แอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคล

Facial Recognition Application on iOS Devices for Identifying People

นางสาวประภัสสร จักรพันธุ์ ณ อยุธยา

นายพงษ์สิทธิ์ น่องเวหา

นางสาวมณฑรัตน์ อูบคุ้ย

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

แอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคล

Facial Recognition Application on iOS Devices for Identifying People

นางสาวประภัสสร จักรพันธุ์ ณ อยุธยา

นายพงษ์สิทธิ์ ผ่องเวหา

นางสาวมณฑรัตน์ อูบตุ้ย

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

Facial Recognition Application on iOS Devices for Identifying People

PRAPASSORN CHAKKAPHAN NA AYUTTHAYA

PONGSIT PONGVEAHA

MONTHARAT OUPTUI

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2013

หัวข้อโครงงานพิเศษ	แอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคล Facial Recognition Application on iOS Devices for Identifying People	
ชื่อนักศึกษา	นางสาวประภัสสร จักรพันธุ์ ณ อยุธยา	53051021
	นายพงษ์สิทธิ์ ผ่องเวหา	53051032
	นางสาวมณฑรัตน์ อูบตุ้ย	53051055
ปริญญา	วิทยาศาสตรบัณฑิต	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	ดร.สายชล ใจเย็น	

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2556

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ธีระ ศิริธีรากุล (ประธานกรรมการ)	
ดร.อดิศักดิ์ สุกุล (กรรมการ)	
ดร.สายชล ใจเย็น (อาจารย์ที่ปรึกษา)	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงงานพิเศษ	แอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคล	
ชื่อนักศึกษา	นางสาวประภัสสร จักรพันธุ์ ณ อยุธยา	53051021
	นายพงษ์สิทธิ์ ผ่องเวหา	53051032
	นางสาวมณฑรัตน์ อูบตุ้ย	53051055
ปริญญา	วิทยาศาสตรบัณฑิต	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2556	
อาจารย์ที่ปรึกษา	ดร.สายชล ใจเย็น	

บทคัดย่อ

แอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS เป็นระบบตรวจหาใบหน้าของบุคคลหนึ่งๆจากภาพถ่ายจากกล้องหรือภาพนิ่ง และนำมาวิเคราะห์เปรียบเทียบภาพใบหน้าที่มีอยู่ในฐานข้อมูล เพื่อระบุตัวบุคคลได้อย่างถูกต้อง แอพพลิเคชันรู้จำใบหน้านี้ทำงานบนระบบปฏิบัติการ iOS แบ่งการทำงานที่สำคัญออกเป็น 2 ขั้นตอน โดยขั้นตอนแรก คือ ขั้นตอนการเรียนรู้ซึ่งจะนำภาพใบหน้าบุคคลมาทำการวิเคราะห์ด้วยเทคนิคการวิเคราะห์องค์ประกอบหลัก(Principal Component Analysis, PCA) เพื่อให้ได้ภาพใบหน้าไอเกนและค่าน้ำหนักของภาพใบหน้า และขั้นตอนที่ 2 คือ ขั้นตอนการรู้จำ เป็นกระบวนการประมวลผลภาพใบหน้าที่ตรวจจับได้ เพื่อทำการค้นหาในฐานข้อมูลที่ได้มาจากขั้นตอนการเรียนรู้ ซึ่งภายในแอพพลิเคชันประกอบด้วยฟังก์ชันหลักในการทำงานดังนี้ การเพิ่มบุคคล การค้นหาบุคคล และรายชื่อบุคคล ผู้ใช้งานสามารถเพิ่มและค้นหาบุคคลโดยใช้ใบหน้าภาพที่ได้จากการถ่ายภาพหรือภาพนิ่ง อีกทั้งยังสามารถตรวจสอบรายชื่อและลบรายชื่อของบุคคลที่ได้ทำการบันทึกในฐานข้อมูล ที่สำคัญแอพพลิเคชันนี้มีความพิเศษที่สามารถระบุตัวบุคคลได้มากกว่าหนึ่งบุคคลขึ้นไป โดยความแม่นยำและความน่าเชื่อถือนั้นจะอยู่ในค่าที่กำหนดจากขั้นตอนการรู้จำ หากค่าอยู่เกินหรือน้อยกว่าขอบเขตที่กำหนด จะไม่สามารถจำภาพนั้นได้

Title	Facial Recognition Application on iOS Devices for Identifying People	
Students	Ms.Prapassorn Chakkaphan Na Ayutthaya	53051021
	Mr.Pongsit Pongveaha	53051032
	Ms.Montharat Ouptui	53051055
Degree	Bachelor of Science	
Major Program	Computer Science	
Academic Year	2556	
Advisor	Saichon Jaiyen, Ph.D.	

ABSTRACT

Facial recognition application on iOS for identifying people is the system which scan people's face image from photo or camera and then compare with the exiting database for correctly identify person. This application works on iOS operating system by dividing the operation into two steps. The first step is the learning operation. It analyzes the person face using principal component analysis (PCA) for creating eigenface and estimating the weight of the face image. The second step is the recognition operation which evaluates the detected face to find data from the previous step. The application has two main functions consist of adding a new contact and finding the contact. Moreover, user can check and delete contact list form the exiting database. Especially, the application can accurately identify the person more than one person and the confidential interval can be determined from the recognition operation. The application cannot recognize image that are not in this interval value.

กิตติกรรมประกาศ

การดำเนินโครงงานพิเศษในการพัฒนาแอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคลสามารถดำเนินมาจนประสบความสำเร็จบรรลุตามวัตถุประสงค์ที่ตั้งเป้าไว้ได้ ทั้งนี้เนื่องจากคณะผู้จัดทำได้รับความอนุเคราะห์และความกรุณารวมถึงได้รับคำแนะนำที่ก่อให้เกิดคุณประโยชน์จากคณาจารย์และบุคลากรหลายท่าน ดังต่อไปนี้

ดร.สายชล ใจเย็น ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการ ที่ได้กรุณาให้ความรู้ คำปรึกษาและคำแนะนำถึงวิธีการในแต่ละขั้นตอนอย่างละเอียดของกระบวนการทำงานในโครงงานพิเศษดังกล่าว โดยเสียสละเวลาอันมีค่าเพื่อชี้แนวทาง การแก้ไขข้อบกพร่องและช่วยตรวจสอบ ซึ่งเป็นกำลังใจที่สำคัญของการทำโครงงานพิเศษนี้มาโดยตลอด ทางคณะผู้จัดทำขอขอบพระคุณถึงความอนุเคราะห์มา ณ โอกาสนี้

ขอขอบพระคุณท่านกรรมการคุมสอบทั้ง 2 ท่าน คือ ผศ.ธีระ ศิริธีรากุลและดร.อดิศักดิ์ สุกุล ที่ได้ให้โอกาสในการนำเสนอโครงงานพิเศษ และชี้แนวทางกระบวนการทำงาน ที่จะเกิดข้อบกพร่องของโครงการ เพื่อเป็นแนวทางให้สามารถพัฒนาโครงการได้อย่างมีประสิทธิภาพและก่อประโยชน์ต่อสังคมมากที่สุด

ขอบคุณสถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้โอกาสได้มาพบแต่สิ่งดี ที่มีสังคมและสิ่งแวดล้อมที่ดี คณาอาจารย์และบุคลากรมีความพร้อมและศักยภาพในการพัฒนาทักษะและมอบความรู้ให้แก่นักศึกษาได้อย่างมีคุณภาพ

รวมถึงขอบคุณทุกท่านที่ไม่ได้กล่าวมา ณ ที่นี้ ที่คอยให้คำปรึกษา ช่วยเหลือและเป็นกำลังใจที่ดีในการดำเนินโครงการที่สามารถสำเร็จผลลุล่วงไปได้ด้วยดี

คณะผู้จัดทำ

มีนาคม 2557

สารบัญ

		หน้า
บทคัดย่อภาษาไทย		I
บทคัดย่อภาษาอังกฤษ		II
กิตติกรรมประกาศ		III
สารบัญ		IV
สารบัญตาราง		VII
สารบัญรูป		VIII
บทที่ 1 บทนำ		1
1.1	ที่มาและความสำคัญ	1
1.2	วัตถุประสงค์ของปัญหาพิเศษ	2
1.3	ข้อจำกัดและขอบเขตของโครงงานพิเศษ	2
1.4	ประโยชน์ที่คาดว่าจะได้รับ	2
1.5	ขั้นตอนการดำเนินการ	3
1.6	อุปกรณ์ที่ใช้ในการทำโครงงานพิเศษ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง		5
2.1	ภาพและการรู้จำใบหน้า	5
2.1.1	ภาพ	5
2.1.2	ภาพดิจิตอล	7
2.1.3	ประเภทของภาพ	8
2.1.3.1	ภาพระดับความเข้มเทา (Intensity Image or Gray Scale Image)	8
2.1.3.2	ภาพสี (Color Image)	9
2.1.3.3	ภาพไบนารี (Binary Image)	9
2.1.3.4	ภาพแบบดัชนี (Index Image)	9
2.1.4	การรู้จำใบหน้า	9
2.1.4.1	การรู้จำรูปแบบ (Pattern recognition)	9
2.1.5	อัลกอริทึมการรู้จำใบหน้า	12
2.1.5.1	ทฤษฎีการวิเคราะห์องค์ประกอบหลัก	12

สารบัญ(ต่อ)

		หน้า
	2.1.6 กระบวนการทำ Histogram	15
2.2	ข้อมูลอุปกรณ์ iOS	20
	2.2.1 เกี่ยวกับไอโฟน	20
	2.2.2 สถาปัตยกรรมของระบบปฏิบัติการ iOS	21
	2.2.3 OpenCV	22
	2.2.4 ชุดพัฒนา iOS SDK	27
	2.2.5 องค์ประกอบของชุดพัฒนา iPhone SDK	28
	2.2.5.1 เครื่องมือ XCode สำหรับสร้างโปรแกรมประยุกต์บน iPhone	28
	2.2.5.2 เครื่องมือ interface Builder สำหรับการออกแบบอินเตอร์เฟซ	29
	2.2.5.3 เครื่องมือ instruments	30
	2.2.5.4 เครื่องมือ simulator	31
	2.2.6 ส่วนประกอบของโปรแกรมบน iOS	32
	2.2.7 ข้อจำกัดของแพลตฟอร์ม	35
	2.2.8 การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming)	38
	2.2.9 หลักการ Model-View-Controller (MVC)	38
	2.2.10 โปรแกรมภาษา Objective-C 2.0	40
	2.2.10.1 ส่วนประกาศ	40
	2.2.10.2 ส่วนระบุวิธีการทำงาน	40
	2.2.10.3 การประกาศเมธอด	41
	2.2.10.4 การประกาศออบเจ็กต์	42
	2.2.10.5 การเรียกเมธอด	42
2.3	ระบบงานที่เกี่ยวข้อง	42
	2.3.1 การเรียนรู้ของเครื่องจักร (Machine Learning)	42
	2.3.1.1 ทำความรู้จักกับการเรียนรู้ของเครื่องจักร	43
	2.3.1.2 Ensemble Learning	45
	2.3.1.3 Nearest Neighbor Classification	46
บทที่ 3	**ขั้นตอนการพัฒนาและออกแบบระบบ**	48

สารบัญ(ต่อ)

		หน้า
3.1	Use Case Diagram	48
3.2	E-R Diagram	52
3.3	User Interface	53
	3.3.1 รูปแบบหน้าเมนูหลัก	53
	3.3.2 รูปแบบหน้าอื่นๆ	54
บทที่ 4	**การพัฒนาแอพพลิเคชันรู้จำใบหน้า**	57
4.1	การสร้างภาพใบหน้าไอเกน	57
4.2	ขั้นตอนการเตรียมรูปภาพที่ใช้ในการเรียนรู้	57
	4.2.1 การตรวจจับใบหน้า	57
	4.2.2 การครอบตัดภาพใบหน้า	57
	4.2.3 การประมวลผลภาพ	58
4.3	ขั้นตอนการเรียนรู้	58
4.4	ขั้นตอนการรู้จำ	59
4.5	การใช้งานแอพพลิเคชัน	60
	4.5.1 การเพิ่มบุคคล	61
	4.5.2 การค้นหาบุคคล	63
	4.5.3 บัญชีรายชื่อบุคคล	64
4.6	การทดสอบ	66
4.7	การประเมินผล	67
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	68
5.1	สรุปผล	68
5.2	ข้อเสนอแนะ	68
เอกสารอ้างอิง		69

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 2.1 ตารางเก็บจำนวนของค่าความสว่าง	17
ตารางที่ 2.2 ตารางค่าความถี่สะสมของค่าความสว่าง	18
ตารางที่ 3.1 คำอธิบายของ U1: เพิ่มรูปภาพและชื่อ	49
ตารางที่ 3.2 คำอธิบายของ U2: ระบุตัวบุคคล	50
ตารางที่ 3.3 คำอธิบายของ U3: รายชื่อบุคคล	51
ตารางที่ 4.1 การประเมินผล	67

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 จุดพิกัดของแต่ละพิกเซลของภาพดิจิตอล	7
รูปที่ 2.2 ขั้นตอนการรู้จำรูปแบบ (Pattern recognition)	11
รูปที่ 2.3 ภาพ 2 มิติขนาด (N x N) เมื่อทำเป็นรูปภาพแบบ 1 มิติ N^2 x 1 (1-D Vector)	12
รูปที่ 2.4 Histogram การแบ่งโทนภาพออกเป็น 256 ตั้งแต่โทน 0-255	15
รูปที่ 2.5 กราฟแสดงความสัมพันธ์ระหว่าง ค่าความสว่าง และจำนวนพิกเซล	16
รูปที่ 2.6 ภาพก่อนและหลังทำ histogram Equalization	16
รูปที่ 2.7 ภาพจำลองโดยมีค่าความสว่างในแต่ละพิกเซล โดยที่ช่องแต่ละช่องแทนพิกเซล	17
รูปที่ 2.8 แสดงภาพจำลองหลังผ่านกระบวนการ Histogram	19
รูปที่ 2.9 ภาพตัวอย่างก่อนและหลังการทำ Histogram Equalization จากตารางข้างต้น	19
รูปที่ 2.10 ภาพก่อนและหลังการทำ Histogram Equalization	19
รูปที่ 2.11 ตัวอย่าง iPhone 5	20
รูปที่ 2.12 ระบบปฏิบัติการ iOS เป็นตัวกลางระหว่างโปรแกรมประยุกต์และฮาร์ดแวร์	21
รูปที่ 2.13 รูปชั้นโครงสร้างของระบบปฏิบัติการ iOS	21
รูปที่ 2.14 หน้าต่างของ Xcode	29
รูปที่ 2.15 หน้าต่างของ Interface Builder	29
รูปที่ 2.16 หน้าต่างของชุด Instruments	30
รูปที่ 2.17 หน้าของ iPhone Simulator	31
รูปที่ 2.18 ภาพรวมการทำงานของหลักการ MVC	39
รูปที่ 2.19 แสดงตัวอย่างระยะทางระหว่างชุดข้อมูล [Jones, 2008]	46
รูปที่ 3.1 Use Case Diagram	48
รูปที่ 3.2 E-R Diagram	52
รูปที่ 3.3 หน้าหลักของแอพพลิเคชัน	53
รูปที่ 3.4 หน้าจอสำหรับเลือกรูปภาพจากคลังรูปภาพ	54
รูปที่ 3.5 หน้าจอสำหรับเลือกรูปภาพจากการถ่ายภาพ	54
รูปที่ 3.6 หน้าจอสำหรับค้นหาบุคคลจากกล้อง	55
รูปที่ 3.7 หน้าจอสำหรับค้นหาบุคคลด้วยภาพ	55
รูปที่ 3.8 หน้าจอแสดงบัญชีรายชื่อ	56

สารบัญรูป(ต่อ)

รูปที่	หน้า
รูปที่ 4.1 ขั้นตอนการเรียนรู้	58
รูปที่ 4.2 ขั้นตอนการรู้จำ	59
รูปที่ 4.3 (ก)ไอคอนแอพพลิเคชัน Face Recognition (ข)แสดงหน้าจอหลัก	60
รูปที่ 4.4 ขั้นตอนการเพิ่มบุคคลจากการเลือก Photo Library	61
รูปที่ 4.5 ขั้นตอนการเพิ่มบุคคลจากการเลือก Take a photo	61
รูปที่ 4.6 ขั้นตอนการบันทึกข้อมูลลงในฐานข้อมูล	62
รูปที่ 4.7 ขั้นตอนการค้นหาบุคคลด้วย Camera	63
รูปที่ 4.8 ขั้นตอนการค้นหาบุคคลด้วย Image	63
รูปที่ 4.9 แสดงบัญชีรายชื่อทั้งหมด	64
รูปที่ 4.10 แสดงข้อมูลของแต่ละบุคคล	64
รูปที่ 4.11 การลบรายการข้อมูล	65
รูปที่ 4.12 การแก้ไขรายการข้อมูล	65
รูปที่ 4.13 ภาพถ่ายใบหน้าบุคคลทั้งหมดที่ทำการทดสอบ	66
รูปที่ 4.14 ภาพใบหน้าที่มีแสงสว่างมากเกินไป	66
รูปที่ 4.15 ภาพใบหน้าที่มีแสงสว่างน้อยเกินไป	66

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

เทคโนโลยีทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น หนึ่งในเทคโนโลยีที่มีความโดดเด่นและเป็นที่น่าสนใจ ได้แก่ การรู้จำใบหน้า (Face Recognition) เป็นงานวิจัยส่วนหนึ่งที่มีความสำคัญในระบบปัญญาประดิษฐ์ โดยใช้หลักการ การวิเคราะห์ภาพใบหน้าเพื่อแยกแยะและระบุตัวบุคคลในภาพ ซึ่งถือเป็นหนึ่งในระบบที่ใช้ในการพิสูจน์ยืนยันตัวบุคคล โดยใช้คุณลักษณะจำเพาะทางสรีระ(Biometric) เป็นรูปแบบการพิสูจน์ตรวจสอบบุคคลที่มีความแม่นยำและความถูกต้อง จึงเห็นได้ว่าการรู้จำใบหน้ามีประโยชน์และมีความสำคัญ ซึ่งสามารถนำมาประยุกต์ใช้ในระบบงานด้านต่างๆ โดยส่วนใหญ่จะเน้นที่การรักษาความปลอดภัย เช่น ใช้ในการระบุตัวบุคคล การทำทะเบียนประวัติคนร้ายของกรมตำรวจโดยใช้ภาพใบหน้า งานรักษาความปลอดภัยในอาคารและสถานที่สำคัญต่างๆ นอกจากนี้ยังรวมไปถึงระบบงานในเชิงพาณิชย์ด้านอื่นๆ ได้แก่ การจัดเก็บฐานข้อมูล,การสร้างสรรค์งานบันเทิง, การแพทย์และสาธารณสุข, การเรียนและการสอน, การสื่อสารโทรคมนาคม เป็นต้น

จะเห็นว่าการใช้เทคโนโลยีการรู้จำใบหน้ามีความสำคัญและเป็นประโยชน์ในชีวิตประจำวัน จึงได้นำเทคโนโลยีนี้มาประยุกต์และพัฒนาเป็นแอพพลิเคชัน ซึ่งเดิมทีแอพพลิเคชันในปัจจุบันที่เกี่ยวกับระบบการรู้จำใบหน้า(Face Recognition) ส่วนใหญ่นั้นเน้นไปทางการจดจำและการค้นหาเพื่อระบุตัวบุคคลเพียงเฉพาะแค่ผู้ใช้งานเพียงคนเดียว การพัฒนาให้ระบบสามารถรู้จำใบหน้าและระบุตัวบุคคลได้หลายคนบนอุปกรณ์มือถือนั้น จะสามารถนำไปใช้ประโยชน์ได้มากกว่าระบบที่จดจำได้เพียงคนเดียว

1.2 วัตถุประสงค์ของปัญหาพิเศษ

1.2.1 เพื่อพัฒนาระบบรู้จำใบหน้าบนมือถือแพลตฟอร์ม iOS

1.2.2 ระบบที่พัฒนาขึ้นสามารถระบุตัวบุคคลได้

1.3 ข้อจำกัดและขอบเขตของโครงงานพิเศษ

1.3.1 พัฒนาแอพพลิเคชันที่ทำงานบนระบบแพลตฟอร์ม iOS

1.3.2 แอพพลิเคชันสามารถเข้าใช้งานได้เพียงหนึ่งคน

1.3.3 ภาพใบหน้าที่ถ่ายจะต้องมีแสงที่มีความสม่ำเสมอไม่มืดที่ด้านใดด้านหนึ่ง

1.3.4 ภาพใบหน้าต้องเป็นภาพใบหน้าตรง

1.3.5 รูปแบบของแอพพลิเคชันบนมือถือแพลตฟอร์ม iOS จะแสดงการจับใบหน้าโดยใช้กล้องถ่ายรูป แล้วนำภาพใบหน้าที่กล้องจับไว้ไปทำการตรวจสอบค้นหา โดยการเปรียบเทียบกับข้อมูลใบหน้าในฐานข้อมูล ซึ่งภายในระบบจะต้องมีฐานข้อมูลเกี่ยวกับบุคคลที่ต้องการค้นหา

1.3.6 ส่วนของผู้ใช้งาน ผู้ใช้งานสามารถเริ่มการใช้งานได้โดย

- เพิ่มภาพและชื่อ จากคลังภาพที่มีอยู่ภายในโทรศัพท์หรือถ่ายรูปจากกล้อง เข้าสู่ฐานข้อมูล
- ตรวจสอบใบหน้าโดยการสแกนกล้องหรือภาพนิ่ง เพื่อค้นหาบุคคลโดยนำไปเปรียบเทียบในฐานข้อมูลที่สร้าง
- บัญชีรายชื่อบุคคล แสดงรายชื่อบุคคลที่ได้ทำการบันทึกลงในฐานข้อมูล สามารถแก้ไขและลบรายชื่อบุคคลได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถตรวจสอบและยืนยันบุคคลได้จากโทรศัพท์มือถือแพลตฟอร์ม iOS ในทุกที่ทุกเวลา

1.4.2 สามารถประยุกต์ใช้กับเทคโนโลยีด้านการรักษาความปลอดภัย

1.4.3 สามารถประยุกต์ใช้ในการระบุตัวบุคคล

1.5 ขั้นตอนการดำเนินการ

1.5.1 วางแผนระบบงาน ทำการวางแผนเพื่อเป็นแนวทางในการดำเนินงานและเข้าใจถึงปัญหาและหลักการการทำงานในแต่ละขั้นตอนของระบบงาน

1.5.2 ศึกษาเอกสารและทฤษฎีที่เกี่ยวข้อง ศึกษาทฤษฎี หลักการต่างๆและซอฟต์แวร์รวมไปถึงอุปกรณ์ต่างๆที่มีส่วนเกี่ยวข้องที่ใช้ในการออกแบบระบบงาน

1.5.3 ออกแบบขั้นตอนการทำงาน นำหลักการและทฤษฎีมาวิเคราะห์ เพื่อออกแบบระบบงาน ให้ระบบงานสามารถทำงานได้อย่างครอบคลุม มีความแม่นยำและถูกต้อง

1.5.4 พัฒนาโปรแกรมระบบงาน นำหลักการที่ทำการวิเคราะห์มาแล้ว เพื่อทำการเขียนโปรแกรมให้ถูกต้องตามระบบงานที่ออกแบบไว้

1.5.5 ทดสอบและติดตั้งงาน การทดสอบการใช้งาของโปรแกรมที่ทำการพัฒนาขึ้น เพื่อทดสอบความสามารถและหาข้อจำกัดและปัญหาที่เกิดขึ้น เพื่อนำไปแก้ไขข้อผิดพลาดทีเกิดขึ้น

1.5.6 ทดลองใช้งานจริง นำระบบที่พัฒนาขึ้นมาใช้งานจริง เพื่อหาจุดบกพร่องที่อาจเกิดขึ้นได้

1.5.7 แก้ไขข้อบกพร่องและปรับปรุงงาน การแก้ไขในส่วนของข้อบกพร่องหรือข้อผิดพลาดต่างๆ ที่ได้จากการทดลองใช้งาน เพื่อให้ระบบมีความสามารถและมีประสิทธิภาพตรงกับความต้องการมากที่สุด

1.5.8 สรุปผลและอภิปรายผล การสร้างเอกสารประกอบการใช้งานของระบบงานที่ทำการพัฒนาขึ้น เอกสารสรุปผลและเอกสารอ้างอิง เพื่อให้เป็นคู่มือและสะดวกต่อผู้ใช้งาน

1.6 อุปกรณ์ที่ใช้ในการทำโครงงานพิเศษ

1.6.1 คอมพิวเตอร์ตั้งโต๊ะหรือคอมพิวเตอร์พกพา

1.6.2 ซอฟต์แวร์ที่เกี่ยวข้อง ดังนี้

- ระบบปฏิบัติการ Mac OS X 10.9.2
- Xcode 5.1 เครื่องมือ(IDE) ในการเขียนโปรแกรมบน Mac
- Simulator เครื่องมือที่ใช้จำลองการทำงานของ iPhone บนเครื่อง Mac
- Cocoa touch เครื่องมือที่ใช้สำหรับสร้างส่วนติดต่อกับผู้ใช้

- Interface Builder เครื่องมือที่ใช้ในการรวมและแสดง User Interface ของโปรแกรมประยุกต์

1.6.3 iPhone4s หรือ iPad Air

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีและงานวิจัยที่เกี่ยวข้องที่ใช้ในงานวิจัยนี้ เกี่ยวกับการตรวจสอบและยืนยันตัวบุคคลโดยอาศัยการรู้จำรูปแบบของใบหน้าที่เป็นเอกลักษณ์อย่างหนึ่งของมนุษย์ การระบุตัวบุคคลด้วยใบหน้านั้นมีความสำคัญอย่างยิ่งและสามารถนำไปประยุกต์ใช้งานด้านต่างๆได้หลากหลาย

