สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

# ระบบสืบค้นข้อมูลโดยใช้รูปภาพ

# IMAGE INQUIRY SYSTEM

T046126

โดย

นางสาวธิดา เด่นพฤกษ์ธรรม

นายอาทิตย์ วจนสมบูรณ์กุล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 46126
วัน, เดือน, ปี 20 มี.ค. 2546

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

b 11288292

# ระบบสืบค้นข้อมูลโดยใช้รูปภาพ

# IMAGE INQUIRY SYSTEM

โดย

นางสาวธิดา เด่นพฤกษ์ธรรม

นายอาทิตย์ วจนสมบูรณ์กุล

อาจารย์ที่ปรึกษา

รศ.ดร.เอื้อน ปิ่นเงิน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา2544

ปริญญานิพนธ์ ปีการศึกษา2544

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสืบค้นข้อมูลโดยใช้รูปภาพ

ผู้จัดทำ

1. นางสาวธิดา เด่นพฤกษ์ธรรม
2. นายอาทิตย์ วจนสมบูรณ์กุล

..... อาจารย์ที่ปรึกษา

(รศ.ดร.เอื้อน ปิ่นเงิน)

ระบบสืบค้นข้อมูลโดยใช้รูปภาพ

IMAGE INQUIRY SYSTEM

ผู้จัดทำ

1. นางสาวธิดา เด่นพฤกษ์ธรรม เลขประจำตัว 41014184
2. นายอาทิตย์ วจนสมบูรณ์กุล เลขประจำตัว 41014539

โครงงานนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

ลงชื่อ ..... อาจารย์ที่ปรึกษา

(รศ.ดร.เอื้อน ปิ่นเงิน)

# ระบบสืบค้นข้อมูลโดยใช้รูปภาพ

นางสาวธิดา เด่นพฤกษ์ธรรม
นายอาทิตย์ วจนสมบูรณ์กุล
รศ.ดร.เอื้อน ปิ่นเงิน (อาจารย์ที่ปรึกษา)
ภาคการศึกษาที่ 2 ปีการศึกษา2544

**บทคัดย่อ**

โครงงานนี้ทำการพัฒนาระบบที่รับรูปภาพใบหน้าคนเป็นอินพุท และจัดเก็บลงสู่ฐานข้อมูลเพื่อใช้ในการสืบค้นและทำการบ่งชี้บุคคล สำหรับการจัดเก็บฐานข้อมูลภาพใบหน้าคนเอาไว้นั้น เมื่อระบบได้รับอินพุทภาพใบหน้าเข้ามา ระบบจะประมวลผลและเปรียบเทียบกับข้อมูลในฐานข้อมูลดังกล่าว เพื่อทำการรู้จำ บ่งชี้ภาพแต่ละภาพได้ เทคนิคที่ใช้ในโครงงาน ได้แก่ การสร้างภาพไบนารี่ การแยกขอบวัตถุที่สนใจออกจากภาพ รวมถึงการใช้โครงข่ายประสาทเทียมเข้าช่วยในส่วนการประมวลผลข้อมูล ซึ่งโครงงานนี้ได้ใช้ภาษาวิชวลเบสิกเป็นภาษาเพื่อใช้เขียนโปรแกรมทั้งในส่วนการทำงาน ส่วนติดต่อกับผู้ใช้ รวมถึงการติดต่อกับระบบจัดการฐานข้อมูลออราเคิลด้วย

# Image Inquiry System

Miss Thida Denpruektham

Mr. ArTit Wajanasombulkoon

Assoc. Prof. Dr. Ouen Pinngern (Advisor)

2nd semester, Educational Year 2001

**Abstract**

In this project, we develop a system capable of accepting human face image as input and collecting into a database for inquiring and identifying purposes. By collecting images in the database , when the system receive input image, it performs a matching process with the database for recognition and identification each individual image. The techniques used in the project consist of image binarization, segmentation and neural network for image processing. We use visual basic to program all applications and interfaces, along with Oracle as a DBMS.

# สารบัญ


| | หน้า |
|---|---|
| บทคัดย่อ | I |
| Abstract | II |
| สารบัญ | III |
| สารบัญรูปภาพ | V |
| สารบัญตาราง | VII |
| บทที่ 1 บทนำ | 1 |
| 1.1 วัตถุประสงค์ของโครงงาน | 1 |
| 1.2 ขอบเขตของโครงงาน | 1 |
| 1.3 การนำไปประยุกต์ใช้ของโครงงาน | 5 |
| 1.4 ประโยชน์และผลที่คาดว่าจะได้รับจากโครงงาน | 5 |
| บทที่ 2 หลักการเบื้องต้นของการประมวลผลภาพ | 6 |
| 2.1 การได้มาซึ่งภาพ | 6 |
| 2.2 กระบวนการประมวลผลภาพ | 6 |
| 2.2.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล | 7 |
| 2.2.2 ระดับสีเทา (Grey Scale) | 7 |
| 2.2.3 การสร้างภาพไบนารี่ | 8 |
| 2.2.3.1 ฮิสโตแกรม (Histogram) | 9 |
| 2.2.3.2 การแปลงระดับเกรย์ (Grey-level Transformation) | 10 |
| 2.3 ผลที่ได้และการแสดงผล (Output and Display) | 11 |
| บทที่ 3 โครงข่ายประสาทเทียม (Neural Network) | 12 |
| 3.1 ทฤษฎีพื้นฐานของโครงข่ายประสาทเทียม | 12 |
| 3.2 ลักษณะของโครงข่ายประสาทเทียม | 13 |
| 3.3 ชั้นของโครงข่ายประสาทเทียม (Layer of Neural Network) | 15 |
| 3.3.1 โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Neural Network) | 16 |
| 3.3.2 โครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Neural Network) | 17 |
| 3.4 การฝึกสอนให้กับโครงข่าย (Training Algorithm) | 17 |

3.4.1 ขั้นตอนการเรียนรู้แบบแพร่ย้อนกลับ (Backpropagation Learning) 18
3.5 Weightless Neural Networks 19
3.5.1 The General Neural Unit (GNU) 22
3.5.2 การทำงานของ GNUs 23
บทที่ 4 หลักการทำงานและการออกแบบระบบ 25
4.1 หลักการทำงาน 25
4.1.1 การปรับค่าความสว่าง (Normalization) 25
4.1.2 การฝึกสอนโครงข่ายประสาทเทียม (Training) 26
4.1.3 การเปรียบเทียบเพื่อการสืบค้นภาพใบหน้าคนของแต่ละบุคคล 27
4.2 การออกแบบระบบ 27
4.2.1 ส่วนของโปรแกรม 27
4.2.1.1 โครงสร้างโดยรวมของโปรแกรม 27
4.2.1.2 Keep into Database 29
4.2.1.3 Search 30
4.2..1.4 Advance Search 31
4.2.2 ส่วนของระบบจัดการฐานข้อมูล 34
บทที่ 5 ผลการทดลอง 36
5.1 ประสิทธิภาพการรู้จำภาพใบหน้าคน 36
5.1.1 ผลของจำนวนข้อมูลในฐานข้อมูล 36
5.1.2 ผลจากภาพใบหน้าอินพุทมีลักษณะแตกต่างจากภาพในฐานข้อมูล 37
5.1.3 เมื่อสภาพแวดล้อมเปลี่ยน 39
บทที่ 6 บทสรุปและข้อเสนอแนะ 40
ภาคผนวก
เอกสารอ้างอิง

# สารบัญรูปภาพ

หน้า

รูปที่ 1.1 ขั้นตอนการทำงานรวมของโครงงาน 2
รูปที่ 1.2 ขั้นตอนการทำงานในส่วนการรับภาพใบหน้าอินพุท 3
รูปที่ 1.3 ขั้นตอนการทำงานในส่วนการรับภาพใบหน้าอินพุทพร้อมทั้งข้อมูลที่ต้องการเก็บ 3
รูปที่ 1.4 ขั้นตอนการทำงานในส่วนการทำการประมวลผลรูปภาพที่เข้ามา 4
รูปที่ 1.5 ขั้นตอนการทำงานในส่วนการทำการฝึกสอนให้กับโครงข่ายประสาทเทียม 4
รูปที่ 1.6 ขั้นตอนการทำงานในส่วนการทำการเก็บลงฐานข้อมูล 5
รูปที่ 2.1 ตัวอย่างฮิสโตแกรมของ 8 ระดับเกรย์ 9
รูปที่ 2.2 (a) ฮิสโตแกรมจากภาพต้นฉบับ 10
(b) ฮิสโตแกรมของภาพหลังจากเพิ่มความสว่างเข้าไปอีก 3 ให้กับทุกจุดภาพ 10
รูปที่ 3.1 ลักษณะของนิวรอลเซลล์ของเซลล์ประสาท 13
รูปที่ 3.2 นิวรอลของโครงข่ายประสาทเทียม 14
รูปที่ 3.3 สมการเชิงเส้นอย่างง่าย 14
รูปที่ 3.4 สมการเทรชโฮล 15
รูปที่ 3.5 ซิกมอยด์ฟังก์ชัน (Sigmoid function) 15
รูปที่ 3.6 โครงสร้างชั้นของโครงข่ายประสาทเทียม 16
รูปที่ 3.7 โครงข่ายประสาทเทียมแบบชั้นเดียว 17
รูปที่ 3.8 โครงข่ายประสาทเทียมแบบหลายชั้น 17
รูปที่ 3.9 ไดอะแกรมของ Backpropagation Neural Network แบบ 2 ชั้น 18
รูปที่ 3.10 (a) A RAM-based neuron 20
(b) 4 RAMs discriminators with 3 inputs each 20
รูปที่ 3.11 A schematic view of a basic GNU 22
รูปที่ 4.1 บล็อคไดอะแกรมแสดงการปรับค่าความสว่างของภาพ 25
รูปที่ 4.2 วิธีการฝึกสอนโครงข่ายประสาทเทียมจากภาพอินพุท 26
รูปที่ 4.3 ขั้นตอนการทำงานแสดงส่วนการฝึกสอนให้กับโครงข่ายประสาทเทียม 26
รูปที่ 4.4 แสดงการฝึกสอน 27
รูปที่ 4.5 ลำดับขั้นตอนการทำงานของโปรแกรมทั้งหมด 28

หน้า

รูปที่ 4.6 หน้าจอหลักของโปรแกรม 28
รูปที่ 4.7 ลำดับขั้นตอนการทำงานของโปรแกรมส่วนการเก็บรูปภาพลงสู่ฐานข้อมูล 29
รูปที่ 4.8 หน้าจอการกรอกข้อมูลเพื่อเก็บลงฐานข้อมูล 30
รูปที่ 4.9 ลำดับขั้นตอนการทำงานของโปรแกรมส่วนการค้นหาข้อมูลโดยใช้รูปภาพเพียงอย่างเดียว 31
รูปที่ 4.10 ลำดับขั้นตอนการทำงานของโปรแกรมส่วนการค้นหาข้อมูลโดยใช้ข้อมูลหลายอย่าง 32
รูปที่ 4.11 หน้าจอรับข้อมูลเพิ่มเติมเพื่อช่วยในการค้นหา 33
รูปที่ 4.12 หน้าจอแสดงผลข้อมูลที่ได้ 33
รูปที่ 4.13 ชนิดการจัดเก็บข้อมูลทั้งหมดใน Oracle 34
รูปที่ 5.1 แสดงการรู้จำภาพใบหน้าอินพุทที่เอียงแตกต่างจากฐานข้อมูล 37
รูปที่ 5.2 แสดงการรู้จำภาพใบหน้าอินพุทที่ยิ้มแตกต่างจากฐานข้อมูล 38
รูปที่ 5.3 แสดงการรู้จำภาพใบหน้าอินพุทที่มีแสงแตกต่างจากฐานข้อมูล 39

# สารบัญตาราง

หน้า

ตารางที่ 3.1 Main parameters of a GNU 22
ตารางที่ 5.1 ผลของจำนวนข้อมูลในฐานข้อมูลที่ไม่เท่ากัน 36
ตารางที่ 5.2 ผลจากภาพใบหน้าอินพุทที่เอียงแตกต่างจากภาพในฐานข้อมูล 37
ตารางที่ 5.3 ผลจากภาพใบหน้าอินพุทที่ยิ้มแตกต่างจากภาพในฐานข้อมูล 38
ตารางที่ 5.4 ผลจากภาพเมื่อมีความสว่างต่างจากภาพในฐานข้อมูล 39

# บทที่ 1
# บทนำ

ปัจจุบันนี้เป็นยุคแห่งข้อมูลข่าวสาร ความต้องการบริโภคข้อมูลข่าวสารจึงมีอยู่ทั่วไปและยังมีแนวโน้มที่จะสูงขึ้นอีกมากในอนาคตอันใกล้ ข้อมูลรูปภาพเป็นรูปแบบหนึ่งของข้อมูลที่เราสามารถพบและได้ใช้ในชีวิตประจำวันเป็นอย่างมาก และในบางโอกาสเราอาจมีความจำเป็นที่จะต้องการค้นหาข้อมูลโดยการใช้รูปภาพเป็นสื่อในการค้นหาแทนการใช้คำสำคัญ (keyword) ที่เป็นตัวอักษร เช่น การสืบค้นรูปใบหน้าคน ตัวอย่างการใช้งานเช่น การสืบค้นข้อมูลผู้ต้องสงสัย ของกรมตำรวจ เป็นต้น หรือเพื่อใช้ในการศึกษาหรือเพื่อความบันเทิงต่าง ๆ เช่น การสืบค้นข้อมูลรายละเอียดของสิ่งต่าง ๆ ในฐานข้อมูลที่มีอยู่, การสืบค้นข้อมูลหรือแกลเลอรี่ภาพของบุคคลที่มีชื่อเสียงในวงการต่าง ๆ เป็นต้น

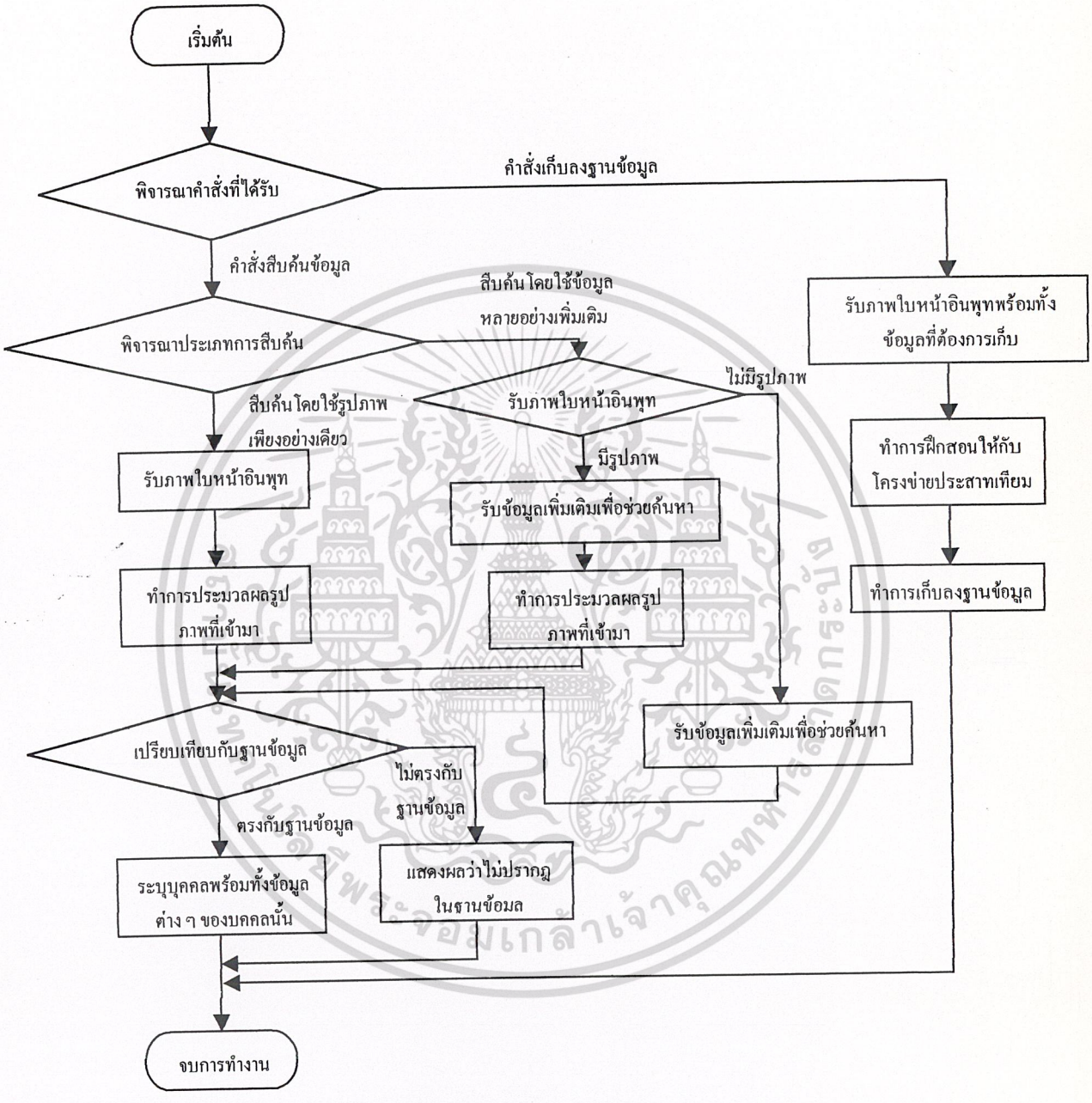
ถึงแม้ว่าปัจจุบันนี้คอมพิวเตอร์จะเข้ามามีบทบาทต่อชีวิตคนเรามากขึ้น สามารถช่วยเหลืองานที่ซับซ้อนยุ่งยากได้อย่างรวดเร็ว และช่วยให้มนุษย์มีความสะดวกสบายมากขึ้นกว่าแต่ก่อน แต่คอมพิวเตอร์ในปัจจุบันนี้ยังไม่สามารถช่วยในงานที่มีลักษณะที่ต้องอาศัยความฉลาดของมนุษย์ในการตัดสินใจได้ดีนัก ซึ่งมีความไม่แน่นอนของการแก้ปัญหาเข้ามาเกี่ยวข้องด้วย เทคโนโลยีทางด้านโครงข่ายประสาทเทียมหรือนิวรอลเน็ตเวิร์คเป็นเทคโนโลยีหนึ่งทางด้านปัญญาประดิษฐ์ (Artificial Intelligent) จึงถูกพัฒนาขึ้น โครงข่ายประสาทเทียมนี้เป็นเทคโนโลยีที่อาศัยรูปแบบการทำงานและโครงสร้างของสมองมนุษย์มาสร้างเครื่องคอมพิวเตอร์ที่มีความรู้อยู่ภายในตัวของมันเอง มีความสามารถในการเรียนรู้ต่อความรู้ที่เข้ามาใหม่ และมีความสามารถในการแก้ปัญหาที่ต้องอาศัยการตัดสินใจได้สูง สำหรับโครงงานนี้ก็ได้นำวิธีการทางโครงข่ายประสาทเทียมเข้ามาช่วยเหลือในขั้นตอนการประมวลผลให้มีความถูกต้องและรวดเร็วมากขึ้น

## 1.1 วัตถุประสงค์ของโครงงาน

เพื่อสร้างโปรแกรมใช้งานที่สามารถประมวลผลภาพเพื่อเก็บเป็นข้อมูลลงสู่ฐานข้อมูล และสามารถบ่งชี้ข้อมูลที่อยู่ในฐานข้อมูลได้

## 1.2 ขอบเขตของโครงงาน

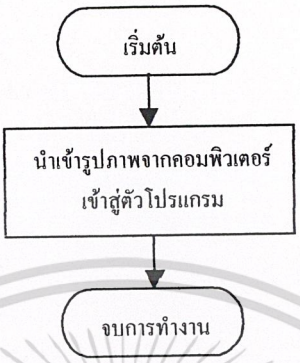
โครงงานนี้จะทำการรับข้อมูลรูปภาพใบหน้าคนเป็นอินพุทเข้ามาจากสแกนเนอร์หรือกล้องดิจิตอล โดยชนิดของไฟล์รูปภาพที่รับเข้ามาสามารถเป็นได้ทุกนามสกุล โดยจะทำการรู้จำและเก็บลงฐานข้อมูล หรือเพื่อทำการบ่งชี้บุคคล สำหรับขั้นตอนการทำงานของโครงงานมีดังนี้




รูปที่ 1.1 ขั้นตอนการทำงานรวมของระบบ

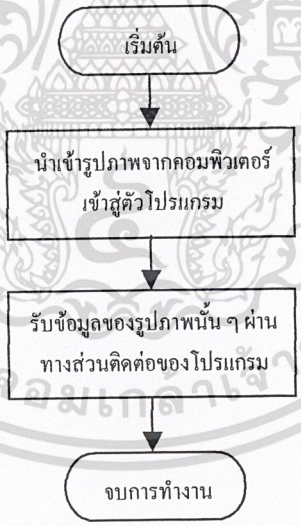
**หลักการทำงานในแต่ละส่วน**

- การรับภาพใบหน้าอินพุท



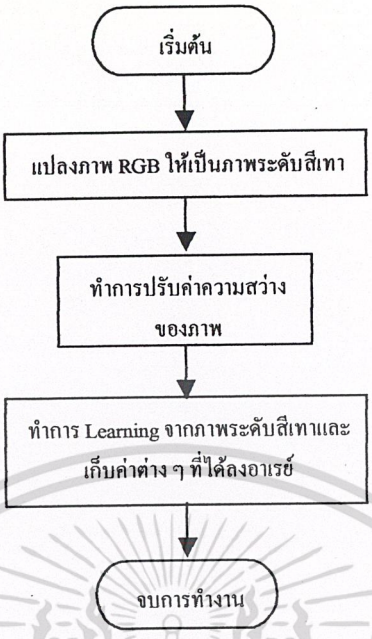

รูปที่ 1.2 ขั้นตอนการทำงานในส่วนการรับภาพใบหน้าอินพุท

- การรับภาพใบหน้าอินพุทพร้อมทั้งข้อมูลที่ต้องการเก็บ



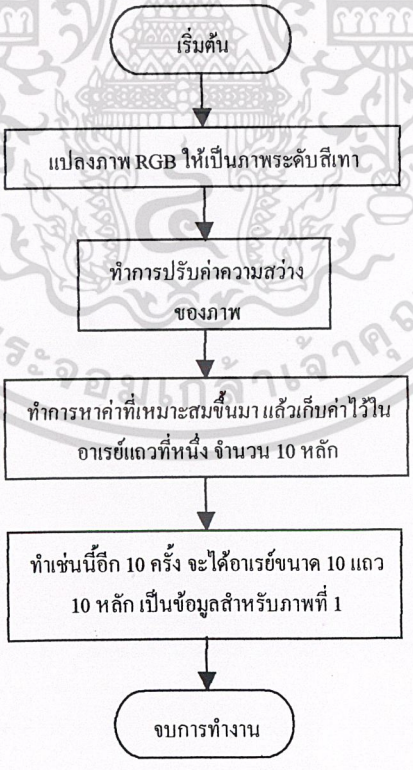

รูปที่ 1.3 ขั้นตอนการทำงานในส่วนการรับภาพใบหน้าอินพุทพร้อมทั้งข้อมูลที่ต้องการเก็บ

