

ระบบติดตามการจราจรรถยนต์ โดยใช้การประมวลผลภาพ
AUTOMATIC TRACKING SYSTEM BASED ON IMAGE PROCESSING



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบติดตามการจราจรรถยนต์ โดยใช้การประมวลผลภาพ
AUTOMATIC TRACKING SYSTEM BASED ON IMAGE PROCESSING



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC TRACKING SYSTEM BASED ON IMAGE PROCESSING



Watcharapan Mongkalangoon

Sahatsawat Kthasattapich

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LARDKRABANG
ACADEMIC YEAR 2021


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2564
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ ระบบติดตามการจราจรรถยนต์ โดยใช้การประมวลผลภาพ
AUTOMATIC TRACKING SYSTEM BASED ON IMAGE PROCESSING

นักศึกษาผู้จัดทำ นาย วัชรพันธุ์ มงคลกลางกูร รหัสศึกษา 61010956
นาย สหัสวรรษ กฤษสัตาพิชญ์ รหัสศึกษา 61011094

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2564

| | |
|-------------------------------|--|
| อาจารย์ผู้ควบคุมปริญญาานิพนธ์ | ลายมือชื่อ |
| ผศ.ดร.นรินทร์ ธรรมารักษ์วัฒน์ |  |

| | | | | |
|---------------------|--|-----------|----------|--|
| หัวข้อปริญญาานิพนธ์ | ระบบติดตามการจราจรรถยนต์ โดยใช้การประมวลผลภาพ AUTOMATIC TRACKING SYSTEM BASED ON IMAGE PROCESSING | | | |
| นักศึกษาผู้จัดทำ | นาย วัชรพันธุ์ มงคลกลางกูร | รหัสศึกษา | 61010956 | |
| | นาย สหัสวรรษ กลาสัตตาพิชญ์ | รหัสศึกษา | 61011094 | |
| อาจารย์ที่ปรึกษา | ผศ.ดร.นรินทร์ ธรรมารักษ์วิณะ | | | |
| ปีการศึกษา | 2564 | | | |

บทคัดย่อ

ในปัจจุบันมีมีจราจรที่ทำการจราจรรถยนต์มากขึ้น โดยเฉพาะในปี พ.ศ. 2564 ที่ผ่านมา มีการแจ้งรถยนต์สูญหายมากกว่า 2000 คัน และส่วนใหญ่ไม่สามารถติดตามคืนมาได้ ประกอบกับเทคโนโลยีปัจจุบันมีการพัฒนาไปก้าวหน้าไปเป็นอันมาก เพื่อให้มีความปลอดภัยในด้านทรัพย์สินมากยิ่งขึ้น จึงเกิดแนวคิดในการตรวจจับและตรวจสอบป้ายทะเบียนรถยนต์ เพื่อช่วยเพิ่มประสิทธิภาพในการตามคืนรถยนต์ที่ถูกโจรกรรม นอกจากนี้ยังมีการประยุกต์เข้ากับระบบฐานข้อมูล และ MQTT เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งจะใช้ภาษา Python ในการพัฒนาโปรแกรมโดย Python คือ ภาษาที่ใช้ในการเขียนโปรแกรม ซึ่งถูกพัฒนาขึ้นโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถใช้ภาษา Python ได้กับหลายระบบปฏิบัติการ ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมได้ จึงทำให้ Python มีความสามารถสูงและใช้งานได้ครอบคลุมกับทุกลักษณะการใช้งาน จึงเกิดแนวคิดในการนำ Python มาเขียนโปรแกรมตรวจจับและตรวจสอบทะเบียนรถยนต์เพื่อนำไปแสดงผลยังจอภาพสำหรับการประมวลผลภาพ โดยมีการใช้การร่วมกับไลบรารี Google cloud vision ซึ่งเป็นไลบรารีที่รวบรวมฟังก์ชันต่างๆ เพื่อนำไปใช้ตามจราจรยนต์ที่ถูกโจรกรรมไป ซึ่งผลที่ได้สามารถตรวจจับ แปรนตร์รถยนต์และป้ายทะเบียนรถยนต์ที่ถูกโจรกรรม ได้ตามเป้าหมายที่ตั้งไว้

| | | |
|-----------------------|---|-----------------|
| Thesis Title | AUTOMATIC TRACKING SYSTEM BASED ON IMAGE PROCESSING | |
| Authors | Mr. Watcharapan | Mongkalangoon |
| | Mr. Sahatsawat | Kthasattapich |
| Thesis Advisor | Asst.Dr. Narin | Tammarugwattana |
| Year | 2021 | |

ABSTRACT

Nowadays, there are more scammers doing car thefts. Especially in 2021, there were over 2000 cars had been stolen and most of them never return to the owner. Therefore, we have developed technology in various fields to be more secure in assets. Thus came the idea of detecting and inspecting car license plates. To help increase the effectiveness of tracking stolen cars. It is also applied to database systems and MQTT to work efficiently, which will use Python language to develop programs. Python is a programming language that was developed independently of the platform. In other words, Python can be used with many operating systems. Making it possible for anyone to use Python to develop programs, Python is highly capable and can be used to cover all applications. Therefore, the idea was to use Python to write a program to detect and verify vehicle registrations to display on the screen for image processing. It is used in conjunction with the Google library. cloud vision, a library that collects various functions to be used to catch stolen cars

กิตติกรรมประกาศ

การดำเนินโครงการ “ระบบติดตามการจรรกรมรณต์ โดยใช้การประมวลผลภาพ” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากขาดการสนับสนุน และ ความช่วยเหลือจากหลายๆฝ่าย บุคคลที่เป็นส่วนสำคัญในการผลักดันให้เกิดโครงการนี้ และ ทำให้สำเร็จลุล่วงได้ ก็คือ ผศ.ดร.นรินทร์ ธรรมารักษ์วัฒน์ อาจารย์ที่ปรึกษาปริญญาโทที่ให้ความเอาใจใส่ แนะนำ และ ช่วยเหลือเสมอมา ต้องขอขอบพระคุณเป็นอย่างสูง และ ต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่งซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดีพร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และ ยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกพระคุณอันสูงสุด และ ขอกราบขอบพระคุณมา ณ ที่นี้ด้วย คุณค่า และ ประโยชน์อันพึงได้รับจากปริญญาโทฉบับนี้ผู้วิจัยขอบแต่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

สารบัญ

| | หน้า |
|---|----------|
| บทคัดย่อ | I |
| บทคัดย่อภาษาอังกฤษ..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ | IV |
| สารบัญตาราง | VII |
| สารบัญรูป..... | VIII |
| | |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ความสำคัญของปริญญาโท | 1 |
| 1.2 วัตถุประสงค์ของปริญญาโท..... | 1 |
| 1.3 ขอบเขตของปริญญาโท..... | 1 |
| 1.4 ขั้นตอนการศึกษา | 2 |
| | |
| บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง..... | 3 |
| 2.1 PYTHON | 3 |
| 2.1.1 หลักปรัชญาของภาษา Python | 3 |
| 2.1.2 คุณสมบัติเด่นของภาษา Python | 3 |
| 2.1.3 ตัวแปรและ คำสั่งงานในภาษา Python..... | 4 |
| 2.1.4 คำสั่งงานในภาษา Python | 5 |
| 2.2 OPENCV (OPEN SOURCE COMPUTER VISION) | 5 |
| 2.2.1 ระบบสี RGB | 6 |
| 2.2.2 ภาพระดับสีเทา (Gray Scale Image)..... | 6 |
| 2.3 GOOGLE CLOUD VISION API..... | 7 |
| 2.4 การเรียนรู้เชิงลึก (DEEP LEARNING)..... | 8 |
| 2.5 CONVOLUTIONAL NEURAL NETWORK (CNN)..... | 8 |
| 2.5.1 ชั้นคอนโวลูชัน (Convolution layer) | 9 |
| 2.5.2 ชั้นพูลลิ่ง (Pooling Layer)..... | 9 |
| 2.5.3 ชั้นเชื่อมโยงสมบูรณ์ (Fully-Connected Layer)..... | 10 |
| 2.6 PANDAS..... | 10 |
| 2.6.1 คุณสมบัติของ Pandas | 10 |

| | |
|--|-----------|
| 2.7 SQL..... | 11 |
| 2.8 MQTT | 11 |
| 2.8.1 หลักการทำงานของ MQTT | 12 |
| 2.8.2 การทำงานของ MQTT Client..... | 12 |
| 2.9 RASPBERRY PI..... | 13 |
| 2.9.1 คุณสมบัติของ Raspberry Pi (Raspberry Pi3 Model B)..... | 14 |
| 2.9.2 ระบบปฏิบัติการของบอร์ด Raspberry Pi | 17 |
| 2.9.3 อุปกรณ์ที่ต้องใช้กับบอร์ด Raspberry Pi..... | 17 |
| 2.9.4 การติดตั้งระบบปฏิบัติการ..... | 17 |
| 2.9.5 การใช้งาน Raspberry Pi แบบ Remote Terminal (SSH)..... | 20 |
| บทที่ 3 วิธีการดำเนินงาน | 29 |
| 3.1 อุปกรณ์ที่ใช้ในระบบ | 29 |
| 3.1.1 Computer..... | 29 |
| 3.1.2 Raspberry Pi 3 | 29 |
| 3.1.3 กล้อง Webcam 1080p รุ่น A875-A..... | 30 |
| 3.1.4 Power Bank..... | 30 |
| 3.2 การออกแบบระบบ | 31 |
| 3.3 กระบวนการติดตามการจราจรรถยนต์ โดยการใช้การประมวลผลภาพ..... | 32 |
| 3.4 การติดตั้งโปรแกรม PYTHON และไลบรารีที่จำเป็น เพื่อให้ใช้งานร่วมกัน | 33 |
| 3.4.1 การติดตั้งโปรแกรม Python..... | 33 |
| 3.4.2 pip..... | 34 |
| 3.4.3 ทำการติดตั้งไลบรารีโดยใช้ pip..... | 34 |
| 3.5 GOOGLE CLOUD VISION API..... | 36 |
| 3.6 DB BROWSER FOR SQLITE..... | 41 |
| 3.6.1 สร้างฐานข้อมูลใน SQLite | 41 |
| 3.7 MQTT (MESSAGE QUEUE TELEMETRY TRANSPORT)..... | 43 |
| 3.7.1 เขียนคำสั่งในการส่งข้อมูลขึ้น | 44 |
| 3.8 การเรียกใช้ฐานข้อมูลโดยใช้ไลบรารี SQLITE3 และจัดเรียงข้อมูลโดยใช้ PANDAS | 47 |
| 3.9 การตรวจจับป้ายทะเบียน | 48 |
| 3.10 ส่งข้อความและรูปภาพเข้าไลน์ NOTIFY..... | 50 |
| 3.11 การใช้งาน RASPBERRY PI แบบ REMOTE TERMINAL (SSH)..... | 54 |
| 3.12 ติดตั้ง SQLITE3 ลงบน RASPBERRY PI..... | 58 |

| | |
|---|-----------|
| บทที่ 4 ผลการทดลอง | 65 |
| 4.1 ผลทดลองประมวลผลภาพป้ายทะเบียน | 65 |
| 4.1.1 ผลการทดลองตรวจจับป้ายทะเบียนโดยกล้องมีระยะห่าง 100 เซนติเมตร..... | 65 |
| 4.1.1.1 เวลากลางวัน..... | 65 |
| 4.1.1.2 เวลากลางคืน | 67 |
| 4.1.2 ผลการทดลองตรวจจับป้ายทะเบียนโดยกล้องมีระยะห่าง 200 cm | 68 |
| 4.1.2.1 เวลากลางวัน..... | 68 |
| 4.1.3 ผลการทดลองตรวจจับป้ายทะเบียนโดยกล้องมีระยะห่าง 300 เซนติเมตร..... | 70 |
| 4.1.3.1 เวลากลางวัน..... | 70 |
| 4.1.4 ผลการทดลองตรวจจับป้ายทะเบียนโดยกล้องมีระยะห่าง 400 เซนติเมตร..... | 72 |
| 4.1.4.1 เวลากลางวัน..... | 72 |
| 4.2 ทดสอบใช้งานจริง..... | 73 |
| 4.2.1 ประสิทธิภาพการประมวลผล การทดสอบตรวจจับรถยนต์จำนวน 50 คัน | 75 |
| 4.2.1.1 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ถูกต้อง | 75 |
| 4.2.1.2 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ไม่ถูกต้อง | 81 |
| 4.2.1.3 ตัวอย่างรถยนต์ที่ไม่สามารถตรวจจับแบรนด์รถยนต์ได้ | 83 |
| 4.2.1.4 ตัวอย่างรถยนต์ที่ไม่สามารถตรวจจับจังหวัดรถยนต์ได้..... | 86 |
| บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ | 90 |
| 5.1 สรุปผลการทดลอง..... | 90 |
| 5.2 ปัญหา และ อุปสรรค | 91 |
| 5.3 ข้อเสนอแนะ..... | 91 |
| บรรณานุกรม..... | 92 |
| ภาคผนวก | 93 |

สารบัญตาราง

| ตารางที่ | หน้า |
|--|------|
| 4.1 ตรวจสอบแบรนด์รถยนต์ได้ถูกต้อง..... | 76 |
| 4.2 ไม่สามารถตรวจสอบแบรนด์รถยนต์ได้..... | 83 |
| 4.3 ไม่สามารถตรวจสอบจังหวัดได้..... | 84 |
| 4.4 ตรวจสอบตัวอักษรภาษาไทยได้ไม่ถูกต้อง..... | 86 |



สารบัญรูป

| รูปที่ | หน้า |
|---|------|
| 2.1 โปรแกรม PYTHON 3.10.0 SHELL | 5 |
| 2.2 วงล้อสี RGB | 6 |
| 2.3 ภาพแสดงระดับสีเทา | 7 |
| 2.4 หลักการทำงาน GOOGLE CLOUD VISION API | 7 |
| 2.5 ภาพรวมการเรียนรู้เชิงลึก | 8 |
| 2.6 โครงสร้างโดยทั่วไปของ CNN | 9 |
| 2.7 RECEPTIVE FIELD | 9 |
| 2.8 ภาพประกอบการรวมค่าสูงสุดด้วยตัวกรอง 2x2 และการเลื่อนตัวกรอง | 10 |
| 2.9 MQTT PROTOCOL | 13 |
| 2.10 บอร์ดและรุ่น RASPBERRY PI | 14 |
| 2.11 พอร์ตของบอร์ด RASPBERRY PI 3 MODEL B | 15 |
| 2.12 จุดเชื่อมต่อ GPIO ของบอร์ด RASPBERRY PI | 16 |
| 2.13 RASPBERRY PI IMAGER | 17 |
| 2.14 เลือกไดรฟ์ที่ต้องการติดตั้ง RASPBERRY PI OS | 18 |
| 2.15 ระบบปฏิบัติการ RASPBERRY PI OS (32BIT) | 18 |
| 2.16 การเชื่อมต่ออุปกรณ์ต่างๆกับบอร์ด RASPBERRY PI | 19 |
| 2.17 หน้าจอแสดงผล GUI | 19 |
| 2.18 ใช้คำสั่ง SUDO APT-GET UPDATE บน TERMINAL | 20 |
| 2.19 เมื่อทำการอัปเดตเสร็จเรียบร้อย | 20 |
| 2.20 ใช้งานคำสั่ง SUDO RASPI-CONFIG | 21 |
| 2.21 การตั้งค่า INTERFACE ของ RASPBERRY PI | 22 |
| 2.22 เลือก SSH ในตั้งค่า | 22 |
| 2.23 เลือกเปิดใช้งาน SSH | 23 |
| 2.24 ทำการเปิดใช้งาน SSH เรียบร้อย | 23 |
| 2.25 ใช้คำสั่ง SUDO APT-GET INSTALL XRDP | 24 |
| 2.26 เมื่อทำการติดตั้ง XRDP เสร็จเรียบร้อย | 25 |
| 2.27 ใช้คำสั่ง IFCONFIG | 25 |
| 2.28 ไอพีแอดเดรสของ RASPBERRY PI | 26 |
| 2.29 เรียกใช้ RUN COMMAND และพิมพ์ MSTSC | 26 |

สารบัญรูป (ต่อ)

| รูปที่ | หน้า |
|---|------|
| 2.30 โปรแกรม REMOTE DESKTOP CONNECTION..... | 27 |
| 2.31 หน้า LOGIN ของ REMOTE DESKTOP CONNECTION..... | 27 |
| 2.32 หน้า GUI ของ RASPBERRY PI OS..... | 28 |
| 3.1 อุปกรณ์ทั้งหมดที่ใช้ในระบบ..... | 29 |
| 3.2 คอมพิวเตอร์..... | 29 |
| 3.3 RASPBERRY PI 3..... | 30 |
| 3.4 กล้อง WEBCAM 1080P รุ่น A875-A..... | 30 |
| 3.5 POWER BANK..... | 30 |
| 3.6 รูปแบบการทำงานของระบบ..... | 31 |
| 3.7 กระบวนการติดตามการจราจรรถยนต์..... | 32 |
| 3.8 HTTPS://WWW.PYTHON.ORG/DOWNLOADS | 33 |
| 3.9 PYTHON 3.10.0 SHELL..... | 34 |
| 3.10 ไฟล์ REQUIREMENTS.TXT..... | 35 |
| 3.11 พิมพ์คำสั่งใน WINDOWS POWERSHELL..... | 35 |
| 3.12 คำสั่งเรียกใช้ไลบรารีที่ทำการติดตั้ง..... | 36 |
| 3.13 การตรวจสอบการติดตั้งไลบรารีทั้งหมด..... | 36 |
| 3.14 เปิดใช้บริการ GOOGLE CLOUD SERVICE..... | 37 |
| 3.15 เลือก CREATE PROJECT..... | 37 |
| 3.16 ทำการตั้งชื่อโปรเจค..... | 38 |
| 3.17 หน้าต่างสร้างโปรเจคเสร็จเรียบร้อยแล้ว..... | 38 |
| 3.18 เปิดใช้บริการ GOOGLE CLOUD VISION API..... | 39 |
| 3.19 เลือก GOOGLE SERVICE ACCOUNT..... | 39 |
| 3.20 ทำการขอ SERVICE ACCOUNT KEY..... | 40 |
| 3.21 เลือกประเภทเป็น JSON..... | 40 |
| 3.22 ไฟล์ SERVICE ACCOUNT KEY..... | 41 |
| 3.23 สร้างไฟล์ฐานข้อมูล (DATABASE)..... | 41 |
| 3.24 TABLE DEFINITION..... | 42 |
| 3.25 BROWSER DATA สำหรับเก็บข้อมูล..... | 43 |
| 3.26 เว็บไซต์ HIVE MQ..... | 43 |

สารบัญรูป (ต่อ)

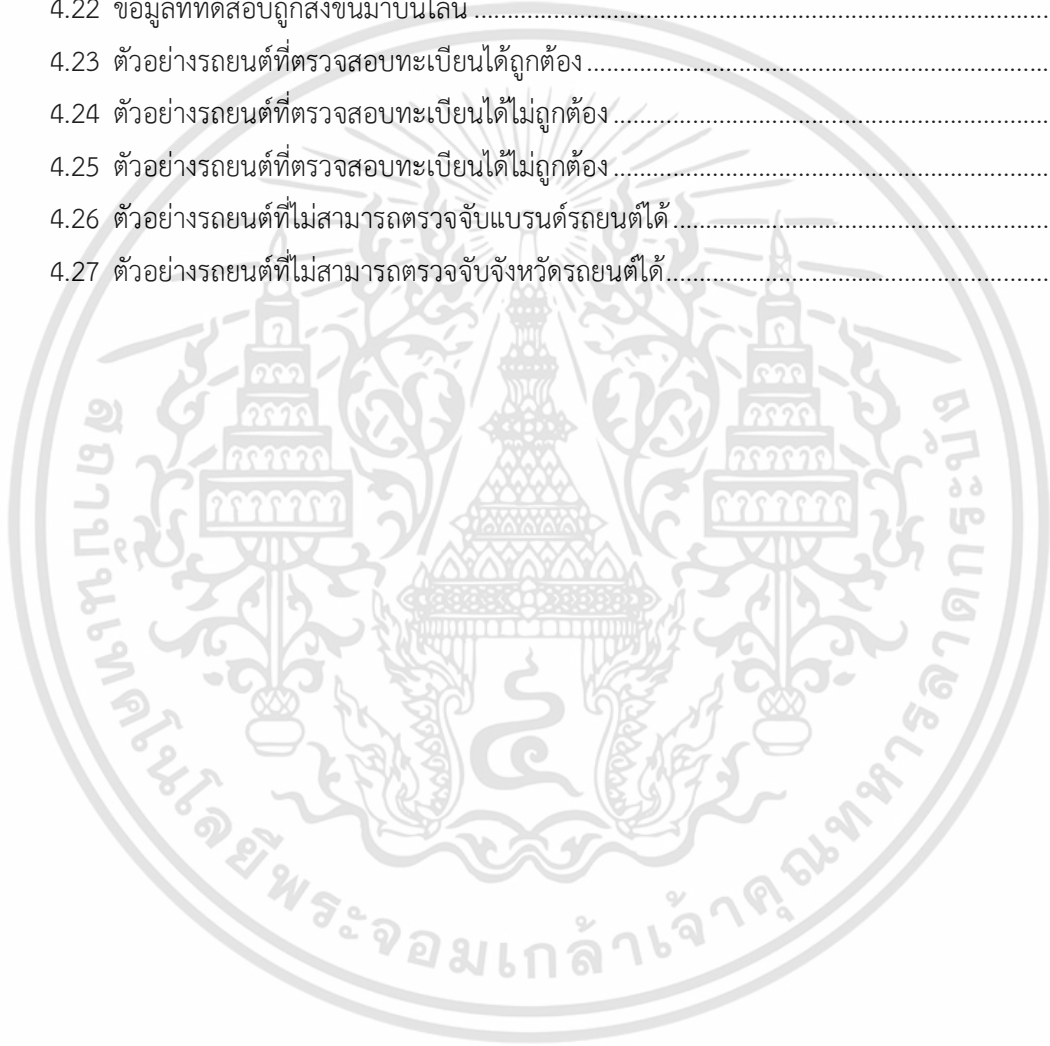
| รูปที่ | หน้า |
|---|------|
| 3.27 ไลบรารีในการส่งข้อมูลจาก MQTT ขึ้น SQLITE | 44 |
| 3.28 คำสั่งในการส่งข้อมูลจาก MQTT ขึ้น SQLITE | 44 |
| 3.29 MQTT WEBSOCKET CLIENT | 45 |
| 3.30 ทดลองส่งค่าบน WEB MQTT (1) | 45 |
| 3.31 เก็บค่าไว้บน DB BROWSER (1) | 46 |
| 3.32 ทดลองส่งค่าบน WEB MQTT (2) | 46 |
| 3.33 เก็บค่าไว้บน DB BROWSER (2) | 47 |
| 3.34 โค้ดการจัดการเรียงข้อมูลโดยใช้ PANDAS | 47 |
| 3.35 ผลลัพธ์การจัดการเรียงข้อมูล | 48 |
| 3.36 โค้ดการดึงข้อมูลจาก MAIN.PY | 48 |
| 3.37 โค้ดการสร้างเงื่อนไขตรวจสอบยี่ห้อและตัวอักษร | 49 |
| 3.38 โค้ดการเปรียบเทียบค่าในฐานข้อมูล (DATABASE) กับ MYDATA | 49 |
| 3.39 ผลลัพธ์เมื่อข้อมูลตรงกัน (1) | 50 |
| 3.40 ผลลัพธ์เมื่อข้อมูลตรงกัน (2) | 50 |
| 3.41 สร้างฟังก์ชัน DEF LINENOTIFY | 51 |
| 3.42 เว็บไซต์ LINE NOTIFY | 51 |
| 3.43 รายชื่อบริการที่เชื่อมต่ออยู่ | 51 |
| 3.44 ผลลัพธ์จากการทำงานของโปรแกรม (1) | 52 |
| 3.45 ผลลัพธ์จากการทำงานของโปรแกรม (2) | 52 |
| 3.46 ข้อมูลที่ตรงพบตรงกันแล้วถูกส่งขึ้นมาบนไลน์ (1) | 53 |
| 3.47 ข้อมูลที่ตรงพบตรงกันแล้วถูกส่งขึ้นมาบนไลน์ (2) | 53 |
| 3.48 ใช้คำสั่ง SUDO APT-GET INSTALL XRDP | 54 |
| 3.49 เมื่อทำการติดตั้ง XRDP เสร็จเรียบร้อย | 55 |
| 3.50 ใช้คำสั่ง IFCONFIG | 55 |
| 3.51 ไอพีแอดเดรสของ RASPBERRY PI | 56 |
| 3.52 เรียกใช้ RUN COMMAND และพิมพ์ MSTSC | 56 |
| 3.53 โปรแกรม REMOTE DESKTOP CONNECTION | 57 |
| 3.54 หน้า LOGIN ของ REMOTE DESKTOP CONNECTION | 57 |
| 3.55 หน้า GUI ของ RASPBERRY PI OS | 58 |

สารบัญรูป (ต่อ)

| รูปที่ | หน้า |
|---|------|
| 3.56 ติดตั้ง SQLITE3 ใน RASPBERRY PI | 58 |
| 3.57 INSTALL SQLITEBROWSER ใน RASPBERRY PI | 59 |
| 3.58 ติดตั้ง DB BROWSER FOR SQLITE..... | 59 |
| 3.59 ติดตั้ง DB BROWSER FOR SQLITE..... | 60 |
| 3.60 CREATE TABLE ใน RASPBERRY PI | 60 |
| 3.61 ไลบรารีในการส่งข้อมูลจาก MQTT ขึ้น SQLITE | 61 |
| 3.62 ส่งค่าบน WEB MQTT ขึ้นไปบน RASPBERRY PI..... | 62 |
| 3.63 BROWSE DATA ใน RASPBERRY PI | 62 |
| 3.64 จัดเรียงข้อมูลโดยใช้ PANDAS ใน RASPBERRY PI | 63 |
| 3.65 เปรียบเทียบค่าในฐานข้อมูล (DATABASE) กับ MYDATA ใน RASPBERRY PI..... | 63 |
| 3.66 ส่งข้อมูลขึ้นไปบนไลน์โดยใช้ RASPBERRY PI | 64 |
| 4.1 มุม 0 องศา ระยะ 100 เซนติเมตร เวลากลางวัน | 65 |
| 4.2 มุม 10 องศา ระยะ 100 เซนติเมตร เวลากลางวัน | 66 |
| 4.3 มุม 20 องศา ระยะ 100 เซนติเมตร เวลากลางวัน | 66 |
| 4.4 มุม 30 องศา ระยะ 100 เซนติเมตร เวลากลางวัน | 66 |
| 4.5 มุม 40 องศา ระยะ 100 เซนติเมตร เวลากลางวัน | 67 |
| 4.6 มุม 0 องศา ระยะ 100 เซนติเมตร เวลากลางคืน | 67 |
| 4.7 มุม 0 องศา ระยะ 100 เซนติเมตร เวลากลางคืน | 68 |
| 4.8 มุม 0 องศา ระยะ 200 เซนติเมตร เวลากลางวัน | 68 |
| 4.9 มุม 10 องศา ระยะ 200 เซนติเมตร เวลากลางวัน | 69 |
| 4.10 มุม 20 องศา ระยะ 200 เซนติเมตร เวลากลางวัน | 69 |
| 4.11 มุม 30 องศา ระยะ 200 เซนติเมตร เวลากลางวัน | 69 |
| 4.12 มุม 40 องศา ระยะ 200 เซนติเมตร เวลากลางวัน | 70 |
| 4.13 มุม 0 องศา ระยะ 300 เซนติเมตร เวลากลางวัน | 70 |
| 4.14 มุม 10 องศา ระยะ 300 เซนติเมตร เวลากลางวัน | 71 |
| 4.15 มุม 20 องศา ระยะ 300 เซนติเมตร เวลากลางวัน | 71 |
| 4.16 มุม 30 องศา ระยะ 300 เซนติเมตร เวลากลางวัน | 71 |
| 4.17 มุม 40 องศา ระยะ 300 CM เวลากลางวันไม่สามารถตรวจจับได้..... | 72 |
| 4.18 มุม 0 องศา ระยะ 400 เซนติเมตร เวลากลางวันไม่สามารถตรวจจับได้..... | 72 |

สารบัญรูป (ต่อ)

| รูปที่ | หน้า |
|---|------|
| 4.19 ทดสอบใช้งานจริงในการตรวจจับรถยนต์..... | 73 |
| 4.20 ทดสอบใช้งานจริงในการตรวจจับรถยนต์..... | 73 |
| 4.21 ทดสอบใช้งานจริงข้อมูลส่งขึ้นมาในไลน์..... | 74 |
| 4.22 ข้อมูลที่ทดสอบถูกส่งขึ้นมาบนไลน์..... | 74 |
| 4.23 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ถูกต้อง..... | 75 |
| 4.24 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ไม่ถูกต้อง..... | 82 |
| 4.25 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ไม่ถูกต้อง..... | 82 |
| 4.26 ตัวอย่างรถยนต์ที่ไม่สามารถตรวจจับแบรนด์รถยนต์ได้..... | 83 |
| 4.27 ตัวอย่างรถยนต์ที่ไม่สามารถตรวจจับจังหวัดรถยนต์ได้..... | 86 |



บทที่ 1

บทนำ

1.1 ความสำคัญของปัญญาประดิษฐ์

ในปัจจุบันปัญญาประดิษฐ์ มีบทบาทในชีวิตประจำวันเป็นอย่างมาก ทำให้การใช้ชีวิตสะดวกสบาย ประกอบกับ การโจรกรรมรถยนต์ที่มากขึ้นเรื่อยๆ โดยในปี 2564 มีรถยนต์ที่ถูกโจรกรรมมากกว่า 2000 คัน เราจึงมีแนวคิดในพัฒนาเทคโนโลยีในด้านต่างๆ เพื่อให้ มีความปลอดภัยในด้านทรัพย์สินมากยิ่งขึ้น จึงเกิดแนวคิดในการตรวจจับและตรวจสอบป้ายทะเบียนรถยนต์ เพื่อช่วยเพิ่มประสิทธิภาพในการตามคืนรถยนต์ที่ถูกโจรกรรม โดยจะประยุกต์เข้ากับระบบฐานข้อมูล และ MQTT เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพและรวดเร็ว ซึ่งจะใช้ภาษา Python ในการพัฒนาโปรแกรม มาเขียนโปรแกรมตรวจจับและตรวจสอบทะเบียนรถยนต์เพื่อนำไปแสดงผลยังจอภาพสำหรับการประมวลผลภาพ โดยมีการใช้การร่วมกับไลบรารี Google cloud vision ซึ่งเป็นไลบรารีที่รวบรวมฟังก์ชันต่างๆ เพื่อนำไปใช้ตามจ็อบรถยนต์ที่ถูกโจรกรรมไป หลังจากนั้นข้อมูลที่ทำการตรวจจับได้อย่างถูกต้องเราจะส่งข้อมูลเหล่านั้นไปในแอปพลิเคชันไลน์ เพราะ เป็นแอปพลิเคชันที่คนไทยใช้กันอย่างแพร่หลายและสะดวกสบาย ทำให้สามารถช่วยเหลือเจ้าหน้าที่หรือผู้รับผิดชอบที่จะติดตามรถยนต์กลับคืนให้แก่เจ้าของ

1.2 วัตถุประสงค์ของปัญญาประดิษฐ์

1. ศึกษาการพัฒนาโปรแกรมด้วยภาษา Python
2. เพื่อศึกษาขั้นตอนวิธีการประมวลผลภาพและการทำงานร่วมกับ Google cloud vision API
3. ศึกษาศึกษาการทำงานไลบรารี Google cloud vision
4. ศึกษาการเก็บข้อมูลในฐานข้อมูล
5. ศึกษาการส่งข้อมูลผ่าน MQTT และส่งข้อความแจ้งเตือนผ่าน Line

1.3 ขอบเขตของปัญญาประดิษฐ์

1. เขียนโปรแกรมด้วยภาษา Python ให้สามารถตรวจจับป้ายทะเบียนและยี่ห้อรถยนต์
2. เขียนโปรแกรมด้วยภาษา Python ร่วมกับ SQLite database
3. เขียนโปรแกรมด้วยภาษา Python ให้สามารถรับ-ส่งข้อมูล MQTT ร่วมกับ SQLite database
4. เขียนโปรแกรมด้วยภาษา Python ให้สามารถแจ้งเตือนผ่านระบบไลน์บนมือถือได้

1.4 ขั้นตอนการศึกษา

1. ศึกษาการเขียนโปรแกรมด้วยภาษา Python
2. ศึกษาการใช้ไลบรารี Google cloud vision ร่วมกับภาษา Python
3. ศึกษาหลักการตรวจจับ และ จัดจำป้ายทะเบียน
4. ศึกษาการรับส่งข้อมูล MQTT และเก็บข้อมูลใน SQL และส่งข้อมูลผ่านระบบไลน์



บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Python

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง เป็นภาษาโปรแกรมระดับสูงที่ใช้กันอย่างแพร่หลาย ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux, Windows N, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัว นี้เป็น Open Source เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรีโดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครอบคลุมกับทุกลักษณะงาน ภาษา Python ถูกสร้างขึ้นมาจากภาษาซี การประมวลผลจะทำในแบบอินเทอร์พรีเตอร์

2.1.1 หลักปรัชญาของภาษา Python

Python เป็นภาษาที่สามารถสร้างงานได้หลากหลายกระบวนทัศน์ (multi-paradigm language) โดยจะมองอะไรที่มากกว่าการ coding เพื่อนำมาใช้งานตามภาพแบบเดิม ๆ แต่จะเป็นการนำเอาหลักการของกระบวนทัศน์ (Paradigm) แบบ Object-oriented programming, Structured programming, Functional programming และ Aspect-oriented programming นำเอามาใช้ทั้งแบบเดียว ๆ และนำมาใช้ร่วมกัน ซึ่ง Python นั้นเป็น ภาษาที่มีการตรวจสอบชนิดตัวแปรแบบยืดหยุ่น (dynamically type-checked) และใช้ Garbage collection ในการจัดการหน่วยความจำ

2.1.2 คุณลักษณะเด่นของภาษา Python

- Python สนับสนุนแนวคิดแบบเชิงวัตถุ หรือ OOP (Object Oriented Programming)
- Python เป็นภาษาคอมพิวเตอร์ที่ไม่คิดมูลค่าการใช้งานและเป็นภาษาที่มีความยืดหยุ่นสูงมาก
- โค้ดที่เขียนด้วย Python สามารถนำไป Run บนระบบปฏิบัติการอื่นๆ ได้ เช่น Linux, MS-windows (95, 98, NT, 2000, XP), Amiga, Be-OS, OS/2, VMS, QNX, และระบบอื่น ๆ อีกมากมาย
- Python สนับสนุนเทคโนโลยี COM ของ MS-windows
- Python รวมมาตรฐานการอินเทอร์เฟซ Tkinter ซึ่งสนับสนุนบนระบบ X windows, MS-windows และ Macintosh การใช้คำสั่ง Tkinter API ช่วยให้โปรแกรมเมอร์ไม่ต้องแก้ไขโค้ดเมื่อนำไปรันบนระบบปฏิบัติการอื่นๆ

