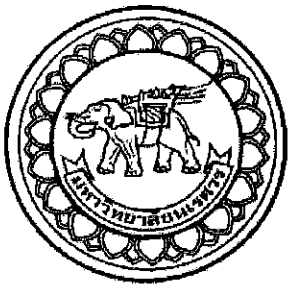
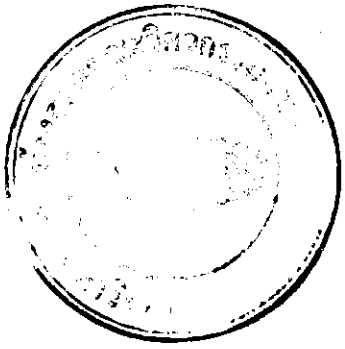




การระบุตัวบุคคลจากการตรวจจับใบหน้า

PERSON IDENTIFICATION BY FACE DETECTION

นางสาวอรุณรัตน์ มานะเพียร รหัส 48364999

นายอุดมศักดิ์ อินทรสูต รหัส 48365019

วิศวกรรมศาสตร์
2 เม.ย. 2553
14997505
ฟร.
ว 413 ก 2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2551

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การระบุตัวบุคคลจากการตรวจจับใบหน้า		
ผู้ดำเนินโครงงาน	นางสาวอรุณรัตน์	มานะเพียร	รหัส 48364999
	นายอุดมศักดิ์	อินทรสูต	รหัส 48365019
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ ริยะมงคล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ
(ดร.พนมขวัญ ริยะมงคล)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มแม่น)

..... กรรมการ
(ดร.ไพศาล มุณีสว่าง)

หัวข้อโครงงาน	การระบุตัวบุคคลจากการตรวจจับใบหน้า		
ผู้ดำเนินโครงงาน	นางสาวอรุณรัตน์	มานะเพียร	รหัส 48364999
	นายอุดมศักดิ์	อินทรสูต	รหัส 48365019
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ	ริยะมงคล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

.....

บทคัดย่อ

โครงงานนี้เป็นโครงงานที่ออกแบบขึ้นเพื่อใช้ในการระบุตัวบุคคลจากการตรวจจับภาพใบหน้า โดยใช้แนวคิดที่ว่าใบหน้าของแต่ละคนเป็นเอกลักษณ์ที่สามารถบ่งบอกความเป็นตัวตนได้ โดยเริ่มจากการตรวจจับหาตำแหน่งของใบหน้าที่ต้องการระบุตัวตน และนำใบหน้าที่ได้มาทำการตรวจสอบหาค่าความเหมือนกับเทมเพลทที่ได้จัดเตรียมไว้ โดยใช้การวิเคราะห์จากค่าสัมประสิทธิ์สหสัมพันธ์ เพื่อนำมาค้นหาข้อมูลตัวบุคคลในภาพจากฐานข้อมูลที่ได้เก็บไว้ ผลที่ได้จากโครงงานนี้ คือ โปรแกรมที่สามารถระบุตัวบุคคล ซึ่งให้ผลค่าความถูกต้องมากกว่า 80%

Project Title	Person Identification by Face Detection		
Name	Miss. Arunrat	Manapian	ID. 48364999
	Mr. Udomsak	Intorasoot	ID. 48365019
Project Advisor	Panomkhawn	Riyamongkol, Ph.D.	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2551		

.....

ABSTRACT

This project has been designed and implemented to identify a person from face detection technology. The core concept lies in the assumption that face is a unique identity of an individual. Our project can identify a person by firstly scanning face position from a still image then examining similarities between the scanned face image and the prepared templates via Cross-Correlation method of calculation and finally identifying a person from the database. This project yields over 80% success rate by correctly identifying persons.

กิตติกรรมประกาศ

โครงงานนี้จะสำเร็จลุล่วงมิได้ หากไม่มี ดร.พนมขวัญ ริยะมงคลที่ได้ให้คำปรึกษาและคำแนะนำในทุกขั้นตอนของการจัดทำโครงงานด้วยดีเสมอมา ผศ.ดร.สุชาติ แย้มเม่น และ ดร.ไพศาล มุณีสว่าง ที่ให้คำแนะนำในการปรับปรุงโครงงานให้ดียิ่งขึ้น รวมถึงเพื่อนๆทุกคนที่ยังไม่ได้เอ่ยนามที่ให้ความสนับสนุน จนทำให้การจัดทำโครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

คณะผู้จัดทำโครงงานจึงขอขอบพระคุณทุกท่านไว้ ณ โอกาสนี้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ซ

สารบัญรูป ..... ซ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบข่ายของโครงงาน 1

1.4 ขั้นตอนการดำเนินงาน 2

1.5 แผนการดำเนินงาน 2

1.6 ผลที่คาดว่าจะได้รับ 3

1.7 งบประมาณที่ใช้ 3

บทที่ 2 หลักการและทฤษฎีที่ใช้

2.1 การพิสูจน์ตัวตนโดยใช้ลักษณะเฉพาะทางชีวภาพของแต่ละบุคคล 4

2.2 ภาพดิจิตอล (Digital Image) 4

2.2.1 พิกเซล (pixel) 5

2.2.2 Digital Image Type 6

2.2.3 รูปร่างของภาพ 6

2.2.4 บิต (Bit) 6

2.3 มาตรฐานของสี 7

2.4 Histogram Processing 12

สารบัญ (ต่อ)

หน้า

2.5 ระบบฐานข้อมูล 12

2.5.1 โครงสร้างข้อมูล 13

2.5.2 รูปแบบของระบบฐานข้อมูล 14

2.5.3 คีย์ในฐานข้อมูลเชิงสัมพันธ์ 14

2.5.4 ความสำคัญของระบบฐานข้อมูล 15

2.5.5 โปรแกรมฐานข้อมูลที่นิยมใช้ 15

2.6 การติดต่อกับฐานข้อมูล ODBC 16

2.7 หลักการค้นหาตำแหน่งของภาพใบหน้า 16

2.7.1 การใช้โครงข่ายประสาทเทียม 17

2.7.2 แบบจำลองวงรีและGenetic Algorithm 17

2.7.3 Genetic Algorithm ในการสร้างแบบจำลองของภาพใบหน้า 17

2.7.4 ใช้แบบจำลองของภาพใบหน้า 17

2.8 วิธีการเปรียบเทียบภาพใบหน้า 17

2.10.1 Principal Component Analysis (PCA) 17

2.10.2 Linear Discriminant Analysis (LDA) 18

2.10.3 Trace Transform 18

2.10.4 หลักการของสหสัมพันธ์ Cross - Correlation 18

2.9 วิธีการทำGraphic User Interface (GUI) บน MATLAB 19

2.9.1 การสร้าง GUI ด้วย GUIDE 19

2.9.2 ส่วนประกอบของGUI ใน MATLAB 19

2.9.3 การสร้าง Application M- file ของ GUIDE 23

2.9.4 การกำหนดค่าคุณสมบัติของส่วนประกอบต่างๆ..... 25

2.9.5 User Interface control 26

2.9.6 Understanding the Application M – file 26

2.9.7 กระบวนการออกแบบ GUI 26

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินโครงงาน

3.1 บล๊อคไดอะแกรม 28

3.2 ขั้นตอนการจัดเก็บ Template 28

3.2.1 เตรียมข้อมูลภาพสำหรับจัดเก็บเพื่อใช้เป็น Template 28

3.2.2 การตรวจจับใบหน้าคนจากภาพนิ่ง 29

3.2.3 เตรียมข้อมูลภาพใบหน้าต้นแบบ (Template) สำหรับการเปรียบเทียบ . 29

3.3 การค้นหาข้อมูลบุคคลในภาพ 32

3.3.1 เตรียมข้อมูลภาพสำหรับการระบุตัวบุคคล 32

3.3.2 การตรวจจับใบหน้าคนจากภาพนิ่ง 33

3.3.3 การเปรียบเทียบระหว่าง Template และภาพใบหน้า 33

3.3.4 การระบุตัวบุคคลด้วยการวิเคราะห์จากค่า Correlation – Coefficient... 34

3.3.5 การติดต่อฐานข้อมูล 36

บทที่ 4 ผลการทดลองและวิเคราะห์ผลการทดลอง

4.1 ผลการทดลอง 37

4.2 วิเคราะห์ผลการทดลอง 48

บทที่ 5 สรุป

5.1 สรุปผลการทดลอง 49

5.2 ปัญหาและแนวทางแก้ไข 49

5.3 แนวทางการพัฒนาในอนาคต 49

เอกสารอ้างอิง 50

ภาคผนวก ก.

การติดตั้งฐานข้อมูล 51

สารบัญ (ต่อ)

หน้า

ภาคผนวก ข.

โปรแกรมควบคุม GUI 53

โปรแกรมการระบุตัวบุคคลจากการตรวจจับใบหน้า 56

โปรแกรมในส่วนของการตรวจจับใบหน้า 57

โปรแกรมในส่วนของการคำนวณค่า Correlation Coefficient 58

ประวัติผู้เขียนโครงงาน 59

สารบัญตาราง

ตารางที่ หน้า

4.1 ผลการทดลองการระบุตัวบุคคลจากการตรวจจับใบหน้า 44

สารบัญรูปภาพ

รูปที่ หน้า

2.1 ตัวอย่างพิกเซลของภาพสีเมื่อขยายรายละเอียด 5

2.2 ระบบสี RGB 8

2.3 ระบบสี HSV 9

2.4 ระบบสี HLS 10

2.5 ระบบสีแบบบวก 11

2.6 ระบบสีแบบลบ 12

2.7 แสดงการสร้าง GUI และ Tool ต่างๆ 21

2.8 กำหนดตัวเลือกต่างๆก่อน โดยภายใต้เมนู Tool Application Option 21

2.9 แสดงหน้าต่าง GUIDE Application Option 22

2.10 แสดงเมนู Context ของ Figure 25

2.11 แสดงหน้าต่างของ Alignment Tool 25

2.12 แสดงขั้นตอนกระบวนการการออกแบบ GUI 26

3.1 แผนภาพแสดงการทำงานของการระบุตัวบุคคลจากการตรวจจับใบหน้าคนจากภาพนิ่ง ... 28

3.2 แสดงผลลัพธ์การตรวจจับใบหน้าคน 29

3.3 ตัวอย่างรูป Template จากการตรวจจับใบหน้าคนจากภาพนิ่ง..... 29

3.4 E/R diagram ของข้อมูลที่จัด 30

3.5 เริ่มสร้างฐานข้อมูลเชื่อ “faceRecognize”..... 30

3.6 การกำหนดคุณสมบัติให้กับตาราง 31

3.7 การจัดเก็บข้อมูลลงฐานข้อมูล 32

3.8 แสดงผลลัพธ์จากการตรวจจับใบหน้าคน 33

3.9 แผนภาพการระบุตัวบุคคลด้วยการวิเคราะห์จากค่า Correlation-Coefficient 35

สารบัญรูปภาพ (ต่อ)

รูปที่ หน้า

4.1 รูปภาพ Template ที่ใช้ในการเปรียบเทียบ 37

4.2 รูปภาพที่นำมาใช้ในการทดสอบ 38

4.3 หน้าต่างโปรแกรมในการระบุตัวบุคคล 40

4.4 แสดงการเลือกไฟล์ภาพที่จะนำมาระบุตัวบุคคล 41

4.5 แสดงผลจากการระบุไฟล์ภาพที่ต้องการระบุตัวบุคคล 42

4.6 แสดงผลที่ได้จากการระบุตัวบุคคล 43

ก-1 ส่วนของการติดตั้ง Data source(ODBC)เพื่อทำการติดตั้ง Database 51

ก-2 ทำการเปลี่ยนชื่อData Source Name 52

ก-3 คลิกเลือกไปยังที่อยู่ของไฟล์ 52

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

เทคโนโลยีทางด้านการประมวลผลลักษณะทางกายภาพของมนุษย์ เช่น การตรวจจับใบหน้า การรู้จำลายนิ้วมือ การรู้จำเสียง เพื่อช่วยในการยืนยันและระบุตัวบุคคล กำลังเป็นที่นิยมและเป็นที่สนใจโดยมีการนำเอามาประยุกต์ใช้ในการดำเนินกิจกรรมต่างๆในชีวิตประจำวัน ตัวอย่างเช่น หน่วยงานทางภาครัฐ ได้นำมาประยุกต์ใช้ในกระบวนการสืบสวนในทางกฎหมาย มีการตรวจวิเคราะห์ใบหน้าจากภาพนิ่ง เพื่อเปรียบเทียบหาตัวผู้กระทำผิดหรือเพื่อเฝ้าระวังป้องกันกลุ่มผู้ก่อการร้ายที่จะลักลอบเข้าประเทศ อีกทั้งยังมีการนำเอาคุณลักษณะทางกายภาพของมนุษย์เหล่านี้ไปบันทึกลงในฐานข้อมูลเพื่อเก็บเป็นประวัติ ส่วนทางด้านของภาคเอกชน ก็มีการนำเอามาประยุกต์ใช้ในด้านการรักษาความปลอดภัยและความลับขององค์กร เพื่อยืนยันตัวบุคคลก่อนเข้าใช้งานข้อมูลที่เป็นความลับของบริษัท นอกจากนี้ในระยะไม่กี่ปีที่ผ่านมา ได้มีการนำไปประยุกต์ใช้งานร่วมกับคอมพิวเตอร์โน้ตบุ๊กสำหรับการล๊อกอินเข้าใช้งานเครื่องคอมพิวเตอร์ ซึ่งใช้วิธีการเปรียบเทียบใบหน้าผ่านทางกล้องแทนการใช้รหัสผ่าน

ด้วยสาเหตุดังกล่าวนี้ ผู้จัดทำจึงเล็งเห็นถึงความสำคัญในการสร้างซอฟแวร์เพื่อใช้ในการระบุตัวบุคคลด้วยการตรวจจับภาพใบหน้า ด้วยเหตุผลที่ว่ามนุษย์แต่ละคนมีใบหน้าเป็นเอกลักษณ์ที่สามารถบ่งบอกความเป็นตัวตนได้ ปลอมแปลงได้ยาก และสามารถถ่ายรูปใบหน้าได้ง่าย โดยซอฟแวร์ดังกล่าวมีลักษณะการทำงานคือ รับข้อมูลเข้าเป็นรูปใบหน้าแบบภาพนิ่ง จากนั้นเข้าสู่กระบวนการตรวจจับภาพใบหน้าเพื่อนำมาวิเคราะห์ ให้สามารถระบุได้ว่าบุคคลในภาพคือบุคคลใด

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษากระบวนการในการตรวจจับใบหน้าจากภาพนิ่ง

1.2.2 เพื่อออกแบบซอฟต์แวร์ที่ใช้ในการระบุตัวบุคคลจากการตรวจจับใบหน้า

1.2.3 เพื่อทำการสร้างซอฟต์แวร์ที่ใช้ในการระบุตัวบุคคลจากการตรวจจับใบหน้า

1.3 ขอบข่ายของโครงงาน

1.3.1 สามารถระบุตัวบุคคลจากการตรวจจับใบหน้า ภายใต้ข้อมูลนำเข้าแบบภาพนิ่ง

1.3.2 สร้างซอฟต์แวร์ที่ใช้ในการระบุตัวบุคคลจากการตรวจจับใบหน้า รวมทั้งจำกัดสิ่งแวดล้อมในการถ่ายรูปใบหน้าดังนี้

- รูปภาพใบหน้าตรง ไม่สวมแว่นตา ไม่มีเส้นผมมาปรกหน้า
- แสงที่ใช้ในการถ่ายรูปอยู่ในระดับเพียงพอที่สามารถมองเห็นใบหน้าชัดเจน
- ไม่มีเงา หรือแสงสะท้อนปรากฏอยู่บนใบหน้า

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและรวบรวมข้อมูลที่เกี่ยวข้องกับหัวข้อโครงงาน

1.4.2 ออกแบบขั้นตอนการทำงานของซอฟแวร์ที่ใช้ในการระบุตัวบุคคลจากการตรวจจับใบหน้า

1.4.3 ทำการสร้างซอฟต์แวร์ที่ใช้ในการระบุตัวบุคคลจากการตรวจจับใบหน้า

1.4.4 ตรวจสอบและแก้ไขข้อผิดพลาดของซอฟต์แวร์ในการระบุตัวบุคคลจากการตรวจจับใบหน้า

1.4.5 จัดทำคู่มือโครงงาน

1.5 แผนการดำเนินงาน

กิจกรรม	ปี 2551						ปี 2552	
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ศึกษาและรวบรวมข้อมูลที่เกี่ยวข้องกับหัวข้อโครงงาน								
2. ออกแบบขั้นตอนการทำงานของซอฟแวร์								
3. ทำการสร้างซอฟต์แวร์								
4. ตรวจสอบและแก้ไขข้อผิดพลาดของซอฟต์แวร์								
5. จัดทำคู่มือโครงงาน								

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ซอฟแวร์ที่ใช้ในการระบุตัวบุคคลจากการตรวจจับใบหน้า ภายใต้ข้อมูลนำเข้าแบบภาพนิ่ง

1.6.2 สามารถนำเอาไปประยุกต์ใช้ในงานด้านการตรวจจับภาพบุคคลต้องสงสัย และใช้เป็นแนวทางสำหรับการค้นคว้าวิจัยเพิ่มเติมเพื่อพัฒนาเทคโนโลยีใหม่ๆในอนาคต

1.6.3 มีความรู้ทางด้าน Image Processing

1.7 งบประมาณของโครงงาน

1.7.1 ค่าพิมพ์(Print) โครงงานและถ่ายเอกสาร	700	บาท
1.7.2 ค่าเข้าเล่มโครงงาน	700	บาท
1.7.3 อุปกรณ์ทางด้านคอมพิวเตอร์	300	บาท
1.7.4 อื่นๆ	300	บาท
รวมเป็นเงิน	2,000	บาท

(สองพันบาทถ้วน)

หมายเหตุ ขออนุมัติถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 การพิสูจน์ตัวตนโดยใช้ลักษณะเฉพาะทางชีวภาพของแต่ละบุคคล [1, 2]

ลักษณะทางชีวภาพของแต่ละบุคคลเป็นลักษณะเฉพาะและลอกเลียนแบบกันไม่ได้ การนำมาใช้ในการพิสูจน์ตัวตนจะเพิ่มความน่าเชื่อถือได้มากขึ้น จึงมีการนำเทคโนโลยีนี้มาช่วยในการพิสูจน์ตัวตน เพื่อเพิ่มความปลอดภัยก่อนเข้าสู่ระบบ เช่นการใช้ควบคู่กับการใช้รหัสผ่าน

Biometric Technology เป็นเทคโนโลยีที่สำหรับยืนยันตัวบุคคลด้วยการเปรียบเทียบ Pattern ของ Physical หรือ พฤติกรรมต่างๆ ของมนุษย์ด้วยคอมพิวเตอร์ เทคโนโลยี Biometric ที่สำคัญได้แก่ การตรวจจับใบหน้า (Face scan), การตรวจสอบลายนิ้วมือ (Finger print scan), การเปรียบเทียบมือ (Hand scan), การสแกนม่านตา (Iris scan), ตรวจสอบลายมือ (Palm scan), ตรวจจับลายเซ็น (Signature scan), สแกนผิว (Skin scan) หรือ ตรวจสอบเสียง (Voice scan) เป็นต้น Biometric จะมาแทนที่เครื่องหมายและรหัสลับซึ่งปกติเป็นสิ่งที่ควบคู่กัน เช่น เลขที่บัตร ATM และรหัสผ่านเป็นต้น ซึ่งไม่สามารถยืนยันความเป็นตัวบุคคลได้ เครื่องหมายเป็นสิ่งที่สามารถปลอมแปลงและลอกเลียนได้ รหัสผ่านก็สามารถลืม ถูกเดา หรือสุ่มได้เช่นกัน ดังนั้น Biometric จะเป็นสะพานเชื่อมโยงความเป็นตัวบุคคลคนนั้นอย่างไม่มีผิดพลาด ไม่ถูกขโมย ไม่สามารถปลอมแปลงหรือสูญหายได้

2.2 ภาพดิจิตอล (Digital Image) [3]

ภาพดิจิตอล เป็นการแสดงผลภาพในลักษณะ 2 มิติ ในหน่วยที่เรียกว่า "พิกเซล" ซึ่งจะสามารถนิยามในรูปของฟังก์ชันสองมิติ f(x,y) โดยที่ x และ y จะแสดงค่าพิกัดของภาพ และแอมพลิจูดของ f ที่พิกัด (x,y) ใดๆ ภายในภาพจะแสดงถึงค่าความเข้มแสงของภาพ (Intensity) ที่ตำแหน่งนั้นๆ และเมื่อค่าของ x,y และแอมพลิจูดของ f เป็นค่าจำกัด (Finite value) จึงเรียกรูปภาพนี้ว่าเป็นภาพดิจิตอล (Digital Image) และถ้ากำหนดให้ภาพ f(x,y) มีขนาด M แถวและ N คอลัมน์ และพิกัดของจุดกำเนิด (Original) ของภาพคือที่ตำแหน่ง (x,y) = (0,0) แล้ว จะสามารถเขียนสมการให้อยู่ในรูปเมทริกซ์ได้ดังนี้

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(0,N-1) \\ \vdots & \vdots & \dots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

ค่าแต่ละค่าที่อยู่ในเมตริกซ์จะเรียกว่า พิกเซล (Pixel) โดยที่ตำแหน่ง (0,0) จะอยู่ทางด้านซ้ายมือสุด ด้านบนของภาพ การจัดลำดับตำแหน่งของจุดภาพจะเรียงจากซ้ายไปขวา ในแต่ละเส้นจุดและจัดลำดับของเส้นจุดจะเรียงจากบนลงล่าง การเก็บค่าของความเข้มแสงของภาพดิจิตอลลงหน่วยความจำในลักษณะเส้นจุด (raster) นี้ จะเรียกว่าภาพบิตแมป (bit-map image) หรือภาพราสเตอร์ (raster image) แต่ภาพที่จัดเก็บในลักษณะนี้จะมีขนาดใหญ่จึงมีการบีบอัดภาพ (image compression) เพื่อให้ข้อมูลภาพมีขนาดเล็กลง