- ทำการประมวลผลรูปภาพที่เข้ามา



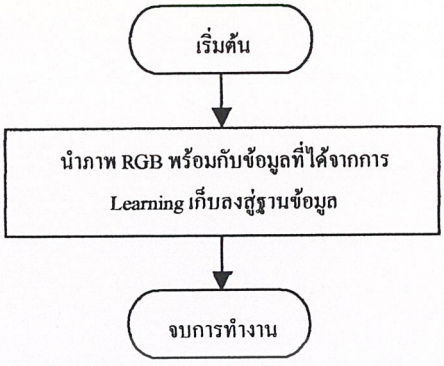

รูปที่ 1.4 ขั้นตอนการทำงานในส่วนการทำการประมวลผลรูปภาพที่เข้ามา

- ทำการฝึกสอนให้กับ โครงข่ายประสาทเทียม




รูปที่ 1.5 ขั้นตอนการทำงานในส่วนการทำการฝึกสอนให้กับ โครงข่ายประสาทเทียม

- การทำการเก็บลงฐานข้อมูล




รูปที่ 1.6 ขั้นตอนการทำงานในส่วนการทำการเก็บลงฐานข้อมูล

**1.3 การนำไปประยุกต์ใช้ของโครงงาน**

1. การสืบค้นข้อมูลจากฐานข้อมูล เพื่อใช้ในเชิงการศึกษา หรือเพื่อความบันเทิง

**1.4 ประโยชน์และผลที่คาดว่าจะได้รับจากโครงงาน**

1. ได้ทราบถึงหลักการประมวลผลภาพโดยใช้เทคโนโลยีโครงข่ายประสาทเทียมเข้าช่วย
2. ได้ศึกษาถึงรายละเอียดการทำงานของระบบจัดการฐานข้อมูลที่ใช้
3. ได้ศึกษาการเขียนโปรแกรมประยุกต์ใช้งานด้วยวิชวลเบสิก (Visual Basic)

# บทที่ 2
# หลักการเบื้องต้นของการประมวลผลภาพ

กลไกระบบการมองเห็นภาพนั้น จะหมายความรวมถึงทุกสิ่งที่จำเป็นต่อการได้มาซึ่งรหัสทางดิจิตอลที่ใช้แทนภาพนั้น ๆ การปรับปรุงเปลี่ยนแปลงข้อมูล และรวมทั้งการนำเสนอภาพที่ได้หลังการปรับปรุงแล้วด้วย ซึ่งระบบการมองเห็นภาพ (Vision System) นี้อาจมีความยุ่งยากซับซ้อนต่างกันตามลักษณะการใช้งาน แต่ก็สามารถแบ่งเป็นขั้นตอนกว้าง ๆ ได้ 3 ขั้นตอน ดังนี้

2.1) การได้มาซึ่งภาพ (Image Acquistion)

2.2) กระบวนการประมวลผลภาพ (Image Processing)

2.3) ผลที่ได้และการแสดงผล (Output and Display)

## 2.1 การได้มาซึ่งภาพ (Image Acquistion)

การได้มาซึ่งภาพนี้ จะหมายถึงการแปลงภาพที่เราเห็นในลักษณะทางกายภาพนั้นให้เป็นเซตของข้อมูลทางดิจิตอล โดยในโครงงานนี้จะรับรูปภาพผ่านทางสแกนเนอร์ ทำการแปลงให้เป็นภาพระดับสีเทา และเก็บค่าความสว่างของแต่ละพิกเซลอยู่ในรูปแบบเลขฐานสองจำนวน 8 บิต ซึ่งเซตของพิกเซลเหล่านี้จะถูกส่งไปยังส่วนประมวลผลภาพต่อไป

## 2.2 กระบวนการประมวลผลภาพ (Image Processing)

หน้าที่พื้นฐานของกระบวนการประมวลผลภาพ ได้แก่ การสร้างภาพใหม่โดยแยกแยะส่วนที่เราต้องการหรือสนใจกับสิ่งรบกวนออกจากกัน (Noise elimination), การหาขอบภาพ (Edge enhancement), การกรอง (Filtering), การปรับปรุงเปลี่ยนแปลงค่าระดับเกรย์ (Grey scale modification)

การประมวลผลภาพเชิงตัวเลข (Digital Image Processing) หมายถึง การนำภาพมาประมวลผลด้วยเครื่องคอมพิวเตอร์ โดยภาพที่นำมาประมวลผลนี้จะถูกแทนที่ด้วยตัวเลขให้อยู่ในรูปแบบของเมตริกซ์ แต่ภาพที่ได้โดยส่วนมากแล้วจะเป็นภาพที่ได้จากตัวรับสัญญาณ ซึ่งอยู่ในรูปของฟังก์ชัน f(x, y) ที่ต่อเนื่องในระนาบสองมิติ (คือแกน x และแกน y) โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพ ที่ตำแหน่ง (x, y) ซึ่งเรียกว่า ระดับสีเทา (Grey Level)

**2.2.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล**

ภาพข้อมูลแบบดิจิตอล (Digital Image) เป็นภาพที่ถูกแปลงมาจากภาพอนาลอกให้อยู่ในรูปของตัวเลข โดยการนำภาพอนาล็อกมาแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า พิกเซล (pixel) ในแต่ละพิกเซลจะถูกระบุตำแหน่งโดย (x, y) และค่าระดับสีเทาของพิกเซล โดยเราสามารถแปลงภาพเป็นข้อมูลแบบดิจิตอลได้ โดยมีขั้นตอนและวิธีการดังนี้

เมื่อเรานำสัญญาณอนาล็อกที่ต้องการประมวลผลมาผ่านส่วนที่เรียกว่า ดิจิไตเซอร์ (Digitizer) ซึ่งจะมีหน้าที่ในการเปลี่ยนสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอล จากนั้นทำการควอนไตซ์ (Quantizing) เพื่อที่จะประมวลสัญญาณด้วยระบบคอมพิวเตอร์ ฟังก์ชันของภาพ f(x, y) จะถูกทำให้เป็นสัญญาณไม่ต่อเนื่องทั้งระนาบของภาพ ซึ่งเราเรียกว่าการสุ่มภาพ (Image Sampling) ของฟังก์ชันที่ได้เรียกว่า การควอนไตเซชันระดับสีเทา (Grey level quantization) ก็จะได้ข้อมูลที่เป็นดิจิตอล

**2.2.2 ระดับสีเทา (Grey Scale)**

หากเราต้องการค่าข้อมูลที่ละเอียดมากขึ้น ก็จำเป็นที่จะต้องเพิ่มจำนวนบิตการแสดงค่าของแต่ละพิกเซล ยกตัวอย่างเช่น หากแบ่งความเข้มของการส่องสว่างให้มี 4 ระดับ ก็ต้องใช้เลขฐานสองจำนวน 2 บิต และจำนวน 4 บิตสำหรับ 16 ระดับ, 8 บิตสำหรับ 256 ระดับ ซึ่งจำนวนระดับที่ใช้ในระดับสีเทาหรือระดับเกรย์นี้มักเป็นเลขยกกำลังของ 2, ค่าที่ต่ำที่สุดคือ 0 กำหนดให้เป็นสีดำ และ 1 หรือตัวเลขที่น้อยกว่าค่าสูงสุดของระดับเกรย์อยู่ 1 (เช่น 15 สำหรับระดับเกรย์ 16 ระดับ) แทนสีขาว ค่าที่กำหนดให้ในแต่ละพิกเซลนี้นิยมใช้เป็นจำนวนเต็ม

ในยุคแรก ๆ ของระบบการมองเห็นภาพจะใช้ระบบเลขฐานสอง แต่ในปัจจุบันเทคโนโลยีไมโครโปรเซสเซอร์เข้ามามีบทบาทมากขึ้น การแบ่งระดับเป็น 16, 64 หรือ 256 เป็นเรื่องธรรมดา แต่ทั้งนี้ในการมองเห็นของมนุษย์จะสามารถแยกแยะความแตกต่างได้เพียง 10-15 ระดับเท่านั้น การแบ่งโดยละเอียดเป็น 64 หรือ 256 ระดับอาจนำไปประยุกต์ใช้กับงานการประมวลผลภาพหรือกระบวนการทางอุตสาหกรรมอื่น ๆ

ซึ่งจะเห็นว่าจำนวนระดับเกรย์จะเป็นตัวจำกัดรายละเอียดของภาพ โดยทั่วไปแล้วยิ่งแบ่งระดับเกรย์เป็นหลายระดับก็จะเป็นการเพิ่มคุณภาพของภาพด้วย และการเพิ่มจำนวนพิกเซล เช่น จาก 35 * 35 เป็น 250 * 250 ก็จะเป็นการเพิ่มความละเอียด (Resolution) และรายละเอียด (Detail) ของภาพเช่นกัน จะเห็นว่าจะแตกต่างกับการขยายภาพ (Zoom) โดยการขยายภาพนี้เป็นการขยายแต่ละพิกเซลให้ใหญ่ขึ้น ไม่ได้เป็นการเพิ่มจำนวนพิกเซล

**2.2.3 การสร้างภาพไบนารี่**

อุปกรณ์ที่มีความสามารถในการแสดงผลได้แค่ 2 ระดับ หรือ 2 สี คือ สีขาวกับสีดำยังมีการใช้กันอย่างแพร่หลาย เช่น เครื่องพิมพ์ (Printer), เครื่องโทรสาร (Fax), จอภาพแสดงผลแบบโมโนโครม (Monochrome Monitor) เป็นต้น เนื่องจากอุปกรณ์เหล่านี้เป็นอุปกรณ์ที่มีราคาถูก ดังนั้นจึงเกิดปัญหาในการแสดงผลขึ้น การที่จะแสดงผลหรือพิมพ์รูปภาพที่มีระดับความเข้มของภาพหลายระดับ ซึ่งมีมากกว่าความสามารถในการแสดงผลของอุปกรณ์เหล่านั้นที่มีเพียงแค่ 2 ระดับ เราจะต้องทำการแปลงข้อมูลภาพให้เป็นภาพไบนารี่ (Binary Image) เสียก่อน ซึ่งการสร้างภาพไบนารี่นั่นก็หมายถึงการแปลงข้อมูลภาพที่มีระดับความเข้มหลายระดับ (Multilevel Image) ให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ นั่นคือ 1 จุดภาพ มีได้ 2 ค่าเท่านั้น คือ 0 กับ 1 โดยจุดภาพที่แทนด้วย 0 หมายถึงจุดภาพที่มีสีดำ ส่วนจุดที่แทนด้วย 1 จะหมายถึงจุดภาพที่มีสีขาว เมื่อทำการแปลงเป็นภาพไบนารี่แล้ว จึงสามารถนำภาพนั้นไปแสดงผลบนอุปกรณ์เหล่านั้น จะเห็นได้ว่าการแปลงข้อมูลภาพหลายระดับเป็นภาพไบนารี่จึงมีความจำเป็นและมีประโยชน์มากในการแสดงผลภาพที่มีความเข้มของภาพหลายระดับบนอุปกรณ์ที่มีความสามารถในการแสดงผลได้ 2 ระดับ สำหรับประโยชน์อีกประการหนึ่งในการแปลงข้อมูลภาพนั้นเป็นภาพไบนารี่ คือการลดเนื้อที่การเก็บข้อมูลภาพ ภาพไบนารี่นี้จะสามารถลดเนื้อที่ลงได้ถึง 8 เท่า นั่นคือถ้าเดิม 1 จุดภาพใช้เนื้อที่ในการเก็บ 8 บิต เมื่อทำการแปลงเป็นภาพไบนารี่แล้วจะใช้เนื้อที่ในการเก็บ 1 จุดภาพนั้นเพียง 1 บิตเท่านั้น อีกทั้งยังสามารถนำไปประยุกต์ใช้งานได้อย่างแพร่หลาย เช่น นำไปประยุกต์ใช้ในการวิคราะห์เอกสารในขั้นตอนที่เรียกว่าการประมวลผลขั้นต้น (Preprocessing) เป็นต้น

ในการสร้างภาพไบนารี่ สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล (Thresholding Technique) โดยพิจารณาว่าจุดภาพใดควรจะเป็นจุดขาวหรือจุดดำ จะกระทำโดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งที่เรียกว่า ค่าเทรชโฮล (Threshold Value) เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุ (Object) และพื้นหลัง (Background) โดยค่าของจุดภาพใด ๆ ที่มีค่าน้อยกว่าค่าเทรชโฮลจะถูกกำหนดให้เป็น 0 (จุดดำ) และถ้าค่าของจุดภาพใด ๆ ที่มีค่ามากกว่าหรือเท่ากับค่าเทรชโฮล จะถูกเปลี่ยนให้เป็น 1 (จุดขาว)

ในการสร้างภาพไบนารี่โดยใช้เทคนิคเทรชโฮลเพื่อให้ได้ผลลัพธ์ที่เหมาะสมและคมชัด สิ่งที่สำคัญที่สุดคือ ค่าเทรชโฮล เนื่องจากถ้าเลือกค่าเทรชโฮลที่ไม่เหมาะสม (ค่าเทรชโฮลที่มีค่าน้อยเกินไปหรือมากเกินไป) ภาพที่ได้อาจจะไม่เหมาะสม ขาดความคมชัด และอาจทำให้รายละเอียดบางส่วนขาดหายไป กล่าวคือภาพที่ได้อาจจะมืดเกินไปหรือสว่างเกินไป หรือภาพที่ได้มีสิ่งรบกวน (Noise) เกิดขึ้น อันเป็นผลทำให้ภาพผลลัพธ์ที่ได้ไม่สวยงามเท่าที่ควร ดังนั้นปัญหาของการสร้างภาพไบนารี่โดยวิธีเทรชโฮลนี้คือ ทำอย่างไรจึงจะสามารถคำนวณหาค่าเทรชโฮลที่เหมาะสมสำหรับ

แต่ละภาพที่จะนำมาทำการสร้างภาพไบนารี่ ซึ่งมีวิธีคำนวณหาค่าเทรชโฮลหลายวิธี โดยแต่ละวิธีเหมาะสมกับลักษณะการทำงานที่แตกต่างกันไป เช่น การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า (Pre-assigned Threshold Value) การหาค่าเทรชโฮลจากค่ากลาง (Mid-range Threshold Value) ซึ่งแต่ละวิธีอธิบายได้ดังนี้

- **การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า (Pre-assigned Threshold Value)**

การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้านี้เป็นวิธีที่ง่ายที่สุด เป็นการคำนวณค่าเทรชโฮลโดยการกำหนดเองจากผู้ใช้ ซึ่งการกำหนดนี้จะขึ้นอยู่กับประสบการณ์ของผู้ใช้นั้น ๆ ทำได้โดยการเลือกค่าคงที่ค่าหนึ่ง ซึ่งเรียกค่านั้นว่า “ค่าเทรชโฮล” โดยค่าที่เลือกมานี้จะเป็นค่าที่อยู่ระหว่างค่าต่ำสุดและค่าสูงสุดของระดับความเข้มของข้อมูลภาพอินพุท เช่น ภาพข้อมูลอินพุทมีเกรย์สเกล 256 ระดับ จะมีค่าเกรย์สเกลได้ตั้งแต่ 0-255

- **การหาค่าเทรชโฮลจากค่ากลาง (Mid-Range Threshold Value)**

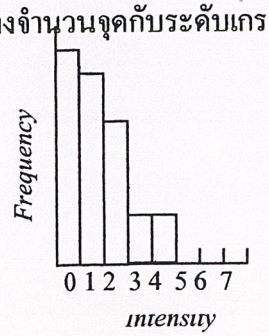
การหาค่าเทรชโฮลโดยพิจารณาจากค่ากลาง เป็นการหาค่าเทรชโฮลที่แตกต่างจากการหาค่าเทรชโฮลวิธีแรก สำหรับวิธีนี้จะเป็นการคำนวณหาค่าเทรชโฮลโดยอัตโนมัติโดยผู้ใช้ไม่ต้องเป็นผู้กำหนดเอง การคำนวณหาค่าได้อาศัยการคำนวณพื้นฐานทางสถิติในเรื่องของการหาค่ากลาง หรือค่าเฉลี่ย (Mean) มาประยุกต์ใช้ ซึ่งค่าเทรชโฮลที่คำนวณได้จะเป็นค่าที่เหมาะสมสำหรับภาพนั้น ๆ

**2.2.3.1 ฮิสโตแกรม (Histogram)**

ฮิสโตแกรมเป็นการแสดงให้เห็นถึงความถี่ของการนับจำนวนพิกเซล ที่มีค่าความเข้มแต่ละค่าหนึ่ง ๆ ในภาพระดับเกรย์ โดยแกน x ในกราฟแสดงค่าระดับเกรย์ และแกน y แสดงค่าจำนวนพิกเซล

ขั้นตอนการสร้างฮิสโตแกรม มีดังนี้

1. ทำการดิจิไตซ์ (Digitizing) ภาพ
2. นับจุดพิกเซลในแต่ละระดับเกรย์
3. พลอตกราฟระหว่างจำนวนจุดกับระดับเกรย์




รูปที่ 2.1 ตัวอย่างฮิสโตแกรมของ 8 ระดับเกรย์

รูปร่างของฮิสโตแกรมสามารถบอกลักษณะบางประการของภาพได้ เช่นหากฮิสโตแกรมมีลักษณะที่แคบ จะหมายถึงภาพนั้นขาดการแยกแยะความผิดแผกของสี (Contrast) เป็นต้น ฮิสโตแกรมนี้มีประโยชน์ในการกำหนดค่าเทรชโฮล เพื่อสำหรับทำการเปลี่ยนภาพระดับเกรย์ให้เป็นภาพไบนารี่

**2.2.3.2 การแปลงระดับเกรย์ (Grey-level Transformation)**

ประโยชน์ของการแปลงระดับเกรย์

1. การเปลี่ยนแปลงความสว่าง (Global alterations in brightness)

ทำได้โดยการใช้ค่าคงที่บวกหรือลบออกจากทุกพิกเซลของภาพ เพื่อเพิ่มหรือลดความสว่างของภาพ

รูปที่ 2.2 (a) ฮิสโตแกรมจากภาพต้นฉบับ

(b) ฮิสโตแกรมของภาพหลังจากเพิ่มความสว่างเข้าไปอีก 3 ให้กับทุกจุดภาพ

2. การทำเทรชโฮล (Threshold)

เป็นการเปลี่ยนแปลงหรือหาแนวโน้มของค่าระดับเกรย์ในภาพเพื่อทำให้ค่าระดับเกรย์เกิดความไม่ต่อเนื่อง (discrete) มากขึ้น โดยจะนำภาพมาทำฮิสโตแกรม แล้วกำหนดค่าระดับเกรย์ที่แน่นอนขึ้นมาเพื่อที่จะเป็นจุดตัดในการทำเทรชโฮล จากนั้นจะทำการตัดหรือปิดส่วนของระดับเกรย์ที่เราไม่ต้องการออกไป

3. บันช์ชิ่ง (Bunching) และการควอนไตซ์ (Quantize)

ใช้เพื่อลดระดับเกรย์ที่ไม่ต้องการในภาพลง ในบางครั้งบันช์ชิ่งจะมีความหมายรวมถึงการ ควอนไตซ์ด้วย

4. สปลิตติ้ง (Splitting)

เป็นการเพิ่มความแตกต่างระหว่าง 2 กลุ่มของระดับเกรย์ ยกตัวอย่างเช่น หากเรามีตัวอักษรเขียนอยู่บนฉาก โดยตัวอักษรมีระดับเกรย์ที่ 98 และฉากมีระดับเกรย์ที่ 99 ซึ่งตา

ของมนุษย์ไม่สามารถแยกแยะความแตกต่างเพียงเท่านี้ได้, ฮิสโตแกรมที่ได้ก็จะมีลักษณะแบน ดังนั้นจึงแก้ปัญหาด้วยการทำการสปลิตฮิสโตแกรม โดยทำการดึงค่า 99 ขึ้นเป็น 120 และดึงค่า 98 ลงมาเป็น 80 ซึ่งก็จะทำให้ระหว่างตัวเลขและตัวอักษรมีความแตกต่างกันมากขึ้นจนสามารถสังเกตแยกแยะได้

เทคนิคนี้จะมีประโยชน์มากในกรณีที่เราต้องการดึงเฉพาะบางส่วนของภาพที่มีความสว่างใกล้เคียงกันมากออกมา

**2.3 ผลที่ได้และการแสดงผล (Output and Display)**

รูปแบบของการแสดงผลที่ได้นั้นขึ้นอยู่กับว่าเราจะนำข้อมูลที่ได้ไปใช้ต่อในลักษณะใด โดยอาจแสดงผลทางเครื่องพิมพ์ หรือแสดงผลผ่านทางหน้าจอ เป็นต้น

# บทที่ 3
# โครงข่ายประสาทเทียม (Neural Network)

โครงข่ายประสาทเทียม หรือนิวรอลเน็ตเวอร์ค (Neural Network) เป็นความพยายามอีกอย่างหนึ่งของมนุษย์ในการที่จะลอกเลียนแบบธรรมชาติ โดยทั่วไปแล้วนิวรอลเน็ตเวอร์คจะถูกประกอบขึ้นจากหน่วยย่อยที่มีหน้าที่การทำงานคล้ายคลึงกัลเซลล์สมองของมนุษย์ จากนั้นหน่วยย่อยเหล่านี้จะถูกจัดเชื่อมต่อกันเป็นโครงข่ายในลักษณะที่สัมพันธ์กับกายวิภาคทางสมองหรือบางครั้งอาจจะไม่ก็ได้ โดยการเชื่อมต่อกันในโครงข่ายนี้ทำให้โครงข่ายประสาทเทียมสามารถแสดงพฤติกรรมมากมายที่มีการตอบสนองคล้ายคลึงกับระบบประสาทในสิ่งมีชีวิต ยกตัวอย่างเช่น ความสามารถในการที่จะเรียนรู้จากประสบการณ์, การจัดกลุ่มของข่าวสารจากตัวอย่างที่ให้หรือจับลักษณะเด่นของอินพุทที่ประกอบจากข้อมูลที่ไม่สมบูรณ์ เป็นต้น