ในการระบุตัวบุคคลบนอุปกรณ์ iOS ระบบจะต้องมีภาพใบหน้าของบุคคลนั้นอยู่ในฐานข้อมูลก่อน จากนั้นนำกล้องแสกนใบหน้าบุคคลที่ต้องการระบุตัวบุคคล ระบบจะทำการ detect ภาพใบหน้า เพื่อนำมาเปรียบเทียบกับภาพใบหน้าที่อยู่ในฐานข้อมูล

2.1 ภาพและการรู้จำใบหน้า

2.1.1 ภาพ

โดยปกติแล้ว ข้อมูลภาพทั่วๆไปนั้นได้มาจากการที่แสงตกกระทบกับวัตถุแล้วเกิดการสะท้อนผ่านเลนส์เข้าสู่ตัวบันทึกภาพ อาจจะอยู่ในรูปแบบของตัวตรวจจับ (Sensor) หรือฟิล์ม (Film) หากเราย้อนนึกถึงเมื่อเวลาเราถ่ายภาพด้วยกล้องดิจิตอล วัตถุหรือภาพที่เราเห็นด้วยตานั้นเป็นข้อมูลสามมิติ (Three-dimension) ที่ประกอบไปด้วยความลึก ความสูง และความกว้าง แต่เมื่อเราแปลงข้อมูลภาพออกมาเป็นข้อมูลภาพดิจิตอล (Digital Image) ข้อมูลของภาพนั้นจะประกอบไปด้วยความกว้างและความสูงของภาพ (Width and Height) เท่านั้น ซึ่งในปัจจุบันมีอุปกรณ์อิเล็กทรอนิกส์หลายประเภทที่สามารถสร้างสัญญาณภาพดิจิตอลได้สะดวกและง่ายขึ้น เช่น กล้องถ่ายภาพดิจิตอล กล้องวีดีทัศน์ดิจิตอล เป็นต้น

โดยทั่วไปแล้วเราสามารถที่จะแบ่งรูปภาพที่ปรากฏและใช้งานบนเครื่องคอมพิวเตอร์ออกเป็น 2 ประเภทคือ รูปภาพแบบบิตแมป (Bitmap Image) และรูปภาพแบบเวกเตอร์ (vector Image) โดยรูปภาพแบบบิตแมปจะพิจารณาตัวรูปภาพซึ่งถูกแบ่งออกเป็นส่วนย่อยเล็กๆหลายๆ ส่วนหรือที่เรียกว่าพิกเซล (Pixels) ที่ถูกนำมารวมกันและใช้แสดงผลภาพ ส่วนรูปภาพแบบเวกเตอร์จะประกอบด้วยเส้นลายต่างๆ ที่สร้างขึ้นจากการคำนวณทางคณิตศาสตร์ของลักษณะทางเรขาคณิตเพื่อสร้างรูปทรงต่างๆที่เราเห็น ซึ่งเรียกว่าเวกเตอร์ (vectors)

2.1.1.1 ภาพแบบบิตแมป (Bitmap Image)

ภาพบิตแมปหรือเรียกอีกอย่างหนึ่งว่าภาพแบบราสเตอร์ (Raster Image) เป็นภาพที่ประกอบขึ้นจากจุดขนาดเล็กๆ หรือที่เรียกว่าพิกเซลจำนวนมากที่เรียงต่อกันจนเป็นภาพภาพหนึ่ง เพื่อให้เห็นภาพลักษณะนี้ชัดเจนยิ่งขึ้น ให้นึกถึงการสร้างภาพบนตารางสี่เหลี่ยมเล็กๆ เราจะใช้สีแต้มลงในช่องสี่เหลี่ยมแต่ละช่องจนกลายเป็นภาพสมบูรณ์ที่มีขนาดใหญ่ ซึ่งเหมือนกับการป้ายสีแปลอักษรบนอัฒจันทร์ ภาพแบบบิตแมปนี้จะมีจำนวนจุดเล็กๆ จำนวนมาก ดังนั้นดวงตาของมนุษย์ไม่สามารถที่จะมองเห็นและแยกแยะรายละเอียดส่วนย่อยเล็กๆนั้นได้ แต่เมื่อลองขยายภาพดูจะเห็นเป็นรูปตาราง ยิ่งขยายใหญ่เท่าไร ตารางสี่เหลี่ยมก็ยิ่งมีขนาดใหญ่ขึ้นจนทำให้มองเห็นจุดของภาพหรือพิกเซล จึงมีผลทำให้ไม่สามารถบอกได้ว่าภาพนั้นเป็นภาพอะไร สิ่งนั้นส่งผลเช่นเดียวกันหากเราขยายภาพบิตแมปบนหน้าจอคอมพิวเตอร์ ซึ่งจะมีผลทำให้รายละเอียดของภาพมีความไม่ โดยทั่วไปแล้วภาพบิตแมปเป็นภาพประเภทที่นิยมใช้กันมากในภาพถ่ายหรือภาพวาด เนื่องจากมันสามารถไล่โทนสีและแสงเงาได้เหมือนจริงที่สุด

2.1.1.2 ภาพแบบเวกเตอร์ (Vector Image)

ภาพแบบเวกเตอร์จะมีคุณสมบัติที่แตกต่างกับแบบบิตแมป คือภาพประเภทนี้ไม่ว่าจะขยายให้ใหญ่แค่ไหน ก็ยังคงรายละเอียดและความคมชัดไว้ได้เหมือนเช่นเดิมโดยไม่ผิดเพี้ยน เนื่องจากภาพแบบเวกเตอร์นั้นประกอบด้วยเส้นตรง เส้นโค้ง และรูปทรงต่างๆ ภาพที่ได้จะสร้างขึ้นจากคำสั่งที่บอกถึงลักษณะของในรูปแบบทางเรขาคณิตด้วยสมการทางคณิตศาสตร์ ดังนั้นโปรแกรมที่ต้องการเปิดรูปภาพจะต้องนำสมการต่างๆ ที่บันทึกเอาไว้มาคำนวณและสร้างรูปทรงของภาพขึ้นมาใหม่ จุดเด่นคือไม่ว่าจะขยายภาพให้ใหญ่แค่ไหน คอมพิวเตอร์ก็จะคำนวรค่าต่างๆ ให้ใหม่ทุกครั้ง ทำให้ภาพที่เกิดขึ้นมามีความคมชัด ภาพแบบเวกเตอร์จึงเหมาะกับงานที่มีความแม่นยำและต้องการความละเอียดสูง เช่น การสร้างภาพโลโก้ การสร้างภาพสามมิติ การสร้างแบบร่างทางวิศวกรรม

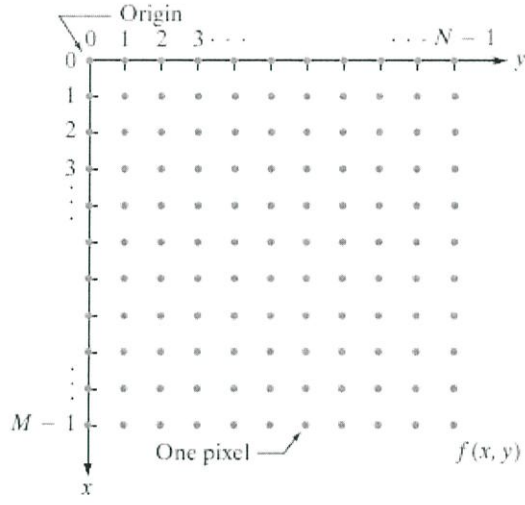
2.1.2 ภาพดิจิตอล

ข้อมูลภาพดิจิตอลระดับความเข้มเทาสามารถที่จะเขียนให้อยู่ในรูปของฟังก์ชันสองมิติของค่าระดับความเข้มแสง โดยค่าของฟังก์ชัน f(x, y) จะแสดงถึงค่าความเข้มแสงที่จุดพิกัดสเปเชียล (Spatial) x, y ใดๆ โดยทั่วไปนิยมเขียนแทนภาพดิจิตอลสองมิติ f(x, y) ขนาด M x N ให้อยู่ในรูปของเมทริกซ์ (Matrix) สองมิติได้ดังสมการที่ 2-1 ส่วนรูปที่ 2.1 แสดงตำแหน่งของจุดพิกัดใดๆ ของภาพดิจิตอลในแต่ละจุดพิกเซลของภาพ

$$f(x,y) = \begin{vmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(0,0) & \dots & f(M-1,N-1) \end{vmatrix} \quad (2\text{-}1)$$

โดยที่ $f(x, y)$ แทนค่าระดับความเข้มเทาที่จุดพิกัด x, y ใดๆ

x, y คือจุดพิกัดของลำดับแถวและหลักของภาพ ตามลำดับ



รูปที่ 2.1 จุดพิกัดของแต่ละพิกเซลของภาพดิจิตอล

คุณภาพของรูปภาพดิจิตอลจะขึ้นอยู่กับจำนวนพิกเซลและจำนวนค่าระดับความเข้มเท่า กล่าวคือ จำนวนพิกเซลในแต่ละแถวและแต่ละหลักของรูปภาพดิจิตอลเป็นจำนวนค่าสุ่มตัวอย่างทั้งหมด หรือเป็นความละเอียดของสัญญาณภาพดิจิตอลนั่นเอง ดังนั้นการลดจำนวนพิกเซลลงจะเป็นการลดความละเอียดของรูปภาพดิจิตอล จึงมีผลทำให้รูปภาพผลลัพธ์ที่ได้อาจผิดเพี้ยนไปจากรูปภาพต้นฉบับได้ ในทำนองเดียวกัน จำนวนค่าระดับของสัญญาณดิจิตอลจะถูกกำหนดด้วยค่าระดับความเข้มเทาที่ใช้ในการแสดงผล

ภาพดิจิตอล นั่นคือคุณภาพของรูปภาพดิจิตอลจะขึ้นอยู่กับจำนวนค่าระดับความเข้มเทาที่ใช้ แต่อย่างไรก็ตาม การเพิ่มพิกเซลหรือจำนวนค่าระดับความเข้มเทาในบางครั้งก็อาจจะไม่เพิ่มคุณภาพของรูปภาพดิจิตอลหรือเพิ่มน้อยมาก

โดยปกติแล้ว ภาพจะดูคมชัดเท่าไรก็ขึ้นอยู่กับว่าภาพนั้นมีจำนวนพิกเซลอยู่มากเท่าใด เรามักจะบอกความละเอียดของภาพ (Resolution) โดยดูจากจำนวนพิกเซลต่อนิ้ว (Pixel/Inch : PPI) หรือพิกเซลต่อตารางนิ้ว (Pixels/Inch^2) หากภาพยิ่งมีจำนวนพิกเซลต่อหนึ่งตารางนิ้วมาก ภาพก็ยิ่งมีความละเอียดเหมือนจริงมากขึ้น โดยภาพที่มีความละเอียดมากๆ จะเรียกว่าภาพนี้มี Resolution สูง ส่วนภาพที่มีความละเอียดน้อยก็เรียกว่าภาพนี้มี Resolution ต่ำ เช่น ภาพขนาดกว้าง 1 นิ้ว ยาว 1 นิ้ว ที่มีความละเอียด 72 PPI จะมีจำนวนพิกเซลทั้งหมด 5184 จุดพิกเซล (72 x 72 = 5184)

2.1.3 ประเภทของภาพ

โดยทั่วไปแล้ว เราสามารถแบ่งประเภทของภาพบิตแมปตามคุณสมบัติการแสดงผลของสีภาพเป็น 4 ประเภทดังนี้คือ

2.1.3.1 ภาพระดับความเข้มเทา (Intensity Image or Gray Scale Image)

ลักษณะของภาพชนิดนี้ ในแต่ละพิกเซลจะมีคาความเข้มแสงในแต่ละระดับที่แตกต่างกันไป ตั้งแต่ระดับเทาดำไปยังระดับสีขาว เราสามารถกำหนดระดับความเข้มแสงนั้นได้โดยใช้ค่าระดับความเข้มเทา (Gray Scale หรือ Gray Level) โดยปกติทั่วๆไป ภาพแบบระดับสีเทาจะมีค่าระดับความเข้มเทาเท่ากับ 8 บิต ดังนั้นค่าความเข้มแสงจะถูกแบ่งออกเป็น 256 ระดับ เมื่อค่าระดับความเข้มเทามีค่าเป็น 0 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงต่ำ จะทำให้จุดภาพเป็นสีดำ ในทางกลับกัน หากค่าระดับความเข้มเท่าเป็น 255 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงมาก จะทำให้จุดภาพเป็นสีขาว ซึ่งสีขาวจะถูกแทนด้วยค่าความเข้มเทาเท่ากับ 255 (11111111) และสีดำจะถูกแทนด้วยค่าความเข้มเทาเท่ากับ 0 (00000000) ส่วนค่าระหว่าง 0-255 ก็จะมีค่าไล่เฉดสีจากสีดำไปหาสีขาวนั่นเอง

2.1.3.2 ภาพสี (Color Image)

ภาพชนิดนี้ แต่ละจุดภาพหรือพิกเซลของภาพจะเก็บค่าระดับความเข้มเทาของแต่ละแถบแสงของแม่สีหลัก 3 สีที่ซ้อนกันอยู่คือ สีแดง (Red) สีเขียว (Green) สีน้ำเงิน (Blue) ซึ่งในแต่ละพิกเซลตามระดับความเข้มในแต่ละแถบสีนั้น

2.1.3.3 ภาพไบนารี (Binary Image)

ภาพไบนารีจะแสดงลักษณะของข้อมูลภาพในรูปแบบขาวดำ กล่าวคือ ในแต่ละพิกเซลของภาพจะถูกแสดงด้วยค่าแบบไบนารี (Binary) คือมี 1 บิต ซึ่งประกอบไปด้วยค่า 1 และ 0 โดยที่ 1 หมายถึงจุดภาพสีดำ ภาพประเภทนี้เหมาะสำหรับภาพที่เกี่ยวกับตัวอักษร (Text) ภาพลายนิ้วมือ (Finger Print) เป็นต้น

2.1.3.4 ภาพแบบดัชนี (Index Image)

ภาพประเภทนี้ ในแต่ละพิกเซลของภาพจะเก็บค่าดัชนี (Index Number) ซึ่งเป็นตัวเลขจำนวนเต็ม จะถูกนำค่าดัชนีดังกล่าวไปเทียบกับตารางสี (Color Table) ซึ่งเป็นตารางแสดงค่าแสงสีแดง สีเขียว และสีน้ำเงิน ซึ่งค่าดัชนีนี้จะเป็นตัวบ่งชี้ให้เห็นว่าภาพในแต่ละตำแหน่งพิกเซลใดๆ มีค่าอัตราส่วนของแม่แสง 3 สีในอัตราส่วนละเท่าไร

2.1.4 การรู้จำใบหน้า

การรู้จำใบหน้าคือการระบุตัวตนโดยการเปรียบเทียบกับฐานข้อมูลภาพใบหน้าที่มีอยู่ว่าตรงกับบุคคลใดส่วนมากแล้วจะใช้ในงานด้านอาชญากรรมเป็นหลัก การรู้จำใบหน้าประกอบด้วยสองส่วนหลักคือ การเรียนรู้ และ การรู้จำ การเรียนรู้คือขั้นตอนของการสอนให้ระบบรู้จักกับบุคคลที่ต้องการป้อนเข้าสู่ระบบ การรู้จำจะเป็นขั้นตอนเปรียบเทียบภาพที่ต้องการทำการสืบค้นกับภาพใบหน้าต่างๆ ที่มีอยู่ในฐานข้อมูล ภาพใบหน้าที่ใช้ในการรู้จำใบหน้าจะต้องเป็นภาพดิจิตอลและนำไปทำการตรวจจับใบหน้า เพื่อครอบตัดเอาเฉพาะส่วนที่เป็นภาพใบหน้าเท่านั้นไปทำการวิเคราะห์ต่อไป

2.1.4.1 การรู้จำรูปแบบ (Pattern recognition)

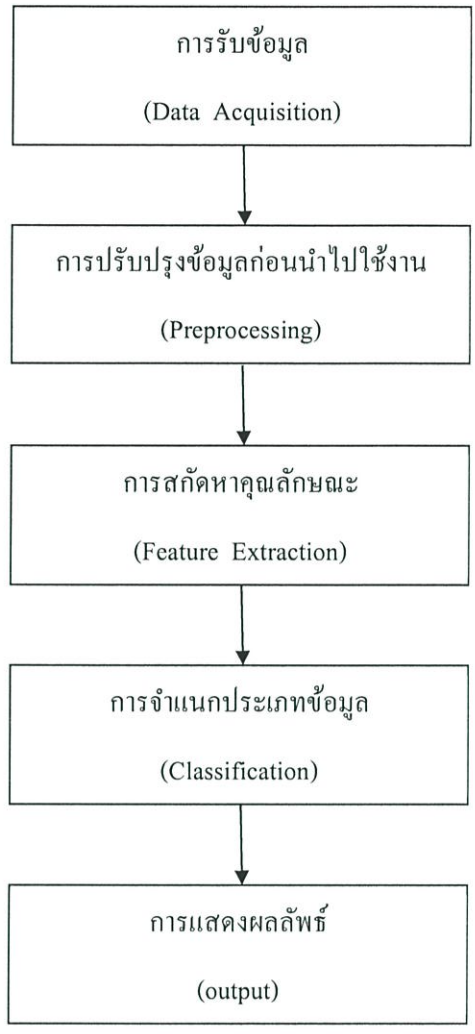
การรู้จำรูปแบบ (Pattern Recognition) เป็นศาสตร์ที่ว่าด้วยกระบวนการตัดสินใจที่เกี่ยวกับการจำแนกกลุ่ม (Classification) การจัดกลุ่ม (Clustering) การรู้จำ (Recognition) ศึกษาแนวคิดต่างๆให้คอมพิวเตอร์สามารถทำงานเหล่านี้ได้โดยใช้เหตุผลหรือคณิตศาสตร์เพื่อหารูปแบบ (Pattern) โดยจะใช้ความรู้ด้านอื่นๆ

มาช่วยในการวิเคราะห์ เป็นวิทยาการที่สามารถประยุกต์ใช้ได้กับงานทุกสาขา และเป็นพื้นฐานสำคัญสำหรับงานวิจัยในด้านปัญญาประดิษฐ์ หรือการสร้างความฉลาดให้คอมพิวเตอร์ กระบวนการเหล่านี้เป็นพื้นฐานที่สำคัญของความฉลาดของมนุษย์ที่ติดตัวมาตั้งแต่แรกเกิด และยังคงเป็นปัญหาที่ยังท้าทายนักวิจัยอยู่ถึงปัจจุบัน และสามารถประยุกต์ใช้ในสาขาอื่นได้อีกมาก

การรู้จำรูปแบบสามารถแบ่งได้เป็น

1. การรู้จำรูปแบบทางสถิติ (Statistic Pattern Recognition) หรือทฤษฎีการตัดสิน (Decision Theory) โดยจะใช้พื้นฐานของทฤษฎีความน่าจะเป็นในการวิเคราะห์
2. การรู้จำรูปแบบสังเคราะห์ (Syntactic Pattern Recognition) หรือ Structural Pattern Recognition (Linguistic Method) โดยจะใช้อัลกอริทึมอื่นๆมาวิเคราะห์

กระบวนการในการรู้จำรูปแบบโดยทั่วไปจะประกอบด้วยขั้นตอนการรับข้อมูล การปรับปรุงข้อมูลก่อนนำเข้าสู่กระบวนการประมวลผล การสกัดลักษณะเด่นของข้อมูลที่ต้องการออกจากข้อมูลอื่น การจำแนกประเภทข้อมูล และการแสดงผลลัพธ์ของการรู้จำ ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ขั้นตอนการรู้จำรูปแบบ (**การรับข้อมูล (Data Acquisition)**

การรับข้อมูลเพื่อนำเข้าสู่กระบวนการรู้จำสามารถทำได้หลายวิธีขึ้นอยู่กับคุณลักษณะของไบโอเมตริกที่ต้องการนำไปใช้งาน ซึ่งในงานวิจัยนี้จะเป็นข้อมูลภาพถ่ายหน้าตรงของใบหน้าแบบสองมิติ

- **การปรับปรุงข้อมูลก่อนนำไปใช้งาน (Preprocessing)**

ข้อมูลที่ถูกนำไปใช้คือภาพถ่ายบริเวณใบหน้า ดังนั้นการปรับปรุงข้อมูลก่อนนำไปใช้ในการรู้จำก็คือกระบวนการในการประมวลผลภาพ (Image Processing) ซึ่งภาพที่ได้จากการรับข้อมูลอาจมีทั้งข้อมูลที่ต้องการ ฉากหลัง และสัญญาณรบกวนรวมอยู่ด้วย จึงจำเป็นต้องกำจัดสัญญาณรบกวนนั้นออกจากภาพ หรืออาจเป็นการแปลงภาพเหล่านั้นให้อยู่ในลักษณะที่เหมาะสมต่อการนำไปใช้งาน

- **การสกัดหาคุณลักษณะเฉพาะ (Feature Extraction)**

เป็นขั้นตอนของการสกัดเอาลักษณะเฉพาะของแต่ละภาพใบหน้าอินพุตออกมาเป็นเวกเตอร์เพื่อนำไปใช้เป็นอินพุตในการเรียนรู้ฝึกระบบและทดสอบระบบ

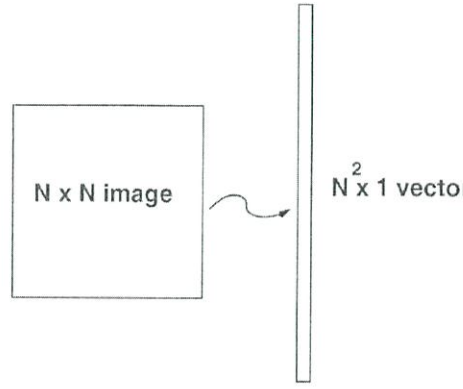
- **การจำแนกประเภทข้อมูล (Classifications)**

เป็นขั้นตอนในการจำแนกและตัดสินใจว่าอินพุตที่เข้ามานั้นเป็นภาพของใคร โดยในขั้นตอนนี้มีหลายวิธีด้วยกัน เช่น Nearest Neighbor, การใช้โครงข่ายประสาทเทียม (Neural Network) เป็นต้น ซึ่งในงานวิจัยนี้จะใช้เทคนิค Nearest Neighbor ในการรู้จำ

2.1.5 อัลกอริทึมการรู้จำใบหน้า

2.1.5.1 ทฤษฎีการวิเคราะห์องค์ประกอบหลัก

วิธีการวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis-PCA) เป็นวิธีการทางสถิติ ซึ่งถูกนำไปประยุกต์ใช้ในงานต่าง ๆ เช่น การบีบอัดข้อมูลและการสร้างภาพใบหน้าไอเกนเพื่อใช้ในระบบรู้จำ เป็นต้น วิธีนี้ถูกนำมาสร้างเป็นภาพใบหน้าไอเกน (Eigen Faces) ครั้งแรกโดย Sirovich และ Kirby ในปี 1987 และต่อมา Turk และ Pentland ได้สร้างระบบรู้จำใบหน้าโดยการนำภาพใบหน้าไอเกนในการรู้จำโดยกำหนดให้ I(x,y) คือภาพที่ใช้เป็น (2-D Image) 2 มิติ มีขนาด 80 x 80 Pixel ขนาด (N x N) ชนิด สีเทา เมื่อทำเป็นรูปภาพแบบ 1 มิติ N^2 x 1 (1-D Vector) มีความเท่ากับ 6,400 จุด ต่อ1 ภาพ เพื่อทำเป็นชุด Training Set Image จากภาพใบหน้าต้นฉบับ



รูปที่ 2.3 ภาพ 2 มิติขนาด (N x N) เมื่อทำเป็นรูปภาพแบบ 1 มิติ N^2 x 1 (1-D Vector)

โดยกำหนดให้ T1, T2, T3, ..., TM เป็นกลุ่มเรียนรู้ ของภาพใบหน้า การสร้างกลุ่มเรียนรู้ จะคำนวณค่าเฉลี่ย ค่าโควาเรียนเมตริกซ์และคำนวณค่าไอเกนเวกเตอร์ตามลำดับ มีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 การคำนวณหาค่าเฉลี่ย (Ψ =Average Face) จากทุกใบหน้า ดังสมการที่ 2-2

$$\Psi = \frac{1}{\mathrm{M}}\sum_{1}^{\mathrm{M}} \mathrm{T}_i \quad (2\text{-}2)$$

โดยที่ Ψ คือ ใบหน้าเฉลี่ย

M คือ จำนวนใบหน้า

T คือ กลุ่มเรียนรู้

ขั้นตอนที่ 2 คำนวณหาผลต่างระหว่างกลุ่มเรียนรู้และค่าเฉลี่ย หรือเรียกว่าค่าเบี่ยงเบนมาตรฐานของภาพใบหน้า ดังสมการที่ 2-3

$$\Phi = \mathrm{T}_1 - \Psi \quad (2\text{-}3)$$

โดยที่ Ψ คือ ใบหน้าเฉลี่ย

Φ คือ ใบหน้าผลต่าง

T คือ กลุ่มเรียนรู้

ขั้นตอนที่ 3 คำนวณหาค่าเมตริกซ์ความแปรปรวนร่วม โดยนำค่าเบี่ยงเบนมาตรฐานจากขั้นตอนที่ 2 แล้ว คูณกับค่าเบี่ยงเบนมาตรฐานที่ทำการทรานโพส ดังสมการที่ 2-4

$$C = \frac{1}{\mathrm{M}}\sum_{1}^{\mathrm{M}} \Phi_i \Phi_i^r \quad (2\text{-}4)$$

$$= AA^T$$

โดยที่ A คือ $[\Phi_1, \Phi_2, \ldots, \Phi_{\mathrm{M}}]$ เป็นชุดข้อมูลค่าเบี่ยงเบนมาตรฐาน

