

กล่องสังเกตการณ์อัจฉริยะขนาดเล็ก

นางสาวดวงทอง คำรงเจริญ

นายวิสูตร      กุศลส่ง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตร์

สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2561

# Smart Trail Camera

Ms. Tuangthong Dumrongjalearn

Mr. Wisarut Kussonsong

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF BACHELOR OF ELECTRICAL ENGINEERING  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK  
ACADEMIC YEAR 2018

ปริญญานิพนธ์เรื่อง : กล้องสังเกตการณ์อัจฉริยะขนาดเล็ก  
ชื่อ : นางสาวดวงทอง คำรงเจริญ  
นายวิศรุต กุศลส่ง  
สาขา : วิศวกรรมไฟฟ้า  
ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
คณะ : วิศวกรรมศาสตร์  
อาจารย์ที่ปรึกษา : อาจารย์วัชรระ ภัคมาตร์  
ปีการศึกษา : 2561


คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ อนุมัติให้  
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขา  
วิศวกรรมไฟฟ้า

  
.....  
(ผู้ช่วยศาสตราจารย์ ดร.นภดล วิวัชรโกเศศ)

หัวหน้าภาควิชาวิศวกรรมไฟฟ้า  
และคอมพิวเตอร์

  
.....  
(อาจารย์วัชรระ ภัคมาตร์)

ประธานกรรมการ

  
.....  
(ผู้ช่วยศาสตราจารย์สิทธิพร เกิดสำอางค์)

กรรมการ

  
.....  
(รองศาสตราจารย์ ดร.เอกรัฐ บุญญา )

กรรมการ

ลิขสิทธิ์ของภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

Project Report Title : Smart Trail Camera

Name : Ms. Tuangthong Dumrongjalearn  
Mr. Wisarut Kussonsong

Major Field : Electrical Engineering

Department : Electrical and Computer Engineering

Faculty : Engineering

Project Advisor : Mr. Watchara Pakkamart

Academic Year : 2018

Accepted by the Faculty of Engineering, King Mongkut's University of Technology North Bangkok in Partial Fulfillment of the Requirement for the Degree of Bachelor of Electrical Engineering.

.....  
(Asst. Prof. Dr. Nopadon Wiwatcharagoses)

*W. Pakkamart*  
..... Chairperson  
(Mr. Watchara Pakkamart)

S. Kerdsun-ang Member  
(Asst. Prof. Sithiporn Kerdsun-ang)

A. Boonpoonga Member  
(Assoc. Prof. Dr. Akkarat Boonpoonga)

Copyright of the Department of Electrical and Computer Engineering, Faculty of Engineering  
King Mongkut's University of Technology North Bangkok

## บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการจัดทำระบบถ่ายภาพนิ่งที่สามารถประมวลผลตรวจจับใบหน้าบุคคลรวมไปถึงถ่ายทอดสดวิดีโอผ่านระบบอินเทอร์เน็ตด้วยบอร์ดราสเบอร์รี่พาย ซึ่งเป็นบอร์ดคอมพิวเตอร์ขนาดเล็กใช้งานร่วมกับเซ็นเซอร์ตรวจจับความเคลื่อนไหวและเว็บแคม เพื่อทำการถ่ายภาพแบบ Snapshot และนำมาประมวลผลรวมไปถึงการถ่ายทอดสดผ่าน Streaming server โดยทั้งหมดนี้จะถูกนำไปแสดงผลทั้งภาพนิ่งและวิดีโอใน Web server อีกทั้งภาพนิ่งจะถูกทำการสำรองข้อมูลบนเว็บฝากไฟล์ ประเภท Google drive เพื่อป้องกันการสูญหายและสามารถย้อนหลังซึ่งระบบการถ่ายภาพ ประมวลผลภาพนิ่งและการถ่ายทอดสดสามารถนำไปใช้ป้องกันการบุกรุกการโจรกรรม และความปลอดภัยต่าง ๆ ในทรัพย์สิน รวมทั้งยังไปประยุกต์ใช้ในการตรวจสอบการถ่ายทำงานต่าง ๆ ทั้งอุปกรณ์เครื่องจักรหรือตัวบุคคลได้อีกด้วย

## **Abstract**

This project represents systems with organism photography, processed to detection face of human and VDO Streaming on internet by Raspberry Pi 3 model b+. Raspberry Pi is a small computer board. Also, connect with PIR motion sensor to detect organism and webcam will take snapshot photo and send image to process and VDO streaming by streaming server. All of this will take show image and video on web servers and backup in Google Drive for protect lost and to look backward. For the image processing and VDO streaming can be protect invasion, depredation and safety in your asset as well as to apply in the examination of various work. Both of machine and person, too.

## กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ได้มีการออกแบบ พัฒนา ดำเนินการทดลองตลอดจนทดสอบเพื่อใช้งานได้สำเร็จลุล่วงไปด้วยดีตามความคาดหวังของผู้เขียนนั้นเพราะความเมตตากรุณาของอาจารย์ที่ปรึกษาและความช่วยเหลือจากหลาย ๆ ฝ่ายในการทำงานรวมถึงกำลังใจอันดีจากเพื่อน ๆ บิดา มารดา และครอบครัวของผู้เขียน

ขอกราบขอบพระคุณ อาจารย์วัชร ภัคมาตร์ ที่กรุณารับเป็นที่ปรึกษาและเสียสละเวลาอันมีค่าในการให้คำปรึกษาเกี่ยวกับ แนวคิด และคอยรับฟังปัญหาที่เกิดขึ้นในการดำเนินงานครั้งนี้คอยแนะแนววิธีการแก้ไขปัญหา กระบวนการคิด รวมไปถึงคำแนะนำที่ดีทั้งในทางวิชาการและ การทำงานตลอดมา

ขอขอบคุณเพื่อนนักศึกษาคณะวิศวกรรมศาสตร์ที่ได้ให้ความช่วยเหลือแนะแนว การออกแบบและการจัดระบบ เป็นที่ปรึกษาคอยรับฟังปัญหาต่าง ๆ

ขอขอบพระคุณคณะกรรมการทุกท่านที่ให้ความกรุณาเป็นกรรมการในการคุมสอบและเสนอปัญหาและข้อบกพร่องต่าง ๆ เพื่อที่จะได้นำไปพัฒนาและแก้ไขปรับปรุง ตลอดจนอาจารย์ทุกท่านที่ได้ให้ความรู้ และอบรมสั่งสอนในการประยุกต์ใช้ความรู้ รวมถึงชี้แนะ แนวความคิดที่ลึกซึ้งตลอดจนประสบการณ์อันมีค่าของท่านอาจารย์ทุกท่านในการใช้ชีวิตครั้งนี้

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ และคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่คอยติดต่อประสานงานและให้ความสะดวกตลอดมาในการดำเนินงานต่าง ๆ

ขอกราบขอบพระคุณ บิดา มารดาและครอบครัวของผู้เขียนปริญญานิพนธ์นี้ทุกท่าน ที่คอยให้การสนับสนุนด้านการเรียนและคอยให้กำลังใจทุกเวลาที่เหนื่อยล้าจนการเรียนผ่านพ้นไปด้วยดีในการทำปริญญานิพนธ์ครั้งนี้จนกระทั่งประสบความสำเร็จไปได้ด้วยดี

ขอขอบพระคุณแหล่งความรู้ต่าง ๆ ที่เกี่ยวกับวิทยานิพนธ์ครั้งนี้ ที่เป็นแหล่งความรู้ในการเขียนโปรแกรมเพื่อให้การทำงานสามารถดำเนินไปได้ด้วยดี ถ้าหากปริญญานิพนธ์เล่มนี้มีข้อผิดพลาดประการใด ทางผู้จัดต้องขออภัยมา ณ ที่นี้ด้วย

ดวงทอง คำรงเจริญ

วิศรุต กุศลส่ง

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	จ
บทคัดย่อภาษาอังกฤษ	ฉ
กิตติกรรมประกาศ	ช
สารบัญภาพ	ญ
บทที่ 1. บทนำ	
1.1 วัตถุประสงค์ของโครงการ	1
1.2 ขอบเขตของการทำโครงการ	1
1.3 แนวเหตุผล ทฤษฎีสำคัญหรือสมมุติฐาน	1
1.4 ผลประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2. ทฤษฎีที่เกี่ยวข้อง	3
2.1 บอร์ดราสเบอร์รี่พาย (Raspberry Pi)	3
2.2 เซ็นเซอร์ตรวจจับความเคลื่อนไหว	4
2.3 การเปลี่ยนภาพสีเป็นภาพระดับสีเทา	6
2.4 การตรวจจับใบหน้า	6
2.5 วิธีการตรวจจับใบหน้าของ Viola-Jones	7
2.6 เว็บเซิร์ฟเวอร์ Apache	12
2.7 phpMyadmin	12
2.8 Mysql	13
2.9 Cloud storage	13
บทที่ 3. ขั้นตอนและวิธีการดำเนินงาน	14
3.1 การออกแบบระบบ	14
3.2 การทำงานของระบบ	15
บทที่ 4. การทดลองและวิจารณ์ผล	42
บทที่ 5. สรุป วิจารณ์ และข้อเสนอแนะ	51
5.1 สรุป	51
5.2 วิจารณ์ผลการทดลอง	51
5.3 ข้อเสนอแนะ	52



## สารบัญ (ต่อ)

	หน้า
เอกสารอ้างอิง	53
ภาคผนวก ก Source Code	54
ภาคผนวก ข การติดตั้งระบบปฏิบัติการ	62
ประวัติผู้แต่ง	66

## สารบัญภาพ

ภาพที่		หน้า
2-1	Raspberry Pi 3 model B+ และพอร์ตเชื่อมต่อต่าง ๆ	3
2-2	PIR Motion Sensor	4
2-3	ลักษณะการทำงานของอุปกรณ์ตรวจจับความเคลื่อนไหว	5
2-4	ลักษณะวงจรการทำงานทั่วไปของอุปกรณ์ตรวจจับความเคลื่อนไหว	5
2-5	ตัวอย่างภาพ RGB และภาพ Grayscale	6
2-6	ตัวอย่างการจำลองรูปแบบ Haar-like	8
2-7	วิธีการทั่วไปในการหาค่าผลรวมความเข้มของพื้นที่สี่เหลี่ยมใด ๆ ในรูปภาพ	8
2-8	Integral image	9
2-9	การหาผลรวมความเข้มของพื้นที่สี่เหลี่ยมด้วยวิธี Integral image	9
2-10	Cascade classifier	11
3-1	การทำงานของระบบ	14
3-2	คำสั่งถ่ายภาพ	16
3-3	ข้อมูลแสดงรายละเอียดการถ่ายภาพ image.jpg	16
3-4	ภาพนิ่ง image.jpg	16
3-5	การต่อสาย Raspberry Pi 3 กับเซ็นเซอร์ตรวจจับความเคลื่อนไหว	17
3-6	แผนภาพการทำงานของโปรแกรมตรวจจับความเคลื่อนไหว	17
3-7	คำสั่งการตรวจจับความเคลื่อนไหวในไฟล์ pircam.py	18
3-8	การเรียกใช้ Opencv และการเช็คเวอร์ชัน	19
3-9	หน้าต่างเว็บให้ไฟล์ชุดเทรน	20
3-10	ไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล	20
3-11	คำสั่งการดาวน์โหลดไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล	21
3-12	แผนภาพการทำงานของโปรแกรมการตรวจจับใบหน้าบุคคล	21
3-13	คำสั่งการตรวจจับใบหน้าบุคคลในไฟล์ detect.py	23
3-14	หน้าต่างแสดง web browser ของ apache2	24
3-15	หน้าต่างแสดง web browser ของข้อมูล php	25

## สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3-16	การใช้งานคำสั่งสำหรับการตั้งค่าภายในการใช้งานฐานข้อมูล MySQL	26
3-17	การตั้งค่ารหัสผ่านของการใช้งานฐานข้อมูล MySQL	26
3-18	การลบ anonymous user ในระบบฐานข้อมูล	27
3-19	การจัดการระบบการ login ในฐานะ root ผ่านระบบ remote	27
3-20	การลบ database ชื่อ test ออกจากระบบฐานข้อมูล MySQL	27
3-21	การจัดการคาว์ดโฮลด์ Privilege table	27
3-22	การเลือกใช้งานเว็บเซิร์ฟเวอร์ หน้าต่างคอนฟิก	28
3-23	การเข้าใช้งานการตั้งค่ารหัสผ่าน เพื่อเชื่อมโยงฐานข้อมูลของ root	28
3-24	หน้าต่างใช้งานของซอฟต์แวร์บริการจัดการฐานข้อมูล PhpMyadmin	29
3-25	รูปแบบการทำงานของแสดงผลภาพนิ่งผ่านเว็บเบราว์เซอร์	30
3-26	ไฟล์ไพลอนสำหรับการส่งข้อมูลจาก Raspberry Pi 3 ไปยัง phpMyadmin	31
3-27	การแสดงผลการใช้ phpMyAdmin ในการเก็บข้อมูลผ่านการจัดการ MySQL	32
3-28	ขั้นตอนการแสดงผลแบบถ่ายทอดสด	32
3-29	หน้าต่างแสดงระบบปฏิบัติการมองเห็นการเชื่อมต่อเว็บแคม	33
3-30	การแก้ไขไฟล์ motion.conf ในส่วน Daemon	33
3-31	การแก้ไขไฟล์ motion.conf ในส่วน stream_localhost	34
3-32	หมายเลขพอร์ตของเซิร์ฟเวอร์	34
3-33	การแก้ไขไฟล์ motion ในส่วน stream_motion_deamon	34
3-34	หน้าต่างแสดงผลแบบถ่ายทอดสด	35
3-35	การตั้งชื่อการเชื่อมต่อระยะไกล	36
3-36	ประเภทของ Cloud storage ที่ซอฟต์แวร์ Rclone สามารถซิงค์ได้	37
3-37	เลือกประเภทการตั้งค่า	37
3-38	หน้าต่างเว็บเบราว์เซอร์	38
3-39	หน้าต่างการอนุญาตซอฟต์แวร์ rclone สามารถเข้าถึงบัญชีผู้ใช้ Google	38
3-40	หน้าต่างแสดงโค้ด	39

## สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3-41	ตรวจสอบการตั้งค่าซอฟต์แวร์ rclone	39
3-42	คำสั่งการอัปโหลดภาพขึ้น Google drive ในไฟล์ up2drive.py	40
3-43	คำสั่งของระบบในไฟล์ system.py	41
4-1	การเชื่อมต่ออุปกรณ์ร่วมกับบอร์ด Raspberry Pi 3	42
4-2	การเข้าสู่หน้าต่าง Raspberry Pi Configuration	43
4-3	หน้าต่าง Raspberry Pi Configuration	43
4-4	หน้าต่างแสดงสถานะพร้อมทำงานของระบบและข้อความเริ่มการตรวจจับ	44
4-5	หน้าต่างแสดงข้อความตรวจพบความเคลื่อนไหวและรายละเอียดการถ่ายภาพนิ่ง	44
4-6	ภาพนิ่งถูกถ่ายจากการตรวจพบความเคลื่อนไหวของเซ็นเซอร์	45
4-7	การแสดงพิกัดบริเวณที่น่าจะเป็นใบหน้าบุคคล	45
4-8	ภาพนิ่งที่ผ่านการตรวจจับใบหน้าบุคคล	46
4-9	หน้าต่างแสดงผลการทำงานของระบบ	46
4-10	หน้าต่างระบบการจัดการฐานข้อมูล phpMyAdmin	47
4-11	หน้าต่างเว็บ SmartTrailCamera	47
4-12	หน้าต่างแสดงผลภาพนิ่งที่ผ่านการประมวลผลตรวจจับใบหน้าบุคคล	48
4-13	หน้าต่างหลัก	48
4-14	หน้าต่างแสดงผลการถ่ายทอดสด	49
4-15	แสดงรายละเอียดการอัปโหลดภาพนิ่งขึ้นเว็บฝากไฟล์	49
4-16	การอัปโหลดภาพนิ่งไปยัง Google drive	50
ข-1	ดาวน์โหลด NOOBS แบบไฟล์ .zip	64
ข-2	หน้าต่างแสดงระบบปฏิบัติการที่มีให้เลือกลง	64
ข-3	รูปแสดงขณะทำการติดตั้ง OS	65
ข-4	รูปแสดงการติดตั้งเสร็จสิ้น	65

# บทที่ 1

## บทนำ

ปัจจุบันกล้องวงจรปิดมีบทบาทสำคัญมากในสังคม จึงเกิดเทคโนโลยีใหม่ๆ ใช้ในการพัฒนาอุปกรณ์ ปรับปรุงประสิทธิภาพการใช้งานให้ดีขึ้น เพื่อเพิ่มมูลค่าสินค้า คณะผู้จัดทำจึงได้เกิดแนวคิดในการออกแบบและพัฒนากล้องสังเกตการณ์อัจฉริยะขนาดเล็ก เพื่อผลิตอุปกรณ์ที่มีประสิทธิภาพในราคาต้นทุนที่ต่ำลง อีกทั้งการทำงานที่เข้าใจได้ง่าย และสามารถพัฒนาเพิ่มเติมการตรวจจับวัตถุอื่น ๆ ในอนาคตได้อีกด้วย

### 1.1 วัตถุประสงค์ของโครงการ

- 1.1.1 เพื่อศึกษาการทำงานของบอร์ด Raspberry Pi 3 ร่วมกับเว็บแคม
- 1.1.2 เพื่อพัฒนาอุปกรณ์ถ่ายภาพให้มีประสิทธิภาพดีขึ้น สามารถตรวจจับวัตถุได้
- 1.1.3 เพื่อศึกษานำบอร์ด Raspberry Pi 3 มาประยุกต์ใช้งานร่วมกับไลบรารี OpenCV
- 1.1.4 เพื่อศึกษาวิธีการส่งภาพผ่านระบบเครือข่ายไร้สาย

### 1.2 ขอบเขตของการทำโครงการ

- 1.2.1 อุปกรณ์ถ่ายภาพสามารถถ่ายภาพได้เมื่อเซ็นเซอร์ตรวจจับความเคลื่อนไหว ตรวจพบการเคลื่อนไหวของวัตถุ
- 1.2.2 โปรแกรมสามารถตรวจจับวัตถุที่เคลื่อนไหวได้
- 1.2.3 ส่งข้อมูลภาพที่มีขนาดเล็กผ่านระบบเครือข่ายไร้สาย
- 1.2.4 จัดทำฐานข้อมูลและวิเคราะห์ข้อมูลของวัตถุที่บันทึกได้

### 1.3 แนวเหตุผล ทฤษฎีสำคัญหรือสมมติฐาน

กล้องวงจรปิดมีบทบาทสำคัญมากในสังคม เนื่องจากผู้คนให้ความสำคัญในเรื่องของความปลอดภัยในด้านต่าง ๆ มากขึ้น เช่น การรักษาความปลอดภัยในชีวิตและทรัพย์สิน รวมทั้งการตรวจสอบการทำงานของเครื่องจักรหรือตัวบุคคล การดูแลคนในบ้าน เป็นต้น แต่ด้วยเทคโนโลยีที่พัฒนาให้อุปกรณ์มีประสิทธิภาพดีขึ้น ทำให้กล้องวงจรปิดมีราคาสูง ทางคณะผู้จัดทำจึงมีแนวคิดในการออกแบบและพัฒนากล้องสังเกตการณ์อัจฉริยะขนาดเล็ก โดยกล้องสังเกตการณ์

อัจฉริยะขนาดเล็กสามารถถ่ายภาพเมื่อมีความเคลื่อนไหวผ่านเซ็นเซอร์ตรวจจับความเคลื่อนไหว สามารถตรวจจับวัตถุหรือระบุชนิดของวัตถุนั้นได้ โดยมุ่งเน้นวัตถุบริเวณใบหน้าเป็นหลัก ใช้หลักการตรวจจับใบหน้าบุคคลด้วยเทคนิคการจำแนกคุณลักษณะคล้ายฮาร์ (Haar-like feature) ของ Viola – Jones ภาพที่ผ่านการตรวจจับแล้วจะถูกส่งผ่านระบบเครือข่ายไร้สาย บันทึกรูปภาพบนเว็บฟลากไฟล์ (Cloud storage) หรือแสดงผลผ่านเว็บเบราว์เซอร์ ทำให้สามารถเปิดดูภาพที่ตรวจจับย้อนหลังได้

#### 1.4 ผลประโยชน์ที่คาดว่าจะได้รับ

1.4.1 อุปกรณ์ถ่ายภาพที่ติดกับบอร์ดราสเบอร์รี่พายและเซ็นเซอร์ตรวจจับความเคลื่อนไหวสามารถถ่ายภาพได้

1.4.2 ระบบสามารถตรวจจับ จำแนกและระบุชนิดวัตถุในภาพได้

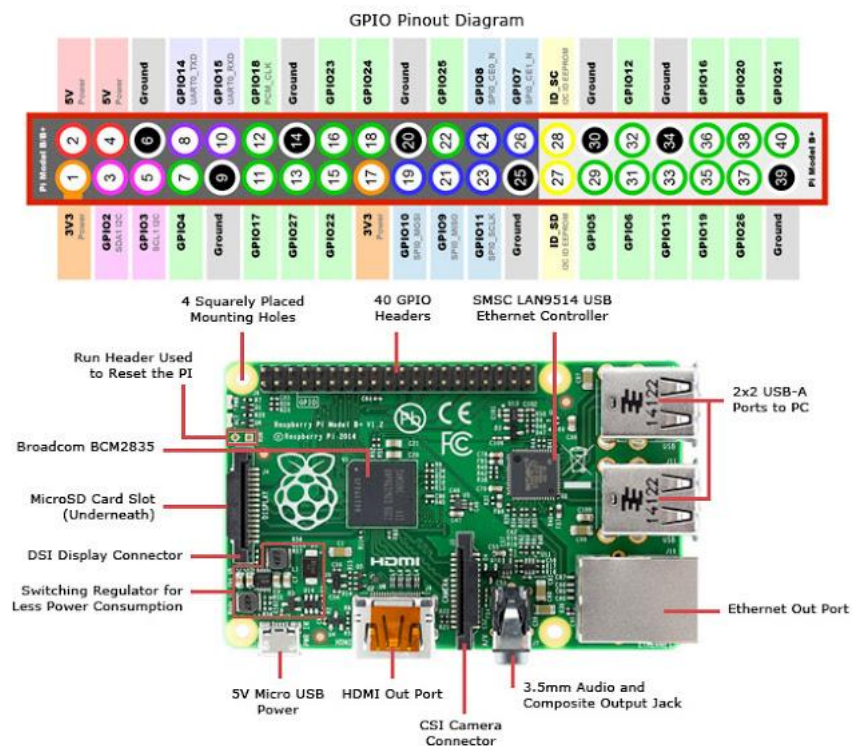
1.4.3 ได้รับความรู้ความเข้าใจในระบบการวิเคราะห์ภาพ