- Python เป็น Dynamic typing คือ สามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก
- Python มี Built-in Object Types คือ โครงสร้างของข้อมูลที่สามารถใช้ได้ ใน Python ประกอบด้วย ลิสต์, ดิกชันนารี, สตริง ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง
- Python มีเครื่องมือต่างๆ มากมาย เช่น การประมวลผลเท็กซ์ไฟล์ การเรียงข้อมูล การเชื่อมต่อสตริง การตรวจสอบเงื่อนไขของข้อความ การแทนค่า เป็นต้น
- Python มีมอดูลสำหรับจัดการ Regular Expression
- Python มีมอดูลที่สร้างขึ้นจากนักพัฒนาสนับสนุนมากมาย ได้แก่ COM, Image, CORBA, ORBs, XML เป็นต้น
- Python จัดการหน่วยความจำอย่างอัตโนมัติ สามารถจัดการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ
- Python อนุญาตให้ฝังชุดคำสั่งของ Python เอาไว้ภายในโคดภาษา C/C++ ได้
- Python อนุญาตให้โปรแกรมเมอร์สร้าง Dynamic Link Library (DLL) เพื่อใช้ร่วมกับ Python
- Python มีมอดูลสนับสนุนเกี่ยวกับเน็ตเวิร์ก โปรเซส ธรด regular, expression, xml, GUI และอื่นๆ
- Python ประกอบด้วยมอดูลสำหรับสร้าง Internet Script และติดต่อกับอินเทอร์เน็ตผ่าน Sockets, และทำหน้าที่เป็น CGI Script ครอบคลุมใช้งานคำสั่ง FTP, Gopher, XML และอื่นๆ อีกมาก
- Python สามารถประมวลผลทางด้านวิทยาศาสตร์ และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ
- Python มีฟังก์ชันสนับสนุนฐานข้อมูล เช่น MySQL, SQLite, Sybase, Oracle, Informix, ODBC และอื่นๆ
- Python มีไลบรารีสนับสนุนด้านการสร้างภาพกราฟฟิก เช่น ทำภาพเบลอ หรือภาพชัด หรือเขียนข้อความบนภาพ ตลอดจนบันทึกไฟล์ในภาพแบบต่างๆ ได้อย่างสะดวกและมีประสิทธิภาพ
- Python มีไลบรารีสนับสนุนด้านปัญญาประดิษฐ์
- Python มีไลบรารีสำหรับสร้างเอกสาร PDF โดยไม่ต้องติดตั้ง Acrobat Writer
- Python มีไลบรารีสำหรับสร้าง Shockwaves Flash (SWF) โดยไม่ต้องติดตั้ง Macromedia Flash

2.1.3 ตัวแปรและ คำสั่งงานในภาษา Python

Python ไม่จำเป็นต้องประกาศตัวแปรก่อนใช้งาน สามารถกำหนดค่าขึ้นมา และเรียกใช้ได้เลย แต่การตั้งชื่อตัวแปรของ Python ต้องเป็นไปตามกฎดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) ขึ้นต้นด้วยตัวอักษรตั้งแต่ 1 ตัวขึ้นไป
- 2) ตัวแปรห้ามมีช่องว่าง
- 3) ห้ามมีสัญลักษณ์พิเศษ เช่น # , ? , \$, ...
- 4) ตัวแปรต้องไม่ซ้ำกับคำสงวน คำสงวนได้ทำยบความนี้
- 5) ตัวแปรใช้ตัวพิมพ์เล็กพิมพ์ใหญ่ มีความหมายต่างกัน (case-sensitive แบบเดียวกับ PHP)

2.1.4 คำสงวนในภาษา Python

and, as, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while, with, yield

```

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

```

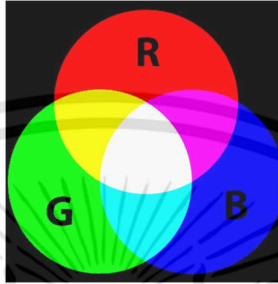
รูปที่ 2.1 โปรแกรม Python 3.10.0 Shell

2.2 OpenCV (Open Source Computer Vision)

OpenCV ย่อมาจาก (Open source Computer Vision) เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเข้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision) เดิมทีแล้วถูกพัฒนาโดย Intel แต่ภายหลังได้รับการสนับสนุนโดย Willow Garage ตามมาด้วย Itseez (ซึ่งต่อมาถูกเข้าซื้อโดย Intel) OpenCV เป็นไลบรารีแบบข้ามแพลตฟอร์ม (Cross-Platform) และใช้งานได้ฟรีภายใต้ลิขสิทธิ์ของ BSD แบบโอเพ่นซอร์ส (Open-Source BSD License) OpenCV ยังสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึก (Deep Learning Frameworks) ได้แก่ TensorFlow, Torch/PyTorch และ Caffe

2.2.1 ระบบสี RGB

RGB ย่อมาจาก Red, Green และ Blue หน้าจอแสดงผลของอุปกรณ์อิเล็กทรอนิกส์เกือบทั้งหมดใช้เทคโนโลยีนี้ในการทำงาน ไม่ว่าจะเป็นจอ LCD, CRT, กล้องถ่ายภาพ, สแกนเนอร์



รูปที่ 2.2 วงล้อสี RGB

2.2.2 ภาพระดับสีเทา (Gray Scale Image)

ภาพระดับสีเทา หรือภาพขาว ดำ เทา ภาพระดับสีเทาเป็นภาพซึ่งแต่ละจุดในภาพคือความเข้มของสี โดยแต่ละตำแหน่งมีความเข้มไม่เท่ากัน มีค่าระดับสีเทาอยู่ในช่วง 0 ถึง 255 เนื่องจากภาพสี RGB เกิดจากการผสมกันของแสงสีแดง สีเขียว สีน้ำเงิน ในอัตราส่วนที่แตกต่างกัน จึงทำให้เป็นไปได้ยาก ที่จะนำภาพสีมาทำการประมวลผล เพื่อให้ง่ายต่อการคำนวณจึงต้องทำการแปลงภาพสีให้เป็นภาพระดับสีเทาก่อน ซึ่งภาพระดับสีเทานั้นเกิดจากการแปลงภาพสี

โดยใช้สูตรทางคณิตศาสตร์ :

$$\text{Gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

Gray คือค่าความเข้มของสีเทามีค่า 0 – 255

R : มีค่า 0 – 255, G : มีค่า 0 – 255, B : มีค่า 0 – 255

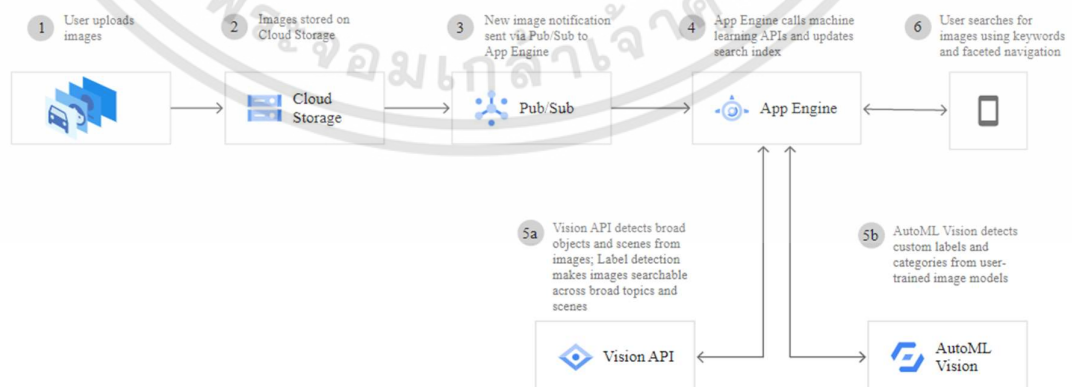


รูปที่ 2.3 ภาพแสดงระดับสีเทา

2.3 Google cloud vision API

API ย่อมาจาก Application Programming Interface คือ ช่องทางการเชื่อมต่อ ช่องทางหนึ่งที่จะเชื่อมต่อกับเว็บไซต์ผู้ให้บริการ API จากที่อื่น เป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น หรือเชื่อมการทำงานเข้ากับระบบปฏิบัติการ

Cloud Vision API นั้นเป็นบริการ machine learning สำเร็จภาพที่สามารถวิเคราะห์ภาพถ่ายได้อย่างแม่นยำที่สามารถนำไปใช้งานได้ทันที บริการนี้ช่วยให้ไม่ต้องกังวลถึงขั้นตอนการพัฒนา ติดตั้ง คุณภาพของโมเดล เพราะกูเกิลจะเป็นผู้จัดการพัฒนาและปรับปรุงประสิทธิภาพอยู่เบื้องหลังทั้งหมด เป็นต้นว่าหากผู้ใช้ลองนำภาพถ่ายในบ้านไปวิเคราะห์ในระบบ บริการ Cloud Vision ก็จะสามารถตอบได้อย่างแม่นยำว่าในภาพนั้นประกอบไปด้วย โต๊ะ โคมไฟ เก้าอี้ และโซฟา อย่างแม่นยำ โดยในบริการ Cloud Vision API จะประกอบไปด้วย Vision API และ AutoML ซึ่งใช้หลักการรู้จำอักขระด้วยแสง (Optical character recognition) และ การเรียนรู้ของเครื่อง (Machine Learning)

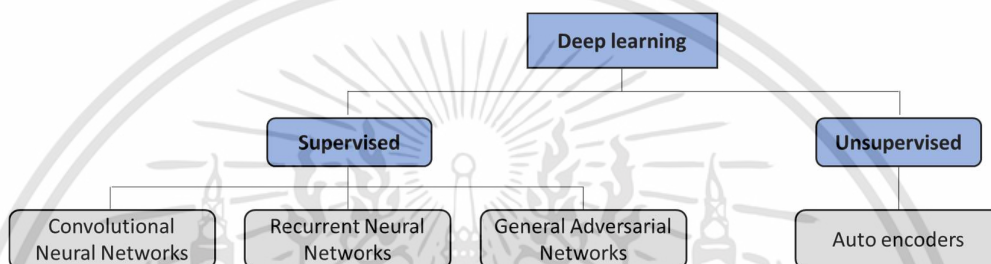


รูปที่ 2.4 หลักการทำงาน Google cloud vision API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การเรียนรู้เชิงลึก (Deep learning)

การเรียนรู้เชิงลึก (deep learning) เป็นส่วนหนึ่งของวิธีการการเรียนรู้ของเครื่อง (Machine learning) บนพื้นฐานของโครงข่ายประสาทเทียมและการเรียนเชิงคุณลักษณะ การเรียนรู้สามารถเป็นได้ทั้งแบบการเรียนรู้แบบมีผู้สอน การเรียนรู้แบบกึ่งมีผู้สอน และการเรียนรู้แบบไม่มีผู้สอน คำว่า "ลึก" ในที่นี้ความหมายมาจากการที่มีชั้นของโครงข่ายหลายชั้น ที่มีประสิทธิภาพมากขึ้น การเรียนที่สะดวกขึ้น และการเข้าใจในโครงสร้างที่ชัดเจนขึ้น การเรียนรู้เชิงลึกมีหลายรูปแบบดังแสดงในรูปที่ 2.5

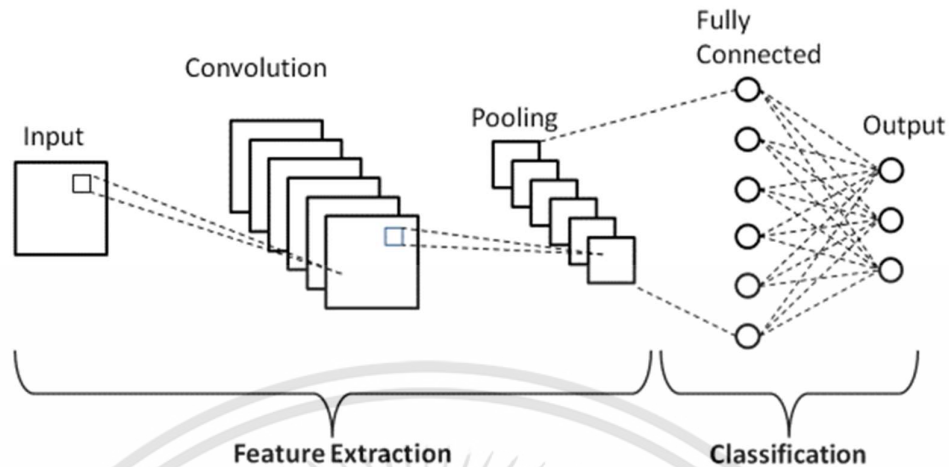


รูปที่ 2.5 ภาพรวมการเรียนรู้เชิงลึก

การเรียนรู้เชิงลึกสามารถแบ่งออกได้หลายประเภท ดังต่อไปนี้ การเรียนรู้แบบมีผู้สอน (Supervised) Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), General Adversarial Networks และการเรียนรู้แบบไม่มีผู้สอน (Unsupervised) Auto encoders โดยในวิทยานิพนธ์นี้เลือกใช้อัลกอริทึม Convolutional Neural Networks (CNN)

2.5 Convolutional Neural Network (CNN)

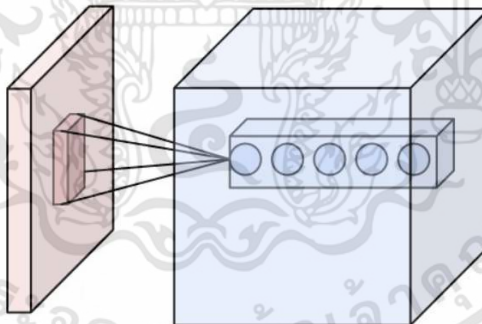
Convolutional Neural Network หรือ CNN ซึ่งเป็นโครงสร้าง Neural network แบบพิเศษ ที่มีความสามารถในการจำแนกข้อมูลประเภทรูปภาพได้ดีกว่า Neural network ทั่วไปมาก หลักการของ CNN คือการใช้ Layer ชนิดพิเศษ ที่เรียกว่า Convolution layer ซึ่งทำหน้าที่สกัดเอาส่วนต่างๆ ของภาพออกมา เช่น เส้นขอบของวัตถุต่างๆ เพื่อให้โมเดลสามารถเรียนรู้ลักษณะของภาพได้อย่างมีประสิทธิภาพและแม่นยำใน CNN จะใช้ Convolution layer มาประกอบกับ Layer ชนิดอื่น เช่น Pooling layer แล้วนำกลุ่ม Layer ดังกล่าวมาซ้อนต่อกัน โดยอาจเปลี่ยน Hyperparameter บางอย่าง เช่นขนาดของ Filter layer (ซึ่งเป็นส่วนหนึ่งของ Convolution layer) และจำนวน Channel ของ layer วิธีการนำเอาส่วนต่างๆ มาประกอบกันนี้ เรียกว่าเป็นโครงสร้าง (Architecture) ของ CNN ซึ่งมีหลายแบบ เช่น LeNet, AlexNet, VGG, ResNet, Inception Network เป็นต้น



รูปที่ 2.6 โครงสร้างโดยทั่วไปของ CNN

2.5.1 ชั้นคอนโวลูชัน (Convolution layer)

Convolution layer คือระดับชั้นที่อยู่แรก ๆ ของโมเดล CNN, Convlayer ทำหน้าที่สกัดเอา Feature สำคัญ จากรูปภาพ, Convlayer มีความพิเศษตรงที่ คงความสัมพันธ์ของ Pixel ที่อยู่บริเวณพื้นที่ใกล้เคียงกันเอาไว้ด้วย เปรียบเทียบกับการทำงานของ Neural network ธรรมดาที่เชื่อมทุก ๆ Neuron ของ layer ก่อนหน้าเข้าด้วยกัน แต่ Convlayer เลือกที่จะเชื่อมแค่บริเวณที่ต้องการเรียกว่า Receptive field



รูปที่ 2.7 Receptive field

2.5.2 ชั้นพูลลิ่ง (Pooling Layer)

Pooling Layer เป็นชั้นที่เชื่อมจาก Convolutional Layer โดยมีเป้าหมายคือทำให้ขนาดของ Feature Map ลดลง ในการคำนวณสามารถใช้ค่าต่ำสุด (Min Pooling) ค่าสูงสุด (Max Pooling) ผลรวม (Sum Pooling) และค่าเฉลี่ย (Average Pooling) ในการคำนวณ Feature Map จะถูกแบ่งออกเป็น บล็อกขนาด ซึ่งหากใช้วิธี Max Pooling ในการคำนวณ ค่าที่ได้ก็คือค่าสูงสุด (Max Value) ของแต่ละบล็อก

| | | | | | |
|---|---|---|---|---|---|
| 2 | 1 | 5 | 7 | 4 | 6 |
| 3 | 4 | 1 | 6 | 2 | 8 |
| 5 | 4 | 6 | 1 | 4 | 7 |
| 8 | 9 | 0 | 1 | 3 | 8 |
| 2 | 4 | 6 | 3 | 5 | 1 |
| 3 | 0 | 0 | 2 | 2 | 6 |

Max-pool
2x2 filters

| | | |
|---|---|---|
| 4 | 7 | 8 |
| 9 | 6 | 8 |
| 4 | 6 | 6 |

รูปที่ 2.8 ภาพประกอบการรวมค่าสูงสุดด้วยตัวกรอง 2x2 และการเลื่อนตัวกรอง

รูปที่ 2.8 แสดงวิธีการคำนวณ Max Pooling จาก Feature Map ที่มีขนาด 6x6 บล็อก ในการคำนวณกำหนด ให้ Pool มีขนาด 2x2 บล็อก ข้อมูลที่อยู่ในบล็อกที่ $f(m_1, n_1)$ ประกอบด้วย $\begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix}$ ดังนั้นผลลัพธ์ที่ได้จากการทำ Max Pooling คือ 4 จากนั้นจึงเลื่อน Pool ไปยังบล็อกถัดไป $f(m_i, n_{j+1})$ และทำไปจนกระทั่งบล็อกสุดท้าย

2.5.3 ชั้นเชื่อมโยงสมบูรณ์ (Fully-Connected Layer)

Fully-Connected Layer ก็คือ Hidden Layer และ Output Layer ของโครงข่ายประสาทเทียม ดังนั้น Fully connected Layer จึงทำหน้าที่ในการเรียนรู้ (Training) และ การจำแนกประเภทของวัตถุ โดยผลลัพธ์ ที่ได้ก็คือจำนวนของ Class ที่ต้องการจำแนก โครงข่ายแบบ CNN สามารถที่จะเพิ่ม Convolutional layer และ Pooling layer ได้อย่างไม่จำกัด

2.6 Pandas

Pandas เป็นไลบรารี Python แบบ open-source ที่มีเครื่องมือจัดการและวิเคราะห์ข้อมูลประสิทธิภาพสูงโดยใช้โครงสร้างข้อมูลที่ชื่อ Pandas มาจากคำว่า Panel data มีจุดเด่นด้านการวิเคราะห์ข้อมูล (Data analysis) และการทำความสะอาด (Data cleaning) ซึ่งเป็น Process ที่สำคัญมากในการทำงานกับข้อมูล

2.6.1 คุณสมบัติของ Pandas

- 1) ออบเจ็กต์ Data frame ที่รวดเร็วและมีประสิทธิภาพพร้อมการสร้าง index เริ่มต้นและ index ที่กำหนดเองได้
- 2) เป็นเครื่องมือสำหรับโหลดข้อมูลลงใน in-memory data objects จากสกุลไฟล์ต่างๆ
- 3) การจัดตำแหน่งข้อมูลและรวมการจัดการข้อมูลที่ขาดหายไป
- 4) Reshaping และ Pivoting data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) การทำ label สำหรับการ slicing, การ indexing และ sub setting ชุดข้อมูลที่มีขนาดใหญ่
- 6) โครงสร้างข้อมูลสามารถ delete หรือ insert ได้
- 7) จัดกลุ่มตาม engine เพื่อให้สามารถใช้ในการดำเนินการ split-apply-combine กับ data set
- 8) การ merging และ joining ของ Data set ที่มีประสิทธิภาพสูง
- 9) ฟังก์ชัน time series: การสร้าง range ของวันและความถี่ของการเปลี่ยนแปลง, การย้าย window statistics, การย้าย window linear regressions, การ shift วัน และการ lagging

2.7 SQL

SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) หมายถึงเราสามารถใส่คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อส่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์และเป็นภาษาหนึ่ง ซึ่งแบ่งการทำงานได้ 4 ประเภทดังนี้

- 1) Select query ใช้สำหรับดึงข้อมูลที่ต้องการ
- 2) Update query ใช้สำหรับแก้ไขข้อมูล
- 3) Insert query ใช้สำหรับการเพิ่มข้อมูล
- 4) Delete query ใช้สำหรับลบข้อมูลออกไป

2.8 MQTT

MQTT (Message Queue Telemetry Transport) คือโพรโตคอลในการส่งข้อมูลที่พัฒนา มาเพื่อใช้ในระบบ IoT มันทำงานแบบ Broker and Clients Network มันถูกออกแบบให้สามารถส่งข้อมูลแบบ Real-Time ในปริมาณข้อมูลทีน้อย ทำให้ใช้พลังงานต่ำมันถูกพัฒนามาจาก TCP/IP ที่มีการส่งข้อมูลแบบ One-To-One ทำให้สิ้นเปลืองทรัพยากรมากซึ่งไม่เหมาะกับระบบ IOT เนื่องจากในระบบ IOT มีการส่งข้อมูลตลอดเวลา และ อุปกรณ์หนึ่งอาจรับหรือส่งข้อมูลไปยังหลายอุปกรณ์ หรือการส่งข้อมูลแบบ One-To-All โดยอุปกรณ์ทุกตัวที่ทำการ Subscriber ไปยัง Topic ใดๆ บน Broker จะได้รับข้อมูลที่ Publisher ส่งให้ Topic นั้นๆบน Broker ทั้งหมด

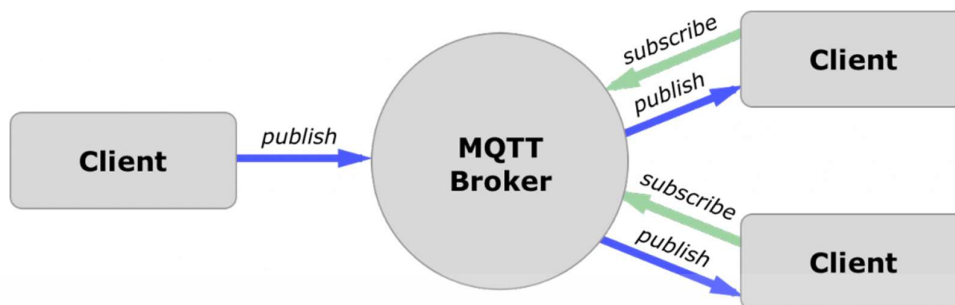
2.8.1 หลักการทำงานของ MQTT

โพรโทคอล MQTT แบ่งบทบาทของอุปกรณ์ในเน็ตเวิร์คออกเป็น 2 ประเภท ได้แก่ Message Broker และ Client MQTT Broker เป็นเซิร์ฟเวอร์ซึ่งทำหน้าที่รับข้อความที่ส่งจาก Client ตัวหนึ่งและนำส่ง (Route) ไปยัง Client อีกตัวหนึ่ง MQTT Broker สามารถรันได้หลาย Platform ไม่ว่าจะเป็น Window, MAC OS, Linux, Ubuntu และ Raspberry Pi โดยที่เราสามารถสร้าง Broker ขึ้นใช้งานเอง (On-premises) เนื่องจากเป็น Open Source MQTT Client เป็นอุปกรณ์ที่ทำหน้าที่ส่งข้อความไปยัง Broker และในขณะเดียวกันก็สามารถรับข้อความจาก Broker ได้ด้วย โดย Client นั้น อาจจะเป็นอุปกรณ์ประเภทใดก็ได้ เช่น ไมโครคอนโทรลเลอร์, พีแอลซี, อาδυโน่, โทรศัพท์มือถือ รวมไปถึงคอมพิวเตอร์ด้วย ขอให้มีไลบรารีสำหรับรัน MQTT Client ได้ด้วยโมเดลการเชื่อมต่อแบบ Publish-Subscribe ทำให้การวางเน็ตเวิร์คสำหรับ MQTT นั้นไม่จำเป็นต้องให้อุปกรณ์ทุกตัวอยู่บนเน็ตเวิร์คเดียวกัน หรือทำ VPN กัน เพียงแต่ทำให้ MQTT Broker อยู่บน IP Address ที่เป็น Public เพียงตัวเดียว ส่วน Client ตัวอื่นๆ ไม่จำเป็นต้องเป็น Public IP ขอเพียงแค่สามารถเข้าถึง Internet ได้ เพียงเท่านี้ Client1 ก็จะสามารถส่งข้อความหา Client2 ได้โดยที่ Client1 ไม่จำเป็นต้องเชื่อมต่อกับ Client2 โดยตรง แต่ใช้วิธีส่งหา Broker ที่ทำหน้าที่เป็นตัวกลางในการรับ-ส่งข้อมูลแทนนั่นเอง

2.8.2 การทำงานของ MQTT Client

MQTT Client นั้นจะมีคำสั่งในการส่งไปยัง Broker ทั้งหมด 5 คำสั่ง

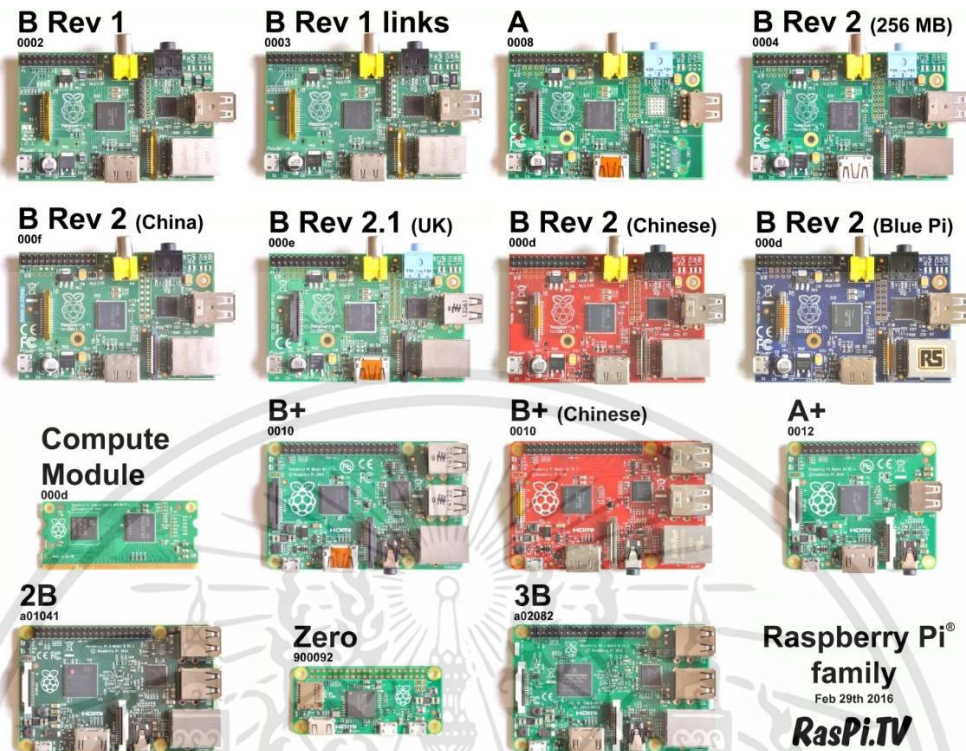
1. Connect เป็นการร้องขอการเชื่อมต่อไปยัง Broker ในขั้นตอนนี้ Client จะต้องระบุ Address ของ Broker, Port ที่ใช้ซึ่งโดยปกติจะใช้ Port 1883
2. Disconnect เป็นการร้องขอให้ตัดการเชื่อมต่อจาก Broker
3. Publish เป็นการส่งข้อความจาก Client ขึ้นไปที่ Broker โดยการกำหนดตัวแปรบน Broker จะใช้สิ่งที่เรียกว่า Topic เป็นตัวกำหนด
4. Subscribe Client จะสามารถ Subscribe Topic ที่อยู่บน Broker ได้และเมื่อมี Client อื่นทำการ publish ค่าใหม่ไปที่ Topic ที่กำลัง Subscribe อยู่เมื่อนั้นตัว Client จะได้รับข้อความใหม่ทันทีโดยไม่ต้องร้องขอ
5. Unsubscribe คือคำสั่งยกเลิกการ Subscribe จาก Topic ใด Topic หนึ่งโดยที่สถานะยัง Connect กับ Broker อยู่



รูปที่ 2.9 MQTT Protocol

2.9 Raspberry Pi

Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็กพัฒนาขึ้นโดยมูลนิธิ Raspberry Pi ซึ่งเป็นองค์การกุศลของสหราชอาณาจักร ที่ทำงานเพื่อนำพลังด้านดิจิทัลเข้าสู่ผู้ใช้งานทั่วโลก ดังนั้นผู้ใช้งานสามารถทำความเข้าใจและสร้างโลกดิจิทัลเพิ่มขึ้นได้โดยง่าย สามารถแก้ปัญหาต่าง ๆ ที่สำคัญได้และเตรียมพร้อมสำหรับงานในอนาคต ซึ่ง Raspberry Pi เป็นคอมพิวเตอร์ที่มีประสิทธิภาพสูง ราคาประหยัด และมีประสิทธิภาพสูงที่ผู้คนใช้เพื่อเรียนรู้ในการแก้ปัญหาและได้รับความสนุกสนาน อีกทั้งมีชุมชนออนไลน์พัฒนาแหล่งข้อมูลฟรี เช่น บทความ ตัวอย่างโครงการ เพื่อช่วยให้ผู้คนเรียนรู้เกี่ยวกับคอมพิวเตอร์และวิธีการทำสิ่งต่าง ๆ กับคอมพิวเตอร์ ไม่ว่าจะใช้งานด้านทั่วไป หรือ ทักษะการเขียนโปรแกรม ซึ่งสามารถประหยัดค่าใช้จ่ายในการเรียนรู้โดยเฉพาะการเขียนโปรแกรม



รูปที่ 2.10 บอร์ดและรุ่น Raspberry Pi

2.9.1 คุณสมบัติของ Raspberry Pi (Raspberry Pi3 Model B)

Raspberry Pi สามารถเชื่อมต่อระบบเครือข่ายแบบใช้สายหรือไร้สายได้ ทำให้กลายเป็นอุปกรณ์ Internet of Things โดยสมบูรณ์ ช่วยให้นักวิจัยและผู้สนใจอื่น ๆ สามารถนำไปประยุกต์ใช้เพื่อเชื่อมต่อกับเซนเซอร์ ในการเก็บข้อมูลตามต้องการ รวมถึงสามารถเชื่อมต่อกับแป้นพิมพ์และเมาส์ได้ง่ายอีกด้วย โดยระบบปฏิบัติการที่ใช้กันนั้น คือ Raspbian ซึ่งเป็นระบบปฏิบัติการลินุกซ์เป็นฐานถูกปรับแต่งมาใช้กับ Raspberry Pi โดยเฉพาะ และระบบปฏิบัติการ ติดตั้งผ่าน Micro SD Card สามารถตั้งค่าเป็นเครื่องแม่ข่ายและใช้งานบริการต่าง ๆ เช่น Web Server, FTP Server ได้เป็นต้น โดยมีคุณสมบัติดังนี้

CPU: Quad-core 1.2 GHz ARM Cortex-A53 แบบ 64 bits

GPU: Broadcom Video Core IV @ 400 MHz

หน่วยความจำ ขนาด 1 GB (LPDDR2-900 SDRAM)

หน่วยความจุแบบ microSD

4 USB ports

1 Ethernet port

802.11n Wireless LAN

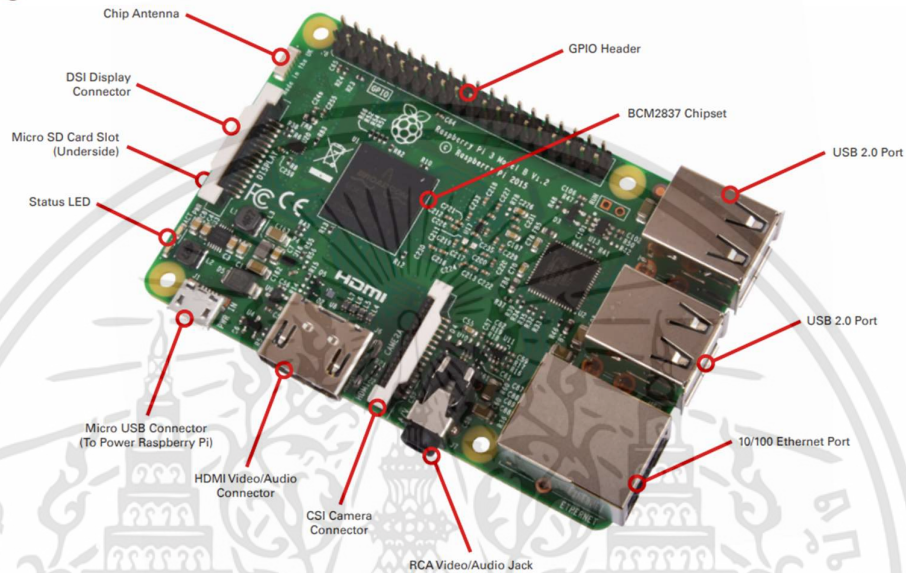
Bluetooth 4.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รองรับ HDMI/Composite ผ่านทาง RCA Jack
GPIO 40 pins