จุดอ่อนของโครงข่ายประสาทเทียมมีอยู่เช่นเดียวกับสมองมนุษย์ นั่นคือขีดจำกัดในการคำนวณตัวเลข ในขณะที่โครงข่ายประสาทเทียมนี้มีความชำนาญพิเศษในการประมวลผลข้อมูลที่ไม่สมบูรณ์, การเรียนรู้จากประสบการณ์ ฯลฯ แต่ในข้อมูลปกติทั่วไปการใช้คอมพิวเตอร์ธรรมดาจะเหมาะสมกว่า ดังนั้นการจะนำโครงข่ายประสาทเทียมไปทำการคำนวณบัญชีเงินเดือน, บัญชีลูกหนี้ ฯลฯ จึงไม่อยู่ในวิสัยที่ควรจะทำ หรือกล่าวอีกอย่างหนึ่งก็คือ แม้วิชาการทางโครงข่ายประสาทเทียมนี้จะถูกพัฒนาก้าวไปไกลเท่าไรก็ตามก็ไม่สามารถเข้ามาแทนที่คอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปแล้วได้ หากแต่จะเข้ามาช่วยเหลือในงานส่วนที่คอมพิวเตอร์ไม่สะดวกเท่านั้น

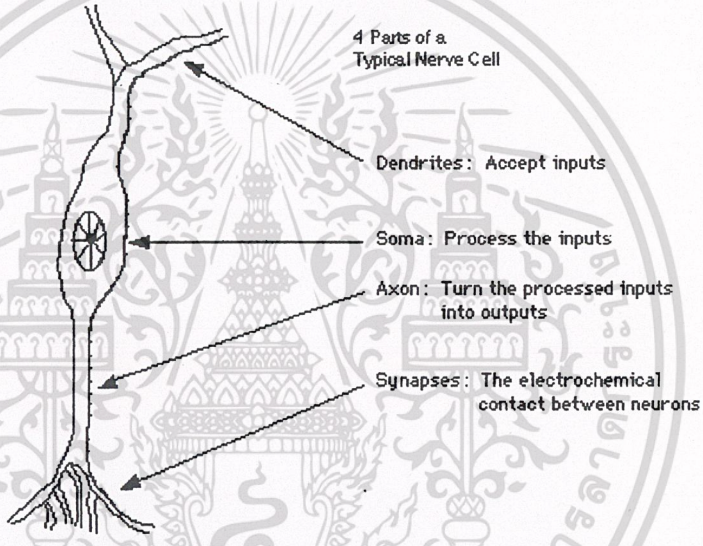
ความแตกต่างระหว่างคอมพิวเตอร์ธรรมดากับโครงข่ายประสาทเทียมอีกอย่างหนึ่งคือ ในคอมพิวเตอร์ธรรมดา เมื่อต้องการให้ทำงานอย่างใดอย่างหนึ่งสามารถทำได้โดยการสร้างโปรแกรม แต่ในทางตรงข้ามโครงข่ายประสาทเทียม เมื่อต้องการให้ทำงานอย่างใดอย่างหนึ่งสามารถทำได้โดยการสอน (Training) โดยการฝึกสอนนี้จะทำให้โครงข่ายประสาทเทียมนำบทเรียนที่ได้จากการฝึกสอนเข้าไปเก็บ และจากนั้นจะถูกเรียกขึ้นมาใช้เมื่ออยู่ในระหว่างประมวลผลข้อมูล

## 3.1 ทฤษฎีพื้นฐานของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมเป็นแนวความคิดหนึ่งทางด้าน AI (Artificial Intelligent) ในการพยายามสร้างเครื่องจักรที่จะสามารถเรียนรู้และจดจำ ซึ่งมีกระบวนการคล้ายคลึงกับกระบวนการทางสมองของมนุษย์

**3.2 ลักษณะของโครงข่ายประสาทเทียม**

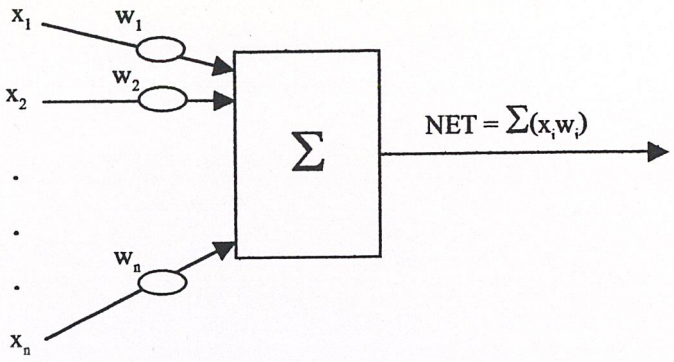
โครงข่ายประสาทเทียม มีลักษณะเป็นการจำลองข้อมูลทางชีววิทยาทางโครงสร้างสมองของมนุษย์ โดยการจำลองเอามาเพียงบางส่วนที่จำเป็นมาใช้ผสมกับรูปแบบการจัดโครงสร้างที่ถูกคิดขึ้น เพื่อทำให้เกิดฟังก์ชัน (Function) ที่ต้องการ โดยทั่วไปแล้วลักษณะโครงสร้างของโครงข่ายประสาทเทียมมีอยู่หลายรูปแบบ แต่ลักษณะโครงสร้างพื้นฐานจะเหมือนกัน กล่าวคือลักษณะทางโครงสร้างพื้นฐานจะประกอบไปด้วยนิวรอล ซึ่งเป็นเซลล์ทางสมองของมนุษย์ ซึ่งโครงสร้างของนิวรอลจำลองแบบมา ลักษณะโดยทั่วไปของนิวรอลเซลล์มีลักษณะดังรูป




**รูปที่ 3.1 ลักษณะของนิวรอลเซลล์ของเซลล์ประสาท**

กล่าวคือจะประกอบไปด้วยส่วนของเดนไดร์ท (Dendrites) ซึ่งยื่นออกมาจากเซลล์ร่างกายไปเชื่อมต่อยังเซลล์อื่น ๆ เพื่อรับสัญญาณ สัญญาณอินพุทที่รับเข้ามานั้นจะถูกประมวลผลโดยส่วนที่เรียกว่าโซมา (Soma) ส่วนสัญญาณเอาท์พุทที่ได้นั้นจะถูกส่งออกทางเซลล์แอกซอน (Axon) และจุดเชื่อมต่อกับเซลล์ประสาทอื่น ๆ เรียกว่า ไซแนปส์ (Synapses)

สำหรับลักษณะของโครงข่ายประสาทเทียม จะมีลักษณะดังรูป

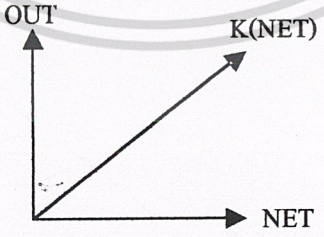



รูปที่ 3.2 นิวรอลของโครงข่ายประสาทเทียม

จากรูปจะมีลักษณะคล้ายกับนิวรอลเซลล์ กล่าวคือสัญญาณอินพุท $x_1, x_2, \ldots, x_n$ จะถูกป้อนเข้ามายังนิวรอล เปรียบได้กับสัญญาณเข้ามายังไซแนปส์ของเซลล์ประสาท ค่าอินพุทเหล่านี้จะคูณด้วยค่าถ่วงน้ำหนัก (Weight) ที่มีค่าตั้งแต่ 0.0-1.0 ซึ่งเปรียบเสมือนส่วนของไซแนปส์ ซึ่งสามารถปรับค่าได้ ผ่านเข้ามาในส่วนที่ทำหน้าที่รวมอินพุทที่เข้ามา และออกเป็นค่าเอาท์พุทค่าหนึ่ง สำหรับค่าเอาท์พุทที่ออกมาสามารถคำนวณได้ดังนี้

$$\begin{aligned} NET &= w_1 x_1 + w_2 x_2 + \ldots + w_n x_n \\ &= \sum_{i=1}^{n} (w_i x_i) \end{aligned}$$

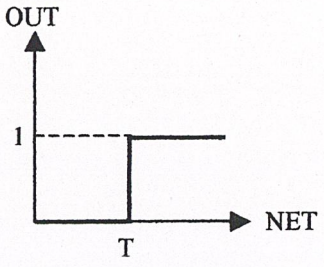
จากนั้นจะถูกกระทำต่อไป โดยส่วนที่เรียกว่าฟังก์ชันการกระตุ้น (Activation Function) และได้ค่าเอาท์พุทออกมา ฟังก์ชันการกระตุ้นนี้มีอยู่หลายรูปแบบ อาจจะมีลักษณะเป็นสมการเชิงเส้นอย่างง่าย ดังรูป 3.3




OUT = K(NET) ; โดย K เป็นค่าคงที่

รูปที่ 3.3 สมการเชิงเส้นอย่างง่าย

หรือในรูปของฟังก์ชันเทรชโฮล (Threshold) แบบ Hard linear function (ดังรูป 3.4)

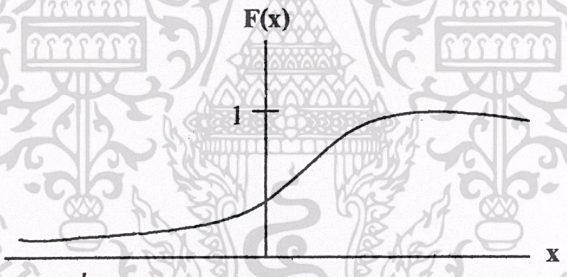



OUT = 1 ถ้า NET > T

OUT = 0 ถ้า NET มีค่าเป็นกรณีอื่น

รูปที่ 3.4 ตัวอย่างเทรชโฮล

หรืออาจเป็นฟังก์ชันเทรชโฮลแบบอื่น ๆ ที่เลียนแบบคุณสมบัติไม่คงที่ของเซลล์ประสาทได้ดีกว่า และใช้เป็นฟังก์ชันให้กับโครงข่ายทั่วไปได้ ฟังก์ชันที่นิยมใช้กันทั่วไปคือ ซิกมอยด์ ฟังก์ชัน (Sigmoid function) ซึ่งมีรูปร่างดังต่อไปนี้ (รูปที่3.5)



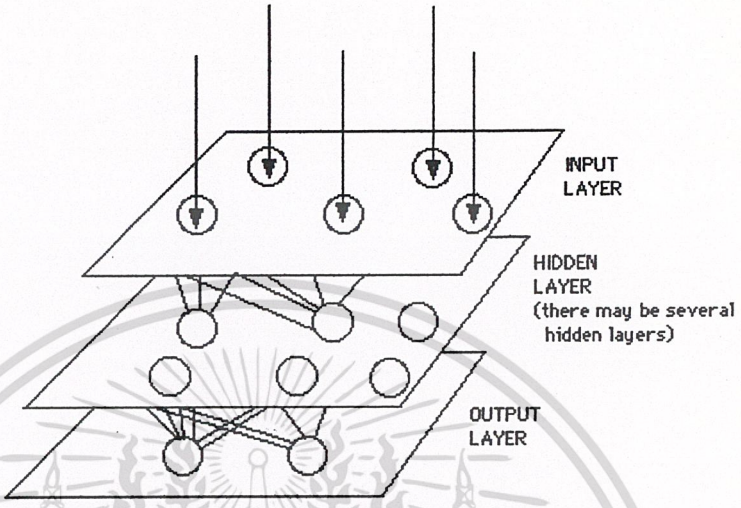

รูปที่ 3.5 ซิกมอยด์ ฟังก์ชัน (Sigmoid function)

การใช้ซิกมอยด์ ฟังก์ชัน จะทำให้เทรชโฮลฟังก์ชันมีลักษณะไม่เป็นเชิงเส้น (Non linear function) ทำให้ได้ค่าอินพุทที่ไวต่อสัญญาณขนาดเล็ก ๆ และเฉลี่ยต่อสัญญาณแรง ๆ คือสัญญาณไปทางบวกเล็กน้อยก็จะมีค่าเอาท์พุทใกล้เคียง “1” และถ้าสัญญาณเป็นลบต่ำ ๆ ก็จะมีค่าใกล้เคียง “0” หรือกล่าวได้ว่ามีคุณสมบัติ Non linear gain นั่นเอง ซึ่ง กลอสเบอร์ก (Grossberg, 1973) พบว่าสามารถแก้ปัญหา Noise Saturation Dilemma ได้ และทำให้นิวรอลสามารถทำงานได้กว้างขวางขึ้น

### 3.3 ชั้นของโครงข่ายประสาทเทียม (Layer of Neural Network)

ตามแนวชีววิทยา นิวรอลเซลล์นั้นสร้างขึ้นอยู่ในรูป 3 มิติ โดยนิวรอลเซลล์เหล่านี้แทบจะไม่มีการเชื่อมต่อถึงกัน ซึ่งไม่เป็นความจริงสำหรับโครงข่ายใด ๆ ที่มนุษย์สร้างขึ้น โครงข่ายประสาทเทียมนี้จะรวมกันเป็นกลุ่มก้อนที่ไม่ซับซ้อนมากนักและก่อตัวขึ้นเป็นชั้น โดยแต่ละชั้นนี้จะ

สามารถเชื่อมต่อถึงกันได้ การเชื่อมต่อถึงกันนี้มีได้หลายรูปแบบ แต่ไม่ว่าจะเป็นรูปแบบใดนั้นก็จะมีโครงสร้างเป็นแบบเดียวกันเสมอ (ดังรูปที่ 3.6)




รูปที่ 3.6 โครงสร้างชั้นของโครงข่ายประสาทเทียม

จากรูปด้านที่ 3.6 จะเห็นได้ว่านิวรอลรวมตัวกันเป็นชั้น ๆ สำหรับชั้นอินพุท (Input layer) นั้นจะทำการรับสัญญาณอินพุทจากสิ่งแวดล้อมภายนอกเข้ามา ส่วนชั้นเอาท์พุท (Output layer) จะทำการส่งสัญญาณเอาท์พุทไปยังนอกระบบต่อไป สำหรับชั้นซ่อน (Hidden layer) นั้นสามารถมีได้หลายชั้นด้วยกัน แต่รูปด้านบนนี้แสดงไว้ให้เห็นเพียงชั้นเดียว

เมื่อชั้นอินพุทได้รับสัญญาณเข้ามาแล้ว นิวรอลในชั้นนี้จะทำการผลิตเอาท์พุทเพื่อส่งไปเป็นสัญญาณอินพุทสำหรับชั้นถัดไป กระบวนการนี้จะถูกกระทำอย่างต่อเนื่องจนกว่าเงื่อนไขถูกยอมรับหรือจนกว่าจะถึงชั้นเอาท์พุทเพื่อทำการส่งสัญญาณนี้ต่อไปยังส่วนนอกระบบ

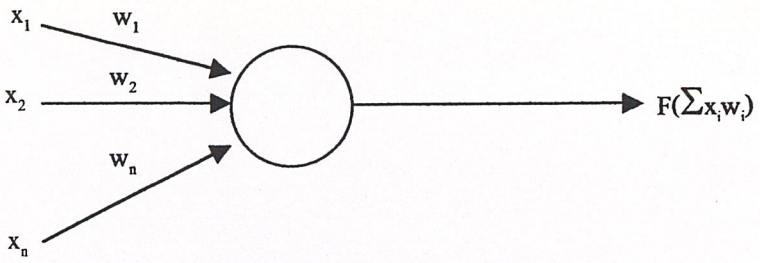
สำหรับชั้นซ่อนนั้นควรจะมีค่าของจำนวนชั้นที่เหมาะสม ที่ไม่มากเกินไปหรือน้อยเกินไป เพราะอาจก่อให้เกิดปัญหาต่าง ๆ ตามมา

### 3.3.1 โครงข่ายประสาทเทียมแบบชั้นเดียว (Single Layer Neural Network)

โครงข่ายประสาทเทียมแบบชั้นเดียวที่ไม่มีการไหลย้อนกลับของค่าสัญญาณเอาท์พุท จะมีความสามารถใช้ตัดสินใจกับปัญหาที่เป็นฟังก์ชันเชิงเส้นได้เท่านั้น

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

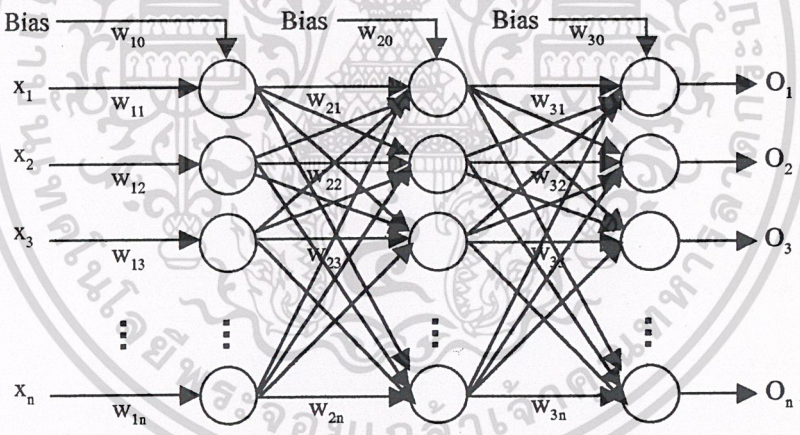





รูปที่ 3.7 โครงข่ายประสาทเทียมแบบชั้นเดียว

**3.3.2 โครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Neural Network)**

ซึ่งประกอบด้วยชั้นอินพุท (Input layer), ชั้นเอาท์พุท (Output layer) และชั้นซ่อน (Hidden layer) จำนวนตั้งแต่ 1 ชั้นขึ้นไป ทำให้ความสามารถของโครงข่ายดีขึ้น คือสามารถแก้ปัญหาที่ไม่เป็นเชิงเส้นได้ ปัญหาที่มักเกิดขึ้นจากการออกแบบโครงข่ายประสาทแบบหลายชั้นจะเป็นแง่ของการกำหนดขนาดจำนวนหน่วยในแต่ละชั้น จำนวนชั้นซ่อนที่เหมาะสม และลักษณะการเชื่อมของแต่ละหน่วย แต่ละชั้น




รูปที่ 3.8 โครงข่ายประสาทเทียมแบบหลายชั้น

**3.4 การฝึกสอนให้กับโครงข่าย(Training Algorithm)**

เทรนนิ่งอัลกอริทึมถูกจัดเป็น 2 ประเภท คือ แบบควบคุม (Supervise Training) และแบบอิสระ (Unsupervise) โดยเทรนนิ่งแบบควบคุมต้องการข้อมูล 2 ชุด คืออินพุทกับชุดเป้าหมาย เรียกเทรนนิ่งแพร์ (Training Pairs) เช่น การเรียนรู้แบบแพร่ย้อนกลับ ส่วนการเทรนนิ่งแบบอิสระสร้างขึ้นโดยเปรียบเทียบการทำงานของสมองที่ไม่จำเป็นต้องมีผู้มาคอยคิดค่าเป้าหมายให้ก่อน โครงข่ายจะรับเพียงค่าอินพุทเข้าไปเท่านั้น เทรนนิ่งอัลกอริทึมจะเปลี่ยนแปลงค่าถ่วงน้ำหนักของโครงข่ายเพื่อสร้างเอาท์พุทที่มั่นคง และเมื่อค่าอินพุทมีการเปลี่ยนแปลงไปเพียงเล็กน้อย โครงข่ายจะยังคง

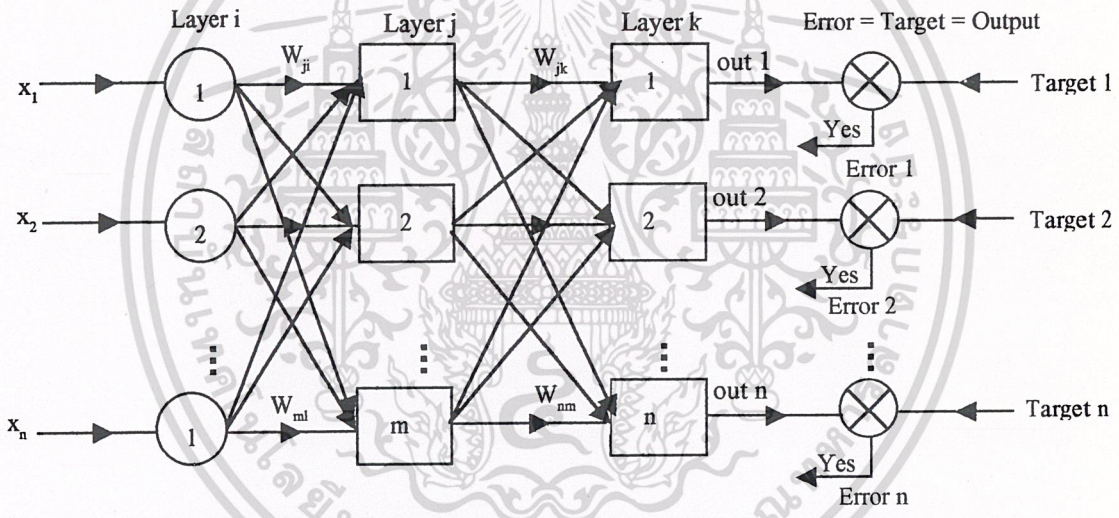
46126

สามารถชี้ได้ว่าเป็นข้อมูลเดิม เนื่องจากเอาท์พุทของโครงข่ายไม่มีการกำหนดมาก่อน ส่วนใหญ่จะถูกแปรรูปให้เข้าใจได้ จึงไม่สามารถใช้ตัดสินใจปัญหาที่มีความยากได้ มักใช้ในงานง่าย ๆ เช่น การเปรียบเทียบเอกลักษณ์, รูปภาพ, รูปแบบที่สัมพันธ์กันระหว่างอินพุทและเอาท์พุท

