ระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้

FIRE SURVEILLANCE AND ALERT SYSTEM

โดย

นายธนบดินทร์ จันทมานิตย์

นายธนปราชญ์ จันทรเสนา

นายพิพัฒน์ คำศรีไพร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้

FIRE SURVEILLANCE AND ALERT SYSTEM

โดย

นายธนบดินทร์	จันทมานิตย์	62010348
นายธนปราชญ์	จันทรเสนา	62010350
นายพิพัฒน์	คำศรีไพร	62010633

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. สมเกียรติ ฤกษ์วรัญญู

รองศาสตราจารย์ ดร. มนตรี คำเงิน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้

FIRE SURVEILLANCE AND ALERT SYSTEM

ผู้จัดทำ

1. นายธนบดินทร์ จันทมานิตย์ 62010348
2. นายธนปราชญ์ จันทรเสนา 62010350
3. นายพิพัฒน์ คำศรีไพร 62010633

..... **อาจารย์ที่ปรึกษา**

(ผู้ช่วยศาสตราจารย์ ดร.สมเกียรติ ฤกษ์วรัญญู)

..... **อาจารย์ที่ปรึกษาร่วม**

(รองศาสตราจารย์ ดร.มนตรี คำเงิน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง “ระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้” สำเร็จลุล่วงไปได้ด้วยดี เนื่องด้วยได้รับความช่วยเหลือและความอนุเคราะห์อย่างดียิ่งจาก ผศ.ดร.สมเกียรติ ฤกษ์วรัญญู และ รศ.ดร.มนตรี คำเงิน ที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษา ค้นคว้าวิจัยให้ปริญญานิพนธ์นี้สำเร็จสมบูรณ์ยิ่งขึ้น รวมถึงสนับสนุนสถานที่ เครื่องมือ และอุปกรณ์ต่าง ๆ ที่ใช้ระหว่างการจัดทำปริญญานิพนธ์

ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้เป็นสถาบันการศึกษาชั้นนำในด้านวิศวกรรมศาสตร์ เป็นสภาพแวดล้อมที่ผู้จัดทำได้ตื่นรู้และเติมเต็มประสบการณ์ ทักษะ ระยะเวลาที่ได้ศึกษา

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอน ประสิทธิ์ประสาทวิชาความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษาอันมีค่ายิ่งแก่ผู้จัดทำด้วยดี

นายธนบดินทร์ จันทมานิตย์
นายธนปราชญ์ จันทรเสนา
นายพิพัฒน์ คำศรีไพร
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้
FIRE SURVEILLANCE AND ALERT SYSTEM

โดย นายธนบดินทร์ จันทมานิตย์ 62010348
นายธนปราชญ์ จันทรเสนา 62010350
นายพิพัฒน์ คำศรีไพร 62010633

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.สมเกียรติ ฤกษ์วรัญญู
อาจารย์ที่ปรึกษาร่วม รองศาสตราจารย์ ดร.มนตรี คำเงิน

บทคัดย่อ

ปัจจุบันอัคคีภัยหรือเหตุเพลิงไหม้โดยเฉพาะในครัวเรือนพบบ่อยและมีแนวโน้มเพิ่มสูงขึ้น ซึ่งเป็นอุบัติเหตุที่ส่งผลกระทบอย่างยิ่งต่อชีวิตและทรัพย์สิน ดังนั้น การเข้าถึงและป้องกันการลุกลามของเพลิงไหม้อย่างทันกาลจึงเป็นเรื่องสำคัญ ปัจจัยที่มีผลต่อการป้องกันคือการตรวจจับและเข้าถึงพื้นที่เพลิงไหม้ได้อย่างรวดเร็วและมีประสิทธิภาพ ซึ่งปริญญานิพนธ์นี้ออกแบบระบบแจ้งเตือนเพลิงไหม้ในครัวเรือน โดยแบ่งระดับการแจ้งเตือนภัยจากระดับความรุนแรงของเพลิงไหม้เพื่อออกแบบและพัฒนาระบบเฝ้าระวังและแจ้งเตือนเหตุเพลิงไหม้ โดยใช้การวิเคราะห์ภาพเคลื่อนไหวและเซ็นเซอร์ (Sensor) ที่สามารถเข้าถึงข้อมูลได้ผ่านอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

Nowadays, fires especially in households, are more common and tend to increase. It's an accident that has a profound impact on life and property. It is important to reach out and prevent the sudden spread of fires. A factor affecting prevention is the detection and quick and effective access to the fire area. This dissertation designed a household fire alarm system, dividing the alarm level from the severity level of the fire to designing and developing a fire surveillance and alert system. It uses motion picture analysis and sensors that can access data over the Internet.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		IV
สารบัญรูป		VI
สารบัญตาราง		IX
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	2
	1.3 ขอบเขตของปริญญานิพนธ์	2
	1.4 แผนการปฏิบัติงาน	3
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 ทฤษฎีการเกิดไฟ	3
	2.2 การตรวจจับวัตถุ (Object Detection)	8
	2.2.1 YOLOv2	10
	2.3 การใช้งานของระบบตรวจจับสัญญาณภาพเพลิงไหม้	10
	2.3.1 ระบบตรวจจับไฟไหม้ (Early Fire Detection)	10
	2.3.2 การจับภาพเคลื่อนไหว (Motion Detection)	11
	2.4 ศึกษาข้อมูลของระบบคลาวด์ ฐานข้อมูล และเว็บไซต์	12
	2.4.1 คลาวด์คอมพิวติ้ง (Cloud Computing)	14
	2.4.2 ฐานข้อมูลระบบคลาวด์ (Cloud Database)	17
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**21**
	3.1 การออกแบบ	21
	3.1.1 การออกแบบการทำงานของระบบ	21
	3.1.2 การติดตั้งระบบปฏิบัติการสำหรับบอร์ด Corgi Dude	24
	3.1.3 การเตรียมชุดข้อมูลสำหรับเทรนโมเดล	25
	3.1.4 ทำการถ่ายคลิปวิดีโอและทำการแปลงเป็นไฟล์รูปภาพ	26
	3.1.5 การออกแบบโมเดล	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	
	3.1.6 การพัฒนาโปรแกรมเทคนิคการตรวจจับ	33
	3.1.7 การออกแบบการส่งแจ้งเตือนเพลิงไหม้ผ่านแอพพลิเคชั่นไลน์	33
	3.2 เครื่องมือที่ใช้ในการทดลอง	37
	3.2.1 บอร์ด Corgi Dude	37
	3.2.2 โมดูลกล้อง OV2640 2MP	37
	3.2.3 โมดูลจอแสดงผลแบบสี ขนาด 1.3 นิ้ว ความละเอียด 240x240	37
	3.2.4 โมดูลบัซเซอร์	37
	3.2.5 อุปกรณ์แปลงแรงดันไฟฟ้า	37
	3.3 การจัดเก็บผลการทดลอง	39
	3.3.1 การทดสอบการทำงานของโมเดลแบบจำลอง	39
	3.3.2 การทดสอบจากแบบจำลองที่ผ่านการเรียนรู้	39
	3.3.3 การทดสอบการทำงานบอร์ด Corgi Dude	39
	3.3.4 การทดสอบการทำงานร่วมกับแอพพลิเคชันไลน์	39
บทที่ 4	**ผลการทดลอง**	**40**
	4.1 ผลการทดสอบการติดตั้งเฟิร์มแวร์ของ Corgi Dude	40
	4.2 ผลการทดสอบการเทรนโมเดล	40
	4.3 ผลการทดสอบจากโมเดลที่ผ่านการเทรน	42
	4.4 ผลการนำโมเดลมาใช้งานจริงในการตรวจจับ	42
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**46**
	5.1 สรุปผล	46
	5.2 ปัญหาและอุปสรรคในการทำปริญญานิพนธ์	47
	5.3 ข้อเสนอแนะงานปริญญานิพนธ์	47
	บรรณานุกรม	**48**
	ภาคผนวก	**50**
	ภาคผนวก ก โค้ดโปรแกรม	**51**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	องค์ประกอบของการติดไฟ (Fire Triangle)	3
2.2	ปฏิกิริยาเคมีของการเผาไหม้	4
2.3	ตัวอย่างการทำ Object Detection	8
2.4	ตัวอย่างการถมสีให้ทุก Pixel ของวัตถุนั้น (Segmentation)	8
2.5	ตัวอย่างการหาค่า TP, FP, FN	9
2.6	วิธีการหาค่า IoU	9
2.7	โครงสร้างของ YOLO	10
2.8	ระบบตรวจจับไฟไหม้ Early Fire Detection	11
2.9	ภาพรวมการให้บริการคลาวด์ (cloud computing)	13
2.10	สถาปัตยกรรมของซอฟต์แวร์คลาวด์	14
2.11	โปรโตคอล OpenFlow	17
2.12	โครงสร้างของคลาวด์	19
2.13	การทำงานของโหนด	20
2.14	เปรียบเทียบความสมบูรณ์ของข้อมูล ความปลอดภัย และตำแหน่งการจัดเก็บ	20
3.1	แผนภาพบล็อกการทำงานของระบบเฝ้าระวังเพลิงไหม้	22
3.2	แสดงผังงานของระบบเฝ้าระวังเพลิงไหม้	23
3.3	วงจรเครื่องตรวจจับเพลิงไหม้	24
3.4	ทำการติดตั้งระบบปฏิบัติการ Firmware MaixPy ลงบอร์ด	25
3.5	การดาวน์โหลดข้อมูลระบบปฏิบัติการสำเร็จ	25
3.6	คลิปแสดงการจำลองไฟไหม้ในห้องครัว	26
3.7	คลิปแสดงการจำลองไฟไหม้ในข้างนอกอาคาร	27
3.8	ข้อมูลได้รับการรวบรวมข้อมูลภาพที่มีควันไฟและไฟไหม้	27
3.9	การทำ Image Labeling	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.10	ตัวอย่างไฟล์ผลลัพธ์	28
3.11	โครงสร้างโฟลเดอร์หลังจาก LabelIMG	29
3.12	ติดตั้งไลบรารี่ aXeleRate	29
3.13	ชุดคำสั่งสำหรับแสดงผล datasets	30
3.14	ชุดคำสั่งที่ทำการแบ่งชุดข้อมูลเอาไว้ใช้ validation ร้อยละ 20 ของชุดข้อมูล	30
3.15	ชุดคำสั่งกำหนด config ในการเทรนโมเดล	31
3.16	ชุดคำสั่งทำการเช็คว่าเครื่องเชื่อมต่อกับการ์ดจอ	32
3.17	ชุดคำสั่งสำหรับการเทรนโมเดล	32
3.18	ชุดคำสั่งสำหรับทดสอบผลลัพธ์	33
3.19	เข้าสู่ระบบแอพพลิเคชันไลน์	33
3.20	เข้าเมนู “My page”	34
3.21	ทำการออก Token สำหรับนำ API มาเชื่อมต่อ	34
3.22	ตั้งชื่อการแจ้งเตือนที่จะแสดงและเลือกผู้ใช้งานที่ต้องการแจ้งเตือนไปยังผู้ใช้	35
3.23	คัดลอก token เพื่อไปใช้ในโปรแกรม Python	35
3.24	ข้อมูล API ในส่วน Notification ของแอพพลิเคชันไลน์	36
3.25	ตัวอย่างโค้ด Micro Python เพื่อส่งข้อความไปยังแอพพลิเคชันไลน์	36
3.26	อุปกรณ์ตรวจจับเพลิงไหม้	37
3.27	อุปกรณ์แปลงแรงดันไฟฟ้า	38
3.28	Laptop Acer Aspire 3 A315-5122	38
3.29	Laptop Lenovo YOGA 530	39
4.1	ผลการอัปโหลดเฟิร์มแวร์	40
4.2	แสดงการเทรนนิ่งโมเดล	40
4.3	ชุดโมเดลหลังจากทำการเทรน	41
4.4	ผลลัพธ์จากการตรวจสอบความแม่นยำสูงที่สุดของโมเดล	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

4.5 ภาพเปรียบระหว่างภาพดั้งเดิมกับแบบจำลองที่ผ่านการเทรน 42
4.6 การส่งข้อมูลระดับไฟไหม้ และภาพทางโปรแกรมแอปพลิเคชันไลน์ 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

1.1 แผนงานที่จะดำเนินการภาคเรียนที่ 1 2

1.2 แผนงานที่จะดำเนินการภาคเรียนที่ 2 2

2.1 การเปรียบเทียบไฮเปอร์ไวเซอร์ (hypervisors) 16

4.1 ค่าความถูกต้องของการเทรนโมเดล 42

4.2 แสดงรูปภาพกับโมเดลที่ทำการตรวจจับ (Detection 43

4.3 เปรียบเทียบถ่ายภาพการตรวจจับในสภาวะที่แตกต่างกัน 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอัคคีภัยหรือเพลิงไหม้โดยเฉพาะในครัวเรือนพบบ่อยและมีแนวโน้มเพิ่มสูงขึ้นเนื่องจากมีพื้นที่เสี่ยงที่มีกิจวัตรการใช้ไฟเป็นประจำ เช่น ห้องครัว หากเกิดเหตุเพลิงไหม้ส่งผลกระทบโดยตรงต่อชีวิตและทรัพย์สิน รวมถึงผลกระทบต่อสิ่งแวดล้อม ดังนั้น การตรวจจับเพลิงไหม้ในพื้นที่เสี่ยงจึงเป็นเรื่องสำคัญ ปัจจัยที่มีผลต่อการป้องกันคือการตรวจจับและเข้าถึงพื้นที่เพลิงไหม้ได้อย่างรวดเร็วและมีประสิทธิภาพ

