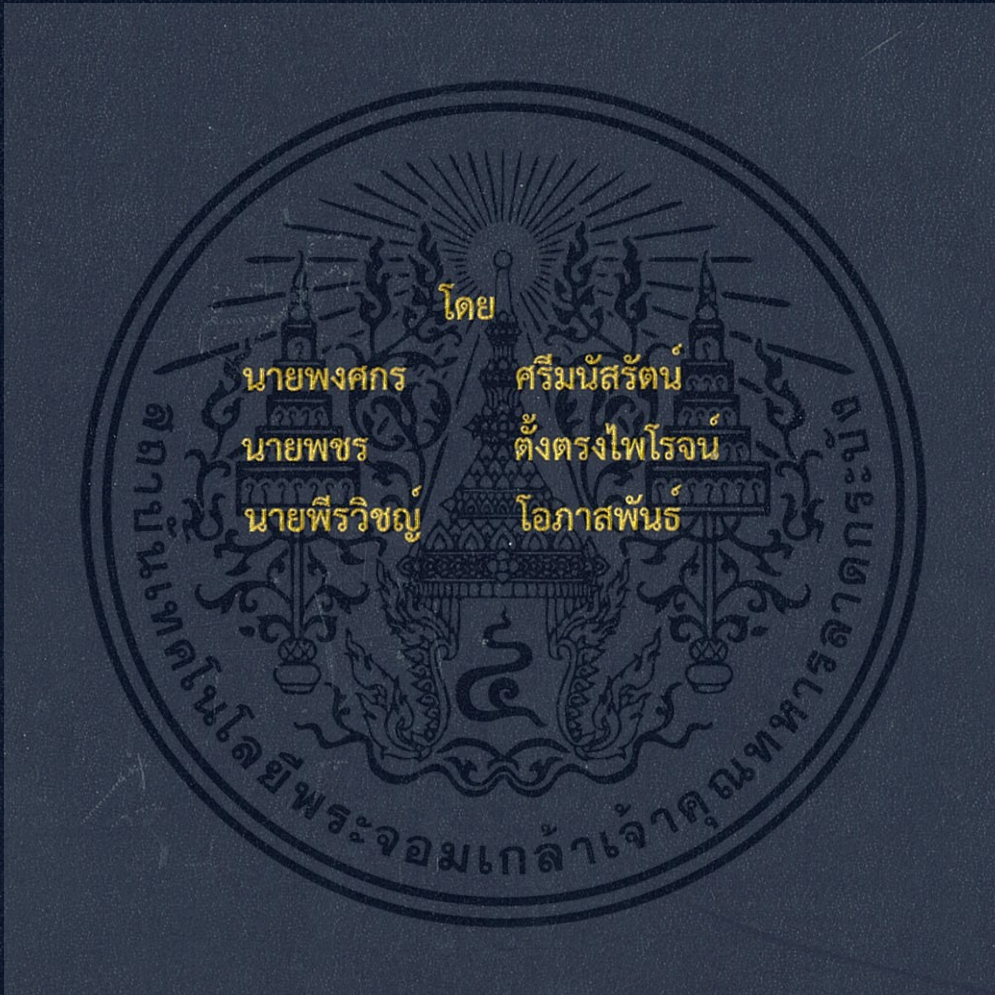


ระบบรักษาความปลอดภัยแบบตรวจจับใบหน้า  
SECURITY SYSTEM BY FACE DETECTOR



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2558

# บันทึกหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรักษาความปลอดภัยแบบตรวจจับใบหน้า  
SECURITY SYSTEM BY FACE DETECTOR



โดย

นายพงศกร	ศรีมนัสรัตน์	55010788
นายเพชร	ตั้งตรงไพโรจน์	55010810
นายพีรวิษณุ	โอภาสพันธ์	55010882



อาจารย์ที่ปรึกษา

ผศ.ดร. นภัทร สระเอี่ยม

ผศ.ดร. ธเนศ พัฒนธาดาพงษ์

เลขหมู่.....  
เลขทะเบียน 144412  
รับเดือนปี 24 พ.ย. 2559

b. 12818859  
f. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

ผ่านการตรวจรูปเล่มแล้ว

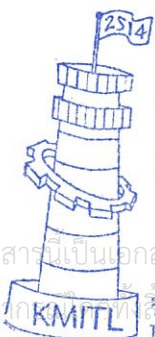
  
.....  
อาจารย์ที่ปรึกษา

ผ่านการตรวจชิ้นงานแล้ว

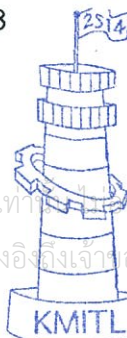
  
.....  
กรรมการผู้ตรวจชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้เพื่อการศึกษาได้โดยไม่เสียค่าใช้จ่าย

ไม่ว่ากรณีใดๆก็ตาม หากมีข้อผิดพลาดประการใด ขออภัยไว้ ณ ที่นี้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วิศวกรรมโทรคมนาคม  
Telecommunications Engineering



วิศวกรรมโทรคมนาคม  
Telecommunications Engineering

ปริญญาโทปีการศึกษา 2558

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เรื่อง ระบบรักษาความปลอดภัยแบบตรวจจับใบหน้า

Security System by Face Detector

ผู้จัดทำ

- |                 |                |          |
|-----------------|----------------|----------|
| 1. นาย พงศกร    | ศรีมันัสรัตน์  | 55010788 |
| 2. นาย พชร      | ตั้งตรงไพโรจน์ | 55010810 |
| 3. นาย พีรวิทย์ | โอภาสพันธ์     | 55010882 |

  
..... อาจารย์ที่ปรึกษา  
( ผศ.ดร.นภัทร สระเอี่ยม )

  
..... อาจารย์ที่ปรึกษา  
( ผศ.ดร.ธเนศ พัฒนธาดาพงษ์ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

โครงการนี้สามารถสำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับคำแนะนำและการให้คำปรึกษาตลอดจนปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างยิ่ง จาก ผู้ช่วยศาสตราจารย์ ดร. นภัทร สระเอี่ยม และ ผู้ช่วยศาสตราจารย์ ดร. ฌเนศ พัฒนธาดาพงษ์ คณะผู้จัดทำตระหนักถึงความตั้งใจและความทุ่มเทของอาจารย์จึงรู้สึกซาบซึ้งสำหรับคำแนะนำ และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับคณะผู้จัดทำ ตลอดจน บิดา มารดา พี่ ๆ น้อง ๆ เพื่อน ๆ ของคณะผู้จัดทำทุกคนที่เป็นกำลังใจและให้การสนับสนุนในทุก ๆ เรื่อง ทำให้กลุ่มของคณะผู้จัดทำสามารถจัดทำปฏิญานีพนธ์นี้สำเร็จลุล่วงด้วยดี

สุดท้ายนี้หากมีข้อบกพร่องประการใดที่อาจจะเกิดขึ้นนั้น คณะผู้จัดทำขอน้อมรับผิดเพียงผู้เดียว และยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ในการพัฒนางานต่อไป

นายพงศกร ศรีมันส์รัตน์  
นายเพชร ตั้งตรงไพโรจน์  
นายพีรวิชญ์ โอภาสพันธ์  
คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยแบบตรวจจับใบหน้า  
Security System by Face Detector

โดย นายพงศกร ศรีมนัสรัตน์ 55010788  
นายพชร ตั้งตรงไพโรจน์ 55010810  
นายพีรวิษณุ โอภาสพันธ์ 55010882

อาจารย์ที่ปรึกษา ผศ.ดร.นภัทร สระเอี่ยม

ผศ.ดร.ธเนศ พัฒนาธาดาพงษ์

### บทคัดย่อ

ในปัจจุบันนี้ความปลอดภัยนั้นเป็นสิ่งที่สำคัญมากสำหรับที่พักอาศัย ซึ่งถ้าที่พักอาศัยนั้นมีความปลอดภัยก็จะไม่ทำให้เกิดความเสียหายต่อชีวิตและทรัพย์สินของผู้อยู่อาศัย โครงการนี้เป็นการพัฒนาาระบบรักษาความปลอดภัยโดยใช้การตรวจจับใบหน้าเพื่อปลดล็อคแทนลูกกุญแจ โดยใช้ Raspberry Pi เป็นบอร์ดประมวลผล เพื่อเพิ่มประสิทธิภาพและความสะดวกของระบบรักษาความปลอดภัยให้มีมากยิ่งขึ้น ซึ่งจะเป็นการพัฒนาาระบบด้วยภาษา Python และใช้โมดูลกล้องทำหน้าที่ในการถ่ายวิดีโอ ผ่านทางโปรแกรม OpenCV แล้วส่งไปประมวลผลที่ Raspberry Pi เพื่อให้แสดงผลว่าใบหน้าที่ตรวจจับได้เป็นใบหน้าของใคร ถ้าหากตรงกับข้อมูลที่มีในฐานข้อมูลก็จะสามารถทำการปลดล็อคได้

### ABSTRACT

Currently, security is very important for the shelter. If shelter has safety, it will not cause damage to life and property of residents. This project is develop a security system with face detection to unlock instead of using the key. Raspberry Pi is used to core processor for improve convenience and efficiency of security system. The program is written by Python language and using Camera Module to record video via the OpenCV. Then sent to the processor at the Raspberry Pi. And in order to show the result of face detected. If the face match with the data in database, it will be unlock.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญ (ต่อ)	IV
สารบัญรูป	V
สารบัญรูป (ต่อ)	VI
สารบัญตาราง	VII
<b>บทที่ 1 บทนำ</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญานิพนธ์	2
<b>บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง</b>	<b>3</b>
2.1 ราสเบอร์รี่พาย 2 (Raspberry Pi 2)	5
2.2 Raspberry Pi Camera Module	11
2.3 ภาษาไพธอน	13
2.4 OpenCV	14
2.5 การประมวลผลภาพเชิงตัวเลข	18
2.6 ประเภทของภาพ	20
2.7 การตรวจจับใบหน้า (Face Detection)	23
2.8 การรู้จำใบหน้า (Face Recognition)	27
2.9 MySQL	30
2.10 ภาษา PHP	31
2.11 phpMyAdmin	34

## สารบัญ (ต่อ)

	หน้า
<b>บทที่ 3 การออกแบบและการจัดทำปฏิญญานิพนธ์</b>	36
3.1 การออกแบบ	36
3.2 เครื่องมือที่ใช้ในการทดลอง	43
3.3 การจัดเก็บผลการทดลอง	45
<b>บทที่ 4 ผลการทดลอง</b>	47
4.1 ผลการทดลองการตรวจจับใบหน้า	47
4.2 ผลการทดลองการรู้จำใบหน้าบุคคล	49
<b>บทที่ 5 สรุปผลและข้อเสนอแนะ</b>	57
5.1 สรุปผล	57
5.2 ปัญหาและอุปสรรค	57
5.3 ข้อเสนอแนะ	57
<b>บรรณานุกรม</b>	58
<b>ภาคผนวก ชุดคำสั่งการตรวจจับใบหน้า</b>	60

## สารบัญรูป

รูปที่	หน้า	
1.1	รูปแสดงบล็อกไดอะแกรมแสดงภาพรวมของระบบ	2
2.1	รูปแสดงบล็อกไดอะแกรมแสดงภาพรวมของระบบ	3
2.2	ไดอะแกรมการทำงานของระบบ	4
2.3	ส่วนประกอบของ Raspberry Pi 2	6
2.4	ขา GPIO ของ Raspberry Pi 2	7
2.5	รูปร่างวงจรที่เชื่อมต่อระหว่าง Raspberry Pi 2 GPIO กับ Solenoid Bolt Lock	7
2.6	ตัวอย่างของไฟล์ Raspbian	8
2.7	หน้าต่างโปรแกรม Win32DiskImage ลง Raspbian ให้กับ Micro SD card	9
2.8	หน้าต่างหลังจาก boot เสร็จ	10
2.9	ภาพหน้าจอ Raspberry Pi หลังจาก reboot เสร็จแล้ว	10
2.10	Raspberry Pi Camera Module	11
2.11	ลักษณะการติดตั้ง Raspberry Pi Camera Module	12
2.12	ระบบหลังการติดตั้งโมดูลกล้องเสร็จเรียบร้อยแล้ว	12
2.13	การทำ Sampling และ Quantization	19
2.14	แสดงวงล้อระบบสี RGB	21
2.15	แสดงระดับความสว่าง	21
2.16	แสดงระดับความสว่างของแต่ละพิกเซล	22
2.17	แสดงตัวอย่างลักษณะเด่นของ Haar ที่ถูกใช้ใน OpenCV	23
2.18	แสดงเทคนิคการรวมภาพ	24
2.19	ตัวอย่างรูปแบบของ Haar like feature	25
2.20	กระบวนการทำงานของ Classifier	25
2.21	แสดงตัวอย่างตัวกรองสองตัวแรกใน Viola-Jones Cascade	26
2.22	ตัวอย่างตัวกรองอื่น ๆ ใน Viola-Jones Cascade	26
2.23	แสดงขั้นตอนการทำงาน PHP Script Request/Response	32
2.24	แสดงหน้า phpMyAdmin	35

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.1 Block Diagram	36
3.2 แสดง Flow Chart ของระบบทั้งหมด	38
3.3 Flow Chart การทำงานของระบบ	39
3.4 Flow Chart การทำงานส่วนตรวจจับใบหน้า	40
3.5 Flow Chart การทำงานส่วนรู้จำใบหน้า	41
3.6 Flow Chart ของการเก็บภาพในฐานข้อมูล	42
3.7 ส่วนประกอบของ Raspberry Pi 2	43
3.8 ตำแหน่งของพอร์ต CSI (Camera Serial Interface)	44
3.9 Raspberry Pi Camera Module	45
3.10 แสดงรูปของระบบเมื่อติดตั้งอุปกรณ์	46
4.1 แสดงตัวอย่างการตรวจจับใบหน้าปกติ	47
4.2 แสดงตัวอย่างการตรวจจับใบหน้าขณะยิ้ม	47
4.3 แสดงตัวอย่างการตรวจจับใบหน้าขณะหลับตา	48
4.4 แสดงตัวอย่างการตรวจจับใบหน้าขณะเอียง 45 องศา	48
4.5 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลทั้งหมด 40 คน	50
4.6 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถปกติ	50
4.7 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถขณะยิ้ม	50
4.8 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถขณะหลับตา	51
4.9 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถขณะแลบลิ้น	51
4.10 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถขณะเอียงหน้า 45 องศา	51
4.11 แสดงตัวอย่างการรู้จำใบหน้าที่ใบหน้าตรงกับฐานข้อมูล	54
4.12 แสดงตัวอย่างการรู้จำใบหน้าที่ใบหน้าไม่ตรงกับฐานข้อมูล	55
4.13 แสดงตัวอย่างตารางบันทึกสถานะของผู้ใช้งานระบบ	56
4.14 แสดงตัวอย่างภาพบุคคลที่ไม่ตรงกับฐานข้อมูลและระบบทำการบันทึกไว้	56

## สารบัญตาราง

ตารางที่		หน้า
4.1	ผลการทดลองตรวจจับภาพใบหน้าในลักษณะต่าง ๆ	49
4.2	ผลการทดลองในส่วนรู้จำใบหน้าในขณะที่มีภาพใบหน้าในฐานข้อมูล 10 ใบหน้า	52
4.3	ผลการทดลองในส่วนรู้จำใบหน้าในขณะที่มีภาพใบหน้าในฐานข้อมูล 20 ใบหน้า	52
4.4	ผลการทดลองในส่วนรู้จำใบหน้าในขณะที่มีภาพใบหน้าในฐานข้อมูล 30 ใบหน้า	53
4.5	ผลการทดลองในส่วนรู้จำใบหน้าในขณะที่มีภาพใบหน้าในฐานข้อมูล 40 ใบหน้า	53



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันความปลอดภัยของทรัพย์สินหรือตัวบุคคลภายในที่พักอาศัยนั้นเป็นสิ่งสำคัญ เพราะไม่มีผู้ใดประสงค์ให้เกิดความเสียหายขึ้นกับตนเอง ซึ่งก็ได้มีการพัฒนาอุปกรณ์รักษาความปลอดภัยต่าง ๆ เพื่อช่วยเพิ่มประสิทธิภาพของการป้องกันมากยิ่งขึ้น เช่น กลุ่บิตประตูดิจิทัลมีการเปลี่ยนลักษณะ แม่กุญแจอาจใช้รหัสแทนใช้ลูกกุญแจ เป็นต้น แต่อุปกรณ์รักษาความปลอดภัยเหล่านี้ก็อาจไม่ปลอดภัยเท่าที่ควรเพราะผู้ไม่หวังดีมีความรู้ความสามารถที่จะหาช่องทางในการทำให้เกิดการสูญเสียมามากมาย ประกอบทั้งในปัจจุบันเทคโนโลยีระบบรักษาความปลอดภัยได้พัฒนาไปจากเดิมเป็นอย่างมาก จึงทำให้อุปกรณ์รักษาความปลอดภัยดังกล่าว อาจจะไม่ล้าสมัยไปบ้าง ซึ่งถ้าหากนำเทคโนโลยีที่ทันสมัยและเกี่ยวข้องกับระบบรักษาความปลอดภัยมาช่วยในการป้องกันความเสียหายที่จะเกิดขึ้นต่อทรัพย์สินและตัวบุคคล ก็จะทำให้ที่พักอาศัยมีความปลอดภัยมากยิ่งขึ้น

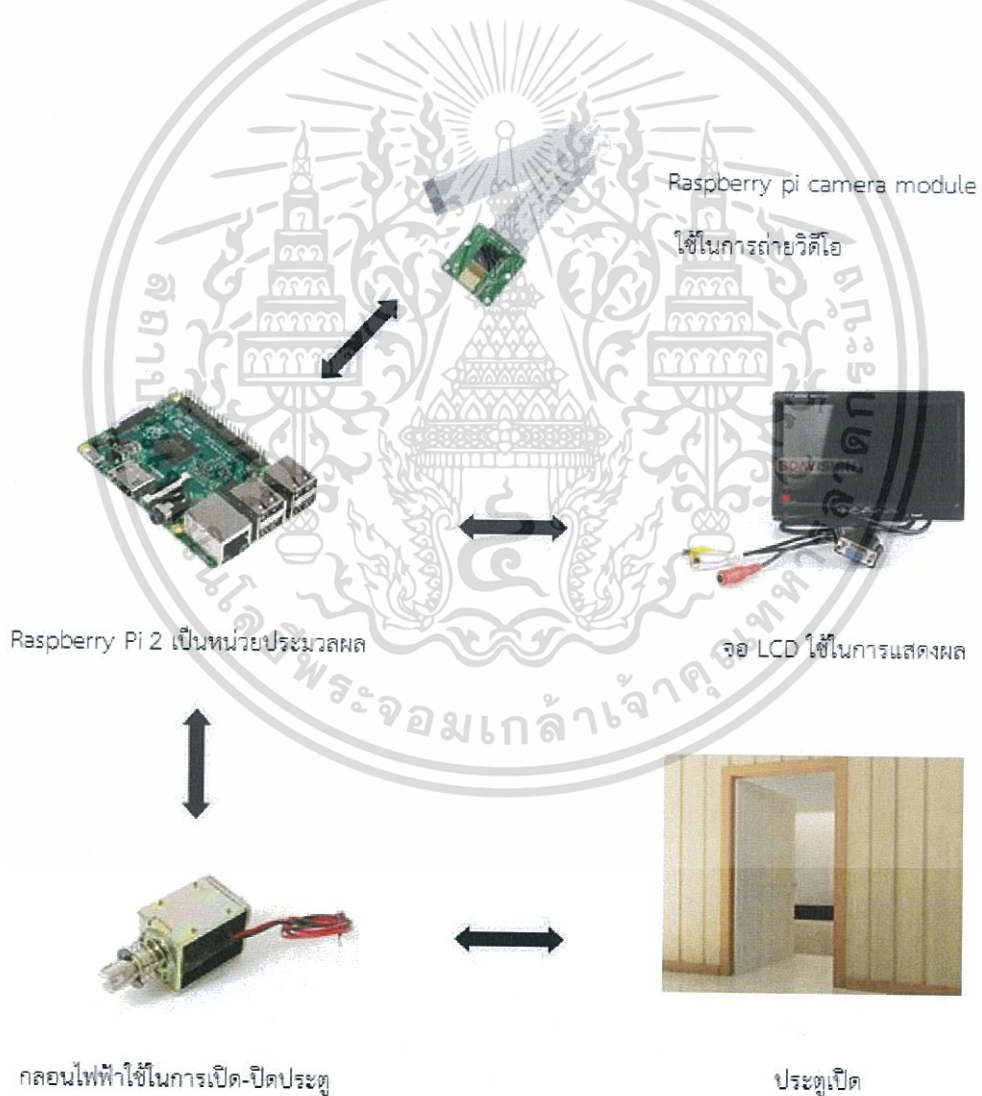
ในปฏิญานิพนธ์นี้ได้นำเทคโนโลยีการตรวจจับใบหน้ามาใช้ ซึ่งจะรับภาพจาก Raspberry Pi Camera Module ผ่านทางโปรแกรม OpenCV ซึ่งเป็นโปรแกรมประมวลผลภาพที่ได้ทำการติดตั้งลงบนบอร์ด Raspberry Pi ที่เป็นหน่วยประมวลผล

### 1.2 วัตถุประสงค์

1. เพื่อให้ระบบรักษาความปลอดภัยมีความสะดวกมากขึ้น
2. เพื่อศึกษาหลักการของการตรวจจับใบหน้าและการเปรียบเทียบใบหน้า
3. เพื่อศึกษาหลักการของโปรแกรม OpenCV ที่ใช้ในการประมวลผลภาพ
4. เพื่อศึกษาการพัฒนาาระบบโดยใช้ภาษา Python
5. เพื่อศึกษาการทำงานของ Raspberry Pi และระบบปฏิบัติการ Raspbian

### 1.3 ขอบเขตของปริญญาโท

ปริญญาโทนี้เป็นการพัฒนาระบบรักษาความปลอดภัยด้วยการตรวจจับใบหน้าและระบุตัวตนของบุคคลนั้น เพื่อใช้การเข้าที่พักอาศัย โดยใช้ Raspberry Pi ซึ่งเป็นการพัฒนาระบบโดยใช้ภาษา Python โดยใช้ Raspberry Pi Camera Module เป็นตัวถ่ายภาพวิดีโอ นำภาพที่ได้ไปประมวลผลที่ Raspberry Pi เพื่อตรวจจับใบหน้าและระบุตัวตนว่าใบหน้าที่ปรากฏเป็นใบหน้าของใคร ซึ่งถ้าใบหน้าที่ปรากฏนั้นตรงกับฐานข้อมูลที่มีอยู่ก็จะทำให้สามารถเข้าที่พักอาศัยได้ ในทางกลับกันถ้าใบหน้านั้นไม่ตรงกับฐานข้อมูลที่มีอยู่จะมีการบันทึกภาพเก็บไว้ เพื่อเป็นประโยชน์ต่อเจ้าของที่พักอาศัยในการระมัดระวังทรัพย์สินมากยิ่งขึ้น



