สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจัดกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค

Feature Clustering on Neural Network

นายวิศรุษ บวรรัตนปราณ
นายศรชัย อุดมธนาพงศ์
นายศรัณย์ เลิศพิพัฒนกุล

เลขทะเบียน 62832
วัน,เดือน,ปี 23 ส.ค. 2549

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค

Feature Clustering on Neural Network

โดย

นายวิศรุษ บวรรัตนปราณ

นายศรชัย อุดมธนาพงศ์

นายศรัณย์ เลิศพิพัฒนกุล

อาจารย์ที่ปรึกษา

ผศ.เกียรติกูล เจียรนัยธนะกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การจัดกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค

Feature Clustering on Neural Network

ผู้จัดทำ

1. นายวิศรุษ บวรรัตนปราณ รหัสนักศึกษา 45010721
2. นายศรชัย อุดมธนาพงศ์ รหัสนักศึกษา 45010742
3. นายศรัณย์ เลิศพิพัฒนกุล รหัสนักศึกษา 45010745

อาจารย์ที่ปรึกษา

(ผศ.เกียรติกูล เจียรนัยธนะกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค

นาย วิศรุต บวรรัตนปราณ 45010721
นาย ศรชัย อุดมธนาพงศ์ 45010742
นาย ศรัณย์ เลิศพิพัฒนกุล 45010745
ผศ. เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษา
ปีการศึกษา 2548

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการจัดกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค โดยใช้ระบบเครือข่ายประสาทเทียมแบบแพร่กลับ (Back-propagation Neural Network: BPNN) โดยจากแนวความคิดที่ว่า ในนิวรอลเน็ทเวิร์คจะมีฟีเจอร์ประกอบกันหลายฟีเจอร์ และฟีเจอร์เหล่านี้อาจจะมีความเกี่ยวข้องกันหรือไม่ก็ได้ ในการพัฒนาจึงมีการจัดกลุ่มของฟีเจอร์ที่เกี่ยวข้องกันไว้ด้วยกัน โดยให้ฟีเจอร์ต่างๆ เป็น อินพุท ของนิวรอลเน็ทเวิร์ค แล้วทำการจัดกลุ่มโดยอาศัยค่าน้ำหนักของระบบนิวรอลเน็ทเวิร์คที่ผ่านการสอนแล้วเป็นค่าอ้างอิงในการจับกลุ่ม โดยระบบเครือข่ายประสาทเทียมที่ทำการจับกลุ่มฟีเจอร์แล้วนี้จะเรียกว่า คลัสเตอร์นิวรอลเน็ทเวิร์ค (Cluster Neural Network) แล้วนำมาเปรียบเทียบกับนิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation Neural Network) ซึ่งถือว่าเป็นแบบมาตรฐาน เพื่อเปรียบเทียบประสิทธิภาพในด้านต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Feature Clustering on Neural Network

Wisarut Borwornrattanapran 45010721

Somchai Udomthanapong 45010742

Saran Lertpipattanagul 45010745

Asst. Prof. Kietkul Jiaranitanakit Advisor

Academic Year 2005

ABSTRACT

This thesis is a part of Feature Clustering on Neural Network. The application use back-propagation neural network. From the concept that back-propagation neural network has many features and these may correlate together, or not. Then in the development we will cluster some features together and use them as input of network and train network. After that we use trained network's weights to cluster features. We will called it as "Cluster Neural Network". Finally we compare performance of "Cluster Neural Network" with "Back-propagation Neural Network".

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก
ผศ. เกียรติกูล เจียรนัยธนะกิจ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้ารู้สึกซาบซึ้งในความกรุณาจากท่าน และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

นาย วิศรุษ บวรรัตนปราณ
นาย ศรชัย อุดมธนาพงศ์
นาย ศรัณย์ เลิศพิพัฒนกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญรูป.....VII

บทที่ 1 บทนำ.....1

1.1 ความสำคัญและที่มาของโครงงาน.....1

1.2 วัตถุประสงค์ของโครงงาน.....1

1.3 ขอบเขตของโครงงาน.....1

1.4 วิธีการดำเนินการ.....2

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....2

1.6 ส่วนประกอบของปริญญานิพนธ์.....2

บทที่ 2 ทฤษฎีพื้นฐานที่ใช้การพัฒนาโปรแกรมการจับกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค.....4

2.1 ระบบการมองเห็นภาพ.....4

2.1.1 หลักการเบื้องต้นของการประมวลผลภาพ.....6

2.1.2 ระดับเกรย์.....6

2.1.3 ฮีสโตแกรม.....7

2.1.4 การแปลงระดับเกรย์.....8

2.1.5 พื้นฐานและระบบของสีโมเดล RGB.....8

2.1.6 ความสว่าง.....9

2.1.7 ชนิดของรูปภาพในกล่องเครื่องมือของโปรแกรมแมทแลป.....9

2.1.8 รูปภาพที่เป็นมัลติเฟรมอาร์เรย์.....15

2.1.9 สรุปชนิดของรูปภาพและคลาสต่างๆ.....16

2.2 นิวรอลเน็ทเวิร์ค.....17

2.2.1 แบบจำลองทางคณิตศาสตร์ของเซลล์ประสาทเดียว.....18

2.2.2 เทคนิคการเรียนรู้แบบแพร่กลับ.....19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3 เทคนิคการแบ่งกลุ่ม.....15

บทที่ 3 โครงสร้างและการทำงานของโปรแกรม.....22

3.1 การสอนนิวรอลเน็ทเวิร์คแบบแพร่กลับ.....22

3.2 การจับกลุ่มฟีเจอร์.....24

3.3 การสร้างและสอนคลัสเตอร์นิวรอลเน็ทเวิร์ค.....26

3.4 โครงสร้างของระบบค้นหาองค์ประกอบของใบหน้า.....26

3.4.1 การประมวลผลภาพ.....27

3.4.2 โครงสร้างของโปรแกรมส่วนค้นหาองค์ประกอบบนใบหน้า.....32

บทที่ 4 การทดลองและผลการทดลองโปรแกรม.....40

4.1 ภาพและชุดข้อมูลที่ใช้ในการทดสอบโปรแกรม.....40

4.1.1 ภาพที่ใช้ในการทดสอบโปรแกรม.....40

4.1.2 ผลที่ได้จากการค้นหาฟีเจอร์ของใบหน้า.....41

4.2 เปรียบเทียบกับระหว่างคลัสเตอร์นิวรอลเน็ทเวิร์คกับนิวรอลเน็ทเวิร์คแบบแพร่กลับ...41

บทที่ 5 บทวิจารณ์และสรุป.....43

5.1 บทสรุป.....43

5.2 วิจารณ์สิ่งที่ได้จากโครงงาน.....44

5.3 ปัญหาอุปสรรคและแนวทางแก้ไข.....44

5.4 แนวทางการพัฒนาต่อ.....44

บรรณานุกรม.....47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ความสัมพันธ์ของโปรแกรมแมทแลปกับชนิดของไฟล์ภาพ.....16

4.1 ผลลัพธ์จากการค้นหาฟีเจอร์บนใบหน้า.....41

4.2 แสดงการเปรียบเทียบระหว่างคลัสเตอร์นิวรอลเน็ทเวิร์คกับ
นิวรอลเน็ทเวิร์คแบบแพร่กลับ ของชุดข้อมูล Breast-cancer-wisconsin 41

4.3 แสดงการเปรียบเทียบระหว่างคลัสเตอร์นิวรอลเน็ทเวิร์คกับ
นิวรอลเน็ทเวิร์คแบบแพร่กลับ ของชุดข้อมูล Iris.....42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 แสดงฮีสโตแกรมของ 256 ระดับเกรย์.....7

2.2 ค่าของพิกเซลชี้ไปยังค่าเทียบเคียงสีในดัชนีรูปภาพ.....10

2.3 ค่าพิกเซลในความหนาแน่นของรูปภาพกำหนดระดับสีเทา.....11

2.4 พิกเซลในไบนารีของรูปภาพมีค่าที่เป็นไปได้สองค่า คือ 0 หรือ 1.....12

2.5 ระนาบสีของรูปภาพอาร์จีบี.....13

2.6 ระนาบสีที่แยกจากกันของรูปภาพอาร์จีบี.....11

2.7 เซลล์สมองของมนุษย์.....13

2.8 Neural Network ของ McCullock & Pitts.....15

2.9 โครงสร้างแบบจำลองอย่างง่ายของเซลล์ประสาท.....18

2.10 โครงประสาทเทียมที่พื้นฐาน.....19

3.1 Flow chart การจับกลุ่มองค์ประกอบในหน้า.....25

3.2 ผลลัพธ์การจับกลุ่มน้ำหนักของนิวรอลเน็ทเวิร์กที่สอนด้วยข้อมูลของ Iris.....25

3.3 แบบจำลองของคลัสเตอร์นิวรอลเน็ทเวิร์ก.....26

3.4 ภาพตัวอย่างรูปถ่ายจากกล้องดิจิตอล.....27

3.5 ภาพที่ได้จากการแปลงเป็นไบนารี.....28

3.6 แสดงการเลือกค่า Threshold ที่ต่ำเกินไป.....29

3.7 แสดงการเลือกค่า Threshold ที่มากเกินไป.....29

3.8 แสดงกรอบสี่เหลี่ยมที่ใช้ในการหาดวงตา.....30

3.9 แสดงกรอบสี่เหลี่ยมที่ใช้หาจมูกและปาก.....30

3.10 แสดงขนาดของดวงตาที่หาได้.....31

3.11 แสดงขนาดของปากและจมูกที่หาได้.....31

3.12 Flow chart ของโปรแกรมทำ Preprocessing.....33

3.13 Flow chart ของโปรแกรมการทำ Binarization.....34

3.14 Flow chart ของโปรแกรมการหาขนาดของตา.....35

3.15 Flow chart ของโปรแกรมการหาขนาดของตา (ต่อ).....36

3.16 Flow chart ของโปรแกรมการหาขนาดของตา (ต่อ).....37

3.17 Flow chart ของการหาขนาดจมูกและปาก.....38

3.18 Flow chart ของการหาขนาดจมูกและปาก (ต่อ).....39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.1 ภาพที่สามารถนำมาใช้กับโปรแกรมได้ 40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

เนื่องจากวิวัฒนาการทางด้านคอมพิวเตอร์ได้ก้าวหน้าไปอย่างรวดเร็วมาก แต่การทำงานของคอมพิวเตอร์ยังมีข้อจำกัดอยู่ คือไม่สามารถทำงานที่ต้องใช้การคิดที่ซับซ้อนเหมือนมนุษย์ได้ เพราะว่าคอมพิวเตอร์ไม่สามารถที่จะเรียนรู้เรื่องราวต่างๆ ได้เหมือนกันมนุษย์ จึงได้มีการพัฒนาเทคนิคที่ช่วยให้คอมพิวเตอร์สามารถทำในสิ่งเหล่านี้ได้ เรียกว่า Neural Network

แต่ว่าฟีเจอร์ของข้อมูลที่นำเข้านิวรอลเน็ทเวิร์คนี้อาจจะมีว่าเกี่ยวข้องกันหรือไม่ก็ได้ จากแนวคิดนี้จึงได้จับกลุ่มของฟีเจอร์บนนิวรอลเน็ทเวิร์ค

1.2 วัตถุประสงค์ของโครงงาน

ปริญญานิพนธ์ฉบับนี้มุ่งหวังเพื่อ พัฒนาโปรแกรมคอมพิวเตอร์ขึ้นมา เพื่อใช้ในการจัดกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค เพื่อทำการเปรียบเทียบกันระหว่าง นิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation Neural Network) ซึ่งถือว่าเป็นนิวรอลเน็ทเวิร์คแบบมาตรฐาน กับคลัสเตอร์นิวรอลเน็ทเวิร์ค (Cluster Neural Network) ซึ่งจะทำการแบ่งกลุ่มของอินพุทของนิวรอลเน็ทเวิร์คโดยอาศัยการแบ่งกลุ่มโดยใช้ค่ากลาง (K-Medoid Clustering Algorithm) ซึ่งพัฒนาขึ้นจากแนวความคิดที่ว่า ในฟีเจอร์ของข้อมูลที่นำเข้านิวรอลเน็ทเวิร์คอาจจะมีความเกี่ยวข้องกันหรือไม่ก็ได้ ซึ่งน่าจะลดการทำงานที่ไม่จำเป็นของนิวรอลเน็ทเวิร์คลงได้

1.3 ขอบเขตของโครงงาน

วิทยานิพนธ์ฉบับนี้เป็นการพัฒนาโปรแกรมที่ใช้เพื่อใช้ในการจัดกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค โดยระบบ Neural Network ที่นำมาใช้นี้มีอยู่ 2 แบบคือ นิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation algorithm) และ คลัสเตอร์นิวรอลเน็ทเวิร์ค (Cluster Neural Network : CNN) เพื่อเปรียบเทียบค่าความถูกต้องและความเร็วที่ในการจำแนก

คลัสเตอร์นิวรอลเน็ทเวิร์ค คือ รูปแบบของนิวรอลเน็ทเวิร์ค ที่มีการเชื่อมต่อกันแบบแบ่งกลุ่มเชื่อมต่อ โดยกลุ่มใดๆ ของอินพุทใน Input Layer จะเชื่อมต่อกับ Hidden layer ในกลุ่มเดียวกัน และ จาก Hidden layer จะเชื่อมต่อกับ Output layer ทั้งหมด ในการแบ่งกลุ่มของอินพุทนี้จะแบ่งโดยอาศัยค่าน้ำหนักของ Input Layer นั้นๆ ไปยัง Hidden Layer ใดๆ ในชั้นที่ติดกัน และใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแบ่งกลุ่มนี้ใช้วิธีการแบ่งกลุ่มโดยอาศัยค่ากลาง (Median) ของกลุ่มนั้นๆ (K-Medoid Clustering Algorithm)

1.4 วิธีการดำเนินการ

1. ศึกษาเกี่ยวกับกระบวนการประมวลผลภาพ เพื่อนำองค์ประกอบของภาพออกมาวิเคราะห์
2. ศึกษาเกี่ยวกับการสร้างนิวรอลเน็ทเวิร์ค (Neural Network) เพื่อนำมาวิเคราะห์ฟีเจอร์
3. ศึกษาเกี่ยวกับกระบวนการแบ่งกลุ่ม (Clustering Algorithm)
4. วิเคราะห์ และออกแบบโปรแกรม
5. จัดหาภาพที่จะนำมาใช้ในการทดสอบโปรแกรม
6. วิเคราะห์ผลของโปรแกรมที่จัดทำขึ้น และแก้ไขส่วนที่ผิดพลาดเพื่อให้สามารถจัดกลุ่มได้ถูกต้องยิ่งขึ้น
7. เปรียบเทียบ และวิเคราะห์ผลระหว่าง คลัสเตอร์นิวรอลเน็ทเวิร์ค (Cluster Neural Network) กับนิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation Neural Network)

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถพัฒนาโปรแกรมคอมพิวเตอร์เพื่อจำแนกใบหน้ามนุษย์ขึ้นมาได้
2. ได้รับความรู้ความเข้าใจเกี่ยวกับ ทฤษฎี Image Processing
3. ได้รับความรู้ความเข้าใจเกี่ยวกับ ทฤษฎี Neural Network
4. ได้รับความรู้ความเข้าใจเกี่ยวกับ ทฤษฎี Clustering Algorithm แบบ K-Medoid

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ซึ่งประกอบด้วยทฤษฎีอะไรบ้าง ได้บรรยายทฤษฎีทั้งหมดโดยละเอียด

บทที่ 3 กล่าวถึง ส่วนที่ได้พัฒนาขึ้น และการทำงานของโปรแกรมจำแนกใบหน้า

บทที่ 4 กล่าวถึงการทดลองและผลการทดลอง การหาค่าสมรรถนะของระบบ การวัดประสิทธิภาพของระบบ พารามิเตอร์ที่ใช้และผลที่ได้จากการจำลองระบบ ผลการทดลองหรือผลการทำงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของโครงงาน วิจารณ์สิ่งที่ได้รับจากโครงงาน และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการพัฒนาโปรแกรมการจับกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค ซึ่งเนื้อหาในบทนี้จะกล่าวถึง ทฤษฎีทาง Image processing นิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation Neural Network) และการแบ่งกลุ่มของข้อมูล (Clustering Algorithm) ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการพัฒนา และประเมินประสิทธิภาพของโปรแกรมการจัดกลุ่มบนนิวรอลเน็ทเวิร์ค

2.1 ระบบการมองเห็นภาพ (Vision System)