C คือ เมตริกซ์ความแปรปรวนร่วม

A คือ กลุ่มของเมตริกซ์ความแปรปรวนร่วม

ขั้นตอนที่ 4 คำนวณหา ไอเกนเวคเตอร์ (v) และค่าไอเกน(μ) ของเมตริกซ์ความแปรปรวนร่วม เนื่องจากเมตริกซ์ C มีขนาด N^2 x N^2 มิติ นำมาสร้างเป็นไอเกนเวคเตอร์ ขนาด N^2 และค่าไอเกน ขนาด N^2 ด้วยรูปภาพ ขนาด 80 x 80 พิกเซล

จะไม่สามารถกระทำได้ซึ่ง N^2 จะได้เมตริกซ์ที่มีขนาดใหญ่มาก ซึ่งวิธี PCA เป็นการลดมิติข้อมูลลง จึงทำให้ได้ผลลัพธ์มีขนาดมากเกินไป และอาจจะมีไอเกนเวกเตอร์บางค่าที่สัมพันธ์กับค่าไอเกนที่มีค่าเป็นศูนย์ จึงไม่นำมาเป็นข้อมูลในการพิจารณา เพื่อแก้ปัญหานี้ Turk และ Pentland ได้ทำการปรับปรุงขั้นตอนวิธีโดยการเสนอวิธีใหม่ โดยการสร้างเมตริกซ์ A^TA แทนที่จะใช้ AA^T ดังนั้นค่าไอเกนเวคเตอร์ สามารถทำได้ ดังสมการที่ 2-5

$$A^TAv_i = \mu_i v_i \quad (2\text{-}5)$$

โดยที่ A คือ กลุ่มของเมตริกซ์ความแปรปรวนร่วม

μ_i คือ ค่าไอเกน (Eigen value)

v_i คือ เวกเตอร์ไอเกน (Eigen vector)

ขนาดของเมตริกซ์ความแปรปรวนร่วม จะมีขนาด M x M จะได้ค่าไอเกนใหญ่มาก จึงแทนค่าไอเกนปีนขนาด N^2 ซึ่งจะได้สมการใหม่ ดังสมการที่ 2-6

$$AA^TAv_i = \mu_i Av_i \quad (2\text{-}6)$$

ในขั้นตอนนี้จะใช้วิเคราะห์องค์ประกอบหลักโดยใช้หน้าไอเกนที่มีนัยสำคัญที่สุด ดังนั้นเพื่อลดพื้นที่ในการเก็บข้อมูลโดยการเลือกเอาเฉพาะใบหน้าไอเกนแรก ๆ มาเก็บไว้เท่านั้น

ขั้นตอนที่ 5 วิธีการรู้จำ เริ่มจากการนำภาพที่ต้องการทดสอบการรู้จำ T จึงนำภาพใบหน้าไอเกนไปทำการหาค่าน้ำหนัก โดยนำภาพใบหน้าไอเกนที่เลือกคูณกับภาพใบหน้าซึ่งเป็นการฉายภาพ (Projection) ภาพใบหน้ากับภาพใบหน้าไอเกนใน Face space ดังสมการที่ 2-7

$$W_k = \mu_k^T(T - \psi) \quad (2\text{-}7)$$

จากสมการที่ 2-7 ค่า k คือจำนวนใบหน้าที่เลือก ซึ่งเป็นการนำภาพใบหน้ามาคำนวณเพื่อหาค่าน้ำหนักที่จะนำไปเป็นตัวแทนจำแนกหรือรู้จำบุคคลต่อไป ซึ่งจะได้ $\Omega^T = [w_1, w_2, w_3, \dots, w_M]$

ขั้นตอนที่ 6 คำนวณหาค่าระยะห่างระหว่างภาพใบหน้าที่นำมาทดสอบกับภาพใบหน้าไอเกน (Euclidean distance) ดังสมการที่ 2-8

$$\varepsilon_k = \|\Omega - \Omega_k\| \quad (2\text{-}8)$$

โดยที่ ε_k คือ ค่าระยะห่าง

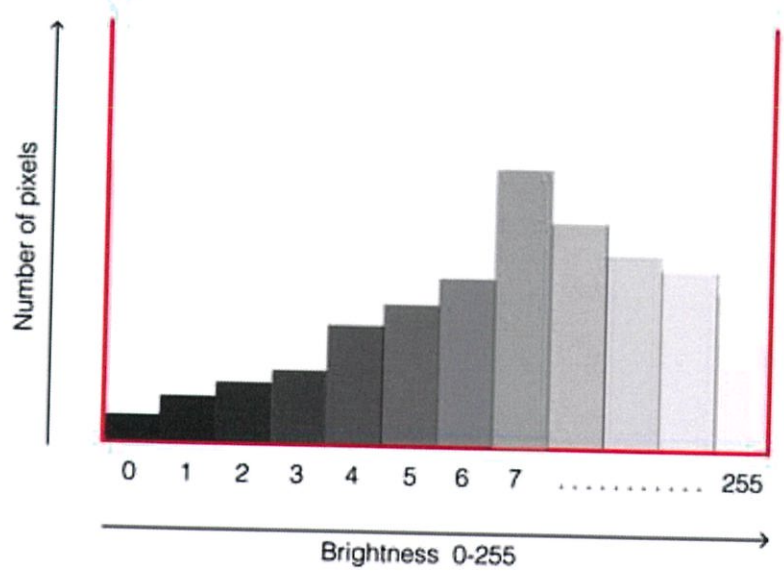
Ω คือ ใบหน้าไอเกน

Ω_k คือ ใบหน้าทดสอบ

จากสมการ 2-8 ค่า Ω_k คือระยะห่างของใบหน้าลำดับที่ k ที่สามารถค้นพบหลังจากมีการคำนวณภาพระยะระหว่างภาพใบหน้ากลุ่มเรียนรู้ กับภาพใบหน้าที่นำมาทดสอบแล้ว นำค่าระยะห่างมาวิเคราะห์ โดยพิจารณาว่าข้อมูลของภาพที่นำมาทดสอบมีค่าใกล้เคียงกับภาพใบหน้าในกลุ่มเรียนรู้ภาพใด มีระยะห่างน้อยที่สุด สรุปผลออกมาได้ว่า ภาพที่นำมาทดสอบนั้นอยู่กลุ่มเดียวกับภาพในกลุ่มเรียนรู้ภาพนั้น

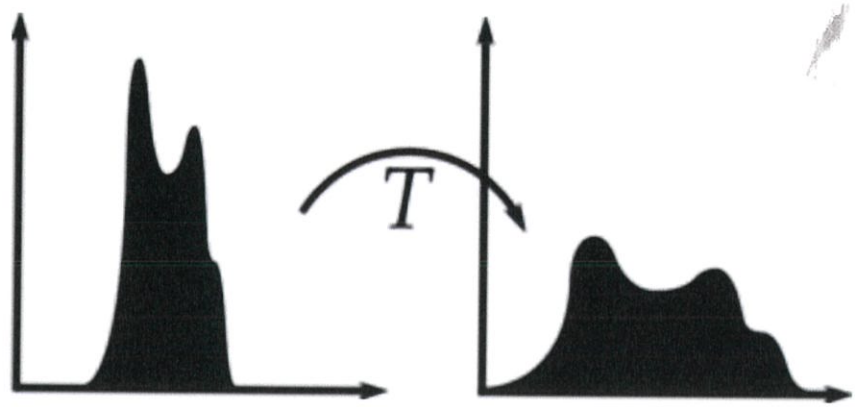
2.1.6 กระบวนการทำ Histogram

กราฟแสดงจำนวนพิกเซลที่ความสว่างต่างๆ ของภาพให้เห็นถึงปริมาณของพิกเซลที่มีค่าความสว่าง (Brightness) ตั้งแต่ค่า “0” ซึ่งหมายถึงตำแหน่งที่ดำที่สุดของภาพ (Shadows) จนกระทั่งถึงค่า “255” ซึ่งเป็นตำแหน่งที่ขาวที่สุด (Highlight) จะเห็นว่าแกนทางแนวนอนของกราฟ แสดงค่าความสว่าง (Brightness) เป็นโทน ขาว-ดำ (Grayscale) 256 ระดับโทน (ตั้งแต่ระดับโทน “0” จนถึงโทน “255”) ขณะที่แกนแนวตั้งแสดงถึงปริมาณของพิกเซล



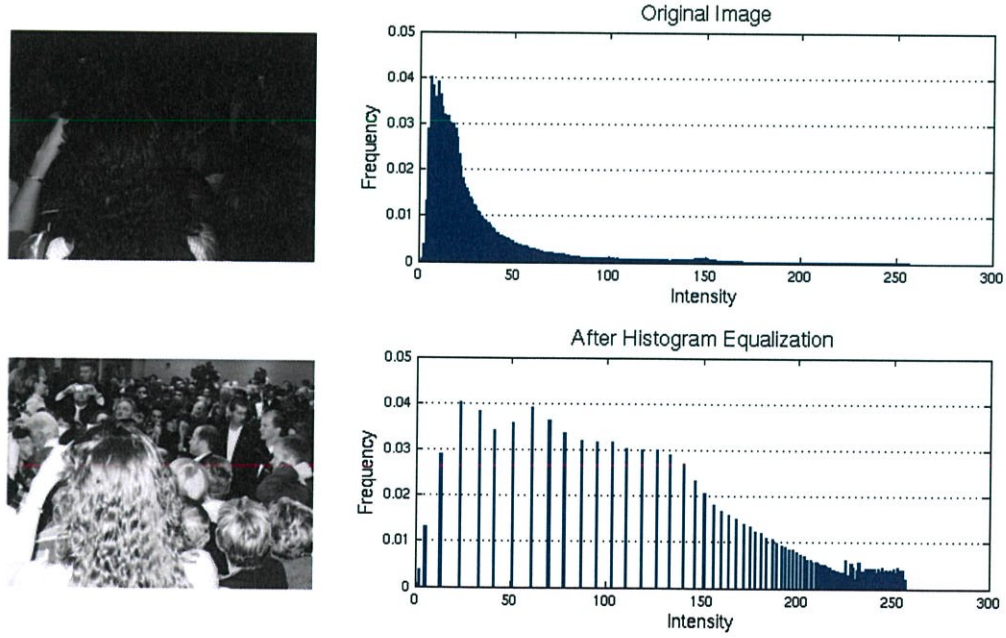
รูปที่ 2.4 Histogram การแบ่งโทนภาพออกเป็น 256 ตั้งแต่โทน 0-255

โดยนำภาพโทนเทาที่ได้ไปพล็อตกราฟระหว่างค่าความสว่างกับจำนวนพิกเซลที่มีค่าความสว่างนั้นๆ จะได้กราฟที่มีความสว่างอยู่ที่ค่าความสว่างช่วงหนึ่ง ไม่ได้กระจายตัวอย่างสม่ำเสมอที่ทุกๆค่าความสว่าง ดังรูป เมื่อผ่านกระบวนการ Histogram Equalization และนำไปพล็อตกราฟอีกครั้ง จะได้การกระจายตัวของความสว่างที่มีความสม่ำเสมอ จึงทำให้เกิดความแตกต่างระหว่างสีที่เรียกว่า contrast มากขึ้น ทำให้มองเห็นภาพชัดเจนมากยิ่งขึ้น



รูปที่ 2.5 กราฟแสดงความสัมพันธ์ระหว่าง ค่าความสว่าง และจำนวนพิกเซล

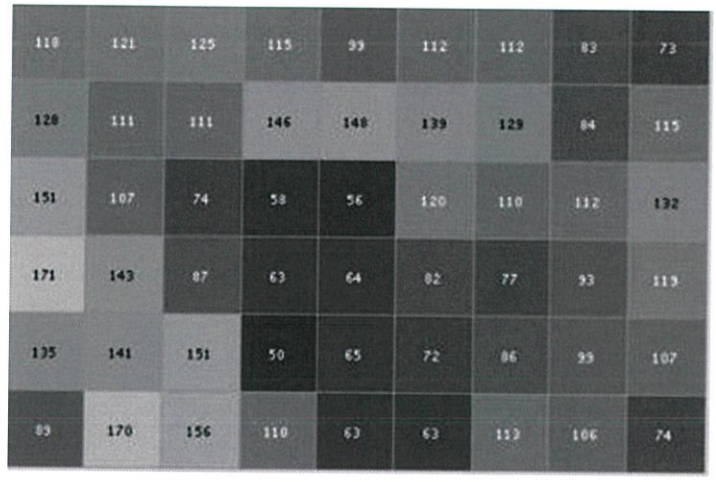
จากรูปที่ 2.6 แสดงภาพก่อนผ่านกระบวนการ histogram Equalization และเมื่อผ่านกระบวนการ histogram Equalization ทำให้ได้ภาพที่มีความชัดเจนขึ้น



รูปที่ 2.6 ภาพก่อนและหลังทำ histogram Equalization

ตัวอย่างขั้นตอนการทำ Histogram Equalization

ให้ภาพโทนเทาที่ต้องการผ่านกระบวน Histogram Equalization เป็นภาพขนาด M x N พิกเซล โดยที่แต่ละพิกเซลเก็บค่าความสว่าง ตั้งแต่ 0-255 แสดงดังรูป



รูปที่ 2.7 ภาพจำลองโดยมีค่าความสว่างในแต่ละพิกเซล โดยที่ช่องแต่ละช่องแทนพิกเซล

1. สร้างตารางนับค่าที่ปรากฏอยู่ในภาพ ดังรูปที่ 2.7 จะได้ตารางเก็บค่าความสว่างตามตาราง

ค่าความสว่าง	จำนวน
50	1
56	1
58	1
63	3
64	1
65	1
72	1
73	1
74	2
77	1
82	1
83	1
84	1
86	1
87	1
89	1
93	1
99	2
106	1
107	2
110	2
111	2

ค่าความสว่าง	จำนวน
112	3
113	1
115	2
118	1
119	1
120	1
121	1
125	1
128	1
129	1
132	1
135	1
139	1
141	1
143	1
146	1
148	1
151	2
156	1
170	1
171	1

ตารางที่ 2.1 ตารางเก็บจำนวนของค่าความสว่าง

2. หาความถี่สะสมของแต่ละอัตรภาคชั้น จากตางราง จะได้ตารางค่าความถี่สะสมของค่าความสว่างดังตารางที่

ค่าความสว่าง	ความถี่สะสม
50	1
56	2
58	3
63	6
64	7
65	8
72	9
73	10
74	12
77	13
82	14
83	15
84	16
86	17
87	18
89	19
93	20
99	22
106	23
107	25
110	27
111	29

ค่าความสว่าง	ความถี่สะสม
112	32
113	33
115	35
118	36
119	37
120	38
121	39
125	40
128	41
129	42
132	43
135	44
139	45
141	46
143	47
146	48
148	49
151	51
156	52
170	53
171	54

ตารางที่ 2.2 ตารางค่าความถี่สะสมของค่าความสว่าง

3. คำนวณสูตร Histogram Equalization ดังสมการ

$$h(v) = \text{round}\left(\frac{cd(v) - cd_{min}}{(M \times N) - cd_{min}}\right) \times (L - 1)$$

โดยที่ cd คือ ความถี่สะสม

v คือ ค่าความสว่าง

L คือ จำนวนค่าความสว่างทั้งหมด (256 ค่า)

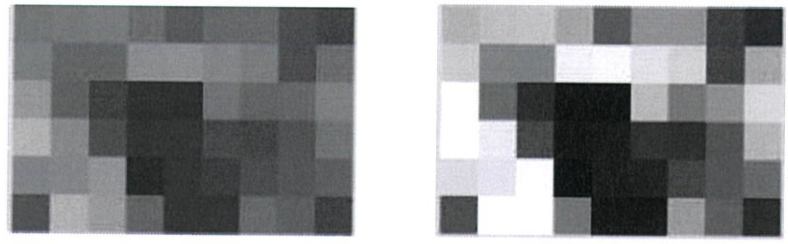
M และ N คือ ขนาดกว้างยาวของรูปภาพตามลำดับ

เมื่อคำนวณแล้วจะได้ค่าความสว่างของรูปภาพใหม่ ดังรูปที่ 2.8

170	182	186	162	97	146	146	69	45
130	134	134	227	231	210	198	73	162
239	113	53	8	4	178	121	146	202
255	223	81	20	28	61	57	93	174
206	215	239	0	32	40	77	97	113
85	251	247	121	20	20	154	105	53

รูปที่ 2.8 แสดงภาพจำลองหลังผ่านกระบวนการ Histogram Equalization

จากรูปที่2.8 สังเกตได้ว่าพิกเซลที่เคยมีค่าความสว่างน้อยสุดคือ ได้กลายเป็น 0 (สีดำ) และพิกเซลที่เคยมีค่าความสว่างมากที่สุดคือ ได้กลายเป็น 255 (สีขาว) รูปที่2.9 แสดงภาพจริงที่ได้จากกระบวนการทำ Histogram Equalization ข้างต้น โดยค่าความสว่างภาพซ้ายมือจะสอดคล้องกับรูปที่ 2.7 และภาพทางขวามือจะสอดคล้องกับรูปที่ 2.8



รูปที่ 2.9 ภาพตัวอย่างก่อนและหลังการทำ Histogram Equalization จากตารางข้างต้น

ตัวอย่างภาพก่อนและหลังกระบวนการทำ Histogram Equalization แสดงดังภาพ

รูปที่ 2.10 ภาพก่อนและหลังการทำ Histogram Equalization

2.2 ข้อมูลอุปกรณ์ iOS

2.2.1 เกี่ยวกับไอโฟน

ไอโฟน (อังกฤษ: iPhone) เป็นโทรศัพท์มือถือที่มีความสามารถใช้งานอินเทอร์เน็ตและมัลติมีเดีย ผลิตและจำหน่ายโดยบริษัทแอปเปิล โดยการทำงานของไอโฟนสามารถใช้งานส่งอีเมล์ ใช้เป็นโทรศัพท์เคลื่อนที่ ส่งเอสเอ็มเอส ท่องอินเทอร์เน็ตผ่านทางซอฟต์แวร์ซาฟารี ค้นหาแผนที่ ฟังเพลง และความสามารถอื่น โดยมีอุปกรณ์หลักประกอบด้วย Wi-Fi (802.11b/g) บลูทูธ 2.0 และกล้องถ่ายภาพ 2.0-megapixel ไอโฟนรุ่นแรกมีลักษณะ 2.5G quad band GSM และ EDGE และรุ่นที่สองใช้ UMTSและ HSDPA

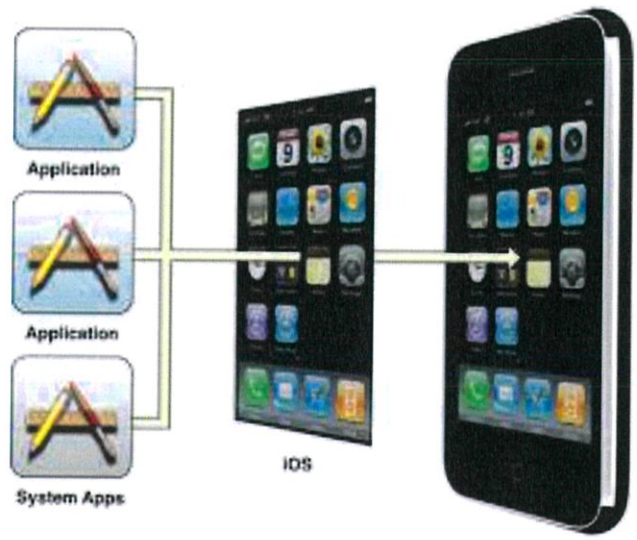
รูปที่ 2.11 ตัวอย่าง iPhone 5

แอปเปิลได้เปิดเผยไอโฟนรุ่นแรกโดย สตีฟ จอบส์ ในงานแม็คเวิลด์ วันที่ 9 มกราคม พ.ศ. 2550 และวางจำหน่ายครั้งแรกในสหรัฐอเมริกาวันที่ 29 มิถุนายน พ.ศ. 2550 ไอโฟนได้ชื่อว่าเป็นสิ่งประดิษฐ์ยอดเยี่ยมประจำปีจากนิตยสารไทม์ ประจำปี 2550[2] โดยมีรุ่นถัดมาคือ ไอโฟน 3G และ ไอโฟน 3GS และไอโฟน 4 ได้เปิดตัวในวันที่ 7 มิถุนายน พ.ศ. 2553 ไอโฟนรุ่นใหม่ล่าสุดในปัจจุบันคือ ไอโฟน 5เอส และ ไอโฟน 5ซี ซึ่งมีเป็นรุ่นที่สมบูรณ์ที่สุดในขณะนี้

การทำงานของโทรศัพท์ไอโฟนนี้จะแตกต่างจากโทรศัพท์มือถืออื่น โดยไอโฟนจะไม่มีปุ่มสำหรับกดหมายเลขโทรศัพท์ โดยการทำงานทั้งหมดจะทำงานผ่านหน้าจอโดยการสัมผัสมัลติทัชผ่านคำสั่งต่างๆ โดยมีระบบปฏิบัติการหลัก iOS และมีระบบเซ็นเซอร์ในการรับรู้สภาพของเครื่องเพื่อกำหนดการแสดงผลของจอภาพ เช่นหากวางเครื่องในแนวตั้งระบบก็จะปรับให้แสดงผลในแนวตั้ง หากวางในแนวนอนระบบก็จะแสดงผลในแนวตั้ง

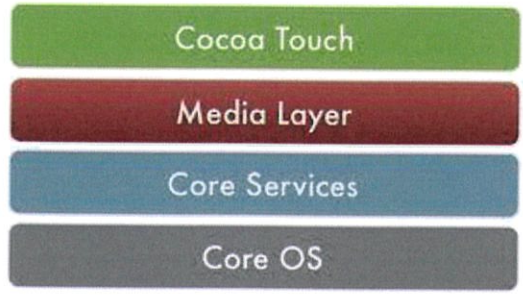
2.2.2 สถาปัตยกรรมของระบบปฏิบัติการ iOS

ระบบปฏิบัติการ iOS จะมีสถาปัตยกรรมที่คล้ายกับสถาปัตยกรรมพื้นฐานที่พบในระบบปฏิบัติการ Mac OS X โดยที่ระบบปฏิบัติการ iOS นี้ จะทำหน้าที่เป็นตัวกลางเชื่อมระหว่างฮาร์ดแวร์พื้นฐานและตัวโปรแกรมประยุกต์ที่ปรากฏบนหน้าจอแสดงผล ดังรูป ซึ่งโดยปกติแล้วโปรแกรมประยุกต์ที่ถูกสร้างขึ้นนั้นแทบที่จะไม่ได้ติดต่อกับฮาร์ดแวร์พื้นฐานโดยตรง ดังนั้นการที่โปรแกรมประยุกต์ที่ติดต่อสื่อสารกับฮาร์ดแวร์ผ่านทางระบบปฏิบัติการ iOS จะมีระบบป้องกันโปรแกรมจากการเปลี่ยนแปลงของฮาร์ดแวร์ได้ ซึ่งส่งผลให้ง่ายต่อการเขียนโปรแกรมที่ทำงานบนอุปกรณ์ที่มีความสามารถของฮาร์ดแวร์ต่างกัน



รูปที่ 2.12 ระบบปฏิบัติการ iOS เป็นตัวกลางระหว่างโปรแกรมประยุกต์และฮาร์ดแวร์

การดำเนินงานของระบบปฏิบัติการ iOS นั้น ถูกแบ่งการทำงานออกเป็นชั้นๆ ซึ่งในแต่ละชั้นนั้นจะมีความสามารถที่แตกต่างกันออกไป ดังต่อไปนี้



รูปที่ 2.13 รูปชั้นโครงสร้างของระบบปฏิบัติการ iOS

2.2.3 OpenCV

เป็น library ของ programming functions ที่ใช้ช่วยในการเขียนโปรแกรมเกี่ยวกับ computer vision เป็นหลัก สามารถนำไปประยุกต์หรือใช้งานได้เลยโดยไม่ต้องคิด algorithm หรือโค้ดขึ้นเอง โดยมี source code และ ตัวอย่าง ถูกพัฒนาจากทีม พัฒนาของ intel และ optimize code สำหรับ cpu 586 ของ intel ทำให้เป็น library ที่มีความเร็วในการประมวล ผลอยู่ในระดับต้นๆ เร็วกว่า ที่มีขายบางตัว โดยเปิดให้บริการฟรีสำหรับการใช้งานภายใต้ใบอนุญาต open source BSD ซึ่ง library เป็นระบบหลายระบบปฏิบัติการ ที่จะเน้นหลักในการประมวลผลภาพแบบเรียลไทม์

เพื่อให้การพัฒนาโปรแกรมทางด้าน การมองเห็นของคอมพิวเตอร์ (Computer Vision) คือสามารถประมวลผลภาพดิจิตอลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวเช่น ภาพจากกล้องVDO หรือ VDO File เป็นไปได้อย่างสะดวก มีฟังก์ชันสำเร็จรูปสำหรับจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐานเช่น การหาขอบภาพ การกรองข้อมูลภาพ การประมวลผลเกี่ยวภาพและสัญญาณ(Image and Signal Processing),การตรวจสอบลักษณะวัตถุจากภาพหรือวิดีโอ(Object Identification,), การจดจำใบหน้า(Face Recognition) ม่านตา(iris Recognition), ตรวจสอบขอบหรือด้านของวัตถุ(Edge Detection), ตรวจสอบความเคลื่อนไหว(Motion Detection) และอื่นๆ

