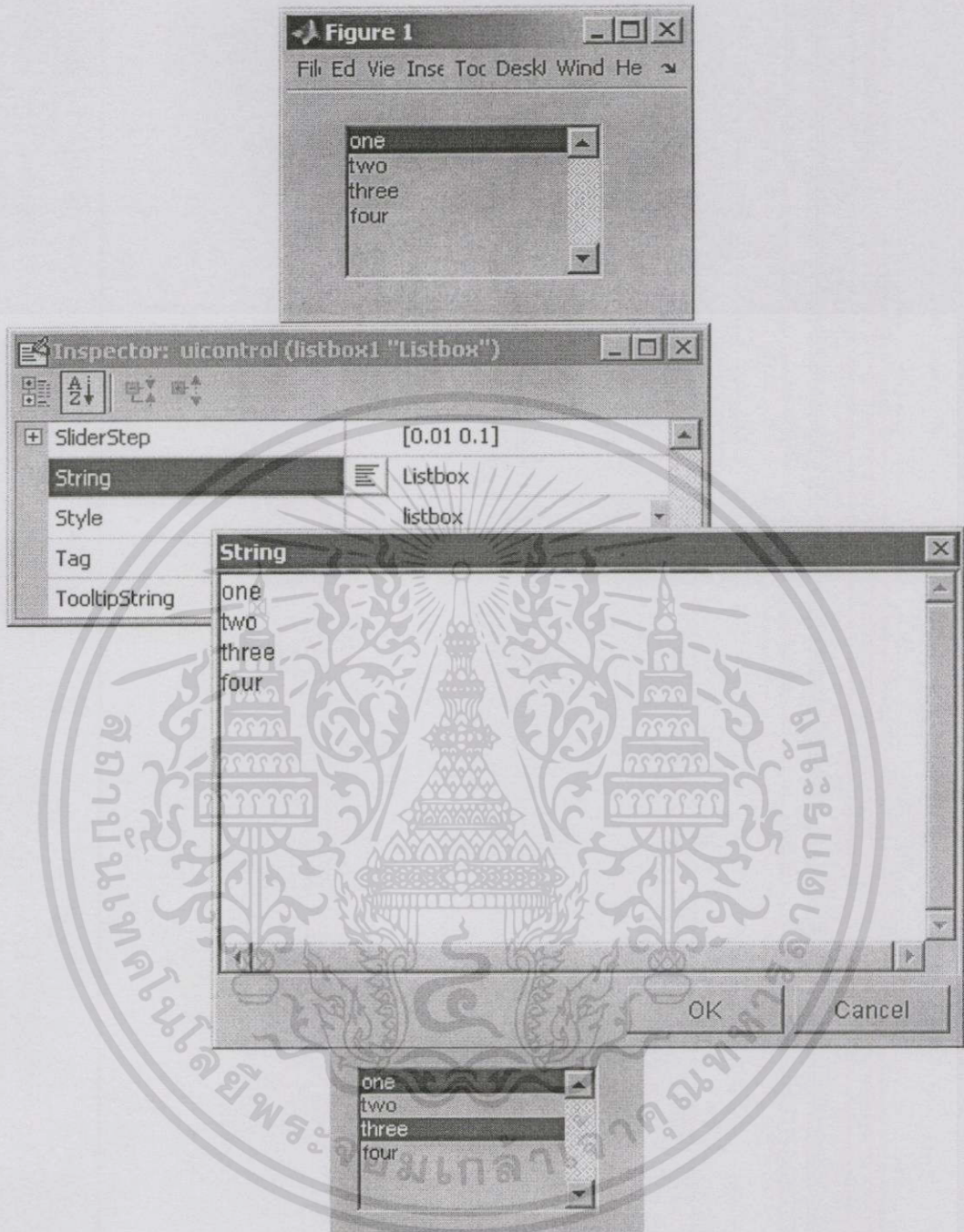


รูปที่ 6.7 หน้าต่าง Pop-Up Menu

List Box

จะเป็นการสร้าง pop-up menu ที่เราสามารถกำหนดข้อความได้มากกว่า 1 ข้อความ และเราสามารถเลือกข้อความได้มากกว่า 1 ข้อความ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