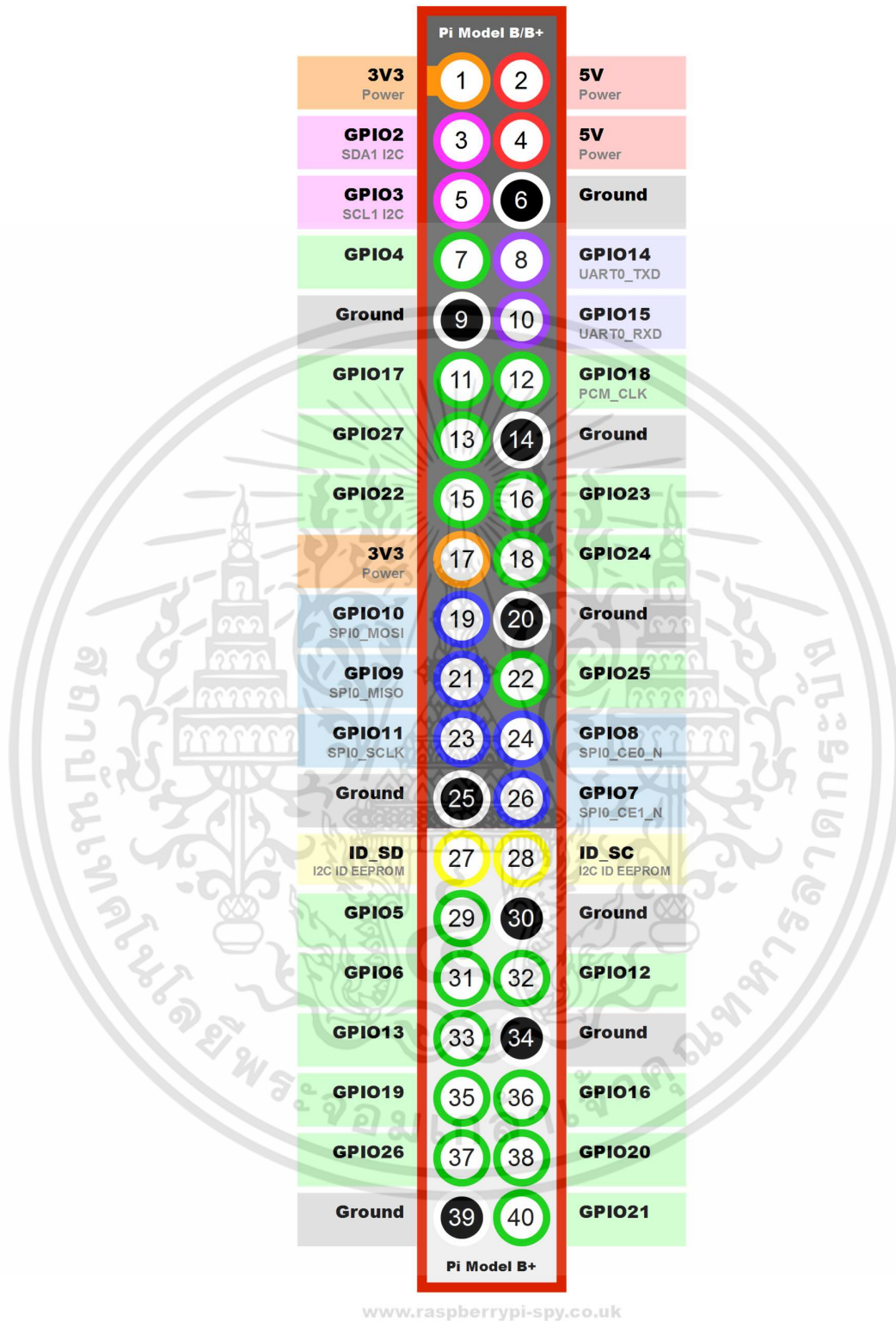


Raspberry Pi 3 Model B



รูปที่ 2.11 พอร์ตของบอร์ด Raspberry Pi 3 Model B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 จุดเชื่อมต่อ GPIO ของบอร์ด Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 ระบบปฏิบัติการของบอร์ด Raspberry Pi

RASPBIAN
 SNAPPY UBUNTU CORE
 OPNELEC
 RASPBMC
 PIDORA
 RISC OS

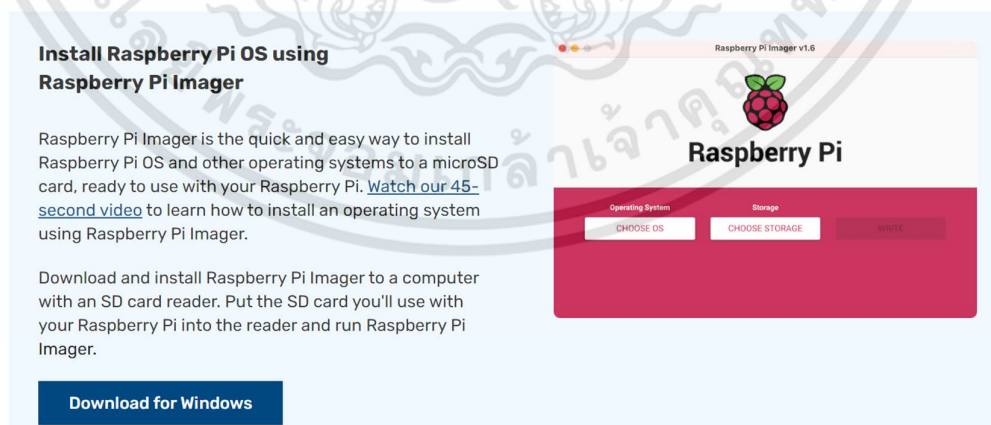
2.9.3 อุปกรณ์ที่ต้องใช้กับบอร์ด Raspberry Pi

บอร์ด Raspberry Pi 3 Model B
 Micro SD card ขนาด 2 GB ขึ้นไป (แนะนำ class 10)
 สาย Micro USB
 สาย HDMI
 หน้าจอแสดงผล (สามารถเชื่อมต่อแบบ HDMI)
 USB Keyboard และ USB Mouse
 WIFI
 คอมพิวเตอร์ และ Card Reader

2.9.4 การติดตั้งระบบปฏิบัติการ

ดาวน์โหลดตัวติดตั้งอิมเมจระบบปฏิบัติการจาก

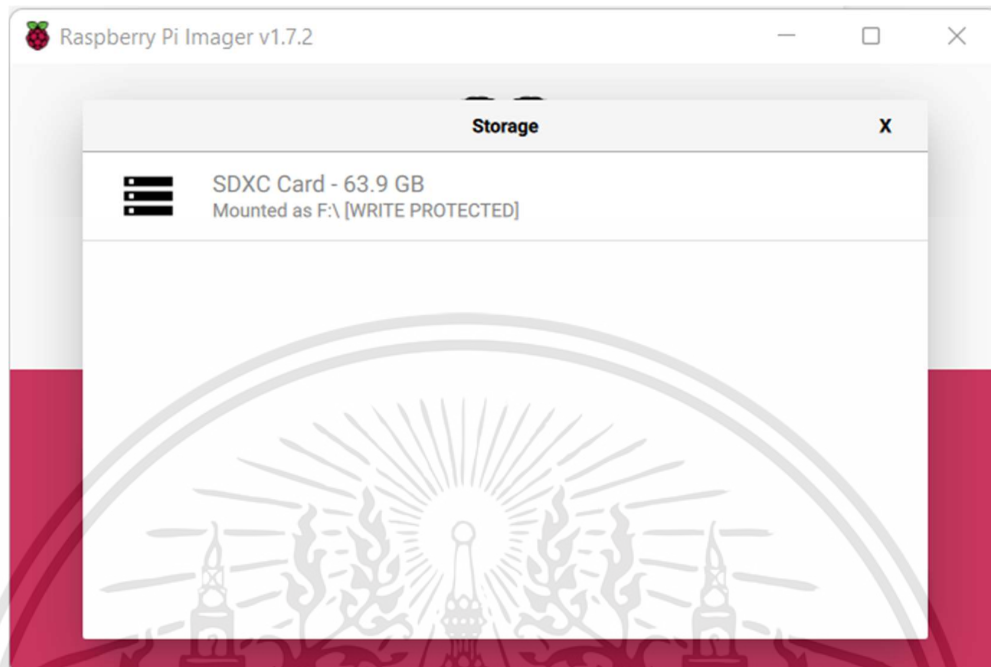
<https://www.raspberrypi.com/software/> ดังรูป



รูปที่ 2.13 Raspberry Pi imager

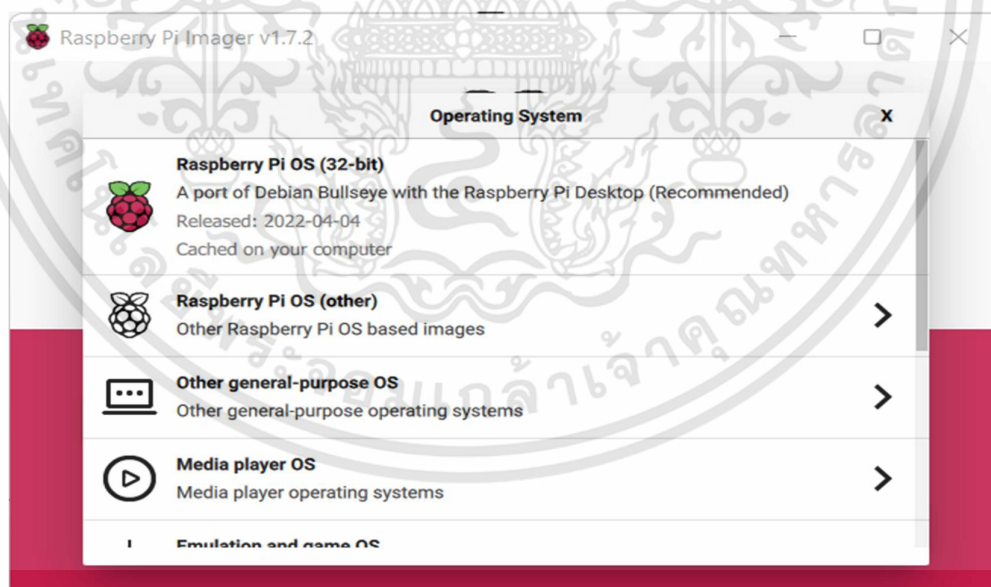
ทำการเลือกที่ใคร่พ็ของ Micro SD card ในที่นี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 เลือกไดรฟ์ที่ต้องการติดตั้ง Raspberry Pi OS

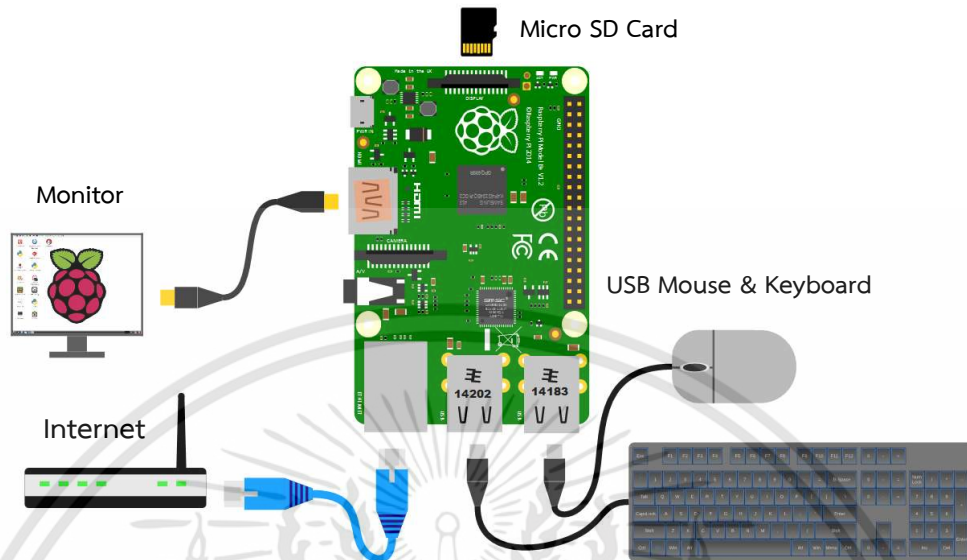
ทำการเลือก Raspberry Pi OS (32bit) ดังรูปที่



รูปที่ 2.15 ระบบปฏิบัติการ Raspberry Pi OS (32bit)

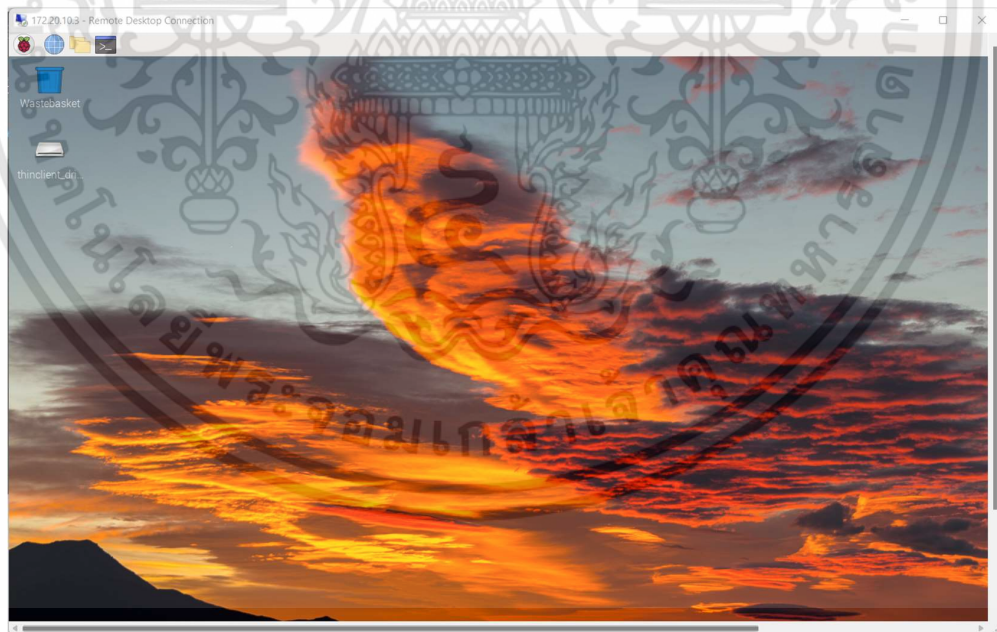
เมื่อทำการติดตั้งเสร็จแล้วให้ถอด Micro SD card ออกจากคอมพิวเตอร์แล้วใส่เข้าไปที่ Raspberry Pi ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 การเชื่อมต่ออุปกรณ์ต่างๆกับบอร์ด Raspberry Pi

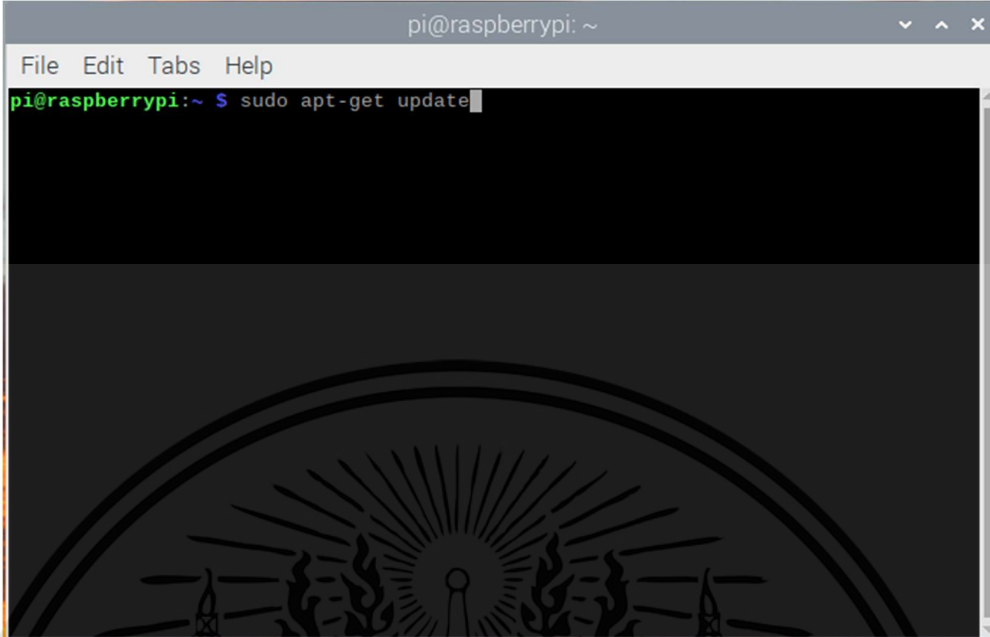
เมื่อทำการเชื่อมต่ออุปกรณ์ที่จำเป็นเข้ากับบอร์ด Raspberry Pi เป็นที่เรียบร้อยแล้ว บอร์ดจะทำการรันและเข้าสู่หน้าจอ GUI



รูปที่ 2.17 หน้าจอแสดงผล GUI

ทำการอัปเดตโปรแกรมให้ใหม่ล่าสุดด้วยคำสั่ง `sudo apt-get update` บน terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



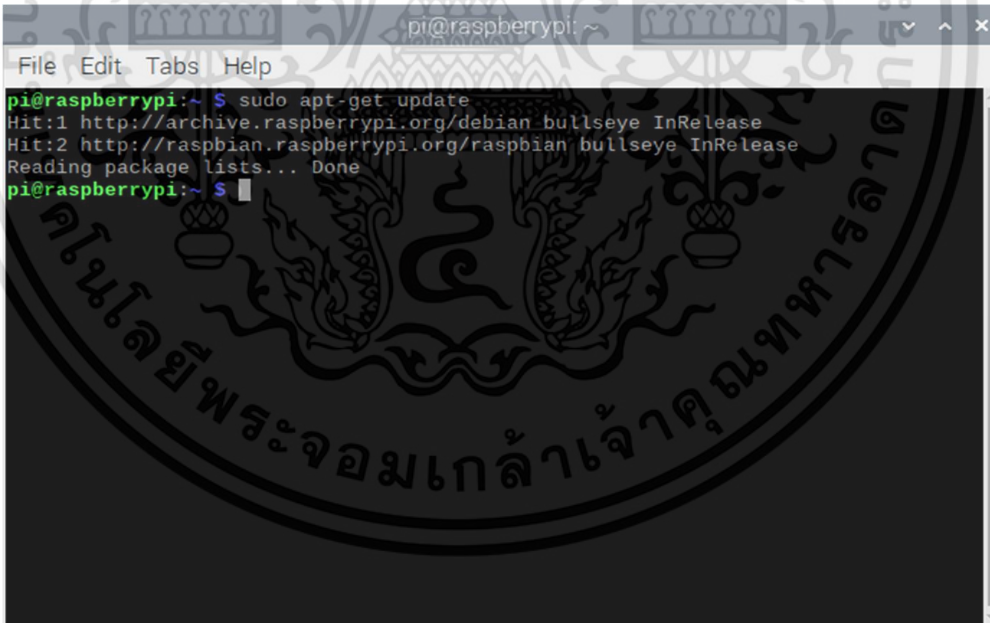
```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get update

```

รูปที่ 2.18 ใช้คำสั่ง sudo apt-get update บน terminal

ทำการรออัปเดต



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get update
Hit:1 http://archive.raspberrypi.org/debian bullseye InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease
Reading package lists... Done
pi@raspberrypi:~ $

```

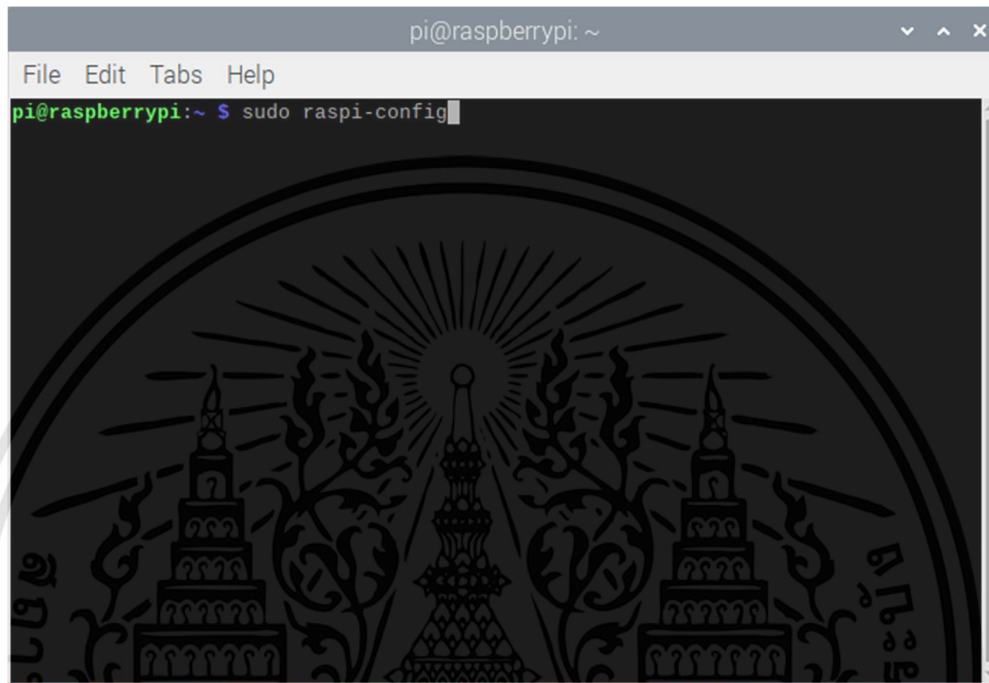
รูปที่ 2.19 เมื่อทำการอัปเดตเสร็จเรียบร้อยแล้ว

2.9.5 การใช้งาน Raspberry Pi แบบ Remote Terminal (SSH)

SSH ย่อมาจาก Secure Shell คือช่องทางที่สามารถแลกเปลี่ยนข้อมูลโดยช่องทางที่ปลอดภัย Secure Channel ระหว่างอุปกรณ์เครือข่ายสองตัว ใช้ Linux หรือ Unix เป็น

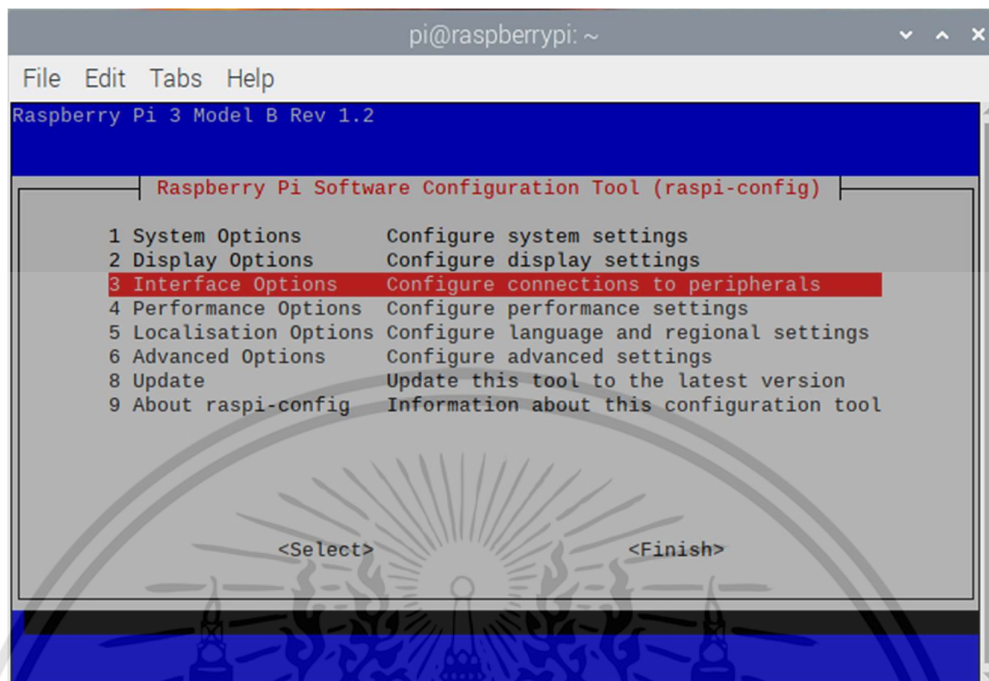
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบปฏิบัติการพื้นฐานในการเข้าถึงบัญชีผู้ใช้ Shell Accounts ซึ่งก็คือช่องทางที่เราสามารถリモทเข้าไปใช้ command line บน Raspberry Pi จากคอมพิวเตอร์เครื่องอื่นได้ บนเน็ตเวิร์คเดียวกัน โดยสามารถทำการเปิดใช้ SSH ได้ตั้งรูป 2.20



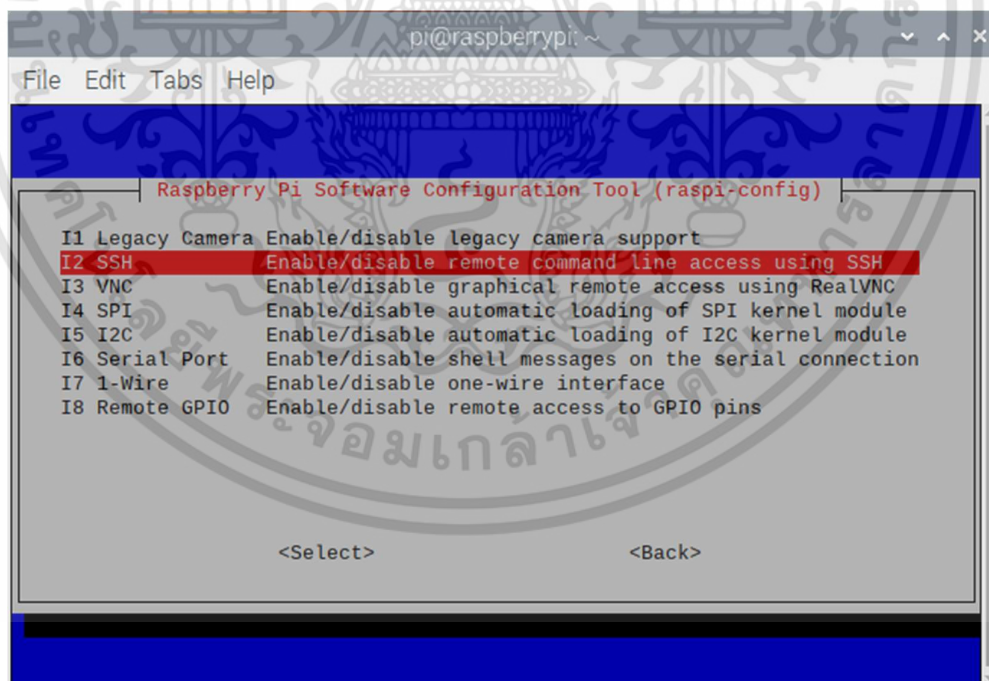
รูปที่ 2.20 ใช้งานคำสั่ง sudo raspi-config

เมื่อพิมพ์คำสั่ง sudo raspi-config เรียบร้อยจะขึ้นหน้าจอตั้งค่าของบอร์ด Raspberry Pi ทำการเลือกเมนูที่ 3 Interface Options เพื่อทำการตั้งค่าในรูปที่ 2.17



รูปที่ 2.21 การตั้งค่า Interface ของ Raspberry Pi

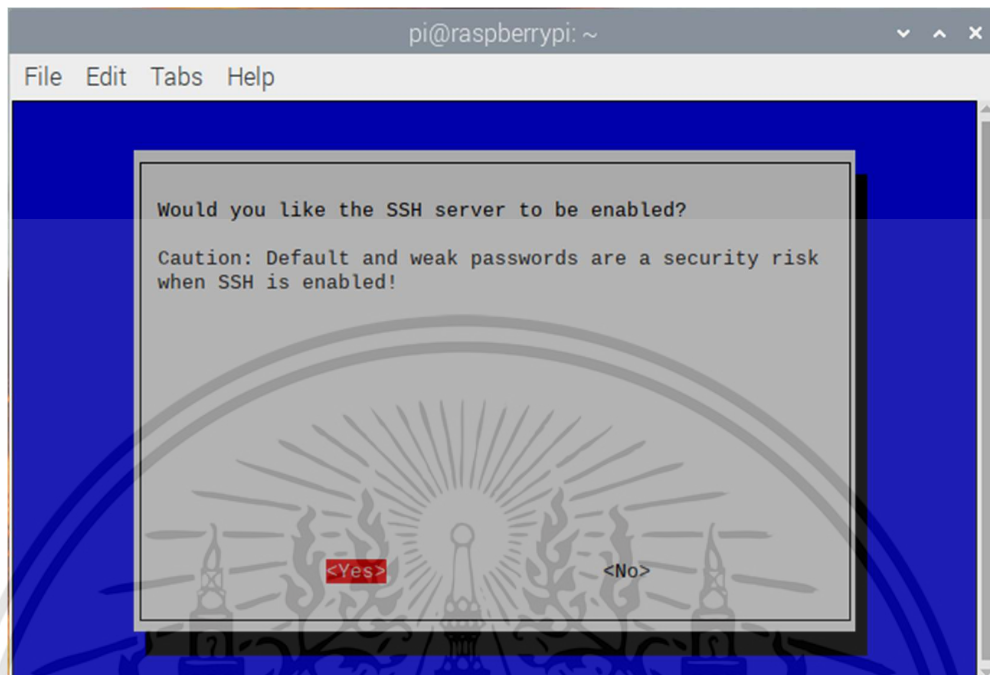
ทำการเลือก SSH แล้วกด Enter



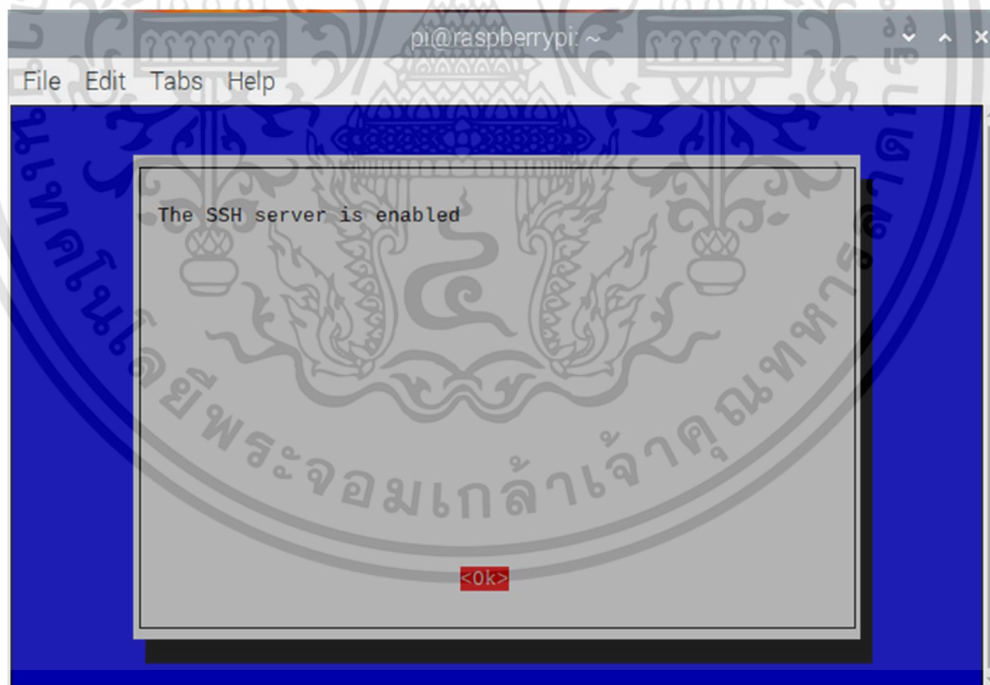
รูปที่ 2.22 เลือก SSH ในตั้งค่า

ทำการเลือกเปิดใช้งาน SSH server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



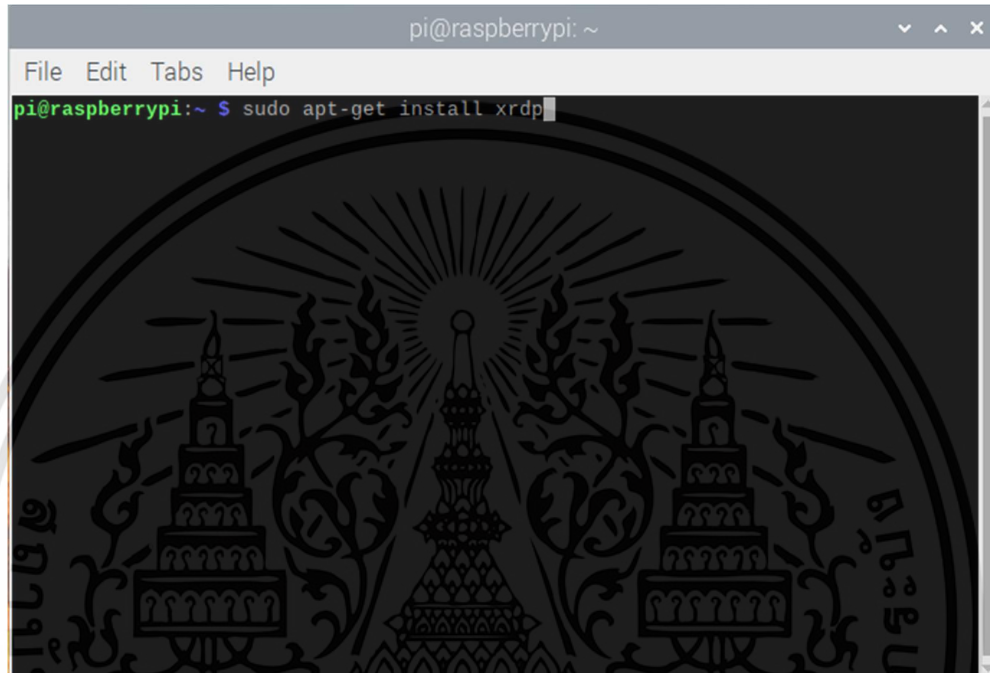
รูปที่ 2.23 เลือกเปิดใช้งาน SSH



รูปที่ 2.24 ทำการเปิดใช้งาน SSH เรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการติดตั้ง RDP หรือ Remote Desktop Protocol คือ โพรโตคอลที่พัฒนาโดย Microsoft ที่ไว้สำหรับเชื่อมต่อกับคอมพิวเตอร์เครื่องอื่น ผ่านระบบเครือข่าย ซึ่งแสดงผลแบบกราฟฟิก หรือ GUI (Graphic User Interface) สามารถติดตั้งโดยใช้คำสั่ง `sudo apt-get install xrdp` ใน Terminal ดังรูปที่ 2.21



รูปที่ 2.25 ใช้คำสั่ง `sudo apt-get install xrdp`

ทำการติดตั้ง

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ sudo apt-get install xrdp
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
xrdp is already the newest version (0.9.17-2~bpo11+rpt1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pi@raspberrypi:~$

```

รูปที่ 2.26 เมื่อทำการติดตั้ง xrdp เสร็จเรียบร้อย

ดูไอพีแอดเรสของ Raspberry Pi โดยใช้คำสั่ง ifconfig

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ ifconfig

```

รูปที่ 2.27 ใช้คำสั่ง ifconfig

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pi@raspberrypi: ~
File Edit Tabs Help
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 42 bytes 4777 (4.6 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 42 bytes 4777 (4.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

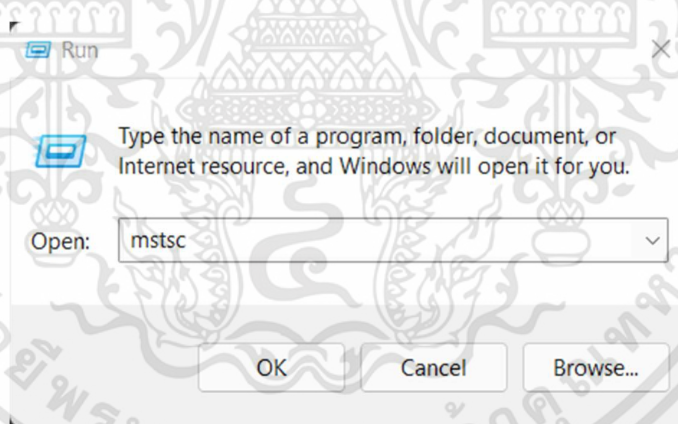
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 172.20.10.5 netmask 255.255.255.240 broadcast 172.20.10.15
inet6 2001:44c8:42c0:676a:ae17:15cf:4ee:a54 prefixlen 64 scopeid 0x0<g
lobal>
inet6 fe80::dec4:862e:6234:89ee prefixlen 64 scopeid 0x20<link>
ether b8:27:eb:ac:a5:f5 txqueuelen 1000 (Ethernet)
RX packets 20099 bytes 1294747 (1.2 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 73067 bytes 99318923 (94.7 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi: ~$

```

รูปที่ 2.28 ไอพีแอดเดรสของ Raspberry Pi

เรียกใช้ Run command โดยการกดปุ่ม Window พร้อมกับปุ่ม R และพิมพ์ mstsc



รูปที่ 2.29 เรียกใช้ Run command และพิมพ์ mstsc

เมื่อกดตกลงจะแสดงหน้าต่างโปรแกรม Remote Desktop Connection ขึ้นมาให้เราใส่ไอพีแอดเดรสของ Raspberry Pi