ชั้นโครงสร้างของระบบปฏิบัติการ iOS

Core OS

คือ ชั้นเลเยอร์ต่ำสุดที่ถูกสร้างขึ้นเมื่อเทียบจากชั้นเลเยอร์อื่นๆ ซึ่งโดยส่วนใหญ่แล้วจะไม่ค่อยได้ใช้ชั้นเลเยอร์นี้โดยตรงในโปรแกรมประยุกต์ แต่ชั้น Core OS มักที่จะถูกเรียกใช้โดยเฟรมเวิร์คอื่นๆ รวมถึงในสถานการณ์ที่จำเป็นที่จะต้องจัดการกับความปลอดภัย หรือการติดต่อการสื่อสารกับอุปกรณ์ฮาร์ดแวร์ภายนอกโดยใช้เฟรมเวิร์คจากชั้นเลเยอร์นี้

ตัวอย่างของเฟรมเวิร์คที่อยู่ในชั้นเลเยอร์ Core OS ได้แก่

- Accelerate Framework เป็นเฟรมเวิร์คที่มีอินเตอร์เฟซสำหรับการดำเนินการ DSP พีชคณิตเชิงเส้น และการคำนวณประมวลผลภาพต่างๆ ซึ่งประโยชน์ของการใช้เฟรมเวิร์คนี้ คือ จะมีการปรับค่าให้เหมาะสมกับทุกการกำหนดค่าของฮาร์ดแวร์ที่มีอยู่ใน

อุปกรณ์ iOS ที่ใช้ ดังนั้น นักพัฒนาสามารถเขียนโค้ดได้ทันที โดยมั่นใจได้ว่าจะทำงานได้อย่างมีประสิทธิภาพในอุปกรณ์ทั้งหมด

- Core Bluetooth เป็นเฟรมเวิร์คที่ช่วยให้นักพัฒนาใช้ในการโต้ตอบกับบลูทูธ ซึ่งประโยชน์ของการใช้เฟรมเวิร์คนี้คือ ช่วยสามารถสแกนหาอุปกรณ์ LE เพื่อทำการเชื่อมต่อและยกเลิกการเชื่อมต่อได้

- External Accessory เป็นเฟรมเวิร์คที่ให้การสนับสนุนสำหรับการติดต่อสื่อสารกับอุปกรณ์ฮาร์ดแวร์ที่เชื่อมต่อกับอุปกรณ์ที่ใช้ iOS โดยอุปกรณ์สามารถเชื่อมต่อผ่านช่องต่อ 30-pin ของอุปกรณ์แบบไร้สายหรือโดยใช้บลูทูธ ซึ่งเฟรมเวิร์คนี้จะช่วยให้นักพัฒนาได้รู้วิธีการ ข้อมูลเกี่ยวกับอุปกรณ์เสริม และการเริ่มต้นเซสชันการสื่อสารต่างๆ หลังจากนั้นนักพัฒนาจะมีอิสระในการจัดการอุปกรณ์โดยตรงโดยใช้คำสั่งใดๆ รองรอบ

- Generic Security Services เป็นเฟรมเวิร์คที่ใช้กำหนดมาตรฐานของการบริการที่เกี่ยวข้องกับความปลอดภัยในการใช้งาน iOS โดยอินเตอร์เฟซพื้นฐานของเฟรมเวิร์คนี้จะถูกระบุไว้ใน IETF RFC 2743และRFC 4401

- Security เป็นเฟรมเวิร์คที่นักพัฒนาใช้เพื่อรับประกันความปลอดภัยของข้อมูลโปรแกรมประยุกต์ที่สร้างขึ้น ช่วยสนับสนุนการสร้างระบบรักษาความปลอดภัย cryptographically รวมถึงตัวเลข pseudorandom นอกจากนี้ยังสนับสนุนการจัดเก็บคีย์ของการเข้ารหัสลับ ซึ่งเป็นพื้นที่เก็บข้อมูลของผู้ใช้

Core Services

คือ ชั้นเลเยอร์ที่อยู่ติดกับชั้นเลเยอร์ Core OS โดยชั้นเลเยอร์นี้จะทำการจัดเตรียมการติดต่อพื้นฐานไปยังบริการต่างๆบนระบบปฏิบัติการ iPhone ซึ่งจะมีหน้าที่ในการจัดหาโครงสร้างส่วนที่ครอบคลุมการบริการทั้งหมดไว้ให้

ตัวอย่างของเฟรมเวิร์คที่อยู่ในชั้นเลเยอร์ Core Services ได้แก่

- Accounts เป็นเฟรมเวิร์คที่ช่วยบริหารจัดการระบบบัญชีของผู้ใช้ และช่วยลดความยุ่งยากของรูปแบบการพัฒนาสำหรับผู้พัฒนา ด้วยการจัดการกระบวนการอนุมัติบัญชีสำหรับโปรแกรมประยุกต์ที่สร้างขึ้น นอกจากนี้ยังสามารถใช้เฟรมเวิร์คนี้ร่วมกันกับการทำงานของ Twitter เพื่อเข้าถึงบัญชีผู้ใช้ Twitter ได้

- Address Book เป็นเฟรมเวิร์คที่ช่วยในการเข้าถึงการเขียนโปรแกรมเพื่อติดต่อกับพื้นที่เก็บข้อมูลที่เก็บไว้บนอุปกรณ์ของผู้ใช้ ไม่ว่าจะเป็นในด้านของการเข้าถึงข้อมูลและแก้ไขข้อมูลของผู้ใช้ในฐานข้อมูล ตัวอย่างเช่น โปรแกรมแชท อาจใช้เฟรมเวิร์คนี้เพื่อดึงรายชื่อผู้ติดต่อที่เป็นไปได้ที่จะเริ่มต้นการสนทนา และแสดงรายชื่อเหล่านั้นในมุมมองที่กำหนดขึ้น

- CFNetwork เป็นเฟรมเวิร์คที่ใช้เชื่อมต่อกับ C-based ที่ใช้ abstractions เชิงวัตถุสำหรับการทำงานกับเครือข่ายโปรโตคอล โดย abstractions เหล่านี้ จะทำให้นักพัฒนาสามารถควบคุมรายละเอียดได้มากกว่าโปรโตคอลสแต็ค และยังง่ายต่อการใช้โครงสร้างระดับที่ต่ำกว่า เช่น ซ็อกเก็ต BSD ซึ่งประโยชน์ของเฟรมเวิร์คนี้ คือ เพื่อลดความซับซ้อนของงานต่างๆ เช่น การสื่อสารกับ FTP และ HTTP เซิร์ฟเวอร์ หรือโฮสต์ นอกจากนี้ยังสามารถแก้ไข DNS ด้วย CFNetwork เฟรมเวิร์ค ดังนี้

  - ใช้ซ็อกเก็ต BSD
  - สร้างการเชื่อมต่อที่เข้ารหัสโดยใช้ SSL หรือ TLS
  - แก้ไข DNS โฮสต์
  - ทำงานร่วมกับเซิร์ฟเวอร์ HTTP ตรวจสอบเซิร์ฟเวอร์ HTTP และเซิร์ฟเวอร์ HTTPS
  - ทำงานร่วมกับเซิร์ฟเวอร์ FTP
  - ประกาศแก้ไขและเรียกดูบริการ Bomjour

- Core Data เป็นเฟรมเวิร์คที่ใช้สำหรับการจัดการรูปแบบข้อมูลจากโปรแกรมประยุกต์ Model-View-Controller Core ซึ่งข้อมูลมีไว้สำหรับใช้งานในรูปแบบของข้อมูลโครงสร้างที่มีอยู่แล้ว แทนการกำหนดโครงสร้างข้อมูลของโปรแกรม โดยนักพัฒนาโปรแกรมจะใช้เครื่องมือแบบกราฟิกใน Xcode ในการสร้างสคีมา เพื่อมาใช้เป็นตัวแทนของรูปแบบข้อมูลที่สร้างขึ้นที่รันไทม์อินสแตนซ์ นอกจากนี้ Core Data ยังมีคุณสมบัติดังต่อไปนี้

  - การเก็บรักษาข้อมูลของวัตถุในฐานข้อมูล SQLife เพื่อประสิทธิภาพสูงสุด
  - ระดับ NSFetchedResultsController ในการจัดการผลสำหรับมุมมองของตาราง

- การบริหารจัดการขอยกเลิก ทำซ้ำ หรือแก้ไขข้อความขั้นพื้นฐาน
- การช่วยเหลือสำหรับการตรวจสอบของค่าสถานที่ให้บริการ
- การช่วยเหลือสำหรับการขยาย เปลี่ยนแปลง และสร้างความมั่นใจว่ามีความสัมพันธ์ระหว่างวัตถุจะสอดคล้องกัน
- การช่วยเหลือสำหรับการจัดกลุ่มข้อมูลการกรองและการจัดระเบียบในหน่วยความจำ

- Core Foundation คือ ชุดของอินเตอร์เฟซ C-based ที่ให้ข้อมูลพื้นฐานในการจัดการ และแสดงคุณลักษณะของการบริการสำหรับการใช้งาน iOS รวมถึงสนับสนุนสิ่งต่อไปนี้

- ชนิดข้อมูลที่เก็บรวมรวม(ชุดอาร์เรย์และอื่นๆ)
- การรวมกลุ่ม
- การจัดการ string
- วันที่และการจัดการเวลา
- วัตถุดิบในการจัดการข้อมูลบล็อก
- การจัดการการตั้งค่า URL และการจัดการลูปการทำงาน
- พอร์ตและซ็อกเก็ตการสื่อสาร

- Core Location เป็นเฟรมเวิร์คที่ให้บริการและข้อมูลเกี่ยวกับสถานที่เพื่อการใช้งาน สำหรับข้อมูลสถานที่จะใช้ onboard GPS มือถือ วิทยุ หรือ Wi-Fi ในการหาเส้นแวงและละติจูดของผู้ใช้ โดยนักพัฒนาสามารถรวมเทคโนโลยีนี้เพื่อให้ข้อมูลตำแหน่งที่ใช้กับผู้ใช้ ตัวอย่างเช่น บริการทีค้นหาร้านอาหาร ร้านค้าหรือสิ่งอำนวยความสะดวกในบริเวณใกล้เคียง และฐานการค้นหาที่เกี่ยวกับตำแหน่งปัจจุบันของผู้ใช้งาน

- Core Media เป็นเฟรมเวิร์คที่ใช้ในการจัดการภาพและเสียง โปรแกรมส่วนใหญ่ไม่จำเป็นต้องใช้เฟรมเวิร์คนี้ แต่มีไว้สำหรับนักพัฒนาเพียงไม่กี่คนที่ต้องการควบคุมที่แม่นยำยิ่งขึ้นมากกว่าการสร้างและนำเสนอเนื้อหาเสียงและวิดีโอแบบธรรมดา

- Core Telephony เป็นเฟรมเวิร์คที่มีอินเตอร์เฟซสำหรับการโต้ตอบกับข้อมูลโทรศัพท์ที่ใช้บนอุปกรณ์ที่มีวิทยุเซลลูลาร์ การประยุกต์การใช้งานนั้น สามารถใช้

เฟรมเวิร์คนี้เพื่อให้ได้ข้อมูลเกี่ยวกับการให้บริการของผู้ใช้โทรศัพท์มือถือ การใช้งาน VoIP และยังสามารถแจ้งเตือนเมื่อเหตุการณ์เหล่านั้นเกิดขึ้นได้

Media

คือ ชั้นเลเยอร์ที่อยู่ติดกับชั้นเลเยอร์ Core Services โดยชั้นเลเยอร์นี้จะมุ่งเน้นการสร้างประสบการณ์ด้านมัลติมีเดียที่ดีที่สุดบนอุปกรณ์ฮาร์ดแวร์ ซึ่งได้รับการออกแบบมาให้ง่ายสำหรับการสร้างโปรแกรมประยุกต์ เพราะได้มีการจัดเตรียมบริการมัลติมีเดียไว้ให้เลือกไปใช้ในโปรแกรมประยุกต์ได้

ตัวอย่างของเฟรมเวิร์คที่อยู่ในชั้นเลเยอร์ Media ได้แก่

- Assets Library เป็นเฟรมเวิร์คที่ให้อินเตอร์เฟซที่ใช้สำหรับการดึงภาพถ่ายและวิดีโอจากอุปกรณ์ของผู้ใช้ โดยสามารถบันทึกภาพถ่ายและวิดีโอใหม่ๆกลับไปที่อัลบั้มภาพของผู้ใช้ที่บันทึกไว้
- Core Graphics เป็นเฟรมเวิร์คที่มีอินเตอร์เฟซสำหรับ Quartz 2D ซึ่งเป็นเครื่องมือที่ใช้ในการวาดรูปภาพแบบเวกเตอร์ที่ใช้ในระบบปฏิบัติการ Mac OS X โดยจะให้การสนับสนุนการวาดภาพ เส้นทางที่ใช้ในการแสดงผล การไล่ระดับสี ภาพ รวมถึงการแปลงพิกัดพื้นที่ และสร้างเอกสาร PDF ซึ่งทั้งหมดนี้จะใช้ abstractions เพื่อแสดงรูปวาดพื้นฐานของวัตถุ ทำให้ง่ายในการจัดเก็บ
- Core Image เป็นเฟรมเวิร์คสำหรับการจัดการวิดีโอและภาพนิ่ง จะมีการดำเนินงานที่เรียบง่าย เช่นการเพิ่มและแก้ไขภาพถ่าย ตลอดจนมีการดำเนินงานที่สูงขึ้น ไม่ว่าจะเป็น การตรวจสอบคุณสมบัติต่างๆ การถ่ายภาพใบหน้า เป็นต้น โดยเฟรมเวิร์คจะไม่ทำลายภาพต้นฉบับ และจะไม่กินพลังงานของ CPU จึงได้ภาพที่รวดเร็วและมีประสิทธิภาพ โดยที่ CIDetector เป็นคลาสที่อยู่ใน CoreImage. framework ซึ่งคลาสนี้มีเมธอดหลัก คือ (NSArray *)featuresInImage:(CIImage *) image ที่จะคอยรับ CIImage เข้าไปและ return Array ของ CIFaceFeature กลับออกมา ซึ่งขนาดของ Array ก็จะขึ้นอยู่กับจำนวนของใบหน้าคนที่มีการตรวจพบ ซึ่ง CIFace Feature ก็จะบรรจุตำแหน่งของ ตาซ้าย ตาขวา และปากไว้
- Core Text เป็นเฟรมเวิร์คที่ใช้สำหรับออกแบบตัวอักษร และการจัดการกรอบข้อความหลักให้ข้อความมีความสมบูรณ์แบบอย่างที่ต้องการ นอกจากนี้ยังสามารถใช้

เพื่อจัดตำแหน่งของข้อความบนหน้าจอ ซึ่งในแต่ละข้อความก็มีสไตล์แบบอักษร และคุณสมบัติการแสดงผลที่แตกต่างกันออกไป เฟรมเวิร์คนี้เหมาะสำหรับโปรแกรมประยุกต์ที่ต้องใช้ความสามารถในการจัดการข้อความที่มีความซับซ้อน เช่น โปรแกรมประมวลคำ เป็นต้น

Cocoa Touch

คือ เลเยอร์ชั้นสูงสุดที่มีเฟรมเวิร์คสำคัญที่ใช้ในการสร้างโปรแกรมประยุกต์ โดยเลเยอร์นี้จะเป็นโครงสร้างของโปรแกรมพื้นฐาน และช่วยสนับสนุนเทคโนโลยีที่สำคัญ เช่น มัลติทาสกิ้ง การสัมผัสอินผุตพื้นฐานการแจ้งเตือน และอื่นๆ ซึ่งชั้นเลเยอร์ Cocoa Touch นี้จะทำการจัดเตรียมชั้นโครงสร้างที่ประกอบไปด้วยไลบรารี่ที่หลากหลาย เพื่อนำมาใช้ในการสร้างโปรแกรมบน iPhone และ iPod Touch

ตัวอย่างของเฟรมเวิร์คที่อยู่ในชั้นเลเยอร์ Cocoa Touch ได้แก่

- Address Book UI เป็นเฟรมเวิร์คที่มีอินเตอร์เฟซการเขียนโปรแกรมแสดงการเชื่อมต่อระบบมาตรฐานสำหรับสร้างการติดต่อ การแก้ไข และการเลือกผู้ติดต่อที่มีอยู่ เฟรมเวิร์คนี้จะช่วยลดความยุ่งของการดำเนินการลง
- Event Kit UI เป็นเฟรมเวิร์คที่ให้การควบคุมมุมมอง เพื่อนำเสนอการเชื่อมต่อระบบมาตรฐานสำหรับการดู และแก้ไขเหตุการณ์ที่เกี่ยวข้องกับปฏิทิน
- Message UI เป็นเฟรมเวิร์คที่ให้การสนับสนุนสำหรับการเขียน และเรียงลำดับข้อความอีเมล์ของผู้ใช้ในกล่องขาออก จะประกอบด้วยอินเตอร์เฟซการควบคุมมุมมอง
- Map Kit เป็นเฟรมเวิร์คที่มีอินเตอร์เฟซแผนที่ สามารถใช้แสดงที่อยู่ของผู้ใช้ แสดงทิศทาง พิกัดหรือจุดไฮไลท์ที่น่าสนใจ การประยุกต์ใช้งานโปรแกรมสามารถตั้งค่าคุณสมบัติของแผนที่ หรือแจ้งให้ผู้ใช้นำทางแผนที่ได้อย่างอิสระ นอกจากนี้ยังสามารถใส่คำอธิบายใต้ภาพแผนที่ได้

2.2.4 ชุดพัฒนา iOS SDK

iPhone SDK เป็นโปรแกรมพัฒนาโปรแกรมประยุกต์สำหรับใช้งานบน iPhone โดยสามารถดาวน์โหลดมาใช้งานได้จากเว็บไซต์ iPhone Dev Center ซึ่ง iPhone SDK นั้นจะประกอบด้วยโค้ดของข้อมูล และเครื่องมือที่ใช้ในการพัฒนา ทดสอบ ใช้งาน และแก้ไข

โปรแกรมประยุกต์สำหรับ iPhone OS โดยโปรแกรมจะมาพร้อมกับ iPhone Emulator สำหรับทดสอบการใช้งานโปรแกรมประยุกต์บนเครื่องคอมพิวเตอร์เสียก่อนการใช้งานจริงกับ iPhone

สำหรับผู้สนใจสามารถดาวน์โหลด iPhone SDK มาใช้ได้ฟรี แต่หากต้องการพัฒนาโปรแกรมเพื่อการค้า จะต้องเสียค่าใช้จ่ายในการเข้าร่วมโครงการ iPhone Developer Program ในราคา $99 สำหรับผู้พัฒนารายบุคคลใน Standard Program และ $299 สำหรับบริษัทพัฒนาใน Enterprise Program ซึ่งจะได้รับสิทธิในการนำโปรแกรมประยุกต์มาจำหน่ายผ่านทางร้านออนไลน์ของ Apple ด้วย โดยมีการหักค่าใช้จ่าย 30 เปอร์เซ็นต์ของกำไรที่ได้รับ

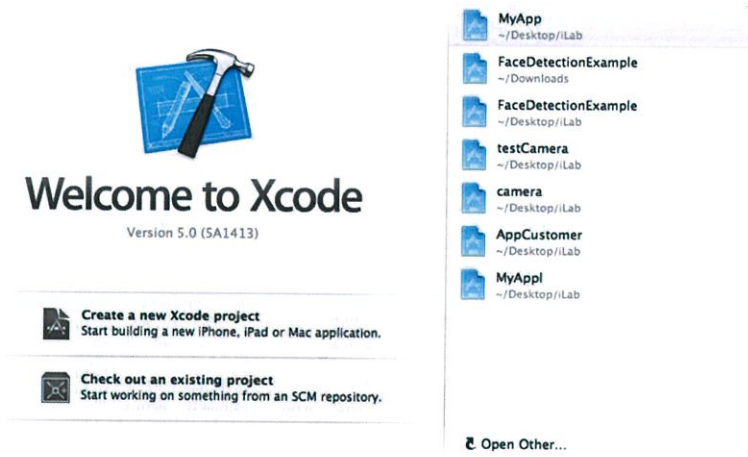
2.2.5 องค์ประกอบของชุดพัฒนา iPhone SDK

ชุดพัฒนา iPhone SDK จะประกอบไปด้วยชุดของเครื่องมือที่ช่วยในการพัฒนาโปรแกรมประยุกต์ต่างๆ สำหรับ iPhone และ iPod Touch ซึ่งจะมีองค์ประกอบดังต่อไปนี้

2.2.5.1 เครื่องมือ XCode สำหรับสร้างโปรแกรมประยุกต์บน iPhone

Xcode เป็นโปรแกรมประเภท IDE หรือ Integrated Development Environment ซึ่งเป็นตัวหลักหัวใจสำคัญในการสร้างและจัดการชิ้นงานโปรแกรมประยุกต์ รวมถึงไฟล์ source ต่างๆ การสร้างโค้ด ใช้งานโค้ด และแก้ไขเมื่อทดลองใช้งานบน iPhone simulator หรือกับเครื่อง iPhone

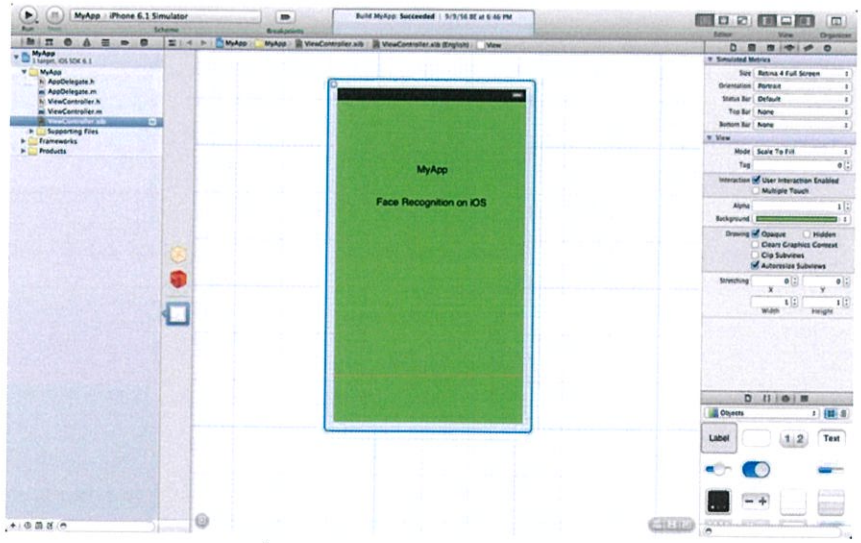
การสร้างโปรแกรมประยุกต์ใหม่สำหรับ iPhone จะต้องเริ่มด้วยการสร้างชิ้นงานบน Xcode โดยตัวโปรแกรมจะจัดการฐานข้อมูลที่เกี่ยวข้องกับโปรแกรมประยุกต์นั้น รวมไปถึงไฟล์ source การติดตั้ง และอื่นๆที่ต้องใช้ในการพัฒนาโปรแกรมให้สามารถใช้งานได้จริง



รูปที่ 2.14 หน้าต่างของ Xcode

2.2.5.2 เครื่องมือ interface Builder สำหรับการออกแบบอินเตอร์เฟซ บนหน้าจอ iPhone

Interface Builder เป็นอุปกรณ์ในการรวมและแสดง User Interface ของโปรแกรมประยุกต์ด้วยการใช้ Interface Builder ทำให้สามารถจัดการหน้าต่างของโปรแกรมประยุกต์ได้ด้วยการลากและวางส่วนต่างๆอย่างง่ายๆ โดยส่วนประกอบต่างๆ รวมถึงระบบควบคุมมาตรฐานต่างๆ เช่น สวิตซ์ ปุ่มกด ช่องตัวหนังสือ และการเลือกแสดงแบบ custom ซึ่งเมื่อได้จัดวางองค์ประกอบต่างๆเรียบร้อย จะสามารถลากไปในทิศทางตามที่ต้องการ จัดแต่ง และสร้างการเชื่อมโยงระหว่างองค์ประกอบนั้นกับโค้ดที่เขียนไว้ เมื่อ Interface มีลักษณะตามที่ต้องการ สามารถบันทึกเป็น .nib ซึ่งเป็นฟอร์แมตไฟล์ประเภท resource

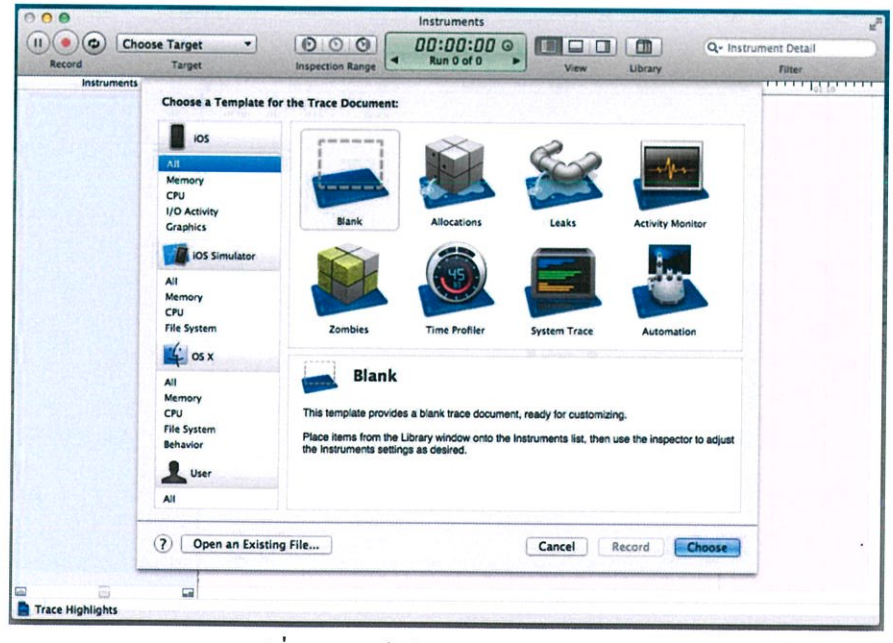


รูปที่ 2.15 หน้าต่างของ Interface Builder

2.2.5.3 เครื่องมือ instruments สำหรับตรวจสอบประสิทธิภาพของโปรแกรมประยุกต์

เพื่อให้การใช้งานโปรแกรมประยุกต์นั้นสมบูรณ์แบบที่สุด ชุด Instruments นี้จะช่วยให้สามารถวิเคราะห์การทำงานของโปรแกรมประยุกต์ในขณะที่ทดลองใช้งานบน simulator หรือ iPhone โดย Instruments จะรวบรวมข้อมูลจากโปรแกรมประยุกต์ขณะที่ทำงาน แล้วนำเสนอข้อมูลเหล่านั้นในการแสดงผลแบบกราฟิกที่เรียกว่า “Timeline” ซึ่งข้อมูลต่างๆที่รวบรวมนั้น ได้แก่ การใช้งานหน่วยความจำ พื้นที่ disk การเชื่อมต่อเครือข่าย และด้านกราฟิกของโปรแกรมประยุกต์ การแสดงผล Timeline นั้นจะแสดงประเภทข้อมูลต่างๆเทียบเคียงกัน เพื่อให้สามารถคำนวณการทำงานโดยรวมระบบของโปรแกรมประยุกต์นั้น ไม่ใช่เพียงแค่เฉพาะบางส่วน และหากต้องการดูข้อมูลแบบละเอียดสามารถเลือกดูได้เช่นกัน

