

การตรวจจับใบหน้าสำหรับระบบรักษาความปลอดภัยโดยใช้  
OpenCV  
FACE DETECTION FOR SECURITY SYSTEM USING OpenCV



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมระบบควบคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2561

การตรวจจับใบหน้าสำหรับระบบรักษาความปลอดภัยโดยใช้

OpenCV

FACE DETECTION FOR SECURITY SYSTEM USING OpenCV



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# FACE DETECTION FOR SECURITY SYSTEM USING OpenCV



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING  
FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การตรวจจับใบหน้าสำหรับระบบรักษาความปลอดภัยโดยใช้ OpenCV  
FACE DETECTION FOR SECURITY SYSTEM USING OpenCV

ผู้จัดทำ นายเชาวนพันธ์ ศุภวิริยกุล 58010302

นายนิรันดร์ แก้วมณี 58010681



.....อาจารย์ที่ปรึกษา  
(ศาสตราจารย์ ดร.วรวงศ์ ตั้งศรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การตรวจจับใบหน้าสำหรับระบบรักษาความปลอดภัยโดยใช้ OpenCV

โดย

นายเชวณพนันท์ ศุภวริยกุล 58010302

นายนิรันดร์ แก้วมณี 58010681

อาจารย์ที่ปรึกษา

ศาสตราจารย์ ดร.วรพงศ์ ตั้งศรีรัตน์

ปีการศึกษา 2561

บทคัดย่อ

ความปลอดภัยเป็นสิ่งที่มีความสำคัญมากสำหรับสภาพความเป็นอยู่ของมนุษย์ ในช่วงหลายปีที่ผ่านมามนุษย์จึงได้มีการคิดค้นระบบรักษาความปลอดภัยหลากหลายรูปแบบขึ้นมา แต่รูปแบบก็ขึ้นอยู่กับเหตุปัจจัย และสภาพแวดล้อมในทีที่ต้องการความปลอดภัย อุปกรณ์ชนิดหนึ่งในปัจจุบันนิยมใช้กันก็คือ “กล้องวงจรปิด” เหตุผลที่นิยมใช้เพราะตัวกล้องมีขนาดเล็ก สามารถนำไปติดตั้งตามที่ต่างๆ ได้ไม่ยาก และทำให้เห็นพื้นที่ต่างๆ ได้อย่างครอบคลุม ในโครงงานนี้ ผู้จัดทำใช้ประโยชน์จากเทคโนโลยีการประมวลผลภาพโดยใช้กล้องขนาดเล็กเพื่อตรวจจับ และระบุใบหน้าบุคคล จากการเปรียบเทียบใบหน้าที่ได้รับจากกล้องกับฐานข้อมูลใบหน้าที่มีอยู่ เมื่อเปรียบเทียบเสร็จก็จะแสดงผลด้วยการแจ้งเตือนผ่านแอปพลิเคชัน LINE และ Facebook

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# FACE DETECTION FOR SECURITY SYSTEM USING OpenCV

By

Mr. Chaowaphun Supawiriyakul 58010302

Mr. Nirun Keawmanee 58010681

Advisor

Prof.Dr. Worapong Tangsirat

Academic Year 2018

## ABSTRACT

Safety is very important for human living conditions. In recent years, human beings have developed a variety of security systems. Each format depends on the factors and the environment in which the safety condition is needed. One device that is currently used is "CCTV". The reason for the use of the camera is its small size and can be installed easily in different places, as well as can be seen in various areas. In this project, we utilize the image processing technology using small cameras to detect and identify the human faces from comparing the detected faces by the camera with the existing face database. The compared results will be alerted and displayed via the application LINE and Facebook.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ประสบความสำเร็จไปได้ด้วยดี ทั้งนี้เนื่องจากคำแนะนำของอาจารย์ที่ปรึกษา ศาสตราจารย์ ดร.วรวงศ์ ตั้งศรีรัตน์ อาจารย์ที่ปรึกษา และอาจารย์ท่านอื่นๆ ในสาขาวิศวกรรมระบบควบคุม ภาควิชาวิศวกรรมการวัดและควบคุม ผู้จัดทำขอขอบพระคุณในความอนุเคราะห์จาก อาจารย์ทุกท่านที่กรุณาสละเวลา ให้ความรู้และคำแนะนำตลอดการทำโครงการ ขอขอบพระคุณ สาขาวิศวกรรมระบบควบคุม ภาควิชาวิศวกรรมการวัดและควบคุม ที่เอื้อเฟื้อสถานที่ วัสดุ อุปกรณ์ต่าง ๆ สำหรับทำโครงการ ขอขอบพระคุณ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้เงินทุนสำหรับสนับสนุนในการทำโครงการนี้ และขอขอบคุณ เพื่อน ๆ พี่ ๆ และ น้อง ๆ ทุกคนที่ช่วยเหลือและให้คำแนะนำในการทำโครงการ

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และถ่ายทอดประสบการณ์ให้แก่ข้าพเจ้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VIII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	2
1.3 ขอบเขตของการดำเนินการของปริญญานิพนธ์	2
1.4 ขั้นตอนการดำเนินงานของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	3
2.1 ภาพดิจิทัล (Digital Image)	3
2.2 ภาพแบบบิตแมป (Bitmap Image)	4
2.3 ภาพแบบเวกเตอร์ (Vector Image)	4
2.4 ประเภทของภาพบิตแมป	4
2.5 ภาพขาวดำ (Binary Image)	5
2.6 ภาพระดับสีเทา (Gray Scale Image)	5
2.7 ภาพสี (Color Image)	6
2.8 ภาพแบบดัชนี (Index Image)	6
2.9 ความรู้เบื้องต้นเกี่ยวกับการประมวลผลภาพดิจิทัล	7
2.9.1 Lossless Algorithm	7
2.9.2 Lossy Algorithm	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
2.10 การตรวจจับใบหน้า (Face Detection)	8
2.10.1 Knowledge – Based Method	8
2.10.2 Feature Invariant Approaches	8
2.10.3 Template Matching Methods	9
2.10.4 Appearance – Based Methods	9
2.11 วิธีการตรวจจับใบหน้าของ Viola – Jones	9
2.11.1 ขั้นตอน Integral Image	10
2.11.2 ขั้นตอนการเรียนรู้ด้วยวิธี Adaboost	12
2.11.3 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)	14
2.12 การรู้จำใบหน้า (Face Recognition)	15
2.12.1 วิธีเชิงความรู้ (Knowledge - Based Methods)	15
2.12.2 วิธีเชิงลักษณะ (Feature – Based Methods)	15
2.12.3 วิธีเทียบเคียงแผ่นแบบ (Template Matching Methods)	16
2.12.4 วิธีเชิงลักษณะปรากฏ (Appearance - Based Methods)	17
2.13 OpenCV	19
2.14 Raspberry Pi	19
2.15 ภาษา Python	20
2.16 Facebook Graph API	21
2.17 LINE Notify	21
<b>บทที่ 3 วิธีการดำเนินงาน</b>	<b>22</b>
3.1 ขั้นตอนการดำเนินงาน	22
3.2 อุปกรณ์ที่เกี่ยวข้อง	22
3.2.1 Raspberry Pi 3 Model B+	22
3.2.2 กล้อง Webcam	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

	หน้า
3.3 ซอฟต์แวร์ที่เกี่ยวข้อง	23
3.3.1 OpenCV	23
3.3.2 Facebook Graph API	23
3.3.3 LINE Notify	24
3.4 วิธีการดำเนินงาน	24
3.4.1 ทดลองการเปิดกล้อง Webcam โดยใช้โปรแกรม IDLEX (python GUI)	24
3.4.2 การสร้างข้อมูลรูปภาพด้วยฟังก์ชัน face_cascade	25
3.4.3 เชื่อมต่อกล้อง Webcam กับบอร์ด Raspberry Pi 3 Model B+	26
3.4.4 การเขียนโปรแกรม Python	27
3.4.5 ทดลองให้กล้องทำการตรวจจับใบหน้าและดวงตา	29
<b>บทที่ 4 ผลการทดลอง</b>	<b>31</b>
4.1 แผนผังการทำงานของระบบตรวจจับใบหน้า	31
4.2 การทดลองการรู้จำใบหน้า	31
4.2.1 ผู้ใช้งานปกติ	32
4.2.2 ผู้ใช้งานสวมแว่น	32
4.2.3 ผู้ใช้งานสวมหมวก	33
4.2.4 บุคคลภายนอก	33
4.3 การทดลองส่งข้อมูลแจ้งเตือนผลทาง Facebook และ LINE	35
<b>บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ</b>	<b>36</b>
5.1 สรุปผลการทดลอง	36
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	36
5.3 แนวทางการแก้ไข	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
เอกสารอ้างอิง	38
ภาคผนวก	39
ภาคผนวก ก โปรแกรม	40
ภาคผนวก ข Raspberry Pi 3 Model B+	45
ภาคผนวก ค Poster	50



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป

รูปที่	หน้า
2.1 ภาพดิจิทัล และภาพบิตแมป	3
2.2 ภาพขาวดำ	5
2.3 ภาพระดับสีเทา	5
2.4 ภาพสี	6
2.5 ภาพแบบดัชนี	6
2.6 การแปลงภาพอนาล็อกให้เป็นภาพดิจิทัล	7
2.7 แบบจำลอง Haar – Like	10
2.8 การคำนวณความเข้มในพื้นที่ D โดยวิธีการทั่วไป	11
2.9 การคำนวณผลรวมความเข้มในพื้นที่ D โดยวิธี Integral Image	11
2.10 ตัวอย่างภาพ Integral Image	12
2.11 ตัวอย่างของ Strong Classifier จากกระบวนการเรียนรู้ด้วย Adaboost	14
2.12 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)	14
2.13 ขอบเขตการพิจารณาของแต่ละลำดับชั้น	15
2.14 แบบจำลองใบหน้ามนุษย์จากฐานข้อมูล MPEG-4	17
2.15 การกระจายตัวของข้อมูลตัวอย่าง	18
2.16 ตัวอย่างโครงข่ายประสาทเทียม	19
2.17 OpenCV	19
2.18 Raspberry Pi	20
2.19 หลักการทำงานของ Graph API	21
2.20 LINE Notify	21
3.1 Raspberry Pi 3 Model B+	22
3.2 กล้อง Webcam	23
3.3 OpenCV	23
3.4 Facebook Graph API	24
3.5 LINE Notify	24
3.6 ฟังก์ชัน show_webcam	24
3.7 ตัวอย่างตอนเปิดกล้อง	25
3.8 การสร้างข้อมูลรูปภาพด้วยฟังก์ชัน face_cascade	25
3.9 ตัวอย่างตอนเปิดกล้อง (2)	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.10 ภาพหลังจากทำการถ่ายภาพออกมา 30 ภาพ	26
3.11 การเชื่อมต่อกล้อง Webcam กับบอร์ด Raspberry Pi 3 Model B+	26
3.12 โปรแกรมในส่วนของการสร้างฐานข้อมูล	27
3.13 โปรแกรม Training	27
3.14 โปรแกรมในส่วนเปิดการทำงานกล้อง Webcam แบบ Real – Time (1)	28
3.15 โปรแกรมในส่วนเปิดการทำงานกล้อง Webcam แบบ Real – Time (2)	28
3.16 โปรแกรมในส่วนเปิดการทำงานกล้อง Webcam แบบ Real – Time (3)	28
3.17 โปรแกรมการแจ้งเตือนผ่านแอปพลิเคชัน LINE	29
3.18 การแจ้งเตือนผ่านแอปพลิเคชัน Facebook	29
3.19 การทดลองการตรวจจับใบหน้า	29
3.20 การทดลองการตรวจจับใบหน้า (2)	30
4.1 แผนผังการทำงานของระบบตรวจจับใบหน้า	31
4.2 การติดตั้งอุปกรณ์	31
4.3 การทดลองโดยผู้ใช้ใช้งานปกติ	32
4.4 การทดลองโดยผู้ใช้ใช้งานสวมแว่น	32
4.5 การทดลองโดยผู้ใช้ใช้งานสวมหมวก	33
4.6 การทดลองโดยใช้นุคคลภายนอก	33
4.7 การแจ้งเตือนผ่าน Facebook	35
4.8 การแจ้งเตือนผ่าน LINE	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
4.1 ผู้ใช้งานในฐานข้อมูล	34
4.2 บุคคลภายนอกฐานข้อมูล	34



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในปัจจุบันความปลอดภัยมีความสำคัญมากต่อสภาพความเป็นอยู่ของมนุษย์ มนุษย์จึงได้มีการคิดค้นระบบรักษาความปลอดภัยหลากหลายรูปแบบขึ้นมา แต่ละรูปแบบก็ขึ้นอยู่กับเหตุปัจจัยสภาพแวดล้อมในที่ต้องการความปลอดภัย เช่น ภายในตัวอาคาร ธนาคาร ร้านค้า พื้นที่ส่วนบุคคล เป็นต้น ซึ่งในแต่ละรูปแบบนั้นก็จะมีขั้นตอน, อุปกรณ์, และทรัพยากรที่ใช้แตกต่างกันไป อุปกรณ์ชนิดหนึ่งที่ระบบรักษาความปลอดภัยส่วนใหญ่ในปัจจุบันนิยมใช้กันก็คือ “กล้องวงจรปิด” เหตุผลที่ปัจจุบันนิยมใช้กล้องวงจรปิดนั้น ก็เพราะตัวกล้องมีขนาดเล็ก สามารถนำไปติดตั้งตามมุมตึกต่างๆ ได้ไม่ยาก เมื่อติดตั้งแล้วก็จะทำให้เห็นภาพในมุมต่างๆ ของสถานที่ และสามารถทำให้เห็นพื้นที่ต่างๆ ได้อย่างครอบคลุม เพียงแค่นั่งรับชมผ่านทางมอนิเตอร์ ถ้าหากเกิดเหตุขึ้นก็จะทำให้เห็นภาพเหตุการณ์ได้อย่างทันท่วงที และภาพเหตุการณ์นั้นที่ถูกบันทึกได้ผ่านกล้องวงจรปิดก็จะถือว่าเป็นหลักฐานสำคัญในการดำเนินคดีต่อไปได้ ซึ่งในปัจจุบันนั้นเทคโนโลยีของกล้องวงจรปิดและเทคโนโลยีทางด้าน Image Processing มีการพัฒนาขึ้นมากกว่าจะเป็นแค่กล้องธรรมดาทั่วไป มีการนำเทคโนโลยีทางด้าน Image Processing มาใช้เพื่อช่วยในงานต่างๆ เช่น ตรวจสอบความเร็วรถยนต์บนท้องถนน ตรวจสอบการเคลื่อนที่ของมนุษย์ วิเคราะห์ใบหน้าของมนุษย์ เป็นต้น ดังนั้นจึงทำให้เกิดแนวคิดที่จะใช้เทคโนโลยีในการประมวลผลภาพในส่วนของการตรวจจับดวงตา (Eye Detection) การรู้จำดวงตา (Eye Recognition) การตรวจจับใบหน้า (Face Detection) และการรู้จำใบหน้า (Face Recognition) มาแก้ปัญหาเรื่องเหล่านี้ โดยทั้ง 4 ฟังก์ชันที่กล่าวมาเป็นวิธีที่ใช้ในการพิสูจน์ยืนยันตัวตนบุคคล ด้วยการใช้คุณลักษณะจำเพาะทางสรีระ (Biometric) ซึ่งจะทำงานโดยการเปรียบเทียบใบหน้าจากภาพถ่ายดิจิทัลหรือภาพจากกล้องวิดีโอกับฐานข้อมูลใบหน้าที่มีอยู่ และแสดงผลออกมาผ่านแอปพลิเคชัน

จากเหตุผลที่กล่าวมาข้างต้น ปริญญานิพนธ์ฉบับนี้จึงนำเสนอการออกแบบระบบรักษาความปลอดภัยด้วยการนำกล้องเว็บแคมมาต่อกับบอร์ด Raspberry pi 3 model B+ และใช้ OpenCV ซึ่งเป็น library ในภาษา Python ที่มีฟังก์ชันสำหรับการตรวจจับดวงตา (Eye Detection) การรู้จำดวงตา (Eye Recognition) การตรวจจับใบหน้า (Face Detection) และการรู้จำใบหน้า (Face Recognition) เพื่อวิเคราะห์ภาพจากกล้องเว็บแคม แล้วจากนั้นแสดงผลผ่านทางแอปพลิเคชัน LINE และ Facebook

