

เทคโนโลยีจดจำใบหน้าเพื่อการจัดการการเข้าออกงาน  
FACIAL RECOGNITION TECHNOLOGY FOR ATTENDANCE MANAGEMENT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีจดจำใบหน้าเพื่อการจัดการการเข้าออกงาน  
FACIAL RECOGNITION TECHNOLOGY FOR ATTENDANCE MANAGEMENT



โดย  
นางสาวนันทรัตน์ สุขสงเคราะห์ 64010429  
นายบุรพา ชาวพัททอง 64010465  
นายปกป้อง โคตรอาษา 64010472

อาจารย์ที่ปรึกษา  
ศ. ดร.ชวรงค์ พงศ์เจริญพาณิชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2567

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เทคโนโลยีจดจำใบหน้าเพื่อการจัดการการเข้าออกงาน

FACIAL RECOGNITION TECHNOLOGY FOR ATTENDANCE MANAGEMENT

ผู้จัดทำ

1. นางสาวนันทรัตน์ สุขสงเคราะห์ 64010429
2. นายบุรพา ชาวพัททอง 64010465
3. นายปกป้อง โคตรอาษา 64010472

(ศ. ดร. ชวงค์ พงศ์เจริญพาณิชย์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

การจัดทำปฏิญานิพนธ์เรื่อง “เทคโนโลยีจดจำใบหน้าเพื่อการจัดการการเข้าออกงาน” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความอนุเคราะห์อย่างดียิ่งจากอาจารย์ที่ปรึกษา ศ. ดร.ชวรงค์ พงศ์เจริญพาณิชย์ ที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษา ค้นคว้าวิจัยให้ปฏิญานิพนธ์นี้สำเร็จสมบูรณ์ยิ่งขึ้น รวมถึงสนับสนุนสถานที่ เครื่องมือ และอุปกรณ์ต่างๆ ที่ใช้ระหว่างการจัดทำปฏิญานิพนธ์

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอนประสิทธิ์ประสาทวิชา ความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษามีค่ายิ่งแก่ผู้จัดทำ

นางสาวนันทรัตน์ สุขสงเคราะห์  
นายบูรพา ชาวพัททอง  
นายปกป้อง โคตรอาษา  
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีจดจำใบหน้าเพื่อการจัดการการเข้าออกงาน  
 FACIAL RECOGNITION TECHNOLOGY FOR  
 ATTENDANCE MANAGEMENT

โดย นางสาวนันทรัตน์ สุขสงเคราะห์ 64010429  
 นายบูรพา ชาวพัททอง 64010465  
 นายปกป้อง โคตรอาษา 64010472

อาจารย์ที่ปรึกษา ศ. ดร. ชวงค์ พงศ์เจริญพาณิชย์

บทคัดย่อ

ปฏิญานีพนธ์นี้นำเสนอเทคโนโลยีจดจำใบหน้าเพื่อการจัดการการเข้าออกงานที่ช่วยเพิ่มความปลอดภัยและลดข้อผิดพลาดจากการใช้บัตรพนักงานหรือการลงชื่อเข้าใช้งานด้วยตนเอง โดยใช้ ESP32-CAM สำหรับตรวจจับใบหน้าและเซ็นเซอร์วัดอุณหภูมิอินฟราเรดเพื่อยืนยันตัวตนพร้อมเชื่อมต่อกับ Firebase เพื่อจัดเก็บข้อมูลแบบเรียลไทม์ ระบบถูกพัฒนาโดยใช้เทคนิคการเรียนรู้เชิงลึกเพื่อเพิ่มความแม่นยำในการตรวจจับและป้องกันการปลอมแปลงตัวตน อีกทั้งยังสามารถนำข้อมูลที่ได้ไปใช้ในกระบวนการบริหารจัดการบุคลากรภายในองค์กรให้มีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ABSTRACT

This thesis presents a facial recognition technology for attendance management, enhancing security and reducing errors associated with employee ID cards or manual sign-ins. The system utilizes an ESP32-CAM for face detection and an infrared temperature sensor for identity verification, with real-time data storage via Firebase. Deep learning techniques are employed to improve detection accuracy and prevent identity spoofing. Additionally, the collected data can be leveraged to enhance personnel management processes within organizations.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	X
<b>บทที่ 1</b>	
<b>บทนำ</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญาานิพนธ์	1
<b>บทที่ 2</b>	
<b>ทฤษฎีและหลักการที่เกี่ยวข้อง</b>	<b>2</b>
2.1 Arduino ESP32-CAM	2
2.2 FTDI	3
2.3 โมดูลกล้องอินฟราเรดวัดอุณหภูมิ (AMG8853)	4
2.4 จอ LCD	5
2.5 การประมวลผลภาพ (Image Processing)	6
2.6 FaceNet	7
2.7 Haar Cascade	9
2.8 MTCNN (Multi-task Cascaded Convolutional Networks)	10
2.9 Support Vector Machine (SVM)	11
2.10 K-nearest Neighbors	12
2.11 Random forest	13
2.12 Gradient Boosting	13
2.13 การทำอินเตอร์โพลชันแบบไบลิเนียร์ (Bilinear interpolation)	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.14 ไลบรารี OpenCV	15
2.15 Firebase	16
2.16 MQTT (Message Queuing Telemetry Transport)	18
2.17 ระบบเว็บไซต์	20
2.18 Vue.js	23
2.19 SketchUp	25
2.20 UltiMakerCura	25
<b>บทที่ 3 การออกแบบและการจัดทำปริญญานิพนธ์</b>	<b>26</b>
3.1 การออกแบบ	26
3.2 เครื่องมือที่ใช้ในการทดลอง	52
3.3 การจัดเก็บผลการทดลอง	52
<b>บทที่ 4 ผลการทดลอง</b>	<b>54</b>
4.1 การเลือกใช้ Model และจำนวนขนาดข้อมูล	54
4.2 การทดสอบตรวจสอบจับใบหน้า	55
4.3 การทดสอบการส่งข้อมูลอุณหภูมิ	57
4.4 การทดสอบแผนที่ความร้อน (Temperature heat map)	58
4.5 การทดสอบระบบลงเวลา	60
4.6 การทดสอบเว็บแอปพลิเคชัน	61
4.7 การนำข้อมูลมาวิเคราะห์ทางสถิติ	64
<b>บทที่ 5 สรุปผลและข้อเสนอแนะ</b>	<b>65</b>
5.1 สรุปผล	65
5.2 ปัญหาและข้อเสนอแนะ	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
บรรณานุกรม	67
ภาคผนวก ก โปรแกรมสำหรับรับภาพและตรวจจับอุณหภูมิ	70
ภาคผนวก ข โปรแกรมสำหรับตรวจจับใบหน้าและลงเวลาการเข้าออก	79
ภาคผนวก ค โปรแกรมสำหรับเตรียมข้อมูลใบหน้า	88
ภาคผนวก ง โปรแกรมสำหรับเทรนโมเดล	92



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่		หน้า
2.1	ESP32-CAM	2
2.2	FTDI	4
2.3	โมดูลกล้องอินฟาเรดวัดอุณหภูมิ (AMG8853)	4
2.4	รูปแบบของอาร์เรย์ 8x8 พิกเซล	5
2.5	จอ LCD 20x4	5
2.6	ข้อมูลแบบพิกเซล	7
2.7	ตัวอย่างการใช้ MTCNN ตรวจสอบใบหน้า	11
2.8	การขีดเส้นแบ่งกลุ่มข้อมูล	11
2.9	ตัวอย่างการใช้ Haar Cascade ร่วมกับ SVM	12
2.10	ไลบรารี OpenCV	15
2.11	บริการบน Firebase	16
2.12	MQTT Clients	18
2.13	ภาษา HTML	21
2.14	ภาษา CSS	21
2.15	ภาษา JavaScript	22
2.16	Vue.js	23
2.17	SketchUp	25
2.18	UltiMaker Cura	25
3.1	บล็อกไดอะแกรมของระบบจดจำใบหน้าเพื่อการจัดการการเข้าออกงาน	26
3.2	การออกแบบการเชื่อมต่ออุปกรณ์บน Fritzing	27
3.3	การเชื่อมต่ออุปกรณ์	27
3.4	คำสั่งการสร้างฐานข้อมูลบุคคล	28
3.5	การแสดง URL สำหรับเข้าถึงภาพบน Serial Terminal	29
3.6	คำสั่งการนำเข้ารูปภาพมายังโปรแกรมประมวลผล	29
3.7	คำสั่งการบันทึกและเก็บภาพ	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.8 ตัวอย่างและรายละเอียดของรูป	31
3.9 คำสั่งสำหรับการเตรียมข้อมูลใบหน้า	32
3.10 คำสั่งการเทรนโมเดล	33
3.11 คำพารามิเตอร์แต่ละโมเดลที่ใช้	33
3.12 คำสั่งการโหลดโมเดลและเตรียมข้อมูล	35
3.13 คำสั่งการเชื่อมต่อ Firebase	35
3.14 คำสั่งตรวจสอบการขาดงาน	36
3.15 คำสั่งการลงเวลา	37
3.16 คำสั่งการกำหนดค่าคงที่ (Constants)	38
3.17 คำสั่งการตรวจจับใบหน้าและตรวจสอบอุณหภูมิ	39
3.18 คำสั่งฟังก์ชัน bilinear_interpolation	40
3.19 (a) Heat Map ของอุณหภูมิที่วัดจากเซ็นเซอร์ AMG 8853	40
(b) Heat Map ของอุณหภูมิที่ผ่านฟังก์ชัน bilinear_interpolation	
3.20 คำสั่งการตั้งค่ากล้องและ MQTT	41
3.21 คำสั่งการประมวลผลใบหน้าและอุณหภูมิ	42
3.22 คำสั่งการแสดงผลและปิดโปรแกรม	42
3.23 คำสั่งการแสดง Classification Report	43
3.24 คำสั่งคำนวณ Accuracy score	44
3.25 คำสั่งคำนวณ Cross-Validation Mean Accuracy	44
3.26 หน้าเข้าสู่ระบบ (Login)	45
3.27 เข้าสู่ระบบด้วย E-mail ผู้ดูแลระบบ (Admin)	46
3.28 เข้าสู่ระบบด้วย E-mail พนักงาน	46
3.29 หน้าแดชบอร์ด (Dashboard)	47
3.30 หน้าผู้ใช้งาน (User)	48
3.31 หน้ารายละเอียดของพนักงาน (Personal)	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญรูป (ต่อ)

รูปที่	หน้า	
3.32	ฟังก์ชันการเข้าสู่ระบบ	49
3.33	ฟังก์ชันการดึงข้อมูลทั้งหมดบน Firebase	50
3.34	การออกแบบกล่องใน Sketchup	50
3.35	โมเดลกล่องใน UltiMaker Cura	51
3.36	อุปกรณ์หลังประกอบ	51
3.37	โครงสร้างของ employees	53
3.38	โครงสร้างของ attendance	53
4.1	ผล Accuracy Score	54
4.2	การเปรียบเทียบอุณหภูมิพื้นหลังกับใบหน้าของคนจริง	56
4.3	การเปรียบเทียบอุณหภูมิพื้นหลังกับใบหน้าของภาพถ่าย	56
4.4	การตรวจจับใบหน้าและยืนยันตัวตน	57
4.5	จอแสดงสถานะ	57
4.6	การส่งข้อมูลอุณหภูมิเป็นชิ้นๆ (Chunk)	58
4.7	การแสดงผลที่ความร้อนโดยที่มีใบหน้า	59
4.8	การแสดงผลที่ความร้อนโดยที่ใบหน้าโดนบังไว้	60
4.9	ตัวอย่างข้อมูลบน Firebase	61
4.10	เข้าสู่ระบบด้วย E-mail ผู้ดูแลระบบ (Admin)	62
4.11	เข้าสู่ระบบด้วย E-mail พนักงาน	62
4.12	โครงสร้างของ employees	63
4.13	โครงสร้างของ attendance	63
4.14	หน้าแดชบอร์ด	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
4.1 ตารางแสดงผล Accuracy Score และ Cross-Validation Mean Accuracy	55



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันองค์กรต่าง ๆ ต่างให้ความสำคัญกับระบบการจัดการเวลาพนักงานที่มีประสิทธิภาพ ระบบการลงเวลาแบบดั้งเดิมเช่นการใช้บัตรหรือลายนิ้วมือ อาจมีข้อจำกัดด้านความสะดวก รวดเร็ว และความปลอดภัย

ระบบสแกนหน้าลงเวลาเข้า-ออกจึงถูกพัฒนาขึ้นเพื่อตอบสนองความต้องการดังกล่าว โดยใช้เทคโนโลยีการจดจำใบหน้าที่ทันสมัยช่วยให้การลงเวลาเข้า-ออกของพนักงานรวดเร็ว ง่ายตาย ปลอดภัย และแม่นยำยิ่งขึ้น

### 1.2 วัตถุประสงค์

- 1) เพื่อพัฒนาระบบสแกนใบหน้าที่มีประสิทธิภาพ รวดเร็ว และแม่นยำ
- 2) เพื่อตรวจสอบการเข้า-ออกสถานที่ทำงานของพนักงาน
- 3) เพื่อออกแบบและพัฒนาระบบสแกนใบหน้าที่ใช้ต้นทุนต่ำ

### 1.3 ขอบเขตของปริญญาานิพนธ์

ระบบสามารถจดจำและระบุตัวบุคคลได้อย่างแม่นยำไม่น้อยกว่า 80% พร้อมระบุเวลา ตรวจจับความรอนใบหน้าเพื่อยืนยันตัวตนป้องกันการนำรูปถ่ายมาแอบอ้าง และค้นหาข้อมูลย้อนหลังจากฐานข้อมูลเพื่อใช้ในการวิเคราะห์ทางสถิติได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

#### 2.1 Arduino ESP32-CAM

ESP32-CAM เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ผสมผสานกล้องและการสื่อสารแบบ Wi-Fi และ Bluetooth ในตัว ทำให้สามารถใช้งานในโปรเจกต์ที่ต้องการการถ่ายภาพและส่งภาพผ่านเครือข่ายไร้สายได้อย่างมีประสิทธิภาพ โดยคุณสมบัติหลัก ๆ ของ ESP32-CAM [1] [2] แสดงดังรูปที่ 2.1 โดยมีส่วนประกอบหลักดังนี้



รูปที่ 2.1 ESP32-CAM

##### 2.1.1 ชิปรประมวลผล ESP32

ESP32 เป็นไมโครคอนโทรลเลอร์จากบริษัท Espressif ที่ออกแบบมาให้มีความสามารถในการประมวลผลสูง ใช้สถาปัตยกรรมแบบ Dual-core 32-bit Xtensa LX6 ทำให้รองรับการประมวลผลที่มีประสิทธิภาพทำให้สามารถทำงานได้พร้อมกันสองงาน รวมถึงมีการจัดการพลังงานที่ดี เหมาะสำหรับการใช้งานในโปรเจกต์ที่ต้องการความรวดเร็วและประหยัดพลังงาน

##### 2.1.2 กล้อง OV2640

ESP32-CAM มาพร้อมกับกล้อง OV2640 ซึ่งรองรับการถ่ายภาพที่มีความละเอียดสูงสุดถึง 1600x1200 พิกเซล (UXGA) กล้อง OV2640 นี้ทำให้ ESP32-CAM เหมาะสำหรับงานที่ต้องการการจับภาพ เช่น การตรวจจับวัตถุ การตรวจจับใบหน้า และการถ่ายภาพตามเวลาจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3 หน่วยความจำ

ESP32-CAM มีหน่วยความจำ SRAM ขนาด 520 KB และสามารถเพิ่มการ์ด microSD ได้สูงสุด 4 GB เพื่อเก็บข้อมูลที่ต้องการหรือไฟล์ภาพในการใช้งาน ทำให้สามารถบันทึกข้อมูลในพื้นที่เก็บข้อมูลภายในของบอร์ดได้ในกรณีที่การเชื่อมต่อเครือข่ายไม่พร้อมใช้งาน

### 2.1.4 การเชื่อมต่อไร้สาย

ด้วยการรองรับการเชื่อมต่อ Wi-Fi และ Bluetooth ESP32-CAM สามารถใช้งานในเครือข่าย IoT ได้อย่างง่ายดาย ไม่ว่าจะเป็นการส่งภาพไปยังเซิร์ฟเวอร์บนคลาวด์ หรือการควบคุมและรับข้อมูลจากระยะไกลผ่านแอปพลิเคชันบนสมาร์ทโฟน

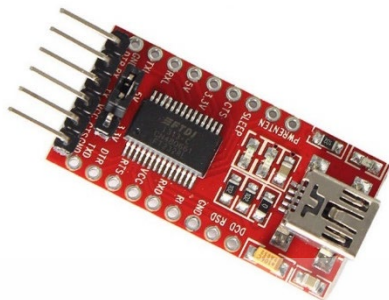
### 2.1.5 GPIO และการเชื่อมต่ออื่นๆ

บอร์ด ESP32-CAM มีพอร์ต GPIO ที่สามารถเชื่อมต่อกับเซ็นเซอร์และอุปกรณ์อื่นๆ ทำให้เหมาะสำหรับการใช้งานในโปรเจกต์ IoT ที่ซับซ้อน เช่น การตรวจสอบสภาพแวดล้อม หรือการควบคุมอุปกรณ์ไฟฟ้า

## 2.2 FTDI

FTDI (Future Technology Devices International) เป็นชิปที่ทำหน้าที่เป็นตัวแปลงสัญญาณระหว่างพอร์ต USB (Universal Serial Bus) และ UART (Universal Asynchronous Receiver-Transmitter) แสดงดังรูปที่ 2.2 ซึ่งทำให้สามารถเชื่อมต่ออุปกรณ์ที่มีพอร์ต UART เช่น ESP32-CAM เข้ากับคอมพิวเตอร์ผ่านพอร์ต USB ได้อย่างสะดวก ในการใช้งาน FTDI ร่วมกับ ESP32-CAM เราจะใช้ FTDI เพื่ออัปโหลดโปรแกรมไปยังบอร์ดและตรวจสอบข้อมูลผ่าน Serial Monitor โดยการต่อขา TX ของ FTDI เข้ากับ RX ของ ESP32-CAM และขา RX ของ FTDI เข้ากับ TX ของ ESP32-CAM รวมถึงต่อขา GND ให้ตรงกัน และจ่ายไฟให้บอร์ดผ่านขา 5V หรือ 3.3V จาก FTDI ซึ่งการใช้งานนี้ทำให้การพัฒนาและทดสอบโปรแกรมบน ESP32-CAM เป็นไปอย่างสะดวกและรวดเร็ว

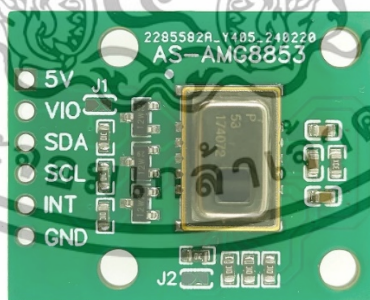
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 FTDI

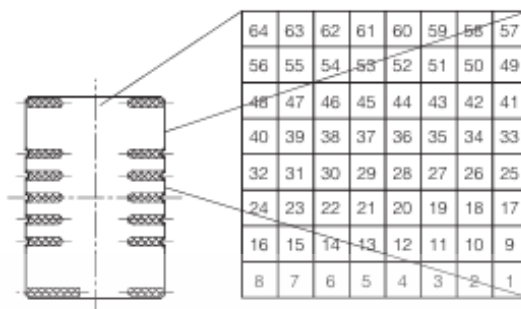
### 2.3 โมดูลกล้องอินฟราเรดวัดอุณหภูมิ (AMG8853)

โมดูลกล้องอินฟราเรดวัดอุณหภูมิ AMG8853 หรือบางครั้งเรียกว่า Grid-EYE sensor เป็นเซ็นเซอร์อินฟราเรดตรวจจับความร้อนที่ผลิตโดย Panasonic แสดงดังรูปที่ 2.3 ซึ่งถูกออกแบบมาเพื่อตรวจจับและแสดงผลพีธในรูปแบบของอาร์เรย์ 8x8 พิกเซล ทำให้สามารถตรวจจับอุณหภูมิได้สูงสุด 64 จุดพร้อมกันในช่วง 0°C ถึง 80°C แสดงดังรูปที่ 2.4 โดยส่งข้อมูลผ่านการเชื่อมต่อแบบ I2C ทำให้ง่ายต่อการใช้งานร่วมกับไมโครคอนโทรลเลอร์ เช่น ESP32 หรือ Arduino ด้วยความสามารถในการตรวจจับความร้อนบนพื้นผิวของวัตถุหรือร่างกาย AMG8853 จึงเหมาะสำหรับการใช้งานในระบบตรวจจับคนในพื้นที่, ระบบตรวจวัดอุณหภูมิร่างกาย และระบบรักษาความปลอดภัยที่ต้องการการตรวจจับอุณหภูมิอย่างละเอียดในมุมมองกว้าง [3]



รูปที่ 2.3 โมดูลกล้องอินฟราเรดวัดอุณหภูมิ (AMG8853)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 รูปแบบของอาร์เรย์ 8x8 พิกเซล

## 2.4 จอ LCD

จอ LCD (Liquid Crystal Display) เป็นหน้าจอแสดงผลที่ใช้เทคโนโลยีผลึกเหลว (Liquid Crystal) ในการสร้างภาพ แสดงดังรูปที่ 2.5 โดยอาศัยหลักการที่ผลึกเหลวจะเปลี่ยนการวางตัวเมื่อได้รับแรงดันไฟฟ้า ซึ่งส่งผลให้แสงสามารถผ่านผลึกเหลวในบางทิศทางเท่านั้น ทำให้เกิดเป็นภาพที่สามารถมองเห็นได้บนจอ LCD โดยทั่วไปจอ LCD มีโครงสร้างหลักคือไฟแบ็คไลท์ (Backlight) ซึ่งเป็นแหล่งกำเนิดแสงที่อยู่ด้านหลังของจอ ผลึกเหลว และฟิลเตอร์สีที่ช่วยในการแสดงสีต่างๆ ในจอภาพ ทำให้เรามองเห็นตัวอักษร รูปภาพ หรือกราฟิกต่างๆ ได้ [4]

การเชื่อมต่อจอ LCD กับ ESP32-CAM เพื่อแสดงผลสามารถทำได้ผ่านการใช้โมดูล LCD ที่มีอินเทอร์เฟซ I2C (Inter-Integrated Circuit) ซึ่งเป็นมาตรฐานการสื่อสารที่ช่วยให้การส่งข้อมูลระหว่าง ESP32-CAM กับจอ LCD ง่ายและสะดวกขึ้น โดยใช้สายสัญญาณเพียง 2 เส้น ได้แก่ SDA (Data) และ SCL (Clock) การทำงานเริ่มจาก ESP32-CAM จะส่งข้อมูลการแสดงผลผ่านทางอินเทอร์เฟซ I2C ไปยังโมดูล LCD ซึ่งจะทำการประมวลผลและแสดงผลตามข้อมูลที่ได้รับ ช่วยในการแสดงข้อความ โดยจะนำจอ LCD มาใช้งานการแสดงผลสถานะการลงเวลาเข้า-ออกงาน



รูปที่ 2.5 จอ LCD 20x4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 การประมวลผลภาพ (Image Processing)

เป็นการประยุกต์ใช้งานการประมวลผลสัญญาณบนสัญญาณ 2 มิติ เช่น ภาพนิ่ง (ภาพถ่าย) หรือภาพวิดีโอ (วิดีโอ) และยังรวมถึงสัญญาณ 2 มิติอื่น ๆ ที่ไม่ใช่ภาพด้วย แนวความคิดและเทคนิค ในการประมวลผลสัญญาณ สำหรับสัญญาณ 1 มิตินั้น สามารถปรับมาใช้กับภาพได้ไม่ยากแต่นอกเหนือจากเทคนิคจากการประมวลผลสัญญาณแล้ว การประมวลผลภาพก็มีเทคนิค และแนวความคิดที่เฉพาะ เช่น (Connectivity และ Rotation Invariance) ซึ่งจะมีความหมายกับสัญญาณ 2 มิติเท่านั้น แต่อย่างไรก็ตามเทคนิคบางอย่างจากการประมวลผลสัญญาณใน 1 มิติจะค่อนข้างซับซ้อนเมื่อนำมาใช้กับ 2 มิติ

การประมวลผลภาพนั้น จะอยู่ในรูปของการประมวลผลสัญญาณแอนะล็อก (Analog) โดยใช้อุปกรณ์ปรับแต่งแสง (Optics) ซึ่งวิธีเหล่านั้นก็ไม่ได้หายสาบสูญ หรือเลิกใช้ไป ยังมีใช้เป็นส่วนสำคัญสำหรับการประยุกต์ใช้งานบางอย่าง เช่น ฮอโลกราฟี แต่เนื่องจากอุปกรณ์คอมพิวเตอร์ในปัจจุบันราคาถูกลงและเร็วขึ้นมาก การประมวลผลภาพดิจิทัล (Digital Image Processing) จึงได้รับความนิยมมากกว่า เพราะการประมวลผลทำได้ซับซ้อนขึ้น แม่นยำ และง่ายในการลงมือปฏิบัติการประมวลผลภาพดิจิทัล หมายถึง การเรียกใช้ขั้นตอนหรือกรรมวิธีใดมากระทำกับภาพ โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณภาพของภาพ ให้ได้ภาพใหม่ที่มีคุณสมบัติตามต้องการ เช่น ความคมชัดหรือการประหยัดพื้นที่ในการเก็บข้อมูล หรือใช้สำหรับการประมวลผลระดับสูง เช่น การจดจำรูปร่างลักษณะให้ได้อย่างแม่นยำ การประมวลผลภาพแบ่งออกได้เป็น 2 ระดับ คือ

1. การประมวลผลภาพระดับต่ำ (Low Level Image Processing) เป็นการประมวลผลขั้นแรกสุดก่อนที่จะนำไปสู่การประมวลผลภาพระดับสูงต่อไป นั่นคือหลังจากที่ได้ภาพมา ซึ่งภาพที่ได้จะมีองค์ประกอบต่าง ๆ มากมาย รวมถึงสิ่งที่ไม่ต้องการด้วย ในที่นี้จะเรียกว่า "สัญญาณรบกวน" ซึ่งทำให้ภาพที่ได้มีคุณภาพไม่ดี ยังไม่สามารถนำไปใช้ในการประมวลผลได้ ดังนั้นการประมวลผลภาพระดับต่ำ จึงต้องมีการกำจัดสัญญาณรบกวนออก การทำภาพให้ชัด (High Pass Filter) การหาขอบภาพ (Edge Detection) การแปลงไปนารีอิมเมจการแบ่งแยกรูปร่าง วัตถุ (Image Segmentation) เป็นต้น เพื่อหาค่าตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ และมีวัตถุประสงค์ที่จะนำตัวแปรเหล่านั้นมาใช้ในการประมวลผลภาพระดับสูงต่อไป