การสร้างภาพดิจิตอลสามารถสร้างได้จากอุปกรณ์รับภาพ เช่น กล้องดิจิตอล (digital cameras) เครื่องกราดภาพ (scanners) เป็นต้น ภาพดิจิตอลยังสามารถสร้างโดยการสังเคราะห์จากสิ่งที่ไม่ใช่ข้อมูลภาพได้อีกด้วย เช่น ฟังก์ชันทางคณิตศาสตร์ หรือ แบบจำลองเรขาคณิตแบบสามมิติ ซึ่งการสร้างภาพลักษณะนี้เป็นส่วนหนึ่งในงานด้านคอมพิวเตอร์กราฟฟิกส์ (computer graphics)

2.2.1 พิกเซล (pixel)

รูปที่ 2.1 ตัวอย่างพิกเซลของภาพสีเมื่อขยายรายละเอียด

จุดภาพ หรือ พิกเซล (pixel) มาจากคำว่า "พิคเจอร์" (picture) ที่แปลว่า รูปภาพ และ "เอเลเมนท์" (element) ที่แปลว่า องค์ประกอบ ซึ่งจะหมายถึง "จุดภาพ 1 จุด" เป็นหน่วยพื้นฐานของภาพ คือ จุดภาพบนจอแสดงผล หรือ จุดภาพในรูปภาพที่รวมกันเป็นภาพขึ้น ซึ่งจะเป็นหน่วยทางตรรกะมากกว่ากายภาค ขนาดของ pixel ขึ้นกับการกำหนดความละเอียด (Resolution) โดยจะเทียบจำนวน pixel กับความยาวต่อนิ้ว ดังนั้นจะมีหน่วยเป็น พิกเซลต่อนิ้ว (ppi : pixel per inch) หรือจุดต่อนิ้ว (dpi : dot per inch) ถ้าภาพมีขนาดเท่ากัน แต่ค่าความละเอียดต่างกัน จำนวนของพิกเซลก็จะต่างกันด้วย ดังนั้นความละเอียด(ความคมชัด)ที่แตกต่างกันไป จึงใช้ในการบอกคุณสมบัติของภาพได้ การกำหนดสีของ pixel ใช้การกำหนดโดยการผสมสีของเสปคตรัม RGB ข้อมูลของสีสามารถคำนวณไบต์ได้ 3 ไบต์ ระบบสีส่วนใหญ่ใช้ 8 บิต ซึ่งจะมีไฟล์สีได้ทั้งหมด 256 สี

2.2.2 Digital Image Type

Digital Image แบ่งออกเป็นหลายประเภท ได้แก่

1) Intensity Image หรือ Monochrome Image หรือ Gray Image

ค่าในแต่ละ pixel ของ gray image คือค่าความเข้มของแสง ณ แต่ละตำแหน่งของ pixel ซึ่งจะอยู่ในรูปของ gray scale (gray level) ค่าที่เป็นไปได้ของ gray scale จะขึ้นอยู่กับจำนวน bit ที่ใช้ ตัวอย่างเช่น 8-bit monochrome จะมี gray scale ทั้งหมด 256 ระดับ

2) Color Image หรือ RGB Image

ค่าในแต่ละ pixel ของ color image จะประกอบไปด้วย vector ที่แสดงค่าของสีแดง สีเขียว และสีน้ำเงิน อย่างละ 8 bit ดังนั้น RGB image 1 pixel จะประกอบไปด้วยจำนวน bit ทั้งหมด 24 bit ทำให้ RGB image มีจำนวนสีที่เป็นไปได้ทั้งหมด 2^{24} สี หรือ 16,777,216 สี

3) Binary Image หรือ Black and White Image

ค่าในแต่ละ pixel ของ binary image จะใช้แค่ 1 bit ซึ่งจะมีค่าที่เป็นไปได้คือ 0 (สีดำ) และ 1 (สีขาว) เท่านั้น

4) Indexed Image

ค่าในแต่ละ pixel ของ indexed image จะประกอบไปด้วยค่า index number ขนาด 8 bit ซึ่งจะชี้ไปยังค่าของสีในตารางสี ดังนั้น ถ้าต้องการที่จะทราบค่าสีในแต่ละ pixel จะต้องไปดูค่าในตารางสีที่ index ตรงกับค่าใน pixel

2.2.3 รูปร่างของภาพ

วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้นมีรูปร่างแตกต่างกันไป ทั้งที่เป็นรูปทรงเรขาคณิตและไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้นจะทำการกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangle image model) เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำ และการแสดงภาพออกทางอุปกรณ์ต่างๆเป็นไปได้อย่างมีประสิทธิภาพ

สำหรับการเก็บข้อมูลภาพลงในหน่วยความจำของคอมพิวเตอร์ต้องทำการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอาร์เรย์ (array) โดยค่าในแต่ละช่องของอาร์เรย์จะแสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอาร์เรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

2.2.4 บิต (Bit)

บิต (Bit : Binary Digit) คือ หน่วยของข้อมูลที่เล็กที่สุดที่เก็บอยู่ในหน่วยความจำภายในคอมพิวเตอร์ ซึ่ง bit จะแทนด้วยตัวเลขหนึ่งตัว คือ 0 หรือ 1 อย่างใดอย่างหนึ่ง เรียกตัวเลข 0 หรือ 1 ว่าเป็น บิต 1 บิต

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพขึ้นอยู่กับจำนวน bit ที่ใช้ เมื่อมีการกำหนดให้ขนาดของ bit ต่อจุดมากขึ้น จะทำให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

1 bit = 2^1 = 2 สี, 2 bit = 2^2 = 4 สี, 4 bit = 2^4 = 16 สี

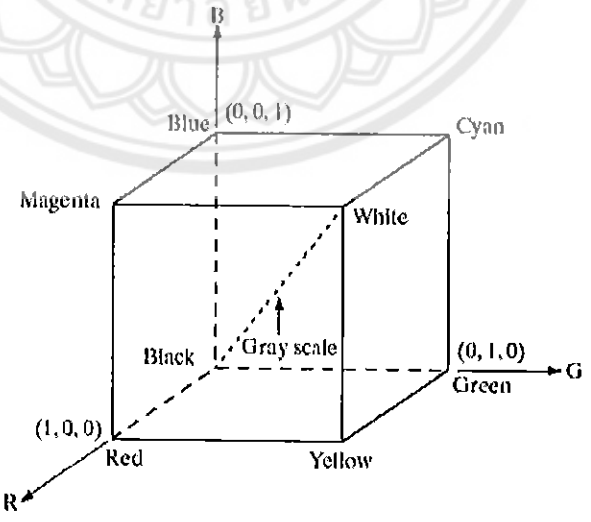
8 bit = 2^8 = 256 สี, 16 bit = 2^{16} = 65536 สี

2.3 มาตรฐานของสี

มาตรฐานของสีในปัจจุบันมีอยู่หลายระบบด้วยกันขึ้นอยู่กับการนำไปใช้ ซึ่งโดยทั่วไปทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปซ 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปซ ซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่น ในระบบ RGB จะมีแกนสีคือ แกนสีแดง สีเขียวและสีน้ำเงิน ในระบบ HLS จะมีแกนเป็นค่าสี (Hue) ความสว่าง (Lightness) และความบริสุทธิ์ของสี (Saturation) ตัวอย่างระบบสีที่นิยมใช้กัน ได้แก่ ระบบ RGB HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation)

1) ระบบสี RGB และ YCrCb

RGB เป็นระบบสีที่แบ่งข้อมูลของสีออกเป็น R (แดง) G (เขียว) B (น้ำเงิน) ซึ่งเป็น 3 สีหลักของแสงสี ส่วน YUV กับ YCrCb จะแบ่งข้อมูลของสีออกในลักษณะเดียวกับกระบวนการรับรู้ของประสาทตาคน นั่นคือ แบ่งเป็นส่วนของความเข้มแสงที่เป็นขาวดำ และส่วนของแสงสีต่างๆ โดย Y จะเก็บข้อมูลของความเข้มแสง เรียกว่า "Luminance" และ U, V กับ Cr, Cb จะเก็บข้อมูลของแสงสี เรียกว่า "Chrominance" ทั้งนี้ YUV และ YCrYCb จะเป็นระบบของสีที่คล้ายๆกัน ซึ่งสามารถแปลงไปเป็น RGB ได้ด้วยสมการเฉพาะแบบ



รูปที่ 2.2 ระบบสี RGB

ระบบสี RGB เป็นระบบสี 24 bit ที่สามารถแสดงสีได้จำนวน 256 สี ที่เห็นบนหน้าจอ ซึ่งระบบสีในหน้าจออิเลคทรอนิกส์โดยทั่วไปจะมีระบบการแสดงผลผ่านหลอดลำแสงที่เรียกว่า CRT (Cathode Ray Tube) โดยที่อาศัยการผสมของแสงสีแดง สีเขียวและสีน้ำเงิน จากการรวมสีของแม่สีหลักนี้จะทำให้เกิดแสงสีขาวซึ่งในลักษณะนี้จะแตกต่างกัน

ในจอคอมพิวเตอร์สีถูกกำเนิดขึ้นประกอบด้วยแสงสีแดง สีเขียวและสีน้ำเงิน ที่เป็นแม่สีหลักทั้งหมดโดยมีลักษณะเป็นจุดเล็กๆในหน้าจอจนไม่สามารถมองเห็นด้วยตาเปล่าได้ ซึ่งดวงตาจะมองเห็นเป็นสีที่ถูกผสมเป็นเนื้อสีเดียวกันแล้ว ถ้าจุดสีแดงและสีเขียวกำลังส่องแสง 100% และสีน้ำเงินไม่แสดงเลยก็จะเห็นเป็นสีเหลืองบริสุทธิ์ แต่ละจุดทั้ง 3 ชุด สามารถปรับค่าใดๆได้จาก 0 ถึง 255 ได้ หมายถึง จำนวนรวมของสีที่เป็นไปได้คือ 256*256*256 = 16,777,216 สี ในจำนวนแรกนั้นเป็นสีแดง (R) จำนวนที่สองคือ สีเขียว (G) และจำนวนสุดท้ายนั้นคือ สีน้ำเงิน (B) หรือที่เรียกว่า "โมเดลแบบ RGB" ดังนั้นถ้าหากจะกำหนดสีใดๆจะต้องระบุจำนวนตัวเลขลงไปด้วย เช่น สีน้ำเงินบริสุทธิ์ คือ 0,0,255 สีน้ำเงินเข้ม คือ 0,0,100 และถ้าหากต้องการสีขาวต้องกำหนดเป็น 255,255,255 ส่วนสีดำกำหนดเป็น 0,0,0 ในการกำหนดด้วยตัวเลขนั้นอาจจะมีความยุ่งยากแต่ก็จะได้สีตามที่ต้องการ สำหรับบนเว็บแล้วไม่จำเป็นต้องกำหนดเป็นตัวเลข แต่จะเขียนแทนด้วยค่าระบบเลขฐานสิบหก

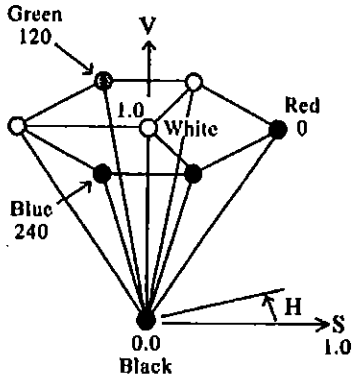
ระบบสีแบบ RGB ของ CIE เป็นระบบสีที่พัฒนาขึ้นโดย CIE (International Commission on Illumination) ซึ่งอ้างสีด้วยสีแดงที่ 700 nm สีเขียวเท่ากับ 546.1 nm และสีน้ำเงิน 435.8 nm

2) ระบบสี HSV

ระบบสี HSV (Hue Saturation Value) ซึ่ง Hue คือ ค่าสีของสีหลัก ในทางปฏิบัติจะอยู่ระหว่าง 0-255 ถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดง และเมื่อเพิ่มค่าขึ้นเรื่อยๆสีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงกลับมาเป็นสีแดงอีกครั้ง ซึ่งแทนค่าให้อยู่ในรูปขององศาได้ คือ สีแดงเท่ากับ 0 องศา สีเขียวเท่ากับ 120 องศาและสีน้ำเงินเท่ากับ 240 องศา

โดยการ Hue สามารถจำนวนได้จากระบบสี RGB ดังนี้

$$\begin{aligned} \text{Red}_h &= \text{red} - \min(\text{red}, \text{green}, \text{blue}) \\ \text{Green}_h &= \text{green} - \min(\text{red}, \text{green}, \text{blue}) \\ \text{Blue}_h &= \text{blue} - \min(\text{red}, \text{greem}, \text{blue}) \end{aligned} \qquad (2.2)$$



รูปที่ 2.3 ระบบสี HSV

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว Hue จะเป็นมุมของสี (ค่าสี) มีค่าเป็นไปตามสีที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้ามีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$\text{Hue} = \frac{(240\text{x}\text{blue}_h)+(120\text{x}\text{green}_h)}{\text{blue}_h+\text{green}_h} \tag{2.3}$$

Saturation คือ ค่าความบริสุทธิ์ของสี ถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วน แต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย Saturation สามารถคำนวณได้ดังนี้

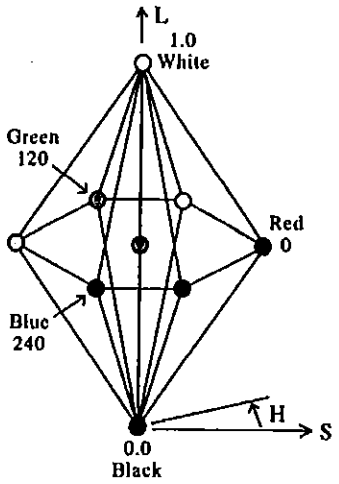
$$\text{Saturation} = \frac{\max(\text{red},\text{green},\text{blue}) - \min(\text{red},\text{green},\text{blue})}{\max(\text{red},\text{green},\text{blue})} \tag{2.4}$$

Value คือ ความสว่างของสี สามารถวัดได้โดยใช้ค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกัน สามารถคำนวณได้จาก

$$\text{Value} = \max(\text{red},\text{green},\text{blue}) \tag{2.5}$$

3) ระบบสีแบบ HLV

ระบบสีแบบ HLS (Hue Lightness Saturation) จะมีลักษณะคล้ายกับ HSV คือ สีของระบบจะขึ้นอยู่กับ Hue Lightness และ Saturation



รูปที่ 2.4 ระบบสี HLS

Hue คือ ค่าของสีหลัก ซึ่งมีสีน้ำเงินอยู่ที่ 0 องศา สีเขียวอยู่ที่ 120 องศา และสีแดงอยู่ที่ 240 องศา

Lightness คือ ความสว่างของสี ซึ่งจะมีค่าเปลี่ยนแปลงตามแนวแกน L โดยที่ L = 0 จะเป็นสีดำ และ L = 1 จะเป็นสีขาว สามารถคำนวณได้ดังนี้

$$\text{Ligthness} = \frac{\max(\text{red},\text{green},\text{blue}) - \min(\text{red},\text{green},\text{blue})}{2} \tag{2.6}$$

Saturation คือ ค่าความบริสุทธิ์ของสี สามารถหาได้ดังนี้

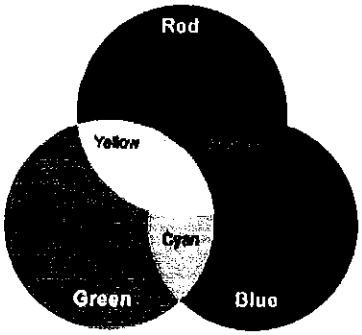
$$\text{Saturation} = \begin{cases} \dfrac{\max(\text{red},\text{green},\text{blue}) + \min(\text{red},\text{green},\text{blue})}{\max(\text{red},\text{green},\text{blue}) - \min(\text{red},\text{green},\text{blue})} & \text{if } L \leq 0.5 \\ \dfrac{\max(\text{red},\text{green},\text{blue}) - \min(\text{red},\text{green},\text{blue})}{2 - \max(\text{red},\text{green},\text{blue}) + \min(\text{red},\text{green},\text{blue})} & \text{otherwise} \end{cases} \tag{2.7}$$

เพื่ออธิบายเกี่ยวกับทฤษฎีสีเพิ่มเติมจึงจำเป็นต้องแนะนำ 2 ระบบสี คือ ระบบสีแบบบวก (additive) และระบบสีแบบลบ (subtractive) มีรายละเอียดดังต่อไปนี้

4) ระบบสีแบบบวก (Additive Color System)

เป็นการผสมสีของแสงที่ประกอบด้วยลำแสงที่มีสีค่าสูงๆ ตามความยาวคลื่นแสงที่ทับซ้อนกันเห็นเป็นแสงสีขาว ถ้าสังเกตให้ดีบนหน้าจอคอมพิวเตอร์ จอโทรทัศน์หรืออุปกรณ์ที่ใช้แสงในการสร้างสีจะเห็นมีจุดเล็กของสีที่ถูกสร้างขึ้นจากแม่สีหลักทางแสงประกอบด้วย 3 สี คือ สีแดง

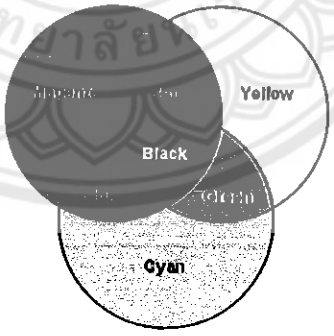
สีเขียวและสีน้ำเงินในสัดส่วนความเข้มที่แตกต่างกันโดยเป็นโมเดลสี RGB และเมื่อคลื่นแสงเหล่านี้มีการทับซ้อนกันแล้วเกิดเป็นสีสันต่างๆนั้นเรียกว่า "สีแบบบวก" ในระบบสีแบบบวกที่เห็นเป็นสีขาวนั้นต่างจากการผสมสีหลักขั้นปฐมภูมิที่เคยรู้จักมาก่อน



รูปที่ 2.5 ระบบสีแบบบวก

5) ระบบสีแบบลบ (Subtractive Color System)

ระบบสีแบบลบมีลักษณะตรงกันข้ามกับระบบสีแบบบวก โดยที่สีแต่ละสีจะได้จากการลบสีต่างๆออกไปจากระบบ ดังนั้นหากไม่มีการแสดงสีใดจะแสดงผลเป็นสีขาว ขณะที่การแสดงสีทุกสีจะเป็นสีดำ ซึ่งสีหลักหรือแม่สีของระบบนี้จะประกอบไปด้วย สีฟ้า (Cyan) สีม่วงแดง (Magenta) และสีเหลือง (Yellow) หรือระบบ CMY เป็นระบบสีที่ใช้กับงานสิ่งพิมพ์ซึ่งมักจะรวมเอาสีดำมาเป็นแม่สีด้วย จึงเรียกว่า "ระบบ CMYK (Cyan, Magenta, Yellow, black)" นั่นเอง



รูปที่ 2.6 ระบบสีแบบลบ

2.4 Histogram Processing [10]

หมายถึงกระบวนการปรับปรุง intensity ของภาพภาพเพื่อให้ได้ histogram ที่มีลักษณะตามต้องการ

Histogram equalization

เป็นการใช้ Histogram ของภาพ กำหนดการเปลี่ยนแปลงระดับของความเข้มแสงให้เป็นแบบ Non-linear โดยทำให้ฟังก์ชันการกระจายความน่าจะเป็น (probability density function : PDF) ของค่าความเข้มแสงของภาพมีการกระจายออก คือทำให้ความสูงของ PDF แบนราบลงโดยให้มีการกระจายตัวแบบภาพแบบ uniform โดยมีสมการการเปลี่ยนแปลงค่าดังสมการที่ 2.8

$$S = T(r) = \int_0^r p_r(w)dw$$

$$S = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n} \quad \text{โดยที่ } k = 0,1,2,\ldots \qquad (2.8)$$

การทำ Histogram equalization จะช่วยเพิ่ม contrast ให้กับภาพโดยเฉพาะบริเวณช่วงที่มีความหนาแน่นสูงใน Histogram ทำให้ Histogram กระจายตัวอย่างสม่ำเสมอ (uniform)

ข้อดี

- ทำงานได้อย่างอัตโนมัติ ไม่ต้องกำหนด Parameter
- Function Transformation ได้จากภาพโดยตรง ปรับ adaptive ตามลักษณะภาพ

ข้อเสีย

- ผลที่ได้อาจมีคุณภาพไม่ดีขึ้นในลักษณะการมองเห็น

2.5 ระบบฐานข้อมูล [4]

ฐานข้อมูล (Database) หมายถึง กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน เพื่อลดความซ้ำซ้อนของข้อมูลและเก็บข้อมูลเหล่านี้ไว้ที่ศูนย์กลาง เพื่อที่จะนำข้อมูลเหล่านี้มาใช้ร่วมกัน

ระบบฐานข้อมูล (Database System) หมายถึง โครงสร้างสารสนเทศที่ประกอบด้วยรายละเอียดของข้อมูลที่เกี่ยวข้องกันที่จะนำมาใช้ในระบบต่างๆร่วมกัน

ระบบฐานข้อมูล จึงนับว่าเป็นการจัดเก็บข้อมูลอย่างเป็นระบบ ซึ่งผู้ใช้สามารถจัดการกับข้อมูลได้ในลักษณะต่างๆทั้งการเพิ่ม การแก้ไข การลบ ตลอดจนการเรียกดูข้อมูล ซึ่งส่วนใหญ่จะเป็นการประยุกต์นำเอาระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการฐานข้อมูล

2.5.1 โครงสร้างข้อมูล

1. บิต (Bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุด

2. ไบท์ (Byte) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำบิตมารวมกันเป็นตัวอักขระ

3. เขตข้อมูล (Field) หมายถึง หน่วยของข้อมูลที่ประกอบขึ้นจากตัวอักขระตั้งแต่หนึ่งตัวขึ้นไปมารวมกันแล้วได้ความหายของสิ่งใดสิ่งหนึ่ง เช่น ชื่อ ที่อยู่ เป็นต้น

4. ระเบียน (Record) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำเอาเขตข้อมูลหลายๆ เขตข้อมูลมารวมกัน เพื่อเกิดเป็นข้อมูลเรื่องใดเรื่องหนึ่ง เช่น ข้อมูลของนักศึกษา 1 ระเบียน (1 คน) จะประกอบด้วย