## 1.2 วัตถุประสงค์ของปฏิญานิพนธ์

1. ออกแบบระบบรักษาความปลอดภัยในอาคารบ้านเรือน
2. พัฒนาระบบรู้จำใบหน้าและดวงตาสำหรับงานด้านรักษาความปลอดภัย
3. ลดต้นทุนในการซื้ออุปกรณ์รักษาความปลอดภัยที่มีราคาสูง

## 1.3 ขอบเขตของการดำเนินการของปฏิญานิพนธ์

1. ศึกษากระบวนการเรียนรู้วิเคราะห์ระบบตรวจจับใบหน้า และระบบรู้จำใบหน้า
2. ศึกษาหลักการทำงานของ Raspberry Pi
3. การออกแบบระบบรักษาความปลอดภัย โดยการเพิ่มฟังก์ชันการวิเคราะห์ใบหน้าให้กับกล้อง Webcam

## 1.4 ขั้นตอนการดำเนินงานของปฏิญานิพนธ์

1. ศึกษาค้นคว้าสิ่งที่น่าสนใจเกี่ยวกับการกำหนดหัวข้อโครงการ
2. นำเสนอหัวข้อโครงการ
3. ศึกษาและรวบรวมข้อมูลเกี่ยวกับการเลือกใช้อัลกอริทึม
4. ศึกษาการใช้ฟังก์ชันต่างๆ ในภาษา Python
5. ศึกษาการใช้งานของ OpenCV
6. ศึกษาการใช้ฟังก์ชันของ Raspberry Pi
7. จัดหาอุปกรณ์ที่ใช้ในโครงการ
8. ทดลองการเปิดกล้อง Webcam โดยใช้โปรแกรม IDLEX (python GUI)
9. ทดลองใช้ฟังก์ชันในการจับภาพจากกล้อง Webcam
10. ทดลองการเก็บภาพเข้าฐานข้อมูลเพื่อนำมาเปรียบเทียบใบหน้าบุคคล
11. ทดลองฟังก์ชัน Face Recognition
12. ทดลองฟังก์ชัน Face Detection
13. แก้ไขข้อผิดพลาดของโปรแกรม
14. สรุปผลและจัดทำเอกสารรายงานการวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

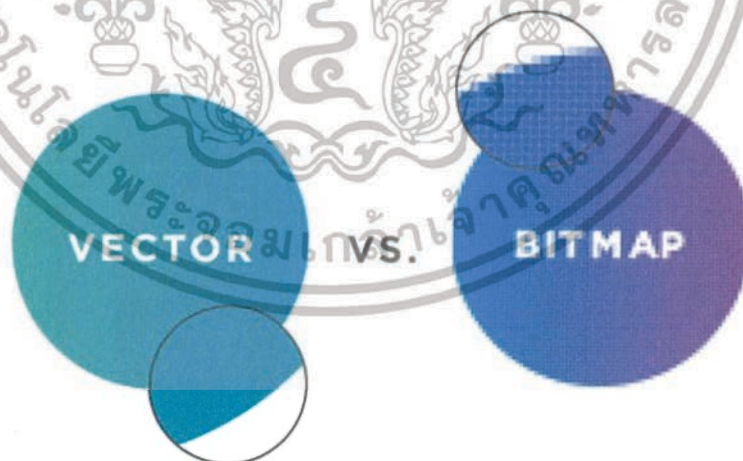
## บทที่ 2

# ทฤษฎีที่เกี่ยวข้อง

### 2.1 ภาพดิจิทัล (Digital Image)

ปกติแล้วข้อมูลภาพต่างๆ ไปนั้นได้มาจากการที่แสงตกกระทบวัตถุแล้วเกิดการสะท้อนผ่านเลนส์เข้าสู่ตัวบันทึกภาพ อาจอยู่ในรูปแบบของตัวตรวจจับ (Sensor) หรือฟิล์ม (Film) หากย้อนนึกถึงเมื่อเวลาถ่ายภาพด้วยกล้องดิจิทัล วัตถุหรือภาพที่เห็นด้วยตาเป็นข้อมูลสามมิติ (Three - dimension) ที่ประกอบไปด้วยความลึก ความสูง และความกว้าง แต่เมื่อแปลงข้อมูลภาพออกมาเป็นภาพดิจิทัล (Digital Image) ข้อมูลของภาพนั้นจะประกอบด้วยความกว้างและความสูง (Width and Height) เท่านั้น ซึ่งในปัจจุบันมีอุปกรณ์อิเล็กทรอนิกส์หลายประเภทที่สามารถสร้างสัญญาณภาพดิจิทัลได้สะดวกและง่ายขึ้น เช่น กล้องถ่ายภาพดิจิทัล กล้องวิดีโอดิจิทัล สแกนเนอร์ และกล้องเว็บแคม เป็นต้น

สามารถที่จะแบ่งรูปภาพที่ปรากฏและใช้งานบนเครื่องคอมพิวเตอร์ออกเป็น 2 ประเภท คือ ภาพแบบบิตแมป (Bitmap Image) และภาพแบบเวกเตอร์ (Vector Image) โดยรูปภาพแบบบิตแมปจะพิจารณาตัวรูปภาพซึ่งถูกแบ่งออกเป็นส่วนย่อยเล็กๆ หลายๆ ส่วนหรือที่เรียกว่า พิกเซล ที่ถูกนำมา รวมกันและแสดงผลเป็นภาพ ส่วนรูปภาพแบบเวกเตอร์จะประกอบด้วยเส้นสายต่างๆ ที่สร้างขึ้นจากการคำนวณทางคณิตศาสตร์ ของลักษณะทางเรขาคณิตเพื่อสร้างรูปทรงต่างๆ ดังรูปที่ 2.1



รูปที่ 2.1 ภาพดิจิทัล และภาพบิตแมป

## 2.2 ภาพแบบบิตแมป (Bitmap Image)

ภาพบิตแมปหรือเรียกอีกอย่างหนึ่งว่าภาพแบบราสเตอร์ (Raster Image) เป็นภาพที่ประกอบขึ้นจากจุดขนาดเล็กๆ หรือที่เรียกว่า พิกเซล จำนวนมากที่เรียงต่อกันจนเป็นภาพภาพหนึ่ง เพื่อให้เห็นภาพลักษณะชัดเจนยิ่งขึ้น ให้นึกถึงการสร้างภาพบนตารางสี่เหลี่ยมเล็กๆ จะใช้สีแต้มลงในช่องสี่เหลี่ยมแต่ละช่องจนกลายเป็นภาพสมบูรณ์ที่มีขนาดใหญ่ ซึ่งเหมือนกับการป้ายสีแปลอักษรบนอัมจันทร์ ภาพแบบบิตแมปนี้จะมีจำนวนจุดขนาดเล็กๆ จำนวนมาก ดังนั้นดวงตาของมนุษย์ไม่สามารถที่จะมองเห็นและแยกแยะรายละเอียดส่วนย่อยเล็กๆ นั้นได้ แต่เมื่อลองขยายภาพดูจะเห็นเป็นรูปตาราง ยิ่งขยายใหญ่เท่าไรตารางสี่เหลี่ยมก็ยังมีขนาดใหญ่ขึ้นจนทำให้มองเห็นจุดของภาพหรือพิกเซล จึงมีผลทำให้รายละเอียดของภาพมีความไม่ชัดเจนมากขึ้น โดยทั่วไปแล้วภาพบิตแมปเป็นภาพประเภทที่นิยมใช้กันมากในภาพถ่าย

## 2.3 ภาพแบบเวกเตอร์ (Vector Image)

ภาพแบบเวกเตอร์จะมีคุณสมบัติที่แตกต่างกับแบบบิตแมปคือ ภาพประเภทนี้ไม่ว่าจะขยายใหญ่แค่ไหน ก็ยังคงรายละเอียด และความคมชัดไวได้เหมือนเช่นเดิมโดยไม่ผิดเพี้ยน เนื่องจากภาพแบบเวกเตอร์นั้นประกอบด้วยเส้นตรง เส้นโค้งและรูปทรงต่างๆ ภาพที่ได้จะสร้างขึ้นจากคำสั่งที่บอกถึงลักษณะของภาพในรูปแบบทางเรขาคณิตด้วยสมการทางคณิตศาสตร์ ดังนั้นโปรแกรมที่ต้องการเปิดรูปภาพจะต้องนำสมการต่างๆ ที่บันทึกเอาไว้มาคำนวณและสร้างรูปทรงของภาพขึ้นมาใหม่ จุดเด่นคือ ไม่ว่าจะขยายภาพให้ใหญ่แค่ไหนคอมพิวเตอร์ก็จะคำนวณค่าต่างๆ ให้ใหม่ทุกครั้ง ทำให้ภาพที่เกิดขึ้นมามีความคมชัด ภาพแบบเวกเตอร์จึงเหมาะกับการงานที่มีความแม่นยำและต้องการความละเอียดสูง เช่น การสร้างภาพโลโก้ การสร้างภาพสามมิติ และการสร้างแบบร่างทางวิศวกรรม

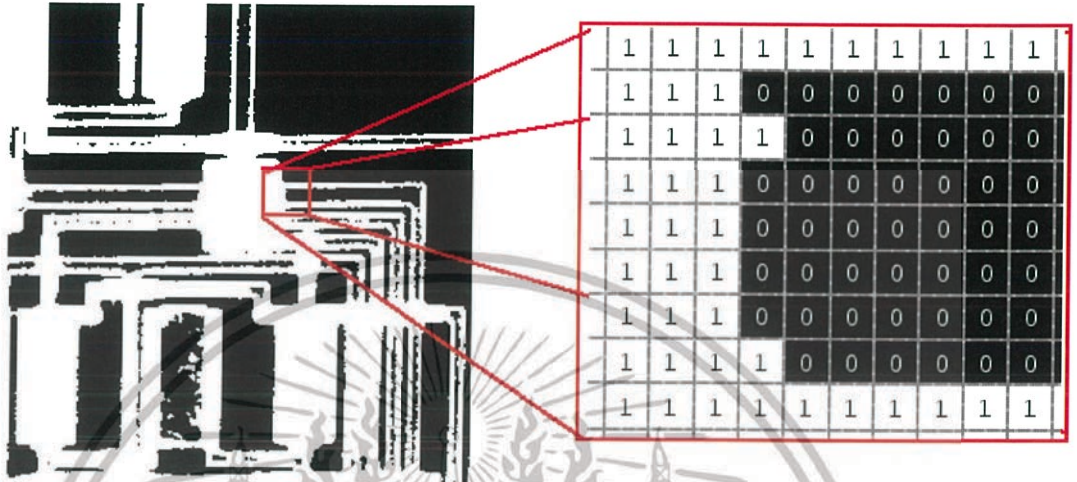
## 2.4 ประเภทของภาพบิตแมป

โดยทั่วไปสามารถแบ่งประเภทของภาพบิตแมปตามคุณสมบัติของสีออกได้เป็น 4 ประเภท ดังนี้

1. ภาพขาวดำ (Binary Image)
2. ภาพระดับสีเทา (Gray Scale Image)
3. ภาพสี (Color Image)
4. ภาพแบบดัชนี (Index Image)

## 2.5 ภาพขาวดำ (Binary Image)

ลักษณะของภาพขาวดำในแต่ละพิกเซลจะแสดงด้วยค่าแบบไบนารี (Binary) คือ มี 1 บิต ซึ่งประกอบไปด้วยค่า 1 และ 0 โดยที่ 1 หมายถึงจุดภาพสีขาว และ 0 หมายถึงจุดภาพสีดำ ภาพประเภทนี้เหมาะสำหรับภาพที่เกี่ยวกับตัวอักษร (Text) ภาพลายนิ้วมือ ดังรูปที่ 2.2



รูปที่ 2.2 ภาพขาวดำ

## 2.6 ภาพระดับสีเทา (Gray Scale Image)

ลักษณะของภาพชนิดนี้ ในแต่ละพิกเซล จะมีค่าความเข้มของแสงในแต่ละระดับที่แตกต่างกันไป ตั้งแต่สีขาวไปยังสีดำ สามารถกำหนดระดับความเข้มของแสงนั้นได้ โดยใช้ค่าระดับเทา ปกติแล้วภาพแบบระดับสีเทาก็จะมีความละเอียด (Resolution) เท่ากับ 8 บิต ซึ่งภาพจะมีค่าระดับความเข้มแสงของสีดำเท่ากับ 0 ส่วนค่าระดับความเข้มของสีขาวจะมีค่าเท่ากับ 255 ดังรูปที่ 2.3

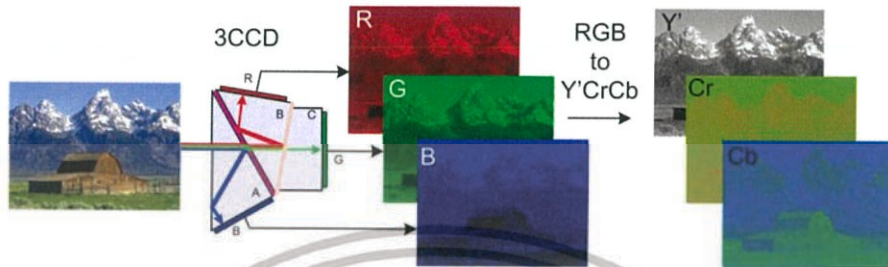


รูปที่ 2.3 ภาพระดับสีเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 ภาพสี (Color Image)

ในแต่ละพิกเซลของภาพสีจะเก็บค่าระดับความเข้มแสงแต่ละแถบแสงของแม่สีหลัก 3 สีที่ซ้อนกัน คือ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) ซึ่งในแต่ละพิกเซลนั้นๆ ก็จะได้แสดงผลของค่าสีของแต่ละพิกเซลตามระดับความเข้มในแต่ละแถบแสงนั้น ดังรูปที่ 2.4



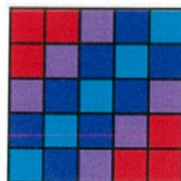
รูปที่ 2.4 ภาพสี

## 2.8 ภาพแบบดัชนี (Index Image)

ภาพประเภทนี้ในแต่ละพิกเซลของภาพจะเก็บค่าดัชนี (Index Number) ซึ่งเป็นตัวเลขจำนวนเต็ม ซึ่งจะถูกนำค่าดัชนีดังกล่าวไปเทียบกับตารางสี (Color Table) ซึ่งเป็นตารางแสดงค่าแสงสีแดง เขียว และน้ำเงิน ซึ่งค่าดัชนีจะเป็นตัวชี้ให้เห็นว่าภาพในตำแหน่งพิกเซลนั้นๆ มีค่าอัตราส่วนของ แม่สีแสง 3 สีในอัตราส่วนละเท่าไร ดังรูปที่ 2.5

0	0	1	2	3
0	1	2	3	2
1	2	3	2	1
2	3	2	1	0
3	2	1	0	0

0 =   
 1 =   
 2 =   
 3 = 



รูปที่ 2.5 ภาพแบบดัชนี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## 2.10 การตรวจจับใบหน้า (Face Detection)

เป็นการตรวจจับบริเวณของภาพส่วนที่เป็นใบหน้าบุคคลโดยใช้เทคนิคต่างๆ เป็นต้นว่า เครือข่ายนิวรัล (Neural Network) เป็นเทคนิคที่ให้ผลลัพธ์ที่มีความแม่นยำสูง แต่ใช้เวลาประมวลผลสูงมาก ขณะที่การใช้โมเดลสี (Color Model) เป็นเทคนิคที่นิยมใช้ แต่การตรวจจับจากสีเพียงอย่างเดียวจะถูกรบกวนได้ง่ายจากสภาพแสงและสิ่งอื่นๆ ที่มารบกวน ส่วนการใช้แม่แบบภาพใบหน้า (Geometric Facial Template) มีการประมวลผลที่รวดเร็ว แต่การค้นหาอวัยวะบนใบหน้ายังคงมีข้อจำกัด นอกจากนี้ยังสามารถใช้หลายเทคนิคพร้อมกัน เพื่อให้ได้คุณภาพและประสิทธิภาพที่ดีขึ้น ในปัจจุบันสามารถจำแนกวิธีการพื้นฐานในการตรวจจับใบหน้าได้เป็น 4 ประเภท