กรมป้องกันและบรรเทาสาธารณภัย (ปภ.) ข้อมูลสถิติสาธารณภัย พุทธศักราช 2565 ตั้งแต่วันที่ 1 มกราคม – 30 เมษายน พุทธศักราช 2565 พบว่าอัคคีภัย มีพื้นที่ประสบภัย จำนวน 956 หมู่บ้าน/ชุมชน[1] (เพิ่มขึ้น 273 จากปีก่อนหน้า) ใน 72 จังหวัดและกรุงเทพมหานคร จากการศึกษาและสำรวจสภาพปัญหาการควบคุมและเฝ้าระวังอัคคีภัยในพื้นที่เสี่ยงสูง เช่น บ้านเรือน ชุมชน เป็นต้น พบว่าไม่มีการใช้เทคโนโลยีการตรวจจับเพลิงไหม้ในพื้นที่ครัวในการช่วยเพิ่มประสิทธิภาพการป้องกันเหตุการณ์เพลิงไหม้

ปริญญานิพนธ์นี้ออกแบบระบบแจ้งเตือนเพลิงไหม้ในครัวเรือน โดยแบ่งระดับการแจ้งเตือนภัยจากระดับความรุนแรงของระดับเพลิงไหม้เพื่อออกแบบและพัฒนาหน้าจอแสดงผลของระบบเฝ้าระวังและแจ้งเตือนเหตุเพลิงไหม้เพื่อพัฒนาและออกแบบระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้ โดยใช้การวิเคราะห์ภาพเคลื่อนไหว และสามารถเข้าถึงข้อมูลได้ผ่านทางอินเตอร์เน็ต เพื่อเพิ่มประสิทธิภาพการป้องกันเหตุการณ์เพลิงไหม้ภายในครัวเรือน

1.2 วัตถุประสงค์

1. เพื่อออกแบบระบบแจ้งเตือนเพลิงไหม้ในภายในครัวเรือน โดยแบ่งระดับการแจ้งเตือนภัยจากระดับความรุนแรงของระดับเพลิงไหม้
2. เพื่อออกแบบและจัดทำระบบเฝ้าระวังเหตุเพลิงไหม้ โดยใช้การวิเคราะห์ภาพเคลื่อนไหวและข้อมูลจากเซ็นเซอร์ (Sensor)
3. เพื่อออกแบบและจัดทำระบบแจ้งเตือนเพลิงไหม้ และสามารถเข้าถึงข้อมูลได้ผ่านอินเตอร์เน็ต

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้ได้ทำการออกแบบและพัฒนาระบบเฝ้าระวังและแจ้งเตือนเหตุเพลิงไหม้ โดยมีขอบเขตการดำเนินงาน ดังนี้

1. จัดทำระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้ในภายในครัวเรือน
2. ใช้ Machine Learning (ML) หรือ Artificial Intelligence (AI) ในการช่วยวิเคราะห์ข้อมูลภาพเคลื่อนไหวในงานเฝ้าระวังเพลิงไหม้
3. จัดทำระบบนำเข้าข้อมูลและแสดงข้อมูลการเฝ้าระวังและแจ้งเตือนเพลิงไหม้ภายในครัวเรือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 แผนการปฏิบัติงาน

1.4.1 ภาคการศึกษาที่ 1

ตารางที่ 1.1 แผนงานที่จะดำเนินการภาคเรียนที่ 1

ช่วงการดำเนินงาน	แผนงานที่จะดำเนินการ
1 สิงหาคม ถึง 30 กันยายน 2565	- ศึกษาข้อมูล และการใช้งานของระบบตรวจจับสัญญาณภาพ - ศึกษาข้อมูลของระบบคลาวด์ (Cloud), ฐานข้อมูล (Database) และเว็บไซต์ (Website) - ศึกษาการทำงานโมดูลที่เหมาะสมกับการใช้งานของระบบ
1 ตุลาคม ถึง 31 ตุลาคม 2565	- ทำการจัดซื้ออุปกรณ์ในการทำงาน - จำลองกล้องตรวจจับที่ใช้ในการทดลอง - เขียนชุดคำสั่งในระบบการเฝ้าระวังและแจ้งเตือนเพลิงไหม้
1 พฤศจิกายน ถึง 30 พฤศจิกายน 2565	- ทำการออกแบบ และเขียนชุดคำสั่งรับข้อมูลจากกล้องที่แสดงบนหน้าจอแสดงผลและวิเคราะห์ถึงโอกาสในการเกิดเพลิงไหม้ - เขียนโปรแกรม และทดสอบการใช้งานของระบบตรวจจับสัญญาณภาพ - ทดลองใช้หน้าจอแสดงผลร่วมกันกับกล้องตรวจจับ

1.4.2 ภาคการศึกษาที่ 2

ตารางที่ 1.2 แผนงานที่จะดำเนินการภาคเรียนที่ 2

ช่วงการดำเนินงาน	แผนงานที่จะดำเนินการ
เดือนที่ 1	- เขียนชุดคำสั่งในส่วนการแสดงผลไปยังส่วนต่าง ๆ และการแจ้งเตือนไปยังแอปพลิเคชันไลน์ - สังเกตข้อผิดพลาด และแก้ไข
เดือนที่ 2	- ทำการพัฒนาระบบการเฝ้าระวังและแจ้งเตือนเพลิงไหม้และส่วนการแสดงผล - พัฒนาหน้าจอแสดงผล และระบบการแจ้งเตือนเพื่อแสดงการแจ้งเตือน
เดือนที่ 3	- จัดทำรายงานฉบับสมบูรณ์ - ตรวจสอบ และแก้ไขข้อผิดผิดพลาดของรายงานฉบับสมบูรณ์ - ส่งรายงานฉบับสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

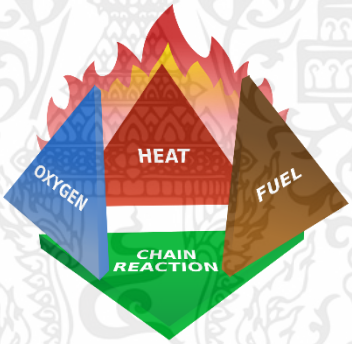
ปริญญานิพนธ์เรื่อง “ระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้” ได้ทำการออกแบบระบบเฝ้าระวังและแจ้งเตือนเพลิงไหม้ ระยะเวลาที่ผ่านมาได้ทำการดำเนินงานการศึกษาข้อมูลเกี่ยวกับ ทฤษฎีการเกิดไฟ ปัจจัยในการเกิดไฟ ระดับไฟ (Fire class) การใช้งานของระบบตรวจจับสัญญาณภาพ ศึกษาโมดูลที่เหมาะสมกับการใช้งานของระบบ การเลือกใช้กล้อง ศึกษาข้อมูลการทำงานของระบบตรวจจับวัตถุ ดังนั้น ปริญญานิพนธ์ที่นำเสนอจึงมีหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 ทฤษฎีการเกิดไฟ

องค์ประกอบของการติดไฟ (Fire Triangle) การที่จะเกิดไฟขึ้นได้นั้น ต้องมีองค์ประกอบ 3 อย่าง คือ

1) เชื้อเพลิง (fuel) ซึ่งจะอยู่ในสภาพของแข็ง ของเหลว หรือแก๊ส
2) ออกซิเจน (oxygen) ซึ่งจะมีอยู่ในอากาศประมาน 21% โดยปริมาณ
3) ความร้อน (heat) พอเพียงที่จะติดไฟได้

เมื่อมีองค์ประกอบทั้ง 3 ครบแล้วไฟจะเกิดลุกไหม้ขึ้นและเกิดปฏิกิริยาลูกโซ่



รูปที่ 2.1 องค์ประกอบของการติดไฟ (Fire Triangle)

สามเหลี่ยมของไฟ (the use of the fire triangle) สามเหลี่ยมของไฟ แสดงให้เห็นว่าไฟจะเกิดขึ้นได้ต้องมีองค์ประกอบ 3 อย่าง คือ เชื้อเพลิง(ในรูปแบบของไอระเหย) อากาศ(ออกซิเจน) และ ความร้อน(ถึงอุณหภูมิติดไฟ) และการที่จะดับไฟนั้นก็ต้องเอาอย่างใดอย่างหนึ่งออกไป ดังนั้น องค์ประกอบในการเผาไหม้มีอยู่ 4 องค์ประกอบ คือ

2.1.1 เชื้อเพลิง (Fuel)

วัตถุใด ๆ ก็ตามที่สามารถทำปฏิกิริยากับออกซิเจนได้อย่างรวดเร็วในการเผาไหม้ เช่น ก๊าซ ไม้ กระดาษ น้ำมัน โลหะ พลาสติก เป็นต้น เชื้อเพลิงที่อยู่ในสถานะก๊าซจะสามารถลุกไหม้ไฟได้ แต่เชื้อเพลิงที่อยู่ในสถานะของแข็งและของเหลวจะไม่สามารถลุกไหม้ไฟได้ ถ้าโมเลกุลที่ผิวของเชื้อเพลิงไม่อยู่ในสภาพที่เป็นก๊าซ การที่โมเลกุลของของแข็งหรือของเหลวนั้นจะสามารถแปรสภาพ กลายเป็นก๊าซได้นั้นจะต้องอาศัยความร้อนที่แตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

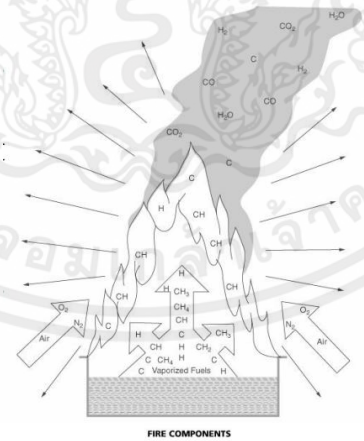
กันตามชนิดของเชื้อเพลิงแต่ละชนิด ความแตกต่างของลักษณะการติดไฟของเชื้อเพลิงดังกล่าวขึ้นอยู่กับคุณสมบัติ 4 ประการ ดังนี้

2.1.1.1 ความสามารถในการติดไฟของสาร (Flamablility Limitts) เป็นปริมาณไอของสารที่เป็นเชื้อเพลิงในอากาศที่มีคุณสมบัติซึ่งพร้อมจะติดไฟได้ในการเผาไหม้นั้นปริมาณไอเชื้อเพลิงที่ผสมกับอากาศนั้นจะต้องมีปริมาณพอ เหมาะจึงจะติดไฟได้ โดยปริมาณต่ำสุดของไอเชื้อเพลิงที่เป็น % ในอากาศ ซึ่งสามารถจุดติดไฟได้เรียกว่า “ ค่าต่ำสุดของไอเชื้อเพลิง (Lower Flammable Limit) ” และปริมาณสูงสุดของไอเชื้อเพลิงที่เป็น % ในอากาศซึ่งสามารถจุดติดไฟได้เรียกว่า “ ค่าสูงสุดของไอเชื้อเพลิง (Upper Flammable Limit) ” ซึ่งสารเชื้อเพลิงแต่ละชนิดจะมีค่าต่ำสุดและค่าสูงสุดของไอเชื้อเพลิงแตกต่างกันไป

2.1.1.2 จุดวาบไฟ (Flash Point) คืออุณหภูมิที่ต่ำที่สุด ที่สามารถทำให้เชื้อเพลิงคายไอออกมาผสมกับอากาศในอัตราส่วน ที่เหมาะสมถึงจุดที่มีค่าต่ำสุดถึงค่าสูงสุดของไอเชื้อเพลิง เมื่อมีประกายไฟก็จะเกิดการติดไฟ เป็นไฟวาบขึ้นและก็ดับ

2.1.1.3 จุดติดไฟ (Fire Point) คืออุณหภูมิของสารที่เป็นเชื้อเพลิงได้รับความร้อน จนถึงจุดที่จะติดไฟได้แต่การติดไฟนั้นจะต้องต่อเนื่องกันไป โดยปกติความร้อนของ Fire Point จะสูงกว่า Flash Point ประมาณ 7 องศาเซลเซียส

2.1.1.4 ความหนาแน่นไอ (Vapor Density) คืออัตราส่วนของน้ำหนักของสารเคมีในสถานะก๊าซต่อน้ำหนักของอากาศเมื่อมีปริมาณเท่ากัน ความหนาแน่นไอ ใช้เป็นสิ่งบ่งบอกให้ทราบว่าก๊าซนั้นจะหนักหรือเบากว่าอากาศซึ่งใช้เป็นข้อมูลในการควบคุมอัคคีภัย



รูปที่ 2.2 ปฏิกิริยาเคมีของการเผาไหม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ออกซิเจน (Oxygen)

อากาศที่อยู่รอบ ๆ ตัวเรา นั้นมีก๊าซออกซิเจนเป็นองค์ประกอบ ประมาณ 21 % แต่การเผาไหม้แต่ละครั้งนั้นจะต้องการออกซิเจนประมาณ 16 % เท่านั้น ดังนั้นจะเห็นว่าเชื้อเพลิงทุกชนิดที่อยู่ในบรรยากาศรอบ ๆ ตัวเรานั้นจะถูกล้อมรอบด้วยออกซิเจน ซึ่งมีปริมาณเพียงพอสำหรับการเผาไหม้ยิ่งถ้าปริมาณออกซิเจนยิ่งมากเชื้อเพลิงก็ยิ่งติดไฟได้ดีขึ้น และเชื้อเพลิงบางประเภทจะมีออกซิเจนในตัวเองอย่างเพียงพอที่จะทำให้ตัวเองไหม้ได้โดยไม่ต้องใช้ออกซิเจนที่อยู่โดยรอบเลย

2.1.3 ความร้อน (Heat)

ความร้อน คือ พลังงานที่ทำให้เชื้อเพลิงแต่ละชนิดเกิดการคายไอออกมา

2.1.4 ปฏิกิริยาลูกโซ่ (Chain Reaction)

การเผาไหม้อย่างต่อเนื่อง คือ กระบวนการเผาไหม้ที่เริ่มตั้งแต่เชื้อเพลิงได้รับความร้อนจนติดไฟเมื่อเกิดไฟขึ้น หมายถึง การเกิดปฏิกิริยา กล่าวคืออะตอมจะถูกเหวี่ยงออกจากโมเลกุลของเชื้อเพลิง กลายเป็นอนุมูลอิสระ และอนุมูลอิสระเหล่านี้จะกลับไปอยู่ที่ฐานของไฟอย่างรวดเร็ว ทำให้เกิดเปลวไฟ