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 บอร์ดราสเบอร์รี่พาย ( Raspberry Pi )

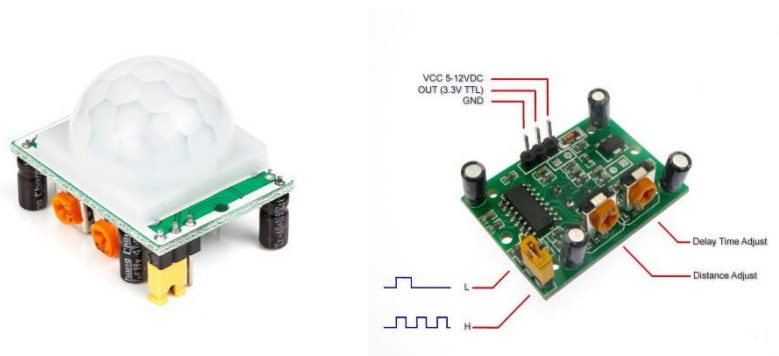
Raspberry Pi คืออุปกรณ์คอมพิวเตอร์ขนาดเล็กที่มีขนาดเท่ากับบัตรเครดิตหรือที่เรียกว่า Single-Board Computer โดยมีระบบปฏิบัติการพื้นฐานเป็น Linux ที่เรียกว่า NOOBS หรือ Raspian ซึ่งจะถูกรวบรวมลงใน SD Card ซึ่งการทำงานไม่ได้มีประสิทธิภาพเท่ากับคอมพิวเตอร์ทั่วไปแต่สามารถนำมาใช้งานในการฝึกใช้คอมพิวเตอร์เบื้องต้น รวมไปถึงการฝึกเขียนภาษา python ในการเรียนรู้ โดย Raspberry Pi มีช่องเชื่อมต่อสำหรับการใช้งานต่าง ๆ ให้เลือกใช้หลายอย่าง อาทิเช่น USB, Micro USB, HDMI port, GPIO port เป็นต้น เพื่อใช้ในการเพิ่มประสิทธิภาพในการทำงาน ดังภาพที่ 2-1



ภาพที่ 2-1 Raspberry Pi 3 model B+ และพอร์ตเชื่อมต่อต่าง

## 2.2 เซ็นเซอร์ตรวจจับความเคลื่อนไหว ( PIR Motion Sensor )

เซ็นเซอร์ตรวจจับความเคลื่อนไหว หรือที่เรียกว่า Passive Infrared Motion อุปกรณ์เซ็นเซอร์ชนิดหนึ่งที่ใช้ตรวจจับคลื่นรังสี Infrared ที่แผ่จากมนุษย์หรือสัตว์ที่มีการเคลื่อนไหว ซึ่งการทำงานของเซ็นเซอร์มีการนำประยุกต์ใช้ในการรักษาความปลอดภัยอย่างมาก ซึ่งมีลักษณะเป็นดังภาพที่ 2-2



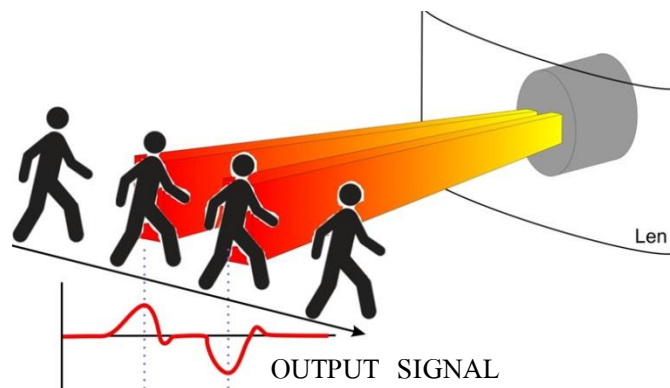
ภาพที่ 2-2 PIR Motion Sensor

### 2.2.1 คุณสมบัติทางเทคนิค

1. ขนาด 32.2 มิลลิเมตร × 24.3 มิลลิเมตร × 25.4 มิลลิเมตร (กว้าง × ยาว × สูง)
2. สามารถตรวจจับความเคลื่อนไหวได้ในช่วง 6 เมตร
3. ใช้เวลาในการปรับตัวเพื่อตรวจจับความเคลื่อนไหว 10 ถึง 60 วินาที
4. ใช้ไฟเลี้ยง +5 V
5. มีขาสำหรับเชื่อมต่อ 3 ขา
6. อุณหภูมิการทำงานอยู่ในช่วง 0 ถึง 60 องศาเซลเซียส (ใช้ในพื้นที่ย่อม)
7. เมื่อตรวจพบความเคลื่อนไหวจะให้สัญญาณ Output เป็น 1
8. รัศมีในการตรวจจับ 70 องศา

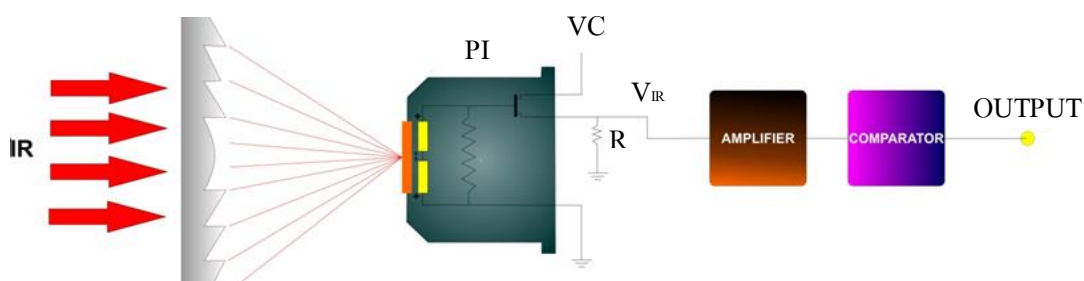


### 2.2.2 ลักษณะการทำงานของอุปกรณ์ตรวจจับความเคลื่อนไหว



ภาพที่ 2-3 ลักษณะการทำงานของอุปกรณ์ตรวจจับความเคลื่อนไหว

ภายใน PIR จะมีอุปกรณ์ตรวจจับรังสี Infrared อยู่ 2 ชุดด้วยกันดังภาพที่ 2-3 เมื่อมีคนหรือสัตว์ที่มีความอบอุ่นในร่างกายเคลื่อนที่ผ่านเข้ามาใน พื้นที่โซนที่ PIR สามารถตรวจจับคลื่นรังสี Infrared ที่แผ่ออกมาจากสิ่งมีชีวิตได้ PIR จะเปลี่ยนคลื่นรังสี Infrared ให้กลายเป็นกระแสไฟฟ้าดังภาพ จะเห็นว่าเมื่อมีสิ่งมีชีวิต เคลื่อนที่ผ่าน อุปกรณ์ตรวจจับรังสี Infrared ตัวที่ 1 จะได้สัญญาณ Output ออกมาสูงกว่าแรงดันปกติ และ เมื่อสิ่งมีชีวิตเคลื่อนที่ผ่าน อุปกรณ์ตรวจจับรังสี Infrared ตัวที่ 2 จะได้แรงดัน Output ต่ำกว่าค่าแรงดันปกติโดยลักษณะ การตรวจสอบเมื่อตรวจจับค่าได้จะส่งค่า 1 ไปออกจากขา Output ของตัวอุปกรณ์ตรวจจับความเคลื่อนไหวไปยังอุปกรณ์ที่เชื่อมต่อ สำหรับรูปแบบของวงจรทั่วไปจะมีลักษณะดังภาพที่ 2-4



ภาพที่ 2-4 ลักษณะวงจรการทำงานทั่วไปของอุปกรณ์ตรวจจับความเคลื่อนไหว

## 2.3 การเปลี่ยนภาพสีเป็นภาพระดับสีเทา

ภาพระดับสีเทา (Grayscale) [1] เป็นการแปลงภาพให้มีสีเทา มีระดับความเข้มของสีคือ 0-255 ซึ่งการแปลงจากภาพสี RGB มีเลเยอร์ชั้น R G และ B มาเป็นภาพระดับสีเทา ดังภาพที่ 2-5 ตามสมการที่ที่ 2-1 ดังนี้

$$\text{Grayscale} = (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \quad (2-1)$$

โดย Grayscale เป็นค่าความเข้มของสีเทา R G B คือค่าความเข้มของสีแดง สีเขียว และสีน้ำเงิน ตามลำดับ การที่ต้องแปลงจากภาพสีเป็นภาพระดับเทา ก่อนประมวลผล เนื่องจาก ณ ตำแหน่งพิกัดหนึ่งในภาพสี มีค่าความเข้ม 3 ชุด หากทำการประมวลผลภาพสีโดยตรง จะมีความยุ่งยาก และซับซ้อนมาก จึงนิยมแปลงเป็นภาพระดับสีเทาก่อน ซึ่งมีเลเยอร์ชั้นเดียว



ภาพที่ 2-5 ตัวอย่างภาพ RGB และภาพ Grayscale

## 2.4 การตรวจจับใบหน้า

ขั้นตอนการตรวจจับใบหน้า ถือเป็นขั้นตอนแรกของการวิเคราะห์ใบหน้า เป็นขั้นตอนที่มีความสำคัญ และมีปัจจัยที่มีผลกระทบต่อการตรวจจับมากมาย เช่น สเกล ตำแหน่ง ทิศทาง สภาพแสง เป็นต้น ซึ่งปัจจัยดังกล่าวมีผลต่อคุณลักษณะของใบหน้า อีกทั้งในการนำไปใช้งานที่ต้องการความเร็ว และความแม่นยำที่สูง ในปัจจุบันสามารถจำแนกวิธีการพื้นฐานในการตรวจจับใบหน้าได้เป็น 4 ประเภท [2] ได้แก่

2.4.1 Knowledge-based methods ใช้ส่วนประกอบของใบหน้ามนุษย์มาหากฎเกณฑ์ในการจำแนกใบหน้าของมนุษย์

2.4.2 Feature invariant approaches ค้นหาลักษณะเด่นที่แตกต่างกันของแต่ละใบหน้า โดยใช้เทคนิคการตรวจจับเส้นขอบ และใช้โมเดลทางสถิติอธิบายความสัมพันธ์

2.4.3 Template matching methods หาคความสัมพันธ์ของใบหน้า โดยการหาค่า สหสัมพันธ์ (Correlation value) กับรูปแบบมาตรฐาน (Standard pattern)

2.4.4 Appearance-based methods ใช้โมเดลที่ได้จากการเรียนรู้ของกลุ่มตัวอย่าง ที่เป็นใบหน้า และไม่ใช่อใบหน้า ใช้เทคนิคการวิเคราะห์ทางสถิติ และเทคนิค machine learning ในการหาคุณลักษณะของรูปภาพ

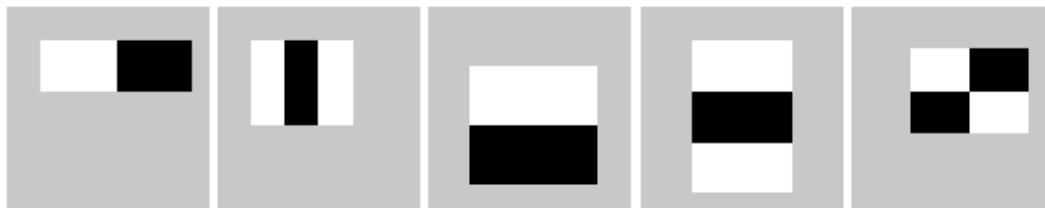
ในงานวิจัยที่เกี่ยวข้องที่ผ่านมาวิธี Appearance-based เป็นวิธีที่มีการพัฒนามากที่สุด และเป็นที่ยอมรับสามารถนำมาใช้งานได้จริงในหลายๆด้าน ในวิธีนี้ใช้กล่าวถึงหลักการตรวจจับใบหน้าของ Viola - Jones มีความสามารถในการตรวจจับใบหน้าได้อย่างรวดเร็ว มีประสิทธิภาพ และสามารถนำไปใช้งานได้จริง

## 2.5 วิธีการตรวจจับใบหน้าของ Viola-Jones

วิธีการตรวจจับใบหน้าด้วยเทคนิคการจำแนกคุณลักษณะคล้ายฮาร์ (Haar-like feature) วิธีการนี้ได้ถูกเสนอเป็นครั้งแรกโดย Viola-Jones ในปี 2001 [3] เทคนิคการตรวจจับใบหน้าของ Viola - Jones เป็นการนำ Sub-window เลื่อนไปตามตำแหน่งต่าง ๆ ในรูปภาพเพื่อพิจารณาและตรวจจับใบหน้าจากรูปภาพที่เข้ามา โดยตรวจหา สแกนหลายครั้งบนภาพเดิม ด้วยขนาดที่แตกต่างกันในแต่ละรอบ และได้มีการสร้างตัวจำแนกประเภทแบบต่อเรียง (Cascaded classifier) เป็นตัวจำแนกหลายตัวต่อกัน เพื่อจำแนกส่วนที่มีโอกาสเป็นใบหน้าและไม่ใช่อใบหน้า ยังมีจำนวนชั้น (Stage) ของตัวจำแนกมาก โอกาสที่ตรวจจับได้ใบหน้าจะมีมากขึ้น วิธีการตรวจจับใบหน้าของ Viola-Jones ประกอบด้วย 3 ขั้นตอน การคำนวณการจำลองรูปแบบ Haar-like ด้วย Integral image การค้นหาการจำลองรูปแบบ Haar-like ด้วย Adaboost และการรวมตัวจำแนกกลุ่มแบบต่อเนื่อง (Cascade classifier)

### 2.5.1 Integral image

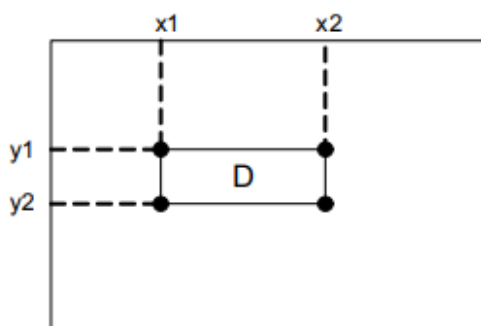
ตัวอย่างของการจำลองรูปแบบ Haar-like ที่ใช้ในเทคนิคของ Viola-Jones ดังภาพที่ 2-6 โดยแต่ละลักษณะเด่นจะประกอบด้วยพื้นที่สี่เหลี่ยม 2 ประเภท คือ ส่วนที่เป็นสีขาวและส่วนที่เป็นสีดำ การหาผลต่างระหว่างความเข้มในส่วนที่เป็นสีดำ กับส่วนที่เป็นสีขาว จากนั้นนำผลที่ได้ไปเปรียบเทียบกับค่าขีดแบ่ง (Threshold) กับขั้ว (Polarity) ในการตัดสินใจว่าภาพที่รับเข้ามาจัดให้เป็น Positive image (ภาพที่มีวัตถุที่สนใจ) หรือ Negative image (ภาพที่ไม่มีวัตถุที่สนใจ) ผลลัพธ์ถูกนำไปพิจารณาในขั้นตอนต่อไป



ภาพที่ 2-6 ตัวอย่างการจำลองรูปแบบ Haar-like

จากภาพที่ 2-7 เป็นวิธีการทั่วไปในการหาค่าผลรวมความเข้มของพื้นที่สี่เหลี่ยมใด ๆ ในรูปภาพ เขียนแทนด้วยสมการที่ 2-2 โดย  $i(x, y)$  แทนค่าความเข้มที่จุด  $x$  และ  $y$  ในรูปภาพ  $i_s(x_1, x_2, y_1, y_2)$  แทนผลรวมความเข้มภายในพื้นที่สี่เหลี่ยม  $x_1, x_2, y_1, y_2$  เป็นพิกัดมุมทั้งสี่ของรูปสี่เหลี่ยม ซึ่งเวลาที่ใช้ในการคำนวณขึ้นอยู่กับขนาดของรูปสี่เหลี่ยม ซึ่งใช้เวลาในการคำนวณมากและไม่เหมาะกับงานแบบเวลาจริง

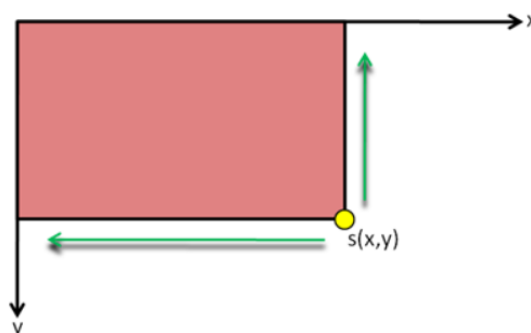
$$i_s(x_1, x_2, y_1, y_2) = \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} i(x, y) \quad (2-2)$$



ภาพที่ 2-7 วิธีการทั่วไปในการหาค่าผลรวมความเข้มของพื้นที่สี่เหลี่ยมใดๆในรูปภาพ

Viola-Jones จึงเสนอเทคนิคการใช้ภาพอินทิกรัล (Integral image) หรือตารางพื้นที่รวม (Summed-area table) เป็นการหาผลรวมค่าความเข้มของพิกเซลที่อยู่บนและอยู่ทางซ้ายทั้งหมดของตำแหน่ง  $x, y$  ดังภาพที่ 2-8 สามารถนิยามได้ตามสมการที่ 2-3 เมื่อ  $s(x, y)$  คือ ค่าของ integral image ที่ตำแหน่งจุด  $(x, y)$  และ  $i(x', y')$  คือค่าความเข้มแต่ละพิกเซลในภาพต้นฉบับ

$$s(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2-3)$$

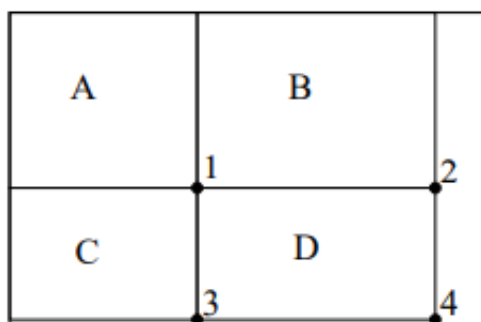


ภาพที่ 2-8 Integral image

ตัวอย่างการหาผลรวมความเข้มของพื้นที่สี่เหลี่ยม โดยใช้ Integral image จำนวน 4 จุด การใช้ Integral image หาผลรวมความเข้มของจุดใดๆ แต่ละจุด ได้ตามสมการที่ 2-4 และหาผลรวมความเข้มพื้นที่สี่เหลี่ยม จำนวน 4 จุด ผลลัพธ์ที่ได้คือ ค่าตำแหน่งที่ 4+1-2-3 โดยค่าที่ตำแหน่ง 1 เป็นผลรวม ความเข้มพิกเซลในสี่เหลี่ยม A ตำแหน่ง 2 เป็นผลรวมพิกเซลในสี่เหลี่ยม A+B ตำแหน่งที่ 3 เป็นผลรวมพิกเซล A+C และตำแหน่ง 4 เป็นผลรวมพิกเซล A+B+C+D จะได้ ผลรวมความเข้มในสี่เหลี่ยม 1234 คือ  $(A+B+C+D) + A - (A+B) - (A+C) = D$  แสดงดังภาพ 2-9 ตามสมการที่ 2-5

$$s(x, y) = i(x, y) + s(x-1, y) + s(x, y-1) - s(x-1, y-1) \quad (2-4)$$

$$\sum_{\substack{x_0 < x \leq x_1 \\ y_0 < y \leq y_1}} i(x, y) = s(D) + s(A) - s(B) - s(C) \quad (2-5)$$



ภาพที่ 2-9 การหาผลรวมความเข้มของพื้นที่สี่เหลี่ยมด้วยวิธี Integral image

การจำลองรูปแบบ Haar-like ที่ได้หลังจากการเปรียบเทียบกับค่าขีดแบ่ง เรียกว่า ตัวจำแนกแบบอ่อนแอ (Weak classifier) เนื่องจากคำตอบที่ได้มีความแม่นยำต่ำ มีความถูกต้องมากกว่าเดสุ่มเพียงเล็กน้อย สามารถหาตัวจำแนกแบบอ่อนแอได้จากสมการที่ 2-6 ดังนี้

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (2-6)$$

โดย  $x$  คือ sub-window ของภาพอินพุต งานวิจัยของ Viola - jones ได้กำหนดขนาดเท่ากับ  $24 \times 24$  พิกเซล ส่วน  $h_j(x)$  คือ ตัวจำแนกแบบอ่อนแอ ที่พิจารณาจากรูปแบบการจำลอง  $f_j$  เทียบกับค่าขีดแบ่ง  $\theta_j$  และ  $p_j$  แสดงทิศทางของสมการ

### 2.5.2 ขั้นตอนการเรียนรู้ด้วย Adaboost

นำรูปแบบการจำลอง Haar-like ที่ได้จากขั้นตอนแรกมาเข้ากระบวนการ machine learning ที่เรียกว่า Adaptive boost หรือ Adaboost เป็นกระบวนการที่หาตัวจำแนกแบบอ่อนแอ (weak classifier) และกำหนดค่าถ่วงน้ำหนักที่ทำให้ค่าผิดพลาดน้อยสุดในแต่ละรอบของกระบวนการ เพื่อสร้างตัวจำแนกแบบแข็งแรง (Strong classifier) ซึ่งขั้นตอนการเรียนรู้ Adaboost มีดังนี้

1. พิจารณาเลือกกลุ่มภาพตัวอย่าง  $(x_1, y_1), \dots, (x_n, y_n)$  เพื่อใช้ในการเรียนรู้ เมื่อ  $y_i = 0, 1$  สำหรับ negative image และ positive image ตามลำดับ
2. กำหนดค่าถ่วงน้ำหนักเริ่มต้น ดังสมการที่ 2-7 สำหรับ  $y_i = 0, 1$  เมื่อ  $m$  และ  $l$  เป็นจำนวนของ negative image และ positive image ตามลำดับ

$$w_{1,i} = \frac{1}{2m}, \frac{1}{2l} \quad (2-7)$$

3. กำหนดจำนวนตัวจำแนกแบบอ่อนแอมีค่าเท่ากับ  $t = 1, 2, \dots, T$  รอบ

- นอร์มัลไลซ์ค่าถ่วงน้ำหนัก ดังสมการที่ 2-8 ซึ่ง  $w_t$  คือการกระจายความน่าจะเป็น

