กล้องวงจรปิดอัจฉริยะ

Smart CCTV Camera

ชัยภัทร สาขามุละ

Chaiyapath Sakhamula

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องวงจรปิดอัจฉริยะ

Smart CCTV Camera

โดย

ชัยภัทร สาขามุละ

อาจารย์ที่ปรึกษา

ผศ.ดร. ยุทธนา คิดใจเดียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2566

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง กล้องวงจรปิดอัจฉริยะ

Smart CCTV Camera

ผู้จัดทำ นายชัยภัทร สาขามุละ รหัสนักศึกษา 62010187

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ผศ.ดร. ยุทธนา คิดใจเดียว)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน กล้องวงจรปิดอัจฉริยะ
นักศึกษา นายชัยภัทร สาขามุละ รหัสนักศึกษา 62010187
ปริญญา วิศวกรรมศาสตร์บัณฑิต
ภาควิชา วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา 2566
อาจารย์ที่ปรึกษาโครงงาน ผศ.ดร. ยุทธนา คิดใจเดียว

บทคัดย่อ

โปรเจ็คกล้องวงจรปิดอัจฉริยะนี้มี วัตถุประสงค์จัดทำขึ้นเพื่อศึกษาระบบ Computer Vision, Machine learning และ วิธีการเขียนโปรแกรมด้วยภาษา Python ในรูปแบบต่างๆ โดยมีจุดมุ่งหมายในการทำโปรเจ็คคือ การทำกล้องวงจรปิดอัจฉริยะ ที่จะคอยรายงานหากตรวจจับเจอคนแปลกหน้า สามารถบันทึกหน้าของคนที่ต้องการให้จดจำได้ สามารถดูกล้องวงจรปิดแบบเรียลไทม์ได้ผ่านแอพพลิเคชั่น และมีแจ้งเตือนเมื่อตรวจเจอคน โดยจะนำความรู้ที่ได้จากโปรเจ็คครั้งนี้นำไปพัฒนาต่อยอดต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	Smart CCTV Camera
Student	Mr. Chaiyapath Sakhamula Student ID 62010187
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2023
Project Advisor	Assistant Professor Dr. Yuttana Kitjaidure

ABSTRACT

This smart CCTV project is aimed at studying Computer Vision, Machine Learning, and various Python programming techniques. The project's goal is to develop an intelligent CCTV system that can detect unfamiliar faces, record faces for identification, provide real-time CCTV viewing through an application, and send alerts upon detecting people. The knowledge gained from this project can be further utilized for future developments and enhancements.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำโครงงานเรื่อง Smart CCTV Camera ให้สำเร็จในครั้งนี้ได้ ต้องขอขอบคุณอาจารย์ ผศ.ดร. ยุทธนา คิดใจเดียว ที่ได้คอยชี้แนะและให้คำปรึกษาในการแก้ปัญหาต่างๆในการทำโครงงานครั้งนี้

สุดท้ายนี้ข้าพเจ้ารู้สึกขอขอบคุณจากใจในทุกความช่วยเหลือในการทำโครงงานชิ้นนี้ให้สำเร็จขึ้นมาได้ และหวังว่าโครงงานเล่มนี้จะเป็นประโยชน์ต่อผู้อ่านและนำไปต่อยอดในอนาคตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติธรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของโครงงาน	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตการวิจัย	1
1.4 ประโยชน์ที่ได้รับ	1
บทที่ 2 หลักการและทฤษฎี	2
2.1 ภาษาโปรแกรม Python	2
ประวัติของภาษาโปรแกรม Python	3
ข้อดีของภาษาโปรแกรม Python	3
2.2 รายละเอียดของหลักการส่วนที่สำคัญ	4
2.2.1 Machine Learning	4
2.2.2 YOLOv5	5
2.3 Firebase	7
2.4 ESP32-CAM	9
บทที่ 3 วิธีดำเนินการวิจัย	10
3.1 ลักษณะการทำงานของโปรแกรม	10
3.2 แผนผังการทำงานของโปรแกรม	10
3.3 Flowchart ของโปรแกรมตรวจจับ	11
3.4 การใช้งาน Firebase	12
3.5 การใช้งาน YOlOv5	13
3.6 โค้ดของโปรแกรม	14
3.7 โค้ดของ Application	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	บทที่ 4 ผลการทำงาน.....	20
	4.1 ผลการทำงานของโปรแกรมตรวจจับ.....	20
	4.2 ผลการทำงานของ Application.....	23
	4.3 ผลการทำงานของ Firebase.....	25
	บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	26
	5.1 สรุปผลการทดลอง.....	26
	5.2 วิจารณ์ผลการทดลอง.....	26
	เอกสารอ้างอิง.....	27
	ภาคผนวก.....	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 รูปสัญลักษณ์ของภาษา Python	2
2.2 รูปภาพที่ถูกแบ่งออกเป็น n x n grid	5
2.3 รูปภาพที่มีหลาย Object	6
2.4 รูปภาพการทำนายหลายกล่อง	6
2.5 รูปภาพแสดงหลักการคำนวณของ IoU	6
2.6 รูปสัญลักษณ์ของ Firebase	7
2.7 รูปของ ESP32-CAM	9
3.1 รูปแสดงกระบวนการทำงานของระบบ	10
3.2 รูป Flowchart ของโปรแกรมตรวจจับ	11
3.3 รูป Firebase Cloud Storage	12
3.4 รูปภาพโมเดลของ YOlOv5	13
4.1 ภาพจากกล้อง	20
4.2 ภาพการทำงานของโปรแกรมในเบื้องหลัง	20
4.3 โปรแกรมเบื้องหลังตรวจพบคนที่ไม่มีในบันทึก	21
4.4 โปรแกรมส่งแจ้งเตือนไปที่ไลน์เพื่อแจ้งเตือนคนแปลกหน้า	21
4.5 โปรแกรมเบื้องหลังตรวจพบคนที่มีในบันทึก	22
4.6 โปรแกรมส่งแจ้งเตือนไปที่ไลน์เพื่อแจ้งเตือนสมาชิกถึงบ้าน	22
4.7 Application หน้า Real-Time CCTV	23
4.8 Application หน้า Upload	23
4.9 ใส่รูปและชื่อใน Upload	24
4.10 รูป Upload สำเร็จ	24
4.11 รูปแสดงข้อมูลที่ถูกนำมาเก็บไว้ที่ Firebase	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ไม่ว่าจะในยุคสมัยใด ความปลอดภัยเป็นสิ่งสำคัญอย่างมากในชีวิตของผู้คน ถึงแม้จะเป็นบ้านของตัวเองก็อาจจะมีอันตรายได้ หรือแม้จะติดกล้องวงจรปิดไว้แล้ว แต่หากผู้ใช้ไม่ได้นั่งดูกล้องวงจรปิด ก็อาจจะมีคนลักลอบเข้ามาโดยไม่รู้ตัวได้เหมือนกัน ผู้จัดทำโครงงานจึงได้ทำโครงงาน Smart CCTV ขึ้นเพื่อสร้างกล้องวงจรปิดที่สามารถแจ้งเตือนไปที่ตัวผู้ใช้เมื่อมีผู้บุกรุกได้ และในโปรเจ็คนี้ กล่องวงจรปิดสามารถแยกแยะใบหน้าของคนที่บันทึกไว้ในระบบกับคนแปลกหน้าได้ โดยสามารถแจ้งเตือนว่าเป็นใครที่เข้ามาที่บ้านของเรา เพื่อยกระดับความปลอดภัยในชีวิตประจำวันขึ้นไปอีก

โดยในปัจจุบันมีภาษาที่สามารถเขียนเกี่ยวกับ Machine Learning ได้หลากหลาย เช่น Python, MATLAB เป็นต้น ผู้จัดทำโครงงานได้เลือกภาษา Python มาใช้เนื่องจากเล็งเห็นว่ามีผู้ใช้งานที่หลากหลายมากมายในปัจจุบัน ซึ่งสามารถนำไปต่อยอดในอนาคตได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของโครงงาน

1.2.1 เพื่อสร้างกล้องวงจรปิดที่สามารถส่งข้อความแจ้งเตือนไปหาผู้ใช้หากมีคนบุกรุกเข้ามาได้

1.2.2 เพื่อสร้างกล้องวงจรปิดที่สามารถแยกแยะใบหน้าของคนที่มีบันทึกไว้ในระบบกับคนที่ไม่ได้บันทึกไว้ได้

1.2.3 เพื่อฝึกกระบวนการคิดและแก้ปัญหาในการเขียนโค้ด

1.3 ขอบเขตการวิจัย

1.3.1 กล้องวงจรปิดสามารถแยกแยะระหว่างหน้าคนที่บันทึกไว้ในระบบกับไม่มีบันทึกไว้ในระบบได้ ระบบสามารถแจ้งเตือนไปยังผู้ใช้ตามคนที่กล้องตรวจจับได้ โดยสามารถอัพโหลดรูปแบบภาพคนที่ต้องการบันทึกไว้เข้าระบบ และสามารถดูกล้องวงจรปิดแบบ Realtime ได้ผ่านแอพพลิเคชั่น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้ระบบ CCTV ที่สามารถแยกแยะใบหน้าคนที่มีบันทึกไว้ในเบื้องต้นได้

1.4.2 มีความรู้และเข้าใจในหลักการทำงานและทฤษฎีที่เกี่ยวข้องของกับ Machine Learning

1.4.3 สามารถคิดวิเคราะห์และแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นในการเขียนโค้ดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในการทดลองครั้งนี้ได้ศึกษาหลักการและทฤษฎีที่เกี่ยวข้อง ดังนี้

2.1 ภาษาโปรแกรม Python

2.1 รูปสัญลักษณ์ของภาษา Python

ภาษาโปรแกรม Python คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยกรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เราเขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่เราต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติของภาษาโปรแกรม Python

Python ได้เริ่มต้นขึ้นในเดือนธันวาคมปี 1989 โดยนาย Guido van Rossum โปรแกรมเมอร์ชาวดัตช์ ในตอนนั้นทำงานอยู่ที่สถาบันวิจัยแห่งชาติ Centrum Wiskunde & Informatica (CWI) ซึ่งเป็นสถาบันวิจัยทางด้านคณิตศาสตร์และวิทยาการคอมพิวเตอร์ในเมืองอัมสเตอร์ดัม ประเทศเนเธอร์แลนด์ ในเวลานั้น Guido ต้องพัฒนาโปรแกรมสำหรับผู้ดูแลระบบ เพื่อใช้ในโครงการ Amoeba ซึ่งเป็นโครงการเกี่ยวกับระบบปฏิบัติการแบบกระจาย (Distributed operating system) อย่างไรก็ตามเขารู้สึกว่าภาษาโปรแกรม ABC, C และ Bourne shell มีข้อจำกัดมากมาย ทั้งเรื่องใช้เวลาในการพัฒนานานมากและไม่สามารถตอบโจทย์หลายประการ ดังนั้น Guido จึงได้ตัดสินใจเริ่มพัฒนาภาษาโปรแกรมระดับสูงขึ้นมาใหม่เพื่อใช้งานเองเป็นงานอดิเรก โดยนำเอาสิ่งที่ชอบในภาษา ABC มาพัฒนาลงไปในภาษาโปรแกรม Python รวมถึงได้พัฒนาส่วนอื่น ๆ เพิ่มเติมเข้าไป และในเวลาต่อมาจึงได้เผยแพร่ Python 1.0 เวอร์ชันแรกในปี 1994 หากเทียบกับภาษา Java ที่ได้ทำการเผยแพร่เวอร์ชันแรกในปี 1996 จะเห็นได้ว่าภาษา Python มีอายุมากกว่าภาษา Java ถึง 2 ปี

ข้อดีของภาษาโปรแกรม Python

- นักพัฒนาสามารถอ่านและทำความเข้าใจโปรแกรม Python ได้อย่างง่ายดาย เนื่องจากมีไวยากรณ์พื้นฐานเหมือนภาษาอังกฤษ
- Python ทำให้นักพัฒนาทำงานได้อย่างมีประสิทธิภาพมากขึ้น เนื่องจากพวกเขาสามารถเขียนโปรแกรม Python ได้โดยใช้โค้ดน้อยลงเมื่อเปรียบเทียบกับภาษาอื่นๆ อีกมากมาย
- Python มีไลบรารีมาตรฐานขนาดใหญ่ที่มีโค้ดที่ใช้ซ้ำได้สำหรับเกือบทุกงาน ด้วยเหตุนี้ นักพัฒนาจึงไม่ต้องเขียนโค้ดขึ้นใหม่ทั้งหมดโดยนักพัฒนาสามารถใช้ Python ร่วมกับภาษาการเขียนโปรแกรมยอดนิยมอื่นๆ เช่น Java, C และ C++ ได้อย่างง่ายดาย
- ชุมชน Python ในปัจจุบันมีนักพัฒนาที่พร้อมให้การสนับสนุนหลายล้านคนทั่วโลก หากประสบปัญหาสามารถรับการสนับสนุนอย่างรวดเร็วได้จากชุมชนโดยมีแหล่งข้อมูลที่เป็นประโยชน์มากมายบนอินเทอร์เน็ต หากต้องการเรียนรู้ Python ตัวอย่างเช่น สามารถค้นหาวิดีโอ บทแนะนำสอนการใช้งาน เอกสารประกอบ และคู่มือนักพัฒนาได้อย่างง่ายดาย
- Python สามารถใช้งานได้บนระบบปฏิบัติการคอมพิวเตอร์ต่างๆ เช่น Windows, macOS, Linux และ Unix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 รายละเอียดของหลักการส่วนที่สำคัญ

2.2.1 Machine Learning

Machine Learning คือ ส่วนการเรียนรู้ของเครื่องคอมพิวเตอร์ ถูกใช้งานเสมือนเป็นสมองของ AI (Artificial Intelligence) เราอาจพูดได้ว่า AI ใช้ Machine Learning ในการสร้างความฉลาด มักจะใช้เรียกโมเดลที่เกิดจากการเรียนรู้ของปัญญาประดิษฐ์ มนุษย์มีหน้าที่เขียนโปรแกรมให้ AI เรียนรู้จากข้อมูลเท่านั้น

Machine Learning เรียนรู้จากสิ่งที่เราส่งเข้าไปกระตุ้น แล้วจดจำเอาไว้เป็นมันสมอง ส่งผลลัพธ์ออกมาเป็นตัวเลข หรือ code ที่ส่งต่อไปแสดงผล หรือให้เจ้าตัว AI นำไปแสดงการกระทำ Machine Learning เอง สามารถเอาไปใช้งานได้หลายรูปแบบ ต้องอาศัยกลไกที่เป็นโปรแกรม หรือเรียกว่า Algorithm ที่มีหลากหลายแบบ โดยมี Data Scientist เป็นผู้ออกแบบ หนึ่งใน Algorithm ที่ได้รับความนิยมสูง คือ Deep Learning ซึ่งถูกออกแบบมาให้ใช้งานได้ง่าย และประยุกต์ใช้ได้หลายลักษณะงาน อย่างไรก็ตาม ในการทำงานจริง Data Scientist จำเป็นต้องออกแบบตัวแปรต่างๆ ทั้งในตัวของ Deep Learning เอง และต้องหา Algorithm อื่นๆ มาเป็นคู่เปรียบเทียบ เพื่อมองหา Algorithm ที่เหมาะสมที่สุดในการใช้งานจริงการปรับคุณภาพภาพ (Image enhancement) ทำให้ภาพดูชัดเจนมากขึ้น ลดสัญญาณรบกวน ปรับปรุงความคมชัด และปรับแสงให้เหมาะสม เพื่อให้ภาพมีความสวยงามและมีคุณภาพที่ดีขึ้น

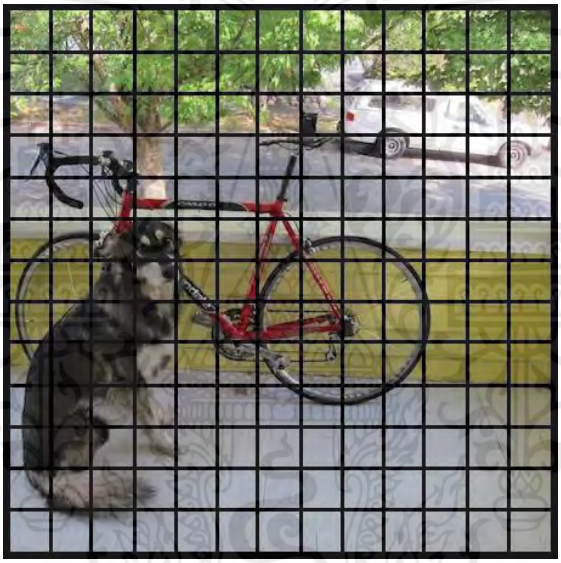
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 YOLOv5

YOLOv5 คือเทคโนโลยีการตรวจจับวัตถุแบบเรียลไทม์ซึ่งเป็นเทคนิคที่ใช้ในการตรวจจับวัตถุจากกล้องวิดีโอ โดยเทคโนโลยีนี้มีการติดตั้งในรถยนต์ไร้คนขับของ Tesla ไปจนถึงร้านค้าปลอดแคชเชียร์ของ Amazon โดย YOLO (“You Only Look Once”) หมายถึง technology การตรวจจับ barcode บนฉลากวัตถุแบบ Realtime ที่ใช้กันอย่างแพร่หลายในปัจจุบัน

YOLO เป็นสถาปัตยกรรมที่ ultralytics ออกแบบไว้เพื่อทำ Image Detection ได้อย่างรวดเร็วและมีประสิทธิภาพ ซึ่งมีข้อดีที่แตกต่างจากการ Detection ตัวอื่นที่ความรวดเร็ว

หลักการทำงานคือ ทำการแบ่งรูปภาพออกเป็น Grid cell ออกมาเป็น n x n grid



2.2 รูปภาพที่ถูกแบ่งออกเป็น n x n grid

ในแต่ละ grid จะต้องใส่ Label เช่น [Pc, bx, by, bh, bw, c1, c2 ,... , cn]

โดยที่ Pc คือ Probability ที่มีวัตถุอยู่ใน Grid นั้นๆ ถ้าไม่มีคือ 0 ถ้ามีคือ 1 bx, by คือตำแหน่งตรงกลางของ Object bh, bw คือขนาดความสูงและกว้างของ Object และ c1,c2,..,cn คือ ผลลัพธ์ว่าเป็น class อะไร ถ้าหากเราต้องการแค่ detect หมา ก็จะมี class เดียว และ grid นั้นมีค่าเป็น 1 แต่หากมีหลาย Object ก็จะมีเลขต่อๆไป

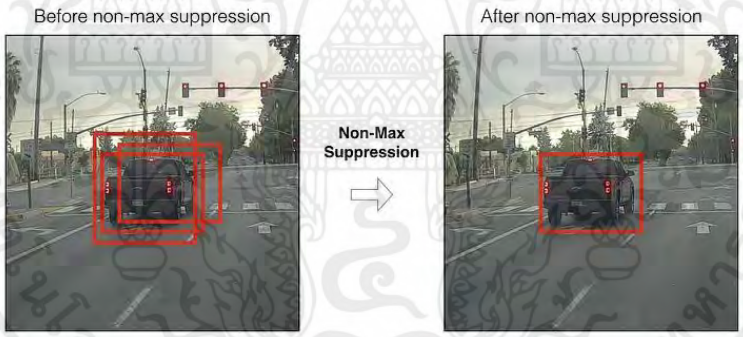
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากในรูปมีอยู่หลาย Object

2.3 รูปภาพที่มีหลาย Object

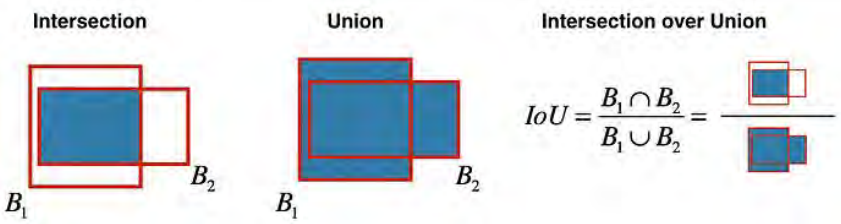
จากรูปภาพ 3.4 ด้านบน จะรับได้เฉพาะ 1 grid คือ 1 Object แต่กรณีถ้ามีหลาย Object จะใช้หลักการที่เรียกว่า Anchor Box โดยสามารถกำหนดจำนวน Box ได้ โดยหากกำหนดเป็น 2 Box ก็จะมี Label แบบด้านบน 2 อันใน 1 Grid และ YOLO จะคำนวณว่ารูปนั้นใกล้เคียงกับ Anchor ใดที่สุดจากค่า IoU ก็จะถูกกำหนดไปที่ Anchor นั้น