### 3.4.1 ขั้นตอนการเรียนรู้แบบแพร่ย้อนกลับ (Backpropagation Learning)

การฝึกสอนให้กับโครงข่ายประสาทเทียมแบบหลายชั้นใช้การเรียนรู้แบบแพร่ย้อนกลับ โดยโครงข่ายจะรับอินพุท X และค่าเป้าหมาย (Target) ไว้ เมื่อคำนวณค่าเอาท์พุทในโครงข่ายแล้วจะเปรียบเทียบค่าผิดพลาดของเอาท์พุทที่คำนวณได้กับค่าเป้าหมาย

$$\text{Error} = \text{Target} - \text{Output}$$




รูปที่ 3.9 ไดอะแกรมของ Backpropagation Neural Network แบบ 2 ชั้น

ค่าผิดพลาดนี้จะนำไปใช้ปรับค่าถ่วงน้ำหนักของโครงข่ายเพื่อให้สอดคล้องกับค่าเป้าหมาย การเรียนรู้จะทำไปเรื่อย ๆ จนโครงข่ายสามารถแยกแยะข้อมูลได้

ขั้นตอนการฝึกสอนสามารถสรุปเป็นขั้น ๆ ได้ดังต่อไปนี้

1. สร้างโครงข่ายประสาทที่มีจำนวนอินพุท และเอาท์พุทเท่ากับจำนวนอินพุท และเอาท์พุทของปัญหา ส่วนจำนวนชั้นซ่อนอาจเริ่มที่ค่ามากพอสมควรก่อน
2. ให้ค่าเริ่มต้นแก่ค่าถ่วงน้ำหนักในโครงข่าย โดยควรมีค่าในช่วง −1 ถึง 1
3. คำนวณค่า $\sum X_i W_i = U_i$ สำหรับแต่ละนิวรอล j ของชั้นซ่อน
4. คำนวณค่ากระตุ้น (Activation) สำหรับแต่ละนิวรอล j ของชั้นซ่อน

$$Y_j = f(U_i)$$

โดยฟังก์ชันที่นิยมใช้คือ sigmoid function $f(x) = 1/[1+\exp(-x)]$

5. คำนวณค่า $\sum Y_i W_i = V_I$ สำหรับแต่ละนิวรอล k ของชั้นเอาท์พุท
6. คำนวณค่ากระตุ้น (Activation) ของแต่ละนิวรอล k

$$OUT = Z_k = f(V_i)$$

7. นำค่า Zk มาเปรียบเทียบกับเป้าหมาย ถ้าค่าผิดพลาดที่คำนวณได้น้อยกว่าระดับที่กำหนดไว้ก็จบการเรียนรู้ ถ้าค่าผิดพลาดมากกว่าระดับที่กำหนดไว้ให้ทำต่อไป
8. คำนวณค่าปรับน้ำหนักสำหรับแต่ละค่าถ่วงน้ำหนัก

8.1 สำหรับเส้นเชื่อมชั้นซ่อนกับชั้นเอาท์พุท

$$\Delta W_{jk} = \alpha \delta_k Y_j$$

โดยที่ $\delta_k = (Target_k - Z_k) f'(V_k)$

8.2 สำหรับเส้นเชื่อมชั้นอินพุทกับชั้นซ่อน

$$\Delta W_{ij} = \alpha \delta_j X_I$$

โดยที่ $\delta_j = \sum \delta_k W_{jk} f'(U_j)$

9. ปรับค่าน้ำหนักเส้นเชื่อมจากนิวรอล r ไปยังนิวรอล s จะได้
10. กลับไปทำข้อ 3

ค่า $\alpha$ : อัตราการเรียนรู้ (Learning rate) มีค่าในช่วง 0 ถึง 1

การปรับความเร็วในการเรียนรู้อาจใช้การปรับค่า Gain ของ sigmoid function ในส่วนดีกรีของ exponential การปรับอัตราการเรียนรู้ ขนาดจำนวนหน่วยในชั้นซ่อน และการเพิ่มโมเมนตัมแฟกเตอร์ เพื่อคำนวณทิศทางการปรับค่าถ่วงน้ำหนัก นอกจากนี้ยังเลือกใช้อัตราการเรียนรู้เป็นฟังก์ชันที่มีความเหมาะสมได้อีกด้วย

ฟังก์ชันซิกมอยด์เป็นฟังก์ชันที่นิยมใช้ในการเทรนนิ่งเนื่องจากมีความใกล้เคียงกับฟังก์ชันขั้นบันได และสามารถคำนวณค่าอนุพันธ์ได้ง่ายอีกด้วย โดยอยู่ในรูปนิพจน์ของตัวฟังก์ชันเอง

$$f'(x) = f(x)[1 - f(x)]$$

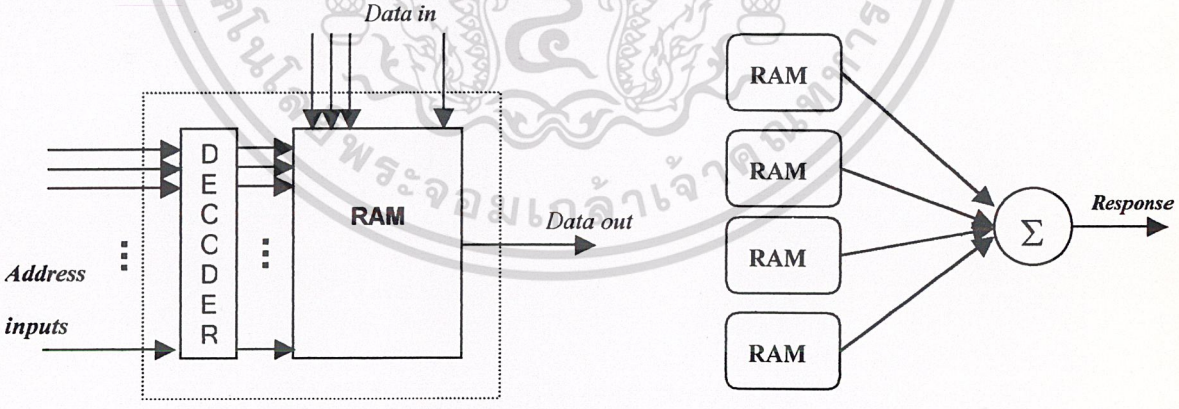
### 3.5 Weightless Neural Networks

Weightless models คือ RAM ที่เป็นพื้นที่สำหรับเก็บข้อมูลการกระตุ้นของอินพุท และผลตอบสนองต่อความสัมพันธ์ใน look-up tables ไปในรูปแบบของปฏิกิริยาทางตรรกศาสตร์ของนิวรอล ต้นแบบของความคิดนี้ถูกเสนอโดย Bledsoe และ Browning (1959) ซึ่งรู้จักกันในชื่อของ

กระบวนการจำแนกข้อมูลแบบ n-tuple (n-tuple recognition process) tuple แต่ละยูนิตจะรับข้อมูลเพียงส่วนเล็ก ๆ ของข้อมูลทั้งหมดและตอบสนองโดยผลที่ได้ของแต่ละยูนิตเป็นอิสระต่อกัน การจำแนกข้อมูลใน n-tuple machine สามารถทำได้โดยเปรียบเทียบลักษณะที่คล้ายกันระหว่างรูปแบบของข้อมูลที่เก็บภายในความจำของคอมพิวเตอร์ซึ่งมาจาก tuples ทั้งหมดกับรูปแบบของข้อมูลที่ป้อนเข้าไปโดยให้ลักษณะที่คล้ายคลึงกันออกมาในรูปของคะแนน ซึ่งข้อได้เปรียบของเทคนิคนี้คือข้อมูลหลายตัวสามารถมีผลตอบสนองร่วมกันใน tuple หนึ่งยูนิต และสามารถแยกรูปแบบของข้อมูลแบบไม่เป็นเชิงเส้นออกมาได้

ระหว่างทศวรรษที่ 70 การพัฒนาเทคโนโลยีของวงจรรวม (Integrated Circuit Technology) สามารถสร้าง Random Access Memories (RAMs) ได้ซึ่งเป็นการนำไปสู่การค้นพลนิวรอลเน็ทเวิร์คที่ใช้หน่วยความจำแบบ RAMs เป็นที่เก็บข้อมูลของ n-tuple machines โดย Aleksander และ Stonham (1979) หลังจากนั้นหลักการเดียวกันนี้ถูกนำไปใช้ในออกแบบ WISARD (Aleksander, Thomas and Bowden 1984, Aleksander 1984)

เมื่อพิจารณาการใช้นิวรอลเน็ทเวิร์คแบบที่ใช้ RAMs แล้วพบว่ามีข้อดีในการเรียนรู้ algorithm เนื่องจากเป็นโหนดแต่ละโหนดอิสระต่อกันระหว่างการอ้างอิงข้อมูลจาก look-up table เว้นแต่ในส่วนของ weighted models ในส่วนที่สองซึ่งน้ำหนักของข้อมูลต้องขึ้นกับการเปลี่ยนแปลงในส่วนแรก ส่วนข้อดีอื่น ๆ นั้นคือฮาร์ดแวร์ซึ่งประกอบด้วย RAMs มีราคาถูกและสามารถหาได้ง่าย โดยบล็อกไดอะแกรมพื้นฐานของนิวรอลเน็ทเวิร์คแบบที่ใช้ RAMs นี้สามารถแสดงได้ดังรูปที่ 3.10




รูปที่ 3.10 (a) A RAM-based neuron (b) 4 RAMs discriminators with 3 inputs each

โมเดลของนิวรอลแบบไร้น้ำหนัก (weightless neural model หรือ WISARD) คือระบบที่มีการป้อนไปข้างหน้าแบบชั้นเดียวที่ประกอบไปด้วยเซ็ตสัญลักษณ์การแบ่งแยกคลาสของ

discriminators รูปแบบของข้อมูลตัวแบ่งแยกทั้งหมดจะเริ่มต้นที่เซ็ตของศูนย์และค่อย ๆ เปลี่ยนไปตามการป้อนข้อมูล (training) ระหว่างการป้อนข้อมูลรูปแบบของข้อมูลแต่ละอันก็จะถูกเก็บอยู่แค่ใน RAM ของการตอบสนองการแบ่งแยกหลังจากการป้อนข้อมูลเสร็จสิ้นก็จะได้ผลการแบ่งแยกของแต่ละคลาส การจำแนกข้อมูล (recognition) ก็จะเริ่มโดยป้อนรูปแบบของข้อมูลที่ไม่ทราบค่าให้กับ discriminators ทุก ๆ ตัวเพื่อหาผลตอบสนองที่มีค่ามากที่สุดโดยหลักการทั่ว ๆ ไปของการคิดคือไม่นำมาคิดแค่เพียงแต่ละโหนดแต่คิดทั้งระดับของเน็ทเวิร์ค

เนื่องจากการทำงานเป็นเส้นตรงของ WISARD จะมีความคลุมเครือระหว่างสถานะ 0 ที่เกิดขึ้น ระหว่างการเก็บรูปแบบของข้อมูลและสถานะ 0 ที่จุดเริ่มต้นก่อนจะเกิดการป้อนข้อมูลหรือ training เพื่อแก้ปัญหาข้อนี้ Kan และ Aleksander (1988) ได้นำเสนอการใช้ Probabilistic Logic Node (PLN) ซึ่งแต่ละโหนดจะยังคงมีเอาท์พุทเป็นเลขฐานสองแต่จะมีสถานะการทำงานเป็น 3 ระดับ โดยสถานะที่เพิ่มขึ้นมาก็คือ u-state ซึ่งจะแทนตรงส่วนที่ค่าเริ่มต้นหรือในช่วงที่ข้อมูลยังไม่ถูกเก็บ ซึ่งโอกาสที่เอาท์พุทของโหนดจะเป็น 1 และ 0 จะมีค่า 50% ใน recalling mode

เมื่อการที่ข้อมูลจะเข้าไปได้มีเพียงแค่การป้อนข้อมูล (trained) สำหรับการประยุกต์การเรียนรู้ข้อมูล แต่ใน PLNs ข้อมูลจะมาจาก delay เมื่อเซ็ตของการป้อนข้อมูลมีค่าน้อยเช่น 2 หรือ 3 รูปแบบ เนื่องจากบนตำแหน่งส่วนใหญ่จะยังคงเหลืออยู่ภายใน u-state อุปสรรคอีกอย่างหนึ่งที่ควรจะต้องกล่าวถึงคือการ generalisation ยังแสดงผลออกมาเป็นระดับต่าง ๆ เนื่องจากปัญหานี้ Aleksnder (1990a) ได้เสนอเทคนิคที่จะไม่เห็นอินพุทเป็นลักษณะระดับของโหนด เราเรียกวิธีการนี้ว่า spreading process อัลกอริทึมของมันคือระบุตำแหน่งของ u-state ที่มาจากเน็ทเวิร์คข้าเคียงที่ใกล้ที่สุดโดยหลักของ Hamming distance หากข้อมูลไม่ตรงกันหรือเกิดความขัดแย้งระหว่างเน็ทเวิร์คที่ใกล้ที่สุดกันเอง ตำแหน่งของ u-state จะไม่เปลี่ยนแปลง การ generalisation ของเน็ทเวิร์คจะขึ้นกับรัศมีของการกระจายชั้นของแต่ละโหนดของเน็ทเวิร์ค ถ้ารัศมีมีค่าเท่ากับผลรวมจำนวนของอินพุทที่เป็นเลขฐานสอง หมายความว่าเน็ทเวิร์คจะมีการกระจายเต็มที่ เราเรียกเน็ทเวิร์คที่มีคุณสมบัติเช่นนี้ว่า Generalising RAMs หรือ GRAMs

ความจุของการ generalisation ใน GRAMs สามารถหาได้โดยจำนวนของรูปแบบของการเรียนรู้ข้อมูล (training patterns) เนื่องจากการกระจายของชั้นของเน็ทเวิร์คจะมีค่าน้อยลงเมื่อมีจำนวนของรูปแบบเพิ่มขึ้น นอกจากนี้ยังมีอัลกอริทึมของการกระจายหรือ spreading ในแบบอื่น ๆ อีกเช่น the Combined Generalisation Algorithm (CGA) ค้นพบโดย Aleksander Clarke และ Braga (1994) โดยเป็นวิธีที่ผสมผสานกันระหว่างการเรียนรู้แบบHebbian และ Weightless

### 3.5.1 The General Neural Unit (GNU)

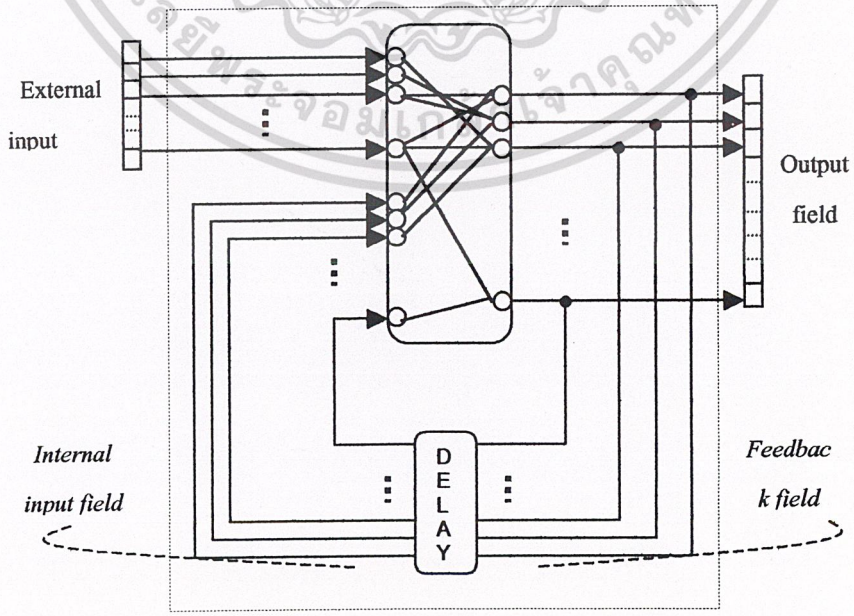
The General Neural Unit (GNU) จะเกิดขึ้นเมื่อมีการเชื่อมโยงโดยใช้ GRAM-based neurons ในเน็ทเวิร์คโมเดล GNU สามารถที่จะ mapping ได้ทั้งแบบอัตโนมัติและแบบ hetero-associative mode (Zurada 1992) กำหนดให้แต่ละโหนดมีจำนวนรวมของอินพุทเป็น I ประกอบด้วยจำนวนอินพุทที่ได้จากการสุ่ม N จำนวน ต่ออยู่กับบิตจำนวน W บิตของ External input field และ F แทนอินพุทที่มาจากการสุ่มซึ่งต่ออยู่กับ Feedback input field จำนวน K บิต สามารถเขียนความสัมพันธ์ได้ดังนี้

$$I = N + F$$

GNU จะมีคุณสมบัติเป็น fully connected หากอัตราส่วน *N/W* และ *F/K* มีค่าเท่ากับ 1 และจะมีคุณสมบัติเป็น partially connected เมื่ออัตราส่วนของจำนวนดังกล่าวมีค่าน้อยกว่า 1 เราสามารถสรุปพารามิเตอร์หลักของ GNU ได้ดังตารางที่ 2.1 และไดอะแกรมของรูปแบบการทำงานดังรูปที่ 3.11

| | |
|---|---|
| ***K*** | ***The number of nodes of a GNU*** |
| ***W*** | ***The number of external input*** |
| ***I*** | ***Total number of inputs to each neuron*** |
| ***N*** | ***The number of inputs connected from the external field*** |
| ***F*** | ***The number of inputs connected from the feedback field*** |

ตารางที่ 3.1 Main parameters of a GNU




รูปที่ 3.11 A schematic view of a basic GNU

การศึกษา GNU ในรูปแบบของ Boolean network ซึ่งทำการวิจัยโดย Wong และ Sherrington (1988, 1989) ส่วนงานวิจัยของ Lucy (1991) แสดงให้เห็นว่าการที่ GNU มีการเชื่อมต่อในแบบ fully connected auto-associative mode หรือ ($W$=0) จะสามารถเก็บข้อมูลได้เป็นจำนวน $2^K$ รูปแบบ ผลลัพธ์ที่ดีที่สุดของการดำเนินงานซึ่งศึกษาโดย Aleksander และ Morton (1991) โดยกำหนดให้จำนวนของการเชื่อมต่อมีค่าระหว่างจำนวนอินพุทกับจำนวน feedback หรือ $N/W = F/K$ การคาดคะเนของความขัดแย้ง GNU ทำการวิจัยโดย Braga( 1993; 1994) ส่วนโปรแกรมที่ทำการจำลอง GNUs กับข้อมูบจริง ๆ มีชื่อว่า MAGNUS (Multi Automata General Neural Units) ร่วมกันพัฒนาโดย Aleksander, Evans และ Penney (1993)

### 3.5.2 การทำงานของ GNUs

เราสามารถอธิบาย GNU โดยแบ่งการทำงานเป็นส่วนของ GRAMs ได้เป็น 3 ขั้นตอนคือ storage spread และ retrieval process ในการทำงานแบบ storage process จะเป็นการ mapping ของเน็ทเวิร์คระหว่างอินพุท (ทั้ง external inputs และ internal representations) และเอาท์พุท โดยการสร้างสถานะเสถียร (stable state) ภายใน Boolean space สถานะของ n-demensional Boolean space สำหรับโมเดลของ single GNU สามารถแสดงได้โดย two-dimensional Boolean transition table ในแบบพื้นฐาน

การทำงานแบบ storage process มีหลักการคือความสัมพันธ์ของอินพุทและเอาท์พุทในทุก ๆ โหนดของเน็ทเวิร์ค ภายใน GRAM ซึ่งจะเริ่มต้นจาก u-state จากนั้นก็จะมีการออกแบบตำแหน่งของสถานะของเอาท์พุท รูปแบบลำดับของการทำงานในแบบ storing นั้นเป็นการ representations ภายในเชื่อมไปถึงลำดับการทำงานเอาท์พุทซึ่งเปรียบเสมือนอ่าง

storage ต่อไปจะเป็นกระบวนการ spreading ซึ่งช่วยในการลดจำนวน delay ของเน็ทเวิร์ค เนื่องจากมีหลาย ๆ ตำแหน่งยังคงเหลืออยู่ภายใน u-state ซึ่งการกระจายจะนำที่ว่างในแต่ละโหนดแยกออกเป็นอิสระต่อกัน การกระจายใน GNUs จะคล้ายคลึงกับแนวคิดในการสร้างพลังงานที่น้อยที่สุดของ Hopfield network โดยมีความแตกต่างกันคือ auto-associative ใน Hopfield model จะมีความเป็นไปได้ที่จะเกิดความผิดพลาดในการป้อนกลับ เนื่องจากความสัมพันธ์ระหว่างข้อมูลที่เข้ามาใหม่และข้อมูลที่ถูกเก็บไว้ก่อนหน้านั้น ขณะที่โมเดลของ GNUs มีการเชื่อมต่อแบบจะไม่เกิดความผิดพลาดระหว่างความสัมพันธ์ของรูปแบบของข้อมูล นอกจากนั้น การกระจายใน GNUs จะสร้าง attractors ด้วยการเปลี่ยนแปลงในแบบ steeper ผลที่ได้คือจะใช้เวลาในการดำเนินการน้อยกว่า อย่างไรก็ตามบางครั้งก็เกิดความผิดพลาดในการสร้าง attractor หากการสร้างนั้นมีขนาดกว้างเกินไป เนื่องจากรูปแบบของข้อมูลที่เก็บมีค่าน้อยเกินไป นอกจากนั้นการ storage ของความสัมพันธ์

ระหว่างรูปแบบของข้อมูล สามารถเกิดความขัดแย้งกันได้ในโมเดลของ GNU ที่มีการเชื่อมโยงกันน้อยเกินไป reprocess จะทำงานได้ดีเมื่อทั่วทั้งเน็ทเวิร์คในทุก ๆ โหนดสามารถเปลี่ยนแปลงข้อมูลได้พร้อม ๆ กันภายในเวลาเดียวกัน ดังเช่นการแสดงภาพจะเกิดขึ้นพร้อม ๆ กันกับอินพุทและในช่วงริ่มต้นของสถานะภายใน จำนวนที่เพิ่มขึ้นของ attractors จะมีผลต่อการทำงานของเน็ทเวิร์คให้มีความรวดเร็วและต่อเนื่องมากขึ้น อย่างไรก็ตามถ้าหากจำนวนของ attractors มีมากเกินไปเมื่อเทียบกับความจุของเน็ทเวิร์คก็อาจจะทำให้การเรียกกลับ (recall) ล้มเหลวก็เป็นได้ เนื่องจากมีสถานะที่ไม่เสถียรมากเกินไปทำให้เกิดความขัดแย้งกัน จำนวนรูปแบบการเก็บข้อมูลจะต้องมีน้อยกว่าความสามารถของเน็ทเวิร์คเพื่อการป้อนกลับที่มีความถูกต้องสูง