อนุมูลอิสระที่ใช้งาน (การเริ่มต้นลูกโซ่) อนุมูลกลาง (การขยายพันธุ์แบบลูกโซ่และต่อเนื่องกัน) การสิ้นสุดลูกโซ่

2.1.5 ระยะการเกิดไฟไหม้

ไฟไหม้ขั้นต้น ตั้งแต่เห็นเปลวไฟจนถึง 4 นาที สามารถดับได้โดยใช้เครื่องดับเพลิงเบื้องต้น

ไฟไหม้ขั้นปานกลาง ระยะเวลาไฟไหม้ไปแล้ว 4 นาที ถึง 8 นาที อุณหภูมิจะสูงมากเกิน กว่า 400 °C

ไฟไหม้ขั้นรุนแรง ระยะเวลาไฟไหม้ต่อเนื่องไปแล้วเกิน 8 นาทีและยังมีเชื้อเพลิงอีกมากมาย อุณหภูมิจะสูงมากกว่า 600 °C ไฟจะลุกลามขยายตัวไปทุกทิศทางอย่างรุนแรงและรวดเร็ว

2.1.6 ระดับของไฟ (Fire class)

ระดับไฟ (Fire class) คือ ระบบการจัดประเภทไฟตามประเภทของวัสดุและเชื้อเพลิงสำหรับการเผาไหม้ มักถูกกำหนดให้กับไฟประเภทต่าง ๆ แตกต่างกันตามประเทศ มีมาตรฐานแยกต่างหากสำหรับสหรัฐอเมริกา ยุโรป และออสเตรเลีย ใช้เพื่อกำหนดชนิดของสารดับเพลิงที่สามารถใช้สำหรับประเภทไฟนั้นได้ [1]

ไฟประเภท A (Class A) สารที่ติดไฟได้ทั่วไป

ไฟประเภท A ประกอบด้วยวัสดุที่ติดไฟได้ทั่วไป เช่น ไม้ กระดาษ ผ้า และขยะเกือบทุกชนิด ฯลฯ

วิธีดับไฟประเภท A ที่ดีที่สดุ คือ การลดความร้อน (Cooling) โดยใช้น้ำ การระงับสารเคมีแบบเปียก หรือผงเคมีแห้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟประเภท B (Class B) ของเหลวไวไฟ

ไฟประเภท B คือไฟที่เชื้อเพลิงติดไฟหรือของเหลวที่ติดไฟได้ ระบบของสหรัฐอเมริการวมถึงก๊าซไวไฟใน ในระบบยุโรป/ออสเตรเลีย ของเหลวไวไฟถูกกำหนดให้ที่มีจุดวาบไฟน้อยกว่า 100 °C (212 °F) ไฟเหล่านี้เป็นไปตามหลักสี่ด้านของไฟ (ความร้อน เชื้อเพลิง ออกซิเจน ปฏิกิริยาเคมี) กับไฟที่ติดไฟได้ทั่วไป ยกเว้นว่าเชื้อเพลิงที่เป็นของเหลวไวไฟ เช่น น้ำมันเบนซิน หรือก๊าซ เช่นก๊าซธรรมชาติ ไม่ควรใช้น้ำที่เป็นของแข็งในการดับไฟประเภทนี้เพราะอาจทำให้เชื้อเพลิงกระจายและลุกลามได้ วิธีที่มีประสิทธิภาพที่สุดในการดับไฟที่เป็นของเหลวคือการยับยั้งปฏิกิริยาลูกโซ่เคมีของไฟ ซึ่งกระทำโดยสารเคมีแห้งและสาร ดับไฟฮาลอน แม้ว่าจะซับด้วย CO_2 หรือสำหรับของเหลว โฟมก็มีประสิทธิภาพเช่นกัน ฮาลอนไม่ได้รับความนิยมในช่วงไม่กี่ครั้งที่ผ่านมา (ยกเว้นระบบดับเพลิงของเครื่องบิน) เนื่องจากเป็นวัสดุที่ทำลายโอโซน พิธีสารมอนทรีออลประกาศว่าไม่ควรใช้ฮาลอนอีกต่อไป สารเคมีเช่น FM-200 ในปัจจุบันเป็นตัวยับยั้งฮาโลเจนที่แนะนำวิธีดับไฟประเภท B ที่ดีที่สุด คือ กำจัดออกซิเจนทำให้อับอากาศ โดยใช้ผงเคมีแห้งคลุมใช้ฟองโฟมคลุม

ไฟประเภท B (Class B)(US) / ไฟประเภท C (Class C)(EU/AU) ก๊าซไวไฟ

ไฟที่เชื้อเพลิงติดไฟหรือก๊าซที่ติดไฟได้จัดประเภทเป็นไฟประเภท C ในระบบยุโรป/ออสเตรเลีย และไฟประเภท B พร้อมกับของเหลวไวไฟในระบบของสหรัฐอเมริกา เนื่องจากเชื้อเพลิงมีลักษณะเป็นแก๊ส จึงดับไฟได้ยาก เทคนิคที่มีประสิทธิภาพสูงสุดสำหรับการควบคุมการเกิดเพลิงไหม้ของก๊าซไวไฟคือการหยุดการไหลของน้ำมันเชื้อเพลิง (โดยการปิดก๊อกหรือวาล์วแก๊ส) หรือเพื่อแทนที่ออกซิเจน การควบคุมเพลิงไหม้ที่เกี่ยวข้องกับก๊าซไวไฟซึ่งไม่สามารถควบคุมแหล่งก๊าซได้ต้องได้รับการจัดการอย่างระมัดระวัง หากเปลวไฟดับแล้ว แต่ก๊าซยังคงรั่วไหลอาจเกิดบรรยากาศที่ระเบิดได้ และก๊าซอาจพบแหล่งกำเนิดสำหรับการลุกไหม้นอกพื้นที่ที่ได้รับผลกระทบเดิม

ไฟประเภท D (Class D) โลหะ สารเคมีติดไฟ ได้แก่ ไฟที่ลุกไหม้โลหะติดไฟ

ไฟประเภท D เกี่ยวข้องกับโลหะที่ติดไฟได้ โดยเฉพาะโลหะอัลคาไลเช่นลิเธียมและโพแทสเซียม โลหะอัลคาไลน์เอิร์ธเช่นแมกนีเซียมและองค์ประกอบกลุ่ม 4 เช่น ไททาเนียมและเซอร์โคเนียม

ไฟโลหะแสดงถึงอันตรายเฉพาะตัว เนื่องจากผู้คนมักไม่ตระหนักถึงลักษณะของไฟเหล่านี้และไม่ได้เตรียมพร้อมรับมือกับไฟเหล่านี้อย่างเหมาะสม ดังนั้น แม้แต่ไฟโลหะขนาดเล็กก็สามารถลุกลามและกลายเป็นไฟที่ใหญ่ขึ้นในวัสดุที่ติดไฟได้ทั่วไปโดยรอบ โลหะบางชนิดเผาไหม้เมื่อสัมผัสกับอากาศหรือน้ำ (เช่นโซเดียม) ซึ่งทำให้ความเสี่ยงนี้รุนแรงขึ้น มวลของโลหะที่ติดไฟได้มักจะไม่ได้แสดงถึงความเสี่ยงจากไฟไหม้มากนัก เนื่องจากความร้อนถูกนำออกจากจุดร้อนอย่างมีประสิทธิภาพจนไม่สามารถรักษาความร้อนจากการเผาไหม้ได้ ด้วยเหตุนี้พลังงานความร้อนจำนวนมากจึงจำเป็นต่อการจุดไฟให้กับโลหะที่ติดไฟได้จำนวนมากที่อยู่ติดกัน โดยทั่วไป ไฟไหม้จากโลหะเป็นอันตรายเมื่อโลหะอยู่ในรูปของขี้เลื่อย , เครื่องจักรขี้เลื่อยหรือ "เศษโลหะ" อื่น ๆ ซึ่งเผาไหม้ได้เร็วกว่าบล็อกขนาดใหญ่เนื่องจากพื้นที่ผิวที่เพิ่มขึ้น ไฟโลหะสามารถจุดไฟได้จากแหล่งกำเนิดประกายไฟเดียวกันกับที่จะทำให้เกิดไฟไหม้ทั่วไปอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องใช้ความระมัดระวังเมื่อดับไฟโลหะ น้ำและสารดับเพลิงทั่วไปอื่น ๆ สามารถกระตุ้นไฟโลหะและทำให้มันแย่ลง สมาคมป้องกันอัคคีภัยแห่งชาติแนะนำว่าควรต่อสู้กับไฟโลหะด้วยสารดับไฟแบบผงแห้งซึ่งทำงานโดยการกลบเกลื่อนและดูดซับความร้อน โลหะที่ต่างกันต้องการสารที่แตกต่างกัน และสำหรับโลหะชนิดใดชนิดหนึ่ง สารไม่สามารถทดแทนกันได้ สารที่พบบ่อยที่สุดคือเม็ดโซเดียมคลอไรด์ และ ผงกราไฟท์ ในช่วงไม่กี่ปีที่ผ่านมา ผงทองแดงก็ถูกนำมาใช้เช่นกัน ไม่ควรสับสนเครื่องดับเพลิงชนิดผงแห้งเหล่านี้ กับถังดับเพลิงที่มี สารเคมีแห้งตัวแทน ทั้งสองไม่เหมือนกันและควรใช้ผงแห้งเท่านั้นในการดับไฟโลหะ การใช้เครื่องดับเพลิงแบบเคมีแห้งโดยไม่ได้ตั้งใจ แทนที่จะใช้ผงแห้ง อาจใช้ไม่ได้ผลหรือเพิ่มความรุนแรงของไฟโลหะได้จริง

ไฟประเภท C (Class C) (US) / ไฟประเภท E (Class E)(AU)/ ไม่ได้จัดประเภท (EU) ไฟฟ้า

ไฟที่เกี่ยวข้องกับไฟฟ้าเป็นส่วนใหญ่มีการจำแนกประเภทที่แตกต่างกันในแต่ละระบบทั้งสาม มันถูกจัดประเภทเป็นไฟประเภท E ภายใต้ระบบของออสเตรเลีย ไฟประเภท C ภายใต้ระบบอเมริกันและจำแนกตามประเภทเชื้อเพลิงที่จุดไฟภายใต้ระบบยุโรป (ซึ่งก่อนหน้านี้มีการจำแนกประเภทไฟประเภท E ด้วยระบบของออสเตรเลีย) ไฟที่เกิดจาก ไฟฟ้าคือไฟที่เกี่ยวข้องกับอุปกรณ์ไฟฟ้า ที่อาจได้รับพลังงาน เพลิงไหม้ประเภทนี้อาจเกิดจากเครื่องจักรลัดวงจรหรือสายไฟเกิน ไฟเหล่านี้อาจก่อให้เกิดอันตรายร้ายแรงต่อนักผจญเพลิงที่ใช้น้ำหรือสารนำไฟฟ้าอื่นๆ เนื่องจากไฟฟ้าอาจมาจากไฟ ผ่านน้ำ ไปยังร่างกายของนักผจญเพลิง และจาก ดินไฟฟ้าช็อตทำให้นักผจญเพลิงเสียชีวิตจำนวนมาก

ไฟไฟฟ้าอาจใช้ในลักษณะเดียวกับไฟที่ติดไฟได้ทั่วไป แต่ห้ามใช้น้ำ โฟม และสารนำไฟฟ้าอื่นๆ แม้ว่าไฟจะหรืออาจเป็นพลังงานจากไฟฟ้าก็ได้ แต่ก็สามารถสู้กับสารดับเพลิงใดๆ ที่จัดอยู่ในประเภทไฟไฟฟ้าได้ คาร์บอนไดออกไซด์ CO_2 , NOVEC 1230, FM-200 และเครื่องดับเพลิงชนิดผงเคมีแห้ง และแม้แต่เบกกิ้งโซดาก็เหมาะอย่างยิ่งสำหรับการดับไฟประเภทนี้ PKP ควรเป็นวิธีสุดท้ายในการดับไฟเนื่องจากมีแนวโน้มกัดกร่อน เมื่อไฟฟ้าดับไปยังอุปกรณ์ที่เกี่ยวข้อง โดยทั่วไปจะกลายเป็นไฟที่ติดไฟได้ตามปกติ

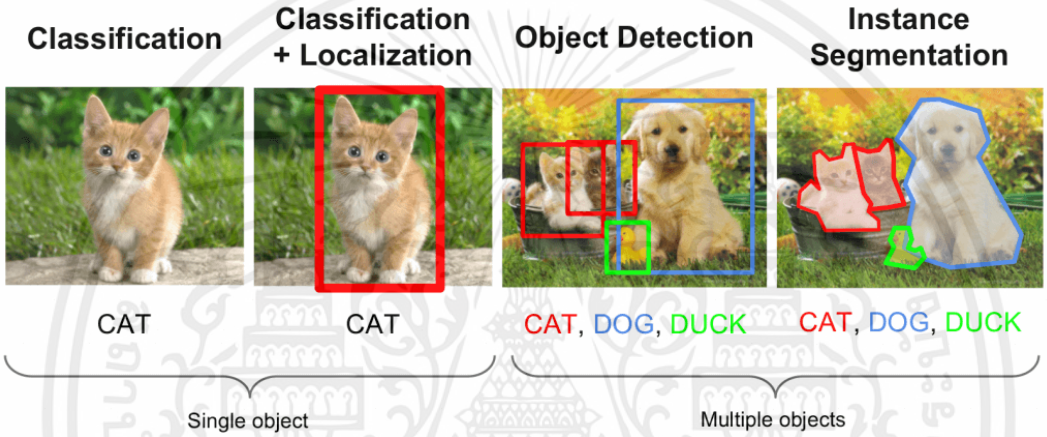
ไฟประเภท F (Class F)(EU/AU) / ไฟประเภท K (Class K)(US) น้ำมันและไขมันประกอบอาหาร (ไฟไหม้ในครัว)