IoU สามารถทำนายรูปเดียวกันแต่หลายกล่องได้ เช่น



2.4 รูปภาพการทำนายหลายกล่อง

เราจะใช้ metric IoU ในเลือกกล่องเดียวเป็นตัวแทนของ Object นั้นๆ จากวิธีการ NMS โดยมีหลักการคำนวณ คือ หาส่วนที่ Intersect หารส่วนที่ Union กัน ถ้ากล่องใดมีค่าเกิน Threshold ที่กำหนด จะนับว่าเป็น Object เดียวกัน



2.5 รูปภาพแสดงหลักการคำนวณของ IoU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Firebase

2.6 รูปสัญลักษณ์ของ Firebase

Firebase เป็นแพลตฟอร์มของบริษัท Google ที่มีเครื่องมือและบริการต่างๆ เพื่อช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันแบบ Real-time ได้อย่างง่ายดายและรวดเร็ว บริการหลักของ Firebase ประกอบด้วย Realtime Database, Firestore, Storage, Authentication, Hosting, Functions และ Analytics ซึ่งนักพัฒนาสามารถนำไปใช้งานเพื่อพัฒนาแอปพลิเคชันต่างๆ ได้ง่ายดายและไม่ต้องใช้เวลานานในการจัดการ Server และ Infrastructure ของแอปพลิเคชันด้วยตัวเอง

Firebase เป็นแพลตฟอร์มที่มีความยืดหยุ่นสูง สามารถใช้งานได้กับหลายภาษาโปรแกรมมิ่ง เช่น JavaScript, Swift, Kotlin, Java, Python และอื่นๆ ทำให้นักพัฒนาสามารถพัฒนาแอปพลิเคชันของตนเองได้อย่างสะดวก และ Firebase ยังมี SDK ที่ช่วยให้การใช้งาน Firebase ในแต่ละภาษาเป็นไปอย่างมีประสิทธิภาพ รวดเร็ว และง่ายต่อการเรียนรู้และใช้งาน

Firebase ถือว่าเป็นแพลตฟอร์มที่เป็นที่นิยมสำหรับนักพัฒนาทุกระดับ เนื่องจากมีความเรียบง่าย มีขนาดเล็กและยืดหยุ่น รองรับหลายแพลตฟอร์มและภาษา

Firebase ทำงานอย่างไร?

Firebase ทำงานโดยใช้โครงสร้าง Cloud Platform ของ Google ที่มีเซิร์ฟเวอร์และโปรแกรมคอมพิวเตอร์หลายตัวที่ทำงานร่วมกันเพื่อให้บริการต่างๆ ให้กับผู้ใช้งาน โดย Firebase มีหลากหลายฟีเจอร์สำหรับการใช้งาน ดังนี้

1. **Firebase Console:** เป็นหน้าเว็บแอปพลิเคชันที่ให้ผู้ใช้งานสามารถจัดการและควบคุม Firebase ได้ ผู้ใช้งานสามารถสร้างโปรเจ็กต์ใหม่, ตั้งค่าแอปพลิเคชัน, แก้ไขโค้ด, จัดการข้อมูล, ติดตั้งเครื่องมือต่างๆ และดูข้อมูลสถิติผู้ใช้งานได้

2. **Firebase SDK:** เป็นชุดเครื่องมือที่ Firebase ให้เพื่อช่วยในการพัฒนาแอปพลิเคชัน ซึ่งประกอบด้วย Library ต่างๆ ที่มีไว้สำหรับภาษาต่างๆ เช่น JavaScript, Android, iOS, C++, Unity ฯลฯ ชุดเครื่องมือนี้จะช่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ผู้พัฒนาสามารถเข้าถึงฐานข้อมูล, ส่งการแจ้งเตือน, ตรวจสอบสิทธิ์การเข้าใช้, ส่งอีเมล, โหลดและอัพโหลดไฟล์, และอื่นๆ

3. **Firebase Services:** Firebase มีบริการต่างๆ ที่มีไว้สำหรับผู้ใช้งาน เช่น Firebase Realtime Database, Firebase Cloud Messaging, Firebase Hosting, Firebase Authentication, Firebase Storage, Firebase Cloud Functions, Firebase Test Lab, และอื่นๆ บริการเหล่านี้เป็นส่วนสำคัญที่ช่วยให้ผู้ใช้งานสามารถจัดการข้อมูล, ทำงานร่วมกับเซิร์ฟเวอร์, ส่งการแจ้งเตือน และทดสอบแอปพลิเคชันได้อย่างมีประสิทธิภาพ เพื่อให้ผู้ใช้งานสามารถพัฒนาแอปพลิเคชันได้อย่างสะดวกและรวดเร็วมากยิ่งขึ้น

4. **Firebase Realtime Database:** เป็นฐานข้อมูลแบบ NoSQL ที่ใช้เก็บข้อมูลแบบ Realtime ซึ่งหมายความว่าข้อมูลที่ถูกเพิ่มหรือแก้ไขจะถูกอัพเดทในเวลาเดียวกันกับที่ข้อมูลถูกเปลี่ยนแปลง

5**. Firebase Cloud Messaging:** เป็นบริการส่งการแจ้งเตือน (Push Notification) ซึ่งช่วยให้ผู้ใช้งานได้รับการแจ้งเตือนในเรื่องต่างๆ แม้ว่าแอปพลิเคชันจะไม่ได้เปิดใช้งานอยู่

6. **Firebase Hosting:** เป็นบริการโฮสต์เว็บไซต์และแอปพลิเคชันของ Firebase โดยเป็นการโฮสต์แบบสแตติก ทำให้ผู้ใช้งานไม่จำเป็นต้องมีเซิร์ฟเวอร์เอง

7. **Firebase Authentication:** เป็นบริการสำหรับการตรวจสอบตัวตนของผู้ใช้งาน ช่วยให้ผู้ใช้งานสามารถเข้าสู่ระบบและใช้งานแอปพลิเคชันได้อย่างปลอดภัย

8. **Firebase Storage:** เป็นบริการเก็บข้อมูลแบบ Cloud Storage ที่ช่วยให้ผู้ใช้งานสามารถเก็บไฟล์ต่างๆ ได้โดยไม่ต้องกังวลเรื่องความปลอดภัยของข้อมูล

9. **Firebase Cloud Functions:** เป็นบริการที่ช่วยให้ผู้ใช้งานสามารถเขียนโค้ดแบบ Serverless โดยไม่ต้องมีเซิร์ฟเวอร์เอง โดยเซิร์ฟเวอร์จะถูกเรียกใช้เมื่อมีการเกิดเหตุการณ์บางอย่างเกิดขึ้น

10. **Firebase Performance Monitoring:** ฟีเจอร์นี้ช่วยตรวจสอบและวัดประสิทธิภาพของแอปพลิเคชันของเรา โดยจะวัดตั้งแต่เวลาโหลดหน้าแรกของแอปพลิเคชัน จนถึงเวลาที่ใช้ในการโหลดข้อมูลและประมวลผลต่างๆ ซึ่งจะช่วยให้เราปรับปรุงประสิทธิภาพของแอปพลิเคชันได้

11. **Firebase Test Lab:** ฟีเจอร์นี้ช่วยให้สามารถทดสอบแอปพลิเคชันบนหลายๆ เครื่องมือและอุปกรณ์ได้ ซึ่งรวมถึงการทดสอบแอปพลิเคชันบน Android, iOS, และเว็บไซต์

12. **Firebase App Distribution:** ฟีเจอร์นี้ช่วยให้เราสามารถแบ่งปันแอปพลิเคชันให้กับผู้ใช้งานที่ต้องการทดสอบได้โดยง่าย โดยเราสามารถสร้างลิงก์สำหรับดาวน์โหลดแอปพลิเคชันได้และส่งต่อให้กับผู้ทดสอบได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. **Firebase Extensions:** ตัวช่วยในการเพิ่มความสามารถของ Firebase โดยใช้งานผ่านแพลตฟอร์ม Firebase Console ซึ่งเป็นแอปพลิเคชันเว็บที่จะช่วยให้เราสามารถเชื่อมต่อและใช้งานบริการต่างๆ ของ Firebase ได้อย่างสะดวกสบายและรวดเร็วมากขึ้นโดยไม่ต้องเขียนโค้ดใดๆ

2.4 ESP32-CAM

2.7 รูป ESP322-CAM

ESP32-CAM คืออะไร?