### 2.10.1 Knowledge – Based Method

เป็นวิธีการที่อาศัยพื้นฐานความรู้ของมนุษย์ ที่มีต่อส่วนประกอบสำคัญของใบหน้ามนุษย์มาหา กฎเกณฑ์ในการจำแนกใบหน้าของมนุษย์ ยกตัวอย่างเช่น รูปที่มีใบหน้านั้นย่อมจะต้องประกอบด้วย ดวงตา 2 ดวง ซึ่งอยู่ในตำแหน่งที่สมมาตรกัน ต้องมีจมูก 1 จมูก และปาก 1 ปาก ซึ่งจะสามารถจำแนกใบหน้าได้จากอวัยวะแต่ละส่วนจะมีความสัมพันธ์กัน โดยใช้ระยะทางและตำแหน่งตามกฎเกณฑ์ที่ได้ตั้งไว้

ข้อเสียของวิธีการนี้คือ การหากฎเกณฑ์ที่จะมาทำการจำแนกใบหน้าที่มนุษย์นั้นเป็นสิ่งที่ยาก เพราะถ้าหากระบุกฎเกณฑ์ละเอียดมากเกินไปนั้น ก็จะทำให้การตรวจจับใบหน้าที่ทำได้ยาก (เนื่องจากไม่มีใบหน้าที่ผ่านกฎเกณฑ์เลย) หรือถ้าระบุกฎเกณฑ์น้อยเกินไปก็ทำให้การตรวจจับใบหน้าเกิดข้อผิดพลาดขึ้น (ตรวจจับสิ่งที่ไม่ใช่ใบหน้าเข้ามาด้วย)

### 2.10.2 Feature Invariant Approaches

เป็นวิธีการที่ใช้สำหรับการค้นหาลักษณะเด่น (Feature) ที่แตกต่างกันของแต่ละใบหน้าที่ในการตรวจจับ สมมติฐานของวิธีนี้นั้น ตั้งอยู่บนพื้นฐานของการสังเกตของมนุษย์ที่จะสามารถจำแนกวัตถุต่างๆ ได้ไม่ว่าจะมีการเปลี่ยนแปลงสภาพแวดล้อมเกิดขึ้น เปลี่ยนตำแหน่ง หรือเปลี่ยนสภาพแสง เป็นต้น ซึ่งในการตรวจหาลักษณะเด่นของใบหน้านั้น ยกตัวอย่างเช่น ลักษณะเด่นของคิ้ว ตา จมูก ปาก และเส้นผม โดยเทคนิคการตรวจจับเส้นขอบ (Edge Detection) จะดึงออกมาเพื่อตรวจสอบ และจากนั้นจะใช้สถิติในการอธิบายความสัมพันธ์และยืนยันการตรวจพบใบหน้า

ข้อเสียของวิธีการนี้คือ การตรวจจับคุณลักษณะเด่นของใบหน้านั้น สามารถดึงออกมาได้ยาก เนื่องจากสภาพแสง สัญญาณรบกวน อีกทั้งไม่สามารถหาขอบของจุดเด่นบนใบหน้าได้

### 2.10.3 Template Matching Methods

วิธีการนี้ใช้ Standard Template ของใบหน้า (โดยทั่วไปจะใช้เป็นของใบหน้าตรง) ซึ่งเกิดจากการกำหนดค่าเองหรือกำหนดโดยฟังก์ชัน เช่น หาความสัมพันธ์โดยใบหน้าที่ได้รับเข้ามานั้น จะนำมาหาค่า Correlation Value กับ Standard Pattern ของใบหน้า

ข้อดีของวิธีการนี้คือ ทำได้ง่าย แต่ข้อเสียคือ เนื่องจากตำแหน่งการวางและรูปทรงที่แตกต่างไปจาก Standard Template จึงทำให้ประสิทธิภาพของการตรวจจับใบหน้าที่ยังทำได้ไม่ดี

### 2.10.4 Appearance – Based Methods

เป็นวิธีการที่ใช้โมเดลของการเรียนรู้ของกลุ่มรูปภาพตัวอย่าง ซึ่งกลุ่มรูปภาพตัวอย่างนั้นจะประกอบด้วยกลุ่มรูปที่เป็นใบหน้าและไม่ใช่ใบหน้า ซึ่งวิธีการนี้ได้ใช้เทคนิคการวิเคราะห์ทางสถิติ และเทคนิค Machine Learning มาใช้ในการหาคุณลักษณะของรูปภาพที่เป็นใบหน้า และไม่ได้เป็นใบหน้า

ซึ่งจากการศึกษาในเรื่องของการตรวจจับใบหน้า ทำให้พบว่าวิธีการ Appearance – Based มีประสิทธิภาพดีกว่าวิธีอื่นๆ ซึ่งโดยเฉพาะงานวิจัยของ Viola และ Jones นั้นมีประสิทธิภาพและความสามารถในการตรวจจับใบหน้าได้อย่างรวดเร็ว

## 2.11 วิธีการตรวจจับใบหน้าของ Viola – Jones

เป็นวิธีการที่นำเสนอโดย Paul Viola และ Michael J. Jones ได้คิดค้นและนำเสนอในปี ค.ศ. 2001 ซึ่งเทคนิคนี้เป็นเทคนิคที่เป็นที่ยอมรับและรู้จักในเรื่องของการตรวจจับใบหน้าอย่างแพร่หลาย โดยวิธีการตรวจจับใบหน้าของ Viola – Jones นี้สามารถแบ่งออกเป็น 3 ขั้นตอน ได้แก่ การคำนวณรูปแบบการจำลองด้วย Integral Image, การค้นหาแบบจำลองด้วย Adaboost และการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)

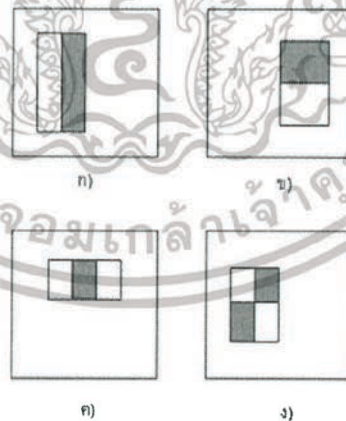
แนวคิดพื้นฐานของเทคนิคการตรวจจับใบหน้าของ Viola – Jones คือ การใช้ Subwindow วิ่งพิจารณาเพื่อทำการตรวจจับใบหน้าจากภาพอินพุต ซึ่งเทคนิคโดยทั่วไปในการตรวจจับหาใบหน้า นั้นจะทำการปรับขนาดภาพอินพุตให้มีขนาดแตกต่างกัน และใช้ตัวตรวจจับ (Detector) ที่มีขนาดคงที่เพื่อวิ่งตรวจจับหาวัตถุ แต่เทคนิคโดยทั่วไปนี้นั้นจะมีข้อเสียคือ ระยะเวลาในการคำนวณไม่คงที่ ซึ่งแตกต่างกับเทคนิคการตรวจจับใบหน้าของ Viola – Jones ที่นำเสนอเทคนิคการตรวจจับใบหน้าในรูปแบบใหม่ โดยใช้การจำลองรูปแบบ Haar – Like เป็นตัวตรวจจับ แทนที่จะทำการปรับขนาดของภาพอินพุต แต่จะไปทำการปรับขนาดของตัวตรวจจับแทน โดยตัวตรวจจับจะทำการวิ่งตรวจจับ

ใบหน้าหลายๆ รอบ ในแต่ละรอบจะใช้ตัวตรวจจับที่มีขนาดแตกต่างกัน เมื่อเปรียบระหว่างวิธีเดิมกับวิธีของ Viola – Jones แล้วนั้น ทั้งสองวิธีใช้เวลาในการคำนวณไม่ได้แตกต่างกันมากนัก แต่เวลาในการคำนวณการตรวจจับภาพใบหน้าแต่ละรอบจะมีค่าคงที่ แม้ขนาดของตัวตรวจจับจะมีขนาดแตกต่างกันก็ตาม โดยเทคนิคนี้นำเสนอการคำนวณรูปแบบจำลอง Haar – Like ด้วยวิธีการ Integral Image

### 2.11.1 ขั้นตอน Integral Image

การจำลองรูปแบบ Haar - Like ในวิธีการตรวจจับใบหน้าของ Viola - Jones นั้นจะมีลักษณะดังรูป 2.7 ซึ่งในพื้นที่สี่เหลี่ยม จะประกอบด้วยพื้นที่สี่เหลี่ยม 2 ประเภท ได้แก่ พื้นที่สี่เหลี่ยมส่วนแรเงา และพื้นที่สี่เหลี่ยมส่วนที่ไม่ได้แรเงา ซึ่งการหาค่าการจำลองรูปแบบ Haar - Like นั้นสามารถทำได้จากการหาผลต่างความเข้มแสงระหว่างพื้นที่ส่วนแรเงา และพื้นที่ส่วนที่ไม่ได้แรเงา เมื่อได้ผลลัพธ์นำไปเปรียบเทียบกับค่าขีดแบ่ง (Threshold) กับขั้ว (Polarity) ซึ่งค่าเหล่านี้จะเป็นการตัดสินใจภาพอินพุตเป็นบวก (ภาพใบหน้า) หรือเป็นลบ (ไม่ใช่ภาพใบหน้า) ซึ่งผลลัพธ์ที่ได้นั้นจะถูกนำไปพิจารณาในขั้นตอนต่อไป

ตัวอย่างการจำลองรูปแบบ Haar - Like ที่แสดงใน รูปที่ 2.7 มี 3 แบบคือ การจำลองรูปแบบ Haar - Like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสองภาพหรือ Two - Rectangle Feature (รูปที่ 2.7 ก และ 2.7 ข) การจำลองรูปแบบ Haar - Like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสามภาพ หรือ Three - Rectangle Feature (รูปที่ 2.7 ค) และการจำลองรูปแบบ Haar - Like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสี่ภาพ หรือ Four - Rectangle Feature (รูปที่ 2.7 ง)

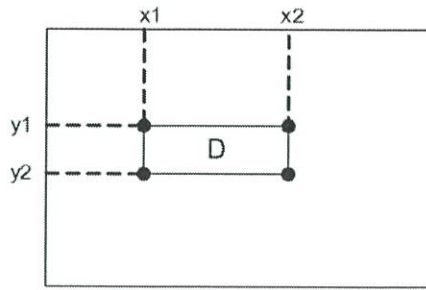


รูปที่ 2.7 แบบจำลอง Haar – Like

ในรูปที่ 2.8 เป็นวิธีการทั่วไปในการหาค่าผลรวมความเข้มของพื้นที่สี่เหลี่ยมใดๆ ในรูปภาพ สามารถเขียนแทนได้ด้วยสมการต่อไปนี้

$$i_s(x_1, x_2, y_1, y_2) = \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} i(x, y) \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การคำนวณความเข้มในพื้นที่ D โดยวิธีการทั่วไป

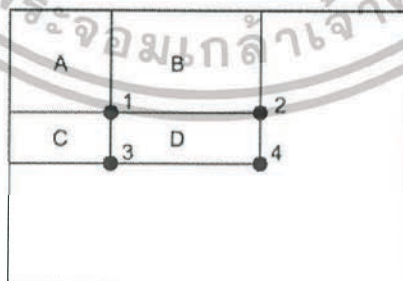
โดยที่  $i(x, y)$  แทนค่าความเข้มที่จุด  $x$  และ  $y$  ในรูปภาพ  $i_s(x_1, x_2, y_1, y_2)$  แทนผลรวมความเข้มภายในพื้นที่สี่เหลี่ยม โดยที่  $x_1, x_2, y_1$  และ  $y_2$  เป็นพิกัดมุมทั้งสี่ของรูปสี่เหลี่ยม จากสมการที่ (2.1) จะเห็นได้ว่าเวลาที่ใช้ในการคำนวณนั้นจะขึ้นอยู่กับขนาดของรูปสี่เหลี่ยม

เพื่อให้ได้ประสิทธิภาพที่ดีมากยิ่งขึ้น Viola และ Jones ได้เสนอเทคนิคที่เรียกว่า Integral Image คือ การรวมความเข้ม (Intensity) ของแต่ละพิกเซลเข้าด้วยกัน ซึ่งเป็นเทคนิคที่ใช้ในการคำนวณรูปแบบจำลอง Haar-Like ซึ่งสามารถเขียนสมการได้ดังต่อไปนี้

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.2)$$

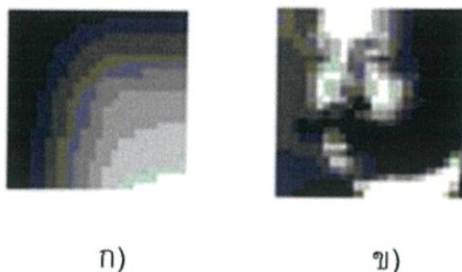
กำหนดให้  $ii(x, y)$  คือ ค่า Integral Image ที่ตำแหน่งจุด  $(x, y)$  และ  $i(x', y')$  คือ ค่าความเข้มในแต่ละพิกเซลของภาพต้นฉบับ

ตัวอย่างการคำนวณหาผลรวมความเข้มของพื้นที่สี่เหลี่ยม D โดยใช้ค่า Integral Image จำนวน 4 จุด ผลลัพธ์ที่ได้ คือ  $4 + 1 - (2 + 3)$  ดังแสดงในรูปที่ 2.9 ซึ่ง จะเห็นได้ว่าไม่ว่าสี่เหลี่ยมจะมีขนาดเท่าใดก็จะใช้เวลา ในการคำนวณค่าผลรวมความเข้มคงที่



รูปที่ 2.9 การคำนวณผลรวมความเข้มในพื้นที่ D โดยวิธี Integral Image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ตัวอย่างภาพ Integral Image

กำหนดให้การจำลองรูปแบบ Haar - Like ที่ได้หลังจากทำการเปรียบเทียบกับค่า Threshold เรียกว่า Weak Classifier เนื่องจากการจำลองรูปแบบ Haar - Like ที่ได้รับจะให้คำตอบที่ถูกต้องกว่าการเดาสุ่มเล็กน้อยเท่านั้น ซึ่งการหา Weak Classifier สามารถเขียนแทนได้ด้วยสมการต่อไปนี้

$$h(x_i, f, p, \theta) = \begin{cases} 1, & \text{if } pf(x_i) < p\theta \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

โดย  $x$  คือ Subwindow ของภาพอินพุต  $h(x_i, f, p, \theta)$  คือ Weak Classifier ที่พิจารณาจากการจำลองรูปแบบ Haar - Like เทียบกับค่าขีดแบ่ง (Threshold,  $\theta$ ) และ ขั้ว ( $p$ ) ซึ่งเป็นตัวกำหนดทิศของสมการ

### 2.11.2 ขั้นตอนการเรียนรู้ด้วยวิธี Adaboost

ในขั้นตอนนี้จะเป็นการจำลองรูปแบบ Haar - Like ที่ได้จากขั้นตอนที่ 1 นำมาเข้ากระบวนการ Machine Learning ซึ่งเรียกว่า “Adaptive Boost” หรือ “Adaboost” ซึ่งขั้นตอนคือ หา Weak Classifier และนำมากำหนดค่าถ่วงน้ำหนัก เพื่อให้ค่าความผิดพลาดน้อยที่สุด เพื่อนำไปสร้าง Strong Classifier

ขั้นตอนการเรียนรู้ด้วย Adaboost มีดังต่อไปนี้

1. พิจารณาเลือกกลุ่มภาพตัวอย่างจำนวนภาพ เพื่อใช้ในเรียนรู้กลุ่มตัวอย่างภาพสามารถเขียน ได้ดังสมการด้านล่าง

$$S = \{(x_i, z_i), i = 1, 2, \dots, N\} \quad (2.4)$$

เมื่อ  $x_i$  คือ ภาพตัวอย่างที่  $i$  และ  $z_i$  คือ ค่าที่ระบุว่าเป็นภาพตัวอย่างที่เป็นใบหน้าคน ( $z_i = 1$ ) หรือไม่ใช่ใบหน้าคน ( $z_i = 0$ )

2. กำหนดจำนวน Weak Classifier มีค่าเท่ากับ  $t = 1, 2, \dots, T$  รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กำหนดค่าถ่วงน้ำหนักเริ่มต้น ( $w_{t,i}$ ) Weak Classifier ตัวแรก (เมื่อ  $t = 1$ ) มีค่าเท่ากับ  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$  สำหรับ  $z_i = 0,1$  เมื่อ  $m$  และ  $l$  คือ จำนวนภาพที่เป็นใบหน้าและไม่ใช่ใบหน้าตามลำดับ