กลไกระบบการมองเห็นภาพ (Vision System) นั้น จะหมายความรวมถึงทุกสิ่งที่จำเป็นที่สามารถแปลงภาพนั้นๆ เป็นรหัสดิจิตอล เพื่อนำไปใช้ในระบบคอมพิวเตอร์ได้. การปรับปรุงเปลี่ยนแปลงข้อมูล และการแสดงภาพที่ได้มาหลังจากการปรับปรุงเปลี่ยนแปลงข้อมูลแล้ว ระบบการมองเห็นภาพนี้ความยุ่งยากจะขึ้นอยู่กับการใช้งาน ซึ่งสามารถแบ่งเป็น 3 ขั้นตอนที่สำคัญ ดังนี้

1. การได้มาซึ่งภาพ (Image Acquisition)
2. กระบวนการการประมวลผลภาพ (Image Processing)
3. ผลที่ได้ หรือ การแสดงผล (Output or Display)

ในปัจจุบันได้มีการประยุกต์ใช้งานระบบการมองเห็นภาพอยู่มากมาย เช่น การใช้บาร์โค้ด , การพิมพ์สิ่งต่างๆ , การประยุกต์ใช้งานในโรงงาน

1. การได้มาซึ่งภาพ (Image Acquisition)

การได้มาซึ่งภาพ หมายถึง การแปลงภาพในลักษณะทางกายภาพ ให้เป็นเซตข้อมูลทางดิจิตอล ซึ่งเซตข้อมูลนี้จะถูกส่งไปยังหน่วยประมวลผลต่อไป ฟังก์ชันของการได้มาซึ่งภาพนี้แบ่งเป็น 4 เฟส คือ

1) การส่องสว่าง (Illumination)
2) รูปแบบของภาพ หรือ การทำให้ภาพชัดขึ้น (Image formation or Focusing)
3) การตรวจจับภาพ หรือ การรับภาพ (Image detection or Sensing)
4) รูปแบบผลสัญญาณที่ได้จากกล้อง (Formatting camera output signal)

การส่องสว่างเป็นตัวแปรสำคัญที่มีอิทธิพลต่อสัญญาณอินพุต (Input signal) ที่จะส่งต่อให้กับ ระบบการมองเห็นภาพ เพราะฉะนั้นเราจึงต้องออกแบบ ให้มีการส่องสว่างที่เหมาะสมกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานที่แตกต่างกัน โดยที่ชนิดและวิธีการส่องสว่างของแหล่งกำเนิดแสง จะมีผลต่อกำลังงานของแสงที่ส่งออกมา ซึ่งจะมีผลต่อกระบวนการประมวลผลภาพ และผลของสัญญาณที่ได้รับ

2. กระบวนการประมวลผลภาพ (Image Processing)

กระบวนการประมวลผลภาพ คือ การสร้างภาพใหม่โดยการแยกส่วนของข้อมูลที่เราสนใจ กับ สิ่งรบกวน (Noise) ออกจากกัน โดยการทำงานพื้นฐานของการประมวลผล คือ การกำจัดสิ่งรบกวนของภาพ (Noise elimination) , การปรับแต่งของภาพให้ดีขึ้น (Edge enhancement) , การกรองภาพ (Filtering) , การปรับปรุงเปลี่ยนแปลงค่าระดับเกรย์ (Grey scale modification) โดยทั่วไปแล้วจะใช้ฮาร์ดแวร์ในการประมวลผลภาพ แต่ถ้ามีการประมวลผลภาพที่ซับซ้อนขึ้น ก็จะใช้ทั้ง ฮาร์ดแวร์ และซอฟต์แวร์ ซึ่งการซับซ้อนของการประมวลผลจะขึ้นอยู่กับลักษณะของการใช้งาน และมีวิธีการประมวลผล 3 อย่าง คือ

1) ที่จุดเดียวกันในรูปภาพเดียวกัน (Point by point in one image)

คือ การสร้างภาพใหม่โดยการเปลี่ยนค่าแบบจุดต่อจุด โดยจุดหนึ่งคือจุดในรูปภาพเดิม และอีกจุดคือจุดในรูปภาพใหม่ที่ได้รับการปรับปรุงเปลี่ยนแปลงจากรูปภาพเดิมแล้ว เช่น การแปลงภาพในระบบเลขฐาน 2 จากจุดที่มีค่าเป็น 0 ถูกเปลี่ยนเป็น 1 และจากค่า 1 ถูกเปลี่ยนกลับเป็น 0

2) ที่จุดเดียวกันในรูปภาพที่แตกต่างกัน (Using corresponding points on different images)

คือ การสร้างภาพใหม่โดยการจับคู่แต่ละจุดจากแหล่งภาพ 2 แหล่ง หรือมากกว่า จุดของรูปภาพที่แตกต่างกันสองจุดนี้ จะนำมาลบกันเพื่อหาค่าที่เปลี่ยนแปลงไป ข้อมูลของจุดในภาพใหม่จะเกิดจากการรวมกันของตัวเลขที่แตกต่างกันของภาพ 2 แหล่งนั้น

3) ที่บริเวณจุดนั้นในภาพนั้น (Using regional points in one image)

คือ การสร้างภาพใหม่โดยการหาค่าเฉลี่ยรอบๆจุดนั้นในภาพนั้น ค่าของจุดในภาพใหม่คือ ค่าเฉลี่ยของจุดในภาพนั้นอันเดิม

3. ผลที่ได้และการแสดงผล (Output or Display)

รูปแบบของการแสดงผลนั้นขึ้นอยู่กับว่า จะนำข้อมูลที่ได้ไปใช้ต่อในลักษณะใด หรือแสดงผลอย่างไรจึงจะเหมาะสมที่สุด เช่น การแสดงผลโดยการพิมพ์ , การแสดงผ่านหน้าจอ , การแสดงผลในรูปแบบของสัญญาณควบคุม เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 หลักการเบื้องต้นของการประมวลผลภาพ (Image Processing Fundamentals)

1. พิกเซล (Pixel)

ในภาพหนึ่งๆ เราสามารถอธิบายได้ด้วยเมทริกซ์ของจุดพิกเซลขนาด N x M โดยใช้คู่ลำดับ p(i, j) แทนค่าของจุดแต่ละจุด โดย i และ j ต้องเป็นจำนวนเต็มบวกหรือศูนย์เท่านั้น และ p(i, j) นี้ จะบ่งชี้ความเข้มของแสงที่จุดนั้นๆ ของภาพ

ค่าที่กำกับแต่ละพิกเซลจะแสดงถึงค่าเฉลี่ยของความเข้มแสงในภาพที่จุดพิกเซลนั้นแทนอยู่ โดยที่ค่าของพิกเซลนี้จะเขียนแทนด้วย P_{ij} ซึ่งมีค่าตั้งแต่ 0 ถึง 1

2. ตำแหน่งพิกเซล (Pixel Location)

ดังที่ได้กล่าวมาแล้วว่า ในภาพหนึ่งๆนั้น เราสามารถแทนอาร์เรย์ (Array) N x M และค่าในแต่ละจุดพิกเซล จะหมายถึงค่าเฉลี่ยของความเข้มของแสงที่ตกกระทบถึงภาพที่จุดพิกเซลนั้น

2.1.2 ระดับเกรย์ (Grey Scale)

หากเราต้องการค่าของข้อมูลที่ละเอียดมากขึ้น จะต้องเพิ่มจำนวนบิตในการแสดงค่าของแต่ละพิกเซล เช่น ถ้าต้องการแบ่งความเข้มของการส่องสว่างให้มี 4 ระดับ ก็ต้องใช้เลขฐานสอง 2 บิต , 4 บิต สำหรับ 16 ระดับ และ 8 บิต สำหรับ 256 ระดับ ซึ่งจำนวนระดับที่ใช้ในระดับเกรย์นี้มักเป็นเลขยกกำลังของ 2 ส่วนค่าต่ำสุด คือ 0 ถูกกำหนดให้เป็นสีดำ และ 1 หรือตัวเลขที่น้อยกว่าค่าสูงสุดของระดับเกรย์อยู่ 1 (เช่น 15 สำหรับระดับเกรย์ 16 ระดับ) ถูกกำหนดให้เป็นสีขาว ค่าที่กำหนดให้ในแต่ละพิกเซลมักเป็นจำนวนเต็ม

Grey Scale		Grey Value Range
2^1	2 values	0 , 1
2^3	8 values	0 to 7
2^4	16 values	0 to 15
2^8	256 values	0 to 255

ในยุคแรกๆ ของระบบการมองเห็นภาพ (Vision System) จะใช้ระบบเลขฐานสอง แต่ในปัจจุบัน เทคโนโลยีของไมโครโพรเซสเซอร์ (Microprocessors) เข้ามามีบทบาทมากขึ้น ซึ่งในไมโครโปรเซสเซอร์จะมีหน่วยความจำอย่างน้อย 8 บิต เพราะฉะนั้นการแบ่งระดับเป็น 16, 64 หรือ 256 จึงเป็นเรื่องธรรมดา และในตอนนี้กำลังจะมีมากกว่า แต่ในการมองเห็นของมนุษย์สามารถแยกความแตกต่างได้เพียง 10 ถึง 15 ระดับ เท่านั้น ดังนั้นการแบ่งโดยละเอียดเป็น 64 หรือ 256 ระดับ ทำให้มนุษย์ไม่สามารถแยกความแตกต่างได้ จึงนำไปประยุกต์ใช้กับงานการประมวลผลภาพ หรือ ใช้ในกระบวนการทางอุตสาหกรรมต่างๆ

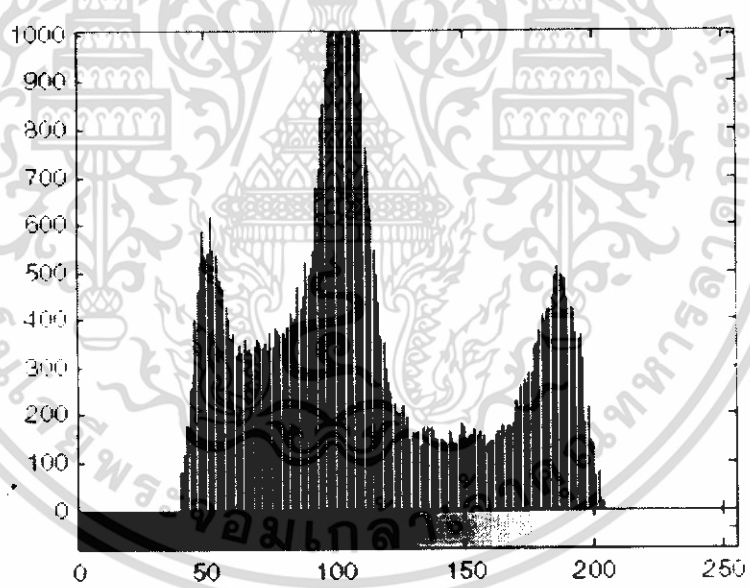
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงกล่าวได้ว่าจำนวนระดับเกรย์เป็นตัวจำกัดรายละเอียดของภาพ ยิ่งแบ่งระดับเกรย์เป็นหลายระดับ ก็จะยิ่งเพิ่มคุณภาพของภาพนั้นด้วย และการเพิ่มจำนวนพิกเซล เช่น จาก 32 × 32 เป็น 250 × 250 ก็จะเป็นการเพิ่มรายละเอียด (Resolution) และรายละเอียด (Detail) ของภาพซึ่งจะแตกต่างกับการขยายภาพ (Zoom) เพราะการขยายภาพ เป็นการเพิ่มขยายของแต่ละพิกเซลให้ใหญ่ขึ้น ไม่ได้เป็นการเพิ่มจำนวนพิกเซล แต่การแบ่งระดับเกรย์เป็นการเพิ่มจำนวนของพิกเซล

2.1.3 ฮีสโตแกรม (Histogram)

ฮีสโตแกรมเป็นการแสดงให้เห็นถึงความถี่ของการนับจำนวนพิกเซล ที่มีค่าความเข้มแต่ละค่าหนึ่งๆในภาพระดับเกรย์ จากรูป 2.3 แกน x ในกราฟแสดงค่าระดับเกรย์ และแกน y แสดงค่าจำนวนพิกเซลในระดับเกรย์นั้น โดยฮีสโตแกรมมีขั้นตอนการสร้างดังนี้

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



รูปที่ 2.1 แสดงฮีสโตแกรมของ 256 ระดับเกรย์

รูปร่างของฮีสโตแกรมสามารถบอกลักษณะบางประการของภาพได้ เช่น ถ้าฮีสโตแกรมมีลักษณะที่แคบ จะหมายถึงขาดการคอนทราสต์ (Contrast. การแยกแยะความผิดแผกของสี) ในภาพนั้น

ฮีสโตแกรมมีประโยชน์ในการกำหนดระดับเทรชโฮล (Threshold) ซึ่งก็คือการเปลี่ยนภาพระดับเกรย์ให้เป็นภาพระบบเลขฐานสอง (Binary) หรือเพื่อการปรับปรุงบางส่วนของสเปกตรัม (Spectrum) ของระดับเกรย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การแปลงระดับเกรย์ (Grey-level Transformation)

1) การเปลี่ยนแปลงความสว่าง(Global Alternative in brightness)

เป็นการใช้ค่าคงที่บวกหรือลบออกจากทุกพิกเซลของภาพ เพื่อเพิ่มหรือลดความสว่างของภาพ

2) การทำเทรชโฮล (Threshold)

เป็นการเปลี่ยนแปลงหรือหาแนวโน้มของค่าระดับเกรย์ในภาพเพื่อทำให้เป็นดิสครีท (Discrete) มากขึ้น โดยจะนำภาพมาทำฮีสโตแกรม แล้วกำหนดค่าระดับเกรย์ที่แน่นอนขึ้นมาเพื่อที่จะเป็นจุดตัดทำเทรชโฮล จากนั้นจะทำการตัดหรือปัดส่วนของระดับเกรย์ที่เราไม่ต้องการออกไป

ประโยชน์ของเทรชโฮล ได้แก่ การทำให้เป็นภาพไบนารี หากดูจากรูปที่ จุดที่อยู่ในช่วงทางซ้ายมือทั้งหมดของจุดตัดเทรชโฮล T จะถูกทำเป็นสีขาว และจุดในช่วงทางขวามือทั้งหมดจะถูกทำเป็นสีดำ หรือ การช่วยให้หาขอบของภาพได้ง่ายขึ้น เป็นต้น แต่ทั้งนี้ก็เป็นการยากที่จะกำหนดจุดตัดเทรชโฮลที่ดีที่สุดออกมาได้

3) บันช์ชิ่ง (Bunching) และการควอนไตซ์ (Quantize)

ในบางครั้ง การบันช์ชิ่ง จะหมายถึง การควอนไตซ์ ด้วย หรือ ใช้ลดระดับเกรย์ของภาพที่ไม่ต้องการลง

หมายเหตุ การควอนไตซ์เป็นการประมาณค่าระดับเกรย์ให้เป็นจำนวนเต็ม ซึ่งจะเป็นไปตามกฎ คือ ถ้าเป็นจุดทศนิยมให้ปัดขึ้นทั้งหมด และการปัดนี้ก็คือความผิดพลาดที่เกิดขึ้น

4) สปลิทติ้ง (Splitting)

เป็นการเพิ่มความแตกต่างระหว่าง 2 กลุ่มของระดับเกรย์ เช่น ถ้าเรามีตัวอักษรเขียนอยู่บนฉาก โดยตัวอักษรมีระดับเกรย์ที่ 98 , ฉากมีระดับเกรย์ที่ 99 ซึ่งตาของมนุษย์ไม่สามารถแยกแยะความแตกต่างเพียงเท่านี้ได้ และจะได้ฮีสโตแกรมที่มีลักษณะแบน เพราะฉะนั้นจึงแก้ปัญหาด้วยการสปลิทฮีสโตแกรม โดยทำการดึงค่า 99 ขึ้นเป็น 120 และดึงค่า 98 เป็น 80 ซึ่งจะทำให้ระหว่างตัวเลขและตัวอักษรมีความแตกต่างกันมากขึ้นจนสามารถสังเกตได้

เทคนิคนี้จะมีประโยชน์มาก ในกรณีที่เราต้องดึงเอาเฉพาะบางส่วนของภาพออกมา

2.1.5 พื้นฐานและระบบของสีโมเดล RGB (Red, Green, Blue)