ESP32-CAM คือ โมดูลกล้องขนาดเล็กที่ใช้พลังงานต่ำ ใช้ชิป ESP32 มาพร้อมกับกล้อง OV2640 และมีช่องเสียบ SD Card ในตัว สามารถเชื่อมต่อ WiFi + Bluetooth เพื่อการควบคุมระยะไกลได้

การ ใช้งาน ESP32-CAM สามาถนำไปใช้ได้ตั้งแต่อุปกรณ์ IoT ธรรมดาไปจนถึงขั้นสูงอื่น ๆ สำหรับการตรวจสอบและจดจำใบหน้าโดยใช้ AI และแม้กระทั่งทำเป็นกล้องวงจรปิด

การใช้งาน ESP32-CAM

ESP32-CAM ถูกนำไป ใช้งาน ในหลากหลายรูปแบบ เนื่องจากมีขนาดที่เล็ก ราคาถูก ใช้กำลังไฟน้อย ตัวอย่างการใช้งานที่เห็นกันทั่วไปคือการนำมาทำเป็นกล้องวงจรปิดไร้สาย การทำระบบจดจำใบหน้าผู้คน หรือเป็นส่วนประกอบของหุ่นยนต์ ใช้งานร่วมกับ Arduino ในโครงงานประดิษฐ์ต่างๆ

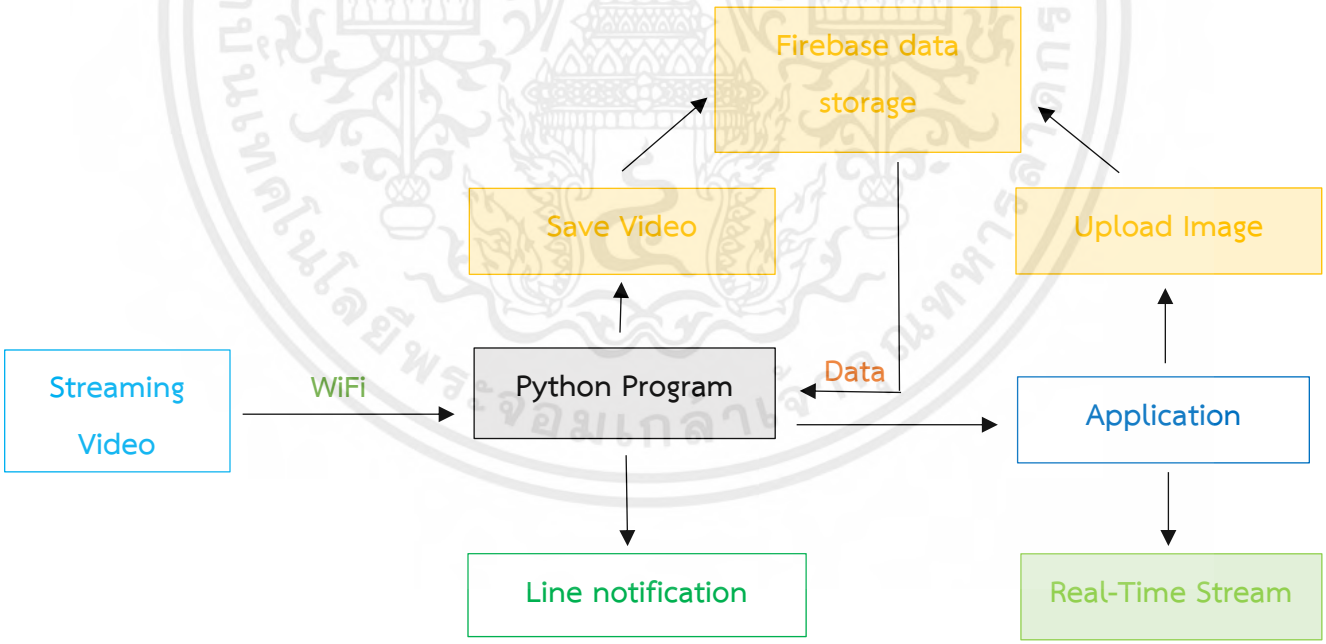
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีดำเนินการวิจัย

3.1 ลักษณะการทำงานของโปรแกรม

โปรแกรมจะเชื่อมกับกล้อง ESP32-CAM ซึ่งทำหน้าที่เป็นกล้องวงจรปิดผ่านไวไฟ หากมีคนเข้ามาในระยะของกล้องวงจรปิด ตัวโปรแกรมจะแยกหน้าของคนที่เข้ามาว่าถูกบันทึกไว้ในระบบหรือไม่ โดยดึงรูปภาพของคนที่ถูกบันทึกไว้ใน Firebase มาทำการตรวจเช็ค หากถูกบันทึกไว้ โปรแกรมจะแจ้งเตือนเป็นชื่อคนที่ถูกบันทึกไว้ในระบบ หากคนที่เข้ามาไม่มีบันทึกไว้ในระบบ โปรแกรมจะส่งสัญญาณเตือนพร้อมกับรูปคนแปลกหน้า ไปที่ไลน์ Notification ของผู้ใช้งานเพื่อแจ้งเตือน ผู้ใช้งานสามารถอัพโหลดรูปคนที่ต้องการบันทึกไว้ในระบบได้ผ่านแอพพลิเคชันซึ่งจะถูกส่งไปเก็บไว้ใน Firebase และสามารถดูกล้องวงจรปิดได้ผ่านแอพพลิเคชันเช่นกัน โดยโปรแกรมจะบันทึกวิดีโอ CCTV ไว้ใน Firebase เผื่อกรณีที่ต้องการดูบันทึกย้อนหลัง

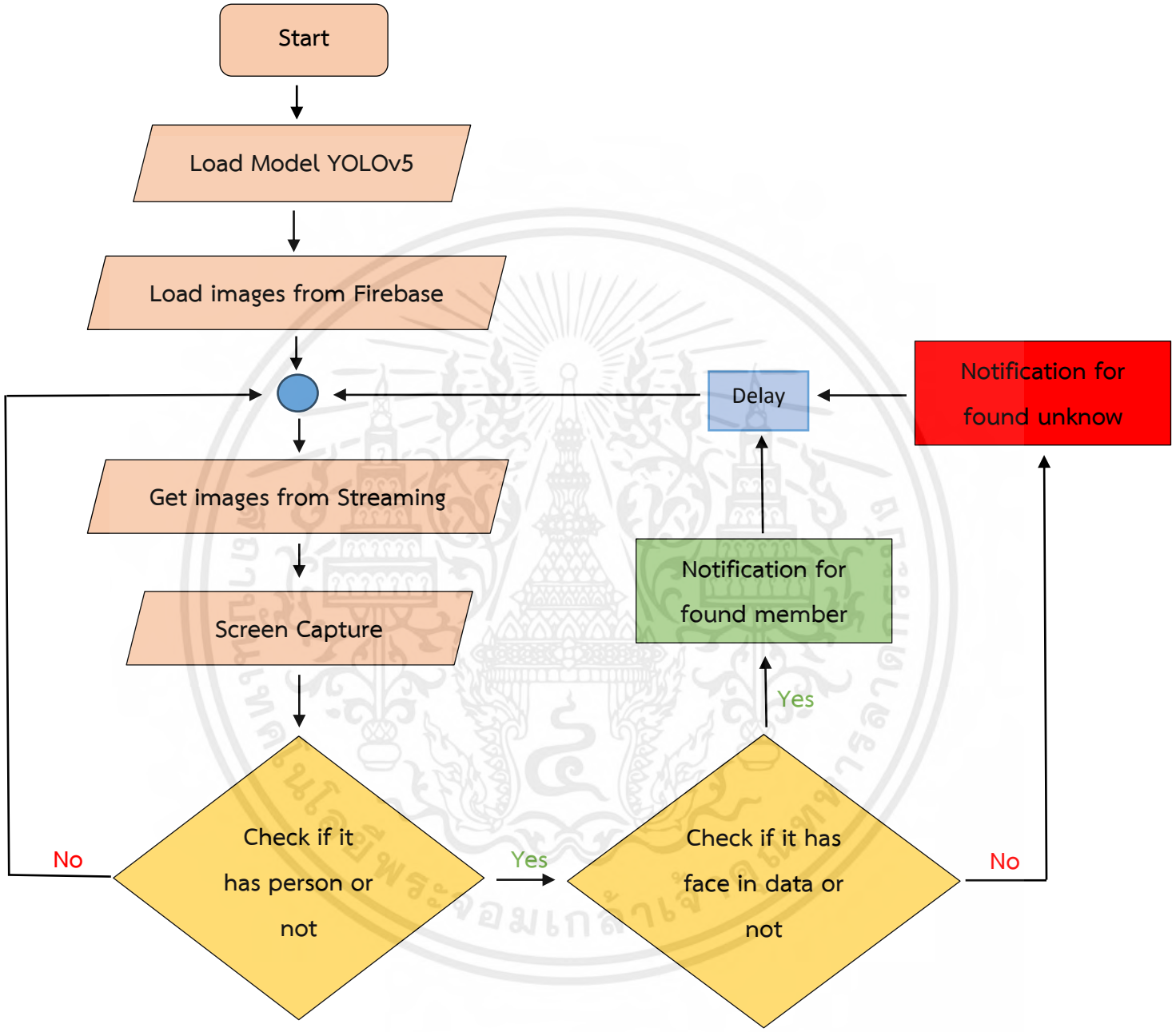
3.2 แผนผังการทำงานของระบบ



3.1 รูปแสดงแผนการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Flowchart ของโปรแกรมตรวจจับ



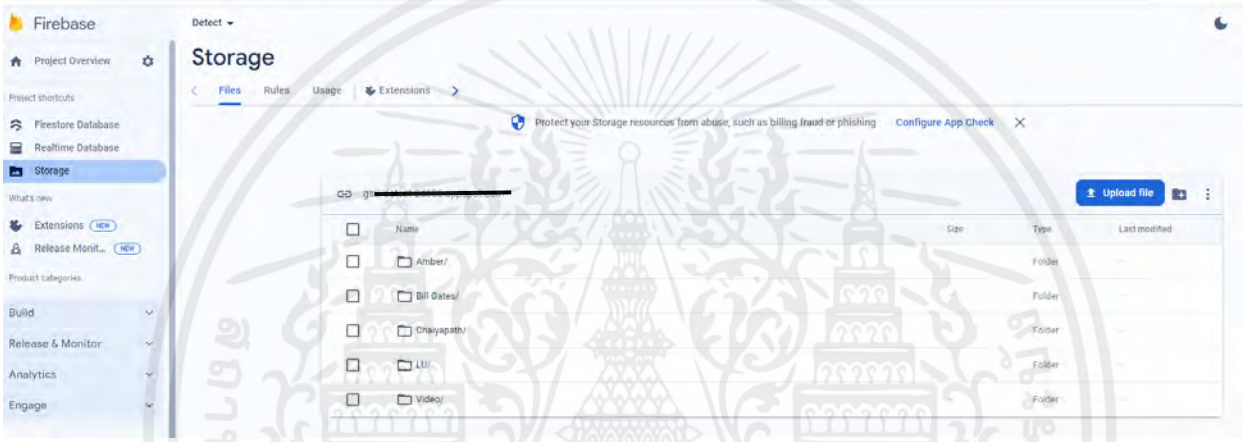
3.2 รูป Flowchart ของโปรแกรมตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การใช้งาน Firebase

```
from firebase_admin import credentials, storage
```