2. การประมวลผลภาพระดับสูง (High Level Image Processing) เป็นการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพนั้นได้ เช่น การจดจำใบหน้าคน หรืออาจจะเป็นการจดจำตัวอักษร เป็นต้น ความแตกต่างของการประมวลผลภาพระดับต่ำและระดับสูงคือ ข้อมูลที่นำมาใช้ในการประมวลผล โดยการประมวลผลระดับต่ำจะใช้ค่าความสว่างหรือความเข้มแสงโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการประมวลผลภาพระดับสูงข้อมูลที่น่ามาใช้ในการประมวลผลจะถูกแสดงในรูปแบบของสัญลักษณ์ โดยสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่าง ๆ ที่อยู่ในภาพและการใช้ตัวแปรที่ได้จากการประมวลผลภาพระดับต่ำมาอธิบายถึงสัญลักษณ์เหล่านี้ การประมวลผลภาพระดับสูงนั้นส่วนใหญ่มักจะใช้ทฤษฎีต่าง ๆ เข้ามาใช้เป็นตัวช่วยในการทำงาน หรือเป็นหัวใจของโปรแกรม เช่น ตรรกศาสตร์คลุมเครือ (fuzzy logic) โครงข่ายประสาท (Neural Network) อย่างที่กล่าวไปแล้วว่าการประมวลผลภาพระดับสูง จำเป็นต้องใช้ข้อมูลที่ได้มาจากการประมวลผลระดับต่ำ ดังนั้นจะเห็นได้ว่าการประมวลผลภาพระดับต่ำมีความสำคัญมากสำหรับการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพได้ในส่วนการแทนภาพด้วยข้อมูลแบบดิจิทัลจากที่ได้กล่าวไปแล้วในตอนต้น ข้อมูลภาพแบบดิจิทัลเป็นภาพที่ถูกตัดแปลงมาจากภาพแบบต่อเนื่อง หรือภาพแอนะล็อก ให้อยู่ในรูปตัวเลข หรือภาพดิจิทัลด้วยวิธีการดิจิทัลไเซชัน (Digitization) โดยภาพแอนะล็อกจะถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า จุดภาพ (Pixels) โดยแต่ละพิกเซลจะใช้  $(X,Y)$  ในการระบุตำแหน่ง การแสดงข้อมูลภาพดิจิทัลสามารถอธิบายได้ด้วยแมทริกซ์  $(M \times N)$  และให้จุดต่าง ๆ ที่อยู่ ในแมทริกซ์เป็นจุดที่พิกัด  $(X,Y)$  ใด ๆ เป็นส่วนประกอบของภาพ แสดงดังรูปที่ 2.6



รูปที่ 2.6 ข้อมูลแบบพิกเซล

## 2.6 FaceNet

คือระบบที่พัฒนาโดย Google ที่สามารถใช้ในการตรวจจับและระบุใบหน้าได้ โดยการแปลงภาพใบหน้าทุกใบให้เป็นเวกเตอร์ในพื้นที่เชิงมิติที่มีความหมาย (embedding space) ซึ่งสามารถใช้ในการวิเคราะห์ใบหน้าในลักษณะต่างๆ เช่น การจับคู่ใบหน้า หรือการค้นหาคนที่มีความคล้ายกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FaceNet ใช้การเรียนรู้จากข้อมูลใบหน้าในรูปแบบที่เป็น triplet loss function ซึ่งช่วยในการลดความคลาดเคลื่อนระหว่างใบหน้าที่เหมือนกันและเพิ่มความแตกต่างของใบหน้าที่ไม่เหมือนกัน โดยระบบจะถูกฝึกโดยการให้ภาพใบหน้า 3 ภาพ ได้แก่ ภาพของคนเหมือนกัน (positive pair), ภาพของคนที่ไม่เหมือนกัน (negative pair) และภาพที่อยู่ในพื้นที่ที่ต้องการให้ระบบเรียนรู้ [5]

### 2.6.1 กระบวนการทำงานของ FaceNet

1. การตรวจจับใบหน้า (Face Detection) ใช้เทคนิคเช่น MTCNN หรือ Haar Cascade เพื่อตรวจจับตำแหน่งใบหน้าในภาพ และตัดภาพเฉพาะส่วนที่เป็นใบหน้าสำหรับขั้นตอนต่อไป

2. การปรับแต่งใบหน้า (Face Alignment) เป็นทำให้ใบหน้าอยู่ในรูปแบบมาตรฐาน โดยการหมุนหรือปรับขนาด พร้อมทั้งใช้เทคนิค Landmark Detection เพื่อตรวจจับจุดสำคัญของใบหน้า เช่น ดวงตา จมูก และปาก เพื่อให้การเปรียบเทียบใบหน้าที่มีความแม่นยำมากขึ้น [6]

3. การดึงคุณลักษณะสำคัญของใบหน้า (Feature Extraction) เมื่อได้ภาพใบหน้าที่ปรับแต่งแล้ว ระบบจะเข้าสู่กระบวนการ การดึงคุณลักษณะสำคัญของใบหน้าโดยใช้โครงข่ายประสาทเทียมแบบ CNN เพื่อแปลงใบหน้าใหญ่กลายเป็นเวกเตอร์ที่มี 128 มิติ (128-dimensional embedding) ซึ่งเป็นตัวแทนของลักษณะเฉพาะของใบหน้านั้นๆ กระบวนการนี้ช่วยให้สามารถเปรียบเทียบใบหน้าที่แตกต่างกันได้อย่างมีประสิทธิภาพ

4. การเปรียบเทียบใบหน้า (Face Matching)  
หลังจากได้ค่าเวกเตอร์ของใบหน้าแล้ว ขั้นตอน การเปรียบเทียบใบหน้า จะทำงานโดยใช้วิธีการคำนวณระยะห่างระหว่างเวกเตอร์ของภาพใบหน้าที่ต้องการตรวจสอบกับฐานข้อมูลใบหน้าที่มีอยู่ โดยใช้ค่า Euclidean Distance หรือ Cosine Similarity เพื่อวัดความคล้ายคลึงของเวกเตอร์ หากระยะห่างต่ำกว่าเกณฑ์ที่กำหนด ก็สามารถสรุปได้ว่าเป็นบุคคลเดียวกัน

5. การจำแนกหรือจัดกลุ่มใบหน้า (Classification)

สุดท้าย ระบบสามารถนำค่า embedding vector ไปใช้ใน การจำแนกหรือจัดกลุ่มใบหน้า (Classification หรือ Clustering) ได้ โดยสามารถนำไปใช้จำแนกบุคคลในฐานข้อมูล หรือใช้เทคนิค Clustering เพื่อจัดกลุ่มภาพใบหน้าที่คล้ายกันโดยไม่ต้องมีการระบุชื่อมาก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.2 ข้อดีของ FaceNet

FaceNet มีความแม่นยำสูงและได้รับคะแนนใกล้เคียงมนุษย์ใน Labeled Faces in the Wild (LFW) benchmark นอกจากนี้ยังสามารถทำงานได้อย่างรวดเร็ว โดยแปลงภาพใบหน้าเป็น embedding vector ได้อย่างมีประสิทธิภาพ ทำให้สามารถนำไปใช้ในระบบที่ต้องการความเร็วสูง เช่น การตรวจสอบตัวตนแบบเรียลไทม์ อีกทั้งยังสามารถนำไปประยุกต์ใช้ได้หลากหลาย เช่น ระบบควบคุมการเข้าออกอาคาร ระบบเฝ้าระวัง หรือระบบการระบุตัวตนผ่านใบหน้าในแพลตฟอร์มดิจิทัล

## 2.7 Haar Cascade

Haar Cascade เป็นอัลกอริธึมที่ใช้การเรียนรู้ของเครื่อง (machine learning) สำหรับการตรวจจับวัตถุในภาพหรือวิดีโอ อัลกอริธึมนี้ใช้ชุด Haar-like features ซึ่งเป็นรูปแบบสีเหลี่ยมของค่าพิกเซล เพื่อแยกแยะระหว่างวัตถุกับพื้นหลัง อัลกอริธึมจะฝึกตัวจำแนกโดยใช้คุณลักษณะเหล่านี้ ซึ่งสามารถใช้ในการตรวจจับวัตถุในภาพหรือวิดีโอใหม่ๆ [7]

Haar Cascade เหมาะสำหรับการตรวจจับใบหน้าอย่างยิ่ง เนื่องจาก Haar-like features สามารถใช้ในการแยกแยะลักษณะต่างๆ ของใบหน้า เช่น ดวงตา จมูก และปาก อัลกอริธึมจะเริ่มต้นด้วยการสร้าง Haar Cascade classifier โดยใช้ชุดภาพบวก (positive images) และภาพลบ (negative images) ภาพบวกจะมีใบหน้า ส่วนภาพลบจะไม่มี

### 2.7.1 หลักการทำงานของ Haar Cascade

#### 1. Haar-like Features

Haar-like Features เป็นพีเจอร์ที่ใช้ตรวจจับลักษณะเฉพาะของภาพโดยเปรียบเทียบค่าความเข้มของพิกเซลในพื้นที่สี่เหลี่ยมผืนผ้าขนาดต่าง ๆ ตัวอย่างเช่น พีเจอร์ขอบ (Edge Feature) ใช้ตรวจจับความแตกต่างระหว่างส่วนมืดและส่วนสว่าง พีเจอร์เส้นขอบ (Line Feature) ใช้ตรวจจับลวดลาย และพีเจอร์สี่เหลี่ยมกลาง (Four-rectangle Feature) ใช้ตรวจสอบการกระจายของแสงภายในพื้นที่ การใช้พีเจอร์เหล่านี้ช่วยระบุส่วนที่อาจเป็นวัตถุที่ต้องการตรวจจับ

#### 2. Integral Image

Integral Image เป็นเทคนิคที่ช่วยลดภาระการคำนวณของ Haar-like Features โดยการคำนวณค่าผลรวมสะสมของพิกเซลในภาพเพียงครั้งเดียว จากนั้นสามารถดึงค่าผลรวมของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฝึกเซลในพื้นที่ใด ๆ ได้อย่างรวดเร็วโดยใช้เพียงไม่กี่การคำนวณ ซึ่งช่วยลดเวลาในการคำนวณ ฟิวเจอร์จำนวนมากได้อย่างมาก ทำให้การตรวจจับวัตถุสามารถทำได้แบบเรียลไทม์

### 3. AdaBoost (Adaptive Boosting)

AdaBoost เป็นเทคนิค Machine Learning ที่ใช้คัดเลือกฟิวเจอร์ที่สำคัญจากฟิวเจอร์จำนวนมาก โดยจะให้ความสำคัญกับฟิวเจอร์ที่สามารถจำแนกวัตถุได้อย่างแม่นยำ ฟิวเจอร์ที่ผ่านการคัดเลือกจะถูกนำมาใช้ร่วมกันเพื่อสร้างตัวจำแนกวัตถุที่มีประสิทธิภาพสูงขึ้น วิธีนี้ช่วยลดจำนวนฟิวเจอร์ที่ต้องใช้ลงไปอย่างมาก ทำให้การคำนวณมีประสิทธิภาพมากขึ้น

### 4. Cascade Classifier

Cascade Classifier เป็นกระบวนการทำงานแบบเรียงลำดับที่ช่วยลดการคำนวณโดยการกรองภาพที่ไม่ใช่วัตถุออกตั้งแต่ขั้นต้น โดยจะเริ่มจากการตรวจสอบฟิวเจอร์ง่าย ๆ ก่อน หากผ่านการตรวจสอบในขั้นแรกแล้วจึงเข้าสู่ขั้นตอนถัดไปที่ซับซ้อนขึ้น กระบวนการนี้ช่วยลดจำนวนภาพที่ต้องตรวจสอบแบบละเอียดทำให้ระบบสามารถตรวจจับวัตถุได้เร็วขึ้นและมีประสิทธิภาพสูงขึ้น

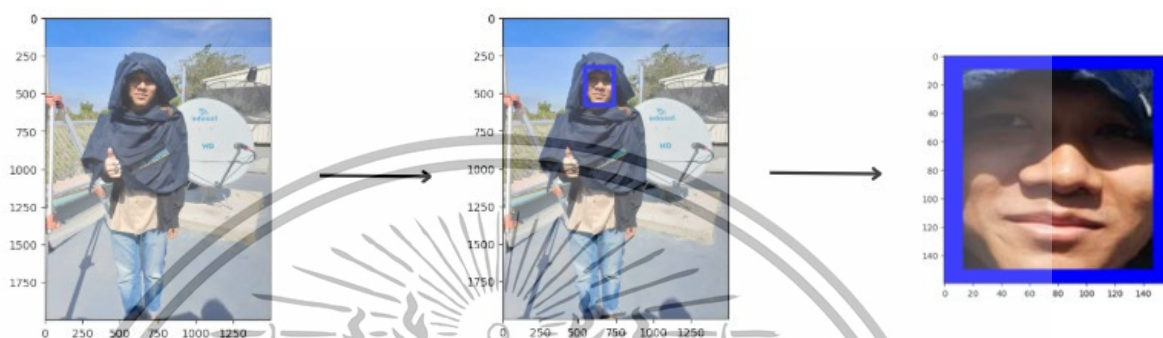
## 2.8 MTCNN (Multi-task Cascaded Convolutional Networks)

MTCNN (Multi-Task Cascaded Convolutional Networks) เป็นอัลกอริธึมการตรวจจับและจัดตำแหน่งใบหน้าที่ใช้หลักการเรียนรู้เชิงลึก (Deep Learning) โดยอาศัยชุดเครือข่ายประสาทเทียมแบบคอนโวลูชัน (CNNs) หลายชั้นที่ทำงานแบบเป็นขั้นตอน เพื่อระบุตำแหน่งและตรวจจับใบหน้าในภาพดิจิทัลหรือวิดีโอ อัลกอริธึมนี้สามารถตรวจจับใบหน้าที่มีขนาดและการหมุนที่หลากหลาย อีกทั้งยังมีความทนทานต่อการเปลี่ยนแปลงของสภาพแสง การแสดงออกทางใบหน้า และสิ่งบดบังบางส่วน [8]

ขั้นตอนการทำงานของ MTCNN ประกอบด้วย 3 เครือข่ายหลัก ได้แก่ Proposal Network (P-Net), Refinement Network (R-Net) และ Output Network (O-Net) โดยในขั้นตอนแรก P-Net จะทำหน้าที่สร้างกล่องกรอบ (bounding boxes) จำนวนมากที่มีความเป็นไปได้ว่าจะมีใบหน้าอยู่ โดยใช้การคอนโวลูชันเพื่อสร้างแผนที่คุณลักษณะ (feature maps) และส่งผ่านไปยังชั้น fully connected เพื่อตรวจหาความน่าจะเป็นของใบหน้าในแต่ละพื้นที่ พร้อมทั้งประมาณค่าพิกัดของ bounding boxes ขั้นตอนถัดไป R-Net จะนำ bounding boxes เหล่านี้มาปรับแต่งตำแหน่งให้แม่นยำยิ่งขึ้น โดยการครอบพื้นที่ที่สนใจ ตัดขนาดให้เท่ากัน และผ่านเครือข่ายคอนโวลูชันและ fully connected อีกครั้งเพื่อคัดกรอง bounding boxes ที่ไม่ใช่ใบหน้าออก และทำการปรับพิกัดให้แม่นยำมากขึ้น สุดท้าย O-Net จะรับ bounding boxes ที่ผ่านการกรองแล้วมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

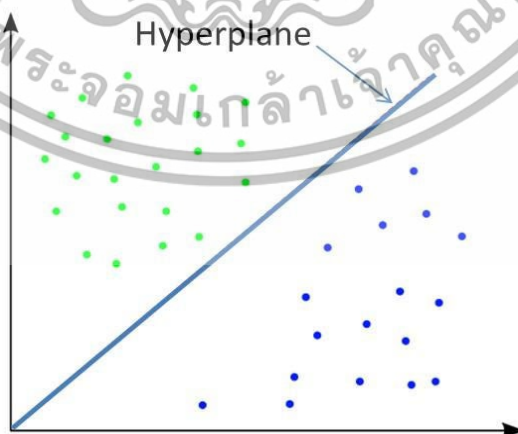
ปรับแต่งเพิ่มเติมอีกครั้ง และตั้งจุด landmark ที่สำคัญ 5 จุดบนใบหน้า ได้แก่ ดวงตาทั้งสองข้าง จมูก และปาก นอกจากนี้ O-Net ยังทำให้พิกัดของ bounding boxes แม่นยำที่สุดเท่าที่จะเป็นไปได้ก่อนจะส่งผลลัพธ์ออกมา แสดงดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างการใช้ MTCNN ตรวจจับใบหน้า

## 2.9 Support Vector Machine (SVM)

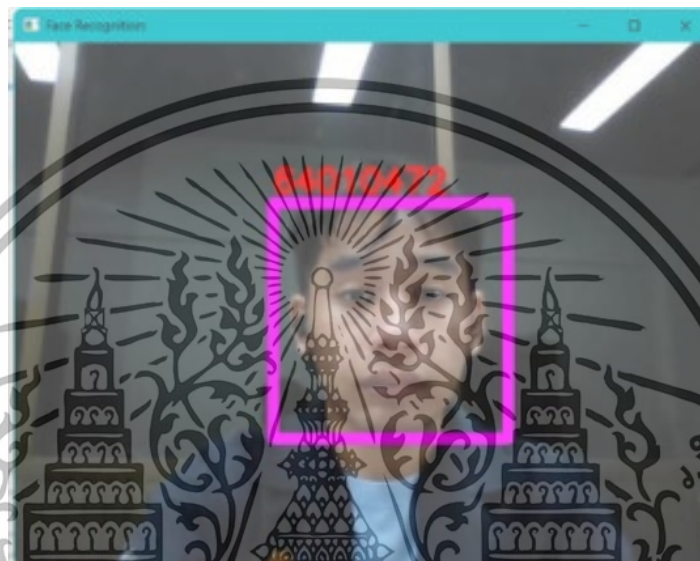
SVM คือ machine learning algorithm ที่ใช้สำหรับการจำแนกประเภท (classification) และการถดถอย (regression) โดยหลักการของมันคือการหาพื้นที่แบ่งที่ดีที่สุด ซึ่งสามารถแยกข้อมูลในกลุ่มต่าง ๆ ออกได้อย่างมีประสิทธิภาพ โดยจะสร้างเส้นตรงที่ใช้แบ่งกลุ่มข้อมูล (Hyperplane) และหา "เส้นที่ดีที่สุด" ที่ห่างจากจุดข้อมูลในแต่ละกลุ่มมากที่สุด ซึ่งช่วยเพิ่มความแม่นยำในการจำแนกประเภท [9] แสดงดังรูปที่ 2.8



รูปที่ 2.8 การขีดเส้นแบ่งกลุ่มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากทำ Model จาก SVM (Support Vector Machine) แล้ว ก็จะนำมาทดสอบกับกล้อง Webcam โดยให้จับภาพหน้าคนในกรอบสี่เหลี่ยม (bounding box) และแสดงคลาสที่ได้จำแนกไว้ โดย คลาสที่ได้จำแนกไว้คือผลลัพธ์จาก SVM ที่ระบุว่าใบหน้านั้นอยู่ในคลาสใด เช่น คลาสของบุคคล หรือประเภทที่จำแนกได้แสดงดังรูปที่ 2.9



รูปที่ 2.9 ตัวอย่างการใช้ Haar Cascade ร่วมกับ SVM

## 2.10 K-nearest Neighbors

k-Nearest Neighbors (KNN) คือ machine learning algorithm ที่ใช้สำหรับการจำแนกประเภท (classification) และการถดถอย (regression) อัลกอริทึมนี้ทำงานโดยการค้นหาตัวอย่างข้อมูลจากชุดฝึกที่ใกล้เคียงกับข้อมูลตัวอย่างใหม่ที่ต้องการทำนาย โดยใช้ระยะทางในเชิงคุณลักษณะ (Feature Space) เช่น ระยะทางแบบ Euclidean จากนั้นจึงกำหนดผลลากหรือค่าตอบแทนโดยพิจารณาจากจำนวนมากของเพื่อนบ้านที่ใกล้เคียง (Majority Voting) หรือการเฉลี่ยค่าในกรณีของการถดถอย

การเลือกค่า K มีความสำคัญต่อความแม่นยำของการจำแนกประเภท ค่า K ที่น้อยเกินไปอาจทำให้โมเดลไวต่อสัญญาณรบกวน ขณะที่ค่า K ที่มากเกินไปอาจทำให้โมเดลไม่สามารถจับความแตกต่างระหว่างประเภทได้อย่างชัดเจน ดังนั้น ควรทดลองกับค่าต่าง ๆ เพื่อหาค่าที่เหมาะสมที่สุด [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.11 Random forest

Random Forest เป็นอัลกอริทึมการเรียนรู้ของเครื่องประเภทการเรียนรู้แบบมีผู้สอน (Supervised Learning) ซึ่งสามารถใช้ได้ทั้งงานจำแนกประเภท (Classification) และงานพยากรณ์ค่า (Regression) โดยเป็นการนำแนวคิดของต้นไม้ตัดสินใจ (Decision Tree) มาขยายให้มีความแม่นยำและลดปัญหาการเกิด Overfitting

หลักการการทำงานของ Random Forest คือการสร้างกลุ่มของต้นไม้ตัดสินใจจำนวนมาก (Forest) โดยต้นไม้แต่ละต้นจะถูกฝึกด้วยข้อมูลที่สุ่มขึ้นมาจากรุ่นข้อมูลทั้งหมด พร้อมทั้งใช้การสุ่มตัวแปรอิสระที่ใช้ในการแยกข้อมูล (Feature Selection) ทำให้แต่ละต้นไม้มีลักษณะแตกต่างกัน เมื่อนำข้อมูลใหม่เข้ามา ระบบจะให้แต่ละต้นไม้ทำการทำนายผลลัพธ์ จากนั้นใช้หลักการให้คะแนนเสียงข้างมาก (Majority Voting) สำหรับปัญหาจำแนกประเภท หรือหาค่าเฉลี่ยของผลลัพธ์ทั้งหมดในกรณีพยากรณ์ค่า

ข้อดีของ Random Forest คือสามารถลดปัญหา Overfitting ของต้นไม้ตัดสินใจเดี่ยวได้ดี ทำให้มีความแม่นยำสูง และสามารถจัดการกับข้อมูลที่มามีมิติสูงได้ นอกจากนี้ยังสามารถใช้วัดความสำคัญของตัวแปรอิสระ (Feature Importance) ได้โดยดูว่าตัวแปรใดถูกใช้ในการแยกข้อมูลบ่อยที่สุด อย่างไรก็ตาม ข้อเสียของ Random Forest คือใช้ทรัพยากรคำนวณมากกว่า Decision Tree ทั่วไป และอาจไม่เหมาะกับงานที่ต้องการความเร็วสูง [11]

## 2.12 Gradient Boosting

Gradient Boosting เป็นเทคนิคการเรียนรู้ของเครื่องที่ใช้แนวคิดของการเรียนรู้แบบเสริมกำลัง (Boosting) โดยสร้างแบบจำลองที่แข็งแกร่งจากแบบจำลองอ่อน ๆ (Weak Learners) หลายตัว ซึ่งมักใช้ต้นไม้ตัดสินใจขนาดเล็กเป็นตัวเรียนรู้หลัก หลักการสำคัญของ Gradient Boosting คือการสร้างแบบจำลองใหม่ให้แก้ไขข้อผิดพลาดของแบบจำลองก่อนหน้า โดยแต่ละต้นไม้จะถูกฝึกให้ลดค่าความคลาดเคลื่อน (Residual) ของการทำนายก่อนหน้า ผ่านการลดทอนค่าคลาดเคลื่อนด้วยขั้นตอนที่เรียกว่า Gradient Descent วิธีนี้ทำให้โมเดลสามารถปรับปรุงความแม่นยำอย่างต่อเนื่อง

Gradient Boosting มีข้อดีที่สามารถเรียนรู้ความซับซ้อนของข้อมูลได้ดีและมีความแม่นยำสูงกว่าวิธีการเรียนรู้ของเครื่องแบบดั้งเดิม อย่างไรก็ตาม โมเดลนี้มีข้อเสียคือใช้เวลาฝึกนาน และมีโอกาสเกิด Overfitting ได้หากไม่มีการปรับแต่งพารามิเตอร์อย่างเหมาะสม เทคนิคที่นิยมใช้ในการลด Overfitting ได้แก่ การกำหนดอัตราการเรียนรู้ (Learning Rate) ที่เหมาะสม การจำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนต้นไม้ (Number of Trees) และการกำหนดความลึกของต้นไม้ (Tree Depth) ให้เหมาะสมกับข้อมูล Gradient Boosting ยังมีเวอร์ชันที่ได้รับการพัฒนาเพิ่มเติม เช่น XGBoost, LightGBM และ CatBoost ซึ่งเพิ่มประสิทธิภาพและลดระยะเวลาการฝึกโมเดล [12]

## 2.13 การทำอินเตอร์โพลेशनแบบไบลิเนียร์ (Bilinear interpolation)

การสร้างภาพให้มีความละเอียดสูงขึ้นหรือมีความชัดเจนมากยิ่งขึ้นวิธีการหนึ่งคือ การใช้เทคนิคการขยายภาพ (Resize) ให้ภาพเดิมมีขนาดของจุดภาพหรือพิกเซล (Pixel) ในภาพที่มากขึ้นด้วยการสุ่มตัวอย่างซ้ำ (Resampling) ซึ่งเป็นการสุ่มระดับสีจากจุดภาพเก่าที่อยู่ใกล้เคียงกับจุดภาพใหม่ ในการทำอินเตอร์โพลेशनแบบไบลิเนียร์ (Bilinear interpolation) จะทำการเพิ่มจุดภาพใหม่ขึ้นมา 1 จุดภาพโดยใช้การคำนวณหาจุดภาพใหม่ดังกล่าวจากจุดภาพเก่าที่อยู่บริเวณรอบ ๆ จำนวน 4 จุด  $(x_1, y_1)$ ,  $(x_2, y_1)$ ,  $(x_1, y_2)$  และ  $(x_2, y_2)$  โดยมีสมการในการคำนวณน้ำหนักของจุดที่สนใจดังสมการที่ 2.1 ซึ่ง Q คือค่าที่ทราบ ณ 4 จุดดังกล่าว [13]

$$f(x, y) = \frac{(x_2-x)(y_2-y)}{(x_2-x_1)(y_2-y_1)} Q_{11} + \frac{(x-x_1)(y_2-y)}{(x_2-x_1)(y_2-y_1)} Q_{21} + \frac{(x_2-x)(y-y_1)}{(x_2-x_1)(y_2-y_1)} Q_{12} + \frac{(x-x_1)(y-y_1)}{(x_2-x_1)(y_2-y_1)} Q_{22} \quad (2.1)$$