4. ขั้นตอนการพิจารณา Weak Classifier ทีละค่ามีดังต่อไปนี้

a. ทำการนอร์มัลไลซ์ (Normalize) ค่าถ่วงน้ำหนัก

$$w_{1,i} \approx \frac{w_{1,i}}{\sum_{j=1}^n w_{1,i}} \quad (2.5)$$

b. ค้นหา Weak Classifier ที่ทำให้ค่าความผิดพลาด  $\epsilon_t$  ต่ำที่สุดตามสมการต่อไปนี้

$$\epsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - z_i| \quad (2.6)$$

c. กำหนดให้

$$h_t(x) = h(x, f_t, p_t, \theta_t)$$

เมื่อ  $f_t, p_t$  และ  $\theta_t$  คือ ค่าที่ทำให้ได้ค่าต่ำที่สุดของ  $\epsilon_t$

d. จากนั้นทำการปรับปรุงค่าถ่วงน้ำหนักของแต่ละตัวอย่างใหม่จะมีค่าเท่ากับ

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_t} \quad (2.7)$$

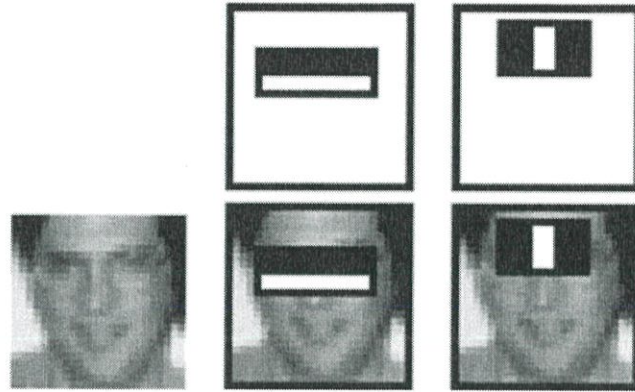
เมื่อ  $e_t = 0$  ถ้าตัวอย่าง  $x_i$  ถูกคัดเลือกอย่างถูกต้อง และ  $e_t = 1$  ในกรณีอื่นๆ และค่า  $\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$

5. ผลลัพธ์สุดท้ายที่ได้ คือ Strong Classifier

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

$$\text{เมื่อ } \alpha_t = \log \frac{1}{\beta_t}$$

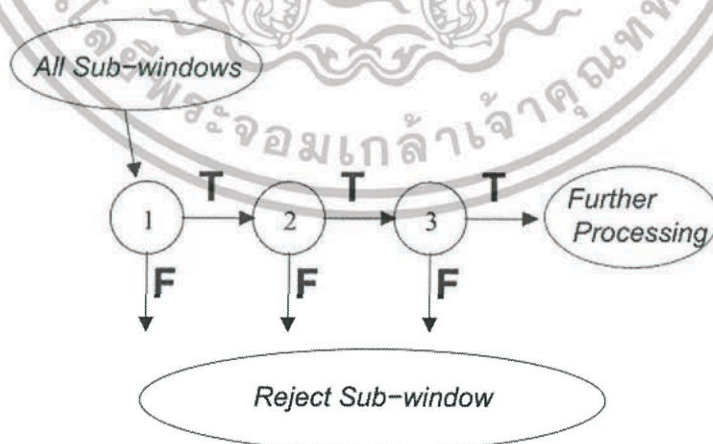
รูปที่ 2.11 แสดงตัวอย่างของการจำลองรูปแบบ Haar - Like สองลำดับแรกที่ได้จากกระบวนการเรียนรู้ด้วยวิธี Adaboost จากรูปที่ 2.11 สังเกตเห็นว่าการจำลอง รูปแบบ Haar - Like แรกเป็นการจำลองรูปแบบ Haar - Like ที่ได้จากความแตกต่างบริเวณดวงตาซึ่งจะมีมากกว่าบริเวณส่วนบนของโหนกแก้ม สำหรับการจำลอง รูปแบบ Haar - Like ตัวที่สองได้จากความแตกต่างบริเวณดวงตาสองข้างซึ่งจะมีมากกว่าบริเวณสันจมูก



รูปที่ 2.11 ตัวอย่างของ Strong Classifier จากกระบวนการเรียนรู้ด้วย Adaboost

### 2.11.3 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)

ขั้นตอนที่ 3 ที่ Viola - Jones นำเสนอเป็นเทคนิคที่ใช้ในการเพิ่มประสิทธิภาพในการตรวจจับให้มีความถูกต้อง และยังใช้เวลาในการคำนวณลดลงด้วย โดยแนวความคิดของขั้นตอนนี้คือการสร้างการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier) โดยเทคนิคนี้จะนำตัวจำแนก (Classifier) หลายๆ ตัวต่อกันเป็นลำดับ ดังแสดงในรูปที่ 2.12 โดยที่ตัวจำแนกในลำดับต้นๆ มักจะมีความซับซ้อนที่น้อยกว่า อาจสร้างมาจาก Weak Classifier ไม่กี่ตัวเพื่อลดปริมาณการคำนวณ และลดระดับของค่าแบ่งขีด (Threshold) ลงเพื่อลดความผิดพลาดในการตรวจจับไม่พบใบหน้า ซึ่งภาพย่อยที่ถูกจัดประเภทเป็นไม่ใช่ใบหน้า (Non - Face) จะถูกทิ้งไปเป็นจำนวนมากในลำดับต้นๆ ในทางตรงข้าม ถ้าภาพย่อยนั้นถูกจำแนกว่ามีโอกาสเป็นภาพใบหน้า ก็จะถูกส่งต่อไปยังตัวจำแนกตัวถัดไปซึ่งมีความซับซ้อนสูงขึ้นตามลำดับ และกล่าวได้ว่ายังมีจำนวนชั้นของตัวจำแนกมากเท่าใด โอกาสที่ภาพย่อยจะเป็นใบหน้าจะยิ่งมีมากขึ้น



รูปที่ 2.12 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.12 การรู้จำใบหน้า (Face Recognition)

การรู้จำใบหน้าคือ กระบวนการที่ได้นำภาพใบหน้าที่ตรวจจับได้ และประมวลผลแล้ว จากขั้นตอนการตรวจจับใบหน้า มาเปรียบเทียบกับฐานข้อมูลของใบหน้าเพื่อระบุว่าใบหน้าที่ตรวจจับได้ตรงกับบุคคลใด โดยจะมีวิธีที่ใช้ในการพิจารณารูปภาพดังนี้

### 2.12.1 วิธีเชิงความรู้ (Knowledge - Based Methods)

เป็นวิธีการพิจารณาความสัมพันธ์ระหว่างลักษณะเด่นเฉพาะต่างๆ บนใบหน้า ที่มีตำแหน่งและองค์ประกอบหลักพื้นฐานตายตัวบนใบหน้า ตัวอย่างเช่น Yang and Huang (1994) ศึกษาวิธีเชิงความรู้แบบลำดับชั้น (Hierarchical) ซึ่งพิจารณาแบ่งเป็น 3 ลำดับชั้น ดังแสดงในรูปที่ 2.13



รูปที่ 2.13 ขอบเขตการพิจารณาของแต่ละลำดับชั้น

ชั้นที่ 1 พิจารณาหาตำแหน่งความน่าจะเป็นของใบหน้าโดยรวม ชั้นที่ 2 พิจารณาโดยค่าฮิสโตแกรม (Histogram) ร่วมกับการหาเส้นขอบ ชั้นที่ 3 พิจารณาลักษณะเด่นภายในของใบหน้าเช่น ความสมมาตรกันของตาทั้ง 2 ข้างตำแหน่งของจมูกและปากในแนวตั้งและแนวนอน ต่อมา Kotropoulos and Pitas (1997) ได้ศึกษาวิธีฐานกฎแบบกำหนดเขต (Rule-Based Localization Method) ซึ่งเป็นวิธีการที่มีคล้ายคลึงกับวิธีพิจารณาของ Yang and Huang แตกต่างกันที่พิจารณาค่าฮิสโตแกรมทั้งใน แนวตั้งและแนวนอน แต่ยังไม่สามารถจัดปัญหาของภาพ ที่มีพื้นหลังซับซ้อนและปัญหาความหลากหลายของใบหน้าบุคคล รวมทั้งการวางท่าที่แตกต่างกันได้

### 2.12.2 วิธีเชิงลักษณะ (Feature - Based Methods)

เป็นการใช้อัลกอริทึมพิจารณาลักษณะเด่นและโครงสร้างของใบหน้า รวมทั้งความเปลี่ยนแปลงขององค์ประกอบต่างๆ ของภาพ เช่น แสงเงา ตัวอย่างเช่น Chetverikov and Lerch (1993) ใช้ความแตกต่างของแสงเงาและเส้นแนวเป็นแบบจำลองในการค้นหาใบหน้า โดยแบบจำลองนั้นประกอบด้วยจุดความสว่างน้อย (จุดมืด) 2 จุดเพื่อแสดงดวงตาและจุดความสว่างมา (จุดอ่อน) 3 จุด เพื่อแสดงโหนกแก้มและจมูกแล้วหาความสัมพันธ์ของระยะห่าง และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของจุดต่างเพื่อคัดเลือกแบบหน้าที่เหมาะสม ระบบนี้มีข้อจำกัดอยู่ที่เมื่อแสงเงาของสภาวะแวดล้อมเปลี่ยนแปลง จะทำให้ประสิทธิภาพในการค้นหาเปลี่ยนไป ต่อมา Graf, Chen, Petajan and Cosatto (1995) พัฒนาทฤษฎีนี้โดยใช้งานในระบบภาพระดับเทา และนำตัวกรองช่วงผ่าน (Band Pass Filter) ตัวกรองเกาส์เซียน (Gaussian Filter) และค่าฮิสโตแกรมมาใช้หาค่าจุดสูงสุดและต่ำสุดเพื่อกำหนดขอบเขตของใบหน้า

ลักษณะเฉพาะอีกประการหนึ่งที่นิยมนำมาพิจารณาหาใบหน้าบุคคลคือ ค่าสีผิวมนุษย์ โดยมีการค้นหาในหลายๆ ปริภูมิสีโดย Crowley and Berard (1997) ทำการเก็บค่าเวกเตอร์ของสีผิวมนุษย์โดยแยกค่าสีออกเป็น R, G และ B เพื่อพิจารณาค่าฮิสโตแกรมของค่าสีผิวมนุษย์ที่ได้จากการสำรวจข้อมูลกลุ่มตัวอย่าง ต่อมา Saxe and Fould (1996) ประยุกต์ใช้วิธีการดังกล่าวกับปริภูมิสี HSV และใช้แนวคิดค่าฮิสโตแกรมตัดผ่าน (Histogram Intersection) เปรียบเทียบค่าฮิสโตแกรมควบคุมกับค่าฮิสโตแกรมปัจจุบัน ซึ่งทำให้ประสิทธิภาพดีกว่าการใช้ค่าเริ่มเปลี่ยน

Chai and Ngan (1998) พิจารณาค่าสีผิวในปริภูมิสี YCrCb โดยพิจารณาค่าเริ่มเปลี่ยนของค่าสีผิวมนุษย์ที่ได้จากการสำรวจข้อมูลกลุ่มตัวอย่าง ซึ่งต้องระมัดระวังในการเลือกใช้ให้เหมาะสมกับภาพที่ทำการทดสอบ และเป็นข้อเสียของการคัดแยกสีผิวมนุษย์ด้วยค่าเริ่มเปลี่ยน ต่อมามีการนำการคัดแยกสีผิวมาใช้ร่วมกับวิธีการอื่นๆ ในการค้นหาใบหน้า ตัวอย่างเช่น Wu, Yokoyama, Pramadihanto and Yachida (1996) ใช้ทฤษฎีคลุมเครือ (Fuzzy Theory) ในการคัดแยกสีผิวมนุษย์และสีผิวในปริภูมิสี CIE XYZ โดยทำการแปลงจากปริภูมิสี RGB นอกจากนี้ยังมีการนำเอาลักษณะเด่นของใบหน้าอื่นๆ มาใช้ในการค้นหา ตัวอย่างเช่น Yokoo and Hagiwara (1996) ใช้คุณสมบัติของโครงหน้าทั่วไปของมนุษย์ที่เป็นรูปไข่ สร้างแบบจำลองรูปวงรีเปรียบเทียบกับพารามิเตอร์ร่วมกับการค้นหาแบบจินเนติกอัลกอริทึม (Genetic Algorithm) จากการศึกษาพบว่าวิธีการนี้มีข้อจำกัดอยู่ที่รายละเอียดของใบหน้า ที่ทำการค้นหาต้องมีความชัดเจนและมีขนาดใหญ่เพียงพอต่อการค้นหา

### 2.12.3 วิธีเทียบเคียงแผ่นแบบ(Template Matching Methods)

เป็นการเปรียบเทียบภาพที่ต้องการค้นหากับโครงสร้างแบบจำลองของใบหน้ามาตรฐาน โดยเก็บข้อมูลความสัมพันธ์อย่างอิสระของส่วนต่างๆ บนใบหน้าได้แก่ โครงสร้างใบหน้า ตา จมูก และปาก ในท่าหน้าตรง ตัวอย่างเช่น Sakai, Nagao and Fujibayashi, (1969) เสนอโครงสร้างแบบจำลองแผ่นแบบย่อย (Subtemplates Model) โดยใช้ตัวกรองโซเบล (Sobel Filter) หาเส้นขอบ เพื่อหาตำแหน่งความน่าจะเป็นของส่วนย่อยต่างๆ บนใบหน้าที่สามารถเข้ากันได้ดีที่สุกกับแบบจำลองแผ่นแบบย่อยให้เป็นตำแหน่งของใบหน้าที่ต้องการ ต่อมา Tsukamoto, Lee and Tsuji (1994) เสนอแบบจำลองคุณภาพสำหรับรูปแบบหน้า (Qualitative Model for Face Pattern : QMF) โดยใช้พารามิเตอร์ของแสงสว่าง และเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบใบหน้าเป็นแบบจำลองของใบหน้า และเทคนิคที่นิยมใช้อีกอันหนึ่งคือ การวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis : PCA) โดย Jung, Lee C.W. and Lee Y.C (2002) ศึกษาการค้นหาค่าตำแหน่งที่แน่นอนของใบหน้า โดยใช้หลักการพื้นฐานทางด้านเรขาคณิต วิเคราะห์องค์ประกอบของใบหน้าให้อยู่ในรูปของสมการทางคณิตศาสตร์

ต่อมาได้มีการพัฒนาแบบจำลองให้มีความสามารถปรับตัวได้เรียกว่า แผ่นแบบเปลี่ยนรูปได้ (Deformable Templates) โดยแผ่นแบบสามารถปรับตัวให้รองรับความยืดหยุ่นของโครงสร้างใบหน้าที่เปลี่ยนแปลงไป และลดปัญหาที่มีในแผ่นแบบคงตัว ตัวอย่างเช่น Malciu and Preteux (2002) นำแบบจำลองหน้าจากฐานข้อมูล MPEG-4 มาสร้างเป็นแบบจำลองแผ่นแบบพารามิเตอร์ (Template Parameterization) และพิจารณาความยืดหยุ่นของโครงสร้างใบหน้าเปรียบเทียบกับความยืดหยุ่นของสปริง เพื่อหาค่าพลังงานที่สปริงใช้น้อยที่สุดในการจะปรับตัวเพื่อให้ภาพที่หาเส้นขอบนั้นเข้ากับแบบจำลองมากที่สุด โดยทำการพิจารณาในส่วนของตาและปากในรูปที่ 2.14



รูปที่ 2.14 แบบจำลองใบหน้ามนุษย์จากฐานข้อมูล MPEG-4

#### 2.12.4 วิธีเชิงลักษณะปรากฏ (Appearance - Based Methods)

เป็นการเปรียบเทียบภาพที่ต้องการค้นหากับโครงสร้างแบบจำลองของใบหน้า ที่ทำการเรียนรู้และฝึกสอนให้ระบบจดจำ และนำความรู้ในฐานข้อมูลมาใช้ในการพิจารณา โดยสามารถแบ่งย่อยได้หลายวิธีด้วยกัน ในที่นี้จะทำการเสนอเพียงวิธีการที่ได้รับความนิยมใช้งานกันอย่างกว้างขวาง 3 วิธีดังนี้