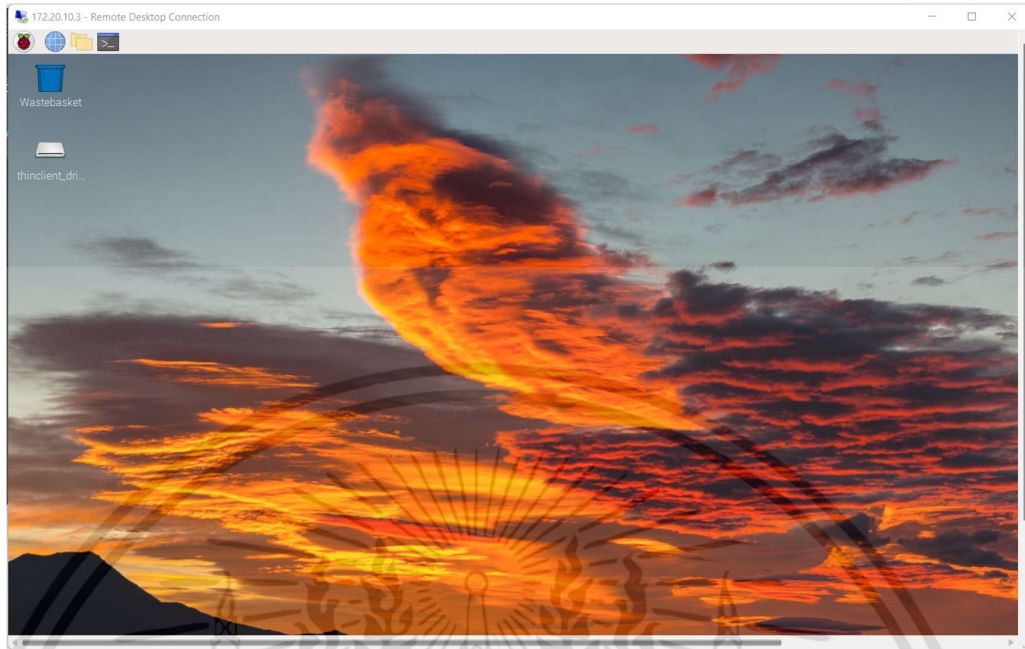
รูปที่ 2.30 โปรแกรม Remote Desktop Connection

เมื่อใส่ไอพีแอดเดรสและทำการกดเชื่อมต่อจะขึ้นหน้าต่าง ให้ทำการใส่ username: pi และ password: raspberry จากนั้นกดตกลงเพื่อเข้าสู่ Raspberry Pi OS



รูปที่ 2.31 หน้า Login ของ Remote Desktop Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.32 หน้า GUI ของ Raspberry Pi OS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงาน

3.1 อุปกรณ์ที่ใช้ในระบบ

ในส่วนนี้จะเป็นอุปกรณ์ที่ใช้ทั้งหมดในระบบติดตามการจราจรรถยนต์ โดยการใช้การประมวลผลภาพ



รูปที่ 3.1 อุปกรณ์ทั้งหมดที่ใช้ในระบบ

3.1.1 Computer

Computer เป็นตัว Input ข้อมูลแบนด์วิดธ์ และ ป้ายทะเบียนเข้าไปในระบบ และมีหน้าที่ Remote Desktop ไปยัง Raspberry Pi



รูปที่ 3.2 คอมพิวเตอร์

3.1.2 Raspberry Pi 3

Raspberry Pi 3 เป็นส่วน Station ที่ใช้ในการเก็บ Data ที่ส่งมาจาก Websocket HIVE MQ และ นำข้อมูลที่ได้ไปใช้ในระบบประมวลผลภาพ



รูปที่ 3.3 Raspberry Pi 3

3.1.3 กล้อง Webcam 1080p รุ่น A875-A

กล้อง Webcam 1080p รุ่น A875-A เป็นรุ่นที่ใช้ในการตรวจจับรถยนต์ โดยเชื่อมต่อกล้องเข้ากับ Raspberry Pi 3



รูปที่ 3.4 กล้อง Webcam 1080p รุ่น A875-A

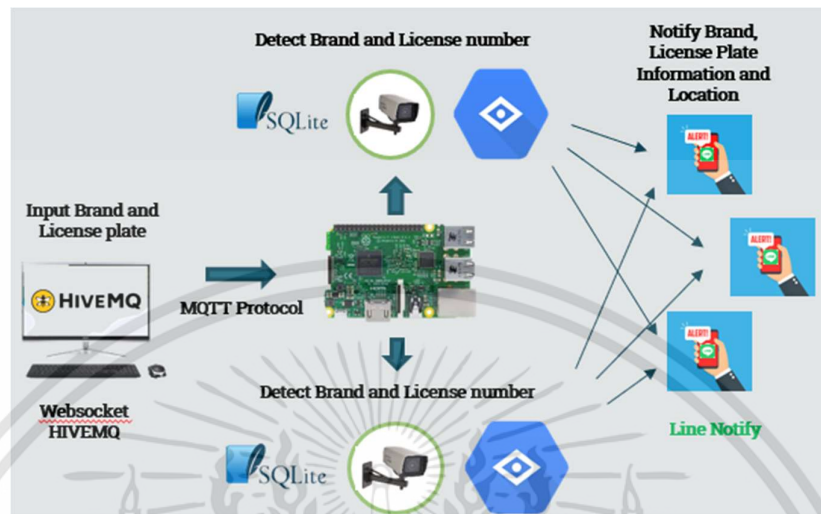
3.1.4 Power Bank

Power Bank ขนาด 5V 3A เป็นแหล่งจ่ายไฟให้กับ Raspberry Pi 3



รูปที่ 3.5 Power Bank

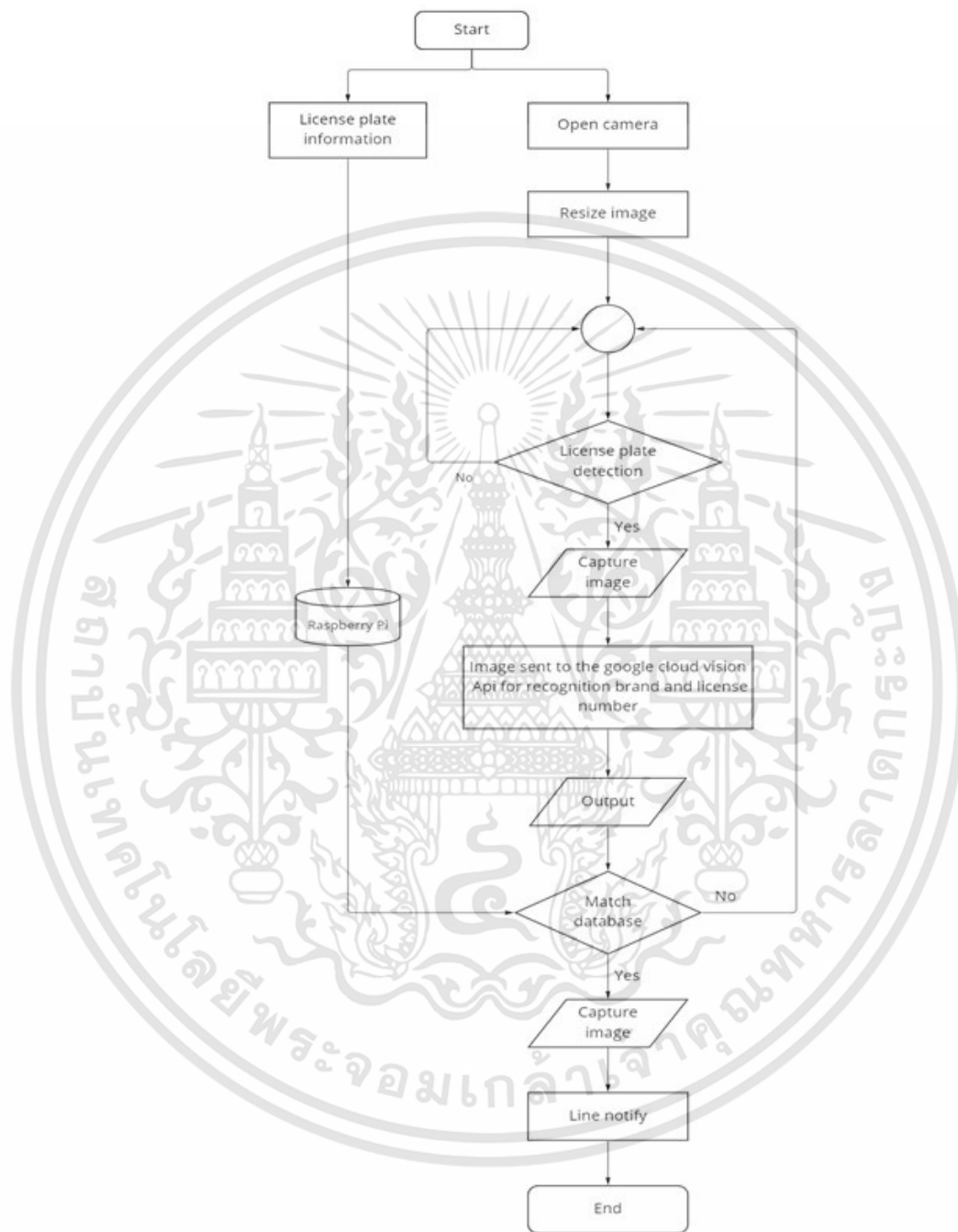
3.2 การออกแบบระบบ



รูปที่ 3.6 รูปแบบการทำงานของระบบ

จากรูปที่ 3.6 จะเป็นการทำงานของระบบติดตามการจราจรรถยนต์ โดยใช้การประมวลผลภาพเริ่มจากการป้อนข้อมูล แบนด์รถยนต์ และ ป้ายทะเบียนรถที่ต้องการตรวจจับ ผ่านโปรแกรม WebSocket HIVEMQ เพื่อส่งข้อมูลไปเก็บใน Raspberry Pi ซึ่งใช้ฐานข้อมูลของ SQLite หลังจากนั้นกล้องจะส่งรูปที่ถ่ายได้ไปให้ระบบ Google Cloud เมื่อกำลังตรวจจับรถยนต์ที่ถูกโจรกรรมได้ จะทำการตรวจจับด้วยระบบการประมวลผลภาพ Google Cloud ได้เป็นแบนด์รถยนต์, ป้ายทะเบียนรถยนต์ ออกมาในรูปแบบตัวอักษร จากนั้นระบบจะนำข้อมูลรถยนต์ที่ได้เปรียบเทียบกับข้อมูลรถยนต์ที่ถูกโจรกรรมในฐานข้อมูล ถ้าพบว่าข้อมูลตรงกับรถยนต์ที่ถูกโจรกรรม ระบบจะส่งตำแหน่งรถยนต์ ข้อมูลรถยนต์ และ รูปถ่ายรถยนต์ ให้กับเจ้าหน้าที่หรือผู้ที่รับผิดชอบผ่านทาง แอปพลิเคชันไลน์

3.3 กระบวนการติดตามการจราจรรถยนต์ โดยใช้การประมวลผลภาพ



รูปที่ 3.7 กระบวนการติดตามการจราจรรถยนต์

3.4 การติดตั้งโปรแกรม Python และไลบรารีที่จำเป็น เพื่อให้ใช้งานร่วมกัน

เนื่องจากโปรแกรม Python และไลบรารีต่าง ๆ นั้นยังไม่สามารถใช้งานร่วมกันได้ตั้งแต่แรก จึงต้องทำให้โปรแกรม Python และไลบรารีสามารถใช้งานร่วมกันได้ โดยจะทำการพูดถึงตั้งแต่เริ่มติดตั้งโปรแกรม Python และไลบรารีที่จำเป็น

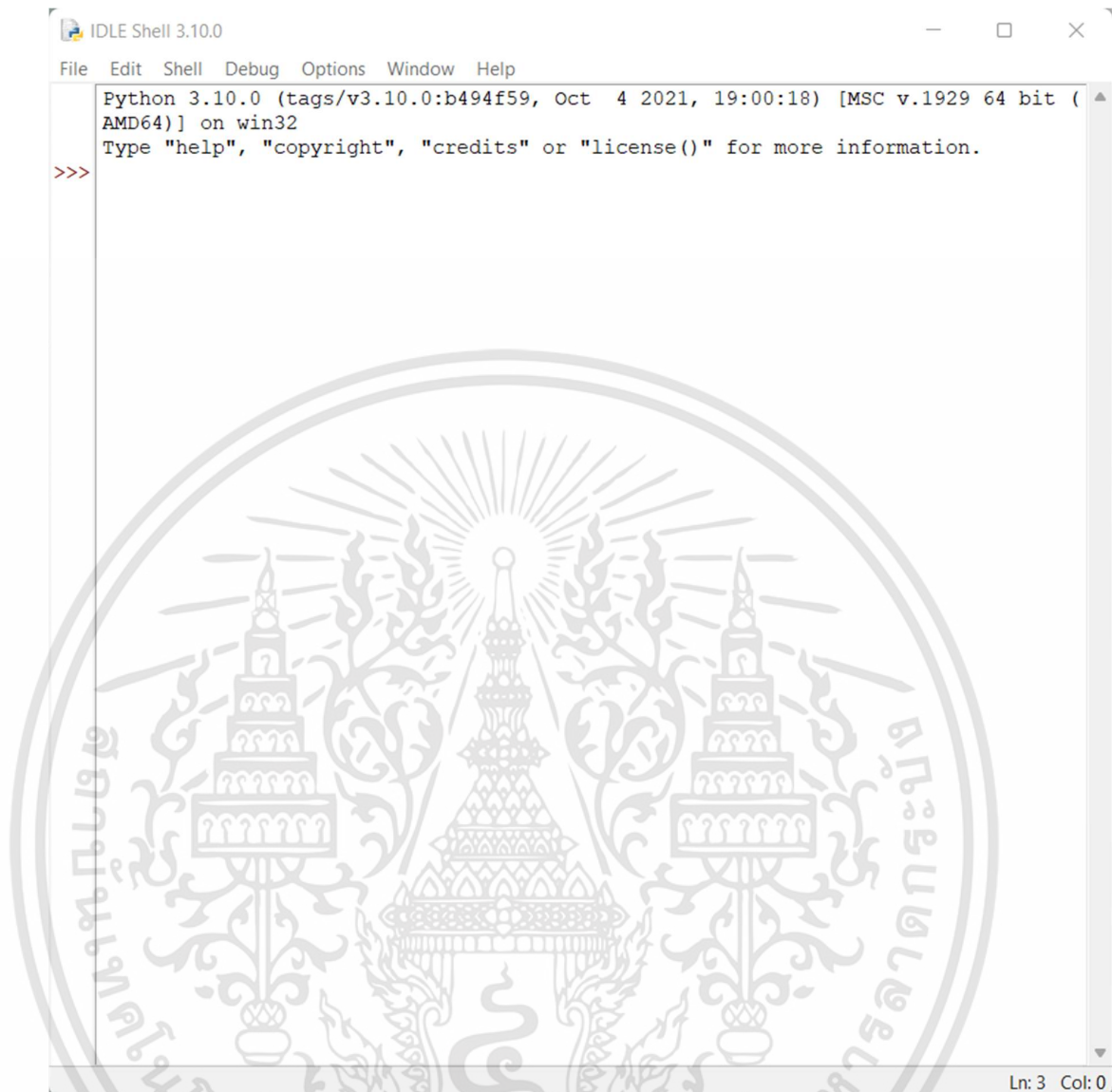
3.4.1 การติดตั้งโปรแกรม Python

เริ่มจากการติดตั้งโปรแกรม Python จาก <https://www.python.org/downloads> และทำการติดตั้ง



รูปที่ 3.8 <https://www.python.org/downloads>

เมื่อทำการติดตั้งเรียบร้อยแล้วจะได้โปรแกรม IDLE Shell 3.10.0 ดังรูปที่ 3.9



```

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
Ln: 3 Col: 0

```

รูปที่ 3.9 Python 3.10.0 Shell

3.4.2 pip

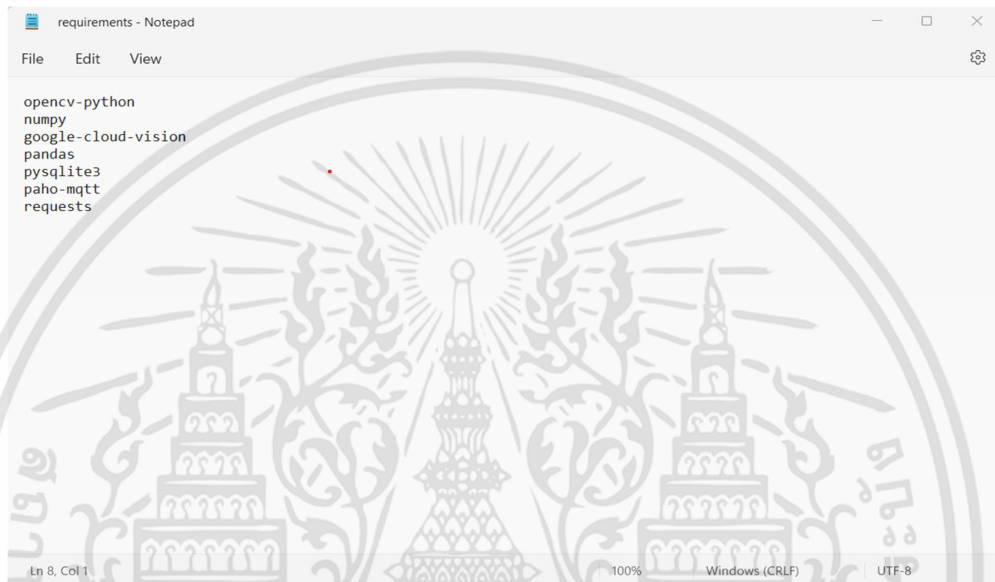
ในการใช้ภาษาไพธอน นอกจากมอดูลมาตรฐานที่มีอยู่แต่แรกแล้ว บ่อยครั้งจำเป็นต้องติดตั้งมอดูลเสริม (แพ็คเกจเสริม) เพิ่มเติมที่จำเพาะสำหรับงานที่ต้องการวิธีการลงมอดูลใหม่นั้นมีหลายวิธีด้วยกัน แต่ที่นิยมมากที่สุดคือใช้ pip

3.4.3 ทำการติดตั้งไลบรารีโดยใช้ pip

ทำการสร้างไฟล์ .txt เพื่อติดตั้งไลบรารีทั้งหมดโดยมีไลบรารีที่สำคัญดังนี้

1. opencv-python
2. numpy

3. google-cloud-vision
4. pandas
5. pysqlite3
6. paho-mqtt
7. requests



```

requirements - Notepad
File Edit View

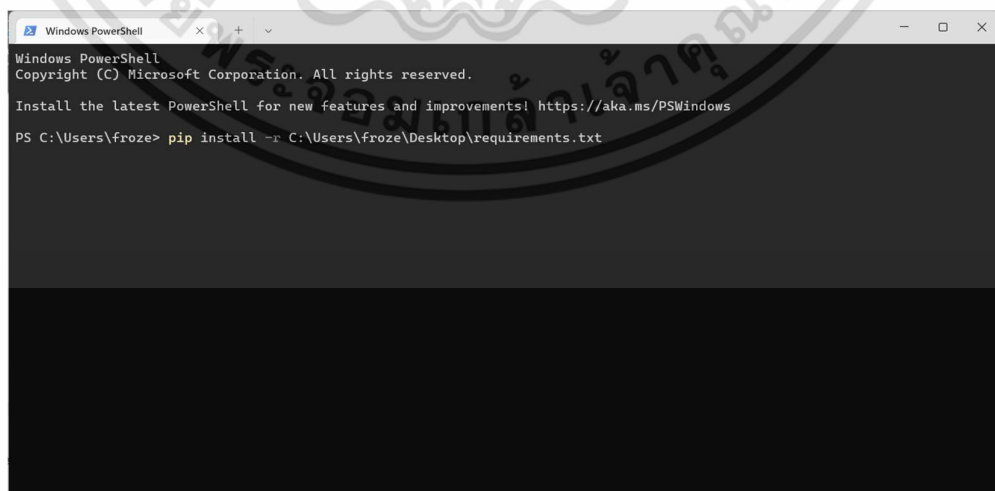
opencv-python
numpy
google-cloud-vision
pandas
pysqlite3
paho-mqtt
requests

Ln 8, Col 1 100% Windows (CRLF) UTF-8

```

รูปที่ 3.10 ไฟล์ requirements.txt

เรียกใช้ Window PowerShell และพิมพ์คำสั่ง pip install r C:\Users\froze\Desktop\requirements.txt โดยต้องกำหนดที่อยู่ของไฟล์หลัง pip install r (ที่อยู่ไฟล์) ดังรูปที่ 3.11



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

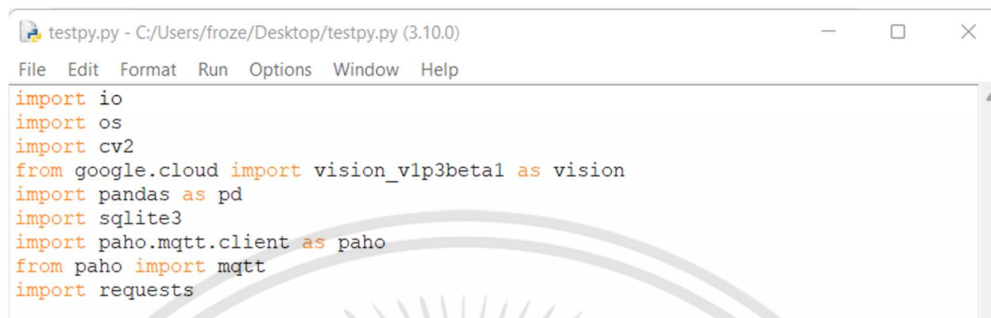
PS C:\Users\froze> pip install -r C:\Users\froze\Desktop\requirements.txt

```

รูปที่ 3.11 พิมพ์คำสั่งใน Windows PowerShell

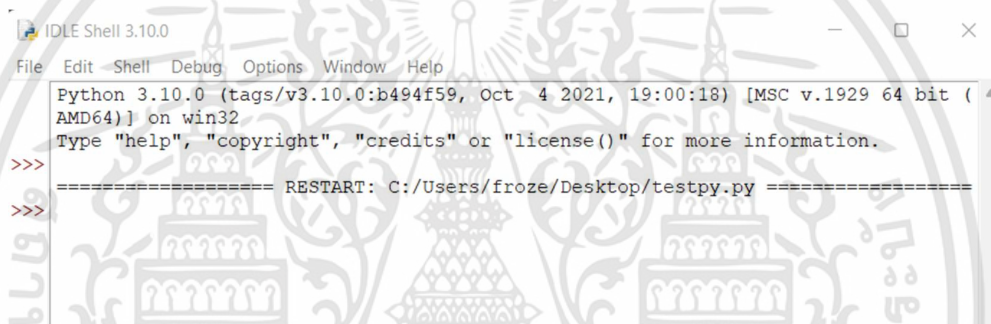
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการตรวจสอบไลบรารีโดยการให้สั่งทำงานหากไม่มีข้อผิดพลาดเกิดขึ้นถือว่าติดตั้งสำเร็จ ดังรูปที่ 3.13



```
testpy.py - C:/Users/froze/Desktop/testpy.py (3.10.0)
File Edit Format Run Options Window Help
import io
import os
import cv2
from google.cloud import vision_v1p3beta1 as vision
import pandas as pd
import sqlite3
import paho.mqtt.client as paho
from paho import mqtt
import requests
```

รูปที่ 3.12 คำสั่งเรียกใช้ไลบรารีที่ทำการติดตั้ง

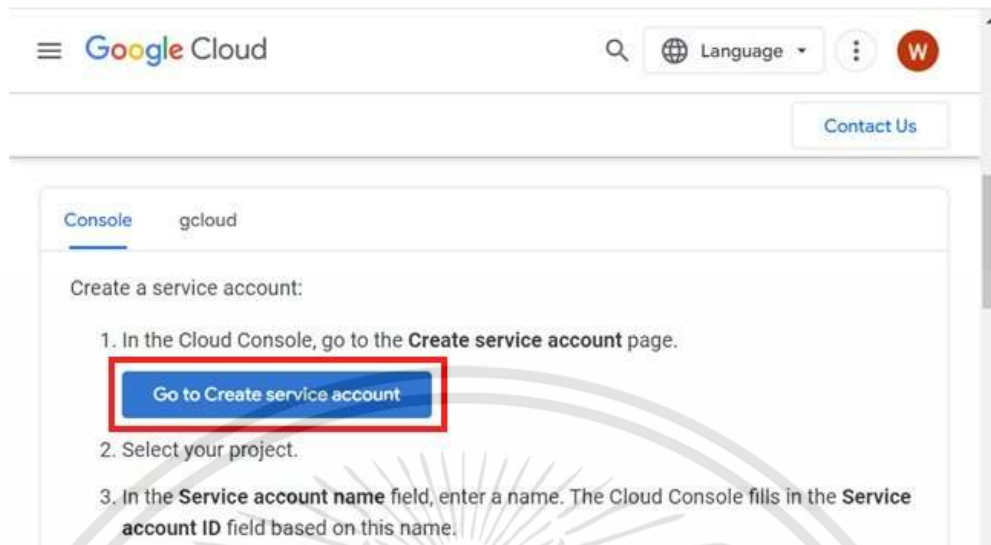