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}} \quad (2-8)$$

- แต่ละคุณลักษณะ ( $j$ ) ทำสอนตัวจำแนก ( $h_j$ ) ถูกจำกัดให้ใช้เพียงคุณลักษณะเดียว ซึ่งความผิดพลาดที่พิจารณาจาก  $w_t$  ตามสมการที่ 2-9 ดังนี้

$$\varepsilon_j = \sum_i w_i |h_j(x_i) - y_i| \quad (2-9)$$

- หาตัวจำแนกแบบ ( $h_t$ ) ที่ทำให้ค่าผิดพลาด ( $\varepsilon_t$ ) ต่ำที่สุด

- ปรับปรุงค่าถ่วงน้ำหนักของแต่ละตัวอย่าง ดังสมการที่ 2-10

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\varepsilon_i} \quad (2-10)$$

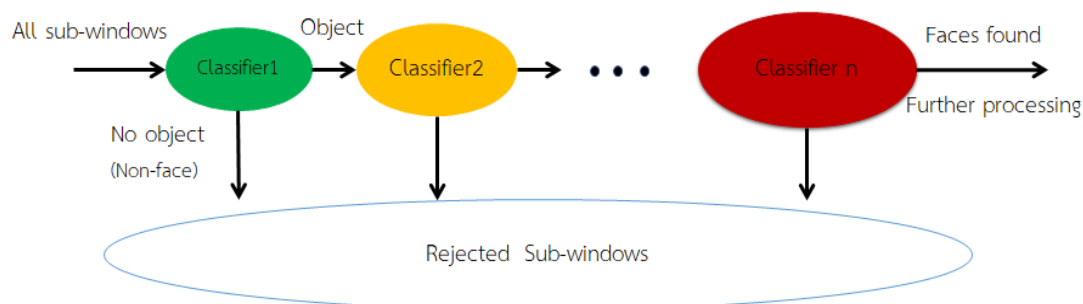
เมื่อ  $e_i = 0$  ถ้าตัวอย่าง  $x_i$  ถูกเลือกอย่างถูกต้อง  $e_i = 1$  ในกรณีอื่น ๆ และค่า  $\beta_i = \frac{\varepsilon_i}{1 - \varepsilon_i}$

4. ผลลัพธ์สุดท้ายได้ ตัวจำแนกแบบแข็งแกร่ง ดังสมการที่ 2-11 โดย  $h_i$  คือ ตัวจำแนกแบบอ่อนแอ และ  $\alpha_i$  คือ ตัวถ่วงน้ำหนักที่กำหนดให้ตัวจำแนกในแต่ละรอบที่คำนวณ มีค่าเท่ากับ

$$C(x) = \begin{cases} 1 & \sum_{i=1}^T \alpha_i h_i(x) \geq \frac{1}{2} \sum_{i=1}^T \alpha_i \\ 0 & \text{otherwise} \end{cases} \quad \text{เมื่อ } \alpha_i = \log \frac{1}{\beta_i} \quad (2-11)$$

### 2.5.3 ขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง

เป็นการเพิ่มประสิทธิภาพในการตรวจจับให้มีความถูกต้อง และยังใช้เวลาในการคำนวณลดลงด้วย โดยการสร้างการรวมตัวจำแนกกลุ่มแบบต่อเรียง โดยเทคนิคนี้จะนำตัวจำแนก (classifier) หลาย ๆ ตัวต่อกันเป็นลำดับ ดังภาพที่ 2-10 โดยตัวจำแนกในลำดับต้นๆ จะมีความซับซ้อนน้อยกว่า สร้างจากตัวจำแนกแบบอ่อนไม่กี่ตัวเพื่อลดปริมาณการคำนวณและลดระดับค่าขีดแบ่ง ซึ่ง Sub-window ที่ถูกจัดประเภทไม่ใช่ใบหน้าจะถูกทิ้งไปเป็นจำนวนมากในลำดับต้น ๆ และหาก sub-window นั้นถูกจำแนกว่ามีโอกาสเป็นภาพใบหน้า จะถูกส่งต่อไปยังจำแนกตัวถัดไปซึ่งมีความซับซ้อนมากขึ้นตามลำดับ โดยชั้นของตัวจำแนกมากขึ้น โอกาสที่ Sub-window จะเป็นใบหน้ามากยิ่งขึ้น หรือกล่าวว่าการตรวจจับใบหน้ามีความแม่นยำขึ้น



ภาพที่ 2-10 Cascade classifier

## 2.6 เว็บเซิร์ฟเวอร์ Apache

คือซอฟต์แวร์สำหรับเปิดให้บริการเซิร์ฟเวอร์บน โพรโทคอล HTTP โดยสามารถทำงานได้บนหลายระบบปฏิบัติการซึ่งถูกพัฒนามาจาก HTTPD Web Server โดย Apache นี้จะทำหน้าที่ในการจัดเก็บ Homepage และส่ง Homepage ไปยัง Browser ที่มีการเรียกเข้า ยัง Web server ที่เก็บ HomePage นั้นอยู่ ซึ่ง Apache เป็นซอฟต์แวร์โอเพนซอร์สที่สามารถใช้งานได้ฟรี โดยมีการใช้โดยรวมประมาณ 67% ของเว็บเซิร์ฟเวอร์ทั้งหมดในโลก ซึ่งรวดเร็วเชื่อถือได้และปลอดภัย สามารถปรับแต่งได้เพื่อตอบสนองความต้องการของสภาพแวดล้อมที่หลากหลาย โดยสามารถเพิ่ม Function พิเศษที่เป็น Module plugin ได้โดยง่าย

### 2.6.1 เว็บเซิร์ฟเวอร์ (Web server)

Web server (เว็บเซิร์ฟเวอร์) เครื่องคอมพิวเตอร์ซึ่งให้บริการที่เก็บเว็บไซต์ (Server) แล้วให้ผู้ใช้ (Client) เรียกชมหน้าเว็บไซต์และสามารถเข้ามาดูข้อมูล ภาพ เสียง ได้โดยใช้โพรโทคอล HTTP ผ่านทางเว็บเบราว์เซอร์

2.6.2 เว็บเบราว์เซอร์ (Web Browser) คือ โปรแกรมที่ทำหน้าที่แสดงเนื้อหาเว็บไซต์ โดยแปลงเป็น html script เป็นข้อความที่เรา สามารถดูได้ เว็บเบราว์เซอร์ที่ได้รับความนิยม ได้แก่ Internet Explorer, Netscape, Mozilla Firefox Google Chrome

## 2.7 phpMyadmin

phpMyAdmin คือโปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูล Mysql แทนการคีย์คำสั่ง เนื่องจากถ้าเราจะใช้ฐานข้อมูลที่เป็น MySQL บางครั้งจะมีความลำบากและยุ่งยากในการใช้งาน ดังนั้นจึงมีเครื่องมือในการจัดการฐานข้อมูล MySQL ขึ้นมาเพื่อให้สามารถจัดการ ตัว DBMS ที่เป็น MySQL ได้ง่ายและสะดวกยิ่งขึ้น โดย phpMyAdmin ก็ถือเป็นเครื่องมือชนิดหนึ่งในการจัดการ

phpMyAdmin เป็นส่วนต่อประสานที่สร้างโดยภาษา PHP ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้าง Table ใหม่ ๆ และยังมี Function ที่ใช้สำหรับการทดสอบการ Query ข้อมูลด้วยภาษา SQL พร้อมกันนั้น ยังสามารถทำการ Insert delete update หรือแม้กระทั่งใช้ คำสั่งต่าง ๆ เหมือนกับกับการใช้ภาษา SQL ในการสร้างตารางข้อมูล

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



## 2.8 Mysql

Mysql คือ โปรแกรมระบบจัดการฐานข้อมูล ที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่น เพื่อให้ได้ระบบงานที่รองรับ ความต้องการของผู้ใช้ เช่น ทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษา php ภาษา asp.net หรือภาษาเจเอสพี เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษาวิซวลเบสิกคอตเน็ต ภาษาจาวา หรือภาษาซีชาร์ป เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนทซอร์ซ (Open Source) ที่ถูกนำไปใช้งานมากที่สุด Mysql จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relational Database Management System) ซึ่งเป็นที่นิยมใช้กันมากในปัจจุบัน

## 2.9 Cloud storage

Cloud Storage หรือเว็บฝากไฟล์ เป็นการอัปโหลดรูปภาพหรือข้อมูลไปยังเว็บฝากไฟล์แบบ คลังข้อมูลสำรองไปฝากไว้บนคราวด์ (Cloud) ทำให้สามารถเปิดดูข้อมูลได้จากที่ไหนก็ได้เพียงแค่มีอินเทอร์เน็ต ซึ่งเว็บฝากไฟล์ในปัจจุบันมีผู้ให้บริการหลายเจ้า เช่น Google drive, Amazon, Dropbox, Nextcloud, owncloud เป็นต้น มีทั้งส่วนที่ให้บริการฟรีและเสียค่าบริการขึ้นอยู่กับความปลอดภัยของข้อมูลหรือความต้องการของผู้ใช้งาน

ในโครงการนี้จะเลือกใช้ Google drive เนื่องจาก Google drive เป็นบริการจาก Google ที่สามารถนำไฟล์ต่าง ๆ ไปฝากไว้กับ Google ผ่านพื้นที่เก็บข้อมูลระบบคราวด์ และสำรองไฟล์ข้อมูลที่มีความปลอดภัย อีกทั้งยังสามารถแชร์ไฟล์กับบุคคลที่ต้องการ รวมถึงแก้ไขร่วมกันได้จากอุปกรณ์หลายประเภท ทำให้ Google drive จึงเป็นที่นิยม เข้าถึงง่ายและให้พื้นที่เก็บข้อมูลฟรี 15 GB ซึ่งถือว่ามากที่สุดในคราวด์ที่ให้บริการฟรี

รวมถึงใช้ Rclone เป็นซอฟต์แวร์แบบ command line ใช้สำรองไฟล์ข้อมูลสำหรับเว็บฝากไฟล์เพื่อซิงค์ไฟล์และโฟลเดอร์จากต้นทางไปยังปลายทาง โดยสามารถเชื่อมต่อเว็บฝากไฟล์ได้หลายชนิด เป็นตัวช่วยในการอัปโหลดหรือดาวน์โหลดข้อมูลระหว่างเว็บฝากไฟล์กับอุปกรณ์ได้สะดวกขึ้น

## บทที่ 3

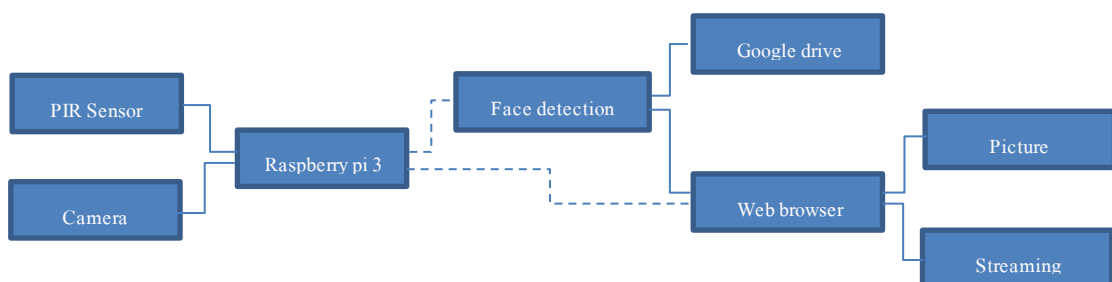
### ขั้นตอนและวิธีการดำเนินงาน

ขั้นตอนและวิธีการดำเนินงานของกล้องสังเกตการณ์อัจฉริยะขนาดเล็ก อธิบายถึงขั้นตอนการออกแบบระบบ การทำงานของระบบ ซึ่งได้แยกการทำงานออกเป็น ส่วน ดังนี้ การเชื่อมต่อบอร์ดร่วมกับเว็บแคมและเซ็นเซอร์ตรวจจับความเคลื่อนไหว การประมวลผลตรวจจับใบหน้าบุคคล การอัปโหลดภาพนิ่งขึ้นเว็บฝากไฟล์ และการแสดงผลผ่านเว็บเบราว์เซอร์

#### 3.1 การออกแบบระบบ

กล้องสังเกตการณ์อัจฉริยะขนาดเล็ก ตรวจจับวัตถุที่เคลื่อนผ่านเซ็นเซอร์ได้ภาพนิ่ง (Snapshot) นำภาพนิ่งที่ได้มาประมวลผลเพื่อตรวจหาบริเวณเฉพาะใบหน้าบุคคล ก่อนอัปโหลดภาพนิ่งที่ผ่านการประมวลผลเก็บไว้บนเว็บฝากไฟล์ และสามารถแสดงผลภาพนิ่ง การถ่ายทอดสด (Streaming) ผ่านทางเว็บเบราว์เซอร์ได้อีกด้วย

ระบบออกแบบโดยใช้เซ็นเซอร์ตรวจจับความเคลื่อนไหว เพื่อจับความเคลื่อนไหวของวัตถุในบริเวณที่ต้องการ มีเว็บแคมสำหรับถ่ายภาพ และบอร์ด Raspberry Pi 3 เป็นศูนย์กลางควบคุมการทำงานของระบบ ไม่ว่าจะเป็นการประมวลผลการตรวจจับใบหน้าบุคคล การอัปโหลดภาพ ไปยังเว็บฝากไฟล์ รวมถึงเว็บเซิร์ฟเวอร์ที่ทำหน้าที่เป็นศูนย์กลางในการเก็บรวบรวมข้อมูล ซึ่งแสดงการทำงานของระบบได้ดังภาพที่ 3-1



ภาพที่ 3-1 การทำงานของระบบ

### 3.2 การทำงานของระบบ

การทำงานของระบบ แบ่งการทำงานออกเป็น 4 ส่วนหลัก ได้แก่ การเชื่อมต่อบอร์ดร่วมกับเว็บแคมและเซ็นเซอร์ตรวจจับความเคลื่อนไหว ส่วนการประมวลผลตรวจจับใบหน้าบุคคล ส่วนการอัปโหลดภาพขึ้นเว็บฝากไฟล์ และส่วนการแสดงผลผ่านเว็บเบราว์เซอร์ ซึ่งแต่ละส่วนมีหน้าที่การทำงานดังนี้

#### 3.2.1 การเชื่อมต่อบอร์ดร่วมกับเว็บแคมและเซ็นเซอร์ตรวจจับความเคลื่อนไหว

เมื่อมีวัตถุซึ่งเป็นสิ่งมีชีวิต มุ่งเน้นที่มนุษย์เป็นหลักเคลื่อนผ่านเซ็นเซอร์ในบริเวณพื้นที่ที่เซ็นเซอร์ทำงาน เซ็นเซอร์จะตรวจพบความเคลื่อนไหวของวัตถุ ทำให้ค่าลอจิกที่ส่งไปยังบอร์ด Raspberry Pi 3 จากเดิมที่ส่งค่าลอจิก ‘0’ เปลี่ยนเป็นส่งค่าลอจิก ‘1’ ซึ่งเป็นการสั่งให้เว็บแคมทำการถ่ายภาพนิ่งเก็บไว้ในโฟลเดอร์ที่กำหนด

การเชื่อมต่อบอร์ดร่วมกับเว็บแคมและเซ็นเซอร์ อธิบายตั้งแต่อุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง การติดตั้งใช้งานเว็บแคม การต่อเซ็นเซอร์ตรวจจับความเคลื่อนไหวในบอร์ด Raspberry Pi 3 และคำสั่งการตรวจจับความเคลื่อนไหว

##### 3.2.1.1 อุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง

ชุดอุปกรณ์ในการใช้งาน Raspberry Pi 3 แบบต่อจอ (ดูได้จากอุปกรณ์ที่ใช้ในการติดตั้ง OS ในภาคผนวก ข) เว็บแคม และเซ็นเซอร์ตรวจจับความเคลื่อนไหว

##### 3.2.1.2 วิธีการติดตั้งใช้งานเว็บแคมในบอร์ด Raspberry Pi 3

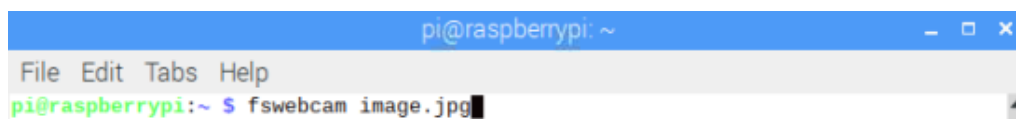
เริ่มติดตั้งซอฟต์แวร์ fswebcam ซึ่งเป็นซอฟต์แวร์ถ่ายภาพจากเว็บแคม โดยป้อนด้านล่างแล้วกดปุ่ม Enter

```
$ sudo apt-get install fswebcam
```

เมื่อติดตั้งซอฟต์แวร์ fswebcam เสร็จเรียบร้อยแล้ว ทำการรีบูตระบบปฏิบัติการใหม่ด้วยคำสั่ง

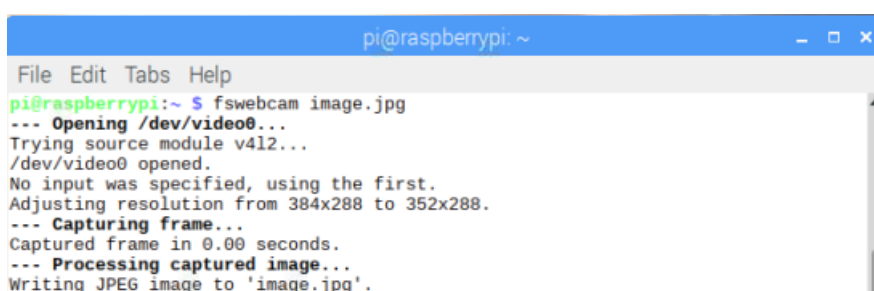
```
$ sudo reboot
```

ทดสอบการใช้งานเว็บแคมในการถ่ายภาพ โดยป้อนคำสั่ง fswebcam ตามด้วยชื่อไฟล์ที่ใช้นับที่ภาพ แล้วกดปุ่ม Enter ดังภาพที่ 3-2



ภาพที่ 3-2 คำสั่งถ่ายภาพ

เมื่อระบบทำการถ่ายภาพเสร็จสิ้น จะแสดงข้อมูลรายละเอียดของภาพหนึ่ง ดังภาพที่ 3-3 และสามารถเปิดดูภาพหนึ่งในโฟลเดอร์ที่ได้บันทึกไว้ ได้ภาพหนึ่งดังภาพที่ 3-4



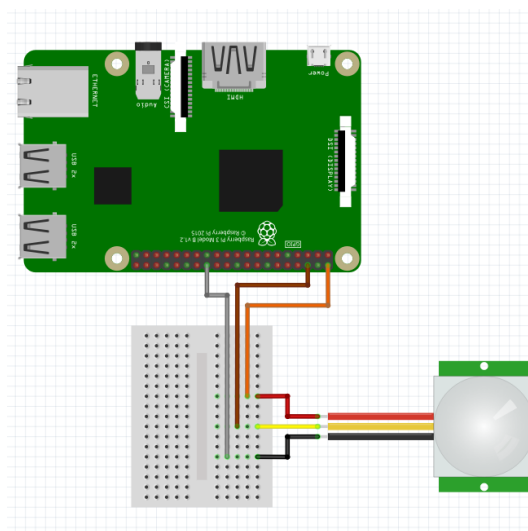
ภาพที่ 3-3 ข้อมูลแสดงรายละเอียดการถ่ายภาพ image.jpg



ภาพที่ 3-4 ภาพนิ่ง image.jpg

### 3.2.1.3 วิธีการต่อเซ็นเซอร์ตรวจจับความเคลื่อนไหว

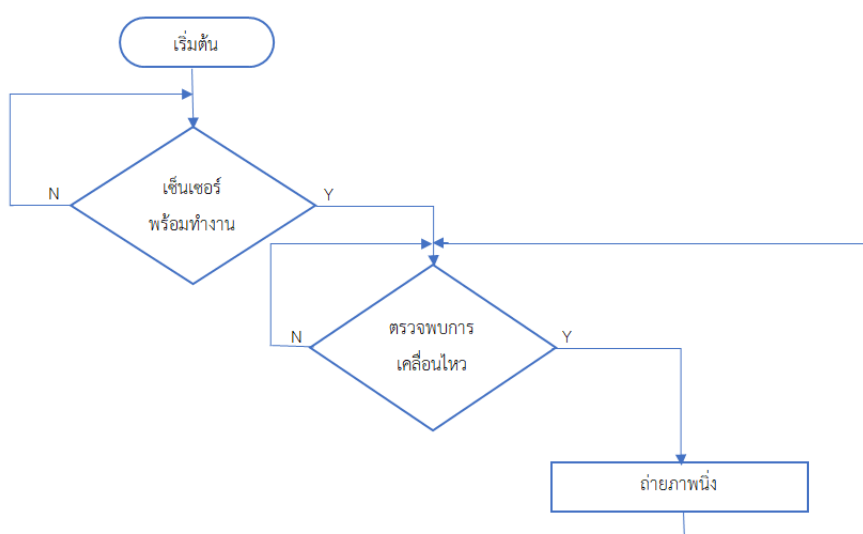
เซ็นเซอร์ตรวจจับความเคลื่อนไหวเชื่อมต่อกับบอร์ด Raspberry Pi 3 ผ่านพอร์ต GPIO โดยใช้ขาไฟ 5 V ต่อกับขา 5 V (Pin 2) ของบอร์ด Raspberry Pi 3 เพื่อไปเลี้ยงการทำงานของเซ็นเซอร์ตรวจจับความเคลื่อนไหว ขา GND ต่อกับขา GND (Pin 6) ของบอร์ด Raspberry Pi 3 และใช้ GPIO7 (Pin 26) ของบอร์ด Raspberry Pi 3 ในการอ่านค่าที่ได้จากเซ็นเซอร์ตรวจจับความเคลื่อนไหว แสดงการต่อวงจรดังภาพที่ 3-5



ภาพที่ 3-5 การต่อสาย Raspberry Pi 3 กับเซ็นเซอร์ตรวจจับความเคลื่อนไหว

#### 3.2.1.4 คำสั่งตรวจจับความเคลื่อนไหว

โปรแกรมตรวจจับความเคลื่อนไหวออกแบบโดยเริ่มจากตรวจสอบความพร้อมทำงานของเซ็นเซอร์ตรวจจับความเคลื่อนไหว หากเซ็นเซอร์ตรวจจับความเคลื่อนไหวไม่พร้อมทำงาน โปรแกรมจะตรวจสอบต่อไปจนกว่าเซ็นเซอร์ตรวจจับความเคลื่อนไหวพร้อมทำงาน เมื่อเซ็นเซอร์ตรวจจับความเคลื่อนไหวพร้อมทำงาน และตรวจพบความเคลื่อนไหวของวัตถุ โปรแกรมจะสั่งให้เว็บแคมทำการถ่ายภาพนิ่ง ดังภาพที่ 3-6 แสดงแผนภาพการทำงานของโปรแกรมตรวจจับความเคลื่อนไหว