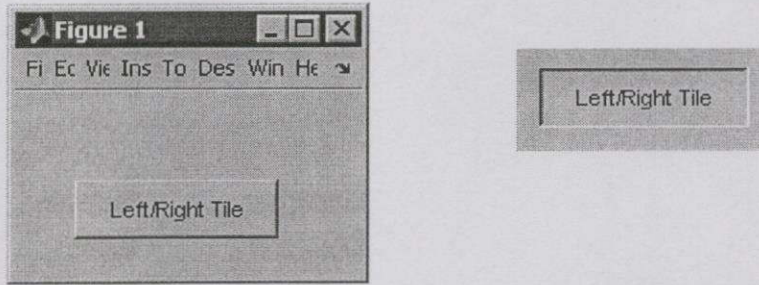


รูปที่ 6.8 หน้าต่าง list Box

Toggle Button

เมื่อเรากดปุ่มนี้ปุ่มจะค้างลงไป เหมือนกับการกดปุ่มค้างไว้ ถ้าหากต้องการให้ค่ากลับมาเหมือนเดิมให้กดอีกครั้งหนึ่ง Toggle Button จึงคืนสู่สภาพเดิม

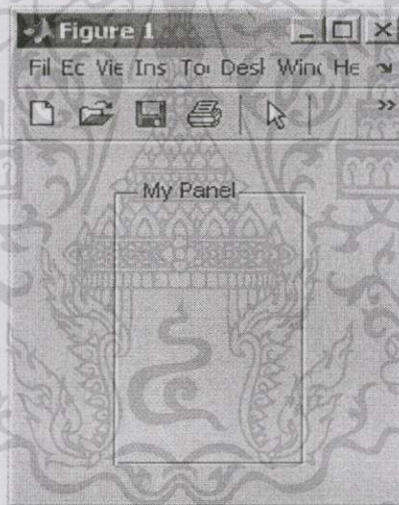
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 หน้าต่าง Toggle Button

Panel

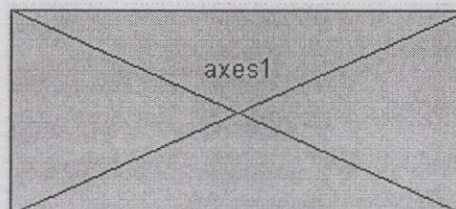
เป็นการสร้าง Panel โดยที่เราสามารถใส่ฟังก์ชันต่างๆ ลงไปในกรอบนี้ได้ เพื่อเป็นการจัดหมวดหมู่ของฟังก์ชัน ให้อยู่ใน Panel