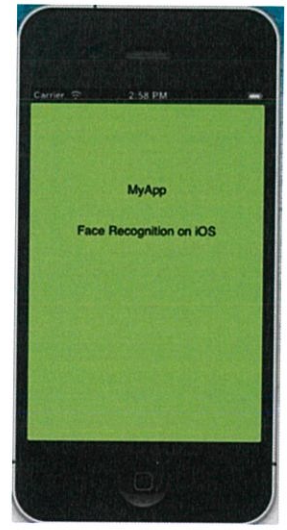
นอกเหนือจากการแสดงข้อมูลแบบ Timeline แล้ว Instruments ยังเป็นเครื่องมือที่ช่วยวิเคราะห์การทำงานของโปรแกรมประยุกต์ในระยะเวลานาน ตัวอย่างเช่น หน้าต่างของ Instruments จะรวบรวมข้อมูลจากการทำงานหลายครั้ง เพื่อนำมาแสดงให้เห็นว่าโปรแกรมประยุกต์ควรได้รับการปรับปรุง หรือว่ามีการทำงานที่ดีขึ้น และยังสามารถบันทึกข้อมูลจากการทำงานดังกล่าว เพื่อเปิดดูอีกครั้งตามที่ต้องการได้



รูปที่ 2.16 หน้าต่างของชุด Instruments

2.2.5.4 เครื่องมือ simulator สำหรับจำลองการทำงานของโปรแกรมประยุกต์ที่พัฒนาขึ้น

Simulator เป็นเครื่องมือที่มีประโยชน์มาก โดยจะรวมอยู่ในชุดพัฒนา iPhone SDK ซึ่งจะสามารถนำมาใช้เพื่อทดสอบโปรแกรมประยุกต์ โดยไม่ต้องใช้ iPhone/ iPod Touch ที่แท้จริง โดย Simulator มี API แบบเดียวกับ iPhone ซึ่งจะทำการแสดงตัวอย่างของโปรแกรมที่สร้างขึ้น เพื่อให้สามารถเห็นภาพเวลาที่ทำงานจริง เมื่อเรียกใช้ simulator โปรแกรม Xcode จะคอมไฟล์โค้ดเป็นแบบ x86 เพื่อให้ทำงานบนเครื่องแม็คได้ แทนที่จะเป็น ARM ที่ทำงานบน iPhone เครื่องจริง



รูปที่ 2.17 หน้าของ iPhone Simulator

จะเห็นส่วนประกอบทั้งหมดในชุดพัฒนา iPhone SDK ได้มีการนำเสนอทางเลือกในการพัฒนาโปรแกรมบน iPhone ทั้งแบบแบบปกติที่ใช้ Objective-C และ Web-based สำหรับการพัฒนาแบบปกติ เครื่องมือที่สำคัญที่สุด คือ Xcode และ Simulator โดยมี Instruments คอยช่วยปรับจูนประสิทธิภาพ และยังมีองค์ประกอบอีกอย่างหนึ่งที่ไม่อยู่ในรายการแต่สำคัญไม่แพ้กัน คือ Cocoa Touch นอกจากนี้การเขียนโปรแกรมบน iPhone ยังอิงอยู่บนแนวคิดที่สำคัญ 2 ประการ คือ การเขียนโปรแกรมเชิงวัตถุ (OOP) และหลักการ Model-View-Controller (MVC)

2.2.6 ส่วนประกอบของโปรแกรมบน iOS

โปรแกรมของ iPhone เหมือนกับของแม็คอินทอชตรงที่จะถูกรวบรวมมาเป็นแพ็กเกจที่เรียกว่า Application Budle ซึ่งคือโฟลเดอร์ธรรมดาที่มีนามสกุล .app ต่อท้าย ทุกอย่างของโปรแกรมไม่ว่าจะเป็นไฟล์ภาพ และเสียงที่ถูกคอมไฟล์เรียบร้อยแล้วไปจนถึงไฟล์พิเศษที่เก็บข้อมูลของโปรแกรม เพื่อให้ระบบปฏิบัติการรับรู้จะถูกเก็บไว้ในโฟลเดอร์นี้ ระบบปฏิบัติการจะมองเห็นโฟลเดอร์ .app เป็นก้อนโปรแกรมหนึ่งก้อน ซึ่งลำกับชั้นของโฟลเดอร์ในโปรแกรม มีดังนี้

ไฟล์ Executable

ไฟล์ Executable หรือไฟล์จริงๆของโปรแกรมที่คอมไฟล์แล้ว และรอการเรียกใช้งานจะถูกเก็บไว้ในโฟลเดอร์ชั้นนอกสุดของโปรแกรม ไฟล์เหล่านี้จะต้องมีสิทธิ์ในการรันและต้องเป็นไฟล์ที่ SpringBoard (ตัวจัดการไฟล์ของ iPhone เทียบได้กับ Finder ของแม็ค) รับรองในเฟิร์มแวร์รุ่น 1.2 เป็นต้นมา SpringBoard จะมีระบบตรวจจับเพื่อป้องกันไม่ให้ไฟล์เหล่านี้ถูกเรียกใช้งานโดยพลการ ฟีเจอร์นี้จะเหมือนกับระบบ UNIX อื่นๆ คือเพิ่มตัวกรองการเรียกใช้คำสั่งแบบคอมมานด์ไลน์(Command-Line)เข้าไป แถมยังจำกัดปริมาณหน่วยความจำที่โปรแกรมสามารถใช้ได้โดยจะปิดการทำงานที่ใช้ทรัพยากรของระบบมากเกินทิ้งไป

ไฟล์ Info.plist

โฟลเดอร์โปรแกรมของ iPhone จะมีไฟล์ Info.plist แบบเดียวกับของแม็ค ไฟล์ Info.plist เป็นไฟล์ XML ที่เก็บคุณสมบัติต่างๆของโปรแกรม เพื่อให้ระบบปฏิบัติการรู้จักกับโปรแกรมได้อย่างถูกต้อง คุณสมบัติเหล่านี้จะถูกเก็บอยู่ในรูปของคู่ลำดับ Key-Value และสามารถบันทึกเป็นไฟล์ข้อความธรรมดาที่มนุษย์อ่านออก หรือแบบไฟล์ไบนารี่ (Binary) ที่ถูกบีบอัดมาแล้วในไฟล์ Info.plist จะต้องระบุค่าไฟล์ Executable (CFBundle-Executable) และ Identifier (ชื่อหรือตัวเลขอ้างอิง) (CFBundleExecutable) โดย Identifier นั้นสำคัญมากในการทำให้โปรแกรมทำงานได้ถูกต้อง

Identifier ช่วยให้สามารถติดต่อสื่อสารกับโปรแกรมอื่นๆ และลงทะเบียนโปรแกรมของนักพัฒนา โดยอาศัย SpringBoard (ตัวจัดการไฟล์ของ iPhone) ได้ SpringBoard จะทำหน้าที่เรียกหน้าแรกของโปรแกรมขึ้นมา เมื่อนักพัฒนาสั่งโปรแกรมเริ่ม

ทำงาน นอกจากนี้ Identifier ยังถูกใช้ในการสร้างหน้า Preferences (หน้าตั้งค่า) แบบมาตรฐานให้โดยอัตโนมัติ

หน้า Preferences จะถูกเก็บไว้ในไลบรารี่ของผู้ใช้(/var/mobile/Library /Preferences)โดยใช้ Identifier นี้จะถูกต่อท้ายด้วยคำว่า .plist ให้โดยอัตโนมัติ เช่น com. sadun.appname.plist เป็นต้น และค่าต่างๆจะถูกเก็บอยู่ในรูปไบนารี่ของ plist นักพัฒนาสามารถอ่านค่า plist เหล่านี้ผ่านแม็คได้ โดยคัดลอกไปไว้ในเครื่องแม็ค แล้วใช้โปรแกรม plutil ของ Apple แปลงค่า plist แบบไบนารี่เหล่านี้กลับมาเป็นไฟล์ XML ที่อ่านได้ คำสั่งที่ใช้คือ plutil convert xml1 plistfile เหตุผลที่ Apple เลือกเก็บค่า plist เป็นไบนารี่เพื่อประหยัดเนื้อที่เก็บข้อมูล และให้ระบบมีประสิทธิภาพดียิ่งขึ้น

นักพัฒนาจะปรับแต่งไฟล์ Info.plist ได้อย่างยืดหยุ่นมากขึ้นถ้านำไปดูบนแม็ค ซึ่งนักพัฒนาสามารถตั้งค่าตัวแปรของ SpringBoard เช่น SBUsersNetwork หรือปรับแต่งวิธีการแสดงไอคอนโปรแกรมได้(UIPrerenderedIcon) นอกจากนี้ ค่าบางอันของ SpringBoard ยังอนุญาตให้นักพัฒนาระบุให้โปรแกรมตัวเดียวทำงานหลายอย่างได้ เช่น Photos และ Camera แท้จริงแล้วคือโปรแกรมเดียวกัน โดย MobileSlideShow จะรับทำงานในอีกบทบาทหนึ่งเท่านั้น นอกจากนี้นักพัฒนายังสามารถสรุปว่าเมื่อไรที่ต้องการซ่อนโปรแกรมจากหน้าจอได้

ค่าอื่นๆที่เก็บไว้ใน Info.plist ได้แก่ UIStatusBarStyle (บอกสีสันและหน้าตาของแถบสถานะ) UIStatusBarHidden (บอกว่าให้ซ่อนแถบสถานะหรือไม่) UIInterface Orientation (กำหนดให้กับ Accelerometer ให้แสดงภาพแนวนอน เมื่อระบุค่า UIInterface OrientationLandscapeRight เท่านั้น) นอกจากนี้ นักพัฒนายังสามารถสร้างรูปแบบ URL พิเศษที่ใช้ในโปรแกรมที่สร้างได้ โดยระบุค่า CFBundleURLTypes

ไอคอนและภาพดีฟอลต์

ไฟล์ภาพสำคัญ 2 ไฟล์ ที่ต้องใช้ในโปรแกรมของ iPhone ได้แก่ Icon.png และ Default.png โดย Icon.png จะเป็นภาพไอคอนของโปรแกรม ซึ่งจะแสดงไว้ในหน้าหลักของ SpringBoard ส่วน Default.png คือภาพที่แสดงบนหน้าจอขณะเรียกให้โปรแกรมเริ่มทำงาน มักจะรู้จักในชื่อ SplashScreen หรือ Launch Image สำหรับ Default.png นักพัฒนาจะต้องตั้งชื่อไฟล์ด้วยชื่อนี้เท่านั้น แต่ในกรณีของภาพไอคอนอาจไม่ต้องใช้ชื่อไฟล์ว่า

Icon.png ก็ได้ ถ้านักพัฒนาอยากใช้ชื่อไฟล์อื่นๆดีสามารถระบุชื่อไฟล์นั้นไว้ใน CFBundleIconFile ใน Info.plist ได้

Apple แนะนำให้สร้างภาพ Default.png ให้มีสีสันและหน้าตาไปกันได้กับสีพื้นหลังของโปรแกรม โดยนักพัฒนาส่วนมากใช้ Default.png เป็นภาพแสดงโลโก้ของโปรแกรมหรือเป็นหน้าจอที่เขียนว่า "กรุณารอสักครู่" ซึ่งไม่ตรงกับจุดประสงค์ที่ Apple แนะไว้มาก แต่ก็เป็นวิธีที่ยอมรับได้ ซึ่ง Apple ได้ระบุไว้ว่าภาพ Launch Image มีหน้าที่แสดงความต่อเนื่องของกระบวนการทำงานของโปรแกรม ไม่ควรใช้เป็นโฆษณาหรือใช้แก้ขัดถ้าโปรแกรมเริ่มทำงานได้ช้า

ตามคำแนะนำของ Apple นั้น ภาพไอคอนต้องมีขนาด 57x57 พิกเซล โดย SpringBoard จะปรับสัดส่วนของไอคอนที่ใหญ่กว่านี้ให้โดยอัตโนมัติ นักพัฒนาไม่จำเป็นต้องสร้างภาพไอคอนให้มีขอบมนหรือเงาสะท้อน เพราะ SpringBoard จะใส่ลูกเล่นพวกนี้เอง นักพัฒนาเพียงแค่สร้างภาพสี่เหลี่ยมของตรง และไอคอนแบบปกติที่ไม่ต้องใส่แสงเงาก็พอ ในกรณีพิเศษที่ต้องการใส่ลูกเล่นในภาพ จะต้องเลือกปิดการทำงานส่วนของ SpringBoard โดยตั้งค่า UIPrerenderedIcon ในไฟล์ Info.plist ให้เป็น <true/> แต่ถ้าหากต้องการจะส่งโปรแกรมประยุกต์ขึ้นไปแจกจ่ายบน App Store ของ Apple จะต้องเตรียมภาพไอคอนขนาด 512x512 พิกเซลเพิ่ม โดยอาจนำภาพขนาด 57x57 พิกเซล มาขยาย หรือเริ่มสร้างจากการสร้างภาพความละเอียดสูงก่อนลดขนาดในภายหลัง

ไฟล์ XIB หรือ NIB

Interface Builder จะสร้างไฟล์ XIB หรือ NIB บนเครื่องแม็ค ซึ่งเก็บคลาสของส่วนติดต่อผู้ใช้ที่ได้สร้างรอไว้แล้ว ไฟล์ชนิดนี้จะถูกเก็บไว้ที่โฟลเดอร์ชั้นนอกสุดของโปรแกรม และจะถูกเรียกใช้จากตัวโปรแกรมประยุกต์โดยตรง ซึ่งนามกุล .xib(Xcode Interface Builder) ถูกใช้เป็นนามสกุลไฟล์มาตรฐาน

ไฟล์ที่ไม่ถูกรวมอยู่ในโฟลเดอร์ของโปรแกรม

ไฟล์บางอย่างจะไม่ถูกรวมอยู่ในโฟลเดอร์ของโปรแกรม เช่น ไฟล์สำหรับการตั้งค่า และปรับแต่ง มักจะถูกเก็บไว้ในโฟลเดอร์ย่อยของ /Library/Preference ปลั๊กอินของโปรแกรม(ปัจจุบันเก็บไว้ใน /System/Library และนักพัฒนาทั่วไปยังไม่สามารถเรียกใช้ได้) และไฟล์เอกสารของโปรแกรม(เก็บไว้ในโฟลเดอร์ย่อยของ Documents)

โฟลเดอร์อีกอันที่เคยมีอยู่ในโปรแกรมของแม็คอินทอช แต่ไม่มีใน iPhone คือ โฟลเดอร์ Application Support ซึ่งถูกย้ายไปไว้ในโฟลเดอร์ Documents หรือ Library แทน

Sandbox

ระบบปฏิบัติการ iPhone OS นั้นจำกัดสิทธิ์ของโปรแกรมทุกตัวที่พัฒนาด้วย iPhone SDK ให้ทำงานแบบ Sandbox เพื่อความปลอดภัยของระบบ Sandbox หรือ iPhone จะจำกัดไม่ให้โปรแกรมเข้าถึงระบบไฟล์ทั้งหมดโดยตรง เข้าถึงได้แค่บางโฟลเดอร์เท่านั้น รวมถึงเรียกใช้ฮาร์ดแวร์ และเชื่อมต่อเครือข่ายได้จำกัดเช่นกัน การทำงานเปรียบได้กับโรงเรียนที่มีกฎระเบียบเข้มงวดแบบสุดๆ

- โปรแกรมทำงานได้อย่างเต็มที่ภายใน Sandbox ของโปรแกรมนั้นๆ แต่ไม่สามารถเข้าถึง Sandbox ของโปรแกรมอื่นได้
- นักพัฒนาสามารถแชร์ข้อมูลกับโปรแกรมตัวอื่นๆได้ โดยไฟล์จะต้องถูกเก็บไว้ในโฟลเดอร์ที่ Sandbox เตรียมไว้ให้ และนักพัฒนาไม่สามารถคัดลอกไฟล์ไปยังโฟลเดอร์ของตัวโปรแกรมอื่นๆได้
- โปรแกรมจะมีโฟลเดอร์ Library Documents และ /tmp ของตัวเอง จุดนี้คล้ายกับโฟลเดอร์มาตรฐานบนระบบปฏิบัติการอื่นๆที่เข้มงวดน้อยกว่านี้ แต่ว่าจะจำกัดสิทธิ์ในการอ่าน-เขียนข้อมูลของโปรแกรม

นอกจากข้อจำกัดเหล่านี้ โปรแกรมของนักพัฒนาจะต้องผ่านการรับรองดิจิทัล (Digital Signation) และแสดงตัวต่อระบบปฏิบัติการด้วยรหัส Identifier ซึ่งต้องขอจากเว็บไซต์นักพัฒนาของ Apple ถ้ามองในแง่ดี ระบบ Sandbox จะรับประกันว่าเมื่อทำการเสียบ iPhone เข้ากับคอมพิวเตอร์ โปรแกรมทุกตัวจะซิงค์ข้อมูลกับคอมพิวเตอร์ได้อย่างถูกต้องเสมอ แต่ในแง่ลบ Apple ยังไม่ได้เผยข้อมูลว่า จะเปิดข้อมูลที่ซิงค์ระหว่างเครื่องได้อย่างไรบนพีซีหรือแม็ค

2.2.7 ข้อจำกัดของแพลตฟอร์ม

สำหรับ iPhone นั้น นักพัฒนาไม่สามารถออกแบบโปรแกรมที่ใช้พื้นที่หน้าจอขนาดใหญ่ ใช้เมาส์ และคีย์บอร์ดในการป้อนข้อมูล และคาดหวังว่าตัวเครื่องจะเสียบไฟอยู่ตลอดเวลา ในทางตรงกันข้ามข้อจำกัดทางกายภาพของแพลตฟอร์มกลับจำกัดอิสรภาพใน

การพัฒนาซอฟต์แวร์ด้วยซ้ำ ซึ่ง Apple ได้สร้างแพลตฟอร์ม iPhone ที่มีความยืดหยุ่นต่อการพัฒนา แม้จะมีข้อกำจัดทางกายภาพก็ตาม

ข้อจำกัดด้านพื้นที่เก็บข้อมูล

iPhone ใช้ระบบปฏิบัติการ OS X ที่ออกแบบมาสำหรับมือถือ และมีขนาดไฟล์ทั้งหมดอยู่ในหลักไม่กี่ร้อยเมกะไบต์ ซึ่งถือว่าเล็กมากเมื่อเทียบกับระบบปฏิบัติการทั่วไป ในยุคนี้ระบบปฏิบัติการ OS X มาพร้อมกับไลบรารี่ของเฟรมเวิร์คให้ใช้งานเป็นจำนวนมาก เช่น เล่นไฟล์เสียง อีเมล์ บราวเซอร์ และ iPhone ได้เตรียมวิธีการสร้างส่วนติดต่อผู้ใช้ที่ยืดหยุ่น แต่เก็บลงในไฟล์ขนาดเล็ก เพื่อให้เหมาะกับเนื้อที่เก็บข้อมูลที่จำกัด

ข้อจำกัดด้านการเข้าถึงข้อมูล

โปรแกรมทุกตัวบน iPhone จะทำงานในโหมด Sandbox ซึ่งจำกัดให้เข้าถึงระบบไฟล์ของ iPhone ได้เพียงบางส่วนเท่านั้น โปรแกรมประยุกต์ที่สร้างขึ้นจะไม่สามารถเข้าถึงได้จากโปรแกรมอื่นๆ และไม่สามารถเข้าถึงโฟลเดอร์ที่ใช้เก็บไฟล์ของระบบ อย่างเช่น ไลบรารี่ของ iTunes สามารถเข้าถึงไฟล์ที่อยู่อินเทอร์เน็ตได้อย่างอิสระถ้า iPhone เชื่อมต่ออินเทอร์เน็ตอยู่

ข้อจำกัดด้านหน่วยความจำ

การจัดการหน่วยความจำถือว่าเป็นเรื่องที่สำคัญบน iPhone เพราะว่า iPhone นั้นไม่มีระบบ swap เนื้อที่บนดิสก์มาใช้เป็นหน่วยความจำสำรอง ดังนั้น หากหน่วยความจำหมด iPhone จะรีบูตตัวเอง ดังนั้น นักพัฒนาจะต้องจัดการหน่วยความจำอย่างระวัง และระบบปฏิบัติการของ iPhone จะสั่งปิดโปแกรมของนักพัฒนาทิ้ง หากพบว่าบริโภคหน่วยความจำมากเกินไป สิ่งที่ช่วยได้ คือ การตรวจสอบขนาดของภาพ หรือไฟล์เสียงที่คุณใช้โปรแกรมไม่ให้ใหญ่เกินไป

ข้อจำกัดด้านวิธีการป้อนข้อมูล

ถึงแม้ว่าจอภาพของ iPhone จะมีขนาดเล็ก และไม่มีคีย์บอร์ดมาให้ในตัว แต่ความยืดหยุ่นของการป้อนข้อมูลก็ไม่ได้เสียไป เพราะว่าจอสัมผัสแบบมัลติทัชของ iPhone ช่วยให้นักพัฒนาสร้างส่วนติดต่อผู้ใช้แบบใหม่ๆ เทคโนโลยีระบบสัมผัส iPhone ได้สร้าง "วิธีการป้อนข้อมูลเสมือน" ซึ่งมีพื้นที่ใช้งานโดยรวมกว้างกว่าคีย์บอร์ดจริงที่มีขนาดเท่ากับตัวเครื่อง iPhone

เทคโนโลยีที่สำคัญ 2 อัน ที่ทำให้ iPhone แตกต่างจากแพลตฟอร์มมือถืออื่นๆในท้องตลาด ได้แก่ คีย์บอร์ดเสมือนบนหน้าจอที่มีระบบแก้ไขคำผิดที่ชาญฉลาด และ Accelerometer ซึ่งจะตรวจจับว่า iPhone เอียงอยู่ในทิศทางใด และปรับหน้าจอเป็นแนวนอนให้อัตโนมัติ

นักพัฒนาควรหลีกเลี่ยงการป้อนข้อมูลที่เป็นข้อความจนล้นหน้าต่าง และต้องเลื่อนลงเพื่อดูข้อมูล แล้วหันมาเน้นส่วนติดต่อผู้ใช้ที่ใช้กา “แตะ” หน้าจอเป็นหลัก แทนที่จะเลียนแบบวิธีทำงานของโปรแกรมคอมพิวเตอร์อย่างที่เคยเป็นมา ซึ่งสามารถเปิดโปรแกรมได้ทีละหนึ่งหน้าต่างเท่านั้น จุดนี้เป็นความแตกต่างที่สำคัญเมื่อเทียบกับการทำงานบนคอมพิวเตอร์

ข้อจำกัดด้านพลังงาน

การพัฒนาโปรแกรมบนมือถือไม่ว่าจะเป็นแพลตฟอร์มใดๆนั้นไม่สามารถมองข้ามเรื่องพลังงานไปได้ ชุดพัฒนา iPhone SDK มีฟีเจอร์ที่ช่วยจำกัดการใช้งานของซีพียู เพื่อยืดอายุของแบตเตอรี่ด้วยการเรียกใช้ฟีเจอร์อย่างเหมาะสม เช่น สั่งโปรแกรมหลับเมื่อไม่ใช้งานเป็นเวลานาน ทำให้โปแกรมของนักพัฒนาน่าใช้ขึ้นมาก โปรแกรมประยุกต์บางตัวที่ทำงานอยู่ตลอดเวลาถึงแม้ว่าจะไม่ถูกใช้งาน อาจมีผลทำให้ตัวเครื่องร้อนจนใช้มือจับไม่ได้ และเปลืองแบตเตอรี่อย่างมาก

ข้อจำกัดโปรแกรม

Apple บังคับโปรแกรมทำงานทีละตัวอย่างเข้มงวดมาตั้งแต่แรก หมายความว่าในฐานะนักพัฒนาที่ไม่ใช่ของ Apple จะไม่สามารถสร้างโปรแกรมที่ทำงานเบื้องหลังได้ เช่นเดียวกับโปรแกรม Mail และ iPhone ของ Apple ทุกครั้งที่ปิดโปรแกรม จะต้องทำความสะอาดตัวเองก่อนจะส่งสิทธิ์ในการทำงานให้กับโปรแกรมถัดไปที่ถูกเรียกใช้งาน ไม่สามารถรันโปรแกรม Demon ที่คอยตรวจสอบอีเมล์ใหม่ หรือส่งข้อมูลขึ้นอินเทอร์เน็ตเป็นระยะๆค้างไว้ได้ ถึงแม้มีนักพัฒนาบางกลุ่มได้สร้างชุดพัฒนา “Open SDK” ที่ปิดข้อจำกัดพวกนี้ทิ้งไป แต่โปรแกรมที่สร้างด้วยชุดพัฒนานี้ไม่สามารถส่งขึ้นไปบน iPhone App Store เพื่อขายหรือแจกได้