ทั้งนี้ในโครงงานนี้ได้เลือกใช้โครงข่ายประสาทเทียมแบบหลายชั้นในการใช้ทำการประมวลผลภาพ ซึ่งจะได้ค่าตัวแทนของรูปภาพรูปนั้นเก็บลงสู่ฐานข้อมูล และทำการฝึกสอนให้กับโครงข่ายประสาทเทียมโดยเลือกใช้การฝึกสอนแบบอิสระ เนื่องจากไม่มีการกำหนดค่าเป้าหมายไว้ให้ก่อน ซึ่งการเลือกใช้เทคนิคในรูปแบบเหล่านี้จะทำให้ได้รับค่าตัวแทนของรูปภาพซึ่งสามารถทำการบ่งชี้รูปภาพนั้น ๆ ได้อย่างถูกต้อง

# บทที่ 4
# หลักการทำงานและการออกแบบระบบ

ในปัจจุบันนี้โปรแกรมการสืบค้นโดยการใช้ภาพใบหน้าคนได้มีการพัฒนาและปรับปรุงอย่างกว้างขวาง และมีผู้ที่สนใจศึกษาทำการเขียนโปรแกรมด้วยภาษาต่าง ๆ หลายภาษา ซึ่งในปริญญานิพนธ์ฉบับนี้ได้เลือกใช้ภาษาวิชวลเบสิก (Visual Basic) ในการเขียนโปรแกรมเพื่อทำการสืบค้นโดยการใช้รูปภาพใบหน้าคน โดยมีรูปแบบการเขียนโปรแกรมดังต่อไปนี้

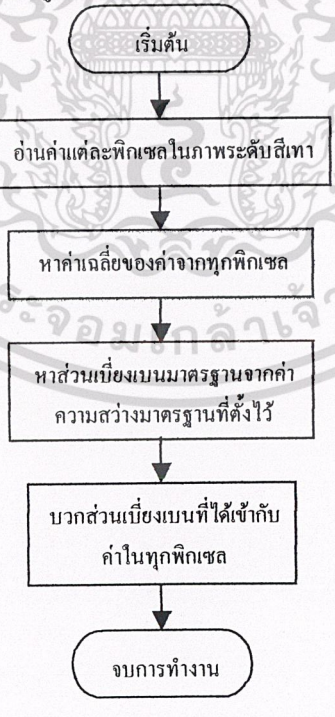
## 4.1 หลักการทำงาน

### 4.1.1 การปรับค่าความสว่าง (Normalization)

การนอร์มอลไลซ์เป็นการปรับค่าความสว่างของภาพเพื่อช่วยลดผลกระทบจากการเปลี่ยนแปลงสิ่งแวดล้อมทางแสง

วิธีการคือ ทำการหาค่าความสว่างมาตรฐานเก็บเป็นค่าคงที่ค่าหนึ่งไว้ และเมื่อรับรูปภาพเข้ามาจะทำการหาค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐานจากค่าความสว่างมาตรฐานที่ตั้งไว้ และทำการปรับค่าความสว่างโดยบวกส่วนเบี่ยงเบนที่ได้กับค่า RGB ของทุกพิกเซล

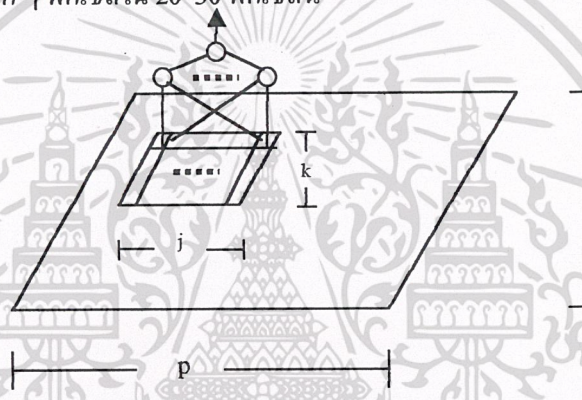
ลำดับการทำงานแสดงไว้ดังรูปที่ 4.1




รูปที่ 4.1 ขั้นตอนการทำงานในส่วนการปรับค่าความสว่างของภาพ

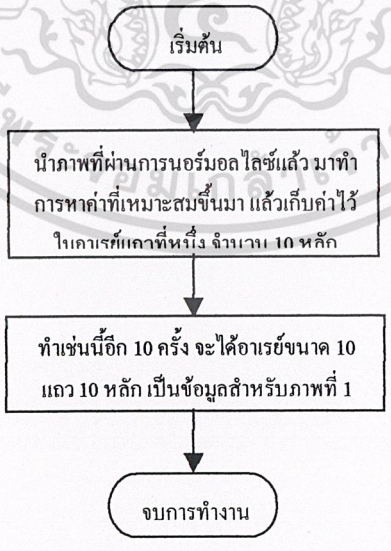
### 4.1.2 การฝึกสอนโครงข่ายประสาทเทียม (Training)

การฝึกสอนเป็นการพยายามสอนให้โครงข่ายประสาทเทียมมีความรู้ตามที่เราต้องการ เพื่อทำการเก็บข้อมูลลงสู่ฐานข้อมูล หลักการฝึกสอนนี้สามารถแสดงให้เห็นได้ดังรูปที่ 4.4 โดยขนาดของภาพอินพุทมีขนาดเป็น p*q พิกเซล และอาเรย์สำหรับเก็บค่าตัวแทนรูปภาพเพื่อทำการฝึกสอนมีขนาด m*n พิกเซล ดังนั้นกลุ่มของพิกเซลที่จะรับเพื่อลงเก็บใน 1 ช่องของอาเรย์ มีขนาด p/m * q/n พิกเซล หรือ j = p/m และ k = q/n สำหรับค่าที่จะนำไปใส่ลงในช่องอาเรย์นี้ จะเป็นค่าที่ได้จากการนำค่าทุกพิกเซลใน j*k พิกเซล มาหาค่าเฉลี่ย ยกตัวอย่างเช่น ภาพอินพุทมีขนาด 200 * 300 พิกเซล และอาเรย์สำหรับเก็บค่าตัวแทนรูปภาพในโครงงานนี้มีขนาด 10 * 10 ช่องอาเรย์ ดังนั้นกลุ่มของพิกเซลที่รับเพื่อลงเก็บใน 1 ช่องของอาเรย์มีขนาด 20* 30 พิกเซล และค่าที่จะนำลงในช่องอาเรย์นี้มาจากการหาค่าเฉลี่ยของทุก ๆ พิกเซลใน 20*30 พิกเซลนี้



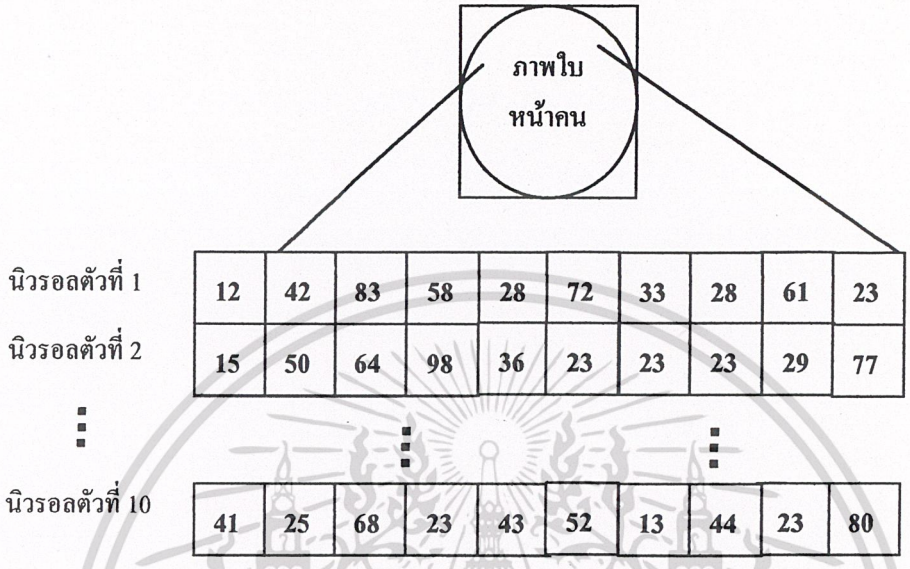

รูปที่ 4.2 วิธีการฝึกสอนโครงข่ายประสาทเทียมจากภาพอินพุท

ขั้นตอนการฝึกสอนแสดงในรูปที่ 4.3 ซึ่งเป็นการฝึกสอนสำหรับภาพข้อมูล 1 ภาพ




รูปที่ 4.3 ขั้นตอนการแสดงส่วนการฝึกสอนให้กับโครงข่ายประสาทเทียม

ทำเช่นนี้กับภาพทุกภาพที่ต้องการเก็บ และสุดท้ายจะได้ฐานข้อมูลของภาพใบหน้าแต่ละภาพเป็นอาเรย์ (Array) ขนาด 10 แถว 10 หลัก




รูปที่ 4.4 แสดงการฝึกสอน

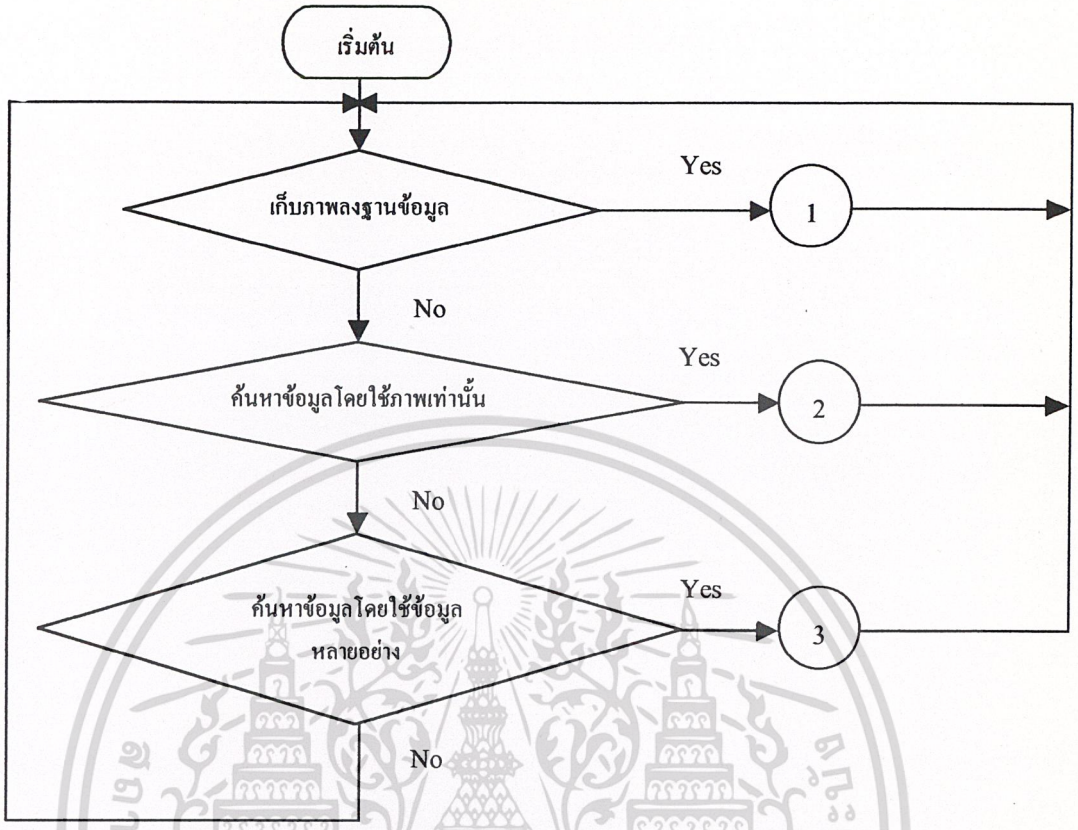
**4.1.3 การเปรียบเทียบเพื่อการสืบค้นภาพใบหน้าคนของแต่ละบุคคล**

เป็นการนำภาพใบหน้าที่เป็นอินพุทมาหาอาเรย์ด้วยวิธีการเดียวกันกับการฝึกสอนดังที่กล่าวไว้ข้างต้น แล้วนำอาเรย์ของภาพอินพุทที่ได้มาเปรียบเทียบกับอาเรย์ข้อมูลในฐานข้อมูล หากอาเรย์ของภาพอินพุทใกล้เคียงกับอาเรย์ของข้อมูลชุดใดมากที่สุด กล่าวคือ ค่าความแตกต่างของอาเรย์ทั้งสองอยู่ในช่วงที่เรายอมรับได้ ก็จะทำการดึงข้อมูลทั้งหมดของชุดนั้นออกมาทำการแสดงผลบนหน้าจอ หากอาเรย์ของภาพอินพุทไม่ใกล้เคียงกับภาพใด ๆ ในฐานข้อมูลเลย การแสดงผลก็จะบอกเพียงว่าไม่พบข้อมูลที่ต้องการ

## 4.2 การออกแบบระบบ

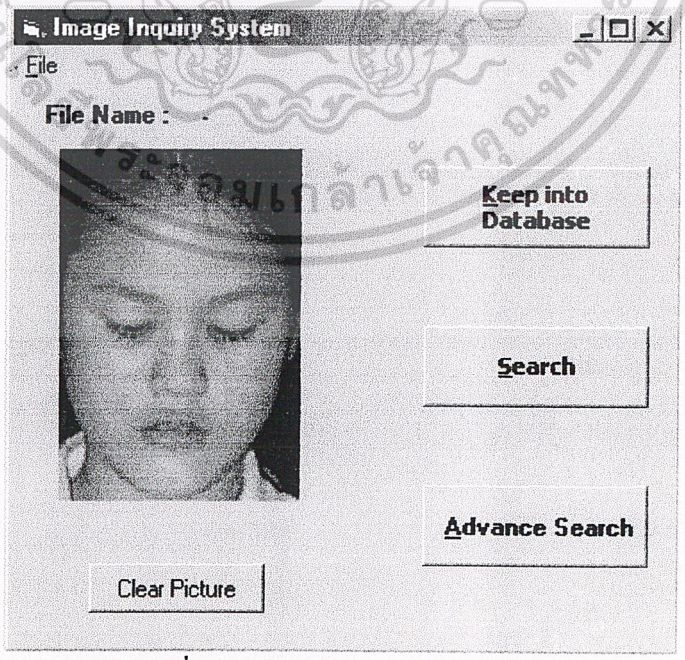
### 4.2.1 ส่วนของโปรแกรม

**4.2.1.1 โครงสร้างโดยรวมของโปรแกรม** มีลำดับขั้นตอนการทำงานของโปรแกรมดังรูปที่ 4.5



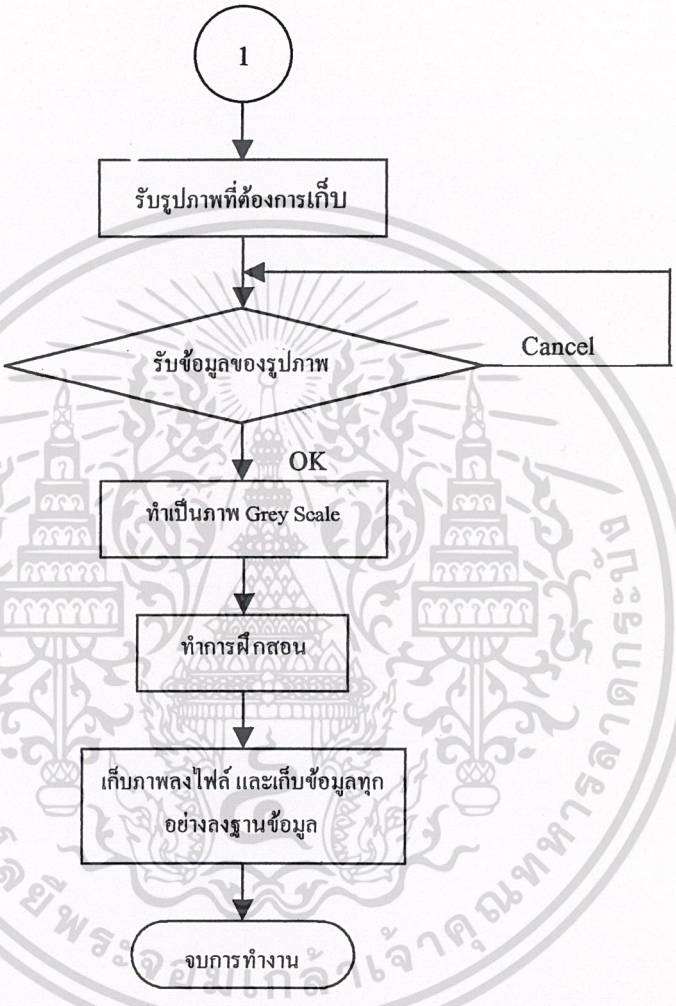

รูปที่ 4.5 ลำดับขั้นตอนการทำงานของโปรแกรมทั้งหมด

สำหรับหน้าจอหลักของโปรแกรม มีลักษณะดังรูปที่ 4.6



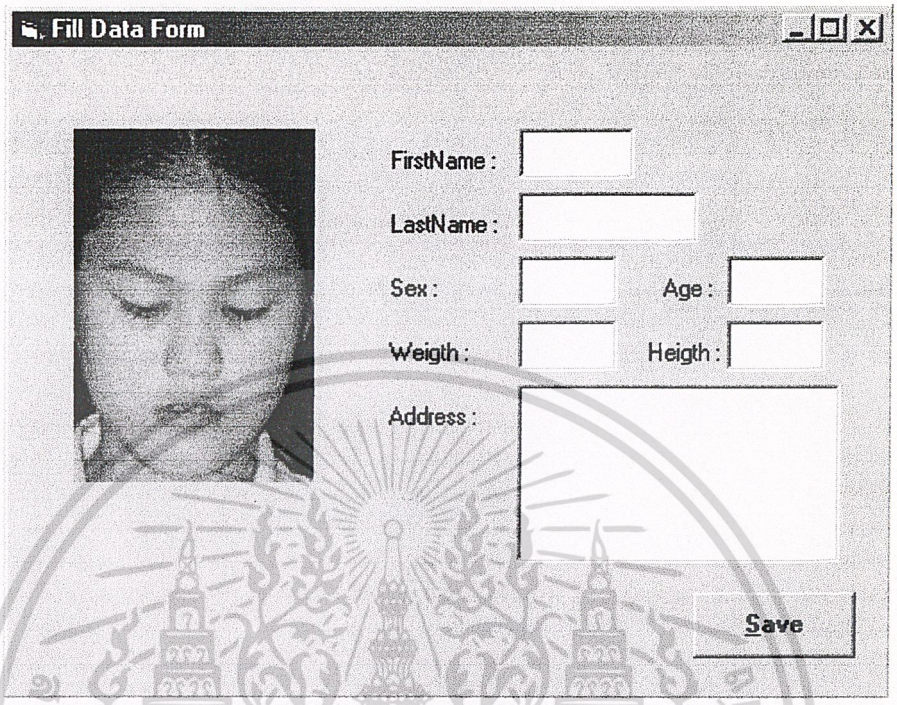

รูปที่ 4.6 หน้าจอหลักของโปรแกรม

**4.2.1.2 Keep into Database** เป็นปุ่มเก็บรูปภาพลงสู่ฐานข้อมูล โดยมีลำดับการทำงานของโปรแกรมดังรูปที่ 4.7



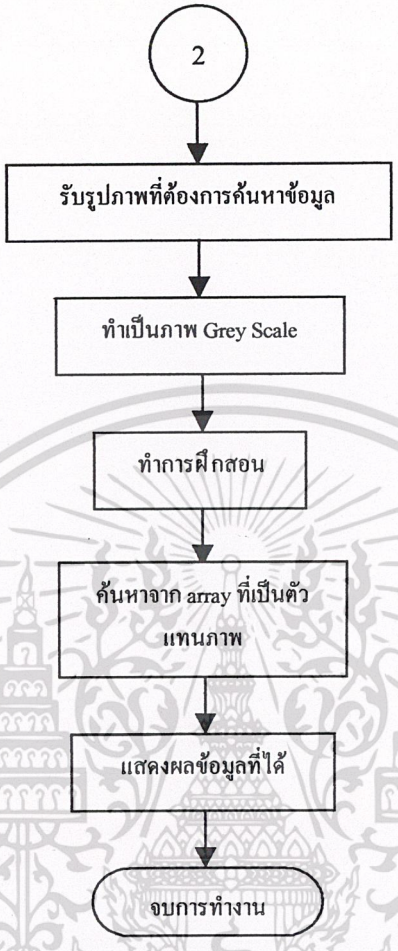

รูปที่ 4.7 ลำดับขั้นตอนการทำงานของโปรแกรมส่วนการเก็บรูปภาพลงสู่ฐานข้อมูล

สำหรับหน้าจอการกรอกข้อมูลเพื่อเก็บลงฐานข้อมูล มีลักษณะดังรูปที่ 4.8



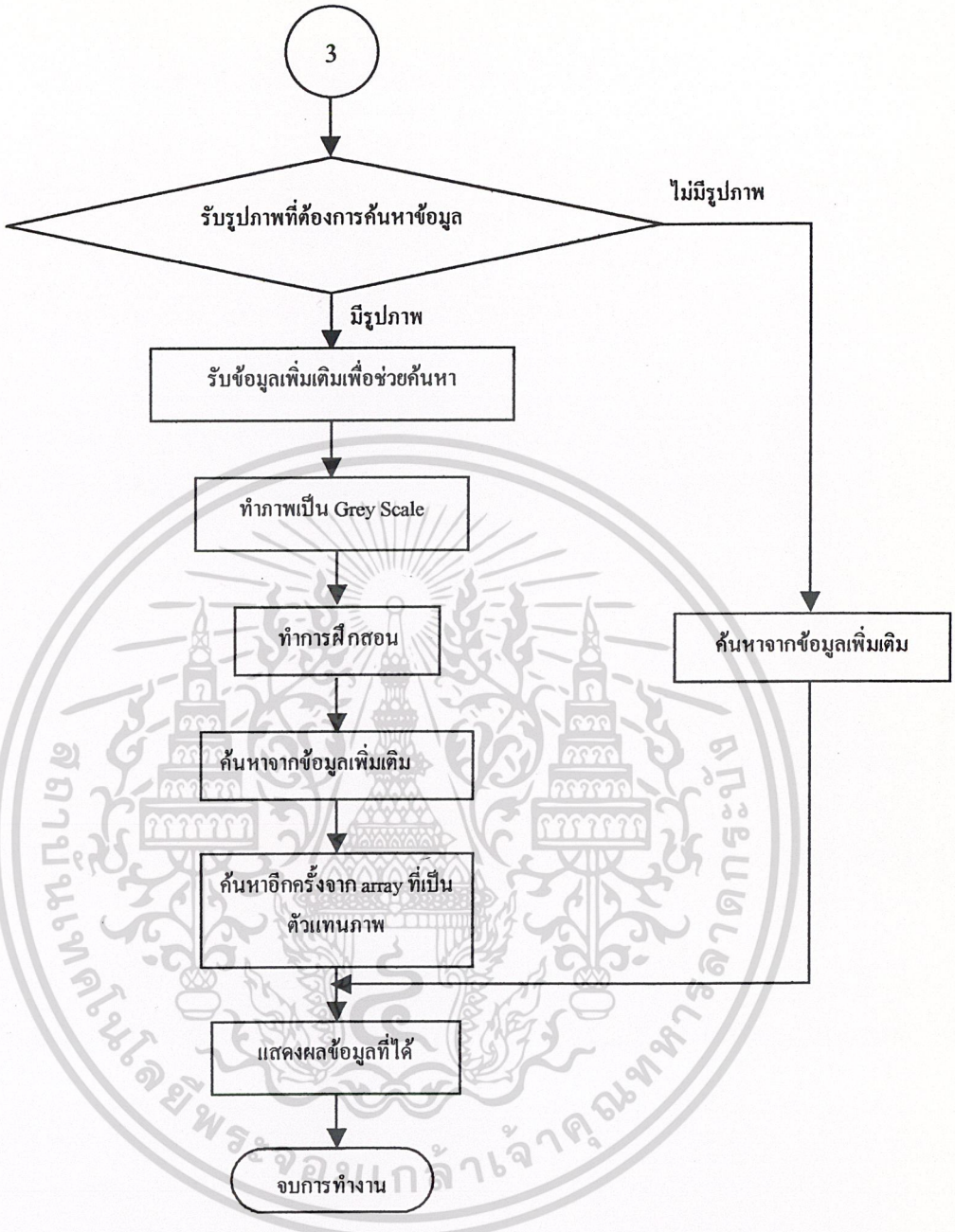

รูปที่ 4.8 หน้าจอการกรอกข้อมูลเพื่อเก็บลงฐานข้อมูล

**4.2.1.3 Search** เป็นปุ่มประมวลผลภาพและค้นหาเพื่อเรียกดูข้อมูลของรูปภาพนั้น ๆ ซึ่งปุ่มนี้จะใช้ ในกรณีที่ใช้รูปภาพเป็น keyword เพียงอย่างเดียว โดยมีลำดับการทำงานของโปรแกรมดังรูปที่ 4.9