```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/froze/Desktop/testpy.py =====
>>>
```

รูปที่ 3.13 การตรวจสอบการติดตั้งไลบรารีทั้งหมด

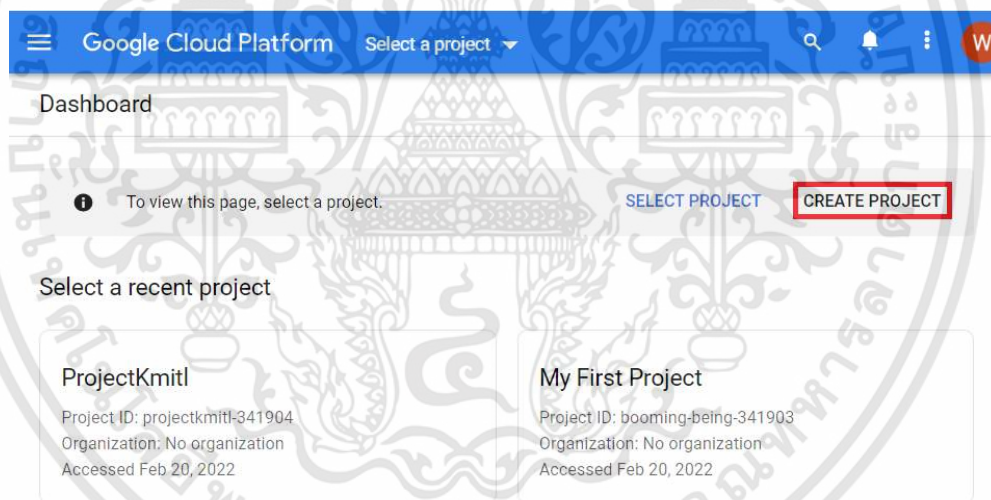
3.5 Google Cloud vision API

ทำการเปิดใช้บริการ Google cloud service โดยเข้าไปที่ URL <https://cloud.google.com/docs/authentication/getting-started> แล้วกด Go to Create service account โดยให้ใส่อีเมลล์ที่ต้องการสมัครใช้งาน

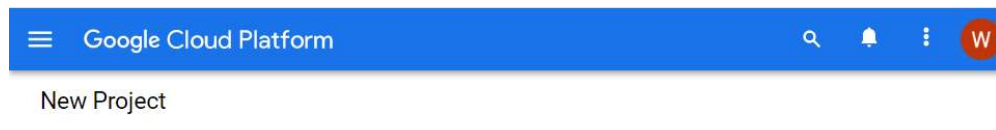


รูปที่ 3.14 เปิดใช้บริการ Google cloud service

จากนั้นทำการสร้างโปรเจกต์โดยกดที่ CREATE PROJECT และตั้งชื่อโปรเจกต์ของเรา

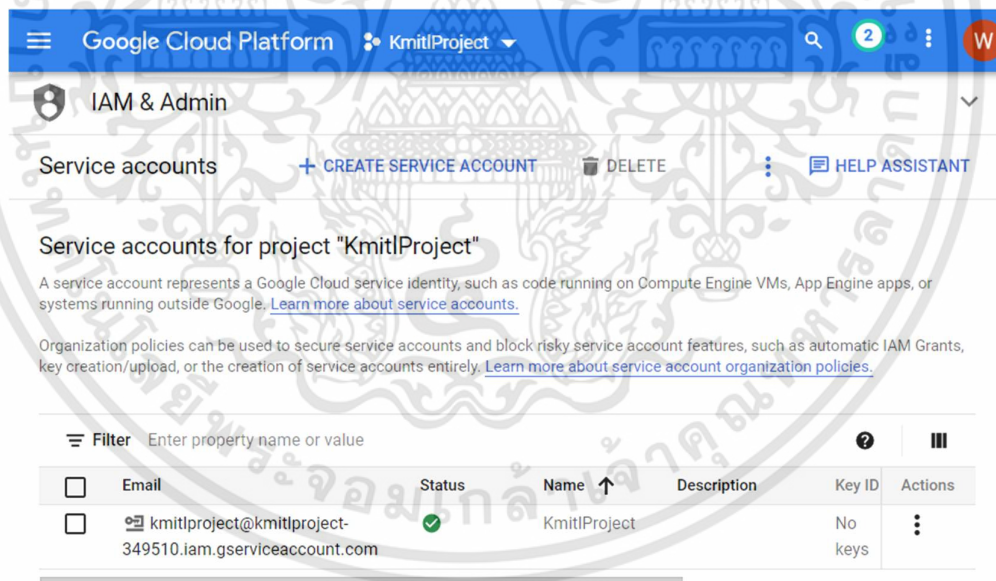


รูปที่ 3.15 เลือก CREATE PROJECT



รูปที่ 3.16 ทำการตั้งชื่อโปรเจกต์

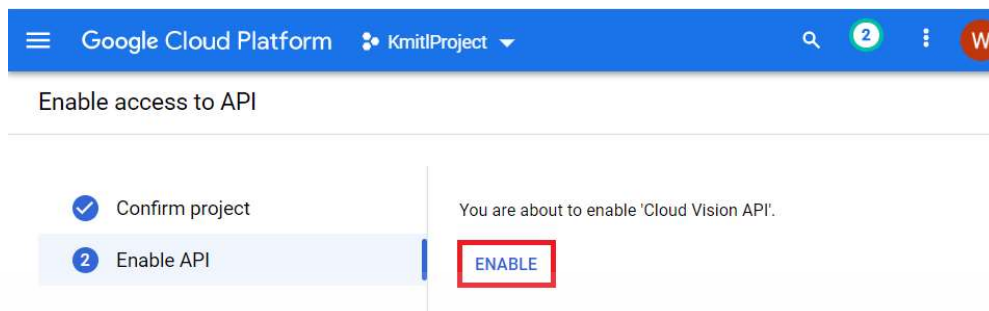
เมื่อทำการสร้างโปรเจกต์เสร็จแล้วจะได้หน้าต่าง ดังรูปที่ 3.17



รูปที่ 3.17 หน้าต่างสร้างโปรเจกต์เสร็จเรียบร้อยแล้ว

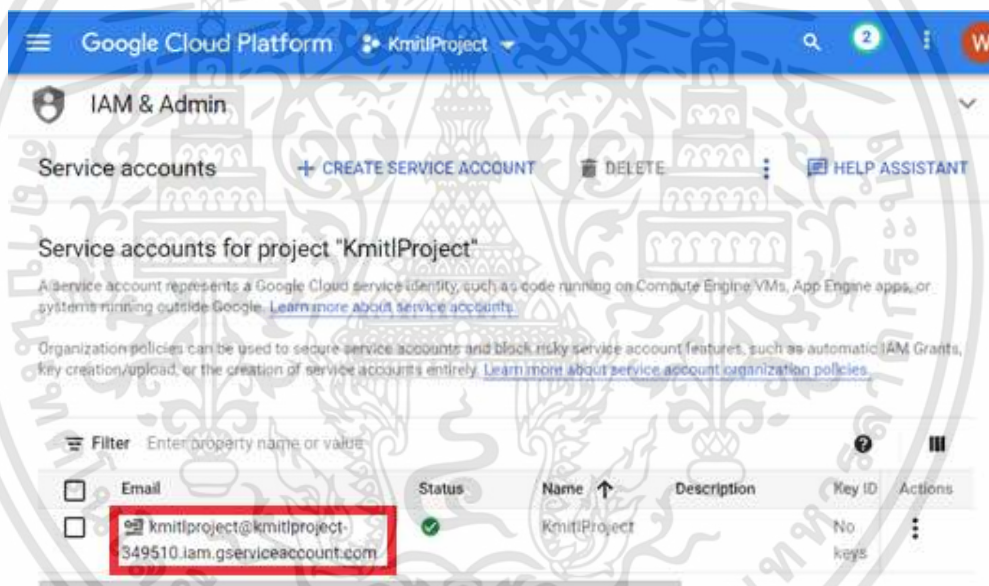
โดยในตอนแรกเราจะยังไม่สามารถเปิดใช้บริการ Google cloud vision API ได้จึงต้องทำการเปิดใช้บริการก่อน โดยไปที่ <https://console.cloud.google.com/apis> แล้วเลือกที่ ทำการเลือก ENABLE API ดังรูปที่ 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



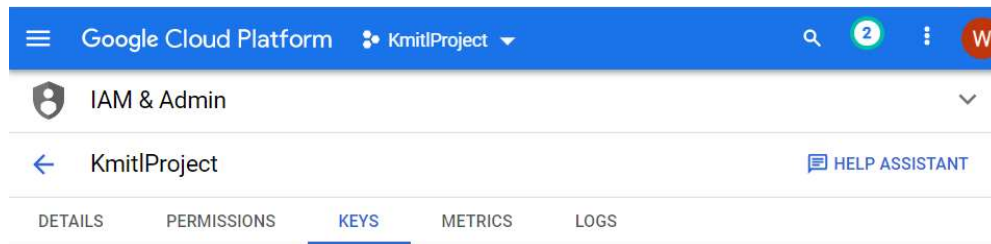
รูปที่ 3.18 เปิดใช้บริการ Google cloud vision API

จากนั้นเข้าไปที่ <https://console.cloud.google.com/iam-admin/serviceaccounts/> เพื่อทำการขอ Service account key ไว้เชื่อมต่อกับ Google cloud vision API เข้ากับโปรแกรมของเรา



รูปที่ 3.19 เลือก Google service account

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Keys

Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the [Workload Identity Federation](#). You can learn more about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).
[Learn more about setting organization policies for service accounts](#)

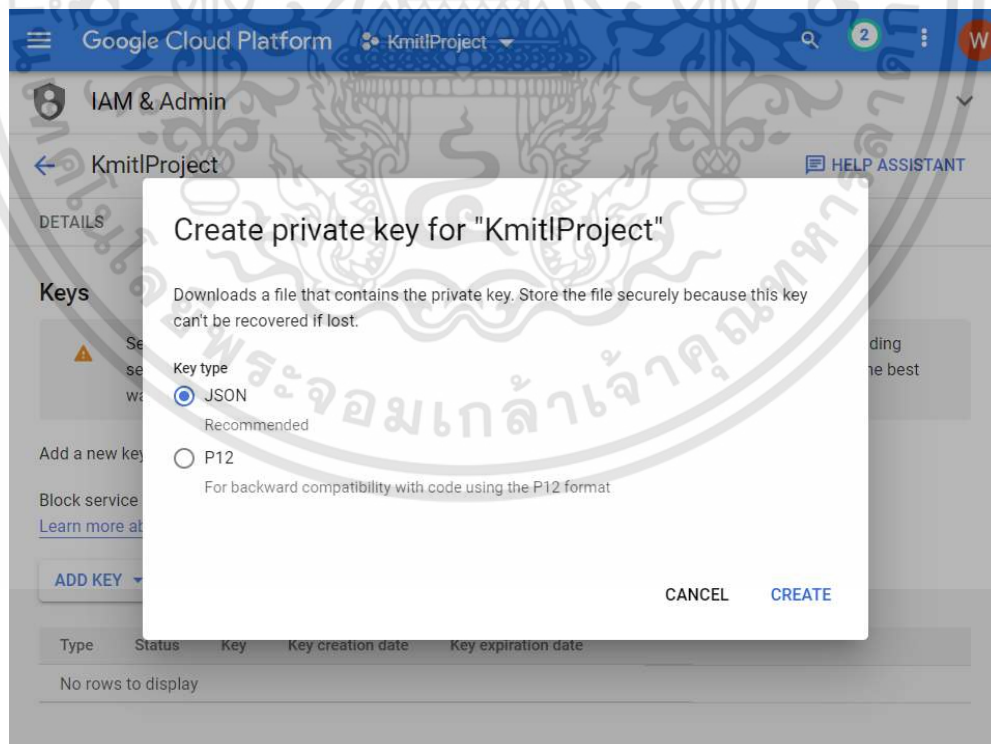
ADD KEY

| Type | Status | Key | Key creation date | Key expiration date |
|------|--------|-----|-------------------|---------------------|
|------|--------|-----|-------------------|---------------------|

No rows to display

รูปที่ 3.20 ทำการขอ Service account key

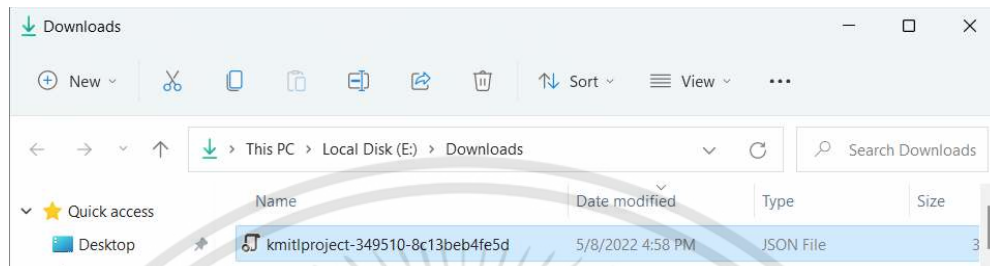
ให้เลือกประเภทเป็น JSON ดังรูปที่ 3.21



รูปที่ 3.21 เลือกประเภทเป็น JSON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการขอคีย์สำเร็จจะได้รับไฟล์ ดังรูปที่ 3.22 ไฟล์นี้ต้องเก็บไว้เพื่อเรียกใช้บริการ Google cloud vision API

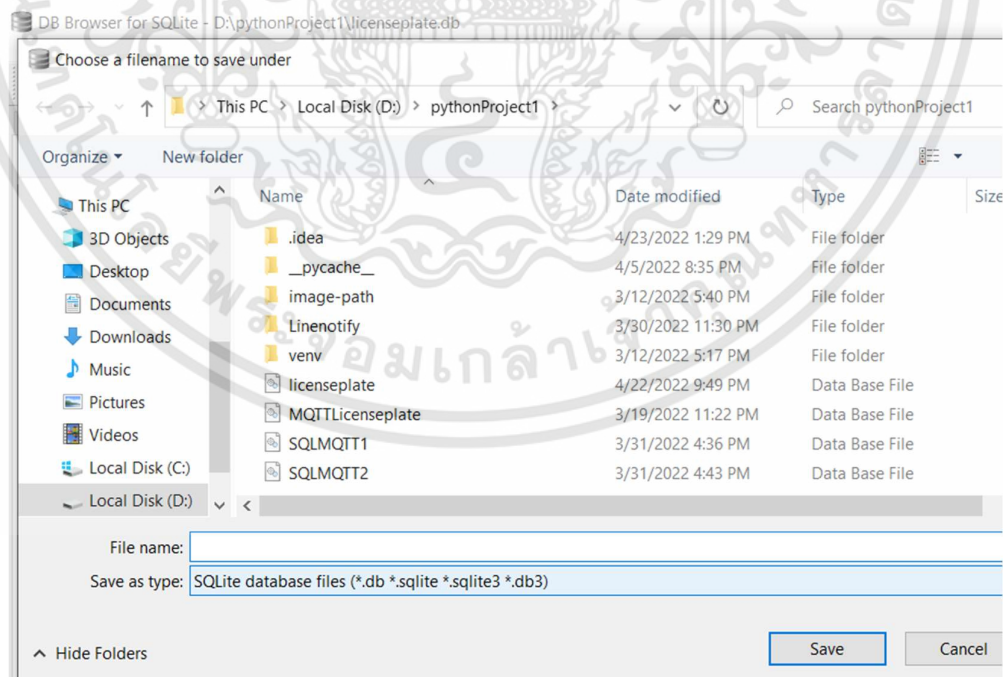


รูปที่ 3.22 ไฟล์ Service account key

3.6 DB Browser for SQLite

SQLite เป็นโปรแกรมฐานข้อมูลที่มีขนาดเล็กไม่ถึง 1 เมกะไบต์ เก็บฐานข้อมูลเป็นไฟล์โดยไม่จำเป็นต้องมีเซิร์ฟเวอร์ ทำให้ถูกใช้ในหลายๆ โปรแกรม หรือ ถูกติดตั้งลงในอุปกรณ์พกพาหลายๆชนิด เช่น ไอโฟน, แอนดรอยด์ เพื่อใช้ในการเก็บข้อมูล

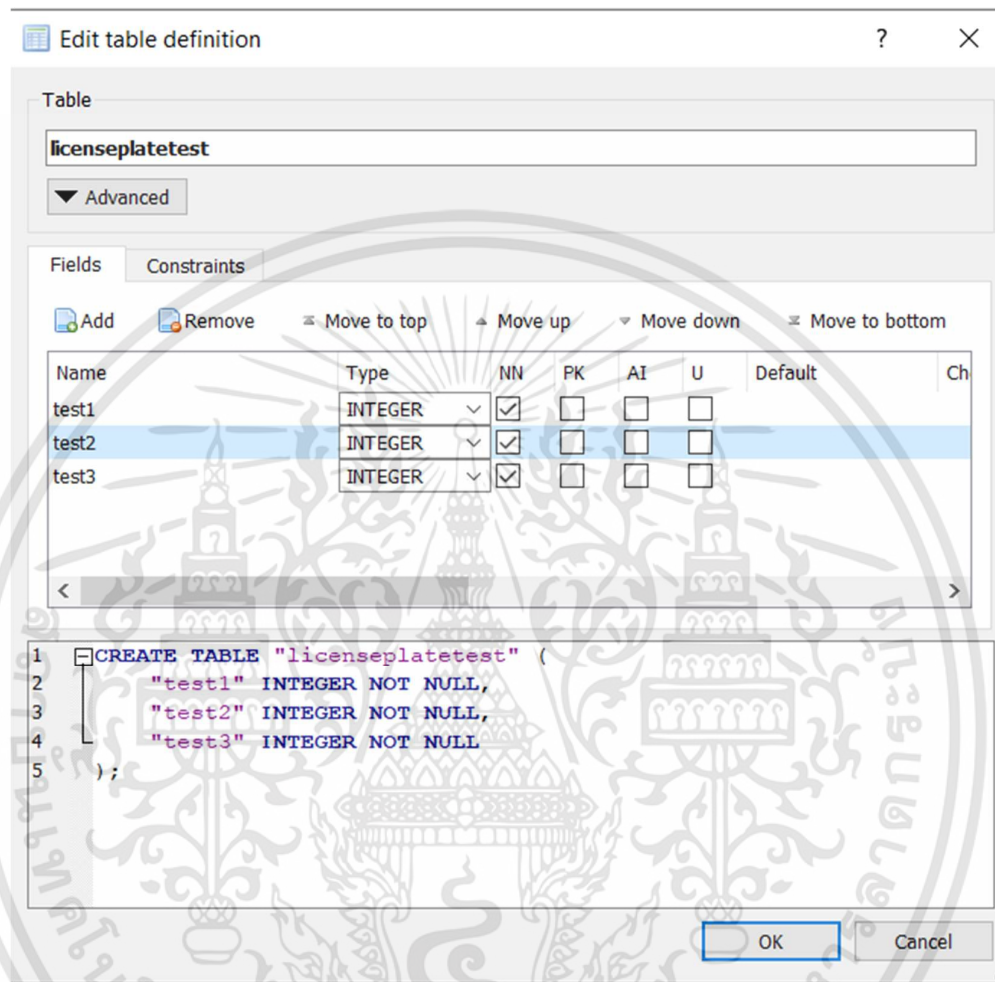
3.6.1 สร้างฐานข้อมูลใน SQLite



รูปที่ 3.23 สร้างไฟล์ฐานข้อมูล (Database)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างไฟล์ไว้ในโฟลเดอร์ Python ที่ได้ทำการเขียนโค้ดเพื่อใช้ในการเก็บข้อมูลป้ายทะเบียน

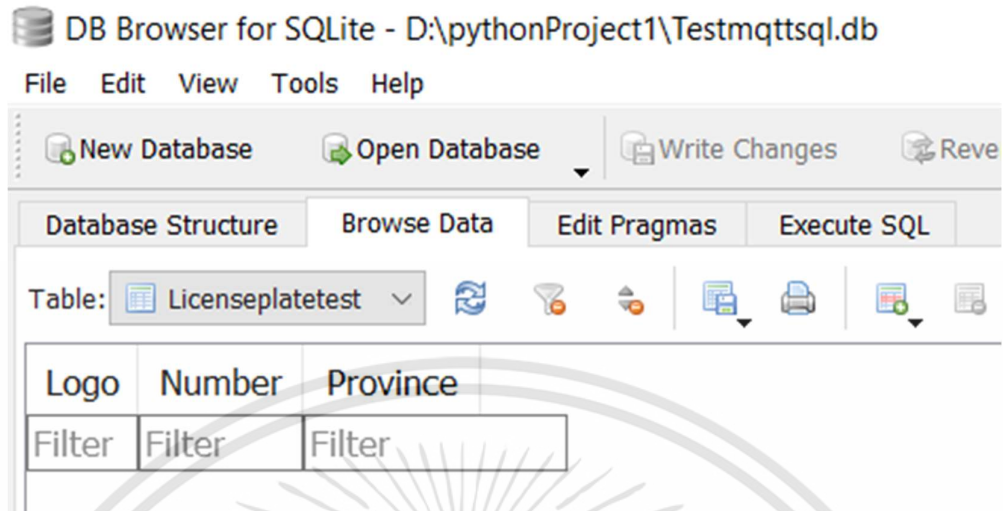


รูปที่ 3.24 Table definition

หลังจากตั้งชื่อไฟล์เสร็จ จะได้ตารางฐานข้อมูลมา ตั้งชื่อตารางสำหรับการเก็บค่าจากนั้นกด Add field เพื่อเอาค่าที่ได้จาก ยี่ห้อ และ ป้ายทะเบียนมาใช้ เลือกชนิดข้อมูลเลขจำนวนเต็มและเลือก NN (Not Null) สำหรับเก็บข้อมูลของป้ายทะเบียน ในที่นี้เราจะเก็บทั้งหมด 3 ค่า ได้แก่

1. ยี่ห้อ
2. เลขทะเบียน
3. จังหวัด

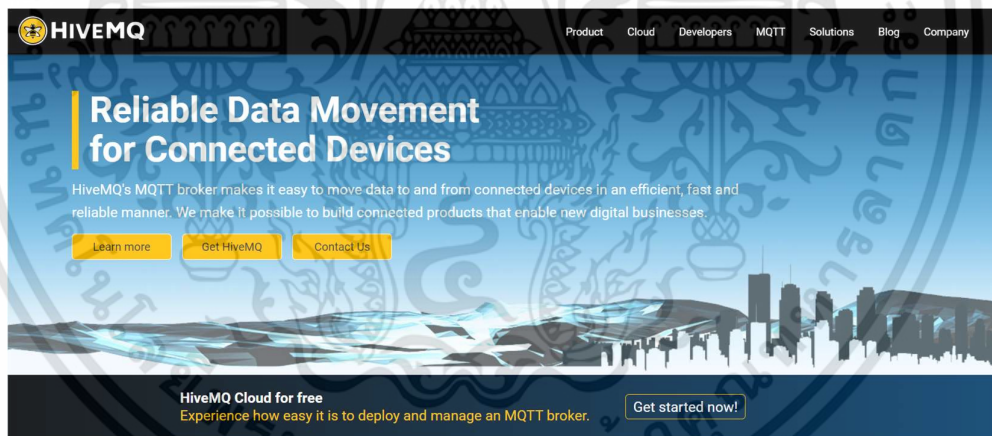
เสร็จเรียบร้อยแล้วจึงกด OK



รูปที่ 3.25 Browser data สำหรับเก็บข้อมูล

จะได้ฐานข้อมูลสำหรับเก็บข้อมูลขึ้นมาไฟล์ที่เราใช้ชื่อ Testmqttsql.db

3.7 MQTT (Message Queue Telemetry Transport)

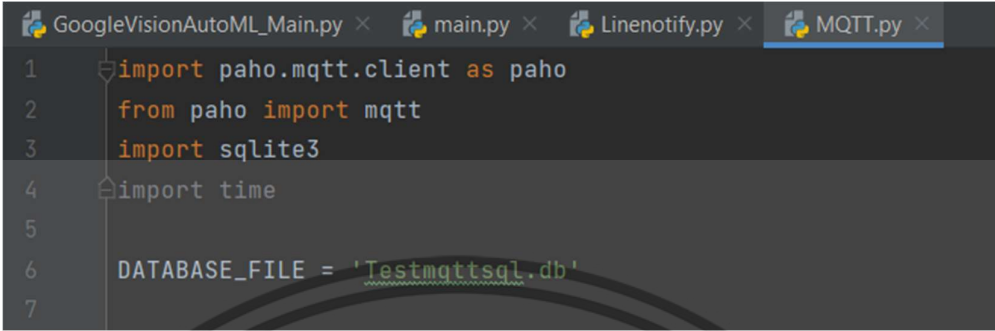


รูปที่ 3.26 เว็บไซต์ HIVEMQ

ในการสื่อสาร การส่งข้อมูล เลือกใช้โปรโตคอล MQTT เพราะ โนโพรเจกนี้ต้องการที่จะ รับ/ส่ง ข้อมูลที่ต้องอาศัยความเร็ว สามารถรับข้อมูลจากระยะไกลได้แบบเรียลไทม์ เพราะ MQTT ส่งข้อมูลแบบ low bandwidth และสามารถรับ ส่งข้อมูลได้พร้อมจากหลายๆที่ ซึ่ง MQTT ที่ใช้เป็น HIVEMQ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.1 เขียนคำสั่งในการส่งข้อมูลขึ้น



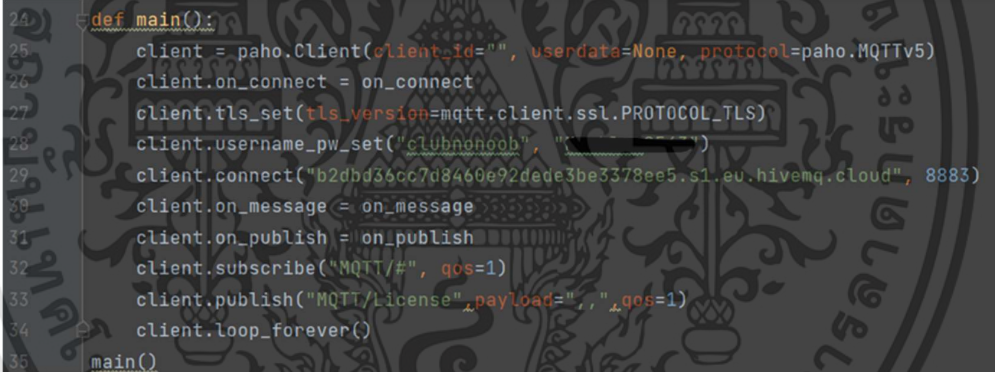
```

1 import paho.mqtt.client as paho
2 from paho import mqtt
3 import sqlite3
4 import time
5
6 DATABASE_FILE = 'Testmqttsql.db'
7

```

รูปที่ 3.27 โไลบรารีในการส่งข้อมูลจาก MQTT ขึ้น SQLite

ทำการ Import Library paho.mqtt และ import sqlite3 เพื่อที่จะเขียนคำสั่งส่งข้อมูล MQTT ไปเก็บใน SQL DATABASE_FILE = 'Testmqttsql.db' ส่วนฐานข้อมูล (database) ใช้เป็นชื่อไฟล์ตามที่ตั้งไว้ใน SQLite



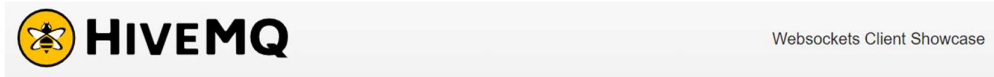
```

24 def main():
25     client = paho.Client(client_id="", userdata=None, protocol=paho.MQTTv5)
26     client.on_connect = on_connect
27     client.tls_set(tls_version=mqtt.client.ssl.PROTOCOL_TLS)
28     client.username_pw_set("clubkookob", " ")
29     client.connect("b2dbb36c67d8460e92dede3be3378ee5.s1.eu.hivemq.cloud", 8883)
30     client.on_message = on_message
31     client.on_publish = on_publish
32     client.subscribe("MQTT/#", qos=1)
33     client.publish("MQTT/license", payload="", qos=1)
34     client.loop_forever()
35     main()

```

รูปที่ 3.28 คำสั่งในการส่งข้อมูลจาก MQTT ขึ้น SQLite

ในส่วนโค้ดสิ่งที่ต้อง แก่ไขตรง Username, Password ในบรรทัดที่ 28 และ Host, Port ในบรรทัดที่ 29 ให้ตรงกัน และเริ่มทำงานโค้ดจึงจะสามารถเชื่อมต่อได้



Connection

Host: b2dbd36cc7d8460e92dede3be3378ee5.s1.eu.hivemq.c Port: 8884 ClientID: clientId-ySMVvYGTY9 Connect

Username: clubnonoob Password: Keep Alive: 60 SSL: Clean Session:

Last-Will Topic: Last-Will QoS: 0 Last-Will Retain:

Last-Will Message:

Publish

Subscriptions

Messages

รูปที่ 3.29 MQTT WebSocket Client



Connection

connected

Publish

Subscriptions

Topic: MQTT/License QoS: 0 Retain: Publish

Message: Toyota,7กย 3603,กรุงเทพมหานคร

Add New Topic Subscription

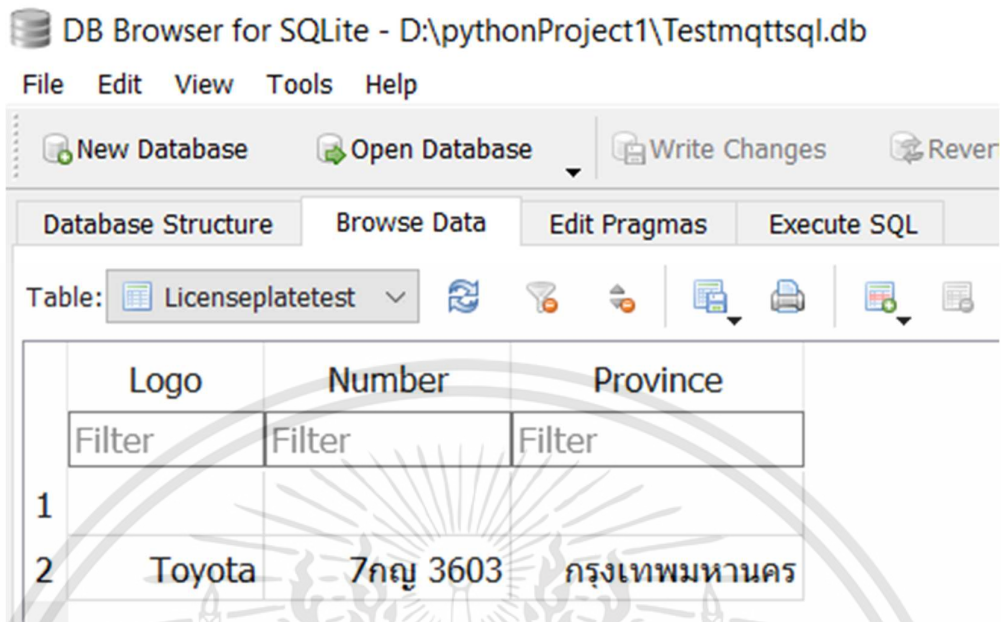
Qos: 2 MQTT/#

Messages

2022-04-29 13:45:09 Topic: MQTT/License Qos: 0
Toyota,7กย 3603,กรุงเทพมหานคร

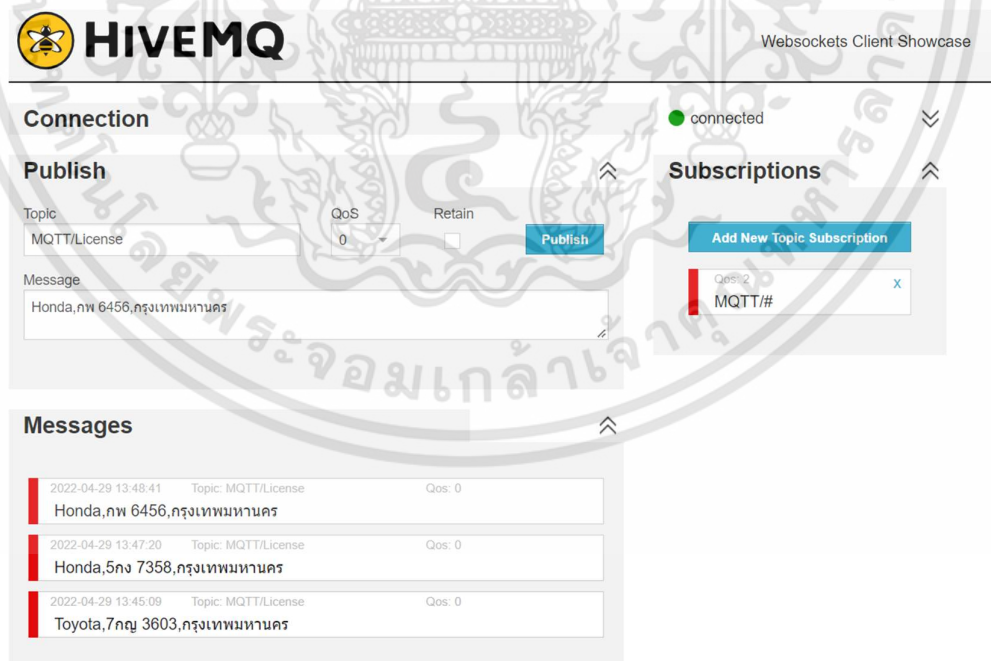
รูปที่ 3.30 ทดลองส่งค่าบน Web MQTT (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



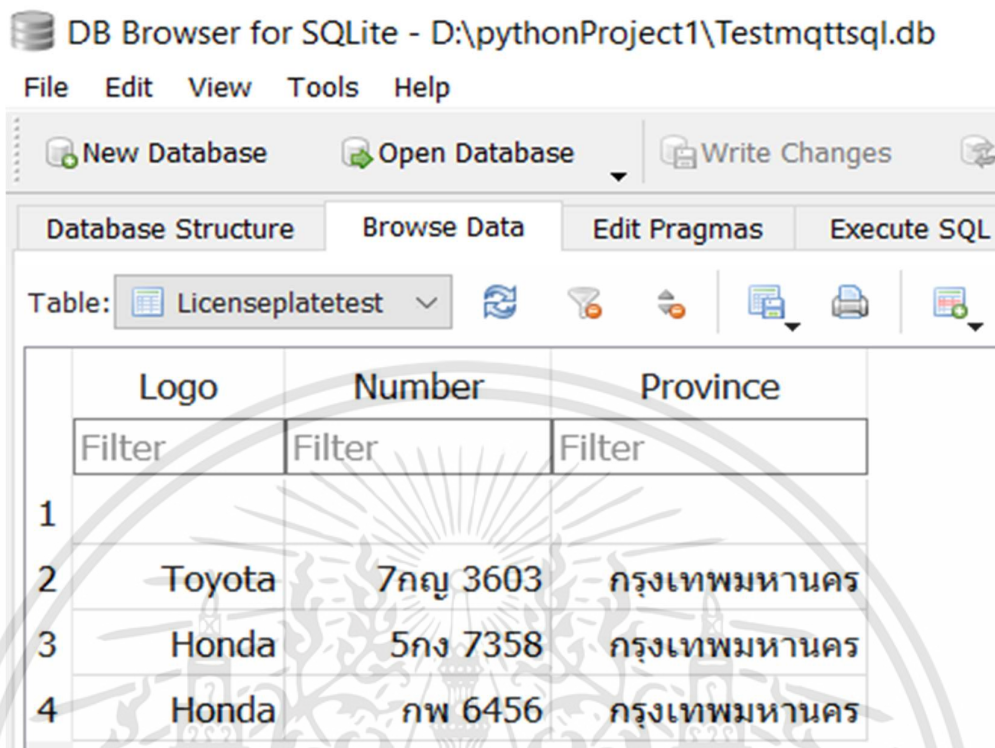
รูปที่ 3.31 เก็บค่าไว้บน DB browser (1)

กำหนดหัวข้อ Topic ให้ตรงกับโค้ดใน Python ใส่ให้ตรงกับโมเดล Publish/Subscribe หลังจากนั้นทดลองพิมพ์ ยี่ห้อ, รถยนต์, ป้ายทะเบียน, จังหวัด ที่ต้องการทดลอง แล้วกด Publish ค่าจะขึ้นไปใน Browse Data ของ 'Testmqttsql'



รูปที่ 3.32 ทดลองส่งค่าบน Web MQTT (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.33 เก็บค่าไว้บน DB browser (2)

หลังจากได้ค่าแล้วก็จะนำค่าไปทดลองในขั้นตอนการจัดเรียงข้อมูล

3.8 การเรียกใช้ฐานข้อมูลโดยใช้ไลบรารี Sqlite3 และจัดเรียงข้อมูลโดยใช้ Pandas

ในส่วนของการกำหนดฟังก์ชัน getData คือการเรียกใช้งานฐานข้อมูลผ่านไลบรารี sqlite3 โดยใช้คำสั่งในการเชื่อมต่อฐานข้อมูล sqlite3.connect("ชื่อไฟล์ฐานข้อมูล.db") ดังบรรทัดที่ 6

```

1 import sqlite3
2 import pandas as pd
3
4 def getData():
5
6     con = sqlite3.connect("Testmqttsql.db")
7     df = pd.read_sql_query("SELECT * from Licenseplatetest", con)
8
9     con.close()
10    return df.to_dict(orient="records")
11
12 a = getData()
13 print(a)

```

รูปที่ 3.34 โค้ดการจัดเรียงข้อมูลโดยใช้ Pandas

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ไลบรารี Pandas ในการจัดการข้อมูลโดยทำการ import pandas ตั้งชื่อว่า pd เพื่อสะดวกต่อการใช้งาน นำตารางฐานข้อมูลของไฟล์ SQL มาใส่ในบรรทัดที่ 7 จะได้ผลลัพธ์มาดังรูปที่ 3.35

```
Run: MQTT x main x
D:\pythonProject1\venv\Scripts\python.exe D:\pythonProject1/main.py
[{'Logo': '', 'Number': '', 'Province': ''}, {'Logo': 'Toyota', 'Number': '7กย 3663', 'Province': 'กรุงเทพมหานคร'}]
Process finished with exit code 0
```

รูปที่ 3.35 ผลลัพธ์การจัดเรียงข้อมูล

3.9 การตรวจจับป้ายทะเบียน

```
31
32 #os.system("start cmd /k python MQTT.py")
33 sqldata = getData()
34
35 # Setup google authn client key
36 os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'projectkmitl-api.json'
37
38 # Source path content all images
39 SOURCE_PATH = "image-path/"
40 path = SOURCE_PATH + 'output.jpg'
41
42 print('----- Start recognize license palate -----')
43
44 DATABASE_FILE = 'Testmqttsql.db'
45
46 cap = cv2.VideoCapture(0)
47 while (True):
```

รูปที่ 3.36 โค้ดการดึงข้อมูลจาก Main.py

ส่วนของการตรวจจับป้ายทะเบียน ทำการดึง 'getData' มาจาก Main.py เพื่อนำข้อมูลมาใช้ในการเทียบ database กับ mydata ตั้งชื่อว่า 'sqldata' และ ส่วนที่ใช้ตรวจจับป้ายทะเบียนทั้งหมด ตั้งชื่อว่า 'mydata'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

70 data = {}
71 for logo in logos:
72     #print(logo.description)
73     data = {'Logo': logo.description}
74     #print(data)
75     cv2.putText(img, logo.description, (200, 200), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 3)
76     cv2.imshow('Recognize & Draw', img)
77     i+=1
78     for text in texts:
79         if i==0:
80             i=i+1
81             license_plate = text.description
82             #print(license_plate)
83             vertices = [(vertex.x, vertex.y)
84                         for vertex in text.bounding_poly.vertices]
85             cv2.rectangle(img, (vertices[0][0]-10, vertices[0][1]-10), (vertices[2][0]+10, vertices[2][1]+10), (0, 255, 0), 3)
86             cv2.imshow('Recognize & Draw2', img)
87             #df = pd.DataFrame([x.split(';') for x in license_plate.split('\n')])
88             resultArray = [x.split(';') for x in license_plate.split('\n')]
89             if 'Logo' in data:
90                 logo = data['Logo']
91                 data = {'Logo': logo, 'Number': resultArray[0], 'Province': resultArray[1]}
92             df = pd.DataFrame(data)
93             try:

```

รูปที่ 3.37 โค้ดการสร้างเงื่อนไขตรวจจับยี่ห้อและตัวอักษร

โดยส่วนแรกที่โปรแกรมจะทำงานก็คือ การตรวจจับหายี่ห้อ ข้อเสียของโปรแกรมคือ ถ้าตรวจจับไม่เจอยี่ห้อ โค้ดจะไม่ทำงานในขั้นตอนต่อไป หมายความว่าทุกครั้งที่ทำกาเปิดใช้งานโปรแกรม ต้องเห็น ยี่ห้อทุกครั้ง ปัญหานี้จะเป็นเฉพาะตอนเริ่มเปิดใช้งานโปรแกรมเท่านั้น หลังจากนั้นจะทำการสร้างกรอบเพื่อตรวจจับหาข้อความ

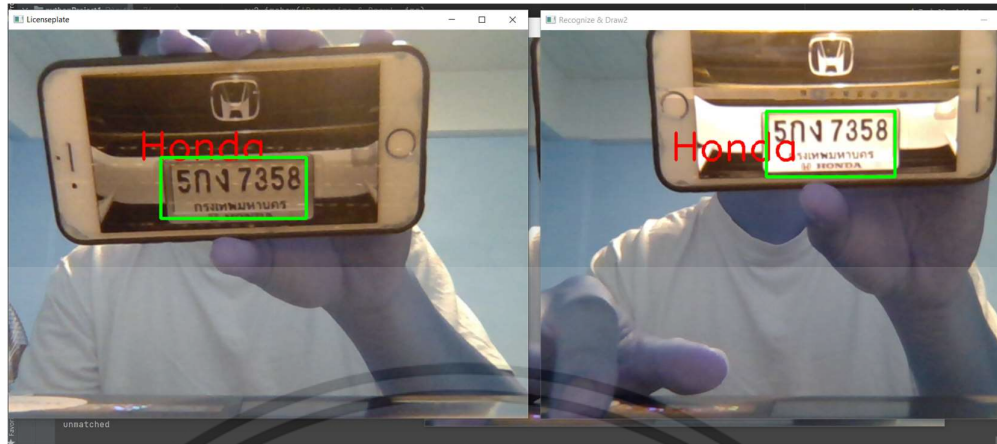
```

96     try:
97         for item in sqldata:
98             print("database", item)
99             print("mydata", data)
100             if item['Logo'] == logo and item['Number'] == resultArray[0][0] and item['Province'] == resultArray[1][0]:
101                 print("matched!")
102                 msg = '---\n' + logo + '\n' + str(resultArray[0][0]) + '\n' + str(resultArray[1][0]) + "\n(คลิกขวาคลิกขวาคลิกขวา)"
103                 line.lineNotify(msg)
104
105                 namefile = "licenseplate.png"
106                 cv2.imwrite(namefile, img)
107                 cv2.imshow("licenseplate", img)
108
109                 message = "Picture license plate" # Set your message here!
110                 line.notify(message)
111             else:
112                 print("unmatched")
113
114         except:
115             pass
116             #df.reset_index(level=0, inplace=True)
117             df.to_csv("license.csv")
118
119         cv2.waitKey(1)
120
121         if cv2.waitKey(1) & 0xFF == ord('q'):
122             break
123     cap.release()
124     cv2.destroyAllWindows()

```

รูปที่ 3.38 โค้ดการเปรียบเทียบค่าในฐานข้อมูล (database) กับ mydata

พอทำการตรวจจับ ยี่ห้อ, ป้ายทะเบียน, จังหวัด เรียบร้อย ก็จะมาเข้า for loop ใช้ร่วมกับ 'sqldata' โปรแกรมจะดึงฐานข้อมูล ที่มีข้อมูลเก็บไว้มาใช้งาน และ เปรียบเทียบกับ 'mydata' ที่ตรวจจับผ่านกล้องเว็บแคมถ้าค่าของ 'mydata' ตรงกับฐานข้อมูล (database) จะให้ทำการส่งข้อมูล และภาพขึ้นไปบนไลน์ ในขณะที่ข้อมูลที่ตรวจจับได้ตรงกันพอดี เป็นอันเสร็จสิ้น



รูปที่ 3.39 ผลลัพธ์เมื่อข้อมูลตรงกัน (1)



รูปที่ 3.40 ผลลัพธ์เมื่อข้อมูลตรงกัน (2)

ถ้าข้อมูลตรงกันจะสร้างภาพขึ้นมาดังรูปที่ 3.39 และรูปที่ 3.40 แล้วนำไปเก็บบนโฟลเดอร์ของโปรเจกต์ เพื่อที่จะทำการส่งภาพไปขึ้นบนไลน์