ภาพที่ 3-6 แผนภาพการทำงานของโปรแกรมตรวจจับความเคลื่อนไหว

การเขียนโปรแกรมตรวจจับความเคลื่อนไหว บันทึกในไฟล์ชื่อ pircam.py เริ่มจากตั้งค่าพอร์ต GPIO ที่ใช้ในการเชื่อมต่อระหว่างเซ็นเซอร์ตรวจจับความเคลื่อนไหวและบอร์ด Raspberry Pi 3 รวมถึงกำหนดสถานะก่อนหน้าและสถานะปัจจุบันเป็น '0' ก่อนตรวจสอบการทำงานของเซ็นเซอร์ตรวจจับความเคลื่อนไหวให้อยู่ในสภาวะพร้อมทำงาน หากค่าลอจิกที่รับมาจากเซ็นเซอร์ตรวจจับความเคลื่อนไหวมีค่าลอจิกเป็น '1' กำหนดให้สถานะ ปัจจุบันมีค่าเป็น '0' เมื่อเซ็นเซอร์ตรวจจับความเคลื่อนไหวพร้อมทำงานจะเข้าสู่ช่วงเช็คเงื่อนไขสถานะก่อนหน้าและสถานะปัจจุบันเพื่อควบคุมการถ่ายภาพ โดยรับค่าที่ได้จากเซ็นเซอร์ตรวจจับความเคลื่อนไหวเพื่อกำหนดค่าสถานะปัจจุบัน ถ้าสถานะปัจจุบันเป็น '1' และสถานะก่อนหน้าเป็น '0' แสดงว่าเซ็นเซอร์ตรวจจับความเคลื่อนไหวตรวจพบความเคลื่อนไหวของวัตถุ กำหนดค่าให้สถานะปัจจุบันเป็น '1' และถ่ายภาพผ่านเว็บแคม จำนวน 3 ภาพ ซึ่งแต่ละภาพมีความละเอียด 400x400 พิกเซล บันทึกในไฟล์เดอร์ที่กำหนดด้วยรูปแบบการตั้งชื่อไฟล์ ปี:เดือน:วัน - ชั่วโมง:นาที:วินาที เมื่อถ่ายภาพเสร็จเรียบร้อยแล้ว ระบบจะกลับไปรับค่าที่ได้จากเซ็นเซอร์ตรวจจับความเคลื่อนไหวเพื่อกำหนดสถานะปัจจุบันอีกครั้ง หากสถานะปัจจุบันเป็น '0' และสถานะก่อนหน้าเป็น '1' แสดงว่าเซ็นเซอร์ตรวจจับความเคลื่อนไหวไม่พบความเคลื่อนไหวของวัตถุ กำหนดค่าให้สถานะปัจจุบันเป็น '0' และแสดงข้อความพร้อมทำงานอีกครั้ง ระบบจะดำเนินต่อไปจนกว่าจะสั่งหยุดการทำงานของโปรแกรม ซึ่งมีโปรแกรมดังภาพที่ 3-7

```

pi@raspberrypi: ~/test
File Edit Tabs Help
GNU nano 2.7.4 File: pircam.py

#!/bin/sh
import RPi.GPIO as GPIO
import cv2
import time
import os
from subprocess import call
from datetime import datetime
import cv2.cv as cv
from os import mkdir
GPIO.setmode(GPIO.BCM)
GPIO_PIR = 7
GPIO.setup(GPIO_PIR, GPIO.IN)
Current_State = 0
Previous_State = 0
try:
    print "Waiting for PIR to settle"
    while GPIO.input(GPIO_PIR)==1:
        Current_State = 0
    print "Ready"
    while True:
        Current_State = GPIO.input(GPIO_PIR)
        if Current_State==1 and Previous_State==0:
            print "Motion detected!"
            Previous_State=1
            for x in range(0,3):
                os.system("fswebcam -r 400x400 -F $ /home/pi/$
                time.sleep(2)
            elif Current_State==0 and Previous_State==1:
                print "Ready"
                Previous_State=0
                time.sleep(2)
except KeyboardInterrupt:
    print "Quit"
GPIO.cleanup

```

ภาพที่ 3-7 คำสั่งการตรวจจับความเคลื่อนไหวในไฟล์ pircam.py

### 3.2.2 การประมวลผลตรวจจับใบหน้าบุคคล

นำภาพนิ่งที่ได้จากการถ่ายภาพในขั้นตอนการเชื่อมต่อบอร์ดร่วมกับเว็บแคมและเซ็นเซอร์ตรวจจับความเคลื่อนไหวมาประมวลผลตรวจจับใบหน้าบุคคล โดยหาบริเวณที่น่าจะเป็นใบหน้าบุคคลด้วยเทคนิคการจำแนกคุณลักษณะคล้ายฮาร์ (Haar-like feature) ของ Viola – Jones ซึ่งภาพนิ่งหลังผ่านการประมวลผลจะถูกบันทึกไว้อีกไฟล์เดอร์หนึ่งเพื่อนำไปใช้ในการส่งไปยังเว็บฝากไฟล์หรือแสดงผลผ่านเว็บไซต์

การประมวลผลตรวจจับใบหน้า อธิบายถึงอุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง วิธีการติดตั้งซอฟต์แวร์ Opencv วิธีการเตรียมไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล และคำสั่งการตรวจจับใบหน้า

#### 3.2.2.1 อุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง

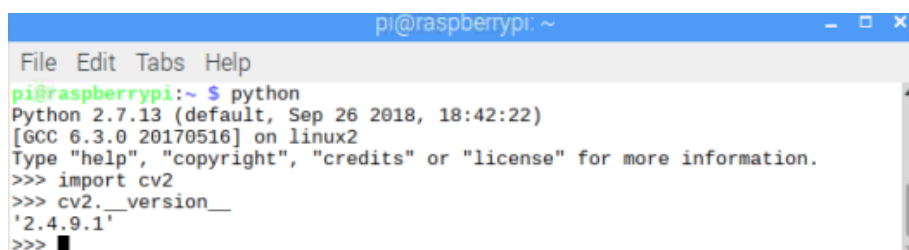
เชื่อมต่ออุปกรณ์ในการใช้งาน Raspberry Pi 3 แบบต่อจอ (คู่ได้จากอุปกรณ์ที่ใช้ในการติดตั้ง OS ในภาคผนวก ข) ซอฟต์แวร์ Opencv และ ไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล

#### 3.2.2.2 วิธีการติดตั้งซอฟต์แวร์ Opencv

ทำการติดตั้งซอฟต์แวร์ Opencv ด้วยการป้อนคำสั่งด้านล่าง แล้วกดปุ่ม Enter

```
$ sudo apt-get install python-opencv -y
```

เมื่อติดตั้งซอฟต์แวร์ Opencv เสร็จเรียบร้อยแล้ว พิมพ์คำสั่ง python เพื่อทำการ import cv2 และเรียกดูเวอร์ชันดังภาพที่ 3-8 เมื่อปรากฏเวอร์ชันแสดงว่าการติดตั้งเรียบร้อยแล้ว

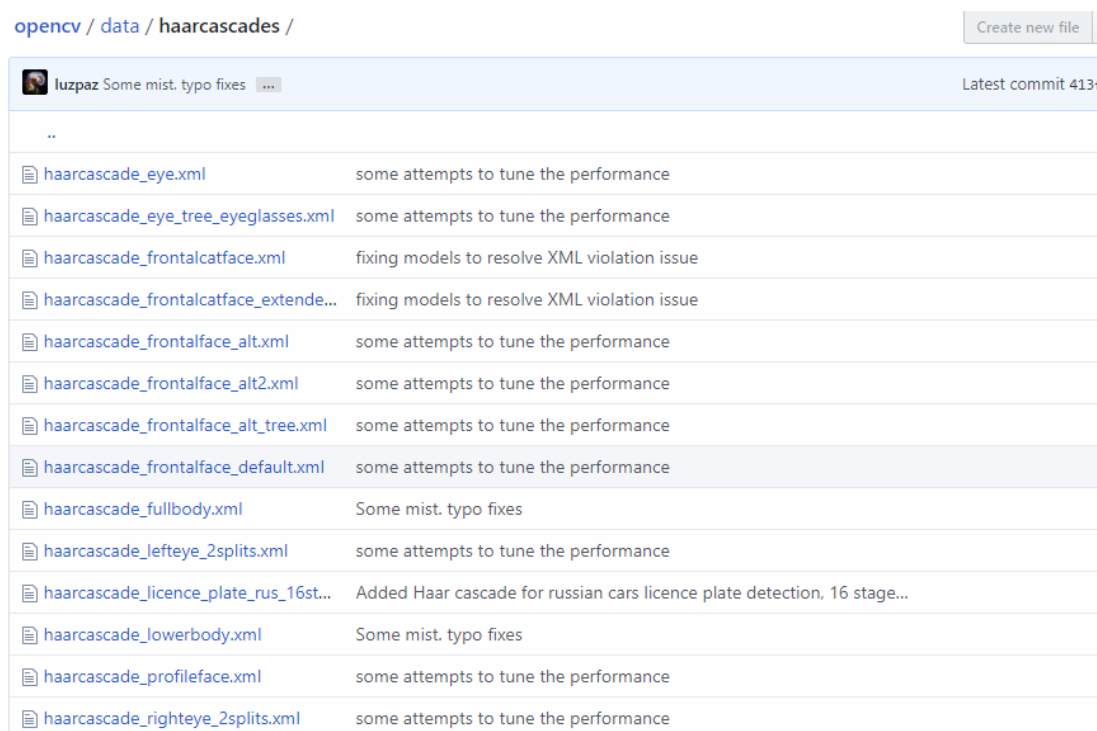


```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python
Python 2.7.13 (default, Sep 26 2018, 18:42:22)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'2.4.9.1'
>>>
```

ภาพที่ 3-8 การเรียกใช้ Opencv และการเช็คเวอร์ชัน

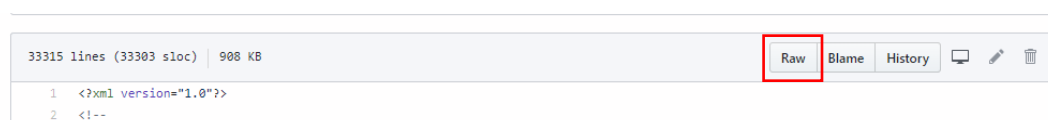
### 3.2.2.3 วิธีการเตรียมไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล

ทำการดาวน์โหลดไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล ได้จากเว็บ <https://github.com/opencv/opencv/tree/master/data/haarcascades> ซึ่งจะมีไฟล์ชุดเทรนมากมายให้เลือกใช้งาน



ภาพที่ 3-9 หน้าต่างเว็บให้ไฟล์ชุดเทรน

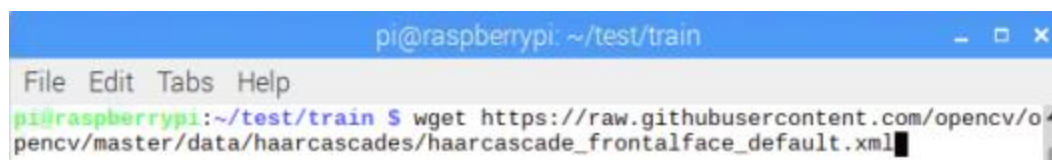
เลือกไฟล์ haarcascade\_frontalface\_default.xml เป็นไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล จากนั้นกด Raw (ปุ่มด้านบนขวา) เพื่อทำการคัดลอกโค้ด



ภาพที่ 3-10 ไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล



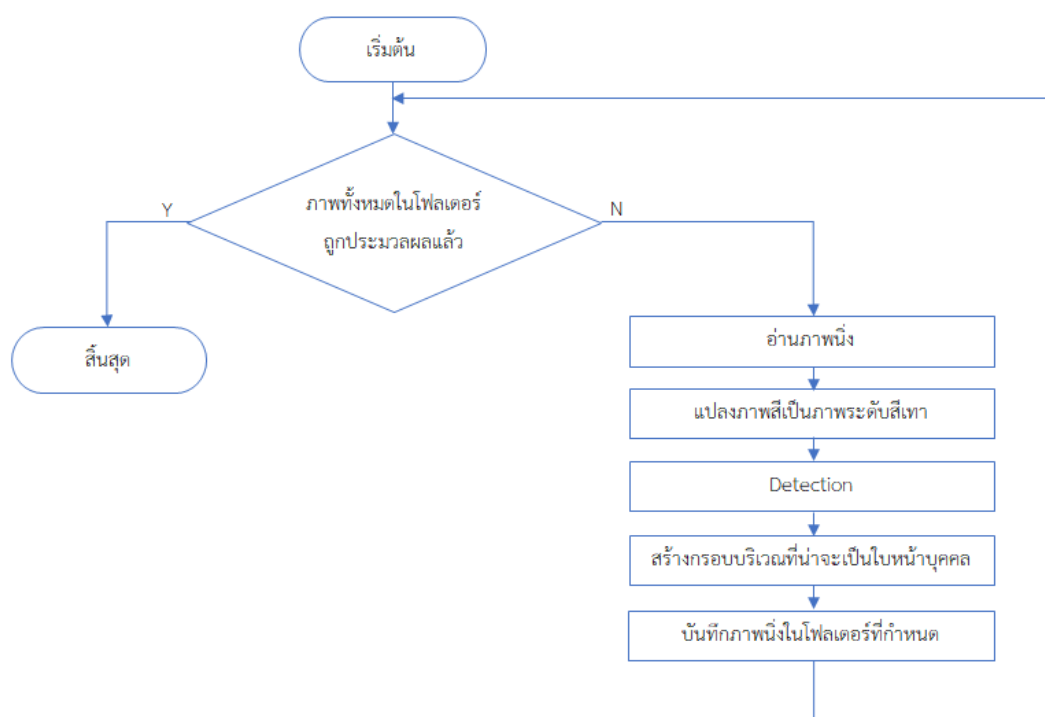
คัดลอกไฟล์ชุดเทรนตรวจจับใบหน้าบุคคลบนที่ลงบนบอร์ด Raspberry Pi 3 โดยป้อนคำสั่ง wget ตามด้วยลิงค์ที่คัดลอกมา ดังภาพที่ 3-11 แล้วกดปุ่ม Enter เมื่อดาวน์โหลดเสร็จสิ้น จะได้ไฟล์ชุดเทรนตรวจจับใบหน้าบุคคลอยู่ในโฟลเดอร์ที่กำหนด



ภาพที่ 3-11 คำสั่งการดาวน์โหลดไฟล์ชุดเทรนตรวจจับใบหน้าบุคคล

### 3.2.2.4 คำสั่งตรวจจับใบหน้าบุคคล

โปรแกรมตรวจจับใบหน้าบุคคลถูกออกแบบให้ตรวจหาบริเวณความน่าจะเป็นใบหน้าบุคคลในภาพนิ่ง แต่ละภาพ ตั้งแต่ภาพแรกจนถึงภาพสุดท้ายที่อยู่ในโฟลเดอร์ โดยเริ่มจากนำภาพนิ่งแปลงภาพสีเป็นภาพระดับสีเทา ก่อนเข้าประมวลผลเพื่อหาบริเวณความน่าจะเป็นใบหน้าบุคคล จากนั้นทำการสร้างกรอบล้อมรอบบริเวณที่ตรวจพบ และบันทึกภาพนิ่งที่ได้ในโฟลเดอร์ที่กำหนด แสดงแผนภาพดังภาพที่ 3-12



ภาพที่ 3-12 แผนภาพการทำงานของโปรแกรมการตรวจจับใบหน้าบุคคล

การเขียนโปรแกรมตรวจจับใบหน้าบุคคล จำเป็นต้องสร้างโฟลเดอร์เพื่อบันทึกภาพนิ่งที่ถ่ายได้จากเว็บแคมในโฟลเดอร์ชื่อ Inpic และสร้างอีกโฟลเดอร์บันทึกภาพนิ่งที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคลในโฟลเดอร์ชื่อ GDrive ทั้งสองโฟลเดอร์อยู่ในพาธ home/pi/test หลังจากสร้างโฟลเดอร์เรียบร้อยแล้ว เริ่มเข้าสู่การเขียนโปรแกรมตรวจจับใบหน้าบันทึกในไฟล์ชื่อ detect.py ส่วนแรกของฟังก์ชันหลักเป็นการกำหนดพาธของภาพนิ่งที่จะนำมาประมวลผล พาธของภาพนิ่งที่ผ่านการประมวลผล และพาธของไฟล์ชุดเทรนตรวจจับใบหน้าบุคคลที่ได้เตรียมไว้ในขั้นตอนที่ 3.2.2.3 เก็บพาธไว้ในอีกตัวแปรหนึ่งเพื่อความสะดวกในการใช้งาน จากนั้นนำภาพนิ่งแต่ละภาพที่อยู่ในโฟลเดอร์ Inpic มาประมวลผล เริ่มจากการอ่านภาพนิ่งขึ้นมา ด้วยคำสั่ง cv2.imread ทำการแปลงภาพจากภาพสีเป็นภาพระดับสีเทา ด้วยคำสั่ง cv2.cvtColor ก่อนเข้าฟังก์ชันย่อย detection ทั้งภาพสี (color\_img) และภาพระดับสีเทา (gray\_img) โดยภาพระดับสีเทาใช้ในการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคล ส่วนภาพสีใช้ในการแสดงผลสร้างกรอบสี่เหลี่ยมบริเวณความน่าจะเป็นใบหน้าบุคคล ซึ่งในฟังก์ชันย่อย detection ประกอบด้วยการตรวจหาบริเวณความน่าจะเป็นใบหน้าบุคคล ด้วยเมธอด detectMultiScale ในภาพระดับสีเทาที่ชื่อว่า frame ซึ่งมีพารามิเตอร์ คือ ปรับขนาดภาพ (scaleFactor) จำนวนพิกเซลข้างเคียงที่ยังคงรักษาความน่าจะเป็นใบหน้าบุคคลในบริเวณนั้น (minNeighbors) ขนาดเล็กสุดของใบหน้าบุคคลที่ตรวจหาในภาพนิ่ง (minSize) การปรับขนาดแบบฮาร์ใน flags ช่วยให้การประมวลผลเร็วขึ้น เมื่อตรวจพบบริเวณความน่าจะเป็นใบหน้าบุคคลจะทำการสร้างกรอบสี่เหลี่ยมด้วยคำสั่ง cv2.rectangle ในภาพสีที่ชื่อว่า color เริ่มจากตำแหน่ง (x, y) สิ้นสุดที่ตำแหน่ง (x+w, y+h) ค่าสี่กรอบสี่เหลี่ยมคือ (255, 0, 0) เป็นการใส่ค่าสีน้ำเงินจาก (B, G, R) และมีความหนาของกรอบสี่เหลี่ยมเป็น 2 อีกทั้งเหนือกรอบสี่เหลี่ยมแสดงข้อความระบุใบหน้าบุคคลว่า "face" ด้วยคำสั่ง cv2.putText ก่อนเข้าใช้ฟังก์ชันย่อย saveCropped เพื่อบันทึกภาพนิ่งที่ได้หลังจากประมวลผลแล้ว สามารถเขียนคำสั่งดังภาพที่ 3-13

```

pi@raspberrypi: ~/test
File Edit Tabs Help
GNU nano 2.7.4 File: detect.py

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

def detection(frame,color):
    # detection frontal_face
    bBoxes_frontal_face_default=frontal_face_default.detectMultiScale(frame, s$
    print("bBoxes_frontal_face::")
    print(bBoxes_frontal_face_default)

    for (x,y,w,h) in bBoxes_frontal_face_default:
        color_rect= cv2.rectangle(color,(x,y),(x+w,y+h),(255,0,0),2)
        cv2.putText(color,"face", (x,y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 2$
        saveCropped(color,name)

def saveCropped(img, name):
    cv2.imwrite(output_path+ name , img)

if __name__ == "__main__":

    # paths to input and output images
    input_path= "/home/pi/test/Inpic/"
    output_path= "/home/pi/test/8Drive/"
    input_names= listdir(input_path)

    frontal_face_default= cv2.CascadeClassifier("/home/pi/test/train/haarcasc$
    print("$starting to detect faces in images and save the cropped images to $
    sttime= time.clock()

    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)
        detection(gray_img,color_img)

Get Help Write Out Where Is Cut Text Justify Cur Pos
Exit Read File Replace Uncut Text To Linter Go To Line

```

ภาพที่ 3-13 คำสั่งการตรวจจับใบหน้าบุคคลในไฟล์ detect.py

### 3.2.3 การแสดงผลผ่านเว็บเบราว์เซอร์

การแสดงผลผ่านเว็บเบราว์เซอร์ ประกอบด้วย การแสดงผล 2 แบบ คือ การแสดงผลแบบถ่ายทอดสด (Streaming) และการแสดงผลแบบภาพนิ่ง (Snapshot) ที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคล จากหัวข้อที่ 3.2.2 ซึ่งการแสดงผลผ่านเว็บเบราว์เซอร์ทำให้ผู้ใช้งานสามารถตรวจสอบบริเวณที่สนใจผ่านการถ่ายทอดสด หรือดูภาพนิ่งย้อนหลังจากเว็บแคมที่ติดตั้งไว้ได้ตลอดเวลาไม่ว่าผู้ใช้งานจะอยู่ไหนผ่านอุปกรณ์อิเล็กทรอนิกส์ที่เชื่อมต่อกับอินเทอร์เน็ตได้ อาทิเช่น คอมพิวเตอร์ โทรศัพท์มือถือ แท็บเล็ต เป็นต้น

การแสดงผลผ่านเว็บเบราว์เซอร์ อธิบายตั้งแต่อุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง วิธีการติดตั้งใช้งานเว็บเซิร์ฟเวอร์ รวมถึงการตั้งค่าซอฟต์แวร์ต่าง ๆ การแสดงผลแบบถ่ายทอดสด และการแสดงผลแบบภาพนิ่งที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคล

#### 3.2.3.1 อุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง

ชุดอุปกรณ์ในการใช้งาน Raspberry Pi 3 แบบต่อจอ (ดูได้จากอุปกรณ์ที่ใช้ในการติดตั้ง OS ในภาคผนวก) ซอฟต์แวร์ Apache2 MySQL และ PhpMyAdmin ภาษา PHP