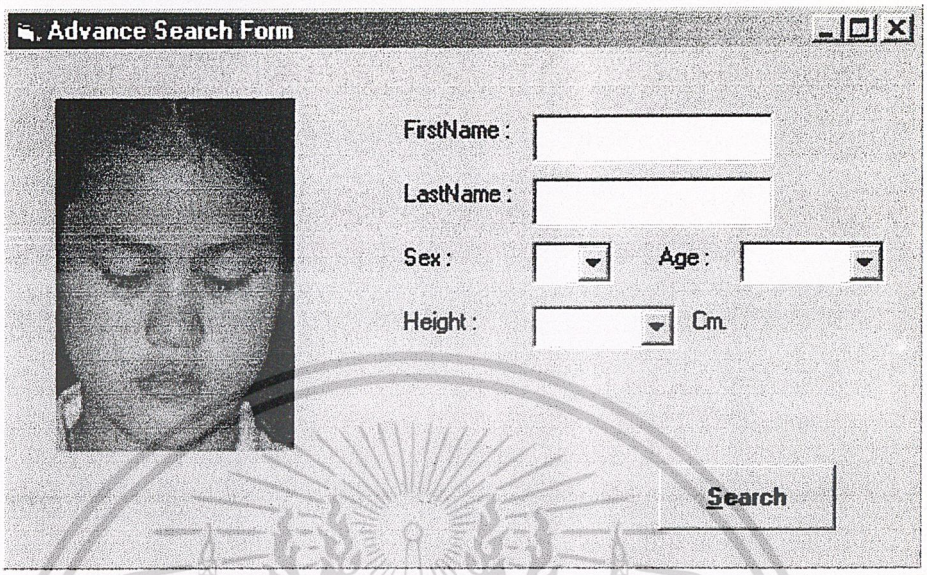

รูปที่ 4.9 ลำดับขั้นตอนการทำงานของโปรแกรมส่วนการค้นหาข้อมูลโดยใช้รูปภาพเพียงอย่างเดียว

**4.2.1.4 Advance Search** เป็นปุ่มที่ช่วยในการค้นหาโดยใช้ข้อมูลอื่น ๆ เพิ่มเติมด้วย ซึ่งมีลำดับ ขั้นตอนการทำงานของโปรแกรมดังรูปที่ 4.10



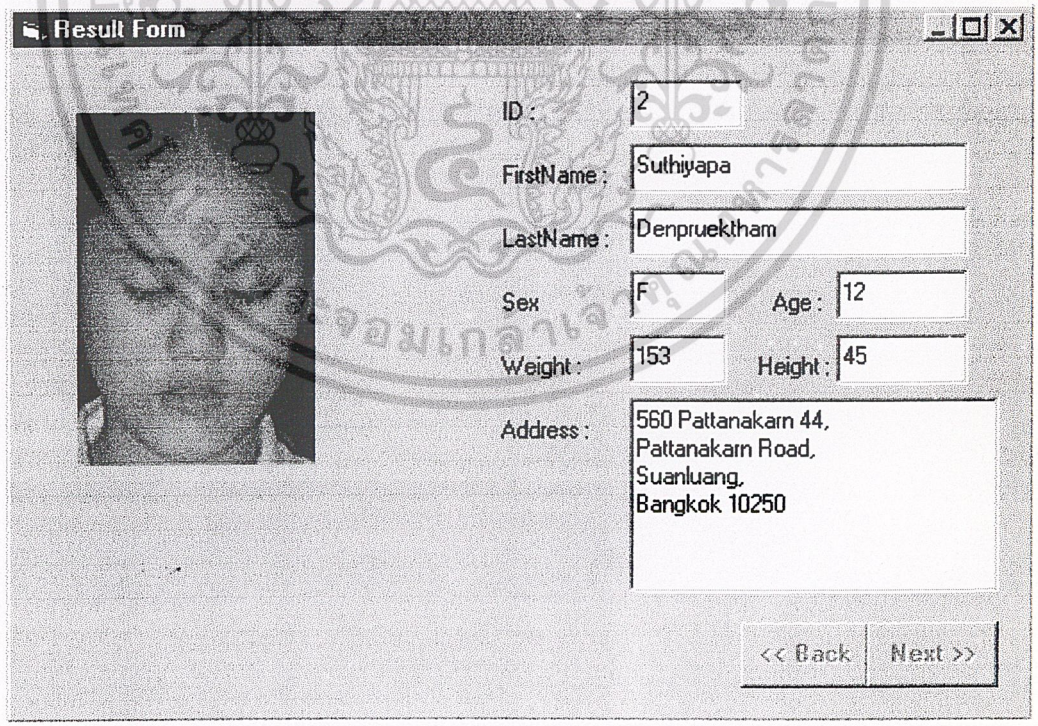

รูปที่ 4.10 ลำดับขั้นตอนการทำงานของโปรแกรมส่วนการค้นหาข้อมูลโดยใช้ข้อมูลหลายอย่าง

สำหรับหน้าจอรับข้อมูลเพิ่มเติมเพื่อช่วยในการค้นหา มีลักษณะดังรูปที่ 4.11




รูปที่ 4.11 หน้าจอรับข้อมูลเพิ่มเติมเพื่อช่วยในการค้นหา

และสำหรับหน้าจอแสดงผลข้อมูลที่ได้ มีลักษณะหน้าจอดังรูปที่ 4.12

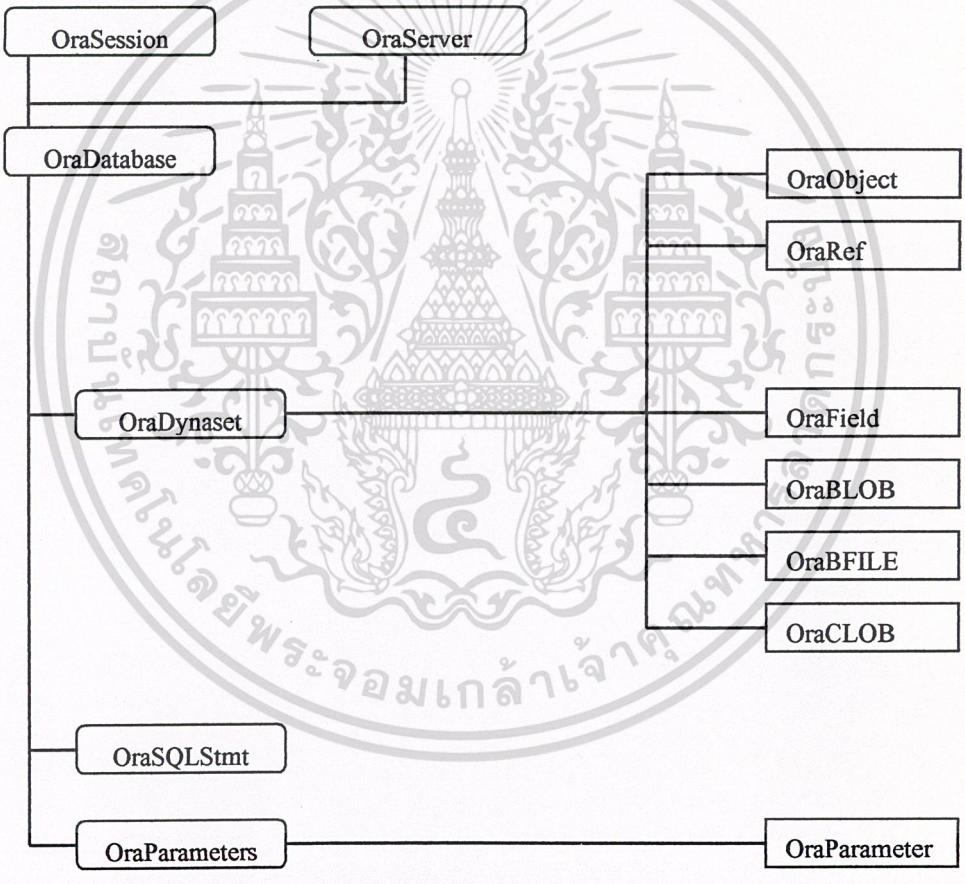



รูปที่ 4.12 หน้าจอแสดงผลข้อมูลที่ได้

### 4.2.2 ส่วนของระบบจัดการฐานข้อมูล

สำหรับในโครงงานนี้ได้เลือกใช้ Oracle 9i เป็น DBMS โดยใช้เทคโนโลยี OO4O คือ Oracle Objects For OLE เป็นตัวช่วยติดต่อระหว่างวิชวลเบสิกกับ Oracle

OO4O คือ เทคโนโลยีที่ถูกพัฒนาโดยบริษัท Oracle เพื่อรองรับการเข้าถึงและใช้งานฐานข้อมูล Oracle โดยเฉพาะเพื่อให้มีประสิทธิภาพสูงสุด การเขียนโปรแกรมด้วย OO4O นั้นใช้ Oracle In Process Automation Server (InProc Server) เป็นส่วนประกอบสำคัญที่ใช้ในการสร้างการติดต่อฐานข้อมูล ประมวลผลคำสั่ง SQL ตลอดจนการเรียกชุดคำสั่งที่ใช้ทำงานกับฐานข้อมูล




รูปที่ 4.13 ชนิดการจัดเก็บข้อมูลทั้งหมดใน Oracle

สำหรับในโครงงานนี้ได้ใช้ชนิดข้อมูลต่าง ๆ เหล่านี้เพียง 3 ชนิดเท่านั้นดังนี้ คือ

**1. OraSession Object** ใช้เป็นตัวแทนของ InProc Server ซึ่งจะประกอบไปด้วยวิธีที่ใช้สร้างการติดต่อ กับฐานข้อมูล

**2. OraDatabase Object** เป็นวิธีการที่จะประมวลคำสั่ง SQL จะถูกสร้างเมื่อมีการเรียกใช้ OpenDatabase Method ของ OraSession Object

**3. OraDynaset Object** เป็นวิธีการจัดการข้อมูลใน Recordset ไม่ว่าจะเป็นการเรียกดูหรือปรับปรุงข้อมูล จะถูกสร้างเมื่อมีการเรียกใช้ CreateDynaset Method ของ OraDatabase Object

สำหรับตารางที่ใช้จัดเก็บข้อมูลในโครงงานมีเพียง 1 ตารางเท่านั้น โดยมีโครงสร้างการจัดเก็บดังต่อ ไปนี้

<u>ชื่อตาราง</u> : Image

**คอลัมน์ที่ 1** : ID **ชนิดข้อมูล** : Number
**คอลัมน์ที่ 2** : ImagePath **ชนิดข้อมูล** : VarChar(20)
**คอลัมน์ที่ 3** : Data **ชนิดข้อมูล** : VarChar2(1000)
**คอลัมน์ที่ 4** : FirstName **ชนิดข้อมูล** : VarChar2(20)
**คอลัมน์ที่ 5** : LastName **ชนิดข้อมูล** : VarChar2(30)
**คอลัมน์ที่ 6** : Sex **ชนิดข้อมูล** : Char
**คอลัมน์ที่ 7** : Age **ชนิดข้อมูล** : Number
**คอลัมน์ที่ 8** : Weight **ชนิดข้อมูล** : Number
**คอลัมน์ที่ 9** : Height **ชนิดข้อมูล** : Number
**คอลัมน์ที่ 10** : Address **ชนิดข้อมูล** : VarChar2(100)

สำหรับในการเข้าถึงข้อมูลในฐานข้อมูลนั้นได้ใช้ SQL PLUS เป็นเครื่องมือช่วยในการจัดการกับฐานข้อมูล โดย SQL PLUS นี้เป็นโปรแกรมมาตรฐานที่มีมากับโปรแกรมระบบจัดการฐานข้อมูลของออราเคิลทุกรุ่นอยู่แล้ว

# บทที่ 5
# ผลการทดลอง

สำหรับข้อกำหนดของภาพอินพุทที่จะทำการรับเข้ามานั้น ต้องมีลักษณะดังต่อไปนี้

1. ต้องเป็นภาพที่ครอบเฉพาะบริเวณใบหน้าแล้ว
2. ต้องเป็นภาพที่มีปริมาณแสงไม่ต่างไปจากภาพที่เก็บอยู่ในฐานข้อมูลมากนัก
3. ต้องเป็นภาพที่มีลักษณะใบหน้าไม่ต่างไปจากฐานข้อมูลมากนัก

## 5.1 ประสิทธิภาพการรู้จำภาพใบหน้าคน

### 5.1.1 ผลของจำนวนข้อมูลในฐานข้อมูล

<u>วิธีทดลอง</u> : เพิ่มจำนวนข้อมูลในฐานข้อมูลจาก 3, 6 และ 10 ตามลำดับ พิจารณาความถูกต้องของการรู้จำภาพใบหน้าคน

ทั้งนี้ขนาดของภาพใบหน้าอินพุทควรจะมีขนาดพอ ๆ กับขนาดที่เก็บเป็นฐานข้อมูล และควบคุมสิ่งแวดล้อมทางแสง การรู้จำจึงจะมีประสิทธิภาพดีขึ้น

| จำนวนฐานข้อมูล | จำนวนครั้งที่ทดลอง | จำนวนครั้งที่ถูกต้อง | % ความถูกต้อง |
|---|---|---|---|
| 3 | 10 | 9 | 90% |
| 6 | 10 | 8 | 80% |
| 10 | 10 | 8 | 80% |

ตารางที่ 5.1 ผลของจำนวนข้อมูลในฐานข้อมูลที่ไม่เท่ากัน

<u>ผลการทดลอง</u> : ถ้าจำนวนภาพที่ใช้เป็นฐานข้อมูลเพื่อการรู้จำมีมาก ประสิทธิภาพในการรู้จำจะลดลงตามลำดับ

**5.1.2 ผลจากภาพใบหน้าอินพุทมีลักษณะแตกต่างจากภาพในฐานข้อมูล**

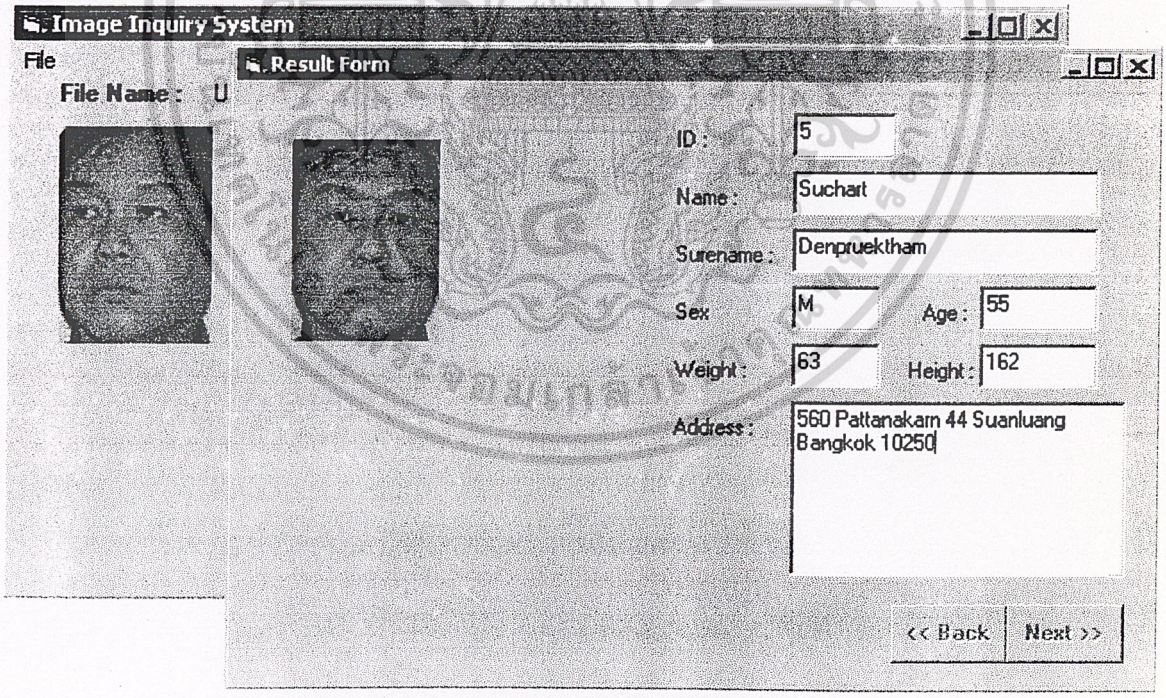
<u>วิธีทดลอง</u> : พิจารณาว่าหากภาพใบหน้าอินพุทมีการเปลี่ยนแปลงผิดไปจากภาพที่เก็บในฐานข้อมูล เช่น ใบหน้าเอียง ใบหน้ายิ้ม เป็นต้น โปรแกรมจะสามารถรู้จำได้อย่างถูกต้องคิดเป็นเปอร์เซ็นต์เท่าไร โดยทดสอบกับขนาดฐานข้อมูล 5 คน เพื่อลดความคลาดเคลื่อนจากจำนวนฐานข้อมูลที่มากขึ้น

1) เมื่อภาพใบหน้าเอียง

| คนที่ | จำนวนครั้งที่ทดลอง | จำนวนครั้งที่ถูกต้อง | % ความถูกต้อง | % ความถูกต้องเฉลี่ย |
|---|---|---|---|---|
| 1 | 10 | 8 | 80% | 75% |
| 2 | 10 | 7 | 70% | |

ตารางที่ 5.2 ผลจากภาพใบหน้าอินพุทที่เอียงแตกต่างจากภาพในฐานข้อมูล

<u>ผลการทดลอง</u> : ภาพใบหน้าเอียงทำให้องค์ประกอบของภาพเปลี่ยนไป ความถูกต้องในการรู้จำจึงลดลง



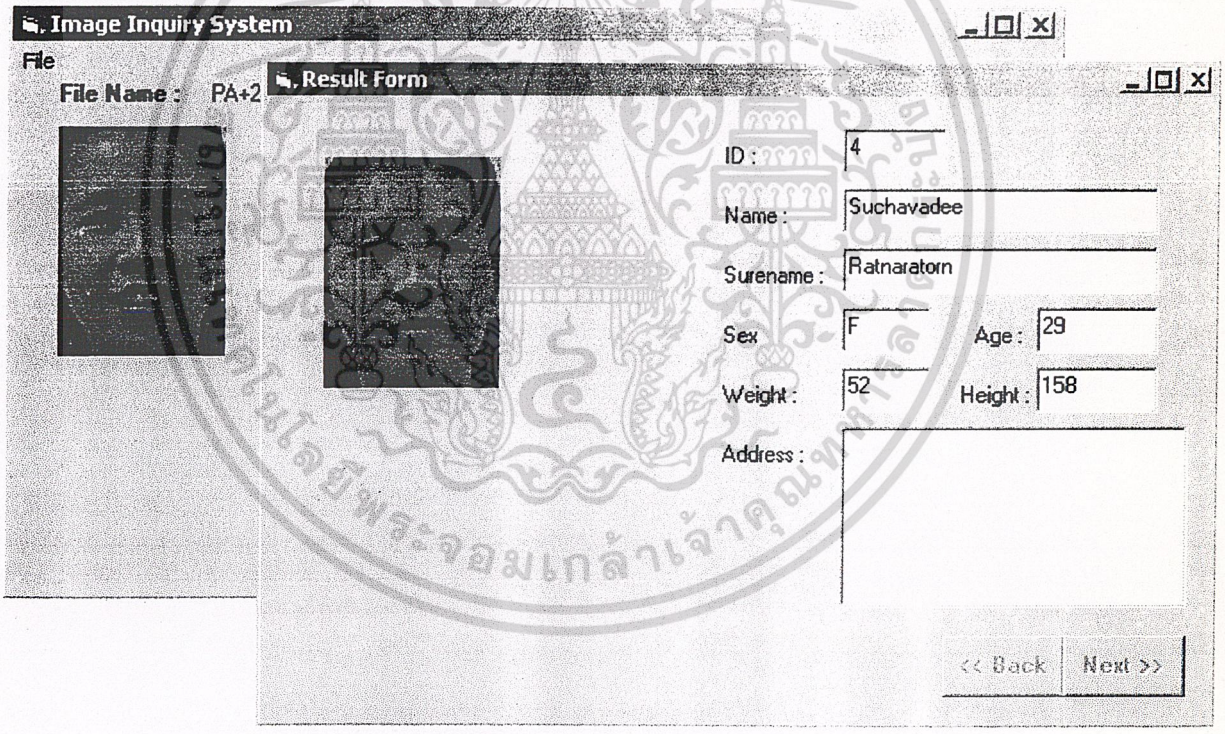

รูปที่ 5.1 แสดงการรู้จำภาพใบหน้าอินพุทที่เอียงแตกต่างจากฐานข้อมูล

2) เมื่อภาพใบหน้ายิ้ม

| คนที่ | จำนวนครั้งที่ทดลอง | จำนวนครั้งที่ถูกต้อง | % ความถูกต้อง | % ความถูกต้องเฉลี่ย |
|---|---|---|---|---|
| 1 | 10 | 9 | 90% | 85% |
| 2 | 10 | 8 | 80% | |