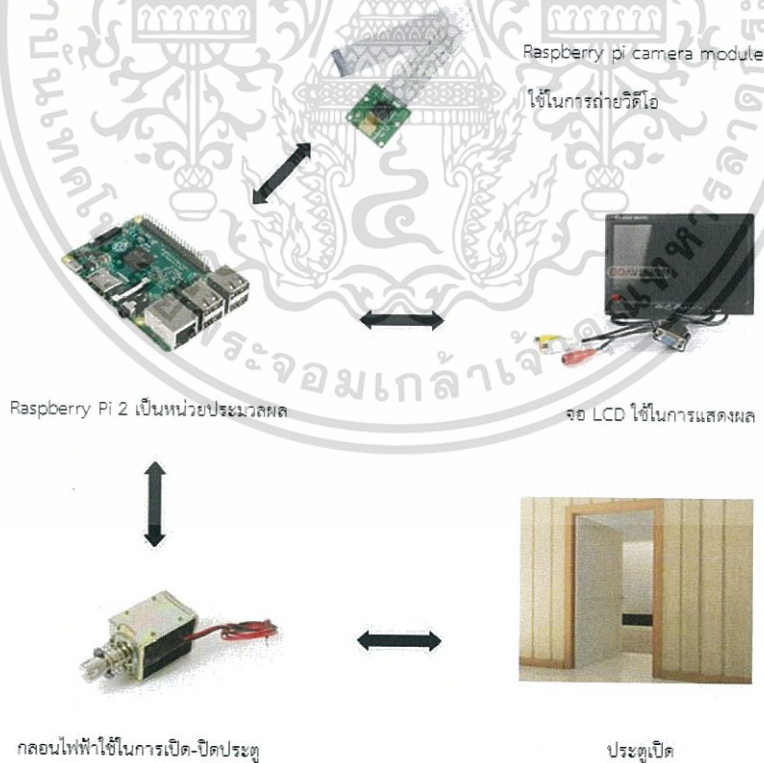
รูปที่ 1.1 บล็อกไดอะแกรมภาพรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

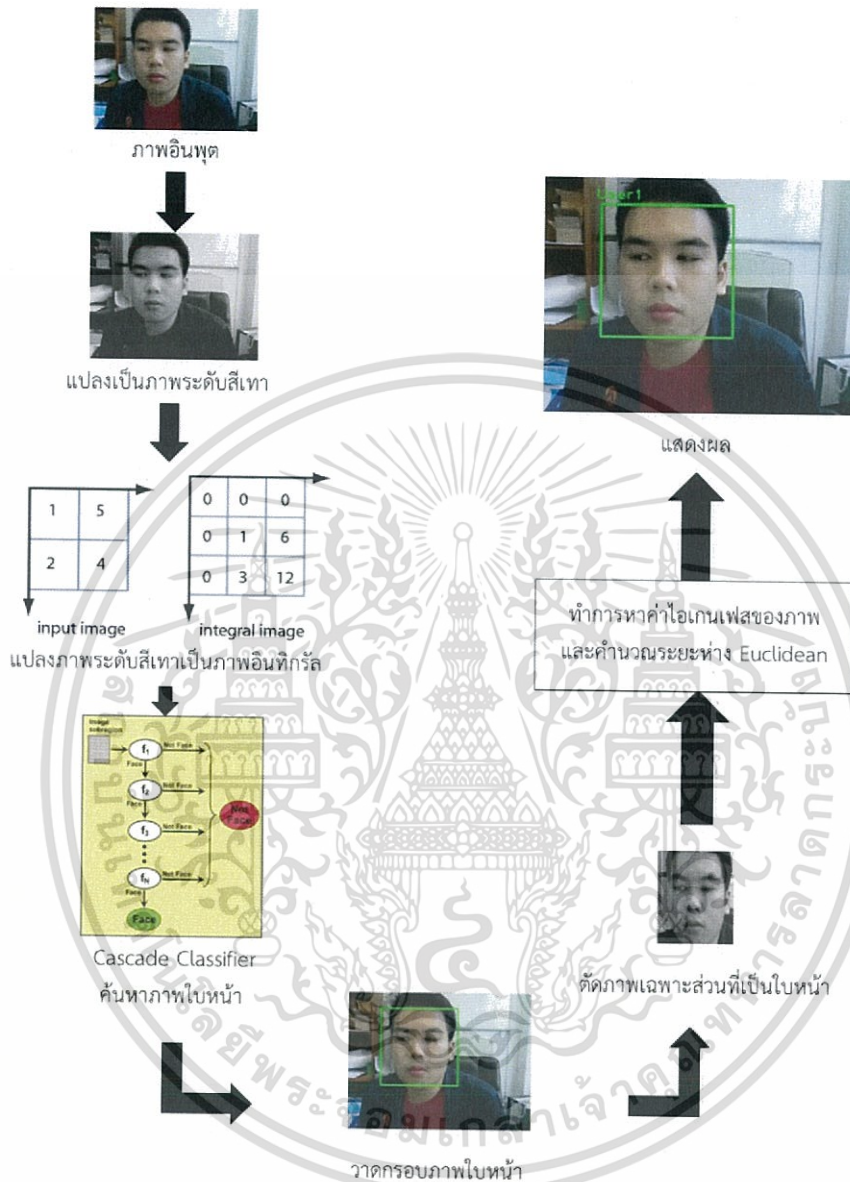
### ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์เรื่อง “ระบบรักษาความปลอดภัยแบบตรวจจับใบหน้า” เป็นการพัฒนาระบบรักษาความปลอดภัยด้วยการตรวจจับใบหน้าและระบุตัวตนของบุคคลนั้น เพื่อใช้ในการเข้าที่พักอาศัย โดยใช้ Raspberry Pi ที่มีระบบปฏิบัติการแบบ Raspbian และพัฒนาระบบโดยใช้ภาษา Python และนำ Raspberry Pi Camera Module มาเป็นอุปกรณ์ในการถ่ายวิดีโอ แล้วส่งภาพไปประมวลผลที่ Raspberry Pi เพื่อตรวจจับใบหน้าและระบุตัวตนว่าใบหน้าที่ปรากฏเป็นใบหน้าของผู้ใด โดยโปรแกรมที่ใช้ในการประมวลผลภาพและระบุตัวตนนั้นคือโปรแกรม OpenCV จากนั้นเมื่อทำการระบุตัวตนได้แล้ว ก็จะมีการส่งค่าไปยังพอร์ต GPIO เพื่อทำให้พอร์ต GPIO มีการจ่ายไฟไปยังกลอนไฟฟ้าเพื่อทำการปลดล็อกประตู แต่ถ้าไม่สามารถระบุตัวตนของใบหน้านั้นได้ ก็จะมีการบันทึกภาพใบหน้านั้นไว้ เพื่อนำไปใช้ประโยชน์เมื่อเกิดอันตรายต่อทรัพย์สินหรือตัวบุคคลต่อไป ซึ่งทฤษฎีและหลักการทั้งหมดที่เกี่ยวข้องมีดังต่อไปนี้



รูปที่ 2.1 บล็อกไดอะแกรมภาพรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ไดอะแกรมการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.1 ราชเบอร์รี่พาย 2 (Raspberry Pi 2)

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกม อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition Video)

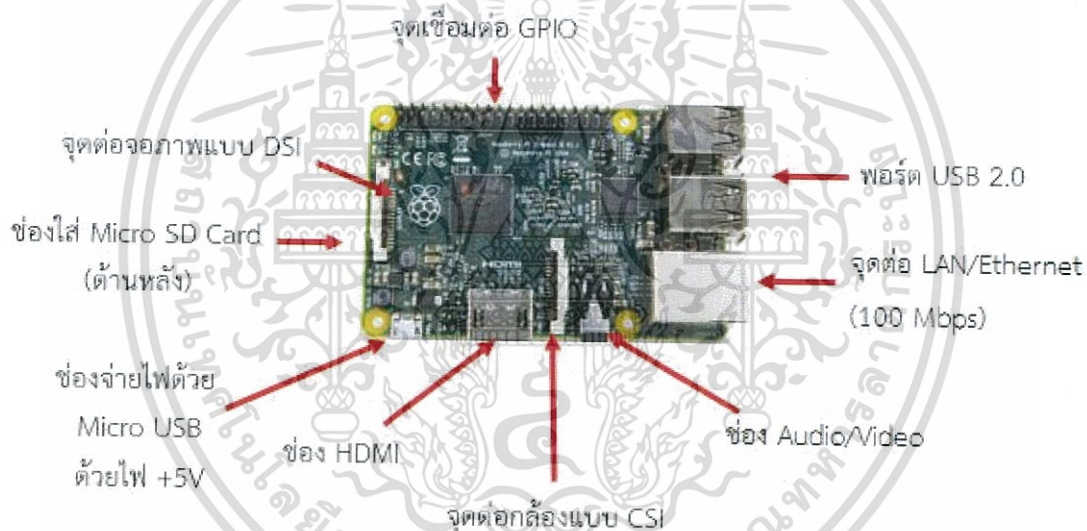
บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน Micro SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่น ๆ ได้อีกด้วย โดยปริยณานิพนธ์นี้ได้ใช้บอร์ด Raspberry Pi 2 Model B

### 2.1.1 คุณสมบัติทางเทคนิค

- ชิปควบคุมหลัก : Broadcom BCM2836 ซึ่งรวมหน่วยประมวลผลกลาง (CPU) หน่วยประมวลผลกราฟฟิก (GPU) และหน่วยความจำ SDRAM ไว้ภายในตัวเดียวกัน
- หน่วยประมวลผลกลางหรือ CPU : ARM Cortex-A7 4-core ความถี่สัญญาณนาฬิกา 900 MHz
- หน่วยประมวลผลกราฟฟิกหรือ GPU : Broadcom VideoCore IV รองรับการแสดงผลผ่านจอภาพที่ใช้จุดต่อแบบ HDMI
- หน่วยความจำ SDRAM : 1GiB
- จุดต่อ : USB 2.0 (4 พอร์ต), HDMI เอาต์พุตสัญญาณวิดีโอสำหรับต่อกับโทรทัศน์หรือจอแสดงผลที่มีจุดต่อแบบ HDMI, จุดต่อเอาต์พุตเสียงเป็นแจ็คหูฟัง 3.5 มิลลิเมตร, จุดต่ออีเธอร์เน็ตหรือจุดต่อระบบ LAN, จุดต่อพอร์ตอินพุตเอาต์พุต (General Purpose Input/Output : GPIO) ที่มีขาต่อบัส SPI (Serial Peripheral Interface Bus), I<sup>2</sup>C (Inter-Integrated Circuit), ขาสัญญาณรับส่งข้อมูลอนุกรมหรือ UART และซ็อกเก็ตของ Micro SD card สำหรับเสียบ Micro SD card ที่ติดตั้งระบบปฏิบัติการเรียบร้อยแล้ว
- ความต้องการไฟเลี้ยง : +5V 650mA (3.0 W)
- ขนาด : 85 x 56 x 17 มิลลิเมตร น้ำหนัก : 40 กรัม

## 2.1.2 ส่วนประกอบของบอร์ด Raspberry Pi

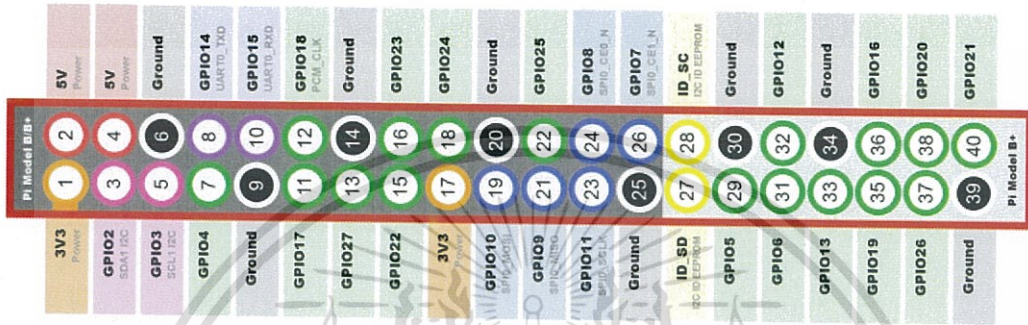
- HDMI Out (ใช้เชื่อมต่อพอร์ต HDMI เพื่อแสดงผลบนหน้าจอ)
- Ethernet Out (ใช้เชื่อมต่อหัวต่อแบบ RJ-45 เพื่อเชื่อมต่อระบบเครือข่าย LAN)
- USB 2.0 (ใช้เชื่อมต่ออุปกรณ์ USB)
- Audio Out (ใช้เชื่อมต่ออุปกรณ์เสียง)
- Camera Serial Interface (ใช้เชื่อมต่อ Raspberry Pi Camera Module)
- Micro SD Card Slot (ใช้เชื่อมต่อ Micro SD Card)
- Display Serial Interface (ใช้เชื่อมต่อจอภาพแบบ DSI หรือต่อจอภาพเพิ่มเติม)
- GPIO Headers (ใช้เชื่อมต่อ GPIO)



รูปที่ 2.3 ส่วนประกอบของ Raspberry Pi 2

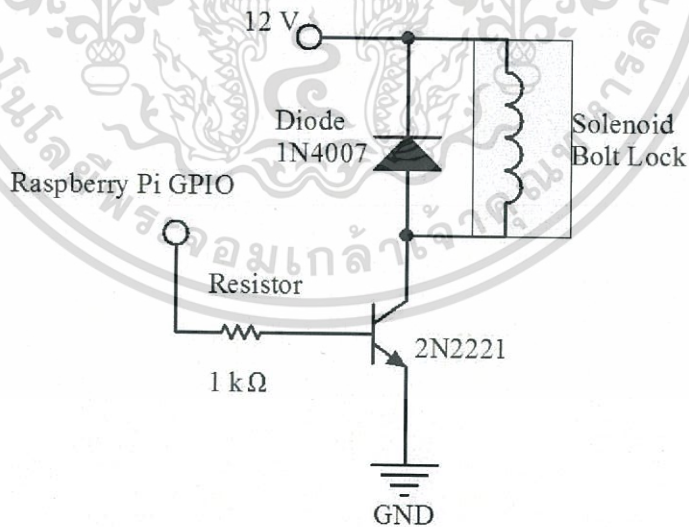
### 2.1.3 การจัดการขา GPIO

GPIO ย่อมาจาก General Purpose Input/Output ขา Pin เหล่านี้มีไว้ใช้งานทั่ว ๆ ไป โดยควบคุมผ่านซอฟต์แวร์ ขา GPIO นั้นจึงสามารถใช้งานได้หลากหลายรูปแบบขึ้นอยู่กับผู้ใช้งานต้องการที่จะใช้งานอะไร เช่น ในปริญญานิพนธ์นี้จะใช้ขา GPIO ในการสั่งการกลอนไฟฟ้า



รูปที่ 2.4 ขา GPIO ของ Raspberry Pi 2

ปริญญานิพนธ์นี้ได้นำขา GPIO มาต่อกับวงจรขับเพื่อใช้งานร่วมกับกลอนไฟฟ้า (Solenoid Bolt Lock) ดังรูปที่ 2.5



รูปที่ 2.5 รูปวงจรขับที่เชื่อมต่อระหว่าง Raspberry Pi 2 GPIO กับ Solenoid Bolt Lock

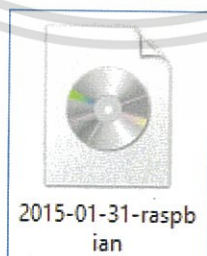
## 2.1.4 การเริ่มต้นการใช้งานบอร์ด Raspberry Pi

ก่อนเริ่มต้นการใช้งานบอร์ด Raspberry Pi จำเป็นที่จะต้องติดตั้งระบบปฏิบัติการให้กับบอร์ดก่อน เนื่องจากบอร์ดไม่มีหน่วยความจำแบบแฟลชเมมโมรี่มาบนบอร์ดด้วย ดังนั้นจำเป็นที่จะต้องเตรียมอุปกรณ์ต่าง ๆ ให้พร้อมเพื่อให้สามารถใช้งานบอร์ดได้ ซึ่งมีรายละเอียดอุปกรณ์ดังนี้

1. บอร์ด Raspberry Pi
2. Micro SD card ขนาด 8 GiB คลาส 4 หรือเร็วกว่า
3. เม้าส์และคีย์บอร์ดแบบ USB สำหรับควบคุมการทำงานของบอร์ด Raspberry Pi
4. สาย MicroUSB ใช้สำหรับต่อกับแหล่งจ่ายไฟ +5V 650mA
5. สาย HDMI สำหรับต่อกับจอแสดงผล

### 2.1.4.1 เตรียม Software สำหรับติดตั้งระบบปฏิบัติการ Linux ลงบนบอร์ด Raspberry Pi

1. เตรียม Micro SD card ขนาดที่พอเหมาะ โดยคณะผู้จัดทำได้ใช้ Micro SD card ขนาด 8 GiB
2. ดาวน์โหลดไฟล์ระบบปฏิบัติการ Raspbian ซึ่งเป็นระบบปฏิบัติการแบบ Debian ที่ถูกปรับแต่งให้ใช้สำหรับบอร์ด Raspberry Pi โดยเฉพาะ สามารถดาวน์โหลดได้จาก <http://www.raspberrypi.org/downloads> ปริญญาพันธ์นี้จะใช้รุ่น Wheezy
3. เมื่อดาวน์โหลดมาแล้วจะได้ไฟล์มาดังรูป



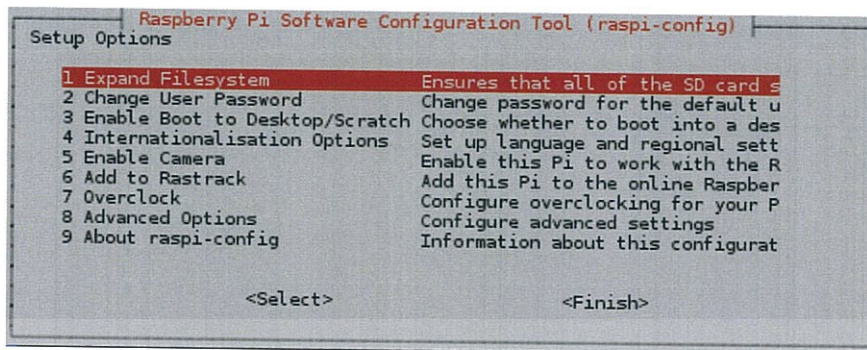
รูปที่ 2.6 ตัวอย่างของไฟล์ Raspbian

4. ดาวน์โหลดโปรแกรม Win32DiskImager จาก <http://sourceforge.net/projects/win32diskimager/> ซึ่งเป็นโปรแกรมที่ใช้ในการเขียนไฟล์ image ลง Micro SD card ที่เตรียมไว้ เมื่อดาวน์โหลดเสร็จให้ทำการเปิดโปรแกรม
5. ให้ Browse ไฟล์ image ระบบปฏิบัติการ Raspbian (\*.img) และเลือก Device ให้ถูกต้อง แล้วคลิกปุ่ม Write แสดงดังรูป เมื่อปรากฏหน้าต่างยืนยันให้คลิกปุ่ม Yes



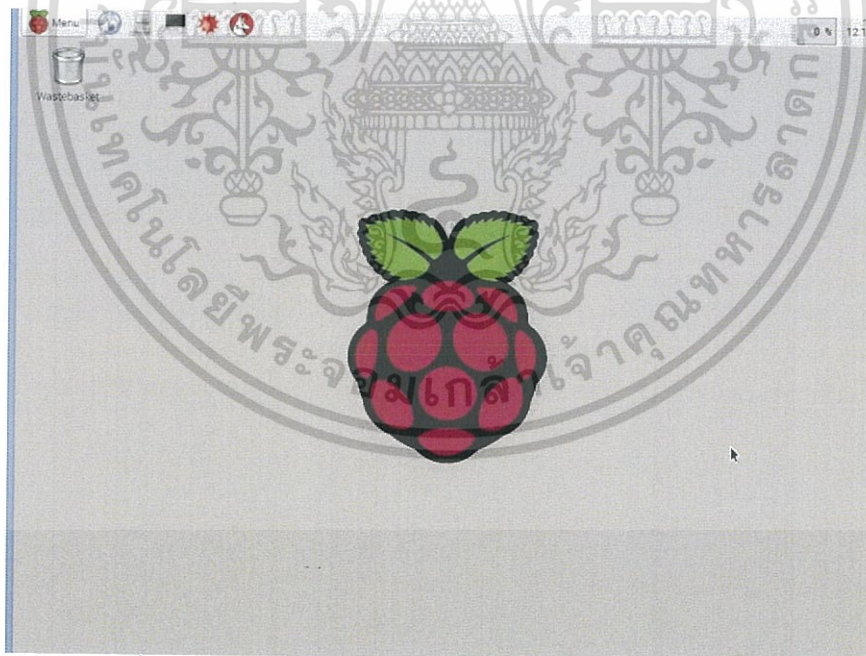
รูปที่ 2.7 หน้าต่างโปรแกรม Win32DiskImager ลง Raspbian ให้กับ Micro SD card

6. ทำการติดตั้ง Micro SD card ที่ลง Raspbian แล้วให้กับ Raspberry Pi
7. หลังจากเสียบ Micro SD Card เรียบร้อยแล้ว ให้เสียบเมาส์ คีย์บอร์ด สายต่อจอแสดงผล HDMI จากนั้นจึงป้อนไฟเลี้ยงวงจรบอร์ดผ่านพอร์ต MicroUSB
8. หลังจากนั้นบอร์ด Raspberry Pi ก็จะเริ่มทำงาน และเริ่ม Boot ระบบ
9. หลังจากระบบปฏิบัติการ Boot เสร็จเรียบร้อย ในครั้งแรกจะปรากฏหน้าต่างแสดงดังรูป ให้เลือกเมนู 1 Expand Filesystem เพื่อขยายพื้นที่บน Micro SD Card ให้ใช้งานได้เต็มความจุ เนื่องจากการลง Raspbian ครั้งแรกจะใช้พื้นที่บน Micro SD Card เพียง 4 GiB และจะไม่สามารถมองเห็นพื้นที่ส่วนที่เหลือของการ์ดได้



รูปที่ 2.8 หน้าต่างหลังจาก Boot เสร็จ

10. จากนั้นเลือก Set up ความสามารถอื่นตามที่ต้องการใช้ เช่น ในปริญญาบัตรนี้  
ต้องการใช้ควบคุมกับกล้องก็ทำการเลือกข้อ 5 ตาม รูปที่ 2.8 เพื่อเปิดใช้งานกล้อง  
เป็นต้น จากนั้นให้กด <Finish> แล้วทำการ Reboot
11. หลังจาก Reboot แล้วจะปรากฏหน้าจอของ Raspbian ดังภาพ



รูปที่ 2.9 ภาพหน้าจอ Raspberry Pi หลังจาก reboot เสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 Raspberry Pi Camera Module



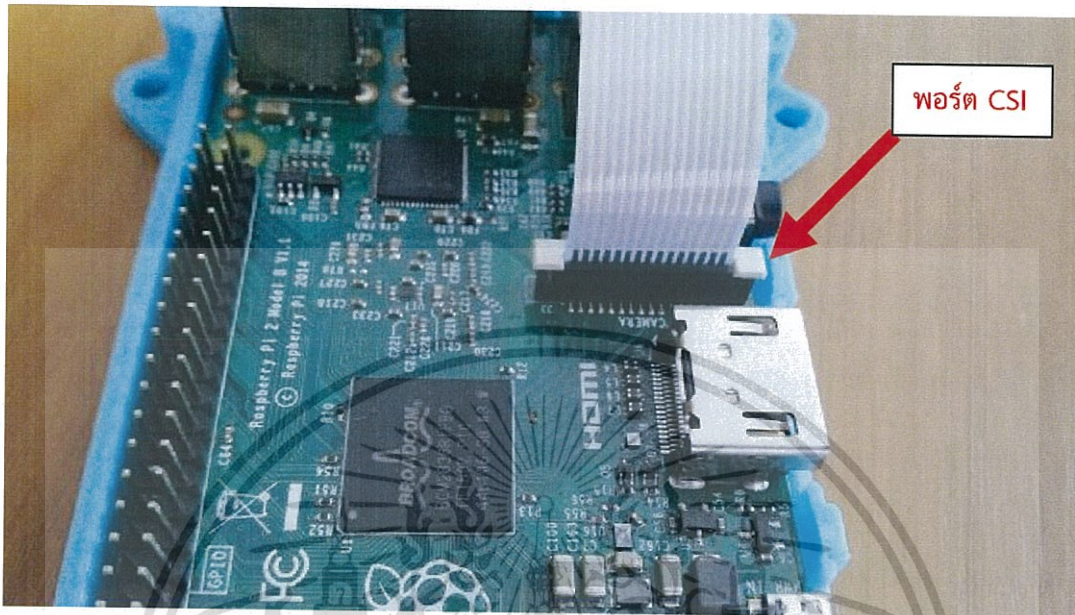
รูปที่ 2.10 Raspberry Pi Camera Module

Raspberry Pi Camera Module เป็นโมดูลกล้องที่ออกแบบมาเพื่อใช้งานร่วมกับบอร์ด Raspberry Pi โดยสามารถเชื่อมต่อกับพอร์ต CSI บนบอร์ดได้ทันที