โดยมีขั้นตอนในการคำนวณดังต่อไปนี้

### 2.13.1 ระบุจุด 4 จุดรอบจุดที่สนใจ

กำหนดจุดเป้าหมายที่ต้องการประมาณค่า  $(x, y)$  จากนั้นหาตำแหน่งของสี่จุดที่ล้อมรอบจุดนั้น ซึ่งเป็นจุดที่มีค่าที่ทราบอยู่แล้ว โดยจุดเหล่านี้จะกำหนดเป็น  $Q_{11}$ ,  $Q_{12}$ ,  $Q_{21}$ ,  $Q_{22}$  และเรียงตัวกันเป็นรูปสี่เหลี่ยม

### 2.13.2 ทำการอินเตอร์โพลेशनในแนวแกน x

คำนวณค่าที่ตำแหน่ง  $(x, y_1)$  โดยใช้ค่าจาก  $Q_{11}$  และ  $Q_{21}$  และคำนวณค่าที่ตำแหน่ง  $(x, y_2)$  โดยใช้ค่าจาก  $Q_{12}$  และ  $Q_{22}$  โดยใช้อินเตอร์โพลेशनแบบลิเนียร์ในการคำนวณ

### 2.13.3 ทำการอินเตอร์โพลेशनในแนวแกน y

นำค่าที่ได้จากการทำอินเตอร์โพลेशनในแนว X มาทำอินเตอร์โพลेशनแบบลิเนียร์ในแนว Y จะได้ค่าประมาณที่ตำแหน่งเป้าหมาย  $(x, y)$

### 2.13.4 รวมผลลัพธ์เข้าด้วยกัน

ค่าที่ประมาณสุดท้ายเป็นค่าผสมเชิงเส้น (Linear Combination) ของค่าที่ทราบทั้งสี่จุดซึ่งน้ำหนักของแต่ละค่าจะขึ้นอยู่กับระยะห่างของจุดเป้าหมายจากจุดรอบข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.14 ไลบรารี OpenCV

ไลบรารี OpenCV ใน Python เป็นเครื่องมือที่ทรงพลังสำหรับการประมวลผลภาพและการวิเคราะห์วิดีโอแบบเรียลไทม์ ไลบรารีนี้ได้รับการออกแบบเพื่อช่วยให้นักพัฒนาสามารถทำงานด้านการประมวลผลภาพและการวิเคราะห์วิดีโอได้อย่างง่ายดาย ไม่ว่าจะเป็นการตรวจจับวัตถุ การรู้จำใบหน้า การติดตามการเคลื่อนไหวของวัตถุ หรือการปรับแต่งภาพ (Image Enhancement) โดย OpenCV ถูกใช้กันอย่างแพร่หลายในงานวิจัยด้านคอมพิวเตอร์วิทัศน์ งานด้านปัญญาประดิษฐ์ และการพัฒนาแอปพลิเคชันที่เกี่ยวข้องกับการวิเคราะห์ภาพ แสดงดังรูปที่ 2.10 โดยหลักการทำงานมีดังนี้



รูปที่ 2.10 ไลบรารี OpenCV

### 2.14.1 การอ่านและเขียนไฟล์ภาพและวิดีโอ

OpenCV สามารถอ่านภาพจากไฟล์และแสดงผลภาพได้ง่ายๆ ด้วยฟังก์ชัน `cv2.imread()` และ `cv2.imshow()` นอกจากนี้ยังสามารถบันทึกภาพลงไฟล์หรือสตรีมวิดีโอได้ด้วย `cv2.imwrite()` และ `cv2.VideoCapture()` ซึ่งทำให้สะดวกสำหรับการสร้างแอปพลิเคชันที่ต้องใช้การประมวลผลภาพจากกล้อง

### 2.14.2 การประมวลผลภาพ (Image Processing)

OpenCV รองรับการปรับแต่งภาพต่างๆ เช่น การแปลงสีด้วย `cv2.cvtColor()` ซึ่งใช้ในการแปลงภาพจาก RGB ไปยังโหมดสีอื่นๆ เช่น Grayscale หรือ HSV นอกจากนี้ยังรองรับการปรับค่าความเข้มของภาพ การเบลอภาพ การตรวจจับขอบด้วย Canny Edge Detection และการสร้างฟิลเตอร์ที่ช่วยปรับปรุงคุณภาพของภาพ

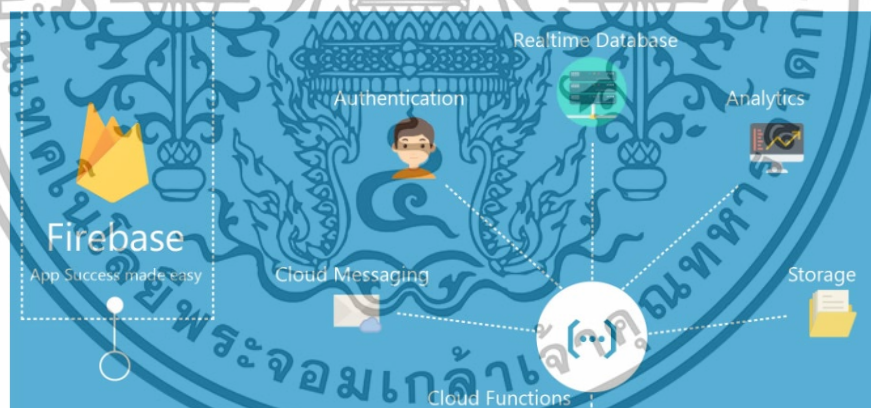
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.14.3 การแปลงภาพเชิงเรขาคณิต (Geometric Transformations)

OpenCV รองรับการเปลี่ยนแปลงภาพในเชิงเรขาคณิต เช่น การหมุนภาพ (Rotate) การย่อ-ขยาย (Resize) การตัดภาพ (Crop) และการเปลี่ยนมุมมอง (Perspective Transform) ซึ่งเป็นประโยชน์ในการปรับแต่งภาพให้อยู่ในรูปแบบที่ต้องการสำหรับการประมวลผลในขั้นตอนต่อไป

## 2.15 Firebase

Firebase เป็นแพลตฟอร์มคลาวด์ที่ให้บริการโดย Google สำหรับช่วยนักพัฒนาในการสร้างแอปพลิเคชันที่ต้องการการจัดการข้อมูลเรียลไทม์ การยืนยันตัวตน และเครื่องมือสำหรับการพัฒนาและบริหารจัดการแอปพลิเคชันในระบบคลาวด์อย่างสะดวก โดย Firebase รองรับการใช้งานทั้งบนเว็บ, iOS, Android และแพลตฟอร์มอื่นๆ จึงทำให้นักพัฒนาสามารถเชื่อมต่อและซิงค์ข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ ข้อดีของ Firebase คือลดความซับซ้อนของโครงสร้างพื้นฐาน ทำให้ทีมพัฒนาสามารถเน้นการสร้างคุณสมบัติใหม่ของแอปพลิเคชันได้อย่างเต็มที่ โดยไม่ต้องกังวลเรื่องการดูแลเซิร์ฟเวอร์และการจัดการข้อมูล แสดงดังรูปที่ 2.11 โดยการใช้งานและฟังก์ชันการทำงานของ Firebase [14] มีดังนี้



รูปที่ 2.11 บริการบน Firebase

### 2.15.1 Firebase Realtime Database

เป็นฐานข้อมูล NoSQL ที่ช่วยให้แอปพลิเคชันสามารถจัดเก็บและซิงค์ข้อมูลระหว่างผู้ใช้หลายคนได้แบบเรียลไทม์ Firebase Realtime Database จะซิงค์ข้อมูลทั้งหมดที่เปลี่ยนแปลงไปยังอุปกรณ์ทั้งหมดที่เชื่อมต่อ ทำให้ข้อมูลของทุกอุปกรณ์ถูกอัปเดตแบบเรียลไทม์ ฟังก์ชันนี้เหมาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับการพัฒนาแอปพลิเคชันที่ต้องการให้ข้อมูลเป็นปัจจุบันอยู่เสมอ เช่น แอปสนทนา, ระบบจัดการข้อมูล, หรือการแสดงผลข้อมูลเรียลไทม์

### 2.15.2 Cloud Firestore

Firebase ยังมี Cloud Firestore ซึ่งเป็นฐานข้อมูล NoSQL แบบเชิงเอกสารที่ช่วยให้จัดเก็บข้อมูลในรูปแบบที่ยืดหยุ่นกว่า Realtime Database โดยสามารถจัดโครงสร้างข้อมูลให้เป็นคอลเลกชันและเอกสาร เหมาะกับการใช้งานที่ต้องการเก็บข้อมูลเชิงโครงสร้าง เช่น ระบบจัดการสินค้าคงคลัง หรือระบบสั่งซื้อสินค้าในร้านค้าออนไลน์

### 2.15.3 Firebase Authentication

บริการนี้ช่วยในการยืนยันตัวตนของผู้ใช้ (Authentication) โดยรองรับหลายวิธี เช่น การเข้าสู่ระบบด้วยอีเมลและรหัสผ่าน, Google, Facebook, Twitter, Apple และเบอร์โทรศัพท์ Firebase Authentication จะช่วยให้ผู้ใช้สามารถยืนยันตัวตนได้อย่างปลอดภัยและสะดวก ซึ่งเป็นฟังก์ชันพื้นฐานของการพัฒนาแอปพลิเคชันที่ต้องการจัดการบัญชีผู้ใช้

### 2.15.4 Firebase Storage

บริการ Firebase Storage ช่วยในการจัดเก็บไฟล์ขนาดใหญ่ เช่น รูปภาพ วิดีโอ ไฟล์เสียง หรือเอกสาร โดยมีการรักษาความปลอดภัยของข้อมูลที่ผ่านการกำหนดสิทธิ์และการเข้ารหัส เหมาะกับการจัดการสื่อไฟล์ที่ต้องการจัดเก็บในระบบคลาวด์และเข้าถึงได้จากแอปพลิเคชันต่างๆ ที่เชื่อมต่อกับ Firebase ทำให้นักพัฒนาสามารถเก็บและดึงไฟล์ได้โดยไม่ต้องจัดการเซิร์ฟเวอร์เอง

### 2.15.5 Firebase Cloud Messaging (FCM)

FCM ช่วยให้สามารถส่งข้อความแจ้งเตือนไปยังอุปกรณ์ของผู้ใช้ผ่านช่องทางต่างๆ โดยไม่เสียค่าใช้จ่าย ไม่ว่าจะป็นข้อความแจ้งเตือนธรรมดา การแจ้งเตือนแบบกำหนดเป้าหมายที่เฉพาะเจาะจง หรือข้อความที่แจ้งเตือนข่าวสารสำคัญ FCM เหมาะกับแอปพลิเคชันที่ต้องการสื่อสารหรือแจ้งเตือนกับผู้ใช้อย่างทันทีทันใด เช่น แอปพลิเคชันส่งข้อความ แอปขายของ หรือแอปเกม

### 2.15.6 Firebase Analytics และ Crashlytics

Firebase Analytics ช่วยเก็บข้อมูลการใช้งานของผู้ใช้บนแอปพลิเคชัน ซึ่งจะช่วยให้เห็นพฤติกรรมผู้ใช้และข้อมูลการใช้งานที่สำคัญ ทำให้ทีมพัฒนาสามารถปรับปรุงการใช้งานให้ตรงกับความต้องการของผู้ใช้มากขึ้น ส่วน Crashlytics เป็นเครื่องมือที่ช่วยแจ้งข้อผิดพลาดหรือการล่มของแอปพลิเคชันแบบเรียลไทม์ จึงช่วยในการแก้ไขปัญหาได้อย่างรวดเร็ว

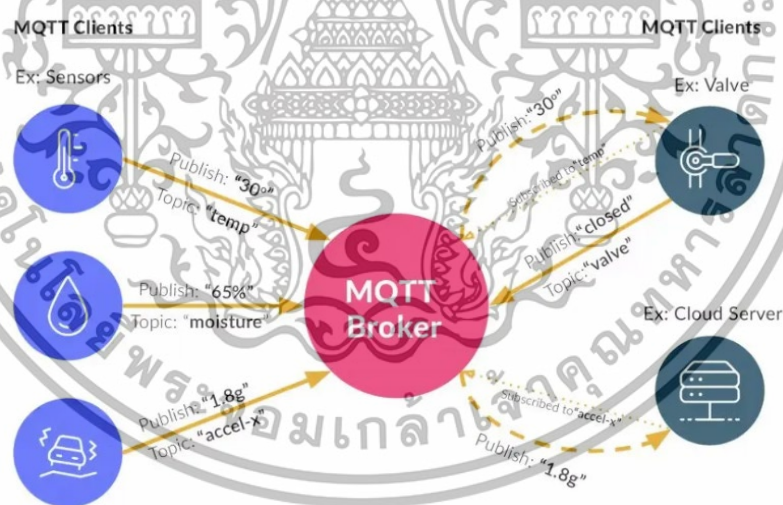
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.15.7 Firebase Cloud Functions

Cloud Functions เป็นบริการที่ช่วยให้สามารถรันโค้ด backend บนระบบคลาวด์ โดยไม่ต้องจัดการเซิร์ฟเวอร์เอง ตัวอย่างเช่น สามารถตั้งค่าให้ทำงานเมื่อมีการสร้างผู้ใช้ใหม่หรือเมื่อข้อมูลในฐานข้อมูลมีการเปลี่ยนแปลง ซึ่งเหมาะกับการทำงานที่ต้องการตรรกะการประมวลผลที่ไม่ต้องการการดูแลเซิร์ฟเวอร์

### 2.16 MQTT (Message Queuing Telemetry Transport)

MQTT (Message Queuing Telemetry Transport) เป็นโพรโตคอลสื่อสารที่ออกแบบมาเพื่อการส่งข้อมูลระหว่างอุปกรณ์ที่มีข้อจำกัดด้านทรัพยากร เช่น พลังงานหรือความเร็วของอินเทอร์เน็ต โดยเฉพาะในระบบ IoT (Internet of Things) ซึ่งมีการเชื่อมต่อและสื่อสารกันระหว่างอุปกรณ์จำนวนมาก MQTT มีการทำงานที่เรียบง่ายและมีประสิทธิภาพ เนื่องจากถูกพัฒนาให้ใช้ปริมาณข้อมูลน้อย มีการรับส่งข้อมูลแบบทันที (real-time) และรองรับการเชื่อมต่อที่ไม่เสถียรได้ดี [15] แสดงดังรูปที่ 2.12



รูปที่ 2.12 MQTT Clients

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.16.1 หลักการทำงานของ MQTT

MQTT ใช้โมเดลแบบ "Publish/Subscribe" ที่ประกอบไปด้วยองค์ประกอบหลัก 3 ส่วน ได้แก่

1. Broker เป็นตัวกลางหรือเซิร์ฟเวอร์ที่ทำหน้าที่จัดการการรับส่งข้อมูลระหว่างอุปกรณ์ต่าง ๆ โดยการรับข้อมูลจาก Publisher และกระจายไปยัง Subscriber ที่ต้องการข้อมูลใน "Topic" นั้น ๆ โดยไม่จำเป็นต้องให้ Publisher และ Subscriber ติดต่อกันโดยตรง
2. Publisher เป็นอุปกรณ์หรือโปรแกรมที่ส่งข้อมูลไปยัง Broker โดยระบุ "Topic" เพื่อกำหนดประเภทหรือหมวดหมู่ของข้อมูล เช่น ข้อมูลสภาพอากาศ ข้อมูลอุณหภูมิ เป็นต้น Publisher สามารถส่งข้อมูลไปยัง Broker ได้ทันทีตามความถี่ที่กำหนด
3. Subscriber เป็นอุปกรณ์หรือโปรแกรมที่ต้องการรับข้อมูลจาก Broker โดย Subscriber จะเลือก Topic ที่ต้องการ (Subscribe) เพื่อรับข้อมูลที่ถูกส่งเข้ามาใน Topic นั้น หากมีข้อมูลใหม่ใน Topic ที่สมัครไว้ Broker จะส่งข้อมูลไปยัง Subscriber โดยอัตโนมัติ

### 2.16.2 Topic และการจัดการข้อมูล

Topic ใน MQTT มีโครงสร้างเป็นลำดับชั้น เช่น home/livingroom/temperature โดย Subscriber สามารถเลือกสมัครเฉพาะ Topic ที่ต้องการรับข้อมูลได้ หรือใช้สัญลักษณ์ + และ # ในการเลือก Topic หลาย ๆ ระดับพร้อมกันได้ เช่น home/+/temperature หมายถึงรับข้อมูลจากทุกห้องที่มีหัวข้อย่อยว่า temperature ทำให้ MQTT มีความยืดหยุ่นสูงในการส่งข้อมูลไปยังกลุ่มอุปกรณ์ที่หลากหลาย

### 2.16.3 คุณสมบัติสำคัญของ MQTT

1. QoS (Quality of Service) MQTT มีระดับ QoS สามระดับเพื่อควบคุมความน่าเชื่อถือในการส่งข้อมูล คือ QoS 0, QoS 1, และ QoS 2
2. Retained Messages เมื่อส่งข้อมูลล่าสุดไปยัง Topic Broker สามารถเก็บข้อความนี้ไว้ และเมื่อมี Subscriber ใหม่เข้ามาใน Topic นี้ ก็จะได้รับข้อมูลที่เก็บไว้ที่ทันที
3. Last Will and Testament พี่เจอร์นี่ช่วยให้ MQTT สามารถแจ้งเตือนเมื่ออุปกรณ์ใดอุปกรณ์หนึ่งตัดการเชื่อมต่ออย่างไม่คาดคิด โดยอุปกรณ์จะประกาศข้อความสุดท้ายไปยัง Topic ที่ตั้งไว้ล่วงหน้า เพื่อให้ Subscriber ที่สนใจทราบว่าอุปกรณ์นั้น ๆ ออฟไลน์ไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.17 ระบบเว็บไซต์

เว็บไซต์ คือ สื่อนำเสนอข้อมูลบนเครื่องคอมพิวเตอร์ หรือคือการรวบรวม หน้าเว็บเพจหลายหน้า ซึ่งเชื่อมโยงกันผ่านทางไฮเปอร์ลิงก์ ซึ่งต้องเปิดด้วยโปรแกรมเฉพาะทางที่เรียกว่า WebBrowser โดยถูกจัดเก็บไว้ในเวิลด์ไวด์เว็บ และเว็บไซต์นั้นถูกสร้างขึ้นด้วยภาษาทางคอมพิวเตอร์ที่เรียกว่า HTML (Hyper Text Markup Language) และได้มีการพัฒนาและนำภาษาอื่นๆ เข้ามาร่วมด้วย เพื่อให้มีความสามารถมากขึ้น เช่น PHP, SQL, Java ฯลฯ

### 2.17.1 User Interface

User Interface หรือ UI คือ ทุกอย่างที่คุณควบคุมผ่านระบบคอมพิวเตอร์ อินเทอร์เน็ต เชื่อมต่อบนหน้าจอของผู้ใช้งาน ประกอบไปด้วยการออกแบบในส่วนต่าง ๆ ที่ส่งผลต่อผู้ใช้งาน โดยตรงได้แก่ การออกแบบกราฟิก การออกแบบเมนูตอบโต้ การออกแบบรูปแบบตัวอักษร สี ฯลฯ โดยเป้าหมายของ UI คือ การออกแบบให้ผู้ใช้งานเห็นแล้วสามารถเข้าใจได้ทันที เช่น เครื่องหมายถูกแล้วเข้าใจว่าตกลงหรือเห็นด้วย [16]

### 2.17.2 ภาษา HTML

HTML Tag คือ คำสั่งที่คอยควบคุมการแสดงผลของข้อความและสื่อต่าง ๆ บนหน้าเว็บเพจ เป็นคำสั่งที่เอาไว้ปรับแต่งการแสดงผลของเว็บเพจให้สวยงาม อ่านง่าย และเป็นที่ยึดจำของผู้เยี่ยมชม โดยชุดคำสั่งจะอยู่ในเครื่องหมาย "<" และ ">" โดยเราสามารถแบ่ง HTML Tag ออกเป็น 2 ประเภทหลัก ๆ คือ [17]

1. HTML Tag ที่เป็นคำสั่งเดี่ยว หรือ Empty Tag คือ มีเฉพาะ Tag เปิดเท่านั้น จะวางอยู่หน้าข้อความ เช่น <br> ที่ใช้สำหรับการขึ้นบรรทัดใหม่

2. HTML Tag ที่เป็นคำสั่งคู่ หรือ Container Tag HTML ย่อมาจากคำว่า Hypertext Markup Language แสดงดังรูปที่ 2.12 เป็นภาษาหลักที่ใช้ในการสร้างไฟล์เว็บเพจ โดยมีแนวคิดจากการสร้างเอกสารไฮเปอร์เท็กซ์ (Hypertext Document) ซึ่งพัฒนาขึ้นมาจากภาษา SGML (Standard Generalized Markup Language) โดย Tim Berners-Lee เป็นภาษามาตรฐานที่ใช้พัฒนาเอกสารในรูปแบบของเว็บเพจเผยแพร่บนระบบเครือข่ายอินเทอร์เน็ต มีโครงสร้างการเขียนที่อาศัยตัวกำกับ เรียกว่า แท็ก (Tag) ควบคุมการแสดงผลของข้อความ รูปภาพ หรือวัตถุอื่นๆ เรียกใช้เอกสารเหล่านี้โดยการใช้โปรแกรมเว็บเบราว์เซอร์ (Web Browser) คือ มีทั้ง Tag เปิดและปิด โดย Tag เปิดจะอยู่ด้านหน้าข้อความ และ Tag ปิด จะเป็นข้อความเดียวกับ tag เปิด แต่จะมีเครื่องหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

slash (/) อยู่ภายใน และจะอยู่หลังข้อความ เช่น เมื่อเริ่มต้นไฟล์ HTML Tag ที่ต้องใช้คือ <html> และเมื่อสิ้นสุด HTML Tag ที่ต้องใช้ก็คือ </html> แสดงดังรูปที่ 2.13

# HTML



รูปที่ 2.13 ภาษา HTML

### 2.17.3 ภาษา CSS

CSS ย่อมาจาก Cascading Style Sheet หรือที่มักเรียกกันสั้นๆ ว่า 'สไตล์ชีต' เป็นภาษาที่ใช้ส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML แสดงดังรูปที่ 2.13 โดยที่ CSS กำหนดและระบุรูปแบบ หรือ Style ของเนื้อหาในเอกสาร เช่น สีของข้อความ สีพื้นหลัง ประเภทของตัวอักษร และการจัดวางข้อความ [18]

การกำหนดรูปแบบ หรือ Style นี้ ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสาร ไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลพร้อมของเอกสาร HTML แสดงดังรูปที่ 2.14

# CSS



รูปที่ 2.14 ภาษา CSS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.17.4 ภาษา JavaScript

JavaScript เป็นภาษาโปรแกรมมิ่งที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันและเว็บไซต์ เป็นภาษาที่ใช้งานได้ทั้งด้าน ฝั่งของผู้ใช้ (client-side) และ ฝั่งของเซิร์ฟเวอร์ (server-side) โดยส่วนใหญ่มักนำมาใช้ในการเพิ่มความประสมความสามารถให้กับเว็บไซต์อย่างเช่น การสร้างการโต้ตอบแบบไดนามิก (Dynamic interaction) การเปลี่ยนแปลงสถานะของหน้าเว็บ (DOM manipulation) และการทำงานกับการเชื่อมต่อเครือข่าย (Networking) [19]

นอกจากนี้ JavaScript ยังเป็นภาษาที่มีความยืดหยุ่นสูงที่สามารถนำมาใช้ในการพัฒนาแอปพลิเคชันแบบเต็มรูปแบบ (Full-stack application) ได้ด้วย เช่นเดียวกับ Node.js ซึ่งเป็นระบบเซิร์ฟเวอร์ที่ใช้ JavaScript ในการพัฒนาแอปพลิเคชัน server-side อีกด้วย การเรียกใช้งาน JavaScript บนเว็บเบราว์เซอร์มักจะเกิดขึ้นผ่านทางคำสั่ง JavaScript โดยใช้แท็ก '<script>' ภายใน HTML หรือผ่านไฟล์ JavaScript ที่เรียกใช้งานภายในแท็ก <script> นั้นเป็นภาษาโปรแกรมมิ่งที่มีความนิยมและใช้งานกันอย่างแพร่หลายในอินเทอร์เน็ตและอุปกรณ์ต่าง ๆ ที่มีเบราว์เซอร์ รวมถึงมือถือและแท็บเล็ตต่าง ๆ ด้วย การพัฒนา JavaScript เองได้รับการพัฒนาอย่างต่อเนื่อง เพื่อให้สามารถรองรับการพัฒนาและการใช้งานในงานต่าง ๆ ได้อย่างเหมาะสม และมีการปรับปรุงความเร็วและประสิทธิภาพในการทำงานต่าง ๆ อยู่เสมอ โดยเฉพาะในปัจจุบัน JavaScript เป็นส่วนสำคัญของเว็บไซต์และแอปพลิเคชันในโลกของเทคโนโลยีสารสนเทศและการสื่อสารอินเทอร์เน็ต โดยมีการนำไปใช้ในหลายสถาบันและองค์กรต่าง ๆ ทั่วโลกในการพัฒนาและการทำงานที่ต่างกัน แสดงดังรูปที่ 2.15 [20]

**JavaScript**



รูปที่ 2.15 ภาษา JavaScript

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.18 Vue.js

Vue.js แสดงดังรูปที่ 2.16 เป็นเฟรมเวิร์ก JavaScript แบบโอเพนซอร์สที่ใช้ในการพัฒนา UI (User Interface) และ SPA (Single Page Applications) ได้อย่างมีประสิทธิภาพ ถูกออกแบบมาให้มีความยืดหยุ่น เรียบง่าย และสามารถปรับขยายได้ตามความต้องการของโปรเจกต์

Vue.js ถูกออกแบบมาให้รองรับการพัฒนาได้ทั้งในระดับเล็กและใหญ่ นักพัฒนาสามารถเลือกใช้ Vue ได้ตั้งแต่การเพิ่มฟีเจอร์เล็ก ๆ ให้กับเว็บไซต์ที่มีอยู่ ไปจนถึงการสร้างเว็บแอปพลิเคชันที่ซับซ้อน โดยมีเครื่องมือที่ช่วยให้การพัฒนาเป็นไปได้อย่างรวดเร็ว เช่น Vue CLI, Vue Router และ Vuex นอกจากนี้ Vue ยังรองรับการใช้งานร่วมกับ TypeScript เพื่อเพิ่มความปลอดภัยและความยืดหยุ่นในการพัฒนาอีกด้วย [21] [22]