3.10 ส่งข้อความและรูปภาพเข้าไลน์ Notify

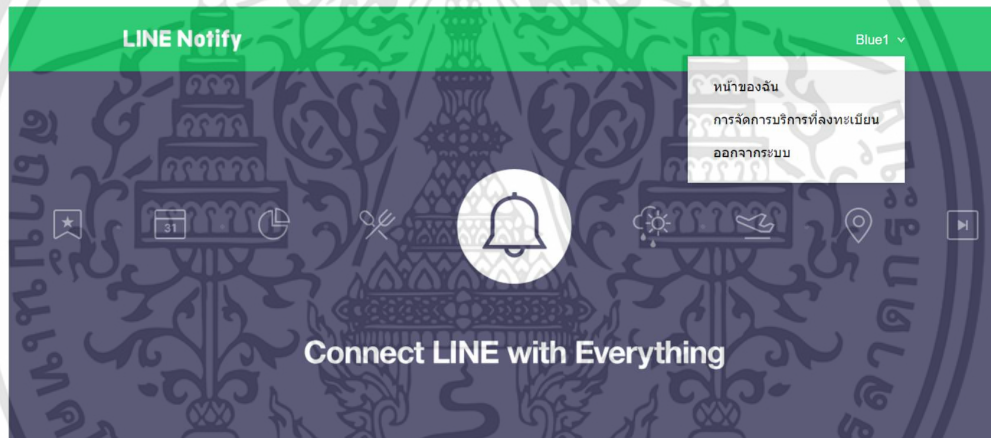
ทำการสร้างฟังก์ชัน def lineNotify เพื่อนิยามฟังก์ชันส่วนที่ใช้แจ้งเตือนไลน์ขึ้นมา

```

14 url = 'https://notify-api.line.me/api/notify'
15 token = '...'
16 headers = {'Authorization': 'Bearer ' + token}
17
18 def lineNotify(mid):
19     payload = {'message':mid}
20     return _lineNotify(payload)
21
22 def _lineNotify(payload,file=None):
23     return requests.post(url, headers=headers, data=payload, files=file)
24
25 def lineNotify(message):
26     img2 = {'imageFile': open('D:/pythonProject1/Licenseplate.png', 'rb')}
27     data2 = {'message': message}
28     session = requests.session()
29     session_post = session.post(url, headers=headers, files=img2, data=data2)
30     return (session_post.text)

```

รูปที่ 3.41 สร้างฟังก์ชัน def lineNotify

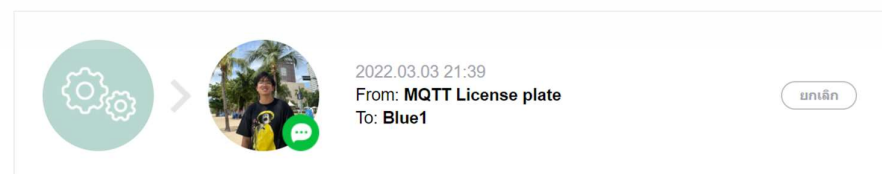


รูปที่ 3.42 เว็บไซต์ LINE Notify



บริการที่เชื่อมต่อ

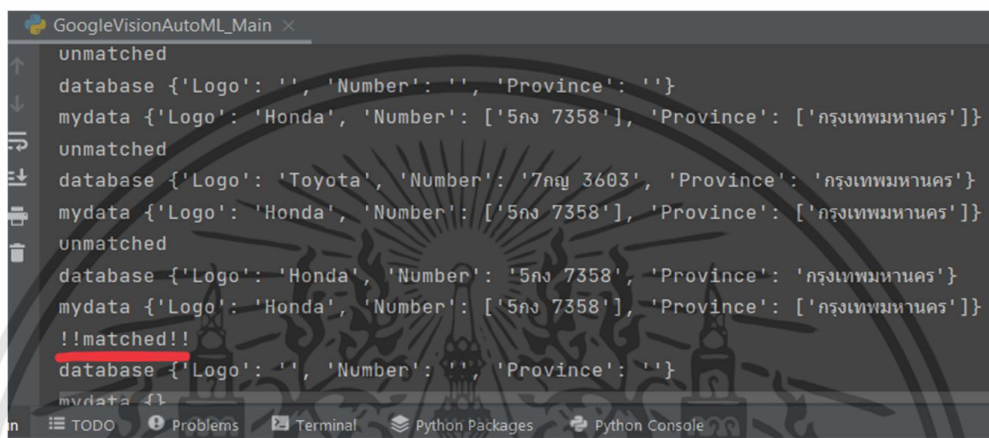
รายชื่อบริการที่เชื่อมต่ออยู่ในขณะนี้ หากต้องการยกเลิกการเชื่อมต่อ โปรดคลิกปุ่ม "ยกเลิก"



รูปที่ 3.43 รายชื่อบริการที่เชื่อมต่ออยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการส่งข้อความ และ ภาพเข้าไลน์ขั้นตอนแรกที่ต้องทำคือเริ่มจากการสร้าง Token ก่อนโดยเข้าไปที่ Line Notify ทำการเข้าสู่ระบบ ไปที่โปรไฟล์ กดที่หน้าของฉัน เลือกที่ออก Token ใส่ชื่อให้ Token สำหรับการแจ้งเตือนและเลือกที่ตัวเอง เสร็จสิ้นจะได้ Token มา คัดลอกส่วนนั้นแล้วนำมาใส่ในโค้ด ส่วน URL ก็ใช้ของ line ตามปกติ ในส่วน def lineNotify คือการสร้างฟังก์ชัน จะเป็นการดึงข้อความและการดึงภาพภาพ เพื่อส่งไปบนไลน์ที่เชื่อมต่อ Token ไว้



```

GoogleVisionAutoML_Main <
unmatched
database {'Logo': '', 'Number': '', 'Province': ''}
mydata {'Logo': 'Honda', 'Number': ['5กง 7358'], 'Province': ['กรุงเทพมหานคร']}
unmatched
database {'Logo': 'Toyota', 'Number': '7กย 3603', 'Province': 'กรุงเทพมหานคร'}
mydata {'Logo': 'Honda', 'Number': ['5กง 7358'], 'Province': ['กรุงเทพมหานคร']}
unmatched
database {'Logo': 'Honda', 'Number': '5กง 7358', 'Province': 'กรุงเทพมหานคร'}
mydata {'Logo': 'Honda', 'Number': ['5กง 7358'], 'Province': ['กรุงเทพมหานคร']}
!!matched!!
database {'Logo': '', 'Number': '', 'Province': ''}
mydata {}

```

รูปที่ 3.44 ผลลัพธ์จากการทำงานของโปรแกรม (1)



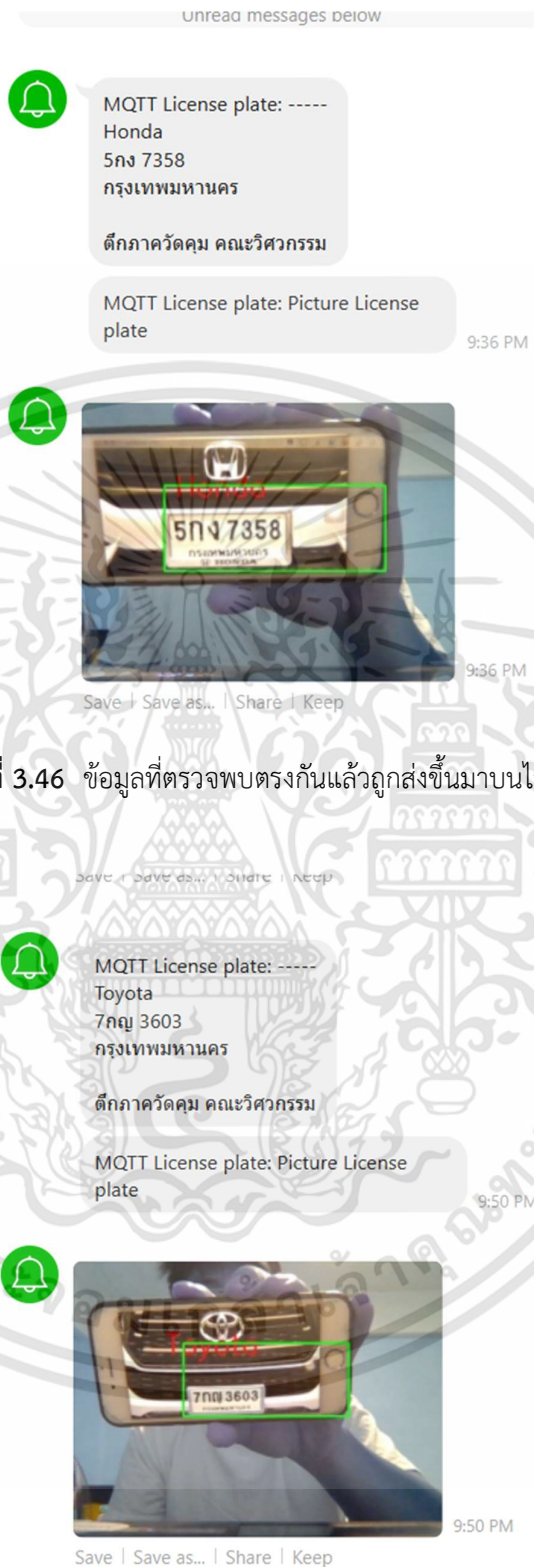
```

GoogleVisionAutoML_Main <
database {'Logo': 'Honda', 'Number': '5กง 7358', 'Province': 'กรุงเทพมหานคร'}
mydata {'Logo': 'Toyota', 'Number': ['73603'], 'Province': ['ภย']}
unmatched
database {'Logo': '', 'Number': '', 'Province': ''}
mydata {'Logo': 'Toyota', 'Number': ['7กย 3603'], 'Province': ['กรุงเทพมหานคร']}
unmatched
database {'Logo': 'Toyota', 'Number': '7กย 3603', 'Province': 'กรุงเทพมหานคร'}
mydata {'Logo': 'Toyota', 'Number': ['7กย 3603'], 'Province': ['กรุงเทพมหานคร']}
!!matched!!
database {'Logo': 'Honda', 'Number': '5กง 7358', 'Province': 'กรุงเทพมหานคร'}
mydata {'Logo': 'Toyota', 'Number': ['7กย 3603'], 'Province': ['กรุงเทพมหานคร']}
unmatched

```

รูปที่ 3.45 ผลลัพธ์จากการทำงานของโปรแกรม (2)

เมื่อโปรแกรมทำการประมวลผลภาพแล้ว ตรวจสอบว่ายี่ห้อรถยนต์และป้ายทะเบียนรถยนต์ตรงกับฐานข้อมูลที่ได้กำหนดไว้ ก็จะทำการบันทึกภาพและส่งข้อมูล ยี่ห้อรถยนต์ เลขทะเบียนรถยนต์ จังหวัด ตำแหน่งของรถยนต์ ณ ขณะนั้น และส่งภาพรถยนต์ ขึ้นแจ้งเตือนบนไลน์

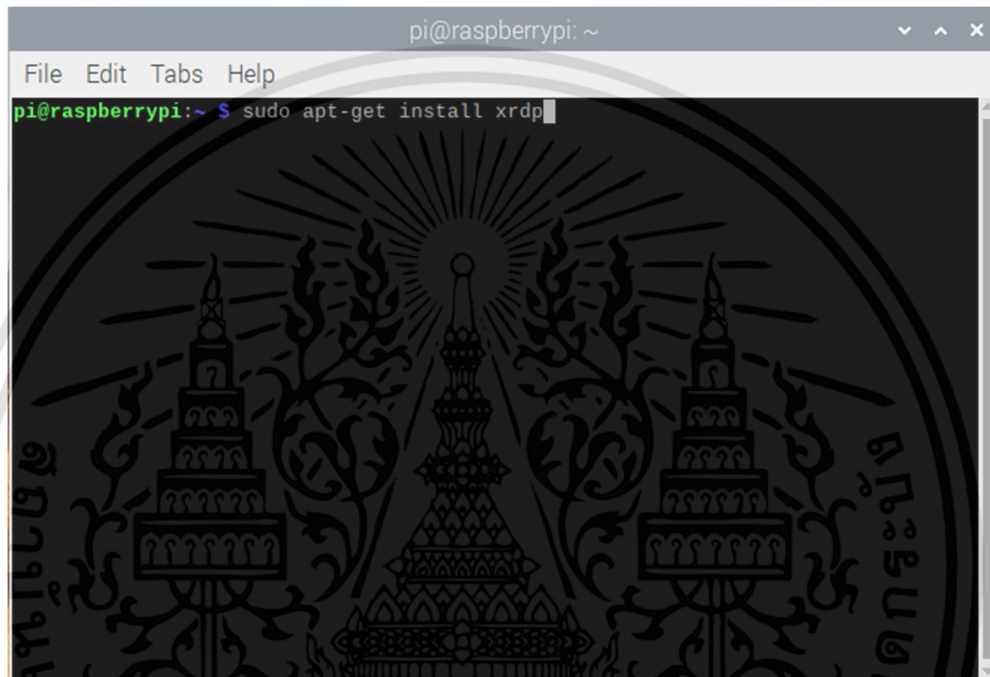


รูปที่ 3.47 ข้อมูลที่ตรวจพบตรงกันแล้วถูกส่งขึ้นมานบนไลน์ (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

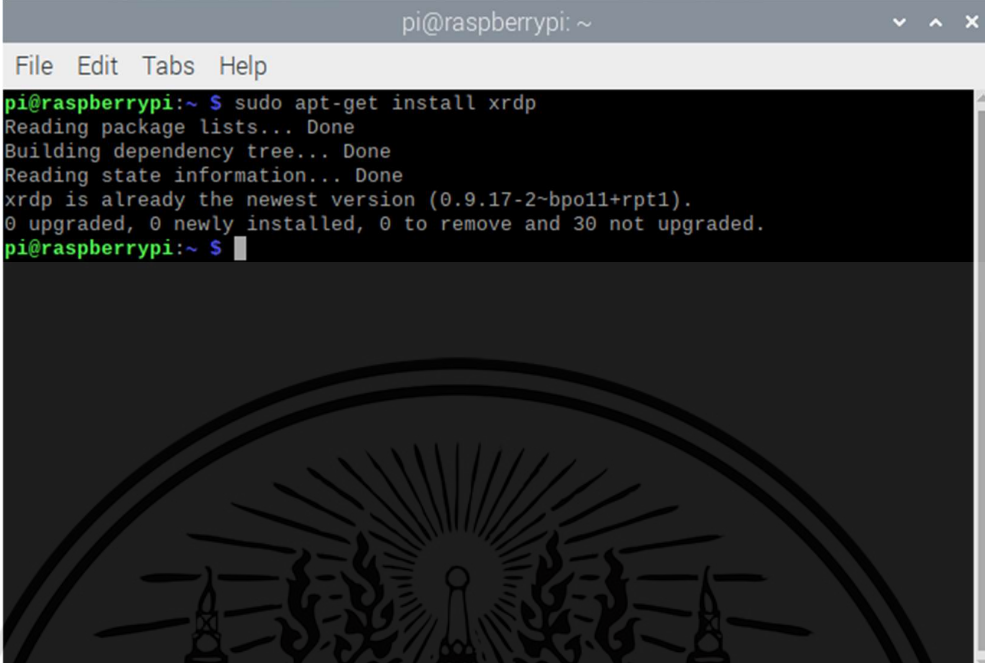
3.11 การใช้งาน Raspberry Pi แบบ Remote Terminal (SSH)

ทำการติดตั้ง RDP หรือ Remote Desktop Protocol คือ โพรโตคอลที่พัฒนาโดย Microsoft ที่ไว้สำหรับเชื่อมต่อกับคอมพิวเตอร์เครื่องอื่น ผ่านระบบเครือข่าย ซึ่งแสดงผลแบบกราฟฟิก หรือ GUI (Graphic User Interface) สามารถติดตั้งโดยใช้คำสั่ง `sudo apt-get install xrdp` ใน Terminal ดังรูปที่



รูปที่ 3.48 ใช้คำสั่ง `sudo apt-get install xrdp`

ทำการติดตั้ง xrdp



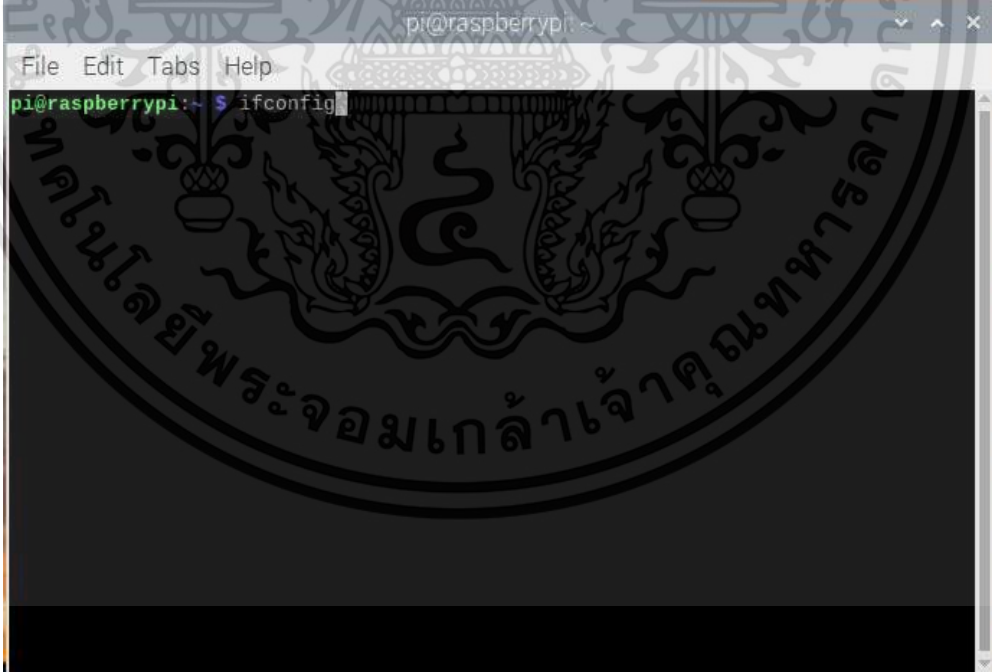
```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get install xrdp
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
xrdp is already the newest version (0.9.17-2~bpo11+rpt1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pi@raspberrypi:~ $

```

รูปที่ 3.49 เมื่อทำการติดตั้ง xrdp เสร็จเรียบร้อย

ดูไอพีแอดเดรสของ Raspberry Pi โดยใช้คำสั่ง ifconfig



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ ifconfig

```

รูปที่ 3.50 ใช้คำสั่ง ifconfig

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pi@raspberrypi: ~
File Edit Tabs Help
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 42 bytes 4777 (4.6 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 42 bytes 4777 (4.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

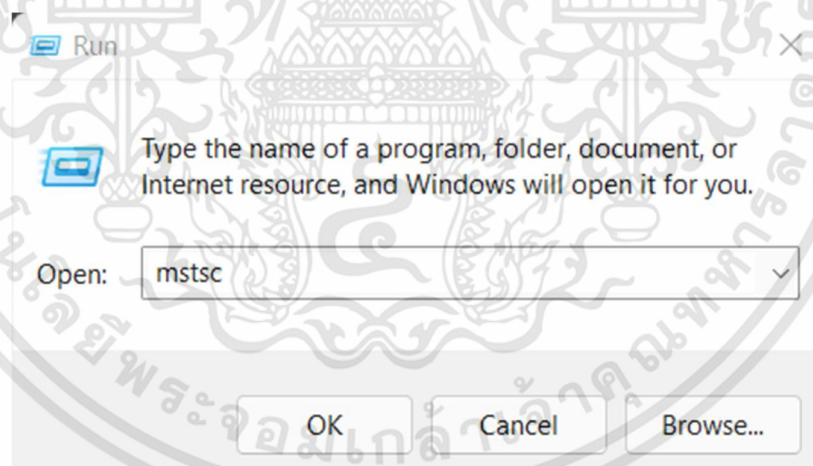
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 172.20.10.5 netmask 255.255.255.240 broadcast 172.20.10.15
inet6 2001:44c8:42c0:676a:a617:15cf:4ee:a54 prefixlen 64 scopeid 0x0<g
lobal>
inet6 fe80::dec4:862e:6234:89ee prefixlen 64 scopeid 0x20<link>
ether b8:27:eb:ac:a5:f5 txqueuelen 1000 (Ethernet)
RX packets 20099 bytes 1294747 (1.2 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 73067 bytes 99318923 (94.7 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$

```

รูปที่ 3.51 ไอพีแอดเดรสของ Raspberry Pi

เรียกใช้ Run command โดยการกดปุ่ม Window พร้อมกับปุ่ม R และพิมพ์ mstsc



รูปที่ 3.52 เรียกใช้ Run command และพิมพ์ mstsc

เมื่อกดตกลงจะแสดงหน้าต่างโปรแกรม Remote Desktop Connection ขึ้นมาให้เราใส่ไอพีแอดเดรสของ Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



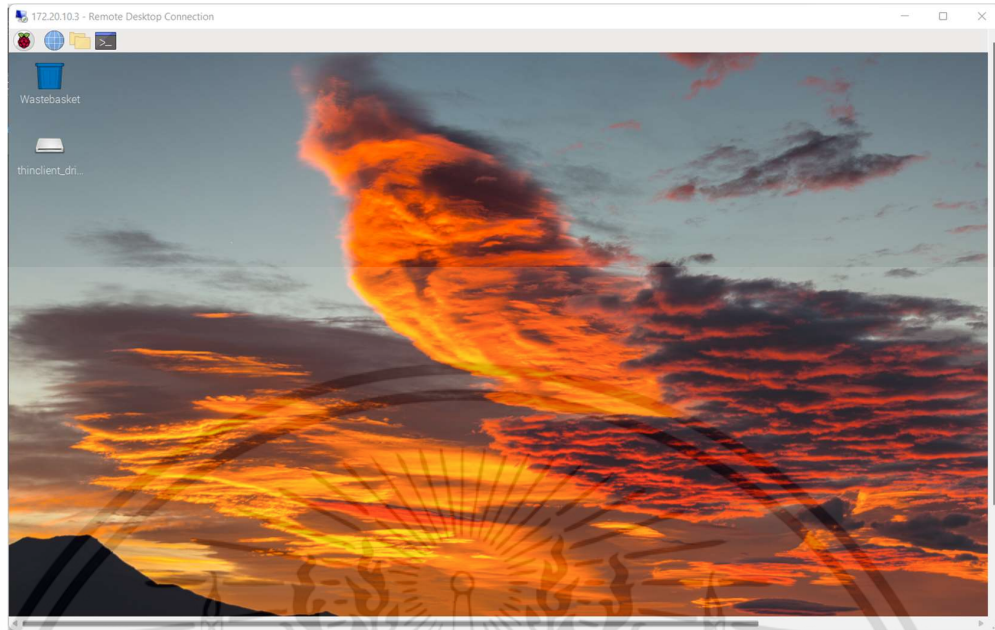
รูปที่ 3.53 โปรแกรม Remote Desktop Connection

เมื่อใส่ไอพีแอดเดรสและทำการกดเชื่อมต่อจะขึ้นหน้าต่างให้ทำการใส่ username: pi และ password: raspberry จากนั้นกดตกลงเพื่อเข้าสู่ Raspberry Pi OS



รูปที่ 3.54 หน้า Login ของ Remote Desktop Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.55 หน้า GUI ของ Raspberry Pi OS

3.12 ติดตั้ง Sqlite3 ลงบน Raspberry Pi

```

pi@raspberrypi:~$ sudo apt install sqlite3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
sqlite3 is already the newest version (3.34.1-3).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pi@raspberrypi:~$
  
```

รูปที่ 3.56 ติดตั้ง sqlite3 ใน Raspberry Pi

ใช้คำสั่ง `sudo apt install sqlite3` ลงบน Raspberry Pi Terminal เพื่อติดตั้ง sqlite3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

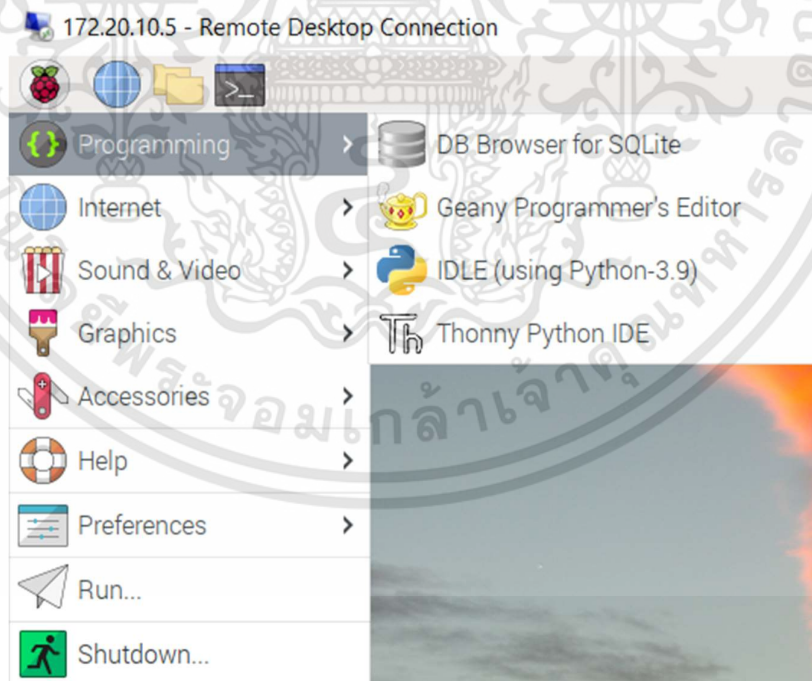
```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt install sqlite3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
sqlite3 is already the newest version (3.34.1-3).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pi@raspberrypi:~ $ sudo apt install sqlitebrowser
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
sqlitebrowser is already the newest version (3.12.1-2).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
pi@raspberrypi:~ $

```

รูปที่ 3.57 install sqlitebrowser ใน Raspberry Pi

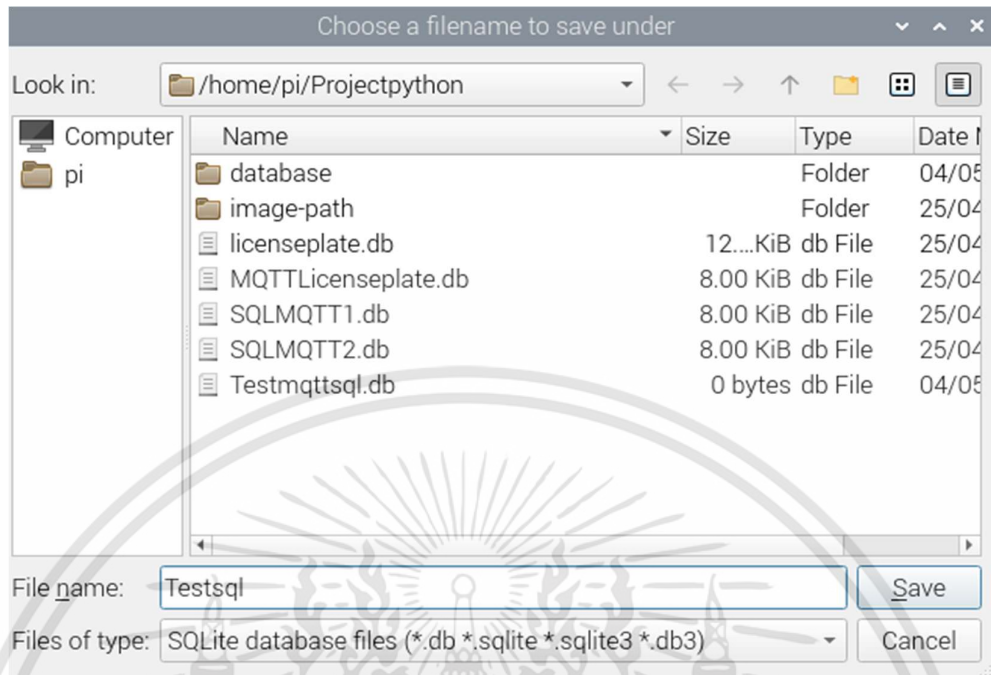
ใช้คำสั่ง `sudo apt install sqlitebrowser` เพื่อติดตั้งโปรแกรม DB Browser for SQLite



รูปที่ 3.58 ติดตั้ง DB Browser for SQLite

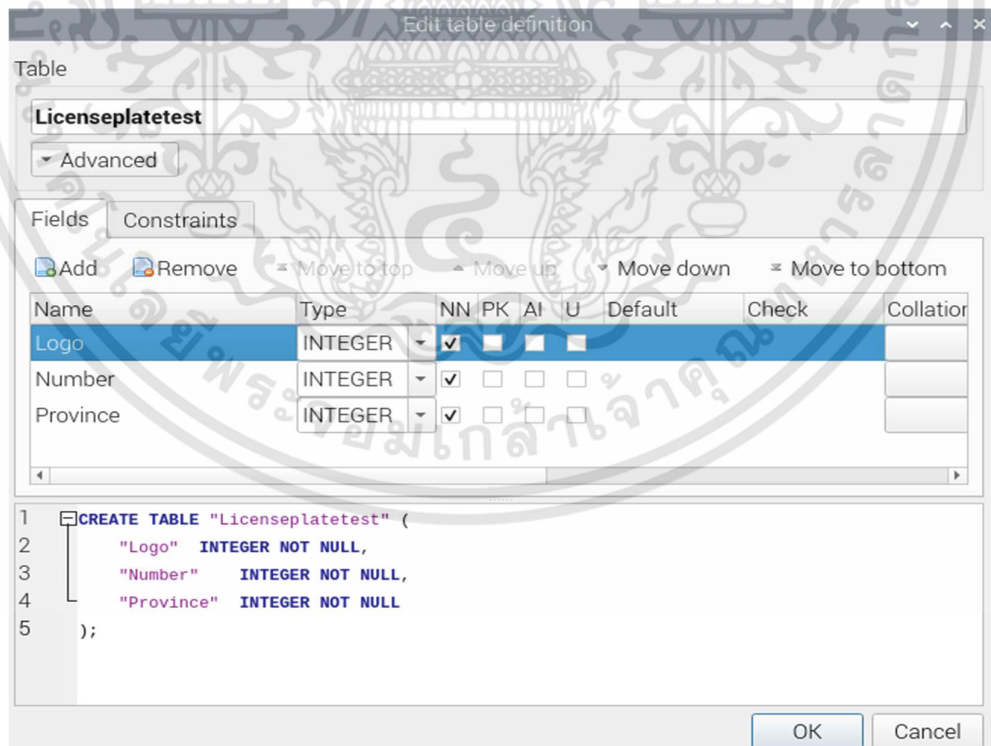
เมื่อติดตั้งเสร็จจะได้ DB Browser for SQLite

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.59 ติดตั้ง DB Browser for SQLite

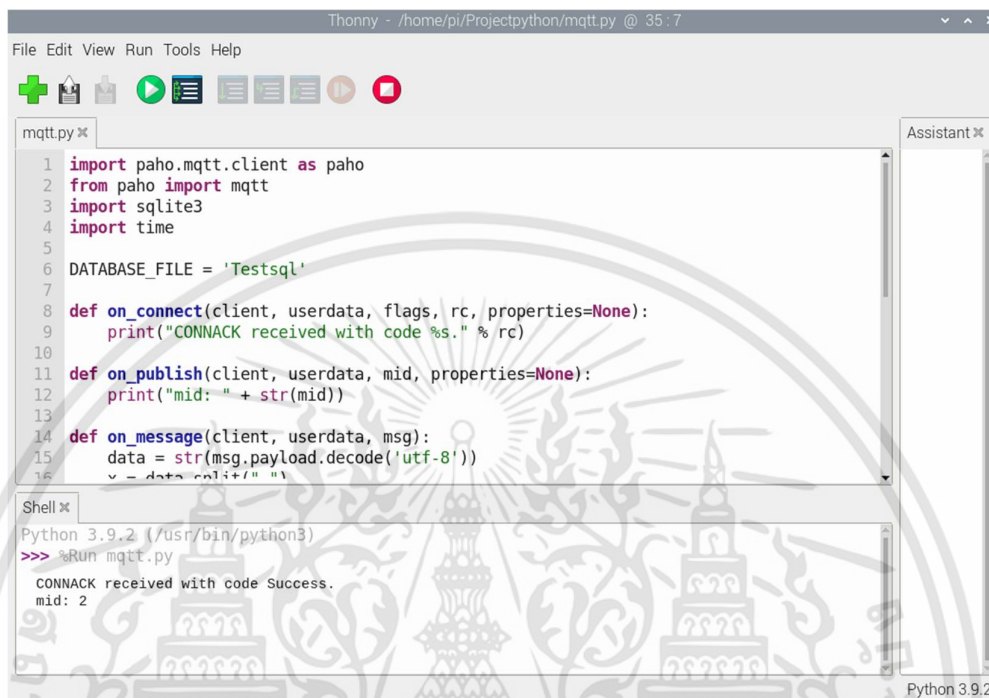
ตั้งชื่อไฟล์ใน DB browser for SQLite ตั้งชื่อว่า Testsql เพื่อไว้เก็บข้อมูลใน Raspberry Pi



รูปที่ 3.60 Create Table ใน Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตั้งชื่อตารางฐานข้อมูล (Table) สำหรับใช้ในการเก็บค่า จากนั้นกด Add field เพื่อเอาค่าที่ได้จากยี่ห้อและป้ายทะเบียนมาใช้ เสร็จเรียบร้อยแล้วจึงกด OK



```

Thonny - /home/pi/Projectpython/mqt.py @ 35 : 7
File Edit View Run Tools Help
mqt.py x Assistant x
1 import paho.mqtt.client as paho
2 from paho import mqtt
3 import sqlite3
4 import time
5
6 DATABASE_FILE = 'Testsql'
7
8 def on_connect(client, userdata, flags, rc, properties=None):
9     print("CONNACK received with code %s." % rc)
10
11 def on_publish(client, userdata, mid, properties=None):
12     print("mid: " + str(mid))
13
14 def on_message(client, userdata, msg):
15     data = str(msg.payload.decode('utf-8'))
16     v = data.split(" ")
17
Shell x
Python 3.9.2 (/usr/bin/python3)
>>> %Run mqt.py
CONNACK received with code Success.
mid: 2
Python 3.9.2

```

รูปที่ 3.61 โลโก้ในการส่งข้อมูลจาก MQTT ขึ้น SQLite

เริ่มการทำงานโค้ด MQTT ลงบน Raspberry Pi เพื่อที่จะส่งข้อมูลไปเก็บไว้ใน SQLite

Connection ● connected

Publish

Topic: MQTT/License QoS: 0 Retain:

Message: Honda, กพ 6456, นนทบุรี

Subscriptions

Add New Topic Subscription

Qos: 2 MQTT/#

Messages

- 2022-05-04 16:18:52 Topic: MQTT/License Qos: 0
Honda, กพ 6456, นนทบุรี
- 2022-05-04 16:17:22 Topic: MQTT/License Qos: 0
Toyota, 7กญ 3603, กรุงเทพมหานคร
- 2022-05-04 16:15:38 Topic: MQTT/License Qos: 0
Honda, 5กง 7358, กรุงเทพมหานคร

รูปที่ 3.62 ส่งค่าบน Web MQTT ขึ้นไปบน Raspberry Pi

ทดลองส่งค่าจาก MQTT ไปเก็บค่าไว้ใน DB browser for SQLite บน Raspberry Pi

DB Browser for SQLite - /home/pi/Projectpython/Testsql

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project

Database Structure Browse Data Edit Pragmas Execute SQL Edit Database Cell

Table: Licenseplattete Filter in a...