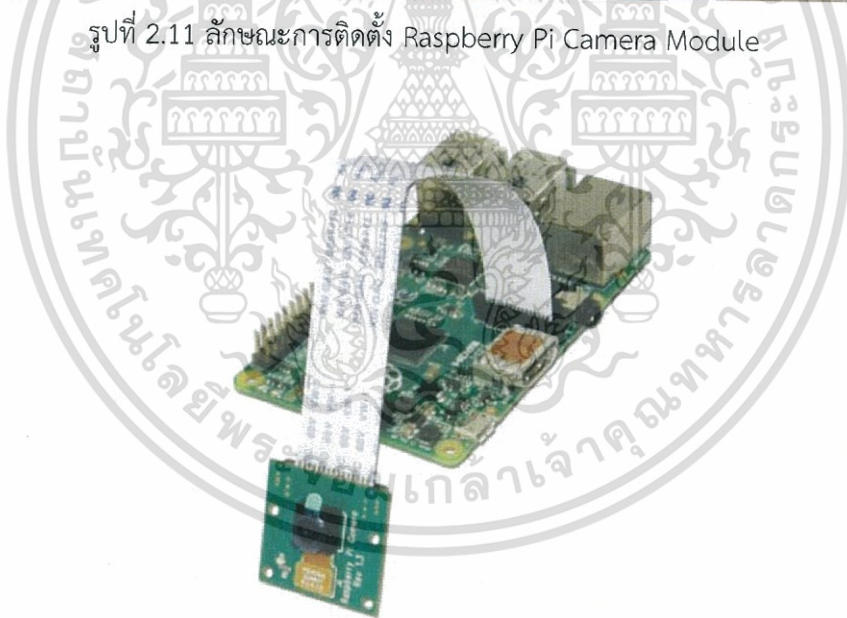
### 2.2.1 คุณสมบัติทางเทคนิค

- ขนาดบอร์ด 25 x 20 x 9 มิลลิเมตร
- ความละเอียดของกล้อง 5MP (2592 x 1944 พิกเซล)
- ใช้เซ็นเซอร์ Omnivision OV5647 โดยการโฟกัสเป็นแบบคงที่
- สามารถถ่ายวิดีโอความละเอียด 1080p30, 720p60 และ 640x480p60/90

## 2.2.2 การติดตั้งโมดูลกล้องเข้ากับ Raspberry Pi และการใช้งานเบื้องต้น



รูปที่ 2.11 ลักษณะการติดตั้ง Raspberry Pi Camera Module



รูปที่ 2.12 ระบบหลังการติดตั้งโมดูลกล้องเสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อติดตั้งเสร็จแล้วให้ทำการเชื่อมต่อ Raspberry Pi กับอินเทอร์เน็ต แล้วทำการอัปเดตไฟล์ของระบบปฏิบัติการเป็นเวอร์ชันล่าสุดก่อนโดยใช้คำสั่ง

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

2. ทำการ Enable Camera ในหน้า config

3. ทดสอบการใช้งานโดยการพิมพ์คำสั่ง `raspistill -o test.jpg` จะเป็นการถ่ายรูป แล้วเซฟรูปในชื่อของ test.jpg

4. ทดสอบการใช้งานโดยการพิมพ์คำสั่ง `raspivid -o test.h264` จะเป็นการถ่ายวิดีโอแล้วบันทึกเก็บไว้ 5 วินาทีตามค่า default

## 2.3 ภาษาไพธอน

ภาษาไพธอน พัฒนาโดย Guido van Rossum ชาวดัตช์ ซึ่งมีคุณสมบัติเด่นดังต่อไปนี้

- สามารถใช้ได้ทุกแพลตฟอร์ม กล่าวคือ สามารถทำงานได้ทุก ๆ CPU หลาย ๆ ระบบปฏิบัติการ เพียงแค่ผู้เขียนโปรแกรมเขียนบนแพลตฟอร์มใด ๆ แล้วนำโปรแกรมที่ได้ไปให้ทำงานต่างแพลตฟอร์มกันได้

- ไม่ต้องเสียค่าใช้จ่ายในการจัดซื้อโปรแกรมแปลภาษา (Interpreter) โดยปกติแล้วการเขียนโปรแกรมด้วยภาษาโปรแกรมมิ่งทั่ว ๆ ไปจะต้องจัดซื้อโปรแกรมแปลภาษาเพื่อนำมาติดตั้งในราคาแพง แต่โปรแกรมแปลภาษาไพธอนสามารถดาวน์โหลดจาก <http://www.python.org/> ได้โดยตรง แล้วนำมาติดตั้งและศึกษาการใช้ด้วยตนเอง เพราะเป็นโปรแกรมประเภท Open Source

- ภาษาไพธอนได้นำเอาข้อดีของภาษาโปรแกรมมิ่งอื่น ๆ เข้ามาไว้ด้วยกัน เช่น ภาษา C, C++, Java และ Perl เป็นต้น

- มีความปลอดภัยสูง เนื่องจากภาษาไพธอนทำงานอยู่ด้าน Server เป็นหลัก เมื่อมีการร้องขอจากเครื่อง Client จะประมวลผลที่เครื่อง Server ทำให้ผู้ใช้ทั่วไปไม่สามารถเข้าถึงเครื่อง Server ได้โดยตรงจึงมีความปลอดภัยสูงกว่า

- ใช้ในการพัฒนา Web Service ได้ ปัจจุบันการพัฒนาซอฟต์แวร์ได้เน้นในการแลกเปลี่ยนข้อมูลซึ่งกันและกัน ทั้งในองค์กรเดียวกันหรือแม้แต่ต่างองค์กรกัน เพื่อสะดวกในการเรียกใช้ข้อมูลทำงานร่วมกันระหว่างแอปพลิเคชัน ไม่ต้องใช้ซอฟต์แวร์อื่น ๆ มาแปลงข้อมูลเพื่อให้เข้ากันได้อีกต่อไป

## 2.4 OpenCV

OpenCV เป็นไลบรารีสำหรับใช้งานเรื่องการประมวลผลภาพ (Image Processing) ซึ่งมีประสิทธิภาพสูง ใช้งานง่ายและดาวน์โหลดได้ฟรี มีความสามารถมากมาย เช่น สามารถทำภาพเบลอหา threshold หา Histogram ของภาพได้ แต่ความสามารถโดยส่วนใหญ่แล้วจะทำการค้นหาขอบของภาพ การตรวจสอบการเคลื่อนไหวและการทำ Image segmentation และสามารถจัดการกับข้อมูลแบบวิดีโอได้ด้วย

OpenCV เป็นชุดคำสั่งที่ไม่ได้เป็นตัวโปรแกรม การเรียกใช้งานจึงต้องเขียนโปรแกรมเพื่อเรียกชุดคำสั่งจากไลบรารีเหล่านั้น ภาษาที่นิยมเขียน เช่น ภาษา C, C++ และ Python ไลบรารี OpenCV จะประกอบด้วยสองส่วนคือ data structure ที่ใช้ในการระบุโครงสร้างข้อมูลต่าง ๆ เช่น รูปภาพ เมตริกซ์ และพิกัด สำหรับอีกส่วนคือ algorithm หรือกระบวนการที่จะใช้ในการประมวลผลภาพในรูปแบบต่าง ๆ OpenCV จะประกอบด้วยไลบรารีอยู่ 4 ชุด ได้แก่ CXCORE, CV, Machine Learning และ HighGUI เช่น ตัวอย่างคำสั่งในการประกาศรูปภาพคือ IplImage, CvMat ใช้ในการประมวลผลและการวิเคราะห์รูปภาพ ฟังก์ชันส่วนใหญ่จะทำงานกับภาพที่เป็นอาเรย์ 2 มิติ เช่น การหาขอบหรือมุมของภาพ การทำฮิสโตแกรม (Histogram) เป็นต้น

ตัวอย่างคำสั่ง OpenCV ที่เกี่ยวข้องกับปริญญานิพนธ์นี้

- cv2.CascadeClassifier  
เป็นคำสั่งสำหรับเรียกใช้การจำแนกแบบ Cascade Classification
- cv2.resize  
เป็นคำสั่งที่ใช้ในการเปลี่ยนขนาดของรูปภาพ
- cv2.rectangle  
เป็นคำสั่งที่ใช้ในการวาดกรอบสี่เหลี่ยมรอบรูปภาพในส่วนที่ตรวจจับ
- cv2.imshow  
เป็นคำสั่งในการเปิดหน้าต่างเพื่อแสดงผลข้อมูล
- cv2.putText  
เป็นคำสั่งในการแสดงข้อความบนหน้าต่างแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4.1 ขั้นตอนการลง OpenCV บน Raspberry Pi

1. อัปเดต repository เพื่อให้ไฟล์ของ Raspbian เป็นเวอร์ชันล่าสุด

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

2. ติดตั้งเครื่องมือ

```
$ sudo apt-get -y install build-essential pkg-config cmake
cmake-curses-gui libpng12-0 libpng12-dev libpng++-dev libpng3
libpnglite-dev zlib1g-dbg zlib1g zlib1g-dev pngtools libtiff4
libtiff4-dev libtiffxx0c2 libtiff-tools libeigen3-dev
```

```
$ sudo apt-get -y install libjpeg8 libjpeg8-dev libjpeg8-dbg
libjpeg-progs ffmpeg libavcodec-dev libavcodec53 libavformat53
libavformat-dev libgstreamer0.10-0-dbg libgstreamer0.10-0
libgstreamer0.10-dev libxine1-ffmpeg libxine-dev libxine1-bin
libunicap2 libunicap2-dev swig libv4l-0 libv4l-dev python-numpy
libpython2.7 python-dev python2.7-dev libgtk2.0-dev
```

3. ติดตั้ง PIP (Python Package Index) เพื่อใช้ในการติดตั้ง NumPy ซึ่งเป็นไลบรารีในการประมวลผลตัวเลขทางวิทยาศาสตร์

```
$ wget https://bootstrap.pypa.io/get-pip.py
$ sudo python get-pip.py
$ pip install numpy
```

4. ดาวน์โหลดและแตกไฟล์ OpenCV

```
$ wget -o opencv-2.4.10.zip
http://sourceforge.net/projects/
opencvlibrary/files/opencv-unix/2.4.10/
opencv-2.4.10.zip/download
$ unzip opencv-2.4.10.zip
$ cd opencv-2.4.10
$ mkdir release
$ cd release
$ cmake ../
```

จากนั้นกด “c” เพื่อปรับค่า ดังนี้

ANT_EXECUTABLE	ANT_EXECUTABLE-NOTFOUND
BUILD_DOCS	ON
BUILD_EXAMPLES	ON
BUILD_JASPER	ON
BUILD_JPEG	ON

BUILD_OPENEXR	ON
BUILD_PACKAGE	ON
BUILD_PERF_TESTS	ON
BUILD_PNG	ON
BUILD_SHARED_LIBS	ON
BUILD_TBB	OFF
BUILD_TESTS	ON
BUILD_TIFF	ON
BUILD_WITH_DEBUG_INFO	ON
BUILD_ZLIB	ON
BUILD_opencv_apps	ON
BUILD_opencv_calib3d	ON
BUILD_opencv_contrib	ON
BUILD_opencv_core	ON
BUILD_opencv_features2d	ON
BUILD_opencv_flann	ON
BUILD_opencv_gpu	ON
BUILD_opencv_highgui	ON
BUILD_opencv_imgproc	ON
BUILD_opencv_legacy	ON
BUILD_opencv_ml	ON
BUILD_opencv_nonfree	ON
BUILD_opencv_objdetect	ON
BUILD_opencv_ocl	ON
BUILD_opencv_photo	ON
BUILD_opencv_python	ON
BUILD_opencv_stitching	ON
BUILD_opencv_superres	ON
BUILD_opencv_ts	ON
BUILD_opencv_video	ON
BUILD_opencv_videostab	ON
BUILD_opencv_world	OFF
CLAMDBLAS_INCLUDE_DIR	CLAMDBLAS_INCLUDE_DIR-NOTFOUND
CLAMDBLAS_ROOT_DIR	CLAMDBLAS_ROOT_DIR-NOTFOUND
CLAMDFFT_INCLUDE_DIR	CLAMDFFT_INCLUDE_DIR-NOTFOUND
CLAMDFFT_ROOT_DIR	CLAMDFFT_ROOT_DIR-NOTFOUND
CMAKE_BUILD_TYPE	Release
CMAKE_CONFIGURATION_TYPES	Debug;Release
CMAKE_INSTALL_PREFIX	/usr/local
CMAKE_VERBOSE	OFF
CUDA_BUILD_CUBIN	OFF
CUDA_BUILD_EMULATION	OFF
CUDA_HOST_COMPILER	/usr/bin/gcc
CUDA_SDK_ROOT_DIR	CUDA_SDK_ROOT_DIR-NOTFOUND
CUDA_SEPARABLE_COMPILATION	OFF
CUDA_TOOLKIT_ROOT_DIR	CUDA_TOOLKIT_ROOT_DIR-NOTFOUND
CUDA_VERBOSE_BUILD	OFF
EIGEN_INCLUDE_PATH	/usr/include/eigen3
ENABLE_NEON	OFF
ENABLE_NOISY_WARNINGS	OFF
ENABLE_OMIT_FRAME_POINTER	ON
ENABLE_PRECOMPILED_HEADERS	ON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ENABLE_PROFILING OFF
ENABLE_SOLUTION_FOLDERS OFF
ENABLE_VFPV3 OFF
EXECUTABLE_OUTPUT_PATH /home/pi/opencv-2.4.10/release/bin
GIGEAPI_INCLUDE_PATH GIGEAPI_INCLUDE_PATH-NOTFOUND
GIGEAPI_LIBRARIES GIGEAPI_LIBRARIES-NOTFOUND
INSTALL_CREATE_DISTRIB OFF
INSTALL_C_EXAMPLES OFF
INSTALL_PYTHON_EXAMPLES OFF
INSTALL_TO_MANGLED_PATHS OFF
OPENCV_CONFIG_FILE_INCLUDE_DIR
/home/pi/opencv/opencv-2.4.10/release

OPENCV_EXTRA_MODULES_PATH
OPENCV_WARNINGS_ARE_ERRORS OFF
OPENEXR_INCLUDE_PATH OPENEXR_INCLUDE_PATH-NOTFOUND
PVAPI_INCLUDE_PATH PVAPI_INCLUDE_PATH-NOTFOUND
PYTHON_NUMPY_INCLUDE_DIR
/usr/lib/pymodules/python2.7/numpy/core/include

PYTHON_PACKAGES_PATH lib/python2.7/dist-packages
SPHINX_BUILD SPHINX_BUILD-NOTFOUND
WITH_1394 OFF
WITH_CUBLAS OFF
WITH_CUDA OFF
WITH_CUFFT OFF
WITH_EIGEN ON
WITH_FFMPEG ON
WITH_GIGEAPI OFF
WITH_GSTREAMER ON
WITH_GTK ON
WITH_JASPER ON
WITH_JPEG ON
WITH_LIBV4L ON
WITH_NVCUVID OFF
WITH_OPENCV ON
WITH_OPENCV_AMDGPU ON
WITH_OPENCV_FFMPEG ON
WITH_OPENCV_GSTREAMER ON
WITH_OPENCV_IPP ON
WITH_OPENCV_OPENCL ON
WITH_OPENCV_OPENCLAMDBLAS ON
WITH_OPENCV_OPENCLAMDFFT ON
WITH_OPENCV_OPENEXR ON
WITH_OPENCV_OPENGL ON
WITH_OPENCV_OPENMP OFF
WITH_OPENCV_OPENNI OFF
WITH_PNG ON
WITH_PVAPI ON
WITH_QT OFF
WITH_TBB OFF
WITH_TIFF ON
WITH_UNICAP OFF
WITH_V4L ON
WITH_XIMEA OFF
WITH_XINE OFF

```

กต “c” อีกครั้งเพื่อใช้การตั้งค่า

กต “g” เพื่อสร้าง Makefile

5. ทำการติดตั้ง OpenCV โดยใช้ Makefile

```
$ make -j4
$ sudo make install
$ sudo ldconfig
```

6. ทำการตรวจสอบ OpenCV ว่าลงสำเร็จหรือไม่