ในโมเดลนี้ สีแต่ละสีจะอยู่ในรูปของสีปฐมภูมิ (แดง, เขียว, น้ำเงิน) โมเดลนี้มีโครงสร้างอยู่ในระบบของพิกัดคาร์ทีเซียน (Cartesian coordinate) มีลักษณะเป็นทรงลูกบาศก์ ดังรูปที่ ค่าสีแดง, เขียว, น้ำเงิน จะอยู่ที่มุมทั้งสามในลักษณะเดียวกัน ส่วนสีดำจะอยู่ที่จุดกำเนิด สีขาวอยู่ที่มุมที่มีระยะทางไกลที่สุดจากแหล่งกำเนิด ค่าของระดับสีเทา (Grey Scale) จะอยู่บนเส้นที่เชื่อมระหว่างสีดำกับสีขาว ค่าสี คือ จุดที่อยู่บนผิวหรือในลูกบาศก์ถูกกำหนดค่าโดยเวกเตอร์ที่ชี้ออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยากต่อการนับ เพื่อความสะดวกเราจะสมมุติให้ค่าสีถูกนอร์มอลไลซ์ (Normalize) ทั้งสามสี โดยให้มีค่าอยู่ในช่วง 0 ถึง 1 ลูกบาศก์ที่แสดงในรูปที่ นี้ จึงเป็นลูกบาศก์หนึ่งหน่วย

ภาพในโมเดล RGB ประกอบด้วยภาพสามระนาบที่เป็นอิสระจากกัน สำหรับแต่ละสีปฐมภูมิเมื่อถูกป้อนเข้าไปในมอนิเตอร์ (monitor) ที่เป็นแบบ RGB ภาพทั้งสามสีจะรวมตัวกันที่จอภาพกลายเป็นภาพสีผสม ดังนั้นการใช้โมเดล RGB ในการประมวลผลภาพนั้นจะสมเหตุสมผลเมื่อภาพถูกแยกออกโดยธรรมชาติให้อยู่ในเทอมของทั้งสามสี กล้องภาพสีส่วนใหญ่ที่ให้ภาพสีดิจิตอลจะอยู่ในรูปแบบของโมเดล RGB ดังนั้นโมเดลนี้จึงเป็นโมเดลที่สำคัญมากในการประมวลผลภาพ

2.1.6 ความสว่าง (Brightness)

ความสว่างเป็นคุณสมบัติของแสงที่มนุษย์เป็นผู้กำหนดให้ระหว่างความสว่างมากที่สุดกับความมืด ความสว่างเป็นสิ่งที่รับรู้ได้แต่ไม่สามารถสัมผัสได้ ความสว่างสามารถรับรู้ได้เมื่อมีแสงตกกระทบกับเรตินารูปแท่งและรูปกรวยในตาของคนเรา ซึ่งจะก่อให้เกิดการตอบสนองที่ไม่เป็นเชิงเส้นและสลับซับซ้อน ความไวในการรับแสงของคนเราจะลดลงเมื่อแสงมีขนาดใหญ่มากขึ้น สำหรับความสว่างที่อยู่ในรูปภาพนั้น สามารถแสดงได้โดย Brightness Histogram ซึ่งเป็นกราฟการกระจายตัวของกลุ่มตัวเลข ในกราฟนี้จะแสดงถึงระดับสีเทาของแต่ละพิกเซลที่อยู่ในรูปภาพ หรืออาจจะกล่าวได้ว่าเป็นการแสดงให้เห็นถึงจำนวนของพิกเซลในรูปภาพที่อยู่ในแต่ละระดับของสีเทาว่ามีอยู่เป็นจำนวนเท่าไร

2.1.7 ชนิดของรูปภาพในกล่องเครื่องมือของโปรแกรมแมทแลป

เครื่องมือในการประมวลผลรูปภาพรองรับรูปแบบพื้นฐาน 4 อย่างของรูปภาพ คือ

- ดัชนีรูปภาพ (Index images)
- ความหนาแน่นของรูปภาพ (Intensity images)
- ไบนารีของรูปภาพ (Binary images)
- อาร์จีบีของรูปภาพ (RGB images)

1. ดัชนีรูปภาพ (Index images)

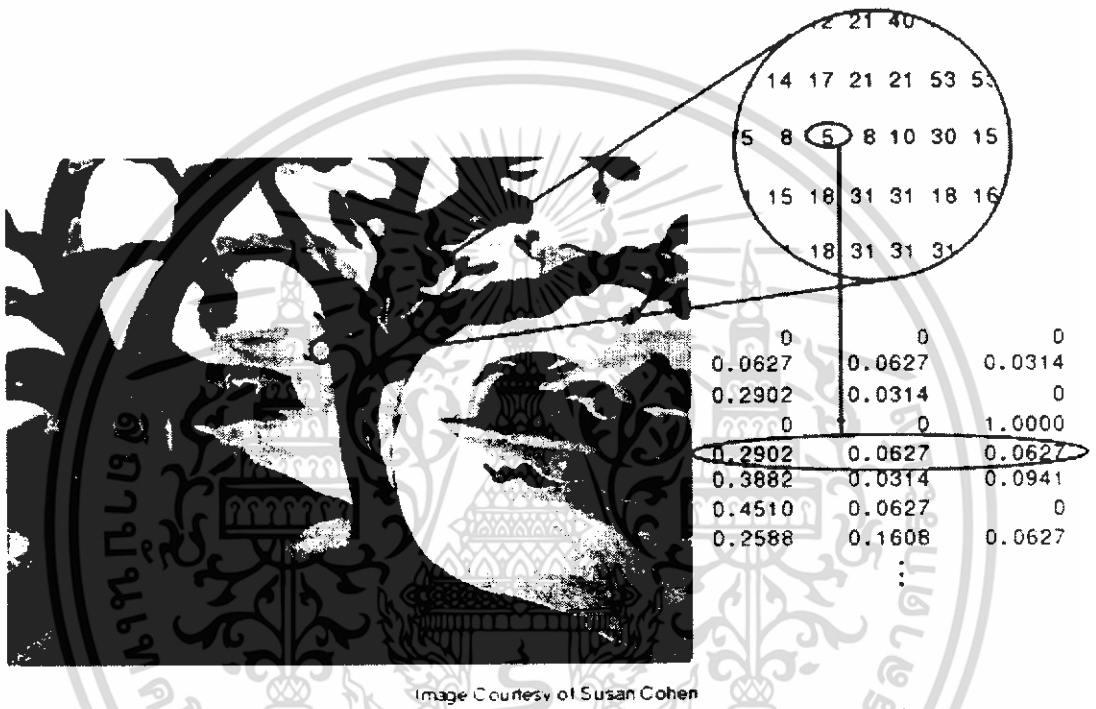
ดัชนีรูปภาพประกอบไปด้วยเมทริกซ์ของข้อมูล X และเมทริกซ์ของสีที่จะนำไปเทียบเคียงเมทริกซ์ของข้อมูลสามารถแบ่งออกเป็นคลาสของ uint8, uint16 หรือ double ส่วนเมทริกซ์ของสีนั้นจะเป็นอาร์เรย์ขนาด m×3 ของคลาส double ที่ประกอบด้วยเลขทศนิยมที่อยู่ในช่วง [0,1] แต่ละแถวของการเทียบเคียงนั้นจะระบุคอมโพเนนท์ สีแดง, สีเขียว และสีน้ำเงินของแต่ละสีนั้น ดัชนีรูปภาพจะใช้การเทียบเคียงโดยตรงของแต่ละค่าในพิกเซลนั้นไปยังค่าเทียบเคียงสี สีในแต่ละพิกเซลบนรูปภาพนั้นจะถูกกำหนดโดยใช้ค่าที่มีลักษณะเช่นเดียวกันกับ X เป็นดัชนีในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทียบเคียง ค่าของจุดแรกแทนแถวแรกในการเทียบเคียง ค่าของจุดที่สองแทนแถวที่สอง และเป็นเช่นนี้เรื่อยไป

การเทียบเคียงสีมักจะถูกเก็บด้วยดัชนีรูปภาพและจะนำมาใช้อย่างอัตโนมัติกับรูปภาพเมื่อเรียกใช้ฟังก์ชัน imread อย่างไรก็ตามในการใช้งานนั้น ไม่ได้

ถูกจำกัดการใช้งานเฉพาะค่าพื้นฐานของการเทียบเคียงสีเท่านั้น แต่ยังสามารถเรียกใช้งานค่าใดๆในการเทียบเคียงสีได้ รูปภาพดังต่อไปนี้แสดงถึงโครงสร้างของดัชนีรูปภาพ พิกเซลที่อยู่ในรูปภาพจะถูกแสดงด้วยเลขจำนวนเต็ม ซึ่งจะชี้ไปยังค่าของสีที่เก็บอยู่ในการเทียบเคียงสีนั้น



รูปที่ 2.2 ค่าของพิกเซลชี้ไปยังค่าเทียบเคียงสีในดัชนีรูปภาพ

- **คลาสและค่าออฟเซตของการเทียบเคียงสี**

ความสัมพันธ์ระหว่างค่าที่อยู่ในเมทริกซ์รูปภาพกับค่าในการเทียบเคียงสีนั้นขึ้นอยู่กับคลาสของเมทริกซ์รูปภาพถ้าเมทริกซ์รูปภาพเป็นคลาสของ double ค่าของจุดแรกจะชี้ไปยังแถวแรกของการเทียบเคียง ค่าของจุดที่สองจะชี้ไปยังแถวที่สอง และเป็นเช่นนี้ไปเรื่อยไป ถ้าเมทริกซ์รูปภาพเป็นคลาสของ uint8 หรือ uint16 มันจะเป็นค่าออฟเซต ค่าของจุด 0 จะชี้ไปยังแถวแรกของการเทียบเคียงสี ค่าของจุดที่หนึ่งจะชี้ไปยังแถวที่สอง และเป็นเช่นนี้เรื่อยไป

ค่าออฟเซตมักจะถูกใช้ในไฟล์ที่มีรูปแบบกราฟิกเพื่อจะทำให้จำนวนของสีที่สามารถรองรับได้มีมากที่สุด จากภาพข้างบนนี้เมทริกซ์รูปภาพเป็นคลาส double เพราะมันไม่มีออฟเซต ดังนั้นค่าของจุดที่ห้าจะชี้ไปยังแถวที่ห้าของการเทียบเคียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

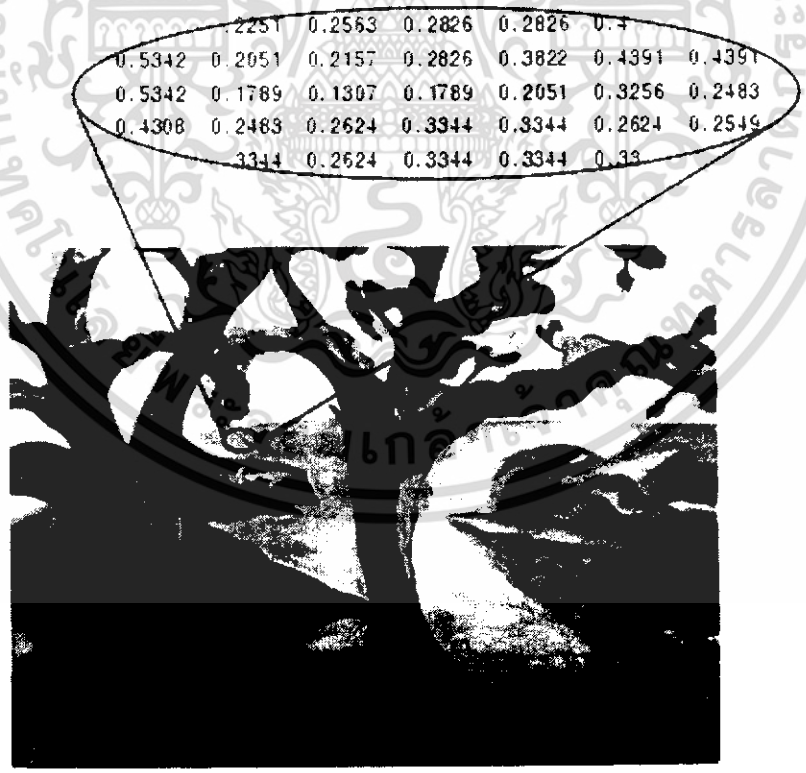
- **ข้อจำกัดของการใช้ uint16**

ในกล่องเครื่องมือนั้นจะมีข้อจำกัดของการใช้ดัชนีรูปภาพที่เป็น คลาส uint16 เราสามารถที่จะอ่านรูปภาพนี้เข้าไปยังโปรแกรมแมทแลป และแสดงมันได้ แต่ก่อนที่จะทำกระบวนการของดัชนีรูปภาพแบบ uint16 จะต้องทำการเปลี่ยนให้เป็น double หรือ uint8 เสียก่อน ในการที่จะเปลี่ยนเป็น double ให้เรียกใช้ im2double เพื่อที่จะลดรูปภาพไปเป็น 256 สีน้อยกว่านั้น (uint8) เรียก imapprox

2. ความหนาแน่นของรูปภาพ (Intensity images)

ความหนาแน่นของรูปภาพคือเมทริกซ์ข้อมูล I ซึ่งค่าของมันจะแทนความหนาแน่นภายในระยะเวลาหนึ่งๆ โปรแกรมแมทแลปจะเก็บค่าความหนาแน่นของรูปภาพเป็นเมทริกซ์เดี่ยว โดยที่แต่ละค่าในเมทริกซ์ นั้นจะมีลักษณะเช่นเดียวกับหนึ่งพิกเซลในรูปภาพ เมทริกซ์สามารถเป็นคลาสของ double, uint8, uint16

ค่าที่อยู่ในเมทริกซ์ความหนาแน่นจะแทนความหนาแน่นที่หลากหลาย หรือระดับสีเทา (Grey levels) โดยค่าความหนาแน่น 0 มักจะแทนด้วยสีดำ และค่าความหนาแน่น 1, 255 หรือ 65535 มักแทนด้วยสีขาว



รูปที่ 2.3 ค่าพิกเซลในความหนาแน่นของรูปภาพกำหนดระดับสีเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ไบนารีของรูปภาพ (Binary images)

ในไบนารีของรูปภาพ แต่ละพิกเซลจะสมมุติให้เป็นหนึ่งในสองค่าที่ไม่ต่อเนื่อง ค่าสองค่านี้จะเป็นลักษณะของ เปิด หรือ ปิด ไบนารีของรูปภาพจะถูกเก็บในเมทริกซ์สองมิติของ 0 (ปิดพิกเซล) และ 1 (เปิดพิกเซล)

ไบนารีของรูปภาพถูกพิจารณาเป็นชนิดพิเศษของความหนาแน่นของรูปภาพ ซึ่งประกอบด้วยสีขาวและสีดำเท่านั้น อย่างไรก็ตาม เราสามารถคิดถึงไบนารีของรูปภาพเป็นลักษณะของดัชนีรูปภาพที่มีเพียงสองสีเท่านั้นก็ได้

ไบนารีของรูปภาพจะถูกเก็บเป็นอาร์เรย์ของคลาส double หรือ uint8 (ในกล่องเครื่องมือไม่รองรับไบนารีของรูปภาพที่เป็นคลาส uint16) อาร์เรย์ของคลาส uint8 โดยทั่วไปแล้วจะถูกนำมาใช้มากกว่าอาร์เรย์ของคลาส double เพราะ uint8 ใช้หน่วยความจำน้อยกว่ามาก ในกล่องเครื่องมือการประมวลผลรูปภาพ ฟังก์ชันใดๆที่คืนค่ากลับมาเป็นไบนารีของรูปภาพจะคืนค่ากลับมาเป็นอาร์เรย์ uint8 ถ้าแฟลกเปิดใช้งานอยู่ ระยะของข้อมูลจะเป็น [0,1] ถ้าแฟลกปิดใช้งานระยะข้อมูลเป็น [0,255]