- รหัสประจำตัวนึกศึกษา 1 เขตข้อมูล
- ชื่อนักศึกษา 1 เขตข้อมูล
- ที่อยู่ 1 เขตข้อมูล

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

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

7. แอททริบิวส์ (Attribute) หมายถึง รายละเอียดข้อมูลที่แสดงลักษณะและคุณสมบัติของเอนทิตี้หนึ่งๆ เช่น เอนทิตี้นักศึกษา ประกอบด้วย

- แอททริบิวต์รหัสนักศึกษา
- แอททริบิวต์ชื่อนักศึกษา
- แอททริบิวต์ที่อยู่นักศึกษา

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

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-One Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตี้หนึ่ง ที่มีความสัมพันธ์กับข้อมูลในอีกเอนทิตี้หนึ่งในลักษณะ (1:1)

- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-Many Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลในเอนทิตี้หนึ่ง ที่มีความสัมพันธ์กับข้อมูลหลายๆข้อมูลในอีกเอนทิตี้หนึ่ง ในลักษณะ (1:m)

- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to- Many Relationship) เป็นการแสดงความสัมพันธ์ของข้อมูลสองเอนทิตี้ในลักษณะกลุ่มต่อกลุ่ม (m:n)

จากคำศัพท์ต่างๆที่เกี่ยวข้องกับระบบฐานข้อมูลที่ได้กล่าวมาแล้วข้างต้น จึงอาจให้ นิยามของฐานข้อมูลในอีกลักษณะได้ว่า "ฐานข้อมูล" อาจหมายถึง โครงสร้างสารสนเทศ ที่ประกอบด้วยหลายๆเอนทิตี้ที่มีความสัมพันธ์กัน

2.5.2 รูปแบบของระบบฐานข้อมูล

รูปแบบของระบบฐานข้อมูล มีอยู่ด้วยกัน 3 ประเภท คือ

1. ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นการเก็บฐานข้อมูลในรูปแบบที่เป็นตาราง หรือเรียกว่า รีเลชั่น (Relation) มีลักษณะเป็น 2 มิติคือเป็นแถว และเป็นคอลัมน์ การเชื่อมโยงข้อมูลระหว่างตาราง จะเชื่อมโยงโดยใช้แอททริบิวต์ (Attribute) หรือคอลัมน์ที่เหมือนกันทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล ฐานข้อมูลเชิงสัมพันธ์นี้จะเป็นรูปแบบของฐานข้อมูลที่นิยมใช้ในปัจจุบัน

2. ฐานข้อมูลแบบเครือข่าย (Network Database) ฐานข้อมูลแบบเครือข่ายจะเป็นการรวมระเบียนต่างๆ และความสัมพันธ์ระหว่างระเบียน แต่จะต่างกับฐานข้อมูลเชิงสัมพันธ์ คือ ในฐานข้อมูลเชิงสัมพันธ์จะแฝงความสัมพันธ์เอาไว้ โดยระเบียนที่มีความสัมพันธ์กันจะต้องมีค่าของข้อมูลในแอททริบิวต์ใดแอททริบิวต์หนึ่งเหมือนกัน

3. ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database) ฐานข้อมูลแบบลำดับชั้น เป็นโครงสร้างที่จัดเก็บข้อมูลในลักษณะความสัมพันธ์แบบพ่อ – ลูก (Parent-Child Relationship Type : PCR Type) หรือเป็นโครงสร้างรูปแบบต้นไม้ (Tree) ข้อมูลที่จัดเก็บในที่นี้คือ ระเบียน (Record) ซึ่งประกอบด้วยค่าของเขตข้อมูล (Field) ของเอนทิตี้หนึ่งๆ ฐานข้อมูลแบบลำดับชั้นนี้คล้ายคลึงกับฐานข้อมูลแบบเครือข่าย แต่ต่างกันที่ฐานข้อมูลแบบลำดับชั้น มีกฎเพิ่มขึ้นมาอีกหนึ่งประการ คือ ในแต่ละกรอบจะมีลูกศรวิ่งเข้าหาได้ไม่เกิน 1 หัวลูกศร

2.5.3 คีย์ในฐานข้อมูลเชิงสัมพันธ์

ในระบบฐานข้อมูลเชิงสัมพันธ์จะต้องกำหนดชนิดของคีย์ เพื่อเป็นแอททริบิวต์พิเศษที่ทำหน้าที่บางอย่างเช่น เป็นตัวแทนของตาราง ซึ่งถูกกำหนดขึ้นมาเพื่อความสะดวกในการอ้างถึงทฤษฎีเกี่ยวกับการทำ Normalization ซึ่งมีดังนี้

- Primary Key (คีย์หลัก) จะเป็นฟิลด์ที่ไม่ซ้ำกันเลยในแต่ละเร็ดคอร์ดในตารางนั้น และสามารถใช้คีย์นี้เป็นตัวแทนของตารางนั้นได้ทันที

- Candidate Key (คีย์คู่แข่ง) เป็นฟิลด์หนึ่งหรือหลายฟิลด์ที่พอเอามารวมกันแล้วคุณสมบัติเป็น Primary Key และไม่ได้ถูกใช้เป็นคีย์หลัก

- Composite Key บางตารางหาฟิลด์ที่ไม่ซ้ำกันไม่ได้เลย จึงต้องใช้หลายๆฟิลด์มารวมกันเป็น Primary Key ซึ่งฟิลด์ที่ใช้รวมกันนี้เรียกว่า Primary Key

- Foreign Key เป็นฟิลด์ใดๆในตารางหนึ่ง (ฝั่ง many) ที่มีความสัมพันธ์กับฟิลด์ที่เป็น Primary Key ในอีกตารางหนึ่ง (ฝั่ง one) โดยที่ทั้งสองตารางมีความสัมพันธ์แบบ One - to - Many

2.5.4 ความสำคัญของระบบฐานข้อมูล

- จัดเก็บและบันทึกข้อมูล (Data Storage)
- ลดความซ้ำซ้อนของข้อมูล (Reduce Data Redundancy)
- สามารถใช้ข้อมูลร่วมกันได้ (Data Concurrency)
- ลดความขัดแย้งหรือแตกต่างกันของข้อมูล (Reduce Data Inconsistency)
- ป้องกันการแก้ไขข้อมูลต่างๆ (Protect Data Editing)
- ความถูกต้องของข้อมูลมีมากขึ้น (Data Accuracy)
- สะดวกในการสืบค้นข้อมูล (Data Retrieval or Query)
- ป้องกันการสูญหายของข้อมูล หรือฐานข้อมูลถูกทำลาย (Data Security)
- เกิดการประยุกต์ใช้ระบบสารสนเทศ (Apply Information System)

2.5.5 โปรแกรมฐานข้อมูลที่นิยมใช้

โปรแกรมฐานข้อมูล เป็นโปรแกรมหรือซอฟแวร์ที่ช่วยจัดการข้อมูลหรือรายการต่าง ๆ ที่อยู่ในฐานข้อมูล ไม่ว่าจะเป็นการจัดเก็บ การเรียกใช้ การปรับปรุงข้อมูล

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

- โปรแกรม Access นับเป็นโปรแกรมที่นิยมใช้กันมากในขณะนี้ โดยเฉพาะในระบบฐานข้อมูลขนาดใหญ่ สามารถสร้างแบบฟอร์มที่ต้องการจะเรียกดูข้อมูลในฐานข้อมูล หลังจากบันทึกข้อมูลในฐานข้อมูลเรียบร้อยแล้ว จะสามารถค้นหาหรือเรียกดูข้อมูลจากเขตข้อมูลใดก็ได้ นอกจากนี้ Access ยังมีระบบรักษาความปลอดภัยของข้อมูล โดยการกำหนดรหัสผ่านเพื่อป้องกันความปลอดภัยของข้อมูลในระบบได้ด้วย

- โปรแกรม FoxPro เป็นโปรแกรมฐานข้อมูลที่มีผู้ใช้งานมากที่สุด เนื่องจากใช้ง่ายทั้งวิธีการเรียกจากเมนูของ FoxPro และประยุกต์โปรแกรมขึ้นใช้งาน โปรแกรมที่เขียนด้วย FoxPro จะสามารถใช้กลับ dBase คำสั่งและฟังก์ชั่นต่าง ๆ ใน dBase จะสามารถใช้งานบน FoxPro ได้ นอกจากนี้ใน FoxPro ยังมีเครื่องมือช่วยในการเขียนโปรแกรม เช่น การสร้างรายงาน

- โปรแกรม dBase เป็นโปรแกรมฐานข้อมูลชนิดหนึ่ง การใช้งานจะคล้ายกับโปรแกรม FoxPro ข้อมูลรายงานที่อยู่ในไฟล์บน dBase จะสามารถส่งไปประมวลผลในโปรแกรม Word Processor ได้ และแม้แต่ Excel ก็สามารถอ่านไฟล์ .DBF ที่สร้างขึ้นโดยโปรแกรม dBase ได้ด้วย

- โปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่งที่มีผู้นิยมใช้กันมาก โดยทั่วไปโปรแกรมฐานข้อมูลของบริษัทต่าง ๆ ที่มีใช้อยู่ในปัจจุบัน เช่น Oracle, DB2 ก็มักจะมีคำสั่ง SQL ที่ต่างจากมาตรฐานไปบ้างเพื่อให้เป็นจุดเด่นของแต่ละโปรแกรมไป

2.6 การติดต่อกับฐานข้อมูล ODBC [5, 6]

ODBC ย่อมาจาก Open Database Connectivity เป็น Application Programming Interface (API) คือ มาตรฐานในการเข้าถึงข้อมูลที่ถูกกำหนดขึ้น ที่อนุญาตให้เชื่อมต่อฐานข้อมูลอื่นๆได้ ซึ่ง ODBC เป็นเสมือนตัวกลางระหว่างโปรแกรมที่ใช้งานกับฐานข้อมูลเหล่านั้น เช่น MS Access

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

ODBC จะใช้ในการติดต่อกับฐานข้อมูลในลักษณะ Client/Server โดยการติดต่อกับฐานข้อมูลใดๆจะไม่ขึ้นอยู่กับชนิดของโปรแกรมระบบฐานข้อมูล ซึ่งถ้ามีการประยุกต์ใช้ ODBC จะทำให้ลักษณะของการเขียนโปรแกรมเหมือนกันทั้งหมด หน้าที่หลักๆของ ODBC คือการติดต่อกับฐานข้อมูล ทำการจัดเตรียมคำสั่ง SQL เพื่อทำการส่งให้ DBMS อีกที แล้วจัดการประมวลผลของ Transaction ทำการรับ-ส่งข้อมูลที่ได้จากฐานข้อมูลกลับไปยังโปรแกรม พร้อมทั้งแจ้งข้อผิดพลาดถ้ารูปแบบคำสั่งนั้นผิดและความสามารถในลักษณะเช่นนี้ถือว่าเป็นคุณลักษณะเด่นที่สำคัญที่สุดของ ODBC

2.7 หลักการค้นหาตำแหน่งของภาพใบหน้า [7]

การค้นหาตำแหน่งของภาพใบหน้าเป็นงานส่วนหนึ่งที่มีความสำคัญต่อการระบุตัวบุคคล ซึ่งสามารถจะนำไปประยุกต์ใช้งานได้หลายอย่างเช่น ระบบรักษาความปลอดภัยโดยการตรวจจับภาพใบหน้าคนร้าย, การทำทะเบียนประวัติคนร้ายของกรมตำรวจโดยใช้ภาพใบหน้า เป็นต้น ในการระบุตัวบุคคลโดยทั่วไปนั้น การค้นหาตำแหน่งของภาพใบหน้าจะเป็นงานที่อยู่ส่วนแรกของ

ระบบ หากการค้นหาตำแหน่งของภาพใบหน้าสามารถทำได้อย่างถูกต้อง ก็จะมีส่วนช่วยให้การระบุตัวบุคคลทำงานได้อย่างถูกต้องมากยิ่งขึ้น ซึ่งวิธีการที่ใช้ในการค้นหาตำแหน่งของภาพใบหน้าได้ถูกพัฒนาออกมามากมาย ในที่นี้จะยกมากล่าวถึงเพียงแค่บางส่วนเท่านั้น

2.7.1 การใช้โครงข่ายประสาทเทียมในการค้นหาตำแหน่งของภาพหน้าตรง

การทำงานจะคล้ายกับระบบรู้จำ คือต้องมีการเลือกภาพมาทำการรู้จำก่อน โดยมีข้อมูลสองชุดที่มีภาพใบหน้ากับไม่มีภาพใบหน้า จากนั้นจึงใช้ข้อมูลที่ได้เรียนรู้เหล่านี้ในการตรวจจับภาพใบหน้าต่อไป ซึ่งจะมีข้อเสียคือ ต้องมีภาพใบหน้าคนจำนวนมากๆเพื่อใช้ในการรู้จำและในการรู้จำจะเวลาในการคำนวณมากขึ้นอยู่กับจำนวนภาพที่นำมารู้จำ

2.7.2 แบบจำลองวงรีและ Genetic Algorithm

ซึ่งวิธีการนี้เป็นการค้นหาตำแหน่งของโครงภาพใบหน้าเท่านั้นยังไม่ใช่ภาพใบหน้าที่แท้จริง และการใช้ Genetic Algorithm จะต้องใช้เวลาในการคำนวณมาก

2.7.3 Genetic Algorithm ในการสร้างแบบจำลองของภาพใบหน้า

ซึ่งวิธีการนี้เป็นการค้นหาภาพที่มีการเปลี่ยนแปลง โดยดึงภาพจากกล้องและนำภาพนั้นมาค้นหาเพื่อกำหนดตำแหน่งของภาพใบหน้า

2.7.4 ใช้แบบจำลองของภาพใบหน้า

โดยแยกคิ้ว ตา จมูก ปาก และทำการค้นหาภาพใบหน้าโดยใช้ Bayesian networks ในการวิเคราะห์ว่าภาพนั้นมีคิ้ว ตา จมูก ปากหรือไม่

2.8 วิธีการเปรียบเทียบภาพใบหน้า [3, 8]

วิธีการที่ใช้ในขั้นตอนการเปรียบเทียบของใบหน้าได้ถูกพัฒนาออกมามากมายในเวลามากกว่าสิบปีที่ผ่านมา มีทั้งแบบที่ยังใช้งานอยู่ในปัจจุบันและบางแบบไม่มีการใช้งานอยู่แล้ว ในที่นี้จะยกมากล่าวถึงเพียงแค่บางส่วนเท่านั้น

2.8.1 Principal Component Analysis (PCA)

PCA ถูกคิดขึ้นในยุคแรกๆของการพัฒนาระบบจดจำใบหน้า ซึ่งอาศัยหลักการทางคณิตศาสตร์เรื่องเวคเตอร์มาใช้ในการทำงานเป็นหลัก โดยจะแปลงภาพถ่ายใบหน้าบุคคลสองมิติไปเป็นเวคเตอร์หนึ่งมิติ และเก็บไว้ในฐานข้อมูล และเมื่อต้องการนำรูปภาพหน้าบุคคลที่สนใจมาเปรียบเทียบก็จะทำการแปลงภาพใบหน้านั้นเป็นเวคเตอร์หนึ่งมิติด้วย แล้วนำเวคเตอร์ไปเปรียบเทียบกับภาพในฐานข้อมูลเพื่อหาผลลัพธ์ ซึ่งต่อมาวิธีนี้ถูกพบว่ามีปัญหาค่อนข้างมากในเรื่องของมิติของเวคเตอร์จึงถูกพัฒนาต่อไปเป็น 2D-PCA (Two-Dimensional Principle Component Analysis) เพื่อแก้ปัญหาเรื่องมิติของเวคเตอร์ โดยไม่ทำการแปลงภาพถ่ายใบหน้าบุคคลสองมิติไปเป็นเวคเตอร์หนึ่งมิติ แต่จะทำการแปลงภาพถ่ายใบหน้าบุคคลสองมิติไปเป็นเวคเตอร์สองมิติด้วย

การทำงานของ PCA จะนำภาพใบหน้าบุคคลในรูปแบบธรรมดามารวมกับรูปที่ทำการหาค่าเฉลี่ยของ pixel แล้ว ได้เป็นรูปใหม่ขึ้นมาแล้วแล้วนำไปแปลงให้เหลืออยู่ในรูปเวคเตอร์หนึ่งมิติ แล้วนำไปใช้เปรียบเทียบ

2.8.2 Linear Discriminant Analysis (LDA)

เป็นวิธีการในยุคแรกๆเหมือนกับ PCA ถูกพัฒนามาควบคู่กันและมีหลักการทำงานค่อนข้างที่จะคล้ายกัน และต่อมาก็ประสบปัญหาในเรื่องเกี่ยวกับมิติของเวคเตอร์เหมือนกับ PCA จึงทำให้ต้องมีการพัฒนาไปเป็นแบบ 2D-LDA (Two-Dimensional Linear Discriminant Analysis) แต่การพัฒนาของ 2D-LDA ไม่ได้ถูกพัฒนามาจาก LDA แต่เป็นการพัฒนาตามอย่างแบบ 2D-PCA เพื่อพยายามจะแก้ไขปัญหาในเรื่องเดียวกัน ทำให้วิธีทั้งสองแบบนี้มี การทำงานค่อนข้างที่จะคล้ายๆกันจึงมีการออกแบบระบบบางระบบที่นำทั้งสองแบบมาใช้ร่วมกันด้วย

2.8.3 Trace Transform

Trace Transform เป็นวิธีการที่ถูกออกแบบมาช่วยทำให้ระบบจดจำใบหน้าทำงานได้ถูกต้องแม้ใบหน้าของบุคคลที่เราสนใจที่จะนำมาเปรียบเทียบจะมีการเปลี่ยนแปลงไปของมุมภาพขนาดภาพ ที่ไม่เท่ากับภาพในฐานข้อมูล โดย Trace Transform คือการทำภาพใบหน้าให้เหลือเป็นรูปร่างแบบ 2D Gray โดยการแปลงด้วยฟังก์ชั่น Trace แล้วทำการวาดเป็นรูปร่างที่ได้จากค่าที่ผ่านการคำนวณด้วยสมการ โดยสมการ Trace จะมีด้วยกันทั้งหมด 22 สมการ โดยภาพใบหน้าเดียวกันถ้าแปลงด้วยสมการ Trace ต่างกันจะให้ผลลัพธ์ภาพ 2D-Gray ออกมาแตกต่างกัน ซึ่งสมการ Trace มีทั้งหมด 22 สมการ นั่นก็หมายถึงว่าสามารถสร้างภาพ 2D-Gray ออกมาจากภาพใบหน้าเดียวกันให้แตกต่างกันได้ทั้งหมด 22 แบบ ซึ่งจากความแตกต่างกันนี้นำไปใช้ประโยชน์ในการรองรับการเปลี่ยนแปลงไปของมุมมอง และขนาดของภาพที่จะนำมาเข้าระบบเพื่อให้ระบบสามารถจดจำใบหน้าของบุคคลนั้นๆได้แม้จะอยู่ในสภาพที่แตกต่างกันไป ไม่ถูกจำกัดว่าจะต้องถ่ายจากมุมตรงๆเท่านั้น

2.8.4 หลักการของสหสัมพันธ์ Cross - Correlation

Cross – Correlation การวิเคราะห์ความสัมพันธ์ของตัวแปร 2 ตัวขึ้นไป โดยจะเป็นการหาค่าความสัมพันธ์ระหว่างข้อมูล 2 ชุด ซึ่งอาจจะมีความสัมพันธ์กันมากหรือน้อยหรือไม่มีความสัมพันธ์กันเลย และอาจจะสัมพันธ์ไปในทิศทางเดียว

ในการศึกษาแต่ละครั้ง ถ้าประกอบด้วยตัวแปรสุ่มเพียง 2 ตัว เช่น X และ Y จะเรียกความสัมพันธ์ระหว่าง 2 ตัวแปรสุ่มว่า สหสัมพันธ์อย่างง่าย (simple correlation) แต่ถ้าหากมีความสัมพันธ์ ระหว่างตัวแปรสุ่มมากกว่า 2 ตัว เรียกว่า สหสัมพันธ์พหุคูณ (multiple correlations) ในกรณีสหสัมพันธ์อย่างง่ายนั้นความสัมพันธ์ระหว่างตัวแปรทั้ง 2 ตัวแปรอาจมีสมการในรูปเส้นตรงหรือมิใช่ก็ได้ โดยถ้ามีสมการอยู่ในรูปของเส้นตรงจะเรียกว่า สหสัมพันธ์เชิงเส้นอย่างง่าย (simple linear correlation)

Cross – Correlation หรือ สหสัมพันธ์ เป็นวิธีการหนึ่งที่สามารถใช้ในการเปรียบเทียบค่าความเหมือนกันของภาพ นอกจากนี้ยังถูกใช้ในกระบวนการของ Image-Matching และ Registration โดยจะทำการเปรียบเทียบโดยคำนวณจากการกระจายตัวของค่า Gray-Level หรือค่าความเข้มของภาพ หลักการของ Cross – Correlation จะนำภาพมาเป็นต้นแบบ (Template) แล้วนำภาพต้นแบบ ซ้อนทับกับภาพที่ต้องการเปรียบเทียบทุกๆจุดที่เป็นไปได้ จากนั้นจึงทำการคำนวณหาค่าของ Correlation Coefficient โดยการคำนวณจะเป็นไปตามสมการ

$$R = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} X(i,j)Y(i,j)}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} X(i,j)\sum_{i=1}^{m}\sum_{j=1}^{n} Y(i,j)}} \qquad (2.9)$$

โดย

R คือ ค่าสัมประสิทธิ์สหสัมพันธ์ระหว่างข้อมูลภาพทั้งสอง

X คือ ค่าเบี่ยงเบนของข้อมูลภาพ template ($X = x - \mu_X$)

Y คือ ค่าเบี่ยงเบนของข้อมูลภาพที่ต้องการเปรียบเทียบ ($Y = y - \mu_Y$)