```
$ workon cv
$ python
>>> import cv2
>>> cv2.__version__
'2.4.10'
```

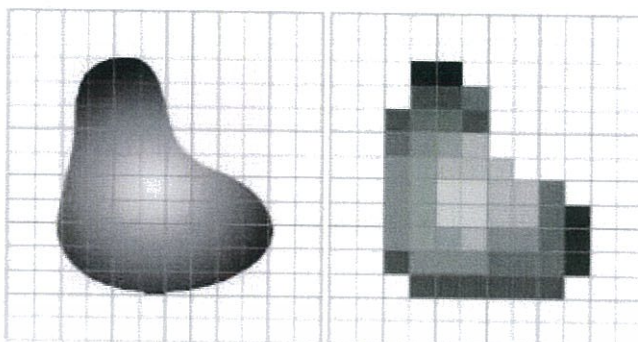
## 2.5 การประมวลผลภาพเชิงตัวเลข (Digital Image Processing)

เป็นการนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่ต้องการทั้งในเชิงคุณภาพและปริมาณ โดยมีขั้นตอนต่าง ๆ ที่สำคัญคือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่สนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนที่ของวัตถุในภาพ จากนั้นสามารถนำข้อมูลเชิงปริมาณเหล่านี้ไปวิเคราะห์ และสร้างเป็นระบบเพื่อใช้ประโยชน์ในงานด้านต่าง ๆ

### 2.5.1 การแทนภาพด้วยภาพแบบดิจิทัล

ภาพแบบดิจิทัล (Digital Image) เป็นภาพที่ถูกแปลงมาจากอนาล็อก ให้อยู่ในรูปของตัวเลขโดยภาพอนาล็อกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า พิกเซล ในแต่ละพิกเซลจะถูกระบุตำแหน่งด้วยคู่อันดับ  $x, y$  และค่าระดับความเข้มของแสงของพิกเซลนั้น ๆ โดยสามารถแปลงภาพเป็นแบบดิจิทัลโดยมีขั้นตอนและวิธีการดังนี้

เมื่อนำสัญญาณอนาล็อกที่ต้องการประมวลผลผ่านส่วนที่เรียกว่า ดิจิไทเซอร์ (Digitizer) ซึ่งจะทำหน้าที่ในการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล จากนั้นทำการควอนไทซ์ (Quantizing) เพื่อแปลงค่าความเข้มของแสงให้เป็นตัวเลขโดยฟังก์ชันของภาพ  $f(x, y)$  จะถูกทำให้เป็นสัญญาณไม่ต่อเนื่องทั้งระนาบของภาพซึ่งเรียกว่า การสุ่มภาพ (Image Sampling) ของฟังก์ชันที่ได้เรียกว่า การควอนไทซ์ระดับความเข้มของแสง ก็จะได้ข้อมูลที่เป็นดิจิทัลดังภาพที่ 2.13



รูปที่ 2.13 การทำ Sampling และ Quantization

### 2.5.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิทัล

โดยทั่วไปแล้วภาพจะมีความเข้มของพิกเซลตั้งแต่ 2 ระดับขึ้นไปซึ่งจะมีเพียงพิกเซลสีขาวและสีดำเท่านั้น แต่ที่นิยมใช้กันมากคือ ค่าระดับความเข้มของพิกเซลที่เท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของพิกเซลอยู่ในช่วง 0-255 โดยใช้เนื้อที่ในการเก็บข้อมูลขนาด 1 ไบต์ หรือ 8 บิต สำหรับข้อมูล 1 พิกเซล ในกรณีที่ต้องการภาพที่มีความเข้มสูงอาจจะต้องการจำนวนบิตเก็บข้อมูลมากกว่า 8 บิต คืออาจจะเป็น 16 หรือ 256 บิตโดยจะแยกความแตกต่างของภาพแต่ละประเภทให้เห็นอย่างชัดเจนได้ดังนี้

- ภาพ 2 ระดับ คือ มีพิกเซลสีขาวกับสีดำเท่านั้น โดยแต่ละพิกเซลจะมีขนาดของข้อมูลเท่ากับ 1 บิต
- ภาพ 16 ระดับ คือ ในแต่ละพิกเซลจะมีขนาดของข้อมูล 4 บิต ซึ่งสามารถแสดงภาพได้ความเข้มถึง  $2^4$  ระดับ
- ภาพ 256 ระดับ คือ ในแต่ละพิกเซลจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงภาพได้ความเข้มถึง  $2^8$  ระดับ

โดยทั่วไปวิธีการประมวลผลภาพเชิงตัวเลขที่ทำให้คอมพิวเตอร์สามารถรู้วัตถุในภาพได้ นั้นแบ่งออกได้เป็นสองระดับด้วยกันคือ การประมวลผลภาพในระดับต่ำ (Low-Level Image Processing) และการประมวลผลภาพในระดับสูง (High-Level Image Processing) ซึ่งการประมวลผลภาพในระดับต่ำจะเป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมด เพื่อหาตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ โดยมีจุดประสงค์ที่จะนำตัวแปรเหล่านั้นไปใช้ในการประมวลผลภาพระดับสูงต่อไป โดยทั่วไปแล้ว การประมวลผลภาพระดับต่ำจะประกอบด้วย การประมวลผลภาพก่อน (Preprocessing) เช่น การกำจัดสัญญาณรบกวน เป็นต้น

การประมวลผลระดับสูงเป็นการนำผลลัพธ์ หรือสัญลักษณ์ที่ได้จากการประมวลผลภาพระดับต่ำมาตีความหรือประมวลผลเพื่อให้คอมพิวเตอร์สามารถรู้จักและเข้าใจภาพได้ สำหรับความแตกต่างของการประมวลผลภาพ ทั้ง 2 ประเภทนั้นคือ การประมวลผลภาพระดับต่ำจะใช้ค่าความสว่างของจุดภาพ (พิกเซล) ส่วนการประมวลผลระดับสูงนั้นข้อมูลภาพที่นำมาประมวลผลจะถูกแสดงในรูปสัญลักษณ์ ซึ่งสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่าง ๆ ที่อยู่ในภาพ เช่น ขนาดของวัตถุ รูปร่าง และความสัมพันธ์กันระหว่างวัตถุภาพ

## 2.6 ประเภทของภาพ

ประเภทของภาพนั้นแบ่งได้หลายประเภท ได้แก่ RGB, CMYK, HSB, Grayscale ซึ่งในปริญญาานิพนธ์นี้ได้นำภาพมาใช้ด้วยกัน 2 ประเภท คือ ภาพสี RGB และภาพสีเทา

### 2.6.1 ภาพสี RGB

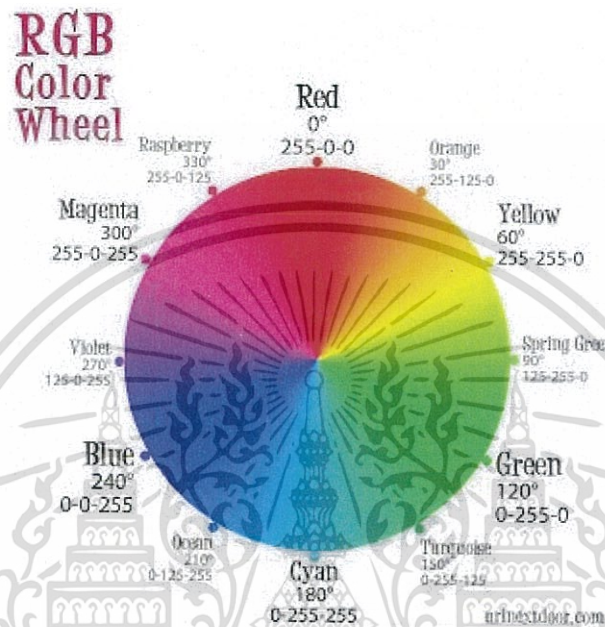
ภาพสี RGB เป็นระบบสีของแสง โดยอาศัยหลักการหักเหของแสงผ่านแท่งแก้วปริซึม จะเกิดแถบสีที่เรียกว่า สเปกตรัม (Spectrum) ซึ่งแยกสีตามที่ยาวตามองเห็นได้ 7 สี คือ แดง แสด เหลือง เขียว น้ำเงิน คราม ม่วง ซึ่งเป็นพลังงานอยู่ในรูปของรังสีที่มีช่วงคลื่นที่สายตาสามารถมองเห็นได้ แสงสีม่วง (Violet) มีความถี่คลื่นสูงที่สุด คลื่นแสงที่มีความถี่สูงกว่าแสงสีม่วง เรียกว่า อัลตราไวโอเล็ต (Ultraviolet) และคลื่นแสงสีแดง (Red) มีความถี่คลื่นต่ำที่สุด คลื่นแสงที่ต่ำกว่าแสงสีแดง เรียกว่า อินฟราเรด (Infrared) คลื่นแสงที่มีความถี่สูงกว่าสีม่วงและต่ำกว่าสีแดงนั้นสายตาสมองมนุษย์ไม่สามารถรับได้ แสงสีในระบบ RGB นี้เกิดจากแสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue) และสีเขียว (Green) ทั้งสามสีถือเป็นแม่สีของแสง เมื่อนำมาฉายรวมกันจะทำให้เกิดสีใหม่อีก 3 สี คือ สีแดงมาเจนต้า (Magenta) สีฟ้าไซแอน (Cyan) และสีเหลือง (Yellow) และถ้าฉายแสงสีทั้งหมดรวมกันจะได้แสงสีขาว จากคุณสมบัติของแสงนี้ได้นำมาใช้ประโยชน์โดยทั่วไป เช่น ในการฉายภาพยนตร์ การบันทึกภาพวิดีโอ ภาพโทรทัศน์ การสร้างภาพเพื่อการนำเสนอทางจอคอมพิวเตอร์ และการจัดแสงสีในการแสดง เป็นต้น

RGB เป็นระบบสีที่ประกอบด้วยแม่สี 3 สีคือ แดง (Red), เขียว (Green) และ น้ำเงิน (Blue) ในสัดส่วนความเข้มที่แตกต่างกัน หากใช้ความเข้มแต่ละสีในช่วง 0-255 หรือ  $2^8$  เมื่อนำมาผสมกันทำให้เกิดสีต่าง ๆ บนจอคอมพิวเตอร์ได้มากถึง  $2^8 \times 2^8 \times 2^8$  หรือ  $2^{24}$  สีหรือ 16.7 ล้านสี และจุดที่สีทั้งสามสีมีความเข้มสูงสุดรวมกันจะกลายเป็นสีขาว นิยมเรียกการผสมสีแบบนี้ว่าแบบ “Additive” หรือการผสมสีแบบบวก ซึ่งเป็นการผสมสีขั้นที่ 1 หรือถ้านำเอา Red Green Blue มาผสมครั้งละ 2 สี ก็จะทำให้เกิดสีใหม่ เช่น

Blue + Green = Cyan

Red + Blue = Magenta

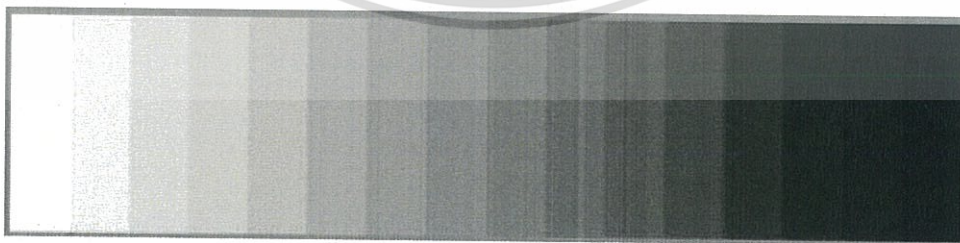
Red + Green = Yellow



รูปที่ 2.14 แสดงวงล้อระบบสี RGB

### 2.6.2 ภาพระดับสีเทา (Grayscale Image)

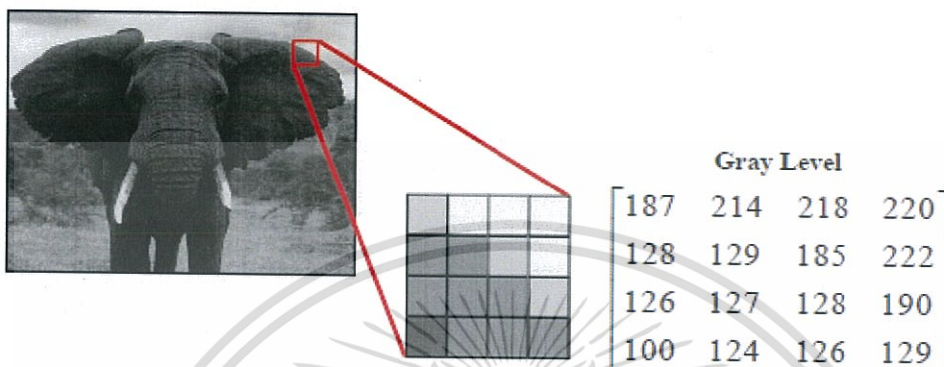
ภาพระดับสีเทาเป็นภาพซึ่งค่าในแต่ละจุดภาพคือค่าความสว่างของแต่ละตำแหน่งของจุดภาพนั้น ซึ่งค่าที่เป็นไปได้ของภาพระดับสีเทาทั้งหมดขึ้นอยู่กับจำนวนบิตที่ใช้ ตัวอย่างเช่น ภาพระดับสีเทา 8 บิต จะมีระดับความสว่างทั้งหมด  $2^8$  หรือ 256 ระดับในช่วง 0-255 แสดงระดับความสว่างดังในรูปที่ 2.15



รูปที่ 2.15 แสดงระดับความสว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการระบุความสว่างหรือความเข้มของภาพระดับสีเทานั้นจะมีค่าตั้งแต่ 0-255 เก็บเป็นข้อมูลในอาร์เรย์สองมิติ จากรูปตัวอย่างที่ 2.16 รูปทางซ้ายมีพื้นที่กรอบสี่เหลี่ยมเล็กในส่วนประกอบเล็ก ๆ นั้นนำมาแสดงเป็นข้อมูลตัวเลขได้ในรูป 2.16 รูปขวา



รูปที่ 2.16 แสดงระดับความสว่างของแต่ละพิกเซล

### 2.6.3 การแปลงภาพสีเป็นภาพระดับสีเทา (Color Conversion to Grayscale)

ภาพที่รับเข้ามาในขั้นตอนแรกเป็นภาพที่อยู่ในระบบปริภูมิสีแบบ RGB ดังนั้นแต่ละพิกเซลของภาพจะประกอบด้วยค่าของเซตที่แสดงถึง ค่าของ R ค่าของ G และค่าของ B ระบบจะทำการเปลี่ยนให้เป็นภาพระดับสีเทา (Grayscale) เพื่อให้สามารถวิเคราะห์ภาพได้ง่ายขึ้น เพราะเมื่อแปลงภาพเป็นระดับสีเทาแล้วจะทำให้แต่ละจุดภาพของภาพจะเหลือเพียงค่าความเข้มของสีมีค่าตั้งแต่ 0 ถึง 255 สามารถทำได้หลายแบบขึ้นอยู่กับอัตราส่วนที่ใช้ แต่อัตราส่วนที่นิยมใช้แสดงดังสมการที่ 2.1

$$G_R = 0.299R_S + 0.587G_S + 0.114B_S \quad (2.1)$$

โดยที่

$G_R$  หมายถึง ค่าเอาต์พุต พิกเซลของภาพระดับสีเทา

$R_S$  หมายถึง ค่าอินพุต พิกเซลสีแดง

$G_S$  หมายถึง ค่าอินพุต พิกเซลสีเขียว

$B_S$  หมายถึง ค่าอินพุต พิกเซลสีน้ำเงิน

## 2.7 การตรวจจับใบหน้า (Face Detection)

การตรวจจับใบหน้าคือ กระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ หลังจากนั้น จะทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไป เพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการ จำแนก และอัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธี ซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนกใบหน้าได้แม่นยำและรวดเร็วขึ้นเป็นอย่างมาก

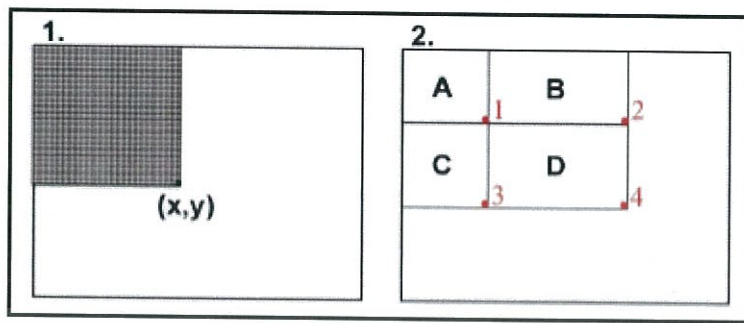
วิธีการที่ใช้ในการตรวจจับใบหน้าที่มีความสามารถในการประมวลผลได้รวดเร็วและมีอัตรา ความถูกต้องในการตรวจหาสูง ซึ่ง Paul Viola และ Michael J. Jones ได้คิดค้นและตีพิมพ์ในปี ค.ศ. 2001 โดยทั่วไปมักจะเรียกว่า Viola-Jones method ซึ่งอัลกอริทึมที่ได้นำเสนอนั้นมีการนำเสนอ วิธีการแทนรูปภาพที่เรียกว่า “Integral Image” ซึ่งช่วยให้การคำนวณ feature ทำได้รวดเร็วขึ้น และได้มีการปรับปรุงอัลกอริทึมการเรียนรู้โดยมีพื้นฐานจาก AdaBoost ซึ่งเลือกเอาเฉพาะ critical features (features ที่ให้ classifier ที่มีประสิทธิภาพสูงสุด) นอกจากนี้ยังได้อธิบายถึงการ รวม classifier แบบ cascade ซึ่งช่วยให้ส่วนพื้นหลังของภาพถูกปฏิเสธได้เร็วและเน้นการคำนวณไป ที่บริเวณที่มีลักษณะคล้ายวัตถุที่สนใจมากขึ้น

หลักการพื้นฐานของอัลกอริทึมของ Viola-Jones คือการมีอยู่ของลักษณะเด่นของ Haar ถูก กำหนดโดยการลบค่าเฉลี่ยบริเวณจุดภาพมืดออกจากค่าเฉลี่ยบริเวณจุดภาพสว่าง ถ้าผลลัพธ์ที่ได้มีค่า มากกว่าค่าขีดแบ่ง แสดงว่ามีลักษณะเด่นของ Haar อยู่



รูปที่ 2.17 แสดงตัวอย่างลักษณะเด่นของ Haar ที่ถูกใช้ใน OpenCV

การตรวจสอบหาลักษณะเด่นของ Haar ในแต่ละส่วนของภาพอย่างมีประสิทธิภาพนั้น Viola และ Jones ใช้เทคนิคที่เรียกว่าการรวมภาพ (Integral Image) โดยการรวมจุดภาพเล็ก ๆ เข้าด้วยกัน จากรูปที่ 2.18 การจะหาค่าของรูปสี่เหลี่ยม  $D = G(x_4, y_4) - G(x_2, y_2) - G(x_3, y_3) + G(x_1, y_1)$



(a)

(b)

รูปที่ 2.18 แสดงเทคนิคการรวมภาพ (a) หลังจากทีรวมภาพแล้ว จุดภาพที่ตำแหน่ง  $(x,y)$  จะรวมค่าของทุกจุดภาพในสี่เหลี่ยมสีดำ, (b) ผลรวมค่าของทุกจุดภาพในสี่เหลี่ยม

$$D = G(x_4,y_4) - G(x_2,y_2) - G(x_3,y_3) + G(x_1,y_1)$$

ในการเลือกลักษณะเด่นของ Haar และการตั้งระดับค่าขีดแบ่ง วิธีการของ Viola-Jones ใช้ machine-learning method ที่เรียกว่า “AdaBoost” ซึ่งรวมตัวกรองจำนวนมาก (weak classifier) เข้าด้วยกันเพื่อสร้าง strong classifier ที่นำคำตอบที่ weak classifier คิดว่าภาพในส่วนนั้นเป็นใบหน้ามารวมกันเป็นคำตอบสุดท้ายว่าภาพส่วนใดบ้างเป็นภาพใบหน้าซึ่ง AdaBoost จะทำการเลือกชุดของ weak classifier ในการรวมและกำหนดน้ำหนักของแต่ละตัว การรวมกันของน้ำหนักก็คือ strong classifier นั้นเอง

### 2.7.1 การเรียนรู้แบบ AdaBoost และ Cascade Classifier

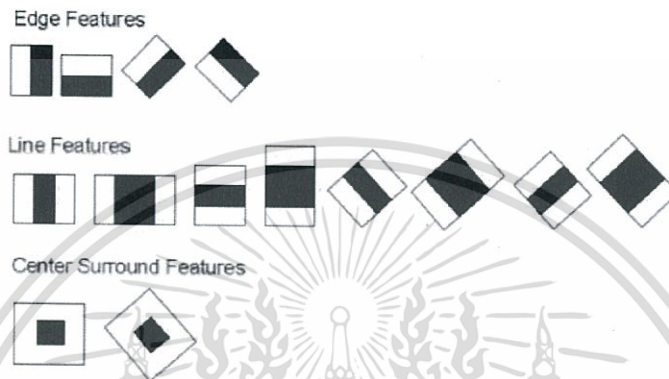
ขั้นตอนวิธีการเรียนรู้แบบ AdaBoost (AdaBoost learning algorithm) เป็นขั้นตอนวิธีการเรียนรู้เพื่อค้นหาค่าของกลุ่มฟังก์ชันที่มีลักษณะใกล้เคียงกับภาพใบหน้า โดยที่ภาพ positive คือ ภาพใบหน้าที่ต้องการตรวจจับ ส่วนภาพ negative คือ ภาพทั่ว ๆ ไปที่ไม่ใช่ใบหน้า ซึ่งการจำแนกกลุ่มของฟังก์ชันจะทำภายในส่วนย่อย (sub window) ของภาพโดยใช้ weak classifier ของ feature ที่  $j$  ( $h_j$ )

$$h_j(x) = \begin{cases} -1 & ; \text{if } f_j(x) \leq 0 \\ 1 & ; \text{otherwise} \end{cases} \quad (2.2)$$

เมื่อ  $f(x)$  คือค่าความแตกต่างของผลรวมฟังก์ชันในพื้นที่ขาว และดำของ Haar-like feature มีค่าตามสมการที่ 2.3

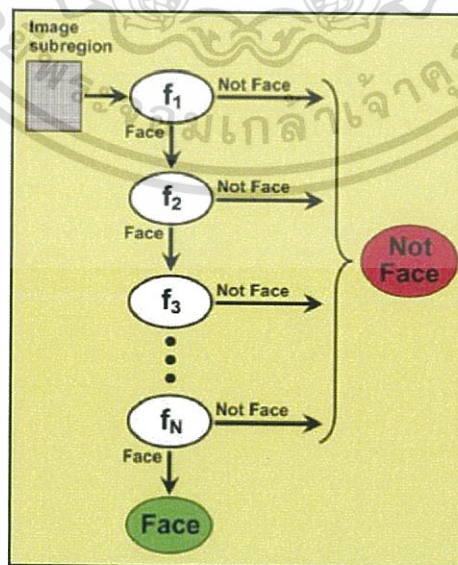
$$f(x) = \text{Sum}_{\text{Black rectangle}} - \text{Sum}_{\text{White rectangle}} \quad (2.3)$$

โดยที่  $x$  คือภาพตัวอย่าง และ Sum คือผลรวมของภาพอินทิกรัลในบริเวณพื้นที่ขาวและดำของ  $x$



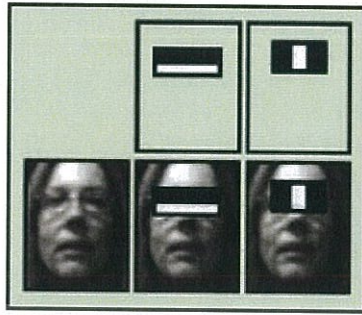
รูปที่ 2.19 ตัวอย่างรูปแบบของ Haar like feature

Cascade Classifier เป็นวิธีการจำแนกรูปร่างที่ต้องการ โดยนำการจำแนกตั้งที่กล่าวมาแล้วมาทำซ้ำหลาย ๆ รอบ (stage) ซึ่งในแต่ละรอบก็จะตัดพื้นที่ที่เป็นภาพ negative ออกไปในทุกรอบที่พบ เมื่อจบกระบวนการแล้วจำนวนของ sub window ที่เป็น negative จะลดลงจนได้ใบหน้าที่ต้องการ และในการจำแนกเพื่อหารูปร่างที่ต้องการนี้จะทำกับภาพอินทิกรัล ซึ่งขั้นตอนของ Cascade Classifier แสดงได้ดังรูปที่ 2.20



รูปที่ 2.20 กระบวนการทำงานของ Classifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงตัวอย่างตัวกรองสองตัวแรกใน Viola-Jones Cascade

จากรูปที่ 2.21 จะพบว่าตัวกรองแรกใช้หลักการที่ว่าบริเวณแก้มสว่างกว่าบริเวณดวงตา และตัวกรองตัวที่สองใช้หลักการที่ว่าบริเวณสันจมูกจะสว่างกว่าบริเวณดวงตา



รูปที่ 2.22 ตัวอย่างตัวกรองอื่น ๆ ใน Viola-Jones Cascade

หลักการของอัลกอริทึมค้นหาหน้าของ Viola-Jones คือการใช้ตัวตรวจหาสแกนหลาย ๆ ครั้งบนภาพเดิมด้วยขนาดที่แตกต่างกัน ถึงแม้ว่าจะมีใบหน้ามากกว่าหนึ่งหน้า ผลลัพธ์ของ sub-window จำนวนมากยังคงเป็นลบ (negative non-faces) ซึ่งปัญหานี้แก้ได้โดยใช้หลักการ “ปฏิเสธสิ่งที่ไม่ใช่ใบหน้า แทนการค้นหาใบหน้า” เพราะการตัดสินใจว่าบริเวณใด ๆ ไม่ใช่ใบหน้านั้น ทำได้เร็วกว่าการค้นหาใบหน้า และได้มีการสร้างตัวจำแนกประเภทแบบ cascaded (Cascaded Classifier) คือเป็น Classifier หลายตัวต่อกันเป็นลำดับดังแสดงในรูปที่ 2.20 ซึ่งเมื่อ sub-window ถูกจัด

ประเภทเป็น ไม่ใช่ใบหน้า (non-face) จะถูกปฏิเสธทันที แต่ในทางตรงกันข้าม ถ้า sub-window นั้น ถูกจำแนกเป็นมีโอกาสเป็นใบหน้า (maybe-face) จะถูกส่งต่อไปยัง Classifier ตัวถัดไปตามลำดับ และกล่าวได้ว่ายังมีจำนวนชั้นของ Classifier มากเท่าใด โอกาสที่ sub-window จะเป็นใบหน้าจะยิ่งมีมากขึ้น

## 2.8 การรู้จำใบหน้า (Face Recognition)

ระบบรู้จำใบหน้า ถือว่าเป็นหนึ่งในระบบที่ใช้ในการพิสูจน์ยืนยันตัวตนบุคคลโดยใช้คุณลักษณะจำเพาะทางสรีระ (Biometric) ซึ่งระบบจะทำงานโดยการเปรียบเทียบใบหน้าจากภาพถ่ายดิจิทัลหรือภาพจากกล้องวิดีโอของบุคคลที่สนใจกับฐานข้อมูลใบหน้าที่มีอยู่ และเมื่อเปรียบเทียบเสร็จก็จะแสดงผลใบหน้าที่อยู่ในฐานข้อมูลที่มีใบหน้าเหมือนกับภาพที่นำมาเปรียบเทียบออกมา ระบบรู้จำใบหน้านั้นได้ถูกพัฒนาอย่างต่อเนื่องเป็นเวลานานมากแล้ว เนื่องจากเป็นระบบที่ได้รับความสนใจมาจากนักวิชาการหลายสาขาวิชา จึงทำให้ระบบรู้จำใบหน้ามีผู้คนสนใจศึกษาและพัฒนากันอย่างมากมาย ทำให้มีการพัฒนาอัลกอริทึมในการทำงานของระบบออกมามากมายหลายรูปแบบแตกต่างกันไป อันเนื่องมาจากปัจจัยด้านองค์ความรู้และเทคโนโลยีของอุปกรณ์ต่าง ๆ ที่พัฒนาขึ้นให้มีความเหมาะสมที่จะนำมาใช้ในระบบ จึงทำให้ต้องออกแบบอัลกอริทึมใหม่ให้เหมาะสมกับอุปกรณ์ใหม่ ๆ ด้วยในปัจจุบันระบบรู้จำใบหน้าได้มีการพัฒนาไปอย่างมาก ทำให้ระบบมีความน่าเชื่อถือมากขึ้น จนมีการนำระบบรู้จำใบหน้ามาใช้ประโยชน์กันอย่างแพร่หลายในต่างประเทศหลาย ๆ ประเทศได้มีการนำระบบรู้จำใบหน้ามาติดตั้งในสนามบินเพื่อป้องกันคนร้ายหนีเข้าออกนอกประเทศ และใช้สำหรับการยืนยันตัวตนคนร้ายในคดีต่าง ๆ ด้วย

### 2.8.1 หลักการทำงานของระบบรู้จำใบหน้า

ระบบรู้จำใบหน้าถูกออกแบบมาให้ทำการเปรียบเทียบใบหน้าบุคคลที่สนใจกับฐานข้อมูลใบหน้าที่มีอยู่ โดยระเบียบวิธีการที่ใช้ในขั้นตอนการสร้างแม่แบบและขั้นตอนการเปรียบเทียบอาจแตกต่างกันไป แล้วแต่การออกแบบระบบของแต่ละระบบ แต่ไม่ว่าจะมีระเบียบวิธีการในการทำงานในขั้นตอนการสร้างอย่างไร แต่ขั้นตอนการทำโดยรวมของระบบก็ยังคงเหมือนกันอยู่ ตัวอย่างของระเบียบวิธีการรู้จำใบหน้าได้แก่ PCA หรือ Principal Component Analysis หรือเรียกว่า การวิเคราะห์องค์ประกอบหลัก คือเทคนิคในการผสมลักษณะเด่นในเวกเตอร์นำเข้าไปเพื่อสร้างเวกเตอร์ใหม่ที่อยู่ในปริภูมิ (Subspace) ที่มีมิติน้อยกว่าเวกเตอร์เดิมโดยการผสมที่ใช้นั้นจะเป็นการผสมเชิงเส้นหรือ Linear Combination นั่นคือการเอาลักษณะเด่นมาคูณค่าคงที่แล้วจึงบวกกัน

ปริญญานิพนธ์นี้ได้ใช้หลักการ Fisherfaces ซึ่งจะเป็นวิธีการที่ปรับปรุงประสิทธิภาพของเทคนิคใบหน้าไอเกน (Eigenfaces) โดยใช้วิธี class-specific transformation matrix ดังนั้นจึงไม่จำเป็นต้องจับภาพให้ชัดเจนเหมือนวิธี Eigenfaces การวิเคราะห์ค่าความต่างของใบหน้าจะถูกนำมาใช้แทนการวิเคราะห์คุณลักษณะของใบหน้าแทน เพื่อระบุความแตกต่างระหว่างตัวบุคคล ที่สำคัญคือการทำงานแบบ Fisherfaces มีปัจจัยหลักที่ขึ้นอยู่กับข้อมูลอินพุตอีกด้วย ในทางปฏิบัติแล้วการใช้งาน Fisherfaces กับภาพในฐานข้อมูลมีลักษณะองค์ประกอบของแสงที่พอเหมาะก็จะทำให้สามารถระบุตัวตนได้อย่างถูกต้อง เมื่อทดลองสแกนใบหน้าที่มีองค์ประกอบของแสงที่ไม่เพียงพอจะทำให้ผลลัพธ์ที่แสดงออกมานั้นอาจไม่สามารถระบุตัวตนได้

### 2.8.1.1 Fisherfaces

คือ การวิเคราะห์จำแนกประเภทเชิงเส้น (Linear Discriminate Analysis : LDA) มีจุดประสงค์เพื่อจำแนกวัตถุต่าง ๆ LDA จะนิยมใช้ในการจำแนกข้อมูลและการลดมิติ LDA รู้จักกันในอีกชื่อหนึ่งคือ Fisher Linear Discriminate (FLD) เป้าหมายของการทำ LDA คือการเพิ่มปริมาณการกระจายระหว่างกลุ่มให้มากที่สุด และลดปริมาณการกระจายภายในกลุ่มให้น้อยที่สุด โดย LDA แบ่งเป็นขั้นตอนการฝึกสอน และขั้นตอนการทดสอบเช่นเดียวกับ Eigenfaces

1. การกระจายตัวภายในกลุ่ม (within-class scatter matrix) แทนโดย  $S_w$  สามารถหาได้จากสมการ 2.4

$$S_w = \sum_{j=1}^C \sum_{i=1}^{N_j} (\Gamma_i^j - \mu_j)(\Gamma_i^j - \mu_j)^T \quad (2.4)$$

เมื่อ  $C$  คือ จำนวนกลุ่ม

$N_j$  คือ จำนวน Sample ในกลุ่มที่  $j$

$\Gamma_i^j$  คือ Sample ที่  $i$  ของกลุ่มที่  $j$

$\mu_j$  คือ Mean ของกลุ่มที่  $j$

2. การกระจายตัวระหว่างกลุ่ม (between-class scatter matrix) แทนโดย  $S_B$  สามารถหาได้จากสมการ 2.5

$$S_B = \sum_{j=1}^C (\mu_j - \mu)(\mu_j - \mu)^T \quad (2.5)$$

เมื่อ  $\mu$  คือ Mean รวมของกลุ่มทั้งหมด

3. จากนั้นสามารถหาค่า eigenvectors ของ  $S_W, S_B$  จากเมตริกซ์  $W$  จากสมการ 2.6

$$W_{\text{opt}} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} \quad (2.6)$$

### 2.8.1.2 การคำนวณหาค่าระยะห่าง Euclidean distance

คำนวณหาค่าระยะห่างระหว่างค่าสัมประสิทธิ์ใบหน้าที่จะนำมาทดสอบ กับค่าสัมประสิทธิ์ภาพใบหน้าในฐานข้อมูลด้วยวิธีหาค่าระยะห่าง Euclidean ดังสมการที่ 2.7

$$\begin{aligned} \varepsilon_k &= \|\Omega_p - \Omega_k\| \\ &= \sqrt{\sum_{i=1}^k (\omega_i - \omega_{ki})^2} \end{aligned} \quad (2.7)$$

โดยที่  $1 \leq k \leq M$

เมื่อ  $\varepsilon_k$  = ค่าระยะห่าง Euclidean

$\Omega_p$  = ใบหน้าไอเทมที่  $k$

$\Omega_k$  = ใบหน้าทดสอบ

$\omega_{ki}$  = สัมประสิทธิ์แต่ละตัวใน  $\Omega_k$

$\omega_i$  = สัมประสิทธิ์แต่ละตัวใน  $\Omega_p$

$M$  = จำนวนรูปภาพที่จะนำมาทดสอบ

จากสมการที่ 2.7 คือสมการหาค่าระยะห่าง Euclidean ระหว่างใบหน้าในฐานข้อมูลและใบหน้าอินพุต จะได้ค่าระยะห่าง Euclidean ทั้งหมด  $M$  ค่า เป็นตัวแทนแต่ละภาพในฐานข้อมูล โดยวิธีการ Fisherfaces จะนำค่าระยะห่าง Euclidean ที่น้อยที่สุดมาตัดสินว่ารูปที่เป็นอินพุตนั้น มีความเหมือนกับภาพใดในฐานข้อมูลมากที่สุด

## 2.9 MySQL (มายเอสคิวแอล)

MySQL จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ตัวหนึ่ง สร้างขึ้นโดย David Axmark, Allan Larsson ชาวสวีเดน และ Michael Widenius ชาวฟินแลนด์ ปัจจุบันผลิตภัณฑ์ภายใต้ MySQL เป็นลิขสิทธิ์ของบริษัท Oracle MySQL เป็นที่นิยมกันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกของอินเทอร์เน็ต สาเหตุเพราะว่า MySQL เป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูง เป็นทางเลือกใหม่จากผลิตภัณฑ์ระบบจัดการฐานข้อมูลในปัจจุบันที่มักจะเป็นการผูกขาดของผลิตภัณฑ์เพียงไม่กี่ตัว นักพัฒนาระบบฐานข้อมูลที่เคยใช้ MySQL ต่างยอมรับในความสามารถความเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมาก ทั้งยังสนับสนุนการใช้งานบนระบบปฏิบัติการมากมาย ไม่ว่าจะเป็น Linux, Mac OS, Windows หรือ Raspbian ก็ตาม นอกจากนี้ MySQL ยังสามารถใช้งานร่วมกับ Web Development Platform ทั้งหลาย เช่น Java, Perl, PHP, Python เป็นต้น

MySQL จัดเป็นซอฟต์แวร์ประเภท Open Source Software สามารถดาวน์โหลด Source Code ต้นฉบับได้จากอินเทอร์เน็ต โดยไม่เสียค่าใช้จ่ายใด ๆ การแก้ไขก็สามารถทำได้ตามความต้องการ MySQL ยึดถือสิทธิบัตรตาม GPL (GNU General Public License) ซึ่งเป็นข้อกำหนดของซอฟต์แวร์ประเภทนี้ส่วนใหญ่ โดยจะเป็นการชี้แจงว่า สิ่งใดทำได้หรือทำไม่ได้สำหรับการใช้งานในกรณีต่าง ๆ

MySQL ได้รับการยอมรับและทดสอบเรื่องของความรวดเร็วในการใช้งาน โดยจะมีการทดสอบและเปรียบเทียบกับผลิตภัณฑ์ทางด้านฐานข้อมูลอื่นอยู่เสมอ มีการพัฒนาอย่างต่อเนื่อง โดยเริ่มตั้งแต่เวอร์ชันแรก ๆ ที่ยังไม่ค่อยมีความสามารถมากนัก มาจนถึงทุกวันนี้ MySQL ได้รับการพัฒนาให้มีความสามารถมากยิ่งขึ้น รองรับข้อมูลจำนวนมาก สามารถใช้งานหลายผู้ใช้ได้พร้อม ๆ กัน (Multi-user) มีการออกแบบให้สามารถแบ่งงานออก (Multi-threaded) เพื่อช่วยให้การทำงานเร็วยิ่งขึ้น และมีการเชื่อมต่อที่ดีขึ้น การกำหนดสิทธิและการรักษาความปลอดภัยของข้อมูลมีความรัดกุมน่าเชื่อถือยิ่งขึ้น เครื่องมือหรือโปรแกรมสนับสนุนทั้งของตัวเองและของผู้พัฒนาอื่น ๆ มีมากยิ่งขึ้น นอกจากนี้สิ่งหนึ่งที่สำคัญคือ “MySQL ได้รับการพัฒนาไปในแนวทางตามข้อกำหนดมาตรฐาน SQL ดังนั้นสามารถใช้คำสั่ง SQL ในการทำงาน MySQL ได้” นักพัฒนาที่ใช้ SQL มาตรฐานอยู่แล้วไม่ต้องศึกษาคำสั่งเพิ่มเติม แต่อาจจะต้องเรียนรู้ถึงรูปแบบและข้อจำกัดบางอย่างโดยเฉพาะ

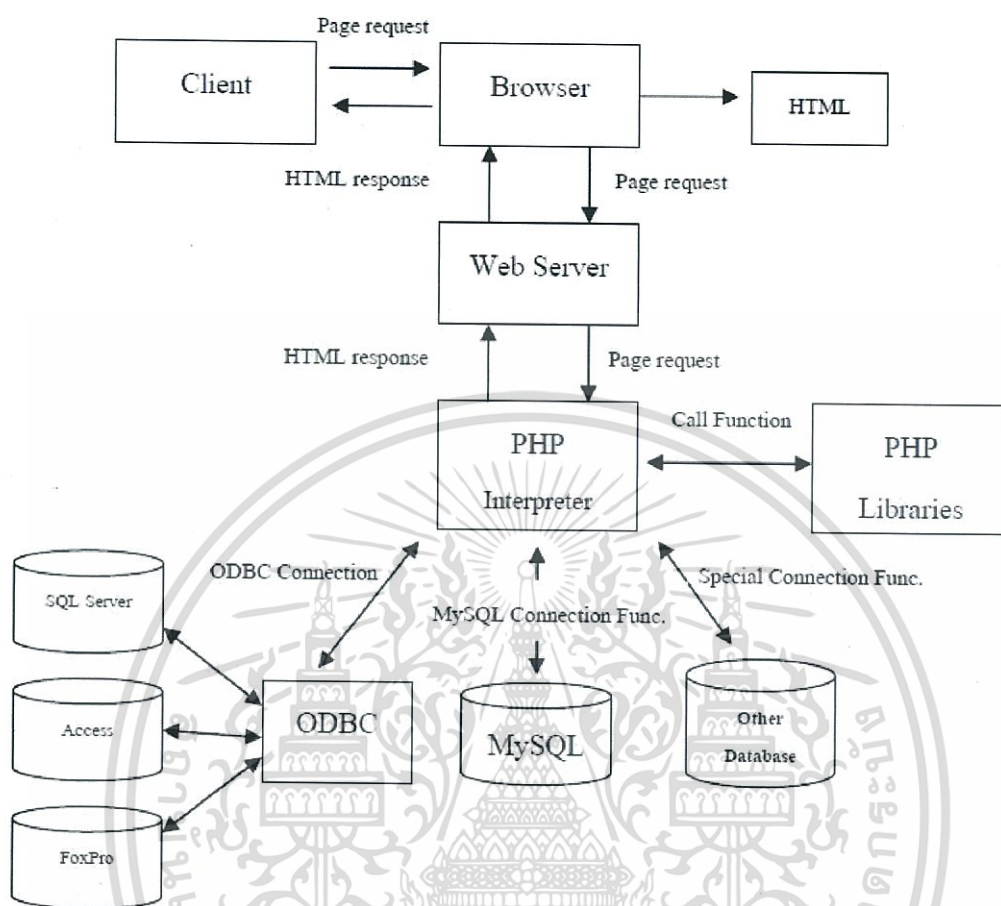
ทุกวันนี้มีการนำ MySQL ไปใช้ในระบบต่าง ๆ มากมาย ไม่ว่าจะเป็นระบบเล็ก ๆ ที่มีจำนวนตารางข้อมูลน้อย มีความสัมพันธ์ของข้อมูลในแต่ละตารางไม่ซับซ้อน เช่น ระบบฐานข้อมูลบุคคลในแผนกเล็ก ๆ ไปจนถึงระบบจัดการข้อมูลขนาดใหญ่ที่ประกอบด้วยตารางข้อมูลมากมาย มีความสัมพันธ์ของข้อมูลในแต่ละตารางซับซ้อน เช่น ระบบคลังสินค้า ระบบบัญชีเงินเดือน เป็นต้น

โดยเฉพาะอย่างยิ่งในปัจจุบัน มีการใช้ MySQL เป็น Database Server เพื่อการทำงานสำหรับ Web Database Application ในโลกของอินเทอร์เน็ตมากขึ้น

## 2.10 ภาษา PHP

PHP ย่อมาจาก “PHP : Hypertext Preprocessor” เป็นภาษาที่นิยมใช้กันมากในการเขียน สคริปต์บนเว็บไซต์ (สคริปต์เป็นชุดคำสั่งควบคุมการทำงานของโปรแกรม มีรูปแบบง่าย ๆ ไม่ซับซ้อน และมีความยาวไม่มากนัก) PHP เป็นภาษาสคริปต์ที่เป็น Server Side Script และเป็น Open Source ที่ผู้ใช้สามารถดาวน์โหลดตัวแปลภาษาได้ฟรี ภาษา PHP ช่วยให้นักพัฒนาเว็บเพจสามารถเขียนเว็บเพจที่เป็นแบบไดนามิกได้อย่างรวดเร็ว ภาษา PHP จะทำงานร่วมกับเอกสาร HTML โดยการสร้างชุดคำสั่งแทรกระหว่าง Tag HTML และสร้างเป็นไฟล์ที่มีนามสกุล .php ไวยากรณ์ที่ใช้ใน PHP เป็นการนำรูปแบบของภาษาหลายภาษามาใช้ร่วมกัน เช่น C Perl และ Java ทำให้ผู้ใช้ที่มีพื้นฐานของภาษาเหล่านี้สามารถใช้งาน PHP ได้ไม่ยาก

PHP ทำงานโดยใช้ตัวแปลภาษาและทำงานหรือเอ็กซิคิวต์ที่ฝั่งเซิร์ฟเวอร์ เรียกการทำงานว่า เซิร์ฟเวอร์ไซด์ (Server Side) ส่วนการทำงานของบราวเซอร์ของฝั่งผู้ใช้ เรียกว่าไคลเอนต์ไซด์ (Client Side) การทำงานจะเริ่มต้นที่ผู้ใช้ส่งความต้องการผ่านเว็บเบราว์เซอร์ทาง HTTP (HTTP Request) ซึ่งอาจเป็นการกรอกแบบฟอร์ม หรือใส่ข้อมูลที่ต้องการ โดยเรียกไฟล์ PHP (ไฟล์นี้จะมีส่วนขยายเป็น php) เช่น test.php เมื่อเว็บเซิร์ฟเวอร์พบว่ามีคำร้องขอไฟล์ PHP (request) มันจะดึงชุดคำสั่งภายในไฟล์นี้ส่งไปให้ PHP Interpreter เพื่อทำหน้าที่แปลคำสั่งแล้วเอ็กซิคิวต์คำสั่งตามบรรทัดที่ระบุ คำสั่งนั้น ๆ จากนั้น PHP จะสร้างผลลัพธ์ในรูปแบบเอกสาร HTML ส่งกลับไปให้เว็บเซิร์ฟเวอร์เพื่อส่งต่อไปให้เบราว์เซอร์แสดงผลทางฝั่งผู้ใช้ต่อไป (HTTP Response)



รูปที่ 2.23 แสดงขั้นตอนการทำงาน PHP Script Request/Response

### 2.10.1 PHP และระบบการจัดการฐานข้อมูล

PHP มีความสามารถในการติดต่อกับฐานข้อมูลต่าง ๆ มากมาย เช่น การติดต่อผ่าน ODBC (Open Database Connectivity) เพื่อเชื่อมต่อกับระบบจัดการฐานข้อมูลอื่น ๆ เช่น MS Access, MS SQL Server หรือติดต่อกับ MySQL ที่เป็นระบบจัดการฐานข้อมูลที่ได้รับความนิยมสูงในปัจจุบัน การใช้ PHP เชื่อมต่อกับระบบจัดการฐานข้อมูลสามารถทำงานได้ในระบบปฏิบัติการต่างชนิดกัน เช่น UNIX, Linux และ Windows เป็นต้น ในการใช้ PHP ติดต่อกับระบบจัดการฐานข้อมูลนั้นจำเป็นต้องใช้คำสั่ง SQL ร่วมด้วย

- การสร้างส่วนเชื่อมโยงฐานข้อมูล  
คำสั่ง PHP ในการเชื่อมต่อกับระบบจัดการฐานข้อมูล MySQL มีดังนี้