รูปที่ 6.10 หน้าต่าง Panel

Axes

Axes เป็นหน้าต่างการแสดงผล ซึ่งสามารถแสดงผลในรูปแบบกราฟหรือรูปภาพได้

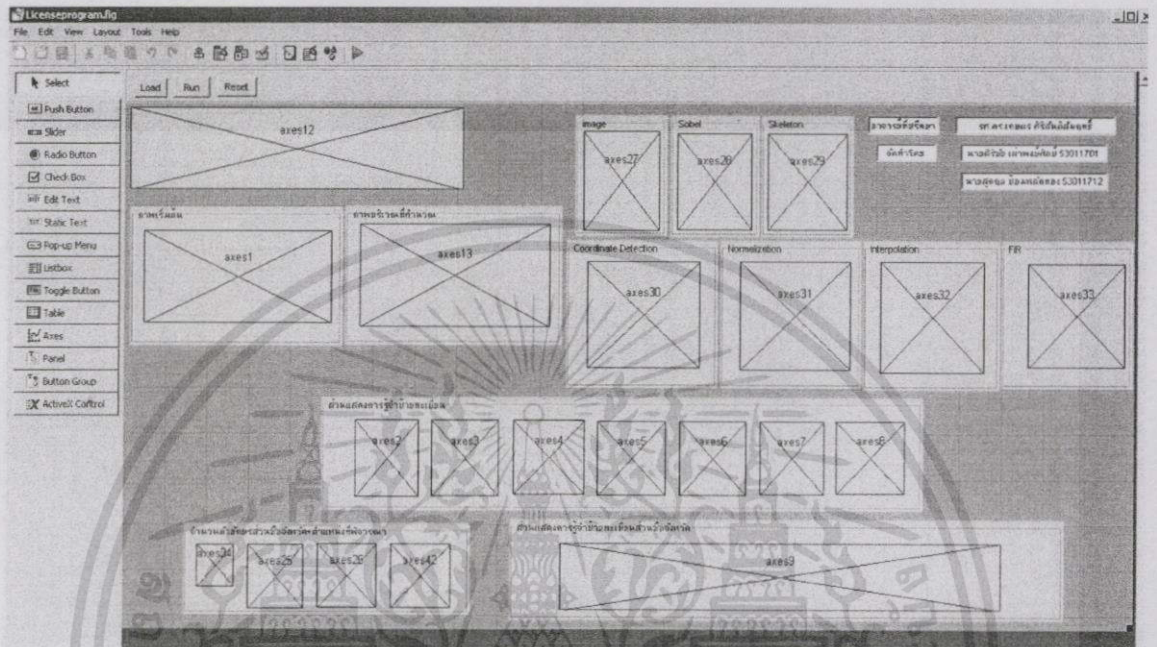


รูปที่ 6.11 หน้าต่าง Axes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การใช้ GUI ในการแสดงผลขั้นตอนและการรู้จำตัวอักษรจากแผ่นป้ายทะเบียน

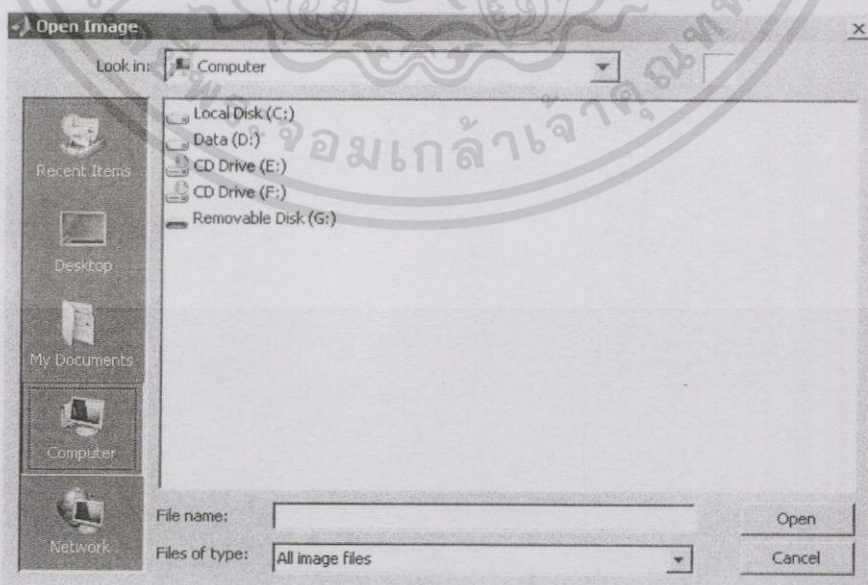
ตัวอย่างการใช้ GUI ในการแสดงผลขั้นตอนและการรู้จำตัวอักษรจากแผ่นป้ายทะเบียนทั้งในส่วนของด้านบนของแผ่นป้ายทะเบียนและชื่อจังหวัด แสดงดังรูปที่ 6.12



รูปที่ 6.12 รูปตัวอย่างการใช้ GUI

คำสั่งเรียกไฟล์ภาพจากคอมพิวเตอร์

เมื่อเรากดปุ่ม load จะมีหน้าต่างขึ้นมาให้เลือกไฟล์ภาพในคอมพิวเตอร์ของเรา โดยใช้คำสั่ง `[path,user_cancel]=imgetfile();` โดยที่ข้อมูลภาพจะเก็บในตัวแปร path



รูปที่ 6.13 หน้าต่างแสดงการโหลดข้อมูลภาพ

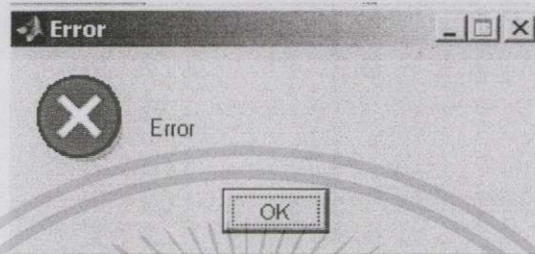
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และถ้าหากไม่มีการเรียกไฟล์ภาพ จะโชว์หน้าต่าง error ขึ้นมาดังภาพที่ 6.14 และใช้คำสั่งดังนี้

```

if user_cancel
msgbox(sprintf('Error'),'Error','Error');
return
end