| Logo | Number | Province |
|--------|-----------------|------------------|
| Filter | Filter | Filter |
| 1 | | |
| 2 | Honda 5กง 7358 | กรุงเทพมหานคร... |
| 3 | Toyota 7กญ 3603 | กรุงเทพมหานคร... |
| 4 | Honda กพ 6456 | นนทบุรี... |

Mode: Text

Type of data currently in 0 characters

รูปที่ 3.63 Browse Data ใน Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ากล้องเว็บแคมเราทำการจับภาพข้อมูลที่ตรงกันได้ จะส่งข้อมูลและภาพไปขึ้นบนไลน์



รูปที่ 3.66 ส่งข้อมูลขึ้นไปบนไลน์โดยใช้ Raspberry Pi

ข้อมูลและภาพส่งไปให้บนไลน์เมื่อข้อมูลในฐานข้อมูลกับที่ตรวจจับได้ตรงกัน

บทที่ 4

ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลอง และ ผลการทดลองของเทคโนโลยีการตรวจจับป้ายทะเบียนผ่าน Raspberry Pi ด้วยเทคนิคการประมวลผลภาพแบบออนไลน์

4.1 ผลทดลองประมวลผลภาพป้ายทะเบียน

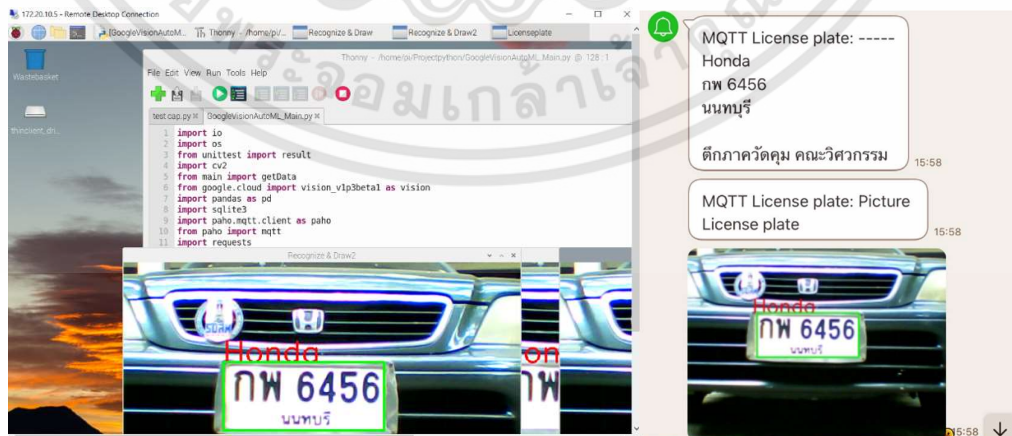
ทำการทดลองตรวจจับภาพป้ายทะเบียนโดย จะกำหนดเงื่อนไขในการทดลองอยู่ 4 อย่าง คือ

1. ระยะห่างระหว่างกล้อง กับ ป้ายทะเบียน
2. มุมองศาระหว่างกล้องกับป้ายทะเบียน
3. ความแตกต่างระหว่างใช้งานเวลาเช้า กับ เวลากลางวัน
4. ประสิทธิภาพการประมวลผล จากตัวอย่างรถยนต์ 50 คัน

กรณีแรกเป็นการทดลองตรวจจับมุมองศาและระยะห่างระหว่างกล้องกับป้ายทะเบียนรถยนต์ โดยจะทำมุม 0 องศา 10 องศา 20 องศา 30 องศา และ 40 องศา กับป้ายทะเบียนรถยนต์ กำหนดตัวแปรควบคุม คือความสูงของกล้องจากพื้น ผู้ทดลองได้กำหนดไว้ที่ 40 เซนติเมตร เพื่อหาตำแหน่งในการตั้งกล้องที่ดีที่สุดในการตรวจจับป้ายทะเบียนโดยกล้องที่เราใช้เป็นกล้องเว็บแคม 1080p รุ่น A875-A

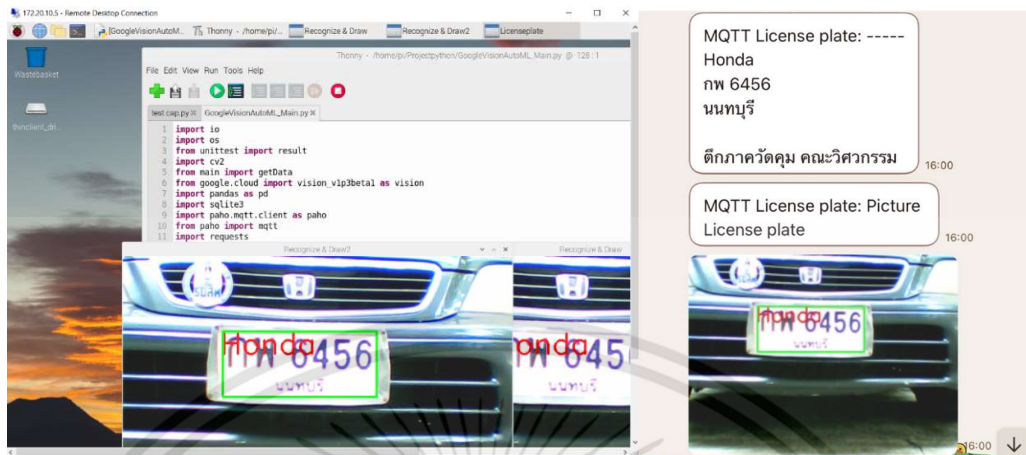
4.1.1 ผลการทดลองตรวจจับป้ายทะเบียนโดยกล้องมีระยะห่าง 100 เซนติเมตร

4.1.1.1 เวลากลางวัน

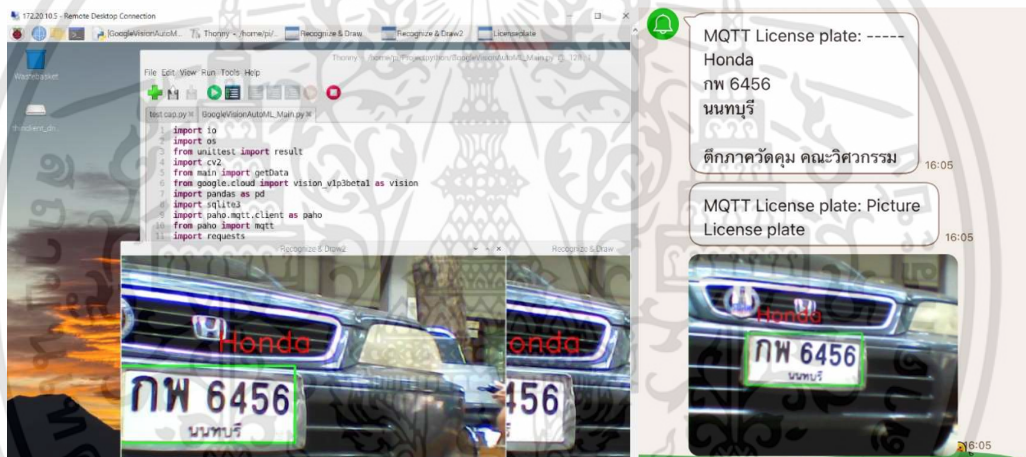


รูปที่ 4.1 มุม 0 องศา ระยะ 100 เซนติเมตร เวลากลางวัน

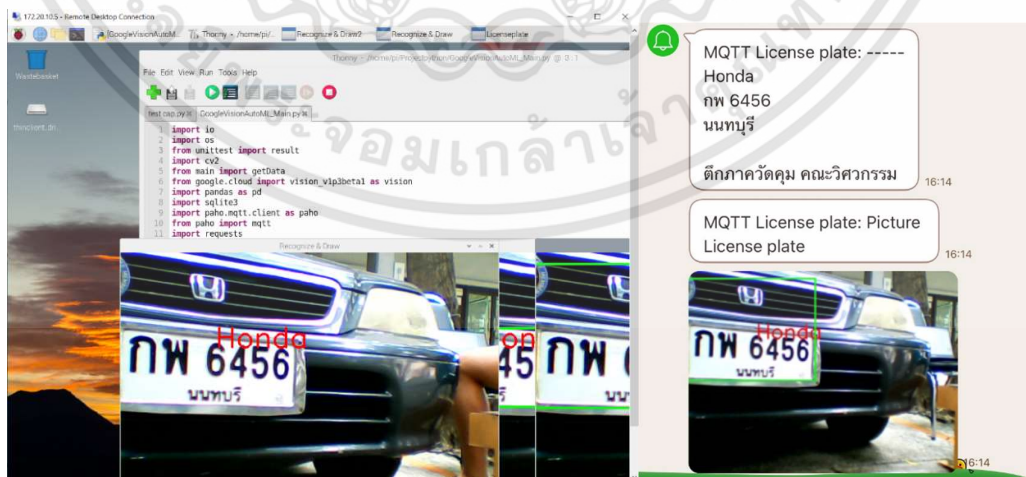
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 มุม 10 องศา ระยะ 100 เซนติเมตร เวลากลางวัน

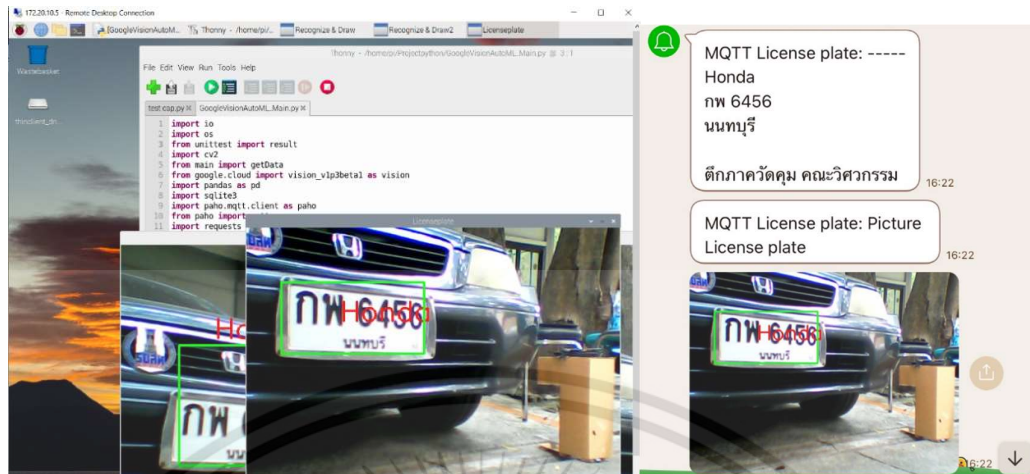


รูปที่ 4.3 มุม 20 องศา ระยะ 100 เซนติเมตร เวลากลางวัน



รูปที่ 4.4 มุม 30 องศา ระยะ 100 เซนติเมตร เวลากลางวัน

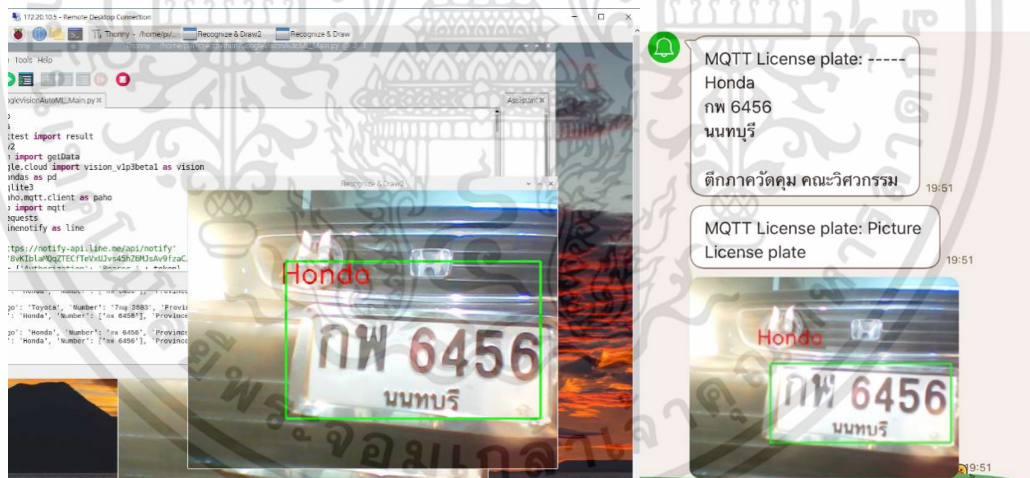
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่หวังผลใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 มุม 40 องศา ระยะ 100 เซนติเมตร เวลากลางวัน

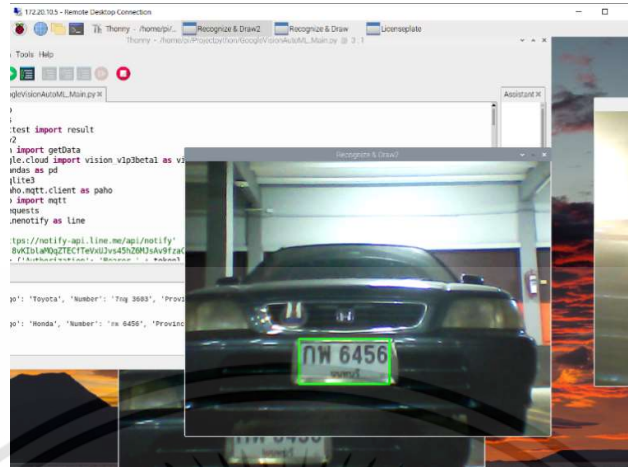
ผลการทดลองในระยะ 100 เซนติเมตร ในเวลากลางวัน พบว่ากล้องสามารถตรวจจับป้ายทะเบียนได้ในมุม 0 - 40 องศา ซึ่งในมุมมองที่มากกว่า 40 องศา นั้นมีมุมที่เอียงจากป้ายทะเบียนมากเกินไปทำให้ไม่สามารถตรวจจับป้ายทะเบียนได้

4.1.1.2 เวลากลางคืน



รูปที่ 4.6 มุม 0 องศา ระยะ 100 เซนติเมตร เวลากลางคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 มุม 0 องศา ระยะ 100 เซนติเมตร เวลากลางคืน

ผลการทดลองในเวลาเย็น ทำการเปิดแสงหน้ารถ ซึ่งกล้องเว็บแคม 1080p รุ่น A875-A มีประสิทธิภาพในการตรวจจับตอนกลางคืนได้ไม่ดีพอ จึงทำให้แสงสะท้อนกลับเข้าตัวเลนส์ กล้อง ทำให้ภาพในโปรแกรมเห็นยี่ห้อและป้ายทะเบียนได้ไม่ชัดเจน

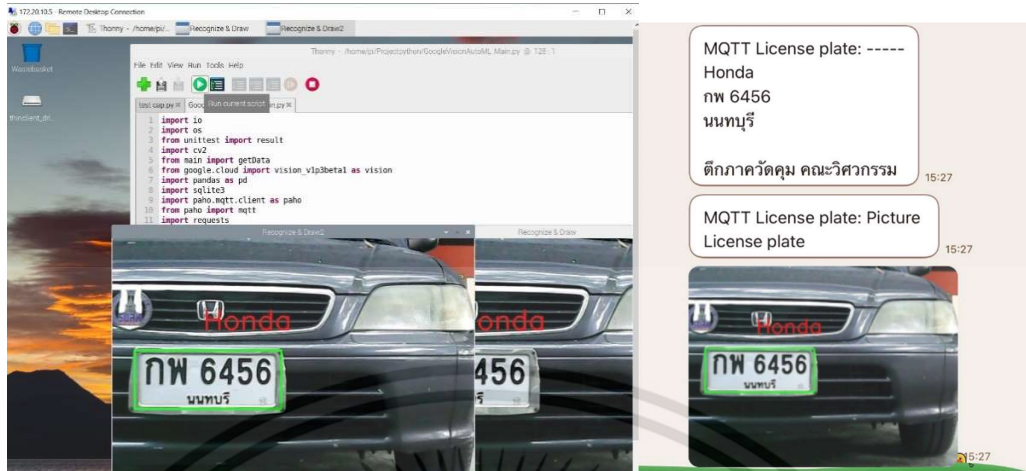
4.1.2 ผลการทดลองตรวจจับป้ายทะเบียนโดยกล้องมีระยะห่าง 200 cm

4.1.2.1 เวลากลางวัน

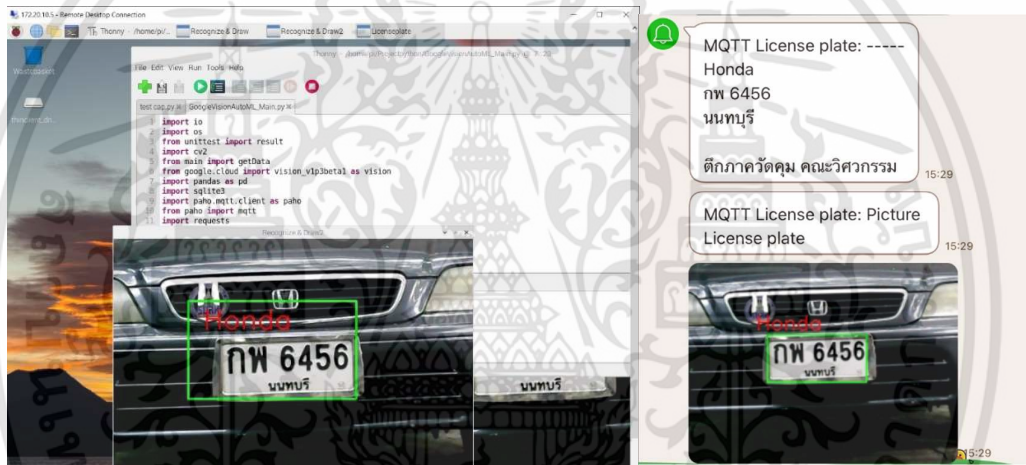


รูปที่ 4.8 มุม 0 องศา ระยะ 200 เซนติเมตร เวลากลางวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 มุม 10 องศา ระยะ 200 เซนติเมตร เวลากลางวัน

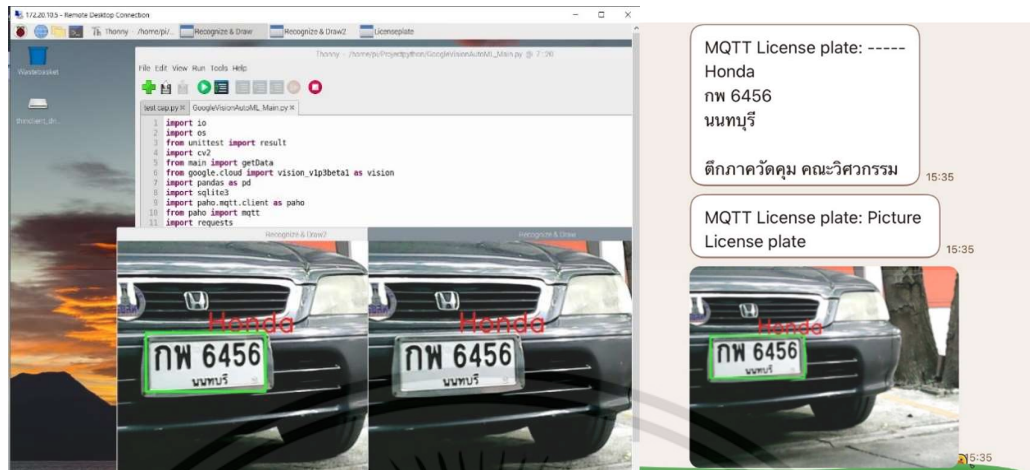


รูปที่ 4.10 มุม 20 องศา ระยะ 200 เซนติเมตร เวลากลางวัน



รูปที่ 4.11 มุม 30 องศา ระยะ 200 เซนติเมตร เวลากลางวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 มุม 40 องศา ระยะ 200 เซนติเมตร เวลากลางวัน

ผลการทดลองในระยะ 200 เซนติเมตร ในเวลากลางวัน พบว่ากล้องสามารถตรวจจับป้ายทะเบียนได้ในมุม 0 - 40 องศา ซึ่งในมุมมองที่มากกว่า 40 องศา นั้นมีมุมที่เอียงจากป้ายทะเบียนมากเกินไป ทำให้ไม่สามารถตรวจจับป้ายทะเบียนได้

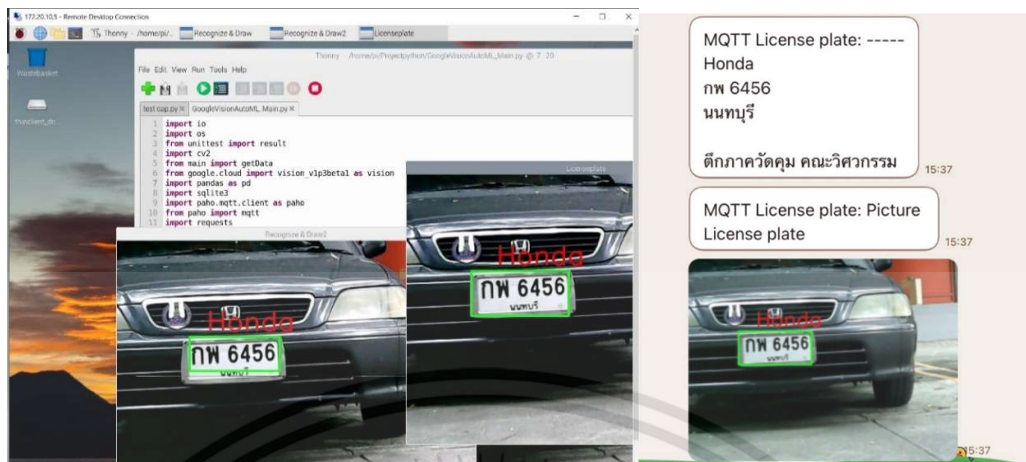
4.1.3 ผลการทดลองตรวจจับป้ายทะเบียนโดยกล้องมีระยะห่าง 300 เซนติเมตร

4.1.3.1 เวลากลางวัน

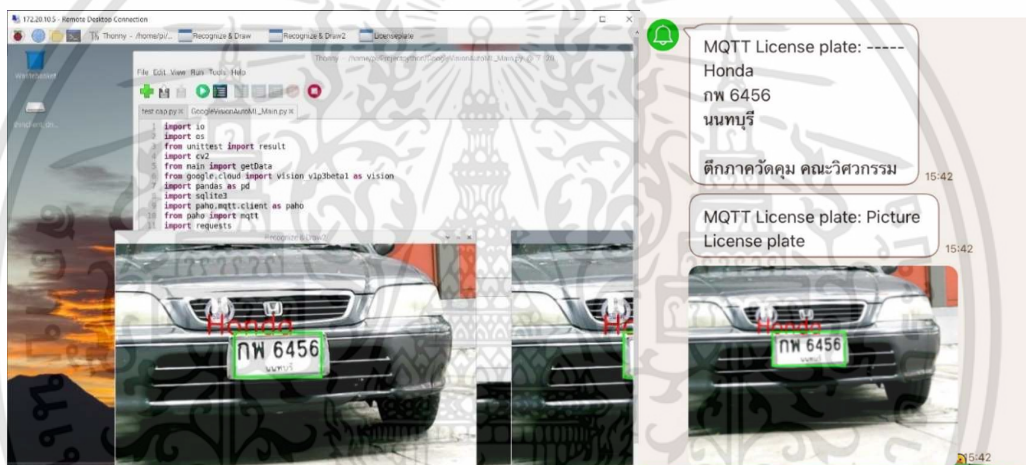


รูปที่ 4.13 มุม 0 องศา ระยะ 300 เซนติเมตร เวลากลางวัน

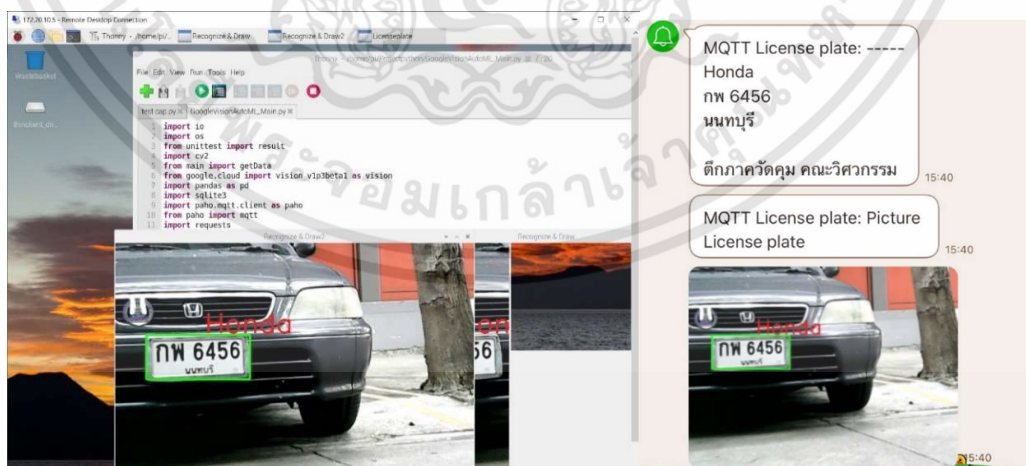
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 มุม 10 องศา ระยะ 300 เซนติเมตร เวลากลางวัน



รูปที่ 4.15 มุม 20 องศา ระยะ 300 เซนติเมตร เวลากลางวัน



รูปที่ 4.16 มุม 30 องศา ระยะ 300 เซนติเมตร เวลากลางวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

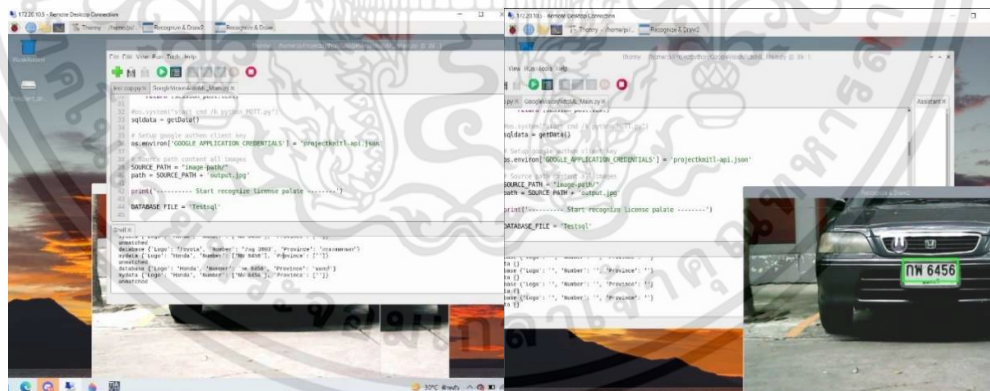


รูปที่ 4.17 มุม 40 องศา ระยะ 300 cm เวลากลางวันไม่สามารถตรวจจับได้

ผลการทดลองในระยะ 300 เซนติเมตร ในเวลากลางวัน พบว่ากล้องสามารถตรวจจับป้ายทะเบียนได้ในมุม 0 - 30 องศา ซึ่งในมุมมองที่มากกว่า 30 องศา นั้นมีมุมที่เอียงจากป้ายทะเบียนมากเกินไปทำให้ไม่สามารถตรวจจับป้ายทะเบียนได้

4.1.4 ผลการทดลองตรวจจับป้ายทะเบียนโดยกล้องมีระยะห่าง 400 เซนติเมตร

4.1.4.1 เวลากลางวัน



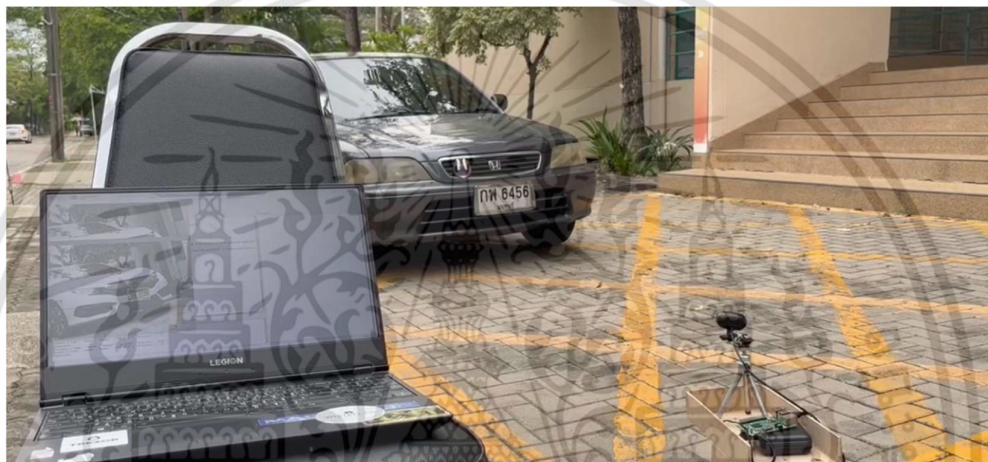
รูปที่ 4.18 มุม 0 องศา ระยะ 400 เซนติเมตร เวลากลางวันไม่สามารถตรวจจับได้

ผลการทดลองในระยะ 400 เซนติเมตร ในเวลากลางวัน พบว่ากล้องไม่สามารถตรวจจับป้ายทะเบียนได้ เนื่องจากในระยะห่างนี้กล้องไม่สามารถโฟกัสหรือ รถยนต์ได้ ซึ่งทำให้ไม่สามารถที่จะอ่านป้ายทะเบียนได้ด้วยตัวเอง หลังจากที่ได้ทดลององศากล้องเบื้องต้นแล้ว ซึ่งได้รับผลสรุปว่า มุมที่ควรตั้งกล้องที่สุด คือ 0 - 20 องศา และ ระยะห่างที่เหมาะสมที่สุดคือ 100 -

300 เซนติเมตร โดยในเวลากลางวันจะตรวจจับได้ดีกว่าในเวลากลางคืน ทั้งนี้ขึ้นอยู่กับความสามารถของกล้องที่ใช้ตรวจจับ

4.2 ทดสอบใช้งานจริง

หลังจากที่เราได้ทดลององศากล้องเบื้องต้นแล้ว ซึ่งได้รับผลการทดลองว่า มุมที่ควรตั้งกล้องที่สุด คือ 0 – 20 องศา และ ระยะห่างที่เหมาะสมที่สุดคือ 100 - 300 เซนติเมตร ซึ่งเราจะนำผลการทดลองมาใช้ในการตรวจจับรถยนต์ ดังรูปที่แสดง



รูปที่ 4.19 ทดสอบใช้งานจริงในการตรวจจับรถยนต์



รูปที่ 4.20 ทดสอบใช้งานจริงในการตรวจจับรถยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 ทดสอบใช้งานจริงข้อมูลส่งขึ้นมาในไลน์



รูปที่ 4.22 ข้อมูลที่ทดสอบถูกส่งขึ้นมาบนไลน์

จากผลการทดลอง ในระยะห่าง 200 เซนติเมตร กล้องมีการตรวจจับยี่ห้อและป้ายทะเบียนได้แล้วแต่จะไม่เรียลไทม์ เนื่องจากเกิดการดีเลย์เวลาในการรับส่งข้อมูลกับ Google cloud API ประมาณ 10-15 วินาที ซึ่งถือว่าอยู่ในช่วงที่ยอมรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 ประสิทธิภาพการประมวลผล การทดสอบตรวจจ็บบรถยนต์จำนวน 50 คัน

จากผลการตรวจจ็บบรถยนต์ทั้ง 50 คัน แบ่งได้เป็น 2 แบบ คือ

1. รถยนต์ที่ตรวจจ็บบได้
2. รถยนต์ที่ตรวจจ็บบไม่ได้

ซึ่งจากรถยนต์ทั้ง 50 คัน ที่เราใช้ทดสอบ มีรถยนต์ที่ตรวจจ็บบได้ทั้งหมด 34 คัน และมีรถยนต์ที่ตรวจจ็บบไม่ได้ทั้งหมด 16 คัน ทำให้สามารถสรุปประสิทธิภาพในการตรวจจ็บบ ด้วยสาเหตุจากป้ายทะเบียนของรถยนต์บางคัน มีคราบสกปรก มีลักษณะการติดตั้งป้ายทะเบียนที่ทำให้ตรวจจ็บบได้ยาก หรือ มีคุณลักษณะที่คล้ายตัวอักษร ที่ทำให้ไม่สามารถวิเคราะห์เป็นตัวอักษรได้ และ รถยนต์บางคันมีลักษณะแบรนด์ที่ซับซ้อน ทำให้ปัญญาประดิษฐ์ (Artificial Intelligence) ไม่สามารถวิเคราะห์ได้ถูกต้อง

4.2.1.1 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ถูกต้อง

รถยนต์แบรนด์ Nissan ป้ายทะเบียน กฉ 6475 จังหวัด สมุทรปราการ สามารถตรวจจ็บบป้ายทะเบียนได้ถูกต้อง และแจ้งเตือนผ่านไลน์

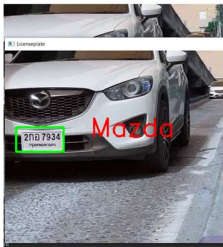
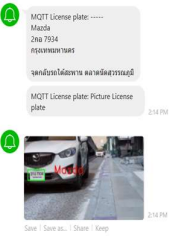

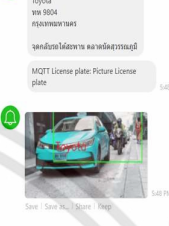


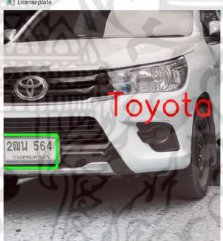







รูปที่ 4.23 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ถูกต้อง

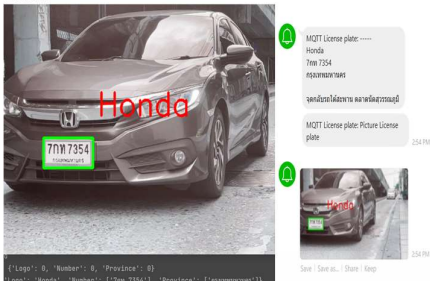

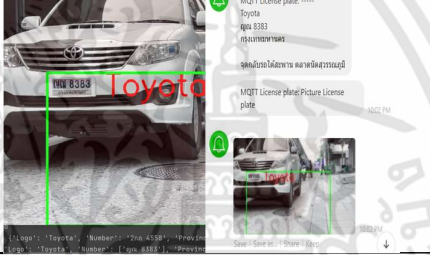


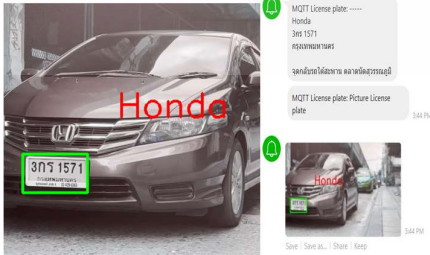
ตารางที่ 4.1 ตรวจสอบแบรนด์รถยนต์ได้ถูกต้อง

| ลำดับที่ | ข้อมูลจาก Database | ข้อมูล my data | ภาพที่ถ่ายได้ | หมายเหตุ |
|----------|---|---|---------------|----------|
| 1 | Mercedes-Benz , ญอ 8866 , กรุงเทพมหานคร | Mercedes-Benz , ญอ 8866 , กรุงเทพมหานคร | | |
| 2 | Nissan กฉ 6475 สมุทรปราการ | Nissan กฉ 6475 สมุทรปราการ | | |
| 3 | Toyota, ญอ 1328, กรุงเทพมหานคร | Toyota, ญอ 1328, กรุงเทพมหานคร | | |
| 4 | Toyota, งท 5889, ชลบุรี | Toyota, งท 5889, ชลบุรี | | |
| 5 | Toyota, 1ขจ7376, กรุงเทพมหานคร | Toyota, 1ขจ7376, กรุงเทพมหานคร | | |