```
MySQL_connect([hostname(port)[:path to socket]], [username], [password])
```

โดยที่ hostname หมายถึง ชื่อของเครื่อง (host) ที่ MySQL กำลังทำงาน เช่นอาจจะเป็นชื่อเครื่องหรือหมายเลข IP Address ที่กำหนดไว้สำหรับเครื่องนั้นก็ได้

username หมายถึง ชื่อผู้ที่กำหนดไว้ในการติดต่อกับฐานข้อมูล MySQL

password หมายถึง รหัสผ่านที่ใช้ร่วมกับ username

หลังจากที่ผู้ใช้ติดต่อกับฐานข้อมูล และทำงานเรียบร้อยแล้ว จะต้องทำการปิดการติดต่อกับฐานข้อมูลทุกครั้ง โดยใช้ คำสั่ง mysql\_close() เพื่อที่จะได้เป็นการลด overhead (การจองเซสชัน : session connection ในการติดต่อเซิร์ฟเวอร์)

- การเรียกใช้ฐานข้อมูลของ MySQL  
คำสั่งใน MySQL ที่ใช้เลือกฐานข้อมูลมาใช้งาน มีดังนี้

```
use "database name";
```

- สำหรับ PHP คำสั่งในการติดต่อกับ MySQL เพื่อเลือกใช้ฐานข้อมูล มีดังนี้

```
mysql_select_db("database name");
```

โดยที่ "database name" คือชื่อฐานข้อมูลที่ต้องการเลือกใช้ เช่น ต้องการเลือกใช้ฐานข้อมูลชื่อ book จะใช้คำสั่งดังนี้ mysql\_select\_db("book")

- เก็บค่าการติดต่อไว้ในตัวแปรเพื่อเรียกใช้งาน

ในการติดต่อกับฐานข้อมูลจะใช้คำสั่ง `mysql_query()` เพื่อเข้าถึงตาราง และข้อมูลภายในฐานข้อมูล แต่ก่อนการใช้คำสั่งนี้ ผู้ใช้ควรเก็บค่าการติดต่อไว้ในรูปของตัวแปรเพื่อความสะดวกในการนำมาใช้ภายหลัง

```
$query="select * from books where ".$searchtype."Like '".$searchterm."'";
```

ในตัวอย่างนี้เป็นการสร้าง query สำหรับค้นหาข้อมูลที่ได้จากข้อมูลของผู้ใช้ที่เก็บไว้ในตัวแปร `$searchtype` โดยค้นหาข้อมูลในตาราง `book` ซึ่งมีรูปแบบของการสืบค้นที่ถูกกำหนดโดยตัวแปร `$searchtype` ผลของการใช้ตัวแปร `$query` จะถูกส่งไปเก็บค่าในตัวแปร `$result` โดยฟังก์ชัน `mysql_query($query)` ดังนี้

