สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจดจำใบหน้าโดยใช้หลักการของ 2DLDA

FACE RECOGNITION USING 2DLDA ALGORITHM

T 061197

สิทธินนท์ คงสนทนา

SITTINON KONGSONTANA

วพ.
ส722ก
2548

เลขหมู่.....
เลขทะเบียน.....61197
วัน,เดือน,ปี.....17 ก.ค. 2549

b.11513590
i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2548

ISBN 974-15-1991-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FACE RECOGNITION USING 2DLDA ALGORITHM

SITTINON KONGSONTANA

THESIS SUBMITED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN TELECOMMUNICATIONS ENGINEERING
SCHOOL OF GRADUATE STUDENTS
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2005

ISBN 974-15-1991-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2005
SCHOOL OF GRADUATE STUDENTS
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การจดจำใบหน้าโดยใช้หลักการของ 2DLDA
นักศึกษา	นายสิทธินนท์ คงสนทนา
รหัสนักศึกษา	45060502
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมโทรคมนาคม
พ. ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. ยุทธพงษ์ รังสรรค์เสรี

บทคัดย่อ

วิทยานิพนธ์นี้ได้เสนอวิธีการ Two – Dimensional Linear Discriminant Analysis (2DLDA) สำหรับการหาฟีเจอร์ของภาพ เพื่อนำมาใช้ในงานด้านการจดจำใบหน้า วิธีการนี้พัฒนามาจากพื้นฐานของวิธีการแบ่งแยกเชิงเส้นของฟิชเชอร์และการวิเคราะห์องค์ประกอบหลักของเมตริกภาพ โดยที่วิธี 2DLDA สามารถใช้เมตริกภาพมาคำนวณหาเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่มได้โดยตรง การคำนวณด้วยวิธี 2DLDA ยังสามารถแก้ปัญหาการเป็นเมตริกเอกฐานของเมตริกการกระจายภายในกลุ่มได้ จากผลการทดลองพบว่าวิธีใหม่นี้มีประสิทธิภาพในการคำนวณดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Face Recognition Using 2DLDA Algorithm
Student	Mr. Sittinon Kongsontana
Student ID.	45060502
Degree	Master of Engineering
Programme	Telecommunications Engineering
Year	2005
Thesis Advisor	Assc. Prof. Dr.Yuttapong Rangsanseri

ABSTRACT

This thesis proposes Two–Dimensional Linear Discriminant Analysis (2DLDA) for feature extraction used for face recognition application. This method is developed from Fisher Linear Discriminated (FLD) and Image Principle Component Analysis (IMPCA). In this method 2DLDA directly uses the image matrix to calculate the between-class scatter matrix and within-class scatter matrix. Moreover, 2DLDA will be handling the problem that the within-class scatter matrix maybe singular. The experimental results indicated that the 2DLDA method is more computationally efficient than the conventional methods.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบคุณเพื่อน ๆ และ น้อง ๆ ทุก ๆ คนที่เป็นกำลังใจ ให้คำแนะนำ และให้ความช่วยเหลือจนทำให้การจัดพิมพ์วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดา มารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

สิทธินนท์ คงสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของวิทยานิพนธ์.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของวิทยานิพนธ์.....1

1.3 สมมุติฐานของวิทยานิพนธ์.....1

1.4 ขอบเขตของวิทยานิพนธ์.....2

1.5 รายละเอียดของวิทยานิพนธ์.....2

บทที่ 2 การประมวลผลภาพและการรู้จำภาพใบหน้า.....4

2.1 ความซับซ้อนและความยุ่งยากในการรู้จำใบหน้าด้วยคอมพิวเตอร์.....4

2.2 การนำระบบรู้จำใบหน้าไปใช้จริงในปัจจุบัน.....4

2.2.1 เฝ้าดูบุคคลแปลกหน้า.....4

2.2.2 เฝ้าดูบุคคลที่รู้จัก.....5

2.2.3 ล็อคหน้าจอ (Screen Lock).....5

2.2.4 การค้นหาในฐานข้อมูล.....5

2.3 สิ่งที่เปลี่ยนแปลงได้บนภาพใบหน้าที่ใช้ในการรู้จำ.....6

2.3.1 สภาพแวดล้อม.....6

2.3.2 การแสดงออกทางอารมณ์บนใบหน้า.....6

2.3.3 อื่น ๆ.....6

2.4 วิธีการในการรู้จำภาพใบหน้าที่มีการวิจัยกันมาจนถึงปัจจุบัน.....6

2.4.1 ลักษณะทางเรขาคณิต (Geometrical Features).....6

2.4.2 การเปรียบเทียบกราฟ (Graph Matching).....7

2.4.3 วิธีการทางสถิติ (Statistical).....8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4.3.1 การวิเคราะห์องค์ประกอบหลัก.....8
2.4.3.2 การวิเคราะห์การแบ่งแยกเชิงเส้น.....10
2.4.4 โครงข่ายประสาทเทียม (Neural network)..... 10
2.4.5 วิธีอื่น ๆ (Other Methods)..... 11

บทที่ 3 ทฤษฎีพื้นฐานที่เกี่ยวข้อง.....12
3.1 การวิเคราะห์องค์ประกอบหลัก (PCA).....12
3.1.1 แนวความคิดของการวิเคราะห์องค์ประกอบหลัก.....12
3.1.2 หลักการคำนวณของการวิเคราะห์องค์ประกอบหลัก.....17
3.1.3 การหาฟีเจอร์ของภาพ 20
3.1.4 การสร้างภาพใหม่.....20
3.2 การวิเคราะห์การแบ่งแยกเชิงเส้นของฟิชเชอร์ (FLD).....20
3.2.1 แนวความคิดของการวิเคราะห์การแบ่งแยกเชิงเส้นของฟิชเชอร์.....20
3.2.2 หลักการคำนวณของการวิเคราะห์การแบ่งแยกเชิงเส้นของฟิชเชอร์.....28
3.2.3 การหาฟีเจอร์ของภาพ.....30
3.2.3 การสร้างภาพใหม่.....30
3.3 Two – Dimensional Principal Component Analysis (2DPCA).....31
3.3.1 แนวความคิดของวิธี 2DPCA.....31
3.3.2 การหาฟีเจอร์ของภาพ.....33
3.3.3 การสร้างภาพใหม่.....34
3.3.4 หลักการคำนวณของวิธี 2DPCA.....34
3.4 Two – Dimensional Linear Discriminant Analysis (2DLDA).....35
3.4.1 แนวความคิดของวิธี 2DLDA.....35
3.4.2 การหาฟีเจอร์ของภาพ.....35
3.4.3 การสร้างภาพใหม่.....35
3.4.4 หลักการคำนวณของวิธี 2DLDA.....38
3.5 วิธีการค้นหาภาพใบหน้า.....40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 การทดลองและผลการทดลอง.....41

4.1 การทดลองบนฐานข้อมูลใบหน้า ORL.....41

4.1.1 การทดสอบการสร้างภาพใหม่ด้วยวิธี 2DLDA บนฐานข้อมูล ORL.....42

4.1.2 การทดลองเปรียบเทียบความถูกต้องในการรู้จำระหว่างวิธี 2DLDA กับวิธี Fisherface.....43

4.1.3 การทดลองเปรียบเทียบกระบวนการของ 2DLDA กับวิธีอื่น ๆ.....45

4.2 การทดลองบนฐานข้อมูลใบหน้า AR.....48

4.2.1 การทดลองกับเงื่อนไขการถ่ายภาพต่างเวลา.....50

4.2.2 การทดลองกับเงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า.....52

4.2.3 การทดลองภายใต้เงื่อนไขของความสว่างบนใบหน้า.....54

4.3 การทดลองบนฐานข้อมูลใบหน้า Yale.....55

4.3.1 การทดลองภายใต้เงื่อนไขการเปลี่ยนแปลงของการแสดงออกบนใบหน้าและความสว่างบนใบหน้า.....56

4.3.2 การทดลองเปรียบเทียบกระบวนการของ 2DLDA ภายใต้เงื่อนไขของระยะทางที่ใช้ในการจำแนกบุคคล.....59

บทที่ 5 สรุปผลการวิจัยและแนวทางในการพัฒนา.....62

5.1 สรุปผลการวิจัย.....62

5.2 แนวทางในการพัฒนา.....63

เอกสารอ้างอิง.....64

ภาคผนวก.....67

ภาคผนวก ก ผลงานที่ได้รับการตีพิมพ์.....68

ภาคผนวก ข โปรแกรมที่ใช้ในการทดลอง.....69

ประวัติผู้เขียน.....126

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

4.1 เปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดระหว่างวิธี Fisherface กับ วิธี 2DLDA ที่สัมพันธ์กับจำนวนตัวอย่างที่ใช้ฝึกสอน.....44

4.2 เปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดของทั้ง 4 วิธี.....48

4.3 เปรียบเทียบเวลา (วินาที) ที่ใช้คำนวณหาฟีเจอร์ของภาพ.....48

4.4 เปรียบเทียบวิธี 2DLDA กับวิธี Fisherface ภายใต้เงื่อนไขการถ่ายภาพต่างเวลา เงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า เงื่อนไขของแสงบนใบหน้า.....52

4.5 เปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดของวิธีทั้งสี่วิธีบนฐานข้อมูล Yale.....58

4.6 เปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดของระยะทางที่ใช้ในการจำแนกบุคคล.....61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

3.1 ตัวอย่างภาพที่เปลี่ยนจากเมตริกภาพเป็นเวกเตอร์ภาพ....18

3.2 ตัวอย่างค่าไอเกนที่เรียงจากมากไปน้อยของวิธี Eigenface จากฐานข้อมูล ORL....19

3.3 ตัวอย่างภาพใบหน้าบุคคลหนึ่งที่สร้างใหม่ด้วยวิธี Eigenface จากฐานข้อมูล ORL....20

3.4 ความแตกต่างระหว่างวิธี PCA และวิธี FLD....21

3.5 ผลของการเลือกค่าของ w ที่ใช้ในการฉายภาพ....22

3.6 การเลือกแกนการฉายจากระยะทางระหว่างค่าเฉลี่ยของตัวอย่างที่ถูกฉาย 2 กลุ่ม....23

3.7 ตัวอย่างการฉายภาพที่มีการแบ่งแยกได้ดีของตัวอย่าง 2 กลุ่ม....24

3.8 ตัวอย่างค่าไอเกนที่เรียงจากมากไปน้อยของวิธี FLD จากฐานข้อมูล ORL....30

3.9 ตัวอย่างภาพใบหน้าบุคคลหนึ่งที่สร้างใหม่ด้วยวิธี Fisherface จากฐานข้อมูล ORL....31

3.10 ตัวอย่างค่าไอเกนที่เรียงจากมากไปน้อยของวิธี 2DPCA จากฐานข้อมูล ORL....33

3.11 ตัวอย่างภาพใบหน้าที่สร้างใหม่ด้วยวิธี 2DPCA จากฐานข้อมูล ORL....34

4.1 ตัวอย่างภาพใบหน้า 1 บุคคลของฐานข้อมูล ORL....41

4.2 ตัวอย่างภาพใบหน้าที่สร้างขึ้นใหม่ของบุคคลคนหนึ่ง....42

4.3 ค่าไอเกนที่เรียงจากมากไปน้อยของวิธี 2DLDA....42

4.4 ร้อยละความถูกต้องในการรู้จำของวิธี Fisherface ภายใต้เงื่อนไขของจำนวนตัวอย่างภาพใบหน้าที่ใช้ฝึกสอน....43

4.5 ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA ภายใต้เงื่อนไขของจำนวนตัวอย่างภาพใบหน้าที่ใช้ฝึกสอน....44

4.6 เปรียบเทียบความถูกต้องในการรู้จำระหว่างวิธี Fisherface กับวิธี 2DLDA+Fisherface....45

4.7 เปรียบเทียบร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA กับวิธี Eigenface วิธี Fisherface และวิธี 2DPCA....47

4.8 ตัวอย่างภาพใบหน้าของบุคคลคนหนึ่งที่ถูกแบ่งเป็นสองกลุ่ม....49

4.9 ตัวอย่างภาพใบหน้าของ 1 บุคคลที่ถูกเลือกมาใช้ในการทดลอง....50

4.10 ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA และ วิธี Fisherface ภายใต้ตัวแปรของการถ่ายภาพต่างเวลา....51

4.11 ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA และ วิธี Fisherface ภายใต้เงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.12 ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA และ วิธี Fisherface ภายใต้เงื่อนไขการความสว่างบนใบหน้า.....55

4.13 ตัวอย่างภาพใบหน้า 1 บุคคลของฐานข้อมูล Yale.....56

4.14 เปรียบเทียบร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA กับวิธี Eigenface วิธี Fisherface และวิธี 2DPCA โดยใช้วิธีลีฟวิงวันเอาท์.....58

4.15 เปรียบเทียบร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA กับวิธี Eigenface วิธี Fisherface และวิธี 2DPCA ภายใต้เงื่อนไขของระยะทาง.....60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของวิทยานิพนธ์

เนื่องจากงานด้านการจดจำใบหน้าเป็นเรื่องที่กำลังได้รับความสนใจอย่างมากในปัจจุบัน วิธีการทางด้านสถิติถูกนำมาประยุกต์ใช้กับงานด้านการจดจำใบหน้าอย่างกว้างขวาง แต่วิธีการที่ใช้อยู่ในปัจจุบันยังให้ความถูกต้องในการจดจำใบหน้าได้ไม่ดีเท่าที่ควรและใช้เวลาในการคำนวณมาก จากการศึกษาพบว่าวิธีการที่ใช้อยู่ในปัจจุบันสามารถพัฒนาให้มีการคำนวณที่ง่ายขึ้นและประสิทธิภาพเพิ่มขึ้นได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของวิทยานิพนธ์

1.2.1 เพื่อศึกษาทฤษฎีการรู้จำใบหน้า โดยเฉพาะหลักการหาฟีเจอร์ของภาพด้วยวิธีการแปลงเชิงเส้น

1.2.2 ศึกษาแนวทางการปรับปรุงวิธีการแบ่งแยกเชิงเส้นของฟิชเชอร์ (FLD) เพื่อให้ได้วิธีการที่สามารถใช้ภาพสองมิติมาคำนวณได้โดยตรง ซึ่งจะทำให้มีการคำนวณที่ง่ายขึ้นและมีประสิทธิภาพเพิ่มขึ้น

1.2.3 วิเคราะห์และเปรียบเทียบความถูกต้องในการรู้จำของหลักการแปลงเชิงเส้นระหว่างวิธีการวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis: PCA) การแบ่งแยกเชิงเส้นของฟิชเชอร์ (Fisher Linear Discriminant: FLD) Two – Dimensional Principal Component Analysis (2DPCA) และวิธีการใหม่ที่วิทยานิพนธ์นี้ได้นำเสนอคือ Two – Dimensional Linear Discriminant Analysis (2DLDA)

1.3 สมมุติฐานของการศึกษาและทฤษฎีแนวความคิดที่ใช้ในการวิจัย

การวิเคราะห์องค์ประกอบหลัก (PCA) และการวิเคราะห์การแบ่งแยกเชิงเส้นของฟิชเชอร์ (FLD) เป็นวิธีทางการทางสถิติซึ่งได้รับความนิยมสำหรับนำมาใช้ในการจดจำใบหน้า แต่ว่าวิธี PCA และวิธี FLD จะใช้การคำนวณเฉพาะเวกเตอร์ 1 มิติ ดังนั้นเมื่อต้องการใช้วิธีนี้กับภาพจึงจำเป็นจะต้องเปลี่ยนเมตริกภาพให้เป็นเวกเตอร์ภาพ หลังจากนั้นจึงใช้พื้นฐานของเวกเตอร์ในการคำนวณหาเมตริกความแปรปรวนร่วม (covariance matrix) เพื่อคำนวณหาเวกเตอร์ที่เหมาะสมที่สุดที่ใช้ในการฉายภาพ (projection) หลักการดังกล่าวนี้ถูกเรียกว่า Eigenface [1] และ Fisherface [2] เมื่อเรานำหลักการดังกล่าวมาใช้ในงานด้านการจดจำใบหน้าเป็นผลให้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการจดจำที่ดี แต่ว่าภาพใบหน้าที่ใช้ในกระบวนการจดจำจะมีมิติมาก เช่น ภาพใบหน้าที่มีขนาดความละเอียด 112 x 92 จุดภาพ เมื่อเปลี่ยนเป็นเวกเตอร์จะมีขนาด 10,304 มิติ จากเหตุผลดังกล่าวจึงทำให้การคำนวณด้วยวิธี Eigenface และ Fisherface ใช้เวลามาก

เมื่อไม่นานมานี้ Yang [3] ได้พัฒนาหลักการใหม่ที่นำแนวคิดมาจากวิธี PCA เพื่อแก้ไขจุดบกพร่องนี้ จึงได้เสนอวิธีการคำนวณหาเมตริกความแปรปรวนร่วมโดยใช้ภาพที่เป็น 2 มิติมาคำนวณโดยตรง วิธีการนี้มีชื่อว่า การวิเคราะห์องค์ประกอบหลักของภาพ (Image Principal Component Analysis: IMPCA หรือ 2DPCA) จากผลการทดลองของ Yang แสดงให้เห็นถึงประโยชน์ของเวลาที่ใช้ในการคำนวณและความถูกต้องในการรู้จำที่ดีกว่าวิธี PCA แต่ว่าวิธีการนี้ได้แนวความคิดมาจากวิธี PCA จึงไม่ได้พิจารณาถึงข้อมูลของกลุ่มเข้าไปคำนวณด้วย

วิทยานิพนธ์นี้จึงได้เสนอวิธีการใหม่ที่มีชื่อว่า Two – Dimensional Linear Discriminant Analysis (2DLDA) มาใช้ในการรู้จำใบหน้า ซึ่งวิธีการนี้พัฒนามาจากพื้นฐานวิธีการแบ่งแยกเชิงเส้นของ ฟิชเชอร์ที่นำเอาข้อมูลของกลุ่มมาคำนวณและการวิเคราะห์องค์ประกอบหลักของเมตริกภาพที่สามารถทำให้ขนาดของเมตริกการกระจายมีขนาดเล็ก โดยที่วิธี 2DLDA สามารถใช้เมตริกภาพมาคำนวณหาเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่มได้ การคำนวณด้วยวิธีนี้ยังสามารถแก้ปัญหาการเป็นเมตริกเอกฐาน (singular matrix) ของเมตริกการกระจายภายในกลุ่มได้เพราะว่า มิติของเมตริกการกระจายภายในกลุ่มมีค่ามากเมื่อเปรียบเทียบกับจำนวนของตัวอย่างภายในกลุ่มที่ใช้ในการฝึกสอน และวิธี 2DLDA ยังสามารถลดเวลาการคำนวณได้มากกว่าวิธี Fisherface

1.4 ขอบเขตของวิทยานิพนธ์

วิทยานิพนธ์นี้ทำการศึกษาหลักการใหม่สำหรับการรู้จำใบหน้าที่มีชื่อว่า 2DLDA ซึ่งนำแนวคิดมาจากพื้นฐานวิธีการแบ่งแยกเชิงเส้นของฟิชเชอร์และการวิเคราะห์องค์ประกอบหลักของเมตริกภาพ แล้วทำการเปรียบเทียบความถูกต้องในการรู้จำกับวิธี Eigenface,วิธี Fisherface และวิธี 2DPCA กับฐานข้อมูล 3 แหล่งได้แก่ ฐานข้อมูลภาพใบหน้า ORL ฐานข้อมูลภาพใบหน้า AR และฐานข้อมูลภาพใบหน้า Yale โดยที่ฐานข้อมูลภาพใบหน้า ORL และฐานข้อมูลภาพใบหน้า Yale นั้นภาพใบหน้าทั้งหมดเป็นภาพระดับสีเทา ส่วนฐานข้อมูลภาพใบหน้า AR เป็นภาพใบหน้าทั้งหมดเป็นภาพสีจึงทำการเปลี่ยนภาพทั้งหมดเป็นภาพระดับสีเทา และถ้าฐานข้อมูลใดมีภาพใบหน้าขนาดใหญ่ก็จะทำการลดขนาดภาพโดยการตัดภาพด้วยมือจากโปรแกรม Photoshop และในการคำนวณหาความถูกต้องในการรู้จำได้ใช้โปรแกรม Matlab 6.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 รายละเอียดของวิทยานิพนธ์

เนื้อหาในวิทยานิพนธ์ได้แยกออกเป็น 5 บท เพื่อแสดงถึง วิธีการที่เคยใช้ในการรู้จำใบหน้า ทฤษฎีที่นำมาใช้ในวิทยานิพนธ์ วิธีการรู้จำใบหน้าที่นำเสนอ ผลการทดลอง สรุป และแนวทางในการพัฒนาต่อไป ดังมีรายละเอียดในบทต่าง ๆ ดังต่อไปนี้

บทที่ 1 บทนำ

กล่าวถึง ความเป็นมาและความสำคัญของวิทยานิพนธ์ ความมุ่งหมายและวัตถุประสงค์ของวิทยานิพนธ์ สมมติฐานของการศึกษาและทฤษฎีแนวความคิดที่ใช้ในการวิจัย ขอบเขตของวิทยานิพนธ์ รายละเอียดของวิทยานิพนธ์

บทที่ 2 การประมวลผลภาพและการรู้จำใบหน้า

กล่าวถึงความยุ่งยากในการรู้จำด้วยคอมพิวเตอร์ ประโยชน์ในการนำไปประยุกต์ใช้งาน สิ่งที่เปลี่ยนแปลงได้บนภาพใบหน้าที่มีผลกระทบในการรู้จำใบหน้า และวิธีการในการรู้จำภาพใบหน้าที่มีการวิจัยกันมาจนถึงปัจจุบัน

บทที่ 3 ทฤษฎีพื้นฐานที่เกี่ยวข้อง

กล่าวถึงทฤษฎีต่าง ๆ ที่วิทยานิพนธ์นี้ได้นำมาใช้ประกอบคือ การวิเคราะห์องค์ประกอบหลัก (PCA) การแบ่งแยกเชิงเส้นของฟิชเชอร์ (FLD) Two – Dimensional Principal Component Analysis (2DPCA) Two– Dimensional Linear Discriminant Analysis (2DLDA) วิธีการค้นหาภาพใบหน้า

บทที่ 4 การทดลองและผลการทดลอง

กล่าวถึงการทดลองและผลการทดลองเพื่อทดสอบแนวคิดที่นำเสนอมา

บทที่ 5 สรุปผลการวิจัยและแนวทางในการพัฒนา

กล่าวถึงผลสรุปจากการวิจัยและข้อเสนอแนะสำหรับแนวทางการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การประมวลผลภาพและการรู้จำใบหน้า

2.1 ความซับซ้อนและความยุ่งยากในการรู้จำใบหน้าด้วยคอมพิวเตอร์

ในการรู้จำใบหน้านั้นมนุษย์ดูเหมือนจะทำได้อย่างรวดเร็วและง่ายดาย เหมือนกับไม่ต้องใช้ความพยายามมากนัก แต่ก็ไม่ได้หมายความว่าการใช้คอมพิวเตอร์รู้จำได้นั้นจะเป็นเรื่องง่าย ๆ ไปด้วย

ความยากอันดับแรก คือ ในการประยุกต์ใช้งานจริง ๆ ระบบจำเป็นต้องมีความถูกต้องสูงมาก เช่น ระบบรักษาความปลอดภัย และระบบ ATM Cash Machine ที่ใช้การรู้จำใบหน้าในการระบุบุคคลนั้นจำเป็นจะต้องมีความเร็วสูงและต้องไม่ผิดพลาดเลย การประยุกต์ทางด้านการตรวจจับคนที่เจ้าหน้าที่ทางกฎหมายต้องการจับตัวและผู้ลี้ภัยนั้น ระบบต้องมีการติดตั้งที่สถานที่สาธารณะต่าง ๆ ตัวอย่างเช่น สนามบิน ระบบจะต้องสามารถตรวจจับบุคคลผิดกฎหมายที่ต้องการออกมาให้ได้ในขณะที่ทุกคนต่างกำลังเดินหรือเคลื่อนไหวกันอยู่ ซึ่งการทำงานลักษณะนี้เป็นเรื่องยากมาก ดังนั้นระบบส่วนใหญ่ที่ทำกัน หรือบทความที่วิจัยกันอยู่ต่างมีขอบเขตความสามารถและขีดจำกัดในการประยุกต์ด้วยกันทั้งหมด

ความยากอันดับที่สอง คือ ในการรู้จำใบหน้าที่มีประสิทธิภาพจริง ๆ นั้นไม่สามารถใช้การประมวลผลภาพแบบเก่า ๆ เช่น การเทียบภาพ (Image Matching) ได้ ซึ่งส่วนที่ท้าทายนั้นแบ่งเป็นสองส่วน คือ ส่วนแรกเป็นส่วนการตรวจจับใบหน้า ที่จะต้องทำการตัดสินใจว่าในภาพนั้นมีภาพใบหน้าอยู่หรือไม่ และอยู่ตำแหน่งใดของภาพ และส่วนที่สองการรู้จำใบหน้าที่จะสามารถระบุได้ว่าเป็นบุคคลใดจากฐานข้อมูลที่มีอยู่ ซึ่งใบหน้านั้นยากที่จะตรวจจับและรู้จำทั้งนี้เพราะข้อเท็จจริงที่ว่าภาพใบหน้านั้นไม่มีรูปแบบตายตัว ซึ่งใบหน้าอาจมีการเปลี่ยนแปลงลักษณะต่าง ๆ ได้ทั้ง การส่องสว่าง การแสดงออกทางอารมณ์ การวางตัวของศีรษะ ทรงผม คิ้ว หนวด ใส่แว่น สวมหมวก ทำให้สิ่งเหล่านี้เป็นอุปสรรคทั้งสิ้น

2.2 การนำระบบรู้จำใบหน้าไปใช้จริงในปัจจุบัน

2.2.1 เฝ้าดูบุคคลแปลกหน้า

เป็นการทำงานอัตโนมัติ ในการเฝ้าดูบุคคลที่ไม่มีอยู่ในฐานข้อมูล มันสามารถส่งเสียงเตือนเมื่อเห็นคนแปลกหน้า และสามารถเก็บบันทึกภาพใบหน้านั้นได้ด้วย ซึ่งทำให้สามารถกลับมาดูภาพใบหน้าของคนบุกลุกได้เมื่อต้องการหรือภายหลังสิ้นสุดวัน อีกทั้งยังมีการบันทึกเวลาที่บุคคลนั้นเข้ามาให้เห็นด้วย ซึ่งจะสามารถดูได้ง่ายกว่าการดูวีดีโอทั้งหมดเพราะการทำงานด้วยวิธีนี้จะเก็บเฉพาะภาพใบหน้าของผู้บุกลุกเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 **เฝ้าดูบุคคลที่รู้จัก**

เป็นการติดตามบุคคลในครอบครัวหรือบุคคลในฐานข้อมูล ที่เข้ามาและออกไป ตัวอย่างเช่น ตำรวจต้องการเฝ้าดูบุคคลต่าง ๆ ที่เข้าออกอาคาร หรือการลงเวลาการเข้าและออกของพนักงานบริษัท ก็สามารถใช้ระบบดังกล่าวได้ การทำงานในลักษณะเช่นนี้สามารถบันทึกข้อมูลวันเวลาและจำนวนครั้งที่บุคคลนั้นเข้ามาและออกไปให้เห็นด้วย

2.2.3 **ล็อคหน้าจอ** (Screen Lock)

เป็นการทำงานที่เหมือนกับการล็อคหน้าจอแบบปกติที่ใช้การป้อนรหัสเพื่อผ่านเพื่อเข้าสู่ระบบคอมพิวเตอร์หรือระบบเครือข่ายนั่นเอง เว้นแต่ว่ามันจะมีการปลดล็อคอัตโนมัติเมื่อมีผู้ใช้กลับมาและมองไปที่กล้องจับภาพ หมายความว่าจะไม่ยอมให้บุคคลอื่น ๆ เข้ามาใช้เครื่องหรือเข้าสู่ระบบได้ ระบบนี้สามารถออกแบบให้ปลดล็อคเพียงคนเดียวที่ยินยอมนั้นกลับมาอีกครั้ง หรือสามารถออกแบบให้ปลดล็อคเมื่อกลุ่มใบหน้าที่อยู่ในบานข้อมูลคนใดคนหนึ่งกลับมาก็ได้

2.2.4 **การค้นหาในฐานข้อมูล**

การทำงานนี้สามารถค้นหาภาพใบหน้าในฐานข้อมูลภาพขนาดใหญ่ หรือเป็นฐานข้อมูลภาพเคลื่อนไหวที่เป็นวีดีโอขนาดใหญ่ก็ได้ โดยจะมีการตรวจหาอย่างอัตโนมัติในทุก ๆ ภาพหรือในทุก ๆ เฟรมของวีดีโอ ที่มีภาพบุคคลที่เราต้องการหาซึ่งอาจจะต้องการหาหลายคนในคราวเดียวตัวอย่างเช่น การค้นหาภาพจากแถบบันทึกวีดีโอ ซึ่งเป็นเรื่องที่น่าเบื่อและเป็นกระบวนการที่ต้องใช้เวลามากคือการค้นหาภาพบุคคลที่ต้องการ ซึ่งถ้าหากวีดีโอมีขนาดใหญ่หรือต้องทำบ่อย ๆ กลับไปกลับมาจะเป็นเรื่องที่เสียเวลามาก แต่ถ้าหากใช้โปรแกรมการรู้จำใบหน้าทำการค้นหาแทนซึ่งสามารถทำงานได้อย่างไม่รู้จักเหน็ดเหนื่อยในการค้นหาต่อเนื่องกันเป็นชั่วโมง ๆ โดยไม่มีการหยุดพัก หรือถ้าเป็นการค้นหาในฐานข้อมูลขนาดใหญ่เช่น ในฐานข้อมูลของกรมตำรวจเป็นต้น มันจะเป็นการหาบุคคลที่มีใบหน้าคล้ายกับภาพที่เราต้องการค้นหา โดยจะแสดงรายชื่อ ประวัติและภาพออกมาตามลำดับความคล้าย อาจจะเป็น 5 หรือ 10 ภาพเป็นต้น

สามารถสรุปเป็นสองแนวทางใหญ่ ๆ คือ

1) กรณีต้องการค้นหาบุคคลในฐานข้อมูลใบหน้าขนาดใหญ่ (เช่น ฐานข้อมูลของกรมตำรวจ) ระบบเช่นนี้จะต้องแสดงภาพและประวัติของบุคคลที่มีความคล้ายออกมา โดยที่ระบบไม่จำเป็นต้องมีความเร็วแบบเวลาจริงในการทำงาน (Non Real Time)

2) กรณีต้องการชี้ระบุตัวบุคคลแบบเวลาจริง (เช่น ในระบบเฝ้าดูเพื่อรักษาความปลอดภัย และระบบติดตามตำแหน่งเป็นต้น) ระบบนี้จำเป็นจะต้องใช้เวลาในการทำงานที่รวดเร็วและมีความถูกต้องแม่นยำที่สูงมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สิ่งที่เปลี่ยนแปลงได้บนภาพใบหน้าที่มีผลกระทบในการรู้จำใบหน้า

2.3.1 สภาพแวดล้อม

การส่องสว่าง

พื้นหลัง

2.3.2 การแสดงออกทางอารมณ์บนใบหน้า

การขมวดคิ้ว

ลักษณะของปาก (ยิ้ม หัวเราะ)

กระพริบตา

2.3.3 อื่น ๆ

การใส่แว่น

สวมหมวก

ทรงผม

หนวด

ขนาดของศีรษะเนื่องจากระยะใกล้ไกลจากกล้อง(Scaling)

ตำแหน่งและการวางตัวของศีรษะ

2.4 วิธีในการรู้จำภาพใบหน้าที่มีการวิจัยกันในปัจจุบัน

จากการสำรวจในบทความต่าง ๆ มีวิธีการมากมายที่ใช้ในการรู้จำใบหน้าซึ่งสามารถแบ่งเป็นกลุ่มได้ดังนี้

2.4.1 ลักษณะทางเรขาคณิต (Geometrical Features)

2.4.2 การเปรียบเทียบกราฟ (Graph Matching)

2.4.3 วิธีการทางสถิติ (Statistical)

2.4.4 โครงข่ายประสาทเทียม (Neural network)

2.4.5 วิธีอื่น ๆ (Other Methods)

2.4.1 ลักษณะทางเรขาคณิต (Geometrical Features)

มีนักวิจัยจำนวนมากที่เลือกใช้การรู้จำใบหน้าในลักษณะทางเรขาคณิต คืออาศัยอัตราส่วนหรือสัดส่วนของระยะทางจากตาซ้ายไปตาขวา จากตาไปจมูก จากจมูกไปปาก หรือใช้รูปร่างของปาก รูปร่างของตา และรูปร่างของคาง ซึ่งเป็นลักษณะทางกายภาพของใบหน้า ดังมีรายละเอียดของงานวิจัยดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kanade [4] ได้นำเสนอวิธีการแยกลักษณะเด่น (Automatic Feature Extraction) โดยใช้อัตราส่วนของระยะทาง (Ration of Distance) โดยมีผลการรู้จำอยู่ระหว่าง 45% -75% ของฐานข้อมูลภาพทั้งหมด 20 บุคคล

Brunelli และ Poggio [5] ได้คำนวณเซตของลักษณะทางเรขาคณิตเช่น ความกว้างของจมูก ตำแหน่งของปากบนภาพใบหน้า และรูปร่างของคาง โดยมีอัตราการรู้จำ 90% ของฐานข้อมูล 47 คน อย่างไรก็ตามในงานวิจัยนี้ได้แสดงให้เห็นว่าเมื่อถ้าใช้วิธีเปรียบเทียบเทมเพล็ตอย่างง่ายนั้น จะให้ผลการรู้จำ 100% ที่ฐานข้อมูลเดียวกัน

Cox และคณะ [6] ได้เสนอเทคนิคระยะทางผสม (Mixture Distance) ที่ทำให้ได้อัตราการรู้จำสูงถึง 95% โดยใช้ฐานข้อมูลที่สุ่มมาที่ละ 96 ภาพ จากทั้งหมด 685 ภาพ ซึ่งใบหน้าของแต่ละภาพนั้นจะถูกแทนด้วยระยะทางต่าง ๆ ถึง 30 อย่าง