ตามสมการที่ 2.8 ซึ่งค่าของ R จะมีค่าอยู่ระหว่าง 1 ถึง -1 การพิจารณาความเหมือนของภาพจะพิจารณาจากค่าของ Correlation Coefficient คือ ถ้าภาพที่นำมาเปรียบเทียบ มีส่วนที่เหมือนหรือคล้ายกับภาพ Template ค่าของ Correlation Coefficient จะมีค่าใกล้เคียงกับ 1

2.9 วิธีการทำ Graphic User Interface(GUI) บน MATLAB [11]

2.9.1 การสร้าง GUI ด้วย GUIDE

การสร้าง Graphic User Interface จะใช้การสร้างอยู่บนหน้าต่างรูปภาพ (figure window) ซึ่งภายใต้หน้าต่างนี้จะมีส่วนประกอบต่าง ๆ อยู่ เช่น axes, uicontrol หรือวัตถุอื่น ๆ

การสร้างGUI มีส่วนประกอบดังนี้

1. กำหนดและวางส่วนประกอบต่าง ๆ ลงบน GUI

2. เขียนโปรแกรมเพื่อกำหนดการทำงานของส่วนประกอบต่าง ๆ ใน GUI

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

2.9.2 ส่วนประกอบของ GUI ใน MATLAB

ดังที่ได้กล่าวมาก่อนแล้วว่าสามารถสร้าง GUI ขึ้นมาได้โดยการเขียนเป็น M-file ขึ้นมาล้วน ๆแต่การใช้ GUIDE จะทำให้การทำงานง่ายขึ้นมากเพราะจะช่วยให้สามารถกำหนดตำแหน่ง

ของวัตถุต่าง ๆ ได้โดยง่าย หลังจากนั้น GUIDE จะสร้างไฟล์ขึ้งมา 2 ไฟล์เพื่อเก็บและนำ GUI มาใช้ต่อไปซึ่งจะประกอบด้วย

- FIG-file ซึ่งจะบรรจุรายละเอียดของวัตถุต่างที่เป็นองค์ประกอบอยู่ในหน้าต่างรูปภาพที่เป็น GUI

- M-file ที่จะบรรจุฟังก์ชันที่กำหนดการทำงานของ GUI รวมถึง callback ทั้งหมดซึ่ง callback เหล่านี้จะบรรจุเป็น sub function อยู่ใน M-file และเรียก M-file ที่ควบคุมการทำงานของ GUI นี้ว่า Application M-file

ดังนั้น Application M-file จะไม่มีข้อมูลใด ๆ เกี่ยวกับรูปแบบของส่วนประกอบที่บรรจุอยู่ใน GUI เช่นสี ขนาด ตำแหน่ง หรือ อื่น ๆ เลย เพราะข้อมูลเหล่านั้นจะบรรจุอยู่ใน FIG-file

1. ส่วนประกอบสำคัญของ Application M-file ที่สร้างโดย GUIDE

GUIDE จะรวบรวมองค์ประกอบต่าง ๆ ภายใน GUI แล้วสร้าง Application M-file โดยอัตโนมัติโดยมีรูปแบบของการสร้างที่ชัดเจน เพื่อให้ได้โครงสร้างของ Application M-file จากนั้นสามารถนำโครงสร้างที่สร้างโดยอัตโนมัตินั้นมาปรับแก้ เพื่อให้เกิดการควบคุม GUI ตามที่ต้องการ การกระทำดังกล่าวทำให้ได้ข้อได้เปรียบหลายประการ เช่น

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

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

- เลือก GUIDE Application option แล้วเลือกให้ GUIDE สร้างทั้ง FIG-file และ M-file

Introduction to GUIDE 245

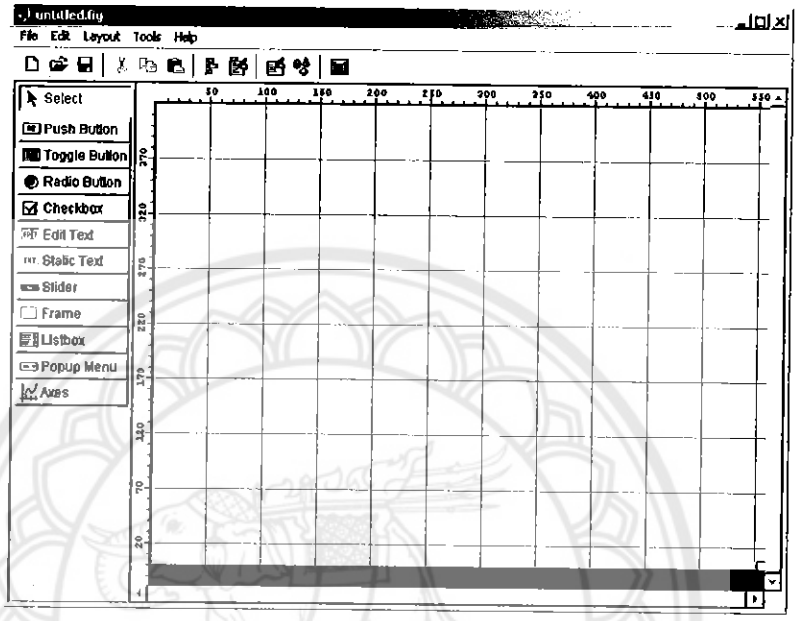
- การใช้ Layout Editor เพื่อวางรูปแบบของ GUI
- เรียนรู้การสร้าง Application M-file จาก GUIDE และเข้าใจถึงวิธีการทำเพื่อจะนำไปใช้ต่อ
- ปรับแก้ Application M-file ให้ทำงานตามที่กำหนด

2. การเลือก GUIDE Application Options

เมื่อต้องการจะใช้ GUIDE นั้น ครั้งแรก บน MATLAB COMMAND WONDOW ที่ prompt สั่ง

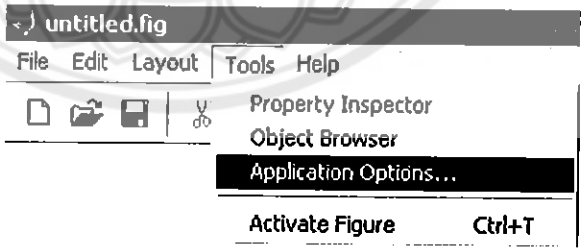
» guide

จากนั้น Layout Editor จะปรากฏขึ้น ซึ่งมีลักษณะดังรูป



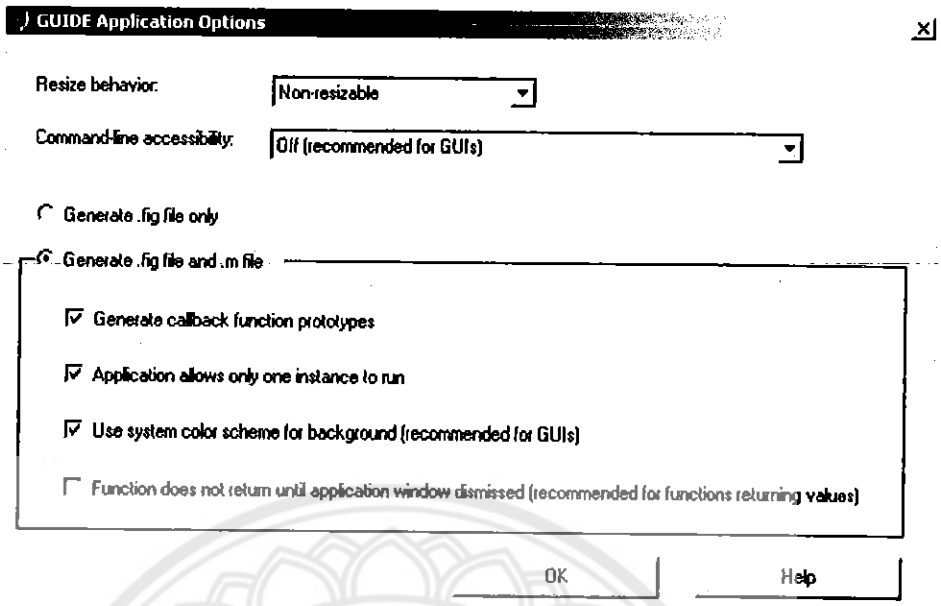
รูปที่ 2.7 แสดงการสร้าง GUI และ Tool ต่างๆ

ก่อนที่จะทำการเพิ่มส่วนประกอบต่าง ๆ ลงใน GUI ควรกำหนดตัวเลือกต่าง ๆ ก่อน โดยภายใต้เมนู Tool เลือก Application Options



รูปที่ 2.8 กำหนดตัวเลือกต่าง ๆ ก่อน โดยภายใต้เมนู Tool เลือก Application Options

ซึ่งจะทำให้ได้ หน้าต่าง GUIDE Application Options ซึ่งมีลักษณะตามรูป



รูปที่ 2.9 แสดงหน้าต่าง GUIDE Application Options

นอกเหนือจากการเลือกให้ GUIDE จะสร้างเฉพาะ FIG-file หรือสร้างทั้ง Fig-file และ M-file แล้วยังสามารถ กำหนดค่าต่าง ๆ ในหน้าต่างตัวเลือกนี้ได้

เป็นการกำหนดว่าผู้ใช้สามารถเปลี่ยนขนาดของหน้าต่าง GUI ที่สร้างขึ้นได้หรือไม่ และถ้าได้ จะให้MATLAB ควบคุมการเปลี่ยนขนาดโดยผู้ใช้อย่างไร ซึ่ง GUIDE ให้ตัวเลือก 3 แบบ คือ

- Non-Resizable ผู้ใช้ไม่สามารถเปลี่ยนขนาดของหน้าต่างได้ (default)

- Proportional ให้ผู้ใช้สามารถปรับขนาดของหน้าต่าง GUI ได้โดย MATLAB จะปรับขนาดขององค์ประกอบต่าง ๆ ใน GUI ให้มีสัดส่วนตามขนาดของหน้าต่าง GUI ที่เปลี่ยนไป

- User-Specifide มีการเขียนโปรแกรมกำหนดให้ GUI ปรับเปลี่ยนขนาดและตำแหน่งของ

องค์ประกอบต่าง ๆ ใน GUI ซึ่งการเลือกตัวเลือกนี้ผู้เขียน GUI ต้องเขียนคำสั่งเพื่อปรับขนาดและตำแหน่งขององค์ประกอบต่าง ๆ ใน GUI ให้ชัดเจน

เมื่อ MATLAB สร้าง graph จะ figure และ axes ที่จะต้องเป็น parents ของรูปกราฟนั้น ซึ่ง MATLAB จะทำการมองหาก่อนว่ามี figure และ Axes เกิดขึ้นอยู่ในขณะนั้นหรือไม่ ถ้ามี MATLAB จะทำการเขียนกราฟลงใน figure และ Axes ที่มีอยู่ถ้าไม่ MATLAB จะทำการสร้าง figure และ axes ขึ้นมาใหม่

ในการสร้าง GUI ของนั้น ส่วนมากแล้ว คงไม่ต้องการให้ผู้ใช้เขียนกราฟลงใน axes ที่ปรากฏอยู่ใน GUI ของ แต่บางกรณีอาจจะต้องการให้ผู้ใช้เขียนกราฟลงใน axes ที่ปรากฏอยู่ในGUI ของก็ได้ ดังนั้น GUIDE จึงมีตัวเลือกให้เลือกสำหรับ Command-Line Accessibility ดังนี้

- Off - ป้องกันการสั่งเขียนกราฟผ่าน command-line บน GUI ของ (default)

- On - ให้มีการเขียนกราฟผ่าน command-line บน GUI ของได้

- User-Specified GUIDE จะให้ GUI ใช้ค่าที่กำหนดโดยคุณสมบัติ Handle Visibility และInteger Handle ของ figure

2.9.3 การสร้าง Application M-file ของ GUIDE

1. การสร้างต้นแบบของ Application M-file

เมื่อเลือกตัวเลือก Generate Callback Function Prototype ในการเลือกตัวของ GUIDE Application Option ก็จะทำให้ GUIDE เพิ่ม sub function ให้กับ application M-file สำหรับทุกวัตถุที่สร้างขึ้นใน GUI (ยกเว้น frame และ Static text) อย่างไรก็ตาม GUEDE จะสร้างเฉพาะ sub function เป็นต้นแบบไว้เท่านั้นส่วนคำสั่งต่าง ๆนั้นต้องเป็นผู้เขียนใน sub function นั้นเอง นอกเหนือจากนั้น GUIDE ยังจะเพิ่ม sub function ทุกครั้งเมื่อแก้ไข callback จากการใช้เมาส์ปุ่มขวาในเมนู context

2. ใช้สีพื้นที่กำหนดด้วย System ที่ MATLAB ทำงานอยู่

สีที่ใช้ในระบบและส่วนประกอบของ GUI จะเปลี่ยนไปตามระบบคอมพิวเตอร์ที่ใช้ตัวเลือกนี้ยอมให้ใช้สีพื้นของ uicontrol เป็นสีเดียวกับสีพื้นของ figure ซึ่งจะทำให้ GUI ของดูมีความกลมกลืนเข้ากับสีพื้น แต่หากต้องการปรับเปลี่ยนสีพื้นของ uicontrol ที่ใช้ให้เป็นไปตามต้องการ ก็ไม่ต้องเลือกตัวเลือกนี้

3. การให้รอ input ของผู้ใช้

สำหรับตัวเลือกใน GUIDE application option

Function does not return until application window dismissed

เป็นการสร้าง application M-file ที่ออกแบบให้รอ input จากผู้ใช้ ซึ่งสามารถทำได้โดยใช้ฟังก์ชัน uiwait ซึ่งจะป้องกันการทำงานต่อไปของ M-file

ในขณะการทำงานนี้ถูกสั่งให้รอ MATLAB จะจัดคิวให้กับคำสั่งต่าง ๆ ที่สั่งเข้าไปไว้ตามลำดับแต่ยังไม่ให้เกิดการทำงานตามคำสั่งนั้น จนกว่าจะเกิดปรากฏการณ์ต่อไปนี้เกิดขึ้น

- รูป GUI ถูกลบ

- ใน GUI มี callback ที่ให้คำสั่ง uiresume

คำสั่งนี้มีประโยชน์ที่จะป้องกัน MATLAB ที่จะใช้คำสั่งจาก command line จนกว่ามีการตอบสนองต่อ dialog box แต่ในขณะเดียวกันยอมให้ callback ทำงานได้

4. การตั้งชื่อไฟล์และ Tag

ในการกำหนดชื่อของไฟล์ หรือวัตถุต่าง ๆ ที่ใช้ใน GUI ซึ่งจะตั้งชื่อโดยคุณสมบัติ Tag สำหรับGUIDE กำหนดค่าคุณสมบัติ Tag (หรือกำหนดชื่อของวัตถุนั้น) ให้กับวัตถุทุกแบบที่สร้างขึ้นโดยอัตโนมัติ เช่น pushbutton1 และให้ string นี้ จะนำไปใช้เป็นชื่อ callback sub function เช่น pushbutton1_callback อย่างไรก็ดี เพื่อให้ชื่อของวัตถุบ่งบอกถึงหน้าที่ของมันมากขึ้น อาจจะตั้งชื่อของวัตถุนั้นให้สื่อถึงหน้าที่ของมันมากขึ้น ดังนั้นแนะนำว่า หลังจากที่สร้างวัตถุนั้นขึ้นมาแล้ว จะต้องตั้งชื่อให้ด้วย การตั้งชื่อก็คือ การกำหนด คุณสมบัติ Tag นั่นเอง และควรจะทำก่อนที่จะ active หรือ save GUI นี้ด้วยการใช้ save as จะทำให้ GUIDE ได้สร้าง application M-file ขึ้นมาใหม่ และปรับค่าคุณสมบัติ callback ให้เหมาะสมกับ callbacks ที่มีอยู่ด้วย

5. การวางส่วนประกอบต่างๆ ลงใน GUI โดย Layout Editor

การใช้ Layout Editor จะช่วยให้สามารถกำหนดส่วนประกอบต่าง ๆ ว่าจะมีอะไร และมีตำแหน่งอยู่ที่ใด การวางตำแหน่งวัตถุต่าง ๆ ซึ่งก็จะมี uicontrol และ axes ก็จะเหมือนการใช้โปรแกรมวาดรูปทั่ว ๆ ไป มีขั้นตอนดังนี้คือ

- เลือก uicontrol หรือ axes ที่ต้องการจะเพิ่มไปใน GUI จาก component palette
- เลื่อนเมาส์เข้ามาในบริเวณพื้นที่ของ GUI ลักษณะ cursor จะเปลี่ยนเป็นรูปกากบาทซึ่งจะสามารถใช้กำหนดตำแหน่ง มุมซ้ายบน ของวัตถุนั้นได้ โดยการกดเมาส์ปุ่มซ้ายที่ตำแหน่งที่ต้องการแล้วลากเมาส์ขณะกดปุ่มเมาส์ด้านซ้ายอยู่ เพื่อกำหนดตำแหน่งด้านขวาล่างของวัตถุ
เมื่อให้ตำแหน่งที่ต้องการให้ปล่อยปุ่มเมาส์
- สามารถปรับปรุงขนาดและเลื่อนตำแหน่งของวัตถุนั้นได้ โดยใช้เมาส์เลือกวัตถุนั้น แล้วเลื่อนหรือปรับขนาดได้ตามต้องการ

6. Activating the Figure

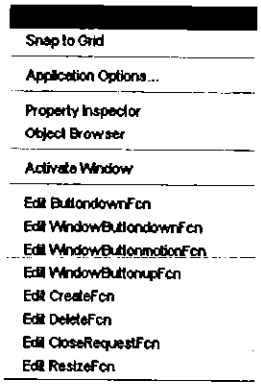
สามารถสร้างการทำงานของ GUI ได้โดยสั่ง activate figure ได้ออกแบบมาแล้วด้วยLayout Editor สามารถ Activate รูปได้โดยเลือก activate figure ภายใต้เมนู Tool หรือโดยการกด Activator button บน Toolbar

เมื่อสั่ง activate figure สิ่งต่อไปนี้จะเกิดขึ้นก่อนอื่น GUIDE จะทำการ SAVE ไฟล์ทั้ง M-file และ Fig-file เป็นอันดับแรก ถ้า file ทั้งสองไม่เคยถูกSAVE มาก่อน ก็จะมี Dialog box SAVE AS เกิดขึ้น เพื่อถามชื่อ FILE ที่ต้องการ SAVE ถ้าใช้ชื่อไฟล์ที่มีอยู่แล้ว MATLAB จะถามว่าต้องการเขียนทับ เขียนเพิ่ม (Append) หรือยกเลิกการใช้ชื่อนั้น

เมื่อทำงานภายใต้ Layout Editor สามารถเลือกวัตถุนั้นด้วยเมาส์ปุ่มซ้าย และเมื่อกดเมาส์ปุ่มขวาเหนือวัตถุนั้น ก็จะปรากฏ context menu ขึ้น นอกเหนือจากที่จะใช้เมนูที่ปรากฏอยู่ด้านบนของหน้าต่าง ซึ่งสามารถใช้ context menu นี้ สร้าง subfunction ให้กับ application M-file ได้สำหรับทุกวัตถุชนิดที่มี callback routine

1497505

ในรูปข้างล่างนี้แสดง เมนู Context ของ figure

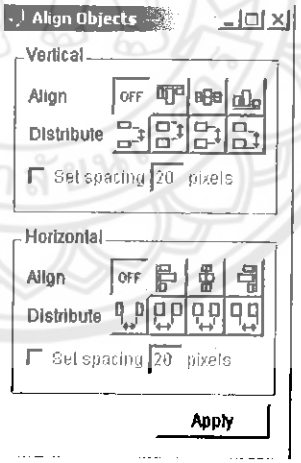


ฝร.
ด 4138
2551

รูปที่ 2.10 แสดงเมนู Context ของ figure

7. Aligning Component in The Layout Editor

ในการจัดเรียงส่วนประกอบต่างๆ ที่มีอยู่ แม้ว่าสามารถที่จะใช้เมาส์เลื่อนวัตถุต่างๆ ได้อยู่แล้วแต่การจัดเรียงส่วนประกอต่างๆ ให้วางอยู่ในแนวเดียวกัน มีระยะห่างเท่าๆ กันนั้นจะมีความสะดวกขึ้นหากใช้ Alignment Tool สามารถเลือก Alignment Tool ได้โดยเลือก Alignment Tool จากปุ่มบนเมนูซึ่ง Alignment Tool จะมีลักษณะดังนี้



รูปที่ 2.11 แสดงหน้าต่างของ Alignment Tool

2.9.4 การกำหนดค่าคุณสมบัติของส่วนประกอบต่างๆ

สามารถที่จะกำหนดค่าคุณสมบัติของส่วนประกอบต่างๆใน GUI ได้ด้วยการใช้ PropertyInspector ซึ่งจะให้รายการคุณสมบัติทั้งหมดของวัตถุที่เลือกและแสดงค่าปัจจุบันของ

คุณสมบัติเหล่านั้น สำหรับคุณสมบัติแต่และตัวในรายการนั้น จะมีอุปกรณ์ที่ใช้ในการแก้ไขคุณสมบัติแต่ละตัวไว้ด้วย คุณสมบัติบางตัวซึ่งมีตัวเลือกอุปกรณ์แก้ไขก็จะแสดงตัวเลือกไว้ให้ ส่วนคุณสมบัติบางตัวต้องเป็นการกำหนดค่า ก็จะเป็นการกำหนดค่าลงไป

2.9.5 User Interface Controls

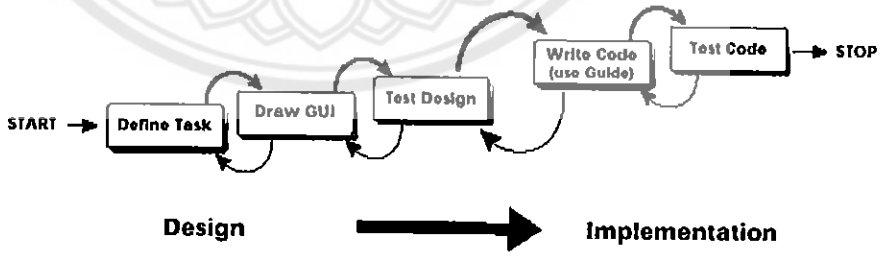
สำหรับ User interface control นั้นจะประกอบด้วย 1) Check Boxes, 2) Editable Text, 3) Frames 4),List boxes, 5) Push Buttons, 6) Radio Buttons, 7) Sliders, 8) Static Text, 9) Toggle Buttons