##### 2.12.4.1 วิธีหน้าลักษณะเฉพาะ (Eigenface Methods)

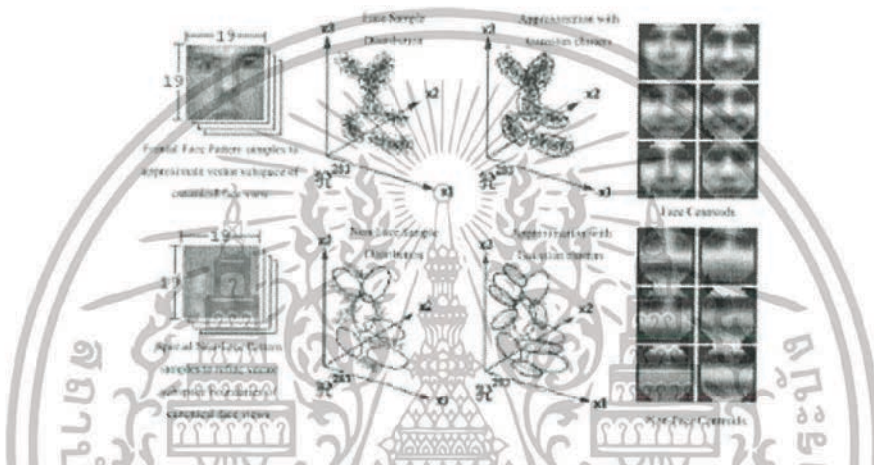
เป็นการวิเคราะห์องค์ประกอบพื้นฐานของใบหน้า ด้วยกระบวนการทางสถิติของใบหน้าที่หลากหลาย โดยหน้าลักษณะเฉพาะคือเซตของเวกเตอร์ลักษณะเฉพาะ (Eigen Vector) ที่สามารถหาได้จากเมทริกซ์ความแปรปรวน ร่วมเกี่ยว (Covariance Matrix) สร้างเป็นแบบจำลองของใบหน้าที่รวมเอาลักษณะเด่นต่างๆ ของภาพใบหน้าตัวอย่างมารวมกัน เพื่อหาค่าเฉพาะเจาะจงขององค์ประกอบบนใบหน้า ซึ่งในแต่ละบุคคลจะเป็นค่าเฉพาะของบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นๆ ตัวอย่างเช่น Turk and Pentland (1991) นำภาพระดับเทามาแปลงเป็นเวกเตอร์เพื่อหาค่าลักษณะเฉพาะ และนำค่าลักษณะเฉพาะของตัวอย่างภาพหน้าบุคคล มาสร้างเป็นแบบจำลองหน้าลักษณะเฉพาะเพื่อค้นหาตำแหน่งของใบหน้า

#### 2.12.4.2 วิธีเชิงการกระจาย (Distribution - Base Methods)

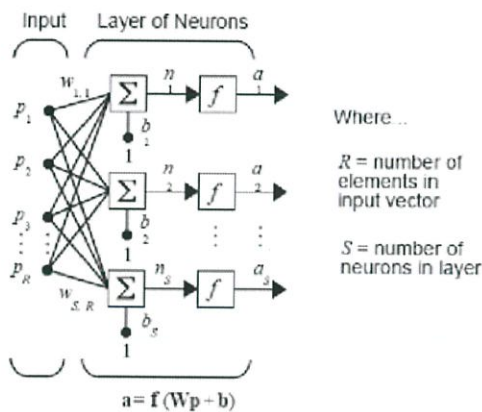
เป็นการแสดงการกระจายตัวของรูปแบบข้อมูลตัวอย่างที่มีความเป็นหน้า และความไม่เป็นหน้าเพื่อใช้เป็นบรรทัดฐานในการตัดสินใจ ตัวอย่างเช่น Sung and Poggio (1998) นำฟังก์ชันเกาส์เซียน (Gaussian Function) มาประมาณกลุ่มการกระจายของค่าเฉลี่ยกลุ่มตัวอย่าง แสดงในรูปที่ 2.15



รูปที่ 2.15 การกระจายตัวของข้อมูลตัวอย่าง

#### 2.12.4.3 โครงข่ายประสาทเทียม (Artificial Neural Networks)

เป็นการฝึกสอนโครงข่ายด้วยโครงสร้างใบหน้าที่มีความซับซ้อน โดยทำการปรับค่าน้ำหนักประสาทเพื่อให้ค่าความผิดพลาดกำลังสองเฉลี่ยมีค่าลดลง ความถูกต้องของโครงข่ายประสาทเทียมขึ้นกับตัวอย่างที่ทำการฝึกสอนจำนวนชั้นนิเวรอนและจำนวนนิเวรอนที่เหมาะสม ตัวอย่างเช่น Rowley, Baluja, and Kanade (1998) ใช้โครงข่ายประสาทเทียมแบบแพร่กลับในการฝึกสอนข้อมูลตัวอย่าง โดยทำการแบ่งข้อมูลออกเป็นภาพตัวอย่างใบหน้าบุคคล และภาพที่ไม่ใช่ใบหน้าบุคคลอย่างละ 1,048 ภาพ และแบ่งชุดข้อมูลออกเป็น 15 ชุดข้อมูล รวมข้อมูลตัวอย่างทั้งหมดได้ 15,720 ภาพ ซึ่งการฝึกสอนที่มีจำนวนข้อมูลตัวอย่างที่ครอบคลุมและมากเพียงพอจะช่วยลดความผิดพลาดของระบบ ดังรูปที่ 2.16



รูปที่ 2.16 ตัวอย่างโครงข่ายประสาทเทียม

### 2.13 OpenCV

OpenCV (Open Source Computer Vision Library) เป็น Library ที่ใช้สำหรับรวบรวมเอาความสามารถทางการประมวลผลภาพ (Image Processing) และความสามารถทางการมองเห็นของคอมพิวเตอร์ (Computer Vision) ความสามารถต่างๆ ของ OpenCV คือ ชุดคำสั่งหรือฟังก์ชันที่สามารถถูกเรียกใช้งานได้อย่างง่ายดาย สะดวก และรวดเร็ว จึงทำให้ OpenCV ถูกนำมาใช้เป็นเครื่องมือในกลุ่มของนักวิจัยและนักพัฒนาโปรแกรมคอมพิวเตอร์ด้าน Image Processing และ Computer Vision อย่างกว้างขวาง



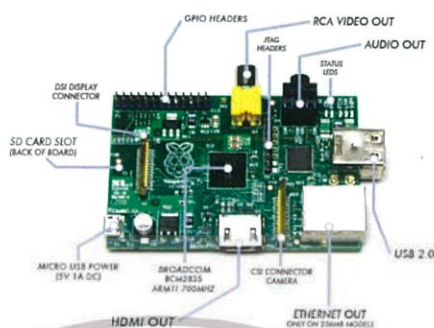
รูปที่ 2.17 OpenCV

### 2.14 Raspberry Pi

คือ คอมพิวเตอร์ขนาดจิ๋ว มีแต่สิ่งที่จำเป็นเพื่อการประมวลผล ราคาไม่แพงเกินไป บอร์ดใช้ชิป SoC ของ Broadcom BCM2835 ซึ่งบรรจุ ARM1176JZFS พร้อมทั้งหน่วยประมวลผลเลขทศนิยม (Floating Point) ทำงานที่ความถี่สัญญาณนาฬิกา 700 MHz DRAM ขนาด 512 MB ซึ่งถ้าเป็น Microcontroller ทั่วไป อาจมี RAM น้อยกว่านี้ (เป็น SRAM เพราะไม่มี MMU) ภายในมีแกนประมวลผลกราฟิก Video Core IV ทำให้สามารถเล่นไฟล์วิดีโอความละเอียด 1080p ได้ ไม่มีฮาร์ดดิสก์มาให้ มีเพียง ROM ที่ไว้ใช้ Boot เข้าระบบปฏิบัติการที่ได้รับการติดตั้งในการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำไดรฟ์ USB หรือผ่านทางเครือข่ายเท่านั้น ไม่มีจอภาพ คีย์บอร์ด หรือเมาส์มาให้ และถูกออกแบบให้รองรับ Linux ซึ่งงานบางอย่างอาจต้องใช้หน่วยความจำมาก เช่น งานประมวลผลภาพ Raspberry Pi จึงมีความเหมาะสมที่จะใช้งานในด้านนี้ ดังรูปที่ 2.18



รูปที่ 2.18 Raspberry Pi

## 2.15 ภาษา Python

Python เป็นภาษาระดับสูงภาษาหนึ่ง ที่มีความสามารถสูง ถูกสร้างขึ้นในปี 1989 โดย Guido Van Rossum ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือ สามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux, Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัวนี้เป็นภาษาลักษณะ Open Source เหมือนอย่าง PHP

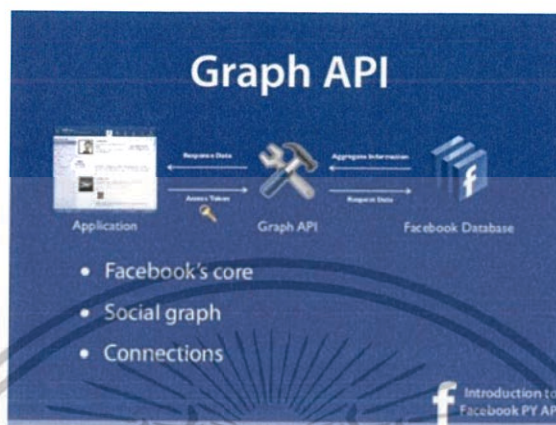
ในปัจจุบันภาษาที่ใช้ในการพัฒนา Web Application มีมากมายหลายภาษา อาทิเช่น ภาษา Perl, PHP, JAVA, ASP, Tcl, Python เป็นต้น สำหรับภาษา Python นับว่ายังใหม่ในวงการพัฒนาโปรแกรมบนเว็บ แต่ด้วยข้อดีหลายประการของภาษา Python ทำให้มีผู้นิยมใช้มากขึ้นเรื่อยๆ ซึ่งพอสรุปข้อดีของภาษา Python ได้ดังนี้

1. ง่ายต่อการเรียนรู้ โดยภาษา Python มีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษา Python จะคล้ายกับภาษา C มาก เพราะภาษา Python สร้างขึ้นมาโดยใช้ภาษา C ทำให้ผู้ที่คุ้นเคยภาษา C อยู่แล้วใช้งานภาษา Python ได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี
2. ไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้น เพราะตัวแปลภาษา Python อยู่ภายใต้ลิขสิทธิ์ GNU
3. ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษา Python ถูกออกแบบใช้งานกับระบบ Unix อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษา Python ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ เช่น Linux, Windows 95/98/ME, Windows NT, Windows 2000, OS/2
4. ภาษา Python ถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน เช่น ภาษา C, C++, Java, Perl
5. ภาษา Python เป็นภาษาประเภท Server Side Script คือ การทำงานของภาษา Python จะทำงานด้านฝั่ง Server แล้วส่งผลลัพธ์กลับมายัง Client ทำให้มีความปลอดภัยสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.16 Facebook Graph API

เป็นการส่งข้อมูลขึ้นบนเซิร์ฟเวอร์ของแอปพลิเคชัน Facebook เมื่อทำการประมวลผลภาพเสร็จสิ้นก็จะส่งข้อมูลเข้าเซิร์ฟเวอร์ของแอปพลิเคชัน Facebook และแสดงผลบนแอปพลิเคชันผ่าน Graph API (Application Program Interface) ดังรูปที่ 2.19



รูปที่ 2.19 หลักการทำงานของ Graph API

## 2.17 LINE Notify

LINE Notify คือ บริการที่คุณสามารถรับข้อความแจ้งเตือนจากเว็บเซอร์วิสต่างๆ ที่คุณสนใจได้ทาง LINE โดยหลังเสร็จสิ้นการเชื่อมต่อกับทางเว็บเซอร์วิสแล้ว คุณจะได้รับการแจ้งเตือนจากบัญชีทางการของ “LINE Notify” ซึ่งให้บริการโดย LINE นั้นเอง คุณสามารถเชื่อมต่อกับบริการที่หลากหลาย และยังสามารถรับการแจ้งเตือนทางกลุ่มได้อีกด้วย ซึ่งบริการหลักๆ ที่สามารถเชื่อมต่อได้แก่ GitHub, IFTTT หรือ Mackerel ดังรูปที่ 2.20



รูปที่ 2.20 LINE Notify

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

# วิธีการดำเนินงาน

### 3.1 ขั้นตอนการดำเนินงาน

1. ศึกษาค้นคว้าสิ่งที่น่าสนใจเกี่ยวกับการกำหนดหัวข้อโครงการงาน
2. นำเสนอหัวข้อโครงการงาน
3. ศึกษาและรวบรวมข้อมูลเกี่ยวกับการเลือกใช้อัลกอริทึม
4. ศึกษาการใช้ฟังก์ชันต่างๆ ในภาษา Python
5. ศึกษาการใช้งานของ OpenCV
6. ทดลองการเปิดกล้อง Webcam โดยใช้โปรแกรม IDLEX (python GUI)
7. ทดลองใช้ฟังก์ชันในการจับภาพจากกล้อง Webcam
8. ศึกษาการใช้ฟังก์ชันของ Raspberry Pi 3-Model B+
9. ทำการต่อกล้อง Webcam เข้ากับ Raspberry Pi 3 Model B+
10. ทดลองการตรวจจับใบหน้าผู้ใช้งานปกติ
11. ทดลองการตรวจจับใบหน้าผู้ใช้งานสวมแว่น
12. ทดลองการตรวจจับใบหน้าผู้ใช้งานสวมหมวก
13. ทดลองการตรวจจับใบหน้าบุคคลภายนอก
14. สรุปผลและจัดทำเอกสารรายงานการวิจัย

### 3.2 อุปกรณ์ที่เกี่ยวข้อง

#### 3.2.1 Raspberry Pi 3 Model B+

ทำหน้าที่รับข้อมูลจากกล้อง Webcam เข้ามาแล้วดำเนินการวิเคราะห์เพื่อตรวจจับหาใบหน้าและดวงตา จากนั้นทำการเปรียบเทียบกับภาพตัวอย่างในฐานข้อมูล หากพบว่าไม่ตรงกับภาพตัวอย่างในฐานข้อมูล ก็จะทำการส่งแจ้งเตือนผ่านแอปพลิเคชัน Facebook และ LINE



รูปที่ 3.1 Raspberry Pi 3 Model B+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2 กล้อง Webcam

ทำหน้าที่รับภาพแล้วส่งต่อไปยังบอร์ด Raspberry Pi 3 Model B+ เพื่อทำการประมวลผล



รูปที่ 3.2 กล้อง Webcam

## 3.3 ซอฟต์แวร์ที่เกี่ยวข้อง

### 3.3.1 OpenCV

OpenCV เป็น Library สำหรับรวบรวมเอาความสามารถทางด้านการประมวลผลภาพ (Image Processing) และการความสามารถทางด้านการมองเห็นของคอมพิวเตอร์ (Computer Vision) ที่ใช้เพิ่มฟังก์ชันให้กับกล้อง Webcam สามารถตรวจจับและแจ้งเตือนได้แบบ Real - Time



รูปที่ 3.3 OpenCV

### 3.3.2 Facebook Graph API

เป็นแอปพลิเคชันที่ใช้แสดงผลการแจ้งเตือนแบบ Real - Time ผ่าน Graph API (Application Program Interface) หากพบว่าภาพที่รับเข้ามาไม่ตรงกับภาพตัวอย่างในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 Facebook Graph API

### 3.3.2 LINE Notify

เป็นแอปพลิเคชันที่ใช้แสดงผลการแจ้งเตือนแบบ Real - Time หากพบว่าภาพที่รับเข้ามาไม่ตรงกับภาพตัวอย่างในฐานข้อมูล