ตารางที่ 5.3 ผลจากภาพใบหน้าอินพุทที่ยิ้มแตกต่างจากภาพในฐานข้อมูล

ผลการทดลอง : ภาพใบหน้าที่ยิ้มทำให้องค์ประกอบของภาพเปลี่ยนไปเพียงเล็กน้อย ดังนั้น ความถูกต้องในการรู้จำจึงลดลงเพียงเล็กน้อยเท่านั้น




รูปที่ 5.2 แสดงการรู้จำภาพใบหน้าอินพุทที่ยิ้มแตกต่างจากฐานข้อมูล

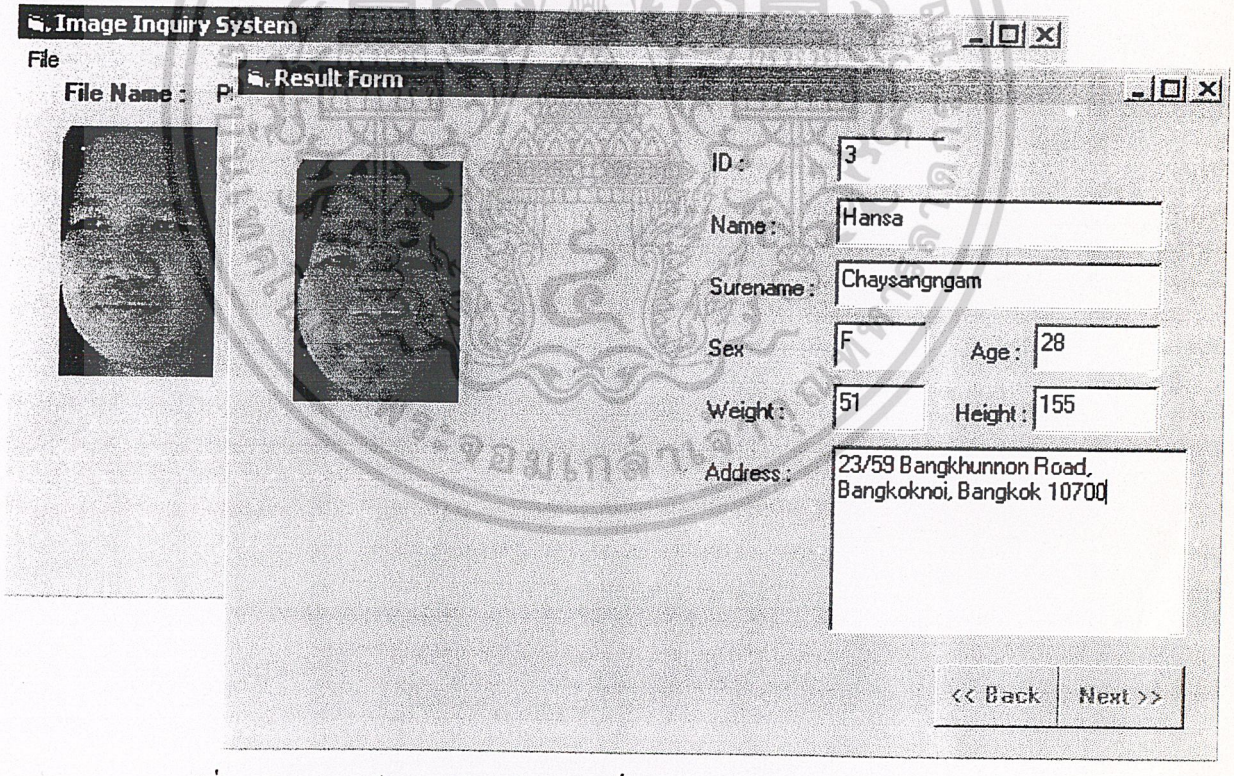
**5.1.3 เมื่อสภาพแวดล้อมเปลี่ยน**

<u>วิธีทดลอง</u> : พิจารณาว่าหากสภาพแวดล้อมเปลี่ยนแปลง โดยในการทดลองนี้จะให้สภาพแสงเปลี่ยนแปลง โปรแกรมจะสามารถรู้จำได้อย่างถูกต้องคิดเป็นเปอร์เซ็นต์เท่าไร โดยทดสอบกับขนาดฐานข้อมูล 5 คน

| คนที่ | จำนวนครั้งที่ทดลอง | จำนวนครั้งที่ถูกต้อง | % ความถูกต้อง | % ความถูกต้องเฉลี่ย |
|---|---|---|---|---|
| 1 | 10 | 3 | 30% | 35% |
| 2 | 10 | 4 | 40% | |

ตารางที่ 5.4 ผลจากภาพเมื่อมีความสว่างต่างจากภาพในฐานข้อมูล

<u>ผลการทดลอง</u> : จะเห็นได้ว่าเมื่อความสว่างเปลี่ยนแปลงจะทำให้องค์ประกอบแสงของภาพเปลี่ยนไปค่อนข้างมาก ดังนั้นจึงมีผลต่อการรู้จำมาก




รูปที่ 5.3 แสดงการรู้จำภาพใบหน้าอินพุทที่มีแสงแตกต่างจากฐานข้อมูล

# บทที่ 6
# บทสรุปและข้อเสนอแนะ

## 6.1 บทสรุป

จากเป้าหมายของโครงงานคือ ให้โปรแกรมสามารถรู้จำภาพใบหน้าคนได้อย่างมีประสิทธิภาพ จากผลการทดลองที่ได้ จะเห็นว่าสิ่งที่มีผลกระทบกับประสิทธิภาพในการรู้จำทำให้การรู้จำมีประสิทธิภาพลดต่ำลง ได้แก่

1. เรื่องของสภาพแวดล้อม คือ แสง ทำให้ความสว่างของภาพไม่เท่ากัน
2. จำนวนคนในฐานข้อมูล ยิ่งมีมากจะทำให้การรู้จำมีประสิทธิภาพลดต่ำลง
3. ความแตกต่างของขนาดภาพใบหน้าที่เก็บในฐานข้อมูลกับภาพอินพุท
4. ภาพใบหน้าที่มีลักษณะแตกต่างจากฐานข้อมูล เช่น ภาพใบหน้าเอียงหรือยิ้ม

สำหรับในข้อ 3 และ 4 ยิ่งภาพใบหน้าอินพุทมีความแตกต่างของขนาดหรือมีลักษณะที่แตกต่างไปจากภาพใบหน้าในฐานข้อมูลมากเท่าไรจะยิ่งทำให้การรู้จำมีประสิทธิภาพน้อยลงไปเท่านั้น

## 6.2 แนวทางการแก้ไข

1. สำหรับในเรื่องของสภาพแสงจะต้องควบคุมไม่ให้ภาพอินพุทแตกต่างไปจากภาพในฐานข้อมูลมากนัก
2. ทำการเพิ่มจำนวนนิวรอลให้มากขึ้น เพื่อให้ได้ความละเอียดของข้อมูลเพิ่มขึ้น
3. หาวิธีการเก็บค่าที่เป็นตัวแทนรูปภาพวิธีอื่น ๆ เข้าช่วยด้วย จะทำให้การรู้จำมีประสิทธิภาพมากยิ่งขึ้นและลดข้อจำกัดของระบบลงได้

# ภาคผนวก

**Module :**

```
Type DifArray
   arr(0 To 31, 0 To 31) As Integer
End Type
Type DifTbl
   arr(0 To 9) As DifArray
   val As Integer
End Type

Global Const size = 10
'Global eliminated_image_array(maxx, maxy) As
Integer    ' For new value that eliminated noise
already
'Global detected_image_array(maxx, maxy) As
Integer
Global represented_array(0 To 31, 0 To 31) As
Integer
Global FileName As String
Global Sql_Command As String
Global arr(size - 1, size - 1) As Integer
Global maxx, maxy As Integer
Global color() As Integer
Global OraSession As OraSession
Global OraDatabase As OraDatabase
Global OraDynaset  As OraDynaset
```

**MainForm :**

```
Dim OraSession As OraSession
Dim OraDatabase As OraDatabase
Dim OraDynaset  As OraDynaset
Dim maskX, maskY As Integer
Dim same As Integer
Dim FileData, Data As String
Dim Seq As String

Private Sub ClearImage_Click()
   Picture1.Cls
   LabelNameofFile.Caption = "-"
End Sub

Private Sub Command1_Click()
   FrmAdvancedSearch.Picture1 =
MainFrm.Picture1
   FrmAdvancedSearch.Show
End Sub

Private Sub ExitMenu_Click()
   End
End Sub

Private Sub Form_Load()
Dim ClickStatus As Integer        ' if = 0 means
"Keep" was clicked, if = 1 means "Processing".
'Create the OraSession Object.
   Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
'Create the OraDatabase Object by opening a
connection to Oracle
   Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
'Create the OraDynaset Object.
   Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)
End Sub

Sub KeepintoDB_Click()
```

```
    ClickStatus = 0
    FrmKeepIntoDB.Picture1 = MainFrm.Picture1
    FrmKeepIntoDB.Show
End Sub

Private Sub OpenMenu_Click()
On Error GoTo Killer
    'CommonDialog1.Filter = "Image
File(*.jpeg)|*.jpeg "
    CommonDialog1.ShowOpen

    Picture1 =
LoadPicture(CommonDialog1.FileName)
    FileName = CommonDialog1.FileName
    Exit Sub
Killer:
    MsgBox "Error!", 0 & 16, "Error:"

End Sub

Private Sub Processing_Click()
    ClickStatus = 1
        'Create the OraSession Object.
        '   Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
        'Create the OraDatabase Object by opening
a connection to Oracle
        '   Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
        'Create the OraDynaset Object.
         Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)
    Call searchData
    Sql_Command = "id='" & same & "'"
        OraDynaset.FindFirst Sql_Command
        FileName = OraDynaset.Fields("path")
        FrmShowResult.Picture1.Picture =
LoadPicture(FileName)
        FrmShowResult.txtID.Text =
OraDynaset.Fields("id")
        FrmShowResult.txtName.Text =
OraDynaset.Fields("name")
        FrmShowResult.txtSurename.Text =
OraDynaset.Fields("surename")
        FrmShowResult.txtSex.Text =
OraDynaset.Fields("sex")
        FrmShowResult.txtAge.Text =
OraDynaset.Fields("age")
        FrmShowResult.txtHeight.Text =
OraDynaset.Fields("Heigth")
        FrmShowResult.txtWieght.Text =
OraDynaset.Fields("Weigth")
        FrmShowResult.Show
End Sub
Sub search()
Dim fso As New FileSystemObject
Dim objts As TextStream
Dim objFil As File
Dim Seq As String
Dim SeqIn, most, found, num As Integer
Dim Count As Integer
Dim Sql_Criteria, blank As String

    Call getData        'get array representation of
```

```
                'this picture => array(size*size)

   Set objts = fso.OpenTextFile("c:\sequence.txt",
ForReading, False)
   Seq = Trim(objts.ReadLine)     'open file
sequence for reading
                'the amount of picture in
database
   SeqIn = val(Seq)
   most = 0
      'Create the OraSession Object.
      '   Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
      'Create the OraDatabase Object by opening a
connection to Oracle
      '   Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
      'Create the OraDynaset Object.
      '   Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)
      'SeqID = 1


   For b = 1 To SeqIn
      Sql_Criteria = "id =' " & b & "'"
      OraDynaset.FindNext Sql_Criteria
      FileData = OraDynaset.Fields("Data")
'FileData = pathname
                'of file data
representation
      Set objts = fso.OpenTextFile(FileData,
ForReading, False)
      Seq = Trim(objts.ReadLine)
      Count = 0
      For x = 0 To size - 1
         For y = 0 To size - 1
            If x * y <> 81 Then
               blank = " "
               found = InStr(Seq, blank)
               num = Int(Left(Seq, found - 1))
               Seq = Trim(Right(Seq, Len(Seq) -
found + 1))
            Else
               num = Trim(Seq)
            End If
            If arr(x, y) < num + 30 And arr(x, y) >
num - 30 Then
               Count = Count + 1      'count is
number of pixel of image
                'in DB that familiar
with the input image
            End If
         Next y
      Next x
      If Count > most Then
         most = Count          'most is number of
count that have maximum value
         same = b              'same is id of image that
have most similar
      End If
   Next b
End Sub


Public Sub getData()
```

```
Dim x, y  As Integer

   maxx = Picture1.ScaleWidth
   maxy = Picture1.ScaleHeight

   ReDim color(maxx, maxy)        'keep gray scale
value of this image

   For i = 0 To maxx - 1
      For j = 0 To maxy - 1
         tmp = CStr(Hex(Picture1.Point(i, j)))
         tmpR = Mid(tmp, 1, 2)
         tmpG = Mid(tmp, 3, 2)
         tmpB = Mid(tmp, 5, 2)

         tempR = val("&H" & tmpR)
         tempG = val("&H" & tmpG)
         tempB = val("&H" & tmpB)
         Temp = Int((tempR + tempG + tempB) / 3)
         color(i, j) = Temp
      Next j
   Next i

   maskX = Int(maxx / size)
   maskY = Int(maxy / size)

   For i = 0 To size - 1
      x = i * maskX
      For j = 0 To size - 1
         y = j * maskY
         arr(i, j) = GetRepresentBO(x, y)
      Next j
   Next i
```

```
   Dim fso As New FileSystemObject
   Dim objFil As File
   Dim objts As TextStream

   Dim tee As String
   FileData = Left(FileName, Len(FileName) - 4)
& ".txt"   'FileData => path name of
                                          'array
representation
   fso.CreateTextFile (FileData)
   Set objFil = fso.GetFile(FileData)
   Set objts =
objFil.OpenAsTextStream(ForWriting)
   For i = 0 To size - 1
      For j = 0 To size - 1
         objts.Write arr(i, j) & " "
      Next j
   Next i
   objts.Close
End Sub

Private Function GetRepresentBO(X1, Y1 As
Integer) As Integer
Dim Temp As Double
Dim tmp As String
   Temp = 0
   For n = X1 To maskX + X1 - 1
      For m = Y1 To maskY + Y1 - 1
         Temp = Temp + color(n, m)
      Next m
   Next n
   GetRepresentBO = Int(Temp / (maskX *
maskY))
End Function
```

```
Sub searchData()
Dim fso As New FileSystemObject
Dim objts As TextStream
Dim objFil As File
'Dim Seq As String
Dim SeqIn, most, found, num As Integer
Dim blank As String
Dim Count As Integer

  Call getFileData        'get array representation
of
                          'this picture =>
array(size*size)

  Set objts = fso.OpenTextFile("c:\sequence.txt",
ForReading, False)
  Seq = Trim(objts.ReadLine)    'open file
sequence for reading the amount of picture in
database
  SeqIn = val(Seq)
  most = 0
    'Create the OraSession Object.
    '   Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
    'Create the OraDatabase Object by opening a
connection to Oracle
    '   Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
    'Create the OraDynaset Object.
    '   Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)
    'SeqID = 1


  For b = 1 To SeqIn
    Sql_Criteria = "id =' " & b & "'"
    OraDynaset.FindNext Sql_Criteria
    Seq = OraDynaset.Fields("Data")

    'Set objts = fso.OpenTextFile(FileData,
ForReading, False)
    'Seq = Trim(objts.ReadLine)
    Count = 0
    For x = 0 To size - 1
      For y = 0 To size - 1
        If x * y <> 81 Then
        blank = " "
        found = InStr(Seq, blank)
        num = Int(Left(Seq, found - 1))
        Seq = Trim(Right(Seq, Len(Seq) -
found + 1))
        Else
        num = Trim(Seq)
        End If
        If arr(x, y) < num + 30 And arr(x, y) >
num - 30 Then
          Count = Count + 1
        End If
      Next y
    Next x
    MsgBox Count
    If Count > most Then
```

```
        most = Count
        same = b
      End If
    Next b
End Sub

Public Sub getFileData()
Dim x, y  As Integer

   maxx = Picture1.ScaleWidth
   maxy = Picture1.ScaleHeight

   ReDim color(maxx, maxy)

   For i = 0 To maxx - 1
     For j = 0 To maxy - 1
       tmp = CStr(Hex(Picture1.Point(i, j)))
       tmpR = Mid(tmp, 1, 2)
       tmpG = Mid(tmp, 3, 2)
       tmpB = Mid(tmp, 5, 2)

       tempR = val("&H" & tmpR)
       tempG = val("&H" & tmpG)
       tempB = val("&H" & tmpB)
       Temp = Int((tempR + tempG + tempB) / 3)
       color(i, j) = Temp
     Next j
   Next i

   maskX = Int(maxx / size)
   maskY = Int(maxy / size)

   Data = ""
   For i = 0 To size - 1
     x = i * maskX
     For j = 0 To size - 1
       y = j * maskY
       arr(i, j) = GetRepresentFile(x, y)
       Data = Data & arr(i, j) & " "
     Next j
   Next i

End Sub

Private Function GetRepresentFile(X1, Y1 As
Integer) As Integer
Dim Temp As Double
   Temp = 0
   For n = X1 To maskX + X1 - 1
     For m = Y1 To maskY + Y1 - 1
       Temp = Temp + color(n, m)
     Next m
   Next n
   GetRepresentFile = Int(Temp / (maskX *
maskY))
End Function
```

**FrmShowResult :**

```
Private Sub Form_Load()
   'Create the OraSession Object.
   Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
'Create the OraDatabase Object by opening a
connection to Oracle
   Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
'Create the OraDynaset Object.
```

```
   Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)
   'Sql_Command = GetSql_Command
   'Trim (Sql_Command)
End Sub

Private Sub NextBtn_Click()
   OraDynaset.FindNext Sql_Command
    If OraDynaset.NoMatch Then
      MsgBox "Can not Find Record"
   Else
      FileName = OraDynaset.Fields("path")
      Picture1 = LoadPicture(FileName)
      txtID.Text = OraDynaset.Fields("id")
      txtName.Text = OraDynaset.Fields("name")
      txtSurename.Text =
OraDynaset.Fields("surename")
      txtSex.Text = OraDynaset.Fields("sex")
      txtAge.Text = OraDynaset.Fields("age")
      txtHeight.Text =
OraDynaset.Fields("Heigth")
      txtWieght.Text =
OraDynaset.Fields("Weigth")
      'FrmShowResult.Show
   End If
End Sub
```

**FrmKeepintoDB :**

```
Dim Sql, Seq As String
Dim SeqIn As Integer
Dim fso As New FileSystemObject
Dim objts As TextStream
Dim objFil As File
Dim color() As Integer
Dim maskX, maskY As Integer
Dim FileData, Data As String

Private Sub btnSave_Click()

   Set objts = fso.OpenTextFile("c:\sequence.txt",
ForReading, False)   'Refer to file

'sequence.txt for read
   Seq = Trim(objts.ReadLine)
   SeqIn = val(Seq) + 1

   OraDynaset.AddNew

   OraDynaset.Fields("id").Value = SeqIn
   OraDynaset.Fields("name").Value =
txtName.Text
   OraDynaset.Fields("surename").Value =
txtSurename.Text
   OraDynaset.Fields("Path").Value = FileName
   OraDynaset.Fields("sex").Value = txtSex.Text
   OraDynaset.Fields("Age").Value = txtAge.Text
   OraDynaset.Fields("Weigth").Value =
txtWeigth.Text
   OraDynaset.Fields("Heigth").Value =
txtHeigth.Text
   OraDynaset.Fields("Address") =
txtAddress.Text
   Call getFileData
   OraDynaset.Fields("Data").Value = Data

   OraDynaset.Update
   OraDynaset.Refresh
```

```
    MsgBox "Success"
    Set objFil = fso.GetFile("c:\sequence.txt")
    Set objts =
objFil.OpenAsTextStream(ForWriting)
    objts.Write SeqIn
    objts.Close
    Unload Me
End Sub

Public Sub getData()
Dim x, y  As Integer

    maxx = Picture1.ScaleWidth
    maxy = Picture1.ScaleHeight

    ReDim color(maxx, maxy)

    For i = 0 To maxx - 1
      For j = 0 To maxy - 1
        tmp = CStr(Hex(Picture1.Point(i, j)))
        tmpR = Mid(tmp, 1, 2)
        tmpG = Mid(tmp, 3, 2)
        tmpB = Mid(tmp, 5, 2)

        tempR = val("&H" & tmpR)
        tempG = val("&H" & tmpG)
        tempB = val("&H" & tmpB)
        Temp = Int((tempR + tempG + tempB) / 3)
        color(i, j) = Temp
      Next j
    Next i

    maskX = Int(maxx / size)
    maskY = Int(maxy / size)

    For i = 0 To size - 1
      x = i * maskX
      For j = 0 To size - 1
        y = j * maskY
        arr(i, j) = GetRepresent(x, y)
      Next j
    Next i

    Dim fso As New FileSystemObject
    Dim objFil As File
    Dim objts As TextStream

    Dim tee As String
    FileData = Left(FileName, Len(FileName) - 4)
& ".txt"
    fso.CreateTextFile (FileData)
    Set objFil = fso.GetFile(FileData)
    Set objts =
objFil.OpenAsTextStream(ForWriting)
    For i = 0 To size - 1
      For j = 0 To size - 1
        objts.Write arr(i, j) & " "
      Next j
    Next i
    objts.Close
End Sub

Private Function GetRepresent(X1, Y1 As Integer)
As Integer
Dim Temp As Double
Dim tmp As String
    Temp = 0
```

```
  For n = X1 To maskX + X1 - 1
    For m = Y1 To maskY + Y1 - 1
      Temp = Temp + color(n, m)
    Next m
  Next n
  GetRepresent = Int(Temp / (maskX * maskY))
End Function

Private Sub Form_Load()
'Create the OraSession Object.
  Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
'Create the OraDatabase Object by opening a
connection to Oracle
  Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
'Create the OraDynaset Object.
  Set OraDynaset =
OraDatabase.CreateDynaset("select * from
Project", &H0&)
End Sub

Sub searchData()
Dim fso As New FileSystemObject
Dim objts As TextStream
Dim objFil As File
Dim Seq As String
Dim SeqIn, most, found, num As Integer
Dim blank As String
Dim Count As Integer

  Call getFileData          'get array representation
of
                            'this picture =>
array(size*size)

  Set objts = fso.OpenTextFile("c:\sequence.txt",
ForReading, False)
  Seq = Trim(objts.ReadLine)     'open file
sequence for reading the amount of picture in
database
  SeqIn = val(Seq)
  most = 0
    'Create the OraSession Object.
    '  Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
    'Create the OraDatabase Object by opening a
connection to Oracle
    '  Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
    'Create the OraDynaset Object.
    '  Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)
    'SeqID = 1

  For b = 1 To SeqIn
    Sql_Criteria = "id =' " & b & "'"
    OraDynaset.FindNext Sql_Criteria
    FileData = OraDynaset.Fields("Data")

    Set objts = fso.OpenTextFile(FileData,
ForReading, False)
    Seq = Trim(objts.ReadLine)
    Count = 0
```

```
    For x = 0 To size - 1
      For y = 0 To size - 1
        If x * y <> 81 Then
          blank = " "
          found = InStr(Seq, blank)
          num = Int(Left(Seq, found - 1))
          Seq = Trim(Right(Seq, Len(Seq) -
found + 1))
        Else
          num = Trim(Seq)
        End If
        If arr(x, y) < num + 30 And arr(x, y) >
num - 30 Then
          Count = Count + 1
        End If
      Next y
    Next x
    If Count > most Then
      most = Count
      same = b
    End If
  Next b
End Sub

Public Sub getFileData()
Dim x, y  As Integer

  maxx = Picture1.ScaleWidth
  maxy = Picture1.ScaleHeight

  ReDim color(maxx, maxy)

  For i = 0 To maxx - 1
    For j = 0 To maxy - 1
      tmp = CStr(Hex(Picture1.Point(i, j)))
      tmpR = Mid(tmp, 1, 2)
      tmpG = Mid(tmp, 3, 2)
      tmpB = Mid(tmp, 5, 2)

      tempR = val("&H" & tmpR)
      tempG = val("&H" & tmpG)
      tempB = val("&H" & tmpB)
      Temp = Int((tempR + tempG + tempB) / 3)
      color(i, j) = Temp
    Next j
  Next i

  maskX = Int(maxx / size)
  maskY = Int(maxy / size)

  Data = ""
  For i = 0 To size - 1
    x = i * maskX
    For j = 0 To size - 1
      y = j * maskY
      arr(i, j) = GetRepresentFile(x, y)
      Data = Data & arr(i, j) & " "
    Next j
  Next i

End Sub

Private Function GetRepresentFile(X1, Y1 As
Integer) As Integer
Dim Temp As Double
  Temp = 0
  For n = X1 To maskX + X1 - 1
    For m = Y1 To maskY + Y1 - 1
```

```
        Temp = Temp + color(n, m)
      Next m
    Next n
    GetRepresentFile = Int(Temp / (maskX *
maskY))
End Function
```