2.9.6 Understanding the Application M-File

Application M-file เป็นโครงร่างโปรแกรมที่ใช้ควบคุมการทำงานของ GUI ซึ่งจะถูกสร้างขึ้นโดยอัตโนมัติพร้อมกับ Fig-file เมื่อใช้ GUIDE ในการสร้าง GUI โดยที่ Application M-file จะช่วยให้ มีความสะดวก และรวดเร็วมากขึ้น ซึ่งโปรแกรมหรือ Code ทุกส่วนรวมถึง Subfunction จะรวมอยู่ใน Application M-file โดย Callback ทั้งหมดจะถูกเขียนเป็น Subfunction ซึ่งทำให้วิธีการเขียน Callback ง่ายขึ้นและทำให้สามารถปรับค่าเริ่มต้น หรือทำการ Initialize ให้กับ GUI ได้ หัวข้อที่จะกล่าวต่อไปนี้จะเป็นประโยชน์กับผู้ที่จะทำการเขียน M-file ที่ใช้ควบคุม GUI ไม่ว่าจะสร้างขึ้นเอง หรือจะเป็น application M-file ที่สร้างขึ้นโดย GUIDE

2.9.7 กระบวนการออกแบบ GUI

การออกแบบ GUI ที่ดีจำเป็นจะต้องมีการวางแผนการที่ดีหรือกล่าวอีกอย่างหนึ่งกระบวนการการออกแบบ GUI จะมีความสำคัญต่อ GUI ที่ได้เป็นอย่างมาก และอย่าลืมว่าอย่าเริ่มเขียน GUI จนกว่าจะออกแบบมันเสร็จแล้ว ขั้นตอนการออกแบบ GUI เป็นขั้นตอนแสดงดังในรูปต่อไปนี้



รูปที่ 2.12 แสดงขั้นตอนกระบวนการการออกแบบ GUI

- เริ่มจากการวางโครงร่างด้วยแนวความคิดก่อนว่าจะทำอะไร

ขั้นแรกอาจเป็นขั้นที่ยากที่สุดของการออกแบบก็ว่าได้ เพราะเป็นขั้นที่จะต้องคิดว่าจะออกแบบอะไร และทำงานเพื่อให้สำเร็จจุดประสงค์ใด ในขั้นตอนนี้จะต้องวางโครงร่างหลักการการทำงานและสิ่งต่างๆที่เกี่ยวข้องเพื่อจะประกอบเป็น GUI ของ

- ลองเขียน GUI เหล่านี้บนกระดาษ

การเขียนแบบลงบนกระดาษ เป็นขั้นตอนที่สำคัญที่สุดที่จำเป็นในการออกแบบทุกชนิด เพราะสามารถที่จะปรับปรุงและแก้ไขปัญหาที่อาจเกิดขึ้นได้ในขั้นตอนแรก นอกเหนือจากนั้นทำให้เห็นภาพคร่าวๆ ว่าเมื่องานเสร็จสิ้นแล้ว GUI ของจะมีรูปร่างหน้าตาออกมาเป็นอย่างไร และขั้นนี้สามารถที่จะเริ่มถามความเห็นของผู้อื่นว่า มีความคิดเห็นเกี่ยวกับ GUI ของอย่างไรบ้าง ซึ่งสามารถทำให้ปรับแก้สิ่งต่างๆ ได้อย่างง่ายดายในขั้นนี้

นอกจากรูปร่างที่ปรากฏภายนอกแล้วขั้นนี้สามารถที่จะเขียนหน้าที่การทำงานของอุปกรณ์ควบคุมแต่ละส่วนได้ หรือก็คือการเขียน callback ของส่วนควบคุมแต่ละส่วนนั่นเอง ดังนั้นในขั้นตอนนี้270 Introduction to GUIDEจะทราบว่าส่วนควบคุมที่มีอยู่นี้มีอยู่อย่างเพียงพอหรือมากเกินไปหรือไม่ อีกทั้งหน้าที่ของแต่ละส่วนนั้นชัดเจนหรือยังนั่นเอง

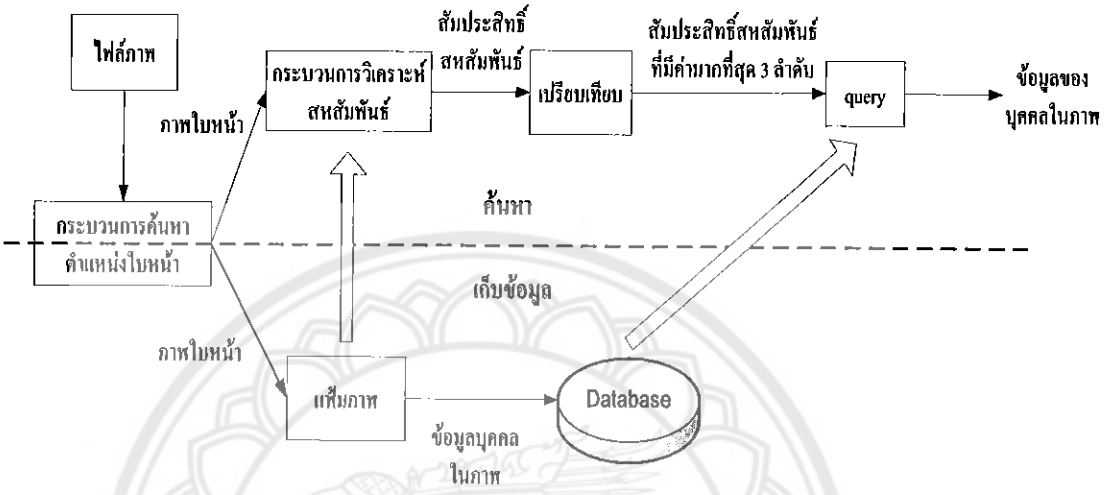
- การสร้างและทดสอบ

ขั้นต่อไปก็จะเป็นการสร้าง GUI และทดสอบ GUI ที่สร้างขึ้น ในส่วนนี้อาจจะเป็นเรื่องยากในMATLAB รุ่นก่อนๆ แต่หลังจากที่มี GUIDE การทำเช่นนี้เป็นเรื่องที่ไม่ยากเลย เพราะการวางส่วนควบคุมต่างๆ เป็นเพียงการวางรูปลงไปบน figure windows เท่านั้น ส่วนการเขียน callback ก็จะมี Callback Editorเป็นอุปกรณ์ที่ช่วยในการเขียนคำสั่งเหล่านี้อยู่แล้ว

หลังจากที่ได้เขียน GUI รวมถึงได้เขียนโปรแกรม callback และอื่นๆที่เกี่ยวข้องเรียบร้อยแล้วจะต้องมีการทดสอบการทำงาน โดยการทดสอบอาจจะทำเอง หรืออาจจะให้เพื่อนๆช่วยใช้โปรแกรมเหล่านี้เพื่อจะได้ช่วยมองหาข้อผิดพลาดของโปรแกรม ซึ่งมักจะเกิดขึ้นเสมอ การให้ผู้อื่นช่วยทดสอบโปรแกรมจะทำให้สามารถมองหาจุดบกพร่องหรือข้อจำกัดของโปรแกรมได้ง่ายขึ้นเพราะผู้สร้างมักจะมีความเคยชินกับสิ่งที่ตัวเองสร้างขึ้นมาและมองข้ามรายละเอียดบางอย่างไปได้

บทที่ 3
วิธีการดำเนินโครงงาน

3.1 บล๊อคไดอะแกรม



รูปที่ 3.1 แผนภาพแสดงการทำงานของการระบุตัวบุคคลจากการตรวจจับใบหน้าคนจากภาพนิ่ง

3.2 ขั้นตอนการจัดเก็บ Template

3.2.1 เตรียมข้อมูลภาพสำหรับจัดเก็บเพื่อใช้เป็น Template สำหรับการเปรียบเทียบ

รูปภาพที่นำมาใช้ในการตรวจจับใบหน้าจะต้องมีคุณสมบัติดังนี้

- รูปที่ใช้เป็นรูปภาพสี
- เป็นรูปภาพใบหน้าตรง ไม่สวมแว่นตา ไม่มีเส้นผมมาปรกหน้า
- แสงที่ใช้ในการถ่ายรูปอยู่ในระดับเพียงพอที่สามารถมองเห็นใบหน้าชัดเจน
- ไม่มีเงา หรือแสงสะท้อนปรากฏอยู่บนใบหน้า

รูปที่จะนำมาเป็น Template ต้องเป็นรูปเดี่ยว แล้วนำรูปไปจัดเก็บไว้ในโฟลเดอร์ ซึ่งการนำข้อมูลรูปมาใช้ในการระบุตัวบุคคลจากการตรวจจับใบหน้าคนจากภาพนิ่งนั้นจำเป็นต้องนำไฟล์รูปภาพใบหน้ามาแปลงเป็นเมตริกซ์ เพื่อให้โปรแกรมที่ใช้ในการประมวลผลสามารถอ่านค่าได้ โดยใช้โปรแกรม MATLAB ในฟังก์ชัน IMREAD ซึ่งผลที่ได้คือค่าเมตริกซ์ ขั้นตอนการแปลงไฟล์ภาพเป็นเมตริกซ์สามารถแสดงได้ดังนี้

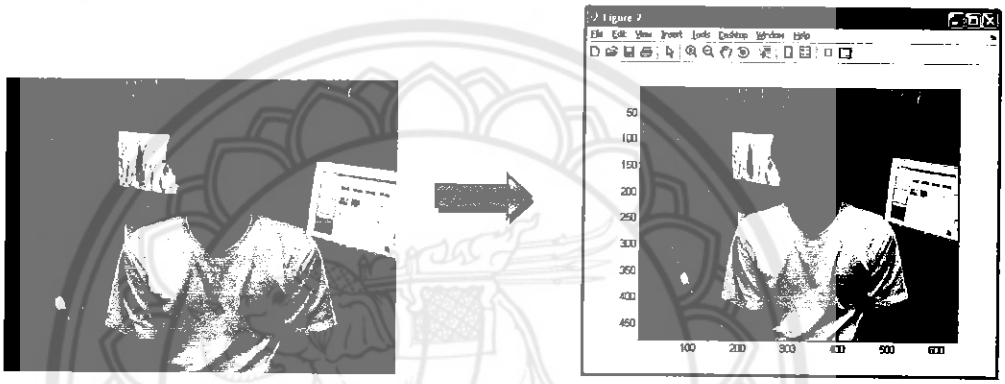
- เปิดไฟล์รูปภาพใบหน้าที่จัดเก็บไว้ในหน้าต่าง current directory ของโปรแกรม MATLAB เพื่อให้โปรแกรมสามารถมองเห็นไฟล์รูปภาพได้

- ทำการแปลงไฟล์รูปภาพเป็นเมตริกซ์ด้วยคำสั่ง " i = imread('ชื่อไฟล์') " ในหน้าต่าง command window ของโปรแกรม MATLAB จากนั้นทำการประมวลผลแล้วจะได้เมตริกซ์

ค่าของเมตริกซ์ที่ได้ในแต่ละแถวและในแต่ละหลักนั้นเป็นค่าเฉพาะของรูปภาพในแต่ละรูปเท่านั้น จึงเกิดความแตกต่างของรูปภาพขึ้น สามารถนำค่าของเมตริกซ์มาทำการประมวลผลแล้วนำค่าที่ได้มาใช้ในการเปรียบเทียบเพื่อระบุตัวบุคคลจากการตรวจจับใบหน้าคนจากภาพนิ่ง

3.2.2 การตรวจจับใบหน้าคนจากภาพนิ่ง [9]

โดยใช้โปรแกรมการตรวจจับใบหน้า ของ Mikael Nilsson, Jorgen Nordberg, and Ingvar Claesson จากเว็บไซต์ http://www.mathworks.com/matlabcentral/fileexchange/13701 ผลที่ได้คือตำแหน่งของส่วนที่เป็นใบหน้า ดังรูปที่ 3.2



รูปที่ 3.2 แสดงผลลัพธ์การตรวจจับใบหน้าคน

3.2.3 เตรียมข้อมูลภาพใบหน้าต้นแบบ (Template) สำหรับการเปรียบเทียบ

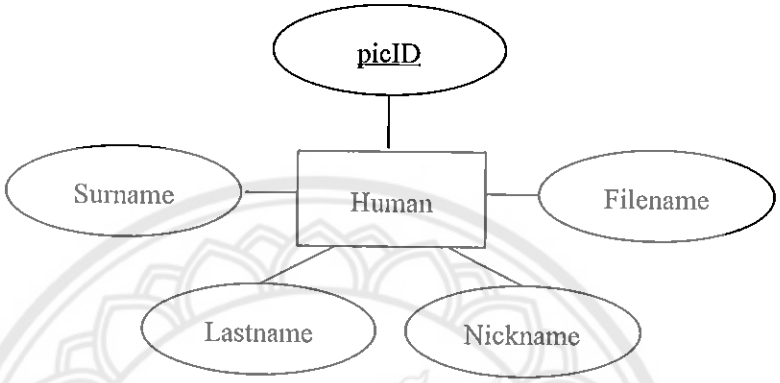
1) การจัดเก็บ Template

หลังจากได้ตำแหน่งใบหน้าของรูปที่ต้องการนำมาเป็น Template ในการเปรียบเทียบจากกระบวนการค้นหาใบหน้าแล้ว นำตำแหน่งของใบหน้ามาใช้เป็นขอบเขตในการตัดส่วนของรูปเพื่อนำรูปใบหน้าที่ได้มาเป็น Template และทำการจัดเก็บเป็นไฟล์รูปแบบไฟล์ฟอร์แมทใดก็ได้

รูปที่ 3.3 ตัวอย่าง รูป Template จากการตรวจจับใบหน้าคนจากภาพนิ่ง

2) การจัดเก็บข้อมูล

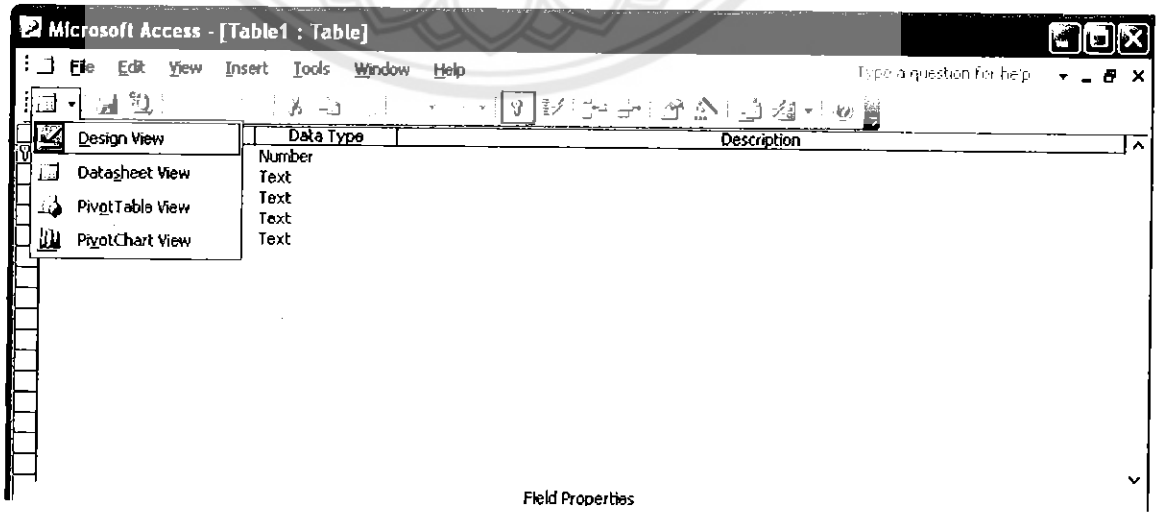
เมื่อได้ Template เรียบร้อยแล้ว ก็จะจัดเก็บข้อมูลของบุคคลในรูป Template ลงในฐานข้อมูล เนื่องจากข้อมูลมีเพียง รหัสรูปภาพ (picID) ชื่อ นามสกุล ชื่อเล่น และชื่อไฟล์รูปภาพ ทำให้ฐานข้อมูลไม่ซับซ้อนมากนัก ง่ายต่อการสืบค้น โดยการจัดเก็บข้อมูลลงฐานข้อมูลนั้นจะใช้ picID เป็น Primary Key สามารถเขียน E/R diagram ได้ดังรูปที่ 3.4 ซึ่งโครงงานนี้ใช้โปรแกรม Microsoft Access ในการจัดเก็บข้อมูล



รูปที่ 3.4 E/R diagram ของข้อมูลที่จะจัดเก็บ

ขั้นตอนการสร้างฐานข้อมูลให้กับโปรแกรมมีดังนี้

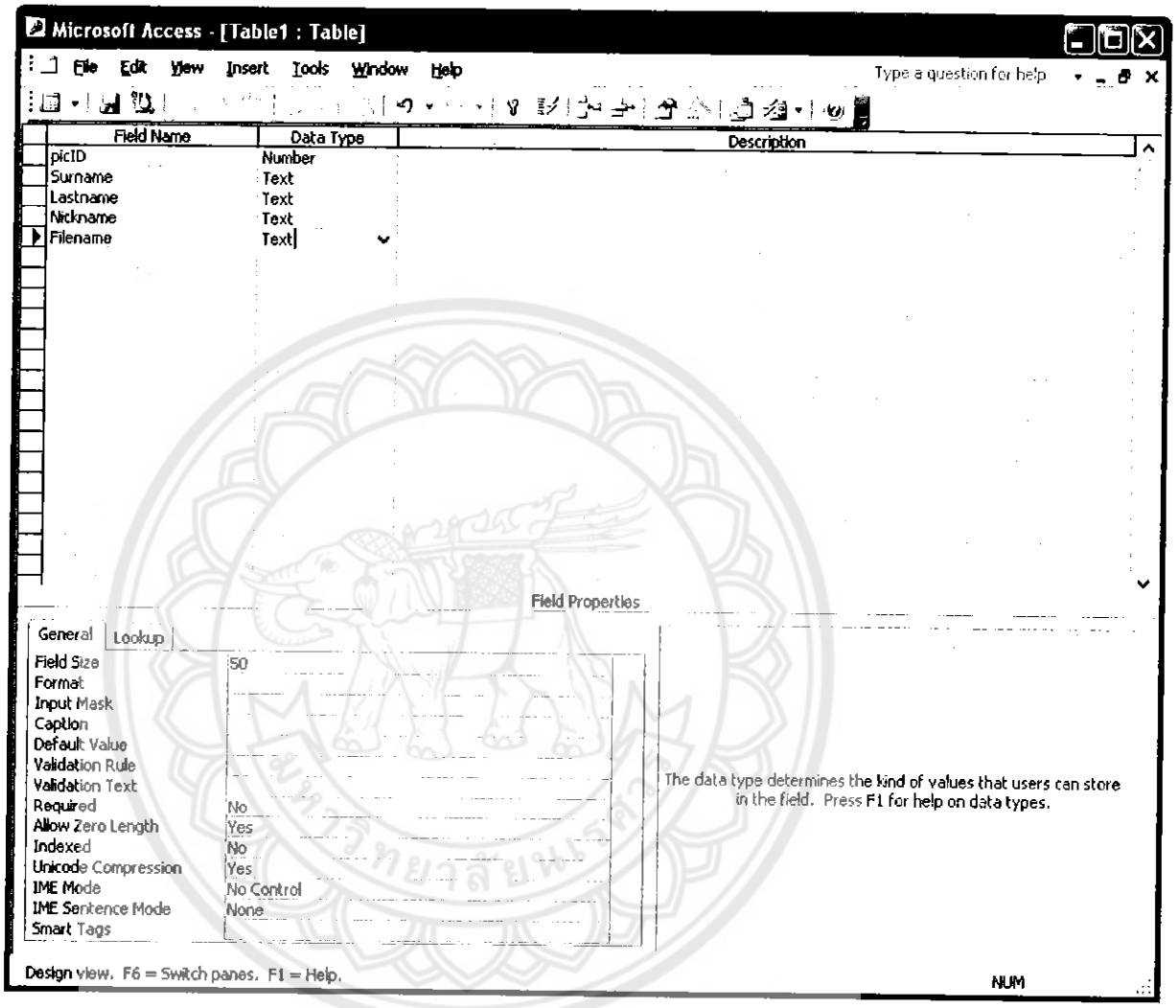
1. เปิดโปรแกรม Microsoft Access
2. เลือกสร้าง "ฐานข้อมูลเปล่า" แล้วทำการตั้งชื่อแฟ้มฐานข้อมูลเป็น "facerecognize.mdb"
3. ไปที่มุมมอง เลือก "มุมมองออกแบบ" ดังรูปที่ 3.5



รูปที่ 3.5 เริ่มสร้างฐานข้อมูลชื่อ "facerecognize"

4. ตั้งชื่อเขตข้อมูลเป็น “picID” ชนิดข้อมูลแบบ Number ,Surname ชนิดข้อมูลแบบ Text , Lastname ชนิดข้อมูลแบบ Text และ Nickname ชนิดข้อมูลแบบ Text