2.2.8 การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming)

ภาษาโปรแกรมเชิงวัตถุนั้นพัฒนามาจากภาษา Smalltalk ซึ่งเป็นภาษาที่สำคัญมากภาษาหนึ่งในโลกของการโปรแกรมเชิงวัตถุ และนำแนวคิดของ Smalltalk มาใช้เป็นอย่างมาก การเขียนโปรแกรมเชิงวัตถุและใช้หลักการ Encapsulation (การซ่อนกลไกการทำงานของโปรแกรมไม่ให้ภายนอกได้ล่วงรู้) และ Inheritance (การสืบทอดคุณสมบัติของวัตถุ) เพื่อสร้างคลาสที่สามารถนำกลับมาใช้ซ้ำได้ สามารถสร้างโปรแกรมขึ้นมาจากการนำคลาสที่มีอยู่แล้วมาประกอบเข้าด้วยกัน เพราะว่าตามแนวคิดเชิงวัตถุ คลาสเหล่านี้ถูกสร้างขึ้นโดยคำนึงว่าจะต้องต่อกันสนิทมาตั้งแต่ต้น (ซึ่งอยู่ในส่วนการประกาศใช้คลาสในซอร์สโค้ดของโปรแกรม)

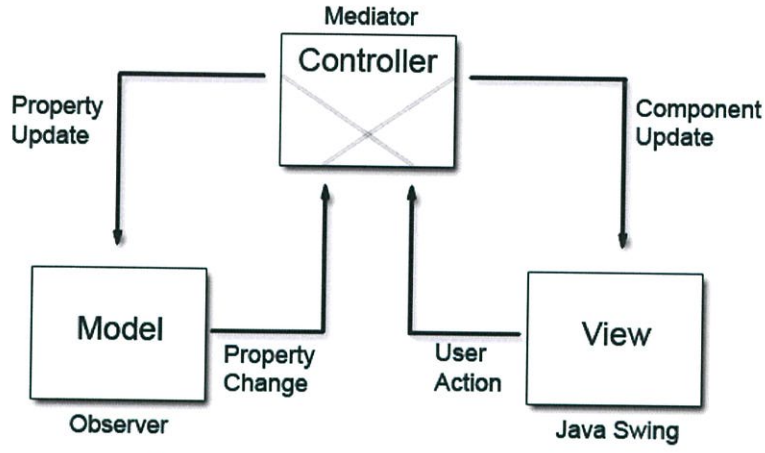
การสืบทอดแบบหลายคลาส หรือ Pseudo-Multiple Inheritance (ผ่านกระบวนการ Invocation Forwarding) ถือเป็นฟีเจอร์ด้านการโปรแกรมเชิงวัตถุที่สำคัญของภาษาโปรแกรมเชิงวัตถุคลาสของ iPhone สามารถสืบทอดพฤติกรรมและชนิดของข้อมูลมาจากคลาสแม่ได้มากกว่าหนึ่งคลาสพร้อมกัน เช่น คลาส UITextView ซึ่งถือเป็นทั้งคลาสแบบ Text และ View สามารถแสดงตัวบนหน้าจอ กำหนดขอบและความโปร่งใสได้เฉกเช่นเดียวกับคลาส View อื่นๆ แต่ในขณะเดียวกันก็มีคุณสมบัติของคลาส Text อย่างการเปลี่ยนแบบขนาดและสีของอักษรด้วยเช่นกัน ทั้งโปรแกรมเชิงวัตถุและ Cocoa Touch ผนวกคุณสมบัติของคลาสเหล่านี้เข้าด้วยกันเป็นคลาสเดียวกันที่ใช้งานสะดวก

2.2.9 หลักการ Model-View-Controller (MVC)

หลักการ MVC คือการแบ่งส่วนของรูปลักษณ์ที่แสดงผลบนหน้าจอออกจากส่วนที่ควบคุมพฤติกรรมการทำงาน ส่วนรูปลักษณ์ที่ถูกเรียกว่า View นั้นไม่มีความหมายหรือคุณค่าใดๆในการทำงาน เป็นเพียงแค่ปุ่มให้ผู้ใช้แตะเท่านั้น เมื่อผู้ใช้แตะปุ่มแล้ว ส่วนที่จะเข้ามารับช่วงต่อคือ Controller ซึ่งจะเชื่อมโยงการแตะปุ่มของผู้ใช้กับกระบวนการทำงานภายในโปรแกรม หรือ Model ผู้เขียนโปรแกรมจะต้องเตรียมว่า ถ้าผู้ใช้มีปฏิสัมพันธ์กับโปรแกรมในแบบต่างๆ เช่น การแตะปุ่มแล้วโปรแกรมจะต้องเก็บข้อมูลหรือตอบรับอย่างไร

แต่ละส่วนใน MVC นั้นทำงานแยกจากกันเป็นอิสระ โดยอาจจะเปลี่ยนปุ่มเป็นสวิตซ์(View)ได้ โดยไม่ต้องสนใจ Model หรือ Controller และโปรแกรมจะทำงานได้ผลลัพธ์ดังเดิมเพียงแต่หน้าตาจะเปลี่ยนไปเท่านั้น ในทางกลับกันอาจคงหน้าตาของ

โปรแกรมไว้แบบเดิมแต่เปลี่ยนวิธีการตอบสนอง เมื่อผู้ใช้แตะปุ่มก็ได้เช่นกัน การแบ่งส่วน MVC ออกจากกันทำให้สามารถสร้างโปรแกรมที่ประกอบด้วยส่วนต่างๆ ซึ่งถูกแก้ไขได้โดยไม่กระทบถึงสิ่งอื่น



รูปที่ 2.18 ภาพรวมการทำงานของหลักการ MVC

หลักการ MVC บน iPhone สามารถแยกได้ดังนี้

Model

Model คือ หัวใจหลักของโปรแกรมประยุกต์ เพราะว่ามันเป็นส่วนที่รับผิดชอบเกี่ยวกับการจัดการข้อมูล ซึ่งแตกต่างจาก View โดย Model นั้นจะรับคำสั่งมาจาก Controller และจะรู้ทุกสิ่งเกี่ยวกับการกระทำของผู้ใช้ หรืออะไรก็ตามที่เห็นบน Display ซึ่งฟังก์ชั่นเฉพาะของ Model คือ ส่วนที่จัดการและดำเนินการเกี่ยวกับข้อมูลของผู้ใช้ที่อยู่ในโปรแกรมประยุกต์ และจะจัดการตรรกะภายในที่มีการจัดเจรียมการกระทำพื้นฐานไว้ให้

View

View เป็นส่วนหลักของส่วนติดต่อผู้ใช้ เช่น ปุ่ม แถบเลื่อน เว็บบราวเซอร์ และสิ่งอื่นๆที่เห็นบนหน้าจอของ iPhone โดยข้อมูลต่างๆ จะถูกส่งมาจาก Controller แล้วนำมาแสดงผลที่ View ถ้ามองด้วยหลัก MVC ระบบ View ของ iPhone จะอยู่ในคลาสสำคัญ 2 คลาส คือ UIView และ UIViewController ซึ่ง 2 คลาสนี้มีหน้าที่สร้างและจัดวางตำแหน่งของส่วนติดต่อผู้ใช้บนหน้าจอ

Controller

เป็นตัวเชื่อมระหว่าง View กับ Model ตอนที่ Apple สร้างติดต่อผู้ใช้อย่างแถบเลื่อนและตารางนั้น Apple ไม่มีแนวคิดว่านักพัฒนาโปรแกรมจะใช้ได้อย่างไร ดังนั้นจึงถูก

สร้างขึ้นแบบกลางๆไม่มีคุณสมบัติอะไรเป็นพิเศษ ซึ่งสอดคล้องกับหลักการ MVC ที่ว่า การแตะปุ่มหรือการเลือกแถวในตารางจะไม่มีความหมายในเชิงโปรแกรมมิ่งแต่อย่างใด สิ่งนี้เป็นหน้าที่ของโปรแกรมเมอร์ในการเชื่อมความหมายของกับ Model ดังนั้น iPhone จึงเตรียมวิธีการเชื่อมต่อคลาสที่ Cocoa Touch เตรียมมาให้แล้วกับคลาสที่สร้างขึ้นมา

2.2.10 โปรแกรมภาษา Objective-C 2.0

Objective-C เป็นภาษาที่ได้รับความนิยมในการเขียนโปรแกรมบน Mac OS X ที่มีรูปแบบการเขียนคำสั่งคล้ายกับภาษา C เนื่องจาก Objective-C เป็น superset หรือภาษาที่อยู่เหนือภาษา C หมายความว่า มีการเพิ่มเติมจากภาษา C เพื่อสร้างภาษาการโปรแกรมที่สามารถสร้างและดูแลจัดการวัตถุได้ (Object-Oriented Programming) โดยจะมีการแบ่งแยกระหว่าง InterFace1 (ส่วนประกาศ) และ Implementation (ส่วนระบุวิธีการทำงาน)

2.2.10.1 ส่วนประกาศ

ส่วนประกาศจะแบ่งไฟล์เป็นส่วนของ header โดยจะมีนามสกุลไฟล์ .h เช่น myClass.h การที่จะสร้าง class ขึ้นมาจะต้องมีการประกาศในส่วนของ interface โดยใช้ @interface นำหน้าชื่อ class ที่ต้องการ แล้วตามด้วยชื่อของ classname และสำหรับ :parentClassname นั้นก็เป็นส่วนที่บอกว่า class นี้สืบทอด (Inheritance) มาจาก class ไหน ในส่วนของ data member นั้นจะประกาศภายในเครื่องหมาย { และ } ส่วน Function จะทำการประกาศนอก {} และปิดท้ายด้วย @end ดังรูปแบบต่อไปนี้

```
@interface classname: parentClassname
{
    // Data member
}
Function
@end
```

2.2.10.2 ส่วนระบุวิธีการทำงาน

ส่วนของการระบุวิธีที่ทำงาน หรือส่วน implement class ที่สร้างขึ้น โดยไฟล์ที่ทำหน้าที่เป็น implementation จะบันทึกโดยใช้นามสกุล .m เช่น myClass.m

```
#import "classname"
@implementation classname
-(void)myMethod:(int) myParameter
{
        //your code here
}
@end
```

2.2.10.3 การประกาศเมธอด

การแปรรูปแบบการประกาศเมธอด มีรูปแบบดังนี้

[methodType](returnType) methodName:(argumentType) argumentName methodName: (argumentType) argumentName ….;

เช่น (void) setNumerator: (int) n;

(void)setTo: (int)n over: (int) d

แต่ส่วนมีความหมายคือ

- [methodType] มี 2 ชนิด คือ

  (-) : เป็นการบอกคอมไฟเลอร์ว่าเมธอดชนิดนี้เป็นเมธอดแบบ instance method จะต้องทำการเรียกผ่านออบเจ็กต์

  (+) : เป็นการบอกคอมไฟเลอร์ว่าเมธอดชนิดนี้เป็นเมธอดแบบ class method สามารถเรียกได้ โดยอ้างอิงผ่านคลาส

- (returnType) เป็นผลลัพธ์จากการทำงานของวัตถุตามข้อความที่ได้รับ ซึ่งผลที่ได้จะเป็นวัตถุเป็นค่าพื้นฐานหรือไม่มีอะไรก็ได้

  กรณีที่เป็นวัตถุ จะคืนค่าเป็น (ชนิดของวัตถุ *) เช่น (NSString*) (GameModel*)

  กรณีที่เป็นค่าพื้นฐาน จะคืนค่าเป็น (ชนิด)เช่น (int)(double)(char)

  กรณีไม่คืนค่าใดๆ จะใช้ (void)

- methodName: ชื่อเมธอด

- (arugmentType) ชนิดของ arugment

- arugmentName ชื่อของ arugment ที่ส่งไป

2.2.10.4 การประกาศออบเจ็กต์

มีรูปแบบดังนี้ ClassName * VariableName = [[ClassName alloc]init];

ตัวอย่างโปรแกรม เช่น

```
Book *scibook;
scibook = [[Book alloc];
scibook init];
```

เริ่มประกาศตัวแปร scibook ให้เป็น pointer ของ book และทำการส่ง message ที่ชื่อว่า alloc ไปให้ Book เพื่อทำการจอง memory แล้วหลังจากการจอง ยังไม่สามารถใช้ได้เลย จ้องทำการ initialize ก่อน ด้วยการส่ง meesage บอกให้ initialize อีกครั้ง หรือจะรวมการเขียนโค้ดเป็นบรรทัดเดียวได้ดังนี้

```
Book*mathbook= [[Book alloc] init];
```

2.2.10.5 การเรียกเมธอด

มีรูปแบบดังนี้ [object methodName]

สำหรับเมธอดที่ส่งพารามิเตอร์ไปให้

[object methodName:parameterName1 methodName:parameterName2….]

ตัวอย่างเช่น [self setPlayerName:name];

[aFraction setTo: 100 over:200];

2.3 ระบบงานที่เกี่ยวข้อง

2.3.1 การเรียนรู้ของเครื่องจักร (Machine Learning)

ในเทคโนโลยีของปัญญาประดิษฐ์มักเกี่ยวข้องกับการควบคุมเครื่องจักรหรือเครื่องคอมพิวเตอร์ให้สามารถทำงานได้ตามที่ต้องการ มีประสิทธิภาพเสมือนการตัดสินใจโดยมนุษย์ เครื่องจักรหรือคอมพิวเตอร์เหล่านั้นจำเป็นต้องมีการเรียนรู้สถานการณ์ต่างๆ เพื่อให้เข้าใจและตัดสินแก้ไขปัญหาให้กับมนุษย์ได้ เรียกกระบวนการนี้ว่า “การเรียนรู้ของเครื่องจักร (Machine Learning)” ซึ่งเป็นเทคโนโลยีที่มุ่งเน้นการพัฒนาและออกแบบ

อัลกอริธึมที่สนับสนุนให้เครื่องจักรตัดสินใจแทนมนุษย์ได้ สำหรับหัวข้อนี้จะแนะนำให้รู้จักกับการเรียนรู้ของเครื่องจักร และวิธีการสร้างเทคนิคการเรียนรู้

2.3.1.1 ทำความรู้จักกับการเรียนรู้ของเครื่องจักร

ในปัญญาประดิษฐ์การเรียนรู้เป็นปัจจัยสำคัญที่ทำให้เครื่องคอมพิวเตอร์หรือเครื่องจักรสามารถตัดสินใจ เข้าใจ และแก้ไขปัญหาตามสถานการณ์ที่เกิดขึ้นได้ โดยทั่วไปเครื่องคอมพิวเตอร์หรือเครื่องจักรจำเป็นต้องอาศัยมนุษย์เป็นผู้ควบคุมการทำงานทุกขั้นตอน ตั้งแต่การป้อนข้อมูล จนถึงการแสดงผลลัพธ์ ด้วยเหตุนี้จึงมีแนวคิดที่จะลดบทบาทการทำงานของมนุษย์ลงด้วยปัญญาประดิษฐ์ ที่มีประสิทธิภาพใกล้เคียงกับสมองมนุษย์ การที่จะให้เครื่องจักรทำงานได้อย่างน่าเชื่อถือ ถูกต้อง และรวดเร็วได้นั้น ปัญญาประดิษฐ์ในเครื่องจักรจะต้องมีการเรียนรู้ข้อมูลต่างๆ เพื่อให้สามารถปฏิบัติงานไปพร้อมกับการพัฒนาประสิทธิภาพการทำงานให้สูงขึ้น เปรียบเสมือนการเรียนรู้ของมนุษย์ที่สามารถจดจำ และศึกษาวิธีแก้ไขปัญหาในสถานการณ์ที่คล้ายคลึงกันได้ ยิ่งมีประสบการณ์มากก็ยิ่งแก้ปัญหาได้อย่างมีประสิทธิภาพมากขึ้น

หากกล่าวถึงการเรียนรู้ของเครื่องจักร ในด้านวิทยาการคอมพิวเตอร์ (Computer Science) จะหมายถึง เทคนิค หรือ กระบวนการที่ใช้สำหรับปรับแต่ง ปรับปรุง อุปกรณ์ เครื่องจักร หรือคอมพิวเตอร์ให้มีความสามารถเพิ่มขึ้น มีพฤติกรรมเฉพาะตัวที่สนับสนุนให้อุปกรณ์เหล่านี้สามารถเรียนรู้ เข้าใจ และศึกษาข้อมูลจากประสบการณ์ หรือการแสดงผลที่ผ่านมาในอดีตได้ เพื่อใช้เป็นข้อมูลสำหรับพัฒนาการทำงานต่อไป จากที่กล่าวมาเป็นความหมายโดยรวมของการเรียนรู้ของเครื่องจักร ซึ่งเป็นวิธีการที่ช่วยให้เครื่องจักรสามารถเรียนรู้ได้เหมือนกับมนุษย์ สำหรับในปัญญาประดิษฐ์คำว่า “การเรียนรู้ของเครื่องจักร” จะมีความหมายที่เฉพาะเจาะจงลงไปถึงการนำเทคนิคต่างๆ มาพัฒนาเครื่องจักรให้มีประสิทธิภาพในการเรียนรู้สิ่งต่างๆ เปรียบเสมือนการสร้างสมองกลให้กับเครื่องจักร เพื่อใช้ในการเรียนรู้และแก้ไขปัญหา

การเรียนรู้ของเครื่องจักรในปัญญาประดิษฐ์ จะเกี่ยวข้องกับการพัฒนาและออกแบบอัลกอริธึม (Algorithm) หรือเทคนิคต่างๆ ที่ช่วยให้เครื่องจักรหรือเครื่องคอมพิวเตอร์มีความสามารถในการเรียนรู้ข้อมูลที่สนใจได้ เพื่อนำไปใช้ในการ

แก้ไขปัญหาเฉพาะหน้าที่ต้องเผชิญ เสมือนกับการเรียนรู้ของมนุษย์ ที่อาศัยประสบการณ์ที่ได้จากเหตุการณ์ต่างๆ ที่เคยผ่านมา เก็บสะสมไว้เป็นความรู้ซึ่งสามารถนำไปใช้ในการแก้ไขปัญหา และคิดค้นวิธีการที่เหมาะสมจนได้ผลลัพธ์ที่ต้องการ

การเรียนรู้โดยทั่วไปแบ่งได้ 2 ชนิด ดังนี้

1. Deductive เป็นการเรียนรู้โดยอาศัยความรู้ที่เป็นจริงอยู่แล้ว มีความจริงเป็นสากลซึ่งทุกคนต่างยอมรับ โดยสามารถคาดการณ์ได้ว่าเหตุการณ์ดังกล่าวจะเกิดขึ้นแน่นอนตามรูปแบบของสิ่งแวดล้อม เช่น ในกล่องใบหนึ่งบรรจุลูกบอลอยู่ 2 ลูก คือ ลูกสีฟ้าและสีแดง แสดงว่าโอกาสที่จะหยิบลูกบอล 1 ลูก เป็นสีใดสีหนึ่งเป็น 1 ใน 2 หรือ 50 เปอร์เซ็นต์ ซึ่งความน่าจะเป็นดังกล่าวทุกคนต่างยอมรับ และเชื่อว่าหากเราหยิบลูกบอล 1 ลูก โอกาสที่จะได้บอลสีใดๆ จะมีเท่ากัน เป็นต้น

2. Inductive เป็นการเรียนรู้จากเหตุการณ์หรือสิ่งที่สนใจ โดยทราบข้อมูลหรือค่าความจริงเพียงบางส่วน ซึ่งจะนำมาใช้เป็นข้อมูลในการศึกษาและทำความเข้าใจเพื่อให้ได้ความจริงของข้อมูลส่วนอื่นๆ ทั้งหมด จนเป็นความจริงสากลที่คนส่วนใหญ่ยอมรับ เช่น การเรียนรู้ของพนักงานขาย โดยศึกษาจากพฤติกรรม ลักษณะของลูกค้า และความสนใจในขณะที่นำเสนอสินค้า เพื่อค้นหาความต้องการที่แท้จริงของลูกค้า ช่วยให้ทราบถึงแนวทางการขายสินค้าแก่ลูกค้าให้ประสบผลสำเร็จ ซึ่งเป็นการศึกษาจากข้อมูลเพียงบางส่วน จนทราบถึงความต้องการที่แท้จริงของลูกค้า เป็นต้น

สำหรับการเรียนรู้ของเครื่องจักร จะเป็นการเรียนรู้แบบ Inductive ซึ่งเป็นการเรียนรู้จากข้อมูลเพียงบางส่วนก่อนจะดำเนินการหาความจริงที่เป็นสากล ในกระบวนการของการเรียนรู้ของเครื่องจักร จะมีการนำเทคนิคหรืออัลกอริธึมที่มีรูปแบบที่แตกต่างกันมาแก้ไขปัญหาในลักษณะต่างๆ ซึ่งอัลกอริธึมในการเรียนรู้ของเครื่องจักรมีอยู่หลายชนิด สำหรับในหัวข้อนี้จะกล่าวถึงอัลกอริธึมเพียงบางชนิดเท่านั้น ได้แก่ Supervised Learning และ Unsupervised Learning

2.3.1.2 Ensemble Learning

ปัญญาประดิษฐ์จะต้องเรียนรู้และวิเคราะห์ข้อมูลที่มาจากหลายสมมติฐานเพื่อการตัดสินใจ กล่าวคือข้อมูลหนึ่งชุดอาจสร้างเป็น Decision Tree ที่มีกิ่งก้านและโหนดจำนวนมาก จึงจำเป็นต้องมีการจำแนก แยกแยะ จัดกลุ่ม รวมถึงคัดเลือกตัวอย่างที่ดีที่สุด สำหรับนำไปใช้ในงานต่อไปตามกระบวนการหรือขั้นตอนการเรียนรู้ ด้วยเหตุนี้จึงมีทฤษฎีที่ช่วยสนับสนุนให้ปัญญาประดิษฐ์ที่พัฒนาขึ้นสามารถรวบรวมและวิเคราะห์สมมติฐาน เพื่อคัดเลือกข้อมูลที่เหมาะสมและมีประสิทธิภาพได้ ทฤษฎีนี้เรียกว่า "Ensemble Learning" [Russell and Norvig, 2003]

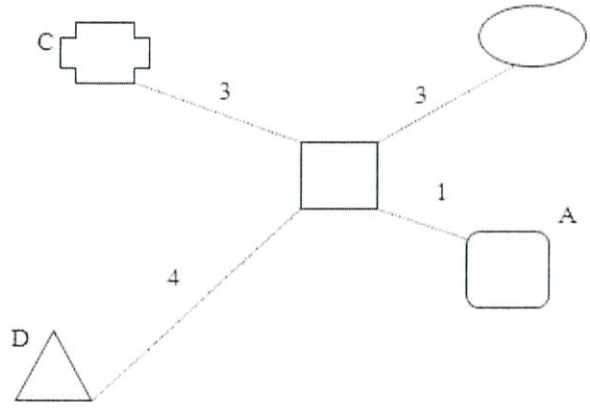
สมมติฐานใน Ensemble Learning จะถูกจัดกลุ่มและรวบรวมก่อนที่จะนำไปใช้วิเคราะห์ เนื่องจากต้องทำการคัดเลือกกลุ่มหรือชุดของข้อมูลจากสมมติฐานต่างๆ ที่มีความเป็นไปได้และเหมาะสมที่สุดในการนำไปวิเคราะห์ เพื่อหาข้อมูลที่ดีที่สุดต่อไป โดยแต่ละสมมติฐานจะมีส่วนที่สัมพันธ์กันและสามารถนำมาใช้เป็นข้อมูลหรือแนวทางในการเรียนรู้ ซึ่งเป็นส่วนที่เกี่ยวข้องกับสิ่งที่สนใจอย่างแท้จริง หลักการของ Ensemble Learning นี้ช่วยให้การเรียนรู้มีประสิทธิภาพสูงขึ้น รวมถึงสามารถแก้ไขปัญหาที่มีความซับซ้อนมากได้

สำหรับวิธีการที่นิยมนำทฤษฎีของ Ensemble Learning มาใช้ คือ "Boosting" ซึ่งเป็นหลักการที่อาศัยการกำหนดน้ำหนักให้กับชุดข้อมูล หรือที่เรียกว่า "Weighted Training Set" โดยในแต่ละชุดข้อมูลจะมีค่าน้ำหนักกำหนดให้มากกว่าหรือเท่ากับศูนย์ ค่าน้ำหนักของชุดข้อมูลใดสูงกว่าแสดงว่ามีความสำคัญที่จะต้องพิจารณาเป็นอันดับแรกในกระบวนการเรียนรู้ อีกทั้งชุดข้อมูลดังกล่าวยังบ่งบอกความสำคัญของสมมติฐานที่เกี่ยวข้องกับชุดข้อมูลนั้นได้ด้วย หลักการ Boosting จะเริ่มด้วยการกำหนดค่าน้ำหนักให้ชุดข้อมูลเท่ากับ 1 ทั้งหมด ซึ่งค่าดังกล่าวจะทำให้เกิดข้อสมมติฐาน h_1 ขึ้น โดยในชุดข้อมูลนี้เมื่อทำการวิเคราะห์แล้วจะมีข้อมูลทั้งข้อมูลที่ถูกและผิดปะปนกัน ทำให้หลังการวิเคราะห์ h_1 ค่าน้ำหนักของชุดข้อมูลจะเปลี่ยนแปลงไป มีทั้งเพิ่มขึ้นและลดลงเพื่อให้สามารถจัดกลุ่มชุดได้ถูกต้องและเหมาะสม ซึ่งจะดำเนินการวิเคราะห์ต่อไปในขั้นตอนของ h_2 และขั้นตอนต่อไปจนกว่าจะได้ตามที่ต้องการ