ระบบการรู้จำใบหน้าแบบที่มีการพิจารณาลักษณะทางเรขาคณิตนี้ เป็นระบบที่มีประโยชน์และสามารถนำไปใช้ได้ในทางปฏิบัติในการค้นหาในฐานข้อมูลขนาดใหญ่ได้ ถ้ามีการวัดระยะทางจากจุดที่เป็นลักษณะเด่นได้อย่างถูกต้องทั้งนี้เพราะข้อมูลที่ได้จะมีจำนวนน้อยมากเมื่อเทียบกับการคำนวณแบบอื่น ทำให้เวลาในการค้นหาน้อยกว่าวิธีอื่น ๆ ดังนั้นจึงมีการคิดสร้างระบบฐานข้อมูลขึ้นมาสนับสนุนวิธีการนี้ ซึ่งฐานข้อมูลดังกล่าวถูกเรียกว่าฐานข้อมูลภาพหลายด้าน (Mug shot Database) ทั้งนี้เพราะถ้ามีภาพด้านข้างด้วยจะช่วยในการหาลักษณะเด่นได้ถูกต้องมากขึ้น แต่ความถูกต้องก็ยังขึ้นอยู่กับอัลกอริทึมที่ใช้ด้วย

2.4.2 การเปรียบเทียบกราฟ (Graph Matching)

เป็นการมองภาพในลักษณะที่เป็นเวกเตอร์ของกราฟซึ่งมีจุดและเส้นในการเชื่อมต่อเป็นโครงร่างที่แนบไปบนใบหน้าตามลักษณะเด่นที่พิจารณา เช่น รูปร่างกราฟของตา ปาก คาง หู และคิ้ว เป็นต้น

Lades และคณะ [7] เสนอสถาปัตยกรรมการเชื่อมต่อแบบยืดหยุ่น (Dynamic Link Architecture) สำหรับการรู้จำใบหน้าแบบไม่ขึ้นอยู่กับความเพี้ยน โดยใช้การเทียบกราฟแบบยืดหดได้ (Elastic Graph Matching) เพื่อหากราฟในฐานข้อมูลที่ใกล้เคียงที่สุด เมื่อวัตถุเป้าหมายจะถูกแทนด้วยกราฟห่าง ๆ กันที่มีจำนวนไม่มากนัก (Sparse Graph) ซึ่งข้อมูลจะถูกติดป้าย (Label) แทนด้วยฟังก์ชันความโค้งและขอบต่าง ๆ ก็ถูกติดป้ายซึ่งแทนด้วยระยะทางเรขาคณิต เมื่อทำการงป้อนภาพเข้ามาหนึ่งภาพ จะทำการค้นหาในฐานข้อมูลทั้งหมด 87 คน โดยภาพที่นำมาทดสอบอาจมีหลายลักษณะที่แตกต่างกัน เช่น มีการหมุนของศีรษะ ดังนั้นเวลาที่ใช้ในการค้นต่อครั้งเป็น 25 วินาที แต่บทความนี้ไม่ได้แสดงผลการรู้จำไว้เลย

เมื่อ Wiskott และคณะ [8] ได้ปรับปรุงกระบวนการของ Lades โดยทำการทดสอบฐานข้อมูล 300 คน จากฐานข้อมูล FERET ปรากฏว่ามีผลการรู้จำสูงถึง 97.30% แต่ว่าการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้ไม่ได้บอกเวลาที่ใช้ในการรู้จำ และตัวประมวลผลที่ใช้ ซึ่งอาจเป็นไปได้ว่าระบบดังกล่าวจำเป็นต้องใช้ตัวประมวลผลที่มีประสิทธิภาพและความเร็วสูงมาก เพราะวิธีการนี้คงใช้เวลาในการประมวลผลนาน

2.4.3 วิธีการทางสถิติ

วิธีการทางสถิติประกอบด้วยวิธีการเปรียบเทียบเทมเพล็ตซึ่งเป็นพื้นฐานของระบบ โดยที่ภาพที่ใช้สอนและภาพที่ใช้ทดสอบจะถูกเปรียบเทียบกัน (Match) โดยการวัดความสัมพันธ์ระหว่างภาพที่ใช้สอนกับภาพที่ใช้ทดสอบ อย่างไรก็ตามวิธีการทางสถิติยังประกอบด้วยวิธีที่ใช้การแปลงเชิงเส้น (Linear Transform) อันได้แก่ การวิเคราะห์องค์ประกอบหลัก (PCA) การวิเคราะห์การแบ่งแยกเชิงเส้น (LDA) เป็นต้น โดยระบบที่ใช้การแปลงเชิงเส้นเป็นวิธีที่เข้ามาแก้ไขวิธีการเปรียบเทียบเทมเพล็ต เพราะว่าวิธีวิธีการเปรียบเทียบเทมเพล็ตจะทำการแยกแยะภาพที่ใช้ในการสอนกับภาพที่ใช้ทดสอบในปริภูมิที่มีขนาดใหญ่

วิธีการเปรียบเทียบเทมเพล็ต [5] นี้ทำงานโดยการหาสหสัมพันธ์ของภาพ 2 ภาพโดยตรง เมื่อภาพมีขนาดเดียวกัน มีองค์ประกอบของใบหน้าวางอยู่ตรงกัน และมีการส่องสว่างของแสงในทิศทางเดียวกัน จะให้ผลการรู้จำอย่างมีประสิทธิภาพที่ดีถึง 100% แต่ถ้าภาพที่เข้ามาทดสอบมีการเปลี่ยนแปลงไปจากเดิม ต้องมีการประมวลผลก่อน (preprocessing) เพื่อทำให้ภาพเข้าสู่ตำแหน่งหรือการส่งสว่างใกล้เคียงกัน

เนื่องจากวิธีการเปรียบเทียบเทมเพล็ตนี้ ทำงานโดยการหาสหสัมพันธ์ของภาพ 2 ภาพจึงเกิดข้อจำกัดมาก ดังนั้นวิธีการที่สามารหาฟีเจอร์ที่แท้จริงของภาพและลดขนาดของภาพได้คือวิธีที่ใช้การระบบการแปลงเชิงเส้น วิธีการนี้สามารถตัดข้อมูลที่เป็นผลกระทบต่อการรู้จำใบหน้าอันได้แก่ การส่องสว่าง การแสดงทางอารมณ์บนใบหน้าได้

2.4.3.1 การวิเคราะห์องค์ประกอบหลัก (PCA)

การวิเคราะห์องค์ประกอบหลักเป็นที่ทราบกันดีว่าเหมือนวิธีคาร์ฮูเนนเลิฟ (Karhunen Loeve Expansion) โดยวิธีการนี้สามารถหาฟีเจอร์ของภาพและการสร้างภาพใหม่ได้ ซึ่งวิธีการนี้ถูกนำมาใช้ในงานต่าง ๆ เช่น การจดจำรูปแบบและการมองเห็นของคอมพิวเตอร์

Sirowich และ Kirby [9] ซึ่งพวกเขาเป็นกลุ่มแรกที่ได้ประยุกต์การวิเคราะห์องค์ประกอบหลักมาใช้ในการรู้จำใบหน้ามนุษย์

Turk และ Pentland [1] ได้เสนอวิธีการรู้จำใบหน้าที่มีชื่อว่า ใบหน้าไอเกน (Eigenface) วิธีการนี้สามรถทำได้โดยการฉายภาพใบหน้าลงในปริภูมิใบหน้าไอเกน (Eigenface space) โดยใช้ไอเกนเวกเตอร์เป็นแกนการฉาย ซึ่งไอเกนเวกเตอร์คำนวณมาจากเมตริกความแปรปรวนของภาพใบหน้า หลังจากได้ใบหน้าไอเกนแล้วนำไปค้นหาในฐานข้อมูลใบหน้าไอเกน ในการทดลองนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ใช้ฐานข้อมูลใบหน้า 2,500 ภาพ ของ 16 บุคคล ที่มีการเลื่อน การวางตำแหน่งศีรษะในหลาย ๆ ลักษณะ มีการย่อขยายและมีการเปลี่ยนแปลงของแสง ผลการทดลองเมื่อมีการเปลี่ยนแปลงของแสงมีความสามารถในการรู้จำ 96% เมื่อการวางตำแหน่งศีรษะมีการเปลี่ยนแปลงจะมีความสามารถในการรู้จำถึง 85% และในการรู้จำภาพที่มีการย่อขยายมีความถูกต้องสูงถึง 64% โดยตัวแปรของการย่อขยายของภาพนั้นได้ใช้อัลกอลิทึมทำการปรับขนาดศีรษะให้มีขนาดเดียวกับฐานข้อมูลใบหน้าไอเกน หลังจากนั้นทำการตัดภาพให้เหลือเฉพาะส่วนที่เป็นใบหน้าเพื่อลดผลจากการเปลี่ยนแปลงของผมและฉากหลัง

จากแนวความคิดของ Turk และ Pentland เป็นผลให้มีการนำไปประยุกต์กับงานด้านต่าง ๆ อีกมากอันได้แก่

Pentland และคณะ [10] มีผลการรู้จำที่ดีในฐานข้อมูลขนาดใหญ่ โดยมีความถูกต้องในการรู้จำ 95% ในฐานข้อมูล 200 คน ที่สุ่มมาจากฐานข้อมูลขนาด 3,000 คน

Moghaddame และ Pentland [11] ได้ใช้วิธีการหาใบหน้าไอเกนในการเข้ารหัสภาพใบหน้าสำหรับงานด้านการโทรภาพ (video telephone)

Lee และคณะ [12] ได้เสนอวิธีการใช้ PCA ในการตรวจหาภาพส่วนหัวในภาพพื้นหลังที่ซับซ้อนของแต่ละบุคคลและทำการรู้จำภาพ โดยทำการเปรียบเทียบกับฐานข้อมูลแต่ละบุคคลที่รู้จักอยู่แล้ว

Lee และคณะ [13] ได้ทำการศึกษาวิธี PCA เมื่อใช้กับฐานข้อมูลภาพใบหน้าขนาดใหญ่

Crowly และ Schwerdt [14] ได้เสนอวิธีการ PCA ในการเข้ารหัสและบีบอัดข้อมูลวิดีโอ

Filters ในการหาจุดที่เป็นรูปร่างและองค์ประกอบของใบหน้า และใช้วิธี PCA ในการแยกแยะ ในวิธีการนี้การใช้ Gabor Filter จะแก้ข้อบกพร่องของวิธี PCA ในเรื่องของความเปลี่ยนแปลงของภาพ เช่น แสง การหมุน ความสว่าง เป็นต้น

ในการรู้จำใบหน้าด้วยวิธี PCA ภาพใบหน้าที่เป็น 2 มิติ ซึ่งเก็บในรูปเมตริก ก่อนการคำนวณจำเป็นต้องเปลี่ยนจากเมตริกภาพเป็นเวกเตอร์ภาพ จึงเป็นผลให้ภาพมีมิติที่มากในปริภูมิ (space) ดังนั้นจึงเป็นการยากที่จะคำนวณหาเมตริกการกระจายร่วม เนื่องจากเมตริกการกระจายร่วมจะมีขนาดใหญ่เมื่อเปรียบเทียบกับจำนวนภาพที่ใช้สอน ทำให้เมื่อคำนวณหาไอเกนเวกเตอร์และค่าไอเกนแล้วคำนวณได้ยากและใช้เวลามาก Yang และคณะ ได้พัฒนาหลักการใหม่ที่นำแนวคิดมาจากวิธี PCA เพื่อแก้ไขจุดบกพร่องนี้ จึงได้เสนอวิธีการคำนวณหาเมตริกความแปรปรวนร่วมโดยใช้ภาพที่เป็น 2 มิติมาคำนวณโดยตรง วิธีการนี้มีชื่อว่า การวิเคราะห์องค์ประกอบหลักของภาพ (Image Principal Component Analysis: IMPCA หรือ 2DPCA) จากผลการทดลองของ Yang แสดงให้เห็นถึงประโยชน์ของเวลาที่ใช้ในการคำนวณและความถูกต้องในการรู้จำที่ดีกว่าวิธี PCA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.2 **การวิเคราะห์การแบ่งแยกเชิงเส้น** (Linear Discriminant Analysis)

R.A. Fisher [15] ได้พัฒนาวิธีการแบ่งแยกเชิงเส้น มาใช้ในงานด้านการจดจำวัตถุ โดยให้ชื่อว่าวิธีการแบ่งแยกเชิงเส้นของฟิชเชอร์ (Fisher Linear Discriminant: FLD)

Etemad และ Chellappa [16] ได้เสนอวิธีการแบ่งแยกเชิงเส้นของฟิชเชอร์ (Fisher Linear Discriminant: FLD) สำหรับกระบวนการรู้จำใบหน้า จุดมุ่งหมายของวิธีการนี้ต้องการหาการฉายภาพของข้อมูลภาพใบหน้าที่มีการกระจายระหว่างกลุ่มมาก ๆ และการกระจายภายในกลุ่มน้อย ๆ วิธีการนี้สามารถทำได้โดยคำนวณหาเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่มของภาพที่ใช้สอน แล้วนำเมตริกทั้งสองมาคำนวณหาไอเกนเวกเตอร์และค่าไอเกนที่สมนัยกันจากมากไปน้อย เพื่อใช้ในการฉายภาพ ดังนั้นเราจะได้ฟีเจอร์ของภาพใหม่โดยการฉายภาพที่ใช้สอนลงบนไอเกนเวกเตอร์

ปริภูมิย่อยของ LDA [2], [17], [18], [19] วิธีการนี้ได้รวมเอาวิธี PCA และ LDA เข้าด้วยกัน ดังนั้นวิธีนี้จะประกอบด้วย 2 ขั้นตอน ขั้นตอนแรก ภาพจะถูกฉายลงบนปริภูมิของใบหน้าไอเกนโดยวิธี PCA และขั้นตอนที่สอง นำใบหน้าไอเกนมาฉายด้วยวิธี LDA ในวิธีการนี้จะสามารถลดผลกระทบของตัวแปรที่มีผลกระทบในการรู้จำใบหน้า และสามารถพิจารณาข้อมูลการแบ่งแยกของกลุ่มได้ด้วย

2.4.4 **โครงข่ายประสาทเทียม** (Neural network)

Temdee [20] และคณะ ได้ทำการหาเส้นขอบของภาพใบหน้าและนำมาตัดเป็นส่วน ๆ ขององค์ประกอบภาพใบหน้าเช่น ตา จมูก ผม แล้วนำมาเข้ารหัสภาพแต่ละส่วนเพื่อใช้เป็นอินพุตให้กับโครงข่ายประสาทเทียมแบบแพร่กระจายกลับ (Backpropagation Neural Network) สำหรับใช้ในการรู้จำใบหน้า ซึ่งได้ทำการทดลองกับฐานข้อมูล ORL ซึ่งได้ผลความถูกต้องในการรู้จำ 85%

Er และคณะ [21] โดยใช้หลักการวิเคราะห์องค์ประกอบหลัก (PCA) และการวิเคราะห์การแบ่งแยกเชิงเส้น (LDA) มาทำการลดขนาดของข้อมูลเพื่อเป็นอินพุตให้กับ Radial Basis Function (RBF) ซึ่งวิธีการนี้ให้ผลความถูกต้องในการรู้จำ 94%

Demer [22] โดยใช้หลักการวิเคราะห์องค์ประกอบหลัก (PCA) มาคำนวณหาองค์ประกอบหลัก 50 ค่าแรกแล้วนำมาเป็นอินพุตของโครงข่ายประสาทเทียมแบบหลายชั้น (Standard Multilayer Perceptron) เพื่อใช้ในการรู้จำใบหน้า ซึ่งผลการรู้จำใบหน้าด้วยวิธีนี้ให้ผลการรู้จำที่ดี แต่ฐานข้อมูลที่นำมาทดลองนั้นค่อนข้างง่ายเกินไป โดยที่ภาพที่ใช้ทดลองเป็นภาพด้านหน้า ไม่มีการเปลี่ยนแปลงของแสง ไม่มีการเปลี่ยนแปลงจากการหมุน และฐานข้อมูลมีเพียง 20 คนเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lawrence และคณะ [23] ได้ใช้โครงข่ายประสาทเทียมแบบผสม (Hybrid Neural Network) โดยใช้การผสมของโครงข่ายประสาทเทียมแบบ Self-Organizing Map Neural Network (SOM) เพื่อลดข้อมูลภาพใบหน้า และใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network: CNN) ในการรู้จำใบหน้า ซึ่ง Lawrence และคณะได้เปรียบเทียบผลการรู้จำกับฐานข้อมูล ORLได้ดังนี้

1. ใช้ SOM ในการลดมิติข้อมูล และใช้ CNN ในการรู้จำ ให้ความถูกต้องในการรู้จำ 96.20%

2. ใช้การแปลงแบบคาร์ฮูเนนเลิฟ (KL Transform) ในการลดข้อมูลภาพใบหน้า และใช้ CNN ในการรู้จำ ให้ความถูกต้องในการรู้จำ 94.70%

3. ใช้ SOM ในการลดข้อมูล และใช้โครงข่ายประสาทเทียมแบบเปอร์เซบตรอนหลายชั้น (Multilayer Perceptron) ในการรู้จำ ซึ่งให้ความถูกต้องในการรู้จำ 60.00%

4. ใช้การแปลงแบบคาร์ฮูเนนเลิฟให้ความถูกต้อง 89.50%

2.4.5 วิธีอื่น ๆ (Other Methods)

ข้อมูลเชิงระยะทาง (Range Data)

เป็นวิธีการรู้จำใบหน้าที่แตกต่างจากวิธีอื่น โดยการใช้ข้อมูลเชิงระยะทาง ซึ่งข้อมูลภาพของแต่ละบุคคลได้มาจากการสแกนด้วยระบบเลเซอร์สแกน ที่มีข้อมูลเชิงลึกประกอบด้วย แล้วทำการชี้เฉพาะบุคคลในรูป 3 มิติ [24]

การสแกนด้วยอินฟาเรด (Infrared Scanning)

Yoshitomi [25] วิธีการนี้ได้ใช้แสงอินฟาเรดตรวจจับใบหน้าเพื่อหาการกระจายของอุณหภูมิบนใบหน้าและโครงร่างใบหน้า เมื่อนำการกระจายของอุณหภูมิมาหาค่าเฉลี่ยของอุณหภูมิเพื่อใช้เป็นอินพุตให้กับโครงข่ายประสาทเทียม และข้อมูลโครงร่างใบหน้านำมาปรับปรุงขนาดและตำแน่งเพื่อใช้กับวิธีการแยกแยะโดยการสอน (Supervisor Classification) ผลที่ได้จากทั้งสองวิธีนำมารวมกันเพื่อชี้เฉพาะบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีพื้นฐานที่เกี่ยวข้อง

3.1 การวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis: PCA)

3.1.1 แนวความคิดของการวิเคราะห์องค์ประกอบหลัก

การวิเคราะห์องค์ประกอบหลักเป็นการแทนเซตของข้อมูลด้วยเซตข้อมูลใหม่ที่มีขนาดเล็กกว่าข้อมูลเดิม โดยข้อมูลใหม่นี้จะไม่มีความสัมพันธ์กันระหว่างข้อมูลภายในเซตเลย ซึ่งข้อมูลใหม่นี้ถูกเรียกว่า องค์ประกอบหลัก (Principal Components) ถ้ามีเซตข้อมูลที่มีความสัมพันธ์กันอยู่ภายในข้อมูล ซึ่งมีขนาด D มิติ สามารถใช้การรวมกันเชิงเส้นในการลดข้อมูลลงเป็น d มิติ โดยที่ d มีค่าน้อยกว่า D

ในการพิสูจน์ทฤษฎี PCA นั้นสามารถกระทำได้สองทางโดยที่ วิธีที่หนึ่งพิสูจน์มาจากผลรวมของระยะทางที่น้อยที่สุด (Least Squared Distance) ที่นำเสนอโดย Duda [26] และวิธีที่สองพิสูจน์มาจากการเปลี่ยนตัวแปรที่เสนอโดย Hotelling [27]

1) ผลรวมของระยะทางที่น้อยที่สุด (Least Squares Distance)

Duda เป็นคนแรกที่แนะนำวิธี PCA จากแนวความคิดในการหาจุดและเส้นเหมาะสมที่สุด โดยกำหนดให้มีจำนวนตัวอย่าง M ค่าและแต่ละค่ามีขนาด D มิติ ดังนั้นจำนวนตัวอย่างทั้งหมดคือ $t_1, t_2, \ldots, t_M$ วิธีการในการลดขนาดข้อมูลทั้งหมดให้มีขนาด d มิติ โดยที่ $d < D$ จะทำได้อย่างไร คำตอบในการแก้ปัญหาคือการทดลองแทนข้อมูลใหม่ (representation) เป็น 0 มิติ และ 1 มิติ โดยที่ T_0 คือข้อมูลที่ถูกแทนเป็น 0 มิติ (0-dimensional representation) เพื่อหาความสัมพันธ์ข้อมูลต้นแบบกับข้อมูลที่แสดงใหม่จึงทดลองใช้ T_0 มาคำนวณหาค่าผลรวมความผิดพลาดน้อยที่สุด (Least Squares Error) ระหว่าง T_0 กับ t_M ดังนั้นสมการผลรวมความผิดพลาดน้อยที่สุด คือ

$$J_0(T_0) = \sum_{k=1}^{M} \left\| T_0 - t_k \right\|^2 \tag{3.1}$$

วิธีการหาค่า T_0 ที่ทำให้ J_0 มีค่าน้อยที่สุดสามารถพิสูจน์ได้จาก
เมื่อกำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\bar{t} = \frac{1}{M}\sum_{k=1}^{M} t_k \tag{3.2}$$

$\bar{t}$ คือค่าเฉลี่ยตัวอย่าง

$$\begin{aligned} J_0(T_0) &= \sum_{k=1}^{M} \left\| (T_0 - \bar{t}) - (t_k - \bar{t}) \right\|^2 \\ &= \sum_{k=1}^{M} \left\| (T_0 - \bar{t}) \right\|^2 - 2\sum_{k=1}^{M} (T_0 - \bar{t})^T (t_k - \bar{t}) + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2 \\ &= \sum_{k=1}^{M} \left\| (T_0 - \bar{t}) \right\|^2 - 2(T_0 - \bar{t})^T \sum_{k=1}^{M} (t_k - \bar{t}) + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2 \\ &= \sum_{k=1}^{M} \left\| (T_0 - \bar{t}) \right\|^2 + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2 \end{aligned} \tag{3.3}$$

จากสมการที่ (3.3) เทอมสุดท้ายไม่มีตัวแปร T_0 สามารถตัดทิ้งได้ และจากสมการที่ (3.3) จะสังเกตเห็นว่าค่าที่ทำให้สมการมีค่าน้อยที่สุดก็ต่อเมื่อ $T_0 = \bar{t}$ ดังนั้นค่าเฉลี่ยของตัวอย่าง $\bar{t}$ ก็คือข้อมูลที่ถูกแสดงใหม่ใน 0 มิติ

จากการทดลองใช้ T_0 ไม่สามารถแสดงให้เห็นถึงความสัมพันธ์ของตัวแปรต่างได้ ดังนั้นจึงทดลองใช้ข้อมูลที่ถูกแทนเป็น 1 มิติ (1-dimensional representation) โดยการฉายข้อมูลลงบนเส้นผ่านทางค่าเฉลี่ยของตัวอย่าง

เมื่อกำหนดให้ u เป็นเวกเตอร์ทิศทางตามเส้นที่มีขนาด 1 หน่วย ดังนั้นสามารถสร้างสมการความสัมพันธ์ของตัวแปรในรูปแบบทางคณิตศาสตร์ของการสร้างค่าใหม่ คือ

$$t = \bar{t} + yu \tag{3.4}$$

โดยที่ y คือขนาดที่สัมพันธ์ของระยะทางจาก t ถึง $\bar{t}$ ถ้าต้องการแทนค่าข้อมูลใหม่ของ $t_1, t_2, \ldots, t_M$ ทั้งหมดจะต้องใช้สมการ $t_k = \bar{t} + y_k u$ จึงสามารถหาค่าที่เหมาะสมที่สุดของค่าสัมประสิทธิ์ (Coefficients) y_k จากฟังชันค่าผิดพลาดที่น้อยที่สุด

$$\begin{aligned} J_1(y_k) &= \sum_{k=1}^{M} \left\| (\bar{t} + y_k u) - t_k \right\|^2 \\ &= \sum_{k=1}^{M} \left\| y_k u - (t_k - \bar{t}) \right\|^2 \end{aligned} \tag{3.5}$$

ขยายสมการโดยใช้ความสัมพันธ์ $\|t\|^2 = t^T t$ และ $u^T u = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$J_1(y_k) = \sum_{k=1}^{M}(y_k u - (t_k - \bar{t}))^T (y_k u - (t_k - \bar{t})) \quad (3.6)$$
$$= \sum_{k=1}^{M}(y_k^2 u^T u - 2y_k u^T (t_k - \bar{t}) + (t_k - \bar{t})^T (t_k - \bar{t}))$$
$$= \sum_{k=1}^{M} y_k^2 - 2\sum_{k=1}^{M} y_k [u^T (t_k - \bar{t})] + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2$$

จากสมการที่ (3.6) ทำการหา Partial Differentiate แล้วกำหนดให้มีค่าเท่ากับศูนย์ ดั้งนั้น จะได้สมการการฉายข้อมูลลงในปริภูมิ (Space) โดยใช้เส้นตรงเพียงหนึ่งเส้นที่มาจากค่า J_1 น้อยที่สุด สามารถทำได้โดยการฉายข้อมูล t_k ลงบนเส้นในทิศทางของ u ที่ผ่านทาง $\bar{t}$

$$y_k = u^T (t_k - \bar{t}) \quad (3.7)$$

แทนค่า (3.7) ใน (3.6) เพื่อคำนวณหาทิศทางที่ดีที่สุดของ u ที่มาจากค่าผลรวมความผิดพลาดต่ำสุด

$$J_1(y_k) = \sum_{k=1}^{M} y_k^2 - 2\sum_{k=1}^{M} y_k [y_k] + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2 \quad (3.8)$$
$$= -\sum_{k=1}^{M} y_k^2 + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2$$
$$= -\sum_{k=1}^{M} [u^T (t_k - \bar{t})]^2 + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2$$

ขยายสมการโดยใช้ความสัมพันธ์ $t^2 = tt^T$ และ $(AB)^T = B^T A^T$

$$J_1(u) = -\sum_{k=1}^{M} [u^T (t_k - \bar{t})][u^T (t_k - \bar{t})]^T + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2 \quad (3.9)$$
$$= -\sum_{k=1}^{M} u^T (t_k - \bar{t})(t_k - \bar{t})^T u + \sum_{k=1}^{M} \left\| (t_k - \bar{t}) \right\|^2$$

แทนค่าสมการ $\sum_{k=1}^{M}(t_k - \bar{t})(t_k - \bar{t})^T$ ด้วย S ซึ่ง S คือเมตริกการกระจาย โดยเปรียบเทียบกับเมตริกความแปรปรวนร่วมจะมีค่าเป็น $(M-1)$ เท่าของเมตริกความแปรปรวนร่วม Σ (Covariance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Matrix) โดยที่เมตริกความแปรปรวนร่วมคือ $\Sigma = \frac{1}{M-1}\sum_{k=1}^{M}(t_k - \bar{t})(t_k - \bar{t})^T$ ดังนั้นเมตริกการกระจาย คือ

$$S = \sum_{1=k}^{M}(t_k - \bar{t})(t_k - \bar{t})^T \tag{3.10}$$

แทนค่าสมการที่ (3.10) ลงในสมการที่ (3.9) จะได้

$$J_1(u) = -\sum_{k=1}^{M} u^T S u + \sum_{k=1}^{M}\left\|(t_k - \bar{t})\right\|^2 \tag{3.11}$$

จากสมการที่ (3.11) เทอมที่สองไม่มีค่าที่สัมพันธ์กับ u จึงกำหนดเป็นค่าคงที่ ดังนั้นในการที่ J_1 จะมีค่าน้อยที่สุดก็ต่อเมื่อ $u^T S u$ มีค่ามากที่สุด

การแก้ปัญหาสมการที่ (3.11) จะใช้วิธี Lagrange multipliers จะได้สามาการใหม่คือ

$$L(u) = u^T S u - \lambda(u^T u - 1) \tag{3.12}$$

การหาค่ามากที่สุดสามารถทำได้โดยการหา Gradient vector แล้วเซตให้มีค่าเท่ากับ 0

$$\frac{\partial L}{\partial u} = 2Su - 2\lambda u = 0 \tag{3.13}$$

$$Su = \lambda u \tag{3.14}$$

เมื่อตรวจสอบแนวทางการแก้สมการแล้วจะพบว่า u ก็คือไอเกนเวกเตอร์ (Eigenvector) และ λ เป็นค่าไอเกน ของ S ถ้า u เป็นอินพุตเวกเตอร์ที่ใช้สำหรับการแปลง (Transform) ซึ่งคำนวณมาจากเมตริก S ดังนั้นเอาท์พุตก็คือขนาดของเวกเตอร์ที่มีทิศทางเดียวกับอินพุต

แทนค่าสมการที่ (3.14) ลงในสมการวัตถุประสงค์จะทำให้การหาค่ามากสุดของสมการมาจากการเลือกไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนที่มากที่สุดตามลำดับ

$$\max(u^T S u) = \max u^T(\lambda u) \tag{3.15}$$

$$= \max \lambda$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลลัพธ์ที่ได้จากข้อมูลที่สร้างใหม่ 1 มิติแสดงให้เห็นว่าแกนการฉายก็คือไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนที่มากที่สุดตามลำดับ ในทางเดียวกันสามารถขยายการแทนข้อมูลใหม่เป็น d มิติ โดยใช้การฉายข้อมูลลง d เส้น ดังนั้นสมการที่ (3.4) และ (3.5) สามารถแทนค่าสมการใหม่ได้เป็น

$$\mathrm{t} = \bar{\mathrm{t}} + \sum_{i=1}^{d} y_d \mathrm{u}_d \tag{3.16}$$

$$J_d = \sum_{k=1}^{M} \left\| (\bar{\mathrm{t}} + \sum_{i=1}^{d} y_{ki} \mathrm{u}_i) - \mathrm{t}_k \right\|^2 \tag{3.17}$$

สรุปได้ว่าเมื่อมีข้อมูลเป็น D มิติสามารถแทนข้อมูลเป็น d มิติ โดยให้มีค่าความผิดพลาดน้อยที่สุด สามารถทำได้โดยการคำนวณหาไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนที่มากที่สุดตั้งแต่ 1 ถึง d ของเมตริกการกระจาย S โดยที่ S เป็นเมตริกสมมาตรและเป็นจำนวนจริง u_i เป็นโอทอโกนอล (Orthogonal) ที่ประกอบด้วยแกนการฉาย ค่าสัมประสิทธิ์ y_k ในสมการที่ (3.16) ถูกเรียกว่าองค์ประกอบหลัก (Principal component) ดังนั้น y_k จะเป็นตัวแทนข้อมูลที่มีขนาด d ของข้อมูล t_k ที่มีขนาด D มิติ

2) **การเปลี่ยนตัวแปร**

Hotelling ได้นำเสนอ PCA ที่แตกต่างออกไปและมีหนังสืออีกหลายเล่มได้นำเสนอเหมือนหลักการนี้อันได้แก่ Chatfield, Joliffe, Johnson ซึ่งนำแนวคิดมาจากการการแปลงเชิงเส้นของตัวแปรที่มีความสันพันธ์กัน (correlate) ไปเป็นตัวแปรใหม่ที่ไม่สัมพันธ์กันตามลำดับการลดลงของความแปรปรวน

กำหนดให้มีเวกเตอร์ t ใด ๆ ที่มีขนาด D มิติ และมีค่าเฉลี่ย $\bar{\mathrm{t}}$ และค่าเมตริกความแปรปรวนร่วม $\sum$ ขนาด DxD ต้องการคำนวณหาเวกเตอร์ใหม่ y ที่ไม่มีความสัมพันธ์กันและมีค่าความแปรปรวนลดลงตามลำดับ จากหลักการรวมกันเชิงเส้นสามารถแสดงเป็นสมการได้ดังนี้

$$y_k = \mathrm{u}_{1k}\mathrm{t}_1 + \mathrm{u}_{2k}\mathrm{t}_2 \ldots + \mathrm{u}_{Dk}\mathrm{t}_D = \mathrm{u}_k^{\mathrm{T}}\mathrm{t} \tag{3.18}$$

จากสมการสามารถเขียนให้อยู่ในรูปของเมตริกได้ โดยที่ Y เวกเตอร์มีขนาด dx1 t มีขนาด Dx1 และ U มีขนาด Dxd

$$\mathrm{y} = \mathrm{U}^{\mathrm{T}}\mathrm{t} \tag{3.19}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

$$\begin{bmatrix} \uparrow \\ y \\ \downarrow \end{bmatrix}_{dx1} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ u_1 & u_2 & \cdots & u_d \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}^T_{Dxd} \begin{bmatrix} \uparrow \\ t \\ \downarrow \end{bmatrix}_{Dx1}$$

จากสมการที่ (3.18) จะเห็นว่า u เป็นเพียงแกนของทิศทางเท่านั้นจึงกำหนดให้ u เป็นเวกเตอร์หนึ่งหน่วยที่มีคุณสมบัติคือ

$$u_k^T u_k = \sum_{k=1}^{D} u_{kj}^2 = 1 \tag{3.20}$$

จากแนวคิดการการแปลงเชิงเส้นของตัวแปรที่มีความสัมพันธ์กัน (Correlate) ไปเป็นตัวแปรใหม่ที่ไม่สัมพันธ์กันตามลำดับการลดลงของความแปรปรวน ดังนั้นการคำนวณหาเมตริกความแปรปรวนร่วมของ Y ทำได้โดย

k^{TH} องค์ประกอบหลัก = การรวมกันเชิงเส้นของ $u_k^T t$ ที่มีค่าความแปรวนมากที่สุด $\max(\mathrm{Var}(u_k^T t))$

$$\begin{aligned} \mathrm{Var}(y_k) &= \mathrm{Var}(u_k^T t) \\ &= E[(U^T(t-\bar{t}))^2] \\ &= E[U^T(t-\bar{t})(U^T(t-\bar{t}))^T] \\ &= E[U^T(t-\bar{t})(t-\bar{t})^T U \\ &= U^T E[(t-\bar{t})(t-\bar{t})^T] U \\ &= U^T \Sigma U \end{aligned} \tag{3.21}$$

จากสมการที่ (3.21) สามารแก้ปัญหาได้เหมือนการแก้ปัญหาในสมการ (3.11) โดยการใช้วิธีการ Lagrange multipliers ดังนั้นจึงสามารถสรุปได้ว่าแกนการฉายก็คือไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกน λ ที่เรียงจากมากไปน้อยตามลำดับ ซึ่งคำนวณมาจากเมตริกการกระจาย