```
$result = mysql_query($query);
```

## 2.11 phpMyAdmin

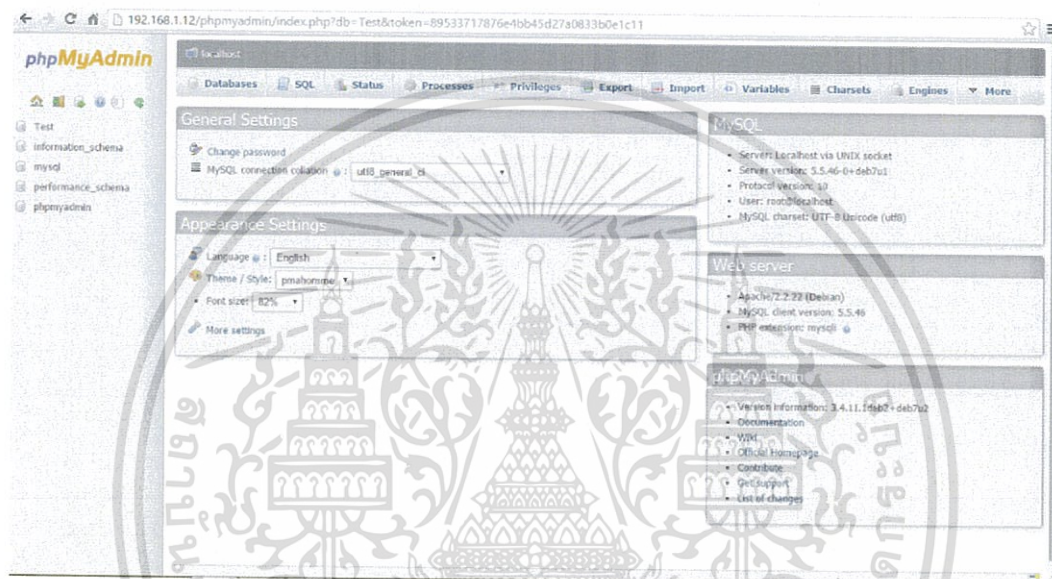
phpMyAdmin คือโปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูลของ MySQL แทนการสืบทอดคำสั่ง เนื่องจากถ้าต้องใช้คำสั่งจัดการกับฐานข้อมูล MySQL โดยตรงบางครั้งจะลำบากและยุ่งยากในการใช้งาน จึงมีเครื่องมือในการจัดการฐานข้อมูลขึ้นมาเพื่อให้สามารถจัดการตัว Database Management System (DBMS) ที่เป็น MySQL ได้ง่ายและสะดวกยิ่งขึ้น โดย phpMyAdmin ก็ถือเป็นเครื่องมือชนิดหนึ่งในการจัดการ

phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการข้อมูล MySQL ผ่าน Web browser ได้โดยตรง phpMyAdmin ตัวนี้จะทำงานบน Web server เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server

phpMyAdmin เป็นส่วนต่อประสานที่สร้างโดยภาษา PHP ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ สามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้าง TABLE ใหม่ ๆ และยังมี function ที่ใช้สำหรับการทดสอบการ query ข้อมูลด้วยภาษา SQL พร้อมกันนั้นยังสามารถทำการแทรก ลบ อัปเดต หรือแม้กระทั่งใช้คำสั่งต่าง ๆ เหมือนกับการใช้ภาษา SQL ในการสร้างตารางข้อมูล

ตัวอย่างความสามารถของ phpMyAdmin คือ

- 1.สร้างและลบ Database
- 2.สร้างและจัดการ Table เช่น แทรก record, ลบ record, แก้ไข record, ลบ Table, แก้ไข field
- 3.โหลดเท็กซ์ไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้
- 4.ทำการจัดการกับข้อมูล เช่น การสืบค้นข้อมูล (Query) ด้วยคำสั่ง SQL



รูปที่ 2.24 แสดงหน้า phpMyAdmin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

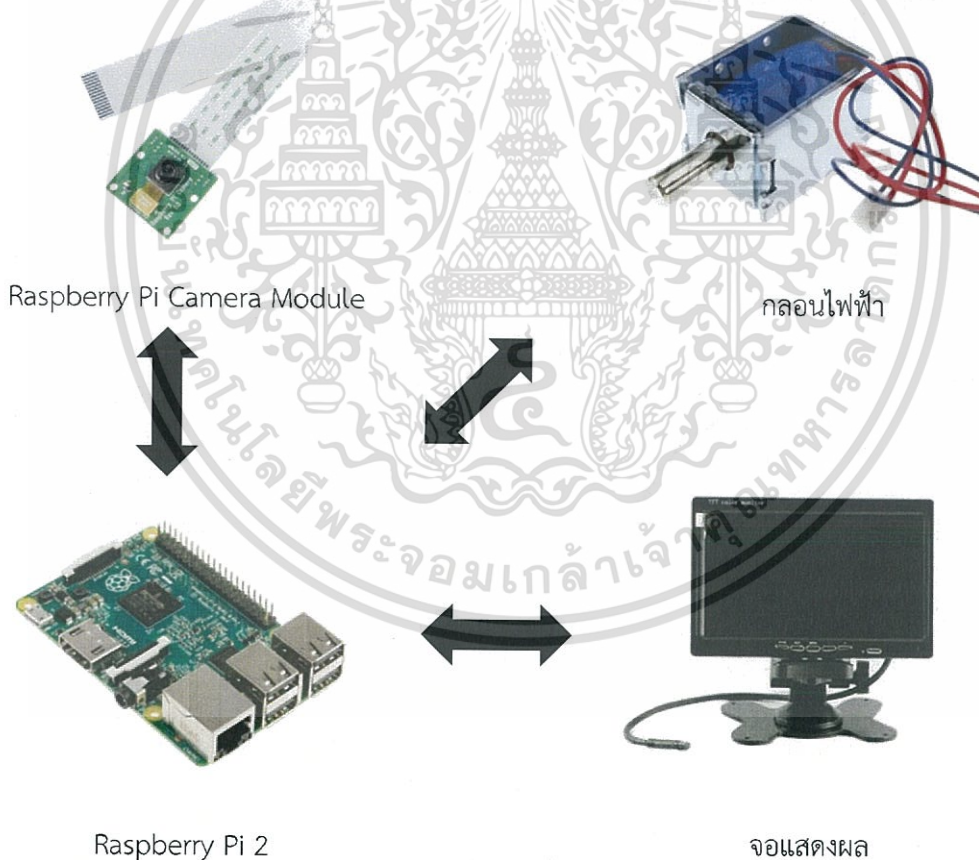
### บทที่ 3

## การออกแบบและการจัดทำปฏิญญาฉบับ

### 3.1 การออกแบบ

#### 3.1.1 Block Diagram

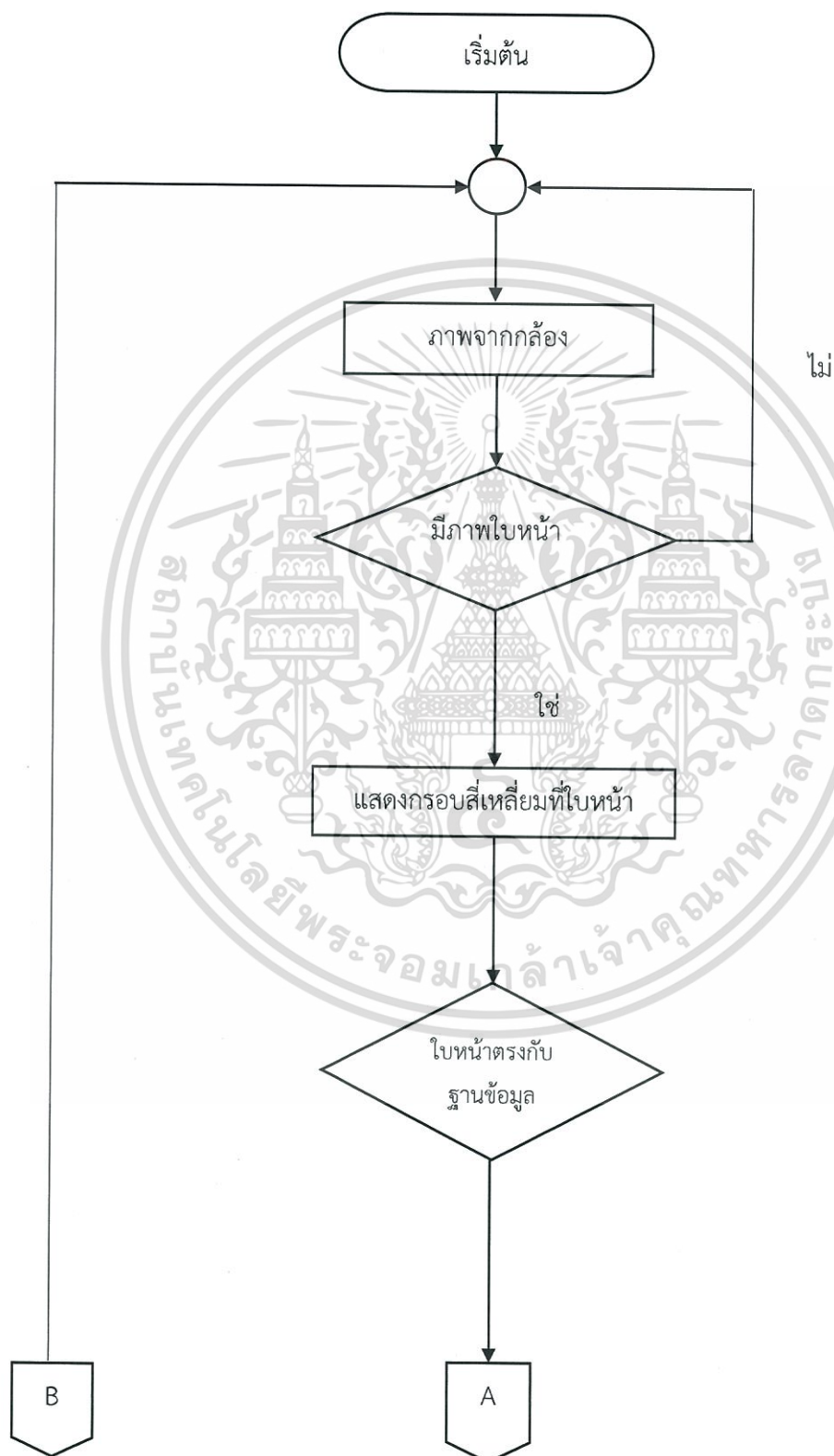
การทำงานของระบบแสดงได้ดังรูปที่ 3.1 ซึ่งแบ่งออกเป็น 4 ส่วน คือ 1) Raspberry Pi Camera Module จะทำงานในส่วนของการถ่ายวิดีโอ 2) Raspberry Pi เป็นส่วนประมวลผลรูปภาพที่ได้รับมาจาก Raspberry Pi Camera Module 3) จอแสดงผล เป็นจอแสดงผลผลลัพธ์ที่ได้ออกมาจากส่วนประมวลผล 4) กลอนไฟฟ้า (Solenoid Bolt Lock) ใช้สำหรับล็อกและปลดล็อกประตู



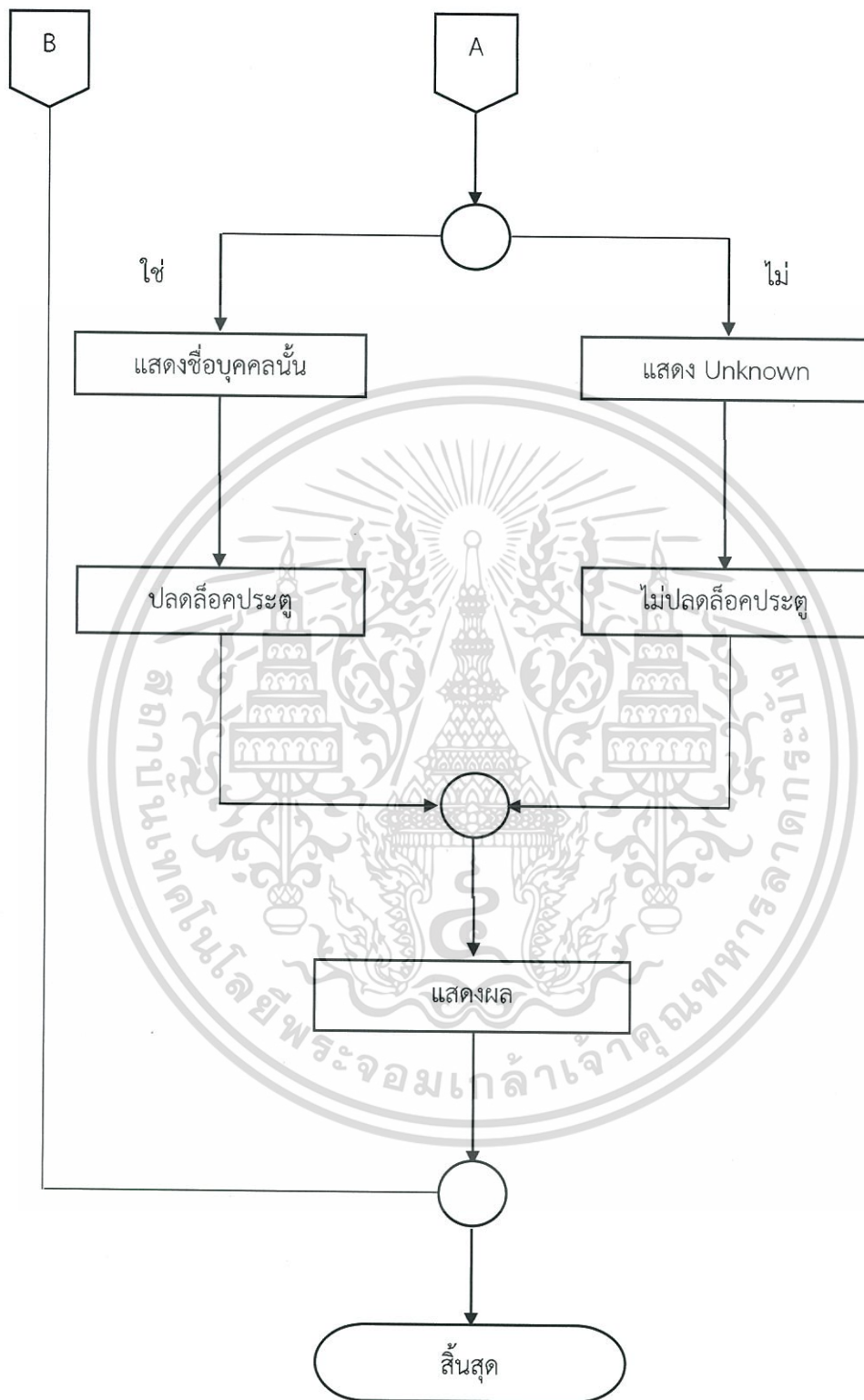
รูปที่ 3.1 บล็อกไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.1.2 Flow Chart กระบวนการทำงานของระบบ



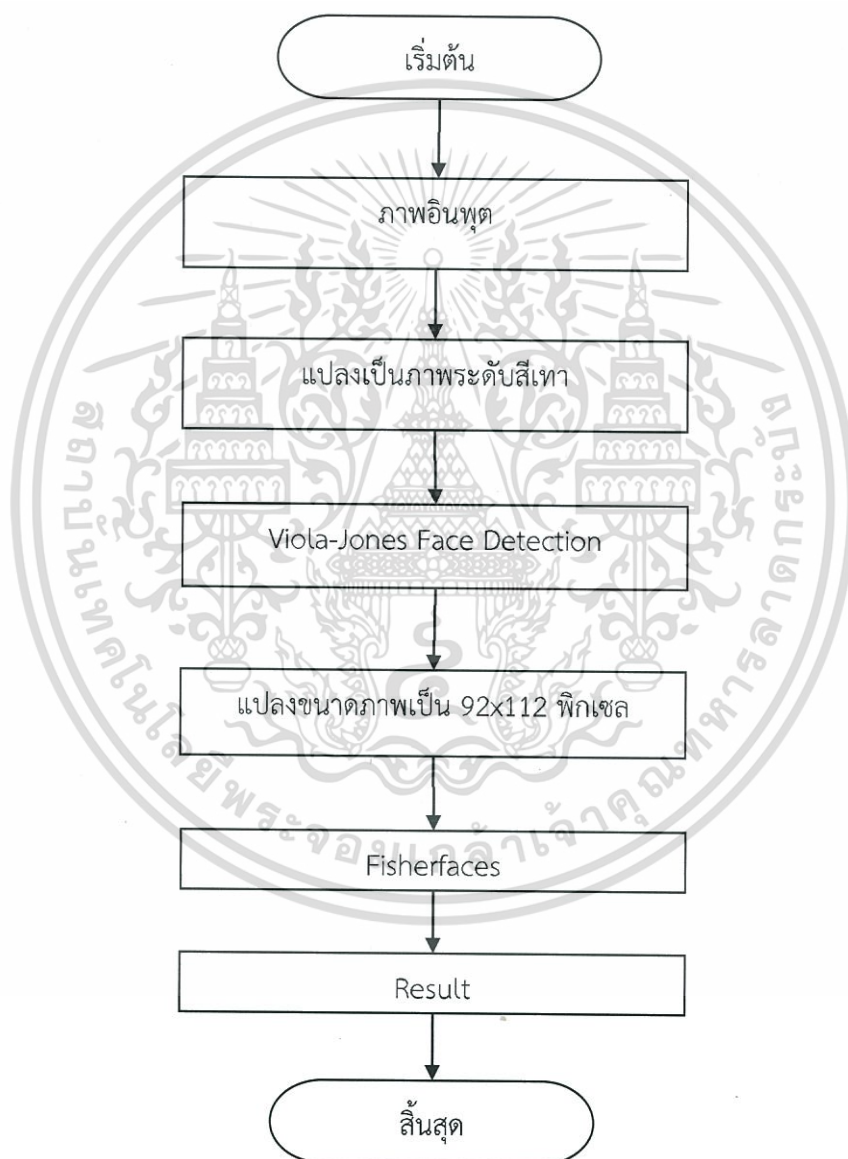
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดง Flow Chart ของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการทำงานของระบบรู้จำใบหน้าบุคคล ภายในหน่วยประมวลผลนั้นเริ่มจากภาพถ่ายอินพุตที่เข้ามา จากนั้นแปลงเป็นภาพระดับสีเทาเพื่อเข้าสู่กระบวนการ Face Detection โดยใช้วิธีการของ Viola-Jones เมื่อภาพอินพุตผ่านกระบวนการ Face Detection แล้วจะทำการย่อขนาดของภาพเหลือ 92x112 พิกเซล เพื่อจะเข้าสู่กระบวนการ Face Recognition โดยใช้หลักการ Fisherfaces และแสดงผลออกมาเป็นการระบุตัวตนของบุคคลนั้นต่อไป



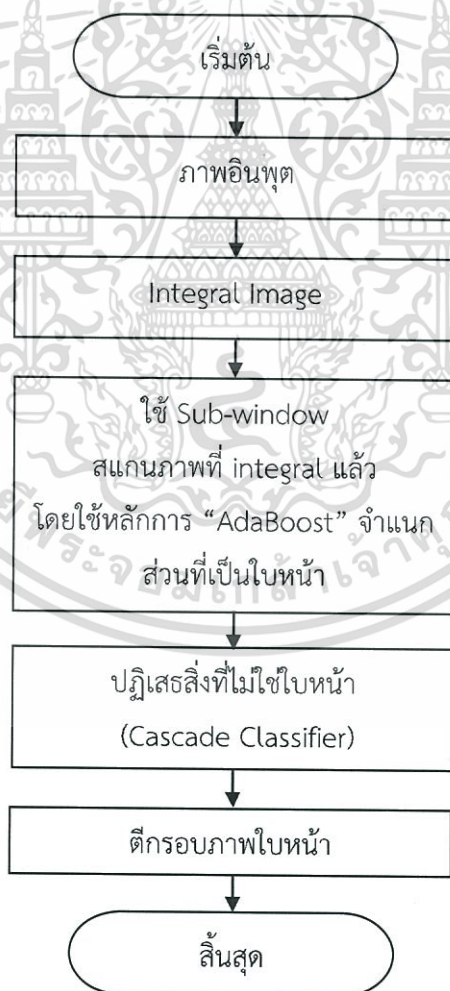
รูปที่ 3.3 Flow Chart การทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.1 ส่วนการตรวจจับภาพใบหน้าบุคคล (Face Detection)

การตรวจจับภาพใบหน้าบุคคลใช้ระเบียบวิธีการของ Viola-Jones คือ การสแกน Sub-window หรือการสแกนหน้าต่างเล็ก ๆ ภายในภาพนั้น เพื่อตรวจหาใบหน้าจากรูปภาพอินพุต โดยจะใช้วิธีการตรวจค้นหาหลาย ๆ รอบ (แต่ละรอบใช้ขนาดของหน้าต่างที่ต่างกัน) โดยการตรวจนั้นจะใช้ Feature ของ Haar Wavelets และ Integral Image โดยการตรวจหาในแต่ละรอบนั้นจะใช้หลักการ “ปฏิเสธสิ่งที่ไม่ใช่ใบหน้า แทนการค้นหาใบหน้า” เพราะการตัดสินใจในบริเวณใด ๆ นั้น ไม่ใช่ใบหน้าทำได้รวดเร็วกว่าการค้นหาใบหน้า

การตรวจจับภาพใบหน้าจะเริ่มต้นที่นำภาพที่ต้องการเข้าสู่ระบบ จากนั้นวิธีการของ Viola-Jones จะตรวจจับส่วนที่คาดว่าจะจะเป็นใบหน้า ระบบจะเลือกชิ้นส่วนที่ใหญ่ที่สุด เพื่อตัดสินใจเป็นภาพใบหน้า และเลือกเฉพาะภาพใบหน้า แสดงได้ดังรูปที่ 3.4

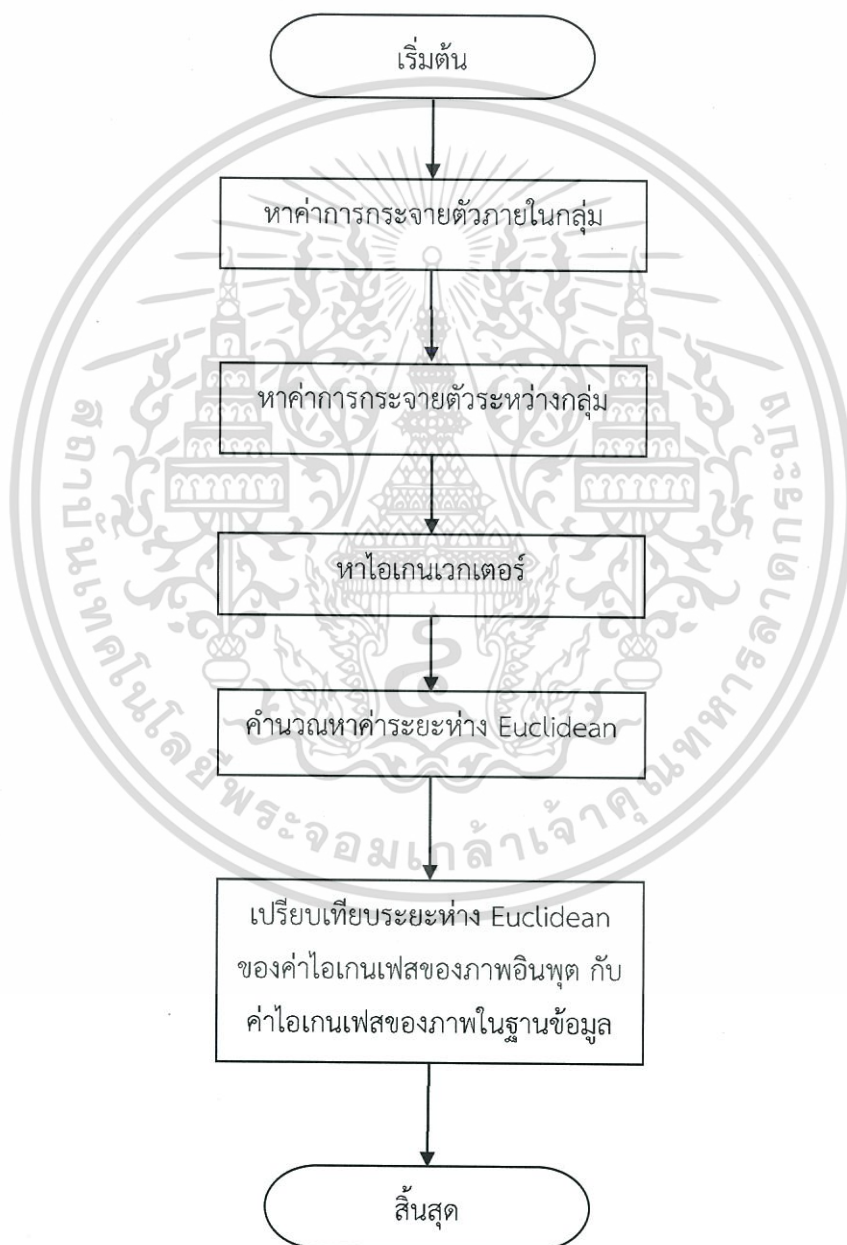


รูปที่ 3.4 Flow Chart การทำงานส่วนตรวจจับใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 ส่วนการรู้จำใบหน้าบุคคล (Face Recognition)

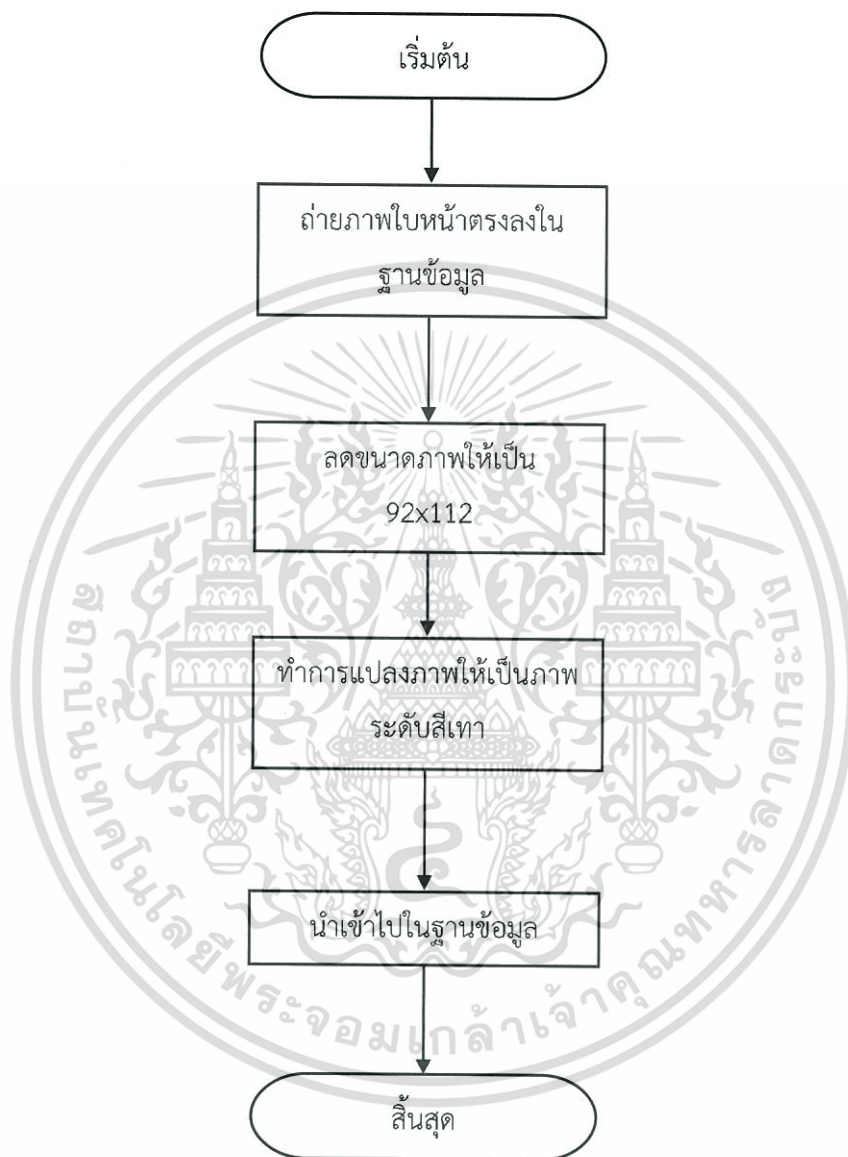
ภาพที่เข้ามาในกระบวนการรู้จำภาพใบหน้า ระบบจะคำนวณหาค่าไอเจนเวกเตอร์ เมื่อได้ค่าไอเจนเวกเตอร์แล้วจะนำไปคูณกับรูปภาพในฐานข้อมูล เพื่อให้ได้ค่าไอเจนเฟสออกมา จากนั้นระบบจะระบุตัวบุคคลออกมาจากค่าระยะห่าง Euclidean ที่น้อยที่สุด ซึ่งระยะห่าง Euclidean คือระยะห่างระหว่างค่าไอเจนเฟสของภาพอินพุตกับค่าไอเจนเฟสของภาพในฐานข้อมูล



รูปที่ 3.5 Flow Chart การทำงานส่วนรู้จำใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3 Flow Chart ของการเก็บข้อมูลในฐานข้อมูล

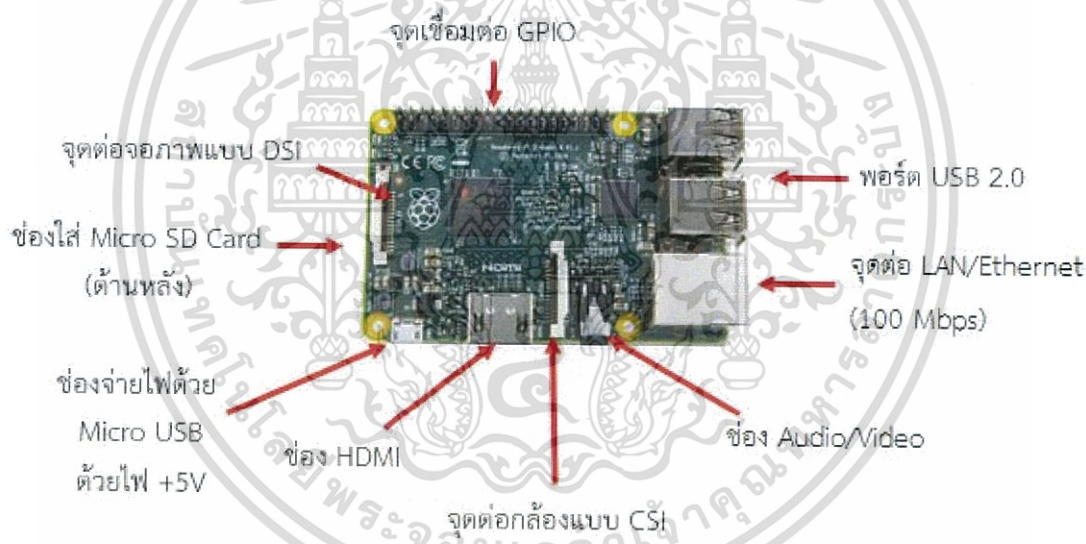


รูปที่ 3.6 Flow Chart ของการเก็บภาพในฐานข้อมูล

## 3.2 เครื่องมือที่ใช้ในการทดลอง

### 3.2.1 Raspberry Pi 2 Model B

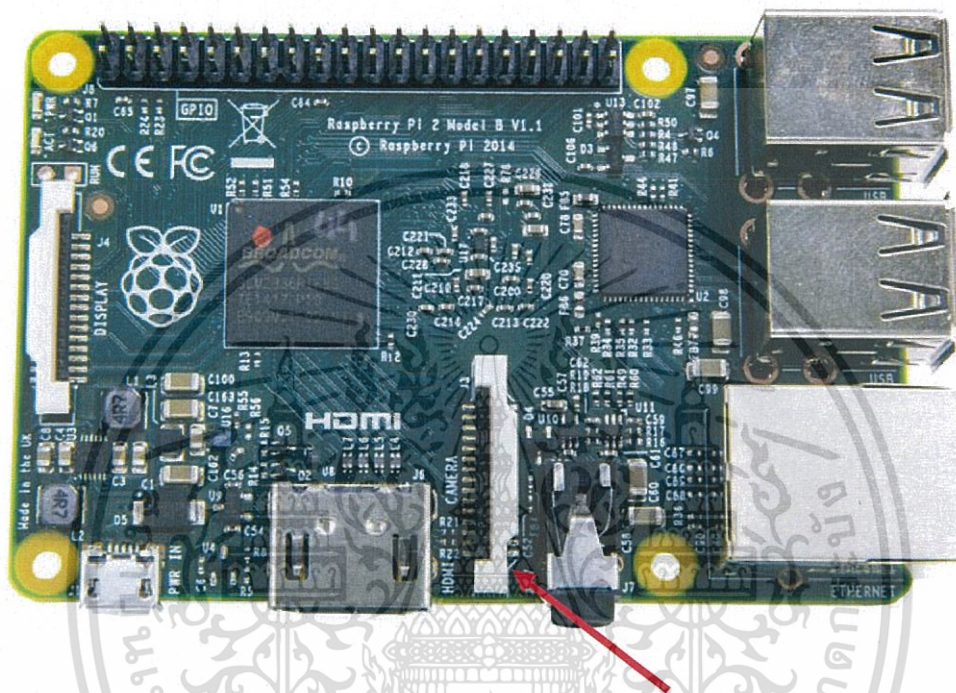
บอร์ด Raspberry Pi 2 Model B เป็นบอร์ดคอมพิวเตอร์ขนาดเล็ก (Single Board Computer: SBC) สามารถเชื่อมต่อกับหน้าจอคอมพิวเตอร์หรือโทรทัศน์ผ่านพอร์ต HDMI เชื่อมต่อกับ USB Mouse/Keyboard เพื่อใช้งานเป็นเครื่องคอมพิวเตอร์ขนาดเล็ก ใช้ทำงานเอกสาร Word Processing, Spreadsheet ใช้เชื่อมต่ออินเทอร์เน็ตด้วย Web Browser สามารถเล่นไฟล์มัลติมีเดียต่าง ๆ ทั้ง Audio และ Video รองรับระบบปฏิบัติการ Linux ต่าง ๆ เช่น Raspbian (พื้นฐานมาจาก Linux Debian), Snappy Ubuntu Core, OpenELEC, Raspbmc, Pidora (พื้นฐานมาจาก Linux Fedora), RISC OS เป็นต้น



รูปที่ 3.7 ส่วนประกอบของ Raspberry Pi 2

### 3.2.2 พอร์ต CSI (Camera Serial Interface)

เป็นพอร์ตสำหรับการเชื่อมต่อ Raspberry Pi Camera Module โดยเฉพาะ จะอยู่ในตำแหน่งที่ลูกศรสีแดงชี้ ดังรูป

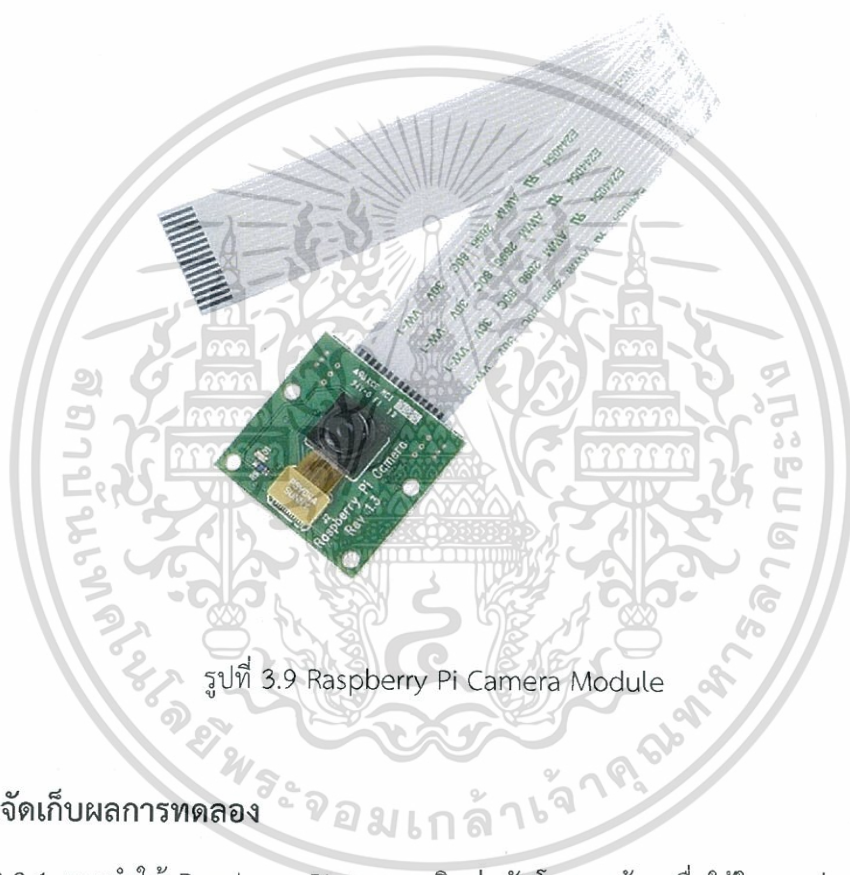


รูปที่ 3.8 ตำแหน่งของพอร์ต CSI (Camera Serial Interface)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3 Raspberry Pi Camera Module

Raspberry Pi Camera Module เป็นโมดูลกล้องสำหรับต่อใช้งานร่วมกับบอร์ด Raspberry Pi ขนาดความละเอียด 5 ล้านพิกเซล สามารถถ่ายวิดีโอระดับ HD ที่ความละเอียด 1080p, 720p และ 640x480 ด้วยอัตราแสดงผล 30 (1080p), 60 (720p และ 640x480) และ 90 (640x480) เฟรมต่อวินาที



รูปที่ 3.9 Raspberry Pi Camera Module

## 3.3 การจัดเก็บผลการทดลอง

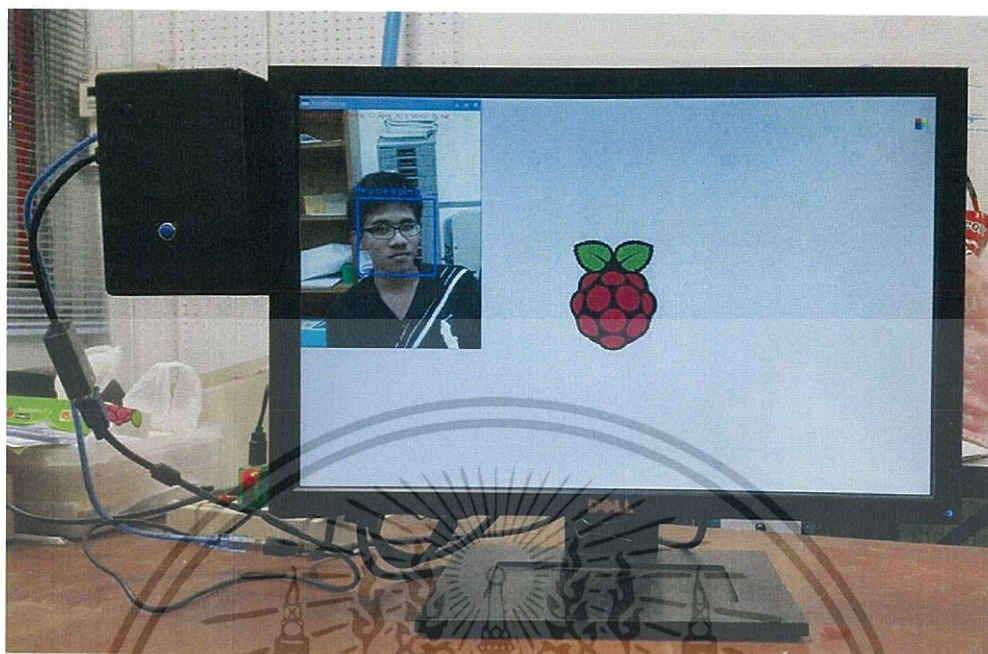
3.3.1 การทำให้ Raspberry Pi สามารถติดต่อกับโมดูลกล้องเพื่อใช้ในการถ่ายภาพ และประมวลผลเพื่อวิเคราะห์ใบหน้า

3.3.2 การเขียนชุดคำสั่งเพื่อนำภาพที่ได้จากกล้องไปใช้ในการประมวลผลเพื่อวิเคราะห์ใบหน้า

3.3.3 การทดลองส่วนตรวจจับใบหน้าบุคคล ว่าสามารถตรวจจับใบหน้าบุคคลในอริยาบถต่าง ๆ ได้มีความแม่นยำมากเพียงใด

3.3.4 การทดลองส่วนรู้จำภาพใบหน้าบุคคล ว่าสามารถระบุได้ว่าบุคคลในภาพคือใครได้อย่างแม่นยำหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงรูปของระบบเมื่อติดตั้งอุปกรณ์

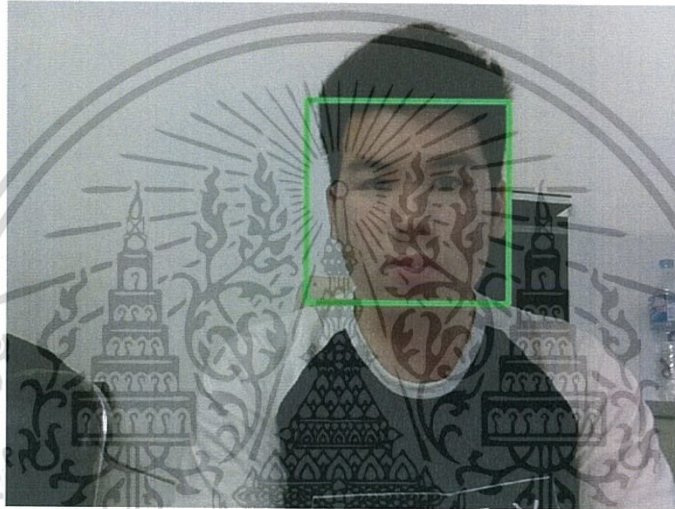
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

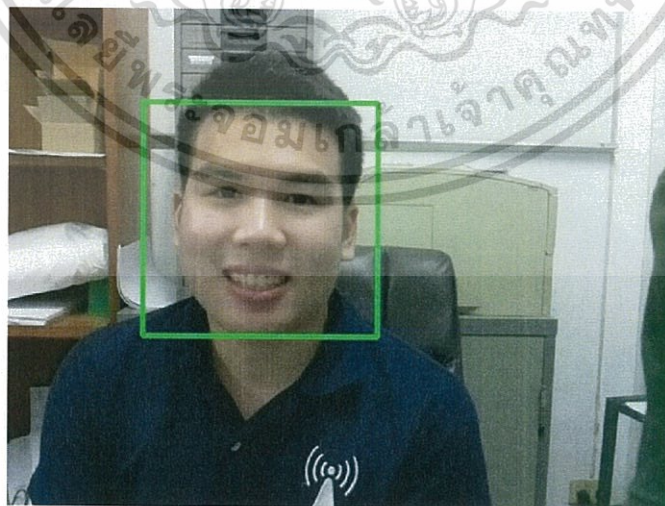
### ผลการทดลอง

#### 4.1 ผลการทดลองการตรวจจับใบหน้า

จากการทดลองจะเป็นการนำภาพที่ได้จากกล้องมาแสดงผลและตรวจสอบว่ามีใบหน้าอยู่ในภาพนั้นหรือไม่ เพื่อดูประสิทธิภาพของระบบว่ามีความสามารถตรวจจับใบหน้าได้ถูกต้องมากน้อยเพียงใด



รูปที่ 4.1 แสดงตัวอย่างการตรวจจับใบหน้าปกติ

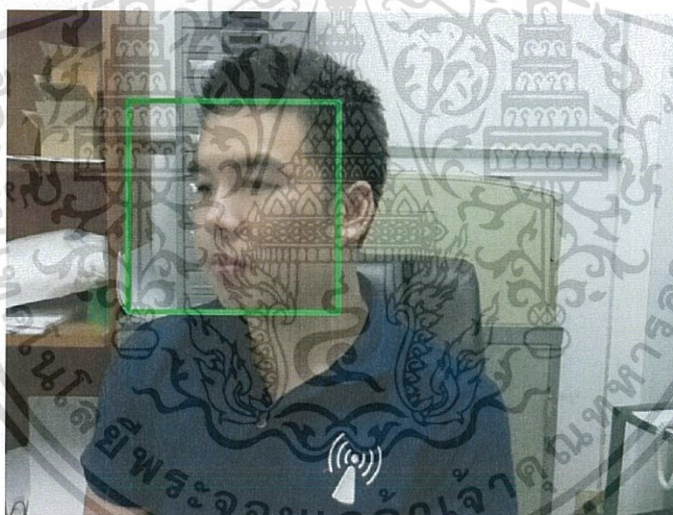


รูปที่ 4.2 แสดงตัวอย่างการตรวจจับใบหน้าขณะยิ้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงตัวอย่างการตรวจจับใบหน้าขณะหลับตา



รูปที่ 4.4 แสดงตัวอย่างการตรวจจับใบหน้าขณะเอียง 45 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการทดลองตรวจจับภาพใบหน้าในลักษณะต่าง ๆ

ลักษณะใบหน้า	จำนวนครั้ง ที่ทำการทดลอง	ตรวจจับถูกต้อง (ครั้ง)	เปอร์เซ็นต์ ความถูกต้อง
ใบหน้าปกติ	30	30	100%
ใบหน้าที่ยิ้ม	30	30	100%
ใบหน้าที่หลับตา	30	30	100%
ใบหน้าที่แลบลิ้น	30	30	100%
ใบหน้าที่เอียง 45 องศา	30	24	80%
ใบหน้าที่เอียง 90 องศา	30	0	0%

จากผลการทดลองในส่วนของการตรวจจับใบหน้า ระบบสามารถที่จะระบุได้ว่าส่วนไหนของภาพนั้นเป็นใบหน้าบุคคล ในด้านของความแม่นยำในส่วนของการตรวจจับใบหน้านั้น สามารถทำได้อย่างถูกต้องมีความแม่นยำ ดังตารางที่ 4.1 ซึ่งแสดงให้เห็นว่าเมื่อมีใบหน้าที่ปรากฏที่กล้อง ก็จะมีการตรวจจับใบหน้าบุคคลได้อย่างแม่นยำ

#### 4.2 ผลการทดลองการรู้จำใบหน้าบุคคล

การรู้จำใบหน้าบุคคลนั้นจะทดสอบต่อจากหัวข้อ 4.1 ที่ว่าเมื่อทำการระบุได้แล้วว่าส่วนใดของภาพเป็นใบหน้าบุคคล จะทำการระบุชื่อบุคคลนั้นด้วยว่า บุคคลที่ระบุได้แล้วในส่วนที่ 4.1 คือบุคคลใด โดยได้ทำการเก็บภาพของบุคคลไว้ก่อนแล้วในฐานข้อมูล แล้วเมื่อบุคคลนั้นเข้าไปอยู่ภายในภาพ ก็จะมีการแสดงชื่อของบุคคลนั้นขึ้นมา สำหรับในการทดลองจะทำการทดลองโดยใช้ภาพใบหน้าที่มีอริยาบถแตกต่างกัน และมีจำนวนภาพใบหน้าในฐานข้อมูล 10, 20, 30, 40 ภาพใบหน้าตามลำดับ เพื่อทดสอบว่าเมื่ออริยาบถเปลี่ยนไป และจำนวนภาพใบหน้าในฐานข้อมูลต่างกัน จะมีผลต่อความแม่นยำของระบบหรือไม่



รูปที่ 4.5 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลทั้งหมด 40 คน

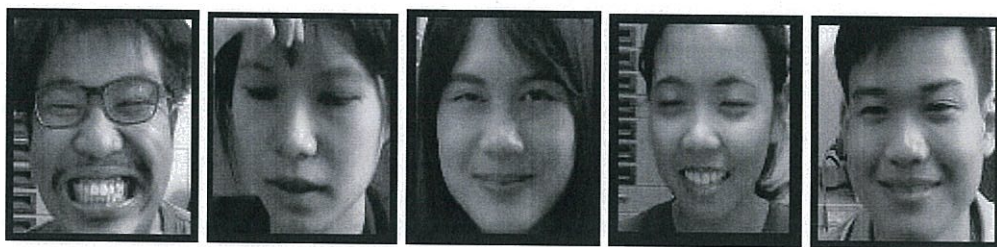


รูปที่ 4.6 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถปกติ



รูปที่ 4.7 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถขณะยิ้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถขณะหลับตา



รูปที่ 4.9 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถขณะแลบลิ้น



รูปที่ 4.10 แสดงตัวอย่างภาพบุคคลที่อยู่ในฐานข้อมูลในอิริยาบถขณะเอียงหน้า 45 องศา

#### 4.2.1 ภาพใบหน้าในฐานข้อมูล 10 ใบหน้า

ทดสอบความแม่นยำของระบบโดยใช้รูปภาพใบหน้าในฐานข้อมูล 10 รูป และมี 5 อิริยาบถ ได้แก่ ภาพใบหน้าตรงปกติ ภาพใบหน้าที่ยิ้ม ภาพใบหน้าหลับตา ภาพใบหน้าแลบลิ้น และ ภาพใบหน้าเอียง 45 องศา

ตารางที่ 4.2 ผลการทดลองในส่วนรู้จำใบหน้าในขณะที่มีภาพใบหน้าในฐานข้อมูล 10 ใบหน้า

ลักษณะใบหน้า	จำนวนครั้งที่ทำการทดลอง	ตรวจจับถูกต้อง (ครั้ง)	ตรวจจับผิดพลาด (ครั้ง)		เปอร์เซ็นต์ความถูกต้อง
			ไม่สามารถตรวจจับได้	ตรวจจับได้แต่ผิดบุคคล	
ใบหน้าตรงปกติ	20	17	3	0	85%
ใบหน้าที่้ม	20	15	4	1	75%
ใบหน้าหลับตา	20	15	4	1	75%
ใบหน้าแลบลิ้น	20	14	5	1	70%
ใบหน้าเอียง 45 องศา	20	8	10	2	40%

#### 4.2.2 ภาพใบหน้าในฐานข้อมูล 20 ใบหน้า

ทดสอบความแม่นยำของระบบโดยใช้รูปภาพใบหน้าในฐานข้อมูล 20 รูป และมี 5 อิริยาบถ ได้แก่ ภาพใบหน้าตรงปกติ ภาพใบหน้าที่้ม ภาพใบหน้าหลับตา ภาพใบหน้าแลบลิ้น และ ภาพใบหน้าเอียง 45 องศา

ตารางที่ 4.3 ผลการทดลองในส่วนรู้จำใบหน้าในขณะที่มีภาพใบหน้าในฐานข้อมูล 20 ใบหน้า

ลักษณะใบหน้า	จำนวนครั้งที่ทำการทดลอง	ตรวจจับถูกต้อง (ครั้ง)	ตรวจจับผิดพลาด (ครั้ง)		เปอร์เซ็นต์ความถูกต้อง
			ไม่สามารถตรวจจับได้	ตรวจจับได้แต่ผิดบุคคล	
ใบหน้าตรงปกติ	20	16	4	0	80%
ใบหน้าที่้ม	20	15	4	1	75%
ใบหน้าหลับตา	20	14	4	2	70%
ใบหน้าแลบลิ้น	20	14	5	1	70%
ใบหน้าเอียง 45 องศา	20	7	10	3	35%

#### 4.2.3 ภาพใบหน้าในฐานข้อมูล 30 ใบหน้า

ทดสอบความแม่นยำของระบบโดยใช้รูปภาพใบหน้าในฐานข้อมูล 30 รูป และมี 5 อิริยาบถ ได้แก่ ภาพใบหน้าตรงปกติ ภาพใบหน้าที่้ม ภาพใบหน้าหลับตา ภาพใบหน้าแลบลิ้น และ ภาพใบหน้าเอียง 45 องศา

ตารางที่ 4.4 ผลการทดลองในส่วนรู้จำใบหน้าในขณะที่มีภาพใบหน้าในฐานข้อมูล 30 ใบหน้า