การใช้งาน Firebase ใน Python ถูกดึงมาใช้โดยใช้ Library firebase admin ซึ่งเป็นเครื่องมือที่ใช้ในการจัดการและเชื่อมต่อกับ Firebase services ผ่านโมดูล credentials ซึ่งเป็นการโหลดข้อมูลตรวจสอบสิทธิ์เพื่อให้สามารถเข้าถึง Firebase services ได้และโมดูล storage เพื่อจัดการการเข้าถึง Firebase Cloud Storage โดยในโครงงานนี้ได้ใช้เป็นตัวอ่านและเขียนข้อมูลลงใน Firebase Cloud Storage



3.3 รูป Firebase Cloud Storage

ในเว็บ Firebase ส่วนที่ใช้งานคือ Firebase Cloud Storage ซึ่งเป็นบริการใน Firebase ที่ให้พื้นที่เก็บไฟล์ไว้บน Cloud ได้ โดยสามารถเก็บไฟล์เช่น รูปภาพ วิดีโอ ไฟล์เสียง และเอกสารอื่น ๆ ไว้ในคลังข้อมูลออนไลน์ และทำให้สามารถเข้าถึงไฟล์ได้จากทุกที่ที่มีการเชื่อมต่ออินเทอร์เน็ต

ในโครงงานนี้ได้ใช้ Firebase Cloud Storage ในการเก็บไฟล์ข้อมูลรูปภาพของหน้าคนไว้เพื่อเป็นฐานข้อมูลในการใช้ตรวจจับหน้าบุคคล และเก็บวิดีโอแบ็คอัพไว้ในการดูย้อนหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การใช้งาน YOlOv5

```
import torch

model = torch.hub.load('ultralytics/yolov5', 'yolov5s')
model.classes = 0
```

ในการใช้งาน YOlOv5 ในโครงงานนี้ ได้เรียกใช้โมดูล torch ใน Python โมดูล torch เป็นส่วนหนึ่งของ PyTorch ซึ่งเป็นไลบรารีที่มาพร้อมกับเครื่องมือที่ใช้ในงาน deep learning และการวิเคราะห์ข้อมูล ในส่วนของโครงงานนี้ได้เรียกใช้โมดูลนี้เพื่อดึงโมเดลของ YOlOv5 มาลงใน Python สำหรับตรวจจับบุคคลในกล้อง

การเลือกโมเดลของ YOlOv5

Model	size (pixels)	mAP^{box} 50-95	mAP^{mask} 50-95	Train time 300 epochs A100 (hours)	Speed ONNX CPU (ms)	Speed TRT A100 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n-seg	640	27.6	23.4	80:17	**62.7**	**1.2**	**2.0**	**7.1**
YOLOv5s-seg	640	37.6	31.7	88:16	173.3	1.4	7.6	26.4
YOLOv5m-seg	640	45.0	37.1	108:36	427.0	2.2	22.0	70.8
YOLOv5l-seg	640	49.0	39.9	66:43 (2x)	857.4	2.9	47.9	147.7
YOLOv5x-seg	640	**50.7**	**41.4**	62:56 (3x)	1579.2	4.5	88.8	265.7

3.4 รูปภาพโมเดลของ YOlOv5

รูปภาพที่ 3.7 จะแสดงโมเดลที่มีของ Yolov5 ซึ่งมีความเร็วและความถูกต้องในการตรวจจับที่ต่างกัน โมเดลที่มีความแม่นยำต่ำที่สุดคือ YOLOv5n ซึ่งมีความรวดเร็วที่สุด และโมเดลที่มีความแม่นยำสูงที่สุดคือ YOLOv5x แต่จะใช้เวลาสูงไปด้วยในการตรวจจับ ในโครงงานนี้ผู้จัดทำได้เลือกใช้โมเดล YOLOv5s ที่มีความรวดเร็วสูงเกือบที่สุดและมีความแม่นยำในระดับที่รับได้ โดยกำหนด classes ที่ตรวจจับคือ 0 ซึ่งเป็นการตรวจจับบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 โค้ดของโปรแกรม

```
import cv2
import torch
import os
import time
import requests
import face_recognition
import firebase_admin
from firebase_admin import credentials, storage

cred = credentials.Certificate("your-link.json")
firebase_admin.initialize_app(cred, {
    'storageBucket': 'your-link.com'
})
bucket = storage.bucket()

token = 'your-token'

# Load YOLOv5 model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')
model.classes = 0
model.conf = 0.7

# Define video parameters
frame_width = 640
frame_height = 480
fps = 30

video_dir = "Video"
if not os.path.exists(video_dir):
    os.makedirs(video_dir)


fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('Video/output.avi', fourcc, fps, (frame_width,
frame_height))

def send_line_notification_with_image(token, message, image_path):  # คำสั่งส่งข้อมูลไป
ที่ไลน์ Notification
    headers = {
        "Authorization": f"Bearer {token}"
    }

    data = {
        "message": message
    }

    with open(image_path, 'rb') as img_file:
        files = {"imageFile": img_file}
        response = requests.post('https://notify-api.line.me/api/notify',
headers=headers, data=data,
                                 files=files)

    return response.json()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
def download_images_and_save(folder_path):
    blobs = bucket.list_blobs()
    for blob in blobs:
        filename = blob.name.split('/')[-1]

        save_path = os.path.join(folder_path, filename)

        image_data = blob.download_as_string()

        with open(save_path, 'wb') as file:
            file.write(image_data)

        print(f"Saved image '{filename}' to '{save_path}'")


folder_path = "downloaded_images"

if not os.path.exists(folder_path):
    os.makedirs(folder_path)

download_images_and_save(folder_path)

known_face_encodings = []
known_faces_dir = "downloaded_images"
for filename in os.listdir(known_faces_dir):
    if filename.endswith(".jpg") or filename.endswith(".png"):
        face_image =
face_recognition.load_image_file(os.path.join(known_faces_dir, filename))
        face_encodings = face_recognition.face_encodings(face_image)
        if face_encodings:
            face_encoding = face_encodings[0]
        else:
            continue
        known_face_encodings.append(face_encoding)

url = "your-url-live-video"

cap = cv2.VideoCapture(url)

save_dir = "detected_people"
if not os.path.exists(save_dir):
    os.makedirs(save_dir)

desired_width = 640
scale_factor = None

image_counter = 0
last_saved_timestamp = 0
delay_seconds = 10

frame_skip = 5  # process every 5th frame
frame_count = 0

while True:
    ret, frame = cap.read()
    frame_count += 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if ret:
        # Write the frame to the video file
        out.write(frame)

    if frame is not None and frame_count % frame_skip == 0:
        # Calculate the scaling factor if it's not already determined
        if scale_factor is None:
            scale_factor = desired_width / frame.shape[1]

        frame = cv2.resize(frame, None, fx=scale_factor, fy=scale_factor)

        results = model(frame)

        if len(results.pred[0]) > 0:
            face_locations = face_recognition.face_locations(frame)
            face_encodings = face_recognition.face_encodings(frame,
face_locations)
            name = "Unknown"

            for face_encoding, face_location in zip(face_encodings,
face_locations):
                # Compare with known faces
                matches =
face_recognition.compare_faces(known_face_encodings, face_encoding)

                if True in matches:
                    name = filename.split('.')[0]

                top, right, bottom, left = face_location
                cv2.rectangle(frame, (left, top), (right, bottom), (0, 255,
0), 2)
                cv2.putText(frame, name, (left + 6, bottom - 6),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)

        if (time.time() - last_saved_timestamp) > delay_seconds:
            image_counter += 1
            save_path = os.path.join(save_dir, f"person_{image_counter}.jpg")
            cv2.imwrite(save_path, frame)
            last_saved_timestamp = time.time()
            print(f"Saved detected person image to {save_path}")

            # Send simple message to LINE
            message = f"{name} is home"

            # Send image to LINE
            image_result_path = f"{save_path}"
            response = send_line_notification_with_image(token, message,
image_result_path)

        frame_person = results.render()[0]
        cv2.imshow('Person Detection', frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

out.release()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cap.release()
cv2.destroyAllWindows()

blob = bucket.blob('Video/output.avi')
blob.upload_from_filename('Video/output.avi')

print("Video uploaded to Firebase Storage")
```