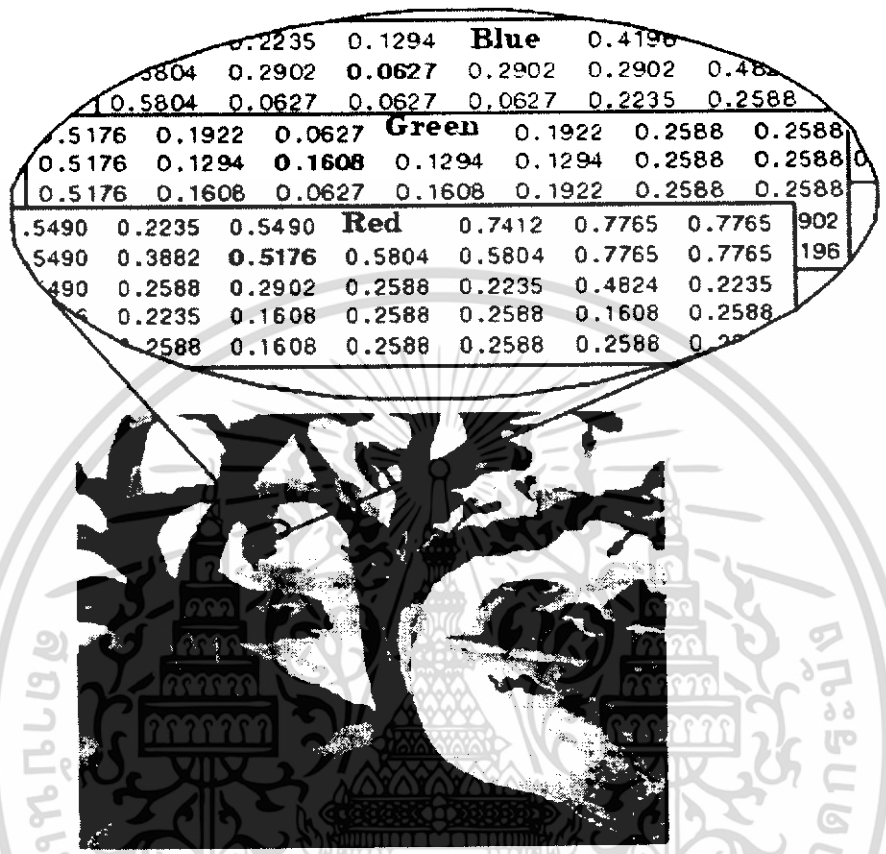
รูปที่ 2.4 พิกเซลในไบนารีของรูปภาพมีค่าที่เป็นไปได้สองค่า คือ 0 หรือ 1

4. อาร์จีบีของรูปภาพ (RGB images)

อาร์จีบีของรูปภาพ หรืออาจเรียกเป็น สีแท้จริงของรูปภาพ (true color image) จะถูกเก็บในโปรแกรมแมทแลป เป็นอาร์เรย์ m×n×3 ซึ่งจะกำหนดคอมโพเนนท์สีแดง, สีเขียวและสีน้ำเงินสำหรับแต่ละพิกเซล อาร์จีบีของรูปภาพไม่ได้ใช้แพทเลตของสี สีในแต่ละพิกเซลจะถูกกำหนดโดยการรวมกันของความหนาแน่นของสีแดง, สีเขียว และสีน้ำเงินที่เก็บอยู่ในแต่ละระนาบของสีที่ตำแหน่งของแต่ละพิกเซล รูปแบบไฟล์กราฟิกจะเก็บอาร์จีบีของรูปภาพเป็น 24 บิต ซึ่งแต่ละ

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมโพเนนท์สีแดง, สีเขียว และสีน้ำเงินจะมีขนาด 8 บิต ซึ่งจะทำให้เกิดจำนวนของสีที่เป็นไปได้ 16 ล้านสี เนื่องจากความละเอียดนี้เอง เราจึงสามารถแทนรูปภาพต่างๆที่พบเห็นได้ในชีวิตจริงด้วยรูปภาพแบบสีแท้จริง



รูปที่ 2.5 ระนาบสีของรูปภาพอาร์จีบี

ในการกำหนดสีของพิกเซลที่จุด (2,3) จะต้องดูที่อาร์จีบีที่เก็บใน (2,3,1:3) สมมุติว่า (2,3,1) ประกอบไปด้วยค่า 0.5176 (2,3,2) ประกอบไปด้วยค่า 0.1608 และ (2,3,3) ประกอบไปด้วยค่า 0.0627 ดังนั้นสีของพิกเซลที่จุด (2,3) คือ

0.5176 0.1608 0.0627

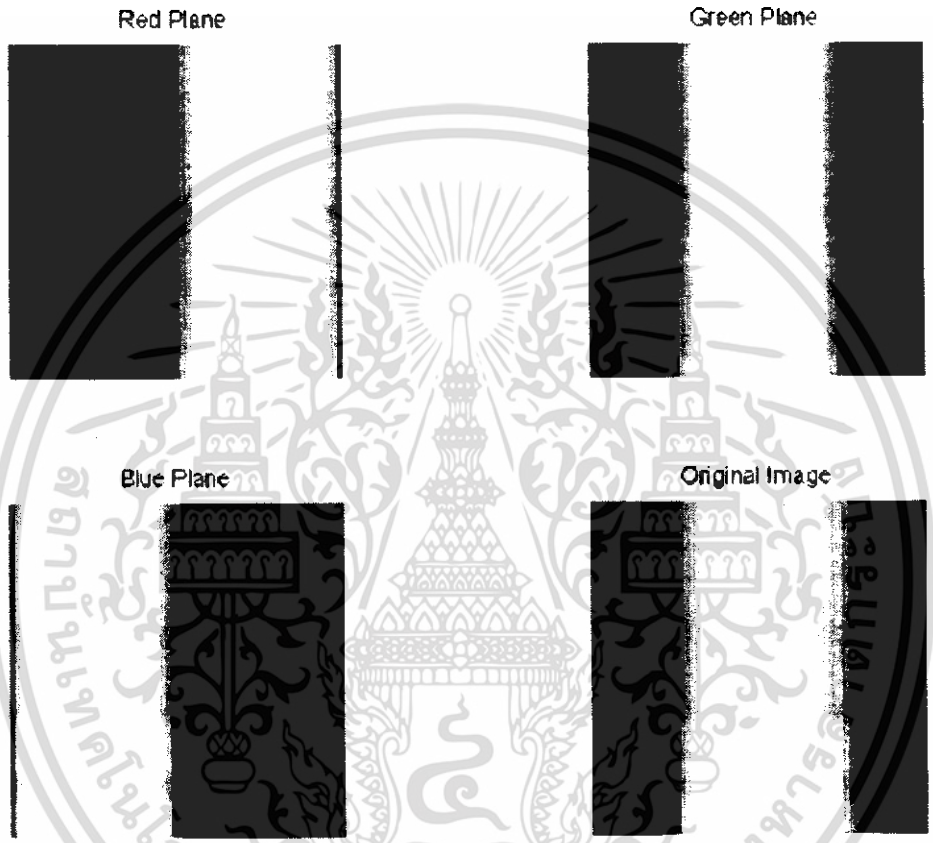
เพื่อที่จะแสดงหลักการของระนาบสีสามสีในอาร์จีบีของรูปภาพนั้น โค้ดที่อยู่ข้างล่างนี้เป็น ตัวอย่างของการสร้างอาร์จีบีของรูปภาพที่ประกอบด้วยการไม่รบกวนกันของพื้นที่ในสีแดง. สีเขียว และสีน้ำเงิน จากนั้นจึงสร้างรูปภาพสำหรับแต่ละระนาบของสี (แดง,เขียวและน้ำเงิน) โดยจะแสดงแต่ละระนาบของสีที่แยกจากกัน และยังแสดงถึงรูปภาพต้นฉบับด้วย

```
RGB = reshape(ones(64,1) × reshape(jet(64),1,192),[64,64,3]);

R = RGB(:,:,1);
G = RGB(:,:,2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
B    RGB(:,:,3):
imshow(R)
figure, imshow(G)
figure, imshow(B)
figure, imshow(RGB)
```



รูปที่ 2.6 ระนาบสีที่แยกจากกันของรูปภาพอาร์จีบี

ในแต่ละสีระนาบสีที่แยกจากกันในรูปภาพประกอบไปด้วยพื้นที่สีขาว สีขาวนี้จะแสดงถึงค่าสูงสุดของแต่ละสีที่แยกจากกัน ตัวอย่างเช่น ในรูภาพที่เป็นระนาบของสีแดง สีขาวจะแสดงถึงค่าสูงสุดของสีแดง ในขณะที่สีแดงถูกผสมด้วยสีเขียวหรือสีน้ำเงิน พิกเซลที่เป็นสีเทาจะปรากฏออกมา ส่วนพื้นที่ที่เป็นสีดำในรูปภาพนั้นจะแสดงค่าของพิกเซลที่ไม่มีค่าของสีแดงอยู่เลย เช่น R= = 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8 รูปภาพที่เป็นมัลติเฟรมอาร์เรย์ (Multiframe image Arrays)

ในโปรแกรมประยุกต์บางตัวนั้น จำเป็นต้องทำงานร่วมกับรูปภาพจำนวนมากที่เกี่ยวข้องกับเวลาของการเปิดดู เช่น Magnetic Resonance Imaging (MRI) หรือ เฟรมของภาพยนตร์ เป็นต้น

ในกล่องเครื่องมือการประมวลผลภาพนั้นรองรับการเก็บรูปภาพจำนวนมากในอาร์เรย์เดียวกัน แต่ละรูปภาพที่แยกจากกันจะเรียกว่าเฟรม อาร์เรย์รองรับหลายๆเฟรม มันจะถูกนำมาต่อกันเป็นขนาด 4 มิติ เช่น อาร์เรย์ของภาพห้าภาพที่มีอาร์จีบีแบบ 400 × 300 จะเป็น 400×300×1×5

โดยการใช้คำสั่ง cat ในการเก็บรูปแต่ละรูปที่แยกจากกันในไฟล์ของมัลติเฟรม เช่น ถ้ามีกลุ่มของรูปภาพ A1,A2,A3,A4 และ A5 เราสามารถที่จะเก็บทั้งหมดนี้ภายในอาร์เรย์เดียวโดยใช้คำสั่ง

A = cat(4, A1, A2, A3, A4, A5)

เมื่อต้องการที่จะแยกแต่ละเฟรมออกจากรูปภาพที่เป็นมัลติเฟรม ตัวอย่างเช่น มีรูปภาพที่เป็นมัลติเฟรม MULTI คำสั่งต่อไปนี้จะทำการแยกเฟรมที่สามออกมา

FRM3 = MULTI(:, :, :, 3)

ในรูปภาพที่เป็นมัลติเฟรมอาร์เรย์นั้น แต่ละรูปต้องมีขนาดเท่ากันมีจำนวนของระนาบที่เท่ากันด้วย ในดัชนีของรูปภาพที่เป็นมัลติเฟรม แต่ละรูปจะต้องใช้การเทียบเคียงแบบเดียวกัน

ข้อจำกัดในการใช้มัลติเฟรม

ฟังก์ชันจำนวนมากในกล่องเครื่องมือนั้นจะทำงานเฉพาะสองหรือสามมิติแรกเท่านั้น เราสามารถที่จะใช้อาร์เรย์ขนาดสี่มิติด้วยฟังก์ชันต่อไปนี้ได้ แต่จำเป็นต้องทำงานแต่ละเฟรมโดยแยกจากกัน เช่น ตัวอย่างนี้เป็นการแสดงเฟรมที่เจ็ดในอาร์เรย์ MULTI

imshow(MULTI(:, :, :, 7))

ถ้าเราส่งอาร์เรย์ไปยังฟังก์ชันและอาร์เรย์นั้น มีมิติมากกว่าการทำงานที่ฟังก์ชันกำหนดไว้ ผลลัพธ์ที่เกิดขึ้นไม่อาจคาดเดาได้ ในบางกรณีฟังก์ชันจะทำงานในเฟรมแรกของอาร์เรย์อย่างถูกต้อง แต่ในกรณีอื่นๆผลลัพธ์ที่เกิดจากกาทำงานนั้นอาจไม่มีความหมายอะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.9 สรุปชนิดของรูปภาพและคลาสต่างๆ

ตารางนี้เป็นการสรุปวิธีที่โปรแกรมแมทแลป แปลข้อมูลเมทริกซ์เป็นสีของพิกเซล ซึ่งขึ้นอยู่กับชนิดของรูปภาพและคลาสที่ใช้ในการเก็บ

ตารางที่ 2.1 ความสัมพันธ์ของโปรแกรมแมทแลปกับชนิดของไฟล์ภาพ

ชนิดรูปภาพ	**คลาส double**	**คลาส uint8 หรือ uint16**
ไบนารี	รูปภาพเป็นอาร์เรย์ขนาด m×n ของหนึ่งกับศูนย์ ซึ่งแฟลกจะเปิดใช้งาน	รูปภาพเป็นอาร์เรย์ขนาด m×n ของหนึ่งกับศูนย์ ซึ่งแฟลกจะเปิดใช้งานกล่องเครื่องมือไม่รองรับรูปภาพไบนารีที่เป็น uint16
ดัชนี	รูปภาพเป็นอาร์เรย์ขนาด m×n ของเลขจำนวนเต็มที่อยู่ในช่วง [1,p] การเทียบเคียงสีเป็นอาร์เรย์ขนาด p×3 ของเลขทศนิยมซึ่งค่าอยู่ในช่วง [0,1]	รูปภาพเป็นอาร์เรย์ขนาด m×n ของเลขจำนวนเต็มที่อยู่ในช่วง [0,p-1] การเทียบเคียงสีเป็นอาร์เรย์ขนาด p×3 ของเลขทศนิยมซึ่งค่าอยู่ในช่วง [0,1]
ความหนาแน่น	รูปภาพเป็นอาร์เรย์ขนาด m×n ของเลขทศนิยมซึ่งค่าเป็นความสัมพันธ์แบบเชิงเส้นโดยทั่วไปค่าอยู่ในช่วง [0,1] การเทียบเคียงสีเป็นขนาดอาร์เรย์ขนาด p×3 ของเลขทศนิยม ซึ่งค่าอยู่ในช่วง [0,1] และเป็นระดับของสีเทา	รูปภาพเป็นอาร์เรย์ขนาด m×n ของเลขจำนวนเต็มซึ่งค่าเป็นความสัมพันธ์แบบเชิงเส้นโดยทั่วไปค่าจะอยู่ในช่วง [0,255] หรือ [0,65535] การเทียบเคียงสีเป็นอาร์เรย์ขนาด p×3 ของเลขทศนิยม ซึ่งค่าอยู่ในช่วง [0,1] และเป็นระดับของสีเทา
อาร์จีบี	รูปภาพเป็นอาร์เรย์ขนาด m×n×3 ของเลขทศนิยม ซึ่งค่าอยู่ในช่วง	รูปภาพเป็นอาร์เรย์ขนาด m×n×3 ของเลขจำนวนเต็ม ซึ่งค่าอยู่ในช่วง
ชนิดรูปภาพ	**คลาส double**	**คลาส uint8 หรือ uint16**
	[0,1]	[0,255] หรือ [0,65535]

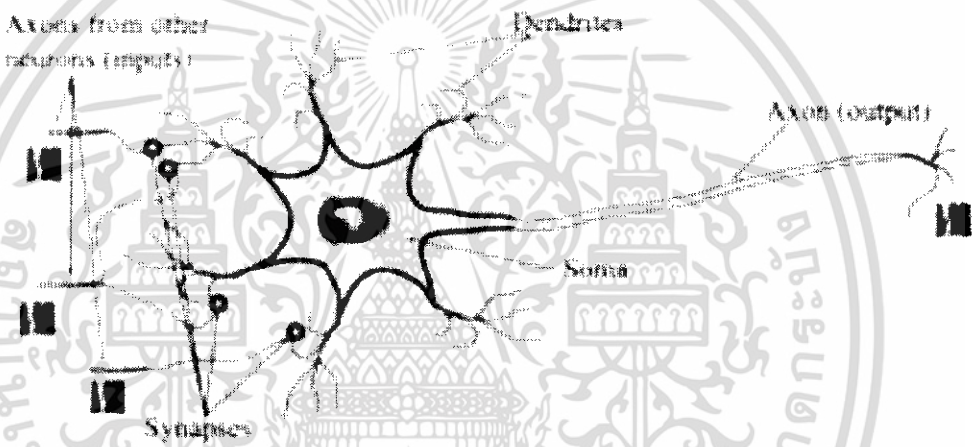
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 นิวรอลเน็ทเวิร์ค (Neural Network : NN)

Neural Network (NN) เป็นคำย่อมาจาก Artificial Neural Network (ANN) ซึ่งมีความหมายว่า "ระบบการทำงานเรียนแบบเซลล์สมองของมนุษย์" โดยที่กล่าวถึงนี้ สามารถใช้เป็นได้ทั้ง โปรแกรมคอมพิวเตอร์ วงจรอิเลคทรอนิคส์ วงจรดิจิตอล หรือ ระบบผสมทั้งซอฟแวร์และฮาร์ตแวร์ แต่ NN นั้นจะต้องมีคุณสมบัติที่สำคัญๆ 2 ข้อ ได้แก่

1. สามารถเรียนรู้ และเก็บความรู้ที่เรียนรู้ไว้ได้ (Learn and Knowledge)
2. สามารถนำความรู้ที่เก็บไว้มาใช้งานในภายหลังได้อย่างถูกต้อง