3.1.2 หลักการคำนวณของการวิเคราะห์องค์ประกอบหลัก

วิธีหาปริภูมิใบหน้าไอเกน (Eigenface) ที่นำเสนอโดย Turk และ Pentland เป็นวิธีที่ประยุกต์มาจากวิธี PCA ซึ่งนำมาใช้ในการรู้จำใบหน้า โดยการนำภาพใบหน้าแต่ละภาพมากำหนดเป็นจุดในปริภูมิภาพใบหน้า ปริภูมิภาพใบหน้าจะมีมิติเท่ากับจำนวนพิกเซลของภาพ เช่น ภาพระดับสีเทามีขนาดเท่ากับ m x n จะได้มิติของปริภูมิภาพคือ D ซึ่ง D จะเท่ากับ m x n แล้ว

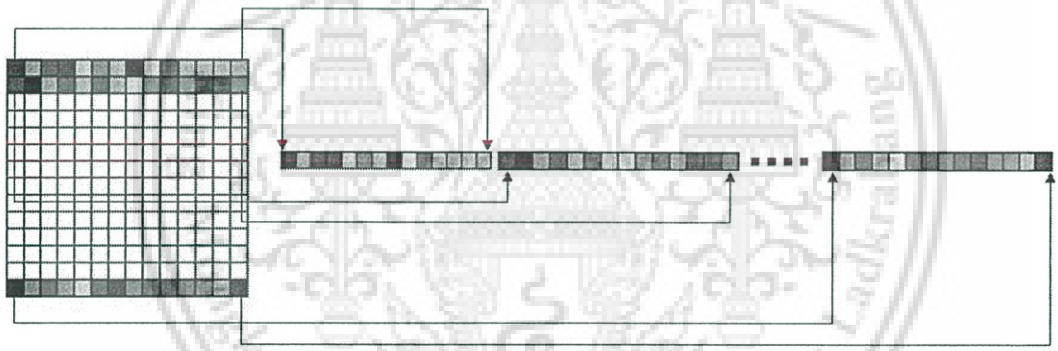
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61197

ทำการลดมิติไปเป็นปริภูมิที่มีมิติน้อยลงเพื่อหาฟีเจอร์ของภาพ ซึ่งถูกเรียกว่าปริภูมิใบหน้าไอเกน (Eigenface Space)

วิธีการหาปริภูมิใบหน้าไอเกน ทำโดยคำนวณหาไอเกนเวกเตอร์ของเมตริกซ์ความแปรปรวนของภาพที่ใช้สอน ที่สมนัยกับค่าไอเกนที่มีค่าความแปรปรวนจากมากไปน้อย เมื่อนำภาพที่ใช้สอนไปฉายด้วยไอเกนเวกเตอร์ที่ได้ จะได้ปริภูมิใบหน้าไอเกน

ในการทดสอบ จะนำภาพที่ใช้สอนทั้งหมดฉายด้วยวิธี Eigenface จะได้ปริภูมิใบหน้าไอเกนที่ใช้สอน แล้วนำภาพที่ใช้ในการทดสอบมาฉายด้วยไอเกนเวกเตอร์ของภาพที่ใช้ในการสอน จะได้ปริภูมิใบหน้าไอเกนที่ใช้ในการทดสอบ ในการแยกแยะภาพใบหน้าทำโดยหาระยะทางของปริภูมิที่ใช้สอนกับปริภูมิที่ใช้ทดสอบ

วิธีการคำนวณ เมื่อกำหนดให้มีภาพที่ใช้สอน M ภาพ คือ $A_1, A_2, \ldots, A_M$ แต่ละภาพมีขนาด m x n จุดภาพ โดยที่แต่ละภาพถูกจะถูกเปลี่ยนให้เป็นเวกเตอร์ t ตามรูปที่ 3.1 จากรูปที่ 3.1 นำแต่ละแถวของภาพมาเรียงต่อกันจากบนลงล่างให้เป็นเวกเตอร์ ดังนั้นจะได้แต่ละเวกเตอร์จะมีขนาด $1 \text{x } D$ ($D = m \text{ x } n$)



รูปที่ 3.1 ตัวอย่างภาพที่เปลี่ยนจากเมตริกภาพเป็นเวกเตอร์ภาพ

หลังจากภาพที่ใช้สอนทั้งหมดถูกเปลี่ยนให้เป็นเวกเตอร์ทั้งหมดแล้ว นำเวกเตอร์ทั้งหมดมาเรียงตามตามคอลัมน์จะได้เป็นเมตริกของชุดภาพตัวอย่างที่ใช้ในการสอนตามสมการ

$$t = [t_1, t_2, \ldots, t_M] \tag{3.22}$$

คำนวณหาค่าเฉลี่ยของภาพใบหน้า(Mean Face) เพื่อหาจุดศูนย์กลางของข้อมูลใหม่

$$\bar{t} = \frac{1}{M} \sum_{i=1}^{M} t_i \tag{3.23}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณหาภาพที่ถูกลบด้วยค่าเฉลี่ยของภาพใบหน้า (Mean subtracted image)

$$T_i = t_i - \bar{t} \tag{3.24}$$

สร้างเมตริกที่ใช้ในการคำนวณหาเมตริความแปรปรวนร่วม

$$T = [T_1, T_2, \ldots, T_M] \tag{3.25}$$

คำนวณหาเมตริกความแปรปรวนร่วม (Covariance Matrix) เพื่อวัดการกระจายของข้อมูล

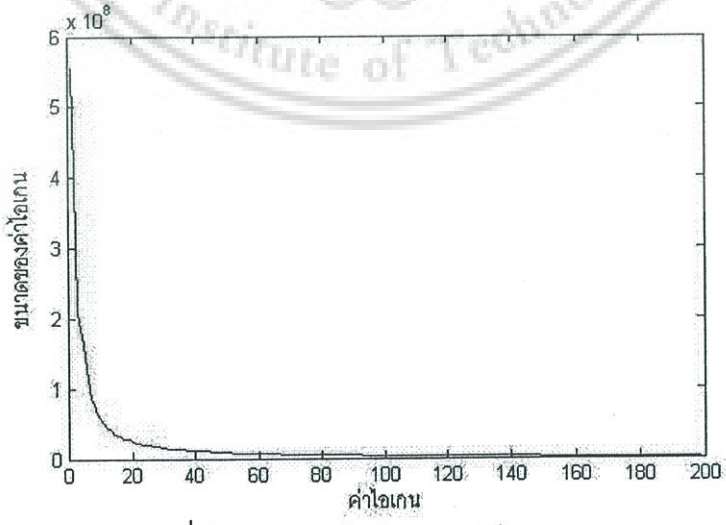
$$S = T.T^T = \frac{1}{M}\sum_{i=1}^{M} T_i T_i^T \tag{3.26}$$

คำนวณหาคำนวณหาค่าไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกน จากเมตริกความแปรปรวนร่วมด้วยสมการ

$$Su = \lambda u \tag{3.27}$$

เรียงลำดับค่าไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนจากมากไปน้อย แล้วเลือกเฉพาะไอเกนเวกเตอร์ที่สัมพันธ์กับค่าไอเกนที่ไม่เท่ากับศูนย์ ดังนั้นจะได้เมตริกใหม่ของไอเกนเวกเตอร์คือ

$$U_{PCA} = [u_1\ u_2\ u_3\ \ldots\ u_{M-1}] \tag{3.28}$$



รูปที่ 3.2 ตัวอย่างค่าไอเกนที่เรียงจากมากไปน้อยของวิธี Eigenface จากฐานข้อมูล ORL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การหาฟีเจอร์ของภาพ (Feature Extraction)

เมื่อได้เวกเตอร์ที่เหมาะสมที่ใช้ในการฉายภาพของ Eigenface คือ $U_{PCA}= [u_1, u_2, \ldots, u_{M-1}]$ โดยที่ d เป็นค่าที่เลือกระหว่าง 1 ถึง M-1 และถ้ามีภาพที่ใช้สอนทั้งหมด M ภาพ จะสามารถหาฟีเจอร์ที่ของภาพที่ใช้สอนทั้งหมดจากสมการ

$$Y_k = U_d^T (t_k - \bar{t}) ; k = 1, 2, \ldots, M \tag{3.29}$$

3.1.4 การสร้างภาพใหม่ (Reconstruction)

วิธีของ Eigenface สามารถนำเอาฟีเจอร์ของภาพและแกนการฉายที่เหมาะสมที่สุดของ Eigenface มาสร้างเป็นภาพใบหน้าใหม่ได้ตามขั้นตอนดังนี้

โดยที่ $y = [y_1,\ldots,y_d]$ เป็นฟีเจอร์ของภาพ และ$U_d = [u_1, u_2, \ldots,u_d]$ เป็นแกนการฉายที่เหมาะสมที่สุด ดังนั้นจาก $y= U_d^T (t - \bar{t})$ และ U_d เป็นออโธนอมอล($U^{-1} = U^T$) จะได้สมการการสร้างภาพใหม่คือ

$$\tilde{t} = U_d y + \bar{t} \tag{3.30}$$

d =20 d = 40 d=60 d = 80

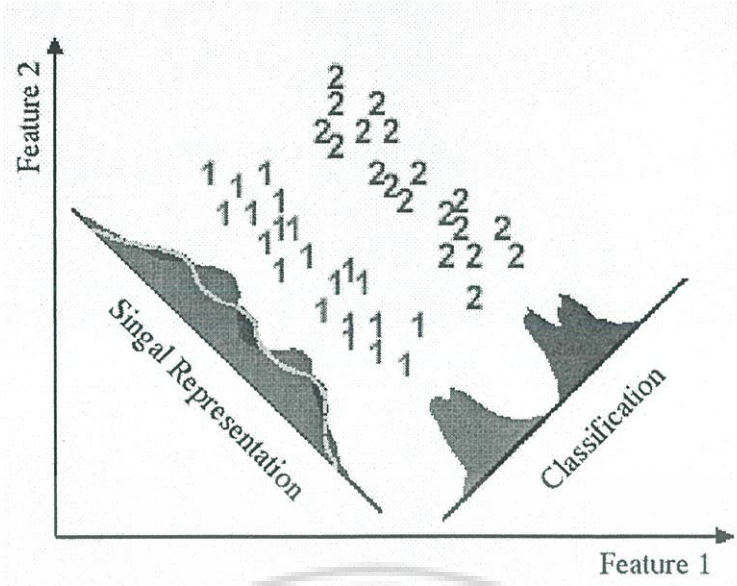
รูปที่ 3.3 ตัวอย่างภาพใบหน้าบุคคลหนึ่งที่สร้างใหม่ด้วยวิธี Eigenface จากฐานข้อมูล ORL

3.2 การวิเคราะห์การแบ่งแยกเชิงเส้นของฟิชเชอร์ (Fisher Linear Discriminant : FLD)

3.2.1 แนวความคิดของการวิเคราะห์การแบ่งแยกเชิงเส้นของฟิชเชอร์

เมื่อวิธี PCA ถูกใช้สำหรับการแทนข้อมูลใหม่ (Representation data) โดยไม่คำนึงถึงการแบ่งแยกข้อมูลระหว่างกลุ่มที่ต่างกัน ดังตัวอย่างในรูปที่ 3.4 เมื่อมีข้อมูลอยู่สองกลุ่มได้แก่ ตัวเลข 1 และ ตัวเลข 2 เมื่อนำข้อมูลมาคำนวณด้วยวิธี PCA จะเห็นว่าตัวเลข 1 และ สอง 2 จะถูกแทนด้วยค่าข้อมูลใหม่ซึ่งจะเห็นว่าข้อมูลใหม่นี้มีการผสมกันระหว่างข้อมูลทั้งสองกลุ่ม แต่เมื่อคำนวณด้วยวิธี FLD ข้อมูลใหม่ของแต่ละกลุ่มจะถูกแบ่งแยกกันอย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ความแตกต่างระหว่างวิธี PCA และวิธี FLD

แนวคิดของวิธี FLD มาจากความต้องการที่จะฉายข้อมูลจาก D มิติลงบนเส้น เพื่อให้ข้อมูลนั้นมีการแบ่งแยกของกลุ่มที่ดี ถ้าตัวอย่างข้อมูลมีขนาด D มิติ มีการรวมตัวกันแน่น เมื่อทำการฉายลงบนแกนใด ๆ ทำให้เกิดการปนกันของตัวอย่างทุก ๆ กลุ่ม ดังนั้นจึงทำให้กระบวนการรู้จำแย่ลงไปด้วย อย่างไรก็ตามการแก้ปัญหาสามารถทำได้การทดลองเคลื่อนเส้นไปรอบ ๆ เพื่อดูตำแหน่งของเส้นที่ทำให้การฉายตัวอย่างข้อมูลมีการแบ่งแยกที่ดี

การหาแกนการฉายที่ทำให้ตัวอย่างแบ่งแยกได้ดี ขั้นแรกได้ทำการทดลองกับตัวอย่างเพียง 2 กลุ่มเพื่อดูความสัมพันธ์ แล้วหลังจากนั้นจึงขยายเป็นจำนวนกลุ่มที่มากขึ้น

เมื่อกำหนดให้จำนวนตัวอย่างทั้งหมดคือ $t_1, t_2,\ldots, t_M$ แต่ละตัวอย่างมีขนาด D มิติ โดยถูกแบ่งออกเป็นสองกลุ่ม T_1 และ T_2 กลุ่มที่ 1 มีจำนวนตัวอย่าง N_1 และกลุ่มที่ 2 มีจำนวนตัวอย่าง N_2 จากหลักการรวมกันเชิงเส้น (Linear Commination) จะได้สมการ

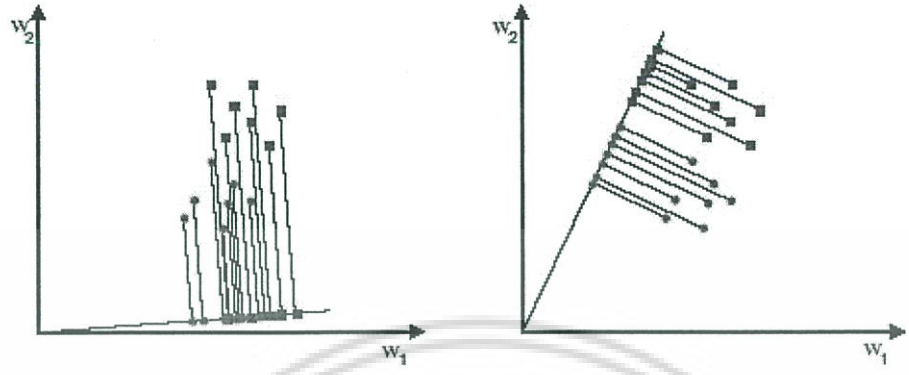
$$Y = W^T t \tag{3.31}$$

จะได้ $y_1, y_2,\ldots,y_M$ เป็นตัวอย่างที่ถูกฉายซึ่งถูกแบ่งเป็นสองกลุ่มคือ $\tilde{T}_1$ และ $\tilde{T}_2$ โดยที่แต่ละค่าของ y_M มาจากการฉายค่าของ t_M ลงบนเส้นในทิศทางของ w

จากแนวความคิดของวิธีการนี้คือต้องการให้ตัวอย่างในกลุ่ม T_1 ที่ถูกฉายตกลงบนเส้นให้อยู่ในกลุ่มเดียวกัน ส่วนตัวอย่างที่เหลือก็ให้ตกลงบนเส้นแล้วมาอยู่ในกลุ่มที่เหลือจึงจะทำให้การแบ่งแยกมีประสิทธิภาพ ดังนั้นสิ่งที่สำคัญที่สุดคือการเลือกแกนการฉาย w รูปที่ 3.5 แสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของการเลือกค่าของ w ที่ใช้ในการฉายของตัวอย่างภาพ 2 มิติ จากรูปจะเห็นว่าเมื่อเลือกแกนการฉาย w เหมือนรูปที่ 3.5 (ก) จะทำให้ตัวอย่างที่ถูกฉายของสองกลุ่มมีการปนกัน แต่ถ้าเลือกแกนการฉาย w เหมือนรูปที่ 3.5 (ข) จะทำให้กลุ่มตัวอย่างมีการแบ่งแยกกันอย่างชัดเจน



(ก) (ข)

รูปที่ 3.5 ผลของการเลือกค่าของ w ที่ใช้ในการฉายภาพ

(ก) การเลือกค่าของ w ที่ทำให้ตัวอย่างที่ถูกฉายของสองกลุ่มมีการปนกัน

(ข) การเลือกค่าของ w ที่ทำให้ตัวอย่างที่ถูกฉายมีการแบ่งแยกกันอย่างชัดเจน

วิธีการที่จะหาทิศทางการฉายที่ดีที่สุด โดยทดลองวัดการแบ่งแยกของกลุ่มจากผลต่างของค่าเฉลี่ยตัวอย่างที่ถูกฉายในแต่ละกลุ่ม โดยที่ผลต่างของค่าเฉลี่ยตัวอย่างที่ถูกฉายมีค่ามากจะแสดงให้เห็นว่าข้อมูลของแต่ละกลุ่มตัวอย่างที่ถูกฉายห่างกันมาก ถ้ากำหนดให้ $\bar{t}_i$ คือ ค่าเฉลี่ยแต่ละกลุ่มตัวอย่าง และ $\tilde{m}_i$ คือค่าเฉลี่ยแต่ละกลุ่มของตัวอย่างที่ถูกฉาย

$$\bar{t}_i = \frac{1}{N_i} \sum_{t \in T_i} t \tag{3.32}$$

และ

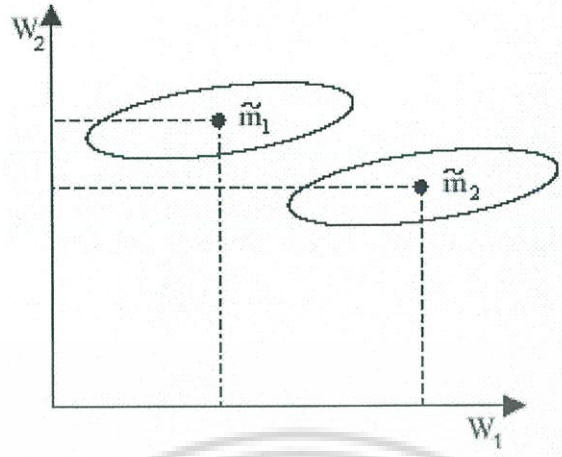
$$\tilde{m}_i = \frac{1}{N_i} \sum_{y \in \tilde{T}_i} y \tag{3.33}$$

$$= \frac{1}{N_i} \sum_{t \in \tilde{T}_i} w^T t \tag{3.34}$$

จากการทดลองเลือกระยะทางระหว่างค่าเฉลี่ยของตัวอย่างที่ถูกฉายเป็นสมการวัตถุประสงค์ (Objective function) จะได้สมการวัตถุประสงค์เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$J(w) = |\tilde{m}_1 - \tilde{m}_2| = |w^T(\bar{t}_1 - \bar{t}_2)| \quad (3.35)$$



รูปที่ 3.6 การเลือกแกนการฉายจากระยะทางระหว่างค่าเฉลี่ยของตัวอย่างที่ถูกฉาย 2 กลุ่ม

จากรูปที่ 3.6 เมื่อเลือกแกนการฉายเป็น w_2 ตัวอย่างที่ถูกฉายลงบนแกนนี้จะมีการปนกันน้อยกว่าเมื่อเลือกแกนการฉายเป็น w_1 ทั้งที่ระยะห่างระหว่างค่าเฉลี่ยของตัวอย่างที่ถูกฉายของแกนการฉาย w_2 มีค่าน้อยกว่าแกนการฉาย w_2 ซึ่งแสดงให้เห็นว่าการใช้ระยะห่างระหว่างค่าเฉลี่ยของตัวอย่างที่ถูกฉายเพียงอย่างเดียวยังไม่เพียงพอที่จะทำให้การฉายตัวอย่างมีการแบ่งแยกที่ดี

ฟิชเชอร์ได้นำเสนอการแก้ปัญหานี้ โดยการหาค่ามากที่สุดของฟังก์ชันอัตราส่วนของระยะทางระหว่างค่าเฉลี่ยของตัวอย่างที่ถูกฉายกับผลรวมการกระจายภายในกลุ่มของตัวอย่างที่ถูกฉาย เมื่อการกระจายภายในกลุ่มจะมีค่าเท่ากับค่าความแปรปรวนของตัวอย่างภายในกลุ่ม โดยที่ค่าการกระจายของแต่ละกลุ่มตัวอย่างที่ถูกฉายคือ

$$\tilde{S}_i^2 = \sum_{y \in \tilde{T}_i} (y - \tilde{m}_i)^2 \quad (3.36)$$

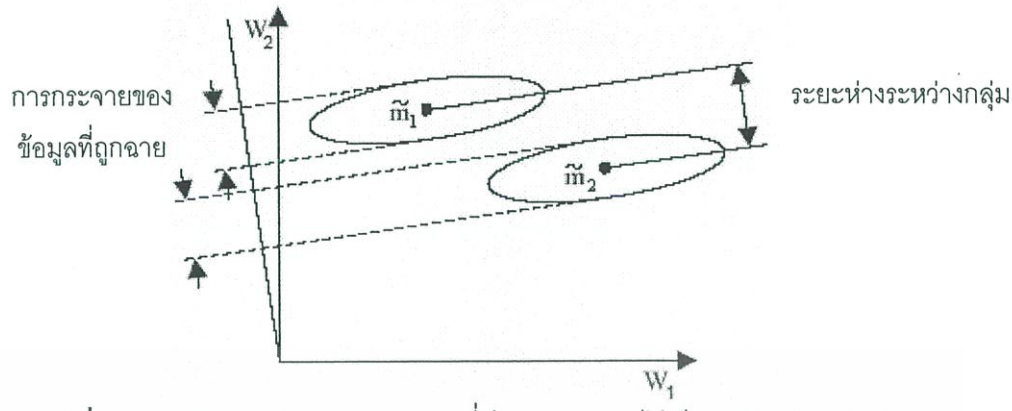
และ $\tilde{S}_1^2 + \tilde{S}_2^2$ ถูกเรียกว่า การกระจายภายในกลุ่มของตัวอย่างที่ถูกฉาย

ดังนั้นสมการการแบ่งแยกเชิงเส้นของฟิชเชอร์ที่ได้จากการแปลงเชิงเส้น $w^T t$ ซึ่งมีค่ามากที่สุดของข้อมูล 2 กลุ่มจะได้สมการวัตถุประสงค์เป็น

$$J(w) = \frac{|\tilde{m}_1^2 + \tilde{m}_2^2|}{\tilde{S}_1^2 + \tilde{S}_2^2} \quad (3.37)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสมการที่ (3.37) มีค่ามากที่สุดจะทำให้การแบ่งแยกระหว่างข้อมูล 2 กลุ่มทำได้ดีดังรูปที่ 3.7



รูปที่ 3.7 แสดงตัวอย่างการฉายภาพที่มีการแบ่งแยกได้ดีของตัวอย่าง 2 กลุ่ม

จากรูปที่ 3.7 จะเห็นว่าเมื่อใช้ค่าการกระจายภายในกลุ่มมาคิดคำนวณด้วยทำให้การฉายข้อมูลของ 2 กลุ่มแบ่งแยกกันอย่างชัดเจน โดยตัวอย่างที่ถูกฉายในกลุ่มเดียวกันจะรวมตัวอยู่ใกล้ ๆ กัน ขณะที่ยังคงไว้ซึ่งระยะห่างมาก ๆ ของค่าเฉลี่ยแต่ละกลุ่ม

เมื่อได้แนวคิดในการแบ่งแยกข้อมูลที่ดีแล้ว ลำดับต่อมาก็หาวิธีในการหาแกนการฉายแกนของตัวอย่างข้อมูล โดยสามารถหามาจากเมตริกการกระจายภายในกลุ่มและระหว่างกลุ่มของตัวอย่าง กำหนดให้เมตริกการกระจายภายในกลุ่มคือ

$$S_i = \sum_{t \in T_i} (t - \bar{t}_i)(t - \bar{t}_i)^T \qquad (3.38)$$

แต่ในกรณีนี้พิจารณาแค่ 2 กลุ่มจะได้เมตริกการกระจายภายในกลุ่มเป็น

$$S_W = S_1 + S_2 \qquad (3.39)$$

ความสัมพันธ์ของเมตริกการกระจายภายในกลุ่มของตัวอย่างที่ถูกฉายกับเมตริกการกระจายภายในกลุ่มของตัวอย่างคือ

$$\begin{aligned} \tilde{S}_i^2 &= \sum_{y \in \tilde{T}_i} (y - \tilde{m}_i)^2 \\ &= \sum_{t \in T_i} (w^T t - w^T \bar{t}_i)^2 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \sum_{t_k \in T_i} w^T (t_k - \bar{t}_i)(t_k - \bar{t}_i)^T w$$

$$= w^T S_i w \tag{3.40}$$

ดังนั้นผลรวมของเมตริการกระจายของข้อมูลที่ถูกฉาย

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_w w \tag{3.41}$$

ในวิธีเดียวกันผลต่างของค่าเฉลี่ยที่ถูกฉายสามารถแสดงในรูปผลต่างของค่าเฉลี่ยตัวอย่างได้คือ

$$\begin{aligned} |\tilde{m}_1 - \tilde{m}_2|^2 &= |w^T \bar{t}_1 - w^T \bar{t}_2|^2 \\ &= w^T (\bar{t}_1 - \bar{t}_2)(\bar{t}_1 - \bar{t}_2)^T w \\ &= w^T S_B w \end{aligned} \tag{3.42}$$

ซึ่ง S_w ถูกเรียกว่าเมตริกการกระภายในกลุ่มและ S_B เรียกว่าเมตริกการกระจายระหว่างกลุ่ม โดยที่เมตริกทั้งสองจะมีขนาดสมมาตร

ดังนั้นจากสมการที่ 3.37 สามารถแสดงสมการของฟิชเชอร์ในตัวแปรของ S_w และ S_B เป็น

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \tag{3.43}$$

ในการแก้สมการ (3.43) เพื่อหาแกนการฉาย โดยการใช้การหา Differenciate เทียบกับ w และกำหนดให้มีค่าเท่ากับศูนย์จะได้

$$S_B w = \lambda S_w w \tag{3.44}$$

เมื่อ S_w ไม่เป็นเอกฐาน (nonsingular) เราสามารถเปลี่ยนสมการใหม่เป็น

$$S_w^{-1} S_B w = \lambda w \tag{3.45}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในกรณีนี้พิจารณาจำนวนกลุ่มเพียง 2 กลุ่ม จึงไม่จำเป็นจะต้องแก้สมการที่ (3.45) โดยการหาค่าไอเกนเวกเตอร์และค่าไอของ $S_W^{-1} S_B$ เพราะในความเป็นจริง $S_B w$ จะมีทิศทางเดียวกับ $(\bar{t}_1 - \bar{t}_2)$ ดังนั้นการหาค่า w ที่เหมาะสมที่สุดในกรณีนี้คือ

$$w = S_W^{-1} (\bar{t}_1 - \bar{t}_2) \tag{3.46}$$

ดังนั้นจะได้แกนการฉาย w ที่มาจากการแบ่งแยกเชิงเส้นของฟิชเชอร์ สามารถคำนวณได้โดยการหาค่าอัตราส่วนที่มากที่สุดของการกระจายภายในกลุ่มและการกระจายระหว่างกลุ่ม จากการในทดลองใช้ข้อมูล 2 กลุ่มมาทำการฉายจะพบว่าข้อมูลของตัวอย่าง D มิติจะสามารถมิติลงเป็น 1 มิติได้

หลังจากทดลองใช้ข้อมูล 2 กลุ่มมาทดลองหาแกนการฉายที่เหมาะสมที่สุด ซึ่งจะแสดงให้เห็นความสัมพันธ์และวิธีในการหาแกนการฉาย ลำดับต่อมาจึงขยายจำนวนกลุ่มของตัวอย่างข้อมูลเป็นจำนวน L กลุ่ม เป็นผลทำให้สมการการแบ่งแยกเชิงเส้นของฟิชเชอร์จะประกอบไปด้วยฟังก์ชันการแบ่งแยกทั้งหมด L-1 ดังนั้นจะได้การฉายภาพจากปริภูมิ D มิติ ไปเป็นปริภูมิ L-1 มิติ ที่จำนวนมิติ $(L\text{-}1) < D$

จากการทดลองใช้ตัวอย่าง 2 กลุ่มในการลดมิติจะมีเมตริกที่ใช้ในการหาแกนการฉายอยู่สองเมตริกคือ เมตริการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่ม ดังนั้นในกรณีของ L ก็สามารถหาเมตริการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่มเป็น

เมตริกการกระจายภายในกลุ่มคือ

$$S_w = \sum_{i=1}^{L} S_i \tag{3.48}$$

เมื่อการกระจายของแต่ละกลุ่มและค่าเฉลี่ยของแต่ละกลุ่มเป็น

$$S_i = \sum_{t \in T_i} (t - \bar{t}_i)(t - \bar{t}_i)^T \quad \text{และ} \quad \bar{t}_i = \frac{1}{N_i} \sum_{t \in T_i} t \tag{3.49}$$

และสามารถหาเมตริการกระจายระหว่างกลุ่มได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S_B = \sum_{i=1}^{L} N_i (\bar{t}_i - \bar{t})(\bar{t}_i - \bar{t})^T \quad (3.50)$$

โดยที่ $\bar{t}$ คือค่าเฉลี่ยของตัวอย่างทั้งหมด $\bar{t} = \frac{1}{m}\sum_{i=1}^{m} t_i$ (3.51)

การฉายข้อมูลจากปริภูมิ D มิติ ไปเป็นปริภูมิ L-1 มิติด้วยวิธีการแบ่งแยกเชิงเส้นของฟิชเชอร์จะประกอบไปด้วยฟังก์ชันการแบ่งแยกทั้งหมด L-1

$$y_i = w_i^T t \quad i = 1,\ldots, L\text{-}1 \quad (3.52)$$

เปลี่ยนสมการให้อยู่ในรูปของเมตริกจะได้

$$Y = W^T t \quad (3.53)$$

เมื่อตัวอย่างในการฉายมีความสัมพันธ์กันกับตัวอย่างที่ถูกฉาย ดังนั้นจึงสามารถหาเวกเตอร์ค่าเฉลี่ยและเมตริกการกระจายได้ดังนี้

$$\tilde{m}_i = \frac{1}{N_i}\sum_{y \in \tilde{T}_i} y \quad (3.54)$$

$$\tilde{m} = \frac{1}{m}\sum_{i=1}^{L} N_i \tilde{m}_i \quad (3.55)$$

$$\tilde{S}_w = \sum_{i=1}^{L}\sum_{y \in \tilde{T}_i} (y - \tilde{m}_i)(y - \tilde{m}_i)^T \quad (3.56)$$

$$\tilde{S}_B = \sum_{i=1}^{L} N_i (\tilde{m}_i - \tilde{m})(\tilde{m}_i - \tilde{m})^T \quad (3.57)$$

จากแนวความคิดของตัวอย่างที่ถูกแบ่งเป็น 2 กลุ่มจะได้ความสัมพันธ์ สามารถหาเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่มของตัวอย่างที่ถูกฉายได้คือ

$$\tilde{S}_w = w^T S_w w \quad (3.58)$$

$$\tilde{S}_B = w^T S_B w \quad (3.59)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นความสัมพันธ์ระหว่างเมตริการกระจายภายในกลุ่มและเมตริการกระจายระหว่างกลุ่มที่ใช้ในการฉายข้อมูลให้มีปริภูมิมีขนาดลดลงสามารถทำได้โดย หาอัตราส่วนที่มากที่สุดของเมตริการกระจายภายในกลุ่มและเมตริการกระจายระหว่างกลุ่ม แต่เนื่องจากการวัดการกระจายต้องมีขนาดเป็นสเกลาจึงจำเป็นต้องใช้การหาดีเทอร์มิแนนซ์ (determinant) เข้ามาช่วย ดังนั้นสมการวัตถุประสงค์คือ

$$J(\mathrm{w}) = \frac{\left|\widetilde{\mathrm{S}}_{\mathrm{B}}\right|}{\left|\widetilde{\mathrm{S}}_{\mathrm{W}}\right|} = \frac{\left|\mathrm{w}^{\mathrm{T}}\mathrm{S}_{\mathrm{B}}\mathrm{w}\right|}{\left|\mathrm{w}^{\mathrm{T}}\mathrm{S}_{\mathrm{w}}\mathrm{w}\right|} \tag{3.60}$$

การแก้สมการ (3.60) จะใช้การหาไอเกเวกเตอร์ที่สมนัยกับค่าไอเกนที่เรียงจากมากไปน้อยตามลำดับของสมการที่ (3.61)

$$\mathrm{S}_{\mathrm{B}}\mathrm{W}_i = \lambda_i \mathrm{S}_{\mathrm{w}}\mathrm{W} \tag{3.61}$$

เมื่อเรียงลำดับค่าไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนจากมากไปน้อย แล้วเลือกเฉพาะไอเกนเวกเตอร์ที่สัมพันธ์กับค่าไอเกนที่ไม่เท่ากับศูนย์ ดังนั้นจะได้เมตริกใหม่ของไอเกนเวกเตอร์คือ

$$\mathrm{W}_{\mathrm{FLD}} = [\,\mathrm{w}_1\,\mathrm{w}_2\,\mathrm{w}_3 \ldots\, \mathrm{w}_{L\text{-}1}] \tag{3.62}$$

ดังนั้นจะได้การฉายภาพจากปริภูมิ D มิติ ไปเป็นปริภูมิ L-1 มิติ โดยหาอัตราส่วนที่มากที่สุดของเมตริการกระจายภายในกลุ่มและเมตริการกระจายระหว่างกลุ่ม

3.2.2 หลักการคำนวณของการวิเคราะห์การแบ่งแยกเชิงเส้นของฟิชเชอร์

วิธีการหาใบหน้าฟิชเชอร์ (Fisherface) ที่นำเสนอโดย P. N. Belhumeur J. P. Hespanha และ D. J. Kriegman เป็นวิธีที่ประยุกต์มาจากวิธี FLD ซึ่งนำมาใช้ในการรู้จำใบหน้า โดยการนำภาพใบหน้าแต่ละภาพมากำหนดเป็นจุดในปริภูมิภาพใบหน้า ปริภูมิภาพใบหน้าจะมีมิติเท่ากับจำนวนพิกเซลของภาพ เช่น ภาพระดับสีเทามีขนาดเท่ากับ m x n จะได้มิติของปริภูมิภาพคือ D ซึ่ง D จะเท่ากับ m x n แล้วทำการลดมิติไปเป็นปริภูมิที่มีมิติน้อยลงเพื่อหาฟีเจอร์ของภาพ

วิธีการหาใบหน้าฟิชเชอร์ ทำโดยคำนวณหาไอเกนเวกเตอร์จากเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่มของภาพที่ใช้สอน ที่สมนัยกับค่าไอเกนที่มีค่าความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปรปรวนจากมากไปน้อย เมื่อนำภาพที่ใช้สอนไปฉายด้วยไอเกนเวกเตอร์ที่ได้ จะได้ปริภูมิที่มีมิติน้อยลง