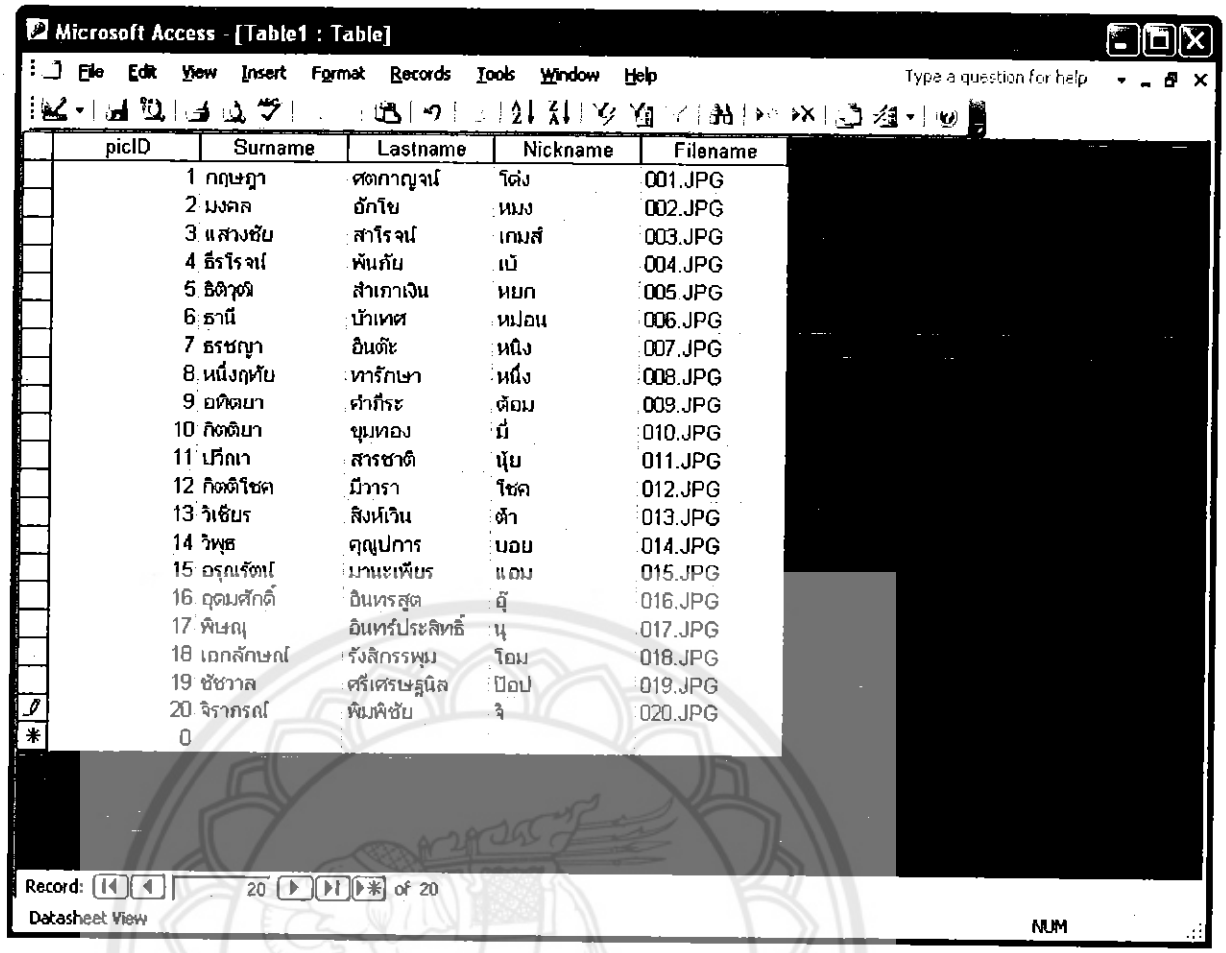
5. กำหนด Primary Key ให้กับเขตข้อมูล “picID” โดยคลิกที่แถบเครื่องมี “คีย์หลัก” ดังรูปที่ 3.6



รูปที่ 3. 6 การกำหนดคุณสมบัติต่างๆให้กับตาราง

6. บันทึกตารางโดยตั้งชื่อว่า “DataFace”

7. ดับเบิลคลิกที่ตาราง “DataFace” จะปรากฏฐานข้อมูลเปล่า ทำการใส่ข้อมูลโดยลำดับ picID ตามลำดับที่จัดเก็บ template ในแฟ้ม ดังรูปที่ 3.7



รูปที่ 3.7 การจัดเก็บข้อมูลลงฐานข้อมูล

8. บันทึกฐานข้อมูล

3.3 การค้นหาข้อมูลบุคคลในภาพ

3.3.1 เตรียมข้อมูลภาพสำหรับการระบุตัวบุคคล

รูปภาพที่นำมาใช้ในการตรวจจับใบหน้าจะต้องมีคุณสมบัติดังนี้

- รูปที่ใช้เป็นรูปภาพสี
- เป็นรูปภาพใบหน้าตรง ไม่สวมแว่นตา ไม่มีเส้นผมมาปรกหน้า
- แสงที่ใช้ในการถ่ายรูปอยู่ในระดับเพียงพอที่สามารถมองเห็นใบหน้าชัดเจน
- ไม่มีเงา หรือแสงสะท้อนปรากฏอยู่บนใบหน้า

นำรูปที่จะทำการทดลองซึ่งเป็นรูปเดี่ยว โดยบุคคล 1 คน จะมีรูปลักษณะหน้าตาต่างกันไม่ต่ำกว่า 5 รูปแล้วนำรูปไปจัดเก็บไว้ในโฟลเดอร์ ซึ่งการนำข้อมูลภาพมาใช้ในการระบุตัวบุคคลจากการตรวจจับใบหน้าคนจากภาพนิ่งนั้นจำเป็นต้องนำไฟล์รูปภาพใบหน้ามาแปลงเป็น

เมตริกซ์ เพื่อให้โปรแกรมที่ใช้ในการประมวลผลสามารถอ่านค่าได้ โดยใช้โปรแกรม MATLAB ในฟังก์ชัน IMREAD ซึ่งผลที่ได้คือค่าเมตริกซ์ ขั้นตอนการแปลงไฟล์รูปเป็นเมตริกซ์สามารถแสดงได้ดังนี้

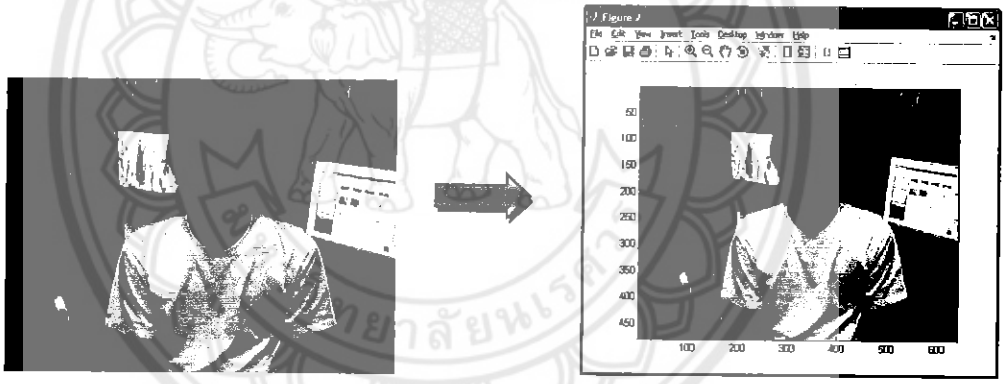
- เปิดไฟล์รูปภาพใบหน้าที่จัดเก็บไว้ในหน้าต่าง current directory ของโปรแกรม MATLAB เพื่อให้โปรแกรมสามารถมองเห็นไฟล์รูปภาพได้

- ทำการแปลงไฟล์รูปภาพเป็นเมตริกซ์ด้วยคำสั่ง " i = imread('ชื่อไฟล์') " ในหน้าต่าง command window ของโปรแกรม MATLAB จากนั้นทำการประมวลผลแล้วจะได้เมตริกซ์

ค่าของเมตริกซ์ที่ได้ในแต่ละแถวและในแต่ละหลักนั้นเป็นค่าเฉพาะของรูปภาพในแต่ละรูปเท่านั้น จึงเกิดความแตกต่างของรูปภาพขึ้น สามารถนำค่าของเมตริกซ์มาทำการประมวลผลแล้วนำค่าที่ได้มาใช้ในการเปรียบเทียบเพื่อระบุตัวบุคคลจากการตรวจจับใบหน้าคนจากภาพนิ่ง

3.3.2 การตรวจจับใบหน้าคนจากภาพนิ่ง [9]

โดยใช้โปรแกรมการตรวจจับใบหน้า ของ Mikael Nilsson, Jorgen Nordberg, and Ingvar Claesson จากเว็บไซต์ http://www.mathworks.com/matlabcentral/fileexchange/13701 ผลที่ได้คือตำแหน่งของส่วนที่เป็นใบหน้า ดังรูปที่ 3.2



รูปที่ 3.8 แสดงผลลัพธ์การตรวจจับใบหน้าคน

3.3.3 การเปรียบเทียบระหว่าง Template และภาพใบหน้า [7]

จากบทความเรื่อง "การค้นหาภาพใบหน้าโดยใช้การวิเคราะห์สหสัมพันธ์" ของ ดร. สนั่น ศรีสุข, ดร.คำรณ สุนัติ และ ดร.วีระศักดิ์ คุรุธัช สามารถนำมาเป็นแนวทางในการหาค่าความเหมือนระหว่าง Template และภาพใบหน้า โดยมีขั้นตอนดังนี้

1. อ่านไฟล์ภาพใบหน้าที่ต้องการระบุตัวบุคคลและ Template ทั้งหมด

2. หาค่า Correlation ระหว่างภาพใบหน้ากับ Template ทั้งหมด ตามสมการ

$$R = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} X(i,j)Y(i,j)}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} X(i,j)\sum_{i=1}^{m}\sum_{j=1}^{n} Y(i,j)}} \quad (3.1)$$

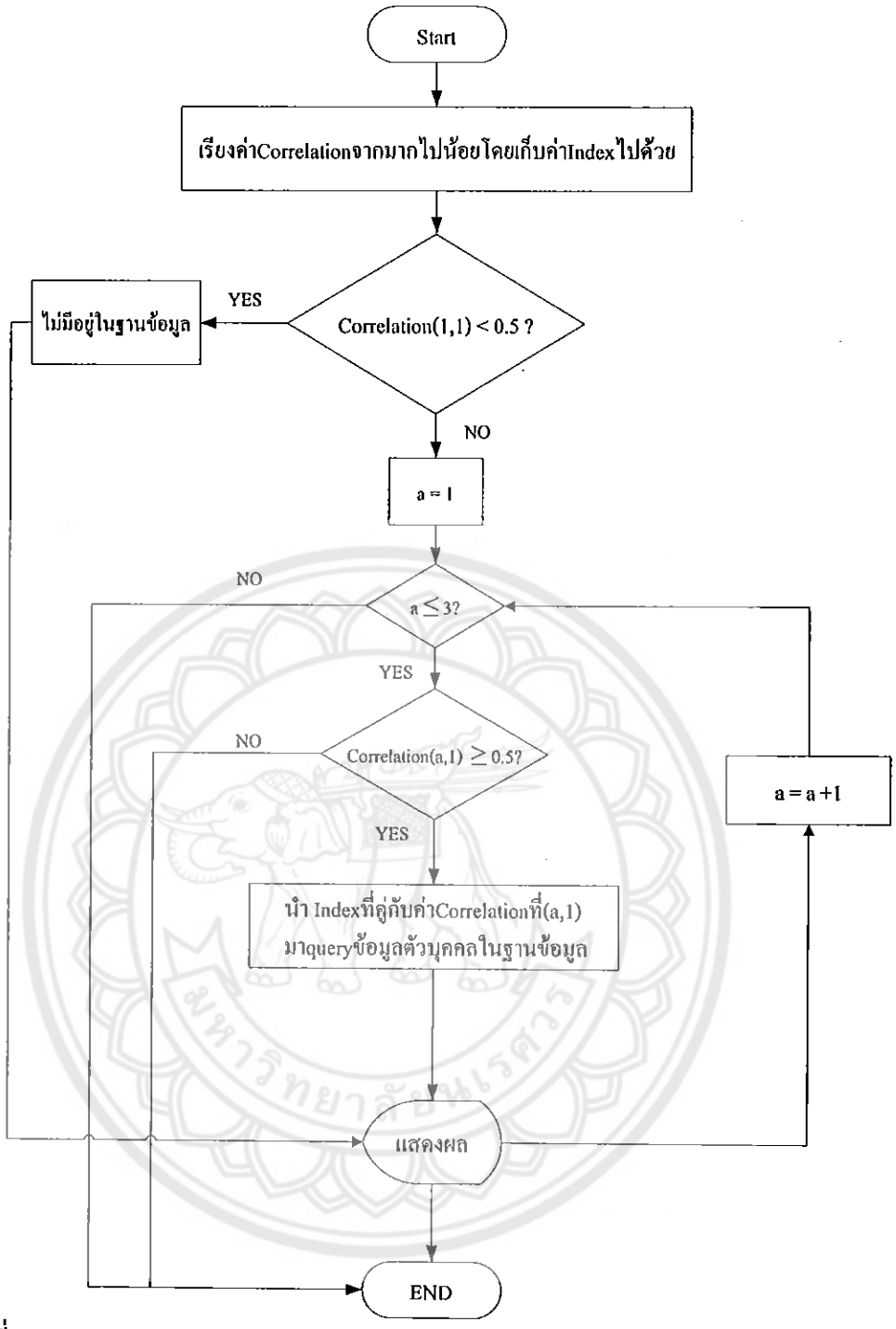
3. เก็บค่า Correlation – Coefficient ระหว่างระหว่างภาพใบหน้ากับ Template แต่ละ Template ลงในเมตริกซ์

3.3.4 การระบุตัวบุคคลด้วยการวิเคราะห์จากค่า Correlation – Coefficient

เปรียบเทียบค่า Correlation – Coefficient เพื่อค้นหาข้อมูลบุคคลในภาพ โดยที่จะตัดสินว่า ค่าความสัมพันธ์ใดสูงหรือต่ำ จะตัดสินจากระดับของค่าสัมประสิทธิ์สหสัมพันธ์จากการศึกษาและทดลองกับรูปภาพตัวอย่างต่างๆพบว่ามีนัยสำคัญ ดังนี้

- ค่าสัมประสิทธิ์สหสัมพันธ์ประมาณ 0.50 ถึง 1.00 ถือว่าสหสัมพันธ์อยู่ในระดับสูงหรือรูปที่นำมาเปรียบเทียบ มีส่วนที่เหมือนหรือคล้ายกับรูป Template
- ค่าสัมประสิทธิ์สหสัมพันธ์ประมาณ 0.00 ถึง 0.50 ถือว่าสหสัมพันธ์อยู่ในระดับต่ำหรือรูปที่นำมาเปรียบเทียบ ไม่มีส่วนคล้ายกับรูป Template

ซึ่งวิธีในการระบุตัวบุคคลเป็นไปตามแผนรูป ดังรูปที่ 3.7



รูปที่ 3.9 แผนภาพแสดงการระบุตัวบุคคลด้วยการวิเคราะห์จากค่า Correlation – Coefficient

จากแผนภาพขั้นตอนการระบุตัวบุคคลคือ การนำค่า Correlation Coefficient ที่เก็บไว้ในเมตริกซ์มาหาค่ามากสุดในสามลำดับแรกจากการเรียงค่าข้อมูล และต้องมีค่า Correlation Coefficient มากกว่า 0.5 เพื่อใช้ในการเลือกข้อมูลที่มีความคล้ายกับบุคคลในภาพจากฐานข้อมูล

3.3.5 การติดต่อฐานข้อมูล

การติดต่อฐานข้อมูลในโครงงานนี้จะเป็นการใช้ชุดคำสั่งของโปรแกรม Matlab โดยการเรียกดูข้อมูล จะเป็นภาษา SQL ทั่วไป ทำให้สะดวกต่อการติดต่อฐานข้อมูล เพราะมีชุดคำสั่งรองรับ ไม่ต้องพึ่งพาโปรแกรมอื่นเข้ามาช่วย

ตัวอย่างการใช้งานคำสั่ง SQL ผ่านฟังก์ชันของโปรแกรม Matlab

- คำสั่งใช้เชื่อมต่อฐานข้อมูล

```
conn=database(' Datasourcename ',' username ','password')
```

โดย

Datasourcename คือ ชนิดของฐานข้อมูลที่ได้ทำการตั้งค่าบน Data source

username และ password คือ ชื่อและรหัสผ่าน เพื่อเข้าถึงสิทธิต่างๆในฐานข้อมูล

conn คือ ตัวแปรที่เก็บค่าการเชื่อมต่อของฐานข้อมูลไว้

ตัวอย่าง

```
conn = database('Data','','');
```

คือ การเชื่อมต่อฐานข้อมูล MS access มีชื่อเป็น Data ไม่กำหนดชื่อและรหัสผ่าน

- คำสั่งที่ใช้เรียกดูข้อมูลภายในฐานข้อมูลตามเงื่อนไขที่ต้องการด้วยคำสั่ง SQL

```
curs = exec(conn,'sqlquery')
curs = fecth(curs)
```

โดย

conn คือ ตัวแปรที่เก็บค่าการเชื่อมต่อของฐานข้อมูลไว้

sqlquery คือ คำสั่ง SQL ที่ใช้เรียกดูข้อมูล

curs คือ ข้อมูลที่ได้จากการเรียกดู

ตัวอย่าง

```
curs = exec(conn,['SELECT ALL Surname,Lastname,Nicname
                   FROM DataFace WHERE picID = ' '' ind ''']);
curs = fetch(curs);
```

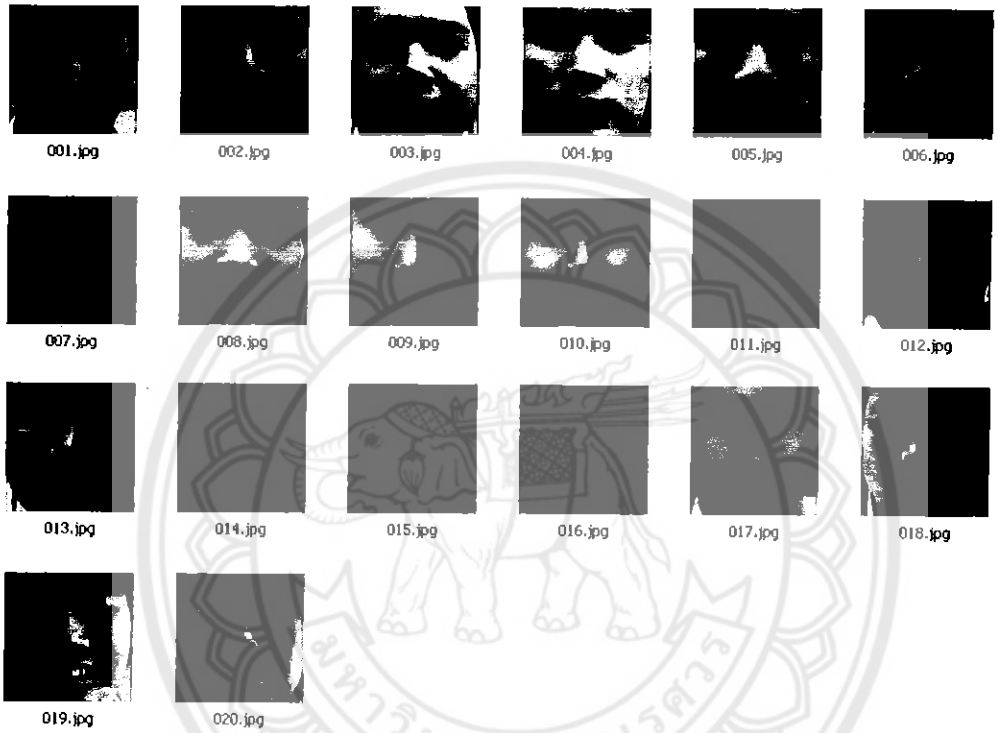
คือ การเรียกดูข้อมูลที่มีค่า picID เท่ากับ ind