### 3.2.3.2 วิธีการติดตั้งใช้งานเว็บเซิร์ฟเวอร์บนบอร์ด Raspberry Pi 3

กำหนดให้บอร์ด Raspberry Pi 3 ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ ด้วยการติดตั้งซอฟต์แวร์ Apache ที่มีระบบจัดการฐานข้อมูลด้วยซอฟต์แวร์ MySQL ซึ่งใช้ซอฟต์แวร์ phpMyAdmin ในการบริหารจัดการฐานข้อมูล MySQL

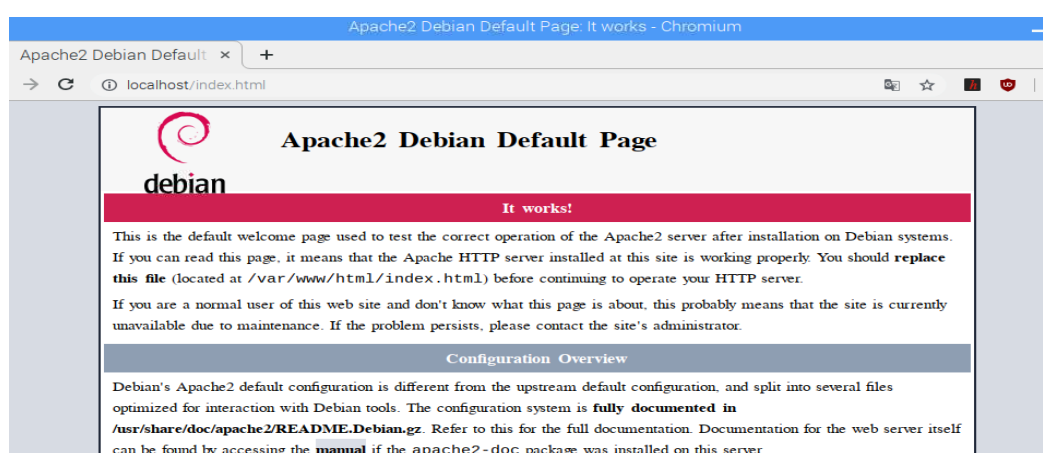
ก่อนการติดตั้งซอฟต์แวร์ต่าง ๆ ทำการอัปเดตระบบปฏิบัติการภายในบอร์ด Raspberry Pi 3 ด้วยคำสั่งด้านล่าง แล้วกด Enter

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

เมื่ออัปเดตเสร็จเรียบร้อยแล้ว เริ่มจากการติดตั้งซอฟต์แวร์ apache โดยป้อนคำสั่งดังนี้

```
$ sudo apt-get install apache2 -y
```

หลังจากติดตั้งเสร็จเรียบร้อยแล้ว ตรวจสอบการติดตั้งซอฟต์แวร์โดยการเปิดเว็บไซต์ พิมพ์ “localhost” หรือ IP Address ของ Raspberry Pi 3 ที่ Address bar ของ Chromium จะปรากฏหน้าต่างเป็น index.html ซึ่งเป็นหน้าเริ่มต้นของการใช้งาน กรณีจะเปลี่ยนหน้าเริ่มต้นให้ตั้งชื่อไฟล์เป็น index.html และลบไฟล์ก่อนหน้า




ภาพที่ 3-14 หน้าต่างแสดง web browser ของ apache2

ต่อมาทำการติดตั้งภาษา PHP เพื่อใช้ในการรันสคริป PHP โดยป้อนคำสั่ง  
ด้านล่างแล้วกดปุ่ม Enter

```
$ sudo apt-get install php -y
```

สามารถตรวจสอบการติดตั้งโดยสร้างไฟล์สคริป ชื่อไฟล์ info.php พิมพ์คำสั่ง  
ด้านล่างและบันทึกไว้ในพื้นที่เว็บ (/var/www/html) จากนั้นลองเปิดโดยพิมพ์ localhost/info.php  
ที่ Address bar ของ Chromium หากปรากฏหน้า PHP info แสดงว่าการติดตั้งเสร็จเรียบร้อยแล้ว ซึ่ง  
หน้าต่างนี้จะแสดงรายละเอียดต่างๆภายในระบบของ php ดังภาพที่ 3-15

```
//info.php  
<?php phpinfo();?>
```

phpinfo() - Chromium	
st/info.php	
<b>PHP Version 7.0.33-0+deb9u3</b> 	
System	Linux raspberrypi 4.14.98-v7+ #1200 SMP Tue Feb 12 20:27:48 GMT 2019 armv7l
Build Date	Mar 8 2019 10:01:24
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/fpm
Loaded Configuration File	/etc/php/7.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/fpm/conf.d
Additional .ini files parsed	/etc/php/7.0/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.0/fpm/conf.d/10-openssl.ini, /etc/php/7.0/fpm/conf.d/10-pdo.ini, /etc/php/7.0/fpm/conf.d/15-xsl.ini, /etc/php/7.0/fpm/conf.d/20-bcmath.ini, /etc/php/7.0/fpm/conf.d/20-calendar.ini, /etc/php/7.0/fpm/conf.d/20-curl.ini, /etc/php/7.0/fpm/conf.d/20-dom.ini, /etc/php/7.0/fpm/conf.d/20-exif.ini, /etc/php/7.0/fpm/conf.d/20-fileinfo.ini, /etc/php/7.0/fpm/conf.d/20-ftp.ini, /etc/php/7.0/fpm/conf.d/20-gd.ini, /etc/php/7.0/fpm/conf.d/20-gettext.ini, /etc/php/7.0/fpm/conf.d/20-ldap.ini, /etc/php/7.0/fpm/conf.d/20-ldap_sasl.ini, /etc/php/7.0/fpm/conf.d/20-mbstring.ini, /etc/php/7.0/fpm/conf.d/20-mysqli.ini, /etc/php/7.0/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.0/fpm/conf.d/20-phar.ini, /etc/php/7.0/fpm/conf.d/20-posix.ini, /etc/php/7.0/fpm/conf.d/20-readline.ini, /etc/php/7.0/fpm/conf.d/20-shmop.ini, /etc/php/7.0/fpm/conf.d/20-simplexml.ini, /etc/php/7.0/fpm/conf.d/20-sockets.ini, /etc/php/7.0/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.0/fpm/conf.d/20-sysvsem.ini, /etc/php/7.0/fpm/conf.d/20-sysvshm.ini, /etc/php/7.0/fpm/conf.d/20-tokenizer.ini, /etc/php/7.0/fpm/conf.d/20-wddx.ini, /etc/php/7.0/fpm/conf.d/20-xmlreader.ini, /etc/php/7.0/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.0/fpm/conf.d/20-xsl.ini, /etc/php/7.0/fpm/conf.d/20-zip.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled

ภาพที่ 3-15 หน้าต่างแสดง web browser ของข้อมูล php

จากนั้นทำการติดตั้งซอฟต์แวร์จัดการฐานข้อมูล MySQL ป้อนคำสั่ง แล้วกดปุ่ม

Enter

```
$ sudo apt-get install mysql-server mysql-client -y
```

เมื่อติดตั้งเสร็จเรียบร้อยแล้ว ทำการตั้งค่ารหัสผ่าน และกำหนดค่าเบื้องต้นให้กับซอฟต์แวร์ ด้วยคำสั่ง

```
$ sudo mysql_secure_installation
```

แสดงข้อความให้ป้อนรหัสผ่านปัจจุบันของ root กดปุ่ม Enter ข้ามขั้นตอนนี้ไป

```
pi@raspberrypi:~ $ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
```

ภาพที่ 3-16 การใช้งานคำสั่งสำหรับการตั้งค่าภายในการใช้งานฐานข้อมูล MySQL

ระบบแสดงข้อความ ให้ทำการตั้งค่ารหัสผ่านใหม่ โดยการพิมพ์ y และกดปุ่ม Enter หลังจากนั้นกรอกรหัสผ่านที่ต้องการ ดังภาพที่ 3-17 เสร็จเรียบร้อยแล้วระบบจะแสดงข้อความ ให้ทำการลบ anonymous user โดยพิมพ์ y แล้วกดปุ่ม Enter ดังภาพที่ 3-18

```
Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

ภาพที่ 3-17 การตั้งค่ารหัสผ่านของการใช้งานฐานข้อมูล MySQL

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
```

```
Remove anonymous users? [Y/n] y
... Success!
```

### ภาพที่ 3-18 การลบ anonymous user ในระบบฐานข้อมูล

จากนั้นระบบขอความต้องการปิดระบบการ login ในฐานะ root ผ่านระบบ remote หรือไม่ ให้พิมพ์ y แล้วกดปุ่ม Enter ดังภาพที่ 3-19 และระบบแสดงข้อความต้องการลบ database ชื่อ test ซึ่งเป็น database ที่มาพร้อมการติดตั้งหรือไม่ ให้พิมพ์ y แล้วกดปุ่ม Enter

```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
```

```
Disallow root login remotely? [Y/n] y
... Success!
```

### ภาพที่ 3-19 การจัดการระบบการ login ในฐานะ root ผ่านระบบ remote

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.
```

```
Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

### ภาพที่ 3-20 การลบ database ชื่อ test ออกจากระบบฐานข้อมูล MySQL

ระบบจะแสดงข้อความต้องการให้ดาวน์โหลด Privilege table หรือไม่ พิมพ์ y แล้วกดปุ่ม Enter ระบบแสดงข้อความตั้งค่าเสร็จเรียบร้อยแล้ว พร้อมใช้งาน MySQL ดังภาพที่ 3-21

```
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
```

```
Reload privilege tables now? [Y/n] y
... Success!
```

```
Cleaning up...
```

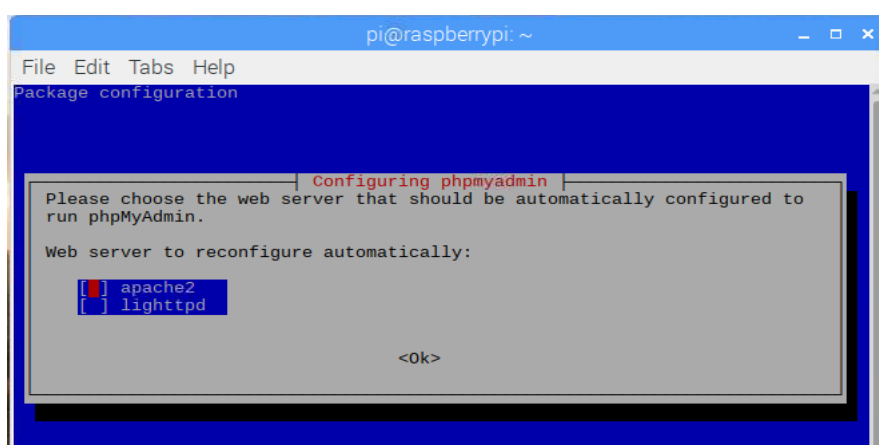
```
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.
```

### ภาพที่ 3-21 การจัดการดาวน์โหลด Privilege table

สุดท้ายติดตั้งซอฟต์แวร์บริการจัดการฐานข้อมูล MySQL หรือ ซอฟต์แวร์ PhpMyAdmin โดยป้อนคำสั่งด้านล่าง แล้วกดปุ่ม Enter

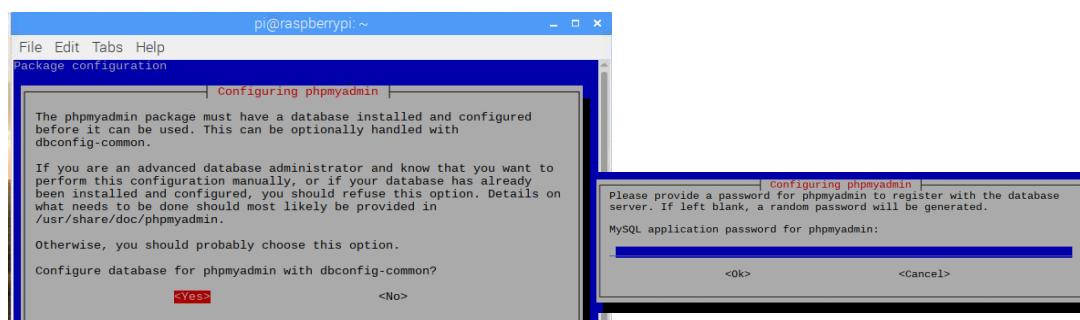
```
$ sudo apt-get install phpmyadmin -y
```

เมื่อติดตั้งเสร็จเรียบร้อยแล้ว จะปรากฏหน้าต่างคอนฟิกเลือกใช้งานเว็บเซิร์ฟเวอร์ ให้เลือก Apache2 แล้วกดปุ่ม Enter



ภาพที่ 3-22 การเลือกใช้งานเว็บเซิร์ฟเวอร์ หน้าต่างคอนฟิก

ปรากฏหน้าต่างต้องการคอนฟิก database ที่ใช้ใน phpMyadmin ด้วย dbconfig-common หรือไม่ ให้เลือก yes จากนั้นจะแสดงหน้าต่างให้กำหนดรหัสผ่านเพื่อเชื่อมต่อกับฐานข้อมูล โดยกรอกรหัสผ่านเดียวกันกับรหัสผ่านของ root ใน MySQL ดังภาพที่ 3-23



ภาพที่ 3-23 การเข้าใช้งานการตั้งรหัสผ่านเพื่อเชื่อมโยงฐานข้อมูลของ root



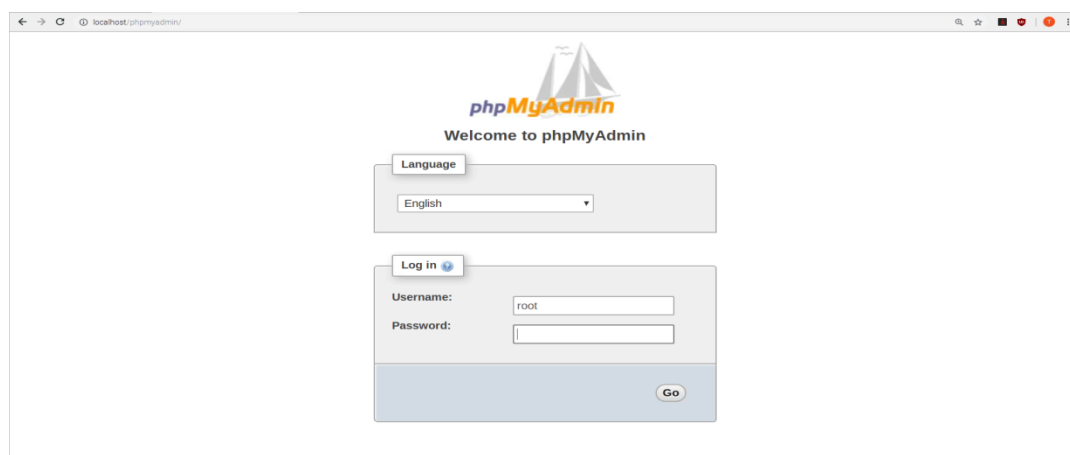
เมื่อคอนฟิกและกำหนดรหัสผ่านเรียบร้อยแล้ว ก่อนทดสอบการใช้งานซอฟต์แวร์ phpMyadmin ต้องทำการสร้างลิ้งค์ชี้ไปยังซอฟต์แวร์ phpMyadmin ที่คอนฟิกไว้ โดยป้อนคำสั่ง

```
$ sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

กรณีหากระบบแจ้งเตือนไม่พบ mbstring ให้ทำการติดตั้งและรีสตาร์ทซอฟต์แวร์ Apache2 โดยป้อนคำสั่ง

```
$ sudo apt-get install phpmyadmin php-mbstring php-gettext
$ sudo service apache2 restart
```

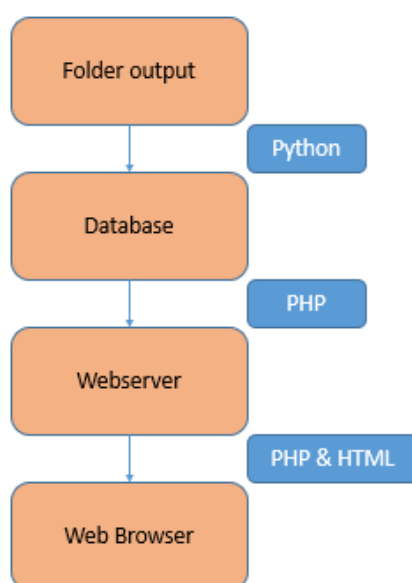
จากนั้นทดสอบการใช้งานซอฟต์แวร์ phpMyadmin โดยพิมพ์ localhost/phpmyadmin ที่ Address bar ของ Chromium จะปรากฏหน้าต่างดังภาพที่ 3-24



ภาพที่ 3-24 หน้าต่างใช้งานของซอฟต์แวร์บริการจัดการฐานข้อมูล PhpMyadmin

3.2.3.3 การแสดงผลแบบภาพนิ่ง (Snapshot) ที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคล

การแสดงผลแบบภาพนิ่งที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคล สามารถเข้าชมภาพนิ่งจากที่ใดก็ได้ตลอดเวลา มีรูปแบบการทำงานของ การแสดงผลภาพนิ่งผ่านเว็บเบราว์เซอร์ ดังภาพที่ 3-25



ภาพที่ 3-25 รูปแบบการทำงานของ การแสดงผลภาพนิ่งผ่านเว็บเบราว์เซอร์

รูปแบบการทำงานของ การแสดงผลภาพนิ่งผ่านเว็บเบราว์เซอร์ โดยนำภาพนิ่งที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคลที่อยู่ภายในโฟลเดอร์ที่ชื่อ GDrive จากหัวข้อที่ 3.2.2.4 แปลงเป็นข้อมูลเลขไบนารี (Binary Large Object : Blob) ก่อนเก็บไว้ที่ MySQL โดยใช้ phpMyadmin ในการเข้าถึง ซึ่งในส่วนนี้ใช้ภาษาไพธอนในการสั่งการ มีคำสั่งดังภาพที่ 3-26 และสามารถดูรายละเอียดข้อมูลใน phpMyadmin ดังภาพที่ 3-27 ส่วนถัดมาเป็นการนำข้อมูลเลขไบนารีแปลงกลับเป็นภาพนิ่ง และนำภาพนิ่งที่ได้ไปแสดงผลผ่านเว็บเบราว์เซอร์ ด้วยภาษา PHP และ HTML ผ่านบนเว็บเซิร์ฟเวอร์ให้ผู้ใช้งานสามารถเข้าผ่านเว็บเบราว์เซอร์ได้

```

import MySQLdb
import sys
from PIL import Image
import base64
import cStringIO
import PIL.Image
from os import listdir
hostname = "localhost";
user="root";
password ="1234"
mydb = "picture"
db = MySQLdb.connect(hostname,user,password,mydb)
if __name__ == "__main__":

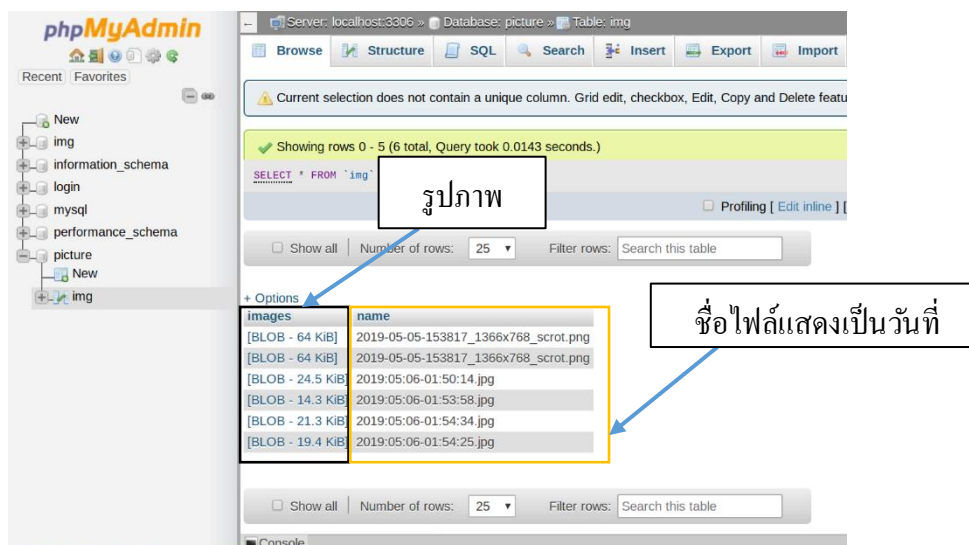
    # paths to input and output images
    input_path= "/home/pi/test/outpic/"
    input_names= listdir(input_path)
    for names in input_names:
        file =names
        image = Image.open(input_path+ names)
        blob_value = open(input_path+ names, 'rb').read()
        sql = 'INSERT INTO img(images,name) VALUES(%s,%s)'

        args = (blob_value,file)
        cursor=db.cursor()
        cursor.execute(sql,args)
        sql1='select * from img'
        db.commit()
        cursor.execute(sql1)
        data=cursor.fetchall()
        print type(data[0][0])
        file_like=cStringIO.StringIO(data[0][0])
        img=PIL.Image.open(file_like)
        img.show()

    db.close()

```

ภาพที่ 3-26 ไฟล์ไพธอนสำหรับการส่งข้อมูลจาก Raspberry Pi 3 ไปยัง phpMyadmin



ภาพที่ 3-27 การแสดงการใช้ phpMyAdmin ในการเก็บข้อมูลผ่านการจัดการ MySQL

### 3.2.3.4 การแสดงผลแบบถ่ายทอดสด

การแสดงผลแบบถ่ายทอดสด ทำให้ผู้ใช้งานสามารถเข้าชมได้ตลอดเวลา ซึ่งมีรูปแบบการทำงานดังนี้



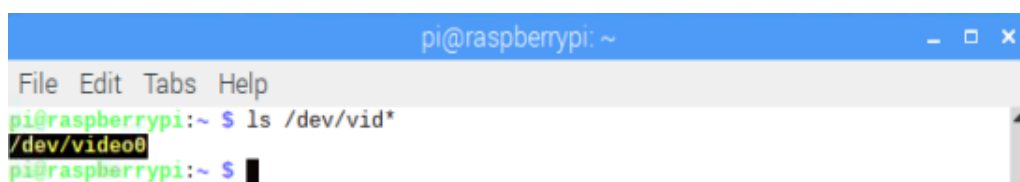
ภาพที่ 3-28 ขั้นตอนการแสดงผลแบบถ่ายทอดสด

เป็นการแสดงผลแบบถ่ายทอดสดแบบ VDO Streaming หรือเรียกว่า IP Camera ลักษณะเหมือนการถ่ายทอดสดผ่านเครือข่ายไร้สายด้วยเว็บแคม โดยติดตั้งให้บอร์ด Raspberry Pi 3 เป็น VDO Streaming Server สามารถเปิดชมการถ่ายทอดสดผ่านเว็บเบราว์เซอร์จากคอมพิวเตอร์หรือสมาร์ทโฟนได้ ซึ่งมีขั้นตอนใช้งานดังนี้

ก่อนติดตั้งซอฟต์แวร์ Motion ตรวจสอบระบบปฏิบัติการมองเห็นการเชื่อมต่อเว็บแคม ด้วยคำสั่ง