ในการทดสอบ จะนำภาพที่ใช้สอนทั้งหมดฉายด้วยวิธี Fisherface จะได้ปริภูมิใบหน้าที่ใช้สอน แล้วนำภาพที่ใช้ในการทดสอบมาฉายด้วยไอเกนเวกเตอร์ของภาพที่ใช้ในการสอน จะได้ปริภูมิใบหน้าที่ใช้ในการทดสอบ ในการแยกแยะภาพใบหน้าทำโดยหาระยะทางของปริภูมิที่ใช้สอนกับปริภูมิที่ใช้ทดสอบ

เมื่อกำหนดให้มีตัวอย่างภาพทั้งหมด M ภาพ ดังนั้น A_j (j = 1,2,...,M) แต่ละภาพมีขนาด m x n จุดภาพ โดยที่แต่ละภาพถูกเปลี่ยนให้เป็นเวกเตอร์ t เมื่อแต่ละ เวกเตอร์ t มีขนาด 1x D ($D = m$ x n) จัดแบ่งเป็นภาพทั้งหมดออกเป็น T_i (i = 1, 2,..., L) โดยที่ L คือจำนวนกลุ่ม $\bar{t}$ คือค่าเฉลี่ยที่คำนวณจากทุกภาพ $\bar{t}_i$ คือค่าเฉลี่ยที่คำนวณเฉพาะภาพภายในกลุ่ม T_i ซึ่งแต่ละกลุ่มมี N_i ภาพ การคำนวณกระทำได้ตามขั้นตอนดังนี้

คำนวณเมตริกการกระจายภายในกลุ่ม

เมตริกการกระภายในกลุ่มจะเป็นการวัดค่าการกระจายระหว่างภาพภายในแต่ละกลุ่ม เมื่อมีภาพทั้งหมด L กลุ่ม ดังนั้นการคำนวณจะใช้การหาผลรวมของเมตริกความแปรปรวน (Covariance matrix) ในแต่ละกลุ่มมาคำนวณหา

$$S_W = \sum_{i=1}^{L} \sum_{t \in T_i} (t - \bar{t}_i)(t - \bar{t}_i)^T \qquad (3.62)$$

คำนวณเมตริกการกระจายระหว่างกลุ่ม

เมตริกการกระจายระหว่างกลุ่มจะเป็นการวัดค่าการกระจายระหว่างกลุ่ม สามารถคำนวณหาโดยการหาผลรวมเมตริกความแปรปรวนของความแตกต่างระหว่างค่าเฉลี่ยรวมกับค่าเฉลี่ยของแต่ละกลุ่ม

$$S_B = \sum_{i=1}^{L} N_i (\bar{t}_i - \bar{t})(\bar{t}_i - \bar{t})^T \qquad (3.63)$$

คำนวณค่าไอเกนเวกเตอร์และค่าไอเกน

คำนวณหาค่าไอเกนเวกเตอร์และค่าไอเกนของเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่ม

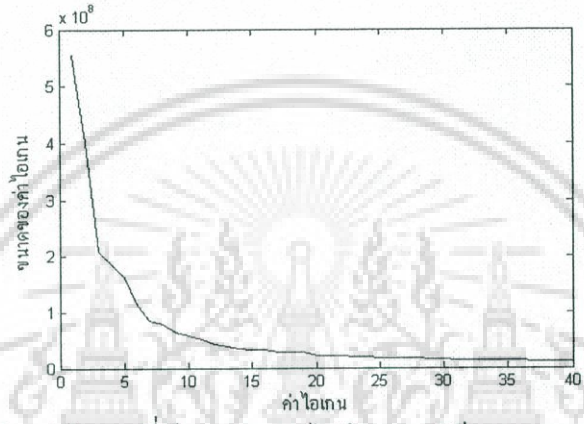
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S_B W = \lambda S_W W \tag{3.64}$$

เลือกค่าไอเกนเวกเตอร์

เรียงไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนจากมากไปน้อยและเลือกไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนที่ไม่เท่ากับศูนย์ ดังนั้นไอเกนเวกเตอร์จะมีทั้งหมด L-1

$$W_{FLD} = [w_1, w_2, \ldots, w_{L-1}] \tag{3.65}$$



รูปที่ 3.8 ตัวอย่างค่าไอเกนที่เรียงจากมากไปน้อยของวิธี FLD จากฐานข้อมูล ORL

3.2.3 การหาฟีเจอร์ของภาพ

เมื่อได้เวกเตอร์ที่เหมาะสมที่ใช้ในการฉายภาพของ FLD คือ $W_{FLD} = [w_1, w_2, \ldots, w_d]$ โดยที่ d เป็นค่าที่เลือกระหว่าง 1 ถึง L-1 และมีภาพที่ใช้สอนทั้งหมด M ภาพ จะสามารถหาฟีเจอร์ของภาพที่ใช้สอนทั้งหมดจากสมการ

$$Y_k = W_d^T (t_k - \bar{t}) ; k = 1, 2, \ldots, M \tag{3.65}$$

3.2.4 การสร้างภาพใหม่

วิธีของ FLD สามารถนำเอาฟีเจอร์ของภาพและแกนการฉายที่เหมาะสมที่สุดของ FLD มาสร้างเป็นภาพใบหน้าใหม่ได้ตามขั้นตอนดังนี้

โดยที่ $y = [y_1, \ldots, y_d]$ เป็นฟีเจอร์ของภาพ และ $U_d = [u_1, u_2, \ldots, u_d]$ เป็นแกนการฉายที่เหมาะสมที่สุด ดังนั้นจาก $y = W_d^T (t - \bar{t})$ และ W_d เป็นออโธนอมอล($W^{-1} = W^T$) จะได้สมการการสร้างภาพใหม่คือ

$$\tilde{t} = W_d y + \bar{t} \tag{3.66}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$d = 9$ $d = 19$ $d = 29$ $d = 39$

รูปที่ 3.9 ตัวอย่างภาพใบหน้าบุคคลหนึ่งที่สร้างใหม่ด้วยวิธี Fisherface จากฐานข้อมูล ORL

3.3 Two – Dimensional Principal Component Analysis (2DPCA)

เมื่อนำหลักการของวิธี PCA มาใช้ในการรู้จำใบหน้าที่เป็นภาพสองมิติ จะเห็นว่า ก่อนการคำนวณจำเป็นต้องเปลี่ยนจากเมตริกภาพให้เป็นเวกเตอร์ภาพทำให้มิติมีค่ามาก เมื่อนำมาหาค่าเมตริกการกระจายรวมเป็นผลให้เมตริกมีขนาดใหญ่ ทำให้การคำนวณหาค่าไอเกนเวกเตอร์ที่ใช้ในการฉายจึงทำได้ยากและใช้เวลาในการคำนวณมาก

เทคนิคใหม่ที่ถูกนำเสนอโดย Yang สามารถนำภาพที่ใช้สอนมาคำนวณได้โดยตรง วิธีการนี้มีชื่อว่า Two – Dimensional Principal Component Analysis (2DPCA) โดยที่ความแตกต่างระหว่าง PCA และ 2DPCA คือ สมการการแปลงเชิงเส้น ซึ่งวิธี 2DPCA จะใช้สมการ $Y = AX$ โดยที่ X คือแกนการฉาย และ A เป็นเมตริกภาพ ขณะที่ PCA จะใช้สมการ $Y = U^T t$ โดยที่ U คือแกนการฉาย และ t เป็นเวกเตอร์ภาพ เป็นผลให้ภาพที่ใช้ในการสอนไม่จำเป็นต้องเปลี่ยนจากเมตริกภาพเป็นเวกเตอร์ภาพและในการคำนวณหาค่าเมตริกการกระจายรวมของภาพสามารถนำภาพที่ใช้สอนมาคำนวณหาได้ตรง ซึ่งเป็นผลทำให้ขนาดของเมตริกการกระจายรวมของภาพที่ใช้วิธี 2DPCA มีขนาดน้อยกว่า วิธี PCA จึงทำให้เวลาในการคำนวณหาค่าของไอเกนเวกเตอร์มีค่าน้อยมาก หลักการคำนวณของวิธี 2DPCA คือการคำนวณหาเมตริกความแปรปรวนร่วมจากภาพที่ใช้ในการสอนทั้งหมด แล้วคำนวณหาค่าไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนที่เรียงลำดับจากมากไปน้อย แล้วจึงนำมาฉายภาพใบหน้าให้มีขนาดลดลง เช่น ภาพใบหน้ามีขนาด 92 x 112 จุดภาพ เมื่อเลือกไอเกนเวกเตอร์ 12 มิติ ฟีเจอร์ของภาพใบหน้าใหม่จะมีขนาดเป็น 92 x 12 จุดภาพ ซึ่งจะเห็นว่าจำนวนมิติของคอลัมน์มีค่าลดลง

3.3.1 แนวความคิดของวิธี 2DPCA

กำหนดให้ X ประกอบด้วยเวกเตอร์ขนาด n มิติ และ A คือภาพที่มีขนาด $m \times n$ จากการฉายภาพ A ลงบนแกน X ตามหลักการแปลงเชิงเส้น จะได้

$$Y = AX \tag{3.67}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะได้ Y ประกอบด้วยเวกเตอร์ขนาด m มิติ โดยแต่ละเวกเตอร์ของ Y คือฟีเจอร์เวกเตอร์ของภาพ A

วิธีในการหาเวกเตอร์ X ที่ใช้ในการฉายที่ดีนั้น สามารถคำนวณจากการหาผลรวมค่าไอเกน (Trace) ของเมตริกความแปรปรวนร่วมของฟีเจอร์เวกเตอร์ Y จากแนวความคิดนี้สามารถสร้างเป็นหลักการได้ตามสมการ

$$J(x) = tr\,(S_x) \tag{3.68}$$

เมื่อ S_x คือเมตริกความแปรปรวนร่วมของฟีเจอร์เวกเตอร์ Y และ $tr\,(S_x)$ คือการหาผลรวมค่าไอเกนของเมตริก S_x

ลักษณะที่สำคัญของการหาค่ามากที่สุด (Maximize) ในสมการที่ (3.68) ก็เพื่อหาทิศทางการฉาย X ที่มีความแปรปรวนมากที่สุด โดยที่เมตริกความแปรปรวนร่วม (S_x) สามารถหาจาก

$$\begin{aligned} S_x &= E(Y - E(Y))\,(Y - E(Y))^T \\ &= [AX - E(AX)]\,[AX - E(AX)]^T \\ &= E[(A - E(A))X]\,[(A - E(A))X]^T \end{aligned} \tag{3.69}$$

ถ้า

$$tr(S_x) = X^T[E(A - E(A))^T(A - E(A))]X \tag{3.70}$$

เมื่อกำหนดให้

$$G = E[(A - EA)^T(A - EA)] \tag{3.71}$$

เมตริก G คือ เมตริกความแปรวนร่วมของภาพ (เมตริกการกระจายของภาพ) ที่มีขนาด n x n

เมื่อกำหนดให้มีภาพตัวอย่างที่ใช้สอนทั้งหมด M ภาพ เป็นเมตริกที่มีขนาด m x n ดังนั้น $A_j (j = 1, 2, \ldots, M)$ และค่าเฉลี่ยภาพของภาพทั้งหมดคือ $\bar{A}$ การคำนวณหาเมตริกความแปรวนร่วมของภาพ สามารถนำภาพมาคำนวณได้โดยตรงจาก

$$G = \frac{1}{M}\sum_{j=1}^{M} [(A_j - \bar{A})^T (A_j - \bar{A})] \tag{3.72}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

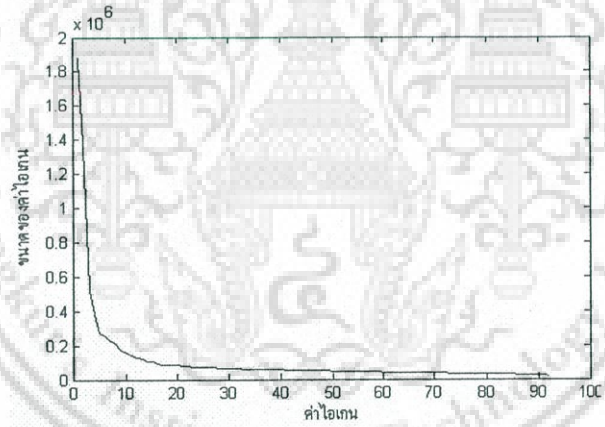
จากสมการที่ (3.68) สามารถแทนค่าสมการที่ (3.70) และสมการที่ (3.72) เป็น

$$J(X) = X^T G X \tag{3.73}$$

สมการที่ (3.73) มีชื่อว่า สมการทั่วไปของผลรวมการกระจาย (generalized total scatter criterion) แกน X ที่ได้จากสมการนี้ที่มีค่ามากที่สุดมีชื่อว่า แกนที่เหมาะสมที่สุดในการฉาย (optimal projection axis) ในการหาค่ามากที่สุดนั้นก็เพื่อต้องการให้การฉายภาพลงบนแกน X แล้วเมตริกใหม่ที่ถูกฉายมีการกระจายตัวมากที่สุด

แกนที่เหมาะสมที่สุดที่ใช้ในการฉาย X_{opt} สามารถหาได้จากการหาค่าไอเกนเวกเตอร์ ของ G ที่สมนัยกับค่าที่มากที่สุดของค่าไอเกนค่าไอเกนที่มากที่สุดตั้งแต่ 1 ถึง n ซึ่งแต่ละไอเกนเวกเตอร์เป็นออโธนอมอล (Orthonomal)

$$X_{2DPCA} = [X_1\ X_2\ \ldots\ X_n] \tag{3.74}$$



รูปที่ 3.10 ตัวอย่างค่าไอเกนที่เรียงจากมากไปน้อยของวิธี 2DPCA จากฐานข้อมูล ORL

3.3.2 การหาฟีเจอร์ของภาพ

เวกเตอร์ที่เหมาะสมที่ที่ใช้ในการฉายภาพของ 2DPCA คือ $X_1, X_2, \ldots, X_n$ ถูกนำมาใช้หาฟีเจอร์ของภาพ แต่เนื่องต้องการหาฟีเจอร์ของภาพฉะนั้นจึงเลือกแกนในการฉายเพียง 1 ถึง d เท่านั้น

เมื่อกำหนดให้ A เป็นภาพตัวอย่างดังนั้นการหาฟีเจอร์ของภาพ A คือ

$$Y_k = AX_k,\ k = 1, 2, \ldots, d \tag{3.75}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวกเตอร์ใหม่ที่ได้จากการฉาย $Y_1, Y_2,\ldots,Y_d$ มีชื่อว่า เวกเตอร์องค์ประกอบหลัก (principal component vector) ของภาพตัวอย่าง A เมื่อนำเวกเตอร์ขององค์ประกอบหลักรวมเป็นเมตริก $Y = [Y_1, Y_2,\ldots, Y_d]$ ที่มีขนาดเป็น $m \times d$ เมตริก Y มีชื่อว่า ฟีเจอร์เมตริกภาพ A

3.3.3 การสร้างภาพใหม่

ในวิธีของ 2DPCA สามารถนำเอาฟีเจอร์เมตริกและแกนการฉายที่เหมาะสมที่สุดมาสร้างเป็นภาพใบหน้าใหม่ได้ตามขั้นตอนดังนี้

กำหนดให้ A เป็นภาพ โดยที่ $Y = [y_1,\ldots,y_d]$ เป็นฟีเจอร์เมตริกของภาพ A และ $X = [x_1,\ldots,x_d]$ เป็นแกนการฉายที่เหมาะสมที่สุด ดังนั้นจาก $Y_k = AX_k$ และ X เป็นออโธนอมอล ($X^{-1} = X^T$) จะได้สมการการสร้างภาพใหม่คือ

$$\tilde{A} = YX^T = \sum_{k=1}^{d} y_k x_k^T \qquad (3.76)$$

ในการสร้างภาพใบหน้าใหม่นั้นถ้าเลือกจำนวนแกนการฉาย d เท่ากับ n จะได้ภาพที่สร้างใหม่เหมือนกับภาพ A แต่ถ้าเลือก d น้อยกว่า n ภาพ $\tilde{A}$ จะความเหมือนกับภาพ A น้อยลงตามลำดับ

d =2 d=6 d= 10 d=14

รูปที่ 3.11 ตัวอย่างภาพใบหน้าที่สร้างใหม่ด้วยวิธี 2DPCA จากฐานข้อมูล ORL

3.3.4 หลักการคำนวณของวิธี 2DPCA

เมื่อกำหนดให้มีภาพที่ใช้สอน M ภาพ คือ $A_1, A_2,\ldots, A_M$ แต่ละภาพมีขนาด $m \times n$ จุดภาพ

คำนวณหาค่าเฉลี่ยของภาพใบหน้า (Mean Face)

$$\bar{t} = \frac{1}{M}\sum_{i=1}^{M} t_i$$

คำนวณหาเมตริกความแปรปรวนร่วม (Covariance Matrix)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G = \frac{1}{M}\sum_{j=1}^{M}[(A_j - \overline{A})^T (A_j - \overline{A})]$$

คำนวณหาคำนวณหาค่าไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกน จากเมตริกความแปรปรวนร่วมด้วยสมการ

$$GX = \lambda X$$

เรียงลำดับค่าไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนจากมากไปน้อย แล้วเลือกแกนการฉายมา d มิติดังนั้นจะได้เมตริกใหม่ของไอเกนเวกเตอร์ คือ

$$X = [x_1 x_2 x_3 \ldots x_d]$$

การหาฟีเจอร์ของภาพ

$$Y_k = AX_k, k = 1, 2, \ldots, d$$

การสร้างภาพใหม่

$$\tilde{A} = YX^T_A = \sum_{k=1}^{d} y_k x_k^T$$

3.4 Two – Dimensional Linear Discriminant Analysis (2DLDA)

วิธีการ 2DLDA เป็นวิธีการใหม่ที่ประยุกต์มาจากวิธี FLD และ 2DPCA โดยสามารถนำเมตริกภาพมาคำนวณหาเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่ม แล้วนำมาคำนวณหาไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนจากมากไปน้อยตามลำดับ จากนั้นนำภาพใบหน้ามาฉายลงบนไอเกนเวกเตอร์ จะได้ฟีเจอร์เมตริกที่มีจำนวนมิติของคอลัมน์ภาพน้อยลง

3.4.1 แนวความคิดของวิธี 2DLDA

กำหนดให้ X ประกอบด้วยเวกเตอร์ขนาด n มิติ และ A เป็นภาพที่มีขนาด $m \times n$ จากการฉายภาพ A ลงบนแกน X ตามหลักการแปลงเชิงเส้นจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Y = AX \tag{3.77}$$

ดังนั้นจะได้ Y ประกอบด้วยเวกเตอร์ขนาด m มิติ โดยแต่ละเวกเตอร์ของ Y คือฟีเจอร์เวกเตอร์ของภาพ A

เมื่อกำหนดให้มีตัวอย่างภาพทั้งหมด M ภาพ A_j ($j = 1,2,\ldots,M$) จัดแบ่งเป็นกลุ่ม T_i ($i = 1, 2,\ldots, L$) โดยที่ L คือจำนวนกลุ่ม $\bar{A}$ คือค่าเฉลี่ยที่คำนวณจากทุกภาพ $\bar{A}_i$ คือค่าเฉลี่ยที่คำนวณเฉพาะภาพภายในกลุ่ม T_i ซึ่งแต่ละกลุ่มมี N_i ภาพ และ P_i คือกลุ่ม T_i ที่ถูกฉายลงบนแกน X

หลังจากภาพที่ใช้ฝึกสอนถูกฉายตามสมการที่ (3.77) ลงบนแกน X จะได้ฟีเจอร์เมตริกเป็น

$$Y_j = A_j X, \; j = 1, 2, \ldots, M \tag{3.78}$$

วิธีในการหาแกนการฉาย X สามารถคำนวณได้โดยการหาผลรวมค่าไอเกน (Trace) ของเมตริกความแปรปรวนร่วมของภาพตัวอย่างที่ถูกฉาย จากแนวความคิดนี้สามารถสร้างเป็นสมการได้คือ

$$J(X) = \frac{P_B}{P_W} \tag{3.79}$$

เมื่อ P_B และ P_W คำนวณจาก

$P_B = tr(\tilde{S}_B)$

$P_W = tr(\tilde{S}_W)$

โดยที่

$\tilde{S}_B$ คือ เมตริกการกระจายระหว่างกลุ่มของภาพตัวอย่างที่ถูกฉาย

$\tilde{S}_W$ คือ เมตริกการกระจายภายในกลุ่มของภาพตัวอย่างที่ถูกฉาย

$$\begin{aligned}\tilde{S}_B &= \sum_{i=1}^{L} N_i (\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \\ &= \sum_{i=1}^{L} N_i [(\bar{A}_i - \bar{A})X][(\bar{A}_i - \bar{A})X]^T \end{aligned} \tag{3.80}$$

$$\tilde{S}_W = \sum_{i=1}^{L} \sum_{Y_k \in P_i} (Y_k - \bar{Y}_i)(Y_k - \bar{Y}_i)^T$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \sum_{i=1}^{L} \sum_{Y_k \in P_i} [(A_k - \overline{A}_i)X][(A_k - \overline{A}_i)X]^T \quad (3.81)$$

ถ้า

$$tr(\tilde{S}_B) = X^T[\sum_{i=1}^{L} N_i (\overline{A}_i - \overline{A})^T (\overline{A}_i - \overline{A})]X$$
$$= X^T S_B X \quad (3.82)$$

$$tr(\tilde{S}_W) = X^T[\sum_{i=1}^{L} \sum_{Y_k \in P_i} (A_k - \overline{A}_i)^T (A_k - \overline{A}_i)]X$$
$$= X^T S_W X \quad (3.83)$$

ในการคำนวณหา S_B และ S_W สามารถนำภาพที่ใช้ฝึกสอนมาคำนวณหาได้โดยตรง และจากสมการที่ (3.79) สามารถเปลี่ยนเป็นสมการใหม่ได้คือ

$$J(X) = \frac{X^T S_B X}{X^T S_W X} \quad (3.84)$$

สมการที่ (3.84) มีชื่อว่าสมการหลักการฉายเชิงเส้นที่ยึดถือประเภทเป็นเกณฑ์ (class specific linear projection criterion) โดยเวกเตอร์ X ที่มาจากการหาค่ามากที่สุดของ $J(X)$ มีชื่อว่าแกนการฉายที่เหมาะสมที่สุด

$$X_{opt} = \arg\max_{X} J(X) \quad (3.85)$$

ถ้า S_W ไม่เป็นเมตริกเอกฐาน (non singular) สามารถหาเมตริก X โดยการคำนวณไอเกนเวกเตอร์และค่าไอเกนได้จากสมการ

$$S_B X_{opt} = \lambda S_W X_{opt} \quad (3.86)$$

โดยปกติแกนการฉายที่เหมาะสมที่สุดที่ใช้ในการฉายมาจากไอเกนเวกเตอร์ของ $S_W^{-1} S_B$ ที่สัมพันธ์กับค่าไอเกนที่มากที่สุดตั้งแต่ 1 ถึง n ซึ่งแต่ละไอเกนเวกเตอร์เป็นออโธนอมอล(Orthonomal)

$$X_{2DLDA} = [x_1 x_2 \ldots x_n] \quad (3.87)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การหาฟีเจอร์ของภาพ

เวกเตอร์ที่เหมาะสมที่ที่ใช้ในการฉายภาพของ 2DLDA คือ $X_1, X_2,\ldots,X_n$ ถูกนำมาใช้หาฟีเจอร์ของภาพ แต่เนื่องต้องการหาฟีเจอร์ของภาพฉะนั้นจึงเลือกแกนในการฉายเพียง 1 ถึง d เท่านั้น

$$y_k = Ax_k, k = 1, 2,\ldots, d \tag{3.88}$$

เวกเตอร์ใหม่ที่ได้จากการฉาย $y_1,\ldots,y_d$ มีชื่อว่าฟีเจอร์เวกเตอร์ เมื่อนำฟีเจอร์เวกเตอร์รวมเป็นเมตริก $Y = [y_1, \ldots,y_d]$ ที่มีขนาดเป็น m x d เมตริก Y ที่ได้มีชื่อว่า ฟีเจอร์เมตริกของภาพ A

3.4.3 การสร้างภาพใหม่

ในวิธีของ 2DLDA สามารถนำเอาฟีเจอร์เมตริกและแกนการฉายที่เหมาะสมที่สุดมาสร้างเป็นภาพใบหน้าใหม่ได้โดย กำหนดให้ A เป็นภาพ โดยที่ $Y = [y_1,\ldots,y_d]$ เป็นฟีเจอร์เมตริกของภาพ A และ $X_{2DLDA} = [x_1,\ldots,x_d]$ เป็นแกนการฉายที่เหมาะสมที่สุด จากสมาการ $Y= AX$ และ X เป็นออโธนอมอล($X^{-1}= X^T$) ดังนั้นสมการในการสร้างภาพใหม่คือ

$$\tilde{A} = YX^T = \sum_{k=1}^{d} y_k x_k^T \tag{3.89}$$

ในการสร้างภาพใบหน้าใหม่นั้นถ้าเลือกจำนวนแกนการฉาย d เท่ากับ n จะได้ภาพที่สร้างใหม่เหมือนกับภาพ A แต่ถ้าเลือก d น้อยกว่า n ภาพ $\tilde{A}$ จะความเหมือนกับภาพ A น้อยลงตามลำดับ

3.3.4 หลักการคำนวณของวิธี 2DLDA

เมื่อกำหนดให้มีตัวอย่างภาพทั้งหมด M ภาพ A_j (j = 1,2,…,M) จัดแบ่งเป็นกลุ่ม T_i ($i = 1, 2, \ldots, L$) โดยที่ L คือจำนวนกลุ่ม

คำนวณเมตริกการกระจายภายในกลุ่ม

เมตริกการกระภายในกลุ่มจะเป็นการวัดค่าการกระจายระหว่างภาพภายในแต่ละกลุ่ม เมื่อมีภาพทั้งหมด L กลุ่ม ดังนั้นการคำนวณจะใช้การหาผลรวมของเมตริกความแปรปรวน (Covariance matrix) ในแต่ละกลุ่มมาคำนวณหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S_W = \sum_{i=1}^{L} N_i \, (\overline{A}_i - \overline{A})^T (\overline{A}_i - \overline{A})$$

คำนวณเมตริกการกระจายระหว่างกลุ่ม

เมตริกการกระจายระหว่างกลุ่มจะเป็นการวัดค่าการกระจายระหว่างกลุ่ม สามารถคำนวณหาโดยการหาผลรวมเมตริกความแปรปรวนของความแตกต่างระหว่างค่าเฉลี่ยรวมกับค่าเฉลี่ยของแต่ละกลุ่ม

$$S_B = \sum_{i=1}^{L} \sum_{Y_k \in P_i} (A_k - \overline{A}_i)^T (A_k - \overline{A}_i)$$

คำนวณค่าไอเกนเวกเตอร์และค่าไอเกน

คำนวณหาค่าไอเกนเวกเตอร์และค่าไอเกนของเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่ม

$$S_B X_{opt} = \lambda S_W X_{opt}$$

เลือกค่าไอเกนเวกเตอร์

เรียงไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนจากมากไปน้อย แล้วเลือกแกนในการฉาย 1 ถึง d โดย $d \leq n$

$$X = [x_1, x_2, \ldots, x_d]$$

การหาฟีเจอร์ของภาพ

$$Y_k = AX_k, \; k = 1, 2, \ldots, d$$

การสร้างภาพใหม่

$$\tilde{A} = YX^T_A = \sum_{k=1}^{d} y_k \, x_k^T$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วิธีการค้นหาภาพใบหน้า

วิธีการจำแนกข้อมูลที่ใช้ในวิทยานิพนธ์นี้มี 2 วิธี คือวิธีการหาระยะทางยูคลิเดียน (Euclidean distance) ที่ใช้กับเวกเตอร์ และประยุกต์วิธีการหาระยะทางยูคลิเดียนที่ใช้กับเมตริก

เมื่อคำนวณหาฟีเจอร์เวกเตอร์ของภาพด้วยวิธี PCA และ FLD แล้ว การจำแนกข้อมูลจะใช้การคำนวณหาระยะทางยูคลิเดียน โดยกำหนดให้มีเวกเตอร์ภาพ t_1 และ t_2 ถูกนำมาหาฟีเจอร์ด้วยวิธี PCA และ FLD จะได้ เวกเตอร์ y_1 และ y_2 ถ้าต้องการหาความเหมือนกันของที่ถูกฉายทั้งสอง ทำได้โดยการหาระยะทาง

$$(y_1, y_2) = \left\| y_1 - y_2 \right\|_2 \tag{3.90}$$

เมื่อ $\left\| y_1 - y_2 \right\|_2$ คือ ระยะทางยูคลิเดียนระหว่างสองฟีเจอร์เวกเตอร์

ถ้าฟีเจอร์เวกเตอร์ของภาพตัวอย่างที่ใช้ฝึกสอนทั้งหมดคือ $y_1, y_2, \ldots, y_M$ (M คือตัวอย่างภาพที่ใช้ฝึกสอนทั้งหมด) และแต่ละภาพเป็นส่วนหนึ่งของกลุ่ม T_i ดังนั้นในการทดสอบภาพตัวอย่าง y จะใช้หลักการหาระยะทางยูคลิเดียนกับภาพทุกภาพ ถ้าภาพใดมีค่าระยะทางที่สั้นที่สุดกับภาพตัวอย่าง และภาพนั้นอยู่ในกลุ่มใด ก็กำหนดให้ภาพ y เป็นกลุ่มนั้นด้วย

เมื่อคำนวณหาฟีเจอร์เมตริกของภาพด้วยวิธี 2DPCA และ 2DLDA แล้ว การจำแนกข้อมูลจะประยุกต์การคำนวณหาระยะทางยูคลิเดียนเพื่อใช้กับเมตริกโดย กำหนดให้ภาพใบหน้า A_1 และ A_2 ถูกนำมาหาฟีเจอร์เมตริกเป็น $Y_1 = [y_1^1, \ldots, y_1^d]$ และ $Y_2 = [y_2^1, \ldots, y_2^d]$ ถ้าต้องการหาความเหมือนกันของภาพที่ถูกฉายทั้งสอง ทำได้โดยการหาระยะทาง

$$d(Y_1, Y_2) = \sum_{k=1}^{d} \left\| y_1^k - y_2^k \right\|_2 \tag{3.91}$$

เมื่อ $\left\| y_1^k - y_2^k \right\|_2$ คือ ระยะทางยูคลิเดียนระหว่างสองฟีเจอร์เวกเตอร์

ถ้าฟีเจอร์เมตริกของภาพตัวอย่างที่ใช้ฝึกสอนทั้งหมดคือ $Y_1, Y_2, \ldots, Y_M$ (M คือตัวอย่างภาพที่ใช้ฝึกสอนทั้งหมด) และแต่ละภาพเป็นส่วนหนึ่งของกลุ่ม T_i ดังนั้นในการทดสอบภาพตัวอย่าง Y จะใช้หลักการหาระยะทางยูคลิเดียนกับภาพทุกภาพ ถ้าภาพใดมีค่าระยะทางที่สั้นที่สุดกับภาพตัวอย่า และภาพนั้นอยู่ในกลุ่มใด ก็กำหนดให้ภาพ Y เป็นกลุ่มนั้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

ในบทนี้ได้ทำการทดลองเปรียบเทียบความถูกต้องในการรู้จำระหว่างวิธี PCA วิธี LDA วิธี 2DPCA และวิธี 2DLDA ภายใต้เงื่อนไขต่าง ๆ ที่ทำให้การรู้จำใบหน้ามีความผิดพลาด ซึ่งในการรู้จำใบหน้านั้น ได้ทดลองกับฐานข้อมูลใบหน้า 3 แห่ง ได้แก่ ฐานข้อมูลใบหน้า ORL [28] ฐานข้อมูลใบหน้า AR [29], [30] และฐานข้อมูลใบหน้า Yale [31] โดยที่ฐานข้อมูลใบหน้า ORL ใช้ทดสอบความถูกต้องในการรู้จำใบหน้าภายใต้เงื่อนไขของการเปลี่ยนแปลงตำแหน่งใบหน้าและจำนวนตัวอย่างในการเรียนรู้ ส่วนฐานข้อมูลใบหน้า AR ใช้ทดสอบความถูกต้องในการรู้จำใบหน้าภายใต้เงื่อนไขของระยะเวลาในการถ่ายภาพ การแสดงออกทางอารมณ์บนใบหน้า ความสว่างบนใบหน้า และฐานข้อมูลใบหน้า Yale ใช้ทดสอบความถูกต้องในการรู้จำใบหน้าภายใต้เงื่อนไขการเปลี่ยนแปลงของการแสดงออกบนใบหน้า ความสว่างบนใบหน้า และระยะทางที่ใช้ในการจำแนกบุคคล

4.1 การทดลองบนฐานข้อมูลใบหน้า ORL

ฐานข้อมูลภาพใบหน้าของ ORL ถูกสร้างขึ้นในช่วงเดือนเมษายน พ.ศ. 2538 ถึง เดือนเมษายน พ.ศ. 2540 ที่ Olivetti Research Laboratory ของมหาวิทยาลัย Cambridge ประเทศสหรัฐอเมริกา ภาพใบหน้าในฐานข้อมูลนี้ทั้งหมดจะเป็นภาพระดับสีเทาและมีพื้นหลังเป็นสีดำ ขนาด 112 x 92 จุดภาพ ถ่ายจากใบหน้าของ 40 บุคคล แบ่งเป็นผู้ชาย 36 คน และเป็นผู้หญิง 4 คน คนละ 10 ภาพที่แตกต่างกัน เช่น ภาพที่มีการแสดงออกทางอารมณ์บนใบหน้า (การปิดตาหรือเปิดตา ยิ้มหรือไม่ยิ้ม) ภาพที่มีลายละเอียดอื่น ๆ (การใส่แว่นหรือถอดแว่น) ภาพที่การก้มหน้าหรือเงยหน้า หันซ้ายหรือขวา ไม่เกินร้อยละ 20 โดยที่ รูปที่ 4.1 เป็นตัวอย่างภาพใบหน้าทั้ง 10 ภาพของบุคคลเดียว