บทที่ 4

ผลการทดลองและวิเคราะห์ผลการทดลอง

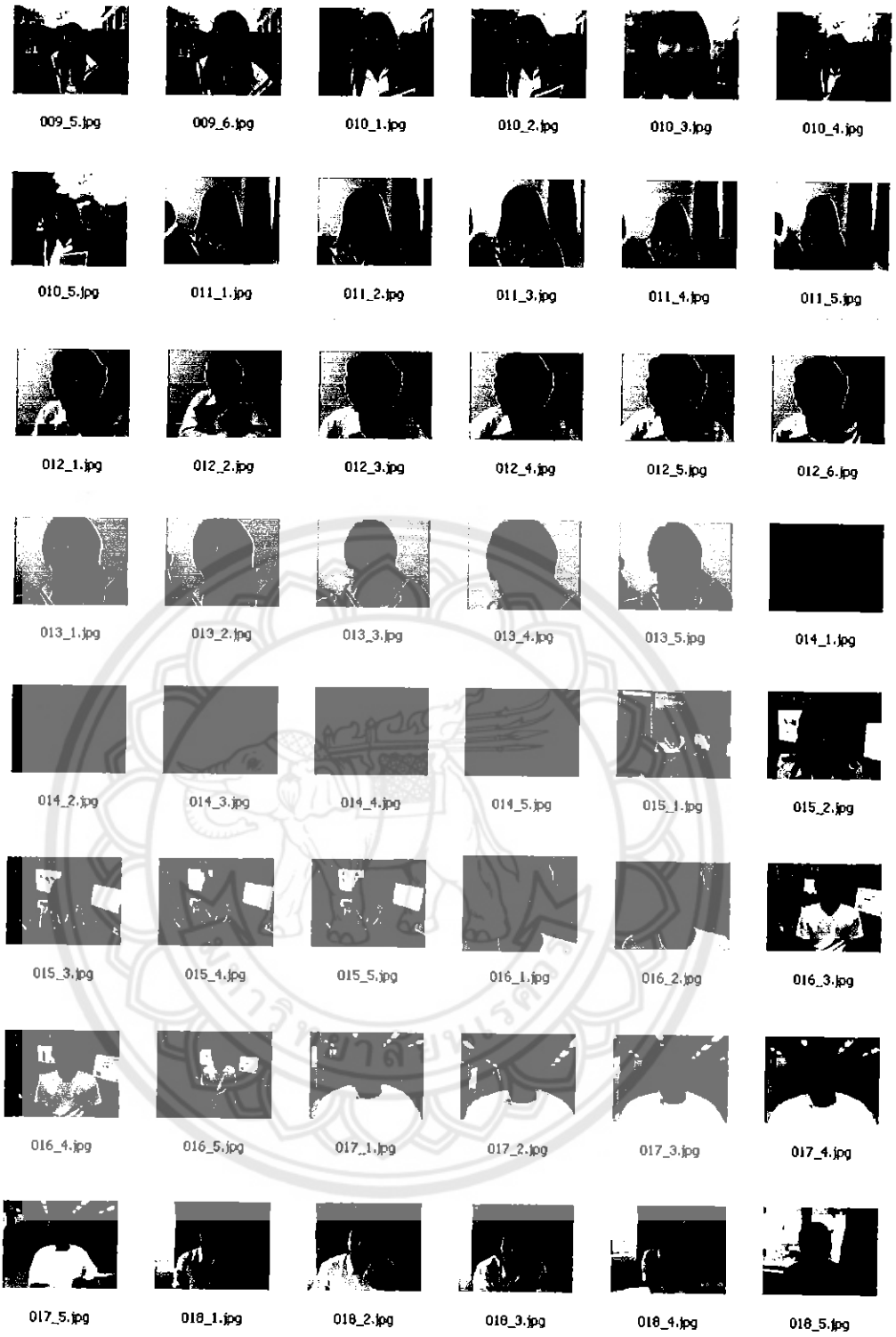
4.1 ผลการทดลอง

รูปภาพที่ใช้ในการทดสอบมีดังนี้

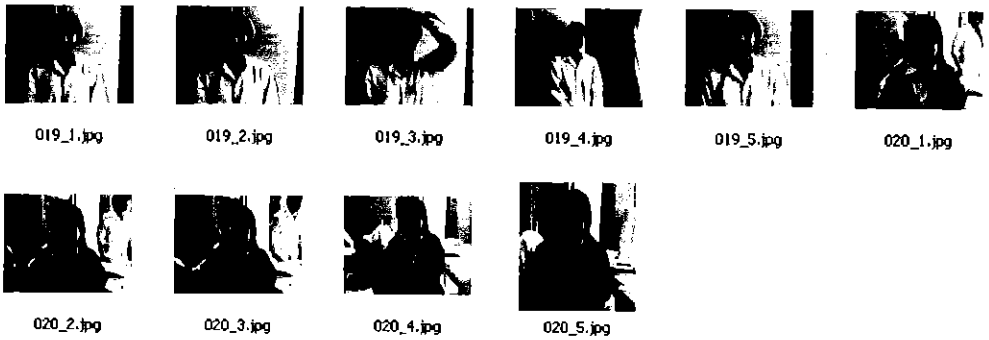


รูปที่ 4.1 รูปภาพ template ขนาด 161x161 ที่ใช้ในการเปรียบเทียบ

รูปที่ 4.2 รูปภาพที่นำมาใช้ในการทดสอบ

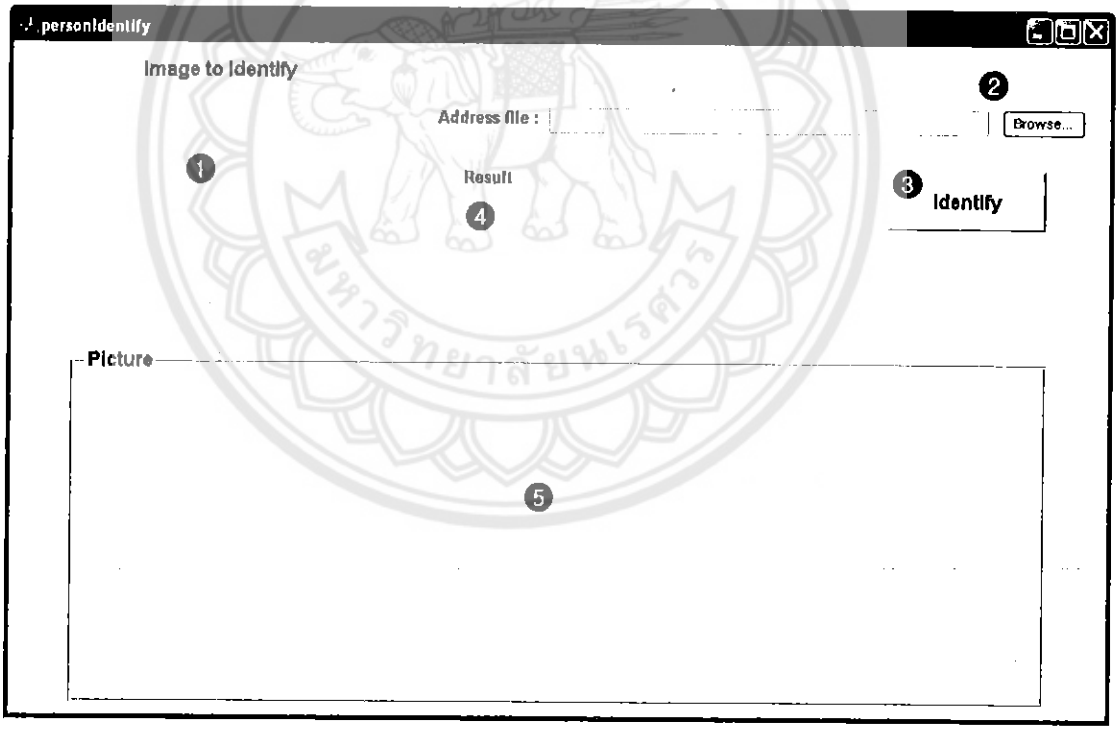


รูปที่ 4.2(ต่อ) รูปภาพที่นำมาใช้ในการทดสอบ



รูปที่ 4.2(ต่อ) รูปภาพที่นำมาใช้ในการทดสอบ

เมื่อเข้าสู่โปรแกรม Matlab ทำการ set Current Directory ให้ตรงกับไฟล์โปรแกรมที่จะใช้ประมวลผลแล้วพิมพ์คำสั่ง personIdentify ที่ Command Window แล้วกด Enter จะปรากฏหน้าต่างโปรแกรมในการระบุตัวบุคคล เพื่อทำการประมวลผล ดังรูปที่ 4.3



รูปที่ 4.3 หน้าต่างโปรแกรมในการระบุตัวบุคคล

ในหน้าต่างโปรแกรมประกอบด้วยส่วนต่างๆดังนี้

1. แสดงรูปภาพที่จะนำมาระบุตัวบุคคล

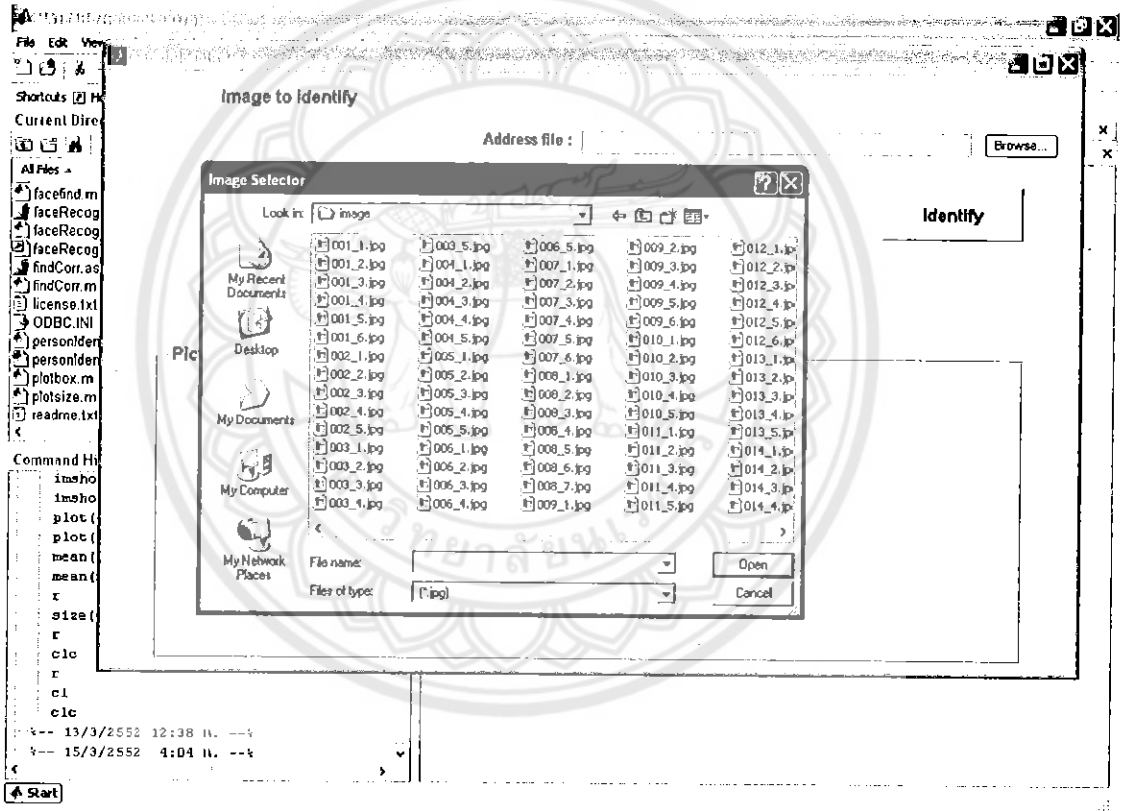
2. ปุ่ม browse สำหรับเลือกไฟล์ภาพที่ต้องการระบุตัวบุคคลและ text box แสดงที่อยู่ของไฟล์ภาพ

3. ปุ่มประมวลผลโปรแกรม

4. ข้อมูลของบุคคลในภาพที่ได้จากการระบุตัวบุคคล

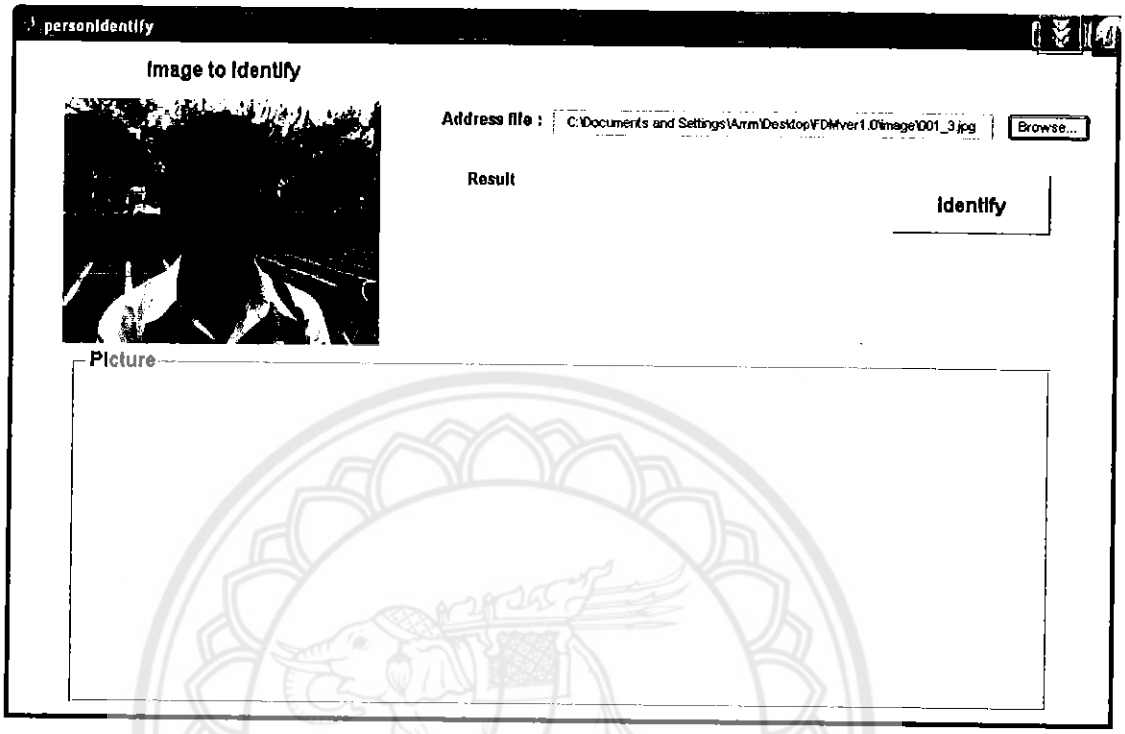
5. แสดงรูปภาพที่สามารถระบุได้ใกล้เคียงมากที่สุด

จากนั้นให้ใส่ไฟล์ภาพที่ต้องการระบุตัวบุคคล โดยค้นหาไฟล์ภาพที่ปุ่ม browse ดังรูปที่ 4.4



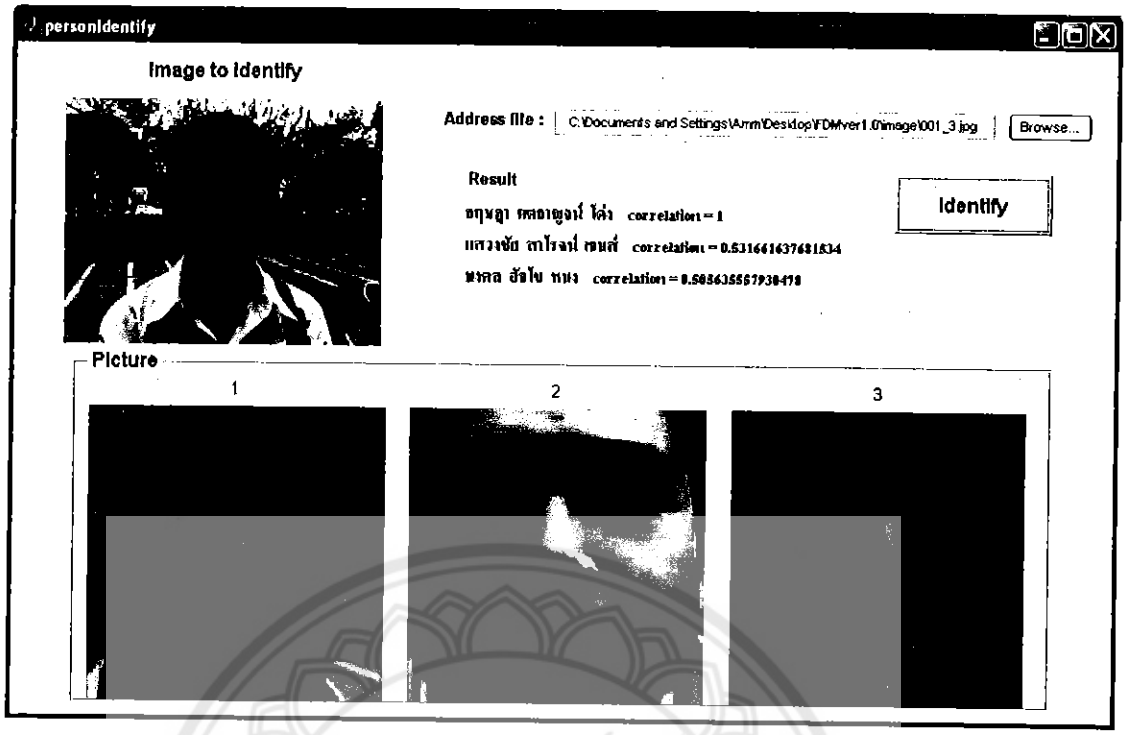
รูปที่ 4.4 แสดงการเลือกไฟล์ภาพที่จะนำมาระบุตัวบุคคล

เมื่อได้ภาพที่ต้องการระบุตัวบุคคล จะปรากฏภาพที่ต้องการระบุและที่อยู่ของไฟล์ภาพขึ้น ดังรูปที่ 4.5



รูปที่ 4.5 แสดงผลหลังจากระบุไฟล์ภาพที่ต้องการระบุตัวบุคคล

เมื่อได้ภาพที่ต้องการระบุตัวบุคคลแล้ว ให้กดปุ่ม Identify เพื่อทำการประมวลผลค้นหาข้อมูลของบุคคลในภาพ หลังจากที่โปรแกรมทำการประมวลผลเสร็จสิ้น จะแสดงข้อมูลบุคคลในภาพ และแสดงรูปภาพที่สามารถระบุตัวบุคคลได้ โดยเรียงลำดับความเหมือนจากมากไปน้อย ดังรูปที่ 4.6



รูปที่ 4.6 แสดงผลที่ได้จากการระบุตัวบุคคล

จากรูปที่ 4.6 แสดงผลที่ได้จากการระบุตัวบุคคลจากโปรแกรม โดยแสดง template ของบุคคลที่มีลักษณะใบหน้าใกล้เคียงกับรูปภาพที่นำมาใช้ทดสอบตามลำดับความใกล้เคียงโดยอ้างอิงจากค่า correlation coefficient และแสดงข้อมูลของบุคคลใน template ตามลำดับการแสดงภาพ

ในทดสอบการระบุตัวบุคคล แบ่งออกเป็นการทดสอบกับภาพที่ไม่ผ่านกระบวนการ histogram equalization ก่อนนำมาเปรียบเทียบ และการทดสอบกับภาพที่ผ่านกระบวนการ histogram equalization ก่อนนำมาเปรียบเทียบ เมื่อนำภาพทั้งหมดมาทดสอบเพื่อระบุตัวบุคคล สามารถจำแนกผลการระบุตัวบุคคลได้เป็น สามารถระบุตัวบุคคลได้ถูกต้องคือ ข้อมูลของบุคคลในภาพอยู่ในอันดับแรกของการแสดงผล ,ระบุตัวบุคคลผิดพลาดคือ ข้อมูลไม่อยู่ในอันดับแรกแต่อยู่ใน 2 อันดับที่เหลือ และไม่สามารถระบุตัวบุคคลได้ โดยมีผลการทดลองตามตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองการระบุตัวบุคคลจากการตรวจจับใบหน้า

ชื่อไฟล์	ไม่ผ่านการ histogram equalization			ผ่านการ histogram equalization		
	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้
001_1.jpg		✓(ลำดับ2)			✓(ลำดับ2)	
001_2.jpg	✓			✓		
001_3.jpg	✓			✓		
001_4.jpg	✓			✓		
001_5.jpg	✓			✓		
001_6.jpg			✓			✓
002_1.jpg	✓			✓		
002_2.jpg	✓			✓		
002_3.jpg	✓			✓		
002_4.jpg	✓			✓		
002_5.jpg			✓			✓
003_1.jpg	✓			✓		
003_2.jpg	✓			✓		
003_3.jpg	✓			✓		
003_4.jpg	✓			✓		
003_5.jpg			✓	✓		
004_1.jpg	✓			✓		
004_2.jpg	✓			✓		
004_3.jpg	✓			✓		
004_4.jpg		✓(ลำดับ2)		✓		
004_5.jpg	✓			✓		
005_1.jpg	✓			✓		
005_2.jpg	✓			✓		
005_3.jpg	✓			✓		
005_4.jpg	✓			✓		
005_5.jpg		✓(ลำดับ2)			✓(ลำดับ2)	

ตารางที่ 4.1(ต่อ) ผลการทดลองการระบุตัวบุคคลจากการตรวจจับใบหน้า

ชื่อไฟล์	ไม่ผ่านการ histogram equalization			ผ่านการ histogram equalization		
	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้
006_1.jpg	✓			✓		
006_2.jpg	✓			✓		
006_3.jpg	✓			✓		
006_4.jpg	✓			✓		
006_5.jpg	✓			✓		
007_1.jpg	✓			✓		
007_2.jpg		✓(ลำดับ3)		✓		
007_3.jpg	✓			✓		
007_4.jpg	✓			✓		
007_5.jpg	✓			✓		
007_6.jpg	✓			✓		
008_1.jpg			✓			✓
008_2.jpg	✓			✓		
008_3.jpg	✓			✓		
008_4.jpg	✓			✓		
008_5.jpg	✓			✓		
008_6.jpg	✓			✓		
008_7.jpg	✓			✓		
009_1.jpg	✓			✓		
009_2.jpg	✓			✓		
009_3.jpg	✓			✓		
009_4.jpg	✓			✓		
009_5.jpg	✓			✓		
009_6.jpg	✓			✓		
010_1.jpg	✓			✓		
010_2.jpg	✓			✓		

ตารางที่ 4.1(ต่อ) ผลการทดลองการระบุตัวบุคคลจากการตรวจจับใบหน้า

ชื่อไฟล์	ไม่ผ่านการ histogram equalization			ผ่านการ histogram equalization		
	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้
010_3.jpg		✓(ลำดับ2)		✓		
010_4.jpg		✓(ลำดับ2)			✓(ลำดับ2)	
010_5.jpg	✓			✓		
011_1.jpg	✓			✓		
011_2.jpg	✓			✓		
011_3.jpg	✓			✓		
011_4.jpg	✓			✓		
011_5.jpg		✓(ลำดับ2)			✓(ลำดับ2)	
012_1.jpg	✓			✓		
012_2.jpg	✓			✓		
012_3.jpg	✓			✓		
012_4.jpg	✓			✓		
012_5.jpg	✓			✓		
013_1.jpg	✓			✓		
013_2.jpg	✓			✓		
013_3.jpg	✓			✓		
013_4.jpg		✓(ลำดับ2)			✓(ลำดับ2)	
013_5.jpg	✓			✓		
014_1.jpg	✓			✓		
014_2.jpg	✓			✓		
014_3.jpg	✓			✓		
014_4.jpg	✓			✓		
014_5.jpg	✓			✓		
015_1.jpg	✓			✓		
015_2.jpg	✓			✓		
015_3.jpg	✓			✓		

ตารางที่ 4.1(ต่อ) ผลการทดลองการระบุตัวบุคคลจากการตรวจจับใบหน้า

ชื่อไฟล์	ไม่ผ่านการ histogram equalization			ผ่านการ histogram equalization		
	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้
015_4.jpg	✓			✓		
015_5.jpg	✓			✓		
016_1.jpg	✓			✓		
016_2.jpg	✓			✓		
016_3.jpg	✓			✓		
016_4.jpg	✓			✓		
016_5.jpg		✓(ลำดับ3)			✓(ลำดับ3)	
017_1.jpg	✓			✓		
017_2.jpg	✓			✓		
017_3.jpg	✓			✓		
017_4.jpg	✓			✓		
017_5.jpg	✓			✓		
018_1.jpg	✓			✓		
018_2.jpg	✓			✓		
018_3.jpg	✓			✓		
018_4.jpg	✓			✓		
018_5.jpg			✓	✓		
019_1.jpg	✓			✓		
019_2.jpg	✓			✓		
019_3.jpg	✓			✓		
019_4.jpg	✓			✓		
019_5.jpg	✓			✓		
020_1.jpg	✓			✓		
020_2.jpg	✓			✓		
020_3.jpg	✓			✓		
020_4.jpg	✓			✓		