ไฟที่เกี่ยวข้องกับน้ำมันปรุงอาหารและไขมันจัดเป็นไฟประเภท F ภายใต้ระบบของยุโรปและออสเตรเลีย และไฟประเภท K ภายใต้ระบบของอเมริกา แม้ว่าไฟดังกล่าวเป็นคลาสย่อยของคลาสของเหลว/ก๊าซไวไฟในทางเทคนิค แต่ลักษณะพิเศษของไฟประเภทนี้ กล่าวคือ จุดวาบไฟที่สูงกว่า ถือว่ามีความสำคัญมากพอที่จะจำแนกแยกกันได้ เครื่องดับเพลิงชนิดพิเศษบางรุ่นออกแบบมาเพื่อการใช้งานนี้ดับไฟโดยเปลี่ยนน้ำมันให้เป็นโฟม สามารถใช้ละอองน้ำได้ เช่น เดียวกับไฟประเภท B ไม่ควรใช้กระแสน้ำที่เป็นของแข็งในการดับไฟประเภทนี้เพราะอาจทำให้เชื้อเพลิงกระจายและลุกลามได้ เครื่องดับเพลิงที่เหมาะสมอาจมีหมวกคลุมเพื่อช่วยดับไฟ บางครั้งผ้าห่มกันไฟใช้สำหรับดับไฟในครัวหรือบนเตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุ (Object Detection) เป็นการค้นหาวัตถุในรูปภาพ ซึ่งสามารถจำแนกและเจาะลึกลงไปได้อีกหลายแขนง เช่น การตรวจจับหน้าคน (Face Detection) ตรวจจับคนเดินถนน (Pedestrian Detection) อีกทั้งยังสามารถนำมาประยุกต์ใช้กับงานได้อย่างหลากหลาย เช่น ใช้ในงานรักษาความปลอดภัย การนับจำนวนวัตถุ เป็นต้น การตรวจจับวัตถุ [3,4] คือ AI ตรวจจับวัตถุ ในงานทางด้าน Computer Vision ที่จะจำแนก และตรวจจับวัตถุที่อยู่ในรูป การตรวจจับ มารุกจุด มารุกพื้นที่ โดยหลักการนี้สามารถทำได้หลายวิธี การทำมารุกพื้นที่ที่นิยมได้แก่วาดกล่องรอบวัตถุ (Bounding Box) หรือ ถมสีให้ทุก Pixel ของวัตถุนั้น (เรียกว่า Segmentation) โดยตัวอย่างตามรูปที่ 2.2 และ 2.3



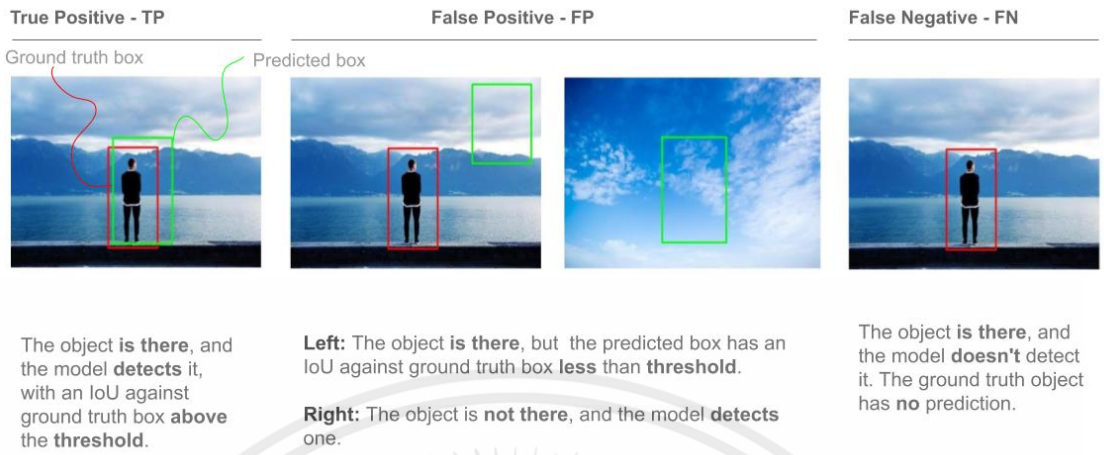
รูปที่ 2.3 ตัวอย่างการทำ Object Detection

(อ้างอิง : Standford University 2016 winter lectures CS231n Fei-Fei Li & Andrej Karpathy & Justin Johnson)

รูปที่ 2.4 ตัวอย่างการถมสีให้ทุก Pixel ของวัตถุนั้น (Segmentation)

(อ้างอิง : http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวอย□างการหาค□า TP, FP, FN

จากรูปที่ 2.5 กรอบสีแดงคือกรอบวัตถุความจริง และกรอบสีเขียวคือกรอบทำนายวัตถุ ค่า IoU ที่ได้จากกรอบทำนายวัตถุและกรอบวัตถุความจริงซ้อนทับตาม รูปที่ 2.5 สามารถตั้งค่า threshold ได้สำหรับค่า IoU เพื่อพิจารณาว่าการตรวจจับวัตถุนั้นถูกต้องหรือไม่ โดยสมมติว่าเราตั้ง

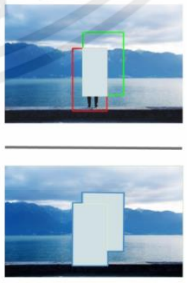
ค่า IoU เป็น 0.5 โดย

ถ้า IoU ≥ 0.5 ให้จัดประเภทการตรวจจับวัตถุเป็น True Positive (TP)

หาก IoU < 0.5 แสดงว่าเป็นการตรวจจับที่ผิดและจัดประเภทเป็น False Positive (FP)

เมื่อกรอบวัตถุความจริงปรากฏอยู่ในภาพและแบบจำลองไม่สามารถตรวจจับวัตถุได้ให้จัดประเภทเป็น False Negative (FN) True Negative (TN) คือทุกส่วนของภาพที่เราไม่ได้ทำนายวัตถุ โดยเมตริกนี้จะไม่มีผลต่อการตรวจจับวัตถุ ดังนั้น เราจึงไม่สนใจ True Negative ส่วนการตั้งค่า threshold ของ IoU เป็น 0.5 หรือสูงกว่า สามารถตั้งค่าเป็น 0.5, 0.75 0.9 หรือ 0.95 เป็นต้น การใช้ค่า Precision และค่า Recall ของเมตริกในการประเมินประสิทธิภาพ Precision และ Recall มาคำนวณโดยใช้ True Positive, False Positive และ False Negative

$$IoU_{pred}^{truth} = \frac{truth \cap pred}{truth \cup pred} =$$

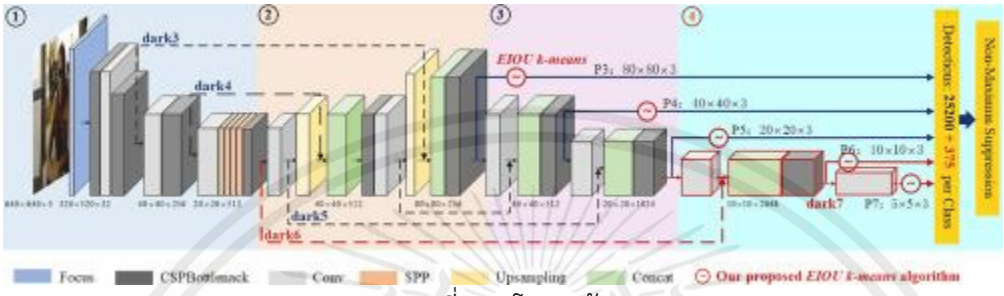


รูปที่ 2.6 วิธีการหาค□า IoU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 YOLOv2

YOLO (You Only Look Once) Version 2 เป็นอัลกอริธึมการรู้จำวัตถุแบบเรียลไทม์ที่มีประสิทธิภาพ ซึ่ง [2] จะกล่าวถึงแนวคิดของการตรวจจับวัตถุอัลกอริทึม โดย [3,4] อัลกอริธึม YOLO เป็น Object Detection Model ตัวหนึ่งที่มีความโดดเด่นอย่างมากในด้านของความเร็ว ตลอดจนการปรับปรุงบางอย่างในขั้นตอนการคาดคะเนกรอบขอบเขตที่แตกต่างกันเพื่อแยกคุณสมบัติ



รูปที่ 2.7 โครงสร้างของ YOLO

โดยจะมีอัลกอริทึมที่แตกต่างกันสำหรับการตรวจจับวัตถุ (Object Detection) และสามารถแบ่งออกเป็นสองกลุ่ม [5,6] อัลกอริทึมที่ใช้กับ Classification มีการดำเนินการในสองขั้นตอน ขั้นแรกเลือกภูมิภาคที่สนใจในรูปภาพ ต่อมาก็จำแนกภูมิภาคเหล่านี้โดยใช้โครงข่ายประสาทเทียม การแก้ปัญหานี้อาจช้าเนื่องจากต้องเรียกใช้การคาดการณ์สำหรับทุกภูมิภาคที่เลือก ตัวอย่างของอัลกอริทึมประเภทนี้คือ Region-based convolutional neural network (RCNN) อัลกอริทึมที่ใช้กับ Regression แทนที่จะเลือกส่วนที่น่าสนใจของรูปภาพ ก็จะทำนายคลาสและขอบเขตสำหรับรูปภาพทั้งหมดในการรันอัลกอริทึมเดียว ตัวอย่างที่ดีที่สุดจากอัลกอริทึมกลุ่มนี้คือ YOLO และ SSD (Single Shot Multibox Detector) มักใช้สำหรับการตรวจจับวัตถุแบบเรียลไทม์เนื่องจากโดยทั่วไปแล้วอาจต้องแลกกับการลดความแม่นยำลงเล็กน้อยเพื่อการปรับปรุงให้ความเร็วเพิ่มขึ้น

2.3 การใช้งานของระบบตรวจจับสัญญาณภาพเพลิงไหม้

2.3.1 ระบบตรวจจับไฟไหม้ (Early Fire Detection)

ระบบตรวจจับไฟไหม้ คือระบบที่นำข้อมูลภาพแวดล้อมที่กล้องจับได้ มาทำการวิเคราะห์ถึงความเปลี่ยนแปลงที่เกิดขึ้น (ความเปลี่ยนแปลงลักษณะเปลวไฟ) โดยจะทำการจับภาพและแจ้งเตือนถึงลักษณะการเปลี่ยนแปลงของพื้นที่ที่เสี่ยงต่อการเกิดอัคคีภัยให้ทราบ โดยระบบจะทำงานร่วมกับกล้อง IP Camera คุณภาพสูงในการตรวจจับการเคลื่อนไหวของภาพในลักษณะเปลวไฟเคลื่อนที่ในแนวตั้งแบบ Real Time และซอร์ฟแวร์สำหรับวิเคราะห์ข้อมูลการเปลี่ยนแปลงของภาพ (ตรวจจับเปลวไฟ) โดยระบบสามารถแจ้งเตือนการเกิดไฟไหม้แบบ Alarm ส่งเมล์แจ้งเตือน หรือประยุกต์ใช้งานร่วมกับซอร์ฟแวร์บริหารจัดการระบบ สำหรับใช้ในการแจ้งเตือนเพื่อป้องกันการเกิดเหตุอัคคีภัยตามอาคาร สถานที่ หรือโรงงานอุตสาหกรรม ฯลฯ [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.8 ระบบตรวจจับไฟไหม้ Early Fire Detection

2.3.2 การจับภาพเคลื่อนไหว (Motion Detection)

ปัจจุบันนี้เทคโนโลยีกล้องวงจรปิดได้พัฒนาก้าวไกลไปมากยิ่งขึ้น มีฟังก์ชันใหม่ ๆ เข้ามาเติมเต็มเพื่อตอบสนองความปลอดภัยให้กับสถานที่และบุคคลที่คุณรักอย่างรัดกุม โดยสิ่งที่จะต้องพูดถึงเลยก็คือ ระบบตรวจจับการเคลื่อนไหว หรือ Motion detection ซึ่งมีบทบาทและประโยชน์อย่างมาก สำหรับระบบรักษาความปลอดภัยกล้องวงจรปิด แต่จะมีประโยชน์และทำะไรได้บ้าง ในบทความนี้เราจะพาไปดูประโยชน์ของระบบจับการเคลื่อนไหวในกล้องวงจรปิดกัน [8]

ลักษณะการทำงานการจับภาพเคลื่อนไหว (Motion Detection)

การทำงานของระบบ Motion detection เกิดจากการนำซิพเซ็นเซอร์แบบพิเศษฝังลงไปในกล้องวงจรปิด ทำให้กล้องวงจรปิดสามารถจับการเคลื่อนไหวของวัตถุหรือสิ่งมีชีวิตที่อยู่ในเฟรมภาพได้ และเมื่อกล้องวงจรปิดจับการเคลื่อนไหวของวัตถุได้ กล้องวงจรปิดจะส่งสัญญาณไปยังระบบเพื่อประมวลผลทำตามฟังก์ชั่นที่เราตั้งไว้ในระบบ ซึ่งมีฟังก์ชั่นมากมาย

การตรวจจับการเคลื่อนไหวพร้อมแจ้งเตือนบนอุปกรณ์ไร้สาย (Mobile)

เป็นฟังก์ชั่นพื้นฐานของกล้องวงจรปิดที่มีระบบ Motion detectionหากผู้บุกรุกเข้ามายังพื้นที่ที่เราติดตั้งกล้องวงจรปิดไว้ แล้วถ้ากล้องวงจรปิดของเราตรวจจับได้ กล้องวงจรปิดจะส่งข้อความแจ้งเตือนส่งมาทาง Line, SMS, หรือ E-mail มาหาคุณเพื่อให้ทราบ และแจ้งความกับเจ้าหน้าที่ตำรวจได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ศึกษาข้อมูลของระบบคลาวด์ ฐานข้อมูล และเว็บไซต์