3.7 โค้ดของ Application (run on pc)

```
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
from PIL import Image, ImageTk
from firebase_admin import credentials, storage, initialize_app
import cv2

class UploadPage(tk.Frame):
    def __init__(self, parent):
        super().__init__(parent)
        self.parent = parent
        self.grid(row=0, column=0, sticky="nsew")
        self.create_widgets()

    def create_widgets(self):
        # Initialize Firebase
        cred = credentials.Certificate("your-link.json")
        initialize_app(cred, {'storageBucket': 'your-link.com'})

        self.selected_image_path = None
        self.folder_name = None

        self.upload_button = tk.Button(self, text="Upload Image",
command=self.upload_image)
        self.upload_button.pack(side=tk.BOTTOM, padx=10, pady=20)

        self.select_button = tk.Button(self, text="Select Image",
command=self.select_image)
        self.select_button.pack(side=tk.BOTTOM, padx=10)

        self.folder_entry = tk.Entry(self)
        self.folder_entry.pack(side=tk.BOTTOM, pady=10)
        self.folder_label = tk.Label(self, text="Folder Name:")
        self.folder_label.pack(side=tk.BOTTOM, pady=10)

        self.image_label = tk.Label(self)  # Initialize image label here

    def select_image(self):
        self.selected_image_path =
filedialog.askopenfilename(filetypes=[("Image files", "*.jpg;*.jpeg;*.png")])
        if self.selected_image_path:
            self.display_image()

    def display_image(self):
        # Clear any existing image
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        self.clear_image()

        # Open and display the new image
        image = Image.open(self.selected_image_path)
        image.thumbnail((400, 400))
        tk_image = ImageTk.PhotoImage(image)
        self.image_label.config(image=tk_image)
        self.image_label.image = tk_image
        self.image_label.pack()

    def upload_image(self):
        if self.selected_image_path:
            self.folder_name = self.folder_entry.get()
            if self.folder_name:
                bucket = storage.bucket()
                destination =
f"{self.folder_name}/{self.selected_image_path.split('/')[-1]}"
                blob = bucket.blob(destination)
                blob.upload_from_filename(self.selected_image_path)
                messagebox.showinfo("Upload Successful", "Image uploaded
successfully.")
                self.clear_image()
            else:
                messagebox.showerror("Error", "Please enter a folder name.")
        else:
            messagebox.showerror("Error", "Please select an image.")

    def clear_image(self):
        self.image_label.pack_forget()

class LivePage(tk.Frame):
    def __init__(self, parent):
        super().__init__(parent)
        self.parent = parent
        self.grid(row=0, column=0, sticky="nsew")
        self.create_widgets()

    def create_widgets(self):
        # Label to display real-time CCTV text
        self.text_label = tk.Label(self, text="Real time CCTV",
font=("Helvetica", 16))
        self.text_label.pack()

        self.video_label = tk.Label(self)
        self.video_label.pack()

        self.cap = cv2.VideoCapture(0)
        self.show_video()

    def show_video(self):
        ret, frame = self.cap.read()
        if ret:
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            frame = Image.fromarray(frame)
            frame = ImageTk.PhotoImage(image=frame)
            self.video_label.config(image=frame)
            self.video_label.image = frame
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

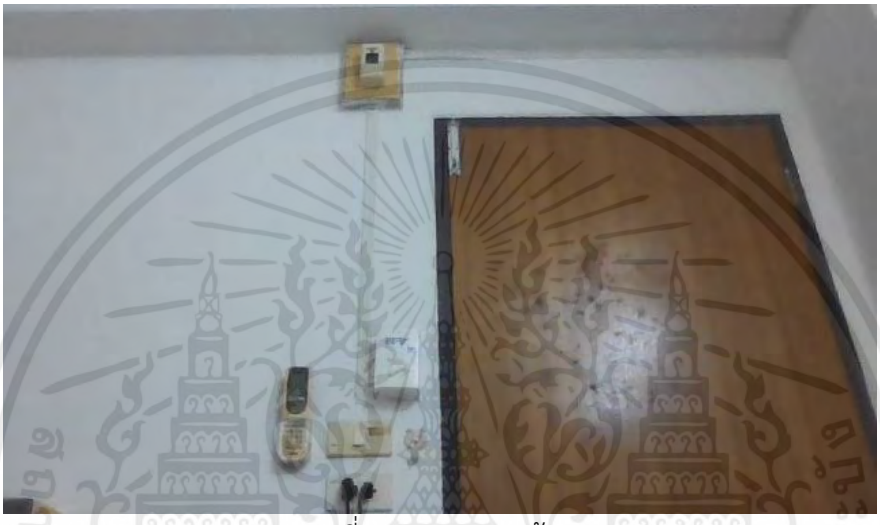
```
            self.video_label.after(10, self.show_video)  # Refresh every 10
milliseconds



class MainPage(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Application")
        self.geometry("500x700")

        self.notebook = ttk.Notebook(self)
        self.notebook.pack(expand=True, fill="both")

        self.live_page = LivePage(self.notebook)
        self.upload_page = UploadPage(self.notebook)

        self.notebook.add(self.live_page, text="Live Stream")
        self.notebook.add(self.upload_page, text="Upload Page")

        self.notebook.select(self.live_page)

if __name__ == "__main__":
    app = MainPage()
    app.mainloop()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

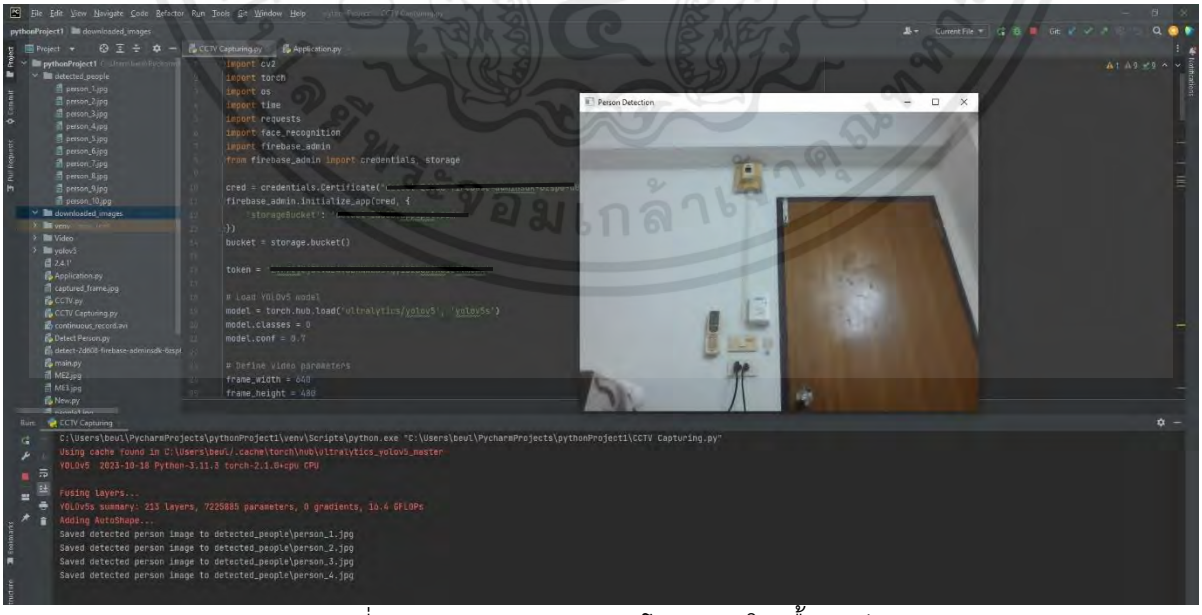
บทที่ 4
ผลการทำงาน

4.1 ผลการทำงานของโปรแกรมตรวจจับ

- การทำงานเบื้องหลัง



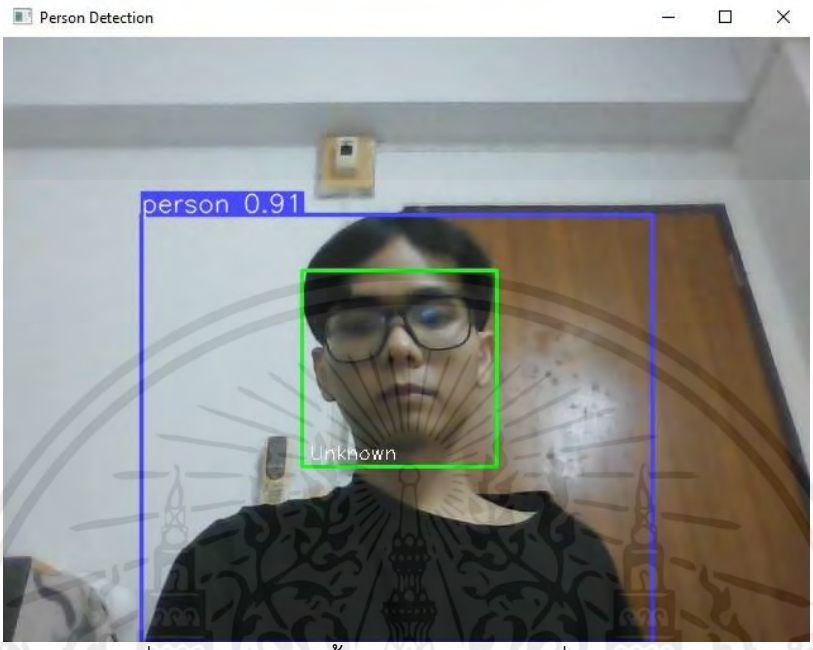
รูปที่ 4.1 ภาพจากกล้อง



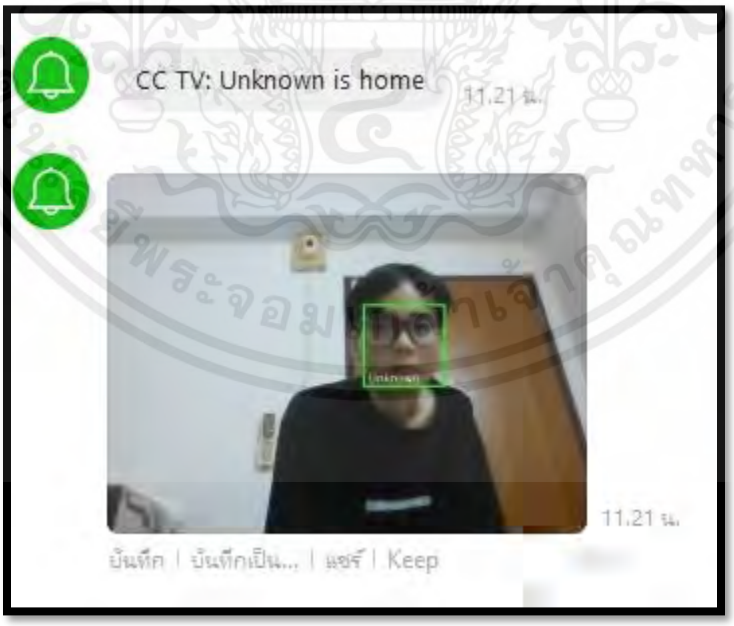
รูปที่ 4.2 ภาพการทำงานของโปรแกรมในเบื้องหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อตรวจเจอคนที่ไม่มีในข้อมูล