รูปที่ 3.5 LINE Notify

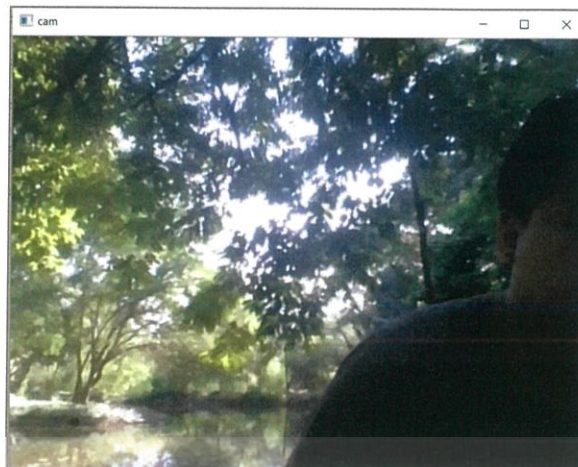
## 3.4 วิธีการดำเนินงาน

3.4.1 ทดลองการเปิดกล้อง Webcam โดยใช้โปรแกรม IDLEX (python GUI) การใช้ฟังก์ชัน show\_webcam เพื่อเปิดกล้อง ดังรูปที่ 3.7

```
import cv2
def show_webcam(mirror=False):
    cam = cv2.VideoCapture(0)
    while True:
        ret_val, img = cam.read()
        if mirror:
            img = cv2.flip(img, 1)
            cv2.imshow('cam', img)
            if cv2.waitKey(1) == 27:
                break
    cv2.destroyAllWindows()
show_webcam(mirror=True)
```

รูปที่ 3.6 ฟังก์ชัน show\_webcam

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ตัวอย่างตอนเปิดกล้อง

### 3.4.2 การสร้างข้อมูลรูปภาพด้วยฟังก์ชัน face\_cascade

เป็นฟังก์ชันที่ใช้สร้าง Datasets โดยจะทำการถ่ายภาพไว้จำนวนมากแล้วนำไปเก็บไว้ใน Folder ที่ได้กำหนดไว้ ดังรูปที่ 3.10

```
import cv2, sys, numpy, os
haar_file = 'haarcascade_frontalface_default.xml'
datasets = 'datasets'
sub_data = 'chcam'

path = os.path.join(datasets, sub_data)
if not os.path.isdir(path):
    os.mkdir(path)
(width, height) = (130, 100)

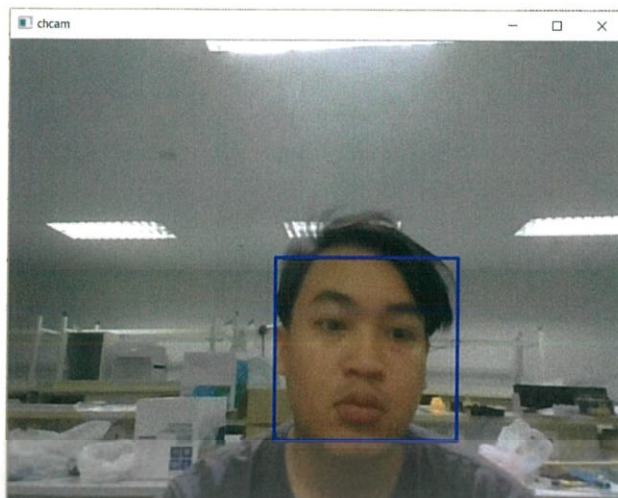
face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)

count = 1
while count < 31:
    (_, im) = webcam.read()
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 4)
    for (x, y, w, h) in faces:
        cv2.rectangle(im, (x, y), (x+w, y+h), (255, 0, 0), 2)
        face = gray[y:y + h, x:x + w]
        face_resize = cv2.resize(face, (width, height))
        cv2.imwrite('%s/%s.png' % (path, count), face_resize)
    count += 1

cv2.imshow('chcam', im)
key = cv2.waitKey(150)
if key == 27:
    break
```

รูปที่ 3.8 การสร้างข้อมูลรูปภาพด้วยฟังก์ชัน face\_cascade

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

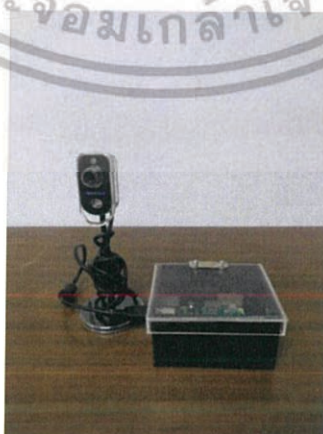


รูปที่ 3.9 ตัวอย่างตอนเปิดกล้อง (2)



รูปที่ 3.10 ภาพหลังจากทำการถ่ายภาพออกมา 30 ภาพ

3.4.3 เชื่อมต่อกล้อง Webcam กับบอร์ด Raspberry Pi 3 Model B+  
ทำการต่อกล้อง Webcam เข้ากับบอร์ด Raspberry Pi 3 Model B+ ดังรูปที่ 3.11

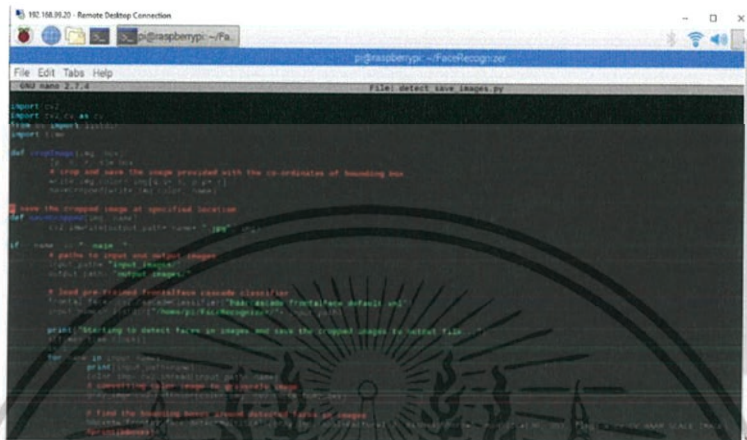


รูปที่ 3.11 การเชื่อมต่อกล้อง Webcam กับบอร์ด Raspberry Pi 3 Model B+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.4 การเขียนโปรแกรม Python

ส่วนที่ 1 เป็นโปรแกรมที่จะนำข้อมูลรูปภาพใน Folder Input เพื่อสร้างฐานข้อมูลสำหรับการนำไปใช้เป็นรูปภาพตัวอย่างในการเปรียบเทียบตอนเปิดใช้เป็นกล้องตรวจจับแบบ Real – Time โดยระบบจะทำการตัดภาพส่วนเกินออกและเก็บไว้เฉพาะใบหน้า ได้ออกมาเป็นรูปภาพใน Folder Output ดังรูปที่ 3.12



```

import cv2
import dlib as dl
import sys
import os

def crop_image(image, bounding_box):
    """Crop the image based on the bounding box coordinates"""
    x1, y1, x2, y2 = bounding_box
    return image[y1:y2, x1:x2]

def find_faces(image):
    """Find faces in the image using dlib's face detector"""
    detector = dl.get_frontal_face_detector()
    faces = detector(image)
    return faces

def crop_faces(image, faces):
    """Crop faces from the image"""
    cropped_faces = []
    for face in faces:
        cropped_face = crop_image(image, face)
        cropped_faces.append(cropped_face)
    return cropped_faces

def main():
    # Get input image path
    input_image = sys.argv[1]

    # Load the image
    image = cv2.imread(input_image)

    # Find faces in the image
    faces = find_faces(image)

    # Crop faces from the image
    cropped_faces = crop_faces(image, faces)

    # Save the cropped faces to the output folder
    output_folder = sys.argv[2]
    for i, cropped_face in enumerate(cropped_faces):
        output_path = os.path.join(output_folder, f'face_{i}.jpg')
        cv2.imwrite(output_path, cropped_face)

if __name__ == '__main__':
    main()
  
```

รูปที่ 3.12 โปรแกรมในส่วนของการสร้างฐานข้อมูล

ส่วนที่ 2 เป็นโปรแกรม Training ที่จะนำภาพจากส่วนที่ 1 มาให้บอร์ดได้เรียนรู้และจดจำโดยใช้ฟังก์ชัน Face Recognition ซึ่งเป็นฟังก์ชันใน Library OpenCV ดังรูปที่ 3.13



```

import cv2
import face_recognition as fr
import sys
import os

def load_image(image_path):
    """Load an image from the given path"""
    return cv2.imread(image_path)

def find_faces(image):
    """Find faces in the image using OpenCV's face_recognition module"""
    faces = fr.face_locations(image)
    return faces

def crop_faces(image, faces):
    """Crop faces from the image"""
    cropped_faces = []
    for face in faces:
        cropped_face = crop_image(image, face)
        cropped_faces.append(cropped_face)
    return cropped_faces

def main():
    # Get input image path
    input_image = sys.argv[1]

    # Load the image
    image = load_image(input_image)

    # Find faces in the image
    faces = find_faces(image)

    # Crop faces from the image
    cropped_faces = crop_faces(image, faces)

    # Save the cropped faces to the output folder
    output_folder = sys.argv[2]
    for i, cropped_face in enumerate(cropped_faces):
        output_path = os.path.join(output_folder, f'face_{i}.jpg')
        cv2.imwrite(output_path, cropped_face)

if __name__ == '__main__':
    main()
  
```

รูปที่ 3.13 โปรแกรม Training

ส่วนที่ 3 เป็นโปรแกรมที่จะทำการเปิดการทำงานกล้อง Webcam แบบ Real – Time โดยกล้องจะทำการตรวจจับหาใบหน้า แล้วนำมาเปรียบเทียบกับฐานข้อมูล หากพบที่ไม่ตรงกับฐานข้อมูล ก็จะทำการแจ้งเตือนผ่านแอปพลิเคชัน LINE และ Facebook ดังรูปที่ 3.14 ถึงรูปที่ 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 โปรแกรมในส่วนเปิดการทำงานกล้อง Webcam แบบ Real – Time (1)



รูปที่ 3.15 โปรแกรมในส่วนเปิดการทำงานกล้อง Webcam แบบ Real – Time (2)



รูปที่ 3.16 โปรแกรมในส่วนเปิดการทำงานกล้อง Webcam แบบ Real – Time (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 3.20 การทดลองการตรวจจับใบหน้า (2)

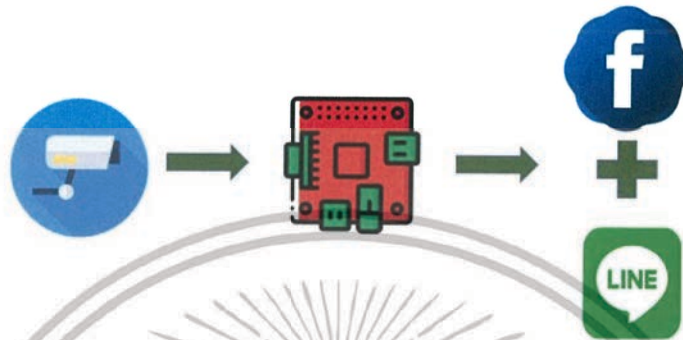


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

#### 4.1 แผนผังการทำงานของระบบตรวจจับใบหน้า



รูปที่ 4.1 แผนผังการทำงานของระบบตรวจจับใบหน้า

#### 4.2 การทดลองการรู้จำใบหน้า

ทำการทดลองโดยให้ผู้ใช้งานที่มีรูปในฐานข้อมูลเดินผ่านมายืนหน้ากล้อง และนั่งหน้ากล้อง เพื่อทำการจับเวลาในการประมวลผลของ Raspberry Pi ซึ่งจะทำการทดลองใส่หมวกและแว่นตา หลังจากนั้นจะส่งการแจ้งเตือนไปยังเว็บไซต์ Facebook และแอปพลิเคชัน LINE ต่อไป จึงได้ทำการทดลองทั้งหมด 10 คน ซึ่งแบ่งเป็น

ผู้ใช้งานที่มีชื่อในฐานข้อมูลจำนวน 2 คน

บุคคลแปลกหน้าจำนวน 8 คน

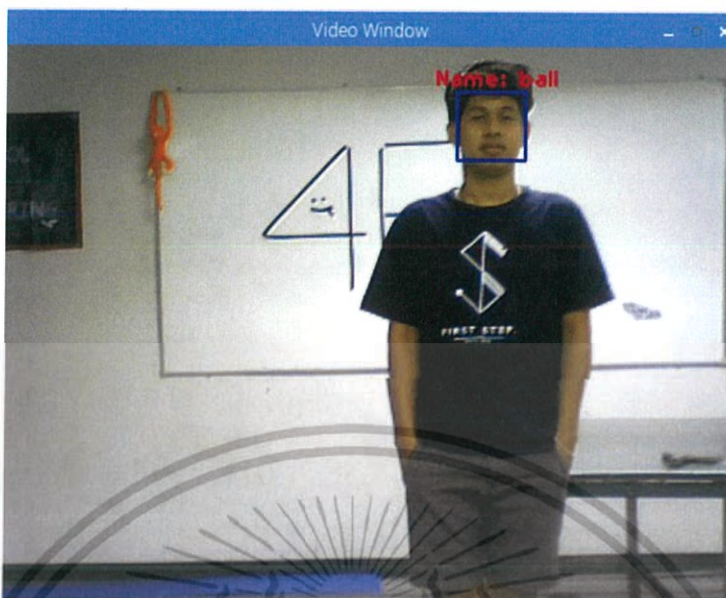
ซึ่งได้ทำการทดลองดังรูปที่ 4.2 ถึงรูปที่ 4.5



รูปที่ 4.2 การติดตั้งอุปกรณ์

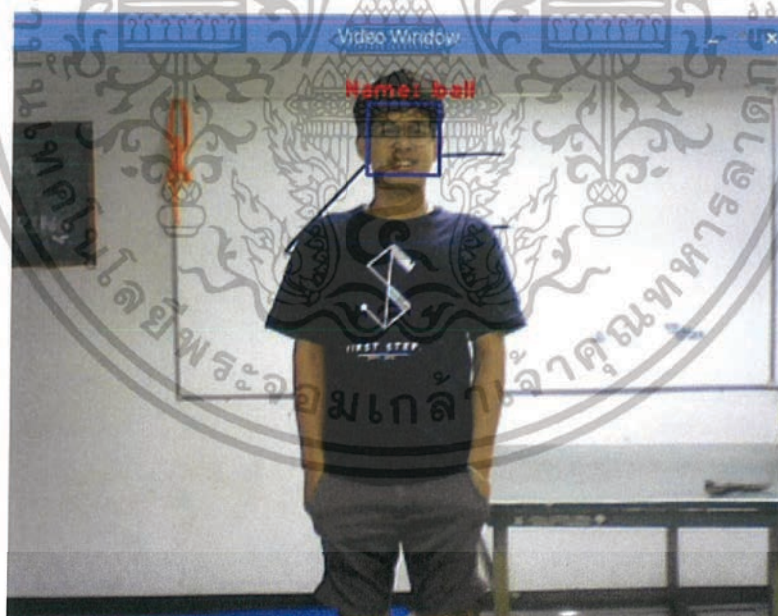
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.1 ผู้ใช้งานปกติ



รูปที่ 4.3 การทดลองโดยใช้ผู้ใช้งานปกติ

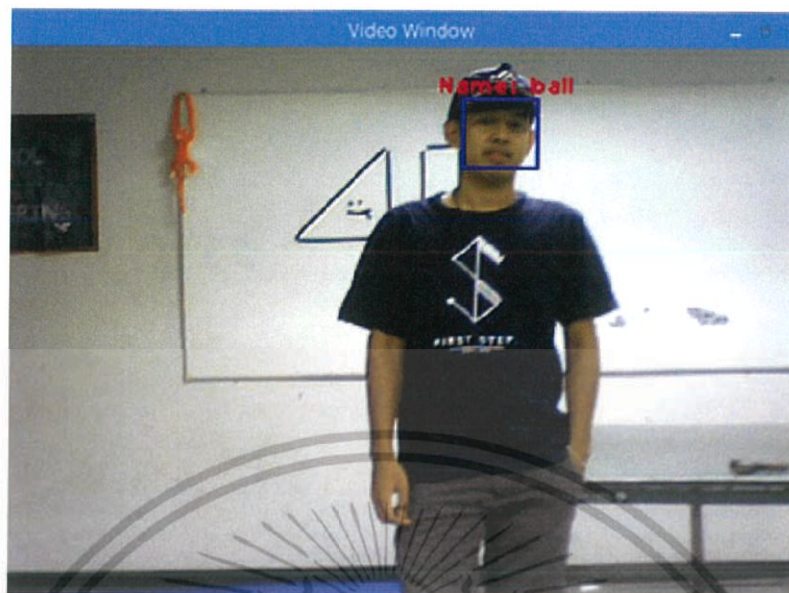
#### 4.2.2 ผู้ใช้งานสวมแว่น



รูปที่ 4.4 การทดลองโดยใช้ผู้ใช้งานสวมแว่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.2.3 ผู้ใช้งานสวมหมวก