2.4.1 คลาวด์คอมพิวติ้ง (Cloud Computing)

กระบวนทัศน์การประมวลผลแบบคลาวด์เป็นกรอบการทำงานใหม่สำหรับการจัดซื้อคอมพิวเตอร์ในฐานะยูทิลิตี้ (Utility Computing) แทนที่จะใช้ศูนย์ข้อมูลแบบเดิมเพื่อให้มีความสามารถในการประมวลผลข้อมูล บางทีภาพรวมที่ดีที่สุดของตัวขับเคลื่อนเทคโนโลยี การประมวลผลแบบคลาวด์ในรายงาน "Above the Clouds" [9, 10] จาก Berkeley ซึ่งกล่าวถึงอุปสรรคและโอกาสในการประมวลผลแบบคลาวด์ไว้ว่า

การประมวลผลแบบคลาวด์ประกอบด้วยเทคโนโลยีจำนวนหนึ่งที่จำเป็นเพื่อให้เป็นไปตามสัญญา "การคำนวณเป็นยูทิลิตี้ (Computing as utility)" ที่ทำโดยผู้จำหน่ายคลาวด์คอมพิวติ้ง เทคโนโลยีหลักหลายอย่างมีอยู่แล้วก่อนที่จะมีการประดิษฐ์คำว่า "การประมวลผลแบบคลาวด์ (Cloud computing)" ในขณะที่เทคโนโลยีอื่น ๆ ถือกำเนิดมาจากการใช้คอมพิวเตอร์ในระดับอินเทอร์เน็ตในศูนย์ข้อมูลแบบกระจายโดยบริษัทขนาดใหญ่หลายแห่ง เช่น Google, Amazon, Alibaba จึงมีคำจำกัดความมากมายของการประมวลผลแบบคลาวด์ โดยคุณสมบัติหลักที่แตกต่างของการประมวลผลแบบคลาวด์ ได้แก่

1) ทรัพยากรคอมพิวเตอร์สามารถซื้อได้ตามต้องการจากอุปทานที่ไม่จำกัด
2) ค่าใช้จ่ายด้านทุนที่จำเป็นในการจัดซื้อทรัพยากรสามารถคำนวณล่วงหน้าค่าใช้จ่ายในการดำเนินงานได้ ซึ่งจะทำให้ความเสี่ยงในการลงทุนลดลงสำหรับผู้ให้บริการคลาวด์คอมพิวติ้งที่ไม่เพียงพอ
3) คอมพิวเตอร์คิดราคาด้วยรูปแบบการกำหนดราคาแบบจ่ายตามการใช้งาน ซึ่งความจุสามารถปรับขนาดขึ้นและลงได้ในระยะสั้น

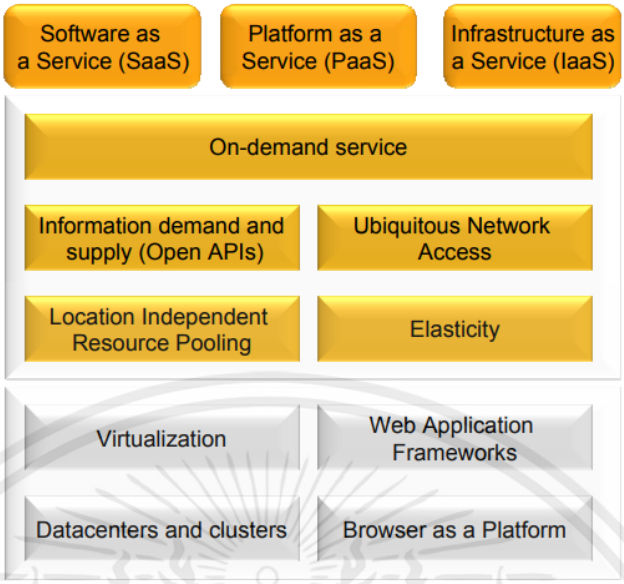
สถาบันมาตรฐานแห่งชาติ (NIST) สหรัฐอเมริกาได้ตัดสินใจที่จะเน้นย้ำถึงความยืดหยุ่นของทรัพยากรคอมพิวเตอร์ในคำจำกัดความของการประมวลผลแบบคลาวด์ [11] ซึ่งเป็นคำจำกัดความที่เราเห็นด้วยเป็นส่วนใหญ่ บทความหนึ่งที่อ้างถึงบ่อยให้คำจำกัดความเกี่ยวกับระบบคลาวด์ ดังนี้

คลาวด์เป็นแหล่งทรัพยากรเสมือนจริงขนาดใหญ่ที่ใช้งานง่ายและเข้าถึงได้ (เช่น ฮาร์ดแวร์ แพลตฟอร์มการพัฒนา และ/หรือบริการ) ทรัพยากรเหล่านี้สามารถกำหนดค่าใหม่แบบไดนามิกเพื่อปรับให้เข้ากับโหลดตัวแปร (สเกล) ซึ่งช่วยให้สามารถใช้ทรัพยากรได้อย่างเหมาะสมที่สุด แหล่งทรัพยากรนี้มักถูกเอารัดเอาเปรียบโดยรูปแบบการจ่ายต่อการใช้ ซึ่งผู้ให้บริการโครงสร้างพื้นฐานจะดำเนินการค้ำประกันโดยวิธีการของ SLA ที่ปรับแต่งเอง [12]

หนึ่งในปัจจัยขับเคลื่อนหลักสำหรับการประมวลผลแบบคลาวด์คือความประหยัดของขนาดในการสร้างศูนย์ข้อมูลและค่าใช้จ่ายในการดำเนินงาน ราคาของฮาร์ดแวร์ ไฟฟ้า การระบายความร้อน และโดยเฉพาะอย่างยิ่งความสามารถของเครือข่ายทั่วโลกนั้นสามารถแข่งขันได้มากขึ้น

เมื่อซื้อสำหรับศูนย์ข้อมูลที่มีเซิร์ฟเวอร์นับหมื่นหรือหลายแสนเครื่อง แทนที่จะเป็นการดำเนินงานศูนย์ข้อมูลขนาดเล็กที่มีเซิร์ฟเวอร์นับร้อยถึงหนึ่งพันเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ภาพรวมการให้บริการคลาวด์ (cloud computing)

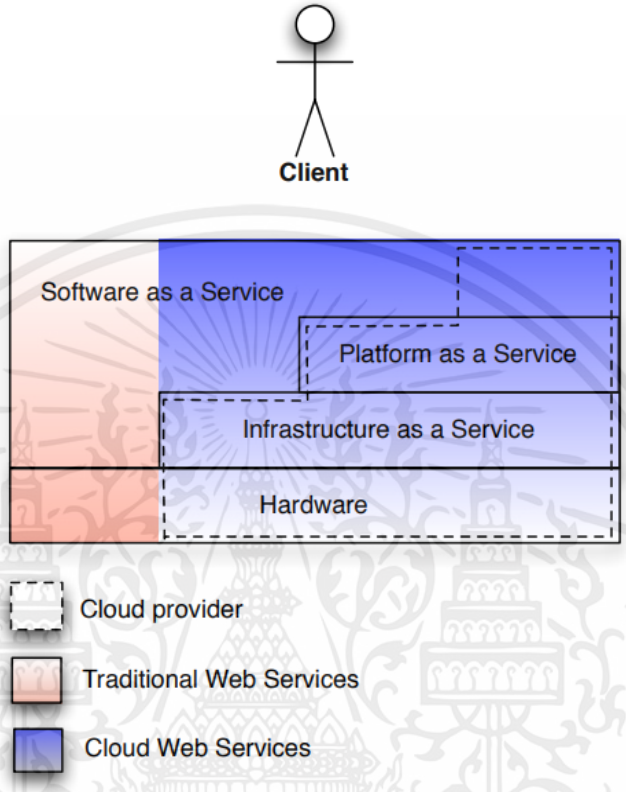
ตัวเลขดังกล่าวเน้นให้เห็นถึงบทบาทที่แตกต่างกันที่พบในคลาวด์คอมพิวติ้ง ได้แก่ ผู้ใช้บริการ ผู้ให้บริการ และนักพัฒนา ผู้ให้บริการเป็นองค์ประกอบสำคัญของคลาวด์คอมพิวติ้ง เนื่องจากอำนวยความสะดวกในการปรับใช้และการดำเนินการของหน่วยการสร้างต่างๆ ผู้ให้บริการได้รับความยืดหยุ่นผ่านการจำลองเสมือนและการกำหนดค่าแบบไดนามิกของระบบรันไทม์ ความยืดหยุ่นที่คำนึงถึงอุปสงค์และอุปทานของเนื้อหานี้ถือได้ว่าเป็นคุณลักษณะสำคัญของระบบคลาวด์ ทรัพยากรเสมือนจริงรวมถึงเครือข่าย ซีพียู และพื้นที่เก็บข้อมูล ในการปรับใช้และจัดการแพลตฟอร์มคลาวด์ จำเป็นต้องมีคุณสมบัติการจัดการจำนวนหนึ่ง คุณลักษณะการจัดการที่จำเป็นได้แก่ การรายงาน ข้อตกลงระดับการให้บริการ การวางแผนกำลังการผลิต และการเรียกเก็บเงิน เลเยอร์ซอฟต์แวร์ที่ให้บริการการจำลองเสมือนเรียกว่าการตรวจสอบเครื่องเสมือนหรือไฮเปอร์ไวเซอร์ ไฮเปอร์ไวเซอร์สามารถทำงานบนฮาร์ดแวร์เปล่า (ประเภท 1 หรือ VM ดั้งเดิม) หรือบนระบบปฏิบัติการ (ประเภท 2 หรือ VM ที่โฮสต์) นักพัฒนาบริการใช้ API ที่เปิดเผยโดยแพลตฟอร์มคลาวด์และซอฟต์แวร์ที่ดำเนินการอยู่ ผู้พัฒนาบริการจำเป็นต้องมีเครื่องมือสำหรับการสร้างบริการ การปรับใช้และการเผยแพร่ และการวิเคราะห์บริการขณะใช้งานจริง ผู้ใช้บริการใช้ API และอินเทอร์เฟซผู้ใช้ที่หลากหลายเพื่อเข้าถึงบริการที่ปรับใช้

บริการของ Cloud Computing สามารถแบ่งออกเป็น 3 ประเภท ดังนี้

1) Software-as-a-Service (SaaS) ซึ่งผู้จำหน่ายจัดหาโครงสร้างพื้นฐานด้านฮาร์ดแวร์ ผลิตภัณฑ์ซอฟต์แวร์ และโต้ตอบกับผู้ใช้โดยใช้พอร์ทัล
2) Platform-as-a-Service (PaaS) ซึ่งผู้ให้บริการโฮสต์ชุดซอฟต์แวร์และเครื่องมือการพัฒนาบนโครงสร้างพื้นฐานของผู้ให้บริการ เช่น AppEngine ของ Google

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Infrastructure-as-a Service (IaaS) ซึ่งเกี่ยวข้องกับอินสแตนซ์เซิร์ฟเวอร์เสมือนที่มีที่อยู่ IP ที่ไม่ซ้ำกันและบล็อกของพื้นที่จัดเก็บตามต้องการ เช่น โครงสร้างพื้นฐานบริการเว็บของ Amazon รูปที่ 2.8 แสดงสถาปัตยกรรมเลเยอร์ของคลาวด์คอมพิวติ้ง



รูปที่ 2.10 สถาปัตยกรรมของซอฟต์แวร์คลาวด์

2.4.1.1. เทคโนโลยีเบื้องหลัง (Background Technologies)

เทคโนโลยีที่ใช้กันทั่วไปบางอย่างที่มีอยู่แล้วก่อนที่จะมีการเปิดตัวคลาวด์คอมพิวติ้ง สำหรับคลาวด์คอมพิวติ้งที่สำคัญมี ดังนี้

1) เทคโนโลยีฮาร์ดแวร์เสมือนจริง (Cloud Hardware Virtualization Technologies)

เสมือนจริง (Virtualization) หรือความสามารถในการทำให้คอมพิวเตอร์เครื่องหนึ่งปรากฏเป็นคอมพิวเตอร์หลายเครื่องหรือคอมพิวเตอร์ที่แตกต่างกันโดยสิ้นเชิง เป็นแนวคิดที่มีอายุ 4 ทศวรรษที่ IBM นำมาใช้ในคอมพิวเตอร์ 7044 ร่วมกับ Compatible Time Sharing System (CTSS) ที่พัฒนาโดย MIT

การจำลองเสมือนเป็นองค์ประกอบหลักในการประมวลผลแบบคลาวด์ เนื่องจากช่วยให้แยกแยะฮาร์ดแวร์พื้นฐานออกจากระบบปฏิบัติการได้ และช่วยให้ผู้ให้บริการฮาร์ดแวร์ระบบคลาวด์อนุญาตให้ไคลเอ็นต์เรียกใช้ระบบปฏิบัติการใดๆ ที่จำเป็นได้อย่างง่ายดาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงาน Secure Virtualization และ Multicore Platforms State-of-theArt โดย Heradon Douglas และ Christian Gehrmann [20] ให้ภาพรวมที่ดีของเทคนิคและปัญหาพื้นฐาน เทคนิคการจำลองเสมือนสามารถแบ่งออกเป็นสองวิธีหลัก ได้แก่