```
$ ls /dev/vid*
```

หากมองเห็นการเชื่อมต่อเว็บแคม จะปรากฏไครเทอริ Device video0 แสดงว่าใช้งานได้ดังภาพที่ 3-29



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ ls /dev/vid*
/dev/video0
pi@raspberrypi:~ $
```

ภาพที่ 3-29 หน้าต่างแสดงระบบปฏิบัติการมองเห็นการเชื่อมต่อเว็บแคม

ต่อมาทำการอัปเดตและอัปเดตระบบปฏิบัติการ และทำการติดตั้งซอฟต์แวร์ Motion ซึ่งเป็นซอฟต์แวร์ใน Streaming Server ด้วยคำสั่ง

```
$ sudo apt-get update && apt-get dist-upgrade -y
```

```
$ sudo apt-get install motion -y
```

แก้ไขไฟล์ motion.conf โดยป้อนคำสั่ง

```
$ sudo nano /etc/motion/motion.conf
```

แก้ไขจาก Daemon off เป็น Daemon on เพื่อให้ทำงานแบบ background

```
# Rename this distribution example file to motion.conf
#
# This config file was generated by motion 4.0

#####
# Daemon
#####

# Start in daemon (background) mode and release terminal (default: off)
daemon on
```

ภาพที่ 3-30 การแก้ไขไฟล์ motion.conf ในส่วน Daemon

จากนั้นแก้ไข stream\_localhost on เป็น stream\_localhost off เพื่อให้สามารถเปิดการถ่ายทอดสดจากเครื่องอื่นได้ ดังภาพที่ 3-31 สุดท้ายทำการดูหมายเลขพอร์ตของเซิร์ฟเวอร์เพื่อใช้งาน VDO Streaming และบันทึกการแก้ไข โดยกดปุ่ม Ctrl + X แล้วกดปุ่ม y ตามภาพที่ 3-32

```
# Output frames at 1 fps when no motion is detected and increase to the
# rate given by stream_maxrate when motion is detected (default: off)
stream_motion off

# Maximum framerate for stream streams (default: 1)
stream_maxrate 1

# Restrict stream connections to localhost only (default: on)
stream_localhost off
```

ภาพที่ 3-31 การแก้ไขไฟล์ motion.conf ในส่วน stream\_localhost

```
#####
# Enable IPv6 (default: off)
ipv6_enabled off

#####
# Live Stream Server
#####

# The mini-http server listens to this port for requests (default: 0 = disabled)
stream_port 8081
```

ภาพที่ 3-32 หมายเลขพอร์ตของเซิร์ฟเวอร์

ต่อมาทำการแก้ไขไฟล์ Motion โดยป้อนคำสั่ง

```
$ sudo nano /etc/default/motion
```

แก้ไขจาก start\_motion\_daemon = no เป็น start\_motion\_daemon = yes และบันทึกการแก้ไข โดยกดปุ่ม Ctrl + X แล้วกดปุ่ม y

```
# set to 'yes' to enable the motion daemon
start_motion_daemon=yes
```

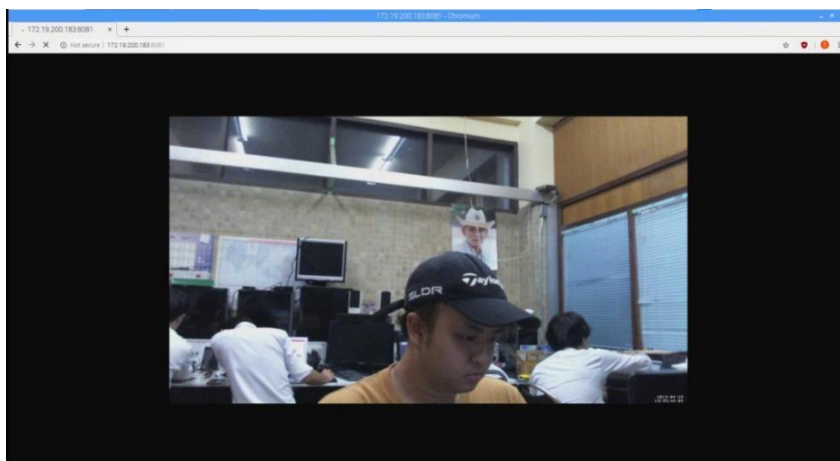
ภาพที่ 3-33 การแก้ไขไฟล์ motion ในส่วน stream\_motion\_daemon

เมื่อทำการตั้งค่าคอนฟิกซอฟต์แวร์ Motion เสร็จเรียบร้อยแล้วทั้ง 2 ไฟล์ให้รีสตาร์ทเซิร์ฟเวอร์ (Streaming Server) ด้วยคำสั่งด้านล่าง ในกรณีเลิกใช้งาน ต้องการหยุดเซิร์ฟเวอร์ พิมพ์คำสั่ง sudo service motion stop

```
$ sudo motion service restart
```

```
$ sudo motion
```

การแสดงผลแบบถ่ายทอดสด สามารถเข้าใช้งานโดยพิมพ์ IP Address ตามด้วยหมายเลขพอร์ตของเซิร์ฟเวอร์ในขั้นตอนก่อนหน้านี้ ซึ่งมีรูปแบบ เช่น 192.168.1.1:8081 ใน Address bar ของ Chromium จะปรากฏหน้าต่างแสดงผลแบบถ่ายทอดสด ดังภาพที่ 3-34



ภาพที่ 3-34 หน้าต่างแสดงผลแบบถ่ายทอดสด

### 3.2.4 การอัปโหลดภาพนิ่งขึ้นเว็บฝากไฟล์

ภาพนิ่งที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคลแล้ว จะถูกบันทึกในฐานข้อมูลก่อนอัปโหลดขึ้นเว็บฝากไฟล์ ซึ่งอัปโหลดขึ้นเว็บฝากไฟล์ด้วยวิธีการย้ายไฟล์ภาพนิ่งจากโฟลเดอร์ภาพนิ่งที่ผ่านการประมวลผลไปเก็บในโฟลเดอร์ที่สร้างขึ้นใน Google Drive เหตุที่ทำให้การย้ายไฟล์ภาพนิ่งแทนการคัดลอก เพื่อเป็นการเคลียร์ไฟล์ภาพนิ่งในบอร์ด Raspberry Pi 3

การอัปโหลดภาพนิ่งขึ้นเว็บฝากไฟล์ อธิบายถึงอุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง วิธีการติดตั้งซอฟต์แวร์ Rclone รวมถึงตั้งค่าการใช้งาน และคำสั่งการอัปโหลดภาพนิ่งขึ้นเว็บฝากไฟล์

#### 3.2.4.1 อุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้อง

เซตอุปกรณ์ในการใช้งาน Raspberry Pi 3 แบบต่อจอ (ดูได้จากอุปกรณ์ที่ใช้ในการติดตั้ง OS ในภาคผนวก) ซอฟต์แวร์ Rclone รวมถึงชื่อบัญชีและรหัสผ่านของบัญชีผู้ใช้งาน Google สำหรับอัปโหลดภาพนิ่ง

#### 3.2.4.2 วิธีการติดตั้งซอฟต์แวร์ Rclone

ก่อนทำการติดตั้งซอฟต์แวร์ให้อัปเดตระบบปฏิบัติการ เมื่ออัปเดตเสร็จเรียบร้อยแล้วทำการติดตั้งซอฟต์แวร์ rclone โดยป้อนคำสั่งดังนี้

```
$ sudo apt-get update
$ sudo apt install rclone
```

หลังจากติดตั้งเสร็จเรียบร้อยแล้ว เข้าไปยังโฟลเดอร์ที่ใช้บันทึกภาพนิ่งที่ผ่านการประมวลผลการตรวจหาความน่าจะเป็นใบหน้าบุคคล ชื่อว่า GDrive จากหัวข้อที่ 3.2.2.4 เพื่อใช้โฟลเดอร์ GDrive ในการซิงค์กับ Google drive จากนั้นตั้งค่า Rclone ตามคำสั่ง

```
$ cd GDrive
$ rclone config
```

พิมพ์ n เพื่อตั้งค่าการเชื่อมต่อระยะไกลใหม่ (new remote) จากนั้นกดปุ่ม Enter ระบบจะให้ใส่ชื่อ ตั้งชื่อว่า “remote” ดังภาพที่ 3-35 แล้วกดปุ่ม Enter

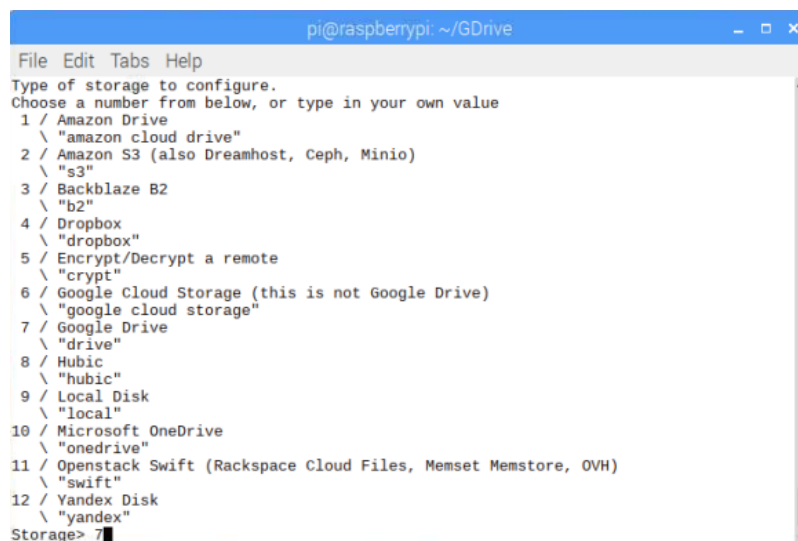


```
pi@raspberrypi: ~/GDrive
File Edit Tabs Help
pi@raspberrypi:~ $ cd GDrive
pi@raspberrypi:~/GDrive $ rclone config
2019/03/06 12:47:40 Config file "/home/pi/.rclone.conf" not found - using defaults
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n
name> remote
```

ภาพที่ 3-35 การตั้งชื่อการเชื่อมต่อระยะไกล

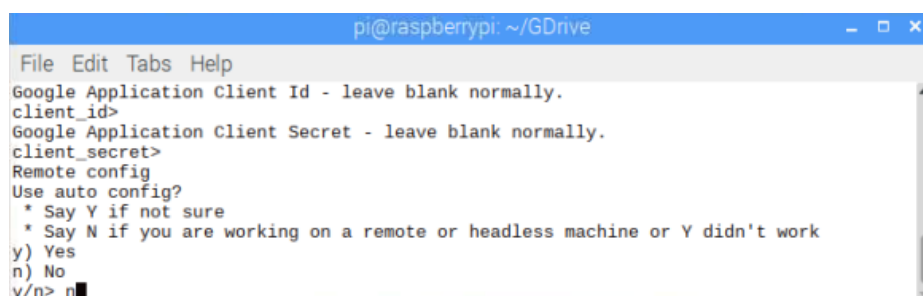
หลังจากนั้นระบบจะแสดงประเภทของ Cloud storage ทั้งหมดที่ซอฟต์แวร์ Rclone สามารถซิงค์ได้ เลือกหมายเลข 7 Google drive แล้วกดปุ่ม Enter





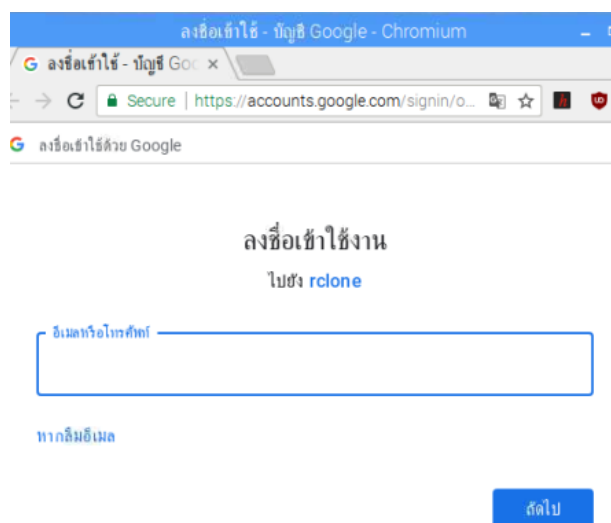
ภาพที่ 3-36 ประเภทของ Cloud storage ที่ซอฟต์แวร์ Rclone สามารถซิงค์ได้

ระบบขึ้นข้อความให้ระบุ client\_id และ client\_secret ไม่ต้องระบุใด ๆ ให้กดปุ่ม Enter ข้ามขั้นตอนนี้ และพิมพ์ n เพื่อยืนยันไม่ใช้การตั้งค่าอัตโนมัติ ดังภาพที่ 3-37 จากนั้นกดปุ่ม Enter



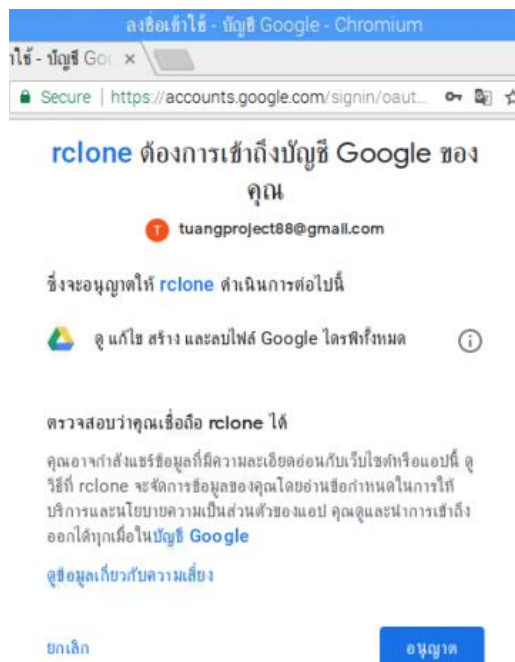
ภาพที่ 3-37 เลือกประเภทการตั้งค่า

หน้าต่างเว็บเบราว์เซอร์เข้าใช้งาน Google เพื่อซิงค์ไปยังซอฟต์แวร์ Rclone จะแสดงขึ้นอัตโนมัติ ดังภาพที่ 3-38 หากหน้าต่างเว็บเบราว์เซอร์ไม่แสดงอัตโนมัติสามารถกดคลิกจากเทอร์มินัล



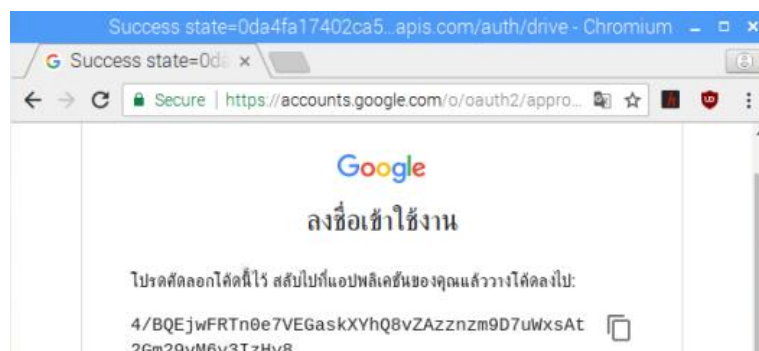
ภาพที่ 3-38 หน้าต่างเว็บเบราว์เซอร์

ทำการป้อนชื่อบัญชีและรหัสผ่านของบัญชีผู้ใช้งาน Google จากนั้นกด อนุญาต เพื่อให้ซอฟต์แวร์ Rclone สามารถเข้าถึงบัญชี Google ได้



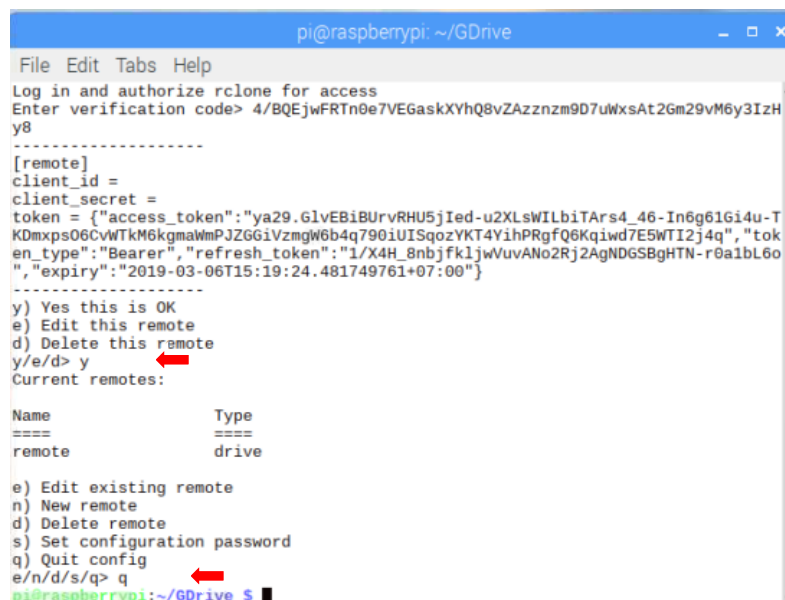
ภาพที่ 3-39 หน้าต่างการอนุญาตซอฟต์แวร์ rclone สามารถเข้าถึงบัญชีผู้ใช้ Google

หน้าต่างถัดไปแสดงโค้ดที่ใช้ในการเข้าถึงบัญชีผู้ใช้งาน Google ดังภาพที่ 3-40  
ทำการคัดลอกโค้ดและกลับไปเทอร์มินัล วางโค้ดแล้วกดปุ่ม Enter



ภาพที่ 3-40 หน้าต่างแสดงโค้ด

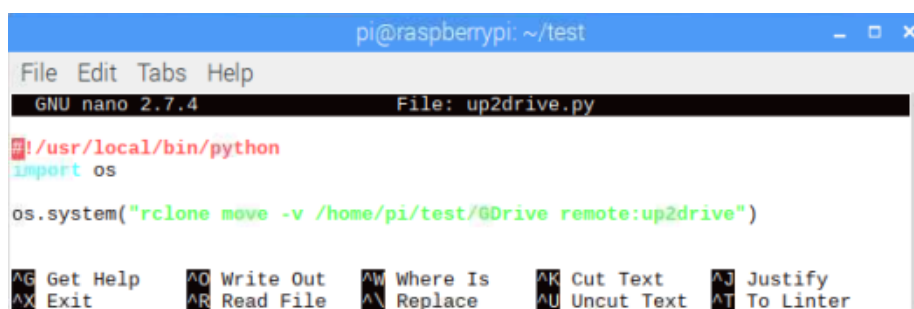
จากนั้นพิมพ์ y แล้วกดปุ่ม Enter ระบบแสดงรายละเอียดการตั้งค่าซอฟต์แวร์  
Rclone ตรวจสอบความถูกต้องก่อนพิมพ์ q แล้วกดปุ่ม Enter



ภาพที่ 3-41 ตรวจสอบการตั้งค่าซอฟต์แวร์ rclone

### 3.2.4.3 คำสั่งการอัปโหลดภาพขึ้นเว็บฝากไฟล์

โปรแกรมการอัปโหลดภาพขึ้นเว็บฝากไฟล์ บันทึกในไฟล์ชื่อ up2drive.py ใช้ซอฟต์แวร์ rclone ในการย้ายไฟล์ภาพนิ่งในโฟลเดอร์ GDrive ไปเก็บในโฟลเดอร์ up2drive ที่สร้างขึ้นใน Google drive ด้วยคำสั่ง `rclone move -v /home/pi/test/GDrive remote:up2drive` ดังภาพที่ 3-42



```

pi@raspberrypi: ~/test
File Edit Tabs Help
GNU nano 2.7.4 File: up2drive.py
#!/usr/local/bin/python
import os

os.system("rclone move -v /home/pi/test/GDrive remote:up2drive")

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit          ^R Read File    ^_ Replace      ^U Uncut Text   ^T To Linter
  
```

ภาพที่ 3-42 คำสั่งการอัปโหลดภาพขึ้น Google drive ในไฟล์ up2drive.py

ดังนั้น โปรแกรมการทำงานของระบบที่รวมการทำงานของแต่ละ ส่วนจากหัวข้อที่ 2.3.1 – 2.3.4 โดยใช้โปรแกรมการเชื่อมต่อออร์ร่วมกับเว็บแคมและเซ็นเซอร์ตรวจจับความเคลื่อนไหว ในหัวข้อที่ 2.3.1 เป็นฐานในการเชื่อมต่อการทำงานส่วนอื่น ๆ ด้วยคำสั่ง `os.system()` ซึ่งเป็นคำสั่งสำหรับการรันคำสั่งคอมมานด์ไลน์ บันทึกในไฟล์ชื่อ system.py ดังภาพที่ 3-43 สามารถดูคำสั่งของโปรแกรมในแต่ละส่วนและโปรแกรมการทำงานของระบบได้ในภาคผนวก ก

```

pi@raspberrypi: ~/test
File Edit Tabs Help
GNU nano 2.7.4 File: system.py Modified

import RPi.GPIO as GPIO
import cv2
import time
import os
from subprocess import call
from datetime import datetime
import cv2.cv as cv
from os import listdir
GPIO.setmode(GPIO.BCM)
GPIO_PIR = 7
GPIO.setup(GPIO_PIR, GPIO.IN)
Current_State = 0
Previous_State = 0
try:
    print "Waiting for PIR to settle"
    while GPIO.input(GPIO_PIR)==1:
        Current_State = 0
    print "Ready"
    while True :
        Current_State = GPIO.input(GPIO_PIR)
        if Current_State==1 and Previous_State==0:
            print "Motion detected!"
            Previous_State=1

            for x in range(0,3):
                os.system("fswebcam -r 400x400 -F 5 /home/pi/te$
                time.sleep(2)
            os.system("python detect.py")
            time.sleep(1)
            os.system("python up2mysql.py")
            time.sleep(1)
            os.system("python up2drive.py")
            time.sleep(1)

        elif Current_State==0 and Previous_State==1:
            print "Ready"
            Previous_State=0
            time.sleep(2)

except KeyboardInterrupt:
    print "Quit"
GPIO.cleanup

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line