```



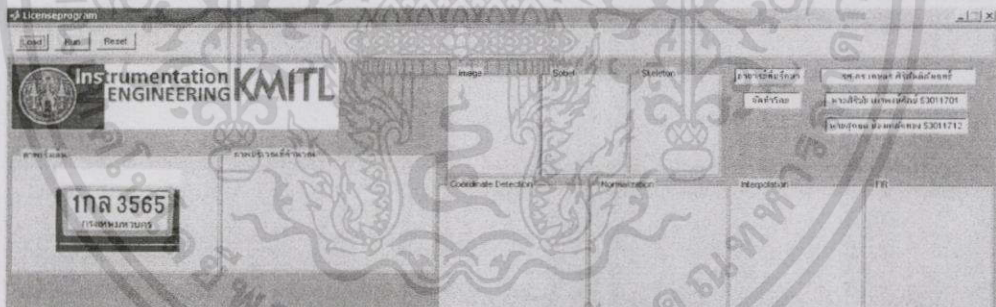
รูปที่ 6.14 หน้าต่างแสดงค่า error หรือ ไม่สามารถโหลด ไฟล์ภาพได้

จากนั้นจะแสดงภาพเริ่มต้นหลังจากที่เรียกไฟล์ภาพมาใน axes1 โดยใช้คำสั่งต่อไปนี้

```

I=imread (path);
axes(handles.axes1);
imshow(I);

```



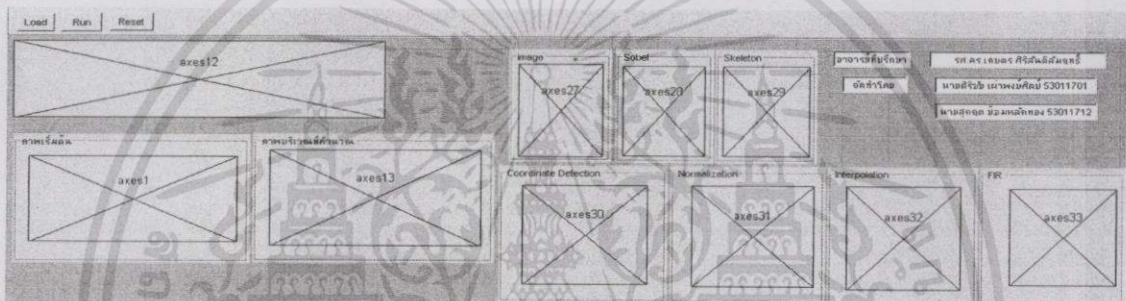
รูปที่ 6.15 หน้าต่างแสดงภาพเริ่มต้นหลังจากเรียกไฟล์ภาพจากคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งเริ่มต้นการรู้จำป้ายทะเบียนรถยนต์

หลังจากเรากดปุ่ม 'Run' โปรแกรมจะแสดงภาพกระบวนการการรู้จำป้ายทะเบียนรถในขั้นตอนต่างๆ ลงในแต่ละ axes ที่เรากำหนดดังนี้

- 1) ภาพบริเวณที่คำนวณ
- 2) ภาพตัวอักษรแต่ละตัวจากแผ่นป้ายทะเบียน
- 3) Edge Detection
- 4) Skeleton
- 5) Coordinate Detection
- 6) Normalization
- 7) Interpolation
- 8) FIR



(ก)



(ข)

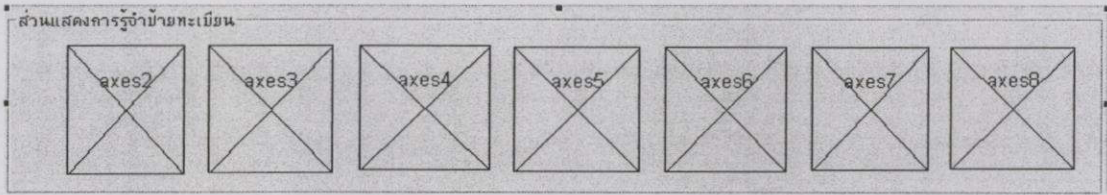
รูปที่ 6.16 ตัวอย่างหน้าต่าง GUI ในส่วนของหมายเลขทะเบียน