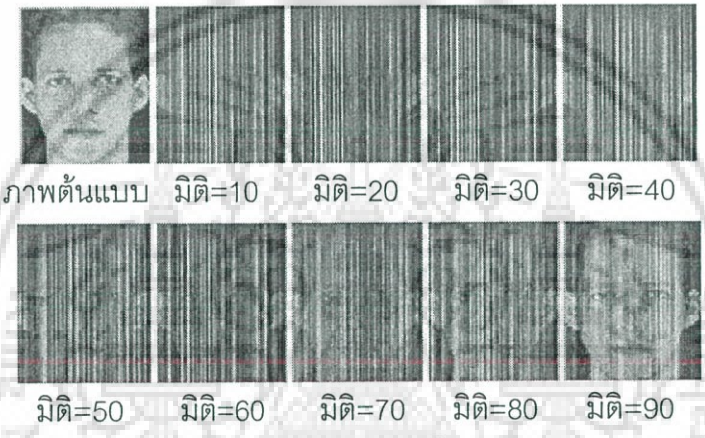
รูปที่ 4.1 ตัวอย่างภาพใบหน้า 1 บุคคลของฐานข้อมูล ORL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 การทดสอบการสร้างภาพใหม่ด้วยวิธี 2DLDA บนฐานข้อมูล ORL

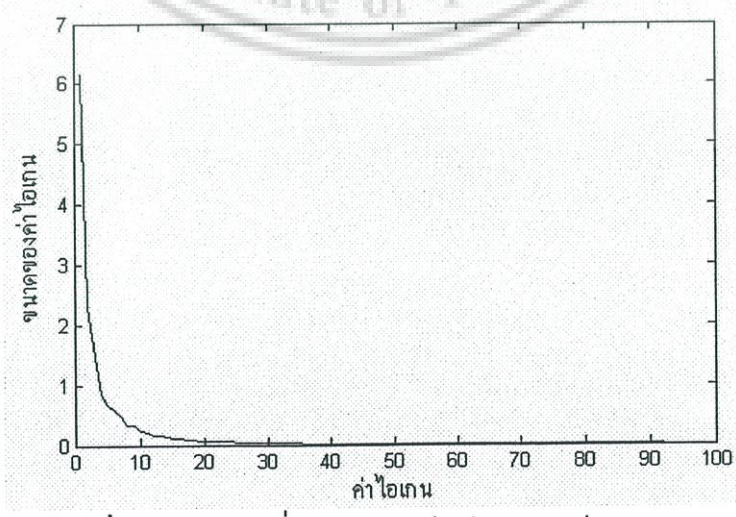
เนื่องจากทฤษฎีของวิธี Eigenface Fisherface และ 2DPCA นั้นมีคุณสมบัติพื้นฐานอย่างหนึ่งที่เหมือนกันคือ สามารถสร้างภาพใบหน้าใหม่ได้ ดังนั้นในการที่จะทดสอบว่าวิธี 2DLDA ที่ได้เสนอในวิทยานิพนธ์นี้มีความถูกต้องหรือไม่จึงทำการทดลองสร้างภาพใบหน้าใหม่

การทดสอบคุณสมบัติการสร้างภาพใหม่ของวิธี 2DLDA นั้น ได้ทำการสุ่มเลือกภาพที่ใช้ฝึกสอนจำนวน 5 ภาพต่อหนึ่งบุคคล เพื่อนำมาหาแกนการฉายที่เหมาะสมที่สุดด้วยวิธี 2DLDA เมื่อนำภาพแรกในรูปที่ 4.1 มาทดลองฉายลงบนแกนการฉายที่เหมาะสมที่สุดจะได้ฟีเจอร์เมตริกที่ใช้ในการสร้างภาพใหม่ โดยรูปที่ 4.2 ภาพแรกเป็นภาพต้นแบบและภาพลำดับต่อมาจะเป็นการสร้างภาพใหม่ตามจำนวนมิติที่เพิ่มขึ้นตามลำดับ



รูปที่ 4.2 ตัวอย่างภาพใบหน้าที่สร้างขึ้นใหม่ของบุคคลคนหนึ่ง

จากรูปที่ 4.2 พบว่าวิธี 2DLDA ก็มีคุณสมบัติการสร้างภาพใหม่เหมือนกับวิธี Fisherface โดยที่ภาพที่สร้างใหม่จะเหมือนกับภาพต้นแบบขึ้นอยู่กับจำนวนมิติของแกนในการฉาย ถ้าจำนวนมิติมากขึ้นภาพใบหน้าที่สร้างใหม่ก็จะมีความเหมือนมาก



รูปที่ 4.3 ค่าไอเกนที่เรียงจากมากไปน้อยของวิธี 2DLDA

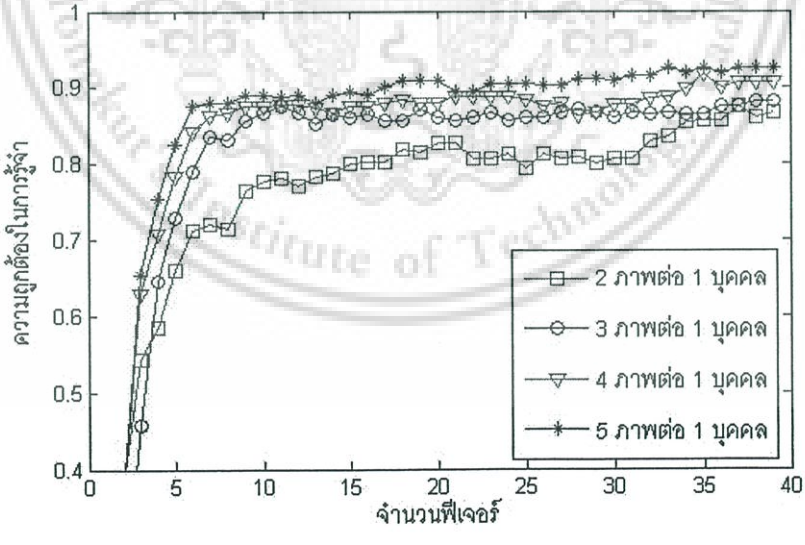
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 เป็นการนำค่าไอเกนมาสร้างกราฟของการสร้างภาพใบหน้าใหม่มาสร้างกราฟ โดยค่าไอเกนจะเรียงจากมากไปน้อย จากรูปที่ 4.3 จะเห็นว่าค่าเกนจะมีค่าเข้าใกล้ศูนย์ในจำนวนไอเกนน้อย ๆ ซึ่งแสดงให้เห็นว่าที่ฟีเจอร์น้อย ๆ วิธี 2DLDA จะสามารถคงข้อมูลที่เป็นประโยชน์ในการแยกแยะบุคคลไว้มาก

4.1.2 การทดลองเปรียบเทียบความถูกต้องในการรู้จำระหว่างวิธี 2DLDA กับวิธี Fisherface

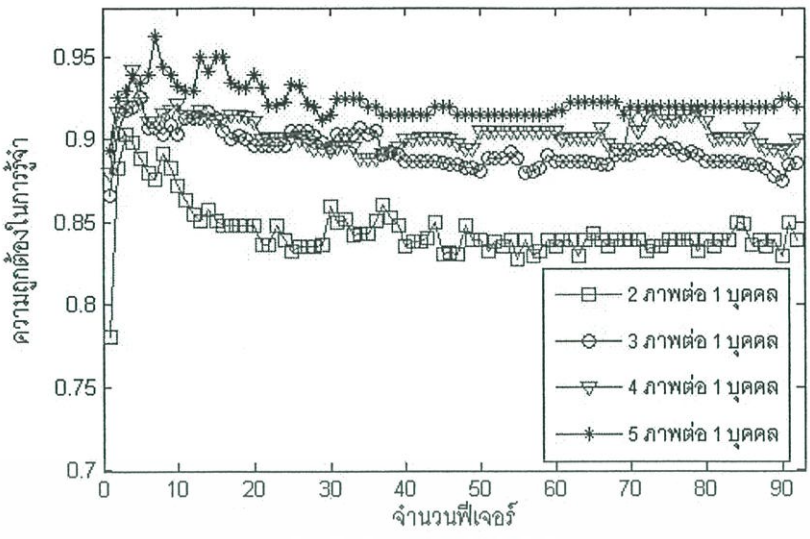
ในการทดลองที่สองทำการเปรียบเทียบความถูกต้องในการรู้จำระหว่างวิธี 2DLDA กับวิธี Fisherface ภายใต้เงื่อนไขของจำนวนตัวอย่างภาพใบหน้าที่ใช้ฝึกสอน โดยแบ่งการทดสอบเป็น 4 ครั้ง ครั้งแรกภาพใบหน้าที่ใช้ฝึกสอนจะถูกเลือกมาอย่างสุ่ม 2 ภาพต่อหนึ่งบุคคล และภาพที่เหลือถูกนำมาใช้ทดสอบความถูกต้อง การทดสอบครั้งต่อ ๆ มา ภาพที่ใช้ฝึกสอนจะถูกเลือกมาอย่างสุ่ม 3, 4 และ 5 ภาพต่อหนึ่งบุคคล ตามลำดับ

การทดลองของวิธี Fisherface ได้ทำการหาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 1 จนถึงฟีเจอร์มากที่สุดที่ 39 (L -1 โดยที่ L คือจำนวนบุคคล) ส่วนวิธี 2DLDA ได้หาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 112 x 1 จนถึงฟีเจอร์มากที่สุดที่ 112 x 92 เพื่อหาค่าร้อยละความถูกต้องในการรู้จำสูงสุดมาเปรียบเทียบระหว่างวิธี Fisherface กับวิธี 2DLDA ซึ่งผลการทดลองของวิธี 2DLDA และวิธี Fisherface แสดงในรูปที่ 4.4



รูปที่ 4.4 ร้อยละความถูกต้องในการรู้จำของวิธี Fisherface ภายใต้เงื่อนไขของจำนวนตัวอย่างภาพใบหน้าที่ใช้ฝึกสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA ภายใต้เงื่อนไขของจำนวนตัวอย่างภาพใบหน้าที่ใช้ฝึกสอน

จากรูปที่ 4.4 และรูปที่ 4.5 สามารถหาค่าร้อยละความถูกต้องในการรู้จำสูงสุดได้ดังตารางที่ 4.1 โดยที่ตารางที่ 4.1 เป็นการเปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดระหว่างวิธี Fisherface กับวิธี2DLDA ที่สัมพันธ์กับจำนวนตัวอย่างที่ใช้ฝึกสอน

ตารางที่ 4.1 เปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดระหว่างวิธี Fisherface กับวิธี 2DLDA ที่สัมพันธ์กับจำนวนตัวอย่างที่ใช้ฝึกสอน

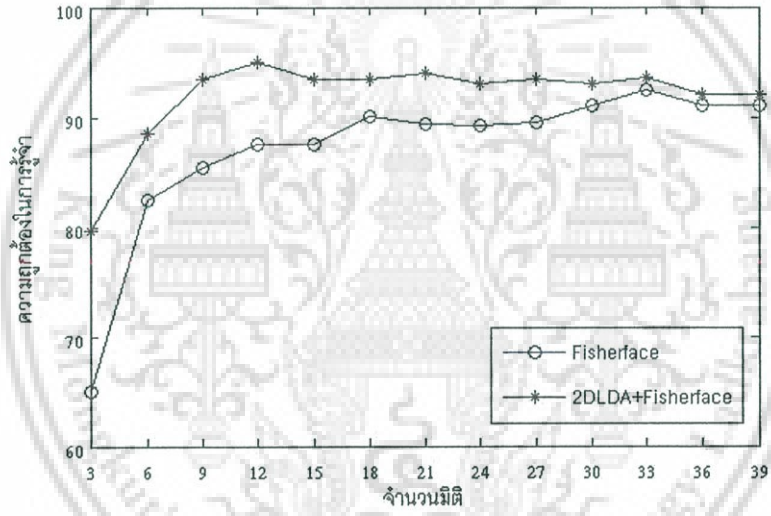
จำนวนภาพต่อบุคคลที่ใช้ในการฝึกสอน	ความถูกต้องในการรู้จำสูงสุดของวิธี Fisherface		ความถูกต้องในการรู้จำสูงสุดของวิธี 2DLDA	
	ร้อยละความถูกต้อง	ขนาดฟีเจอร์	ร้อยละความถูกต้อง	ขนาดฟีเจอร์
2	85.65	39	90.31	112 x 3
3	87.52	37	92.95	112 x 5
4	91.73	35	94.27	112 x 4
5	92.51	33	96.25	112 x 7

จากตารางที่ 4.1 จะพบว่าในการทดลองทั้งสี่ครั้ง ขนาดฟีเจอร์เมตริกที่ให้ค่าร้อยละความถูกต้องในการรู้จำสูงสุดของวิธี 2DLDA มีค่าไม่เกิน 112 x 10 ฟีเจอร์ ส่วนขนาดฟีเจอร์เวกเตอร์ที่ให้ค่าร้อยละความถูกต้องในการรู้จำสูงสุดของวิธี Fisherface มีค่าเกิน 30 ฟีเจอร์เวกเตอร์ ส่วนร้อยละความถูกต้องในการรู้จำสูงสุดของวิธี 2DLDA เมื่อเปรียบเทียบกับวิธี Fisherface จะพบว่าวิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2DLDA มีค่ามากกว่า วิธี Fisherface และถ้าจำนวนตัวอย่างที่ใช้ฝึกสอนมีเพิ่มขึ้นความถูกต้องในการรู้จำของทั้งสองวิธีมีค่ามากขึ้น

ในการเปรียบเทียบร้อยละความถูกต้องในการรู้จำระหว่างวิธี 2DLDA กับวิธี Fisherface ทำได้ยากเนื่องจากขนาดของฟีเจอร์เมตริกมีขนาดต่างกัน โดยที่ 2DLDA มีขนาดฟีเจอร์เมตริกเป็น $112 \times d$ ซึ่งมีขนาดมากกว่าฟีเจอร์เมตริกของ Fisherface มาก ดังนั้นในการเปรียบเทียบจึงจำเป็นต้องลดขนาดฟีเจอร์เมตริกของ 2DLDA โดยใช้ วิธี Fisherface มาคำนวณร่วมกับวิธี 2DLDA วิธีการนี้ถูกเรียกว่า 2DLDA + Fisherface ในการทดสอบความถูกต้องในการรู้จำของวิธี 2DLDA + Fisherface ภาพที่ใช้ฝึกสอนถูกเลือกมาอย่างสุ่มจำนวน 5 ภาพต่อหนึ่งบุคคลและภาพที่เหลือถูกนำมาใช้สำหรับทดสอบ เมื่อนำภาพที่ใช้ฝึกสอนมาคำนวณด้วยวิธี 2DLDA เพื่อหาฟีเจอร์เมตริกที่กำหนดให้มีขนาดเป็น 112×7 แล้วจึงใช้วิธี Fisherface มาคำนวณหาฟีเจอร์เป็นลำดับสุดท้าย



รูปที่ 4. 6 เปรียบเทียบความถูกต้องในการรู้จำระหว่างวิธี Fisherface กับวิธี 2DLDA+ Fisherface

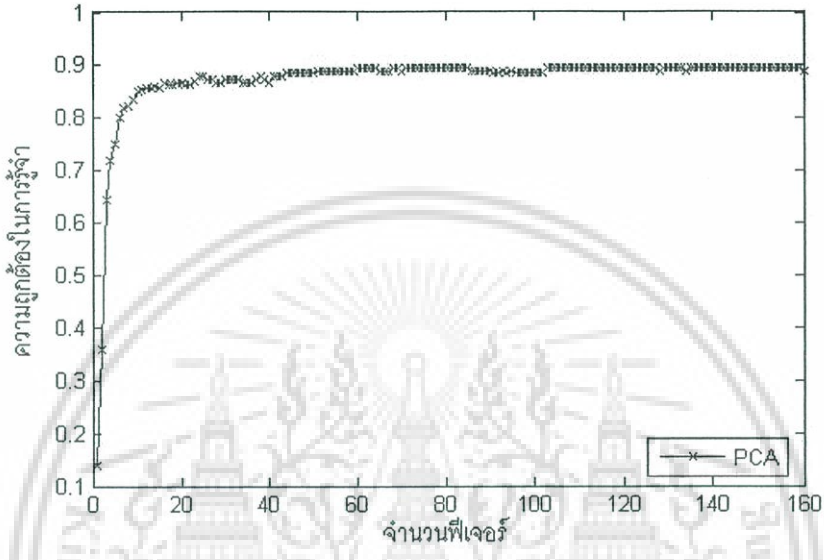
จากรูปที่ 4.6 พบว่ากระบวนการของ 2DLDA + Fisherface ยังคงให้ค่าความถูกต้องในการรู้จำดีกว่าวิธี Fisherface ในฟีเจอร์เดียวกัน

4.1.3 การทดลองเปรียบเทียบกระบวนการของ 2DLDA กับวิธีอื่น ๆ

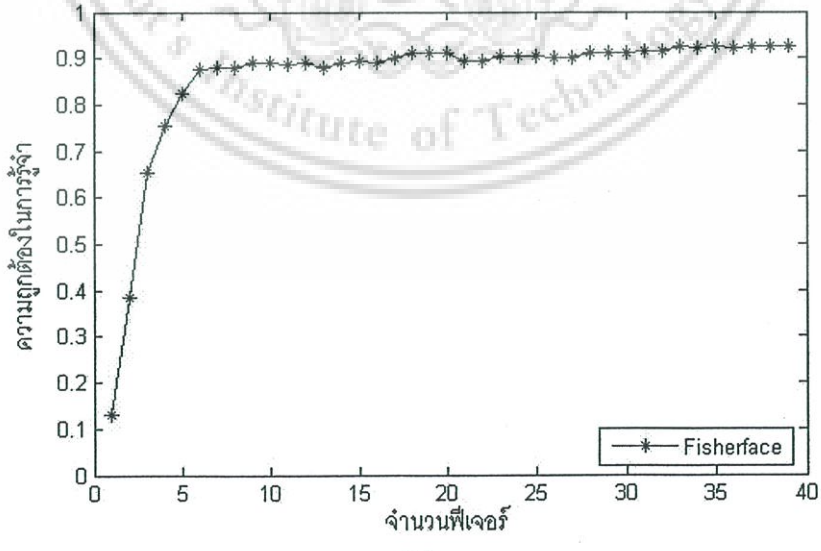
การทดลองที่สามทำการเปรียบเทียบกระบวนการของ 2DLDA, 2DPCA, Eigenface, Fisherface ในการทดลองภาพที่ใช้ฝึกสอนถูกเลือกมาอย่างสุ่มจำนวน 5 ภาพต่อหนึ่งบุคคลและภาพที่เหลือถูกนำมาใช้สำหรับทดสอบ ดังนั้นภาพที่ใช้ฝึกสอนมีจำนวนทั้งหมด 200 ภาพ และภาพที่ใช้ในการทดสอบ 200 ภาพ ในการทดลองวิธี Eigenface ได้ทำการหาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 1 จนถึงฟีเจอร์มากที่สุดที่ 199 ($199 = M - 1$ โดยที่ M คือจำนวนภาพทั้งหมด) และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธี Fisherface ได้หาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 1 จนถึงฟีเจอร์มากที่สุดที่ 39 ส่วนวิธี 2DPCA และวิธี 2DLDA ได้ทำการหาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 112 x 1 จนถึงฟีเจอร์มากที่สุดที่ 112 x 92 ซึ่งผลการทดลองแสดงในรูปที่ 4.7

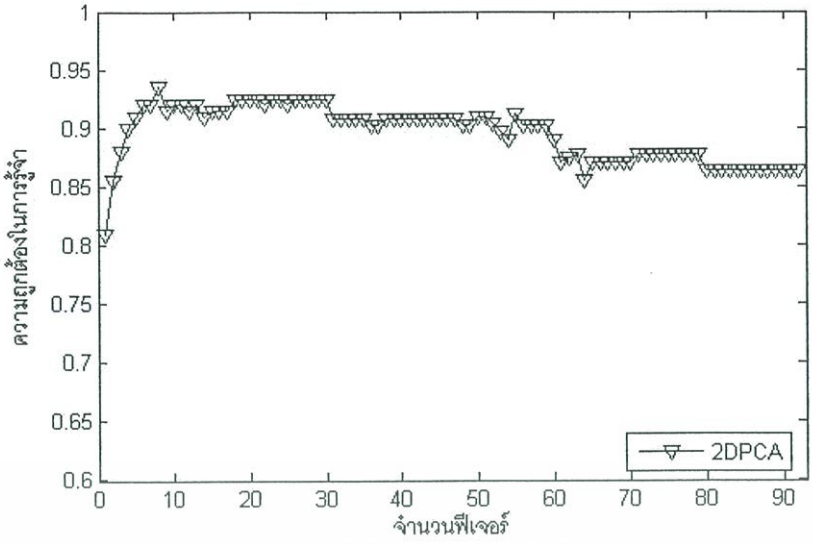


(ก)

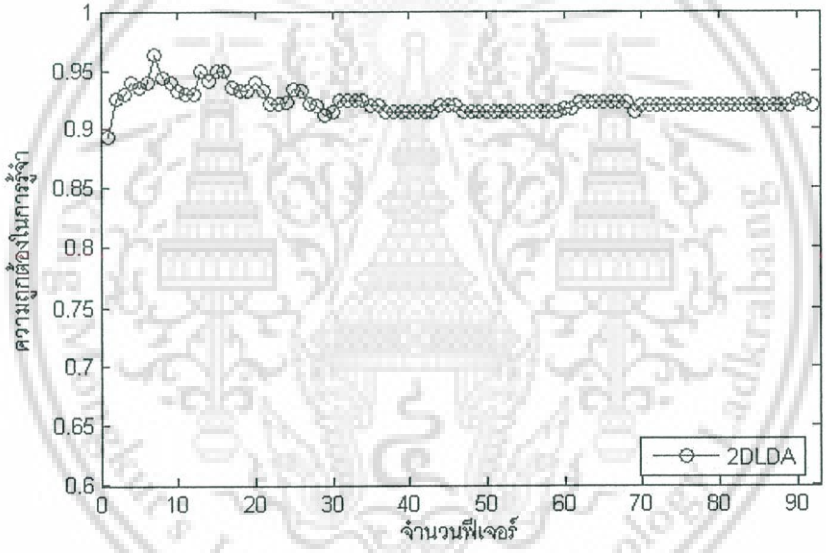


(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ค)



(ง)

รูปที่ 4.7 เปรียบเทียบร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA กับวิธี Eigenface วิธี Fisherface และวิธี 2DPCA

(ก) ร้อยละความถูกต้องในการรู้จำของวิธี Eigenface

(ข) ร้อยละความถูกต้องในการรู้จำของวิธี Fisherface

(ค) ร้อยละความถูกต้องในการรู้จำของวิธี 2DPCA

(ง) ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.7 สามารถหาร้อยละความถูกต้องในการรู้จำสูงสุดได้ดังตารางที่ 4.2 โดยที่ตารางที่ 4.2 แสดงร้อยละความถูกต้องในการรู้จำสูงสุดของทั้ง 4 วิธี

ตารางที่ 4.2 เปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดของทั้ง 4 วิธี

วิธี	ร้อยละความถูกต้องในการรู้จำสูงสุด	ขนาดของฟีเจอร์
Eigenface	89.57	60
Fisherface	92.53	33
2DPCA	93.50	112 x 8
2DLDA	96.00	112 x 7

จากตารางที่ 4.2 พบว่าวิธี 2DLDA มีความถูกต้องในการรู้จำดีที่สุดในสี่วิธี โดยที่ผลการรู้จำที่ดีที่สุดของวิธี 2DLDA คือ ร้อยละ 96.0 ที่ฟีเจอร์มีขนาดเท่ากับ 112 x 7 ซึ่งดีกว่าวิธี 2DPCA ที่ผลการรู้จำที่ดีที่สุดเพียงร้อยละ 93.5 ที่ฟีเจอร์มีขนาดเท่ากับ 112 x 8

ตารางที่ 4.3 เปรียบเทียบเวลา ที่ใช้คำนวณหาฟีเจอร์ของภาพ

วิธี	2DLDA	2DPCA	Eigenface	Fisherface
เวลา (วินาที)	0.621	0.621	32.586	40.978

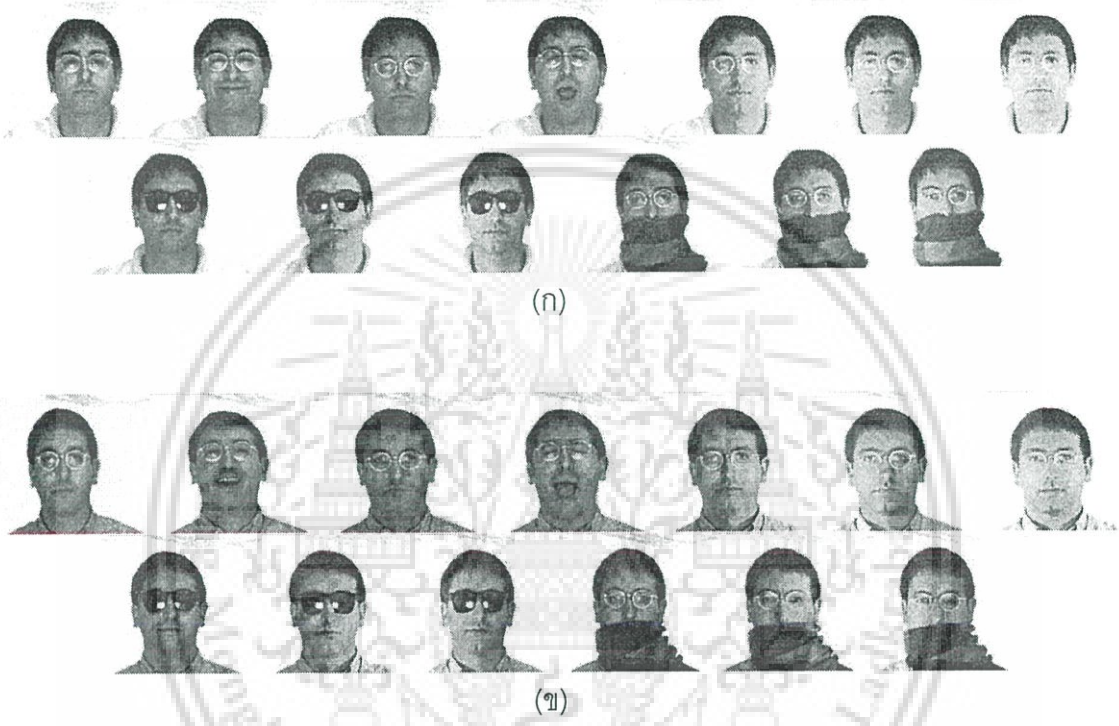
ตารางที่ 4.3 แสดงเวลาที่ใช้ในการฝึกสอนทั้งสี่วิธี (CPU: Pentium IV 1.7 GHz, RAM: 256M) โดยทั้งสี่วิธีถูกคำนวณภายใต้โปรแกรม Matlab 6.5 จากตารางที่ 2 จะเห็นว่าวิธี 2DLDA กับ วิธี 2DPCA ใช้เวลาในการคำนวณน้อยกว่าวิธี Eigenface และ วิธี Fisherface มาก เพราะว่า วิธี 2DLDA และ วิธี 2DPCA จะจัดการกับเมตริกที่มีขนาดเพียง 92 x 92 แต่วิธี Eigenface และ วิธี Fisherface ต้องจัดการกับเมตริกที่ขนาด 10,304 x 10,304 ดังนั้นการคำนวณจึงทำได้ยากและใช้เวลามาก

4.2 การทดลองบนฐานข้อมูลใบหน้า AR

ฐานข้อมูลภาพใบหน้าของ AR ถูกสร้างขึ้นโดย Aleix Martinez และ Robert Benavente ใน Computer Vision Center (CVC) ที่ U.A.B. ประกอบด้วยภาพใบหน้าสีขนาด 768 x 576 จุดภาพที่ความละเอียดของสี 24 บิต มากกว่า 4,000 ภาพ ของ 126 บุคคล แบ่งเป็นผู้ชาย 70 คน และเป็นผู้หญิง 56 คน ซึ่งถ่ายเฉพาะด้านหน้าที่แตกต่างกันตามเงื่อนไขต่าง ๆ เช่น เงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า เงื่อนไขของความสว่างบนใบหน้า เงื่อนไขการสวมแว่นตากันแดดและผ้าพันคอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพใบหน้าของฐานข้อมูล AR ที่สามารถให้นำมาใช้ได้มีเพียง 120 บุคคล แบ่งเป็นชาย 66 คน และหญิง 55 คน โดยภาพใบหน้าของแต่ละบุคคลจะถูกแบ่งออกเป็น 2 กลุ่ม กลุ่มแรกประกอบ ด้วยภาพใบหน้า 13 ภาพที่แตกต่างกัน ส่วนกลุ่มที่สองจะมีลักษณะภาพใบหน้าเหมือนกับกลุ่มแรกแต่ถ่ายภาพห่างกันเป็นระยะเวลา 2 สัปดาห์ ดังแสดงในรูปที่ 4.8 รูปที่ 4.8 เป็นตัวอย่างภาพใบหน้าของบุคคลคนหนึ่งที่ถูกแบ่งเป็นสองกลุ่ม โดยที่รูปที่ 4.8 (ก) เป็นกลุ่มที่ 1 และรูปที่4.8 (ข) เป็นกลุ่มที่ 2



รูปที่ 4.8 ตัวอย่างภาพใบหน้าของบุคคลคนหนึ่งที่ถูกแบ่งเป็นสองกลุ่ม
(ก) ตัวอย่างภาพใบหน้าของบุคคลคนหนึ่งที่ถูกแบ่งเป็นกลุ่มที่ 1
(ข) ตัวอย่างภาพใบหน้าของบุคคลคนหนึ่งที่ถูกแบ่งเป็นกลุ่มที่ 2

ก่อนการทดลองได้ทำการเลือกภาพใบหน้าของแต่ละบุคคล ทั้ง 120 คนมาตามรูปที่ 4.9 โดยที่รูปในกลุ่มที่ 1 ประกอบด้วยรูปที่ 4.9 (ก) เป็นภาพใบหน้าปกติ รูปที่ 4.9 (ข) เป็นภาพใบหน้ายิ้ม รูปที่ 4.9 (ค) เป็นภาพใบหน้าโกรธ รูปที่ 4.9 (ฅ) เป็นภาพใบหน้าแสดงความกลัว รูปที่ 4.9 (ฆ) เป็นภาพใบที่มีแสงส่องด้านขวา รูปที่ 4.9 (ง) เป็นภาพใบที่มีแสงส่องด้านซ้าย และรูปที่ 4.9 (จ) เป็นภาพใบที่มีแสงส่องทั้งสองด้าน ส่วนกลุ่มที่ 2 ทำเหมือนกลุ่มแรกจะได้ดังรูปที่ 4.9 (ฉ) , รูปที่ 4.9(ช), รูปที่ 4.9 (ซ), รูปที่ 4.9 (ฌ), รูปที่ 4.9 (ญ), รูปที่ 4.9 (ฎ), รูปที่ 4.9 (ฏ) ดังนั้นภาพที่ทำการตัดทิ้งจากรูปที่ 4.8 คือภาพที่มีการสวมแว่นตาดำและผ้าพันคอ หลังจากนั้นทำการแปลงภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมดให้เป็นภาพระดับสีเทาที่ความละเอียดของสี 8 บิต และเนื่องจากภาพใบหน้าในข้อมูลนี้ขนาดใหญ่จึงปรับภาพให้มีขนาด 50 x 40 จุดภาพ

กลุ่มที่ 1

(ก) (ข) (ค) (ศ) (ฆ) (ง) (จ)

กลุ่มที่ 2

(ฉ) (ช) (ซ) (ฌ) (ญ) (ฎ) (ฏ)

รูปที่ 4.9 ตัวอย่างภาพใบหน้าของ 1 บุคคลที่ถูกเลือกมาใช้ในการทดลอง

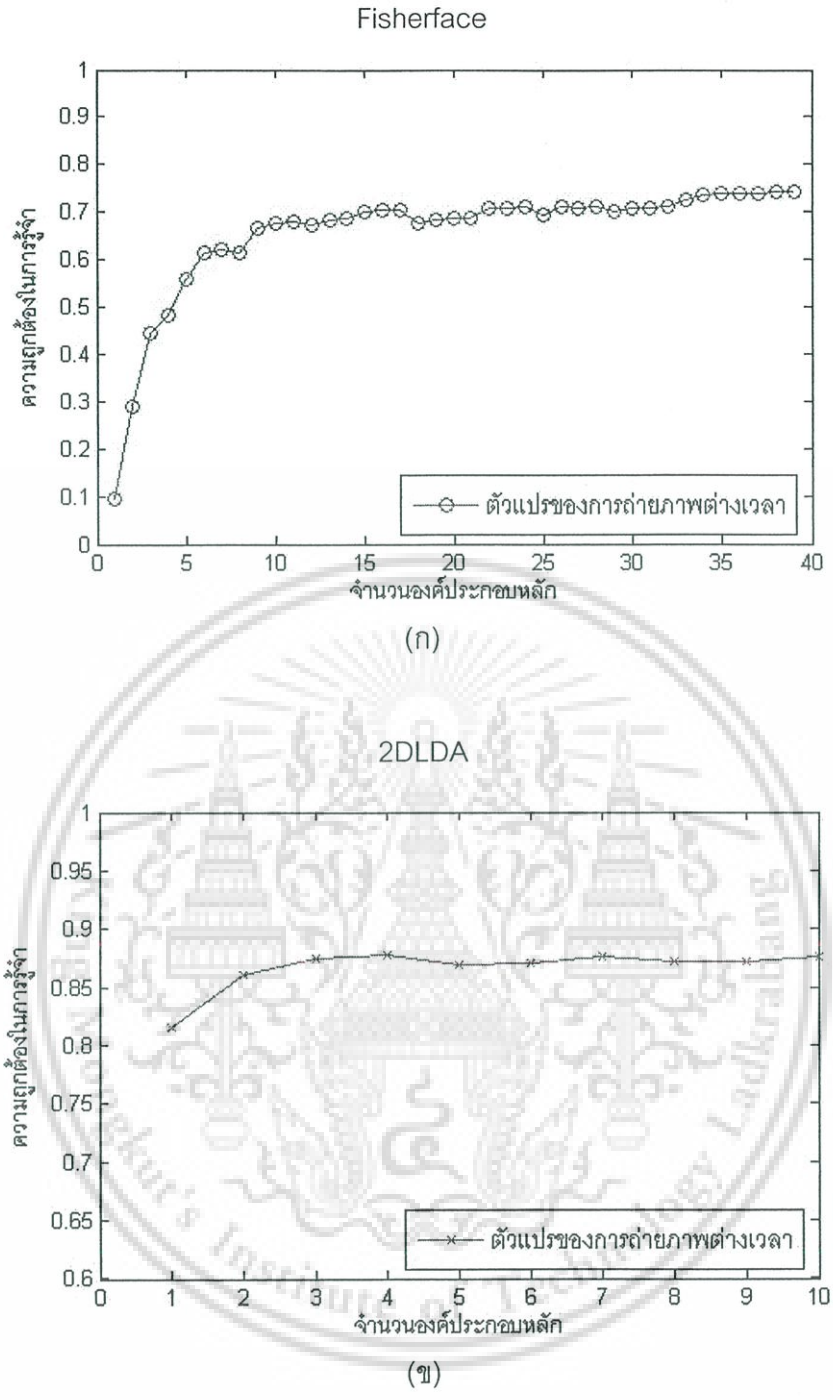
4.2.1 การทดลองกับเงื่อนไขการถ่ายภาพต่างเวลา