ในการปรับค่าน้ำหนักสามารถทำได้หลายรูปแบบทั้งลดค่าน้ำหนัก และเพิ่มค่าน้ำหนัก ปัจจัยเหล่านี้ช่วยให้ชุดข้อมูลมีค่าน้ำหนักที่เหมาะสมมากที่สุด ทฤษฎีการรวมชุดข้อมูลหรือสมมติฐานเข้าด้วยกันตามหลักของ Ensemble Learning ช่วยให้การพยากรณ์และการวิเคราะห์ข้อมูลมีประสิทธิภาพสูงขึ้น เนื่องจากมีการปรับเปลี่ยนน้ำหนักของชุดข้อมูลเพื่อให้มีความเหมาะสมอยู่เสมอ ทำให้ข้อผิดพลาดหรือชุดข้อมูลที่ส่งผลให้เกิดความคลาดเคลื่อนมีลดน้อยลง ดังนั้นจึงช่วยให้การพัฒนาระบบการเรียนรู้แสดงผลและสามารถแก้ไข พร้อมทั้งค้นหาแนวทางให้เหมาะสมกับปัญหาได้ดียิ่งขึ้น

2.3.1.3 Nearest Neighbor Classification

เป็นการเรียนรู้ประเภท Unsupervised Learning ซึ่งเป็นการจำแนกหรือจัดกลุ่มที่มีวิธีการไม่ซับซ้อนมาก โดยพิจารณาจากชุดข้อมูลใกล้เคียงกับค่าของข้อมูลที่พิจารณามากที่สุด ในที่นี้ค่าความใกล้เคียงจะหมายถึงระยะทาง (Distance) ที่มีค่าน้อยที่สุดระหว่างชุดข้อมูลกับข้อมูลที่พิจารณา ข้อมูลดังกล่าวนี้เรียกว่า "Nearest Neighbor" " ดังรูปที่ 2.19



รูปที่ 2.19 แสดงตัวอย่างระยะทางระหว่างชุดข้อมูล [Jones, 2008]

จากรูปที่ 2.19 ชุดข้อมูลที่มีระยะทางใกล้กับศูนย์กลางมากที่สุด คือ A ซึ่งมีระยะเป็น 1 ในทฤษฎีของ Nearest Neighbor การจำแนกชุดข้อมูลที่มีระยะห่าง 1 จะเรียกว่า "1NN (One Nearest Neighbor)" โดยระยะห่างของข้อมูลนั้นสามารถกำหนดได้ว่าต้องการมากน้อยเพียงใด ด้วยเหตุนี้จึงมีการจำแนกที่เรียกว่า "k-NN" ซึ่ง k แทนด้วยค่าระยะทางระหว่างข้อมูลที่ต้องการ สำหรับการหาค่าระยะทางจะใช้สมการทฤษฎีการวัดค่าของ Euclidean ดังนี้

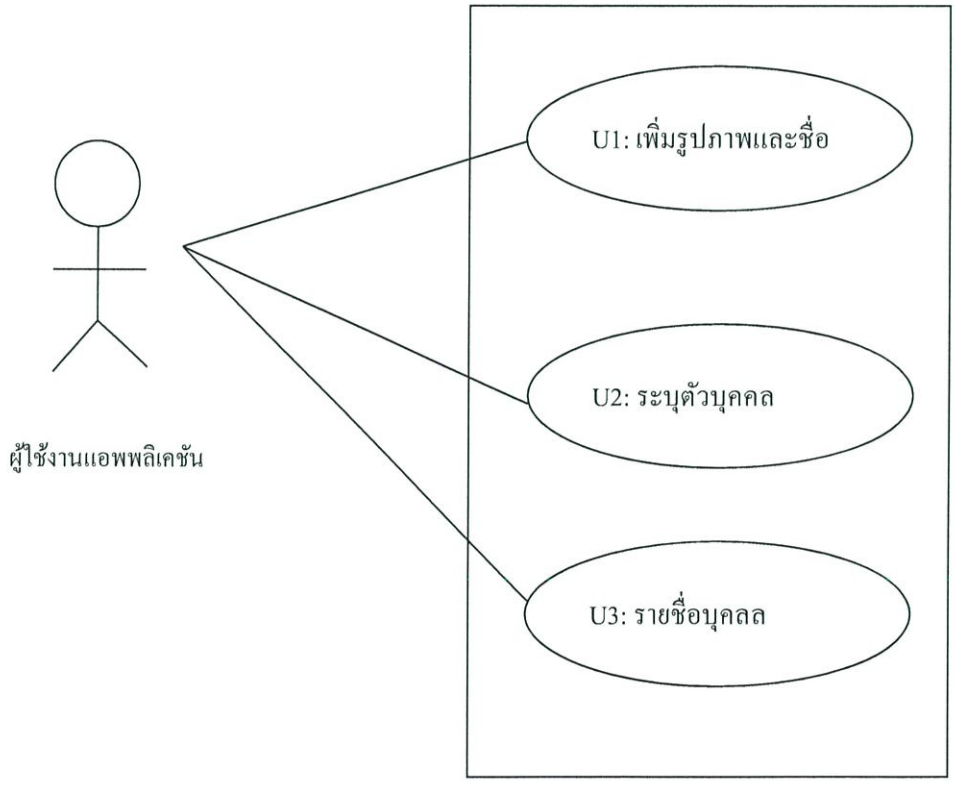
$$d = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

สมการดังกล่าวเป็นการหาระยะทางระหว่างชุดข้อมูล โดยแทนค่าที่บรรจุอยู่ภายในชุดข้อมูลทั้งสอง ซึ่ง p คือ ค่าในชุดข้อมูลที่ต้องการจำแนก มีค่าตั้งแต่ $(p_1, p_2, p_3, ..., p_n)$ และ q คือ ค่าในชุดข้อมูลข้างเคียงที่นำมาพิจารณา มีค่าตั้งแต่ $(q_1, q_2, q_3, ..., q_n)$ ขึ้นอยู่กับจำนวนข้อมูลที่อยู่ภายในชุดข้อมูลนั้น

บทที่ 3
ขั้นตอนการพัฒนาและออกแบบระบบ

ในการพัฒนาแอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคลนั้น จำเป็นต้องมีการวางแผนเพื่อการพัฒนาที่ถูกต้องและถูกวัตถุประสงค์ที่ตั้งไว้ โดยเราต้องคำนึงถึงความเป็นไปได้ว่าจะสามารถพัฒนาแอพพลิเคชันได้ตามแผนที่วางไว้หรือไม่ ดังนั้นการเขียนโครงสร้างของแอพพลิเคชันนี้ก่อนจึงเป็นเรื่องสำคัญที่ควรกระทำ

3.1 Use Case Diagram



รูปที่ 3.1 Use Case Diagram

Use Case Name :	เพิ่มรูปภาพและชื่อ	
Scenario :	เพิ่มรูปภาพใบหน้าและรายละเอียดของบุคคล	
Trigger Event :	ต้องการเพิ่มบุคคลที่อยากจดจำไว้	
Brief Description :	ผู้ใช้เพิ่มรูปภาพใบหน้าจากคลังภาพหรือการถ่ายภาพ และป้อนรายละเอียดชื่อของบุคคล เข้าสู่ฐานข้อมูล SQLite	
Actor :	ผู้ใช้งาน	
Related Use Case :	-	
Stakeholders :	ผู้ใช้งาน	
Precondition :	ต้องใช้ภาพที่มีใบหน้าของบุคคลอยู่ชัดเจน	
Post condition :	บันทึกรูปภาพ	
Flow of Activity :	Actor	System
	1. เข้ามายังแอพพลิเคชัน 2. เลือกเมนูการเพิ่มรูปภาพจากคลังภาพหรือการถ่ายภาพ 3. ป้อนรายละเอียดชื่อของบุคคล	4. บันทึกรูปภาพ
Exception Condition	ถ้าหากในรูปภาพไม่มีใบหน้าบุคคลอยู่ ระบบจะแจ้งเตือน ต้องนำรูปภาพที่มีใบหน้าบุคคลมาใส่แทน จึงจะสามารถเพิ่มรูปภาพใบหน้าบุคคลได้สำเร็จ	

ตารางที่ 3.1 คำอธิบายของ U1: เพิ่มรูปภาพและชื่อ

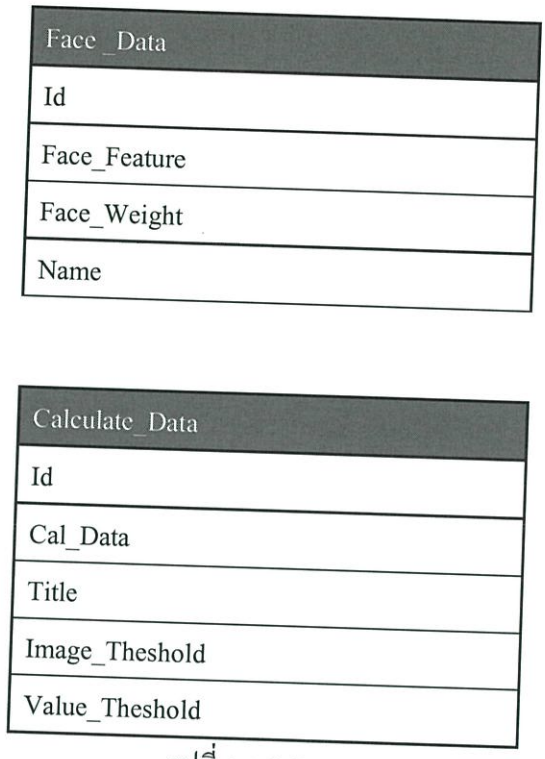
Use Case Name :	ระบุตัวบุคคล	
Scenario :	ตรวจสอบใบหน้าบุคคล	
Trigger Event :	ต้องการรู้ชื่อของบุคคล	
Brief Description :	ผู้ใช้ตรวจสอบใบหน้าบุคคลโดยการใช้กล้องส่องเข้าไปที่ใบหน้าบุคคลที่ผู้ใช้ต้องการรู้ชื่อ	
Actor :	ผู้ใช้งาน	
Related Use Case :	-	
Stakeholders :	ผู้ใช้งาน	
Precondition :	ต้องส่องกล้องไปทางที่มีบุคคลอยู่	
Post condition :	สามารถรู้ชื่อบุคคลหรือไม่ก็ได้ ขึ้นอยู่กับฐานข้อมูล	
Flow of Activity :	Actor	System
	1. เข้ามายังแอพพลิเคชัน 2. เลือกเมนูการตรวจสอบใบหน้า 3. ส่องกล้องไปทางบุคคล	4. แสดงข้อมูลชื่อของบุคคลหรือแสดงว่าไม่ทราบชื่อ (กรณีที่บุคคลนั้นไม่มีอยู่ในฐานข้อมูล)
Exception Condition	-	

ตารางที่ 3.2 คำอธิบายของ U2: ระบุตัวบุคคล

Use Case Name :	รายชื่อบุคคล	
Scenario :	ดูข้อมูลเพื่อแก้ไขรายชื่อและลบข้อมูล	
Trigger Event :	ต้องการรู้รายชื่อพร้อมแก้ไขและลบข้อมูล	
Brief Description :	ผู้ใช้สามารถดูข้อมูลรายชื่อบุคคลที่ทำการบันทึกไว้ และสามารถทำการแก้ไขรายชื่อหรือลบรายชื่อออกจากระบบ	
Actor :	ผู้ใช้งาน	
Related Use Case :	-	
Stakeholders :	ผู้ใช้งาน	
Precondition :	ต้องทำการบันทึกข้อมูลเข้าสู่ฐานข้อมูล SQLite	
Post condition :	สามารถรู้ชื่อบุคคลและแก้ไข้ข้อมูลได้	
Flow of Activity :	Actor	System
	1. เข้ามายังแอพพลิเคชัน 2. เลือกเมนูรายชื่อบุคคล	3. แสดงข้อมูลชื่อของบุคคลทั้งหมดที่มีอยู่ในฐานข้อมูล
Exception Condition	-	

ตารางที่ 3.3 คำอธิบายของ U3: รายชื่อบุคคล

3.2 E-R Diagram



รูปที่ 3.2 E-R Diagram

ตาราง Face_Data

ตารางนี้จะเก็บลักษณะเฉพาะของใบหน้าและชื่อของบุคคล

ตาราง Calculate_Data

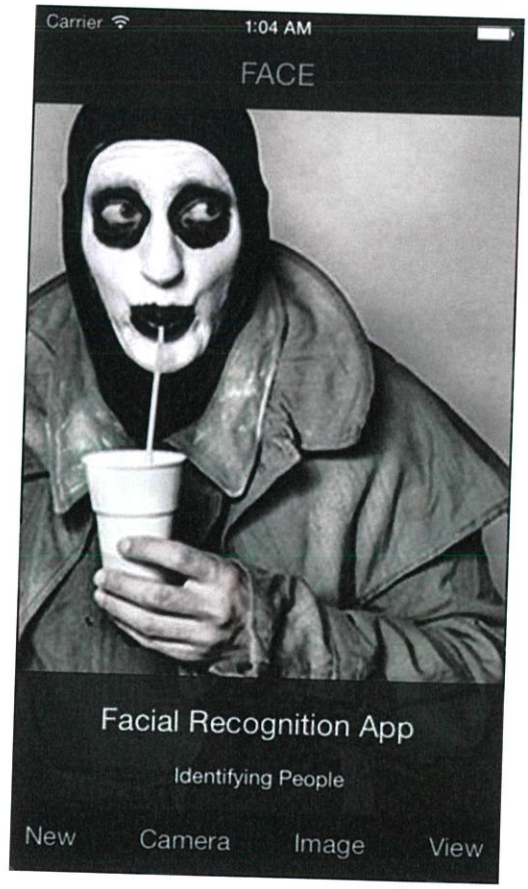
ตารางนี้เก็บข้อมูลที่ใช้สำหรับการคำนวณ

3.3 User Interface

3.3.1 รูปแบบหน้าเมนูหลัก

ประกอบไปด้วยฟังก์ชันการทำงาน 3 ส่วน คือ

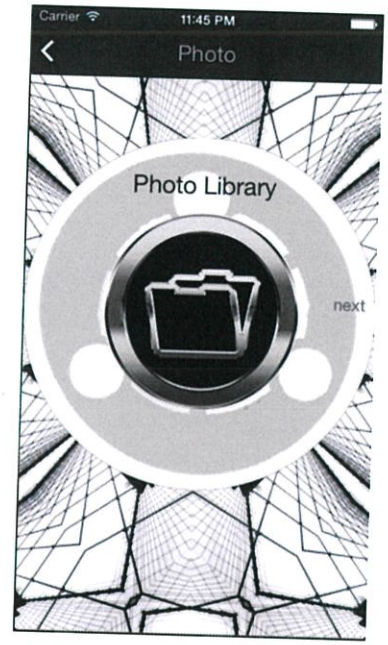
- **New** ส่วนเพิ่มข้อมูลของบุคคลที่ต้องการค้นหา โดยภาพที่ได้จากคลังรูปภาพหรือภาพที่ได้จากการถ่าย
- **Camera/ Image** ส่วนในการค้นหาบุคคลที่ได้ทำการบันทึกข้อมูลลงในฐานข้อมูล โดยสามารถหาได้จากการส่องกล้อง(Camera)หรือค้นหาจากภาพถ่าย(Image)
- **View** บัญชีรายชื่อบุคคลที่ได้ทำการบันทึกข้อมูลลงในฐานข้อมูล



รูปที่ 3.3 หน้าหลักของแอพพลิเคชัน

3.3.2 รูปแบบหน้าอื่นๆ

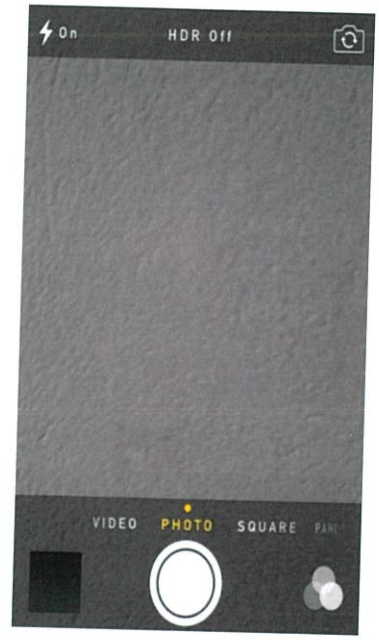
New การเพิ่มบุคคล สามารถเพิ่มบุคคลได้ 2 รูปแบบ แบบแรก คือ Photo Library เป็นการเลือกรูปภาพจากคลังภาพ แบบที่ 2 คือ Camera เป็นการเลือกรูปภาพจากการถ่ายรูป



รูปที่ 3.4 หน้าจอสำหรับเลือกรูปภาพจากคลังรูปภาพ

รูปที่ 3.5 หน้าจอสำหรับเลือกรูปภาพจากการถ่ายภาพ

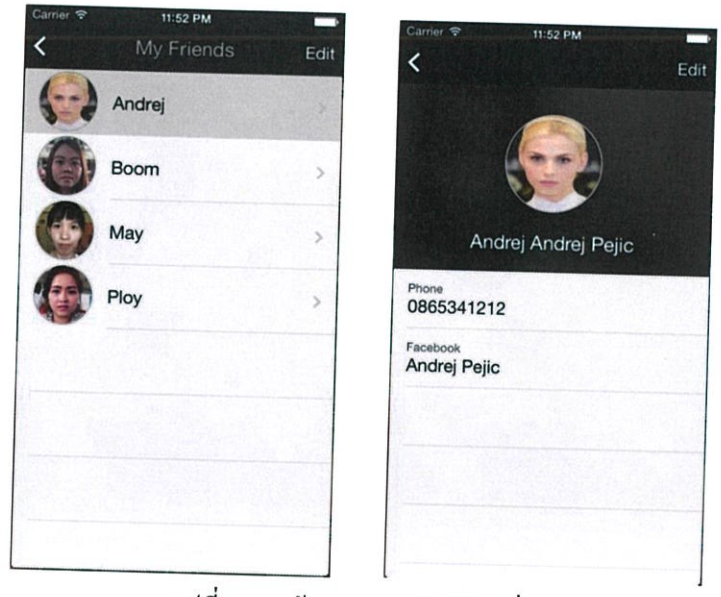
Camera/ Image การค้นหาบุคคล สามารถค้นหาบุคคลได้ 2 รูปแบบ คือ จาก Camera (การใช้กล้องส่องเพื่อค้นหาบุคคล) หรือจาก Image (การค้นหาด้วยภาพถ่าย) ซึ่งบุคคลที่ต้องการค้นหาจะต้องทำการบันทึกข้อมูลลงในฐานข้อมูลก่อนเสมอ



รูปที่ 3.6 หน้าจอสำหรับค้นหาบุคคลจากกล้อง

รูปที่ 3.7 หน้าจอสำหรับค้นหาบุคคลด้วยภาพ

View บัญชีรายชื่อบุคคล แสดงผลรายชื่อบุคคลที่ทำการบันทึกข้อมูลลงในฐานข้อมูล ที่ประกอบไปด้วย ชื่อ นามสกุล ชื่อเล่น เบอร์โทรศัพท์ และ Facebook โดยจะเรียงลำดับข้อมูลตามตัวอักษรที่ได้ทำการบันทึกไว้ในฐานข้อมูล



รูปที่ 3.8 หน้าจอแสดงบัญชีรายชื่อ

บทที่ 4
การพัฒนาแอพพลิเคชันรู้จำใบหน้า

4.1 การสร้างภาพใบหน้าไอเกน

การวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis– PCA)ในการสร้างภาพใบหน้าไอเกน (Eigen Face) เป็นวิธีทางสถิติที่สามารถลดมิติข้อมูลภาพที่ไม่จำเป็นและเลือกใช้ข้อมูลภาพเท่าที่จำเป็นเท่านั้น เริ่มจากภาพแรกที่มีความสำคัญมากที่สุดของภาพใบหน้าทุกๆภาพที่ถูกนำมาคำนวณ ที่จะดึงข้อมูลที่มีความสำคัญของภาพใบหน้าออกมาจากส่วนต่างๆ เช่น โครงภาพใบหน้า จมูก ปาก และตา เป็นต้น และถูกคำนวณในภาพต่อๆไปตามลำดับ โดยที่ภาพลำดับต่อๆไปจะมีความสำคัญน้อยลงๆตามลำดับ ซึ่งในแต่ละภาพจะมีรูปแบบจำเพาะที่แสดงถึงความแตกต่างระหว่างภาพใบหน้าแต่ละบุคคล

4.2 ขั้นตอนการเตรียมรูปภาพที่ใช้ในการเรียนรู้

เริ่มจากนำภาพที่ได้จากการถ่ายจากกล้อง ทำการตรวจจับภาพใบหน้า เมื่อระบบตรวจเจอภาพใบหน้าจะทำการครอบตัดรูปภาพ โดยเอาเฉพาะส่วนที่เป็นใบหน้าเท่านั้น จากนั้นนำภาพที่ได้มาทำการแปลงภาพจากภาพสีให้เป็นภาพโทนเทาหรือ graysale และนำภาพโทนเทาเข้าสู่กระบวนการทำ histogram equalization

4.2.1 การตรวจจับใบหน้า

การหาตำแหน่งของใบหน้า โดยใช้ CIDetector Class จาก CoreImage FrameWork ซึ่งจะได้กรอบของใบหน้าเป็นรูปสี่เหลี่ยมขนาด N X N

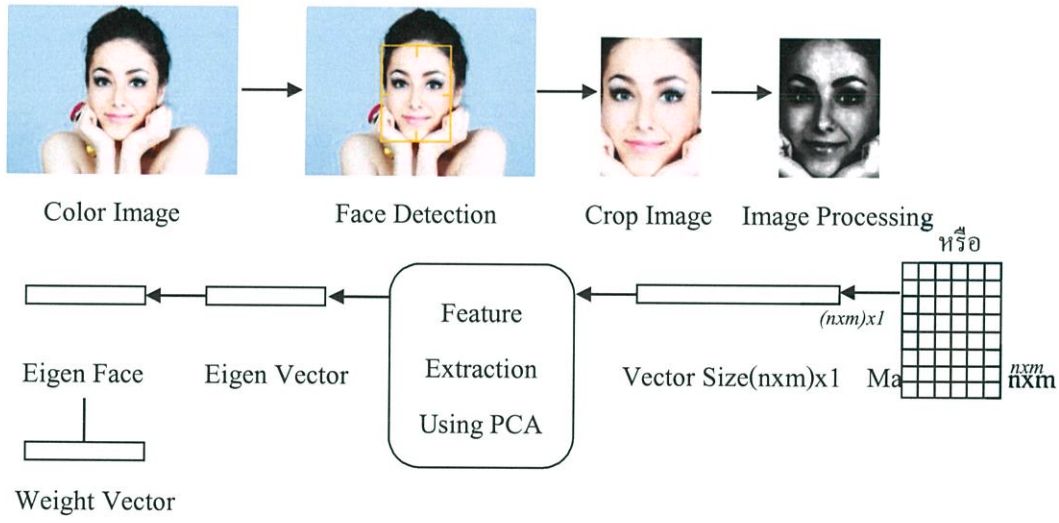
4.2.2 การครอบตัดภาพใบหน้า

หลังจากการสร้างกรอบของใบหน้าแล้ว ในขั้นตอนต่อไปคือการครอบตัดรูปภาพ โดยทำการสร้างรูปใหม่จากรูปเดิม ใช้ตัวกรอบของใบหน้ามาเป็นตัววัดว่าจะสร้างภาพใหม่จากตำแหน่งไหน ขนาดเท่าไหร่

4.2.3 การประมวลผลภาพ

จากการครอบตัดใบหน้า จะได้ภาพใบหน้าที่มีขนาดแตกต่างกัน ซึ่งมีขนาด NxN จึงจำเป็นต้องขยาย/ลดขนาดภาพให้เท่ากัน โดยการสเกลภาพในแอพพลิเคชันนี้ ตั้งให้ขนาดภาพเป็น 100x110 หลังจากที่สเกลภาพตามขนาดที่เหมาะสมแล้ว จะทำการแปลงภาพสีไปเป็นภาพโทนเทา และนำภาพโทนเทาที่ได้ผ่านกระบวนการฮิสโทแกรม เพื่อให้ได้ภาพที่มีการกระจายตัวของแสงที่ดี

4.3 ขั้นตอนการเรียนรู้



รูปที่ 4.1 ขั้นตอนการเรียนรู้

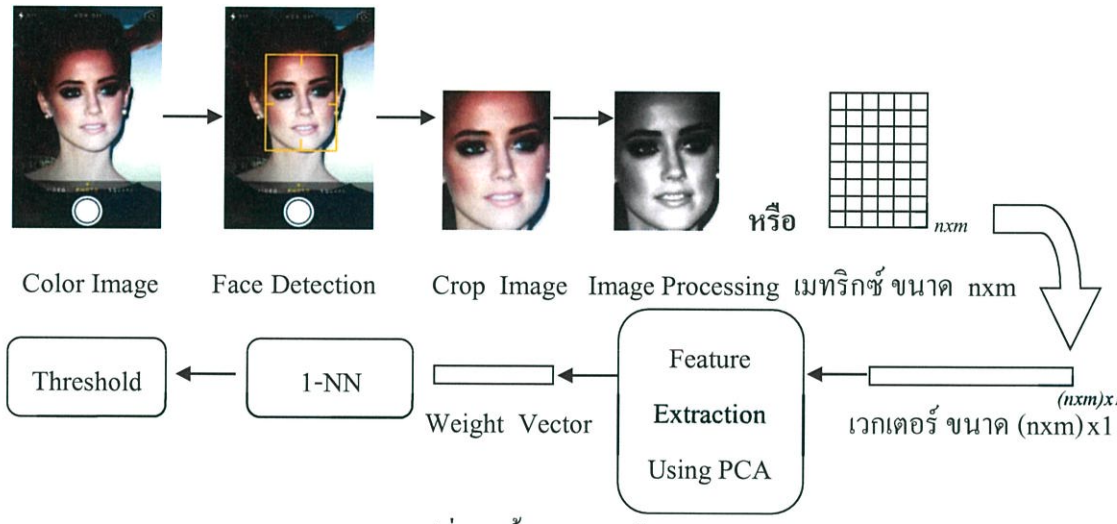
ขั้นตอนที่ 1 : เมื่อผู้ใช้เลือกเมนูเพิ่มบุคคลและชื่อ แล้วส่งเข้าสู่ระบบ ระบบจะทำการตรวจจับใบหน้า หลังจากนั้นจะทำการตัดใบหน้าเฉพาะกรอบของใบหน้า