```

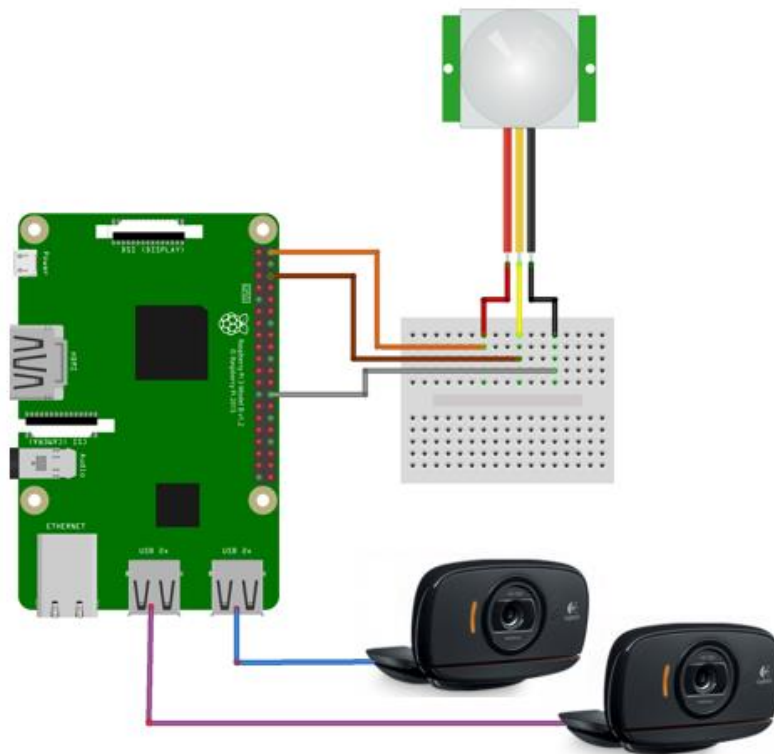
ภาพที่ 3-43 คำสั่งของระบบ ในไฟล์ system.py

## บทที่ 4

### การทดลองและวิจารณ์ผล

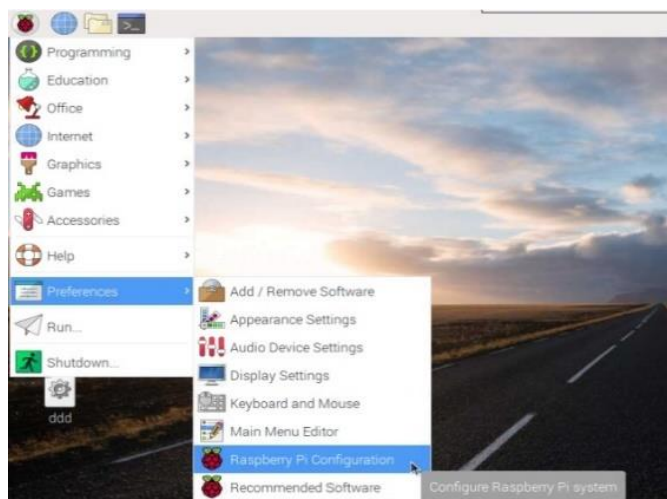
ระบบกล้องสังเกตการณ์อัจฉริยะขนาดเล็กเป็นระบบที่รวมการทำงานในส่วนของการเชื่อมต่อบอร์ดร่วมกับเว็บแคมและเซ็นเซอร์ตรวจจับความเคลื่อนไหว การประมวลผลตรวจจับใบหน้าบุคคล การอัปโหลดภาพขึ้นเว็บฝากไฟล์ และการแสดงผลผ่านเว็บเบราว์เซอร์ ซึ่งมีการทำงานของระบบดังนี้

เริ่มจากติดตั้งอุปกรณ์เว็บแคมและเซ็นเซอร์ตรวจจับความเคลื่อนไหวเชื่อมต่อกับบอร์ด Raspberry Pi 3 ดังภาพที่ 4-1



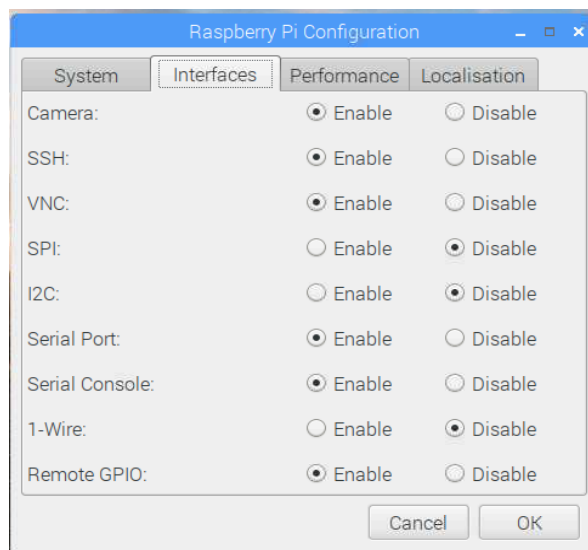
ภาพที่ 4-1 การเชื่อมต่ออุปกรณ์ร่วมกับบอร์ด Raspberry Pi 3

เมื่อติดตั้งอุปกรณ์เรียบร้อยแล้ว ทำการเปิดใช้งานเว็บแคม โดยกดเลือก แถบเมนู > เลือก Preferences > เลือก Raspberry Pi Configuration ตามภาพที่ 4-2



ภาพที่ 4-2 การเข้าสู่หน้าต่าง Raspberry Pi Configuration

จะปรากฏหน้าต่างดังภาพที่ 4-3 เลือกแถบ Interfaces เปลี่ยน Camera จาก Disable เป็น Enable แล้วกด OK จากนั้นระบบปฏิบัติการแจ้งให้ทำการ Reboot กด OK

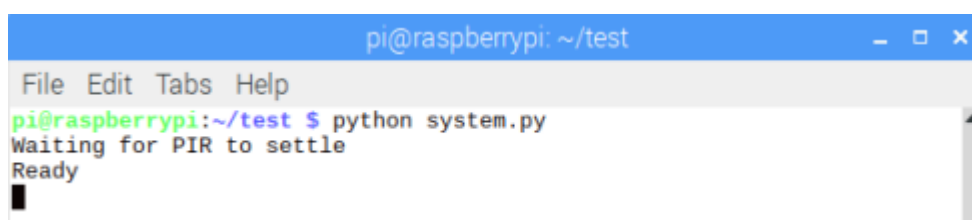


ภาพที่ 4-3 หน้าต่าง Raspberry Pi Configuration

เมื่อระบบปฏิบัติการ Reboot เสร็จเรียบร้อยแล้ว ทดลองการทำงานของระบบ โดยเข้าไปที่เก็บไฟล์ system.py อยู่ในโฟลเดอร์ test ก่อนรันสคริปไพธอน ด้วยคำสั่ง

```
$ cd test
$ python system.py
```

ระบบเริ่มทำงาน จะแสดงสถานะพร้อมทำงานของระบบและข้อความเริ่มการตรวจจับความเคลื่อนไหวของเซ็นเซอร์ตรวจจับความเคลื่อนไหว ดังภาพที่ 4-4



```
pi@raspberrypi: ~/test
File Edit Tabs Help
pi@raspberrypi:~/test $ python system.py
Waiting for PIR to settle
Ready
█
```

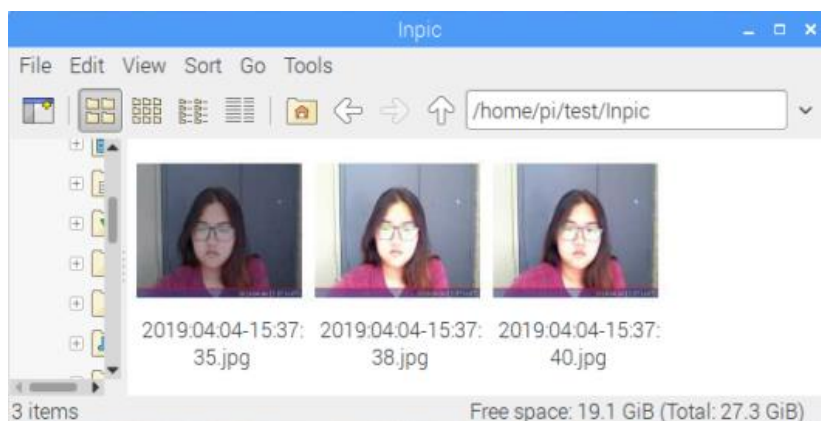
ภาพที่ 4-4 หน้าต่างแสดงสถานะพร้อมทำงานของระบบและข้อความเริ่มการตรวจจับ

เมื่อเซ็นเซอร์ตรวจจับความเคลื่อนไหวตรวจพบความเคลื่อนไหวของวัตถุ ระบบจะแสดงข้อความตรวจพบความเคลื่อนไหว (Motion detected!) ทำการถ่ายภาพนิ่ง จำนวน 3 ภาพ โดยการถ่ายภาพนิ่งแต่ละครั้งจะแสดงรายละเอียดภาพนิ่ง ชื่อไฟล์และพาธที่บันทึก ดังภาพที่ 4-5 ซึ่งภาพนิ่งถูกบันทึกในโฟลเดอร์ Inpic ดังภาพที่ 4-6

```
Waiting for PIR to settle
Ready
Motion detected!
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 400x400 to 352x288.
--- Capturing 5 frames...
Captured 5 frames in 0.26 seconds. (19 fps)
--- Processing captured image...
Writing JPEG image to '/home/pi/test/Inpic/2019:04:04-15:08:14.jpg'.
```

ภาพที่ 4-5 หน้าต่างแสดงข้อความตรวจพบความเคลื่อนไหวและรายละเอียดการถ่ายภาพนิ่ง





ภาพที่ 4-6 ภาพนิ่งถูกถ่ายจากการตรวจพบความเคลื่อนไหวของเซ็นเซอร์

จากนั้นภาพนิ่งในโฟลเดอร์ Inpic แต่ละภาพนิ่งจะถูกนำไปประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคล หากตรวจพบบริเวณที่น่าจะเป็นใบหน้าบุคคล ระบบจะแสดงพิกัดบริเวณใบหน้าบุคคลนั้น ระบบทำการประมวลผลทีละภาพนิ่งจนครบทุกภาพนิ่งที่อยู่ในโฟลเดอร์ Inpic ดังภาพที่ 4-7

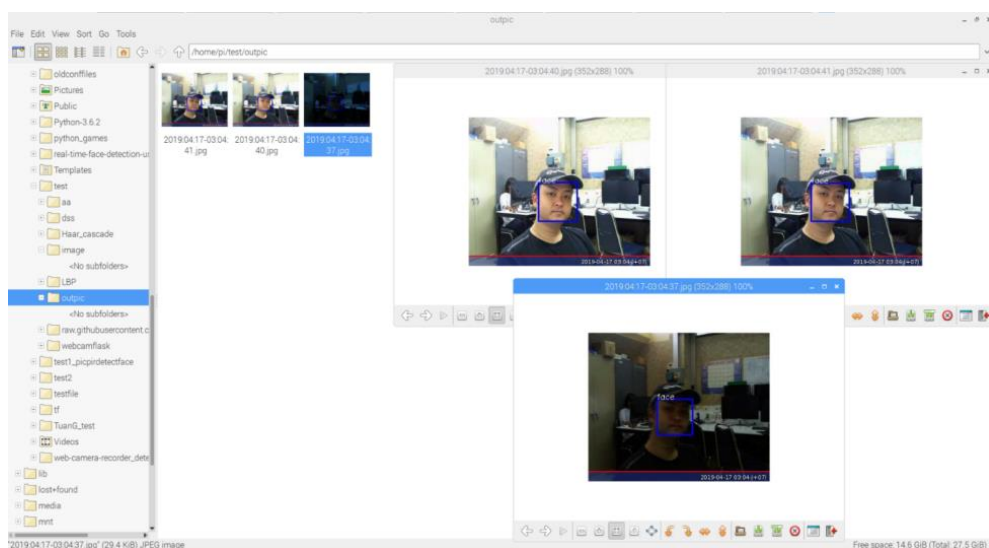
```

--- Capturing 5 frames...
Captured 5 frames in 0.18 seconds. (27 fps)
--- Processing captured image...
Writing JPEG image to '/home/pi/test/image/2019:04:17-03:04:41.jpg'.
Starting to detect faces in images and save the cropped images to output file...
/home/pi/test/image/2019:04:17-03:04:41.jpg
bBoxes_frontal_face::
[[136 127 74 74]]
bBoxes_frontal_profileface::
()
/home/pi/test/image/2019:04:17-03:04:37.jpg
bBoxes_frontal_face::
[[133 128 69 69]]
bBoxes_frontal_profileface::
()
/home/pi/test/image/2019:04:17-03:04:40.jpg
bBoxes_frontal_face::
[[135 126 75 75]]
bBoxes_frontal_profileface::
()
Ready

```

ภาพที่ 4-7 การแสดงพิกัดบริเวณที่น่าจะเป็นใบหน้าบุคคล

ภาพนิ่งที่ผ่านการประมวลผลตรวจจับใบหน้าบุคคลถูกบันทึกในโฟลเดอร์ GDrive ซึ่งจะมีกรอบสี่เหลี่ยมสีน้ำเงินล้อมรอบบริเวณที่น่าจะเป็นใบหน้าบุคคล ดังภาพที่ 4-8

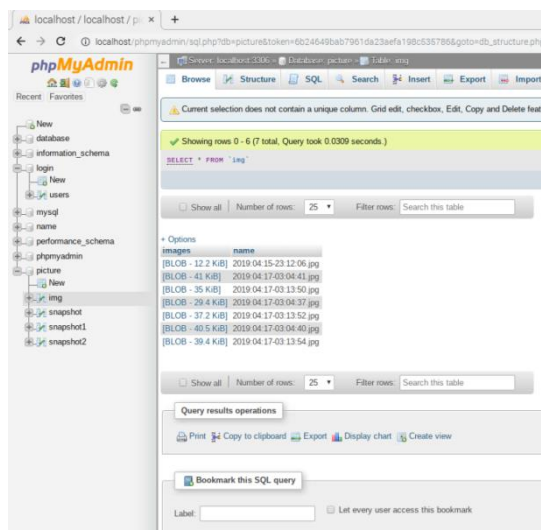


ภาพที่ 4-8 ภาพนิ่งที่ผ่านการตรวจจับใบหน้าบุคคล

ภาพนิ่งที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคลจะถูกแสดงผลผ่านเว็บเบราว์เซอร์ โดยการส่งภาพนิ่งไปเก็บในฐานข้อมูล MySQL ที่สร้างไว้ ในหัวข้อที่ 3.2.3.2 สามารถตรวจสอบการเก็บภาพนิ่งในฐานข้อมูลผ่านผลการการทำงานของระบบ ดังภาพที่ 4-9 หรือระบบการจัดการฐานข้อมูล phpMyAdmin ซึ่งแสดงตารางการเก็บข้อมูลภาพนิ่งและชื่อภาพนิ่ง ดังภาพที่ 4-10

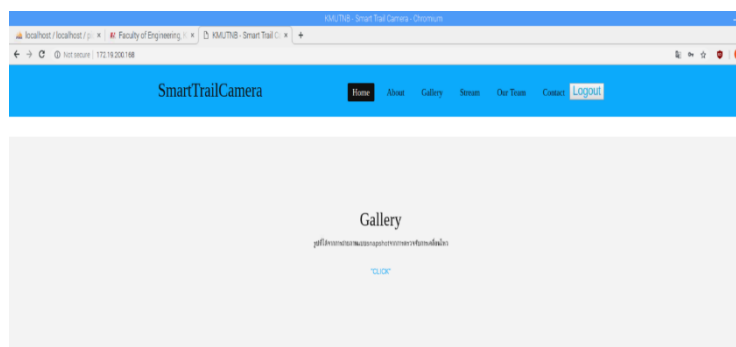
```
bBoxes_frontal_face::
[[118 140 80 80]]
bBoxes_frontal_profileface::
()
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
Ready
```

ภาพที่ 4-9 หน้าต่างแสดงผลการทำงานของระบบ

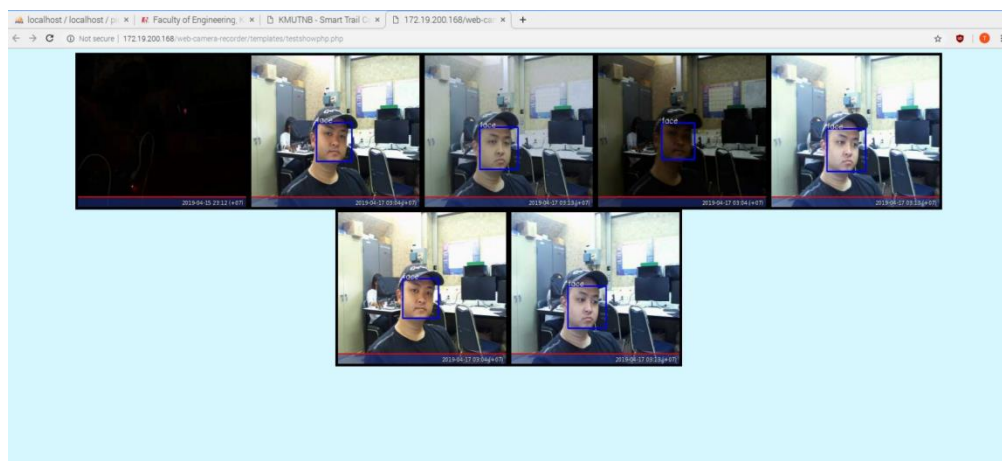


ภาพที่ 4-10 หน้าต่างระบบการจัดการฐานข้อมูล phpMyAdmin

จากนั้นทำการเข้าหน้าเว็บเบราว์เซอร์หลัก โดยพิมพ์ IP Address ของบอร์ด Raspberry Pi 3 ใน Address bar ของ Chromium จะปรากฏหน้าต่างเว็บ SmartTrailCamera เลื่อนหน้าต่างจนพบหัวข้อ gallery คลิกปุ่ม “คลิก” เพื่อเข้าสู่หน้าต่างแสดงภาพหนึ่งทีผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคล ดังภาพที่ 4-11 และ 4-12 ตามลำดับ อีกทั้งภาพหนึ่งจะอัปเดตตลอดเวลาผ่านการกด Refresh ที่หน้าต่าง

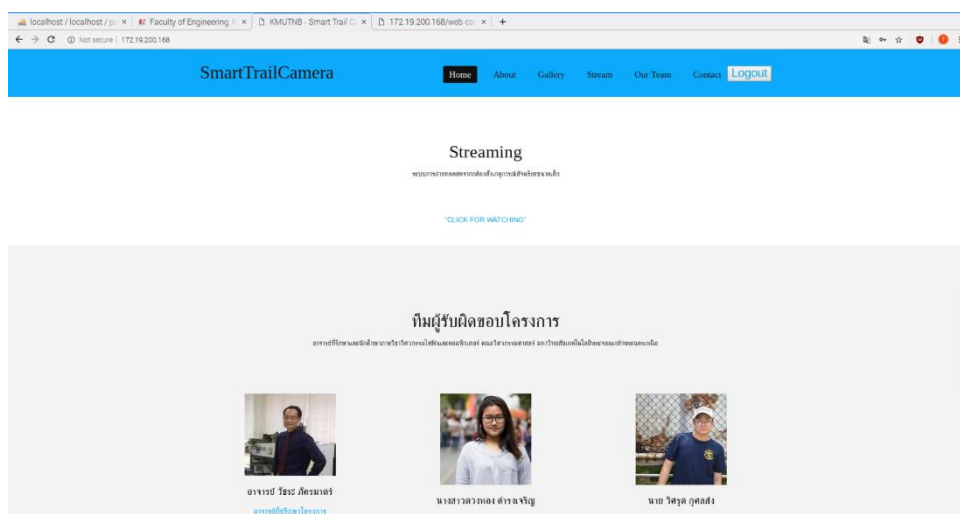


ภาพที่ 4-11 หน้าต่างเว็บ SmartTrailCamera

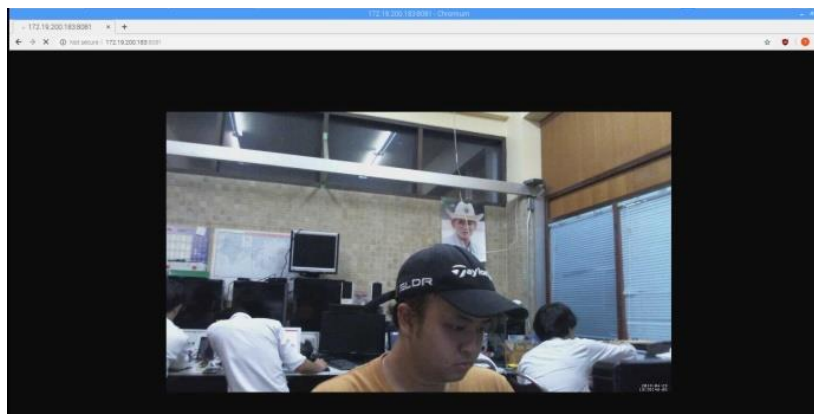


ภาพที่ 4-12 หน้าต่างแสดงผลภาพนิ่งที่ผ่านการประมวลผลหาใบหน้าบุคคล

นอกจากการแสดงผลภาพนิ่งที่ผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคล ยังมีการแสดงผลแบบถ่ายทอดสดผ่านเว็บเบราว์เซอร์ ซึ่งเป็นการใช้งานเว็บแคมอีกตัวหนึ่งสำหรับถ่ายทอดสดโดยเฉพาะ เปิดใช้งานอัตโนมัติเมื่อป้อนไฟเลี้ยงให้บอร์ด Raspberry Pi 3 ในการแสดงผลแบบถ่ายทอดสดทำการพิมพ์ IP Address ของบอร์ด Raspberry Pi 3 ใน Address bar ของ Chromium จะปรากฏหน้าต่างเว็บ SmartTrailCamera เลื่อนหน้าต่างจนพบหัวข้อ streaming คลิกปุ่ม “click for streaming” จะปรากฏหน้าต่างแสดงผลการถ่ายทอดสด ดังภาพที่ 4-13 และ 4-14 ตามลำดับ