การทดลองนี้ได้ทำการเปรียบเทียบวิธี 2DLDA กับ วิธี Fisherface ภายใต้เงื่อนไขการถ่ายภาพต่างเวลา โดยการทดลองได้ใช้ภาพจากรูปที่ 4.9 กลุ่มที่ 1 ทั้งหมดมาใช้สอนและภาพจากรูปที่4.9 กลุ่มที่ 2 ถูกนำมาใช้ทดสอบ ดังนั้นภาพที่ใช้ในการสอนทั้งหมดจึงมี 840 ภาพ และภาพที่ใช้ในการทดสอบทั้งหมด 840 ภาพ

การหาฟีเจอร์ของภาพด้วยวิธี Fisherafce จะเลือกใช้จำนวนองค์ประกอบหลักทั้งหมด 39 เวกเตอร์ ส่วนของวิธี 2DLDA จะใช้องค์ประกอบหลักเพียง 10 เวกเตอร์ ในการจำแนกบุคคลนั้นได้ใช้การวัดความใกล้เคียงกันมากที่สุดของภาพใบหน้าที่ถูกหาฟีเจอร์แล้ว (1NN) ซึ่งผลของความถูกต้องในการรู้จำใบหน้าของทั้งสองวิธีแสดงในรูปที่ 4.10 จากรูปที่ 4.10 จะพบว่าวิธี 2DLDA มีความถูกต้องในการรู้จำสูงกว่าวิธี Fisherface ซึ่งความถูกต้องในการรู้จำสูงสุดของวิธี 2DLDA เป็น ร้อยละ 88.64 ที่ 10 ฟีเจอร์เมตริก ขณะที่ความถูกต้องในการรู้จำสูงสุดของวิธี Fisherface เป็น ร้อยละ 72.31 ที่ 39 ฟีเจอร์เวกเตอร์

การคำนวณหาเวลาที่ใช้ในการหาฟีเจอร์ของภาพของทั้งสองวิธีสามารถแสดงในตารางที่ 4.4 จากตารางที่ 4.4 เวลาที่ใช้ในการคำนวณหาฟีเจอร์ของของวิธี 2DLDA มีความเร็วกว่าการคำนวณหาฟีเจอร์ของของวิธี Fisheraface เป็น 100 เท่า เนื่องจากการคำนวณหาไอเกนเวกเตอร์ของวิธี 2DLDA จะคำนวณจากเมตริกขนาด 40 x 40 ขณะที่วิธี Fisherafce จะคำนวณจากเมตริกขนาด 2,000 x 2,000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA และ วิธี Fisherface ภายใต้ตัวแปรของการถ่ายภาพต่างเวลา

(ก) ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA

(ข) ร้อยละความถูกต้องในการรู้จำของวิธี Fisherface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 เปรียบเทียบวิธี 2DLDA กับวิธี Fisherface ภายใต้เงื่อนไขการถ่ายภาพต่างเวลา เงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า เงื่อนไขของแสงบนใบหน้า

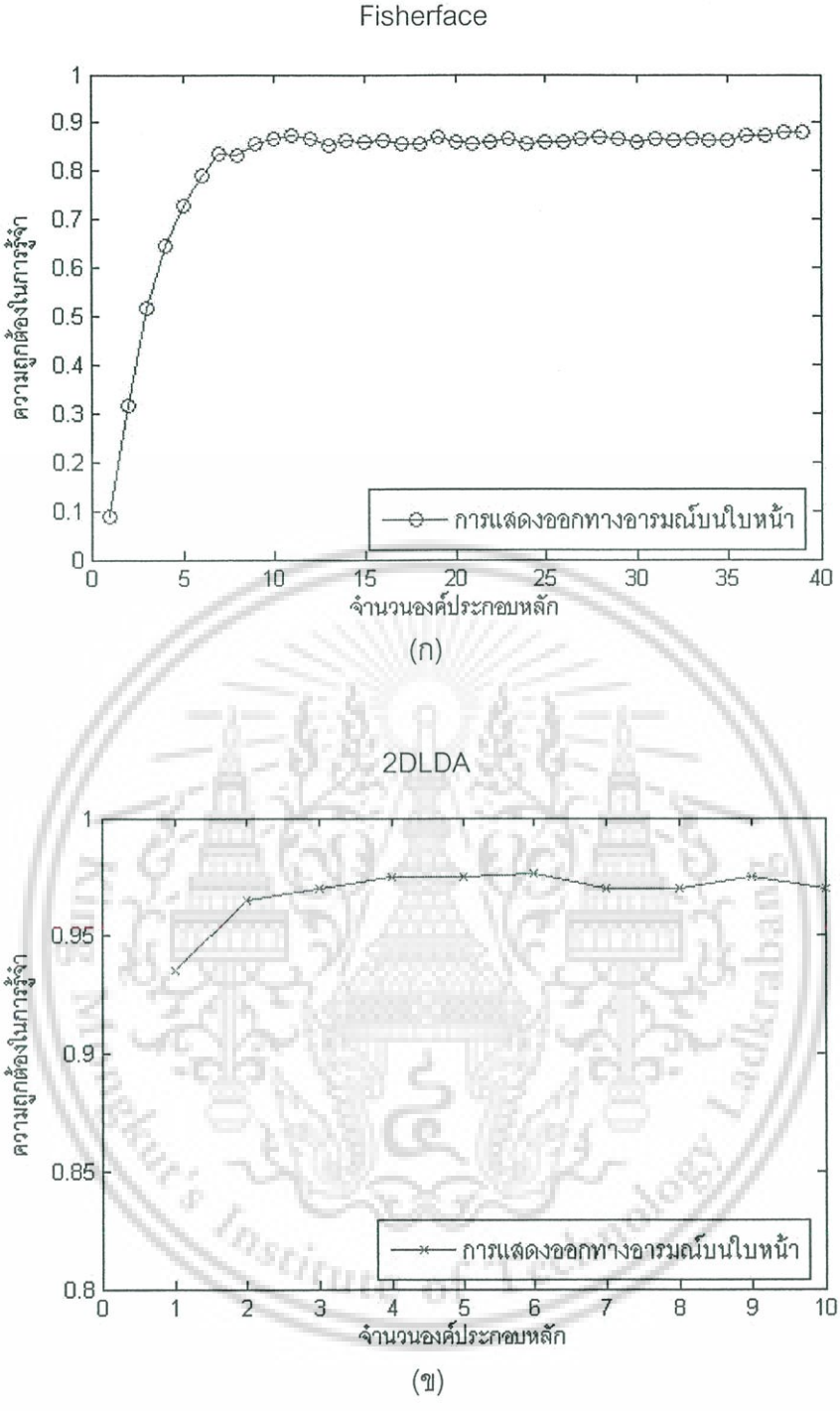
การทดลองภายใต้เงื่อนไขของการเปลี่ยนแปลงบนใบหน้า		เวลาในการคำนวณหาฟีเจอร์ของภาพ (วินาที)	ร้อยละความถูกถูกต้องในการรู้จำสูงสุด	ขนาดเมตริกที่ใช้ในการคำนวณ
เงื่อนไข	วิธี			
การถ่ายภาพต่างเวลา	Fisherface	20.782	72.00	2,000 x 2,000
	2DLDA	0.216	88.00	40 x 40
การแสดงออกทางอารมณ์บนใบหน้า	Fisherface	13.480	89.54	2,000 x 2,000
	2DLDA	0.146	97.45	40 x 40
ตัวแปรทางแสงบนภาพใบหน้า	Fisherface	10.454	85.70	2,000 x 2,000
	2DLDA	0.094	92.50	40 x 40

4.2.2 การทดลองกับเงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า

การทดลองนี้ได้ทำการเปรียบเทียบวิธี 2DLDA กับ วิธี Fisherface ภายใต้เงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า ก่อนการทดลองได้ทำการเลือกภาพจากฐานข้อมูลทั้งหมดมาเป็นฐานข้อมูลใหม่ ดังนั้นภาพใบหน้าของแต่ละบุคคลที่ถูกเลือกจากกลุ่มที่ 1 เป็น 4.9(ก), 4.9(ข), 4.9(ค), 4.9(ศ) และจากกลุ่มที่ 2 เป็น 4.9(ฉ), 4.9(ช), 4.9(ซ), 4.9(ฌ) ซึ่งภาพที่เลือกทั้งหมดจะอยู่ภายใต้เงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า หลังจากนั้นได้ทำการเลือกภาพที่ใช้ในการสอนจากแต่ละบุคคลมาสองภาพเป็น 4.9(ก) และ4.9(ฉ) ส่วนภาพที่เหลือถูกนำมาทดสอบความถูกต้องในการรู้จำเป็น 4.9(ข), 4.9(ค), 4.9(ศ) และ 4.9(ช), 4.9(ซ), 4.9(ฌ) ดังนั้นภาพที่ใช้สอนทั้งหมดจะมี 240 ภาพและภาพที่ใช้ทดสอบ 720 ภาพ

จากการทดลองที่ 4.2.1 จำนวนองค์ประกอบหลักที่ใช้ในวิธีของ Fisherface คือ 39 เวกเตอร์ ขณะที่ 2DLDA ใช้องค์ประกอบหลัก 10 เวกเตอร์ ดังนั้นการทดลองนี้จึงกำหนดให้มีค่าเท่ากัน รูปที่ 4.11 (ก) และรูปที่ 4.11 (ข) แสดงร้อยละความถูกต้องในการรู้จำตามจำนวนขององค์ประกบหลัก จากรูปที่ 4.11 สามารถหาค่าร้อยละความถูกต้องในการรู้จำสูงสุดได้ตามตารางที่ 4.4 ซึ่งจากตารางที่ 4.4 จะพบว่าวิธี 2DLDA มีผลการรู้จำดีกว่าและมีเวลาในการคำนวณน้อยกว่าวิธี Fisherface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA และ วิธี Fisherface ภายใต้เงื่อนไขการแสดงออกทางอารมณ์บนใบหน้า

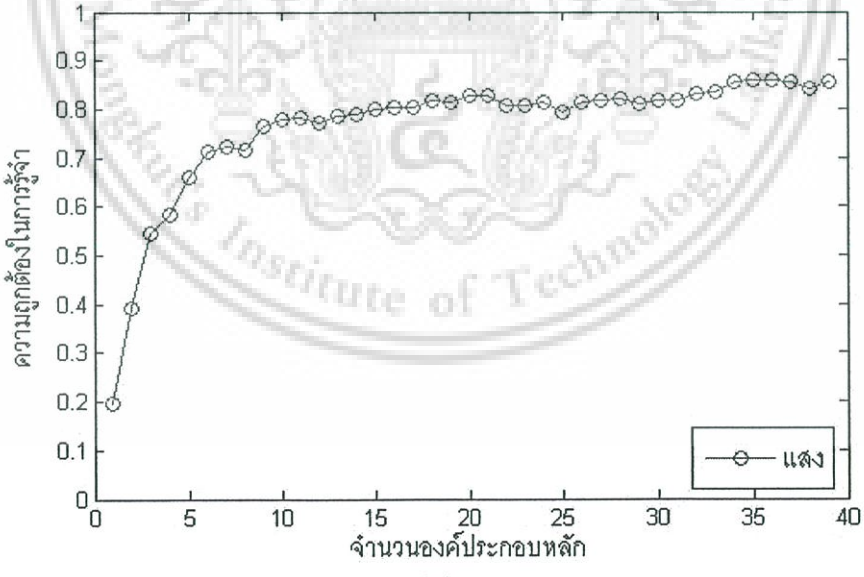
(ก) ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA

(ข) ร้อยละความถูกต้องในการรู้จำของวิธี Fisherface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การทดลองภายใต้เงื่อนไขของความสว่างบนใบหน้า

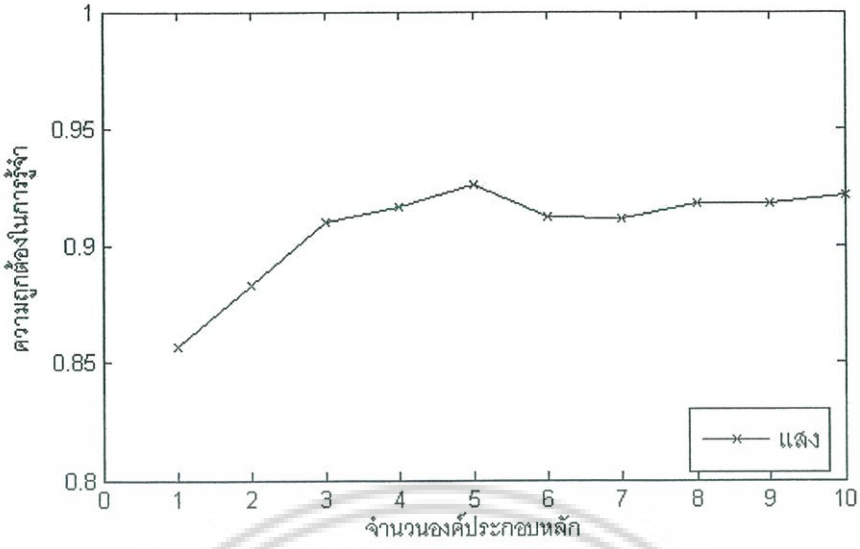
ในการทดลองนี้ทำการเปรียบเทียบวิธี 2DLDA กับวิธี Fisherface ภายใต้เงื่อนไขของความสว่างบนใบหน้า โดยทำการเลือกภาพจากฐานข้อมูลทั้งหมดมาเป็นฐานข้อมูลใหม่ซึ่งจะนำมาใช้ในการทดลอง ดังนั้นภาพใบหน้าของแต่ละบุคคลจากกลุ่มที่ 1 เป็นรูปที่ 4.9(ก), รูปที่ 4.9(ฆ), รูปที่ 4.9(ง), รูปที่ 4.9(จ) และจากกลุ่มที่ 2 เป็นรูปที่ 4.9(ฉ), รูปที่ 4.9(ญ), รูปที่ 4.9(ฎ), รูปที่ 4.9(ฏ) หลังจากนั้นทำการเลือกภาพมาอย่างสุ่มคนละ 2 ภาพ จากกลุ่มที่ 1 หนึ่งภาพและจากกลุ่มที่ 2 อีกหนึ่งภาพเพื่อนำมาใช้ในการสอน ส่วนภาพที่เหลือจากการเลือกถูกนำมาใช้ในการทดสอบความถูกต้องในการรู้จำ ดังนั้นภาพที่ใช้ฝึกสอนมีจำนวนทั้งหมด 240 ภาพ และภาพที่ใช้ในการทดสอบ 720 ภาพ ในการทดลองจำนวนองค์ประกอบหลักที่ใช้ในวิธี Fisherface คือ 39 เวกเตอร์ ขณะที่ 2DLDA ใช้องค์ประกอบหลัก 10 เวกเตอร์ ส่วนการจำแนกบุคคลนั้นได้ใช้การวัดความใกล้เคียงกันมากที่สุดของภาพใบหน้าที่ถูกหาฟีเจอร์แล้ว (1NN) ซึ่งผลของร้อยละความถูกต้องในการรู้จำใบหน้าของทั้งสองวิธีแสดงในรูปที่ 4.12 จากรูปที่ 4.12 สามารถหาร้อยละความถูกต้องในการรู้จำสูงสุดและเวลาที่ใช้ในการคำนวณหาฟีเจอร์ของภาพตามตารางที่ 4.4 จากตารางที่ 4.4 จะพบว่าร้อยละความถูกต้องในการรู้จำสูงสุดของวิธี 2DLDA จะมีค่าเป็น 92.50 ซึ่งมีค่ามากกว่าวิธี Fisherface อยู่ร้อยละ 7 และจากตารางที่ 4.4 ยังแสดงให้เห็นถึงเวลาในการคำนวณหาฟีเจอร์ของภาพจากวิธี 2DLDA ซึ่งจะเร็วกว่าวิธี Fisherface



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2DLDA



(ข)

รูปที่ 4.12 ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA และ วิธี Fisherface ภายใต้เงื่อนไขการความสว่างบนใบหน้า

(ก) ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA

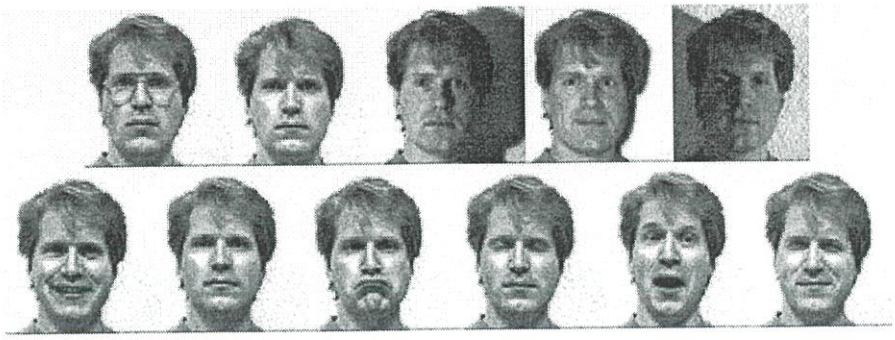
(ข) ร้อยละความถูกต้องในการรู้จำของวิธี Fisherface

4.3 **การทดลองบนฐานข้อมูลใบหน้า** Yale

ฐานข้อมูลภาพใบหน้าของ Yale ถูกสร้างขึ้นมาจาก Yale Center for Computational Vision and Control โดยภาพใบหน้าทั้งหมดเป็นภาพระดับสีเทาฉากหลังมีสีขาวขนาด 320 x 243 จุดภาพ ถ่ายจาก 15 บุคคล คนละภาพ 11 ภาพที่แตกต่างกัน ซึ่งในฐานข้อมูลภาพใบหน้านี้จะมีทั้งผู้หญิงและผู้ชาย โดยรูปที่ 4.13 เป็นตัวอย่างภาพใบหน้าของบุคคลคนหนึ่ง ซึ่งภาพแรกจะเป็นภาพที่มีแสงล้อมรอบบนลักษณะใบหน้าที่เป็นปกติและบุคคลคนนั้นใสแว่น ในภาพที่สองจะเป็นภาพที่มีแสงล้อมรอบบนลักษณะใบหน้าที่เป็นปกติและบุคคลนั้นไม่ใส่แว่น ถ้าปกติบุคคลนั้นใส่แว่น ภาพที่ 3 - 5 เป็นภาพที่ได้มาโดยการสองสว่างของบนใบหน้าธรรมชาติโดยใช้ Luxolamp วางใน 3 จุดตามลำดับ ภาพที่เหลือได้มาจากภาพที่ล้อมรอบไปด้วยแสง แต่มีลักษณะที่แสดงออกทางบนใบหน้าที่แตกต่างกัน (เช่น มีความสุข,เศร้า,หลับ,กระพริบตา,และประหลาดใจ)

เนื่องจากภาพใบหน้าในข้อมูลนี้ขนาดใหญ่ทำให้การคำนวณใช้เวลามาก ก่อนการทดลองภาพทั้งหมดจึงถูกตัดและปรับให้มีขนาด 100 x 80 จุดภาพ

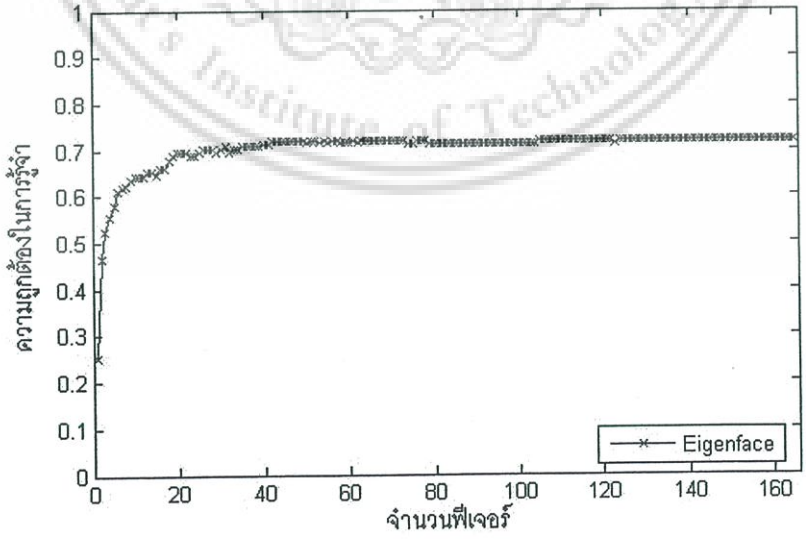
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ตัวอย่างภาพใบหน้า 1 บุคคลของฐานข้อมูล Yale

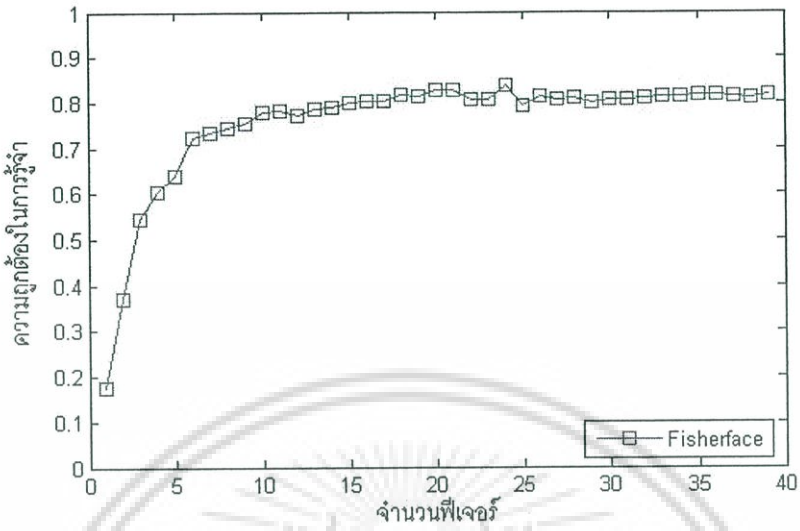
4.3.1 **การทดลองภายใต้เงื่อนไขการเปลี่ยนแปลงของการแสดงออกบนใบหน้าและความสว่างบนใบหน้า**

ในการทดลองกับฐานข้อมูลนี้ได้ใช้วิธีลีฟวิงวันเอาท์ (Leaving-one-out) ซึ่งเป็นการนำภาพทดสอบ 1 ภาพ ออกจากฐานข้อมูลภาพทั้งหมด แล้วนำภาพที่เหลือไปใช้ในการฝึกสอน ดังนั้นจะมีภาพที่ใช้ในการทดสอบ 1 ภาพ และภาพที่ใช้ในการฝึกสอน 164 ภาพ โดยจะนำภาพทุกภาพในฐานข้อมูลมาทำการทดสอบ จึงต้องทำการทดสอบทั้งหมด165 ครั้ง โดยที่วิธี Eigenface ได้ทำการหาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 1 จนถึงฟีเจอร์มากที่สุดที่ 163 และวิธี Fisherface ได้หาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 1 จนถึงฟีเจอร์มากที่สุดที่ 39 ส่วนวิธี 2DPCA และวิธี 2DLDA ได้ทำการหาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 112 x 1 จนถึงฟีเจอร์มากที่สุดที่ 112 x 92 ซึ่งผลการทดลองแสดงในรูปที่ 4.14

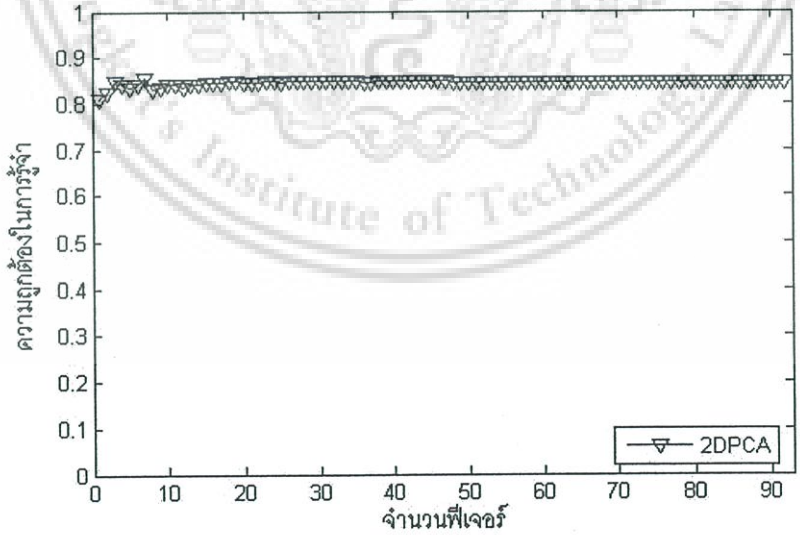


(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

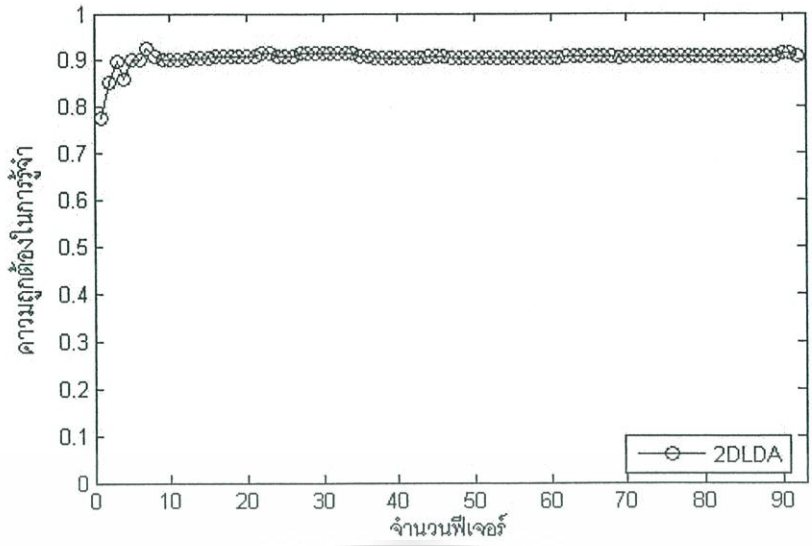


(ข)



(ค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ง)

รูปที่ 4.14 เปรียบเทียบร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA กับวิธี Eigenface วิธี Fisherface และวิธี 2DPCA โดยใช้วิธีลีฟวิงวันเอาท์

(ก) ร้อยละความถูกต้องในการรู้จำของวิธี Eigenface

(ข) ร้อยละความถูกต้องในการรู้จำของวิธี Fisherface

(ค) ร้อยละความถูกต้องในการรู้จำของวิธี 2DPCA

(ง) ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA

จากรูปที่ 4.14 สามารถหาร้อยละความถูกต้องในการรู้จำสูงสุดของของวิธี 2DLDA กับวิธี Eigenface วิธี Fisherface และวิธี 2DPCA ได้ตามตารางที่ 4.6

ตารางที่ 4.5 เปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดของวิธีทั้งสี่วิธีบนฐานข้อมูล Yale

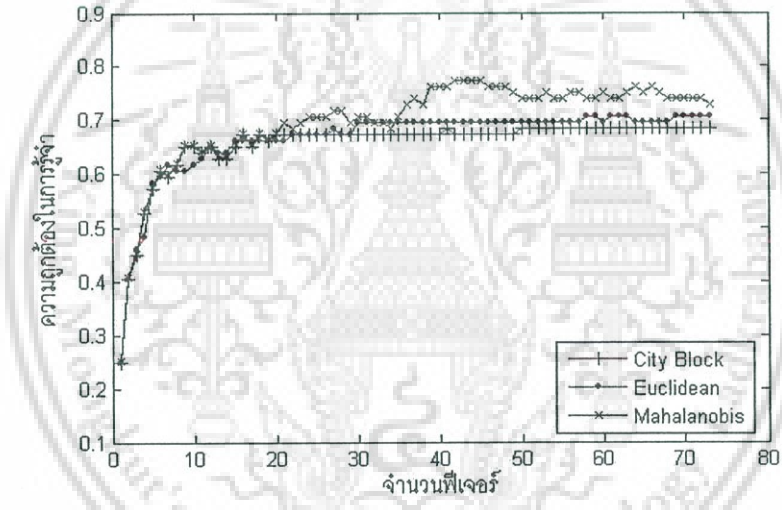
วิธี	ร้อยละความถูกต้องในการรู้จำสูงสุด	ขนาดฟีเจอร์
Eigenface	76.12	78
Fisherface	83.65	24
2DPCA	89.46	112 x 7
2DLDA	93.73	112 x 8

จากตารางที่ 4.5 พบว่าวิธี 2DLDA มีผลการรู้จำสูงสุดดีกว่า วิธี Eigenface, วิธี Fisherface และวิธี 2DPCA

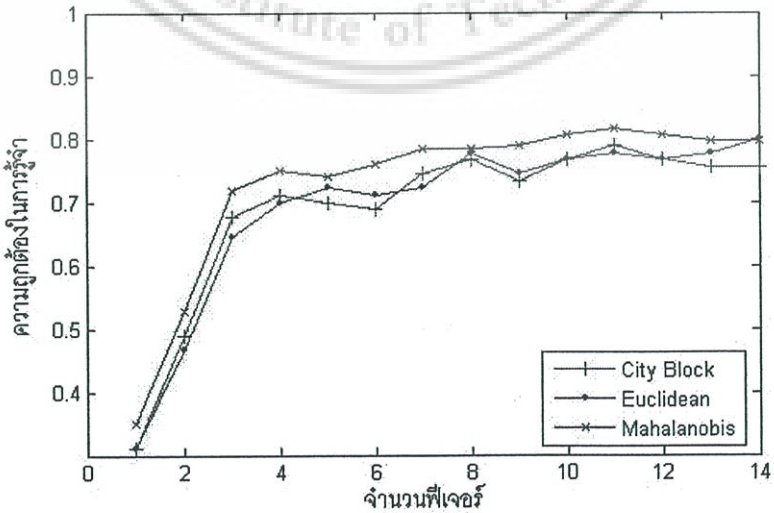
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การทดลองเปรียบเทียบกระบวนการของ 2DLDA ภายใต้เงื่อนไขของระยะทางที่ใช้ในการจำแนกบุคคล

การทดลองที่สองทำการเปรียบเทียบกระบวนการของ 2DLDA, 2DPCA, Eigenface, Fisherface ภายใต้เงื่อนไขของระยะทางที่ใช้ในการจำแนกบุคคลได้แก่ ระยะทางซิตีบล็อก (City Block Distance) ระยะทางยูคลิเดียน (Euclidean Distance) ระยะทางโมหาลาโนบิส (Mohalanobis Distance) การทดลองนี้ ภาพที่ใช้ฝึกสอนถูกเลือกมาอย่างสุ่มจำนวน 5 ภาพต่อหนึ่งบุคคลและภาพที่เหลือถูกนำมาใช้สำหรับทดสอบ ดังนั้นภาพที่ใช้ฝึกสอนมีจำนวนทั้งหมด 75 ภาพ และภาพที่ใช้ในการทดสอบ 90 ภาพ ในการทดลองนี้ วิธี Eigenface ได้ทำการหาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 1 จนถึงฟีเจอร์มากที่สุดที่ 74 และวิธี Fisherface ได้หาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 1 จนถึงฟีเจอร์มากที่สุดที่ 39 ส่วนวิธี 2DPCA และวิธี 2DLDA ได้ทำการหาค่าร้อยละความถูกต้องในการรู้จำตั้งแต่ฟีเจอร์ที่ 112 x 1 จนถึงฟีเจอร์มากที่สุดที่ 112 x 80 ซึ่งผลการทดลองแสดงในรูปที่ 4.15

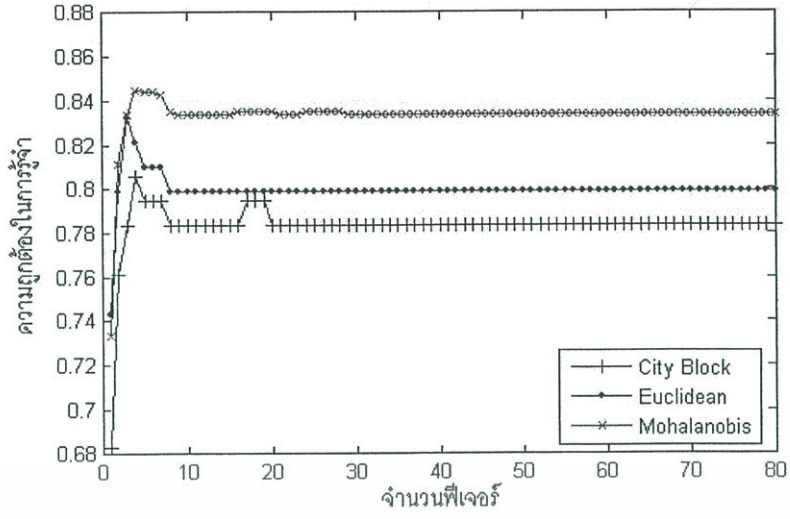


(ก)

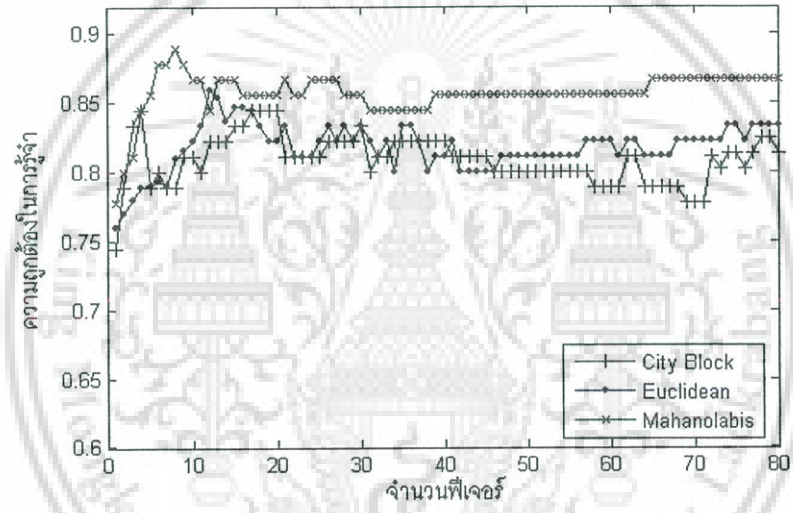


(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ค)



(ง)

รูปที่ 4.15 เปรียบเทียบร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA กับวิธี Eigenface วิธี Fisherface และวิธี 2DPCA ภายใต้เงื่อนไขของระยะทาง