(ก) ตัวอย่างหน้าต่างก่อน run โปรแกรม

(ข) ตัวอย่างหน้าต่างเมื่อ run โปรแกรม

หลังจากนั้นโปรแกรมจะเปรียบเทียบตัวอักษรจากแผ่นป้ายทะเบียนกับฐานข้อมูล และแสดงผลการเปรียบเทียบตัวอักษรแต่ละตัวดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

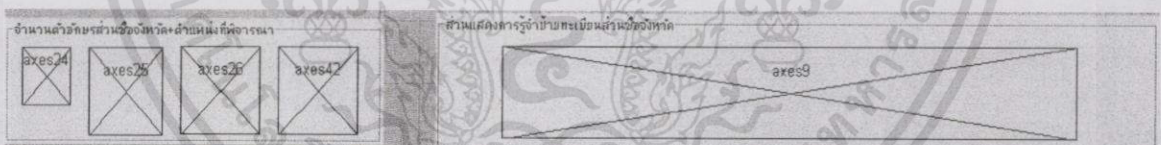
รูปที่ 6.17 ตัวอย่างหน้าต่าง GUI ในส่วนของชื่อจังหวัด

(ก) ตัวอย่างหน้าต่างก่อน run โปรแกรม

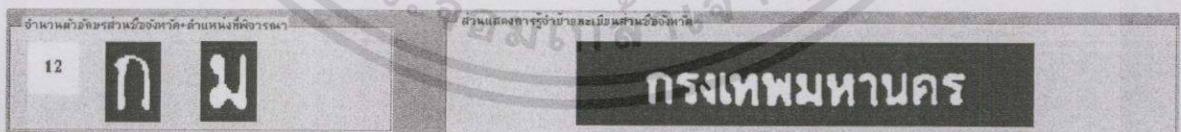
(ข) ตัวอย่างหน้าต่างเมื่อ run โปรแกรม

ส่วนแสดงผลในการรู้จำชื่อจังหวัด

ในส่วนนี้จะแสดงผลการรู้จำในส่วนของชื่อจังหวัด โดยโปรแกรมจะทำการแสดงผลจำนวนตัวอักษรในส่วนชื่อจังหวัดว่ามีกี่ตัว และแสดงตำแหน่งของตัวอักษรที่นำมาพิจารณา หลังจากนั้นนำตัวอักษรที่พิจารณาเปรียบเทียบกับฐานข้อมูล หากตรงตามเงื่อนไขก็จะแสดงผลชื่อจังหวัด ดังรูปที่ 6.17



(ก)



(ข)

รูปที่ 6.18 หน้าต่างการแสดงผลในส่วนชื่อจังหวัด

(ก) ตัวอย่างหน้าต่างก่อน run โปรแกรม

(ข) ตัวอย่างหน้าต่างเมื่อ run โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งรีเซ็ตข้อมูลภาพทั้งหมดของโปรแกรม

โดยเมื่อเรากดปุ่ม 'Reset' โปรแกรมจะทำการเรียกภาพที่มีสีเดียวกับสีพื้นหลังของ Panel มาทับภาพทั้งหมดภายในโปรแกรมโดยใช้คำสั่งต่อไปนี้

```
global im2 < เรียกภาพ im2 โดยที่ภาพ im2 เป็นภาพที่มีสีเดียวกับสีพื้นหลังของ Panel
axes(handles.axes1); < อ้างอิงข้อมูล axes1
```

```
imshow(im2);
```

```
axes(handles.axes2);
```

```
imshow(im2);
```

```
axes(handles.axes3);
```

```
imshow(im2);
```

```
axes(handles.axes4);
```

```
imshow(im2);
```

```
axes(handles.axes5);
```

```
imshow(im2);
```

```
axes(handles.axes6);
```

```
imshow(im2);
```

```
axes(handles.axes7);
```

```
imshow(im2);
```

```
axes(handles.axes8);
```

```
imshow(im2);
```

```
axes(handles.axes9);
```

```
imshow(im2);
```

```
axes(handles.axes27);
```

```
imshow(im2);
```

```
axes(handles.axes28);
```

```
imshow(im2);
```

```
axes(handles.axes29);
```

```
imshow(im2);
```

```
axes(handles.axes30);
```

```
imshow(im2);
```

```
axes(handles.axes31);
```

```
imshow(im2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

axes(handles.axes32);
imshow(im2);
axes(handles.axes33);
imshow(im2);
axes(handles.axes24);
imshow(im2);
axes(handles.axes25);
imshow(im2);
axes(handles.axes26);
imshow(im2);
axes(handles.axes42);
imshow(im2);
axes(handles.axes33);
imshow(im2);
axes(handles.axes13);
imshow(im2);