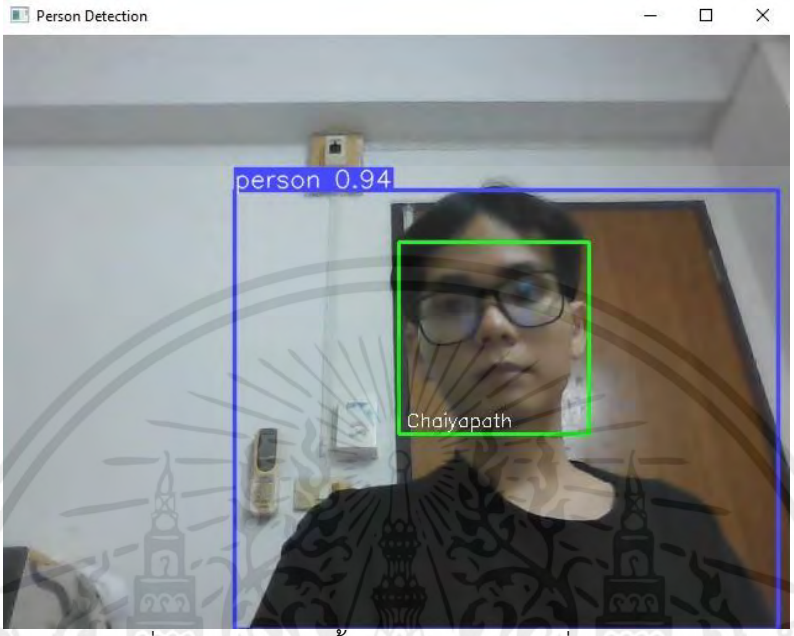
รูปที่ 4.3 โปรแกรมเบื้องหลังตรวจพบคนที่ไม่มีในบันทึก



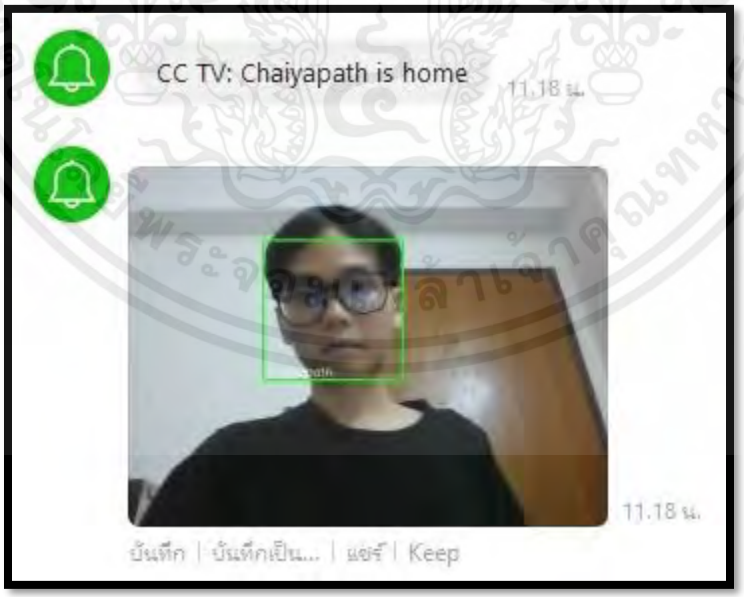
รูปที่ 4.4 โปรแกรมส่งแจ้งเตือนไปที่ไลน์เพื่อแจ้งเตือนคนแปลกหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อตรวจเจอคนที่มีในข้อมูล (เพิ่มข้อมูลเข้าไปแล้ว)



รูปที่ 4.5 โปรแกรมเบื้องหลังตรวจพบคนที่มีในบันทึก



รูปที่ 4.6 โปรแกรมส่งแจ้งเตือนไปที่ไลน์เพื่อแจ้งเตือนสมาชิกถึงบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทำงานของ Application

หน้าแรกเป็นหน้าดู Real-Time CCTV

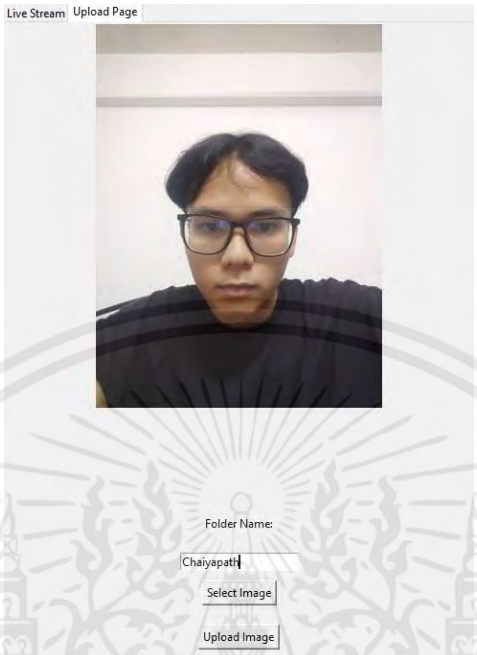
รูปที่ 4.7 Application หน้า Real-Time CCTV

สามารถเข้ามาหน้า Upload เพื่อเพิ่มข้อมูลเข้าไปยัง Firebase ได้

รูปที่ 4.8 Application หน้า Upload

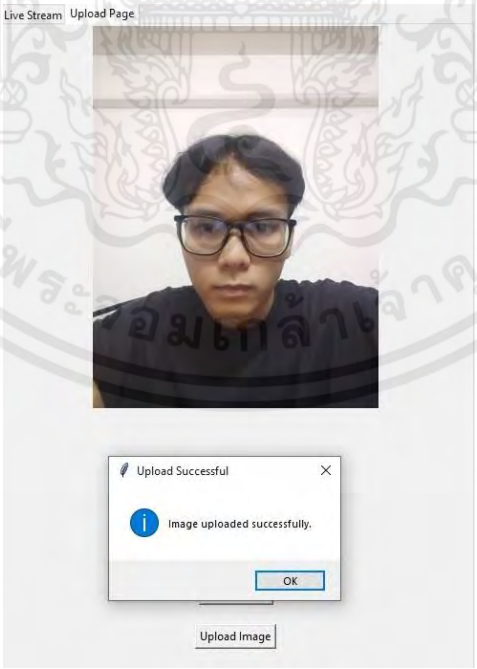
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กด Select Image และใส่รูปภาพ



รูปที่ 4.9 ใส่รูปและชื่อใน Upload

เมื่อกด Upload จะ Upload รูปและชื่อเข้าไปที่ Storage ของ Firebase

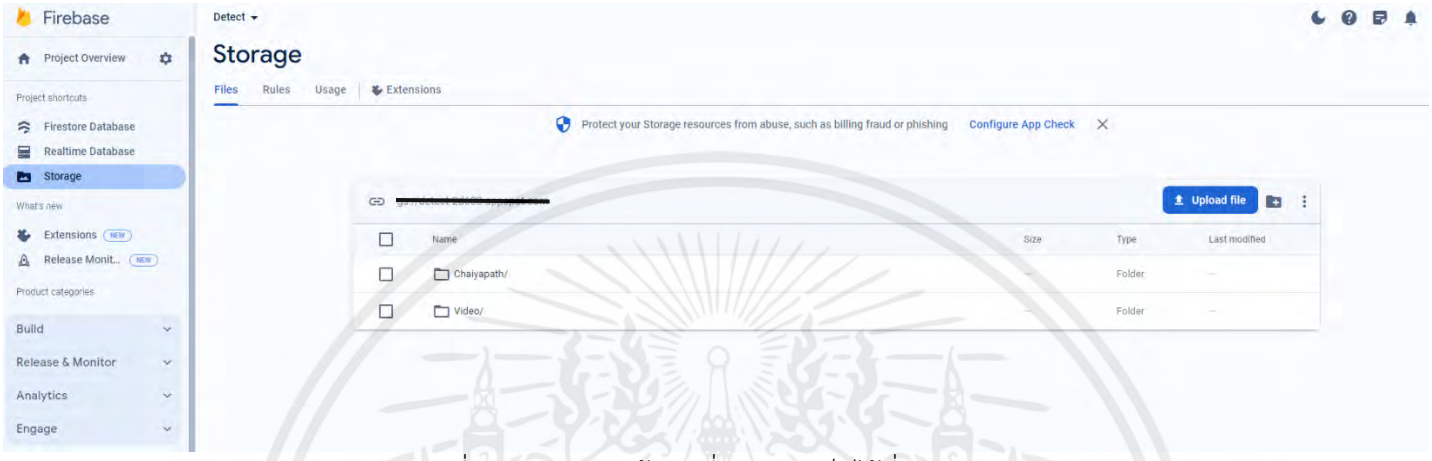


รูปที่ 4.10 รูป Upload สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทำงานของ Firebase

จากที่ Upload มาเมื่อกี้จะถูกเก็บเข้ามาใน Firebase เพื่อใช้เป็นฐานข้อมูลในการตรวจจับใบหน้า และวิดีโอย้อนหลังก็จะถูกเก็บไว้ใน Video/ ด้วยเช่นกัน



รูปที่ 4.11 รูปแสดงข้อมูลที่ถูกนำมาเก็บไว้ที่ Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากผลการทดลอง สามารถสรุปได้ว่า โปรแกรมที่เขียนออกมาเป็นไปตามจุดประสงค์ คือสามารถจับคนที่เข้ามาในระยะกล้องและแยกแยะใบหน้าที่มีในบันทึกฐานข้อมูลได้ โดยโปรแกรมสามารถส่งรูปภาพและชื่อของคนที่ถูกตรวจจับเข้าไปที่ไลน์ของผู้ใช้งานได้ตามที่คาดหวัง

ในส่วนของ Application มีการทำงานเป็นไปตามจุดประสงค์ คือผู้ใช้สามารถเข้ามาเพื่อดูกล้องวงจรปิดแบบ Real-Time ได้ และสามารถอัพโหลดรูปภาพและชื่อเพื่อเพิ่มข้อมูลสมาชิกที่ต้องการเข้าไปเก็บไว้ที่ Firebase ได้ผ่าน Application