(ก) ร้อยละความถูกต้องในการรู้จำของวิธี Eigenface

(ข) ร้อยละความถูกต้องในการรู้จำของวิธี Fisherface

(ค) ร้อยละความถูกต้องในการรู้จำของวิธี 2DPCA

(ง) ร้อยละความถูกต้องในการรู้จำของวิธี 2DLDA

จากรูปที่ 4.15 สามารถหาร้อยละความถูกต้องในการรู้จำสูงสุดของของวิธี 2DLDA กับ วิธี Eigenface วิธี Fisherface และวิธี 2DPCA ได้ตามตารางที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 เปรียบเทียบร้อยละความถูกต้องในการรู้จำสูงสุดของระยะทางที่ใช้ในการจำแนกบุคคล

วิธี	ระยะทางซิตีบล็อก	ระยะทางยูคลิเดียน	ระยะทางโมหาลาโนบิส
Eigenface	68.43	70.49	78.65
Fisherface	78.72	80.15	81.24
2DPCA	81.13	83.21	84.48
2DLDA	84.87	85.83	88.75

จากตารางที่ 4.6 พบว่าวิธี 2DLDA มีผลร้อยละความถูกต้องในการรู้จำสูงสุดดีกว่า วิธี Eigenface, วิธี Fisherface และวิธี 2DPCA ของทุกระยะทางที่ใช้ในการทดลอง โดยที่ระยะทางโมหาลาโนบิสจะให้ผลร้อยละความถูกต้องสูงสุดในการรู้จำดีกว่าการใช้ระยะทางยูคลิเดียนและระยะทางซิตีบล็อกในการจำแนกบุคคลของทุกวิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและแนวทางในการพัฒนา

5.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ศึกษาทฤษฎีการรู้จำใบหน้า โดยเฉพาะหลักการหาฟีเจอร์ของภาพด้วยวิธีการแปลงเชิงเส้นซึ่งประกอบด้วย

วิธีการวิเคราะห์องค์ประกอบหลัก ภาพใบหน้าจำเป็นต้องเปลี่ยนจากเมตริกภาพไปเป็นเวกเตอร์ภาพ นำมาคำนวณหาเมตริกความแปรปรวนร่วม หลังจากนั้นนำเมตริกความแปรปรวนร่วมมาหาไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนจากมากไปน้อยตามลำดับ แล้วนำไอเกนเวกเตอร์มาฉายภาพใบหน้าเพื่อหาฟีเจอร์ของภาพ ซึ่งข้อเสียของวิธีการนี้คือภาพใบหน้าที่เปลี่ยนจากเมตริกภาพไปเป็นเวกเตอร์ภาพ จะมีมิติมากและการคำนวณด้วยวิธีนี้ไม่นำค่าข้อมูลของกลุ่มมาพิจารณา

วิธีการแบ่งแยกเชิงเส้นของฟิชเชอร์ ภาพใบหน้าจำเป็นต้องเปลี่ยนจากเมตริกภาพไปเป็นเวกเตอร์ภาพแล้วนำมาคำนวณหาเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่ม เพื่อนำมาคำนวณหาไอเกนเวกเตอร์ที่สมนัยกับค่าไอเกนจากมากไปน้อยตามลำดับ แล้วนำไอเกนเวกเตอร์มาฉายภาพใบหน้าเพื่อหาฟีเจอร์ของภาพ ซึ่งข้อเสียของวิธีการนี้คือเมื่อหาเมตริกการกระจายภายในกลุ่มด้วยวิธีนี้จะทำให้ เกิดปัญหาการเป็นเมตริกเอกฐานได้เพราะว่า มิติของเมตริกการกระจายภายในกลุ่มมีค่ามากเมื่อเปรียบเทียบกับจำนวนของตัวอย่างภายในกลุ่มและการคำนวณด้วยวิธีการนี้จะใช้เวลามาก

วิธี 2DPCA ทำการคำนวณหาเมตริกความแปรปรวนร่วมโดยใช้ภาพที่เป็น 2 มิติมาคำนวณโดยตรง การคำนวณด้วยวิธีนี้เมตริกความแปรปรวนร่วมของภาพจะมีขนาดเล็ก ซึ่งเป็นผลให้ใช้เวลาในการคำนวณน้อยกว่าวิธี PCA แต่ข้อเสียของวิธีการนี้คือไม่ได้เอาข้อมูลของกลุ่มมาพิจารณาด้วย

วิทยานิพนธ์นี้จึงได้เสนอวิธีการใหม่ที่มีชื่อว่า Two – Dimensional Linear Discriminant Analysis (2DLDA) มาใช้ในการรู้จำใบหน้า ซึ่งวิธีการนี้พัฒนามาจากพื้นฐานวิธีการแบ่งแยกเชิงเส้นของ ฟิชเชอร์ที่นำเอาข้อมูลของกลุ่มมาคำนวณและการวิเคราะห์องค์ประกอบหลักของเมตริกภาพที่สามารถทำให้ขนาดของเมตริกการกระจายมีขนาดเล็ก โดยที่วิธี 2DLDA สามารถใช้เมตริกภาพมาคำนวณหาเมตริกการกระจายภายในกลุ่มและเมตริกการกระจายระหว่างกลุ่มได้ การคำนวณด้วยวิธีนี้ยังสามารถแก้ปัญหาการเป็นเมตริกเอกฐาน

จากผลการทดลองเปรียบเทียบความถูกต้องในการรู้จำของหลักการแปลงเชิงเส้นระหว่างวิธีการวิเคราะห์องค์ประกอบหลัก การแบ่งแยกเชิงเส้นของฟิชเชอร์ 2DPCA และวิธีการใหม่ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิทยานิพนธ์นี้ได้นำเสนอคือวิธี 2DLDA ภายใต้เงื่อนไขต่าง ๆ ที่ทำให้การรู้จำใบหน้ามีความผิดพลาดซึ่งได้แก่ การเปลี่ยนแปลงตำแหน่งและจำนวนตัวอย่างในการเรียนรู้ การเปลี่ยนแปลงของการแสดงออกบนใบหน้าและความสว่างบนใบหน้า พบว่าวิธี 2DLDA ให้ผลของความถูกต้องในการรู้จำดีที่สุดและใช้เวลาในการคำนวณน้อย

5.2 แนวทางในการพัฒนา

จากวิธี 2DLDA เมื่อหาฟีเจอร์ของภาพแล้วภาพจะถูกลดข้อมูลลงเฉพาะคอลัมน์ของภาพ ทำให้ขนาดของข้อมูลภาพยังมีขนาดใหญ่ เช่น ถ้าเราใช้วิธี 2DLDA ลดข้อมูลภาพที่มีขนาด 112 x 92 จุดภาพด้วยไอเกนเวกเตอร์จำนวน 6 เวกเตอร์ จะได้ฟีเจอร์ของภาพที่มีขนาด 112 x 6 ดังนั้นในการพัฒนาต่อ ควรจะหาวิธีใหม่ที่สามารถลดข้อมูลของภาพได้ทั้งคอลัมน์และแถว (row) ของภาพแล้วยังให้ผลการรู้จำที่ดีและใช้เวลาในการคำนวณหาฟีเจอร์ของภาพน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] M. Turk and A. Pentland, "Eigenfaces for Recognition," J. Cognitive Neuroscience, vol. 3, no. 1, 1991.

[2] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces : Recognition Using Class Specific Linear Projection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, Jul. 1997.

[3] J. Yang, "Two-Dimensional PCA: A New Approach to Appearance-Based FaceRepresentation and Recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 1, Jan. 2004.

[4] T. Kanade, "Picture Processing by Computer Complex and Recognition of Human Faces," Ph.D. dissertation, Kyoto Univ., Japan, 1973.

[5] R. Brunelli and T. Poggio, "Face Recognition : Features versus Templates," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, Oct. 1993.

[6] I. J. Cox, J. Ghosn, and P. N. Yianilos, "Feature-based Face Recognition using mixture-distance,"in Vision and Pattern Recognition. Piscataway, NJ: IEEE Press,1996.

[7] M. Lades, J. C. Vorbruggen, J. Buhann, J. Lange, C. Von der Malsburg, R. P.Wurtz, and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture," IEEE Transactions on Computers, Vol. 42, March 1993.

[8] L. Wiskott, J. M. Fellous, N. Kruger, and C. Von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," IEEE, 1997.

[9] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for Characterization of Human Faces," J. Optical Soc. Am., vol. 4, 1987.

[10] A.Pentland, T. Straner, N. Etcoff, A. Masoiu, O. Oliyide, and M. Turk, "Experiments with Eigenface," in Proc. Looking at people Wkshp, Int. Joint Cofn. Artificail Intell., Chamberry, France, 1993.

[11] A. Pentland, B. Moghaddam, and T. Staner, "View-based and Modular Eigenspaces for Face recognition," in Proc. IEEE conf. Computer Vision and Pattern Recognition 1994.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[12] S. J. Lee, S. B. Yung, J. W. Kwon, and S. H. Hong, "Face Detection and Recognition Using PCA," IEEE TENCON, 1999.

[13] S. Z. Lee and J. Lu, "Generalizing Capacity of Face Database for Face Recognition," IEEE, 1999.

[14] J. L. Crowley, and K. Schwerdt, "Robust Tracking and Compression for Video Communication," IEEE, 1999.

[15] R. A. Fisher, "The use of multiple Measures in Taxonomic Problems, Ann. Eugenics, 1936.

[16] K. Etemad and R. Chellappa, "Face Recognition Using Discriminant Eigenvectors," IEEE, 1996.

[17] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, and J. Weng, "Discriminant Analysis of Principal Component for Face Recognition," International Conference on Automatic Face and Gesture Recognition, 1998.

[18] W. Zhao, R. Chellappa , and N. Nandhakumar, "Empirical Performance Analysis of Linear Discriminant Classifiers," IEEE, 1998.

[19] W. Zhao, "Subspace Methods in Object/Face Recognition," IEEE, 1999.

[20] P. Temdee, D. Khawparisuth, and k. Chamnogthai, "Face Recognition by Using Fractal Encoding and Backpropagation Neural Network," Fifth International Symposium on Signal Processing and its Applications, ISSPA '99, Brisbane, Australia, August 1999.

[21] M. J. Er, S. Wu, and J. Lu, "Face Recognition Using Radial Basis Function (RBF) Neural Networks," Proceedings of the 38th Conference on Decision & Control," Phoenix, Arizona USA, IEEE, 1999.

[22] D. Demers and G. W. Cottrell, "Nonlinear Dimensionality Reduction, " in Advances in Neural Information Processing System 5, S. J. Hanson, J. D. Cowan, C. Lee Giles, Eds. San Mateo, Ca:Morgan Kaufmann, 1993.

[23] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolution Neural-network Approach," IEEE, Trans. Neural Networks, vol.8, no. 1, January 1997.

[24] S. A. Rizvi, P. J. Phillips, and H. Moo0n, "A verification protocol and statistical performance analysis for face recognition," IEEE Proc. Conf. Computer Vision and Pattern recognition (CVPR), Santa Barbara, June 1998.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[25] Y. Yoshitomi, T. Miyaura, S. Tomito, and S. Kimura, "Face Identification Using Termal Image Processing," IEEE International Workshop on Robot and Human Communication, IEEE, 1997.

[26] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification," Jhon Wiley & Sons, 2001.

[27] Hotelling. A Primer of Multivariate Statistics. Lawrence Erlbanum Associates, 2001

[28] http://www.cam-orl.co.uk

[29] A.M. Martinez and R. Benavente, "The AR Face Database," CVC Technical Report, no. 24, June 1998.

[30] A.M. Martinezand R.Benavente , "TheARFaceDatabase, "http://rvl1.ecn.purdue.edu /~aleix/aleix_face_DB.html, 2003.

[31] ftp://plucky.cs.yale.edu/CVC/pub/images/yalefaces/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
ผลงานที่ได้รับการตีพิมพ์

[1] สิทธินนท์ คงสนทนา, ยุทธพงษ์ รังสรรค์เสรี, "การรู้จำใบหน้าโดยใช้หลักการของ 2DLD วิศวสารลาดกระบัง, ปีที่ 22, ฉบับที่ 1, หน้า 37-42 , มีนาคม 2548.

[2] Sittinon Kongsontana and Yuttapong Rangsanseri, "Face Recognition Using 2DLDA Algorithm," Eighth International Symposium on Signal Processing and Its Applications, ISSPA 2005, Sydney, Australia, August 2005.

[3] สิทธินนท์ คงสนทนา, ยุทธพงษ์ รังสรรค์เสรี และ วีรรัตน์ ลีลาธรรม "การรู้จำใบหน้าโดยใช้หลักการของ CSLDA," การประชุมทางวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 28, ตุลาคม 2548.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
โปรแกรมที่ใช้ในการคำนวณ

ฟังชันก์ที่ใช้ในการทดลองทั้งหมด

```
%---------------------------------------------------- Eigenface----------------------------------------------
%-------------- General Information ----------------------------
% Program: eigenface.m
% Author: SK
%-------------- Define variables ---------------------------------
% Train - training images
%  P - number of training images
% mn - mean of Train
% msc - meas subtracted training images
% cov = covariance matrix msc' * msc
% V - eigenvectors
% D - eigenvalues
% vects = actual eigenvectors, multipied and normalized
% u - ordered eigenvectors (eigenspace)
% ev - ordered eigenvalues
%-------------------------------------------------------------------
 function [u,mn,msc] = eigenface(Train)
 fprintf('.....eigenface.....\n')
[N1 N2] = size(Train);  % N1 Number of rows and N2 Number of columns
N=N1;                   % Number of pixels in image
P=N2;                   % Number of images
 tol = .000001;     % tolerance

 % compute mean of each class
 fprintf('compute mean \n')
 mn = mean(Train')';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% Mean subtract all the images
for i=1:P
    msc(:,i) = Train(:,i) - mn;
end

% Find an orthonormal basis using k-l
fprintf('compute covarain matrix\n')
cov = msc' * msc;
[V,D] = eig(cov);
vects = msc * V;
for i=1:P
   vects(:,i) = vects(:,i)/norm(vects(:,i));
end

% Order the eigenvects according to the eigenvalues
 fprintf('Order the eigenvects according to the eigenvalues\n')
 for i=1:P
   evals(i) = D(i,i);
end
[a,b]=sort(evals);
for i=1:P
   ind = b(P-i+1);
   if (a(P-i+1) > tol)
     u(:,i) = vects(:,ind);
     ev(i) = D(ind,ind);
   end
end
%-----------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------------------Fisherface-----------------------------------------------
%---------- General Information ---------------------------
% Program: fishface.m
% Author: SK
%---------- Define variables ------------------------------
% Train -  original training images
% mn - mean of Train
% msc - mean (of each class) subtracted training images
% sw - within class scatter matrix
% sb - between class scatter matrix
% w - Fisher basis vectros
%---------------------------------------------------------
 function [w] = fisherface(Train,NumClasses,ImgsPerClass)
 fprintf('.....Fisherface.....\n')
 [N all] = size(Train);

 % compute mean of each class
  fprintf('compute mean of each class\n')
 mn = mean(Train')';
 for i = 1:NumClasses
   m(:,i) = mean(Train(:,(i-1)*ImgsPerClass+1:(i-1)*ImgsPerClass+ImgsPerClass)')';
   msm(:,i) = m(:,i) - mn;
 end

 % Mean subtract all the images
 fprintf('Mean subtract all the images\n')
 for i=1:NumClasses*ImgsPerClass
   msc(:,i) = Train(:,i) - m(:,double(fix((i-1)/ImgsPerClass))+1);
 end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% Calculate the sw matrix (within class scatter)
fprintf('Calculate within class scatter\n')
 sw = zeros(N);
 for i=1:NumClasses*ImgsPerClass
   sw = sw + msc(:,i) * msc(:,i)';
 end

% calculate the sb matrix (between class scatter)
fprintf('Calculate between class scatter\n')
 sb = zeros(N);
 for i=1:NumClasses
   sb = sb + msm(:,i) * msm(:,i)';
 end

 fprintf('Calculate eigenvalue\n')
 % solved the generalized eigenvalue problem sb*w = d*sw*w
 [V,D]=eig(sb,sw);
 szd = size(D);
 for i=1:szd(1)
   evals(i) = D(i,i);
 end
 [a,b]=sort(evals);

 % Pull off the eigenvectors associated with the NumClasses -1 most significant
 % eigenvalues
 for i = 1:NumClasses-1
   w(:,i)=V(:,b(szd(1)-(i-1)));
 end
 % Check that eig worked
 norm((sb*w)-(D(b(szd(1)),b(szd(1)))*sw*w));
%-------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------------------------------------------------2DPCA----------------------------------------------
%---------- General Information ----------
% Program: 2DPCA
% Author: SK
%---------- Define variables ----------
% Train - training images
% P - number of training images
% mn - mean of Train
% msc - meas subtracted training images
% G = covariance matrix
% V - eigenvectors
% D - eigenvalues
% vects = actual eigenvectors, multipied and normalized
% u - ordered eigenvectors (eigenspace)
% ev - ordered eigenvalues
%---------------------------------------------------------
function [u] = TwoDPCA(Train)
fprintf('.....2DPCA.....\n')
[N1 N2 P] = size(Train);
% compute mean of each class
fprintf('compute mean of each class\n')
mn = zeros(N1,N2);
for i=1:P
    mn  =  mn + Train(:,:,i);
end
mn = mn/P;

% Mean subtract all the images
fprintf('compute covarain matrix\n')
G = zeros(N2,N2);
msc = zeros(N2,N2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for i=1:P
  msc = (Train(:,:,i) - mn)' * (Train(:,:,i) - mn);
  G = G + msc ;
end
 G = G/P ;

% Find an orthonormal basis using k-l
 [V,D] = eig(G);
 vects = V;

% Order the eigenvects according to the eigenvalues
 fprintf('Order the eigenvects according to the eigenvalues\n')
 for i=1:N2
  evals(i) = D(i,i);
 end
 [a,b]=sort(evals);
 for i=1:N2
  ind = b(N2-i+1);
  if (a(N2-i+1) > tol)
    u(:,i) = vects(:,ind);
    ev(i) = D(ind,ind);
  end
 end
%-------------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------------------------2DLDA-----------------------------------------------
%------------- General Information ----------------------------
% Program: 2DLDA.m
% Author: SK
%------------- Define variables -------------------------------
% Train - Original training images
% mn - mean of Train
% msc - mean (of each class) subtracted training images
% sw - within class scatter matrix
% sb - between class scatter matrix
% U - Projection vectros
%--------------------------------------------------------------
 function [U] = TwoDLDA(Train,NumClasses,ImgsPerClass)
 fprintf('.....2DLDA.....\n')
 [N1 N2 P] = size(Train);

 % compute mean of each class
 fprintf('compute mean of each class\n')
 mn = zeros(N1,N2);
 for i=1:P
  mn  =  mn + Train(:,:,i); ;
 end
 mn = mn/P;
 ma = zeros(N1,N2);
 msm = zeros(N1,N2);
 for i = 1:NumClasses
    for j = 1:ImgsPerClass
         ma = ma + Train(:,:,((i-1)*ImgsPerClass)+j);
    end
   m(:,:,i)  = ma/ImgsPerClass;
   msm(:,:,i) = m(:,:,i) - mn;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 ma = zeros(N1,N2);
end

% Mean subtract all the images
fprintf('Mean subtract all the images\n')
for i=1:NumClasses*ImgsPerClass
 msc(:,:,i) = Train(:,:,i) - m(:,:,fix((i-1)/ImgsPerClass)+1);
end

% Calculate the sw matrix (within class scatter)
fprintf('Calculate within class scatter\n')
sw = zeros(N2,N2);
for i=1:NumClasses*ImgsPerClass
 sw = sw + msc(:,:,i)' * msc(:,:,i);
end

% calculate the sb matrix (between class scatter)
fprintf('Calculate between class scatter\n')
sb = zeros(N2,N2);
for i=1:NumClasses
 sb = sb + msm(:,:,i)' * msm(:,:,i);
end

% solved the generalized eigenvalue problem sb*w = d*sw*w
fprintf('Calculate eigenvalue\n')
[V D U]=svd(sb*inv(sw));
szd = size(D);
for i=1:szd(1)
  evals(i) = D(i,i);
end
%-----------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%---------------------------------- Load input form ORL Face Database ----------------------------
% Function for load face for Train (loadfaceff.m)
% Load input form ORL Face Database in Folder
% C:\\MATLAB7\\work\\FaceDatabase\\ORL
%-------------- Define variables --------------------------------
% ImgsPerClass =  image per class
% Numc =  number of calss
%.....................
% Defide value
clear
% NumClasses = input;
% ImgsPerClass =input;
TotalImagec =10;
IntTest = 0;
IntTrain = 0;
IntTestOth = 0;
IntTrainOth = 0;
ImgsPerClassTest = TotalImagec-ImgsPerClass;

 for i=1: NumClasses
     random = randperm(TotalImagec);
     fprintf('...train class %d....\n',i)
     for j=1: ImgsPerClass
        count  = random(j);
        sname =sprintf('C:\\MATLAB7\\work\\FaceDatabase\\ORL\\s%d\\%d.pgm',i,count);
         fprintf('%s\n',sname)

            % image Train Other
               IntTrainOth = IntTrainOth+1;
               TrainOth(:,IntTrainOth) = reshape(double(imread(sname)),92*112,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        % image Train Real
            IntTrain = IntTrain+1;
            Train(:,:,IntTrain) = double(imread(sname));
    end
        fprintf('...Test class %d....\n',i)
    for k=1: ImgsPerClassTest
        count  = random(Imagec+k);
        sname = sprintf('C:\\MATLAB7\\work\\FaceDatabase\\ORL\\s%d\\%d.pgm',i,count);
            fprintf('%s\n',sname)

            % Image Test Other
            IntTestOth = IntTestOth+1;
            TestOth(:,IntTestOth) = reshape(double(imread(sname)),92*112,1);

            % Image Test Real
            IntTest = IntTest+1;
            Test(:,:,IntTest) = double(imread(sname));

    end
end
    save Other.mat TrainOth TestOth
    save Real.mat Train Test

clear sname IntTrainOth IntTestOth IntTest Test Train TrainOth TestOth ImgsPerClass
clear random ImgsPerClassTest
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------------------------Function Classify For Feature Matrix (Euclidean)----------------------
% function for classify projection vector (classify.m)
function [True,False] = ...
classify(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest)
True = 0;
False = 0;
allcolum =0;
[L W M] = size(proj);
for kk = 1:NumClassesTest
  for ll = 1:ImgsPerClassTest


   for ii = 1:NumClasses
     for jj = 1:ImgsPerClass
       x =  proj(:,:,(ii-1)*ImgsPerClass+jj);
       y =  projt(:,:,(kk-1)*ImgsPerClassTest+ll);
       d =  y - x;
       distance((ii-1)*ImgsPerClass+jj) = norm(d,'fro');
     end
   end
   [a b] = min(distance);
   Value_Classify = ceil(b/ImgsPerClass) ;


        if Value_Classify == kk
            True = True + 1;
          else
            False = False + 1;
          end
     end
end
%-------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------Function Classify For Future Matrix (Cityblock)-------------------------
% function for classify projection Matrix (classifyothcityblock.m)
function [True,False] = ...
classifycityblock(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest)
True = 0;
False = 0;
allcolum =0;
[L W M] = size(proj);
for kk = 1:NumClassesTest
  for ll = 1:ImgsPerClassTest

   for ii = 1:NumClasses
     for jj = 1:ImgsPerClass
       x =  proj(:,:,(ii-1)*ImgsPerClass+jj);
       y =  projt(:,:,(kk-1)*ImgsPerClassTest+ll);
       d =  y - x;
       distance((ii-1)*ImgsPerClass+jj) = norm(d,1);
     end
   end
   [a b] = min(distance);
   Value_Classify = ceil(b/ImgsPerClass) ;

   if Value_Classify == kk
       True = True + 1;
   else
       False = False + 1;
   end
   end
end
%-------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------Function Classify For Future Matrix (Mahalanobis)----------------------
% function for classify projection Matrix (classifymaha.m)
function [True,False] =...
classifymaha(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest,weights)
True = 0;
False = 0;
allcolum =0;
[L W M] = size(proj);
for kk = 1:NumClassesTest
  for ll = 1:ImgsPerClassTest
    for ii = 1:NumClasses
      for jj = 1:ImgsPerClass
        x =  proj(:,:,(ii-1)*ImgsPerClass+jj);
        y =  projt(:,:,(kk-1)*ImgsPerClassTest+ll);
        d = 0;
        for i=1:W
          d = d + (-sum(y(:,i) .* x(:,i).*weights(:,i)));
        end
        distance((ii-1)*ImgsPerClass+jj) = d;
    end
  end
  [a b] = min(distance);
  Value_Classify = ceil(b/ImgsPerClass) ;
  if Value_Classify == kk
      True = True + 1;
  else
      False = False + 1;
  end
  end
end
%-------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------------------------Function Classify For Future Vector (Euclidean)-----------------------
% function for classify projection Matrix (classifyoth.m)
function [True,False] =...
classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest)
True = 0;
False = 0;
for kk = 1:NumClassesTest
  for ll = 1:ImgsPerClassTest

   for ii = 1:NumClasses
     for jj = 1:ImgsPerClass
      x =  proj(:,(ii-1)*ImgsPerClass+jj);
      y =  projt(:,(kk-1)*ImgsPerClassTest+ll);
      d =  x - y;
      distance((ii-1)*ImgsPerClass+jj) = norm(d,'fro');
      end
   end

   [a b] = min(distance);
   Value_Classify = ceil(b/ImgsPerClass) ;

  % fprintf('distance class%d  real class %d\n' ,Value_Classify,kk)
   if Value_Classify == kk
      True = True + 1;
   else
       False = False + 1;
   End

  end
end
%-------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------------------------Function Classify For Future Vector (Cityblock)-----------------------
% function for classify projection Matrix (classifyothcityblock.m)
function [True,False] =...
classifyothcityblock(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest)
True = 0;
False = 0;
for kk = 1:NumClassesTest
  for ll = 1:ImgsPerClassTest

   for ii = 1:NumClasses
     for jj = 1:ImgsPerClass
      x =  proj(:,(ii-1)*ImgsPerClass+jj);
      y =  projt(:,(kk-1)*ImgsPerClassTest+ll);
      d =  x - y;
      distance((ii-1)*ImgsPerClass+jj) = norm(d,1);
      end
   end

   [a b] = min(distance);
   Value_Classify = ceil(b/ImgsPerClass) ;

   % fprintf('distance class%d  real class %d\n' ,Value_Classify,kk)
   if Value_Classify == kk
       True = True + 1;
   else
        False = False + 1;
   end
     end
end
%-------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%--------------------------Function Classify For Future Vector (Mahalanobis)--------------------
% function for classify projection Matrix (classifyothmaha.m)
function [True,False] =...
classifyothmaha(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest)
True = 0;
False = 0;
for kk = 1:NumClassesTest
  for ll = 1:ImgsPerClassTest

   for ii = 1:NumClasses
     for jj = 1:ImgsPerClass
      x =  proj(:,(ii-1)*ImgsPerClass+jj);
      y =  projt(:,(kk-1)*ImgsPerClassTest+ll);
      d = -sum(y .* x.*weights);
      distance((ii-1)*ImgsPerClass+jj) = d
      end
   end
    [a b] = min(distance);

   Value_Classify = ceil(b/ImgsPerClass) ;
   % fprintf('distance class%d  real class %d\n' ,Value_Classify,kk)

   if Value_Classify == kk
       True = True + 1;
   else
        False = False + 1;
   end
  end
end
%-----------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

%----------การทดลองของฐานข้อมูล ORL ----------

% ----------**การทดลองที่** 1 การสร้างภาพใบหน้าของ 2DLDA----------