- System virtualization ซึ่งทั้งระบบถูกจำลองเสมือน ซึ่งช่วยให้ระบบเสมือนหลายระบบทำงานพร้อมกันโดยแยกจากกันโดยสิ้นเชิง ไฮเปอร์ไวเซอร์หรือมอนิเตอร์เครื่องเสมือนให้การเข้าถึงหน่วยความจำ อุปกรณ์ เครือข่าย รวมถึง CPU ด้วยเหตุนี้ ระบบปฏิบัติการ Guest จึงคิดว่ามีเครื่องสำหรับตัวเอง
- Para-virtualization ซึ่งระบบปฏิบัติการของแขกได้รับการแก้ไขเพื่อให้ความร่วมมือกับไฮเปอร์ไวเซอร์ แขกได้รับการแก้ไขให้ใช้อินเทอร์เฟซที่ปลอดภัยกว่าหรือมีประสิทธิภาพมากกว่าอินเทอร์เฟซระบบปฏิบัติการแขกดั้งเดิม

2) พื้นที่เก็บข้อมูลบล็อกเครือข่ายเสมือนจริง (Virtualized Network Block Storage)

เป้าหมายของการจำลองเสมือนการจัดเก็บข้อมูลคือการสรุปตำแหน่งทางกายภาพของข้อมูลจากผู้ใช้และนักพัฒนา หากมีการนำไปใช้อย่างเพียงพอ จะนำไปสู่ความเป็นอิสระของสถานที่ บทบาทของการจัดเก็บข้อมูลเสมือนจริงคือการจัดทำแผนที่จากข้อมูลที่รับรู้ไปยังตำแหน่งจริง

เทคโนโลยีเหล่านี้มีบทบาทสำคัญในแนวทางปัจจุบันของคลาวด์คอมพิวติ้ง ซึ่งได้ตัดสินใจแยกกันต่างหาก ด้วยเหตุผลที่ชัดเจน รูปแบบที่แท้จริงของการทำแผนที่จะขึ้นอยู่กับการใช้งาน ตัวอย่างเช่น อาจมีข้อจำกัดเกี่ยวกับความละเอียดของการแม็พ โดยที่การใช้งานที่แตกต่างกันจะกำหนดมาตราส่วนจากดิสก์เดี่ยวที่มีอยู่จริงเต็มรูปแบบซึ่งอยู่ในคอมพิวเตอร์เครื่องหนึ่งไปจนถึงชุดย่อยเล็กๆ ของดิสก์ ซึ่งมีให้ในเช่น เมกะไบต์หรือกิกะไบต์

การใช้งานที่มีให้ใช้งานทั่วไปช่วยให้สามารถจัดการระบบจัดเก็บข้อมูลของผู้จำหน่ายหลายรายที่แตกต่างกันได้ ดังนั้นจึงเป็นไปได้ที่จะสร้างระบบเสมือนจริงจากระบบย่อยส่วนประกอบที่เหมาะสมที่สุด ซึ่งอาจให้คุณภาพการบริการที่แตกต่างกันในแง่ของความเร็วในการเข้าถึง

ประโยชน์ของระบบจัดเก็บข้อมูลเสมือนจริงโดยทั่วไปมีมากมาย ซึ่งรวมถึงอย่างน้อยดังต่อไปนี้

- การโยกย้ายข้อมูลแบบไม่รบกวน ด้วยระบบเวอร์ชวลไลซ์ ข้อมูลสามารถโยกย้ายไปยังตำแหน่งต่างๆ ได้ แม้ว่าจะมีการใช้งานอยู่ก็ตาม จึงมีอิสระมากขึ้นในการจัดระเบียบข้อมูลในเครือข่ายตามบริการที่จัดให้และคอมพิวเตอร์และการจัดเก็บข้อมูลที่มีอยู่ในปัจจุบัน
- ปรับปรุงการใช้ประโยชน์ เช่นเดียวกับการประมวลผลแบบคลาวด์ ข้อดีอย่างหนึ่งของการใช้ที่จัดเก็บข้อมูลเสมือนจริงคือความสามารถในการใช้ทรัพยากรที่มีอยู่ในรูปแบบที่เหมาะสมที่สุด
- การจัดการแบบง่าย ประโยชน์ที่ได้รับจากคลาวด์คอมพิวติ้งทั่วไปก็คือความต้องการจุดการจัดการที่น้อยลงเมื่ออาศัยการจัดเก็บข้อมูลเสมือนจริง ซึ่งจะทำให้การจัดการง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) แซนบ็อกซิ่ง (Sandboxing)

แซนด์บ็อกซ์เป็นกลไกความปลอดภัยที่ใช้กันทั่วไปซึ่งแยกโปรแกรมและทรัพยากรออกจากกัน การรวมแอปพลิเคชันต่างๆ ไว้ในแซนด์บ็อกซ์ที่แยกจากกัน ทำให้โครงสร้างพื้นฐานที่ใช้โฮสต์แอปพลิเคชันเหล่านี้สามารถแชร์กับแอปพลิเคชันจำนวนมากได้ ซึ่งบางแอปพลิเคชันอาจมีระดับความเชื่อถือที่แตกต่างกัน ยิ่งไปกว่านั้น มันง่ายที่จะใช้ซอฟต์แวร์ทดลองในโครงสร้างพื้นฐานเดียวกันกับซอฟต์แวร์ที่ใช้งานจริง เนื่องจากเมื่อรวมระบบต่าง ๆ ลงในแซนด์บ็อกซ์ของพวกเขาเอง ระบบจะไม่ก่อให้เกิดอันตรายซึ่งกันและกัน

ตารางที่ 2.1 การเปรียบเทียบไฮเปอร์ไวเซอร์ (hypervisors)

	Xen	KVM	VMWare	Hyper-V
Open source	Yes	Yes	No	No
Type 1/2	1.5 (Domo privileged guest)	1.5 (Linux kernel, Qemu)	2	2
Hardware	x86 / x86_64	x86 / x86_64	x86 / x86_64	x86_64
Virtual hardware	Qemu: x86 / x86_64	Qemu: x86 / x86_64	x86 / x86_64	x86 / x86_64
Features	Nested virtualization, live migration	Nested virtualization, live migration		
Association	Citrix	Red Hat / Intel	VmWare	VmWare

ในรูปแบบที่ง่ายที่สุด ระบบแซนด์บ็อกซ์กำลังแยกแอปพลิเคชันออกจากกันและระบบปฏิบัติการโฮสติ้งทั้งหมด พวกเขาไม่มีทางโต้ตอบ ยกเว้นโดยอ้อมในแง่ของเวลาของตัวประมวลผล ซึ่งพวกเขาต้องแบ่งปัน โดยมีเงื่อนไขว่าใช้โครงสร้างพื้นฐานการคำนวณเดียวกัน อย่างไรก็ตาม มันก็เป็นเรื่องธรรมดาเช่นกันที่ทุกอย่างไม่ได้ถูกแยกออกจากกันในระดับนี้ ให้หลักฐานที่สมเหตุสมผลว่าผู้พัฒนาได้รับอนุญาตให้ใช้บริการบางอย่าง ระบบแซนด์บ็อกซ์แบบละเอียดดังกล่าวสามารถนำไปใช้ได้โดยใช้ความสามารถที่เรียกว่า ซึ่งสามารถใช้เพื่อจัดเตรียมการเข้าถึงทรัพยากรที่แตกต่างกันตามคำจำกัดความที่ละเอียดมากขึ้น

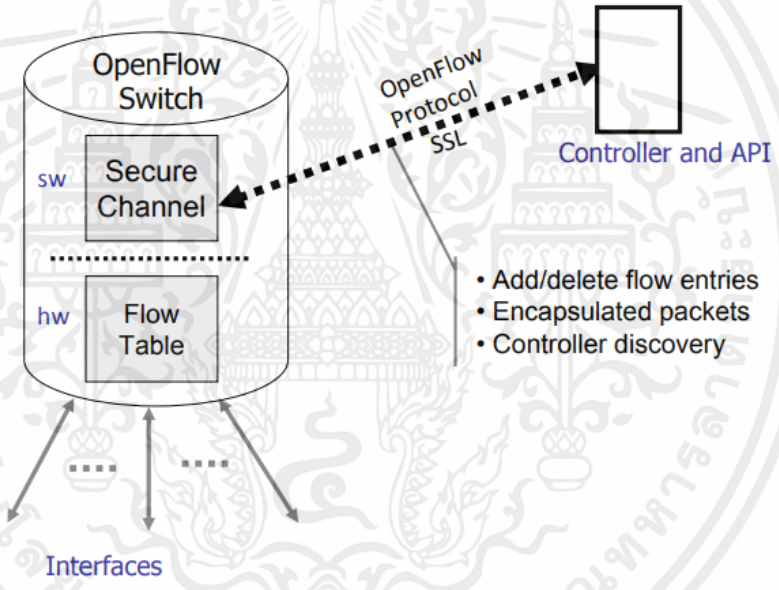
เนื่องจากแซนด์บ็อกซ์ได้รับการพิสูจน์ว่าเป็นเทคโนโลยีที่มีประโยชน์จริงๆ ในด้านการคำนวณหลายด้าน จึงไม่ใช่เรื่องแปลกที่จะพบการใช้งานที่แตกต่างกันซึ่งมุ่งสู่การใช้งานเฉพาะบางด้าน ในระบบคลาวด์คอมพิวติ้ง การใช้แซนด์บ็อกซ์ที่พบบ่อยที่สุดคือร่วมกับระบบเวอร์ชวลไลเซชัน โดยเป้าหมายของแซนด์บ็อกซ์คือการจัดเตรียมภาพลวงตาของคอมพิวเตอร์เครื่องเดียวที่อุทิศให้กับนักพัฒนา ซึ่งแยกออกจากส่วนที่เหลือของแอปพลิเคชันที่ทำงานอยู่ ในเซิร์ฟเวอร์ฟาร์มหรือศูนย์ข้อมูลเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การจำลองเสมือนเครือข่าย (Network Virtualization)

OpenFlow เป็นมาตรฐานเปิดที่อนุญาตให้เรียกใช้โปรโตคอลทดลองในเครือข่ายการผลิตได้ [13] OpenFlow เป็นคุณสมบัติที่เพิ่มให้กับสวิตช์ เราเตอร์ จุดเข้าใช้งาน (AP) และสถานีฐาน ทำให้สามารถควบคุมอุปกรณ์ Datapath เหล่านี้ผ่าน API ภายนอกที่ได้มาตรฐาน ผู้จำหน่ายสวิตช์รายใหญ่กำลังใช้ระบบนี้และมหาวิทยาลัยใช้เพื่อปรับใช้เทคโนโลยีเครือข่ายที่เป็นนวัตกรรมใหม่ โดยทั่วไป OpenFlow เป็นสวิตช์อีเธอร์เน็ตที่กำหนดโดยซอฟต์แวร์ เครือข่ายที่กำหนดโดยซอฟต์แวร์คาดว่าจะเป็นหนึ่งในหัวข้อการวิจัยใหม่ในเครือข่ายคอมพิวเตอร์

รูปที่ 5 แสดงภาพรวมของโปรโตคอล OpenFlow เราเตอร์และสวิตช์ใช้ API แบบเปิดที่ช่วยให้ผู้ดูแลระบบและส่วนประกอบควบคุมสร้าง แก้ไข และลบโฟลว์ในตารางโฟลว์ โปรโตคอลนี้เป็นขั้นตอนสู่เครือข่ายที่กำหนดโดยซอฟต์แวร์



รูปที่ 2.11 โปรโตคอล OpenFlow

NOX เป็นแพลตฟอร์มควบคุมเครือข่ายโอเพ่นซอร์สที่สามารถควบคุมการเชื่อมต่อทั้งหมดบนเครือข่าย รวมถึงการส่งต่อ การเราต์ ซึ่งโฮสต์และผู้ใช้ได้รับอนุญาต NOX เป็นองค์ประกอบเครื่องบินควบคุมที่สามารถใช้กับสวิตช์ OpenFlow [14]

2.4.2. ฐานข้อมูลระบบคลาวด์ (Cloud Database)