ภาพที่ 4-13 หน้าต่างหลัก

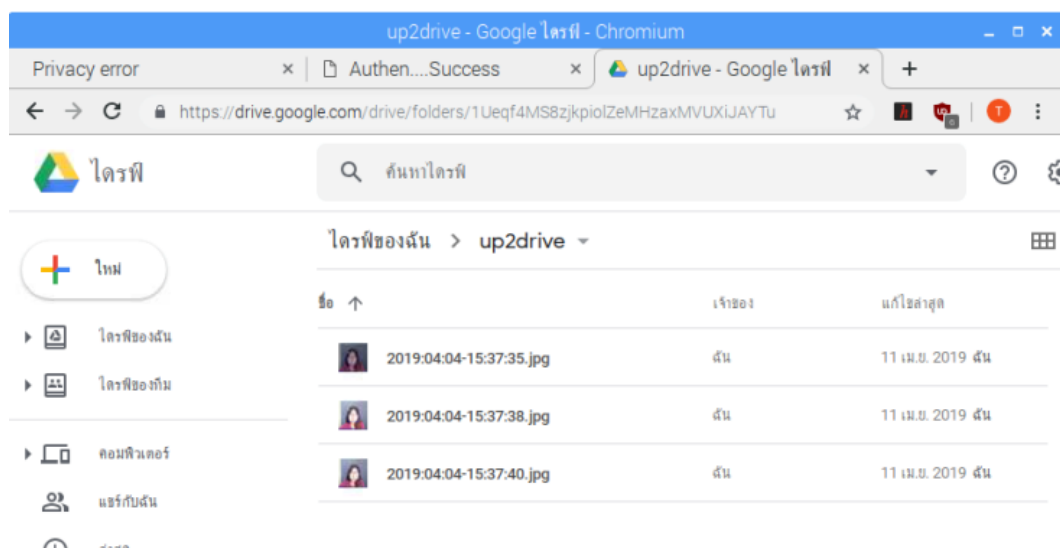


ภาพที่ 4-14 หน้าต่างแสดงผลการถ่ายทอดสด

ท้ายสุดการทำงานของระบบจะนำภาพหนึ่งทีผ่านการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคลอัปโหลดขึ้นเว็บฝากไฟล์ โดยภาพหนึ่งในโฟลเดอร์ GDrive จะถูกย้ายไปเก็บในโฟลเดอร์ up2drive ซึ่งเป็นโฟลเดอร์ที่สร้างขึ้นใน Google drive ระบบจะแสดงรายละเอียดการทำงานตั้งแต่การอัปโหลดภาพหนึ่ง รวมถึงการลบภาพหนึ่งในโฟลเดอร์ ดังภาพที่ 4-15 สามารถตรวจสอบการอัปโหลดภาพหนึ่งไปยัง Google drive ดังภาพที่ 4-16

```
2019/04/17 21:27:42 rclone: Version "v1.35-DEV" starting with parameter
s ["rclone" "move" "-v" "/home/pi/test/GDrive" "remote:up2drive"]
2019/04/17 21:27:43 Time may be set wrong - time from "accounts.google.
com" is -14h39m47.565255923s different from this computer
2019/04/17 21:27:43 remote: Saved new token in config file
2019/04/17 21:27:44 Time may be set wrong - time from "www.googleapis.c
om" is -14h39m47.961879726s different from this computer
2019/04/17 21:27:44 Google drive root 'up2drive': Modify window is 1ms
2019/04/17 21:27:44 Google drive root 'up2drive': Reading ""
2019/04/17 21:27:44 Google drive root 'up2drive': Finished reading ""
2019/04/17 21:27:44 Google drive root 'up2drive': Waiting for checks to
finish
2019/04/17 21:27:44 Google drive root 'up2drive': Waiting for transfers
to finish
2019/04/17 21:27:47 2019:04:04-15:37:40.jpg: Copied (new)
2019/04/17 21:27:47 2019:04:04-15:37:40.jpg: Deleted
2019/04/17 21:27:47 2019:04:04-15:37:35.jpg: Copied (new)
2019/04/17 21:27:47 2019:04:04-15:37:35.jpg: Deleted
2019/04/17 21:27:47 2019:04:04-15:37:38.jpg: Copied (new)
2019/04/17 21:27:47 2019:04:04-15:37:38.jpg: Deleted
2019/04/17 21:27:48 g-dragon-1.jpg: Copied (new)
2019/04/17 21:27:48 g-dragon-1.jpg: Deleted
2019/04/17 21:27:48
Transferred: 232.128 kBytes (43.483 kBytes/s)
Errors: 0
Checks: 4
Transferred: 4
Elapsed time: 5.3s
2019/04/17 21:27:48 Go routines at exit 13
```

ภาพที่ 4-15 แสดงรายละเอียดการอัปโหลดภาพหนึ่งขึ้นเว็บฝากไฟล์



ภาพที่ 4-16 การอัปโหลดภาพนิ่งไปยัง Google drive

## บทที่ 5

### สรุป วิจัย และข้อเสนอแนะ

#### 5.1 สรุป

จากวัตถุประสงค์ของโครงการนี้เป็นการศึกษา และพัฒนาอุปกรณ์การถ่ายภาพให้มีประสิทธิภาพมากยิ่งขึ้น สามารถถ่ายภาพนิ่งตรวจจับวัตถุที่เคลื่อนไหวผ่านเซ็นเซอร์ตรวจจับความเคลื่อนไหว การประมวลผลภาพนิ่งตรวจหาความน่าจะเป็นใบหน้าบุคคล การแสดงผลภาพนิ่งและวิดีโอถ่ายทอดสดผ่านเว็บเบราว์เซอร์ รวมทั้งการอัปโหลดภาพนิ่งขึ้นเว็บฝากไฟล์เพื่อแบ็กอัพข้อมูล เพื่อให้ผู้ใช้งานสามารถเปิดดูภาพนิ่งหรือเปิดชมการถ่ายทอดสดจากที่ใดก็ได้ตลอดเวลา ซึ่งโครงการนี้เป็นการศึกษาการทำงานร่วมกันของเว็บแคม เซ็นเซอร์ตรวจจับความเคลื่อนไหวและบอร์ด Raspberry Pi 3 ผู้ที่มีความสนใจในด้านการถ่ายภาพนิ่งร่วมกับเซ็นเซอร์ การทำเว็บเซิร์ฟเวอร์ การตรวจจับใบหน้าบุคคลหรือด้านอื่น ๆ ที่มีความเกี่ยวข้อง สามารถนำความรู้หรือแนวคิดเบื้องต้นจากโครงการนี้ไปใช้ให้เกิดประโยชน์ได้ต่อไป

#### 5.2 วิจัยผลการทดลอง

จากผลการทดลองและปัญหาที่เกิดขึ้นพบว่าความไวในการตรวจจับความเคลื่อนไหวไม่ไวพอที่จะแยกสองวัตถุที่เคลื่อนผ่านเซ็นเซอร์ตรวจจับความเคลื่อนไหวในเวลาใกล้เคียงกัน การถ่ายภาพนิ่งอย่างต่อเนื่อง เช่น จำนวน 3 ภาพของเว็บแคม ไม่สามารถถ่ายได้ในบางครั้งทำให้ถ่ายภาพขาดหรือถ่ายไม่ครบตามจำนวนที่ต้องการ รวมถึงการถ่ายทอดสดหรือวิดีโอสตรีมมิงช้า มีดีเลย์สูง ปัญหาทั้งหมดนี้อาจเนื่องมาจากการทำงานทั้งหมดทำงานบนบอร์ด Raspberry Pi 3 ซึ่งเป็นบอร์ดคอมพิวเตอร์ขนาดเล็กมีการประมวลผลไม่สูงมากนัก ทำให้ประมวลผลไม่ทัน นอกจากนี้ในการแบ็กอัพข้อมูลหรือการแสดงผลผ่านเว็บเบราว์เซอร์ต้องใช้อินเทอร์เน็ตในการทำงาน หากบริเวณที่นำระบบไปใช้งานจริงมีสัญญาณอินเทอร์เน็ตไม่คงที่หรือไม่เสถียรพอจะทำให้เกิดความผิดพลาดได้ ฉะนั้นควรทำการทดสอบสัญญาณอินเทอร์เน็ตก่อนนำระบบไปใช้งาน

### 5.3 ข้อเสนอแนะ

5.3.1 ปรับปรุงเซ็นเซอร์หรือเปลี่ยนเซ็นเซอร์ให้มีความไวเพิ่มขึ้น

5.3.2 ปรับปรุงเว็บแคมหรือเลือกใช้เว็บแคมที่มีความคมชัดในการถ่ายภาพ

5.3.3 เลือกใช้อุปกรณ์ที่มีการประมวลผลข้อมูลสูง

5.3.4 เลือกไฟลัซด์เทรนนตรวจจับใบหน้าบุคคลที่ได้จากการเทรนนภาพ Positive และ Negative จำนวนมากเพื่อความแม่นยำในการประมวลผลตรวจหาความน่าจะเป็นใบหน้าบุคคลของภาพนิ่งที่ถ่ายได้



### เอกสารอ้างอิง

1. The Wikimedia Foundation, Inc. 2018. (Online). Available from :  
[https://en.wikipedia.org/wiki/Grayscale#Converting\\_color\\_to\\_grayscale](https://en.wikipedia.org/wiki/Grayscale#Converting_color_to_grayscale) . Research date:  
 2 Nov. 2018
2. รุสดี สุทธิวิรุณ และวิลเลพร แซ่ลี. 2011. (Online). Available from :  
<http://ejournals.swu.ac.th/index.php/SwuENGj/article/viewFile/2306/2348>. Research  
 date: 2 Nov. 2018
3. Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of  
 Simple Features. 2001. (Online). Available from :  
[http://wearables.cc.gatech.edu/paper\\_of\\_week/viola01rapid.pdf](http://wearables.cc.gatech.edu/paper_of_week/viola01rapid.pdf). Research date: 2 Nov.  
 2018

ภาคผนวก ก

Source Code

system.py

```
#!/bin/sh

import RPi.GPIO as GPIO
import cv2
import time
import os
from subprocess import call
from datetime import datetime
import cv2.cv as cv
from os import listdir
GPIO.setmode(GPIO.BCM)
GPIO_PIR = 7
GPIO.setup(GPIO_PIR,GPIO.IN)
Current_State = 0
Previous_State = 0
try:
    print"Waiting for PIR to settle"
    while GPIO.input(GPIO_PIR)==1:
        Current_State = 0
    print"Ready"
    while True :
        Current_State = GPIO.input(GPIO_PIR)
        if Current_State==1 and Previous_State==0:
            print"Motion detected!"
            Previous_State=1
```

```
        for x in range(0,3):
            os.system("fswebcam -r 400x400 -F 5 /home/pi/test/Inpic/%Y:%m:%d-%H:%M:%S.jpg")
            time.sleep(2)
            os.system("python detect.py")
            time.sleep(1)
            os.system("python up2mysql.py")
            time.sleep(1)
            os.system("python up2drive.py")
            time.sleep(1)

        elif Current_State==0 and Previous_State==1:
            print"Ready"
            Previous_State=0
            time.sleep(2)

    except KeyboardInterrupt:
        print"Quit"

GPIO.cleanup
```

detect.py

```
import cv2

import cv2.cv as cv

from os import listdir

import time


def detection(frame,color):

    # detection frontal_face

    bBoxes_frontal_face_default=frontal_face_default.detectMultiScale(frame,
scaleFactor=1.3, minNeighbors=5, minSize=(30,30), flags =
cv.CV_HAAR_SCALE_IMAGE)

    print("bBoxes_frontal_face::")

    print(bBoxes_frontal_face_default)


    for (x,y,w,h) in bBoxes_frontal_face_default:

        color_reg= cv2.rectangle(color,(x,y),(x+w,y+h),(255,0,0),2)

        cv2.putText(color,"face", (x,y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255,
255))

        saveCropped(color,name)

def saveCropped(img, name):

    cv2.imwrite(output_path+ name , img)

if __name__ == "__main__":

    # paths to input and output images

    input_path= "/home/pi/test/Inpic/"

    output_path= "/home/pi/test/GDrive/"

    input_names= listdir(input_path)
```

```
frontal_face_default=
cv2.CascadeClassifier("/home/pi/test/train/haarcascade_frontalface_default.xml")
print("Starting to detect faces in images and save the cropped images to output file...")
stime= time.clock()

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)
    detection(gray_img,color_img)
```

## Up2mysql.py

```
import MySQLdb
import sys
from PIL import Image
import base64
import cStringIO
import PIL.Image
from os import listdir

hostname = "localhost";
user="root";
password = "1234"
mydb = "picture"
db = MySQLdb.connect(hostname,user,password,mydb)

if __name__ == "__main__":
    # paths to input and output images
    input_path= "/home/pi/test/outpic/"
    input_names= listdir(input_path)

    for names in input_names:
        file =names

        image = Image.open(input_path+ names)

        blob_value = open(input_path+ names, 'rb').read()

        sql = 'INSERT INTO img(images,name) VALUES(%s,%s)'

        args = (blob_value,file)

        cursor=db.cursor()

        cursor.execute(sql,args)

        sql1='select * from img'

        db.commit()
```

```
cursor.execute(sql1)

    data=cursor.fetchall()

    print type(data[0][0])

    file_like=cStringIO.StringIO(data[0][0])

    img=PIL.Image.open(file_like)

    img.show()

db.close()
```



## Up2drive.py

```
#!/usr/local/bin/python  
import os  
  
os.system("rclone move -v /home/pi/test/GDrive remote:up2drive")
```

ภาคผนวก ข

การติดตั้งระบบปฏิบัติการ

## การติดตั้งระบบปฏิบัติการสำหรับ Raspberry pi

Raspberry pi เป็นคอมพิวเตอร์ที่จำเป็นต้องมีระบบปฏิบัติการ (Operating System: OS) ซึ่งตัว Raspberry pi ใช้ micro SD Card เป็นหน่วยความจำหลักในการติดตั้ง การรันระบบปฏิบัติการ และการเก็บข้อมูล โดย Raspberry pi สามารถลง OS ในตระกูล Linux ได้หลากหลายไม่ว่าจะเป็น Raspbian, Ubuntu, SARPi, หรือ ARCH Linux ซึ่งระบบปฏิบัติการที่นิยมมากที่สุด คือ Raspbian

Raspbian เป็นระบบปฏิบัติการสำหรับติดตั้งใช้งานบน Raspberry Pi พัฒนาจากระบบ Debian Linux เป็น OS ที่ได้รับความนิยมเนื่องจากใช้งานง่าย มี x-windows ติดต่อกับผู้ใช้ได้ง่าย, มี application และ เครื่องมือบางอย่างติดตั้งมาให้แล้ว

การติดตั้ง Raspbian บน Raspberry pi มี 2 วิธี คือ การติดตั้งแบบใช้ NOOBS (ต่อจอ) และการติดตั้งแบบเขียนไฟล์ images ลง micro SD Card (ไม่ต่อจอ)

>การติดตั้งแบบใช้ NOOBS คือ New Out Of Box Software เป็นวิธีที่ง่ายที่สุดเหมาะสำหรับมือใหม่ ที่ไม่มีความรู้เรื่องการติดตั้งระบบปฏิบัติการ เนื่องจากมีความสะดวก รวดเร็ว สามารถถือปวารได้เลย แต่มีระยะเวลาการติดตั้งนานกว่าการติดตั้งแบบเขียนไฟล์ image

>การติดตั้งแบบเขียนไฟล์ images ลง micro SD Card เป็นวิธีที่มีขั้นตอนมากกว่าการติดตั้งแบบใช้ NOOBS อีกทั้งต้องใช้โปรแกรมหา IP Address ของ Raspberry pi เพื่อติดตั้งผ่าน SSH อีกด้วย แต่วิธีนี้มีระยะเวลาในการติดตั้ง (การ write ไฟล์ลงใน micro SD Card) สั้นกว่า เหมาะกับผู้ที่มีความรู้เกี่ยวกับการ write ไฟล์ image ดังนั้นเอกสารฉบับนี้จึงขออธิบายการติดตั้งแบบใช้ NOOBS เท่านั้น

## อุปกรณ์ที่ใช้

1. Raspberry pi 3 model B+
2. Micro SD Card 32 GB Class 10
3. USB micro power supply (Adapter จ่ายไฟเลี้ยง 5 V ขนาดไม่น้อยกว่า 1.5 A)
4. USB Mouse
5. USB Keyboard
6. จอ Monitor
7. HDMI cable สำหรับต่อ Raspberry Pi กับ Monitor

## ขั้นตอนการติดตั้ง

ขั้นตอนการติดตั้งโดยรวม คือ ดาวน์โหลดไฟล์ NOOBS > แยกไฟล์แล้วก๊อปปี้ไปไว้ใน micro SD Card > บูท Raspberry Pi ด้วย NOOBS > เลือกระบบปฏิบัติการที่ต้องการติดตั้ง รายละเอียด ดังนี้

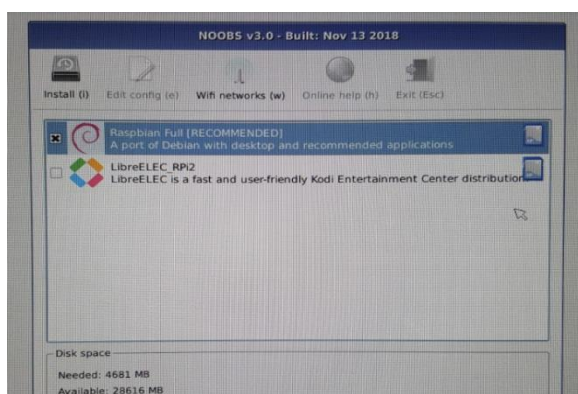
1. ดาวน์โหลด NOOBS ได้จากเว็บ <https://www.raspberrypi.org/downloads/noobs/> ในหน้านี้จะมีให้เลือกระหว่าง NOOBS กับ NOOBS Lite ซึ่ง NOOBS จะมี Raspbian มาให้ไม่ต้องดาวน์โหลดผ่านอินเทอร์เน็ตทีหลัง แต่ NOOBS Lite ทุก OS ต้องโหลดผ่านอินเทอร์เน็ตตอนติดตั้งใน Raspberry pi ดังนั้นให้เลือก NOOBS แบบไฟล์ .zip



ภาพที่ ข-1 ดาวน์โหลด NOOBS แบบไฟล์ .zip

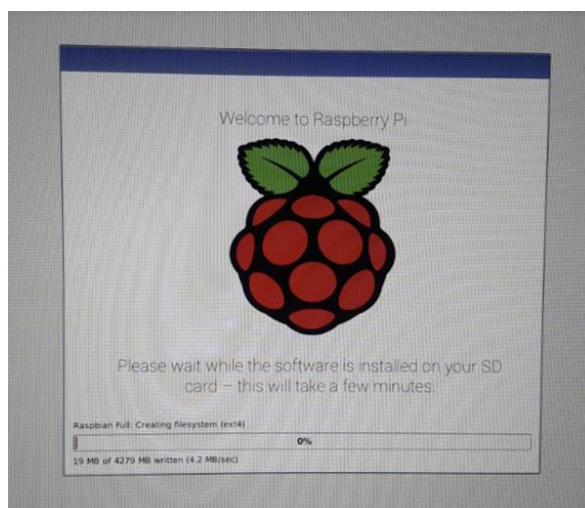
2. เมื่อดาวน์โหลดเสร็จแล้วจะได้ไฟล์ .zip ทำการแตกไฟล์ .zip ออกจะได้ไฟล์ต่าง ๆ หลังจากนั้นทำการ copy ไฟล์ทั้งหมดไปวางไว้ใน micro SD Card ที่ฟอร์แมตไว้

3. เสียบ Micro SD Card เข้า Raspberry Pi และต่อพอร์ตต่างๆ คือ Mouse, Keyboard, HDMI ต่อกับจอ monitor จากนั้นจึงต่อกับ Power Supply หน้าจอจะปรากฏหน้าต่างขึ้นมา ดังภาพที่ ข-2



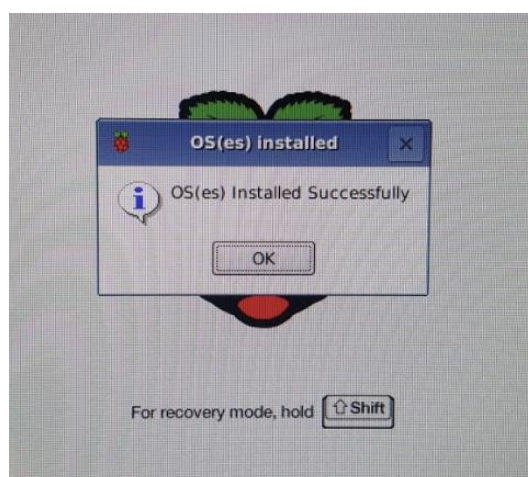
ภาพที่ ข-2 หน้าต่างแสดงระบบปฏิบัติการที่มีให้เลือกลง

4. เลือก Raspbian Full >เลือก install จากนั้นเครื่องจะทำการติดตั้งระบบปฏิบัติการลงใน micro SD Card ซึ่งใช้ระยะเวลาหนึ่ง



ภาพที่ ข-3 รูปแสดงขณะทำการติดตั้ง OS

5. เมื่อการติดตั้งเสร็จสิ้น สามารถบูท Raspberry pi ใช้งานได้เลย



ภาพที่ ข-4 รูปแสดงการติดตั้งเสร็จสิ้น

### ประวัติผู้แต่ง

ปรินญานิพนธ์เรื่อง : กล้องสังเกตการณ์อัจฉริยะขนาดเล็ก  
 สาขาวิชา : วิศวกรรมไฟฟ้า  
 ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
 คณะ : วิศวกรรมศาสตร์  
 ชื่อ : นางสาว ดวงทอง ดำรงเจริญ

#### ประวัติ

เกิดเมื่อวันที่ 19 มิถุนายน พ.ศ. 2538 อยู่บ้านเลขที่ 64 หมู่ 4 ตำบลเสม็ด อำเภอเมืองชลบุรี จังหวัดชลบุรี 20000 สำเร็จการศึกษามัธยมศึกษาตอนปลาย สาขาวิทยาศาสตร์ - คณิตศาสตร์ จากโรงเรียนชลกันยานุกูล จังหวัดชลบุรี ปีการศึกษา 2556 และสำเร็จการศึกษาในระดับปริญญาตรี สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปีการศึกษา 2561

ชื่อ : นายวิศรุต กุศลส่ง

#### ประวัติ

เกิดเมื่อวันที่ 28 กันยายน พ.ศ. 2539 อยู่บ้านเลขที่ 56/1 ซอยทรายทอง 9 ตำบลท่าทราย อำเภอเมืองนนทบุรี จังหวัดนนทบุรี 11000 สำเร็จการศึกษามัธยมศึกษาตอนปลาย สาขาวิทยาศาสตร์ - คณิตศาสตร์ จากโรงเรียนชลประทานวิทยา จังหวัดนนทบุรี ปีการศึกษา 2557 และสำเร็จการศึกษาในระดับปริญญาตรี สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปีการศึกษา 2561