โครงสร้างของสมองมนุษย์ ประกอบด้วยประกอบขึ้นมาจากการรวมกันของเซลล์สมอง (Neuron) โดยจุดที่เชื่อมต่อกันของเซลล์สมองคือ ไซแนฟซ์ (Synapses)

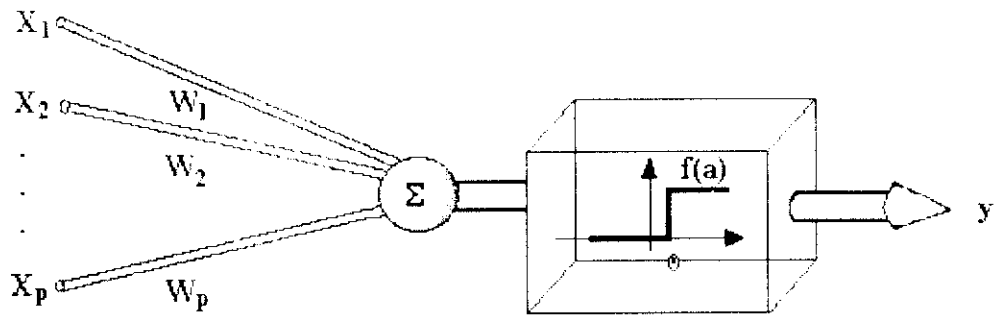


รูปที่ 2.7 เซลล์สมองของมนุษย์

โดยในเซลล์สมองแต่ละเซลล์จะมีส่วนประกอบที่สำคัญๆ คือ นิวเคลียส (Nucleus) ทำหน้าที่ประมวลผลการกระตุ้นทางไฟฟ้า (Electrical Impulses) ที่ส่งมาจากเซลล์อื่นผ่านทาง เดนไดร์ท (Dendrites) และเมื่อประมวลผลแล้วก็จะส่งออกไปสู่ แอกซอน (Axon) ไปยัง เดนไดร์ท (Dendrites) สู่เซลล์อื่นต่อไปเรื่อยๆ และระหว่างไซแนฟส์จะมีช่องว่างไซแนฟส์ (Synaptic Gap) เพื่อปรับระดับสัญญาณไฟฟ้า

จากโครงสร้างของเซลล์ประสาทดังกล่าวจึงเริ่มมีแบบจำลองเซลล์สมองออกมามากมาย แต่ที่ได้รับความนิยมมากคือแบบจำลองของ Warren McCullock กับ Walter Pitts (McCullock & Pitts) ที่นำเสนอด้วยวงจรอิเลคทรอนิคส์

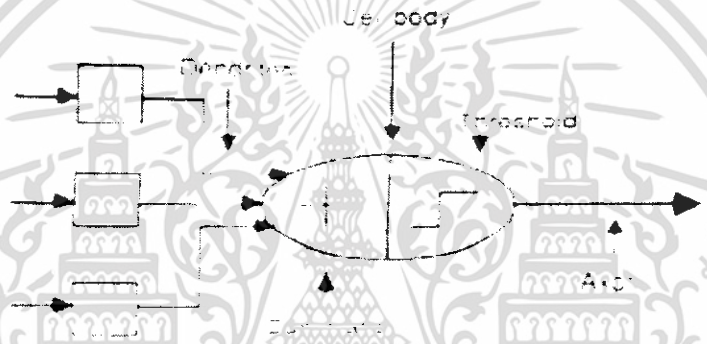
62832
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 Neural Network ของ McCullock & Pitts

2.2.1 แบบจำลองทางคณิตศาสตร์ของเซลล์ประสาทเดียว

โครงสร้างแบบจำลองอย่างง่ายของเซลล์ประสาทสามารถเขียนได้ดังแบบนี้



รูปที่ 2.9 โครงสร้างแบบจำลองอย่างง่ายของเซลล์ประสาท

ถ้าสัญญาณที่ส่งออกมาจากแต่ละเซลล์ประสาท คือ n_i จะสามารถเขียนสมการคณิตศาสตร์ของ n_i ที่เวลา $t+1$ ได้ดังนี้

$$n_i(t+1) = f\left(\sum_j \omega_{ij} n_j(t) - T_i\right) \qquad (2.1)$$

เมื่อ w_{ij} คือ ค่าความต้านทานของไซแนปที่ j ของเซลล์ประสาทตัวที่ i

$n_j(t)$ คือ สัญญาณที่เซลล์ประสาทตัวที่ j ส่งออกมาที่เวลา t

T_i คือ ค่าเปรียบเทียบ ซึ่งเรียกว่า Threshold

* f() คือ ฟังก์ชัน f ที่ถูกนิยามดังต่อไปนี้

f(x)={ 1 ถ้า x>=0

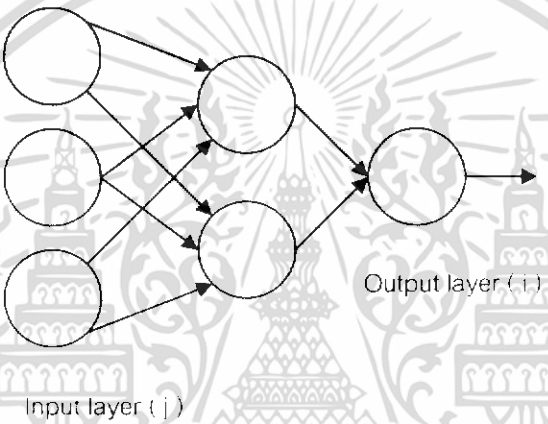
0 ถ้า x<0 }

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 เทคนิคการเรียนรู้แบบแพร่กลับ (Back propagation)

เทคนิคการเรียนรู้ของโครงข่ายประสาทเทียมมีอยู่หลายวิธีด้วยกัน แต่วิธีที่นิยมและแพร่หลายมากที่สุด คือ วิธีการเรียนรู้การเปลี่ยนแปลงของค่า w_{ij} โดยใช้วิธีที่เรียกว่าการแพร่กลับ (Back Propagation) การแพร่กลับพัฒนามาจากสมการพื้นฐานของการเรียนรู้แบบ McCullock และ Pitts โดยการพิจารณาฟังก์ชันของความคลาดเคลื่อนระหว่างค่าเป้าหมายที่ต้องการ กับ ค่าของสัญญาณที่ส่งออกมาจากตัว neuron ในขอบข่ายของค่าจริงด้วย (ไม่ใช่แค่ 0 กับ 1 หรือ -1 กับ +1)

โครงข่ายประสาทเทียมที่มีการเรียนรู้โดยการแพร่กลับนี้จะประกอบไปด้วย กลุ่มของเซลล์ประสาทต่างๆ จัดตัวกันเป็นชั้นๆ ดังนี้



รูปที่ 2.10 โครงประสาทเทียมที่พื้นฐาน

แต่ละขาของสัญญาณเข้าจะมีค่าความต้านทาน หรือ ค่าของสัญญาณออกของแต่ละเซลล์คำนวณได้จากสมการดังต่อไปนี้

$$f(\alpha) = \frac{1}{1+e^{-(\alpha-\theta)}} \quad \text{(Sigmoid function)} \qquad (2.2)$$

เมื่อ e คือค่า log ฐานธรรมชาติ

a คือ $\sum \omega_i X_i$

θ คือค่าคงที่

1.การกำหนดค่าความต้านทาน ω_i และ θ ของแต่ละเซลล์อย่างสุ่ม

2.ป้อนเวคเตอร์ $(X_1, X_2, \ldots, X_n)$ และสัญญาณที่ต้องการให้ปรากฏที่ชั้นนอกสุด $(d_1, d_2, \ldots, d_m)$

3.จากเวคเตอร์ $(X_1, X_2, \ldots, X_n)$ คำนวณหาค่าสัญญาณที่ส่งออกจากเซลล์แต่ละเซลล์ที่ชั้นนอกสุด โดยไล่ไปทีละชั้น จากชั้นที่ 1 การคำนวณโดยใช้สูตร $f(\alpha)$ ข้างต้น ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$(Y_1,Y_2,\ldots,Y_m)$ เป็นค่าของสัญญาณที่ได้จากการคำนวณเวคเตอร์ $(d_1,d_2,\ldots,d_m)$ จะแตกต่างจาก เวคเตอร์ $(Y_1,Y_2,\ldots,Y_m)$

4.ปรับค่าของความต้านทานของแต่ละเซลล์ โดยเริ่มจากชั้นนอกสุดไล่ไปจนถึงชั้นที่ 1 ค่าความต้านทานใหม่ที่ปรับดังสมการต่อไปนี้

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta\delta_i\chi'_i \qquad (2.3)$$

เมื่อ ω_{ij} คือ ความต้านทานระหว่างเซลล์ประสาท i และ j ดังรูปด้านบน

η คือ อัตราการเรียนรู้ โดยที่ จะมีค่าไม่เกิน 1 (กำหนดเอง)

δ_j หาได้โดย

ก.ถ้าเซลล์ j เป็นเซลล์ชั้นนอกสุด ค่าของ j จะเท่ากับ

$$\delta_j=y_j(1-y_j)(d_{ij}-y_{ij}) \qquad (2.4)$$

ข.ถ้าเซลล์ j เป็นเซลล์ชั้นที่ซ่อนอยู่ ค่าของ j จะเท่ากับ

$$\delta_j=X'_j(1-X'_j)\sum_k \delta_k\omega_{jk} \qquad (2.5)$$

เมื่อ X'_j คือ ค่าสัญญาณขาเข้าของเซลล์ j

jk คือ จำนวนเซลล์ทั้งหมดบนชั้นที่อยู่ถัดจากเซลล์ j ขึ้นไป

5.ทำซ้ำข้อ2 ถึง ข้อ5 จนกว่าค่าความแตกต่าง (E) จะยอมรับได้ กำหนดดังนี้

$$E=\frac{1}{2}\sum_i(d_i-y_i)^2$$

หลังจากเราสอนโครงข่ายให้เรียนรู้แล้ว ต่อไปจะนำมันมาใช้งาน โดยป้อนเวคเตอร์เข้าไปแล้วคำนวณหาค่าสัญญาณที่ออกมาจากแต่ละเซลล์โดยสมการข้างต้น(Sigmoid Function)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 เทคนิคการแบ่งกลุ่ม (Clustering Technique)

จากแนวความคิดในเรื่องการจำแนกใบหน้าบุคคลนั้น จะต้องอาศัยปัจจัยหลายๆ อย่างประกอบกัน โดยที่ปัจจัยเหล่านั้นจะเกี่ยวข้องกันหรือไม่ก็ได้ จึงได้ทำการของ ปัจจัยเหล่านั้นมาด้วยกัน โดยในการจับกลุ่มนี้จะแทนด้วยปัจจัยที่จำเป็นด้วย Input node ของนิวรอลเน็ทเวิร์ค แล้วทำการจับกลุ่มด้วยวิธีการที่เรียกว่า K-Medoid

โดยวิธีการ K-Medoid นี้เป็นวิธีการแบ่งกลุ่มข้อมูลออกเป็นจำนวน K กลุ่ม โดยจะอาศัยข้อมูลบางตัวให้เป็นจุดศูนย์กลางของกลุ่มข้อมูล

จากสมการ

$$E = \sum_{j=1}^{K} \sum_{p \in C_j} |p-o_j| \qquad (2.6)$$

เมื่อ E เป็นผลรวมของค่า absolute-error ของข้อมูลทั้งหมด

C_j เป็นกลุ่มที่ j

K เป็นจำนวนของกลุ่มทั้งหมด

p เป็นสมาชิกของแต่ละกลุ่มที่ไม่ใช่จุดอ้างอิง

o_j เป็นจุดอ้างอิงในกลุ่ม j

โดย Set ของ o_j ที่ทำให้มีค่า E ที่น้อยที่สุดจะทำให้ได้กลุ่มของข้อมูลที่เหมาะสม และสามารถหาค่า o_j ได้ดังนี้

กำหนดให้

K เป็นจำนวนของกลุ่มข้อมูลที่ต้องการแบ่ง

1) ทำการสุ่มค่าอ้างอิงขึ้นมาจากข้อมูลทั้งหมด สำหมดหรับเป็นค่าอ้างอิงในแต่ละกลุ่ม
2) จับกลุ่มข้อมูลที่เหลือ โดยให้อยู่ในกลุ่มที่มีค่าอ้างอิงใกล้กันมากที่สุด
3) สุ่มค่าอ้างอิงขึ้นมาใหม่ 1 ค่า
4) คำนวณค่า E ค่าใหม่โดยแทนค่าอ้างอิงเดิมที่ละค่าด้วยค่าอ้างอิงใหม่ที่สุ่มขึ้นมา
5) ถ้าค่า E ที่คำนวณได้ใหม่มีค่าน้อยลง ให้ถือค่าอ้างอิงใหม่แทนค่าเดิมที่ถูกแทนแล้วทำให้มีค่า E ที่น้อยลง
6) ทำซ้ำตั้งแต่ 2 จนกว่าจะไม่มีการเปลี่ยนแปลงค่าอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างและการทำงานของโปรแกรม

ในโปรแกรม นี้จะมีอยู่สองส่วนหลักๆ คือ ส่วนของการค้นหาองค์ประกอบของใบหน้า และ ส่วนของการจัดกลุ่มฟีเจอร์บนนิวรอลเน็ทเวิร์ค

ในบทที่นี้ 3 จะกล่าวถึง โครงสร้างของระบบค้นหาองค์ประกอบของใบหน้า การสอนนิวรอลเน็ทเวิร์ค การแบ่งกลุ่มโดยองค์ประกอบของใบหน้า การสร้างคลัสเตอร์นิวรอลเน็ทเวิร์ค

3.1 การสอนนิวรอลเน็ทเวิร์คแบบแพร่กลับ

เมื่อได้ชุดข้อมูลขององค์ประกอบของใบหน้าแล้ว จะนำเข้ามาสอนนิวรอลเน็ทเวิร์ค ซึ่งนิวรอลเน็ทเวิร์คที่ใช้นี้จะเป็นนิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation Neural Network) ซึ่งถือเป็นรูปแบบทั่วไปที่นิยมกัน และนิวรอลเน็ทเวิร์คแบบแพร่กลับนี้พัฒนาขึ้นมาจากโปรแกรม ภาษา C++ เพื่อที่จะได้สามารถนำมาสร้างเป็นคลัสเตอร์นิวรอลเน็ทเวิร์คต่อไป

ในการสร้างนิวรอลเน็ทเวิร์คนี้จะสร้าง อินพุทยูนิท ขึ้นมาโดยให้มีจำนวนเท่าองค์ประกอบของภาพที่ค้นหามาได้ จากนั้นจึงสร้าง เอาพุทยูนิท ขึ้นมาเท่ากับจำนวนของค่าที่ต้องการ โดยในโปรแกรมนี้จะแบ่งออกมาเป็น ชาย หรือ หญิง จึงมีเอาท์พุทยูนิทเพียง 1 ยูนิท โดยค่าที่ได้จากนิวรอลเน็ทเวิร์คจะกำหนดให้ ชายเป็น 1 และหญิงเป็น 0

จากนั้นจะทำการสุ่มค่าของน้ำหนักของการเชื่อมต่อและค่าเปรียบเทียบ (Threshold)

แล้วจึงทำการสอนนิวรอลเน็ทเวิร์ค โดยการนำองค์ประกอบของใบหน้าเข้ามาในนิวรอลเน็ทเวิร์คทีละชุด มาเป็นค่าเอาพุทของอินพุทยูนิท แล้วอิดเดนยูนิทหาค่าเอาพุทจากสมการ

$$f(\alpha) = \frac{1}{1+e^{-(\alpha-\theta_j)}} \qquad (3.1)$$

$$\alpha \qquad \sum_{j,i} (X_i \times W_{ij}) \qquad (3.2)$$

เมื่อ X เป็นเอาพุทจากยูนิท i ในชั้นก่อนหน้า

W เป็นค่าน้ำหนักจากยูนิท i มายังยูนิท j

เพื่อส่งออกไปเป็นเอาพุทของอิดเดนยูนิท ในแต่ละเอาพุทของแต่ละยูนิท และเอาพุทยูนิทก็จะหาค่าเอาพุทด้วยวิธีเดียวกัน โดยเอาพุทของเอาพุทยูนิทนี้จะมีค่าเป็นเอาพุทของนิวรอลเน็ทเวิร์คด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ค่าเอาพุทของนิวรอลเน็ทเวิร์คแล้ว จะทำการปรับค่าน้ำหนักใหม่ตามสมาการดังต่อไปนี้