รูปที่ 2.16 Vue.js

### 2.18.1 คุณสมบัติหลักของ Vue.js

#### 1. การตอบสนองอัตโนมัติ (Reactivity)

เป็นการช่วยให้ข้อมูลและ UI ซิงค์กันโดยอัตโนมัติผ่าน reactive data binding และใช้ Virtual DOM (Virtual Document Object Model) เพื่อเพิ่มประสิทธิภาพการเรนเดอร์หน้าเว็บ ทำให้การอัปเดต UI รวดเร็วขึ้นและลดภาระของเบราว์เซอร์

#### 2. โครงสร้างแบบคอมโพเนนต์ (Component-based Architecture)

ทำให้นักพัฒนาสามารถแยก UI ออกเป็นคอมโพเนนต์ย่อย ๆ ที่สามารถนำกลับมาใช้ซ้ำได้ ส่งผลให้ได้มีโครงสร้างที่ชัดเจนและง่ายต่อการดูแล

#### 3. Directives Rendering

เป็นเครื่องมือที่ช่วยในการจัดการ DOM อย่างมีประสิทธิภาพ เช่น v-if ใช้สำหรับแสดงหรือซ่อนองค์ประกอบตามเงื่อนไข v-for ใช้ในการวนลูปข้อมูล v-bind ใช้สำหรับผูกค่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอตทริบิวต์ และ v-model สำหรับการเชื่อมโยงข้อมูลแบบสองทาง ไดร็อกทีฟเหล่านี้ช่วยลดโค้ดที่ซับซ้อนและทำให้โค้ดกระชับขึ้น

#### 4. ระบบจัดการเส้นทาง (Vue Router)

เป็นเครื่องมือที่ช่วยให้ Vue สามารถสร้าง SPA ได้อย่างมีประสิทธิภาพ ทำให้สามารถเปลี่ยนหน้าโดยไม่ต้องโหลดใหม่ทั้งหมด ลดระยะเวลาการโหลดและเพิ่มประสบการณ์การใช้งานที่ราบรื่นขึ้น Vue Router ยังรองรับพีเจอร်ชั้นสูง เช่น การกำหนดเส้นทางแบบไดนามิกและการป้องกันเส้นทาง

#### 5. การจัดการสถานะของแอปพลิเคชัน (Vuex)

ช่วยจัดเก็บและจัดการข้อมูลที่ใช้ร่วมกันระหว่างคอมโพเนนต์ต่าง ๆ ทำให้การบริหารจัดการข้อมูลมีความเป็นระเบียบ โดยเฉพาะในแอปพลิเคชันขนาดใหญ่ที่มีข้อมูลซับซ้อน Vuex ช่วยให้สามารถจัดเก็บสถานะของแอปพลิเคชันในที่เดียว ทำให้การดีบั๊กและการจัดการข้อมูลมีประสิทธิภาพมากขึ้น

### 2.18.2 ข้อดีของ Vue.js

1. เรียนรู้ง่าย Vue.js มีโครงสร้างที่เข้าใจง่าย เหมาะสำหรับทั้งมือใหม่และนักพัฒนาที่มีประสบการณ์ สามารถเริ่มต้นใช้งานได้อย่างรวดเร็วโดยใช้เพียง HTML, CSS และ JavaScript พื้นฐาน
2. ขนาดเล็ก Vue.js มีขนาดไฟล์เริ่มต้นที่ประมาณ 20 KB เท่านั้น ซึ่งช่วยให้โหลดหน้าเว็บได้อย่างรวดเร็วเมื่อเทียบกับเฟรมเวิร์กอื่น ๆ เช่น Angular หรือ React ที่มีขนาดใหญ่กว่า
3. ความยืดหยุ่นสูง สามารถใช้งานได้ทั้งกับโปรเจกต์ขนาดเล็กและขนาดใหญ่ Vue สามารถทำงานร่วมกับเทคโนโลยีอื่น ๆ ได้ง่าย เช่น Laravel, Express และ Firebase
4. รองรับการทำ SSR (Server-Side Rendering) Vue.js สามารถใช้ร่วมกับ Nuxt.js เพื่อสร้างเว็บที่รองรับ SEO ได้ดีขึ้น โดยให้เบราว์เซอร์โหลดเนื้อหาจากเซิร์ฟเวอร์แทนการโหลดแบบไคลเอนต์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.19 SketchUp

SketchUp เป็นซอฟต์แวร์ออกแบบ 3D Modeling ที่ใช้งานง่ายและเป็นที่ยอมรับในงานออกแบบสถาปัตยกรรม, ตกแต่งภายใน, วิศวกรรม, และงานออกแบบผลิตภัณฑ์ แสดงดังรูปที่ 2.17



รูปที่ 2.17 SketchUp

## 2.20 UltiMaker Cura

UltiMaker Cura เป็นซอฟต์แวร์สำหรับการพิมพ์ 3 มิติที่ใช้งานฟรีและง่าย ซึ่งได้รับความไว้วางใจจากผู้ใช้หลายล้านคน ปรับแต่งโมเดล 3 มิติของคุณได้อย่างละเอียดด้วยการตั้งค่ากว่า 400 รายการ เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดทั้งในขั้นตอนการแบ่งชั้น (slicing) และการพิมพ์ แสดงดังรูปที่ 2.18



รูปที่ 2.18 UltiMaker Cura

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบและการจัดทำปฏิญานิพนธ์

#### 3.1 การออกแบบ

##### 3.1.1 การออกแบบระบบ

ปฏิญานิพนธ์นี้ต้องการนำการประมวลผลด้วยภาพ (Image Processing) มาใช้เพื่อระบุใบหน้าของมนุษย์เพื่อนำมาต่อยอดในการสร้างระบบจัดการการเข้าออกงานของพนักงาน ซึ่งนอกจากการระบุตัวบุคคลแล้วยังต้องออกแบบระบบเพื่อป้องกันการปลอมแปลงการลงเวลาโดยใช้รูปภาพในการสแกนแทนตัวบุคคลจริงจึงได้นำการตรวจจับความร้อนมาใช้พิจารณาด้วย โดยบล็อกไดอะแกรมของระบบแสดงดังรูปที่ 3.1

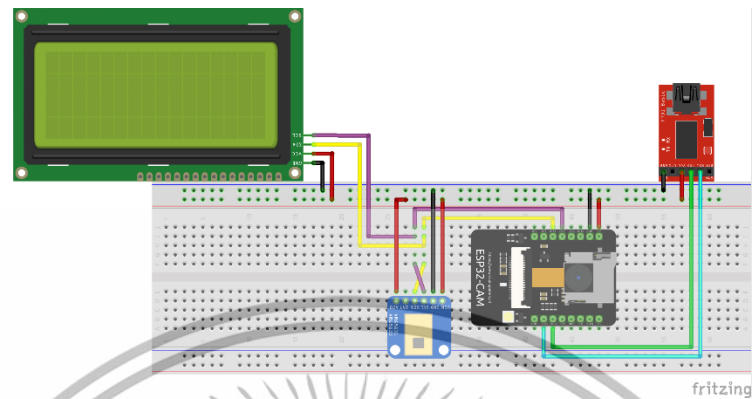


รูปที่ 3.1 บล็อกไดอะแกรมของระบบจดจำใบหน้าเพื่อจัดการการเข้าออกงาน

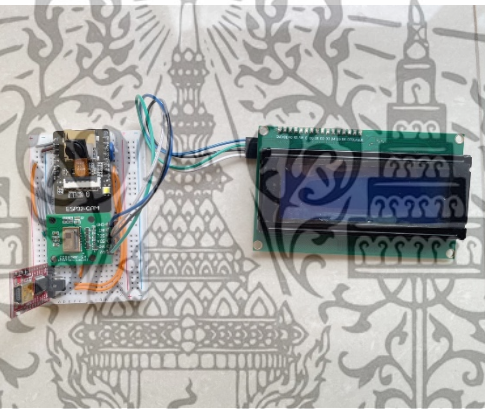
##### 3.1.2 การออกแบบการเชื่อมต่ออุปกรณ์

ในส่วนของการออกแบบการเชื่อมต่ออุปกรณ์เริ่มจากการเลือกอุปกรณ์ที่จะทำมาใช้ในโครงงาน โดยจะมี ESP32-CAM, โมดูลกล้องอินฟราเรดวัดอุณหภูมิ (AMG8853), จอ LCD และ FTDI จากนั้นทำการจำลองการเชื่อมต่ออุปกรณ์บนโปรแกรม Fritzing แสดงดังรูป 3.2 เมื่อออกแบบสำเร็จแล้ว นำมาเชื่อมต่อจริงเพื่อใช้งานการทดสอบ แสดงดังรูป 3.3 อุปกรณ์ทั้งหมดจะต่อไฟเลี้ยงและกราวด์ร่วมกันโดยจะจ่ายไฟผ่าน FTDI นอกจากนี้ FTDI ยังใช้ในการนำเข้าโค้ดจาก Arduino IDE สู่ ESP32-CAM จึงมีการเชื่อมต่อ TX ของ FTDI กับ RX ของ ESP32-CAM และเชื่อมต่อ RX ของ FTDI กับ TX ของ ESP32-CAM และในส่วนของ โมดูลกล้องอินฟราเรดวัดอุณหภูมิ (AMG8853) และจอ LCD จะเป็นอุปกรณ์ที่สื่อสารแบบ I2C ซึ่งสามารถเชื่อมต่อ SDA และ SCL ร่วมกันได้แต่เนื่องจาก ESP32-CAM ไม่มีขา SDA และ SCL ที่กำหนดชัดเจนจึงจำเป็นต้องเขียนคำสั่ง Wire.begin(14, 15); กำหนดไว้ในฟังก์ชัน void setup() คือกำหนดให้ขา 14 คือ SDA และขา 15 คือ SCL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การออกแบบการเชื่อมต่ออุปกรณ์บน Fritzing



รูปที่ 3.3 การเชื่อมต่ออุปกรณ์

### 3.1.3 การออกแบบการเชื่อมต่อกับฐานข้อมูล

#### 3.1.3.1 การออกเชื่อมต่อกับ Firebase Realtime Database

โค้ดนี้เป็นโปรแกรม Python ที่ใช้ Firebase Realtime Database ในการสร้างข้อมูลผู้ใช้ใหม่แสดงดังรูปที่ 3.4 โดยมีขั้นตอนดังนี้

1. นำเข้าโมดูล `firebase_admin` ซึ่งใช้ในการเชื่อมต่อกับ Firebase นำเข้า `credentials` สำหรับการรับรองตัวตน `db` สำหรับการเชื่อมต่อกับ Firebase Realtime Database และนำเข้า `datetime` เพื่อใช้ในการบันทึกเวลาปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โหลดไฟล์ serviceAccountKey.json ซึ่งเป็นไฟล์ที่ใช้รับรองตัวตนของโปรเจกต์ Firebase เพื่อเชื่อมต่อกับฐานข้อมูลและเรียกใช้ initialize\_app เพื่อเริ่มต้นแอปพลิเคชัน Firebase โดยกำหนด databaseURL ของ Firebase Realtime Database ที่ต้องการเชื่อมต่อ

3. สร้างฟังก์ชัน add\_employee สร้างข้อมูลผู้ใช้ใหม่ใน Firebase Realtime Database ภายใต้เรพเพอเรนซ์ employees

4. ในฟังก์ชันจะมีให้มีให้กรอกข้อมูลของพนักงาน ID ของพนักงาน, ชื่อของพนักงาน, แผนก และตำแหน่งงาน เสร็จแล้วจะนำไปเก็บไว้ใน employee\_data

5. ทำการเพิ่มผู้ใช้ใหม่เข้าไปในฐานข้อมูลโดยใช้ employee\_data เป็นคีย์ของผู้ใช้แต่ละคน

6. เรียกใช้ฟังก์ชัน add\_employee() เพื่อสร้างข้อมูลผู้ใช้ใหม่ขึ้นไปบน Firebase Realtime Database [23]



```
import firebase_admin
from firebase_admin import credentials, db
cred = credentials.Certificate("path/to/serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://backend-1e9af-default-rtdb.asia-southeast1.firebaseio.com/'
})

def add_employee():
    emp_id = input("ID: ")
    name = input("Name: ")
    department = input("Department: ")
    position = input("Position: ")

    employee_data = {
        "department": department,
        "id": emp_id,
        "name": name,
        "position": position
    }

    ref = db.reference(f'employees/{emp_id}/attendance')
    ref.set(employee_data)

    print(f"Add (name: {name}, ID: {emp_id}) Success !")

add_employee()
```

รูปที่ 3.4 คำสั่งการสร้างฐานข้อมูลบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.4 การออกแบบการตรวจสอบใบหน้า

#### 3.1.4.1 การเขียนโปรแกรมจับภาพ

การเชื่อมต่อระหว่างกล้อง กับ ESP32-CAM AI-Thinker ใน Arduino IDE จะสามารถตั้งค่าได้ โดยเมื่อ ESP32-CAM เชื่อมต่อกับ WiFi จะทำการแสดง URL สำหรับเข้าถึง ภาพบน Serial Terminal แสดงดังรูปที่ 3.5

```
Connecting to WiFi...
Connected to the WiFi network
The client esp32-client-EG:64:C9:5C:EE:94 connects to the public mqtt broker
Public emqx mqtt broker connected
CAMERA OK
http://172.20.10.4
/cam-lo.jpg
/cam-hi.jpg
/cam-mid.jpg
```

รูปที่ 3.5 การแสดง URL สำหรับเข้าถึงภาพบน Serial Terminal

โดยในส่วนของโปรแกรมที่ใช้ประมวลผลจะใช้ URL ที่กล่าวข้างต้นในการ นำเข้ารูปภาพ ดังรูปที่ 3.6 เพื่อนำไปใช้ร่วมกับส่วนตรวจสอบอุณหภูมิต่อไป [24]

```
# Capture image from ESP32-CAM
img_resp = urllib.request.urlopen(url)
imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)
frame = cv2.imdecode(imgnp, -1)
```

รูปที่ 3.6 คำสั่งการนำเข้ารูปภาพมายังโปรแกรมประมวลผล

#### 3.1.4.2 การเก็บภาพใบหน้า

ในส่วนการเก็บภาพใบหน้านั้นจะใช้กล้อง เว็บแคม (Webcam) จาก แล็ปท็อปเพื่อความสะดวกและไวเพราะต้องใช้รูปจำนวนมาก โดยจะใช้โค้ดแสดงดังรูป 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

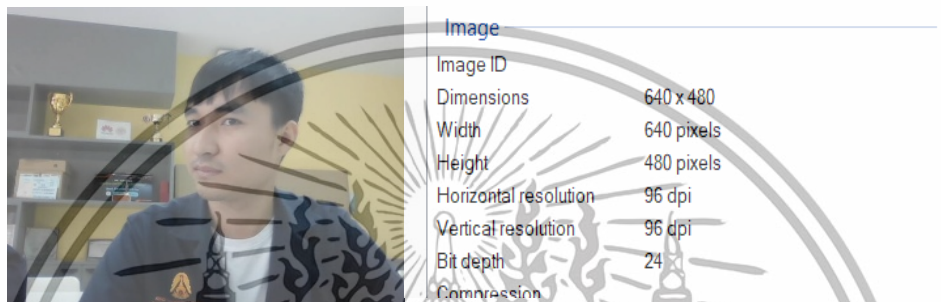
1 import cv2
2 import os
3 import time
4
5 def wait_for_camera_ready(cap):
6     print("Waiting for camera to be ready... ")
7     while True:
8         ret, frame = cap.read()
9         if ret: # If a valid frame is returned
10            print("Camera is ready!")
11            break
12            time.sleep(1) # Wait for 1 second before checking again
13
14 def capture_and_save_images(cap, num_images, folder_name):
15     if not os.path.exists(folder_name):
16         os.makedirs(folder_name)
17
18     for i in range(num_images):
19         ret, frame = cap.read()
20         if not ret:
21             print("Failed to capture an image. Check your webcam connection.")
22             break
23
24         cv2.imshow("Webcam Feed", frame)
25
26         image_filename = os.path.join(folder_name, f"captured_image_{i+1}.jpg")
27         if not cv2.imwrite(image_filename, frame):
28             print(f"Failed to save image {i+1}. Check folder permissions.")
29             break
30
31         print(f"Capturing image {i+1} of {num_images}", end="\r")
32
33         key = cv2.waitKey(1000) & 0xFF # Adds a 1-second delay
34         if key == 27: # Escape key
35             print("\nExiting...")
36             break
37
38     print(f"\nSaving images complete. Total images saved: {i+1}/{num_images}")
39
40 def main():
41     camera_index = int(input("Enter camera index (default is 0): ") or 0)
42     cap = cv2.VideoCapture(camera_index)
43
44     if not cap.isOpened():
45         print(f"Could not access the webcam with index {camera_index}")
46         return
47
48     try:
49         wait_for_camera_ready(cap)
50
51         num_images = int(input(
52             "How many images do you want to capture and save? ")
53         )
54         if num_images <= 0:
55             raise ValueError("Number of images must be greater than zero.")
56         folder_name = input("Enter the folder name to save images on your PC: ")
57
58         capture_and_save_images(cap, num_images, folder_name)
59     except ValueError as e:
60         print(f"Error: {e}")
61     finally:
62         cap.release()
63         cv2.destroyAllWindows()
64         print("Resources released. Goodbye!")
65
66 if __name__ == "__main__":
67     main()

```

รูปที่ 3.7 คำสั่งการบันทึกและเก็บภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดนี้เป็นโปรแกรมใช้กล้องเว็บแคมถ่ายภาพและบันทึกลงไฟล์เดสก์ทอปที่ผู้ใช้กำหนด เริ่มจากตรวจสอบว่ากล้องพร้อมหรือไม่ จากนั้นรับจำนวนภาพและชื่อไฟล์เดสก์ทอป แล้วถ่ายภาพตามจำนวนที่ระบุ โดยบันทึกภาพเป็นไฟล์ .jpg ทีละภาพ พร้อมแสดงภาพจากกล้องแบบเรียลไทม์ ผู้ใช้สามารถกด ESC เพื่อหยุดก่อนครบจำนวนได้ เมื่อเสร็จแล้วจะปิดกล้องและหน้าต่างทั้งหมดอัตโนมัติ ภาพที่บันทึกมาได้จะมีขนาด 640X480 pixel แสดงดังรูป 3.8



รูปที่ 3.8 ตัวอย่างและรายละเอียดของรูป

#### 3.1.4.3 การเขียนโปรแกรมเทรนใบหน้า

หลังจากได้รูปหน้าคนตามจำนวนที่ต้องการแล้วก็จะทำการเตรียมข้อมูลใบหน้าโค้ดนี้ใช้สำหรับเตรียมข้อมูลใบหน้าสำหรับระบบรู้จำใบหน้า โดยเริ่มจากการโหลดรูปภาพใบหน้าจากไฟล์เดสก์ทอปที่กำหนด และใช้ MTCNN เพื่อตรวจจับและตัดเฉพาะส่วนใบหน้า จากนั้นปรับขนาดให้พอดีกับโมเดล FaceNet ที่จะใช้ในการแปลงภาพเป็นเวกเตอร์ตัวเลข ซึ่งเป็นลักษณะเฉพาะของแต่ละใบหน้า เมื่อโหลดภาพเสร็จ โปรแกรมจะนำภาพทั้งหมดไปแปลงเป็น embedding ซึ่งเป็นชุดค่าตัวเลขที่แทนคุณลักษณะของใบหน้าแต่ละคน และบันทึกผลลัพธ์ลงไฟล์ .npz เพื่อให้สามารถนำไปใช้กับโมเดลอื่นๆ เช่น SVM หรือ Neural Network ในการจำแนกบุคคลต่อไป แสดงดังรูป 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1 import os
2 import cv2 as cv
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from mtcnn.mtcnn import MTCNN
6 from keras_facenet import FaceNet
7
8 class FACELOADING:
9     def __init__(self, directory):
10         self.directory = directory
11         self.target_size = (160, 160)
12         self.X = []
13         self.Y = []
14         self.detector = MTCNN()
15     def extract_face(self, filename):
16         img = cv.imread(filename)
17         img = cv.cvtColor(img, cv.COLOR_BGR2RGB)
18         faces = self.detector.detect_faces(img)
19         if len(faces) == 0:
20             print(f"No faces detected in {filename}")
21             return None
22         x, y, w, h = faces[0]['box']
23         x, y = abs(x), abs(y)
24         face = img[y:y+h, x:x+w]
25         face_arr = cv.resize(face, self.target_size)
26         return face_arr
27     def load_faces(self, dir):
28         FACES = []
29         for im_name in os.listdir(dir):
30             try:
31                 path = os.path.join(dir, im_name)
32                 single_face = self.extract_face(path)
33                 if single_face is not None:
34                     FACES.append(single_face)
35             except Exception as e:
36                 print(f"Error processing {im_name}: {e}")
37         return FACES
38     def load_classes(self):
39         for sub_dir in os.listdir(self.directory):
40             path = os.path.join(self.directory, sub_dir)
41             FACES = self.load_faces(path)
42             labels = [sub_dir for _ in range(len(FACES))]
43             print(f"Loaded successfully: {len(labels)}")
44             self.X.extend(FACES)
45             self.Y.extend(labels)
46         return np.asarray(self.X), np.asarray(self.Y)
47     def plot_images(self):
48         plt.figure(figsize=(18, 16))
49         for num, image in enumerate(self.X):
50             ncols = 3
51             nrows = len(self.Y) // ncols + 1
52             plt.subplot(nrows, ncols, num + 1)
53             plt.imshow(image)
54             plt.axis('off')
55
56 faceloading = FACELOADING(r"D:\IndiaFACE\FACERECVSM\DATASET")
57 X, Y = faceloading.load_classes()
58 embedder = FaceNet()
59 def get_embedding(face_img):
60     face_img = face_img.astype('float32') # 3D(160x160x3)
61     face_img = np.expand_dims(face_img, axis=0) # 4D (None, 160, 160, 3)
62     yhat = embedder.embeddings(face_img)
63     return yhat[0] # 512D image (1x1x512)
64 EMBEDDED_X = []
65 for img in X:
66     EMBEDDED_X.append(get_embedding(img))
67 EMBEDDED_X = np.asarray(EMBEDDED_X)
68 np.savez_compressed('faces_embeddings_done_4classes.npz', EMBEDDED_X, Y)

```

รูปที่ 3.9 คำสั่งสำหรับการเตรียมข้อมูลใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากได้ไฟล์ข้อมูลใบหน้าแล้วจะทำการโหลดข้อมูลใบหน้าที่ถูกแปลงเป็นเวกเตอร์เอ็มเบดดิ้ง (EMBEDDED\_X) พร้อมป้ายกำกับ (Y), เซอร์ทิสป้ายกำกับเป็นตัวเลข, แบ่งข้อมูลเป็นชุดฝึกและทดสอบ, ฝึกโมเดล และบันทึกโมเดลที่ฝึกแล้วลงไฟล์ svm\_model500v2.pkl เพื่อใช้งานภายหลังดังรูป 3.10

```

1 import os
2 import pickle
3 import numpy as np
4 import cv2 as cv
5 from mtcnn.mtcnn import MTCNN
6 from keras_facenet import FaceNet
7 from sklearn.model_selection import train_test_split
8 from sklearn.preprocessing import LabelEncoder
9 from sklearn.svm import SVC
10
11 data = np.load('faces_embeddings_done_4classes500v2.npz')
12
13 EMBEDDED_X = data['arr_0']
14 Y = data['arr_1']
15
16 encoder = LabelEncoder()
17 encoder.fit(Y)
18 Y = encoder.transform(Y)
19
20 X_train, X_test, Y_train, Y_test = train_test_split(
21     EMBEDDED_X, Y, shuffle=True, random_state=17
22 )
23 model = SVC(C=0.7, kernel='rbf', gamma='scale', probability=True)
24 model.fit(X_train, Y_train)
25
26 with open('svm_model500v2.pkl', 'wb') as f:
27     pickle.dump(model, f)

```

รูปที่ 3.10 คำสั่งการเทรนโมเดล

โดยในแต่ละโมเดลก็จะกำหนดค่าพารามิเตอร์ดังรูปที่ 3.11

```

1 svm_model = SVC(C=0.7, kernel='rbf', gamma='scale', probability=True)
2 knn_model = KNeighborsClassifier(n_neighbors=5)
3 rf_model = RandomForestClassifier(n_estimators=100, random_state=17)
4 gb_model = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1,
5     random_state=17)

```

รูปที่ 3.11 ค่าพารามิเตอร์แต่ละโมเดลที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### SVM (Support Vector Machine)

- ใช้ RBF Kernel (Radial Basis Function) เพื่อให้สามารถแยกข้อมูลที่ไม่เป็นเส้นตรงได้ดี
- $C=0.7$  เป็นค่าควบคุมความเข้มงวดของเส้นแบ่งข้อมูล (Regularization)
- `probability=True` ช่วยให้สามารถใช้ `predict_proba()` เพื่อดูค่าความมั่นใจของการพยากรณ์ได้

### KNN (K-Nearest Neighbors)

- ใช้  $k=5$  หมายถึงพิจารณา 5 ตัวใกล้ที่สุด ในการตัดสินใจจำแนกข้อมูล

### Random Forest (RF)

- ใช้ 100 ต้นไม้ (`n_estimators=100`) เพื่อรวมผลการทำนายจากหลายต้นไม้
- `random_state=17` ใช้กำหนดค่าเริ่มต้นของการสุ่มเพื่อให้ผลลัพธ์เหมือนกันทุกครั้ง

### Gradient Boosting (GB)

- `n_estimators=100` หมายถึงจำนวนต้นไม้ตัดสินใจ (Decision Trees) ที่โมเดลจะสร้างขึ้น
- `learning_rate=0.1` ควบคุมความเร็วในการเรียนรู้
- `random_state=17` ใช้กำหนดค่าเริ่มต้นของการสุ่มเพื่อให้ผลลัพธ์เหมือนกันทุกครั้ง