ขั้นตอนที่ 2 : เมื่อได้กรอบของใบหน้ามาแล้ว ระบบจะทำการประมวลผลภาพ (Image Processing) โดยการเพิ่มหรือลดขนาดภาพให้มีสัดส่วนที่ 100x110 pixels จากนั้นจะแปลงภาพสี (Color Image) ให้เป็นภาพระดับความเข้มเทา (Gray Scale Image) แล้วนำภาพเข้าสู่กระบวนการทำ Histogram Equalization

ขั้นตอนที่ 3 : การทำ Feature Extraction จะใช้วิธีการ PCA เพื่อลดขนาดข้อมูล โดยการดึงคุณสมบัติเด่นของแต่ละบุคคลที่แตกต่างจากบุคคลอื่น ในขั้นตอนนี้จะได้เวกเตอร์ไอเกน หลังจากนั้นระบบจะนำเวกเตอร์ไอเกนไปสร้างเป็นใบหน้าไอเกน (Eigen Face) ต่อ แล้วนำใบหน้าไอเกน

ไปหาเวกเตอร์ค่าน้ำหนักของแต่ละบุคคล เราจะบันทึกเวกเตอร์ค่าน้ำหนักพร้อมชื่อ ลงในฐานข้อมูล

4.4 ขั้นตอนการรู้จำ



รูปที่ 4.2 ขั้นตอนการรู้จำ

ขั้นตอนที่ 1 : เมื่อผู้ใช้เลือกเมนูระบุตัวบุคคล ระบบจะรับภาพจากกล้องแล้วทำการตรวจจับใบหน้า (Face Detection) จนไปถึงขั้นตอนที่ได้เวกเตอร์ขนาด (nxm)x1 เหมือนในขั้นตอนเรียนรู้

ขั้นตอนที่ 2 : หาส่วนต่างของภาพใบหน้าที่ใช้ทดสอบกับค่าเฉลี่ยของทุกภาพในฐานข้อมูล (ภาพใบหน้าทั้งหมดในขั้นตอนการเรียนรู้) แล้วนำส่วนต่างนี้ไปหาเวกเตอร์ค่าน้ำหนัก

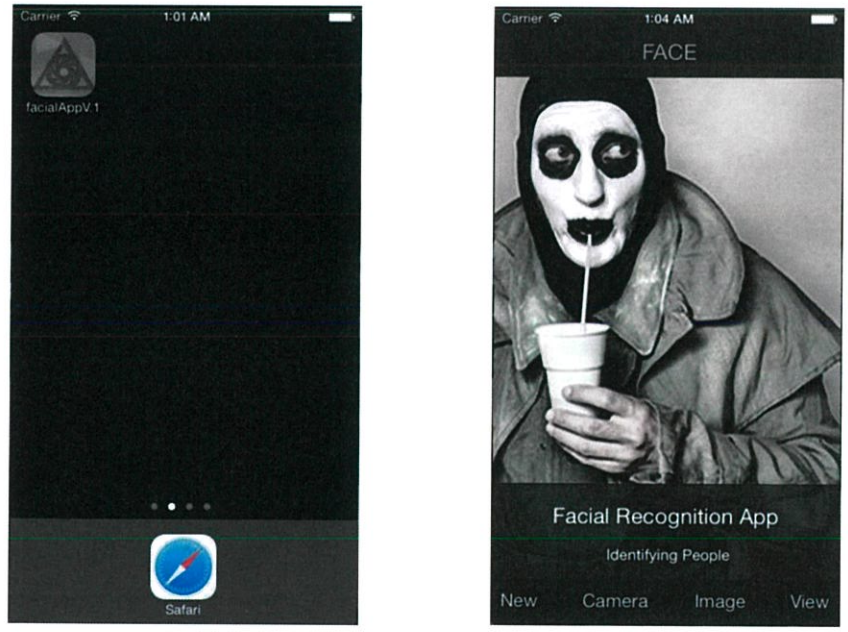
ขั้นตอนที่ 3 : ใช้เทคนิค Nearest Neighbor Classification ซึ่งจะหาความแตกต่างระหว่างเวกเตอร์ค่าน้ำหนักที่ได้รับมาและเวกเตอร์ค่าน้ำหนักที่อยู่ในฐานข้อมูลทั้งหมด ภาพใดที่มีความแตกต่างน้อยที่สุด ย่อมมีโอกาสเป็นคนเดียวกันมากที่สุด โดยใช้สูตร Euclidean distance

ขั้นตอนที่ 4 : นำค่าความแตกต่างที่น้อยที่สุด มาเทียบกับค่า Threshold ถ้าค่าความแตกต่างน้อยกว่าหรือเท่ากับ ค่า Threshold ระบบจะบอกชื่อของบุคคลในภาพนั้น แต่ถ้ามากกว่าค่า Threshold ระบบจะบอกว่าไม่ทราบชื่อ

4.5 การใช้งานแอพพลิเคชัน

เริ่มต้นการใช้งานแอพพลิเคชัน สามารถเข้าใช้งานแอพพลิเคชันโดยการเลือกไอคอมตามที่ปรากฏบนหน้าจอ เมื่อเลือกไอคอนดังกล่าวจะปรากฏหน้าจอหลักของการทำงาน ที่ประกอบด้วยฟังก์ชันการทำงานทั้งหมด 3ส่วน แบ่งเป็นส่วนต่างๆ ดังนี้

- **New** ส่วนเพิ่มข้อมูลของบุคคลที่ต้องการค้นหา โดยภาพที่ได้จากคลังรูปภาพหรือภาพที่ได้จากการถ่าย
- **Camera/ Image** ส่วนในการค้นหาบุคคลที่ได้ทำการบันทึกข้อมูลลงในฐานข้อมูล โดยสามารถหาได้จากการส่องกล้อง(Camera)หรือค้นหาจากภาพถ่าย(Image)
- **View** บัญชีรายชื่อบุคคลที่ได้ทำการบันทึกข้อมูลลงในฐานข้อมูล

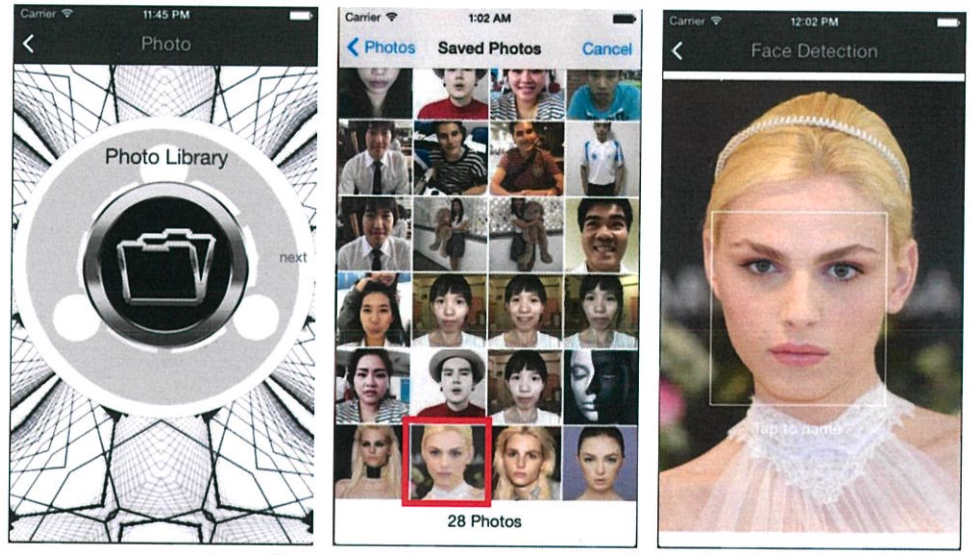


(ก) (ข)

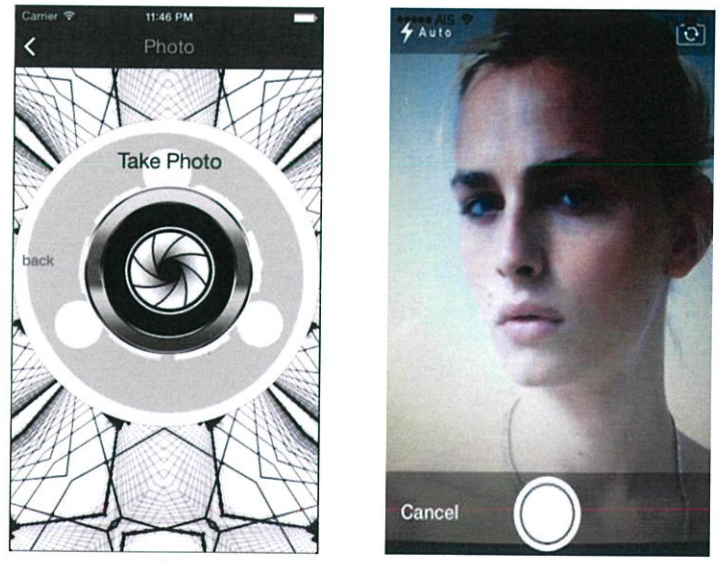
รูปที่ 4.3 (ก)ไอคอนแอพพลิเคชัน Face Recognition (ข)แสดงหน้าจอหลัก

4.5.1 การเพิ่มบุคคล

New ส่วนเพิ่มข้อมูลของบุคคลที่ต้องการ เมื่อผู้ใช้กดเลือก New จะปรากฏหน้าใหม่ ซึ่งแสดงสองเมนู คือ Photo Library การเลือกรูปภาพจากคลังภาพที่มีอยู่บนอุปกรณ์ iOS และ Take a photo การเลือกรูปภาพจากการถ่ายรูป

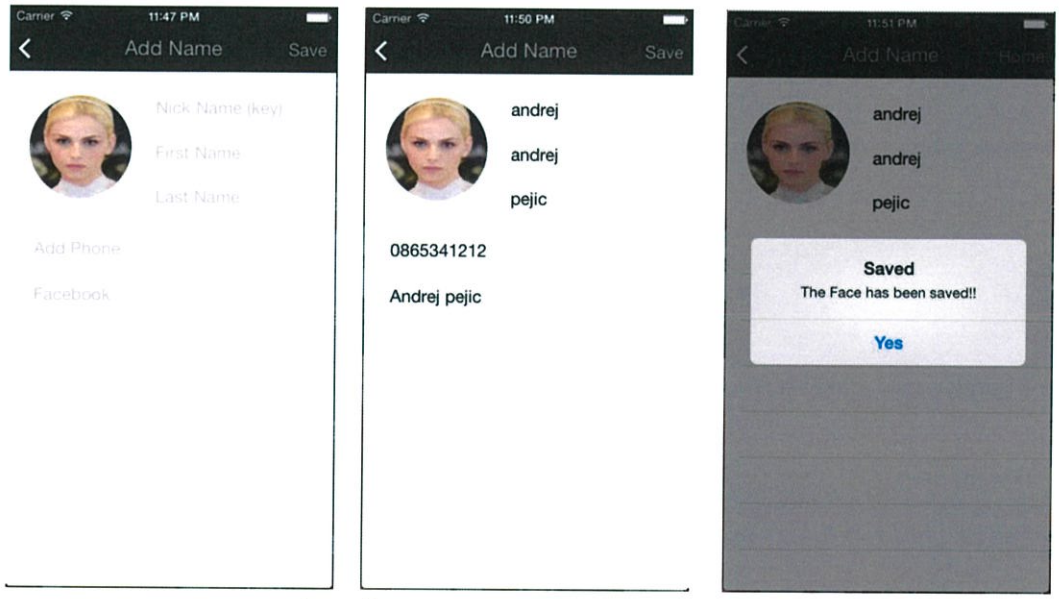


รูปที่ 4.4 ขั้นตอนการเพิ่มบุคคลจากการเลือก Photo Library



รูปที่ 4.5 ขั้นตอนการเพิ่มบุคคลจากการเลือก Take a photo

เมื่อเลือกรูปหรือถ่ายรูปเสร็จแล้วจะปรากฏรายละเอียดให้ผู้ใช้กรอกชื่อ นามสกุล ชื่อเล่น เบอร์โทรศัพท์ และรวมไปถึง Facebook จากนั้นกด save ระบบจะทำการประมวลผลภาพนี้เป็นภาพใบหน้าไอเกน แล้วเก็บลงฐานข้อมูล และจะแสดงชื่อให้ผู้ใช้ทราบว่ามีใบหน้าของคนนี้อยู่ในฐานข้อมูลแล้ว

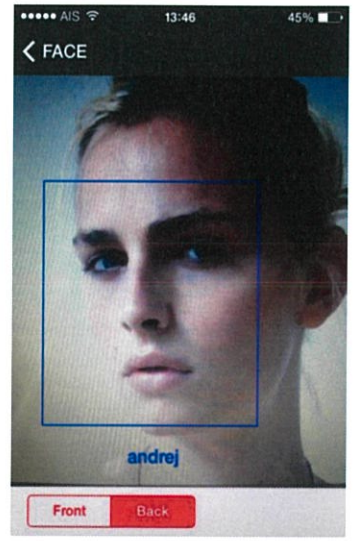


รูปที่ 4.6 ขั้นตอนการบันทึกข้อมูลลงในฐานข้อมูล

4.5.2 การค้นหาบุคคล

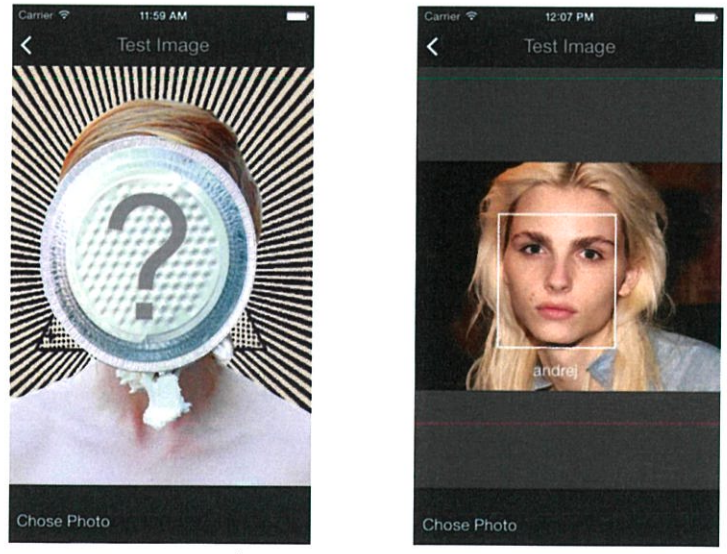
สามารถค้นหาบุคคลได้ 2 รูปแบบ คือ จาก Camera คือ การใช้กล้องส่องเพื่อค้นหาบุคคล หรือจาก Image คือ การค้นหาด้วยภาพถ่าย โดยบุคคลที่ต้องการค้นหาจะต้องทำการบันทึกข้อมูลลงในฐานข้อมูลก่อนเสมอ

- การค้นบุคคลด้วย Camera



รูปที่ 4.7 ขั้นตอนการค้นหาบุคคลด้วย Camera

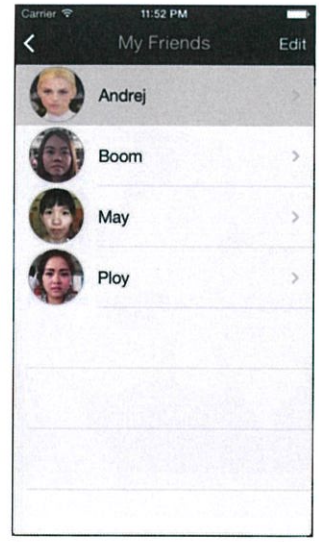
- การค้นหาบุคคลด้วย Image



รูปที่ 4.8 ขั้นตอนการค้นหาบุคคลด้วย Image

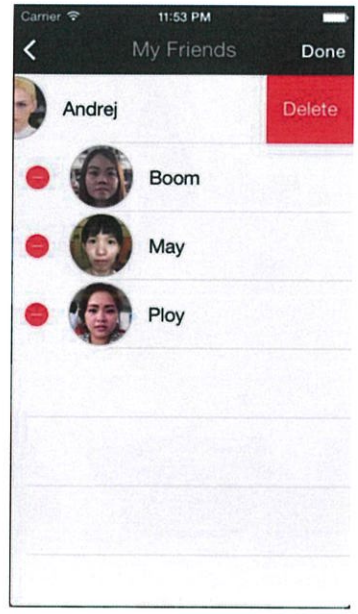
4.5.3 บัญชีรายชื่อบุคคล

View บัญชีรายชื่อบุคคลที่ได้ทำการบันทึกข้อมูลลงในฐานข้อมูล จะต้องบันทึกข้อมูลเป็นภาษาอังกฤษเท่านั้น และจะเรียงลำดับจากตัวอักษรภาษาอังกฤษ ภายในประกอบไปด้วย ข้อมูลชื่อ นามสกุล ชื่อเล่น เบอร์โทรศัพท์ และ Facebook ผู้ใช้งานสามารถดูข้อมูลที่เคยบันทึกไว้แต่เดิม และทำการแก้ไขข้อมูลหรือลบรายการที่ไม่ต้องการ

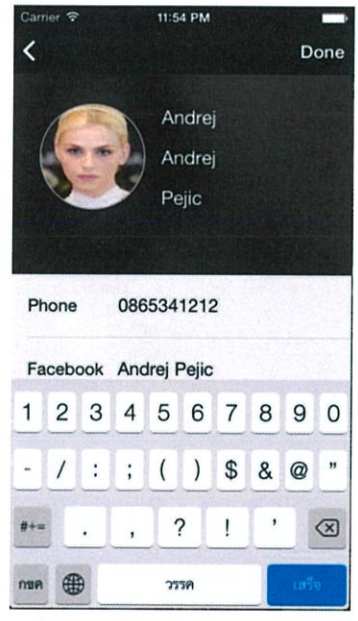


รูปที่ 4.9 แสดงบัญชีรายชื่อทั้งหมด

รูปที่ 4.10 แสดงข้อมูลของแต่ละบุคคล



รูปที่ 4.11 การลบรายการข้อมูล



รูปที่ 4.12 การแก้ไขรายการข้อมูล

4.6 การทดสอบ

แอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคล ทำการทดสอบกับกลุ่มบุคคลทั้งหมด 10 คน โดยเริ่มจากการถ่ายรูปกลุ่มบุคคลทั้งหมด แล้วทำการบันทึกข้อมูลลงในฐานข้อมูลให้ครบตามจำนวนทั้งหมด จากนั้นทำการสอบแอพพลิเคชัน เพื่อบันทึกผลการทดสอบประสิทธิภาพของแอพพลิเคชัน

กลุ่มรูปภาพใบหน้าบุคคลที่ทำการทดสอบทั้งหมด

รูปที่ 4.13 ภาพถ่ายใบหน้าบุคคลทั้งหมดที่ทำการทดสอบ

ตัวอย่างภาพใบหน้าที่ไม่ผ่านการทดสอบ

รูปที่ 4.14 ภาพใบหน้าที่มีแสงสว่างมากเกินไป

รูปที่ 4.15 ภาพใบหน้าที่มีแสงสว่างน้อยเกินไป

ข้อจำกัดในการถ่ายภาพ

1. ภาพใบหน้าที่ได้จากการถ่ายต้องเป็นภาพใบหน้าตรง
2. ภาพใบหน้าจะต้องมีแสงมีที่ความสม่ำเสมอ ไม่มืดหรือสว่างมากเกินไป เพื่อให้เห็นส่วนประกอบทั้งหมดบนใบหน้า

4.7 การประเมินผล

จำนวนบุคคลที่ทดสอบ	ผลลัพธ์การทดสอบ	จำนวนบุคคลที่ทดสอบ	ผลลัพธ์การทดสอบ
1	รู้จำถูกต้อง	6	รู้จำถูกต้อง
2	รู้จำผิดพลาด	7	รู้จำถูกต้อง
3	รู้จำถูกต้อง	8	รู้จำผิดพลาด
4	รู้จำถูกต้อง	9	รู้จำถูกต้อง
5	รู้จำถูกต้อง	10	รู้จำถูกต้อง

ตารางที่ 4.1 การประเมินผล

ผลการทดสอบ

จากการทดสอบแอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคล ในกลุ่มบุคคลทั้งหมด 10 คน แอพพลิเคชันสามารถรู้จำใบหน้าได้ถูกต้องทั้งหมด 8 คน จาก 10 คน ซึ่งคิดเป็นเปอร์เซ็นต์ความถูกได้ ร้อยละ 80 ของจำนวนทั้งหมดที่มี และให้การรู้จำที่ผิดพลาด 2 คน จาก 10 คน คิดเป็นเปอร์เซ็นต์ความผิดพลาดได้ร้อยละ 20 ของจำนวนทั้งหมด ถือว่าแอพพลิเคชันมีประสิทธิภาพที่ดีพอ โดยจะต้องอยู่ภายใต้ข้อจำกัดของการถ่ายภาพที่ได้กำหนดไว้ เพื่อทำให้แอพพลิเคชันสามารถรู้จำได้อย่างถูกต้อง และแม่นยำ

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

จากการศึกษาและพัฒนาเพื่อจัดทำแอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS สำหรับการระบุตัวบุคคล ด้วยเทคนิคการวิเคราะห์องค์ประกอบหลัก(Principal Component Analysis, PCA) ซึ่งภายในประกอบด้วยฟังก์ชันในการทำงานที่สำคัญ คือ การเพิ่มบุคคล การค้นหาบุคคลด้วยภาพหรือกล้อง และสมุดบัญชีรายชื่อบุคคล ที่ผู้ใช้งานสามารถเพิ่มและค้นหาบุคคลได้จากการถ่ายภาพจากกล้อง และสามารถตรวจสอบและแก้ไขข้อมูลได้จากสมุดรายชื่อบุคคล

จากการทดสอบจากกลุ่มบุคคลตัวอย่างทั้งหมด 10 คน ระบบสามารถรู้จำใบหน้าได้ถูกต้องประมาณร้อยละ 80 โดยอยู่ภายใต้เงื่อนไขในการถ่ายภาพที่กำหนดไว้ ควรมีแสงที่สม่ำเสมอทั้งพื้นหลังและตัวบุคคล สามารถมองเห็นองค์ประกอบบนใบหน้าได้อย่างครบถ้วน และกรณีที่มีความคล้ายคลึงกันของใบหน้าบุคคล รวมถึงท่าทางบนใบหน้า จะส่งผลให้การรู้จำของระบบมีความแม่นยำน้อยลง

5.2 ข้อเสนอแนะ

แนวทางสำหรับการพัฒนาแอพพลิเคชันรู้จำใบหน้าบนอุปกรณ์ iOS ที่ควรพัฒนาต่อไป คือ การพัฒนาทางด้านอัลกอรึทึม ที่ควรเพิ่มประสิทธิภาพให้สามารถใช้งานได้กับทุกสภาพแสง ทั้งในเวลากลางวันและกลางคืน ตลอดจนที่ๆอับแสงไปจนถึงแสงจ้า รวมไปถึงตำแหน่งในการถ่ายภาพที่สามารถตรวจจับได้ทุกตำแหน่งไม่ใช่แค่เพียงแต่หน้าตรงเท่านั้น และปรับปรุงในด้านเวลาในประมวลผลข้อมูล ที่ควรพัฒนาให้ระบบมีการประมวลผลที่ไวขึ้น เพื่อรองรับการทำงานที่มีข้อมูลขนาดใหญ่ ซึ่งทำให้เกิดประสิทธิภาพให้แอพพลิเคชันเป็นระบบที่ดีมากยิ่งขึ้น

เอกสารอ้างอิง

สมเกียรติ อุดมหรรษากุล. *การประมวลผลภาพดิจิตอลเบื้องต้น (Fundamentals of Digital Image Processing)*. กรุงเทพฯ : ท้อป; 2554.

Detecting Faces in an Image. iOS Developer Library Web site. 2004. Available at: https://developer.apple.com/library/ios/documentation/graphicsimaging/conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html. Accessed October 8, 2013.

Dowa J. *Tutorial: Easy Face Detection With Core Image In iOS 5 (Example Source Code Provided)*. ManiacDev Web site. 2011. Available at:http://maniacdev.com/2011/11/tutorial-easy-face-detection-with-core-image-in-ios-5. Accessed January 7, 2014.

Taking Pictures and Movies. iOS Developer Library Web site. 2004. Available at:https://developer.apple.com/library/ios/documentation/AudioVideo/Conceptual/CameraAndPhotoLib_TopicsForIOS/Articles/TakingPicturesAndMovies.html. Accessed January 15, 2014.

Turk M. and Pentland A. Eigenfaces for Recognition. *Journal of Cognitive Neurosicence*. 1991; 3:1:71-86

Wagner P. Eigenvalues in OpenCV. 2010. Available at: http://bytefish.de/blog/eigenvalues_in_opencv/. Accessed December 5, 2013.