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p) \quad (3.3)$$

$$\Delta w_{ij}(p) = \alpha + x_i(p) + \delta_j(p) \quad (3.4)$$

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times e \quad (3.5)$$

เมื่อ $w_{ij}(p)$ เป็นค่าน้ำหนักจาก i ไป j ในรอบที่ p

$\Delta w_{ij}(p)$ เป็นค่าสำหรับปรับน้ำหนักจาก i ไป j

α เป็นค่าการเรียนรู้ซึ่งต้องกำหนดโดยผู้ใช้

$\delta_j(p)$ เป็นค่าเกรย์เดียนที่ใช้ในการปรับน้ำหนัก

$x_i(p)$ เป็นค่าอินพุทของยูนิท i จากชั้นก่อนหน้า

y_i เป็นค่าเอาพุทของยูนิท j

ส่วนค่า e จะมีการคำนวณที่แตกต่างกันในเอาพุทยูนิทและฮิดเดนยูนิท โดยในเอาพุทยูนิทจะหาค่า e ดังนี้

$$e = y_{dk}(p) - y_k(p) \quad (3.6)$$

เมื่อ e เป็นค่าความคลาดเคลื่อนของผลลัพธ์ของนิวรอลเน็ทเวิร์ค

$y_{dk}(p)$ เป็นผลลัพธ์ที่ถูกต้อง

$y_k(p)$ เป็นผลลัพธ์ที่ได้ออกมาจากเอาพุทยูนิท

และในฮิดเดนยูนิทจะหาค่า e ดังนี้

$$e = \sum_k \delta_k(p) \times w_{jk}(p) \quad (3.7)$$

เมื่อ $\delta_k(p)$ เป็นค่าเกรเดียน k ที่คำนวณได้ชั้นถัดไปที่เชื่อต่อกันฮิดเดนยูนิท j

$w_{jk}(p)$ เป็นน้ำหนักของชั้นถัดไปที่เชื่อต่อกันฮิดเดนยูนิท j

แล้ววนซ้ำไปเรื่อยๆ จนกว่าจะมีค่า SSE (sum square error) ต่ำกว่า 0.001 โดยค่า SSE นี้หาได้จากสมาการ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$SSE = \sum (y_{dk}(p) - y_k(p))^2 \quad (3.8)$$

เมื่อ $y_{dk}(p)$ เป็นผลลัพธ์ที่ถูกต้อง

$y_k(p)$ เป็นผลลัพธ์ที่ได้ออกมาจากเอาพุทยูนิท

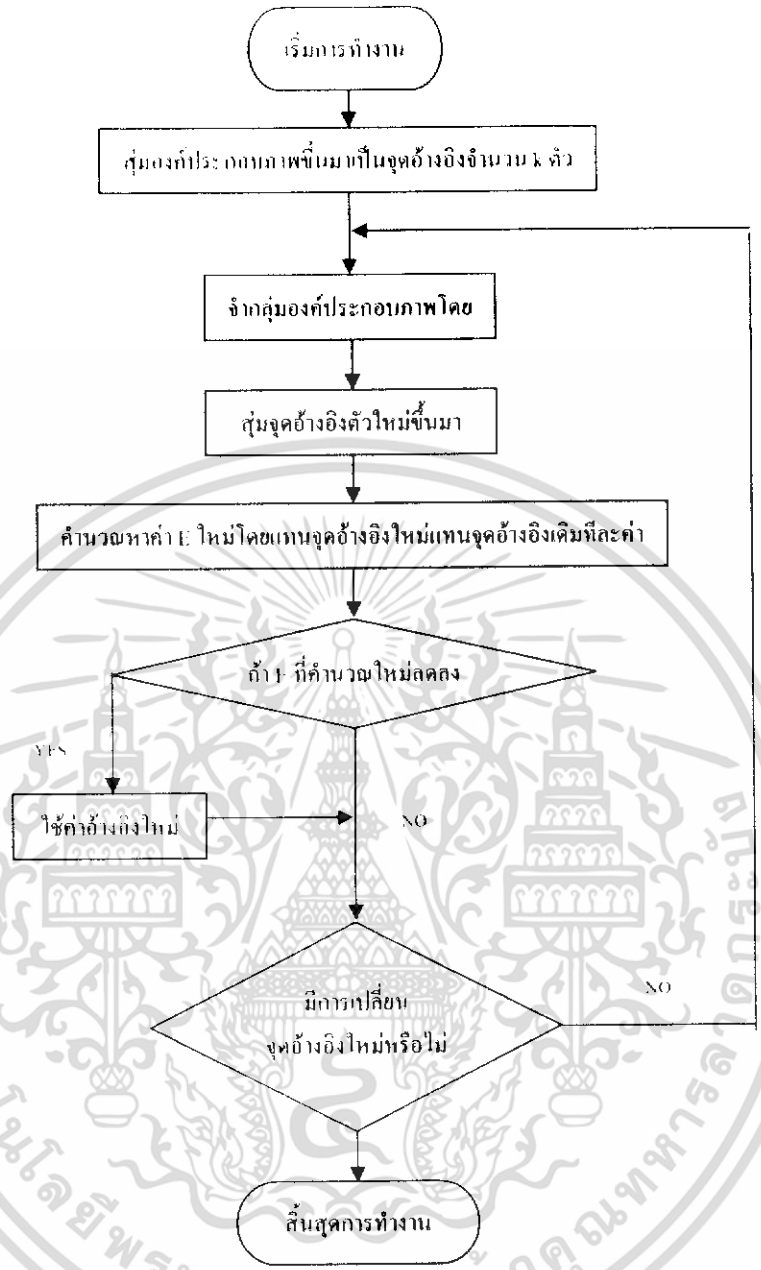
และเมื่อได้ค่า SSE ต่ำกว่า 0.001 แล้วก็จะถือว่าเสร็จสิ้นการสอนนิวรอลเน็ทเวิร์ค

ส่วนการทำงานของนิวรอลเน็ทเวิร์คก็จะเป็นการหาค่าเอาพุทออกมา โดยไม่มีการกลับไปปรับค่าน้ำหนัก

3.2 การจับกลุ่มฟีเจอร์

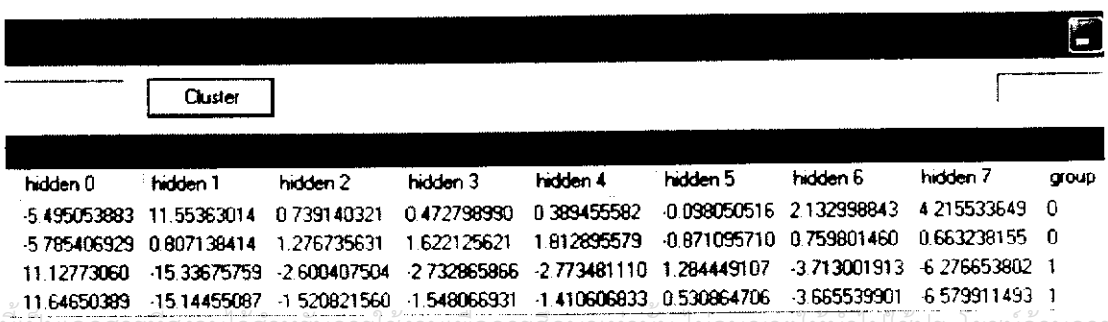
จะทำการจับกลุ่มโดยแทนที่ฟีเจอร์แต่ละอย่าง ด้วยอินพุทยูนิท ของนิวรอลเน็ทเวิร์ค ซึ่งในการจับกลุ่มนี้จะดูจากค่าน้ำหนัก (weight) ที่อยู่บนการเชื่อมต่อจากอินพุทยูนิทนั้นๆ ไปยังอิดเดนยูนิทในชั้นถัดไปของนิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation Neural Network) ที่ผ่านการสอนแล้ว และในการจับกลุ่มนี้ใช้อัลกอริทึมที่ K-Medoid มาจับกลุ่มซึ่งสามารถเขียนเป็น Flow Chart ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 Flow chart การจับกลุ่มองค์ประกอบในหน้า

โดยในการจับกลุ่มนี้เกินจากการสุ่มจึงอาจจะเกิดความผิดพลาดขึ้นได้ จึงต้องมีการทำซ้ำประมาณ 100-1000 รอบเพื่อที่จะลดความผิดพลาดในการจับกลุ่มลง



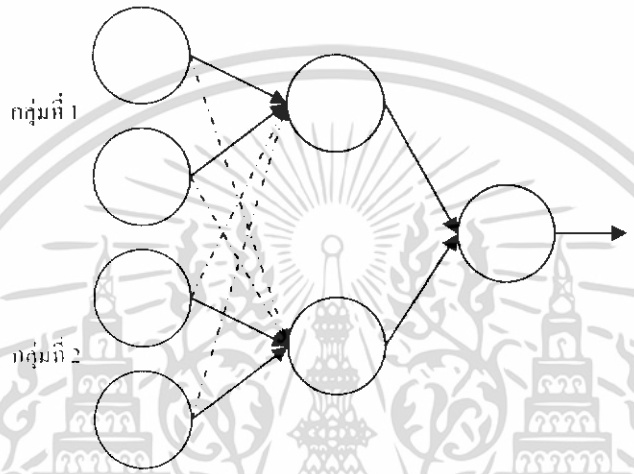
hidden 0	hidden 1	hidden 2	hidden 3	hidden 4	hidden 5	hidden 6	hidden 7	group
-5.495053883	11.55363014	0.739140321	0.472798930	0.389455582	-0.098050516	2.132998843	4.215533649	0
-5.785406929	0.807138414	1.276735631	1.622125621	1.812895579	-0.871095710	0.759801460	0.663238155	0
11.12773060	-15.33675759	-2.600407504	-2.732865866	-2.773481110	1.284449107	-3.713001913	-6.276653802	1
11.64650389	-15.14455087	-1.520821560	-1.548066931	-1.410606833	0.530864706	-3.665539901	-6.579911493	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 ผลลัพธ์การจับกลุ่มน้ำหนักของนิวรอลเน็ทเวิร์คที่สอนด้วยข้อมูลของ Iris

3.3 การสร้างและสอนคลัสเตอร์นิวรอลเน็ทเวิร์ค (Cluster Neural Network)

ในการสร้างคลัสเตอร์นิวรอลเน็ทเวิร์คนี้ จะสร้างขึ้นโดยใช้ผลจากการจัดกลุ่มอินพุทยูนิท (ใช้ผลจากการจัดกลุ่มของกลุ่มองค์ประกอบใบหน้า) มาใช้ในการตัดส่วนเชื่อมต่อที่ไม่เกี่ยวข้องกัน โดยให้อินพุทยูนิทในกลุ่มเดียวกันเชื่อมต่อไปยังฮิดเดน ยูนิทในกลุ่มเดียวกันเท่านั้น ดังภาพต่อไปนี้



รูปที่ 3.3 แบบจำลองของคลัสเตอร์นิวรอลเน็ทเวิร์ค

รูปนี้จะแสดงนิวรอลเน็ทเวิร์คแบบคลัสเตอร์ โดยเส้นประจะแสดงถึงการเชื่อมต่อที่โดนตัดออกไป

จากนั้นจะทำการสุ่มค่าน้ำหนักของส่วนเชื่อมต่อที่เหลืออยู่ทั้งหมดขึ้นมาใหม่ แล้วจึงสอนคลัสเตอร์นิวรอลเน็ทเวิร์คด้วยวิธีการเดียวกับนิวรอลเน็ทเวิร์คแบบแพร่กลับ แต่ไม่

ในการใช้งานนิวรอลเน็ทเวิร์คนั้นจะคล้ายกับการสอนนิวรอลเน็ทเวิร์ค โดยจะต่างกันเพียงไม่จำเป็นต้องมีการปรับค่าน้ำหนักเท่านั้น

3.4 โครงสร้างของระบบค้นหาฟีเจอร์

ในระบบจัดกลุ่มฟีเจอร์นี้ มีการใช้ฟีเจอร์ของใบหน้าของมนุษย์ประกอบด้วย ดังนั้นการประมวลผลภาพจึงเข้ามามีบทบาท ในส่วนของการประมวลผลภาพนั้นมีขั้นตอนในการแยก ตา ปาก จมูก ออกจากข้อมูลภาพทั้งหมด โดยมีขั้นตอนการทำงานของโปรแกรมดังนี้คือ ในขั้นแรกจะเป็นขั้นตอนการเตรียมภาพโดยนำภาพถ่าย RGB มาแปลงเป็นภาพแบบไบนารีซึ่งจะเหลือข้อมูลภาพเพียงสองระดับ คือ 0 กับ 1 จากนั้นก็จะทำการกำจัดสัญญาณรบกวนแล้วทำการหาขอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของรูปภาพแล้วจึงค้นหาบริเวณของตา ปาก และจมูก โดยจะพิจารณาจากตำแหน่งที่การจะเป็นจริงขนาดของฟีเจอร์เหล่านี้เข้าสู่การจำแนกต่อไป

3.1.1 การประมวลผลภาพ

ในการประมวลผลภาพภาพสามารถแบ่งการทำงานออกได้เป็น

1. การเตรียมข้อมูลภาพ
2. การแปลงข้อมูลภาพไบนารี
3. การหาตำแหน่งของดวงตา
4. การหาตำแหน่งของจมูกและปาก
5. การแยกตา ปาก จมูกออกจากภาพ

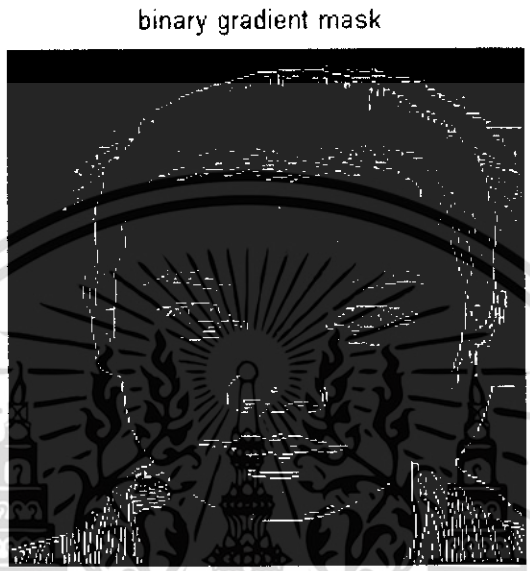
การแปลงข้อมูลภาพไบนารี

ในขั้นตอนแรกจะเป็นการเตรียมข้อมูลภาพเพื่อใช้ในการประมวลผล เนื่องรูปถ่ายที่ได้มาจะเป็นภาพสี RGB

รูปที่ 3.4 ภาพตัวอย่างรูปถ่ายจากกล้องดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

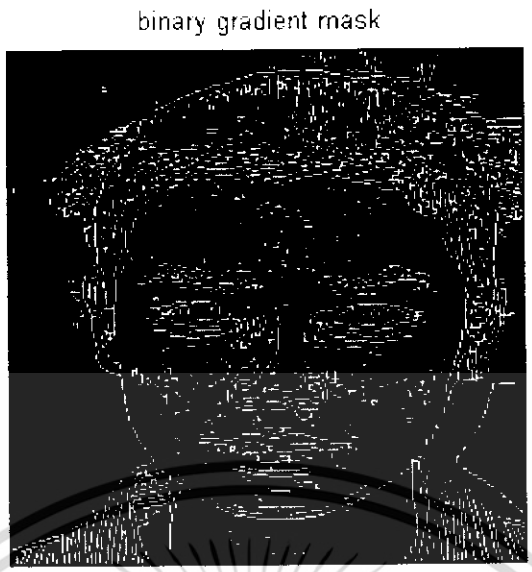
เพื่อทำให้ข้อมูลภาพมีรูปแบบที่เหมาะสมในการประมวลผล จึงต้องทำการแปลงข้อมูลของภาพให้เป็นภาพไบนารี คือ มีระดับสีเพียง 2 ระดับได้แก่สีดำเป็น 1 และสีขาวเป็น 0 วิธีการแปลงข้อมูลภาพให้เป็นไบนารีนั้น ทำได้โดยการเลือกค่า threshold ที่เหมาะสม โดยข้อมูลภาพที่มีระดับความสว่างสูงกว่าค่า threshold จะถูกปรับเป็น 0 หรือสีขาว ส่วนค่าระดับความสว่างที่มีค่าต่ำกว่าค่า threshold จะถูกปรับเป็น 1 หรือสีดำ