```



รูปที่ 6.19 หน้าต่าง GUI หลังจากการกดปุ่ม Reset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดลอง

ในส่วนของผลการทดลองจะแบ่งออกเป็น 2 ส่วน คือ ผลการทดลองการแยกแผ่นป้ายทะเบียนออกจากภาพและผลการทดลองการรู้จำตัวอักษรบนแผ่นป้ายทะเบียนโดยเปรียบเทียบกับแม่แบบ โดยจะทำการเปรียบเทียบออกมาเป็นเปอร์เซ็นต์ความผิดพลาด

7.1 ผลการทดลองการแยกแผ่นป้ายทะเบียนออกจากภาพ

ผลการทดลองการแยกแผ่นป้ายทะเบียนออกจากภาพ ในส่วนของขั้นตอนการแยกส่วนแผ่นป้ายทะเบียนออกมานั้นได้กล่าวถึงไปแล้วในบทที่ 3 หลังจากทีรันโปรแกรมแยกส่วนที่เป็นแผ่นป้ายทะเบียนออกมาแล้ว ผลการทดลองซึ่งใช้ภาพถ่ายรถยนต์ที่นำมาทดลองมีทั้งหมด 154 ภาพ สามารถค้นหาแผ่นป้ายทะเบียนได้ทั้งหมด 133 ภาพ คิดเป็น 86.36% ดังแสดงในตารางที่ 7.1

ตารางที่ 7.1 ผลการทดลองในการแยกแผ่นป้ายทะเบียนทั้งหมด 154 คัน

ผลการทดลองการแยกแผ่นป้ายทะเบียนออกจากภาพ		
จำนวนภาพที่ใช้ทำการทดลอง	จำนวนภาพที่โปรแกรมสามารถทำการแยกแผ่นป้ายทะเบียนออกจากภาพได้	เปอร์เซ็นต์การแยกแผ่นป้ายทะเบียนออกจากภาพได้อย่างถูกต้อง
154	133	86.36%

7.2 ผลการทดลองการรู้จำตัวอักษรบนแผ่นป้ายทะเบียนโดยเปรียบเทียบกับแม่แบบ

สำหรับผลการทดลองการรู้จำตัวอักษรบนแผ่นป้ายทะเบียนโดยเปรียบเทียบกับแม่แบบนั้น ได้ทำการทดลองจากภาพแผ่นป้ายทะเบียนรถยนต์ทั้งหมด 154 แผ่นป้าย ที่ผ่านขั้นตอนการแยกแผ่นป้ายทะเบียนออกจากภาพแล้ว อย่างไรก็ตามเนื่องจากตัวอักษรและตัวเลขที่ซ้ำกันบนแผ่นป้ายทะเบียนทั้ง 133 แผ่นป้าย อาจมีจำนวนไม่เท่ากันบางตัวอักษรมีเพียง 1 ตัว เช่น ตัวอักษร “ฟ” และบางตัวอักษรอาจมีซ้ำกันมากถึง 47 ตัว เช่น ตัวอักษร “ก” เป็นต้น ดังนั้นค่าเปอร์เซ็นต์ความถูกต้องที่สามารถรู้จำได้จึงมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าค่อนข้างต่ำจากความเป็นจริง สำหรับผลการรู้จำตัวอักษรและตัวเลขจากแผ่นป้ายทะเบียนทั้งหมด 154 แผ่นป้ายถูกแสดงดังตารางที่ 7.2