```
clear
% define variable
NumClasses = 40;
ImgsPerClass = 5;
NumClassesTest = 40;
ImgsPerClassTest = 5;
Loadfaceff;
% load Data for train
load Real.mat Train
load Real.mat Test
% Calculate Eigenvactor and Eigenvalue
[u] = TwoDLDA(Train,NumClasses,ImgsPerClass);
I=0;
figure;
colormap('gray');
fprintf('Showing the images\n')
for k=10:10:90
 I=I+1;
 proj = Train(:,:,1)*u(:,1:k);
 im=proj*u(:,1:k)';
 subplot(3,4,I);
 imagesc(im);
 title(['D = ',num2str(k)]);
 proj=[];
 axis equal
 axis off
end
```

%----------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------การทดลองที่ 2 เปรียบเทียบวิธี 2DLDA กับ Fisherface----------
% ----------Run program Fisherface----------
% Load face from ORL Data base
clear
for ImgsPerClass = 2:5
   NumClasses = 40;
   NumClassesTest = 40;
   TotalImage = 10;
   ImgsPerClassTest = TotalImage-ImgsPerClass;
   loadfaceff
   load Other.mat TrainOth

% Calculate Eigenvactor and Eigenvalue
   [w] = fisherface(TrainOth,NumClasses,ImgsPerClass);

% Calculate Projection Data
   load Other.mat TestOth

for Dimension = 1:NumClasses-1
      fprintf('projection data to %d  Dimension \n',Dimension)
   for i=1:NumClasses*ImgsPerClass
      proj(:,i) =  w(:,1:Dimension)' * TrainOth(:,i);
   end

 % projection data test
   for i=1:NumClassesTest*ImgsPerClassTest
         projt(:,i) =  w(:,1:Dimension)' * TestOth(:,i);
   end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 % classify
   [True,False] = ...
classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
   RecogntionRate_fisherface(Dimension)= ((True*100)/(True + False))/100;
   proj=[];
   projt=[];
end
   figure(ImgsPerClass)
   plot(1:1:Dimension,RecogntionRate_fisherface,'r-*')
   max(RecogntionRate_LDA)
   clear True False RecogntionRate_LDA TrainOth TestOth
end
%-------------------------------------------------------------------------------------------

% ------------------------------------Run program 2DLDA-------------------------------------
% Run program 2DLDA
% Load face from ORL Data base
Clear
% define variable
for ImgsPerClass = 2:5
     NumClasses = 40;
     NumClassesTest = 40;
     TotalImage = 10;
     ImgsPerClassTest = TotalImage-ImgsPerClass;
     Loadfaceff;

% load Data for train
load Real.mat Train
load Real.mat Test
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% Calculate Eigenvactor and Eigenvalue
[u] = TwoDLDA(Train,NumClasses,ImgsPerClass);

for Dimension = 1:92
% projection data
 fprintf('projection data to %d Dimension\n',Dimension)
  for i=1:NumClasses*ImgsPerClass
       proj(:,:,i) =  real(Train(:,:,i)*u(:,1:Dimension));
 end
 for i=1:NumClassesTest*ImgsPerClassTest
       projt(:,:,i) =  real(Test(:,:,i) * u(:,1:Dimension));
  end

% classify
 [True,False] = classify(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,
                         ImgsPerClassTest);
% calculate Percent recognition
 RecogntionRate_TwoDLDA(Dimension)= ((True*100)/(True + False))/100;
 proj =[];
 projt = [];
end
 figure(ImgsPerClass)
 plot(1:1:Dimension,RecogntionRate_TwoDLDA,'m-x')
 clear True False RecogntionRate_TwoDLDA Train Test
end
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%---------- เปรียบเทียบวิธี 2DLDA+ fisherface กับ Fisherface ----------
% Run program 2DLDA+Fisherface
% Load face from ORL Data base
% Select  Frist Five Face by Random
% select feture matrix of 2DLDA = 7 Feture matrix
clear
% load Data for train
load Real.mat Train
load Real.mat Test
% define variable
NumClasses = 40;
ImgsPerClass = 5;
NumClassesTest = 40;
ImgsPerClassTest = 5;
Feature_matrix = 7;

% Calculate Eigenvactor and Eigenvalue
[u] = TwoDLDA(Train,NumClasses,ImgsPerClass);

% projection data
 fprintf('projection data to %d Feature_matrix\n',Feature_matrix)

 for i=1:NumClasses*ImgsPerClass
       proj(:,:,i) =  real(Train(:,:,i)*u(:,1:Feature_matrix ));
 end
 for i=1:NumClassesTest*ImgsPerClassTest
       projt(:,:,i) =  real(Test(:,:,i) * u(:,1:Feature_matrix));
 end
%-------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 for i=1:NumClasses*ImgsPerClass
   TrainOth(:,i) = reshape(proj(:,:,i),112*7,1);
 end
for i=1:NumClassesTest*ImgsPerClassTest
   TestOth(:,i) = reshape(projt(:,:,i),112*7,1);
 end
   clear Train Test proj projt

%-------------------------------------------------------------------------
 [w] = fisherface(TrainOth,NumClasses,ImgsPerClass);

for Dimension = 1:39
% projection data
  fprintf('projection data to %d Dimension\n',Dimension)
   proj1=w(:,1:Dimension)'*TrainOth;
   projt1 = w(:,1:Dimension)'*TestOth;

 % classify
[TrueFalse]=classifyoth(proj1,projt1,NumClasses,ImgsPerClass,NumClassesTest,ImgsP
erClassTest);

 % calculate Percent recognition
 RecogntionRate_TwoDLDApfiherface(Dimension)= ((True*100)/(True + False))/100;
 proj1 =[];
 projt1 = [];
end

% save RecogntionRate_TwoDLDApfisherface.mat RecogntionRate_TwoDLDApfisherface
plot(1:1:Dimension,RecogntionRate_TwoDLDApfisherface,'m-x')
clear True False RecogntionRate_TwoDLDApfisherface TrainOth TestOth
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------การทดลองที่ 3 เปรียบเทียบวิธี 2DLDA กับ Eigenface Fisherface 2DPCA----------
% Define Variable
clear
NumClasses = 40;
ImgsPerClass = 5;
NumClassesTest = 40;
ImgsPerClassTest = 5;
% Load face from ORL Data base
% Select  Frist Five Face by Random
Loadfaceff;

%----------------------------------------Run program Eigenface ----------------------------------------
load Other.mat TrainOth
Calculate Eigenvactor and Eigenvalue
Calculate Projection Data
tic
[u mn msc] = eigenface (TrainOth);
toc
% clear TrainOth
load Other.mat TestOth
   for i = 1:NumClassesTest*ImgsPerClassTest
      mst(:,i) = TestOth(:,i) - mn;
   end

clear TestOth
for Dimension = 1:NumClasses*ImgsPerClass -1

% projection data
   fprintf('projection data to %d Dimension\n',Dimension)
      proj=u(:,1:Dimension)'*msc;
      projt = u(:,1:Dimension)'*mst;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% classify
  [True False] = classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
   RecogntionRate_Eigenface(Dimension)= ((True*100)/(True + False))/100;
   proj=[];
   projt=[];
end
   max(RecogntionRate_Eigenface)
   save RecognitionEigenface.mat RecogntionRate_Eigenface
   figure(1)
   plot(1:1:Dimension,RecogntionRate_Eigenface,'g-+')
clear True False RecogntionRate_Eigenface  mst msc
%----------------------------------------------------------------------------------------------------

%-------------------------------------Run program Fisherface-----------------------------------------
 load Other.mat TrainOth

% Calculate Eigenvactor and Eigenvalue
tic
   [w] = fisherface(TrainOth,NumClasses,ImgsPerClass);
toc

% Calculate Projection Data
   load Other.mat TestOth

for Dimension = 1:NumClasses-1
       fprintf('projection data to %d  Dimension \n',Dimension)
   for i=1:NumClasses*ImgsPerClass
       proj(:,i) =  w(:,1:Dimension)' * TrainOth(:,i);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end

% projection data test
    for i=1:NumClassesTest*ImgsPerClassTest
        projt(:,i) =  w(:,1:Dimension)' * TestOth(:,i);
    end

% classify
    [True,False] = classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
    RecogntionRate_fisheraface(Dimension)= ((True*100)/(True + False))/100;
    proj=[];
    projt=[];
end
    figure(2)
    plot(1:1:Dimension,RecogntionRate_fisherface,'r-*')
    save RecogntionRate_fisherface.mat RecogntionRate_fisherface
    max(RecogntionRate_fisherface)
    clear True False RecogntionRate_fisherface TrainOth TestOth
%-------------------------------------------------------------------------------------------------

%-----------------------------------Run program 2DPCA-------------------------------------------
load Real.mat Train

% Calculate Eigenvactor and Eigenvalue
tic
[u] = TwoDPCA(Train);
toc

load Real.mat Test
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% Projection data
for Dimension = 1:92
    fprintf('projection data to %d Dimension\n',Dimension)
   for i=1:NumClasses*ImgsPerClass
        proj(:,:,i) = Train(:,:,i)*u(:,1:Dimension);
   end

   for i=1:NumClassesTest*ImgsPerClassTest
        projt(:,:,i) = Test(:,:,i)*u(:,1:Dimension);
   end

% classify
   [True,False] = classify(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
    RecogntionRate_TwoDPCA(Dimension)= ((True*100)/(True + False))/100;
    proj =[];
    projt =[];
end

   save RecogntionRateTwoDPCA.mat RecogntionRate_TwoDPCA
   Figure(3)
   plot(1:1:Dimension,RecogntionRate_TwoDPCA,'b-o')
   max(RecogntionRate_TwoDPCA)
   clear True False RecogntionRate_TwoDPCA

%----------------------------------------Run program 2DLDA----------------------------------------
% load Data for train
load Real.mat Train
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% Calculate Eigenvactor and Eigenvalue
tic
[u] = TwoDLDA(Train,NumClasses,ImgsPerClass);
toc
load Real.mat Test

for Dimension = 1:92
% projection data
 fprintf('projection data to %d Dimension\n',Dimension)
  for i=1:NumClasses*ImgsPerClass
       proj(:,:,i) =  real(Train(:,:,i)*u(:,1:Dimension));
 end
 for i=1:NumClassesTest*ImgsPerClassTest
       projt(:,:,i) =  real(Test(:,:,i) * u(:,1:Dimension));
  end

% classify
[True,False] = classify(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
 RecogntionRate_TwoDLDA(Dimension)= ((True*100)/(True + False))/100;
 proj =[];
 projt = [];
end
 save RecogntionRateTwoDLDA.mat RecogntionRate_TwoDLDA
 max(RecogntionRate_TwoDLDA)
 figure(4)
 plot(1:1:Dimension,RecogntionRate_TwoDLDA,'m-x')
 clear True False RecogntionRate_TwoDLDA Train Test
end
%-----------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-----------------------------------การทดลองของฐานข้อมูล AR ------------------------------
%-----------------------------------การทดลองที่ 1 Variation Over Time--------------------------
% Database from AR
% Load input form FaceDatabase Folder into C
% Folder of face database is C:\MATLAB7\work\FaceDatabase\AR\Over_Time
% methode – Variation Over Time
% ImgsPerClass =  image per class
% Numc =  number of calss
% NumClasses = number of image within class
%.....
% Define Variable
Clear
NumClasses = 120;
ToltalImage = 14;
ImgsPerClass = 7;
session = 2;
IntTest = 0; IntTrain = 0;
IntTestOth = 0; IntTrainOth = 0;
fprintf('...train class %d....\n',i)
for i=1: NumClasses
        fprintf('...train class %d....\n',i)
    for j=1: ImgsPerClass
sname = sprintf('C:\\MATLAB7\\work\\FaceDatabase\\AR\\Over_Time\\
subject%03d\\session1_%d.bmp',i,j);
          fprintf('%s\n',sname)
            % image Train Other
              IntTrainOth = IntTrainOth+1;
              TrainOth(:,IntTrainOth) = reshape(double(imread(sname)),50*40,1)
            % image Train Real
              IntTrain = IntTrain+1;
              Train(:,:,IntTrain) = double(imread(sname));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end
        fprintf('...Test class %d....\n',i)
        for k=1: ImgsPerClass
sname = sprintf('C:\\MATLAB7\\work\\FaceDatabase\\AR\\Over_Time\\
subject%03d\\session2_%d.bmp',i,j);
            fprintf('%s\n',sname)

            % Image Test Other
                IntTestOth = IntTestOth+1;
                TestOth(:,IntTestOth) = reshape(double(imread(sname)),50*40,1);

            % Image Test Real
                IntTest = IntTest+1;
                Test(:,:,IntTest) = double(imread(sname));
        end
    save Other.mat TrainOth TestOth
    save Real.mat Train Test

clear sname IntTrainOth IntTestOth InTrain IntTest Test Train TrainOth TestOth
%-------------------------------------------------------------------------------------------------------------

%-----------------------------------------Run program Fisherface-----------------------------------------
%Define Variable
NumClassesTest = 120;
ImgsPerClassTest = 7;

load Other.mat TrainOth

% Calculate Eigenvactor and Eigenvalue
tic
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   [w] = fisherface(TrainOth,NumClasses,ImgsPerClass);
toc

% Calculate Projection Data
   load Other.mat TestOth

for Dimension = 1:NumClasses-1
       fprintf('projection data to %d  Dimension \n',Dimension)
   for i=1:NumClasses*ImgsPerClass
       proj(:,i) =  w(:,1:Dimension)' * TrainOth(:,i);
   end

% projection data test
   for i=1:NumClassesTest*ImgsPerClassTest
        projt(:,i) =  w(:,1:Dimension)' * TestOth(:,i);
   end

% classify
   [True,False] = classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
  RecogntionRate_Fisheraface_Over_Time(Dimension)=((True*100)/(True + False))/100;
   proj=[];
   projt=[];
end
   figure(1)
   plot(1:1:Dimension,RecogntionRate_Fisheraface_Over_Time,'r-*')
   max(RecogntionRate_Fisheraface_Over_Time)
   save RecognitionRate_Fisheraface_Over_Time.mat RecognitionRate__Fisheraface_Over_Time
   clear True False RecogntionRate_Fisheraface_Over_Time TrainOth TestOth
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------------------------------------Run program 2DLDA----------------------------------------
% load Data for train
load Real.mat Train
% Calculate Eigenvactor and Eigenvalue
tic
[u] = TwoDLDA(Train,NumClasses,ImgsPerClass);
toc
load Real.mat Test

for Dimension = 1:92
% projection data
 fprintf('projection data to %d Dimension\n',Dimension)
  for i=1:NumClasses*ImgsPerClass
       proj(:,:,i) =  real(Train(:,:,i)*u(:,1:Dimension));
 end
 for i=1:NumClassesTest*ImgsPerClassTest
       projt(:,:,i) =  real(Test(:,:,i) * u(:,1:Dimension));
  end

% classify
[True,False] =
classify(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
 RecogntionRate_TwoDLDA_Over_Time(Dimension) = ((True*100)/(True + False))/100;
 proj =[];
 projt = [];
end
 max(RecogntionRate_TwoDLDA_Over_Time)
 figure(2) plot(1:1:Dimension,RecogntionRate_TwoDLDA_Over_Time,'m-x')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
save RecognitionRate_TwoDLDA_Over_Time.mat
RecognitionRate__TwoDLDA_Over_Time
 clear True False RecogntionRate_TwoDLDA_Over_Time Train Test
%-------------------------------------------------------------------------------------------------------------
```

%----------การทดลองที่ 2 Variation Facial Expressions----------

```
% Database from AR
% Load input form FaceDatabase Folder into C
% Folder of face database is C:\MATLAB7\work\FaceDatabase\AR\Facial Expressions
% methode – Variation Facial Expressions
% ImgsPerClass =  image per class
% NumClasses =  number of class
% Image = number of image within class
%.....
% Define Variable
Clear
NumClasses = 120;
TotalImage = 4;
session=2;
ImgsPerClass = 2;
IntTest = 0; IntTrain = 0;
IntTestOth = 0; IntTrainOth = 0;
fprintf('...train class %d....\n',i)
for i=1: NumClasses
        fprintf('...train class %d....\n',i)
   for j=1:session
sname = sprintf('C:\\MATLAB7\\work\\FaceDatabase\\AR\\Facial_Expressions\\
subject%03d\\session%d_1.bmp',i,j);
          fprintf('%s\n',sname)
             % image Train Other
               IntTrainOth = IntTrainOth+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            TrainOth(:,IntTrainOth) = reshape(double(imread(sname)),50*40,1)
        % image Train Real
            IntTrain = IntTrain+1;
            Train(:,:,IntTrain) = double(imread(sname));
  end %for j

      fprintf('...Test class %d....\n',i)
for l=1:session
   for k=2: TotalImage
sname =
sprintf('C:\\MATLAB7\\work\\FaceDatabase\\AR\\subject%03d\\session%d_%d.bmp',i,l,k);
     fprintf('%s\n',sname)

        % Image Test Other
            IntTestOth = IntTestOth+1;
            TestOth(:,IntTestOth) = reshape(double(imread(sname)),50*40,1);

        % Image Test Real
            IntTest = IntTest+1;
            Test(:,:,IntTest) = double(imread(sname));
   end %for k
end %for l
end %%for i
     save Other.mat TrainOth TestOth
     save Real.mat Train Test

 clear sname IntTrainOth IntTestOth InTrain IntTest Test Train TrainOth TestOth
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-----------------------------------Run program Fisherface-----------------------------------
%Define Variable
NumClassesTest = 120;
ImgsPerClassTest = 6;

load Other.mat TrainOth

% Calculate Eigenvactor and Eigenvalue
tic
   [w] = fisherface(TrainOth,NumClasses,ImgsPerClass);
toc

% Calculate Projection Data
   load Other.mat TestOth

for Dimension = 1:NumClasses-1
       fprintf('projection data to %d  Dimension \n',Dimension)
   for i=1:NumClasses*ImgsPerClass
       proj(:,i) =  w(:,1:Dimension)' * TrainOth(:,i);
   end

% projection data test
   for i=1:NumClassesTest*ImgsPerClassTest
       projt(:,i) =  w(:,1:Dimension)' * TestOth(:,i);
   end

% classify
   [True,False] = classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
RecogntionRate_Fisheraface_Facial_Expressions (Dimension)= ((True*100)/(True + False))/100;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    proj=[];
    projt=[];
end

save RecognitionRate_Fisheraface_Facial_Expressions.mat RecognitionRate__Fisheraface_Facial _Expressions
   figure(1)
    plot(1:1:Dimension,RecogntionRate_Fisheraface_Facial_Expressions,'r-*')
    max(RecogntionRate_Fisheraface_Facial_Expressions)
    clear True False RecogntionRate_Fisheraface_Over_Time TrainOth TestOth
%-------------------------------------------------------------------------------------------------------------------

%-----------------------------------------Run program 2DLDA-----------------------------------------
% load Data for train
load Real.mat Train

% Calculate Eigenvactor and Eigenvalue
tic
[u] = TwoDLDA(Train,NumClasses,ImgsPerClass);
toc
load Real.mat Test

for Dimension = 1:92
% projection data
  fprintf('projection data to %d Dimension\n',Dimension)
   for i=1:NumClasses*ImgsPerClass
        proj(:,:,i) =  real(Train(:,:,i)*u(:,1:Dimension));
  end
  for i=1:NumClassesTest*ImgsPerClassTest
        projt(:,:,i) =  real(Test(:,:,i) * u(:,1:Dimension));
   end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% classify
[True,False] = classify(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
 RecogntionRate_TwoDLDA_Facial_Expressions(Dimension) = ((True*100)/(True + False))/100;
 proj =[];
 projt = [];
end

save RecognitionRate_TwoDLDA_Facial_Expressions.mat
max(RecogntionRate_TwoDLDA_Facial_Expressions)
figure(2)
plot(1:1:Dimension,RecogntionRate_TwoDLDA_Facial_Expressions,'m-x')
RecognitionRate__TwoDLDA_Facial_Expressions
 clear True False RecogntionRate_TwoDLDA_Facial_Expressions Train Test
%-------------------------------------------------------------------------------------------------------

%----------------------------------การทดลองที่ 3 Variation Lighting-----------------------------------
% Database from AR
% Load input form FaceDatabase Folder into C
% Folder of face database is C:\MATLAB7\work\FaceDatabase\AR\Lighting
% methode – Variation Lighting
% ImgsPerClass =  image per class
% NumClasses =  number of class
% Image = number of image within class
%.....
% Define Variable
Clear
NumClasses = 120;
TotalImage = 4;
session=2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ImgsPerClass = 2;
IntTest = 0; IntTrain = 0;
IntTestOth = 0; IntTrainOth = 0;
fprintf('...train class %d....\n',i)
for i=1: NumClasses
        fprintf('...train class %d....\n',i)
        random = randperm(TotalImage);
        random1 = randperm(TotalImage);
        use(1) = random(1);
        use(2) = random1(1);
        uset(1:3) = random(1:3);
        uset(3:6) = random1(1:3);
   for j=1:session
sname = sprintf('C:\\MATLAB7\\work\\FaceDatabase\\AR\\Facial_Expressions\\subject%03d\\...
session%d_%d.bmp',i,j,use(j));
         fprintf('%s\n',sname)
           % image Train Other
             IntTrainOth = IntTrainOth+1;
             TrainOth(:,IntTrainOth) = reshape(double(imread(sname)),50*40,1)
           % image Train Real
             IntTrain = IntTrain+1;
             Train(:,:,IntTrain) = double(imread(sname));
   end %for j

        fprintf('...Test class %d....\n',i)
for l=1:session
    for k=1: TotalImage*session
sname = sprintf('C:\\MATLAB7\\work\\FaceDatabase\\AR\\subject%03d\\session%d_%d.bmp',i,l,uset(k));
      fprintf('%s\n',sname)

          % Image Test Other
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            IntTestOth = IntTestOth+1;
            TestOth(:,IntTestOth) = reshape(double(imread(sname)),50*40,1);

        % Image Test Real
            IntTest = IntTest+1;
            Test(:,:,IntTest) = double(imread(sname));
    end %for k
end %for l
end %%for i
    save Other.mat TrainOth TestOth
    save Real.mat Train Test

clear sname IntTrainOth IntTestOth InTrain IntTest Test Train TrainOth TestOth random
clear random1 use uset
%-------------------------------------------------------------------------------------------

%-----------------------------------Run program Fisherface----------------------------------
%Define Variable
NumClassesTest = 120;
ImgsPerClassTest = 6;

load Other.mat TrainOth

% Calculate Eigenvactor and Eigenvalue
tic
  [w] = fisherface(TrainOth,NumClasses,ImgsPerClass);
toc

% Calculate Projection Data
  load Other.mat TestOth
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for Dimension = 1:NumClasses-1
    fprintf('projection data to %d  Dimension \n',Dimension)
  for i=1:NumClasses*ImgsPerClass
    proj(:,i) =  w(:,1:Dimension)' * TrainOth(:,i);
  end

% projection data test
  for i=1:NumClassesTest*ImgsPerClassTest
    projt(:,i) =  w(:,1:Dimension)' * TestOth(:,i);
  end

% classify
  [True,False] = classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
RecogntionRate_Fisheraface_Lighting (Dimension)= ((True*100)/(True + False))/100;
  proj=[];
  projt=[];
end
  figure(1)
  plot(1:1:Dimension,RecogntionRate_Fisheraface_Lighting,'r-*')
  max(RecogntionRate_Fisheraface_Lighting)
save RecognitionRate_Fisheraface_Lighting.mat RecognitionRate__Fisheraface_Lighting
  clear True False RecogntionRate_Fisheraface_Lighting TrainOth TestOth
%-------------------------------------------------------------------------------------------------------------

%----------------------------------------Run program 2DLDA-----------------------------------------
% load Data for train
load Real.mat Train

% Calculate Eigenvactor and Eigenvalue
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tic
[u] = TwoDLDA(Train,NumClasses,ImgsPerClass);
toc
load Real.mat Test

for Dimension = 1:92
% projection data
 fprintf('projection data to %d Dimension\n',Dimension)
  for i=1:NumClasses*ImgsPerClass
       proj(:,:,i) =  real(Train(:,:,i)*u(:,1:Dimension));
 end
 for i=1:NumClassesTest*ImgsPerClassTest
       projt(:,:,i) =  real(Test(:,:,i) * u(:,1:Dimension));
  end

% classify
[True,False] =
classify(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);

% calculate Percent recognition
 RecogntionRate_TwoDLDA_Lighting(Dimension) = ((True*100)/(True + False))/100;
 proj =[];
 projt = [];
end
 max(RecogntionRate_TwoDLDA_Lighting)
 figure(2)
plot(1:1:Dimension,RecogntionRate_TwoDLDA_Lighting,'m-x')
save RecognitionRate_TwoDLDA_Lighting.mat  RecognitionRate__TwoDLDA_Lighting
 clear True False RecogntionRate_TwoDLDA_Lighting Train Test
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------การทดลองของฐานข้อมูล YALE ------------------------------
%-------------------------------------การทดลองที่ 1 leaving one out----------------------------
%-------------------------------------Load all face form Yale database-------------------------
% Load input form FaceDatabase Folder into C
% Folder of face database is C:\MATLAB7\work\FaceDatabase\yalefaces\
% methode - leaving - one - out
% Database from Yale
% Imagec =  image per class
% Numc =  number of calss
% Image = number of image within class
%.....................................................................
Numc = 15;
Imagec = 11;
fprintf('...train class %d....\n',i)
for i=1:Numc
  for j=1:Imagec

sname=sprintf('C:\\MATLAB7\\work\\FaceDatabase\\yalefaces\\subject%d_%02d.bmp',i,j);
   fprintf('%s\n',sname)
   % image TemTrain
    TemOthTrain(:,(i-1)*Imagec+j) = reshape(double(imread(sname)),100*80,1);
    TemTrain(:,:,(i-1)*Imagec+j) = double(imread(sname));
  end
end
save Other.mat TemOthTrain
save Real.mat TemTrain
clear all
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%---------------------------------------------- Run program PCA----------------------------------------
% Load face from Yale Data base
% Select Leaving - one - out
clear
ImgsTotal = 165;
ImgsTrain = 164;
ImgsTest = 1;
load Other.mat TemOthTrain ;

for class = 1:15
   for imageperclass = 1:11
%-----------------------------------------------------
% load image for Train and Test
OthTrain = TemOthTrain ;
OthTrain(:,(class-1)*11 + imageperclass)=[];
OthTest = TemOthTrain(:,(class-1)*11 + imageperclass);

%-----------------------------------------------
% Calculate Eigenvactor and Eigenvalue

[u mn msc] = eigenface(OthTrain);
%-----------------------------------------------
% test image subtrace mean
mst = OthTest - mn ;

%-------------------------------------------------------
clear TestOth
for Dimension = 1:154

% projection data
   fprintf('projection data to %d Dimension\n',Dimension)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    proj = u(:,1:Dimension)'*msc;
    projt = u(:,1:Dimension)'*mst;
%-------------------------------------------------
% classify
    for i = 1:ImgsTrain
        x =  proj(:,i);
        y =  projt;
        d =  x - y;
        distance(i) = norm(d,'fro');
    end
    [a b] = min(distance);
    distance =[];
    minm = (class-1)*11 +1 ;
    maxm = (class*11)-1 ;
    if  b >= minm & b <= maxm
       Image((class-1)*11 + imageperclass,Dimension)= 1;
    else
       Image((class-1)*11 + imageperclass,Dimension)= 0;
    End
%-------------------------------------------------
    proj=[];
    projt=[];
end % end Dimension
end % end image per calss
end % end classess
RecognitionRate_Eigenface = sum(Image')/165;
max(RecognitionRate_Eigenface)
save RecognitionRate_Eigenface.mat RecognitionRate_Eigenface
plot(1:154,RecogntionRate_Eigenface,'m-x')
clear
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%---------------------------------------- Run program Fisherface----------------------------------------
% Run program Fisherface
% Load face from Yale Data base
% Select Leaving - one - out
clear
ImgsTotal = 165;
ImgsTrain = 164;
ImgsTest = 1;
load Other.mat TemOthTrain ;

for class = 1:15
   for imageperclass = 1:11
%-------------------------------------------------
% load image for Train and Test
OthTrain = TemOthTrain ;
OthTrain(:,(class-1)*11 + imageperclass)=[];
OthTest = TemOthTrain(:,(class-1)*11 + imageperclass);

%-------------------------------------------------
% Calculate Eigenvactor and Eigenvalue

[w msc mn] = fisherface(Train,15,11,class);
%-------------------------------------------------
% test image subtrace mean
mst = OthTest - mn ;

%-------------------------------------------------
clear TestOth
for Dimension = 1:154
% projection data
   fprintf('projection data to %d Dimension\n',Dimension)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    proj = u(:,1:Dimension)'*msc;
    projt = u(:,1:Dimension)'*mst;
%-------------------------------------------------------
% classify
    for i = 1:ImgsTrain
        x =  proj(:,i);
        y =  projt;
        d =  x - y;
        distance(i) = norm(d,'fro');
    end
    [a b] = min(distance);
    distance =[];
    minm = (class-1)*11 +1 ;
    maxm = (class*11)-1 ;

    if  b >= minm & b <= maxm
        Image((class-1)*11 + imageperclass,Dimension)=  1;
    else
        Image((class-1)*11 + imageperclass,Dimension)= 0 ;
    end
%-------------------------------------------------------
    proj=[];
    projt=[];
 end % end Dimension
end % end image per calss
end % end classess
RecognitionRate_Fisherface = sum(Image')/165;
max(RecognitionRate_Fisherface)
save RecognitionRate_Fisherface.mat RecognitionRate_Fisherface
plot(1:154,RecogntionRate_fisherface,'r-*')
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------------------------------------Run program 2DPCA----------------------------------
% Load face from Yale Data base
% Select Leaving - one - out
clear
ImgsTotal = 165;
ImgsTrain = 164;
ImgsTest = 1;
load real.mat TemTrain ;

for class = 1:15
   for imageperclass = 1:11
fprintf('class %d\n',class)
%-------------------------------------------------
% load image for Train and Test
Train = TemTrain ;
Train(:,:,(class-1)*11 + imageperclass)=[];
Test = TemTrain(:,:,(class-1)*11 + imageperclass);

%-------------------------------------------------
% Calculate Eigenvactor and Eigenvalue
[u] = TwoDPCA(Train);

%-------------------------------------------------
for Dimension = 1:92
   % projection data
   fprintf('projection data to %d Dimension\n',Dimension)
  for i=1:ImgsTrain
        proj(:,:,i) = Train(:,:,i)*u(:,1:Dimension);
  end
        projt = Test*u(:,1:Dimension);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------------------------
% classify
   for i = 1:ImgsTrain
       x =  proj(:,:,i);
       y =  projt;
       d =  x - y;
       distance(i) = norm(d,'fro');
   end
   [a b] = min(distance);
   distance =[];
   minm = (class-1)*11 +1 ;
   maxm = (class*11)-1 ;

   if  b >= minm & b <= maxm
       Image2DPCA((class-1)*11 + imageperclass,Dimension)=  1;
   else
       Image2DPCA((class-1)*11 + imageperclass,Dimension)= 0 ;
   end
%-------------------------------------------------------
   proj=[];
   projt=[];
end % end Dimension
end % end image per calss
end % end classess
RecognitionRate_2DPCA = sum(Image')/165;
max(RecognitionRate_2DPCA)
save RecognitionRate_2DPCA.mat RecognitionRate_2DPCA
plot(1:154,RecogntionRate_2DPCA,'g-+')
clear
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-----------------------------------------Run program 2DLDA-----------------------------------------
% Run program 2DLDA
% Load face from Yale Data base
% Select Leaving - one - out
clear
ImgsTotal = 165;
ImgsTrain = 164;
ImgsTest = 1;
load real.mat TemTrain ;

for class = 1:15
   for imageperclass = 1:11
fprintf('class %d\n',class)
%-------------------------------------------------
% load image for Train and Test
Train = TemTrain ;
Train(:,:,(class-1)*11 + imageperclass)=[];
Test = TemTrain(:,:,(class-1)*11 + imageperclass);

%-------------------------------------------------
% Calculate Eigenvactor and Eigenvalue
[u] = TwoDLDA(Train,15,11,class);

%-------------------------------------------------------
for Dimension = 1:92
% projection data
   fprintf('projection data to %d Dimension\n',Dimension)
  for i=1:ImgsTrain
        proj(:,:,i) = Train(:,:,i)*u(:,1:Dimension);
  end
        projt = Test*u(:,1:Dimension);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%-------------------------------------------------------
% classify
   for i = 1:ImgsTrain
       x =  proj(:,:,i);
       y =  projt;
       d =  x - y;
       distance(i) = norm(d,'fro');
   end
   [a b] = min(distance);
   distance =[];
   minm = (class-1)*11 +1 ;
   maxm = (class*11)-1 ;

   if  b >= minm & b <= maxm
        Image2DLDA((class-1)*11 + imageperclass,Dimension)=  1;
   else
        Image2DLDA((class-1)*11 + imageperclass,Dimension)= 0 ;
   end
%-------------------------------------------------------
   proj=[];
   projt=[];
end % end Dimension
end % end image per calss
end % end classess
RecognitionRate_2DLDA = sum(Image')/165;
max(RecognitionRate_2DLDA)
save RecognitionRate_2DLDA.mat RecognitionRate_2DLDA
plot(1:154,RecogntionRate_2DLDA,'b-^')
clear
%-------------------------------------------------------------------------------------------------------
%-----------------------------------การทดลองที่ 2 Distance-----------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%---------------------------------- Load input form YALE Face Database ----------------------
% Load input form Yale Face Database in Folder
% Random Five for Test
% Folder of face database is C:\MATLAB7\work\FaceDatabase\yalefaces\
%------------- Define variables -------------------------------
% ImgsPerClass =  image per class
% Numc =  number of calss
%.....................
% Defide value
clear
NumClasses = 15;
ImgsPerClass =5;
TotalImagec =11;
IntTest = 0;
IntTrain = 0;
IntTestOth = 0;
IntTrainOth = 0;
ImgsPerClassTest = TotalImagec-ImgsPerClass;

 for i=1: NumClasses
     random = randperm(TotalImagec);
     fprintf('...train class %d....\n',i)
     for j=1: ImgsPerClass
        count  = random(j);

sname=sprintf('C:\\MATLAB7\\work\\FaceDatabase\\yalefaces\\subject%d_%02d.bmp',i,count);
         fprintf('%s\n',sname)

            % image Train Other
               IntTrainOth = IntTrainOth+1;
               TrainOth(:,IntTrainOth) = reshape(double(imread(sname)),100*80,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            % image Train Real
               IntTrain = IntTrain+1;
               Train(:,:,IntTrain) = double(imread(sname));
    end
        fprintf('...Test class %d....\n',i)
    for k=1: ImgsPerClassTest
       count  = random(ImgsPerClass+k);

sname=sprintf('C:\\MATLAB7\\work\\FaceDatabase\\yalefaces\\subject%d_%02d.bmp',i,cont);
        fprintf('%s\n',sname)

            % Image Test Other
               IntTestOth = IntTestOth+1;
               TestOth(:,IntTestOth) = reshape(double(imread(sname)),100*80,1);

            % Image Test Real
               IntTest = IntTest+1;
               Test(:,:,IntTest) = double(imread(sname));

    end
end
      save Other.mat TrainOth TestOth
      save Real.mat Train Test

clear sname IntTrainOth IntTestOth IntTest Test Train TrainOth TestOth ImgsPerClass
clear random ImgsPerClassTest
%-------------------------------------------------------------------------------------------------------------

% Define Variable
clear
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NumClasses = 15;
ImgsPerClass = 5;
NumClassesTest = 15;
ImgsPerClassTest = 6;
% Load face from ORL Data base
% Select  Frist Five Face by Random
Loadfaceff;

%---------------------------------------Run program Eigenface ---------------------------------------
load Other.mat TrainOth
% Calculate Eigenvactor and Eigenvalue
% Calculate Projection Data
tic
[u mn msc ev] = eigenface(TrainOth);
toc
% clear TrainOth
load Other.mat TestOth
   for i = 1:NumClassesTest*ImgsPerClassTest
      mst(:,i) = TestOth(:,i) - mn;
   end

clear TestOth
for Dimension = 1:NumClasses*ImgsPerClass -1

% projection data
   fprintf('projection data to %d Dimension\n',Dimension)
      proj=u(:,1:Dimension)'*msc;
      projt = u(:,1:Dimension)'*mst;
     weights = (ones(1, Dimension)./ ev(:,1:Dimension))';
% classify
[True(1) False(1)] = classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[True(2) False(2)] =classifyothcityblock(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);
[True(3) False(3)] = classifyothMoha(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest, weights);

% calculate Percent recognition
for dis=1:3
   RecogntionRate_Eigenface(dis,Dimension)= ((True(dis)*100)/(True(dis) + False(dis)))/100;
   proj=[];
   projt=[];
end
end
save RecognitionEigenface.mat RecogntionRate_Eigenface
for dis=1:3
   max(RecogntionRate_Eigenface(dis,:))
   figure(dis)
   plot(1:1:Dimension,RecogntionRate_Eigenface(dis,:),'g-+')
end
clear True False RecogntionRate_Eigenface  mst msc dis
%-------------------------------------------------------------------------------------------------------

%-----------------------------------Run program Fisherface-----------------------------------
 load Other.mat TrainOth

% Calculate Eigenvactor and Eigenvalue
tic
   [w,ev] = fisherface(TrainOth,NumClasses,ImgsPerClass);
toc

% Calculate Projection Data
   load Other.mat TestOth

for Dimension = 1:NumClasses-1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fprintf('projection data to %d  Dimension \n',Dimension)
  for i=1:NumClasses*ImgsPerClass
    proj(:,i) =  w(:,1:Dimension)' * TrainOth(:,i);
  end

% projection data test
  for i=1:NumClassesTest*ImgsPerClassTest
      projt(:,i) =  w(:,1:Dimension)' * TestOth(:,i);
  end

weights = (ones(1, Dimension)./ ev(:,1:Dimension))';
% classify
[True(1) False(1)] = classifyoth(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);
[True(2) False(2)] =classifyothcityblock(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);
[True(3) False(3)] = classifyothmaha(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest,weights);

% calculate Percent recognition
for dis=1:3
  RecogntionRate_Fisherface(dis,Dimension)= ((True(dis)100)/(True(dis) + False(dis)))/100;
  proj=[];
  projt=[];
end
end
save RecognitionFisherface.mat RecogntionRate_Fisherface
for dis=1:3
  max(RecogntionRate_Fisherface(dis,:))
  figure(dis)
  plot(1:1:Dimension,RecogntionRate_Fisherface(dis,:),'r-*')
end
  clear True False RecogntionRate_Fisherface TrainOth TestOth dis
%-------------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%----------------------------------------Run program 2DPCA----------------------------------------
load Real.mat Train

% Calculate Eigenvactor and Eigenvalue
tic
[u] = TwoDPCA(Train);
toc

load Real.mat Test

% Projection data
for Dimension = 1:92
    fprintf('projection data to %d Dimension\n',Dimension)
   for i=1:NumClasses*ImgsPerClass
       proj(:,:,i) = Train(:,:,i)*u(:,1:Dimension);
   end

   for i=1:NumClassesTest*ImgsPerClassTest
       projt(:,:,i) = Test(:,:,i)*u(:,1:Dimension);
   end

weights = (ones(size(proj,1),Dimension)/ev(:,1:Dimension))';
% classify
[True(1) False(1)] = classify (proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);
[True(2) False(2)] =classifycityblock(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);
[True(3) False(3)] = classifymaha(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest,weights);

% calculate Percent recognition
for dis = 1:3
   RecogntionRate_TwoDPCA(dis,Dimension)= ((True(dis)*100)/(True(dis) + False(dis)))/100;
    proj =[];
    projt =[];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
end

   save RecogntionRateTwoDPCA.mat RecogntionRate_TwoDPCA
for dis=1:3
    Figure(dis)
    plot(1:1:Dimension,RecogntionRate_TwoDPCA(dis,:),'b-o')
    max(RecogntionRate_TwoDPCA(dis,:))
end
    clear True False RecogntionRate_TwoDPCA dis
%-------------------------------------------------------------------------------------------------------------

%-----------------------------------------Run program 2DLDA-----------------------------------------
% load Data for train
load Real.mat Train

% Calculate Eigenvactor and Eigenvalue
tic
[u] = TwoDLDA(Train,NumClasses,ImgsPerClass);
toc
load Real.mat Test

for Dimension = 1:92
% projection data
 fprintf('projection data to %d Dimension\n',Dimension)
  for i=1:NumClasses*ImgsPerClass
       proj(:,:,i) =  real(Train(:,:,i)*u(:,1:Dimension));
  end
  for i=1:NumClassesTest*ImgsPerClassTest
       projt(:,:,i) =  real(Test(:,:,i) * u(:,1:Dimension));
   end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
weights = (ones(size(proj,1),Dimension)/ev(:,1:Dimension))';
% classify
[True(1) False(1)] = classify(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);
[True(2) False(2)] =classifycityblock(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest);
[True(3) False(3)] = classifymaha(proj,projt,NumClasses,ImgsPerClass,NumClassesTest,ImgsPerClassTest,weights);

% calculate Percent recognition
for dis=1:3
 RecogntionRate_TwoDLDA(dis,Dimension)= ((True(dis)*100)/(True(dis) + False(dis)))/100;
 proj =[];
 projt = [];
end
end
save RecogntionRateTwoDLDA.mat RecogntionRate_TwoDLDA
for dis=1:3
 max(RecogntionRate_TwoDLDA(dis,:))
 figure(dis)
 plot(1:1:Dimension,RecogntionRate_TwoDLDA(dis,:),'m-x')
end
clear True False RecogntionRate_TwoDLDA dis
%-------------------------------------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	สิทธินนท์ คงสนทนา
วัน เดือนปีเกิด	26 กันยายน 2523 ที่สมุทรปราการ
ที่อยู่	70 หมู่ 8 ต. บางกอบัว อ. พระประแดง จ. สมุทรปราการ 10130 โทร 02 - 4610894
ประวัติการศึกษา	มีนาคม 2545 ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า มหาวิทยาลัยศรีนครินทรวิโรฒ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้