รูปที่ 3.5 ภาพที่ได้จากการแปลงเป็นไบนารี

ภาพที่เป็นภาพไบนารีแล้วจะได้ ตา ปาก จมูก ออกจากใบหน้าเป็นส่วนๆอย่างชัดเจนเนื่องมาจากการเลือกค่า threshold ที่เหมาะสม แต่บางครั้งมีบางกรณีที่เลือกค่า threshold สูงหรือต่ำเกินไปจะทำให้ไม่สามารถแยกแยะ ตา ปาก จมูก ออกจากใบหน้าได้หรือได้ขนาดที่ผิดพลาดไป ดังนั้นต้องเลือกค่า threshold ที่เหมาะสมที่สุดในการแปลงเป็นภาพไบนารีเพื่อให้เกิดความเสียหายของข้อมูลน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงการเลือกค่า threshold ที่ต่ำเกินไป

รูปที่ 3.7 แสดงการเลือกค่า threshold มากเกินไป

การกำจัดสิ่งรบกวน

ในขั้นตอนของการแปลงให้เป็นภาพไบนารีนั้นอาจทำให้เกิดพิกเซลสีดำเดี่ยวๆจำนวนมาก ซึ่งเป็นสิ่งรบกวน (Noise) ทำให้เกิดผลต่อการประมวลผลในขั้นตอนต่อไปได้ ดังนั้นจึงต้องกำจัดพิกเซลที่เป็นสิ่งรบกวนเหล่านี้ออกจากภาพ โดยการหาตำแหน่งของพิกเซลสีดำที่ไม่มีพิกเซลข้างเคียงเลย แล้วเปลี่ยนให้เป็นพิกเซลสีขาวแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาตำแหน่งของดวงตา

หลังจากที่ภาพได้ผ่านการแปลงให้เป็นไบนารีแล้ว สามารถหาตำแหน่งของดวงตาได้โดยใช้วิธีการพิจารณาว่าดวงตาคือกลุ่มของพิกเซลสีดำหรือพิกเซลที่มีค่าเป็น 1 ติดกัน ดังนั้นภาพที่ผ่านการแปลงเป็นไบนารีแล้วนี้จะมีกลุ่มของพิกเซลดังกล่าวอยู่หลายกลุ่มทั้งที่เป็นดวงตา จมูก ปาก หรือส่วนอื่นๆของภาพ โดยที่ดวงตาควรจะอยู่ในกรอบสี่เหลี่ยม โดยในแนวแกน X=102-153 และแกน Y=61-204

รูปที่ 3.8 แสดงกรอบสี่เหลี่ยมที่ใช้ในการหาดวงตา

การหาตำแหน่งของจมูกและปาก

เช่นเดียวกันดวงตาจมูกและปากจะมีกลุ่มพิกเซลที่มีค่าเป็น 1 ติดกัน โดยตำแหน่งของจมูกและปากควรจะอยู่ในกรอบสี่เหลี่ยม โดยในแนวแกน X=145-200 และแกน Y=84-173

รูปที่ 3.9 แสดงกรอบสี่เหลี่ยมที่ใช้หาจมูกและปาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแยกตา ปาก จมูกออกจากภาพ

หลังจากที่รู้ตำแหน่งของฟีเจอร์ต่างๆแล้ว ก็จะทำการแยกแต่ละฟีเจอร์ออกจากภาพโดยการสแกนจากซ้ายไปขวาเพื่อหาขอบซ้าย สแกนจากบนลงล่างเพื่อหาขอบบน สแกนจากขวาไปซ้ายเพื่อหาขอบขวา และสแกนจากล่างขึ้นบนเพื่อหาขอบล่าง จากนั้นก็ตีกรอบตามขอบที่หาได้จะได้กรอบรอบฟีเจอร์เพื่อนำมาหาขนาดต่อไป

รูปที่ 3.10 แสดงขนาดของดวงตาที่หาได้

รูปที่ 3.11 แสดงขนาดของปากและจมูกที่หาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 โครงสร้างของโปรแกรมส่วนค้นหาองค์ประกอบบนใบหน้า

โปรแกรม Face Classification แบ่งออกเป็น 2 ส่วน คือ ส่วนการประมวลผลภาพและส่วนของการจำแนกใบหน้า โดยในส่วนของการประมวลผลภาพนั้นเขียนขึ้นโดยภาษา Matlab ส่วนการจำแนกใบหน้าเขียนขึ้นโดยภาษา C# และทำงานร่วมกันโดยการ Compile M-File มาเป็นไฟล์สกุล dll เพื่อนำฟังก์ชันไปเรียกใช้ในการจำแนกใบหน้า โดยแบ่งโปรแกรมออกเป็นส่วนย่อยดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

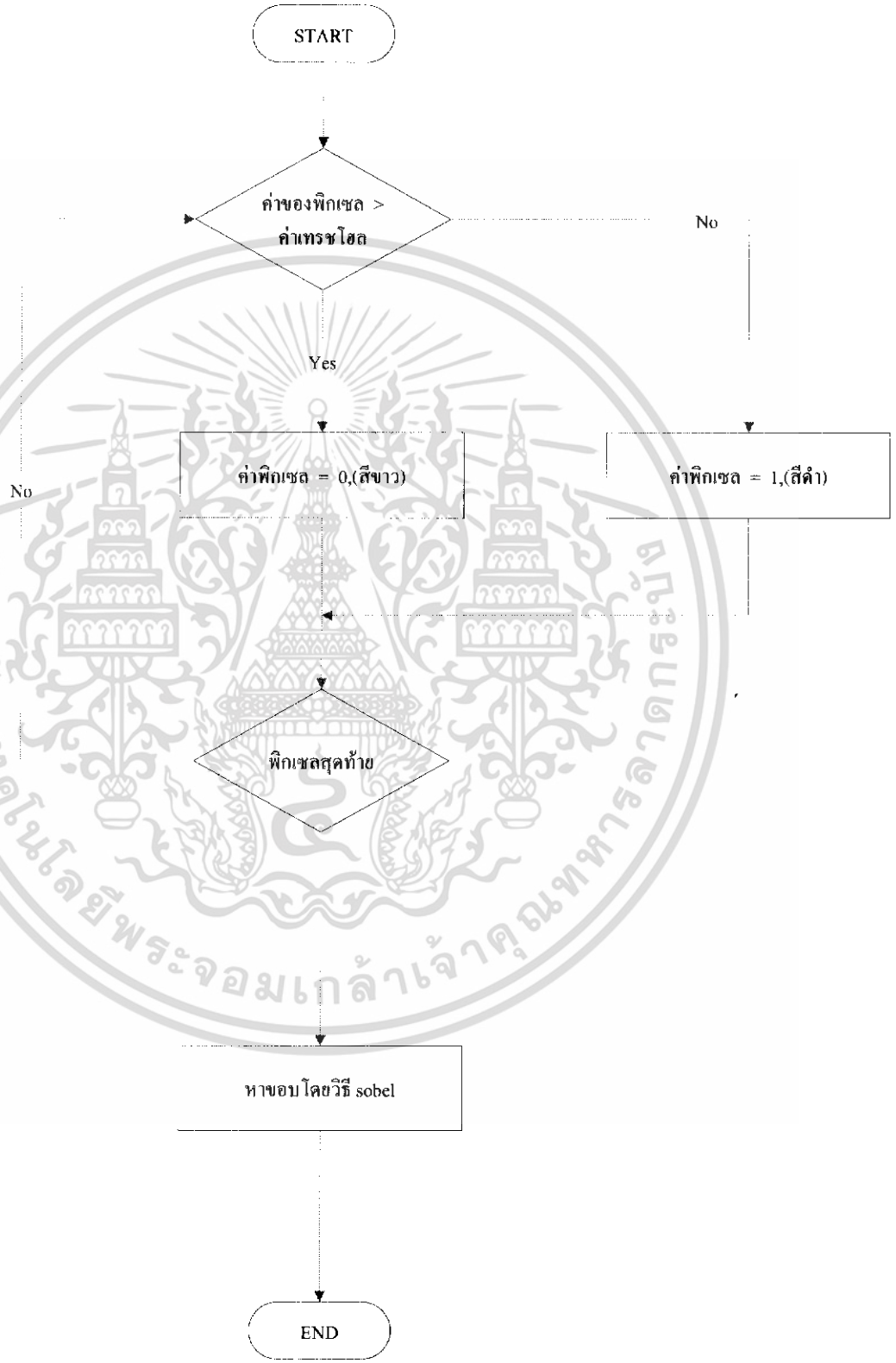
โปรแกรมการทำ Preprocessing

รูปที่ 3.12 Flow chart ของโปรแกรมทำ Preprocessing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการทำ Binarization

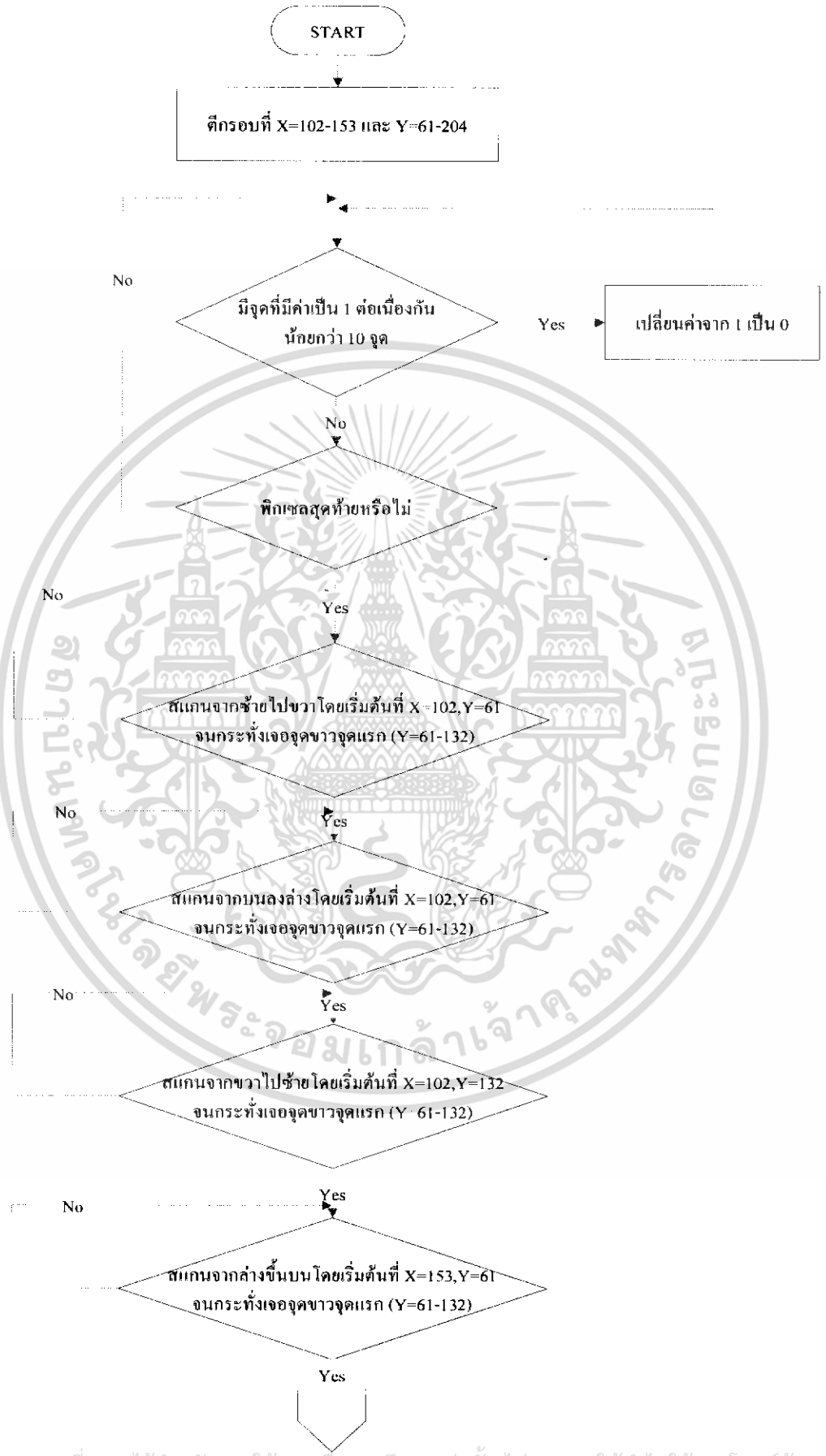
เมื่อทำการปรับปรุงภาพเบื้องต้นแล้วก็จะนำรูปที่ปรับปรุงมาแล้วนั้นมาแปลงเป็นภาพสีขาวดำ แล้วทำการหาขอบของภาพ



รูปที่ 3.13 Flow chart ของโปรแกรมการทำ Binarization

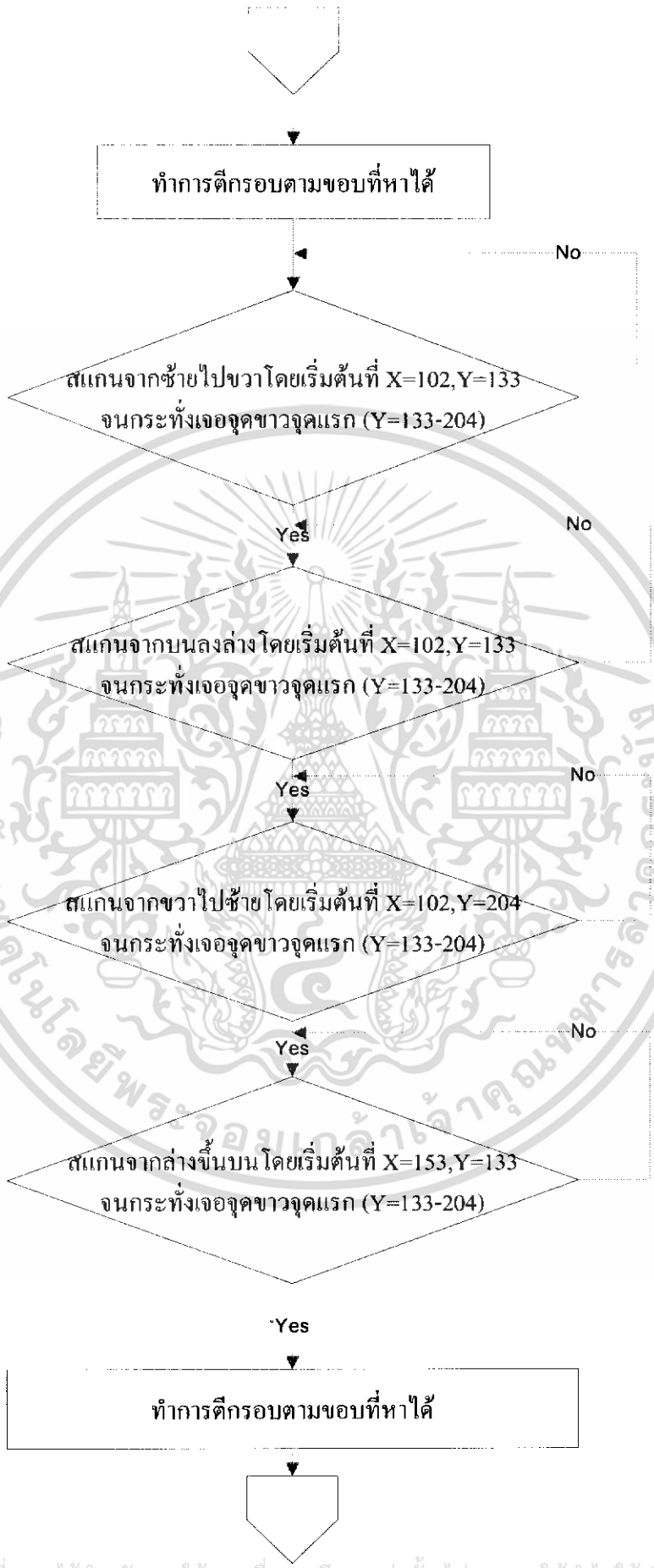
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการหาขนาดของตา



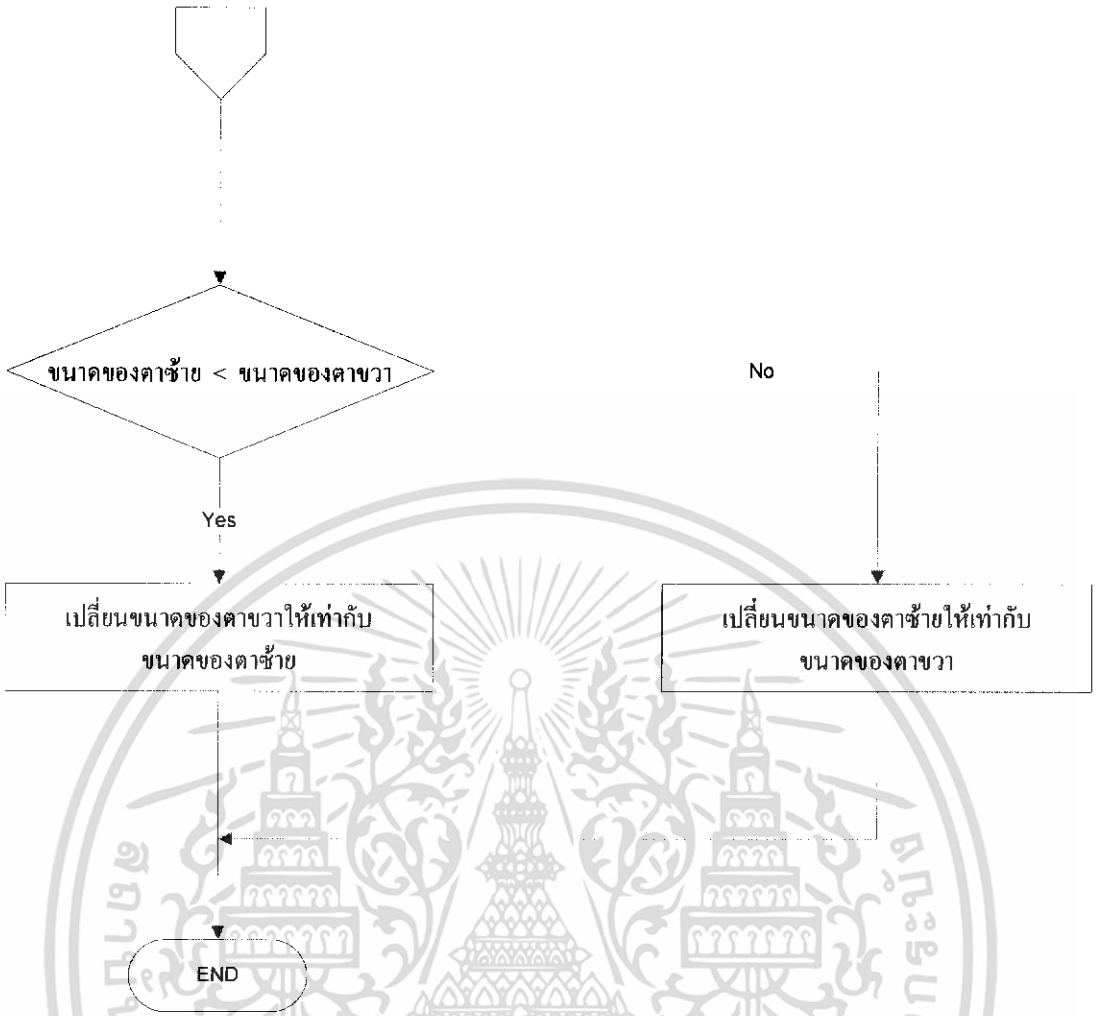
รูปที่ 3.14 Flow chart ของโปรแกรมการหาขนาดของตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 Flow chart ของโปรแกรมการหาขนาดของตา (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 Flow chart ของโปรแกรมการหาขนาดของตา (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการหาขนาดของจมูกและปาก

รูปที่ 3.17 Flow chart ของการหาขนาดจมูกและปาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.15 Flow chart ของการหาขนาดจมูกและปาก (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลองโปรแกรม

ในบทนี้จะกล่าวถึงภาพ และชุดข้อมูลที่ใช้ในการทดสอบของโปรแกรม และผลที่ได้จากการทำงานของโปรแกรม โดยผลที่ได้จะแสดงให้เห็นความถูกต้องแม่นยำและเวลาที่ใช้ไป เมื่อนำมาเปรียบเทียบกันระหว่างคลัสเตอร์นิวรอลเน็ทเวิร์ค (Cluster Neural Network) กับนิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation Neural Network) ซึ่งเป็นแบบที่ใช้กันโดยทั่วไป

4.1 ภาพและชุดข้อมูลที่ใช้ในการทดสอบโปรแกรม

4.1.1 ภาพที่ใช้ในการทดสอบโปรแกรม

ในโปรแกรมจัดกลุ่มฟีเจอร์นี้ ฟีเจอร์ของใบหน้ามนุษย์ก็เป็นส่วนหนึ่งที่ใช้ในทดสอบด้วย จึงนำภาพที่ถ่ายหน้าตรงไม่สวมหมวก ไม่ใส่แว่น และไม่ยิ้ม มาใช้ในการทำงาน แต่ไม่ว่าจะสามารถทำงานได้กับทุกๆ คือ ในกรณีที่สภาพแวดล้อมของภาพถ่ายเปลี่ยนไปจะทำให้เกิดความผิดพลาดในการค้นหาองค์ประกอบของใบหน้าขึ้น เช่น ในกรณีที่ เกิดแสงสะท้อนบนใบหน้า ผู้ถูกถ่ายมีการแต่งหน้า หรือการซูมเข้าหรือออกมากเกินไป เป็นต้น

รูปที่ 4.1 ภาพที่สามารถนำมาใช้กับโปรแกรมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลที่ได้จากการสกัดหาฟีเจอร์ของใบหน้า

เมื่อค้นหาฟีเจอร์ของภาพหลายๆ ภาพแล้วจะได้ ข้อมูลความสูงและความกว้างของตาดำ จมูก และปาก เป็นชุดของข้อมูลดังนี้

ตารางที่ 4.1 ผลลัพธ์จากการค้นหาฟีเจอร์บนใบหน้า

ภาพที่	Eye L	Eye W	M L	M W	N L	N W	Sex
1	20	51	20	49	18	63	Man
2	23	31	13	53	24	66	Man
3	21	34	16	44	18	66	Woman
4	24	48	23	56	12	66	Woman
5	24	49	19	51	12	55	Man
6	23	46	21	50	19	69	Woman
7	18	40	10	54	22	61	Man
8	13	40	7	52	18	65	Woman

4.2 เปรียบเทียบกันระหว่างคลัสเตอร์นิวรอลเน็ทเวิร์ค (Cluster Neural Network) กับนิวรอลเน็ทเวิร์คแบบแพร่กลับ (Back-propagation Neural Network)

ในการทดสอบโปรแกรม ได้ทำการทดสอบโดยใช้ชุดข้อมูลของ Breast-Cancer-Wisconsin และ Iris ซึ่งเป็นชุดข้อมูลที่ใช้กันอย่างแพร่หลาย โดยผลการทดสอบมีค่าดังต่อไปนี้

ตารางที่ 4.2 แสดงการเปรียบเทียบระหว่างคลัสเตอร์นิวรอลเน็ทเวิร์คกับนิวรอลเน็ทเวิร์คแบบแพร่กลับ ของชุดข้อมูล Breast-cancer-wisconsin

	คลัสเตอร์นิวรอลเน็ทเวิร์ค	นิวรอลเน็ทเวิร์คแบบแพร่กลับ
Hidden Neuron ที่ใช้ในเน็ทเวิร์ค	12 unit	12 unit
ค่า Learning Rate ที่ใช้	0.1	0.1
เวลาที่ใช้ในการสอน	378.296875	595.859375
จำนวนรอบการสอน	19720	32363
ค่า SSE	0.009998	0.009999
ค่าความแม่นยำ	93.491124260355	93.1952662721893
เวลาที่ใช้ในการทดสอบใช้งาน	0.03125	0.03125

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงการเปรียบเทียบระหว่างคลัสเตอร์นิวรอลเน็ทเวิร์คกับนิวรอลเน็ทเวิร์คแบบแพร่กลับของ Iris

	คลัสเตอร์นิวรอลเน็ทเวิร์ค	นิวรอลเน็ทเวิร์คแบบแพร่กลับ
Hidden Neuron ที่ใช้ในเน็ทเวิร์ค	8 unit	8 unit
ค่า Learning Rate ที่ใช้	0.01	0.01
เวลาที่ใช้ในการสอน	4.25	34.828125
จำนวนรอบการสอน	1248	10404
ค่า SSE	4.6989	4.69998
ค่าความแม่นยำ	97.3333	97.3333
เวลาที่ใช้ในการทดสอบใช้งาน	0.015625	0.015625

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 บทสรุป

ในส่วนของการจัดกลุ่มฟีเจอร์นั้นจะกระทำโดยใช้นิวรอลเน็ทเวิร์คในการเป็นพื้นฐาน โดยส่วนชุดข้อมูลที่ นิวรอลเน็ทเวิร์คสามารถเรียนรู้ได้คือ Iris, Breast cancer Wisconsin และชุดข้อมูลที่นิวรอลเน็ทเวิร์คไม่สามารถเรียนรู้ได้มีคือ ฟีเจอร์บนใบหน้าได้ ซึ่งอาจจะมีเหตุผลมาจากสาเหตุ 3 ประการดังนี้

1. จำนวนของรูปภาพที่นำมาใช้ในการทดสอบไม่เพียงพอ
2. จำนวนขององค์ประกอบบนใบหน้าที่ค้นหาได้ไม่เพียงพอ
3. ไม่สามารถจำแนกเพศจากฟีเจอร์บนใบหน้าของมนุษย์ได้ตั้งแต่ต้น

และผลที่ได้จากกระบวนการจัดกลุ่มของฟีเจอร์ ยังไม่อยู่ในเกณฑ์ที่พอใจ เนื่องจากจำนวนของข้อมูลมีน้อย ทำให้ในการทำงานของนิวรอลเน็ทเวิร์คใช้เวลาไม่มาก เป็นผลให้สภาพแวดล้อมของเครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ มีผลต่อความเร็วที่เพิ่มขึ้นมาอย่างมาก

ในส่วนของการประมวลผลภาพจะนำรูปถ่ายหน้าตรงมาเพื่อทำการแยก ตา ปาก จมูก ออกจากใบหน้า ซึ่งฟีเจอร์เล่านี้อาจไม่สมบูรณ์หรือหาขนาดไม่ได้เนื่องจากสาเหตุหลายประการ ได้แก่

- ขนาดของภาพไม่เหมาะสม เช่น ซูมภาพมากเกินไปหรือน้อยเกินไป เป็นต้น
- ตำแหน่งของฟีเจอร์ต่างๆไม่อยู่ในกรอบที่กำหนดไว้ ก็จะทำให้ไม่สามารถหาฟีเจอร์นั้นได้สมบูรณ์หรือได้อาจได้ฟีเจอร์ที่ผิดพลาด
- เส้นผมตกลงมาติดกับคิ้วอาจทำให้ขนาดของดวงตาที่ได้ผิดไปจากความเป็นจริง
- รูปที่ได้เอียงซ้าย,ขวา เงยหน้า หรือก้มหน้าจะทำให้ไม่สามารถหาฟีเจอร์ได้

ดังนั้น ในทางปฏิบัติสามารถนำข้อจำกัดดังกล่าวเป็นสิ่งสำคัญในการออกแบบระบบที่ใช้งานจริง โดยเฉพาะในส่วนการประมวลผลภาพจะต้องออกแบบให้ครอบคลุมถึงข้อจำกัดทุกข้อ เพื่อให้ระบบสามารถทำงานได้อย่างถูกต้องและแม่นยำ ซึ่งจะต้องทำการปรับปรุงโปรแกรมที่มีอยู่ให้สมบูรณ์ยิ่งขึ้น ซึ่งจะเป็นแนวทางที่ต้องพัฒนาในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 วิจารณ์สิ่งที่ได้จากโครงงาน

จากโครงงานนี้ ทำให้สามารถพัฒนานิวรอลเน็ทเวิร์คที่สามารถจัดกลุ่มของฟีเจอร์ โดยแต่ละกลุ่มจะมีเพียงฟีเจอร์ที่เกี่ยวข้องกันเท่านั้น ซึ่งเป็นผลให้นิวรอลเน็ทเวิร์คมีความเร็วในการตัดสินใจเพิ่มขึ้น แต่อย่างไรก็ตาม เนื่องจากได้ทำการลดการเชื่อมต่อของ Input unit และ Hidden unit เป็นผลให้จำเป็นต้องให้เวลาในการเรียนรู้ของนิวรอลเน็ทเวิร์คเพิ่มขึ้นด้วยเช่นกัน

5.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข

1. ในการทดสอบนิวรอลเน็ทเวิร์ค สภาพแวดล้อมของเครื่องคอมพิวเตอร์ยังเป็นปัจจัยที่มีผลต่อความเร็วของนิวรอลเน็ทเวิร์คอยู่มาก
2. การหาขนาดของแต่ละฟีเจอร์ยังมีความผิดพลาดอยู่เนื่องจากรูปถ่ายที่จะนำมาใช้ในการทำการทดลอง ไม่มีรูปถ่ายที่เป็นมาตรฐานและรูปแบบเดียวกัน ทำให้ออกแบบอัลกิริทึมเพื่อให้รองรับกับทุกรูปเป็นไปได้ยาก แนวทางแก้ไขคือ ควรจะมีการเก็บรูปถ่ายที่เป็นมาตรฐานไว้จำนวนหนึ่งเพื่อการพัฒนาต่อไปในอนาคต
3. แต่ละรูปถ่ายที่นำมาทำการจำแนกจะมีสภาวะแวดล้อมที่ต่างกัน เช่น มุมของแสง ความเข้มของแสง และการแต่งหน้า เป็นต้น สิ่งเหล่านี้จะทำให้เกิดความผิดพลาดในการค้นหาฟีเจอร์บนใบหน้าขึ้นได้ แนวทางแก้ไขคือ ในการถ่ายภาพควรที่จะเลือกสถานที่ถ่ายภาพที่ไม่มีสภาพแวดล้อมที่ทำให้เกิดปัญหา

5.4 แนวทางการพัฒนาต่อ

1. ในส่วนของการประมวลผลภาพคืออัลกอริทึมที่ใช้ในการหาแต่ละฟีเจอร์จะต้องมีความถูกต้องมากยิ่งขึ้น และรูปถ่ายที่จะนำมาทดลองจะต้องเป็นรูปแบบเดียวกัน
2. พัฒนาในส่วนของการประมวลผลภาพ ให้สามารถรับภาพที่ในสภาพแวดล้อมที่หลากหลายยิ่งขึ้น
3. พัฒนาในส่วนของการประมวลผลภาพให้สามารถค้นหาฟีเจอร์ได้มากยิ่งขึ้นเพื่อที่จะได้สามารถนำมาจำแนกใบหน้าได้
4. พัฒนาในส่วนของตัวแอพพลิเคชันให้เป็นแอพพลิเคชันแบบออนไลน์ ซึ่งมีแนวโน้มที่เป็นไปได้ในอนาคต เพราะแอพพลิเคชันออนไลน์กำลังได้รับความนิยมอย่างมากในปัจจุบัน
5. พัฒนาในส่วนของนิวรอลเน็ทเวิร์คให้สามารถลด Hidden unit ได้ ซึ่งคาดว่าจะทำให้นิวรอลเน็ทเวิร์คมีใช้เวลาน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] รศ. มนัส สังวรศิลป์ ,วรรัตน์ ภัทรอมรกุล, 2000, "คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์", อินโฟเพรส

[2] Rafael C. Gonzalez and Richard E. Woods., 2001, "Digital Image Processing", New Jersey, Prentice-Hall , Inc.

[3] Gregory A.Baxes , 1994, " Digital Image Processing Principles and Processing ", John Wiley & Sons, Inc.

[4] F. Bergholm, "On the Information in Edges and Optical Flow," Ph.D. thesis, University of Stockholm, Sweden, May 1989

[5] Erik Hjelmas, Boon Kee Low, "Face Detection: A Survey," Computer Vision and Image Understanding, Vol.83, 236-274 , Apr. 2001.

[6] L. C. De Silva, K. Aizawa, and M. Hatori, Detection and tracking of facial features by using a facial feature

[7] Model and deformable circular template, IEICE Trans. Inform. Systems E78–D(9), 1995, 1195–1207.

[8] D. E. Goldberg, Genetic Algorithms in Search Optimization, and Machine Learning, Addison-Wesley, 1989.

[9] Li Min Fu, Neural Networks in Computer Intelligence,McGraw Hill, Inc., 1994

[10] R .E. Karjalainen, " Using Genetic algorithms To Find Technical Trading Rules In Financial Markets", Ph D. Dissertation, University of Pennsylvania, 1994.

[11] Hans-Georg Wittkemper; Manfred Steiner, " Using Neural network to Forecast The Systematic Risk of Stocks ",

[12] European Journal of Operational Research, vol.90, iss.3, May 10,1996, pp. 577-588 .

[13] Philip, K. 1994. Combining Genetic Algorithm and Neural Networks: The Encoding Problem.

[14] Michael Negnevitsky, "Artifitial Intelligence A Guide to Intelligent System", Second Edition, Addison-Wesley.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้