ลักษณะใบหน้า	จำนวนครั้งที่ทำการทดลอง	ตรวจจับถูกต้อง (ครั้ง)	ตรวจจับผิดพลาด (ครั้ง)		เปอร์เซ็นต์ความถูกต้อง
			ไม่สามารถตรวจจับได้	ตรวจจับได้แต่ผิดบุคคล	
ใบหน้าตรงปกติ	20	15	4	1	75%
ใบหน้าที่้ม	20	14	5	1	70%
ใบหน้าหลับตา	20	14	4	2	70%
ใบหน้าแลบลิ้น	20	13	5	2	65%
ใบหน้าเอียง 45 องศา	20	7	10	3	35%

#### 4.2.4 ภาพใบหน้าในฐานข้อมูล 40 ใบหน้า

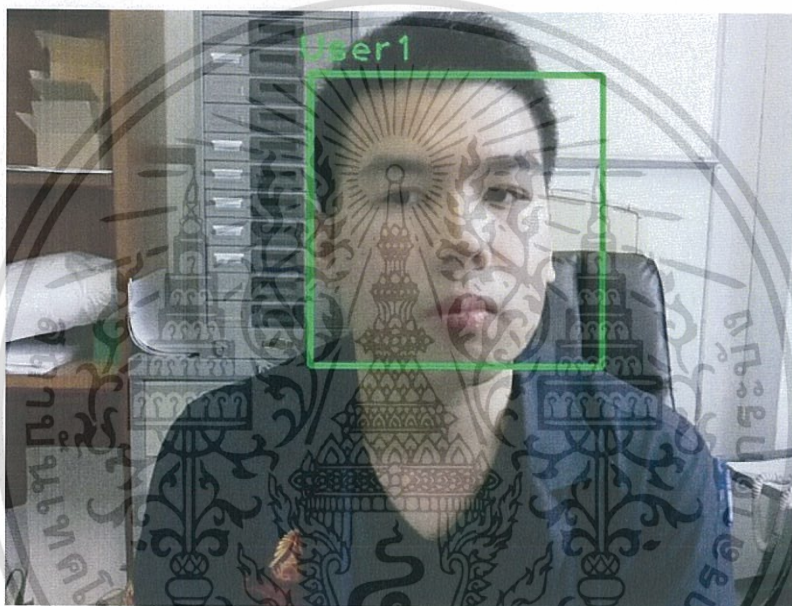
ทดสอบความแม่นยำของระบบโดยใช้รูปภาพใบหน้าในฐานข้อมูล 40 รูป และมี 5 อิริยาบถ ได้แก่ ภาพใบหน้าตรงปกติ ภาพใบหน้าที่้ม ภาพใบหน้าหลับตา ภาพใบหน้าแลบลิ้น และ ภาพใบหน้าเอียง 45 องศา

ตารางที่ 4.5 ผลการทดลองในส่วนรู้จำใบหน้าในขณะที่มีภาพใบหน้าในฐานข้อมูล 40 ใบหน้า

ลักษณะใบหน้า	จำนวนครั้งที่ทำการทดลอง	ตรวจจับถูกต้อง (ครั้ง)	ตรวจจับผิดพลาด (ครั้ง)		เปอร์เซ็นต์ความถูกต้อง
			ไม่สามารถตรวจจับได้	ตรวจจับได้แต่ผิดบุคคล	
ใบหน้าตรงปกติ	20	15	4	1	75%
ใบหน้าที่้ม	20	13	6	1	65%
ใบหน้าหลับตา	20	13	5	2	65%
ใบหน้าแลบลิ้น	20	13	5	2	65%
ใบหน้าเอียง 45 องศา	20	6	11	3	30%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบความแม่นยำของระบบในขณะที่มีจำนวนภาพในฐานข้อมูลที่แตกต่างกันนั้น เมื่อนำตารางมาเปรียบเทียบกันจะเห็นได้ว่า ยิ่งมีจำนวนภาพใบหน้าในฐานข้อมูลมาก จะทำให้เปอร์เซ็นต์ความแม่นยำของระบบลดลง เพราะใบหน้าของบุคคลแต่ละบุคคลในฐานข้อมูลนั้น อาจจะมีลักษณะคล้ายคลึง อีกทั้งยังอาจเป็นเพราะในขณะทดลอง ผู้ทำการทดลองอาจจัดวางหน้าไม่ตรงกับภาพใบหน้าที่มีอยู่ในฐานข้อมูล และสภาพแวดล้อมก็มีส่วนเกี่ยวข้อง เช่น แสง เป็นต้น จากที่กล่าวมาข้างต้นเป็นสาเหตุเบื้องต้นที่ทำให้ระบบอาจมีความผิดพลาดได้



รูปที่ 4.11 แสดงตัวอย่างการรู้จำใบหน้าที่ใบหน้าตรงกับฐานข้อมูล

จากรูปที่ 4.11 จะเป็นการแสดงการทดลองในส่วนของการรู้จำใบหน้า ซึ่งภาพใบหน้านี้เมื่อทำการรู้จำแล้ว จะได้ผลลัพธ์ที่คล้ายคลึงกับภาพใบหน้าในฐานข้อมูล ระบบจึงแสดงผลออกมาโดยการระบุชื่อของบุคคลนั้นดังรูป

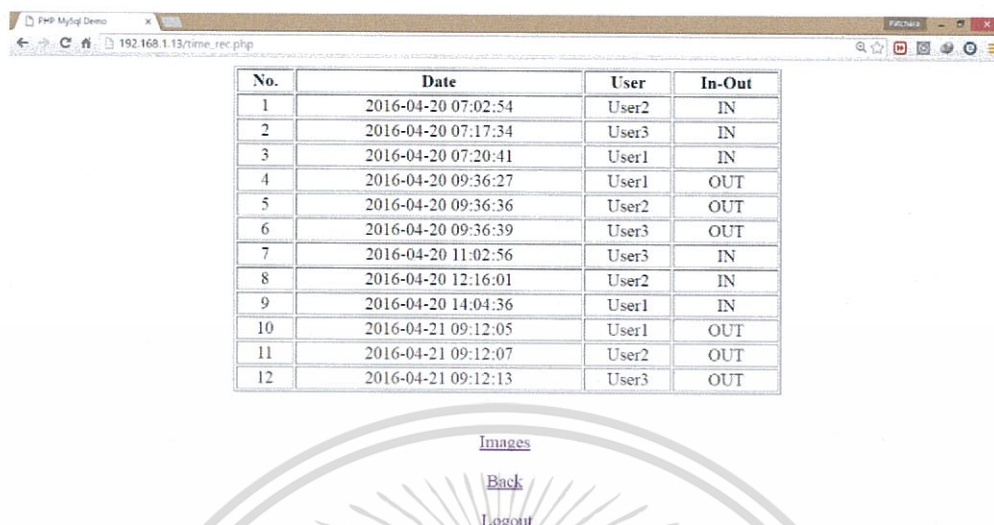


รูปที่ 4.12 แสดงตัวอย่างการรู้จำใบหน้าที่ใบหน้าไม่ตรงกับฐานข้อมูล

จากรูปที่ 4.12 เมื่อภาพที่นำมาทำกระบวนการรู้จำนั้น ไม่มีความคล้ายคลึงกับภาพใบหน้าในฐานข้อมูล จึงทำให้ผลลัพธ์ที่แสดงออกมานั้นแสดงว่า Unknown

#### 4.3 ผลการทดลองการอัปเดตสถานะและเวลาในฐานข้อมูล MySQL

จากการทดลอง เมื่อมีการสแกนใบหน้าแล้วสามารถระบุตัวตนบุคคลนั้นว่าเป็นบุคคลใดในฐานข้อมูล ก็จะมีการส่งค่ามาอัปเดตที่ฐานข้อมูล MySQL ที่เชื่อมต่อกับ Web Server ว่าให้อัปเดตสถานะของบุคคลว่าเข้ามาใช้งานระบบเมื่อวันและเวลาใด และเมื่อทำการสแกนหน้าแล้วไม่สามารถระบุได้ว่าเป็นบุคคลใดในฐานข้อมูล ก็จะแสดงข้อความ Unknown และทำการเซฟภาพใบหน้านั้น พร้อมทั้งระบุวันและเวลาที่ทำการสแกน เพื่อให้ผู้ใช้สามารถทราบได้ว่ามีบุคคลใดมาพยายามสแกนใบหน้าเพื่อจะเข้าที่פקอาศัยนั้น ๆ



No.	Date	User	In-Out
1	2016-04-20 07:02:54	User2	IN
2	2016-04-20 07:17:34	User3	IN
3	2016-04-20 07:20:41	User1	IN
4	2016-04-20 09:36:27	User1	OUT
5	2016-04-20 09:36:36	User2	OUT
6	2016-04-20 09:36:39	User3	OUT
7	2016-04-20 11:02:56	User3	IN
8	2016-04-20 12:16:01	User2	IN
9	2016-04-20 14:04:36	User1	IN
10	2016-04-21 09:12:05	User1	OUT
11	2016-04-21 09:12:07	User2	OUT
12	2016-04-21 09:12:13	User3	OUT

[Images](#)  
[Back](#)  
[Logout](#)

รูปที่ 4.13 แสดงตัวอย่างตารางบันทึกสถานะของผู้ใช้งานระบบ



รูปที่ 4.14 แสดงตัวอย่างภาพบุคคลที่ไม่ตรงกับฐานข้อมูลและระบบทำการบันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผล

ระบบรักษาความปลอดภัยแบบตรวจจับใบหน้าสามารถใช้งานได้บนบอร์ด Raspberry Pi โดยสามารถที่จะตรวจจับภาพได้ว่าในภาพนั้นมีใบหน้าของบุคคลอยู่หรือไม่ และเมื่อสามารถตรวจจับได้ว่ามีใบหน้าของบุคคลอยู่ก็จะทำการโซ่กรอบขึ้นมาครอบใบหน้านั้น ถ้าใบหน้าของบุคคลนั้นตรงกับใบหน้าที่มีอยู่ในฐานข้อมูล ก็จะแสดงชื่อของบุคคลนั้นขึ้นมา แล้วตัวของ Raspberry Pi ก็จะทำการส่งแรงดันไฟฟ้าผ่านพอร์ต GPIO ไปยังกลอนไฟฟ้าเพื่อทำให้กลอนไฟฟ้าปลดล็อก แต่ถ้าใบหน้าของบุคคลนั้นไม่ตรงกับฐานข้อมูลก็ จะไม่มีการปลดล็อกเกิดขึ้น

#### 5.2 ปัญหาและอุปสรรค

ปัญหาในการใช้งานโปรแกรม OpenCV นั้นผู้ใช้ต้องสังเกตว่า ระบบปฏิบัติการของ Raspberry Pi หรือ Raspbian รองรับ OpenCV รุ่นใดบ้าง เนื่องจาก OpenCV แต่ละรุ่นมีความแตกต่างกัน ผู้ใช้จึงควรศึกษาควรใช้รุ่นใดถึงจะเหมาะสมที่สุด

ปัญหาของระบบคือ การรู้จำใบหน้านั้นอาจจะมีการคลาดเคลื่อน ซึ่งขึ้นอยู่กับปัจจัยหลายอย่าง เช่น แสงอาจจะมีแสงน้อยเกินไปจึงทำให้ภาพไม่ชัดและอาจจะระบุผิดพลาดได้, อิริยาบถของบุคคลนั้นไม่ตรงกับที่มีอยู่ในฐานข้อมูลเลยทำให้คลาดเคลื่อน เป็นต้น

#### 5.3 ข้อเสนอแนะ

ในขณะที่ทำการตรวจจับใบหน้า บุคคลนั้นควรที่จะยืนหันหน้าตรงให้ใกล้เคียงกับภาพที่เก็บไว้ในฐานข้อมูลให้มากที่สุด และควรให้มีแสงสว่างที่มากพอ เพื่อเพิ่มความแม่นยำของการตรวจจับและรู้จำใบหน้ามากยิ่งขึ้น

## บรรณานุกรม

[1] วรณพงษ์ ภัททียไพบูลย์ (2558), “ทำระบบตรวจจับใบหน้าด้วย OpenCV กับภาษา Python”

สืบค้นจาก

<https://python3.wannaphong.com/2015/06/ตรวจจับใบหน้าด้วย-opencv-กับ-python.html>

[2] กิตติ ไพฑูรย์วัฒนกิจ (2549), การประมวลผลภาพดิจิทัล. กรุงเทพฯ, สถาบันเทคโนโลยีพระจอม

เกล้าเจ้าคุณทหารลาดกระบัง

[3] Mindphp Videoman (9 พฤษภาคม 2557), สอนเขียน php ติดต่อฐานข้อมูล Mysql อัพโหลด

ภาพ เพิ่ม ลบ แก้ไข Database [video file]

สืบค้นจาก <https://www.youtube.com/watch?v=5UFod7TgzSI>

[4] OpenCV dev team (2008), “Face Recognition with OpenCV”

สืบค้นจาก

[http://docs.opencv.org/2.4/Modules/contrib/doc/facerec/facerec\\_tutorial.html?highlight=fisherface#eigenfaces](http://docs.opencv.org/2.4/Modules/contrib/doc/facerec/facerec_tutorial.html?highlight=fisherface#eigenfaces)

[5] OpenCV dev team (2008), “Cascade Classification”

สืบค้นจาก

[http://docs.opencv.org/2.4/Modules/objdetect/doc/cascade\\_classification.html?highlight=cascade](http://docs.opencv.org/2.4/Modules/objdetect/doc/cascade_classification.html?highlight=cascade)

[6] Robert Castle (2014), “Computer Vision, How To, Raspberry Pi • Comments Off on Installing OpenCV on a Raspberry Pi”

สืบค้นจาก

<http://robertcastle.com/2014/02/installing-opencv-on-a-raspberry-pi/>

[7] LookHin (2013), “ขั้นตอนการติดตั้ง Apache, PHP, MySQL บน Raspberry Pi”

สืบค้นจาก

<https://www.unzeen.com/article/2173/>

[8] วิสิทธิ์ แผ้วกระโทก (2012), How to setup WebServer, DatabaseServer on Raspberry Pi

สืบค้นจาก

<http://raspberry-pi-th.blogspot.com/2012/10/how-to-setup-webserver-databaseserver.html>

[9] Eyal Arubas (2013), “Face Detection and Recognition (Theory and Practice)”

สืบค้นจาก

<http://eyalarubas.com/face-detection-and-recognition.html>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# Security System by Face Detector
# Copyleft (C) 2016 by Patchara, Pongsakon, Peeravit
# Telecommunications Engineering
# King Mongkut's Institute of Technology Ladkrabang

import cv2, numpy, os, time, sys
import RPi.GPIO as GPIO
import datetime
import MySQLdb
from picamera.array import PiRGBArray
from picamera import PiCamera
from multiprocessing import Process                                #.jpg @project
from pyimagesearch.tempimage import Tempimage                   #Face.jpg @project
from pyimagesearch.tempimage2 import Tempimage2                #Face.pgm @project
from pyimagesearch.tempimage3 import Tempimage3                #Unknown.jpg@/var/www
from pyimagesearch.tempimage4 import Tempimage4

# Initialize the Camera and grab a reference to the raw Camera
capture

camera = PiCamera()
camera.resolution = (480, 640)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size= (480, 640))

# Allow the Camera to warmup

time.sleep(0.1)

# Initial setup
size = 5

fn_haar = 'haarcascade_frontalface_default.xml'
fn_dir = 'att_faces_c'

a = 0
b = 0
c = 0
d = 0

# Create fisherRecognizer

print ('Training...')

# Create a list of images and a list of corresponding names

(images, labels, names, id) = ([], [], {}, 0)

for(subdirs, dirs, files) in os.walk(fn_dir):

    for subdir in dirs:

        names[id] = subdir

        subjectpath = os.path.join(fn_dir, subdir)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for filename in os.listdir(subjectpath):

    path = subjectpath + '/' + filename

    lable = id

    images.append(cv2.imread (path, 0))

    lables.append(int(lable))

    id += 1

# Face image size
(im_width, im_height) = (92, 112)

# Create a NumPy array from the two lists above
(images, lables) = [numpy.array(lis) for lis in [images, lables]]

# OpenCV trains a model from the images
model = cv2.createFisherFaceRecognizer()
model.train(images, lables)

# Use fisherRecognizer on Camera stream
haar_cascade = cv2.CascadeClassifier(fn_haar)

# Connect database
conn = MySQLdb.connect(host="localhost", # your host
                        user="root",      # your username
                        passwd="123456789", # your password
                        db="db_sample")    # name of the database

cur = conn.cursor()

# GPIO setup
GPIO.setmode(GPIO.BOARD) # Use board Pin numbering

# LED
GPIO.setup(12, GPIO.OUT) # Setup GPIO Pin 12 to OUT
GPIO.output(12,GPIO.LOW) # Turn off GPIO Pin 12

# Solenoid
GPIO.setup(11, GPIO.OUT) # Setup GPIO Pin 11 to OUT
GPIO.output(11,GPIO.LOW) # Turn off GPIO Pin 11

# Capture and save face
GPIO.setup(13, GPIO.IN, pull_up_down = GPIO.PUD_UP) # Setup GPIO Pin
13 to IN

# Log out User (1-3)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GPIO.setup(31, GPIO.IN, pull_up_down = GPIO.PUD_UP) # Setup GPIO Pin
31 to IN

GPIO.setup(33, GPIO.IN, pull_up_down = GPIO.PUD_UP) # Setup GPIO Pin
33 to IN

GPIO.setup(35, GPIO.IN, pull_up_down = GPIO.PUD_UP) # Setup GPIO Pin
35 to IN

# Sound setup

def correct():

    cmd_string = 'mpg321 -q sound01.mp3'
    os.system(cmd_string)

    return

def again():

    cmd_string = 'mpg321 -q sound02.mp3'
    os.system(cmd_string)

    return

def whol():

    cmd_string = 'mpg321 -q sound03.mp3'
    os.system(cmd_string)

    return

while True:

    # Capture frames from the Camera

    for frame in camera.capture_continuous(rawCapture, format="bgr",
use_video_port=True):

        # Grab the raw NumPy array representing the image, then
initialize the timestamp
        image = frame.array
        timestamp = datetime.datetime.now()
        text = "None"

        # Draw the text and timestamp on the frame

        ts = timestamp.strftime("%A %d %B %Y %I:%M:%S %p")

        # Draw the text and timestamp on the face image
        ts2 = timestamp.strftime("%d %B %Y %I:%M:%S %p")

        cv2.putText (image, "Date Time: {}".format(ts), (10, 20),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1)