รูปที่ 4.5 การทดลองโดยใช้ผู้ใช้งานสวมหมวก

### 4.2.4 บุคคลภายนอก



รูปที่ 4.6 การทดลองโดยใช้บุคคลภายนอก

เมื่อการทดลองให้ผู้ใช้งานแบบปกติ สวมแว่น และใส่หมวก จะเห็นว่า Raspberry Pi สามารถทำการวิเคราะห์แยกใบหน้าได้หมด และยังจับเวลาในการประมวลผลเปรียบเทียบระหว่างบุคคลภายนอกกับผู้ใช้งานจริง ซึ่งเวลาในการประมวลผลนั้น ยิ่งเวลาประมวลผลมากจะทำให้มีประสิทธิภาพมากขึ้น ดังตารางที่ 4.1 และตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผู้ใช้งานในฐานะข้อมูล

ครั้งที่	ระยะเวลาการประมวลผล (วินาที)	ผลการประมวล	แจ้งเตือน	ไม่แจ้งเตือน
1	10	ไม่รู้	✓	
2	14	มี		✓
3	13	มี		✓
4	15	มี		✓
5	12	มี		✓
6	11	ไม่รู้	✓	
7	14	มี		✓
8	11	ไม่รู้	✓	
9	13	มี		✓
10	13	มี		✓

ตารางที่ 4.2 บุคคลภายนอกในฐานะข้อมูล

ครั้งที่	ระยะเวลาการประมวลผล (วินาที)	ผลการประมวล	แจ้งเตือน	ไม่แจ้งเตือน
1	13	ไม่รู้	✓	
2	14	มี		✓
3	9	ไม่รู้	✓	
4	10	ไม่รู้	✓	
5	12	ไม่รู้	✓	
6	15	ไม่รู้	✓	
7	10	มี		✓
8	9	ไม่รู้	✓	
9	11	ไม่รู้	✓	
10	11	ไม่รู้	✓	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 การทดลองส่งข้อมูลแจ้งเตือนผลทาง Facebook และ LINE

จากการทดลองพบว่าเมื่อให้ตัวกล้องจับภาพและต่อกับบอร์ด Raspberry Pi ได้ทำการเชื่อมต่อ Wi - Fi แล้วนั้นจะสามารถเชื่อมต่อ การแจ้งเตือนข้อมูลขึ้นทาง Facebook และ LINE ได้ อย่างเป็นปกติ แล้วมีการรับ - ส่งค่าเป็น Real - Time ดังรูปที่ 4.6 และรูปที่ 4.7 ตามลำดับ



รูปที่ 4.7 การแจ้งเตือนผ่าน Facebook



รูปที่ 4.8 การแจ้งเตือนผ่าน LINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# สรุปผลการทดลองและข้อเสนอแนะ

### 5.1 สรุปผลการทดลอง

จากการดำเนินงานตามขั้นตอนที่ได้กล่าวมานั้น เริ่มจากขั้นตอนการศึกษาหาข้อมูลและหลักการ พร้อมทำความเข้าใจกับขอบเขตของโครงการ ดังนั้นจะแบ่งส่วนของการดำเนินงานเป็นทั้งหมด 2 ส่วน คือ

1. ส่วนของการประดิษฐ์กล่องควบคุม (Hardware) ในส่วนนี้ได้ทำการศึกษาการทำงานของ Raspberry Pi เพื่อที่จะได้นำไปใช้งานได้อย่างสะดวก เหมาะสมตามสถานที่

2. ส่วนของการเขียนโปรแกรม (Software) เป็นส่วนของการเขียนโปรแกรมเพื่อที่จะสามารถนำข้อมูล Input หรือวิดีโอจากหน้ากล้องมาทำการวิเคราะห์ผ่าน Classifiers ที่ได้จัดทำไว้ หลังจากนั้นนำข้อมูลรูปภาพไปตรวจสอบในฐานข้อมูลว่าตรงตามฐานข้อมูลหรือไม่ แล้วแจ้งเตือนผ่านทางเว็บไซต์ Facebook และ แจ้งเตือนผ่าน LINE แบบ Real - Time

หลังจากการออกแบบกล่องควบคุมและส่วนของโปรแกรมเรียบร้อยแล้ว ได้มีการนำไปทดลอง ตรวจสอบใบหน้าจากเหตุการณ์จริง พบว่าสามารถแยกและรู้จำใบหน้าของบุคคลได้เป็นอย่างดี และแสดงข้อมูลผ่านออกทางหน้าเว็บไซต์แบบ Real - Time ได้ทั้งในรูปแบบตัวเลขข้อมูลรูปภาพและตัวอักษรได้เป็นอย่างดี ทั้งนี้ต้องขอขอบคุณผู้ที่เกี่ยวข้องกับการดำเนินงานทุกคน ไม่ว่าจะเป็นเพื่อนร่วมทีม เพื่อนๆ อาจารย์ ที่ปรึกษา และสื่อการสอนต่างๆ ที่มีส่วนช่วยในการให้คำปรึกษา ข้อเสนอแนะ ตลอดจนความช่วยเหลือต่างๆ จนทำให้โครงการสำเร็จลุล่วงไปได้เป็นอย่างดี

### 5.2 ปัญหาและอุปสรรคในการดำเนินงาน

1. เนื่องจากการเขียนโปรแกรมส่วนใหญ่ต้องอาศัยความรู้จาก Pregaming ค่อนข้างสูงอีกทั้งผนวกกับความรู้ที่ได้เรียนมาจึงทำให้เกิดความล่าช้าในการทำงานได้

2. เนื่องจากเพื่อความแม่นยำในการวิเคราะห์ และตรวจสอบใบหน้าจึงจำเป็นต้องมีการเพิ่มรูปในฐานข้อมูลค่อนข้างเยอะ จึงทำให้ใช้เวลามากในการทำงานแต่ละครั้ง

3. ระยะเวลาพักของกล้องค่อนข้างต่ำ จึงทำให้เวลาผู้ใช้งานผ่านมากเกินระยะที่กำหนดจะทำให้เกิดความผิดพลาดในการวิเคราะห์ใบหน้าเพิ่มขึ้น

4. การเชื่อมต่ออินเทอร์เน็ตที่ไม่เสถียรมีผลต่อการแจ้งเตือนผ่าน Facebook และ LINE โดยใช้เวลาเพิ่มมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 แนวทางการแก้ไข

1. ศึกษาและสอบถามแนวทางเพิ่มเติมจากสื่อต่างๆ
2. ใช้หน่วยประมวลผลที่มีความเร็วเพิ่มขึ้น
3. ใช้กล่องที่มีการปรับระยะโฟกัสที่เพิ่มขึ้นได้
4. ใช้การเชื่อมต่ออินเทอร์เน็ตผ่านสาย Lan แทนการเชื่อมต่อผ่าน Wi - Fi



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] KOKZARD. 2561. Face recognition algorithm. [Online].  
Available : <http://kokzard.blogspot.com/2012/04/face-recognition-algorithm.html>
- [2] Tamemo. 2561. รวม syntax พื้นฐานของภาษา Python ฉบับรวบรัด. [Online].  
Available : <https://www.tamemo.com/post/137/python-basic-syntax/>
- [3] Goragod. 2560. การขอ Token สำหรับ LINE Notify. [Online].  
Available : [https://www.gcms.in.th/line\\_notify.html](https://www.gcms.in.th/line_notify.html)
- [4] Divyansh Dwivedi. 2561. Face Recognition for Beginners. [Online].  
Available : <https://towardsdatascience.com/face-recognition-for-beginners>
- [5] Facebook Developers. 2558. คู่มือ Graph API Explorer. [Online]  
Available : <https://developers.facebook.com/docs/graph-api/explorer/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### โปรแกรม

```
import cv2
import cv2.cv as cv
from os import listdir
import time

def cropImage(img, box):
    [p, q, r, s]= box
    # crop and save the image provided with the co-ordinates of bounding box
    write_img_color= img[q:q+ s, p:p+ r]
    saveCropped(write_img_color, name)

# save the cropped image at specified location
def saveCropped(img, name):
    cv2.imwrite(output_path+ name+ ".jpg", img)

if __name__ == "__main__":
    # paths to input and output images
    input_path= "input_images/"
    output_path= "output_images/"
```

#### รูปที่ ก.1 โปรแกรม detect save images (1)

```
for box in bBoxes:
    # crop the image
    cropImage(color_img, box)
    i+= 1

print("Successfully completed the task in %.2f Secs." % (time.clock()- sttime))
```

#### รูปที่ ก.2 โปรแกรม detect save images (2)

```
# load pre-trained frontal face cascade file
frontal_face= cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
input_names= listdir("/home/pi/FaceRecognizer/"+ input_path)

print("Starting to detect faces in images and save the cropped images to output file...")
stime= time.clock()
i= 1
for name in input_names:
    print(input_path+ name)
    color_img= cv2.imread(input_path+ name)
    # converting color image to grayscale image
    gray_img= cv2.cvtColor(color_img, cv2.COLOR_BGR2GRAY)

    # find the bounding boxes of detected faces in image
    bBoxes= frontal_face.detectMultiScale(gray_img, scaleFactor=1.3, minNeighbors=5, minSize=(30, 30),
    #print(bBoxes)
```

#### รูปที่ ก.3 โปรแกรม detect save images (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import cv2
import cv2.cv as cv
import numpy as np
from os import listdir
import sys, time

def get_images(path, size):
    path: path to a folder which contains subfolders of for each subject/person
        which in turn contains pictures of subjects/persons.

    size: a tuple to resize images.
        Ex- (256, 256)
    ...
    sub= 0
    images, labels= [], []
    people= []

    for subdir in listdir(path):

```

#### รูปที่ ก.4 โปรแกรม detect recongnition (1)

```

labels= np.asarray(labels, dtype= np.int32)

# Initializing eigen_model
print("Initializing eigen FaceRecognizer and training...")
stime= time.clock()
eigen_model= cv2.createEigenFaceRecognizer()
eigen_model.train(images, labels)
print("\tSuccessfully completed training in "+ str(time.clock()- stime)+ " Secs!")

# Starting to detect & recognize
for image_name in listdir(sys.argv[2]):
    try:
        color_image= cv2.imread(sys.argv[2]+ "/" + image_name)
        [x, y]= color_image.shape[:2]
        x_factor= (float(y)/4)
        resize_y= 480

        color_image= cv2.resize(color_image, (int(resize_y* x_factor), resize_y))
        pre_image= cv2.imread(sys.argv[1]+ "/" + image_name, cv2.IMREAD_GRAYSCALE)

        for image in listdir(path+ "/" + subdir):
            img= cv2.imread(path+ "/" +subdir+ "/" +image, cv2.IMREAD_GRAYSCALE)
            img= cv2.resize(img, size)

            images.append(np.asarray(img, dtype= np.uint8))
            labels.append(sub)

        people.append(subdir)
        sub+= 1

    return [images, labels, people]

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Wrong number of arguments! See the usage\n")
        print("Usage: face_recognition.py <full/path/to/root/images/folder> <full/path/to/images/1")

```

#### รูปที่ ก.5 โปรแกรม detect recongnition (2)

#### รูปที่ ก.6 โปรแกรม detect recongnition (3)

```

for bbox in bboxes:
    (p,q,r,s)= bbox
    cv2.rectangle(color_image, (p,q), (p+r,q+s), (225,0,25), 2)
    #cv2.rectangle(pre_image, (p,q), (p+r,q+s), (25,0,225), 2)

    #starting to predict subject/ person in cropped (detected) image
    pre_crop_image= pre_image[q:q+s, p:p+r]
    pre_crop_image= cv2.resize(pre_crop_image, (256, 256))

    [predicted_label, predicted_conf]= eigen_model.predict(np.asarray(pre_crop_image))
    print("Predicted person in the image "+ image_name+ " : "+ people[predicted_label])

    box_text= format("Subject: "+ people[predicted_label])
    #print(box_text)

    cv2.putText(color_image, box_text, (p-20, q-5), cv2.FONT_HERSHEY_PLAIN, 1.1, (25,0,225), 1)

```

#### รูปที่ ก.7 โปรแกรม detect recongnition (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

a = 0
def get_images(path, size):
    '''
    path: path to a folder which contains subfolders of for each subject/person
        which in turn contains pictures of subjects/persons.

    size: a tuple to resize images.
        ... Ex- (256, 256)
    ...
    sub= 0
    images, labels= [], []
    people= []

    for subdir in os.listdir(path):
        for image in os.listdir(path+ "/" + subdir):
            #print(subdir, images)
            img= cv2.imread(path+os.path.sep+subdir+os.path.sep+image, cv2.IMREAD_GRAYSCALE)
            img= cv2.resize(img, size)

```

รูปที่ ก.8 โปรแกรม detect video (1)

```

import os
import sys, time
import requests, facebook

confidence = 10000
scan_retry = 2

a = 0
def get_images(path, size):
    '''
    path: path to a folder which contains subfolders of for each subject/person
        which in turn contains pictures of subjects/persons.

    size: a tuple to resize images.
        ... Ex- (256, 256)
    ...
    sub= 0
    images, labels= [], []
    people= []

def get_images(path, size):
    '''
    path: path to a folder which contains subfolders of for each subject/person
        which in turn contains pictures of subjects/persons.

    size: a tuple to resize images.
        ... Ex- (256, 256)
    ...
    sub= 0

images, labels= [], []
people= []

    for subdir in os.listdir(path):
        for image in os.listdir(path+ "/" + subdir):
            #print(subdir, images)
            img= cv2.imread(path+os.path.sep+subdir+os.path.sep+image, cv2.IMREAD_GRAYSCALE)
            img= cv2.resize(img, size)

            images.append(np.asarray(img, dtype= np.uint8))

```

รูปที่ ก.9 โปรแกรม detect video (2)

รูปที่ ก.10 โปรแกรม detect video (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def train_model(path):
    """
    Takes path to images and train a face recognition model
    Returns trained model and people
    """
    [images, labels, people]= get_images(sys.argv[1], (256, 256))
    #print([images, labels])

    labels= np.asarray(labels, dtype= np.int32)

    # initializing eigen_model and training
    print("Initializing eigen FaceRecognizer and training...")
    sttime= time.clock()
    eigen_model= cv2.createEigenFaceRecognizer()
    eigen_model.train(images, labels)
    print("\tSuccessfully completed training in "+ str(time.clock()- sttime)+ " Secs!")

    return [eigen_model, people]

```

### รูปที่ ก.11 โปรแกรม detect video (4)

```

def send_message(r,picture_name):
    msg = ' LINE Notify'
    notifyFile('./frame.jpg')
    notifySticker(3,1)
    lineNotify('มีผู้บุกรุก ตรวจพบโดยระบบ')
def post_on_facebook(intruder, counter, picture_name):
    """
    Takes name of intruder and posts on your facebook timeline.
    You need to get access_token from facebook GraphAPI and paste it below
    """
    # has a life time of 1 hour
    token= "EAAAAAYsX7TsBAHMLBtBmCBglghE:580bnhP91cYSLqP9BGeN5BLra67PUFrN8ktYPm6zokYKdE3B74G13LdGYZBhsnfCml"
    url= "https://graph.facebook.com/me/feed"

    graph= facebook.GraphAPI(access_token= token)

    my_message1= "การแจ้งเตือนระบบแจ้งเตือน "+ intruder+ " "
    my_message2= "ถ้าพบเห็นหรือสงสัยกรุณาแจ้งต่อฝ่ายไอทีของระบบ"
    final_message= my_message1+"\n"+my_message2+"\n"

```

### รูปที่ ก.12 โปรแกรม detect video (5)

```

if __name__ == "__main__":
    if len(sys.argv)!= 2:
        print("Wrong number of arguments! See the usage.\n")
        print("Usage: face_detect_video.py <full/path/to/root/images/folder>")
        sys.exit()

    arg_one= sys.argv[1]
    eigen_model, people= train_model(arg_one)