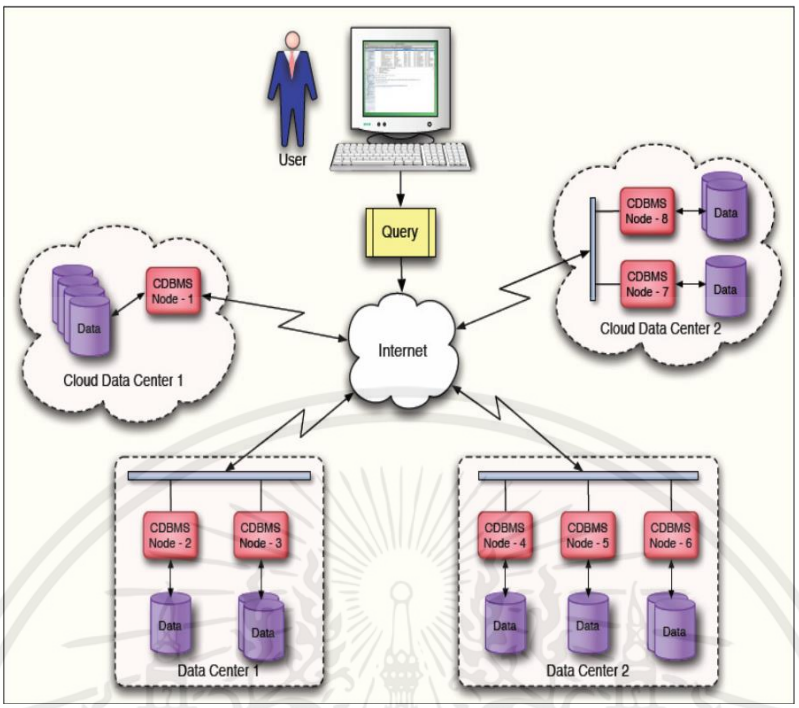
ฐานข้อมูลบนคลาวด์เก็บข้อมูลในศูนย์ข้อมูลต่าง ๆ ซึ่งตั้งอยู่ในสถานที่ต่างกัน ทำให้โครงสร้างฐานข้อมูลบนคลาวด์แตกต่างจากระบบการจัดการฐานข้อมูลที่มีเหตุผล ทำให้โครงสร้างของฐานข้อมูลบนคลาวด์มีความซับซ้อน มีหลายโหนดในฐานข้อมูลบนคลาวด์ ซึ่งออกแบบมาสำหรับบริการสืบค้นข้อมูล สำหรับศูนย์ข้อมูลที่ตั้งอยู่ในตำแหน่งทางธรณีวิทยาต่างๆ และศูนย์ข้อมูลขององค์กรด้วย การเชื่อมโยงนี้เป็นข้อบังคับสำหรับการเข้าถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลที่ง่ายดายและสมบูรณ์ผ่านบริการคลาวด์ มีหลายวิธีในการเข้าถึงฐานข้อมูลผ่านบริการคลาวด์ ผู้ใช้สามารถเข้าถึงได้ผ่านคอมพิวเตอร์ผ่านอินเทอร์เน็ต หรือผู้ใช้ที่ใช้โทรศัพท์มือถือสามารถเข้าถึงฐานข้อมูลบนคลาวด์ผ่านบริการ 3G หรือ 4G (Pizzete และ Cabot 2012) เพื่อให้เข้าใจโครงสร้างของฐานข้อมูลบนคลาวด์ได้ดียิ่งขึ้น เราจะสาธิตตัวอย่างแอปพลิเคชัน Business Intelligence แอปพลิเคชัน BI ใช้สำหรับจัดเก็บข้อมูลขนาดใหญ่ เนื่องจากองค์กรใช้เพื่อจัดเก็บข้อมูลสำหรับลูกค้าของตน ที่นี่เราคิดว่าผู้ใช้กำลังเข้าถึงฐานข้อมูลบนคลาวด์จากคอมพิวเตอร์ผ่านอินเทอร์เน็ต อินเทอร์เน็ตเป็นจุดเชื่อมต่อ ที่ทำหน้าที่เป็นสะพานเชื่อมระหว่างศูนย์ข้อมูล ศูนย์ข้อมูลบนคลาวด์ และผู้ใช้ที่กำลังเข้าถึงข้อมูล เป็นสิ่งสำคัญที่จะต้องทราบที่นี่ว่ามีเพียงโหนดเดียวเท่านั้นที่ไม่ได้ใช้ในฐานข้อมูลบนคลาวด์ อย่างไรก็ตาม มีโหนดต่างๆ ที่ใช้สำหรับฐานข้อมูลบนคลาวด์ (Curino, Madden และ et.al.) เพื่อจุดประสงค์นี้ ควรใช้การสื่อสารแบบเพียร์ทูเพียร์ จุดประสงค์ในการนำการสื่อสารแบบเพียร์ทูเพียร์มาใช้คือ โหนดเดียวสามารถจัดการกับแบบสอบถามประเภทใดก็ได้ที่ผู้ใช้นำไปใช้ ดูเหมือนว่าจะซับซ้อน แต่เป็นวิธีแก้ปัญหาที่ง่ายสำหรับระบบโหนดประเภทนี้ ซึ่งแต่ละโหนดในฐานข้อมูลบนคลาวด์มีการแมปไปยังข้อมูลที่จัดเก็บไว้ในแต่ละโหนด แผนที่นี้ไปยังข้อมูลที่จัดเก็บไว้ช่วยให้เข้าถึงข้อมูลสำหรับการสืบค้นข้อมูลที่เฉพาะเจาะจงได้ง่าย

เมื่อการสืบค้นข้อมูลถูกสร้างขึ้นจากผู้ใช้ผ่านทางคอมพิวเตอร์ อันดับแรก โหนดจะตัดสินใจประเภทของการสืบค้น และโหนดใดจะดีที่สุดสำหรับการสืบค้น หลังจากที่โหนดระบุเคียวรีแล้ว เคียวรีจะถูกโอนไปยังโหนดเฉพาะนั้น จากนั้นโหนดเฉพาะจะดูแลแบบสอบถามและตอบสนองต่อผู้ใช้ ตัวอย่างเช่น เมื่อได้รับข้อความค้นหาอาจถูกส่งไปยังโหนด 1 ก่อน จากนั้นโหนด 1 จะระบุว่าโหนดใดจะแก้ปัญหาการสืบค้นที่เหมาะสม อาจเป็นโหนด 7 เก็บข้อมูลโหนด 1 จะส่งแบบสอบถามไปยังโหนด 7 หลังจากตรวจสอบแผนที่ข้อมูล เมื่อการสืบค้นข้อมูลถูกส่งไปยังการสืบค้นเฉพาะ ข้อมูลจะถูกส่งไปยังผู้ใช้โดยตรงโดยไม่ชักช้า รูปด้านล่างแสดงสถาปัตยกรรมพื้นฐานของฐานข้อมูลบนคลาวด์ หรือจะถือเป็นภาพรวมก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

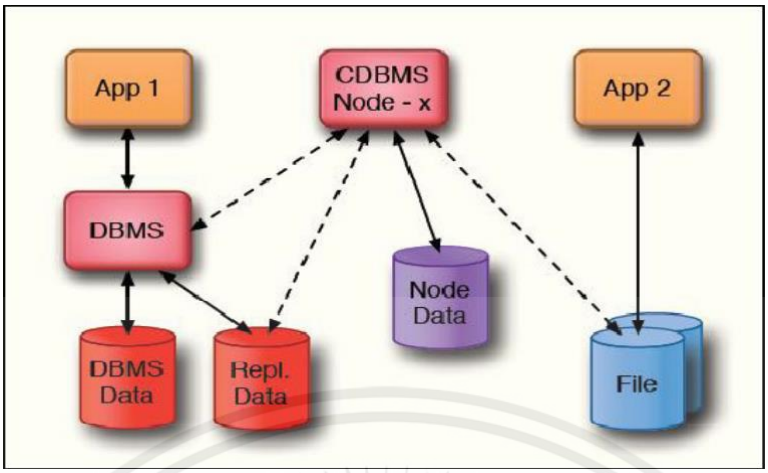


รูปที่ 2.12 โครงสร้างของคลาวด์

2.4.2.1. การทำงานของโหนด (Working of Nodes)

ในส่วนนี้เราจะพูดถึงการทำงานของโหนดในระบบการจัดการฐานข้อมูลบนคลาวด์ ขั้นตอนการทำงานของโหนดใน CDBMS สำหรับการเข้าถึงข้อมูลที่จัดเก็บไว้ในฐานข้อมูลนั้น คล้ายกับเมื่อมีการสอบถามไปยังโหนด โหนดที่พร้อมใช้งานจะมี 2 ทางเลือก ไม่ว่าจะเป็นการเข้าถึงข้อมูลโดยตรงจากฐานข้อมูลหรือโดยการรับ จากการจำลองฐานข้อมูล ฐานข้อมูลที่จำลองแบบไม่มีการเข้าถึงตลอดเวลา เนื่องจากมีไว้เพื่อวัตถุประสงค์ฉุกเฉิน เมื่อฐานข้อมูลไม่สามารถดำเนินการได้ (Bloor, 2011) โหนดส่วนใหญ่รับข้อมูลจากฐานข้อมูลเนื่องจากเพิ่มประสิทธิภาพในการดึงข้อมูล ใน CDBMS; ข้อมูลแอปพลิเคชันถูกเก็บไว้ในแอปพลิเคชัน CDBMS เข้าถึงข้อมูลจากไฟล์โดยตรง หากโหนดเข้าถึงข้อมูลโดยตรง โหนดจะเก็บแมปข้อมูลเมตาของไฟล์ที่รับข้อมูลแอปพลิเคชัน รูปต่อไปนี้แสดงการทำงานของโหนดในระบบจัดการฐานข้อมูลบนคลาวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การทำงานของโหนด

รูปด้านบนแสดงการทำงานของโหนดสำหรับการดึงข้อมูลจากข้อมูลและไฟล์ DBMS นอกจากนี้ CDBMS จะรักษาฐานข้อมูลเพื่อจัดเก็บข้อมูลที่โหนดใช้บ่อย ๆ สิ่งนี้ช่วยปรับปรุงประสิทธิภาพของ CDBMS

2.4.2.2. ความสมบูรณ์ของข้อมูล ความปลอดภัย และตำแหน่งการจัดเก็บ (Data Integrity, Security and Storage Location)

ความปลอดภัยเป็นภัยคุกคามที่สำคัญต่อข้อมูลที่จัดเก็บไว้ในที่เก็บข้อมูลบนคลาวด์ การรักษาความปลอดภัยยังขึ้นอยู่กับวิธีการเข้ารหัสที่ใช้และตำแหน่งการจัดเก็บข้อมูล (Hacigumus, Iyer และ Mehrotra 2004) ข้อมูลจะถูกเก็บไว้ในตำแหน่งต่าง ๆ ในศูนย์ข้อมูล

	Amazon RDS (MySQL)	Microsoft SQL Azure	Google Datastore	Amazon SimpleDB
Type	RDBMS	RDBMS	NoSQL	NoSQL
Maximum amount of data that can be stored	1 terabyte per database[2]	50 gigabytes per database[3]	Not published for entire database, but 1 MB limit on a subset of data (called an entity). Limit to the number of indexes.	10 gigabytes per database domain (roughly equivalent to an RDBMS table)[4]
Ease of software portability with similar, locally hosted capability	High. MySQL instantiation in cloud is very similar to the local instantiated version.	High. Most SQL Server features are available in SQL Azure.	Medium/Low. Requires Java Data Objects or Datastore-specific interface and use of App Engine.	Medium. Requires SimpleDB-specific interface.
Transaction capabilities	Yes	Yes	Yes	Yes
Configurability and ability to tune database	High. MySQL instantiation in cloud.	Medium. Can create indexes and stored procedures, but no control over memory allocation or similar resources.	Low	Low
Database accessible as "stand-alone" offering	Yes	Yes	No. Requires Google App Engine application layer.	Yes
FISMA Certified	No	No	No	No
Can designate where data is stored (e.g., region or data center)	Yes	Yes	No	Yes
Replication	Yes	Yes	Yes	Yes

รูปที่ 2.14 เปรียบเทียบความสมบูรณ์ของข้อมูล ความปลอดภัย และตำแหน่งการจัดเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

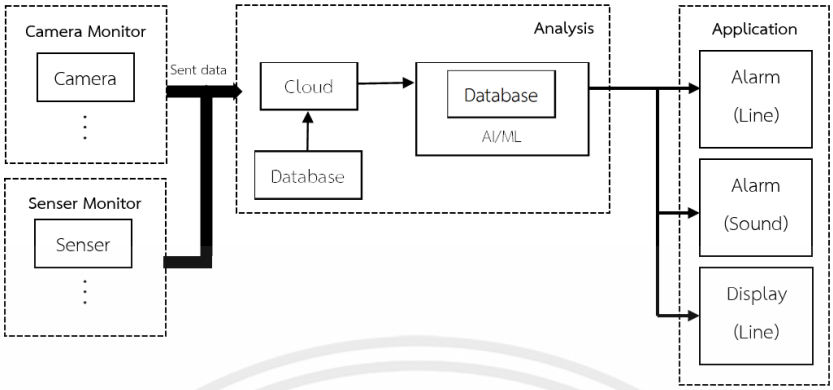
3.1.1 การออกแบบการทำงานของระบบ

การออกแบบและการทำงานของระบบในส่วนนี้จะอธิบายโมเดลการเรียนรู้เชิงลึกที่ใช้สำหรับการตรวจจับเพลิงไหม้ ชุดข้อมูลที่ใช้สำหรับการ Train และตัวชี้วัดการประเมินที่ใช้ในการศึกษานี้ ขั้นตอนการทำงานและการดำเนินการของระบบจะถูกแบ่งออกเป็นส่วนต่างๆ ดังรูปที่ 3.1 และมีกระบวนการทำงานดังนี้

1) ออกแบบโครงสร้างเครื่องตรวจจับไฟไหม้โดยการใช้บอร์ดคอนโทรลเลอร์ Corgi Dude เพื่อทำการเขียนคำสั่งใช้งานลงบนบอร์ดโดยใช้งานร่วมกับโมดูลกล้องและทำการส่งข้อมูลไปยังหน้าจอแสดงภาพ
2) การนำเข้าชุดข้อมูลโดยการทำการอัดคลิปวิดีโอ และทำการแปลงเป็นรูปภาพรูปภาพที่มีควันไฟและไฟไหม้ รวมถึงกรองภาพที่มีความละเอียดต่ำออก หลังจากได้ภาพแล้วจากนั้นทำการเตรียมการกำหนดผลลัพธ์ในแต่ละรูป (Data Labeling) สำหรับการทำ Image Detection
3) นำเข้าข้อมูลจากชุดข้อมูลมาจัดเตรียมสำหรับโมเดลการตรวจจับไฟไหม้ โดยการทำ Image Detection โดยใช้วิธี YOLO โดยที่ YOLO คือสถาปัตยกรรมที่ทาง Ultralytics ได้ออกแบบไว้เพื่อทำ Image Detection ได้อย่างรวดเร็วและมีประสิทธิภาพ
4) ทำการ Train โมเดลสำหรับการทำ image Detection นำข้อมูลจากไฟล์ภาพข้อมูลมาทำการสร้างแบบจำลองตรวจจับไฟไหม้ มาประมวลผลและทดสอบร่วมกับข้อมูลชุดรูปภาพที่มีอยู่
5) นำโมเดลการตรวจจับไฟไหม้ที่ทำการสร้างขึ้นจาก YOLO โดยการใช้บอร์ด และทำการ Detect ภาพไปยังจอแสดงผล โดยนำโมเดลที่ได้ในการเทรนทดสอบการตรวจจับไฟไหม้เพื่อทดสอบความแม่นยำของอัลกอริทึม
6) ทำการส่งข้อมูลแสดงผลและทำการแจ้งเตือนไปยังแอพพลิเคชันไลน์