```

```

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
mini = cv2.resize(gray, (gray.shape[1]/size,gray.shape[0]/size))

faces = haar_cascade.detectMultiScale(mini)

# Try to detect the face

for i in range(len (faces)):
    face_i = faces[i]
    (x, y, w, h) = [v * size for v in face_i]
    face = gray[y:y + h, x:x + w]
    face_resize = cv2.resize (face, (im_height, im_width))
    face_resize2 = face_resize.copy()
    cv2.putText(face_resize2, "{}".format(ts2), (10, 20),
cv2.FONT_HERSHEY_SIMPLEX, 0.3, (255, 255, 255), 1)
    cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0),
3)
    text = "Face"

    # Try to recognize the face
    prediction = model.predict(face_resize)
    accuracy = 100*((1500-prediction[1]) /1500)
    a+=1

    if a>=25:
        b+=1
        # Correct
        if prediction[1]<500:
            c+=1
            cv2.rectangle(image, (x, y), (x + w, y + h),
(0, 255, 0), 3)
            cv2.putText(image,
                '%s - %.0f%s' %
                (names[prediction[0]],accuracy,"%"),
                (x-10,y-10), cv2.FONT_HERSHEY_PLAIN,2,
(0, 255, 0),2)

            if c == 1:

                p = Process(target=correct)

                p.start()

                with conn :

                    cur = conn.cursor()

                    cur.execute("UPDATE tbl_name SET
AvaliableStatus = %s, Lastupdate = NOW() WHERE ID = %s",
("Avaliable",names[prediction[0]]))

                    print "Number of row update
:",cur.rowcount

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Time_Record")
cur.execute("SELECT * FROM
Time_Record(Date,User,InorOut) VALUES (NOW(),%s,'IN')
",names[prediction[0]])

for row in cur.fetchall():
    print row[0]
print "Number of row update2
:",cur.rowcount

conn.commit()
cur.close()

GPIO.output(11,GPIO.HIGH) # Turn on
GPIO.output(12,GPIO.HIGH) # Turn on

# Incorrect
else :
    c+=1
    cv2.rectangle(image, (x, y), (x + w, y + h),
(0, 0, 255), 3)
    cv2.putText(image,
'Unknown',
(x-10,y-10), cv2.FONT_HERSHEY_PLAIN,2,
(0, 0, 255),2)

    if c == 1:
        d+=1
        if d < 4:
            p = Process(target=again)
            p.start()
        else:
            d=0
            p = Process(target=whol)
            p.start()

    t4 = Tempimage4()
    cv2.imwrite(t4.path, face_resize2)

if b>=15 :
    a=0
    b=0
    c=0
    GPIO.output(11,GPIO.LOW) # Turn off GPIO Pin
    GPIO.output(12,GPIO.LOW) # Turn off GPIO Pin

# Show the frame
cv2.imshow('FaceDetector', image)

# Save image

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if GPIO.input(13) == True: # Press button GPIO Pin 13

    if text == "Face":

        # Face.pgm
        t = Tempimage3()
        cv2.imwrite(t.path, face_resize)

        # Face.jpg
        t2 = Tempimage2()
        cv2.imwrite(t2.path, face_resize2)

        # FaceRecog.jpg
        t3 = Tempimage()
        cv2.imwrite(t3.path, image)

# Log out User1
if GPIO.input(31) == True: # Press button GPIO Pin 31
    with conn :

        cur = conn.cursor()

        cur.execute("UPDATE tbl_name SET Status = %s,
Lastupdate = NOW() WHERE ID = %s", ("unavailable","User1"))
        print "Number of row update :",cur.rowcount

        conn.commit()
        cur.close()

# Log out User2
if GPIO.input(33) == True: # Press button GPIO Pin 33
    with conn :

        cur = conn.cursor()

        cur.execute("UPDATE tbl_name SET Status = %s,
Lastupdate = NOW() WHERE ID = %s", ("unavailable","User2"))
        print "Number of row update :",cur.rowcount

        conn.commit()
        cur.close()

# Log out User3
if GPIO.input(35) == True: # Press button GPIO Pin 35
    with conn :

        cur = conn.cursor()

        cur.execute("UPDATE tbl_name SET Status = %s,
Lastupdate = NOW() WHERE ID = %s", ("unavailable","User3"))

        print "Number of row update :",cur.rowcount

        conn.commit()
        cur.close()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# Clear the stream in preparation for the next frame  
rawCapture.truncate(0)
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้