#### 3.1.4.4 การเขียนโปรแกรมตรวจสอบใบหน้า

โค้ดส่วนนี้เริ่มต้นด้วยการนำเข้าไลบรารีต่างๆ ที่จำเป็น เช่น `cv2` สำหรับการประมวลผลภาพ, `numpy` สำหรับการทำงานกับอาร์เรย์, `tensorflow` สำหรับการใช้งานโมเดล Machine Learning, `firebase_admin` สำหรับการเชื่อมต่อกับ Firebase, และ `paho.mqtt.client` สำหรับการสื่อสารผ่าน MQTT นอกจากนี้ยังมีการโหลดโมเดล FaceNet และ SVM ที่ใช้ในการแปลงรูปภาพใบหน้าให้เป็นเวกเตอร์และจำแนกใบหน้า โดยโมเดล FaceNet จะทำหน้าที่แปลงรูปภาพใบหน้าให้เป็นเวกเตอร์ (embedding) ซึ่งจะถูกนำไปใช้ในการเปรียบเทียบกับใบหน้าที่มีอยู่ในฐานข้อมูล ส่วนโมเดล SVM จะทำหน้าที่จำแนกว่าเวกเตอร์ที่ได้มาจาก FaceNet ตรงกับบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใดในฐานข้อมูล นอกจากนี้ยังมีการโหลด Haar Cascade ซึ่งเป็นอัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในภาพ และ LabelEncoder ที่ใช้ในการแปลงชื่อบุคคลให้เป็นตัวเลขและแปลงกลับ แสดงดังรูปที่ 3.12

```
import cv2
import numpy as np
import os
import pickle
import tensorflow as tf
import firebase_admin
from firebase_admin import credentials, db
from datetime import datetime, timedelta
from sklearn.preprocessing import LabelEncoder
from keras_facenet import FaceNet
from collections import defaultdict
import time
import schedule
import paho.mqtt.client as mqtt
import urllib.request

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

facenet = FaceNet()
faces_embeddings = np.load("faces_embeddings_done_4classes500.npz")
Y = faces_embeddings['arr_1']
encoder = LabelEncoder()
encoder.fit(Y)
haarcascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
model = pickle.load(open("svm_model500.pkl", "rb"))
```

รูปที่ 3.12 คำสั่งการโหลดโมเดลและเตรียมข้อมูล

#### 3.1.4.5 การเชื่อมต่อ Firebase

โค้ดส่วนนี้ทำการเชื่อมต่อกับ Firebase Realtime Database โดยใช้ credentials ที่ได้จากไฟล์ serviceAccountKey.json เพื่อเข้าถึงฐานข้อมูล Firebase ซึ่งจะใช้ในการเก็บข้อมูลพนักงานและบันทึกการลงเวลา การเชื่อมต่อ Firebase โดยระบบจะใช้ฐานข้อมูลนี้ในการบันทึกสถานะการทำงานของพนักงาน เช่น เวลาของการลงเวลาเข้าออกงาน, สถานะการเข้าออกงาน แสดงดังรูปที่ 3.13

```
# Firebase Initialization
cred = credentials.Certificate('D:/FACERECOGNITION/Python/serviceAccountKey.json')
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://backend-1e9af-default-rtdb.asia-southeast1.firebaseio.com/'
})
```

รูปที่ 3.13 คำสั่งการเชื่อมต่อ Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.4.6 การตรวจสอบการขาดงาน

ฟังก์ชัน `mark_absent_if_needed` จะถูกเรียกใช้ทุกวันตอน 00:01 AM เพื่อตรวจสอบว่ามีพนักงานคนใดที่ไม่ได้ลงเวลาในวันก่อนหน้า หากพบว่าพนักงานที่ไม่ได้ลงเวลา ระบบจะทำการบันทึกสถานะเป็น "Absent" ใน Firebase โดยฟังก์ชันนี้จะดึงข้อมูลพนักงานจาก Firebase และตรวจสอบว่ามีกรลงเวลาในวันก่อนหน้าหรือไม่ หากไม่มีการลงเวลา ระบบจะบันทึกสถานะ "Absent" พร้อมกับวันที่ที่ขาดงาน แสดงดังรูปที่ 3.14

```
def mark_absent_if_needed():
    """ ตรวจสอบว่ามีใครไม่ได้ลงเวลาวันนี้และกำหนดสถานะเป็น 'Absent' """
    employees_ref = db.reference('employees')
    employees_data = employees_ref.get()

    if not isinstance(employees_data, list):
        print("Error: Employee data format is incorrect.")
        return

    today = datetime.now()
    yesterday = today - timedelta(days=1)
    yesterday_str = yesterday.strftime("%Y-%m-%d")

    for index, employee in enumerate(employees_data):
        attendance_ref = db.reference(f'employees/{index}/attendance')
        existing_data = attendance_ref.get()

        if isinstance(existing_data, list):
            already_checked_in = any(record.get("date") == yesterday_str for record in existing_data)
        else:
            already_checked_in = False

        if not already_checked_in:
            # บันทึกสถานะ Absent
            next_index = len(existing_data) if existing_data else 0
            attendance_ref.child(str(next_index)).set({
                "date": yesterday_str,
                "status": "Absent"
            })
            print(f"✘ Marked {employee.get('name', 'Unknown')} as Absent for {yesterday_str}")

    # ตั้งให้รันอัตโนมัติทุกวันตอน 00:01 AM
    schedule.every().day.at("00:01").do(mark_absent_if_needed)

    print(f"✘ ระบบกำลังรันอัตโนมัติเพื่อตรวจสอบพนักงานที่ขาดงาน...")
```

รูปที่ 3.14 คำสั่งตรวจสอบการขาดงาน

### 3.1.4.7 การลงเวลา

ฟังก์ชัน `log_attendance` ใช้ในการบันทึกการลงเวลาของพนักงาน โดยระบบจะตรวจสอบว่าพนักงานได้ลงเวลาเข้าและออกงานหรือไม่ และจะบันทึกสถานะเป็น "On Time" หรือ "Late" ตามเวลาที่ลงเวลา หากพนักงานลงเวลาเข้างานก่อน 09:00 น. สถานะจะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกเป็น "On Time" แต่หากลงเวลาทำงานหลัง 09:00 น. สถานะจะถูกบันทึกเป็น "Late" นอกจากนี้ระบบยังตรวจสอบว่าพนักงานได้ลงเวลาออกงานหรือไม่ หากพนักงานลงเวลาทำงานแต่ยังไม่ได้ลงเวลาออกงาน และเวลาปัจจุบันเกิน 12:01 PM ระบบจะบันทึกเวลาออกงานให้โดยอัตโนมัติ แสดงดังรูปที่ 3.15

```
def log_attendance(name):
    employees_ref = db.reference('employees')
    employees_data = employees_ref.get()

    if not isinstance(employees_data, list):
        print("Error: Employee data format is incorrect.")
        return

    employee_index = None
    for index, employee in enumerate(employees_data):
        if str(employee.get('name')) == str(name):
            employee_index = index
            break

    if employee_index is None:
        print(f"Employee name {name} not found!")
        return

    now = datetime.now()
    date = now.strftime("%Y-%m-%d")
    current_time = now.strftime("%H:%M")

    attendance_ref = db.reference(f'employees/{employee_index}/attendance')
    existing_data = attendance_ref.get()

    if isinstance(existing_data, list):
        for idx, record in enumerate(existing_data):
            if record.get("date") == date:
                if "time_in" in record and current_time < "12:01":
                    print(f"{name} already checked in before 12:01 PM. No update needed.")
                    return

                if "time_in" in record and "time_out" not in record and current_time >= "12:01":
                    time.sleep(3) # Delay 3 วินาที
                    attendance_ref.child(str(idx)).set({
                        "date": record["date"],
                        "time_in": record["time_in"],
                        "status": record["status"],
                        "time_out": current_time
                    })
                    print(f"Updated time_out for {name} at {current_time}")
                    return

                if "time_out" in record:
                    print(f"{name} already has time_out recorded: {record['time_out']}")
                    return

    status = "Late" if current_time > "09:00" else "On Time"
    next_index = len(existing_data) if existing_data else 0

    time.sleep(3) # Delay 3 วินาที
    attendance_ref.child(str(next_index)).set({
        "date": date,
        "time_in": current_time,
        "status": status
    })
    print(f"Added attendance record for {name}: {status}")
```

รูปที่ 3.15 คำสั่งการลงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.4.8 การกำหนดค่าคงที่

โค้ดส่วนนี้เริ่มต้นด้วยการกำหนดค่าคงที่ต่างๆ ที่ใช้ในระบบ เช่น THRESHOLD ซึ่งเป็นค่าความน่าจะเป็นขั้นต่ำที่ใช้ในการตัดสินใจว่าใบหน้าที่ตรวจจับได้ตรงกับบุคคลในฐานข้อมูลหรือไม่ หากค่าความน่าจะเป็นต่ำกว่า 0.65 ระบบจะถือว่าใบหน้านั้นไม่ตรงกับบุคคลใดในฐานข้อมูล และ TEMPERATURE\_VARIANCE\_THRESHOLD เป็นค่าความแปรปรวนของอุณหภูมิขั้นต่ำที่ใช้ในการตัดสินใจว่าบุคคลที่ตรวจจับได้เป็นมนุษย์จริงหรือไม่ หากความแปรปรวนของอุณหภูมิในบริเวณใบหน้ามีค่าสูงกว่า 0.5 ระบบจะถือว่าบุคคลนั้นเป็นมนุษย์จริง แสดงดังรูปที่ 3.16

THRESHOLD = 0.65  
TEMPERATURE\_VARIANCE\_THRESHOLD = 0.5

รูปที่ 3.16 คำสั่งการกำหนดค่าคงที่ (Constants)

### 3.1.4.9 การตรวจจับใบหน้าและตรวจสอบอุณหภูมิ

ระบบจะรับข้อมูลอุณหภูมิจากเซ็นเซอร์ AMG8853 ผ่าน MQTT โดยข้อมูลอุณหภูมิจะถูกส่งมาในรูปแบบของข้อความที่ประกอบด้วยค่าอุณหภูมิ 8 ค่า (1 แถว) และระบบจะเก็บข้อมูลนี้ไว้ในอาร์เรย์ temperature\_data จนครบ 8 แถว เพื่อสร้างเป็นอาร์เรย์อุณหภูมิขนาด 8x8 จากนั้นระบบจะใช้ข้อมูลอุณหภูมินี้ในการตรวจสอบว่าบุคคลที่ตรวจจับได้เป็นมนุษย์จริงหรือไม่ โดยดูจากความแปรปรวนของอุณหภูมิในบริเวณใบหน้า หากความแปรปรวนของอุณหภูมิในบริเวณใบหน้ามีค่าสูงกว่าค่าที่กำหนด ระบบจะตัดสินใจว่าบุคคลนั้นเป็นมนุษย์จริง แสดงดังรูปที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def on_message(client, userdata, message):
    global temperature_data
    if message.topic == MQTT_TOPIC_SUB1:
        data_str = message.payload.decode()
        data_list = [float(value) for value in data_str.split(',')]
        if len(temperature_data) < 8:
            temperature_data.append(data_list)
        if len(temperature_data) >= 8:
            print("Received 8x8 Temperature Grid:")
            for row in temperature_data:
                print(row)

client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2)
client.on_message = on_message
client.connect(MQTT_BROKER, MQTT_PORT)
client.subscribe(MQTT_TOPIC_SUB1)
client.loop_start()

```

รูปที่ 3.17 คำสั่งการตรวจจับใบหน้าและตรวจสอบอุณหภูมิ

#### 3.1.4.10 ฟังก์ชัน bilinear\_interpolation

ฟังก์ชัน bilinear\_interpolation ใช้ในการปรับขนาดอาร์เรย์อุณหภูมิจากขนาด 8x8 เป็นขนาดที่ใหญ่ขึ้น โดยใช้วิธีการ Interpolation เพื่อให้ได้ข้อมูลอุณหภูมิที่ละเอียดมากขึ้น วิธีการนี้จะคำนวณค่าอุณหภูมิที่ตำแหน่งใหม่โดยใช้ค่าอุณหภูมิจากตำแหน่งรอบๆ และคำนวณค่าเฉลี่ยตามน้ำหนักที่ได้จากระยะห่างระหว่างตำแหน่งเดิมและตำแหน่งใหม่ ตัวอย่างเช่น หากต้องการปรับขนาดอาร์เรย์อุณหภูมิจาก 8x8 เป็น 30x30 ฟังก์ชันนี้จะคำนวณค่าอุณหภูมิที่ตำแหน่งใหม่โดยใช้ค่าอุณหภูมิจากตำแหน่งรอบๆ และคำนวณค่าเฉลี่ยตามน้ำหนักที่ได้จากระยะห่างระหว่างตำแหน่งเดิมและตำแหน่งใหม่ แสดงดังรูปที่ 3.18 และแสดง Heat Map ของอุณหภูมิดังภาพที่ 3.19

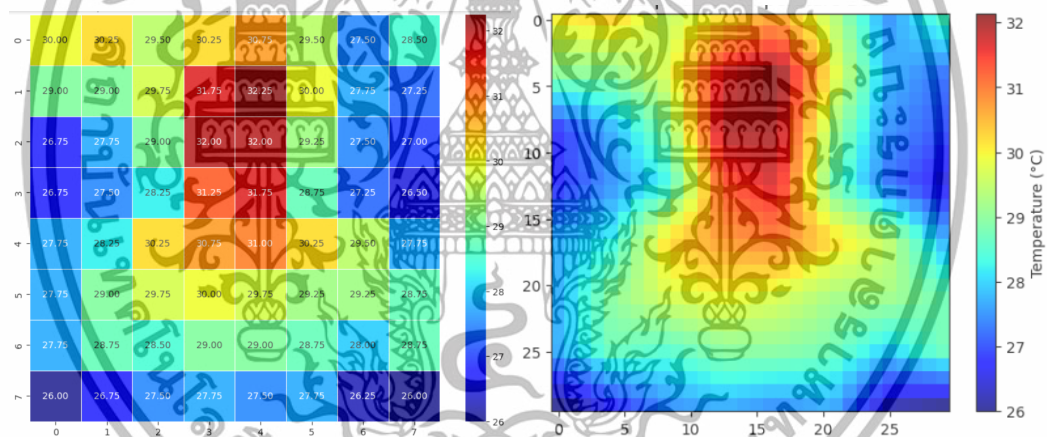
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def bilinear_interpolation(input_array, output_size):
    input_size = input_array.shape[0]
    scale = (output_size - 1) / (input_size - 1)
    output_array = np.zeros((output_size, output_size))
    for i in range(output_size):
        for j in range(output_size):
            x = i / scale
            y = j / scale
            x0, y0 = int(x), int(y)
            x1, y1 = min(x0 + 1, input_size - 1), min(y0 + 1, input_size - 1)
            dx, dy = x - x0, y - y0
            value = (
                input_array[x0, y0] * (1 - dx) * (1 - dy) +
                input_array[x0, y1] * (1 - dx) * dy +
                input_array[x1, y0] * dx * (1 - dy) +
                input_array[x1, y1] * dx * dy
            )
            output_array[i, j] = value
    return output_array

```

รูปที่ 3.18 คำสั่งฟังก์ชัน bilinear\_interpolation



รูปที่ 3.19 (a) Heat Map ของอุณหภูมิที่วัดจากเซ็นเซอร์ AMG 8853

(b) Heat Map ของอุณหภูมิที่ผ่านฟังก์ชัน bilinear\_interpolation

#### 3.1.4.11 การตั้งค่ากล้องและ MQTT

โค้ดส่วนนี้ทำการกำหนดค่าต่างๆ ที่เกี่ยวข้องกับการตั้งค่ากล้องและ MQTT เช่น url ซึ่งเป็น URL ของกล้องที่ใช้ในการรับภาพ MQTT\_BROKER ซึ่งเป็นที่อยู่ของ MQTT broker ที่ใช้ในการสื่อสาร MQTT\_PORT ซึ่งเป็นพอร์ตที่ใช้ในการเชื่อมต่อกับ MQTT broker MQTT\_TOPIC\_SUB1 ซึ่งเป็นหัวข้อที่ใช้ในการรับข้อมูลอุณหภูมิจากเซ็นเซอร์ AMG8853 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MQTT\_TOPIC\_PUB ซึ่งเป็นหัวข้อที่ใช้ในการส่งข้อมูลไปยัง ESP32-CAM นอกจากนี้ยังมีการกำหนด temperature\_data ซึ่งเป็นอาร์เรย์ที่ใช้ในการเก็บข้อมูลอุณหภูมิจากเซ็นเซอร์ AMG8853 แสดงดังรูปที่ 3.20

```
url = 'http://172.20.10.2/cam-lo.jpg'
MQTT_BROKER = 'broker.emqx.io'
MQTT_PORT = 1883
MQTT_TOPIC_SUB1 = 'ESPFaceAtt/amg'
MQTT_TOPIC_PUB = 'ESPFaceAtt/pc'

temperature_data = []
```

รูปที่ 3.20 คำสั่งการตั้งค่ากล้องและ MQTT

#### 3.1.4.12 การประมวลผลใบหน้าและอุณหภูมิ

ในรูปหลักของโปรแกรม ระบบจะทำการตรวจจับใบหน้าในภาพโดยใช้ Haar Cascade จากนั้นจะใช้ FaceNet เพื่อแปลงใบหน้าที่ตรวจจับได้ให้เป็นเวกเตอร์และใช้ SVM เพื่อจำแนกว่าใบหน้านั้นตรงกับบุคคลใดในฐานข้อมูล หากใบหน้าที่ตรวจจับได้ตรงกับบุคคลในฐานข้อมูลและผ่านการตรวจสอบอุณหภูมิ ระบบจะทำการบันทึกการลงเวลาให้กับบุคคลนั้น นอกจากนี้ระบบยังแสดงผลการตรวจจับใบหน้าบนหน้าจอ โดยแสดงชื่อบุคคลและสถานะ (Real Person หรือ Picture) พร้อมกับกรอบสี่เหลี่ยมรอบใบหน้าที่ตรวจจับได้ แสดงดังรูปที่ 3.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while True:

    # Capture image from ESP32-CAM
    img_resp = urllib.request.urlopen(url)
    imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)
    frame = cv2.imdecode(imgnp, -1)

    rgb_img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    gray_img = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = haarcascade.detectMultiScale(gray_img, 1.3, 5)

    classification = "Unknown"

    for (x, y, w, h) in faces:
        img = rgb_img[y:y+h, x:x+w]
        img = cv2.resize(img, (160, 160))
        img = np.expand_dims(img, axis=0)
        ypred = facenet.embeddings(img)
        probabilities = model.predict_proba(ypred) if hasattr(model, "predict_proba") else None
        max_prob = np.max(probabilities) if probabilities is not None else np.max(model.decision_function(ypred))
        predicted_class = np.argmax(probabilities) if probabilities is not None else np.argmax(model.decision_function(ypred))
        face_name = encoder.inverse_transform([predicted_class])[0] if max_prob >= THRESHOLD else "Unknown"

        if face_name != "Unknown":
            x0, y0 = int(x * 30 / frame.shape[1]), int(y * 30 / frame.shape[0])
            x1, y1 = int((x + w) * 30 / frame.shape[1]), int((y + h) * 30 / frame.shape[0])
            face_temperature = interpolated_data[y0:y1, x0:x1]
            temperature_variance = np.var(face_temperature) if face_temperature.size > 0 else 0
            classification = "Real Person" if temperature_variance > TEMPERATURE_VARIANCE_THRESHOLD else "Picture"

            if classification == "Real Person":
                log_attendance(face_name)

        color = (0, 255, 0) if classification == "Real Person" else (0, 0, 255)
        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
        cv2.putText(frame, face_name, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, color, 2)

    schedule.run_pending()
    temperature_data = []

    cv2.imshow("Face Recognition with Attendance", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

```

รูปที่ 3.21 คำสั่งการประมวลผลใบหน้าและอุณหภูมิ

#### 3.1.4.13 การแสดงผลและปิดโปรแกรม

เมื่อผู้ใช้กดปุ่ม 'q' ระบบจะปิดหน้าต่างที่แสดงผลและหยุดการรับข้อความจาก MQTT พร้อมกับปิดการเชื่อมต่อ MQTT ซึ่งเป็นการสิ้นสุดการทำงานของโปรแกรม แสดงดังรูปที่ 3.22

```

cv2.imshow("Face Recognition with Attendance", frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

cv2.destroyAllWindows()
client.loop_stop()
client.disconnect()

```

รูปที่ 3.22 คำสั่งการแสดงผลและปิดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.5 การออกแบบการเลือกใช้ Model

โดยจะใช้ Classification Report เป็นรายงานที่ใช้ประเมินประสิทธิภาพของโมเดลจำแนกประเภท (Classification) โดยแสดงค่าทางสถิติที่สำคัญ ได้แก่ Precision, Recall, F1-score และ Support ทั้งยังมีการใช้ Cross-Validation Mean Accuracy เพิ่มเติมด้วย โดยในส่วนของ Classification Report สามารถหาได้ดังรูป 3.23

```
1 test_report = classification_report(Y_test, ypreds_test, target_names=encoder.
  classes_)
2 print("Classification Report - Test Data:")
3 print(test_report)
```

รูปที่ 3.23 คำสั่งการแสดง Classification Report

Precision คือสัดส่วนของตัวอย่างที่โมเดลทำนายว่าเป็นคลาสหนึ่ง ๆ และเป็นค่าที่ถูกต้องจริง ๆ ค่านี้ชี้วัดว่าจากทั้งหมดที่โมเดลบอกว่าเป็นคลาสนั้น มีจำนวนที่ถูกต้องจริงกี่เปอร์เซ็นต์ ถ้าค่าสูง แสดงว่าโมเดลสามารถแยกแยะได้ดีและมีโอกาสเกิดการทำนายผิดพลาดเป็นคลาสน้อยต่ำ เหมาะกับกรณีที่ต้องการลดการแจ้งเตือนผิดพลาด เช่น โมเดลที่ใช้ตรวจจับอีเมลสแปม เพราะหากทำนายผิดไปเป็นสแปมโดยไม่ใช้จริง อาจทำให้ผู้ใช้พลาดอีเมลสำคัญ

#### 3.1.5.2 Recall (ความสามารถในการตรวจจับค่าจริง)

Recall วัดว่าสำหรับตัวอย่างที่เป็นคลาสหนึ่ง ๆ จริง ๆ โมเดลสามารถตรวจจับได้ถูกต้องกี่เปอร์เซ็นต์ ค่าที่สูงแสดงว่าโมเดลพลาดการตรวจจับค่าจริงน้อย เหมาะกับกรณีที่ต้องการลดการพลาดตัวอย่างสำคัญ เช่น การตรวจจับโรคมะเร็ง หาก Recall ต่ำ หมายความว่าโมเดลพลาดการวินิจฉัยผู้ป่วยที่เป็นมะเร็งจริง ซึ่งอาจส่งผลเสียร้ายแรงได้ ในบางกรณีต้องแลกกับ Precision เพราะหาก Recall สูงเกินไป อาจเพิ่มจำนวนการแจ้งเตือนผิดพลาด

#### 3.1.5.3 F1-score (ค่าความสมดุลระหว่าง Precision และ Recall)

F1-score เป็นค่าเฉลี่ยเชิงฮาร์โมนิกระหว่าง Precision และ Recall เพื่อให้ได้ค่าที่สะท้อนประสิทธิภาพโดยรวมของโมเดล หาก Precision และ Recall มีค่าต่างกันมาก ค่า F1-score จะช่วยให้เห็นภาพรวมได้ดีขึ้น โดยค่าที่ใกล้ 1 แสดงว่าโมเดลมีความแม่นยำและสามารถตรวจจับค่าจริงได้ดีในขณะเดียวกัน ค่านี้เหมาะกับปัญหาที่ต้องการให้ Precision และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Recall มีความสมดุลกัน เช่น การตรวจจับความผิดปกติในระบบความปลอดภัย ที่ต้องลดทั้งการเตือนผิดพลาดและการพลาดเหตุการณ์จริง

#### 3.1.5.4 Support (จำนวนตัวอย่างในคลาสนั้น ๆ)

Support คือจำนวนตัวอย่างที่อยู่ในแต่ละคลาสจริง ๆ ซึ่งมีผลต่อการประเมินผลของ Precision, Recall และ F1-score หากคลาสใดมี Support ต่ำ ค่าประสิทธิภาพของโมเดลอาจไม่น่าเชื่อถือเพราะมาจากตัวอย่างจำนวนน้อย ค่านี้ช่วยให้เข้าใจว่าค่าทางสถิติที่คำนวณออกมานั้นถูกประเมินจากข้อมูลจำนวนมากพอหรือไม่ ในกรณีที่ข้อมูลไม่สมดุล ค่าทางสถิติของคลาสที่มี Support ต่ำอาจสูงหรือต่ำผิดปกติได้ ดังนั้นควรดูค่า Support ควบคู่ไปกับค่าประสิทธิภาพอื่น ๆ เพื่อให้แน่ใจว่าโมเดลสามารถทำงานได้ดีในทุกคลาส

#### 3.1.5.5 Accuracy Score

Accuracy Score เป็นตัวชี้วัดพื้นฐานที่ใช้วัดความแม่นยำ (Accuracy) ของโมเดลจำแนกประเภท (Classification) โดยคำนวณจาก สัดส่วนของจำนวนตัวอย่างที่โมเดลทำนายถูกต้องทั้งหมด เทียบกับจำนวนตัวอย่างทั้งหมด สามารถหาได้จากโค้ดดังรูป 3.24

```
1 accuracy_score(Y_test, ypreds_test)
```

รูปที่ 3.24 คำสั่งคำนวณ Accuracy score

#### 3.1.5.6 Cross-Validation Mean Accuracy

Cross-Validation Mean Accuracy เป็นค่าความแม่นยำเฉลี่ยที่ได้จากการทำ Cross-Validation ซึ่งเป็นเทคนิคที่ช่วยประเมินประสิทธิภาพของโมเดลโดยการแบ่งข้อมูลออกเป็นหลาย ๆ ส่วน แล้วใช้แต่ละส่วนในการฝึกและทดสอบโมเดลเพื่อให้ได้ค่าความแม่นยำที่มีความน่าเชื่อถือมากขึ้น สามารถหาได้จากโค้ดดังรูป 3.25