**FrmAdvanceSearch :**

```
Dim OraDyna As OraDynaset
Dim Command, Sql As String
Dim maskX, maskY As Integer
Dim same As Integer
Dim Data As String
Dim Seq As String

Private Sub cmdSearch_Click()
    Sql_Command = GetSql_Command

    Trim (Sql_Command)
    Sql = "select count(id) as amount from project
where " & Sql_Command

    Set OraDyna = OraDatabase.CreateDynaset(Sql,
&H0&)

    OraDynaset.FindFirst Sql_Command
     If OraDynaset.NoMatch Then
       MsgBox "Can not Find Record"
    ElseIf Picture1.Picture = None Then
       FileName = OraDynaset.Fields("path")
       FrmShowResult.Picture1.Picture =
LoadPicture(FileName)
       FrmShowResult.txtID.Text =
OraDynaset.Fields("id")
       FrmShowResult.txtName.Text =
OraDynaset.Fields("name")
       FrmShowResult.txtSureName.Text =
OraDynaset.Fields("surename")
       FrmShowResult.txtSex.Text =
OraDynaset.Fields("sex")
       FrmShowResult.txtAge.Text =
OraDynaset.Fields("age")
       FrmShowResult.txtHeight.Text =
OraDynaset.Fields("Heigth")
       FrmShowResult.txtWieght.Text =
OraDynaset.Fields("Weigth")
       FrmShowResult.Show
    Else
       Call search
       Sql_Command = "id='" & same & "'"
       OraDynaset.FindFirst Sql_Command
       FileName = OraDynaset.Fields("path")
       FrmShowResult.Picture1.Picture =
LoadPicture(FileName)
       FrmShowResult.txtID.Text =
OraDynaset.Fields("id")
       FrmShowResult.txtName.Text =
OraDynaset.Fields("name")
       FrmShowResult.txtSureName.Text =
OraDynaset.Fields("surename")
       FrmShowResult.txtSex.Text =
OraDynaset.Fields("sex")
       FrmShowResult.txtAge.Text =
OraDynaset.Fields("age")
       FrmShowResult.txtHeight.Text =
OraDynaset.Fields("Heigth")
       FrmShowResult.txtWieght.Text =
OraDynaset.Fields("Weigth")
```

```
    FrmShowResult.Show
  End If
End Sub

Function GetSql_Command() As String
  Command = ""
  If txtName.Text <> "" Then
    Command = "name ='" & txtName.Text & "'"
  ElseIf txtSureName.Text <> "" Then
    If Command <> "" Then
      Command = Command & "and surename =
'" & txtSureName.Text & "'"
    Else: Command = "surename = '" &
txtSureName.Text & "'"
    End If
  ElseIf cmbSex.Text <> "" Then
    If Command <> "" Then
      Command = Command & "and sex = '" &
cmbSex.Text & "'"
    Else: Command = "sex = '" & cmbSex.Text &
"'"
    End If
  ElseIf cmbAge.Text <> "" Then
    Call getAge
  ElseIf cmbHeight <> "" Then
    Call getHeight
  End If
  GetSql_Command = Command
End Function

Private Sub getAge()
Dim strAge  As String
Dim valAge As Integer
  strAge = cmbAge.Text
  valAge = val(Left(strAge, 2))
  For i = 0 To 4
    If Command <> "" Then
      Command = Command & "or age = '" &
Str(valAge + i) & "'"
    Else: Command = "age = '" & Str(valAge + i)
& "'"
    End If
  Next i
End Sub

Private Sub getHeight()
Dim strHeight  As String
Dim valHeight As Integer
  strHeight = cmbHeight.Text
  valHeight = val(Left(strHeight, 3))
  For i = 0 To 4
    If Command <> "" Then
      Command = Command & "or heigth = '" &
Str(valHeight + i) & "'"
    Else: Command = "heigth = '" &
Str(valHeight + i) & "'"
    End If
  Next i
End Sub

Sub search()
Dim fso As New FileSystemObject
Dim objts As TextStream
Dim objFil As File
Dim Seq As String
Dim SeqIn, most, found, num As Integer
Dim FileData As String
Dim blank As String
```

```
'Dim Seq As String
Dim SeqIn, most, found, num As Integer
Dim blank As String
Dim Count As Integer

   Call getFileData          'get array representation
of
                             'this picture =>
array(size*size)

   Set objts = fso.OpenTextFile("c:\sequence.txt",
ForReading, False)
   Seq = Trim(objts.ReadLine)     'open file
sequence for reading the amount of picture in
database
   SeqIn = val(Seq)
   most = 0
      'Create the OraSession Object.
      '   Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
      'Create the OraDatabase Object by opening a
connection to Oracle
      '   Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
      'Create the OraDynaset Object.
      '   Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)
      'SeqID = 1


   For b = 1 To SeqIn
      Sql_Criteria = "id =' " & b & "'"
      OraDynaset.FindNext Sql_Criteria
      Seq = OraDynaset.Fields("Data")

      'Set objts = fso.OpenTextFile(FileData,
ForReading, False)
      'Seq = Trim(objts.ReadLine)
      Count = 0
      For x = 0 To size - 1
        For y = 0 To size - 1
          If x * y <> 81 Then
            blank = " "
            found = InStr(Seq, blank)
            num = Int(Left(Seq, found - 1))
            Seq = Trim(Right(Seq, Len(Seq) -
found + 1))
          Else
            num = Trim(Seq)
          End If
          If arr(x, y) < num + 30 And arr(x, y) >
num - 30 Then
            Count = Count + 1
          End If
        Next y
      Next x
      If Count > most Then
        most = Count
        same = b
      End If
    Next b
End Sub

Public Sub getFileData()
Dim x, y  As Integer
```

```
    Next i

    maskX = Int(maxx / size)
    maskY = Int(maxy / size)

    For i = 0 To size - 1
      x = i * maskX
      For j = 0 To size - 1
        y = j * maskY
        arr(i, j) = GetRepresentBO(x, y)
        Debug.Print arr(i, j)
      Next j
    Next i

    Dim fso As New FileSystemObject
    Dim objFil As File
    Dim objts As TextStream

    Dim tee As String
    FileData = Left(FileName, Len(FileName) - 4)
& ".txt"
    fso.CreateTextFile (FileData)
    Set objFil = fso.GetFile(FileData)
    Set objts =
objFil.OpenAsTextStream(ForWriting)
    For i = 0 To size - 1
      For j = 0 To size - 1

        objts.Write arr(i, j) & " "
      Next j
    Next i
    objts.Close
End Sub
```

```
Private Function GetRepresentBO(X1, Y1 As
Integer) As Integer
Dim Temp As Double
Dim tmp As String
  Temp = 0
  For n = X1 To maskX + X1 - 1
    For m = Y1 To maskY + Y1 - 1
      Temp = Temp + color(n, m)
    Next m
  Next n
  GetRepresentBO = Int(Temp / (maskX *
maskY))
End Function

Private Sub Form_Load()
'Create the OraSession Object.
  Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
'Create the OraDatabase Object by opening a
connection to Oracle
  Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
'Create the OraDynaset Object.
  Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)

End Sub

Sub searchData()
Dim fso As New FileSystemObject
Dim objts As TextStream
Dim objFil As File
```

```
Dim Count As Integer

   Call getData

   Set objts = fso.OpenTextFile("c:\sequence.txt",
ForReading, False)
   Seq = Trim(objts.ReadLine)
   SeqIn = val(Seq)
   most = 0

   For b = 1 To SeqIn
      Sql_Criteria = "id =' " & b & "'"
      OraDynaset.FindNext Sql_Criteria
      FileData = OraDynaset.Fields("Data")

      Set objts = fso.OpenTextFile(FileData,
ForReading, False)
      Seq = Trim(objts.ReadLine)
      Count = 0
      For x = 0 To size - 1
        For y = 0 To size - 1
          If x * y <> 81 Then
            blank = " "
            found = InStr(Seq, blank)
            num = Int(Left(Seq, found - 1))
            Seq = Trim(Right(Seq, Len(Seq) -
found + 1))
          Else
            num = Trim(Seq)
          End If
          If arr(x, y) < num + 30 And arr(x, y) >
num - 30 Then
            Count = Count + 1
          End If
        Next y
      Next x

      If Count > most Then
        most = Count
        same = b
      End If
    Next b

End Sub

Public Sub getData()
Dim x, y  As Integer

    maxx = Picture1.ScaleWidth
    maxy = Picture1.ScaleHeight

    ReDim color(maxx, maxy)

    For i = 0 To maxx - 1
      For j = 0 To maxy - 1
        tmp = CStr(Hex(Picture1.Point(i, j)))
        tmpR = Mid(tmp, 1, 2)
        tmpG = Mid(tmp, 3, 2)
        tmpB = Mid(tmp, 5, 2)

        tempR = val("&H" & tmpR)
        tempG = val("&H" & tmpG)
        tempB = val("&H" & tmpB)
        Temp = Int((tempR + tempG + tempB) / 3)
        Picture1.PSet (i, j), RGB(Temp, Temp,
Temp)
        color(i, j) = Temp
      Next j
```

```
For i = 0 To maxx - 1
  For j = 0 To maxy - 1
    tmp = CStr(Hex(Picture1.Point(i, j)))
    tmpR = Mid(tmp, 1, 2)
    tmpG = Mid(tmp, 3, 2)
    tmpB = Mid(tmp, 5, 2)

    tempR = val("&H" & tmpR)
    tempG = val("&H" & tmpG)
    tempB = val("&H" & tmpB)
    Temp = Int((tempR + tempG + tempB) / 3)
    color(i, j) = Temp
  Next j
Next i

maskX = Int(maxx / size)
maskY = Int(maxy / size)

Data = ""
For i = 0 To size - 1
  x = i * maskX
  For j = 0 To size - 1
    y = j * maskY
    arr(i, j) = GetRepresentFile(x, y)
    Data = Data & arr(i, j) & " "
  Next j
Next i

End Sub

Private Function GetRepresentFile(X1, Y1 As
Integer) As Integer
Dim Temp As Double
  Temp = 0
  For n = X1 To maskX + X1 - 1
    For m = Y1 To maskY + Y1 - 1
      Temp = Temp + color(n, m)
    Next m
  Next n
  GetRepresentFile = Int(Temp / (maskX *
maskY))
End Function
```

**FrmAdvanceSearch :**

```
Dim OraDyna As OraDynaset
Dim Command, Sql As String
Dim maskX, maskY As Integer
Dim same As Integer
Dim Data As String
Dim Seq As String

Private Sub cmdSearch_Click()
  Sql_Command = GetSql_Command

  Trim (Sql_Command)
  Sql = "select count(id) as amount from project
where " & Sql_Command

  Set OraDyna = OraDatabase.CreateDynaset(Sql,
&H0&)

  OraDynaset.FindFirst Sql_Command
  If OraDynaset.NoMatch Then
    MsgBox "Can not Find Record"
  ElseIf Picture1.Picture = None Then
    FileName = OraDynaset.Fields("path")
    FrmShowResult.Picture1.Picture =
LoadPicture(FileName)
    FrmShowResult.txtID.Text =
OraDynaset.Fields("id")
    FrmShowResult.txtName.Text =
OraDynaset.Fields("name")
```

```
    Else: Command = "age = '" & Str(valAge + i)
& "'"
    End If
  Next i
End Sub

Private Sub getHeight()
Dim strHeight  As String
Dim valHeight As Integer
  strHeight = cmbHeight.Text
  valHeight = val(Left(strHeight, 3))
  For i = 0 To 4
    If Command <> "" Then
      Command = Command & "or heigth = '" &
Str(valHeight + i) & "'"
    Else: Command = "heigth = '" &
Str(valHeight + i) & "'"
    End If
  Next i
End Sub

Sub search()
Dim fso As New FileSystemObject
Dim objts As TextStream
Dim objFil As File
Dim Seq As String
Dim SeqIn, most, found, num As Integer
Dim FileData As String
Dim blank As String
Dim Count As Integer

  Call getData

  Set objts = fso.OpenTextFile("c:\sequence.txt",
ForReading, False)
  Seq = Trim(objts.ReadLine)
  SeqIn = val(Seq)
  most = 0

  For b = 1 To SeqIn
    Sql_Criteria = "id =' " & b & "'"
    OraDynaset.FindNext Sql_Criteria
    FileData = OraDynaset.Fields("Data")

    Set objts = fso.OpenTextFile(FileData,
ForReading, False)
    Seq = Trim(objts.ReadLine)
    Count = 0
    For x = 0 To size - 1
      For y = 0 To size - 1
        If x * y <> 81 Then
          blank = " "
          found = InStr(Seq, blank)
          num = Int(Left(Seq, found - 1))
          Seq = Trim(Right(Seq, Len(Seq) -
found + 1))
        Else
          num = Trim(Seq)
        End If
        If arr(x, y) < num + 30 And arr(x, y) >
num - 30 Then
          Count = Count + 1
        End If
      Next y
    Next x

    If Count > most Then
      most = Count
      same = b
    End If
  Next b
```

```
End Sub

Public Sub getData()
Dim x, y  As Integer

   maxx = Picture1.ScaleWidth
   maxy = Picture1.ScaleHeight

   ReDim color(maxx, maxy)

   For i = 0 To maxx - 1
      For j = 0 To maxy - 1
         tmp = CStr(Hex(Picture1.Point(i, j)))
         tmpR = Mid(tmp, 1, 2)
         tmpG = Mid(tmp, 3, 2)
         tmpB = Mid(tmp, 5, 2)

         tempR = val("&H" & tmpR)
         tempG = val("&H" & tmpG)
         tempB = val("&H" & tmpB)
         Temp = Int((tempR + tempG + tempB) / 3)
         Picture1.PSet (i, j), RGB(Temp, Temp,
Temp)
         color(i, j) = Temp
      Next j
   Next i

   maskX = Int(maxx / size)
   maskY = Int(maxy / size)

   For i = 0 To size - 1
      x = i * maskX
      For j = 0 To size - 1
         y = j * maskY
         arr(i, j) = GetRepresentBO(x, y)
         Debug.Print arr(i, j)
      Next j
   Next i

   Dim fso As New FileSystemObject
   Dim objFil As File
   Dim objts As TextStream

   Dim tee As String
   FileData = Left(FileName, Len(FileName) - 4)
& ".txt"
   fso.CreateTextFile (FileData)
   Set objFil = fso.GetFile(FileData)
   Set objts =
objFil.OpenAsTextStream(ForWriting)
   For i = 0 To size - 1
      For j = 0 To size - 1

         objts.Write arr(i, j) & " "
      Next j
   Next i
   objts.Close
End Sub

Private Function GetRepresentBO(X1, Y1 As
Integer) As Integer
Dim Temp As Double
Dim tmp As String
   Temp = 0
   For n = X1 To maskX + X1 - 1
      For m = Y1 To maskY + Y1 - 1
         Temp = Temp + color(n, m)
      Next m
   Next n
   GetRepresentBO = Int(Temp / (maskX *
maskY))
End Function
```

```
Private Sub Form_Load()
'Create the OraSession Object.
   Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
'Create the OraDatabase Object by opening a
connection to Oracle
   Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
'Create the OraDynaset Object.
   Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)

End Sub

Sub searchData()
Dim fso As New FileSystemObject
Dim objts As TextStream
Dim objFil As File
'Dim Seq As String
Dim SeqIn, most, found, num As Integer
Dim blank As String
Dim Count As Integer

   Call getFileData          'get array representation
of
                  'this picture =>
array(size*size)

   Set objts = fso.OpenTextFile("c:\sequence.txt",
ForReading, False)
   Seq = Trim(objts.ReadLine)     'open file
sequence for reading the amount of picture in
database
   SeqIn = val(Seq)
   most = 0
      'Create the OraSession Object.
      '   Set OraSession =
CreateObject("OracleInProcServer.XOraSession")
      'Create the OraDatabase Object by opening a
connection to Oracle
      '   Set OraDatabase =
OraSession.DbOpenDatabase("exOracle",
"teety/bobo", &H0&)
      'Create the OraDynaset Object.
      '   Set OraDynaset =
OraDatabase.CreateDynaset("select * from Project
order by id asc", &H0&)
         'SeqID = 1

    For b = 1 To SeqIn
      Sql_Criteria = "id =' " & b & "'"
      OraDynaset.FindNext Sql_Criteria
      Seq = OraDynaset.Fields("Data")

      'Set objts = fso.OpenTextFile(FileData,
ForReading, False)
      'Seq = Trim(objts.ReadLine)
      Count = 0
      For x = 0 To size - 1
        For y = 0 To size - 1
          If x * y <> 81 Then
            blank = " "
            found = InStr(Seq, blank)
            num = Int(Left(Seq, found - 1))
            Seq = Trim(Right(Seq, Len(Seq) -
found + 1))
          Else
            num = Trim(Seq)
```

```
            End If
            If arr(x, y) < num + 30 And arr(x, y) >
num - 30 Then
                Count = Count + 1
            End If
        Next y
    Next x
    If Count > most Then
        most = Count
        same = b
    End If
  Next b
End Sub

Public Sub getFileData()
Dim x, y  As Integer

  maxx = Picture1.ScaleWidth
  maxy = Picture1.ScaleHeight

  ReDim color(maxx, maxy)

  For i = 0 To maxx - 1
    For j = 0 To maxy - 1
      tmp = CStr(Hex(Picture1.Point(i, j)))
      tmpR = Mid(tmp, 1, 2)
      tmpG = Mid(tmp, 3, 2)
      tmpB = Mid(tmp, 5, 2)

      tempR = val("&H" & tmpR)
      tempG = val("&H" & tmpG)
      tempB = val("&H" & tmpB)
      Temp = Int((tempR + tempG + tempB) / 3)
      color(i, j) = Temp
    Next j
  Next i

  maskX = Int(maxx / size)
  maskY = Int(maxy / size)

  Data = ""
  For i = 0 To size - 1
    x = i * maskX
    For j = 0 To size - 1
      y = j * maskY
      arr(i, j) = GetRepresentFile(x, y)
      Data = Data & arr(i, j) & " "
    Next j
  Next i

End Sub

Private Function GetRepresentFile(X1, Y1 As
Integer) As Integer
Dim Temp As Double
  Temp = 0
  For n = X1 To maskX + X1 - 1
    For m = Y1 To maskY + Y1 - 1
      Temp = Temp + color(n, m)
    Next m
  Next n
  GetRepresentFile = Int(Temp / (maskX *
maskY))
End Function
```

# เอกสารอ้างอิง

"การรู้จำภาพใบหน้าคน (Face Recognition)", ปริญญานิพนธ์ ปีการศึกษา 2543 ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

กฤษณ์รัฐ กลิ่นศรีสุข, "การจัดการฐานข้อมูล Oracle ด้วย Visual Basic", พิมพ์ครั้งที่ 1, บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), 172 หน้า พ.ศ.2545

ทรงผล บูรณะโอสถ, สุรชาติ พงศ์สุธนะ "Oracle", พิมพ์ครั้งที่ 1, บริษัท XLNC จำกัด, 321 หน้า พ.ศ.2544

กิตติ ภักดีวัฒนะกุล, จำลอง ครูอุตสาหะ, "Visual Basic 6 ฉบับโปรแกรมเมอร์", พิมพ์ครั้งที่ 9, หจก.ไทยเจริญการพิมพ์, 621 หน้า, มิถุนายน 2544

www.xlnc.com

www.thaivb.net