ตารางที่ 7.2 แสดงผลการรู้จำตัวอักษรที่อยู่บนแผ่นป้ายทะเบียนทั้ง 154 แผ่นป้าย

ตัวอักษร	จำนวนรูปทั้งหมด	จำนวนรูปที่ออก	จำนวนรูปที่มีปัญหา	เปอร์เซ็นต์ความสำเร็จ	ฝ	-	-	-	-					
					พ	ฟ	ภ	ม	ย	ร	ล	ว	ศ	ช
0	25	24	1	96	พ	2	2	0	100					
1	83	83	0	100	ฟ	1	0	1	0					
2	65	65	0	100	ภ	3	3	0	100					
3	44	43	1	97.72	ม	5	3	2	60					
4	43	43	0	100	ย	5	4	1	80					
5	48	44	4	88	ร	-	-	-	-					
6	50	50	0	100	ล	7	7	0	100					
7	48	39	9	81.25	ว	5	5	0	100					
8	44	43	1	97.72	ศ	11	10	1	90.90					
9	53	53	0	100	ช	8	5	3	62.5					
ก	47	45	2	95.74	ด	11	11	0	100					
ข	2	1	1	50	ท	1	0	1	0					
ค	2	2	0	100	พ	3	0	3	0					
ฅ	22	21	1	95.45	อ	6	6	0	100					
ง	1	1	0	100	ย	2	2	0	100					
จ	2	1	1	50										
ฉ	4	0	4	0										
ช	6	6	0	100										
ฌ	10	8	2	80										
ญ	11	8	3	72.72										
ฎ	16	16	0	100										
ฏ	2	2	0	100										
ฒ	3	3	0	100										
ณ	7	2	5	28.57										
ด	2	2	0	100										
ต	7	2	5	28.57										
ถ	7	7	0	100										
ท	-	-	-	-										
ธ	2	2	0	100										
น	9	7	2	77.78										
บ	2	0	2	100										
ป	-	-	-	-										
ผ	1	1	0	100										

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองการรู้จำตัวอักษรบนแผ่นป้ายทะเบียนสามารถคิดเป็นเปอร์เซ็นต์ค่าเฉลี่ยความถูกต้องในการรู้จำของทุกตัวอักษรรวมกันได้เท่ากับ 92.41% โดยมีจำนวนตัวอักษรและตัวเลขรวมกันทั้งหมด 738 ตัว

สำหรับผลการรู้จำชื่อจังหวัดจากแผ่นป้ายทะเบียนทั้งหมด 144 แผ่นป้ายถูกแสดงดังตารางที่ 7.3

ตารางที่ 7.3 แสดงผลการรู้จำชื่อจังหวัดที่อยู่บนแผ่นป้ายทะเบียนทั้ง 144 แผ่นป้าย

ชื่อจังหวัด	จำนวนรูป	จำนวนรูปที่	จำนวนรูปที่มี	เปอร์เซ็นต์
	ทั้งหมด	ออก	ปัญหา	ความสำเร็จ
กรุงเทพ	109	100	9	91.74
อยุธยา	2	1	1	50
นครราชสีมา	3	3	0	100
ฉะเชิงเทรา	3	3	0	100
ขอนแก่น	3	3	0	100
เชียงใหม่	3	3	0	100
หนองคาย	1	1	0	100
ปราจีนบุรี	1	1	0	100
ร้อยเอ็ด	1	1	0	100
พิษณุโลก	1	1	0	100
ปทุมธานี	1	1	0	100
ศรีสะเกษ	1	1	0	100
สระแก้ว	1	1	0	100
สกลนคร	1	1	0	100
อุดรธานี	1	1	0	100
จันทบุรี	1	1	0	100
นนทบุรี	3	3	0	100
ราชบุรี	1	1	0	100
ยโสธร	1	1	0	100
ชลบุรี	4	3	1	75
เลย	2	2	0	100

จากการทดลองการรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียนสามารถคิดเป็นเปอร์เซ็นต์ค่าเฉลี่ยความถูกต้องในการรู้จำของทุกจังหวัดรวมกันได้เท่ากับ 92.36% โดยมีจำนวนของชื่อจังหวัดรวมกันทั้งหมด 144 ชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพถ่ายรถยนต์ที่ไม่สามารถค้นหาแผ่นป้ายทะเบียนได้ด้วยอย่างจะแสดง ดังรูปที่ 7.2 สาเหตุที่หาแผ่นป้ายทะเบียนไม่ได้เนื่องจากแสงและเงาที่ตกกระทบบนแผ่นป้ายทะเบียน ความสกปรกของแผ่นป้าย ความไม่ชัดเจนของตัวอักษร ป้ายที่มีสัญลักษณ์ประจำจังหวัด และกรอบของแผ่นป้ายทะเบียนที่มีลวดลายต่างๆ จึงเป็นเรื่องยากในการปรับค่าเทรซโฮลที่เหมาะสม ในการค้นหาแผ่นป้ายทะเบียน



รูปที่ 7.2 ภาพถ่ายรถยนต์ที่ไม่สามารถค้นหาแผ่นป้ายทะเบียนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