```
1 scores = cross_val_score(model,
2   EMBEDDED_X, Y, cv=5)
3 print(
4   f"Cross-Validation Mean Accuracy: {
5     scores.mean()}"
```

รูปที่ 3.25 คำสั่งคำนวณ Cross-Validation Mean Accuracy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.7 จำนวนชุดข้อมูลรูปภาพต่อคน 1 คน

ในการเตรียมการทดลองจะทำการสร้างโมเดลจากรูปภาพหน้าคนจำนวน 50 รูป, 200 รูป และ 500 รูปตามลำดับเพื่อเปรียบเทียบประสิทธิภาพของโมเดลว่าจำนวนรูปภาพมีผลต่อความถูกต้องของโมเดลเท่าใด โดยใช้ Accuracy Score เป็นตัวเปรียบเทียบ

### 3.1.8 การออกหน้าเว็บแอปพลิเคชัน (Frontend)

#### 3.1.8.1 ออกแบบหน้าเข้าสู่ระบบ (Login)

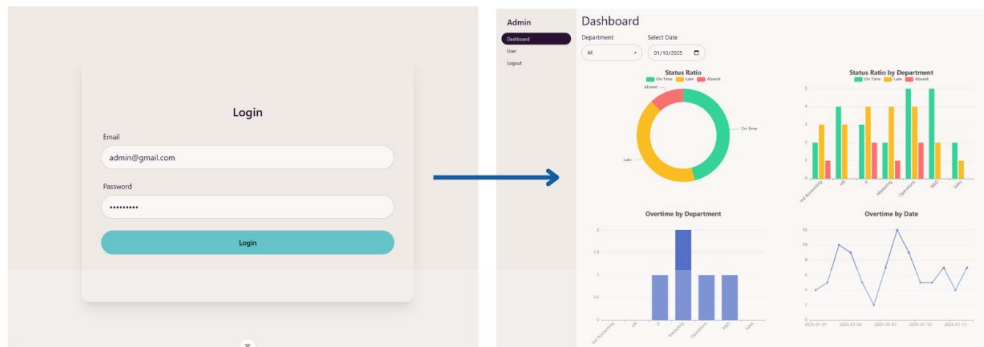
หน้าเข้าสู่ระบบจะมีรายละเอียดที่ผู้ใช้ต้องกรอกดังนี้ E-mail Password และจะมีปุ่ม Login สำหรับการเข้าสู่ระบบของผู้ใช้งาน แสดงดังรูปที่ 3.26



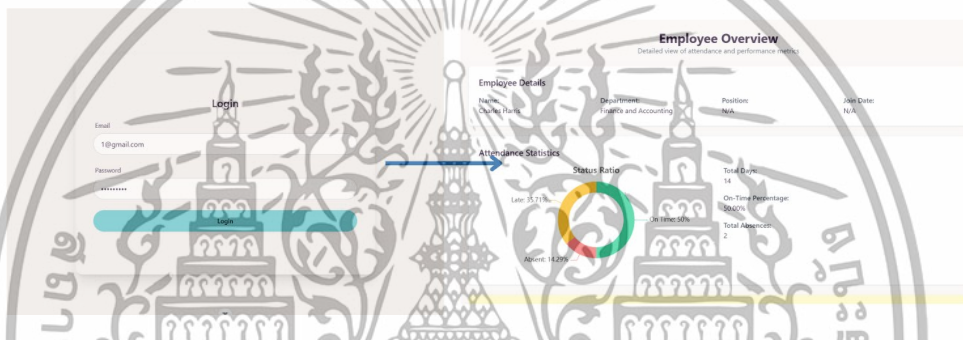
รูปที่ 3.26 หน้าเข้าสู่ระบบ (Login)

โดยเมื่อเข้าสู่ระบบจะมีการตรวจสอบประเภทของผู้ใช้งานว่าถ้าหากเป็น E-mail ของผู้ดูแลระบบ (Admin) ก็จะเข้าไปยัง Backend Dashboard เป็นส่วนหนึ่งของระบบที่ใช้สำหรับดูประวัติการลงเวลาการเข้า-ออกงานของทุกคน ซึ่งสามารถเข้าถึงได้เฉพาะผู้ดูแลระบบเท่านั้น แสดงดังรูปที่ 3.27 แต่ถ้าหากเข้าสู่ระบบด้วย E-mail ของพนักงานก็จะเข้าไปยังหน้าข้อมูลของผู้ใช้งานของแต่ละคนเท่านั้น แสดงดังรูปที่ 3.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 เข้าสู่ระบบด้วย E-mail ผู้ดูแลระบบ (Admin)



รูปที่ 3.28 เข้าสู่ระบบด้วย E-mail พนักงาน

### 3.1.8.2 ออกแบบหน้าแดชบอร์ด (Dashboard)

หน้าแดชบอร์ดจะใช้สำหรับผู้ดูแลระบบที่ออกแบบมาเพื่อแสดงสถิติการเข้าออกงานของพนักงานในแต่ละแผนก โดยมีการเลือกแผนกและวันที่ที่ต้องการดูข้อมูลได้ที่ด้านบน ภายในหน้าจอประกอบด้วยกราฟและแผนภูมิต่าง ๆ ได้แก่

1. กราฟวงแหวน Status Ratio  
เป็นกราฟที่ใช้แสดงสัดส่วนพนักงานที่มาทำงานว่าเข้างานตรงเวลา, สายหรือขาดงาน ออกแบบใช้สีที่แตกต่างกันเพื่อช่วยแยกประเภทข้อมูลได้ชัดเจน
2. กราฟแท่ง Status Ratio by Department  
เป็นกราฟแสดงข้อมูลสถานการณ์เข้างานของแต่ละแผนก โดยหากมีการเลือกแผนกที่สนใจจะเปลี่ยนเป็นกราฟ Status Ratio by Employees ซึ่งแสดงข้อมูลสถานการณ์เข้างานของแต่ละบุคคลในแผนกนั้นๆ

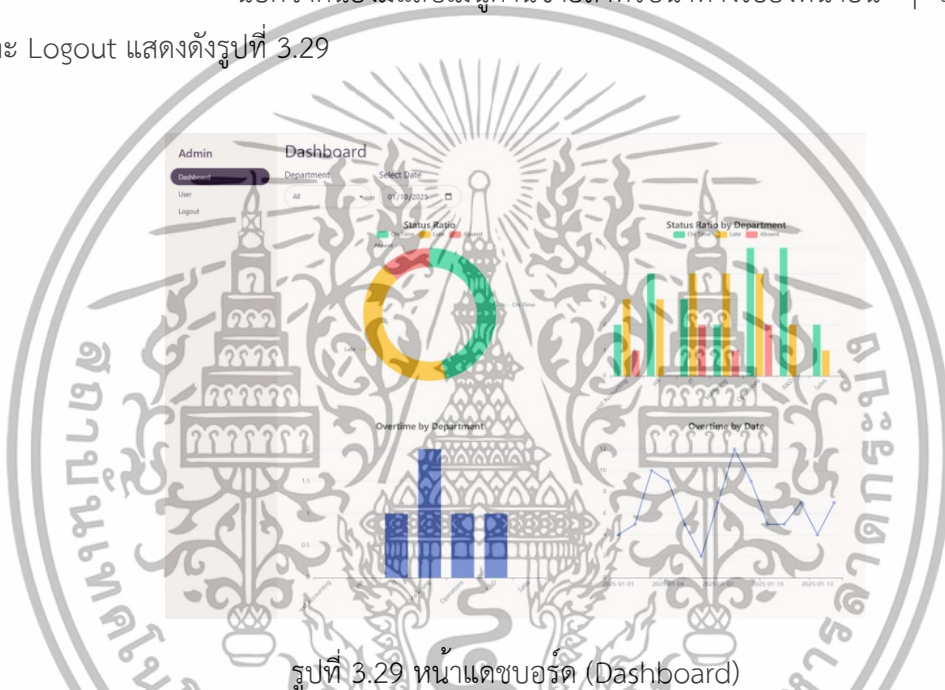
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. กราฟแท่ง Overtime by Department

เป็นกราฟที่แสดงจำนวนชั่วโมงทำงานล่วงเวลาของแต่ละแผนก โดยหากมีการเลือกแผนกที่สนใจจะเปลี่ยนเป็นกราฟ Overtime by Employees ซึ่งแสดงจำนวนชั่วโมงทำงานล่วงเวลาของแต่ละบุคคลในแผนกนั้นๆ

### 4. กราฟเส้น Overtime by Date

เป็นกราฟที่แสดงแนวโน้มการทำงานล่วงเวลาในแต่ละวัน นอกจากนี้ยังมีแถบเมนูด้านซ้ายสำหรับนำทางไปยังหน้าอื่น ๆ เช่น User และ Logout แสดงดังรูปที่ 3.29



รูปที่ 3.29 หน้าแดชบอร์ด (Dashboard)

### 3.1.8.3 หน้าผู้ใช้งาน (User)

หน้านี้เป็นหน้ารายชื่อพนักงาน สำหรับผู้ดูแลระบบ โดยมีการจัดแสดงข้อมูลพนักงานในรูปแบบตารางพร้อมตัวเลือกสำหรับกรองข้อมูลตามแผนกและ ค้นหาตามชื่อพนักงานที่มุมขวาบนตารางมีคอลัมน์แสดงชื่อพนักงาน (Name), แผนก (Department), สถานะการเข้างาน (Status), เวลาล่าสุดที่มีการเช็คอิน (Latest Check) และปุ่ม "Detail" สำหรับดูรายละเอียดเพิ่มเติม สถานะการเข้างานถูกแสดงเป็นแท็กสี เช่น "On Time" (สีเขียว), "Late" (สีเหลือง), และ "Absent" (สีแดง) เพื่อให้ผู้ใช้สามารถมองเห็นและวิเคราะห์ข้อมูลได้ง่ายขึ้น นอกจากนี้ยังมีการแสดงไอคอนแจ้งเตือนเพื่อบอกว่าพนักงานคนนั้นมีการขาดงานเกิน 4 ครั้งแล้ว แสดงดังรูป 3.30

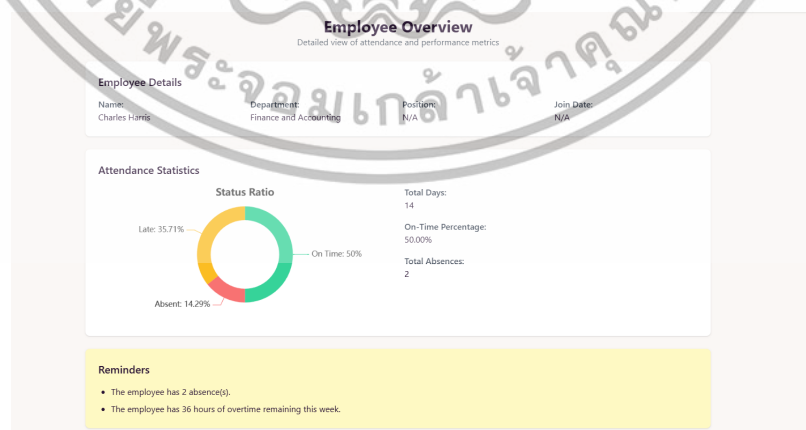
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name	Department	Status	Latest Check	
Charles Harris	Finance and Accounting	Late	2025-01-14, 10:05	Detail
Daniel Parker	Finance and Accounting	On Time	2025-01-14, 08:55	Detail
Nancy Edwards	Finance and Accounting	On Time	2025-01-14, 08:27	Detail
David Miller	Finance and Accounting	On Time	2025-01-14, 08:56	Detail
Christopher Young	Finance and Accounting	On Time	2025-01-14, 08:15	Detail
Deborah Rivera	Finance and Accounting	Late	2025-01-14, 09:52	Detail
Robert Brown	HR	Absent	2025-01-14, N/A	Detail

รูปที่ 3.30 หน้าผู้ใช้งาน (User)

### 3.1.8.4 หน้ารายละเอียดของพนักงาน (Personal)

ซึ่งเป็นหน้าสำหรับแสดงรายละเอียดของพนักงานรายบุคคลในด้านการเข้าทำงานและประสิทธิภาพการทำงาน โครงสร้างของหน้าแบ่งออกเป็นสามส่วนหลัก ได้แก่ Employee Details ที่แสดงข้อมูลพื้นฐานของพนักงาน เช่น ชื่อ แผนก ตำแหน่ง และวันที่เข้าร่วมงาน Attendance Statistics ซึ่งแสดงสถิติการเข้า-ออกงานผ่านกราฟโดนัทที่แสดงสัดส่วนของสถานะการเข้างาน (On Time, Late, Absent) พร้อมตัวเลขสรุป เช่น จำนวนวันทำงานทั้งหมด เปอร์เซ็นต์การมาตรงเวลา และจำนวนครั้งที่ขาดงาน Reminders ที่เป็นกล่องข้อความไฮไลต์สีเหลืองเพื่อแจ้งเตือนข้อมูลสำคัญ เช่น จำนวนครั้งที่พนักงานขาดงาน และชั่วโมงทำงานล่วงเวลาที่ยังเหลือในสัปดาห์ แสดงดังรูปที่ 3.31



รูปที่ 3.31 หน้ารายละเอียดของพนักงาน (Personal)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.9 การออกเว็บแอปพลิเคชันในส่วนของหลังบ้าน (Backend)

#### 3.1.9.1 การแยกประเภทของผู้ใช้งาน

ฟังก์ชันการเข้าสู่ระบบใช้สำหรับเข้าสู่ระบบโดยรีเซ็ทข้อความผิดพลาดก่อน จากนั้นพยายามล็อกอินผ่าน `accountStore.signInAdmin()` หากเป็น Admin จะนำทางไปที่แดชบอร์ด แต่ถ้าเป็นผู้ใช้ทั่วไป จะดึง `userId` จากอีเมลและนำทางไปที่หน้าข้อมูลของพนักงานคนนั้น หากเกิดข้อผิดพลาด จะบันทึกข้อความผิดพลาดและแสดง pop-up แจ้งเตือน เช่น E-mail ผิดพลาด และ Password ผิดพลาด แสดงดังรูปที่ 3.32

```
const login = async () => {
  errorMessage.value = ''
  try {
    await accountStore.signInAdmin(email.value, password.value)
    if (accountStore.isAdmin) {
      router.push({ name: 'dashboard' })
    } else {
      const userEmail = accountStore.user.email
      const userId = userEmail.split('@')[0]
      router.push(`/admin/users/${userId}`)
    }
  } catch (error) {
    errorMessage.value = error.message
    eventStore.popupMessage('error', error.message)
  }
}
```

รูปที่ 3.32 ฟังก์ชันการเข้าสู่ระบบ

#### 3.1.9.2 การดึงข้อมูลจาก Firebase

ระบบจะใช้ Firebase SDK เพื่อตั้งค่าและเชื่อมต่อแอปกับ Firebase โดยกำหนด `firebaseConfig` เพื่อระบุโปรเจกต์ จากนั้นใช้ `initializeApp` เพื่อเริ่มต้น Firebase และเรียกใช้บริการต่างๆ ได้แก่ Firestore (`getFirestore`), Authentication (`getAuth`), และ Realtime Database (`getDatabase`) ซึ่งช่วยให้แอปสามารถจัดเก็บข้อมูล รับรองตัวตน และซิงค์ข้อมูลแบบเรียลไทม์กับ Firebase ได้ และนำไปใช้ในการดึงข้อมูลไปแสดงที่หน้าแดชบอร์ด (Dashboard), หน้าผู้ใช้งาน (User) และหน้ารายละเอียดของพนักงาน (Personal) แสดงดังรูปที่ 3.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import { initializeApp } from 'firebase/app'
import { getFirestore, connectFirestoreEmulator } from 'firebase/firestore'
import { getAuth, connectAuthEmulator } from 'firebase/auth'
import { getDatabase, connectDatabaseEmulator } from 'firebase/database'

const firebaseConfig = {
  apiKey: "AIzaSyBDCbfe8u-D8dsIfIvggi5bKxshD9bDVI0",
  authDomain: "backend-1e9af.firebaseio.com",
  databaseURL: "https://backend-1e9af-default-rtdb.asia-southeast1.firebaseio.com",
  projectId: "backend-1e9af",
  storageBucket: "backend-1e9af.firebaseio.com",
  messagingSenderId: "613614907981",
  appId: "1:613614907981:web:c18ea241f9abecd6efa204"
};

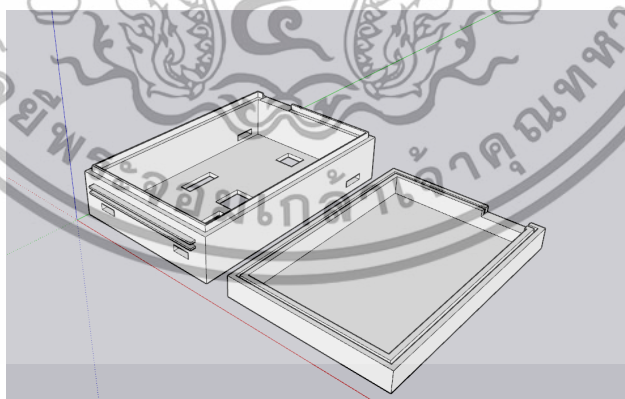
const app = initializeApp(firebaseConfig);
const db = getFirestore(app);
const auth = getAuth(app);
const realtimeDB = getDatabase(app);

```

รูปที่ 3.33 ฟังก์ชันการดึงข้อมูลทั้งหมดบน Firebase

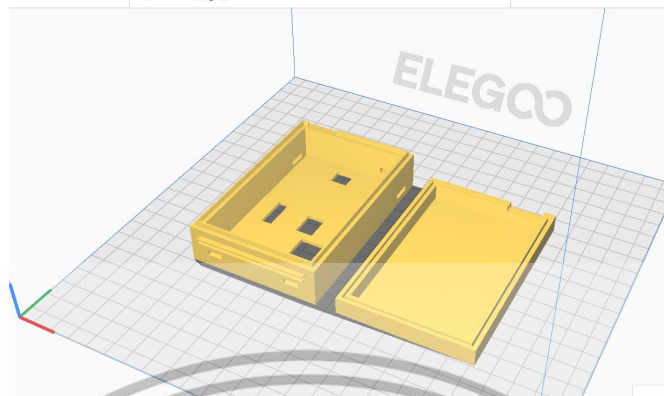
### 3.1.10 การออกแบบกล่อง

โดยจะทำการออกแบบโมเดล 3D ในโปรแกรม SketchUp หลังจากนั้นส่งออกเป็นไฟล์นามสกุล STL จากนั้นนำเข้า UltiMaker Cura นำไป Slicing สำหรับการพิมพ์ 3 มิติ โดยทำการแปลงไฟล์โมเดล 3 มิติ ให้เป็นโค้ด G-code ที่เครื่องพิมพ์ 3 มิติสามารถอ่านและพิมพ์ได้ แสดงดังรูปที่ 3.34 และ 3.35



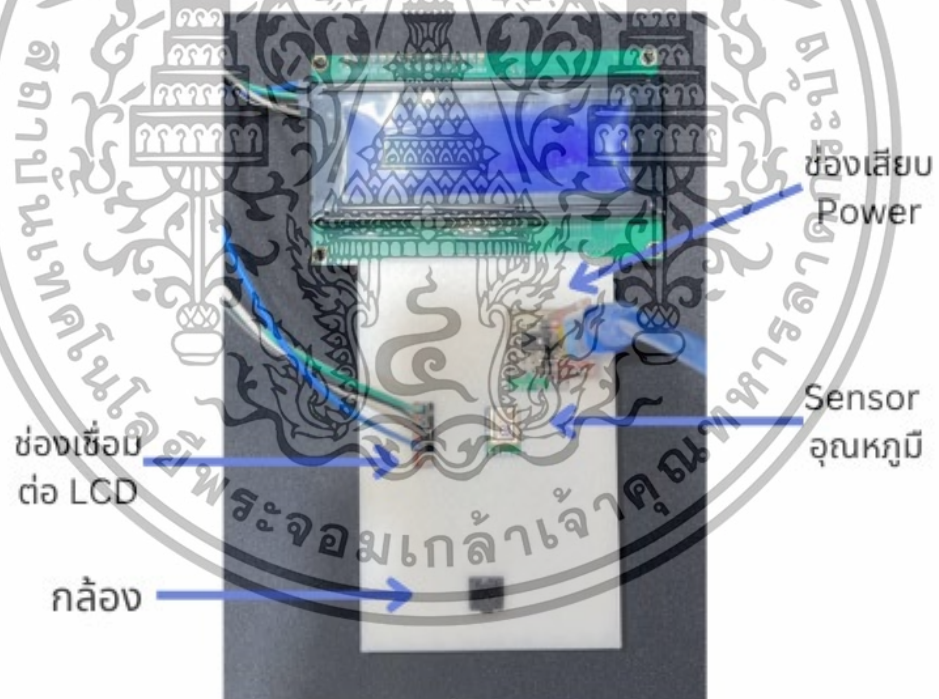
รูปที่ 3.34 การออกแบบกล่องใน Sketchup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.35 โมเดลกล่องใน UltiMaker Cura

และเมื่อนำมาประกอบจะได้ดังรูปที่ 3.36



รูปที่ 3.36 อุปกรณ์หลังประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 เครื่องมือที่ใช้ในการทดลอง

ในปฏิญานิพนธ์นี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลองดังนี้

### 3.2.1 Arduino IDE

เป็นเครื่องมือหลักที่ใช้สำหรับเขียนและอัปโหลดโค้ดไปยัง ESP32-CAM ซึ่งทำหน้าที่ควบคุมโมดูลกล้องและเซ็นเซอร์อินฟราเรดสำหรับวัดอุณหภูมิ นอกจากนี้ ยังใช้ Serial Monitor ใน Arduino IDE เพื่อตรวจสอบค่าต่าง ๆ ที่ได้รับจากเซ็นเซอร์แบบเรียลไทม์

### 3.2.2 Visual Code Studio

เป็นเครื่องมือสำหรับพัฒนาโค้ด Python ที่ใช้ในการประมวลผลภาพใบหน้าและจัดการข้อมูลที่ได้รับจาก ESP32-CAM โดยโค้ด Python จะช่วยวิเคราะห์ข้อมูลที่ได้รับ และส่งไปยังฐานข้อมูล Firebase เพื่อบันทึกเวลาการเข้า-ออกงานของพนักงาน

### 3.2.3 Firebase

ทำหน้าที่เป็นฐานข้อมูลกลางสำหรับจัดเก็บข้อมูลเกี่ยวกับใบหน้าและการลงเวลา ระบบใช้ Firebase Realtime Database ในการเก็บข้อมูลเวลาเข้า-ออก และใช้ Firebase Storage ในการจัดเก็บภาพใบหน้าที่ถ่ายจาก ESP32-CAM เพื่อให้สามารถเรียกดูย้อนหลังได้

### 3.2.4 Fritzing

เป็นซอฟต์แวร์ที่ช่วยออกแบบวงจรการเชื่อมต่อของอุปกรณ์ฮาร์ดแวร์ในระบบ เช่น การเชื่อมต่อ ESP32-CAM, โมดูลวัดอุณหภูมิ และเซ็นเซอร์ต่าง ๆ การใช้ Fritzing ช่วยให้สามารถวางแผนและทดสอบวงจรได้ง่ายขึ้นก่อนนำไปใช้งานจริง

### 3.2.5 MQTT Protocol

เป็นโปรโตคอลที่ใช้สำหรับส่งข้อมูลระหว่าง ESP32-CAM และเซิร์ฟเวอร์ ข้อมูลที่ส่งผ่าน MQTT รวมถึงอุณหภูมิใบหน้าและข้อมูลการลงเวลา ซึ่งช่วยให้สามารถแจ้งเตือนและอัปเดตข้อมูลแบบเรียลไทม์ผ่านเครือข่ายไร้สายได้

## 3.3 การจัดเก็บผลการทดลอง

โดยผลการลงเวลาจะจัดเก็บใน Firebase Realtime Database โดยจะมีการจัดเก็บข้อมูลเกี่ยวกับพนักงานแต่ละคน โดยจะมี รหัสพนักงาน(id), ชื่อพนักงาน(name), ตำแหน่งงาน(position) และแผนกที่สังกัด(department) แสดงดังรูปที่ 3.37 และในส่วนของการลงเวลาเข้าออกพนักงานจะเก็บไว้ใน attendance จะมีข้อมูลของ วันที่(date), สถานะ(status), เวลาการเข้างาน(time\_in) และเวลาการออกงาน(time\_out) แสดงดังรูปที่ 3.38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<https://backend-1e9af-default-rtdb.asia-southeast1.firebaseio.com>

<https://backend-1e9af-default-rtdb.asia-southeast1.firebaseio.com>

```

employees
  0
    attendance
      department: "Finance and Accounting"
      id: "1"
      name: "Charles Harris"
      position: "Accountant"
  
```

รูปที่ 3.37 โครงสร้างของ employees

<https://backend-1e9af-default-rtdb.asia-southeast1.firebaseio.com>

<https://backend-1e9af-default-rtdb.asia-southeast1.firebaseio.com>

```

employees
  0
    attendance
      0
        date: "2025-01-01"
        status: "On Time"
        time_in: "08:51"
        time_out: "15:56"
  
```

รูปที่ 3.38 โครงสร้างของ attendance

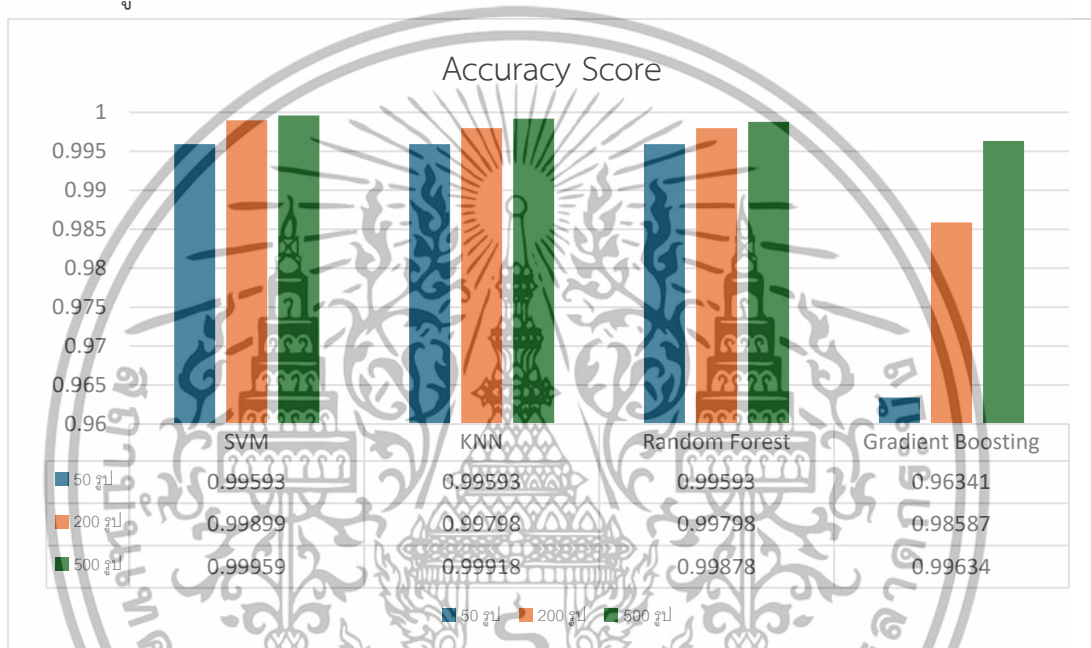
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

#### 4.1 การเลือกใช้ Model และจำนวนขนาดข้อมูล

หลังจากได้ทดสอบโมเดลทั้ง 4 ด้วยภาพหน้าคน 50 รูป 200 รูป และ 500 รูปได้ผลลัพธ์ดังรูปที่ 4.1



รูปที่ 4.1 ผล Accuracy Score

จากรูปกราฟแสดงให้เห็นว่าเมื่อเพิ่มจำนวนรูปภาพเพิ่มขึ้น ส่งผลให้ค่า Accuracy Score มีค่าเพิ่มตามไปด้วยโดยที่เพิ่มจาก 50 รูป ไปเป็น 200 รูป เห็นผลชัดเจนกว่าเพิ่มจาก 200 รูปไปเป็น 500 รูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ตารางแสดงผลที่จำนวนรูป 200 รูป