ตารางที่ 4.1(ต่อ) ผลการทดลองการระบุตัวบุคคลจากการตรวจจับใบหน้า

ชื่อไฟล์	ไม่ผ่านการ histogram equalization			ผ่านการ histogram equalization		
	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้	ระบุตัวบุคคลได้ถูกต้อง	ระบุตัวบุคคลผิดพลาด	ไม่สามารถระบุตัวบุคคลได้
020_5.jpg	✓			✓		
รวม	91	9	5	96	6	3

4.2 วิเคราะห์ผลการทดลอง

จากการทดลองพบว่าในการระบุตัวบุคคลจากการตรวจจับภาพใบหน้า มีผลการทดลองดังนี้ คือในกรณีที่ไม่ทำการ histogram equalization กับภาพก่อนนำไปเปรียบเทียบ พบว่าสามารถระบุตัวบุคคลได้ถูกต้อง 91 ภาพจากภาพที่นำมาทดสอบทั้งหมด คิดเป็นร้อยละ 86.67 ระบุตัวบุคคลผิดพลาดจำนวน 9 ภาพ จากภาพที่นำมาทดสอบทั้งหมด คิดเป็นร้อยละ 8.57 และไม่สามารถระบุได้จำนวน 5 ภาพ จากภาพที่นำมาทดสอบทั้งหมด คิดเป็นร้อยละ 4.76 โดยเมื่อพิจารณาภาพที่เกิดข้อผิดพลาดสามารถวิเคราะห์สาเหตุที่ทำให้เกิดข้อผิดพลาดได้หลายประการ เช่น เส้นผมที่มาบดบังใบหน้า, หรือภาพมีแสงสว่างที่แตกต่างจาก Template มากเกินไป และเมื่อนำภาพมาผ่านกระบวนการ histogram equalization ก่อนนำไปเปรียบเทียบเพื่อแก้ปัญหาภาพมีแสงสว่างที่แตกต่างจาก Template มากเกินไป พบว่าสามารถระบุตัวบุคคลได้ถูกต้อง 96 ภาพจากภาพที่นำมาทดสอบทั้งหมด คิดเป็นร้อยละ 91.43 ระบุตัวบุคคลผิดพลาดจำนวน 6 ภาพ จากภาพที่นำมาทดสอบทั้งหมด คิดเป็นร้อยละ 5.71 และไม่สามารถระบุได้จำนวน 3 ภาพ จากภาพที่นำมาทดสอบทั้งหมด คิดเป็นร้อยละ 2.86 ซึ่งให้ผลการระบุตัวบุคคลได้ดีขึ้นจากเดิม

บทที่ 5
สรุป

5.1 สรุปผลการทดลอง

จากการทดลองการระบุตัวบุคคลจากการตรวจจับใบหน้าโดยใช้วิธีการเปรียบเทียบโดยวิเคราะห์จากค่าสัมประสิทธิ์สหสัมพันธ์ ซึ่งผลที่ได้จากการระบุตัวบุคคลคือข้อมูลของบุคคลในรูปภาพ หลังจากทดลองประมวลผลแล้วพบว่า สามารถระบุตัวบุคคลได้มากกว่า 80% ซึ่งข้อผิดพลาดที่เกิดขึ้นตัดสินจากการระบุตัวบุคคลผิดพลาด และไม่สามารถระบุตัวบุคคลได้ เมื่อพิจารณาภาพที่เกิดข้อผิดพลาดสามารถวิเคราะห์สาเหตุที่ทำให้เกิดข้อผิดพลาดได้หลายประการ เช่น เส้นผมที่มาบดบังใบหน้า , หรือภาพมีแสงสว่างที่แตกต่างจาก Template มากเกินไป และเมื่อนำวิธีการ histogram equalization มาปรับปรุงภาพก่อนนำไปเปรียบเทียบทำให้สามารถระบุตัวบุคคลได้ถูกต้องมากขึ้น โดยสามารถระบุตัวบุคคลได้มากกว่า 90%

5.2 ปัญหาและแนวทางแก้ไข

1. ในการตรวจจับภาพใบหน้าบางภาพเกิดการตรวจจับผิดหรือตรวจจับได้มากกว่า 1 หน้า ซึ่งสามารถแก้ได้โดยกำจัดขอบเขตของรูปภาพให้แคบลง เพื่อช่วยให้สามารถตรวจจับใบหน้าได้ถูกต้อง

2. วิธีการวิเคราะห์ด้วยค่าสัมประสิทธิ์สหสัมพันธ์ ไม่สามารถเปรียบเทียบใบหน้าบุคคลในค่าความสว่างที่แตกต่างกันมากๆ สามารถแก้ไขได้โดยนำวิธีการ histogram equalization มาปรับปรุงภาพก่อนนำไปเปรียบเทียบ

3. ไฟล์ภาพที่ต้องการระบุตัวบุคคลมีขนาดใหญ่ จะทำให้ไม่สามารถตรวจจับภาพใบหน้าได้ ซึ่งอาจเป็นข้อจำกัดของโปรแกรมการตรวจจับภาพใบหน้าที่นำมาใช้ แก้ไขโดยกำหนดขนาดของภาพให้เล็กลง

5.3 แนวทางการพัฒนาในอนาคต

1. สามารถนำวิธีการเปรียบเทียบใบหน้าแบบอื่นมาประยุกต์ใช้งานร่วมกับโครงงานนี้เพื่อเพิ่มความแม่นยำ และประสิทธิภาพในการระบุตัวบุคคล

2. สามารถนำโครงงานนี้ไปพัฒนาต่อให้สามารถระบุตัวบุคคลตรวจจับใบหน้าในลักษณะท่าทางต่างๆได้ เช่น หันหน้าไม่ตรง ก้มหน้า เงยหน้า เป็นต้น

เอกสารอ้างอิง

[1] **"Chapter 5 Input."** [Online]. Available :
http://www.spu.ac.th/~masakul/Chapter05-2006 (thai).ppt. 2008.

[2] [Online]. Available :
http://www.software.co.th/software/ProductDetail.asp?ProductID=31&CategoryTopicID=11. 2008.

[3] กนกวรรณ ชำนาญจุ้ย, วาสนา วงศ์ษา. **"การตรวจหาลายภาพ (Texture Detection)."** วิทยานิพนธ์วิศกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์. มหาวิทยาลัยนเรศวร 2546.

[4] **"ระบบฐานข้อมูล."**[Online]. Available :
http://www.chandra.ac.th/office/ict/document/it/it04/page01.html. 2009.

[5] **"PHP Database ODBC."**[Online]. Available :
http://www.phpstreet.com/tutorials/php/php_db_odbc.php. 2009.

[6] **"Open Database Connectivity (ODBC) ."**[Online]. Available :
http://www.bcoms.net/dictionnary/detail.asp?id=416. 2009.

[7] ดร.สนั่น ศรีสุข, ดร.คำรณ สุนัติ และ ดร.วีระศักดิ์ คุรุธัช. **"การค้นหาภาพใบหน้าโดยใช้การวิเคราะห์สหสัมพันธ์ (Face Detection using Eigen Face and Correlation Analysis)."** [Online]. Available :
http://berry.cpe.mut.ac.th/research/sanun/papers/conferences/local/FDuEFaCA.pdf. 2009.

[8] นายทรรศ จอมขันธิพล. **"ระบบจดจำใบหน้า (Face Recognition)."** [Online]. Available :
http://cpe.rsu.ac.th/ut/courses/T1-51/cpe489/portfolio/482316/research.pdf. 2009.

[9] Mikael Nilsson, Jorgen Nordberg, and Ingvar Claesson. **"FACE DETECTION USING LOCAL SMQT FEATURES AND SPLIT UP SNOWCLASSIFIER."** [Online]. Available :
http://www.bth.se/fou/Forskinfo.nsf/all/367c46cdcf754d17c12572cf002eac41/$file/paper_revisited_070119.pdf. 2009.

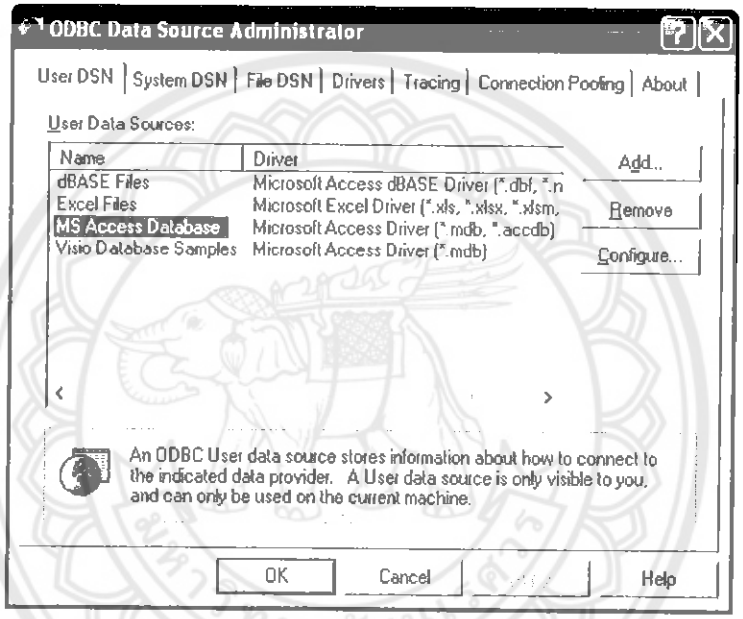
[10] อาจารย์จิระศักดิ์ สิทธิกร. **"Image Enhancement in the Spatial Domain."** [Online]. Available : http://www.kmitl.ac.th/~ksjirasa/Lecture/ImagePro/Lecture04_1.pdf. 2009.

[11] โศรฎา แข็งการ ,กนต์ธร ชำนิประศาสน์. **"การใช้ MATLAB สำหรับงานทางวิศวกรรม"** [pdf.].

ภาคผนวก ก.
การติดตั้งฐานข้อมูลบุคคล

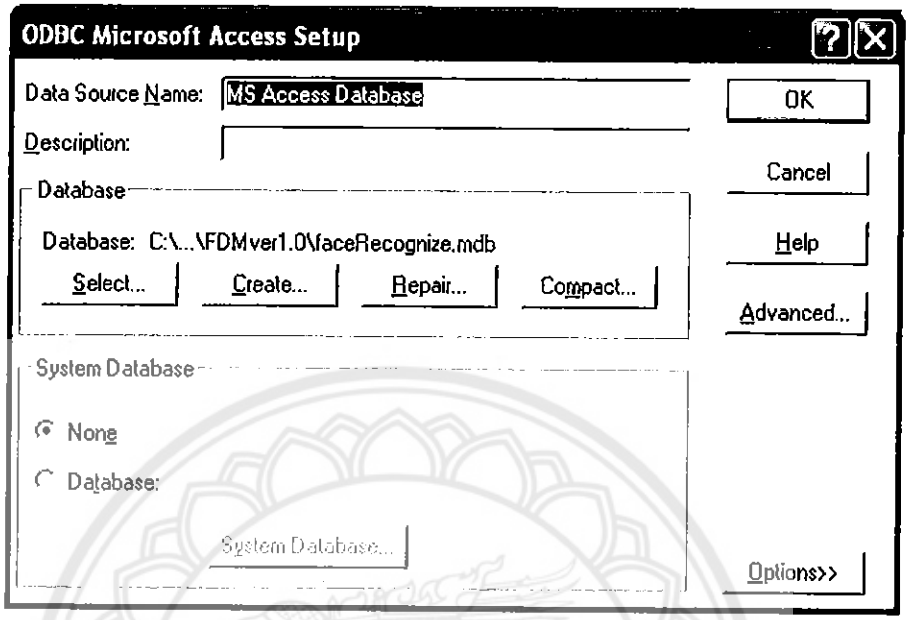
การติดตั้งฐานข้อมูล

1. ทำการ Copy folder "Database" ไว้ในใดก็ได้ตามต้องการ
2. ไปที่ Start -> Control Panel -> Administrative Tools -> Data Sources (ODBC)



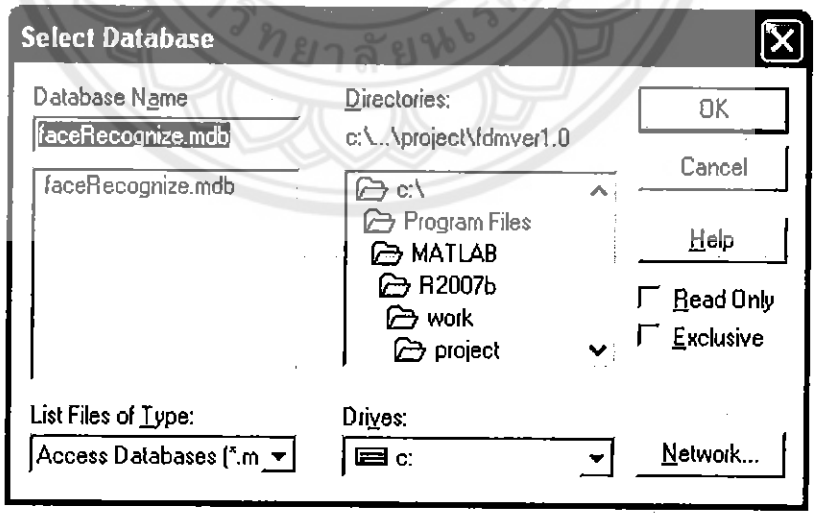
รูปที่ ข-1 ส่วนของการติดตั้ง Data Source (ODBC) เพื่อทำการติดตั้ง Database

3. เลือก MS Access Database แล้วกด Configure หรือ ดับเบิลคลิกที่ชื่อ จะแสดงหน้าต่างนี้ขึ้นมา ทำการเปลี่ยนชื่อ Data Source Name เป็น Data ในการ Connect กับโปรแกรม ดังรูป



รูปที่ ก-2 ทำการเปลี่ยนชื่อ Data Source Name

4. คลิกเลือกที่ Select จะปรากฏหน้าต่าง Select Database เกิดขึ้น



รูปที่ ก-3 คลิกเลือกไปยังที่อยู่ของไฟล์ faceRecognize.mdb

ให้คลิกเลือกที่อยู่โดยไปที่โฟลเดอร์ที่ Copy ไฟล์ไว้จะปรากฏฐานข้อมูลชื่อ "faceRecognize.mdb" คลิกที่ชื่อแล้วกด OK

ภาคผนวก ข.

ตัวอย่างโค๊ดโปรแกรม

โปรแกรมควบคุม GUI

เป็นส่วนควบคุม GUI ต่างๆ

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

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

% Last Modified by GUIDE v2.5 12-Mar-2009 22:21:19

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

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


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

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

% Update handles structure
guidata(hObject, handles);

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


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

% Get default command line output from handles structure
varargout{1} = handles.output;
axis(handles.axes1,'off')

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename, pathname]=uigetfile({'*.jpg';'*.*'},'Image Selector');
ss=[pathname,filename];
set(handles.edit1,'String',ss);
Img=imread(ss);
image(Img,'Parent',handles.axes1)
axis(handles.axes1,'off')


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
```

```
%        str2double(get(hObject,'String')) returns contents of edit1
as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

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


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ss=get(handles.edit1,'String');
if isempty(ss)
    msgbox('Please enter image file','error','warn');
    return
end
faceRecog(handles,ss)


% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox1 contents
as cell array
%        contents{get(hObject,'Value')} returns selected item from
listbox1


% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

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

โปรแกรมการระบุตัวบุคคลจากการตรวจจับใบหน้า

เป็น main ฟังก์ชัน ใช้เรียกฟังก์ชัน ex01 กับ corr แล้วนำค่าต่างๆที่ได้มาจาก 2 ฟังก์ชันมาทำการ Query หาข้อมูลของบุคคลในรูปภาพที่นำมาทดสอบ

```
%-----------------------------------------------------
%function to classifier face
%@since 2009 February 25
%@note Matlab 7.5.0(R2007b)
%@note Windows XP SP2
%-----------------------------------------------------
function [] = faceRecog(handles,ss)
clc
% -----------------------------------------------------
% Main
% @param NameFile is file name
% input is String
% -----------------------------------------------------
% NameFile = input('Enter image to identify : ','s');
faceImg = ex01(ss);
[row col] = size(faceImg);
k=1;

%@record Correlation value in index(k,1) and index in index (k,2)
index =zeros(20,1);
for i = 1:20;
    i = mat2str(i);
    conn = database('Data','','');
    namefile = exec(conn,['SELECT ALL Filename FROM DataFace WHERE
picID = ' '' i ''']);
    namefile = fetch(namefile);
    namefile = char(namefile.data);
    faceData = imread(namefile);
    [rowData colData] = size(faceData);
    cor = corr(faceImg,faceData,row,col,rowData,colData);
    index(k,1) = cor;
    index(k,2) = k;
    k = k+1;
end

%Sort data using bubble sort (most value to less value)
for i = 1:20-1
    for i = 1:20-1
        if(index(i,1)<index(i+1,1))
        tempcor = index(i,1);
        tempin = index(i,2);
        index(i,1) = index(i+1,1);
        index(i,2) = index(i+1,2);
        index(i+1,1) = tempcor;
        index(i+1,2) = tempin;
        end
    end
end

% find correlation coefficient in third higthest and have value more
0.5 to query in database
if index(1,1) < 0.5;
    set(handles.text1,'String','ไม่มีอยู่ในฐานข้อมูลครับ!!!');
else
```

```
    x=0;
    tmppanel=uipanel('Parent',handles.uipanel1,...
                     'Position',[0 0 1 1],'BorderWidth',0);
    for i = 1:3
        if index(i,1) > 0.5;
            ind = mat2str(index(i,2));
            cor = mat2str(index(i,1));
            curs1 = exec(conn,['SELECT ALL Surname,Lastname,...
                  Nickname FROM DataFace WHERE picID = ' '' ind '']);
            curs2 = exec(conn,['SELECT ALL Filename FROM ...
                  DataFace WHERE picID = ' '' ind '']);
            curs1 = fetch(curs1);
            curs2 = fetch(curs2);
            curs2 = char(curs2.data);
            name = [curs1.data cor];
            strname{i,1}=sprintf('%s \t %s \t %s \t correlation =...
            %s',name{1,1},name{1,2},name{1,3},name{1,4});
            order = mat2str(i);
            handles.axes(i)=axes('Parent',tmppanel,'Position',...
            [x 0 0.33 0.9]);
            x=x+0.33;
            imshow(curs2);
            title(order);
        end
    end
    set(handles.text1,'String',strname);
    guidata(handles.figure1,handles);
end
```

โปรแกรมในส่วนของการตรวจจับใบหน้า

โปรแกรมจะทำการรับ Image เข้ามาเพื่อตรวจจับหาบริเวณของใบหน้าแล้วเมื่อตรวจจับเสร็จจะเก็บไว้ในตัวแปรชื่อ faceImg เพื่อนำ faceImg กลับไปเปรียบเทียบกับใบหน้าที่เก็บไว้แล้ว

```
%Simple demonstration of one image face detection operation.
function [faceImg] = ex01(filename)
x=imread(filename);
if (size(x,3)>1)%if RGB image make gray scale
    try
        x=rgb2gray(x);%image toolbox dependent
    catch
        x=sum(double(x),3)/3;%if no image toolbox do simple sum
    end
end
x=double(x);%make sure the input is double format
[output,count,m,svec]=facefind(x);%full scan
% imagesc(x), colormap(gray)%show image
% plotbox(output,[],8)%plot the detections as red squares
% plotsize(x,m)%plot minmum and maximum face size to detect as ...
green squares in top left corner
% crop face from imge
% output : matrix with positionsfor found faces
% output(:,1) - [x1 x2 y1 y2].'
faceImg = x(output(3,1):output(4,1),output(1,1):output(2,1));
faceImg = uint8(faceImg);
```

โปรแกรมในส่วนของการคำนวณค่า Correlation Coefficient

เป็นฟังก์ชันที่ใช้คำนวณค่า Correlation Coefficient ซึ่งจะรับค่าเป็นรูปภาพ กับ ภาพใบหน้าที่ได้จัดเก็บไว้ กับค่าจำนวนของแถวและคอลัมน์ เพื่อนำใช้ในการคำนวณ

```
% ************************************************************
%function to find correlation
%@since 2009 February 25
% @ param Y - face image to recognition
% @ param X - face image in database
% @ param [row col] size of face image to recognition
% @ param[row1 col1] size of face image in database
%@note Matlab 7.5.0(R2007b)
%@note Windows XP SP2
% ************************************************************
function r = corr(Y,X,row,col,row1,col1);
% check if size image are equare
if (row1~= row && col1 == col);
    Y = imresize(Y,[row1 col]);
elseif  (row1 == row && col1 ~= col);
    Y = imresize(Y,[row col1]);
else  (row1 ~= row && col1 ~= col);
    Y = imresize(Y,[row1 col1]);
end
% Image enhancement by histogram equalization
X = histeq(X);
Y = histeq(Y);
X = double(X);
Y = double(Y);
meanY = mean(mean(Y));
meanX = mean(mean(X));
sumXY = 0;
sumX = 0;
sumY =0;
for i = 1 :row1
    for j = 1:col1
        x_bar = X(i,j) - meanX;
        y_bar = Y(i,j) - meanY;
        mul_xy = x_bar*y_bar;
        sumXY = sumXY + mul_xy;
    end
end
for i = 1 :row1
    for j = 1:col1
        x_bar = X(i,j) - meanX;
        sumX = sumX + (x_bar^2);
    end
end
for i = 1 :row1
    for j = 1:col1
        y_bar = Y(i,j) - meanY;
        sumY = sumY + (y_bar^2);
    end
end
r = sumXY/sqrt(sumX*sumY);
```

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวอรุณรัตน์ มานะเพียร

ภูมิลำเนา 97/33 หมู่ 1 ตำบล เขาสามยอด อำเภอเมือง จังหวัด ลพบุรี 15000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพิบูลวิทยาลัย
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **engi_amp@hotmail.com**

ชื่อ นายอุดมศักดิ์ อินทรสูต

ภูมิลำเนา 334/2 หมู่ 5 ตำบล นครชุม อำเภอ เมือง จังหวัด กำแพงเพชร 62000

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนกำแพงเพชรพิทยาคม
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **sak_999_5@hotmail.com**