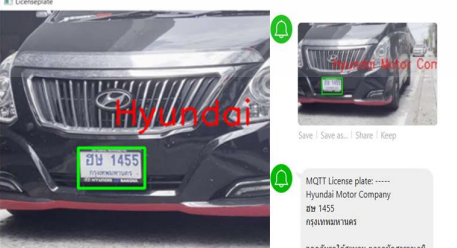
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | |
|----|---------------------------------------|---------------------------------------|--|--|
| 6 | Mazda, 2กอ 7934, กรุงเทพมหานคร | Mazda, 2กอ 7934, กรุงเทพมหานคร |   | |
| 7 | Toyota, ทท 9804, กรุงเทพมหานคร | Toyota, ทท 9804, กรุงเทพมหานคร |   | |
| 8 | Toyota, 2กย 7318, กรุงเทพมหานคร | Toyota, 2กย 7318, กรุงเทพมหานคร |   | |
| 9 | Toyota, 2ฉน 564, กรุงเทพมหานคร | Toyota, 2ฉน 564, กรุงเทพมหานคร |   | |
| 10 | Toyota, 8กพ 7560, กรุงเทพมหานคร | Toyota, 8กพ 7560, กรุงเทพมหานคร |   | |
| 11 | Toyota, 5กส 1413, กรุงเทพมหานคร | Toyota, 5กส 1413, กรุงเทพมหานคร |   | |








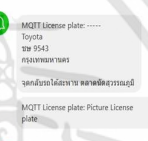







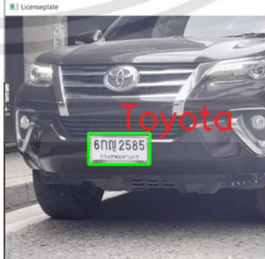


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | |
|----|---|---|--|--|
| 12 | Honda, 7กท 7354, กรุงเทพมหานคร | Honda, 7กท 7354, กรุงเทพมหานคร |  | |
| 13 | Toyota, ทท 267, กรุงเทพมหานคร | Toyota, ทท 267, กรุงเทพมหานคร |  | |
| 14 | Toyota, ญณ 8383, กรุงเทพมหานคร | Toyota, ญณ 8383, กรุงเทพมหานคร |  | |
| 15 | Mitsubishi Motors, 1ฒช 4246, กรุงเทพมหานคร | Mitsubishi Motors, 1ฒช 4246, กรุงเทพมหานคร |  | |
| 16 | Honda, 7กท 953, กรุงเทพมหานคร | Honda, 7กท 953, กรุงเทพมหานคร |  | |
| 17 | Honda, 3กร 1571, กรุงเทพมหานคร | Honda, 3กร 1571, กรุงเทพมหานคร |  | |

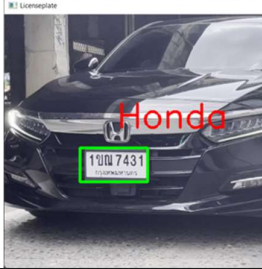
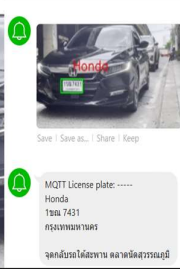
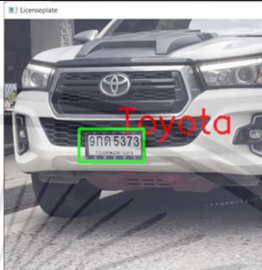
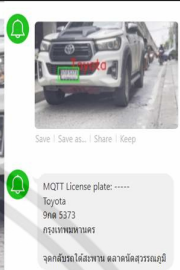

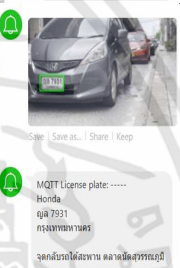
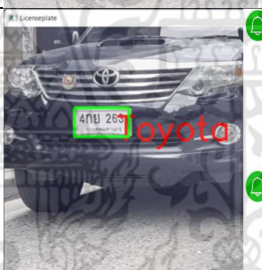
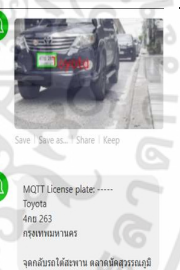

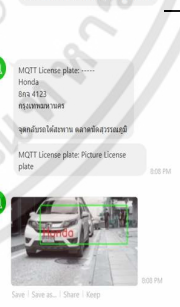
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | |
|-----------|--|--|--|--|
| <p>18</p> | <p>Toyota, ตน 9390, กรุงเทพมหานคร</p> | <p>Toyota, ตน 9390, กรุงเทพมหานคร</p> |  | |
| <p>19</p> | <p>Toyota, ชง 187, กรุงเทพมหานคร</p> | <p>Toyota, ชง 187, กรุงเทพมหานคร</p> |  | |
| <p>20</p> | <p>Toyota, 6กว 726, กรุงเทพมหานคร</p> | <p>Toyota, 6กว 726, กรุงเทพมหานคร</p> |  | |
| <p>21</p> | <p>Toyota, 4กช 5983, กรุงเทพมหานคร</p> | <p>Toyota, 4กช 5983, กรุงเทพมหานคร</p> |  | |
| <p>22</p> | <p>Toyota, มช 8141, กรุงเทพมหานคร</p> | <p>Toyota, มช 8141, กรุงเทพมหานคร</p> |  | |
| <p>23</p> | <p>Hyundai Motor Company, ฮษ 1455, กรุงเทพมหานคร</p> | <p>Hyundai Motor Company, ฮษ 1455, กรุงเทพมหานคร</p> |  | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | |
|----|---------------------------------------|---------------------------------------|--|
| 24 | Toyota, chr 403, ชลบุรี | Toyota, chr 403, ชลบุรี |    |
| 25 | Honda, 2ขถ 9568, กรุงเทพมหานคร | Honda, 2ขถ 9568, กรุงเทพมหานคร |    |
| 26 | Toyota, ชช 9543, กรุงเทพมหานคร | Toyota, ชช 9543, กรุงเทพมหานคร |    |
| 27 | Honda, ชฟ 9332, กรุงเทพมหานคร | Honda, ชฟ 9332, กรุงเทพมหานคร |    |
| 28 | Toyota, 2ตม 8489, กรุงเทพมหานคร | Toyota, 2ตม 8489, กรุงเทพมหานคร |    |
| 29 | Toyota, 6กญ 2585, กรุงเทพมหานคร | Toyota, 6กญ 2585, กรุงเทพมหานคร |    |

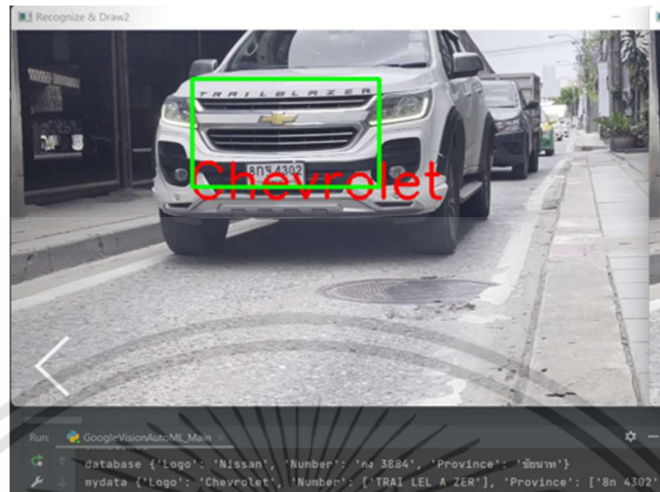
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | |
|----|---------------------------------------|---------------------------------------|--|
| 30 | Honda, 1ขณ 7431, กรุงเทพมหานคร | Honda, 1ขณ 7431, กรุงเทพมหานคร |   |
| 31 | Toyota, 9กต 5373, กรุงเทพมหานคร | Toyota, 9กต 5373, กรุงเทพมหานคร |   |
| 32 | Honda, ญล 7931, กรุงเทพมหานคร | Honda, ญล 7931, กรุงเทพมหานคร |   |
| 33 | Toyota, 4กย 263, กรุงเทพมหานคร | Toyota, 4กย 263, กรุงเทพมหานคร |   |
| 34 | Honda, 8กจ 4123, กรุงเทพมหานคร | Honda, 8กจ 4123, กรุงเทพมหานคร |   |

4.2.1.2 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ไม่ถูกต้อง

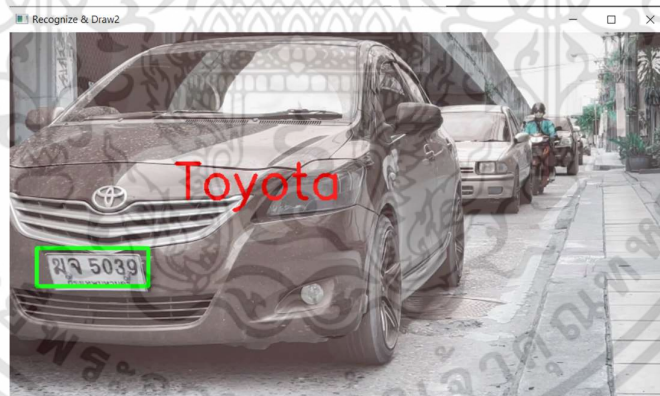
รถยนต์แบรนด์ Chevrolet ป้ายทะเบียน 8กฐ 4302 จังหวัด กรุงเทพมหานคร อ่านได้ แบรนด์ Chevrolet ป้ายทะเบียน TRAI LEL A ZER จังหวัด 8น 4302 เนื่องจากแบรนด์มีลักษณะที่ซับซ้อน จึงทำให้ทำให้ปัญญาประดิษฐ์ (Artificial Intelligence) ไม่สามารถวิเคราะห์ได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ไม่ถูกต้อง

รถยนต์แบรนด์ Toyota ป้ายทะเบียน ขจ 5039 จังหวัด กรุงเทพมหานคร อ่านได้ แบรนด์ Toyota ป้ายทะเบียน 23 5039 ไม่สามารถตรวจจับจังหวัดได้ เนื่องจากป้ายทะเบียนมีการติดตั้งเอียงลงต่ำ จึงทำให้ทำให้ปัญญาประดิษฐ์ (Artificial Intelligence) ไม่สามารถวิเคราะห์ได้ถูกต้อง



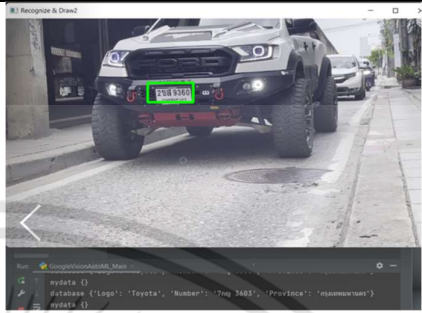
รูปที่ 4.25 ตัวอย่างรถยนต์ที่ตรวจสอบทะเบียนได้ไม่ถูกต้อง

โดยจากตารางแสดงผลรถยนต์ทั้ง 50 คัน เราสามารถจำแนกส่วนที่ตรวจจับผิดพลาดออกมาได้เป็น 3 รูปแบบหลักๆ

1. ไม่สามารถตรวจจับแบรนด์รถยนต์ได้
2. ไม่สามารถตรวจจับจังหวัดได้
3. ตรวจจับตัวอักษรภาษาไทยไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ไม่สามารถตรวจจับแบรนด์รถยนต์ได้

| ลำดับ ที่ | ข้อมูลจาก Database | ข้อมูล my data | ภาพที่ถ่ายได้ | หมายเหตุ |
|--------------|--|----------------|--|---|
| 1 | Ford Motor Company, 2ขส 9360, กรุงเทพมหานคร | , , , |  | ไม่ สามารถ ตรวจจับ แบรนด์ รถยนต์ได้ |

จากรถยนต์ทั้ง 50 คัน ไม่สามารถตรวจจับ แบรนด์รถยนต์ได้ทั้งสิ้น 1 คัน คิดเป็นความแม่นยำ 98% ด้วยสาเหตุจากแบรนด์ของรถยนต์มีสีดำ กลมกลืนกับตัวรถ ทำให้ปัญญาประดิษฐ์ (Artificial Intelligence) ไม่สามารถวิเคราะห์ได้ถูกต้อง

4.2.1.3 ตัวอย่างรถยนต์ที่ไม่สามารถตรวจจับแบรนด์รถยนต์ได้

รถยนต์แบรนด์ Ford Motor Company ป้ายทะเบียน 2ขส 9360 จังหวัด กรุงเทพมหานคร กล้องไม่สามารถอ่านได้เนื่องจาก แบรนด์ของรถยนต์มีสีดำ กลมกลืนกับตัวรถ จึงทำให้ปัญญาประดิษฐ์ (Artificial Intelligence) ไม่สามารถวิเคราะห์ได้ถูกต้อง



รูปที่ 4.26 ตัวอย่างรถยนต์ที่ไม่สามารถตรวจจับแบรนด์รถยนต์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ไม่สามารถตรวจจับจังหวัดได้

| ลำดับที่ | ข้อมูลจาก Database | ข้อมูล my data | ภาพที่ถ่ายได้ | หมายเหตุ |
|----------|--|---|---------------|--|
| 1 | Chevrolet, 8กฐ 4302, กรุงเทพมหานคร | Chevrolet, TRAI LEL A ZER, 8n 4302 | | แบรนด์รถยนต์มีลักษณะซ้ำซ้อน |
| 2 | Mazda, 2กฏ 8623, กรุงเทพมหานคร | Mazda, 2no 8623 , | | กล้องตรวจจับข้อมูลไม่ถูกต้องตามฐานข้อมูล |
| 3 | Mercedes-Benz, 5กค 8388, กรุงเทพมหานคร | Mercedes-Benz, 5nn 8388, | | กล้องตรวจจับข้อมูลไม่ถูกต้องตามฐานข้อมูล |
| 4 | Toyota, 5กข 764, กรุงเทพมหานคร | Toyota, 5NU 764, | | กล้องตรวจจับข้อมูลไม่ถูกต้องตามฐานข้อมูล |
| 5 | Isuzu Motors, ฒน 4770, กรุงเทพมหานคร | Isuzu Motors, ฒน 4770 | | กล้องตรวจจับข้อมูลไม่ถูกต้องตามฐานข้อมูล |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | |
|----|---|---|--|---|
| 6 | Ford Motor Company, 1ขล 4659, กรุงเทพมหานคร | Ford Motor Company, EVEREST, 1ขล 4659 | | แบรนด์มี ลักษณะ ซับซ้อน |
| 7 | Toyota, บล 5108, ฉะเชิงเทรา | Toyota, บล 5108, | | กล้องไม่ ตรวจจับ จังหวัด เนื่องจาก ไม่เห็นใน ป้าย ทะเบียน |
| 8 | Toyota, ฉจ 5039, กรุงเทพมหานคร | Toyota, 23 5039, | | ป้าย ทะเบียน เอียงลง ต่ำทำให้ ไม่ สามารถ ตรวจจับ ได้ |
| 9 | Ford Motor Company, 4กถ 772, กรุงเทพมหานคร | Ford Motor Company, Ford, 400 772 | | กล้อง ตรวจจับ ข้อมูลไม่ ถูกต้อง ตาม ฐานข้อมูล |
| 10 | Nissan, 3กส 7611, กรุงเทพมหานคร | Nissan, 3na 7611, | | กล้องไม่ ตรวจจับ จังหวัด เนื่องจาก ไม่เห็นใน ป้าย ทะเบียน |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรถยนต์ทั้ง 50 คัน ไม่สามารถตรวจจับจังหวัดของป้ายทะเบียนได้ทั้งสิ้น 10 คัน คิดเป็นความแม่นยำ 80% ด้วยสาเหตุจากป้ายทะเบียนของรถยนต์บางคัน มีคราบสกปรก ครอบป้ายทะเบียนบดบังตัวอักษร มีลักษณะการติดตั้งป้ายทะเบียนที่ทำให้ตรวจจับได้ยาก

4.1.2.4 ตัวอย่างรถยนต์ที่ไม่สามารถตรวจจับจังหวัดรถยนต์ได้

รถยนต์แบรนด์ Toyota ป้ายทะเบียน บล 5108 จังหวัด ฉะเชิงเทรา อ่านได้ แบรนด์ Toyota ป้ายทะเบียน บล 5108 จังหวัดไม่สามารถอ่านได้ เนื่องจากกรอบของป้ายทะเบียน บดบังตัวอักษร จึงทำให้ทำให้ปัญญาประดิษฐ์ (Artificial Intelligence) ไม่สามารถวิเคราะห์ได้ถูกต้อง

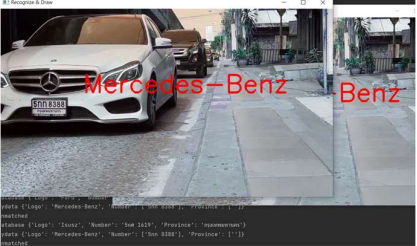






รูปที่ 4.27 ตัวอย่างรถยนต์ที่ไม่สามารถตรวจจับจังหวัดรถยนต์ได้

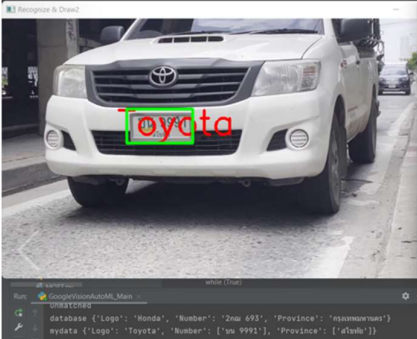


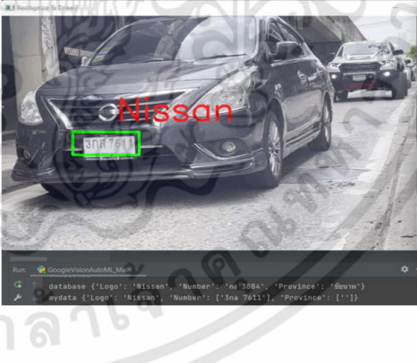
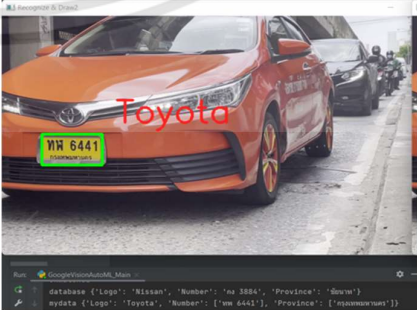
ตารางที่ 4.4 ตรวจจับตัวอักษรภาษาไทยได้ไม่ถูกต้อง

| ลำดับที่ | ข้อมูลจาก Database | ข้อมูล my data | ภาพที่ถ่ายได้ | หมายเหตุ |
|----------|--|---|---------------|--|
| 1 | Chevrolet, 8กฐ 4302, กรุงเทพมหานคร | Chevrolet, TRAI LEL A ZER, 8n 4302 | | แบรนด์รถยนต์มีลักษณะซับซ้อน |
| 2 | Mazda, 2กฏ 8623, กรุงเทพมหานคร | Mazda, 2no 8623 , | | กล้องตรวจจับข้อมูลไม่ถูกต้องตามฐานข้อมูล |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | |
|---|---|---|--|---|
| 3 | Mercedes-Benz, 5กท 8388, กรุงเทพมหานคร | Mercedes-Benz, 5nn 8388, |  | กล้อง ตรวจจับ ข้อมูลไม่ ถูกต้อง ตาม ฐานข้อมูล |
| 4 | Isuzu Motors, 5กต 1619, กรุงเทพมหานคร | Isuzu Motors, 5nn 1619, กรุงเทพมหานคร |  | กล้อง ตรวจจับ ข้อมูลไม่ ถูกต้อง ตาม ฐานข้อมูล |
| 5 | Toyota, 5กข 764, กรุงเทพมหานคร | Toyota, 5NU 764, |  | กล้อง ตรวจจับ ข้อมูลไม่ ถูกต้อง ตาม ฐานข้อมูล |
| 6 | Toyota, ฉจ 5039, กรุงเทพมหานคร | Toyota, 23 5039, |  | ป้าย ทะเบียน เอียงลง ต่ำทำให้ ไม่ สามารถ ตรวจจับ ได้ |
| 7 | Ford Motor Company, 4กถ 772, กรุงเทพมหานคร | Ford Motor Company, Ford, 400 772 |  | กล้อง ตรวจจับ ข้อมูลไม่ ถูกต้อง ตาม ฐานข้อมูล |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | |
|----|---|---|--|---|
| 8 | Toyota, บน 9991, สุโขทัย | Toyota, บน 9991, สุโขทัย |  | กล้อง ตรวจจับ ข้อมูลไม่ ถูกต้อง ตาม ฐานข้อมูล |
| 9 | Honda, 2กณ 693, กรุงเทพมหานคร | Honda, 2กณ 693, กรุงเทพมหานคร |  | เนื่องจาก ตัวอักษร ณ และ ณ มีลักษณะ คล้ายกัน |
| 10 | Mitsubishi Motors, 7กย 8517, กรุงเทพมหานคร | Mitsubishi Motors, 7NU 8517, กรุงเทพมหานคร |  | กล้อง ตรวจจับ ข้อมูลไม่ ถูกต้อง ตาม ฐานข้อมูล |
| 11 | Nissan, 3กส 7611, กรุงเทพมหานคร | Nissan, 3na 7611, กรุงเทพมหานคร |  | กล้องไม่ ตรวจจับ จังหวัด เนื่องจาก ไม่เห็นใน ป้าย ทะเบียน |
| 12 | Toyota, ทพ 6441, กรุงเทพมหานคร | Toyota, ทพ 6441, กรุงเทพมหานคร |  | ตัวอักษร พ และ พ มีลักษณะ คล้ายกัน |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรถยนต์ทั้ง 50 คัน ตรวจจับตัวอักษรภาษาไทยได้ไม่ถูกต้องทั้งสิ้น 12 คัน คิดเป็นความแม่นยำ 76% ด้วยสาเหตุลักษณะของตัวอักษรบางชนิดมีความใกล้เคียงกัน และตัวอักษรบางตัวมีการเทรนData ที่น้อยเกินไป ให้ปัญญาประดิษฐ์ (Artificial Intelligence) ไม่สามารถวิเคราะห์ได้ถูกต้อง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

ในบทนี้จะกล่าวถึง สรุปผลการทดลอง ปัญหา และ อุปสรรคในการทำงาน และ ข้อเสนอแนะต่างๆ ที่ได้จากระบบตรวจจับป้ายทะเบียนรถยนต์ ด้วยการส่งข้อมูลแบบ MQTT การประมวลผลภาพ และการส่งข้อมูลไปให้เจ้าหน้าที่หรือผู้รับผิดชอบผ่านแอปพลิเคชันไลน์

5.1 สรุปผลการทดลอง

จากผลการทดลองการใช้งานระบบติดตามการจราจรรถยนต์ ด้วยการประมวลผลภาพ มีรายละเอียดดังนี้

1. การทดลองสามารถใช้งานได้ดี สะดวก รวดเร็ว สามารถติดตั้งจุดตรวจจับได้ไม่จำกัด
2. การทดลองส่งค่าจาก MQTT ไป เก็บไว้ใน Database SQLiteสามารถส่งจากที่ไหนก็ได้ โดยต้องกำหนด Username , Password, Host , Port ,Topic ให้ตรงกับที่โค้ด Python กำหนดไว้
3. การส่งข้อมูลขึ้นไลน์ สามารถส่งข้อมูลได้ถูกต้องและเรียลไทม์ สามารถเลื่อนดูย้อนหลัง และ ดูวันเวลาที่ส่งมาได้ในไลน์
4. การทดลองตรวจจับป้ายทะเบียนด้วยการประมวลผลภาพ สามารถตรวจจับป้ายทะเบียนได้ดีในมุม 0 – 20 องศา ในระยะ 100 – 300 เซนติเมตร ระยะมากกว่านี้กล้องไม่สามารถตรวจจับได้ แสงสว่างจากพระอาทิตย์ที่สะท้อนกับแบนน มีผลทำให้การตรวจจับยากขึ้น
5. จากการทดลองรถยนต์ทั้งหมด 50 คัน ได้ประสิทธิภาพความแม่นยำ 68 % สามารถตรวจจับได้ถูกต้อง 34 คัน และ ตรวจจับได้แต่ข้อมูลไม่ถูกต้องครบถ้วน 16 คัน เช่น
 - 1) แบนด์ Chevrolet ป้ายทะเบียน 8กฐ 4302 จังหวัด กรุงเทพมหานคร
ตรวจจับได้ แบนด์ Chevrolet ป้ายทะเบียน TRAI LEL AZER จังหวัด 8น 4302
เนื่องจากแบนด์ของรถยนต์มีความซับซ้อนจึงทำให้ตรวจจับได้ยาก
 - 2) แบนด์ Mazda ป้ายทะเบียน 2กฏ 8623 จังหวัด กรุงเทพมหานคร
ตรวจจับได้ แบนด์ Mazda ป้ายทะเบียน 2no 8623 ไม่สามารถตรวจจับจังหวัดได้
เนื่องจากป้ายทะเบียนมีคราบสกปรกจึงทำให้ไม่สามารถตรวจจับจังหวัดได้
 - 3) แบนด์ Nissan ป้ายทะเบียน 3กส 7611 จังหวัด กรุงเทพมหานคร
ตรวจจับได้ แบนด์ Nissan ป้ายทะเบียน 3na 7611 ไม่สามารถตรวจจับจังหวัดได้
เนื่องจากป้ายทะเบียนจางจึงทำให้ไม่สามารถตรวจจับจังหวัดได้

5.2 ปัญหา และ อุปสรรค

- 1) กล้องที่ใช้มีความละเอียดต่ำ ทำให้ตรวจจับได้ในระยะที่ไม่เกิน 400 เซนติเมตรเท่านั้น หากใช้กล้องคุณภาพสูง จะทำให้ระบบทำงานได้อย่างเต็มประสิทธิภาพ
- 2) การเรียกใช้งาน Google cloud ผ่าน API ทำให้การประมวลผลไม่เรียลไทม์เกิดความล่าช้า 10 – 15 วินาที

5.3 ข้อเสนอแนะ

1. การใช้กล้องที่มีความละเอียดสูง จะทำให้การตรวจจับป้ายทะเบียนได้ดีมากยิ่งขึ้น
2. เนื่องจากทะเบียนรถยนต์ในประเทศไทยเป็นตัวอักษรภาษาไทย การเทรนโมเดลอาจมีจำนวนข้อมูลไม่มากพอ จึงทำให้การประมวลผลภาพผิดพลาดหรือถูกต้องไม่ครบถ้วน



บรรณานุกรม

- [1] Allen B. Downey, “Think Python 2nd ed”, California: North Sebastopol, 2015.
- [2] Gary Bradski and Adrian Kaehler. 2008. “Learning OpenCV”, California, O’Reilly
- [3] The Python Tutorial. [Online]. 2020. Available : <https://docs.python.org/3/tutorial/>
[AUG 20, 2021]
- [4] Pandas User Guide. [Online]. 2021. Available :
https://pandas.pydata.org/docs/user_guide/basics.html#sorting [SEP 11, 2021]
- [5] Image Processing in OpenCV. [Online]. 2020. Available :
https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html
[AUG 20, 2021]
- [6] SQL คืออะไร เอสคิวแอล. [ออนไลน์]. 2018. แหล่งที่มา :
<https://www.mindphp.com/AD/73A3/2088-sql-A3.html> [SEP 11, 2021]
- [7] Python Client – documentation. [Online]. 2020. Available :
<https://www.eclipse.org/paho/index.php?page=clients/python/docs/index.php>
[SEP 11, 2021]
- [8] MQTT กับงาน Industrial Internet of Things (IIoT). [ออนไลน์]. 2020. แหล่งที่มา :
<https://sonicautomation.co.th/mqtt-for-iiot-application/> [SEP 11, 2021]
- [9] Cloud Vision documentation. [Online]. 2021. Available :
<https://cloud.google.com/vision/docs/setup> [SEP 11, 2021]
- [10] sqlite3 DB-API 2.0 interface for SQLite databases. [Online]. 2020. Available :
<https://docs.python.org/3/library/sqlite3.html> [SEP 11, 2021]



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก
ข้อมูลอุปกรณ์และวัตถุดิบที่ใช้ในการทดลองและภาษา Python ในการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ข้อมูลอุปกรณ์และวัสดุที่ใช้ในการทดลอง

1.1 กล้องเว็บแคม 1080p รุ่น A875-A

| ข้อมูลอุปกรณ์ กล้องเว็บแคม 1080p รุ่น A875-A | |
|---|---|
| ความต้องการของระบบ | คุณสมบัติทางเทคนิค |
| ทำงานได้ทั้ง IBM PC หรือ PC หรือ Laptop ด้วยพอร์ต USB อย่างสมบูรณ์ Pentium 1 GHz or higher CPU ต้องการพื้นที่ว่างในฮาร์ดดิสก์ 500MB หน่วยความจำ 256MB รองรับ DirectX 9.0 VGA card | Webcam for PC มีไมโครโฟนในตัวแบบ HD ความละเอียด 1920 x 1080p ดีไซน์สวยงาม และเลนส์ Auto Focus ความแม่นยำสูง ออกแบบมาสำหรับทั้งแล็ปท็อปและเดสก์ท็อป ปรับหมุนได้รอบ สามารถวางบนโต๊ะ หรือบนจอก็ได้ ขนาด 70 x 51 x 44 มม. น้ำหนัก 62.8 กรัม สายยาว 1.5 เมตร |



รูปที่ ก กล้องเว็บแคม 1080p รุ่น A875-A

2. ข้อมูลภาษา Python ในการเขียนโปรแกรม

2.1 โปรแกรมส่วนหลัก

```
import io
import os
import cv2
from main import getData
from google.cloud import vision_v1p3beta1 as vision
import pandas as pd
import sqlite3
import paho.mqtt.client as paho
from paho import mqtt
import requests
import Linenotify as line

url = 'https://notify-api.line.me/api/notify'
token = '8vKIblamQqZTECFteVxUJvs45hZ6MJsAv9fzaCJwAfe'
headers = {'Authorization': 'Bearer ' + token}

def lineNotify(mid):
    payload = {'message':mid}
    return _lineNotify(payload)

def _lineNotify(payload,file=None):
    return requests.post(url, headers=headers, data=payload, files=file)

def lineNotify(message):
    img2 = {'imageFile': open('D:/pythonProject1/Licenseplate.png', 'rb')}
    data2 = {'message': message}
    session = requests.session()
    session_post = session.post(url, headers=headers, files=img2, data=data2)
    return (session_post.text)
```

```

sqldata = getData()

os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'projectkmitl-api.json'

SOURCE_PATH = "image-path/"
path = SOURCE_PATH + 'output.jpg'

print('----- Start recognize license palate -----')

DATABASE_FILE = 'licenseplate.db'

cap = cv2.VideoCapture(0)
while (True):
    ret, frame = cap.read()
    img = cv2.cvtColor(frame,1)
    height, width = img.shape[:2]
    img = cv2.resize(img, (800, int((height * 800) / width)))
    cv2.imwrite(SOURCE_PATH + "output.jpg", img)
    img_path = SOURCE_PATH + "output.jpg"
    client = vision.ImageAnnotatorClient()
    with io.open(img_path, 'rb') as image_file:
        content = image_file.read()
    image = vision.types.Image(content=content)
    response = client.logo_detection(image=image)
    logos = response.logo_annotations
    response2 = client.text_detection(image=image)
    texts = response2.text_annotations
    data = {}
    for logo in logos:
        data = {'Logo': logo.description}
        cv2.putText(img, logo.description, (200, 200), cv2.FONT_HERSHEY_SIMPLEX,
2, (0, 0, 255), 3)
        cv2.imshow('Recognize & Draw', img)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

i=0
for text in texts:
    if i==0 :
        i=i+1
        license_plate = text.description
        vertices = [(vertex.x, vertex.y)
                    for vertex in text.bounding_poly.vertices]
        cv2.rectangle(img, (vertices[0][0]-10, vertices[0][1]-10), (vertices[2][0]+10,
vertices[2][1]+10), (0, 255, 0), 3)
        cv2.imshow("Recognize & Draw2", img)
        resultArray = [x.split(';') for x in license_plate.split('\n')]
        if 'Logo' in data:
            logo = data['Logo']
            data = {'Logo': logo, 'Number': resultArray[0], 'Province': resultArray[1]}
            df = pd.DataFrame(data)
            try:
                for item in sqldata:
                    print("database", item)
                    print("mydata", data)
                    if item["Logo"] == logo and item["Number"] == resultArray[0][0]
and item["Province"] == resultArray[1][0]:
                        print("!!matched!!")
                        msg = '-----\n'+ logo +"\n"+ str(resultArray[0][0])+"\n"+
str(resultArray[1][0]) +"\n\nตึกภาควัดคุมฯ คณะวิศวกรรมศาสตร์" )
                        line.lineNotify(msg)
                        namefile = "Licenseplate.jpg"
                        cv2.imwrite(namefile, img)
                        cv2.imshow("Licenseplate", img)
                        message = 'Picture License plate' # Set your message here!
                        lineNotify(message)
            except:
                print("unmatched")
        except:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        pass
        df.to_csv("License.csv")

        cv2.waitKey(1)
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break
    cap.release()
    cv2.destroyAllWindows()

print('----- End -----')
```

2.2 โปรแกรมส่วนเชื่อมต่อและเรียกใช้ฐานข้อมูล (database)

```
import sqlite3
import pandas as pd
def getData():
    con = sqlite3.connect("licenseplate.db")
    df = pd.read_sql_query("SELECT * from License", con)
    con.close()
    return df.to_dict(orient="records")
a = getData()
print(a)
```

2.3 โปรแกรมส่วนรับส่งข้อมูล MQTT

```
import paho.mqtt.client as paho
from paho import mqtt
import sqlite3
import time

DATABASE_FILE = 'licenseplate.db'

def on_connect(client, userdata, flags, rc, properties=None):
    print("CONNACK received with code %s." % rc)

def on_publish(client, userdata, mid, properties=None):
    print("mid: " + str(mid))

def on_message(client, userdata, msg):
    data = str(msg.payload.decode('utf-8'))
    x = data.split(",")
    db_conn = sqlite3.connect(DATABASE_FILE)
    sql = 'INSERT INTO License (Logo, Number, Province) VALUES (?, ?, ?)'
    cursor = db_conn.cursor()
    cursor.execute(sql,(x[0],x[1],x[2]))
    db_conn.commit()
    cursor.close()

def main():
    client = paho.Client(client_id="", userdata=None, protocol=paho.MQTTv5)
    client.on_connect = on_connect

    client.tls_set(tls_version=mqtt.client.ssl.PROTOCOL_TLS)
    client.username_pw_set("clubnonoob", "Namblue2543")
    client.connect("b2dbd36cc7d8460e92dede3be3378ee5.s1.eu.hivemq.cloud",
8883)
```

```
client.on_message = on_message
client.on_publish = on_publish
client.subscribe("MQTT/#", qos=1)
client.publish("MQTT/License",payload=",,",qos=1)
client.loop_forever()
main()
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