    #start video capture
    cap= cv2.VideoCapture(0)

    counter= 0
    last_20= [-1 for i in range(10)]
    final_5= []
    box_text= "Name: "

```

### รูปที่ ก.13 โปรแกรม detect video (6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(True):
    ret, frame= cap.read()
    if ret is True:
        gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        gray_frame = cv2.equalizeHist(gray_frame)

        bBoxes= detect_faces(gray_frame)

    for bBox in bBoxes:
        (p,q,r,s)= bBox
        cv2.rectangle(frame, (p,q), (p+r,q+s), (225,0,25), 2)

        crop_gray_frame= gray_frame[q:q+s, p:p+r]
        crop_gray_frame= cv2.resize(crop_gray_frame, (256, 256))

        [predicted_label, predicted_conf]= eigen_model.predict(np.asarray(crop_gray_frame))
        print(people[predicted_label],predicted_conf)

```

### รูปที่ ก.14 โปรแกรม detect video (7)

```

if counter> 20:
    print("Will post on facebook timeline if this counter reaches to "+str(scan_retry)+
    final_5.append(max_label) #it always takes the label into consideration
    if len(final_5)== scan_retry:
        final_label= majority(final_5)
        if final_label == -1:
            guest = "unknown"
            print("Posting on your facebook timeline...")
            picture_name= "frame.jpg"
            cv2.imwrite(picture_name, frame)
            post_on_facebook(guest, counter, picture_name)
            print("Posting on your line...")
            send_message(r, picture_name)
        else:
            guest = people[final_label]
            print("Intruder is "+ guest)
            final_5= []

    cv2.imshow("Video Window", frame)
    counter += 1

    if (cv2.waitKey(5) & 0xFF== 27):
        break

cv2.destroyAllWindows()

```

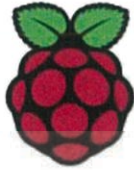
### รูปที่ ก.15 โปรแกรม detect video (8)

### รูปที่ ก.16 โปรแกรม detect video (9)

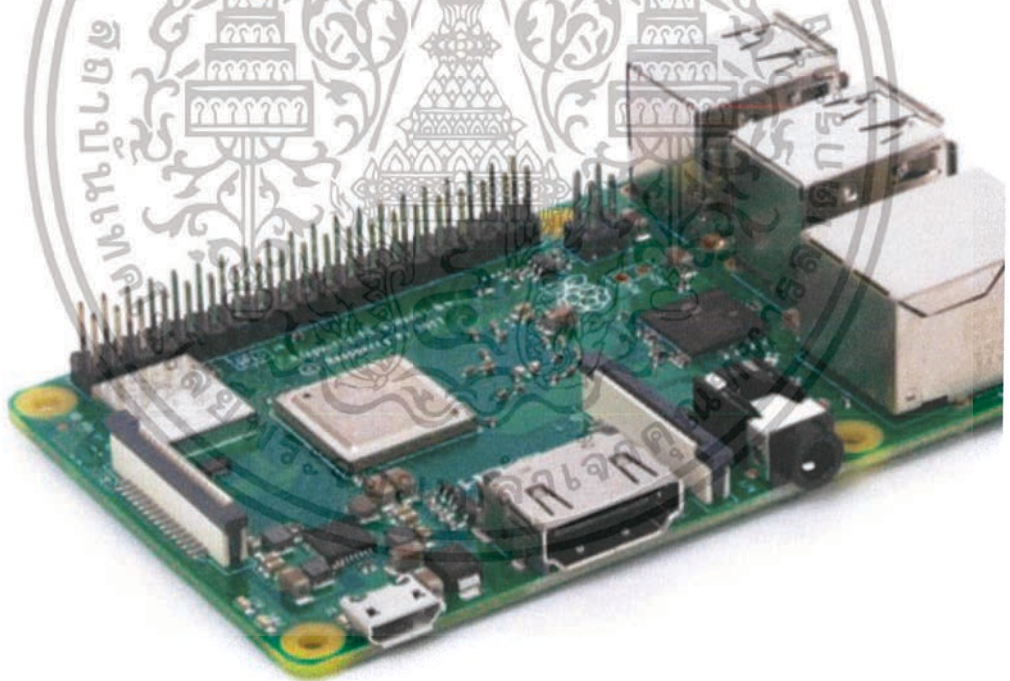
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

## Raspberry Pi 3 Model B+



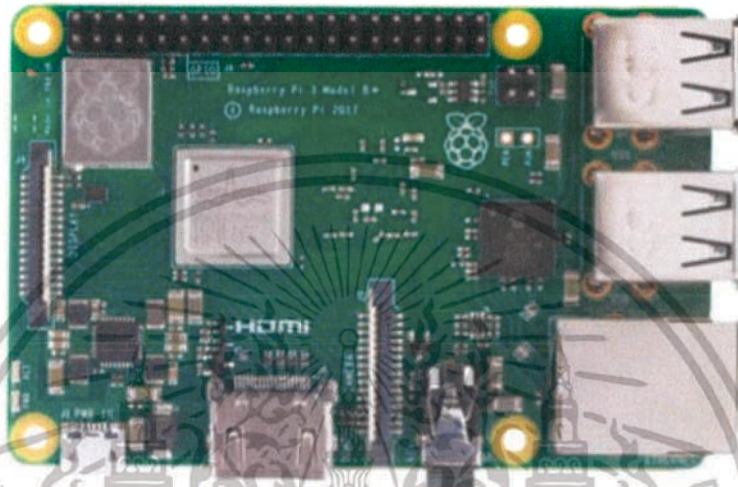
# Raspberry Pi 3 Model B+



รูปที่ ข.1 Raspberry Pi 3 Model B+ product brief (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Overview



The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi-3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT.

The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.

The Raspberry Pi 3 Model B+ maintains the same mechanical footprint as both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B.

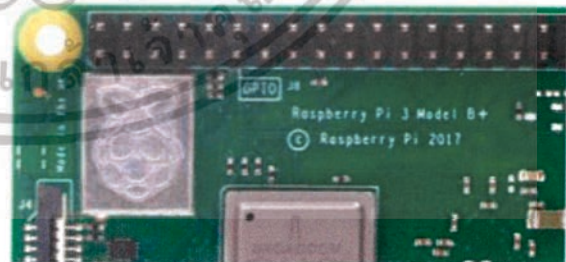


รูปที่ ข.2 Raspberry Pi 3 Model B+ product brief (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Specifications

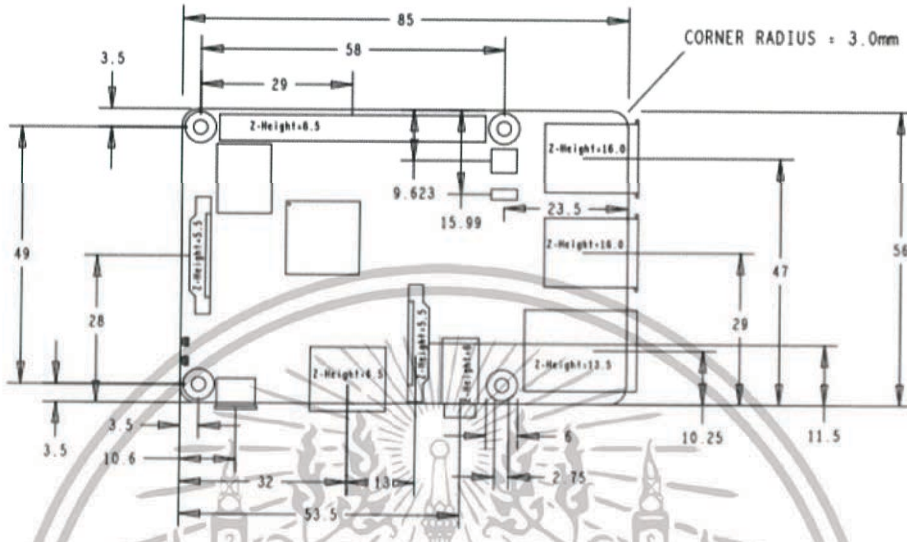
<b>Processor:</b>	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
<b>Memory:</b>	1GB LPDDR2 SDRAM
<b>Connectivity:</b>	<ul style="list-style-type: none"> <li>■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE</li> <li>■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)</li> <li>■ 4 × USB 2.0 ports</li> </ul>
<b>Access:</b>	Extended 40-pin GPIO header
<b>Video &amp; sound:</b>	<ul style="list-style-type: none"> <li>■ 1 × full size HDMI</li> <li>■ MIPI DSI display port</li> <li>■ MIPI CSI camera port</li> <li>■ 4 pole stereo output and composite video port</li> </ul>
<b>Multimedia:</b>	H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30), OpenGL ES 1.1, 2.0 graphics
<b>SD card support:</b>	Micro SD format for loading operating system and data storage
<b>Input power:</b>	<ul style="list-style-type: none"> <li>■ 5V/2.5A DC via micro USB connector</li> <li>■ 5V DC via GPIO header</li> <li>■ Power over Ethernet (PoE)-enabled (requires separate PoE HAT)</li> </ul>
<b>Environment:</b>	Operating temperature, 0–50°C
<b>Compliance:</b>	For a full list of local and regional product approvals, please visit <a href="http://www.raspberrypi.org/products/raspberry-pi-3-model-b+">www.raspberrypi.org/products/raspberry-pi-3-model-b+</a>
<b>Production lifetime:</b>	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.



รูปที่ ข.3 Raspberry Pi 3 Model B+ product brief (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Physical specifications



### Warnings

- This product should only be connected to an external power supply rated at 5V/2.5A DC. Any external power supply used with the Raspberry Pi 3 Model B+ shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.
- Whilst in use, this product should be placed on a stable, flat, non-conductive surface and should not be contacted by conductive items.
- The connection of incompatible devices to the GPIO connection may affect compliance, result in damage to the unit, and invalidate the warranty.
- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors, and mice when used in conjunction with the Raspberry Pi.
- The cables and connectors of all peripherals used with this product must have adequate insulation so that relevant safety requirements are met.

### Safety instructions

To avoid malfunction of or damage to this product, please observe the following:

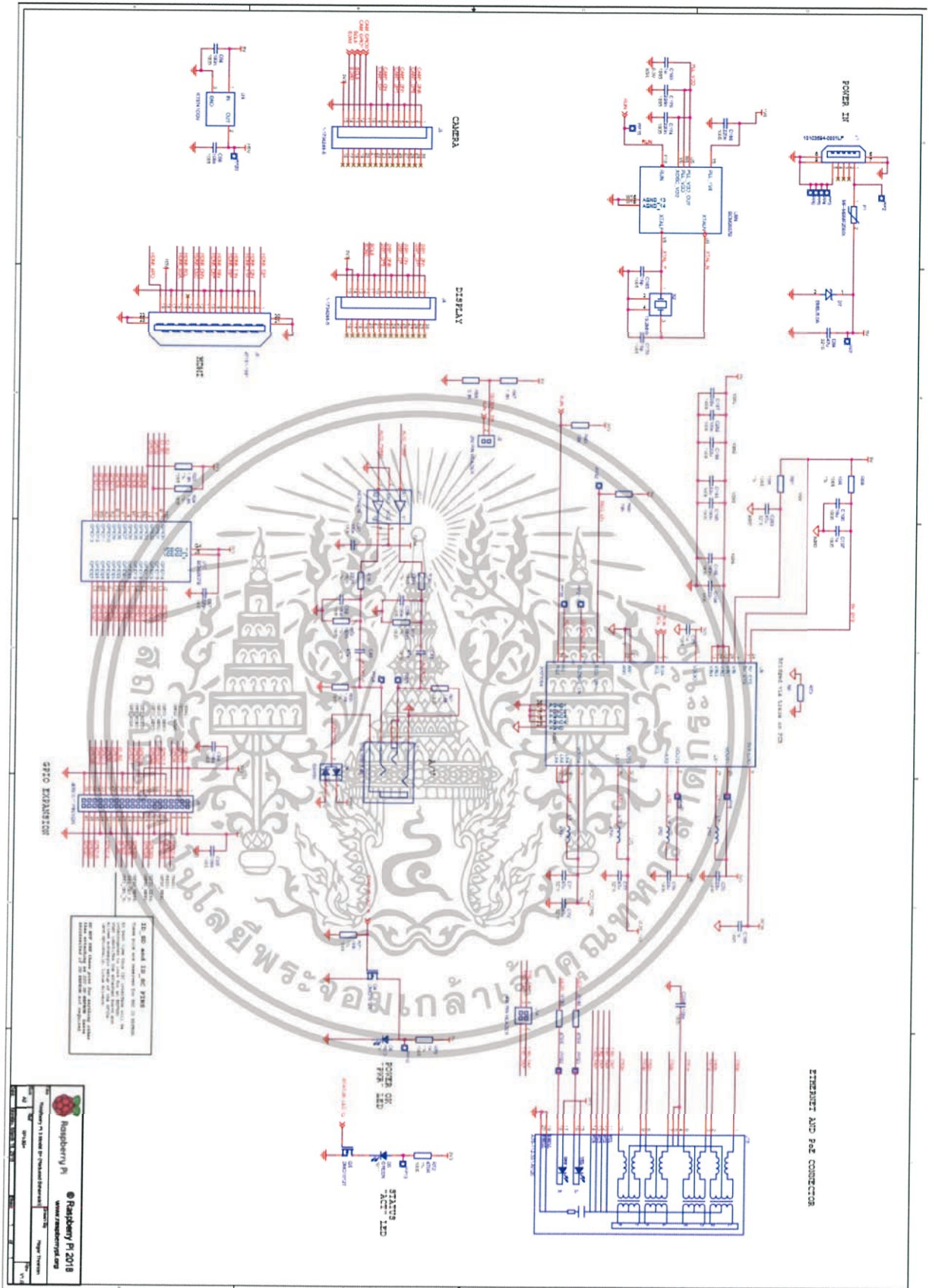
- Do not expose to water or moisture, or place on a conductive surface whilst in operation.
- Do not expose to heat from any source, the Raspberry Pi 3 Model B+ is designed for reliable operation at normal ambient temperatures.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Whilst it is powered, avoid handling the printed circuit board, or only handle it by the edges to minimise the risk of electrostatic discharge damage.



raspberrypi.org

### รูปที่ ข.4 Raspberry Pi 3 Model B+ product brief (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.5 Raspberry Pi 3 Model B+ schematic diagrams

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Face detection for security system using OpenCV

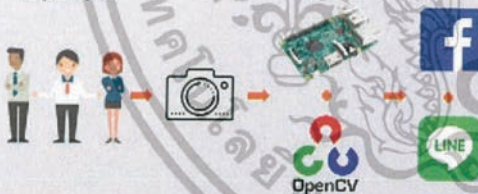
Chaowaphun Supawiriyakul and Nirun Keawmanee

## Abstract

Safety is very important to human living conditions. Human beings have developed a variety of security systems. Each format depends on the factors and the environment in which safety is needed. One device that is currently used is "CCTV", the reason for using it because the camera is small can be installed in different places, not difficult and can be seen in various areas. We wanted to use the Image Processing technology to analyze human faces for use with small cameras. The recognition systems will compare faces from the camera to the existing face database. When the comparison is completed, it will be displayed with notification via the application Facebook and Line.

## Introduction

To create a more effective security system. We use image processing technology to develop cameras to add functionality. Use the Raspberry pi3 board model B+ with fast processing, support for Wi-Fi, which allows real-time data transmission. We have an OpenCV which is a library in Python which has face detection and faces recognition functions used in image analysis.



## Methodology

Receive images from the camera, send data to the board raspberry pi 3 model B+. The board then processes this image using library OpenCV, which has a face recognition function to recognize faces stored as data and use face detection function to detect the person to compare with the information available and send the comparison results to the user via application Facebook and Line.

## Results

From the experiment, when it was detected that someone else entered the area. The camera will take pictures and send notifications via Facebook and Line.



## Conclusion

Image processing will require more processing time to increase accuracy and the faces of each person are similar, therefore resulting in a slight error. In conclusion, we can use image processing to accurately analyze faces in security systems.



## References

- [1] "Face Detection using Haar Cascades" [Online] Available : <https://netpie.io/tutorials>
- [2] "Image Processing with OpenCV" [Online] Available : <https://medium.com/@ariesitr/image-processing-with-opencv-45c3a5cefd10>
- [3] "A face recognition algorithm which notifies you on facebook in case of an Intruder attack" [Online] Available : <https://www.instructables.com/id/IntruderGuest-alert-system/>