ปัญญาที่เกิดขึ้นและข้อเสนอแนะ

8.1 ปัญญาที่เกิดขึ้นในระหว่างการทำโครงการ

จากการทำโครงการนั้นเกิดปัญหาในการทดลองเนื่องจากการเก็บข้อมูลภาพ ดังนั้นจึงต้องใช้เวลา, ประสบการณ์ และคำปรึกษาจาก รศ.ดร. เกษตร์ ศิริสันติสัมฤทธิ์ เพื่อให้สามารถทำงานได้ตามที่กำหนด ซึ่งสามารถแยกปัญหาและวิธีแก้ ได้ดังต่อไปนี้

- การถ่ายภาพด้านหน้าของรถยนต์จะเกิดค่าความผิดพลาดสูง เนื่องจากระยะเวลาในการถ่ายภาพแตกต่างกัน เช่น การถ่ายภาพในช่วงเย็นที่แสงหมดจะไม่สามารถนำมาคำนวณได้ วิธีแก้ปัญหาคือ ทำการถ่ายภาพในช่วงก่อนเที่ยงวัน และช่วงบ่ายของวัน เพื่อจะให้เห็นป้ายทะเบียนได้อย่างชัดเจน และเกิดค่าความผิดพลาดน้อยลง

- ในการเก็บภาพแม่แบบตัวอักษรเป็นไปได้ยากเนื่องจาก แผ่นป้ายทะเบียนด้านหน้าของรถยนต์มีค่าความผิดพลาดสูง เนื่องจากตัวอักษรไม่ชัดเจน มีแมลง คราบต่างๆ ติดอยู่เป็นจำนวนมาก วิธีแก้ปัญหาคือ ทำการถ่ายภาพในส่วนด้านหลังของป้ายทะเบียนรถยนต์แทนการถ่ายภาพจากด้านหน้า จะทำให้ได้ตัวอักษรที่ชัดเจน และสมบูรณ์ในการเก็บภาพแม่แบบ

8.2 ข้อเสนอแนะ

- ในการถ่ายภูม่านั้นควรถ่ายภาพให้มีระยะที่เท่ากัน ลักษณะเดียวกัน และไม่ควรมีแสงตกกระทบบนแผ่นป้ายทะเบียน เนื่องจากสิ่งเหล่านี้จะทำให้เกิดค่าความผิดพลาดสูงและยากต่อการคำนวณ

- แผ่นป้ายทะเบียนที่นำมาทดลองนั้นควรเป็นแผ่นป้ายที่สะอาดและเห็นตัวหนังสือชัดเจน

- ในการทดลองนั้นได้ทดลองเฉพาะภาพหน้าตรงเท่านั้น ซึ่งในความเป็นจริงภาพที่ได้รับอาจบิดเบือนหรือเอียงกว่าภาพที่นำมาทดลอง อาจต้องมีโปรแกรมแก้ไขภาพเหล่านี้ก่อน เพื่อให้การรู้จำสมบูรณ์มากยิ่งขึ้น

สุดท้ายนี้ผู้จัดทำหวังเป็นอย่างยิ่งว่า การรู้จำชื่อจังหวัดบนแผ่นป้ายทะเบียนนี้จะสามารถนำไปใช้ในการอำนวยความสะดวกแก่สังคมและเป็นแนวทางการศึกษาและพัฒนาต่อไป

บรรณานุกรม

Bernd Jähne, Digital Image Processing 5th revised and extended edition, 2002.

INTRODUCTION TO DIGITAL IMAGE PROCESSING WITH MATLAB (Alasdair McAndrew)

นางสาวกาญจนา หน่อแก้ว, นางสาวชนิษฐา นาคทอง, นายจตุพร ภูติศชินภัทร. 2552. “การรู้จำตัวอักษรบนพื้นฐานลักษณะเด่น (Feature Based Character Recognition)”. ปริญญาานิพนธ์ สาขาวิศวกรรมการวัดคุมภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

นายกิตตินันท์ ทองคล้าย, นายคมกริช ขอนเตือ, นายคมศักดิ์ ทองปาน. 2555. “การรู้จำตัวอักษรบนแผ่นป้ายทะเบียน(Character Recognition on License Plate)”. ปริญญาานิพนธ์ สาขาวิศวกรรมการวัดคุมภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้