Model	Accuracy Score	Cross-Validation Mean Accuracy	precision	recall	f1-score	support
SVM	0.99959	0.9962	1.00	1.00	1.00	2440
KNN	0.99919	0.9975	1.00	1.00	1.00	2440
Random Forest	0.99877	0.9955	1.00	1.00	1.00	2440
Gradient Boosting	0.96341	0.9808	0.99	0.99	0.99	2440

จากตาราง SVM, KNN, และ Random Forest มี Accuracy Score สูงมากและ Precision, Recall, F1-score เท่ากับ 1.00 ซึ่งหมายความว่าโมเดลสามารถจำแนกข้อมูลได้สมบูรณ์แบบหรือเกือบสมบูรณ์แบบ อย่างไรก็ตาม Gradient Boosting มี Accuracy Score ต่ำกว่าตัวอื่นเล็กน้อย (0.96341 หรือ 96.34 %) และ Cross-Validation Mean Accuracy ต่ำที่สุดแสดงว่าอาจมีการ Generalization ที่แย่กว่าตัวอื่นตั้งนั้นจากผลลัพธ์ SVM เป็นตัวเลือกที่ดีที่สุด

## 4.2 การทดสอบการตรวจจับใบหน้า

ในการทดสอบระบบการตรวจจับใบหน้าที่มีขั้นตอนทั้งหมด ดังนี้

### 4.2.1 จับภาพจากกล้อง ESP32-CAM และตรวจจับใบหน้า

ระบบรับภาพจากกล้อง ESP32-CAM เพื่อใช้ในการตรวจจับใบหน้าและใช้เทคนิค Haar Cascade เพื่อระบุตำแหน่งของใบหน้าในภาพที่ได้รับ

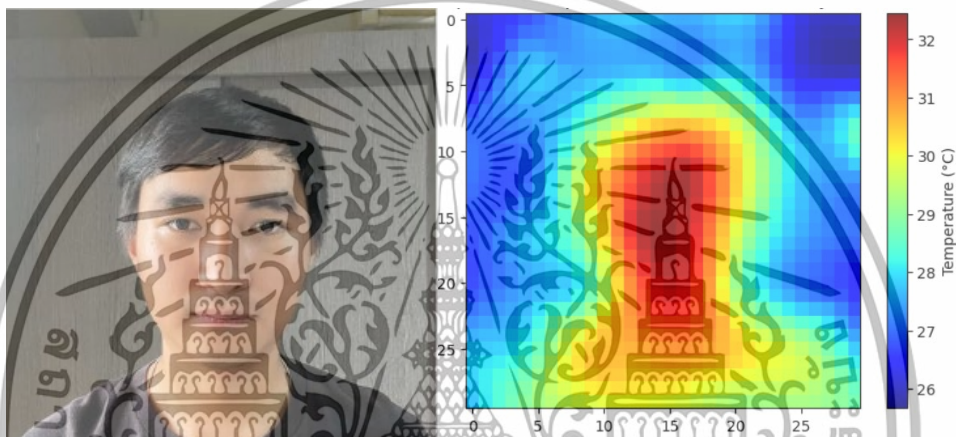
### 4.2.2 แปลงข้อมูลใบหน้าเป็นเวกเตอร์

แปลงข้อมูลใบหน้าเป็นเวกเตอร์โดยใช้โมเดล FaceNet เพื่อแปลงใบหน้าที่ตรวจจับได้ให้เป็นเวกเตอร์สำหรับการเปรียบเทียบ

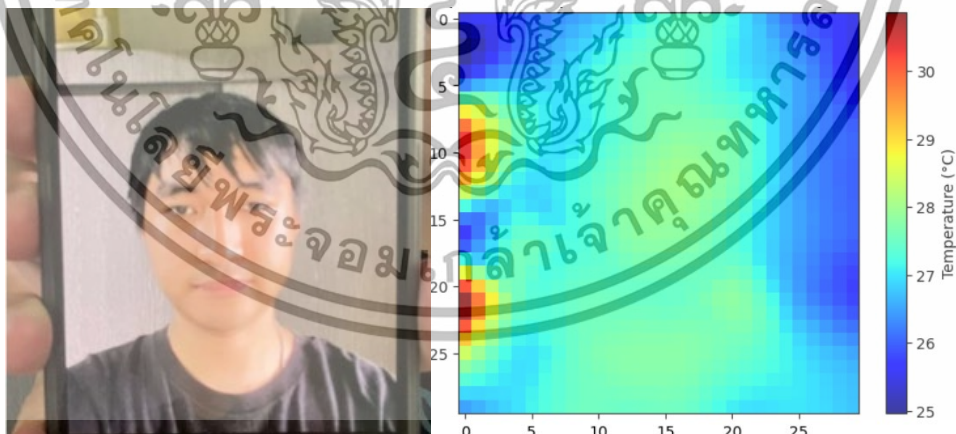
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.2.3 เปรียบเทียบกับฐานข้อมูลและตรวจสอบอุณหภูมิใบหน้า

นำเวกเตอร์ใบหน้ามาเปรียบเทียบกับโมเดลใบหน้าของแต่ละคนที่เทรนไว้เพื่อระบุว่าบุคคลนั้นเป็นใครและใช้เซ็นเซอร์อินฟราเรดตรวจวัดอุณหภูมิบริเวณใบหน้า และเปรียบเทียบกับอุณหภูมิพื้นหลังถ้าอุณหภูมิของใบหน้าสูงกว่าพื้นหลังแสดงว่าบุคคลนั้นเป็นคนจริง แสดงดังรูปที่ 4.2 แต่ถ้าใช้ภาพถ่ายอุณหภูมิจะไม่แตกต่างกันมากทำให้สามารถระบุได้ว่าเป็นภาพถ่าย แสดงดังรูปที่ 4.3



รูปที่ 4.2 การเปรียบเทียบอุณหภูมิพื้นหลังกับใบหน้าของคนจริง

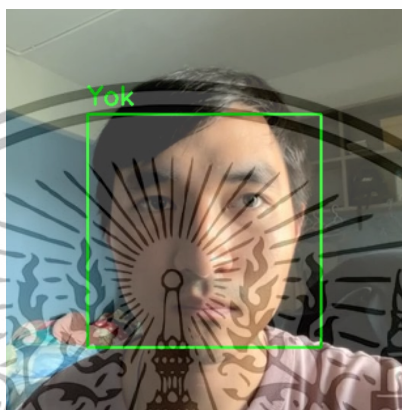


รูปที่ 4.3 การเปรียบเทียบอุณหภูมิพื้นหลังกับใบหน้าของรูปถ่าย

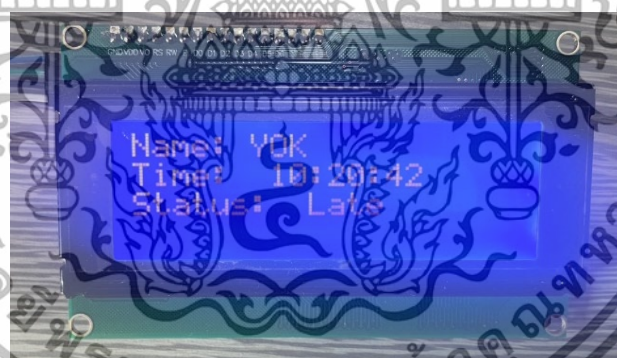
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.4 อัปเดตข้อมูลเวลาเข้า-ออกงานและส่งข้อมูลผ่าน MQTT

จากรูปที่ 4.4 เมื่อระบบยืนยันตัวตนบุคคลได้แล้วระบบจะบันทึกข้อมูลเวลาเข้า-ออกงานลงใน Firebase และระบบส่งข้อความแจ้งเตือนเกี่ยวกับผลการตรวจจับไปยัง MQTT เพื่อแสดงสถานะของบุคคลนั้นบนจอ LCD ได้แบบเรียลไทม์ แสดงดังรูปที่ 4.5



รูปที่ 4.4 การตรวจจับใบหน้าและยืนยันตัวตน



รูปที่ 4.5 จอแสดงสถานะ

### 4.3 การทดสอบการส่งข้อมูลอุณหภูมิ

ในการทดสอบระบบการส่งข้อมูลอุณหภูมิจะมีขั้นตอนทั้งหมด ดังนี้

#### 4.3.1 วัดอุณหภูมิด้วยเซ็นเซอร์อินฟราเรด

วัดอุณหภูมิด้วยเซ็นเซอร์อินฟราเรดเซ็นเซอร์แบบ 8x8 infrared sensor array ตรวจจับอุณหภูมิจาก 64 จุดและข้อมูลอุณหภูมิถูกจัดเก็บเป็นอาร์เรย์ทุก ๆ 5 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.2 แบ่งข้อมูลเป็นชิ้น (Chunk) เพื่อลดขนาดข้อมูล

แบ่งข้อมูลเป็นชิ้น (Chunk) เพื่อลดขนาดข้อมูล แต่ละชิ้นมีค่าอุณหภูมิ 8 ค่า เพื่อลดขนาดและปรับปรุงความเร็วการส่งข้อมูล

### 4.3.3 ส่งข้อมูลผ่าน MQTT

ข้อมูลถูกแปลงเป็นข้อความและส่งไปยังระบบผ่านโปรโตคอล MQTT

### 4.3.4 ตรวจสอบความถูกต้องของการส่งข้อมูล

ตรวจสอบความถูกต้องของการส่งข้อมูลหากการส่งข้อมูลสำเร็จ ระบบจะแสดงข้อความ "Chunk published successfully" แสดงดังรูปที่ 4.6 หากล้มเหลว ระบบจะแสดงสถานะข้อผิดพลาด



```

COMB
PUBLISH CHUNK: 30.25,60.46,30.00,29.75,29.00,29.00,28.50,28.25
Chunk published successfully
PUBLISH CHUNK: 29.00,29.00,29.25,29.75,28.25,28.25,28.25,28.75
Chunk published successfully
PUBLISH CHUNK: 27.75,28.00,27.75,28.00,27.75,27.50,27.50,27.75
Chunk published successfully
PUBLISH CHUNK: 27.00,27.25,27.00,27.75,27.25,27.25,27.00,27.00
Chunk published successfully
PUBLISH CHUNK: 26.75,26.50,26.75,27.00,27.00,27.00,26.75,27.50
Chunk published successfully
PUBLISH CHUNK: 26.25,27.00,27.00,27.75,27.50,27.00,26.50,27.75
Chunk published successfully
PUBLISH CHUNK: 27.25,26.25,25.25,26.25,26.00,26.75,26.75,27.15
Chunk published successfully
PUBLISH CHUNK: 26.50,25.50,25.25,25.50,25.75,26.75,28.25,30.25
Chunk published successfully
CAPTURE OK 26x240x5016
PUBLISH CHUNK: 25.75,30.75,30.25,30.00,25.00,31.50,28.25,25.25
Autoscroll Show timestamp
  
```

รูปที่ 4.6 การส่งข้อมูลอุณหภูมิเป็นชิ้นๆ (Chunk)

## 4.4 การทดสอบแผนที่ความร้อน (Temperature heat map)

จากการทดลองการทำแผนที่ความร้อน สำหรับการตรวจจับอุณหภูมิบนใบหน้า เป็นกระบวนการที่ใช้เซ็นเซอร์อินฟราเรดวัดอุณหภูมิและแสดงผลออกมาในรูปแบบภาพเพื่อบ่งบอกถึงความแตกต่างของอุณหภูมิในแต่ละจุดบนใบหน้า ซึ่งช่วยให้สามารถระบุบริเวณที่มีอุณหภูมิสูงหรือต่ำได้อย่างชัดเจน

ในกรณีนี้ เราจะใช้เซ็นเซอร์แบบ 8x8 infrared sensor array ที่มีจำนวนเซ็นเซอร์รวมทั้งหมด 64 จุด ทำการวัดอุณหภูมิของใบหน้าแต่ละจุด โดยเซ็นเซอร์แต่ละตัวจะบันทึกค่าของอุณหภูมิและส่งข้อมูลเป็นพิกเซลแต่ละจุด ทำให้ได้ข้อมูลความร้อนเป็นชุดคล้ายกับภาพพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเหล่านี้จะถูกแปลงให้อยู่ในรูปแบบของ heat map โดยสีของแต่ละพิกเซลจะบ่งบอกถึงค่าของอุณหภูมิ เช่น สีแดงหรือสีร้อนแทนค่าอุณหภูมิสูง และสีฟ้าหรือสีเย็นแทนค่าอุณหภูมิต่ำ

เพื่อสร้าง heat map บนใบหน้า เราต้องกำหนดขั้นตอนการแปลงข้อมูลอุณหภูมิจากเซ็นเซอร์ให้อยู่ในรูปแบบที่สามารถนำมาแสดงผลได้ ดังนี้

#### 4.4.1 การรับข้อมูลอุณหภูมิ

เซ็นเซอร์อินฟราเรด 8x8 จะวัดอุณหภูมิเป็นค่าแบบ real-time ที่ตำแหน่งต่างๆ บนใบหน้า และส่งข้อมูลนี้ไปยังระบบเพื่อประมวลผล

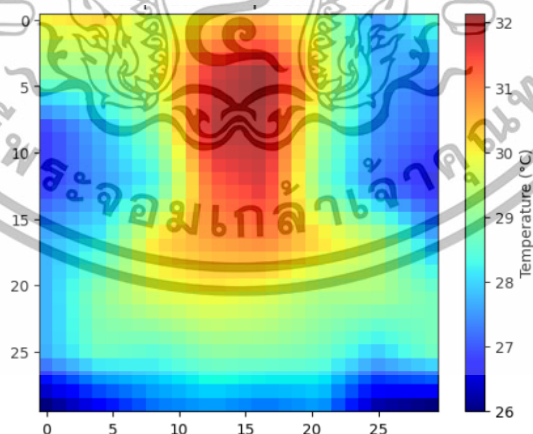
#### 4.4.2 การแปลงข้อมูลเป็นพิกเซล

ข้อมูลอุณหภูมิที่ได้จากแต่ละเซ็นเซอร์จะถูกแปลงเป็นสีที่แทนระดับความร้อนในแต่ละจุด

#### 4.4.3 การแสดงผลในรูปแบบแผนที่ความร้อน

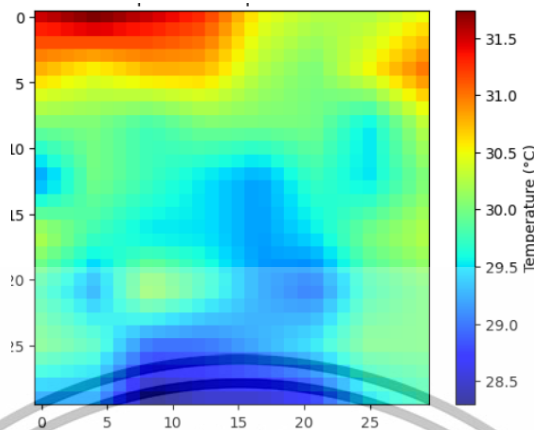
ข้อมูลพิกเซลที่แปลงแล้วจะถูกจัดวางเป็นแผนที่ 8x8 พิกเซล แสดงเป็นภาพของใบหน้าที่มีการแบ่งแยกอุณหภูมิในจุดต่างๆ ทำให้มองเห็นการกระจายของความร้อนบนใบหน้าอย่างชัดเจน

โดยผู้จัดทำได้ทำการทดสอบการหาตำแหน่งของใบหน้าโดยใช้แผนที่ความร้อนในการดูความเปลี่ยนแปลงของอุณหภูมิเมื่อเทียบกับการมีใบหน้าและเมื่อใบหน้าถูกบังไว้ แสดงดังรูปที่ 4.7 และ 4.8



รูปที่ 4.7 การแสดงแผนที่ความร้อนโดยที่มีใบหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



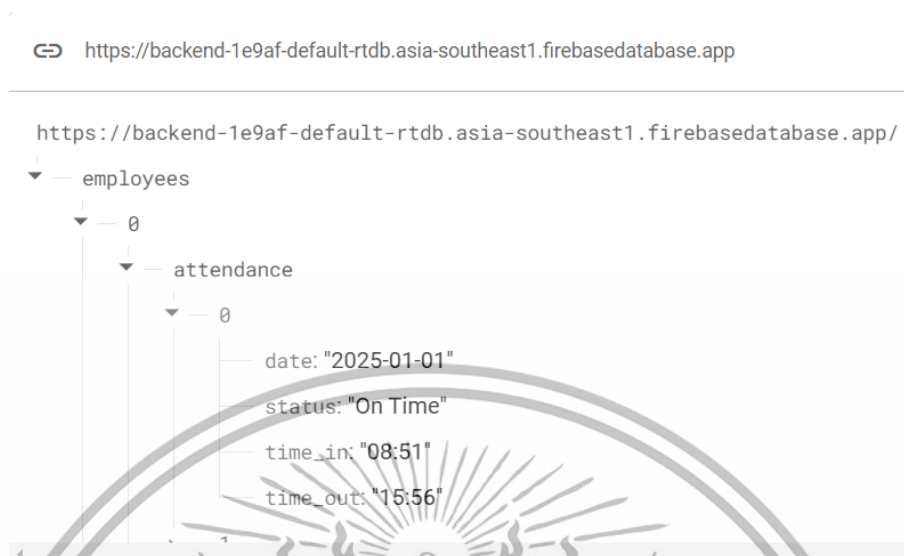
รูปที่ 4.8 การแสดงแผนที่ความร้อนโดยที่ใบหน้าโดนบังไว้

จากรูปที่ 4.7 และ 4.8 เมื่อใบหน้าของคนจริงปรากฏขึ้น อุณหภูมิจะสูงขึ้นเนื่องจากความร้อนจากร่างกาย แต่เมื่อใบหน้าโดนวัตถุมาบัง เช่น กระจาดหรือรูปถ่าย อุณหภูมิจะลดลงอย่างชัดเจน การเปลี่ยนแปลงนี้สามารถใช้แยกแยะระหว่างใบหน้าของคนจริงกับการใช้รูปถ่ายมาปลอมแปลงได้ จึงช่วยเพิ่มความแม่นยำในการยืนยันตัวตน

#### 4.5 การทดสอบระบบลงเวลา

จากการทดสอบระบบบันทึกเวลาทำงาน พบว่าระบบสามารถแยกการบันทึกเวลาเข้าและเวลาเลิกงานได้อย่างถูกต้อง สำหรับการเลิกงาน หากพนักงานบันทึกหลัง 17.00 น. ระบบจะตรวจสอบวันที่และชั่วโมงของการบันทึกล่าสุดเพื่อป้องกันการบันทึกซ้ำ หากพบว่าชั่วโมงของบันทึกล่าสุดน้อยกว่าชั่วโมงปัจจุบัน ระบบจะบันทึกสถานะเป็น "Outtime" พร้อมเวลาที่ออกงาน ในขณะที่การบันทึกเวลาเข้างาน ระบบจะตรวจสอบว่าหากบันทึกก่อน 9.00 น. จะตั้งสถานะเป็น "On Time" หากเกินจากนั้นจะตั้งสถานะเป็น "Late" และบันทึกจำนวนครั้งที่สายลงใน Firebase โดยระบบยังตรวจสอบให้แน่ใจว่าการบันทึกเป็นของวันใหม่เพื่อป้องกันการบันทึกซ้ำ แสดงดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ตัวอย่างข้อมูลบน Firebase

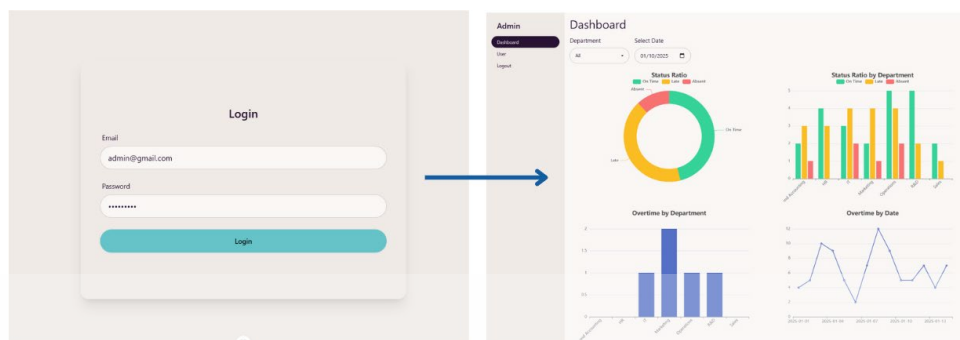
## 4.6 การทดสอบเว็บแอปพลิเคชัน

จากการทดสอบเว็บแอปพลิเคชัน พบว่าเว็บแอปพลิเคชันสามารถเชื่อมต่อกับฐานข้อมูลบน Firebase ได้โดยจะแบ่งการทำงานเป็น ดังนี้

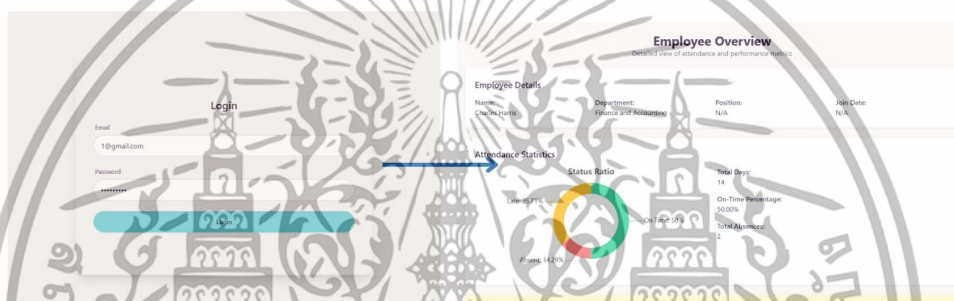
### 4.6.1 การเข้าสู่ระบบ

ในการเข้าสู่ระบบสามารถตรวจสอบบัญชีที่ลงทะเบียนไว้บน Firebase Authentication ได้เพื่อตรวจสอบสิทธิ์ในการเข้าถึงเว็บไซต์และยังสามารถแยกประเภทของผู้ใช้งานได้ โดยเมื่อเข้าสู่ระบบจะมีการตรวจสอบประเภทของผู้ใช้งานว่าถ้าหากเป็น E-mail ของผู้ดูแลระบบ (Admin) ก็จะเข้าไปยัง Backend Dashboard เป็นส่วนหนึ่งของระบบที่ใช้สำหรับดูประวัติการลงเวลาการเข้า-ออกงานของทุกคน ซึ่งสามารถเข้าถึงได้เฉพาะผู้ดูแลระบบ เท่านั้น แสดงดังรูปที่ 4.10 แต่ถ้าหากเข้าสู่ระบบด้วย E-mail ของพนักงานก็จะเข้าไปยังหน้าข้อมูลของผู้ใช้งานของแต่ละคนเท่านั้น แสดงดังรูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 เข้าสู่ระบบด้วย E-mail ผู้ดูแลระบบ (Admin)

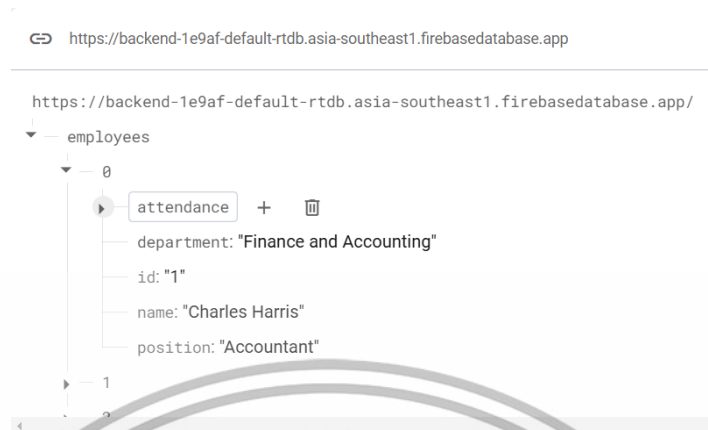


รูปที่ 4.11 เข้าสู่ระบบด้วย E-mail พนักงาน

#### 4.6.2 การแสดงผลหน้าแดชบอร์ดและหน้าผู้ใช้งาน

ในหน้าแดชบอร์ดสามารถดึงข้อมูลจาก Firebase Realtime Database ของพนักงานแต่ละคนมาแสดงโดยจะทำการนำข้อมูล ID พนักงาน, ชื่อพนักงาน, แผนก, ตำแหน่งงาน, วันที่เข้างานล่าสุด, สถานการณ์ลงเวลา, เวลาการเข้างาน และเวลาการออกงาน แสดงดังรูปที่ 4.12 และ 4.13 จากนั้นนำมาแสดงเป็นกราฟแบบต่าง ๆ เพื่อให้สามารถเข้าใจได้มากขึ้น แสดงดังรูป 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 โครงสร้างของ employees



รูปที่ 4.13 โครงสร้างของ attendance



รูปที่ 4.14 หน้าแดชบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.7 การนำข้อมูลมาวิเคราะห์ทางสถิติ

จากรูปที่ 4.14 เป็นแดชบอร์ด ที่ใช้สำหรับวิเคราะห์ข้อมูลการเข้า-ออกงานของพนักงานในรูปแบบกราฟและแผนภูมิ ซึ่งช่วยให้ผู้บริหารสามารถติดตามสถานะของพนักงานแต่ละแผนกและช่วงเวลาได้อย่างมีประสิทธิภาพ โดยการวิเคราะห์ข้อมูลทางสถิติมีรายละเอียดดังนี้

การแสดงผลข้อมูลในแดชบอร์ดนี้มีจุดประสงค์เพื่อให้ผู้ใช้งานสามารถมองเห็นภาพรวมของสถานะการทำงานของพนักงานในแต่ละวันและแต่ละแผนกได้อย่างชัดเจนและเข้าใจง่าย โดยใช้การนำเสนอผ่านรูปแบบกราฟและแผนภูมิที่หลากหลาย เช่น แผนภูมิวงกลม (Donut Chart) ที่แสดงสัดส่วนพนักงานที่มาทำงานตรงเวลา มาสาย และขาดงาน ซึ่งช่วยให้มองเห็นภาพรวมขององค์กรในวันนั้นได้ทันที นอกจากนี้ยังมีกราฟแท่งที่แสดงสถานะของพนักงานแยกตามแผนก ทำให้สามารถเปรียบเทียบได้ว่าแผนกใดมีปัญหาเรื่องการมาทำงานไม่ตรงเวลาหรือขาดงานบ่อยที่สุด

ในส่วนของการวิเคราะห์การทำงานล่วงเวลา มีกราฟแท่งที่แสดงจำนวนการทำโอทีที่แยกตามแผนก ช่วยให้ผู้บริหารเข้าใจถึงภาระงานในแต่ละฝ่าย และสามารถปรับกระจายงานให้เหมาะสมได้ อีกทั้งยังมีกราฟเส้นแสดงจำนวนชั่วโมงโอทีในแต่ละวัน ซึ่งช่วยให้สามารถติดตามแนวโน้มและความเปลี่ยนแปลงของการทำงานล่วงเวลาได้อย่างมีประสิทธิภาพ โดยภาพรวมการแสดงผลทั้งหมดนี้ช่วยให้การตัดสินใจทางบริหารเป็นไปอย่างมีข้อมูลรองรับและทันต่อสถานการณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผล

ปริญญานิพนธ์นี้ประสบความสำเร็จในการพัฒนาระบบตรวจจับใบหน้าและบันทึกเวลาเข้า-ออกงานโดยใช้กล้อง ESP32-CAM และการประมวลผลอุณหภูมิจากเซ็นเซอร์อินฟราเรด ระบบที่ออกแบบประกอบด้วยการจัดการข้อมูล การทดสอบการตรวจจับใบหน้าและการยืนยันบุคคลจริงด้วยข้อมูลอุณหภูมิ รวมถึงระบบบันทึกเวลาเข้า-ออกงานและการแจ้งเตือนผ่าน MQTT ในส่วนของการตรวจจับใบหน้า ระบบสามารถจับภาพและวิเคราะห์คุณลักษณะใบหน้าที่เทียบกับฐานข้อมูลใน Firebase เพื่อยืนยันตัวบุคคล เมื่อระบบพบว่าบุคคลเป็นคนจริง โดยพิจารณาจากอุณหภูมิบริเวณใบหน้าที่สูงกว่าพื้นหลัง ระบบจะส่งข้อความยืนยันผ่าน MQTT การส่งข้อมูลอุณหภูมิใช้เซ็นเซอร์อินฟราเรดขนาด 8x8 ที่จัดเก็บอุณหภูมิทุก 5 วินาที และส่งผ่าน MQTT ในรูปแบบ Chunk เพื่อลดขนาดข้อมูลและเพิ่มประสิทธิภาพการส่งต่อไปยังโปรแกรม Python เพื่อขยายขนาดของเซ็นเซอร์อินฟราเรดจาก 8x8 เป็น 30x30 เพื่อเพิ่มความละเอียดในการตรวจจับอุณหภูมิได้มากยิ่งขึ้นและในส่วนของระบบบันทึกเวลา ระบบสามารถบันทึกเวลาเข้า-ออกงานได้อย่างถูกต้อง โดยการบันทึกเข้างานก่อน 9.00 น. จะแสดงสถานะ "InTime" หากสายจะแสดงเป็น "Late" และเก็บสถิติการมาสาย ในขณะที่บันทึกเลิกงานหลัง 16.00 น. จะแสดงเป็น "Outtime" พร้อมตรวจสอบข้อมูลให้แน่ใจว่าไม่ซ้ำกับบันทึกของวันก่อนหน้า นอกจากนี้ ได้พัฒนาเว็บแอปพลิเคชันที่สามารถดึงข้อมูลจาก Firebase มาใช้ได้ทำให้สามารถแสดงข้อมูลการบันทึกเวลาและรายละเอียดของพนักงานแต่ละบุคคลได้แบบเรียลไทม์ ซึ่งช่วยอำนวยความสะดวกให้ผู้ดูแลสามารถเข้าถึงข้อมูลได้จากทุกที่

จากการทดลองทั้งหมด ระบบมีความแม่นยำและความน่าเชื่อถือในการตรวจจับใบหน้า การประมวลผลอุณหภูมิ และการบันทึกเวลาเข้า-ออกงาน สามารถนำไปประยุกต์ใช้งานจริงในระบบลงเวลางานและเพิ่มความปลอดภัยในการเข้าถึงสถานที่ทำงาน และยังสามารถดูข้อมูลของพนักงานได้จากเว็บแอปพลิเคชันเพื่อนำข้อมูลมาวิเคราะห์ต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 ปัญหาและข้อเสนอแนะ

ปัญหาที่อาจเกิดขึ้นกับระบบนี้คือการตรวจจับใบหน้าที่อาจมีความคลาดเคลื่อนหากสภาพแวดล้อมมีแสงไม่เพียงพอหรือมุมกล้องที่ไม่เหมาะสม รวมถึงการประมวลผลอุณหภูมิที่อาจได้รับผลกระทบจากสภาพอากาศหรือสิ่งแวดล้อมที่มีอุณหภูมิใกล้เคียงกับร่างกายของมนุษย์ ทำให้การตรวจจับใบหน้าที่เป็นความจริงอาจมีความผิดพลาดได้ ข้อเสนอแนะในการปรับปรุงคือการเพิ่มแหล่งแสงที่เหมาะสมและการใช้กล้องที่มีความละเอียดสูงขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] COMPONENTS101. (2565). “ESP32-CAM Camera Module.” เข้าถึงได้จาก :  
<https://components101.com/modules/esp32-cam-camera-module>
- [2] Espressif Systems. “ESP32 Series Datasheet.” เข้าถึงได้จาก :  
[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [3] NET nanoFramework. “AMG8833/AMG8834/AMG8853/AMG8854 Infrared Array Sensor Family.” เข้าถึงได้จาก :  
<https://docs.nanoframework.net/devicesdetails/Amg88xx/README.html>
- [4] cybertice. “LCD 20x4 พร้อม I2C interface.” เข้าถึงได้จาก :  
<https://www.cybertice.com/product/433/2004-lcd-blue-screen-20x4>  
โมดูลจอแสดงผล-lcd-พร้อม-i2cinterface?gad\_source=1&gclid=EAlaQobChMliJO1p5HZiwMVsg6DAX208SCQEAAAYASAAEglwbPD\_BwE
- [5] Medium. “FaceNet และ Triplet Loss ในการจดจำใบหน้า.” เข้าถึงได้จาก :  
<https://medium.com/analytics-vidhya/introduction-to-facenet-a-unified-embedding-for-face-recognition-and-clustering-dbdac8e6f02>
- [6] Vahid Kazemi and Josephine Sullivan. “One Millisecond Face Alignment with an Ensemble of Regression Trees.” เข้าถึงได้จาก :  
<https://www.csc.kth.se/~vahidk/papers/KazemiCVPR14.pdf>
- [7] Medium. “Faces Detection Using Haar Cascade.” เข้าถึงได้จาก :  
<https://medium.com/@baselanaya/faces-detection-using-haar-cascade-3e175aef84f5>
- [8] Medium. “Explore the most advanced deep learning algorithm for face Detection.” เข้าถึงได้จาก :  
<https://medium.com/the-modern-scientist/multi-task-cascaded-convolutional-neural-network-mtcnn-a31d88f501c8>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บรรณานุกรม (ต่อ)

- [9] Medium. “SVM ตอนที่ 1: Support Vector Machine โมเดลคลาสสิกที่น่าศึกษา.”  
เข้าถึงได้จาก :  
<https://beeying.medium.com/support-vector-machine-svm-โมเดลคลาสสิกที่น่าศึกษา-78d1ce6da765>
- [10] Kongruksiam. “สรุป Machine Learning EP 4: เพื่อนบ้านใกล้ที่สุด (K-Nearest Neighbors).” เข้าถึงได้จาก : <https://kongruksiam.medium.com/สรุป-machine-learning-ep-4-เพื่อนบ้านใกล้ที่สุด-k-nearest-neighbors-787665f7c09d>
- [11] Medium. “Random Forest คืออะไร.” เข้าถึงได้จาก :  
<https://medium.com/%40pradyasin/random-forest-คืออะไร-A3-74d2a0af3d7>
- [12] Medium. “Gradient Boosting คืออะไร.” เข้าถึงได้จาก :  
<https://medium.com/@nontakul.p/gradient-boosting-คืออะไร-A3-72d19a7751fa>
- [13] dhakadojp7. “What is Bilinear Interpolation?” เข้าถึงได้จาก :  
<https://www.geeksforgeeks.org/what-is-bilinear-interpolation/>
- [14] Firebase. “Introduction to Firebase Local Emulator Suite.” เข้าถึงได้จาก :  
<https://firebase.google.com/docs/emulator-suite>
- [15] Twilio. “What is MQTT?” เข้าถึงได้จาก :  
<https://www.twilio.com/en-us/blog/what-is-mqtt>
- [16] Criclabs. “User Interface.” เข้าถึงได้จาก :  
<https://www.criclabs.co/post/what-is-user-interface-design>.
- [17] SITVISU ANANTNAKARAKUL. “ภาษา HTML” เข้าถึงได้จาก :  
<https://www.primal.co.th/th/seo/what-is-html/>.
- [18] SEO WINNER. “CSS คืออะไร มีประโยชน์อย่างไรกับเราบ้าง.” เข้าถึงได้จาก :  
<https://seo-winner.com/What-is-CSS-and-how-does-it-benefit-us#:~:text=CSS>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม (ต่อ)

- [19] Amazon Web Services. “ภาษา Javascript.” เข้าถึงได้จาก :  
<https://aws.amazon.com/th/what-is/javascript/>.
- [20] CMPRODEV. “Javascript.” เข้าถึงได้จาก :  
<https://www.cmprodev.com/blog/javascript-for-website.html>
- [21] mikelopster. “Vue Firebase Masterclass.” เข้าถึงได้จาก :  
<https://docs.mikelopster.dev/c/vue-firebase/intro>
- [22] VLink. “Why Use Vue JS? The Ultimate Guide for Frontend Every Aspect Covered.” เข้าถึงได้จาก :  
<https://vlinkinfo.com/blog/why-use-vue-js-the-ultimate-guide/>
- [23] CVZONE. “Face Recognition with Real Time Database.” เข้าถึงได้จาก :  
<https://www.computervision.zone/courses/face-recognition-with-real-timedatabase/>
- [24] Electronic Clinic. (2565). “ESP32 Cam Live Video Streaming in Python OpenCV.” เข้าถึงได้จาก :  
<https://www.electronicclinic.com/esp32-cam-with-python-opencv-yolo-v3-for-object-detection-and-identification/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>
#include <PubSubClient.h>
#include <Adafruit_AMG88xx.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define FLASH_PIN 4

// WiFi and MQTT settings
const char* WIFI_SSID = "Yok";
const char* WIFI_PASS = "12345678";
const char* mqtt_broker = "broker.emqx.io";
const char* mqtt_username = "emqx";
const char* mqtt_password = "public";
const int mqtt_port = 1883;

LiquidCrystal_I2C lcd(0x27, 20, 4);
WiFiClient espClient;
PubSubClient client(espClient);
Adafruit_AMG88xx amg8853;

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned long previousMillis = 0;
const long interval = 3000;
char msgBuffer[1024];
float temperatures[64];

void setup() {
  Serial.begin(115200);
  lcd.begin(20, 4);
  lcd.backlight();

  Wire.begin(14, 15);
  lcd.begin(20, 4);
  lcd.backlight();
  pinMode(FLASH_PIN, OUTPUT);
  digitalWrite(FLASH_PIN, LOW);
  if (!amg8853.begin(0x68)) {
    Serial.println("Could not find a valid AMG88xx sensor, check wiring!");
    while (1);
  }

  connectWiFi();
  setupMQTT();

  {
    using namespace esp32cam;
    Config cfg;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    cfg.setPins(pins::AiThinker);
    cfg.setResolution(hiRes);
    cfg.setBufferCount(2);
    cfg.setJpeg(80);

    bool ok = Camera.begin(cfg);
    Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
}

server.on("/cam-lo.jpg", handleJpgLo);
server.on("/cam-hi.jpg", handleJpgHi);
server.on("/cam-mid.jpg", handleJpgMid);
server.begin();

Serial.print("http://");
Serial.println(WiFi.localIP());
Serial.println(" /cam-lo.jpg");
Serial.println(" /cam-hi.jpg");
Serial.println(" /cam-mid.jpg");
}

void loop() {
    server.handleClient();
    if (!client.connected()) {
        connectMQTT();
    }
    client.loop();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi disconnected, reconnecting...");
    connectWiFi();
}

unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    amg8853.readPixels(temperatures);

    for (int i = 0; i < 64; i += 8) {
        String msgStr = "";
        for (int j = i; j < i + 8; j++) {
            msgStr += String(temperatures[j], 2);
            if (j < i + 7) msgStr += ",";
        }

        msgStr.toCharArray(msgBuffer, msgStr.length() + 1);

        Serial.print("PUBLISH CHUNK: ");
        Serial.println(msgStr);

        if (client.publish("ESPFaceAtt/amg", msgBuffer)) {
            Serial.println("Chunk published successfully");
        } else {
            Serial.println("Failed to publish chunk");
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Serial.print("MQTT state: ");
        Serial.println(client.state());
        break;
    }

    delay(50);
}
}
}

// WiFi connection
void connectWiFi() {
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.println("Connecting to WiFi...");
    }
    Serial.println("Connected to the WiFi network");
}

// MQTT setup
void setupMQTT() {
    client.setServer(mqtt_broker, mqtt_port);
    client.setCallback(mqttCallback);
    connectMQTT();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void connectMQTT() {
    while (!client.connected()) {
        String client_id = "esp32-client-";
        client_id += String(WiFi.macAddress());
        Serial.printf("The client %s connects to the public mqtt broker\n",
client_id.c_str());

        if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
            Serial.println("Public emqx mqtt broker connected");
            client.subscribe("ESPFaceAtt/pc");
        } else {
            Serial.print("Failed to connect to MQTT broker, state: ");
            Serial.println(client.state());
            delay(2000);
        }
    }
}

// MQTT message callback
void mqttCallback(char* topic, byte* message, unsigned int length) {
    char messageTemp[50]; // Adjust size as needed
    memcpy(messageTemp, message, length);
    messageTemp[length] = '\0';

    char name[20], time[20], status[20];
    sscanf(messageTemp, "%19[^\n]%19[^\n]%19[^\n]", name, time, status);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Name: ");
lcd.print(name);

lcd.setCursor(0, 1);
lcd.print("Time: ");
lcd.print(time);

lcd.setCursor(0, 2);
lcd.print("Status: ");
lcd.print(status);
Serial.println(messageTemp);
delay(3000);
lcd.clear();
}

void serveJpg() {
  auto frame = esp32cam::capture();
  if (frame == nullptr) {
    Serial.println("CAPTURE FAIL");
    server.send(503, "", "");
    return;
  }
  Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
    static_cast<int>(frame->size()));
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

server.setContentLength(frame->size());
server.send(200, "image/jpeg");
WiFiClient client = server.client();
frame->writeTo(client);
}

void handleJpgLo() {
  if (!esp32cam::Camera.changeResolution(loRes)) {
    Serial.println("SET-LO-RES FAIL");
  }
  serveJpg();
}

void handleJpgHi() {
  if (!esp32cam::Camera.changeResolution(hiRes)) {
    Serial.println("SET-HI-RES FAIL");
  }
  serveJpg();
}

void handleJpgMid() {
  if (!esp32cam::Camera.changeResolution(midRes)) {
    Serial.println("SET-MID-RES FAIL");
  }
  serveJpg();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import cv2
import numpy as np
import os
import pickle
import tensorflow as tf
import firebase_admin
from firebase_admin import credentials, db
from datetime import datetime, timedelta
from sklearn.preprocessing import LabelEncoder
from keras_facenet import FaceNet
from collections import defaultdict
import time
import schedule
import paho.mqtt.client as mqtt
import urllib.request

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

facenet = FaceNet()
faces_embeddings = np.load("faces_embeddings_done_4classes.npz")
Y = faces_embeddings['arr_1']
encoder = LabelEncoder()
encoder.fit(Y)

haarcascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
model = pickle.load(open("svm_model_160x160.pkl", 'rb'))

cred = credentials.Certificate('D:/FACERECOGNITION/Python/serviceAccountKey.json')
firebase_admin.initialize_app(cred, {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'databaseURL': 'https://backend-1e9af-default-rtdb.asia-
southeast1.firebaseio.com/'
})

def mark_absent_if_needed():
    employees_ref = db.reference('employees')
    employees_data = employees_ref.get()

    if not isinstance(employees_data, list):
        print("Error: Employee data format is incorrect.")
        return

    today = datetime.now()
    yesterday = today - timedelta(days=1)
    yesterday_str = yesterday.strftime("%Y-%m-%d")

    for index, employee in enumerate(employees_data):
        attendance_ref = db.reference(f'employees/{index}/attendance')
        existing_data = attendance_ref.get()

        if isinstance(existing_data, list):
            already_checked_in = any(record.get("date") == yesterday_str for record in
existing_data)
        else:
            already_checked_in = False

        if not already_checked_in:
            next_index = len(existing_data) if existing_data else 0
            attendance_ref.child(str(next_index)).set({

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        "date": yesterday_str,
        "status": "Absent"
    })
    print(f"Marked {employee.get('name', 'Unknown')} as Absent for
        {yesterday_str}")

schedule.every().day.at("00:01").do(mark_absent_if_needed)

def log_attendance(name):
    employees_ref = db.reference('employees')
    employees_data = employees_ref.get()

    if not isinstance(employees_data, list):
        print("Error: Employee data format is incorrect.")
        return

    employee_index = None
    for index, employee in enumerate(employees_data):
        if str(employee.get('name')) == str(name):
            employee_index = index
            break

    if employee_index is None:
        print(f"Employee name {name} not found!")
        return

    now = datetime.now()
    date = now.strftime("%Y-%m-%d")
    current_time = now.strftime("%H:%M")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
attendance_ref = db.reference(f'employees/{employee_index}/attendance')
existing_data = attendance_ref.get()
```

```
if isinstance(existing_data, list):
```

```
    for idx, record in enumerate(existing_data):
```

```
        if record.get("date") == date:
```

```
            if "time_in" in record and current_time < "12:01":
```

```
                print(f" {name} already checked in before 12:01 PM. No update  
needed.")
```

```
                return
```

```
            if "time_in" in record and "time_out" not in record and current_time >=  
"12:01":
```

```
                time.sleep(3)
```

```
                attendance_ref.child(str(idx)).set({
```

```
                    "date": record["date"],
```

```
                    "time_in": record["time_in"],
```

```
                    "status": record["status"],
```

```
                    "time_out": current_time
```

```
                })
```

```
                print(f" Updated time_out for {name} at {current_time}")
```

```
                return
```

```
            if "time_out" in record:
```

```
                print(f" {name} already has time_out recorded: {record['time_out']}")
```

```
                return
```

```
status = "Late" if current_time > "09:00" else "On Time"
```

```
next_index = len(existing_data) if existing_data else 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

time.sleep(3)
attendance_ref.child(str(next_index)).set({
    "date": date,
    "time_in": current_time,
    "status": status
})
print(f" Added attendance record for {name}: {status}")

THRESHOLD = 0.65
FRAME_BUFFER_SIZE = 10
TEMPERATURE_VARIANCE_THRESHOLD = 0.5

def bilinear_interpolation(input_array, output_size):
    input_size = input_array.shape[0]
    scale = (output_size - 1) / (input_size - 1)
    output_array = np.zeros((output_size, output_size))
    for i in range(output_size):
        for j in range(output_size):
            x = i / scale
            y = j / scale
            x0, y0 = int(x), int(y)
            x1, y1 = min(x0 + 1, input_size - 1), min(y0 + 1, input_size - 1)
            dx, dy = x - x0, y - y0
            value = (
                input_array[x0, y0] * (1 - dx) * (1 - dy) +
                input_array[x0, y1] * (1 - dx) * dy +
                input_array[x1, y0] * dx * (1 - dy) +
                input_array[x1, y1] * dx * dy
            )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        output_array[i, j] = value
    return output_array

url = 'http://172.20.10.2/cam-lo.jpg'
MQTT_BROKER = 'broker.emqx.io'
MQTT_PORT = 1883
MQTT_TOPIC_SUB1 = 'ESPFaceAtt/amg'
MQTT_TOPIC_PUB = 'ESPFaceAtt/pc'

temperature_data = []

def on_message(client, userdata, message):
    global temperature_data
    if message.topic == MQTT_TOPIC_SUB1:
        data_str = message.payload.decode()
        data_list = [float(value) for value in data_str.split(',')]
        if len(temperature_data) < 8:
            temperature_data.append(data_list)
        if len(temperature_data) >= 8:
            print("Received 8x8 Temperature Grid:")
            for row in temperature_data:
                print(row)

client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2)
client.on_message = on_message
client.connect(MQTT_BROKER, MQTT_PORT)
client.subscribe(MQTT_TOPIC_SUB1)
client.loop_start()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

interpolated_data = bilinear_interpolation(temperature_data, 30)

cap = cv2.VideoCapture(0)
name_buffer = defaultdict(list)

while True:
    img_resp = urllib.request.urlopen(url)
    imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)
    frame = cv2.imdecode(imgnp, -1)

    rgb_img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    gray_img = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = haarcascade.detectMultiScale(gray_img, 1.3, 5)

    classification = "Unknown"

    for (x, y, w, h) in faces:
        img = rgb_img[y:y+h, x:x+w]
        img = cv2.resize(img, (160, 160))
        img = np.expand_dims(img, axis=0)
        ypred = facenet.embeddings(img)
        probabilities = model.predict_proba(ypred) if hasattr(model, "predict_proba")
            else None
        max_prob = np.max(probabilities) if probabilities is not None else
            np.max(model.decision_function(ypred))
        predicted_class = np.argmax(probabilities) if probabilities is not None else
            np.argmax(model.decision_function(ypred))
        face_name = encoder.inverse_transform([predicted_class])[0] if max_prob >=
            THRESHOLD else "Unknown"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if face_name != "Unknown":
    x0, y0 = int(x * 30 / frame.shape[1]), int(y * 30 / frame.shape[0])
    x1, y1 = int((x + w) * 30 / frame.shape[1]), int((y + h) * 30 / frame.shape[0])
    face_temperature = interpolated_data[y0:y1, x0:x1]
    temperature_variance = np.var(face_temperature) if face_temperature.size
        > 0 else 0
    classification = "Real Person" if temperature_variance >
        TEMPERATURE_VARIANCE_THRESHOLD else "Picture"

    if classification == "Real Person":
        log_attendance(face_name)

    color = (0, 255, 0) if classification == "Real Person" else (0, 0, 255)
    cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
    cv2.putText(frame, face_name, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9,
        color, 2)

    schedule.run_pending()
    temperature_data = []

    cv2.imshow("Face Recognition with Attendance", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cv2.destroyAllWindows()
client.loop_stop()
client.disconnect()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import os
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
from mtcnn.mtcnn import MTCNN
from keras_facenet import FaceNet

```

```

class FACELOADING:

```

```

    def __init__(self, directory):
        self.directory = directory
        self.target_size = (160, 160)
        self.X = []
        self.Y = []
        self.detector = MTCNN()

    def extract_face(self, filename):
        img = cv.imread(filename)
        img = cv.cvtColor(img, cv.COLOR_BGR2RGB)
        faces = self.detector.detect_faces(img)
        if len(faces) == 0:
            print(f"No faces detected in {filename}")
            return None

        x, y, w, h = faces[0]['box']
        x, y = abs(x), abs(y)
        face = img[y:y+h, x:x+w]
        face_arr = cv.resize(face, self.target_size)
        return face_arr

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def load_faces(self, dir):
    FACES = []
    for im_name in os.listdir(dir):
        try:
            path = os.path.join(dir, im_name)
            single_face = self.extract_face(path)
            if single_face is not None:
                FACES.append(single_face)
        except Exception as e:
            print(f"Error processing {im_name}: {e}")
    return FACES

def load_classes(self):
    for sub_dir in os.listdir(self.directory):
        path = os.path.join(self.directory, sub_dir)
        FACES = self.load_faces(path)
        labels = [sub_dir for _ in range(len(FACES))]
        print(f"Loaded successfully: {len(labels)}")
        self.X.extend(FACES)
        self.Y.extend(labels)

    return np.asarray(self.X), np.asarray(self.Y)

```

```

def plot_images(self):
    plt.figure(figsize=(18, 16))
    for num, image in enumerate(self.X):
        ncols = 3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

nrows = len(self.Y) // ncols + 1
plt.subplot(nrows, ncols, num + 1)
plt.imshow(image)
plt.axis('off')

faceloading = FACELOADING(r"D:\IndiaFACE\FACERECSVM\DATASET")
X, Y = faceloading.load_classes()

embedder = FaceNet()

def get_embedding(face_img):
    face_img = face_img.astype('float32') # 3D(160x160x3)
    face_img = np.expand_dims(face_img, axis=0) # 4D (None, 160, 160, 3)
    yhat = embedder.embeddings(face_img)
    return yhat[0] # 512D image (1x1x512)

EMBEDDED_X = []

for img in X:
    EMBEDDED_X.append(get_embedding(img))

EMBEDDED_X = np.asarray(EMBEDDED_X)
np.savez_compressed('faces_embeddings_done_4classes.npz', EMBEDDED_X, Y)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# Importing necessary libraries
import cv2 as cv
import numpy as np
import os
import pandas as pd
import matplotlib.pyplot as plt
from mtcnn.mtcnn import MTCNN
from keras_facenet import FaceNet
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.svm import SVC
import seaborn as sns
import pickle
# Load the .npz file with custom keys
data = np.load('faces_embeddings_done_4classes500v2.npz')
# Access the arrays by the custom keys
EMBEDDED_X = data['arr_0']
Y = data['arr_1']
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()
encoder.fit(Y)
Y = encoder.transform(Y)

plt.plot(EMBEDDED_X[0])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

plt.ylabel(Y[0])

from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(EMBEDDED_X, Y, shuffle=True,
random_state=17)

from sklearn.svm import SVC

model = SVC(C=0.7, kernel='rbf', gamma='scale', probability=True)
model.fit(X_train, Y_train)

ypreds_train = model.predict(X_train)
ypreds_test = model.predict(X_test)

from sklearn.metrics import accuracy_score
accuracy_score(Y_train, ypreds_train)

accuracy_score(Y_test, ypreds_test)

from sklearn.model_selection import cross_val_score

scores = cross_val_score(model, EMBEDDED_X, Y, cv=5)
print(f"Cross-Validation Mean Accuracy: {scores.mean()}")

from sklearn.metrics import classification_report

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

test_report = classification_report(Y_test, ypreds_test,
target_names=encoder.classes_)
print("Classification Report - Test Data:")
print(test_report)

```

```

train_report = classification_report(Y_train, ypreds_train,
target_names=encoder.classes_)
print("Classification Report - Train Data:")
print(train_report)

```

```

from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
cm_test = confusion_matrix(Y_test, ypreds_test)

# Plot confusion matrix for test data
plt.figure(figsize=(8, 6))
sns.heatmap(cm_test, annot=True, fmt='d', cmap='Blues',
xticklabels=encoder.classes_, yticklabels=encoder.classes_)
plt.title('Confusion Matrix - Test Data')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Generate confusion matrix for training data
cm_train = confusion_matrix(Y_train, ypreds_train)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# Plot confusion matrix for training data
plt.figure(figsize=(8, 6))
sns.heatmap(cm_train, annot=True, fmt='d', cmap='Blues',
            xticklabels=encoder.classes_, yticklabels=encoder.classes_)
plt.title('Confusion Matrix - Train Data')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

import pickle
#save the model
with open('svm_model500v2.pkl','wb') as f:
    pickle.dump(model,f)

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้