5.2 วิจารณ์ผลการทดลอง

โปรแกรมนี้สามารถทำตามจุดประสงค์ตั้งต้นได้ แต่โปรแกรมในตอนนี้อาจจะตรวจจับคนได้ยาก ในกรณีที่มีแสงน้อยมากๆ และในบางมุมอาจจะไม่สามารถระบุได้ว่าสิ่งที่เข้ามาในเฟรมเป็นคนหรือเปล่า จึงต้องทำการติดตั้งกล้องในจุดที่เหมาะสม เพื่อให้โปรแกรมสามารถใช้งานได้อย่างมีประสิทธิภาพ และการแยกแยะใบหน้าของโปรแกรมอาจจะยังไม่เสถียรนัก หากมีรูปภาพของสมาชิกในฐานข้อมูลน้อย หรือคนที่เข้ามาปิดหน้าไว้ตัวโปรแกรมก็จะยังไม่สามารถรู้ได้ว่าเป็นใคร และเนื่องจากโปรแกรมนี้ใช้อินเตอร์เน็ตในการเชื่อมคอมพิวเตอร์กับภาพจากกล้อง ESP-CAM32 และตัวแอพพลิเคชั่น หากอินเตอร์เน็ตถูกตัด โปรแกรมก็จะไม่สามารถทำงานได้เช่นกัน โดยทางผู้จัดทำโครงงานจะนำข้อมูลเหล่านี้ไปปรับปรุงพัฒนาแก้ไขโปรแกรมให้ดียิ่งขึ้นในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ภาษาโปรแกรม Python คืออะไร ?
https://www.9experttraining.com/articles/python-คืออะไร
[2] Python คืออะไร
https://aws.amazon.com/th/what-is/python/
[3] YOLO v5 มันคืออะไร?
https://www.tinpa.or.th/yolov5
[4] Machine Learning คืออะไร
https://www.aware.co.th/machine-learning-คืออะไร/
[5] Firebase เครื่องมือ Must-Have สำหรับธุรกิจในยุคนี้
https://predictive.co.th/blog/firebase/
[6] ใช้งาน ESP32-CAM ง่ายๆ ใน 5 นาที
https://techtalk2apply.com/ใช้งาน-esp32-cam/
[7] ESP32-CAM Datasheet
https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf
[8] Image detection โดยใช้ YOLOv5 จากต้นจนจบ (ตอน 1)
https://medium.com/@intouchkunakorntum/image-detection-โดยใช้-yolov5-จากต้นจนจบ-ตอน-1-77faf04b0579
[9] Releases · ultralytics/yolov5
https://github.com/ultralytics/yolov5/releases

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

DFROBOT
DRIVE THE FUTURE

ESP32-CAM Development Board

SKU:DFR0602

INTRODUCTION

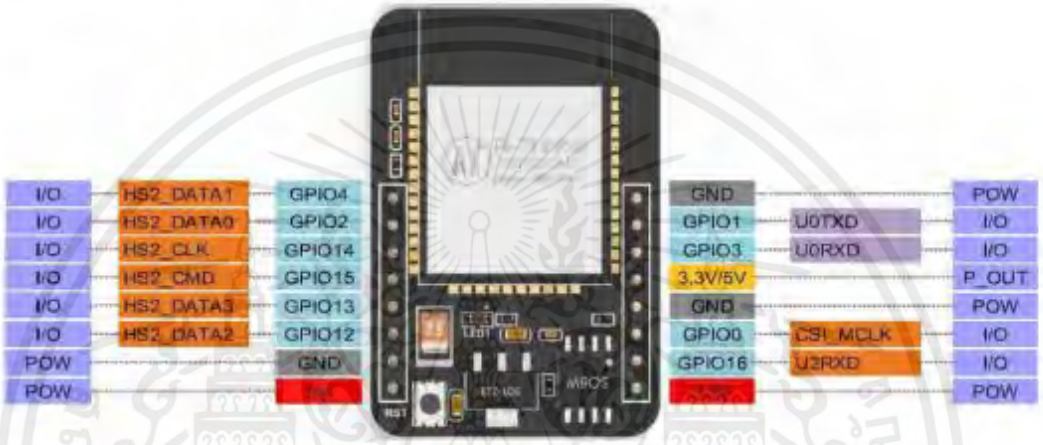
ESP32-CAM is a low-cost ESP32-based development board with onboard camera, small in size. It is an ideal solution for IoT application, prototypes constructions and DIY projects.

The board integrates WiFi, traditional Bluetooth and low power BLE , with 2 high-performance 32-bit LX6 CPUs. It adopts 7-stage pipeline architecture, on-chip sensor, Hall sensor, temperature sensor and so on, and its main frequency adjustment ranges from 80MHz to 240MHz.

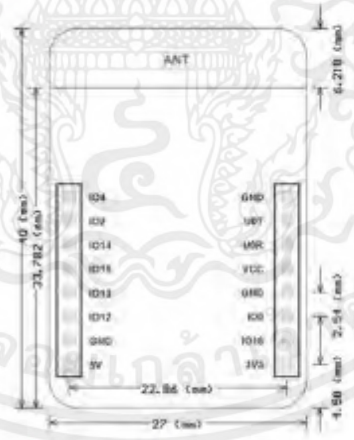
Fully compliant with WiFi 802.11b/g/n/e/i and Bluetooth 4.2 standards, it can be used as a master mode to build an independent network controller, or as a slave to other host MCUs to add networking capabilities to existing devices

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ESP32-CAM can be widely used in various IoT applications. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications. It is an ideal solution for IoT applications.



Schematic Diagram



Dimension Diagram

Notes:
1.Please be sure that the power supply for the module should be at least 5V 2A, otherwise maybe there would be water ripple appearing on the image.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ESP32 GPIO32 pin is used to control the power of the camera, so when the camera is in working, pull GPIO32 pin low.

3.Since IO pin is connected to camera XCLK, it should be left floating in using, and do not connect it to high/low level.

4.The product has been equipped with default firmware before leaving the factory, and we do not provide additional ones for you to download. So, please be cautious when you choose to burn other firmwares.

FEATURES

- Up to 160MHz clock speed, Summary computing power up to 600 DMIPS
- Built-in 520 KB SRAM, external 4MPSRAM
- Supports UART/SPI/I2C/PWM/ADC/DAC
- Support OV2640 and OV7670 cameras, Built-in Flash lamp.
- Support image WiFI upload
- Support TF card
- Supports multiple sleep modes.
- Embedded Lwip and FreeRTOS
- Supports STA/AP/STA+AP operation mode
- Support Smart Config/AirKiss technology
- Support for serial port local and remote firmware upgrades (FOTA)

SPECIFICATION

- SPI Flash: default 32Mbit
- RAM: built-in 520 KB+external 4MPSRAM
- Dimension: 27*40.5*4.5 (±0.2) mm/1.06*1.59*0.18"
- Bluetooth: Bluetooth 4.2 BR/EDR and BLE standards
- Wi-Fi: 802.11b/g/n/e/i
- Support Interface: UART, SPI, I2C, PWM
- Support TF card: maximum support 4G
- IO port: 9
- Serial Port Baud-rate: Default 115200 bps

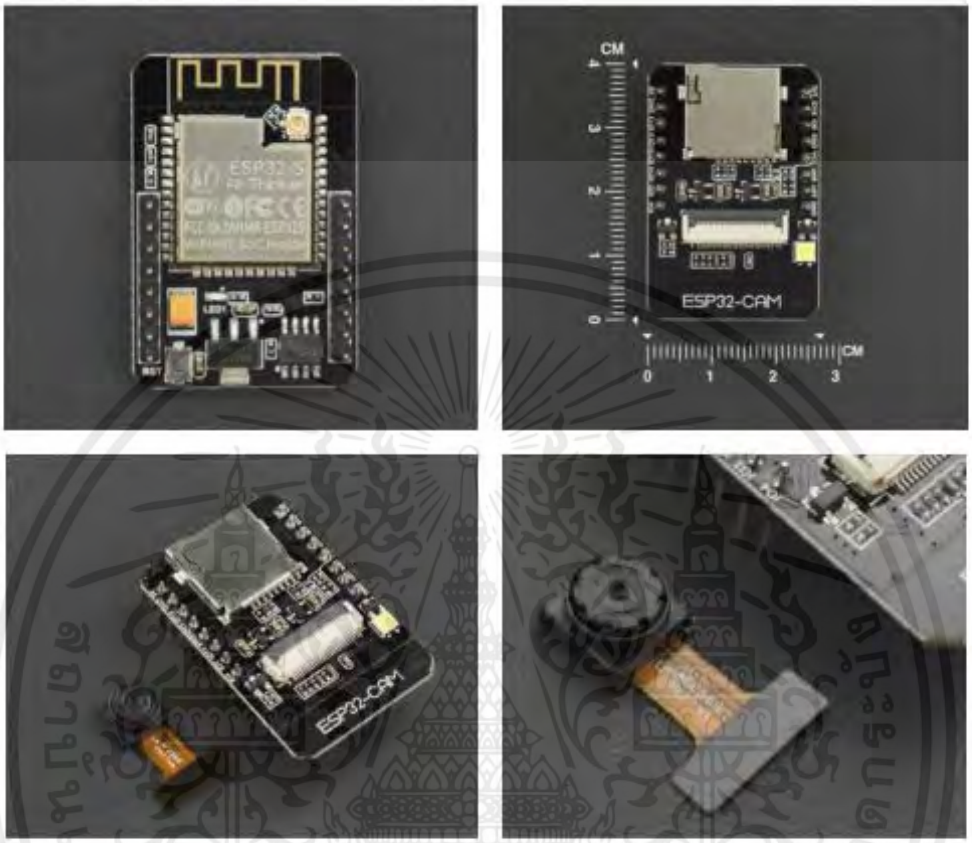
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Image Output Format: JPEG(OV2640 support only), BMP, GRAYSCALE
- Spectrum Range: 2412 ~2484MHz
- Antenna: onboard PCB antenna, gain 2dBi
- Transmit Power: 802.11b: 17±2 dBm (@11Mbps);
  802.11g: 14±2 dBm (@54Mbps);
  802.11n: 13±2 dBm (@MCS7)
- Receiving Sensitivity: CCK, 1 Mbps : -90dBm;
  CCK, 11 Mbps: -85dBm;
  6 Mbps (1/2 BPSK): -88dBm;
  54 Mbps (3/4 64-QAM): -70dBm;
  MCS7 (65 Mbps, 72.2 Mbps): -67dBm
- Power consumption: Turn off the flash: 180mA@5V
  Turn on the flash and adjust the brightness to the maximum: 310mA@5V
  Deep-sleep: the lowest power consumption can reach 6mA@5V
  Moderm-sleep: up to 20mA@5V
  Light-sleep: up to 6.7mA@5V
- Security: WPA/WPA2/WPA2-Enterprise/WPS
- Power supply range: 5V
- Operating temperature: -20 °C ~ 85 °C
- Sorage environment: -40 °C ~ 90 °C, < 90%RH
- Weight: 10g

SHIPPING LIST

- ESP32-CAM Development Board x1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



https://www.dfrobot.com/product-1876.html/8-15-19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้