ในการศึกษานี้ ได้ทำการ Train โมเดลเพื่อตรวจจับไฟไหม้โดยใช้แนวทางการเรียนรู้การถ่ายโอนเพื่อเพิ่มความแม่นยำของโมเดล โมเดลการตรวจจับไฟไหม้ได้มาจากการ Train ชุดข้อมูลควันไฟและไฟไหม้ จากนั้นใช้โมเดลนั้นในการตรวจจับ รูปภาพหรือคลิปวิดีโอรวมไปถึงภาพจากกล้อง โดยนำโมเดลมาทำการเฝ้าระวังเพลิงไหม้ไปยังแอพพลิเคชันต่างๆ

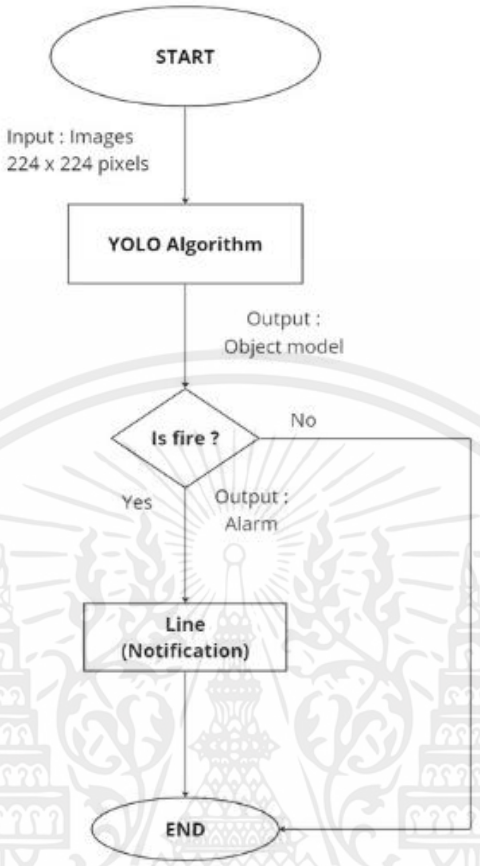
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แผนภาพบล็อกการทำงานของระบบเฝ้าระวังเพลิงไหม้

ขั้นตอนการทำการงานของระบบตรวจจับไฟไหม้ด้วย YOLO Object Detection ผ่านเครื่องตรวจจับเพลิงไหม้ เมื่อระบบเริ่มทำงาน ระบบจะทำการนำไฟล์รูปภาพที่ได้เตรียมไว้ จากนั้นจะนำภาพที่ได้เข้าอัลกอริทึม YOLOv2 เพื่อทำการ ตรวจนับไฟไหม้ที่พบด้วยโมเดลที่ได้จัดทำขึ้น เมื่อได้ข้อมูลแล้วระบบจะทำการส่งข้อมูลที่นับได้ กับภาพถ่ายที่ได้ทำการตีกรอบที่ตรวจ พบไปทาง Line Notification ระบบตรวจจับไฟไหม้ด้วย YOLO Object Detection แสดงผังงานของระบบตามรูปที่ 3.2

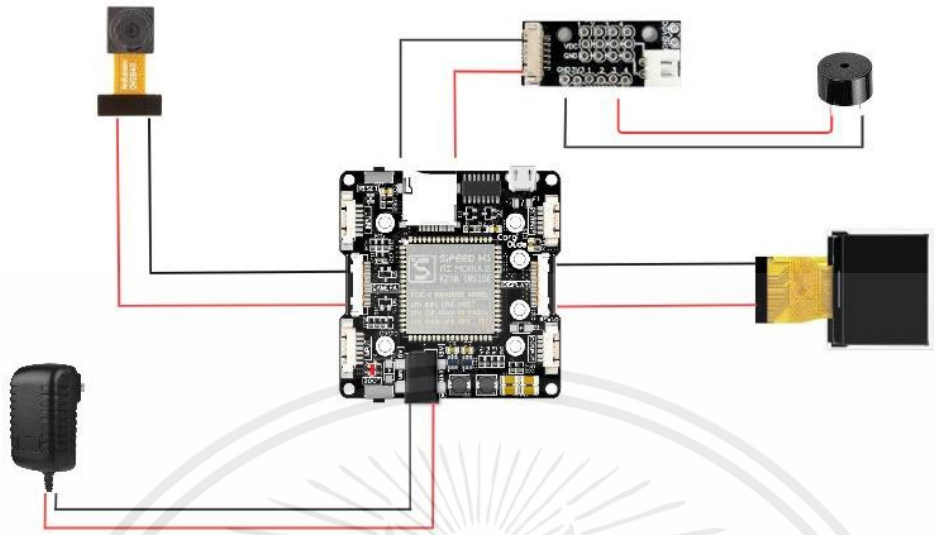
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงผังงานของระบบเฝ้าระวังเพลิงไหม้

การออกแบบวงจรและเขียนชุดควบคุมการทำงานของอุปกรณ์ ตามแผนผังใน รูปที่ 3.2 รูปที่ 3.3 และรูปที่ 3.4 แล้วนั้น โดยในปริญญานิพนธ์นี้ได้ทำการออกแบบชุดคำสั่งร่วมกับชิป Kendryte K210 เพื่อใช้เป็นหน่วยประมวลผลการทำงาน มีช่องต่อกล้อง โดยกล้องจะเป็นรุ่น OV2640 24Pin ช่องต่อจอภาพ จอภาพที่ต่อได้จะเป็นรุ่น ST7789 และต่อกับโมดูลบัซเซอร์ และเขียนคำสั่งให้โปรแกรมทำงานแบบ วนซ้ำ (Loop) ส่วนในโหมดการทำงาน การควบคุมการเปิดและปิดอุปกรณ์ภายในโรงเรียน อุปกรณ์ภายในโรงเรือนจะใช้แรงดันไฟฟ้ากระแสตรง 12 โวลต์ การออกแบบวงจรแสดงดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

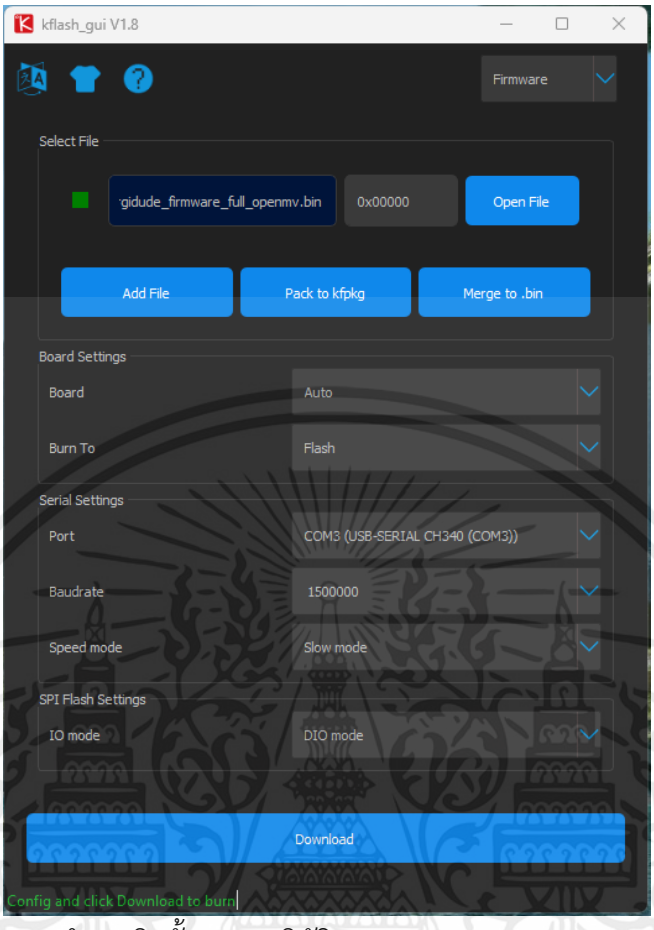


รูปที่ 3.3 วงจรเครื่องตรวจจับเพลิงไหม้

3.1.2 การติดตั้งระบบปฏิบัติการสำหรับบอร์ด Corgi Dude

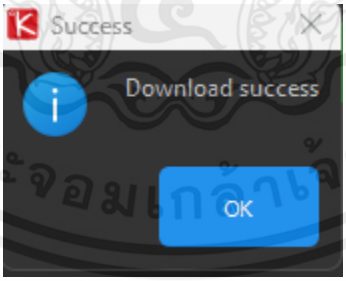
สำหรับ Corgi Dude มีความสามารถในการเร่งการประมวลผลคล้ายกับการ์ดจอ แต่ถอดออกมาเฉพาะบางส่วน ทำให้สามารถทำงาน AI บางอย่างได้เช่น ตรวจจับใบหน้า รู้จำใบหน้า จำแนกวัตถุ รู้จำเสียง ค้นหาวัตถุได้ เป็นต้น ซึ่งทำงานในตัวอุปกรณ์เองได้เลยไม่ต้องส่งข้อมูลไปไหน สามารถจำแนกรูปขนาด 224x224 ในระดับ 10 - 20 รูปต่อวินาทีได้ โดยใช้ชิป Kendryte K210 มีความเร็วในการประมวลผลอยู่ที่ 1 TOPS ที่ 108MHz + Flash 16Mbyte (Tera Operations Per Second : 1 ล้านล้าน การประมวลผล ใน 1 วินาที) ซึ่งใช้ไฟแค่ 0.3W ซึ่งชิปตัวนี้ถูกผลิตขึ้นโดยบริษัท Canaan ในชิปมีหน่วยประมวลผล KPU สำหรับการทำ Convolutional Neural Network (CNN) บนอุปกรณ์ฮาร์ดแวร์ สำหรับทำ Machine Learning โดยทาง sipeed ได้ สร้าง maixpy ที่เป็น micro python ในการที่จะ flash K210 นั้นจะต้องใช้โปรแกรม kflash_gui และทำการติดตั้งระบบปฏิบัติการ Firmware MaixPy ดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ทำการติดตั้งระบบปฏิบัติการ Firmware MaixPy ลงบอร์ด

รอจนทำการลงข้อมูลระบบปฏิบัติการเสร็จ แล้วจะขึ้น download success แปลว่าทำการลงข้อมูลเสร็จแล้ว ดังรูปที่ 3.5



รูปที่ 3.5 การดาวน์โหลดข้อมูลระบบปฏิบัติการสำเร็จ

3.1.3 การเตรียมชุดข้อมูลสำหรับเทรนโมเดล

โดยทางผู้พัฒนา YOLO แนะนำว่ารูปที่เราจะเอาเข้าโมเดลสำหรับเรียนรู้และมีประสิทธิภาพที่ดีนั้น จะมีข้อแนะนำดังต่อไปนี้

1) ควรจะมีจำนวนรูปต่อ class (class คือผลลัพธ์ของรูป เช่น รูปแมว รูปหมา) มากกว่า 1500 รูปขึ้นไป
2) ควรมีจำนวน Object มากกว่า 10000 ชิ้นต่อ class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

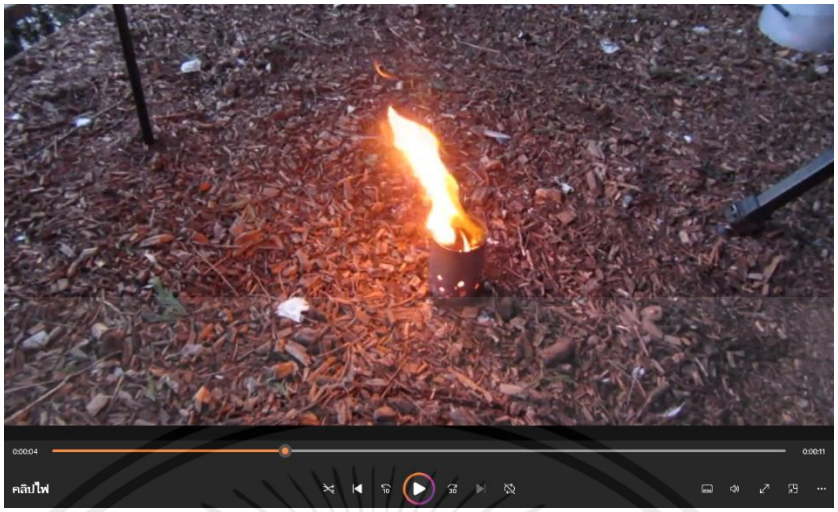
3) ควรมี Image Variety ยกตัวอย่างเช่น รูปถ่ายรถยนต์สำหรับงาน Traffic ก็ควรจะมีรูปทั้งตอนเช้า ตอนเย็น ตอนค่ำ ,รวมถึงช่วงฤดูร้อน ฤดูฝน ฤดูหนาว ,ตอนฝนตก หิมะตก, มีแสงหลายๆแบบ รวมถึงมุมกล้องที่แตกต่างกันออกไปจากหลายๆ sources (เช่น crawl จากอินเทอร์เน็ต หรือเก็บเองจากกล้องหลายตัว)
4) ความสม่ำเสมอในการ Label รูป (จะอธิบายเพิ่มเติมในบทที่ 3)
5) ความถูกต้องในการ Label รูป
6) ควรจะมี Background images โดยเป็นรูปที่ไม่มี Object อยู่เลยประมาณ 1-10% ของรูปทั้งหมด เพื่อลด False Positive Rate

3.1.4 ทำการถ่ายคลิปวิดีโอและทำการแปลงเป็นไฟล์รูปภาพ

การจัดเตรียมข้อมูลเพื่อ นำมาสร้างโมเดล โดยการนำภาพถ่ายวีดีโอคลิปจำลองไฟไหม้บริเวณห้องครัวและไฟไหม้ที่ห้องนั่งเล่น ทำการบันทึกวีดีโอคลิปในด้านต่าง ๆ เพื่อที่จะได้นำภาพวีดีโอคลิปดังกล่าวมาสร้าง โมเดลที่มีความถูกต้องแม่นยำ ทำการแปลงวีดีโอคลิปเป็นภาพที่ละเฟรมโดยใช้โค้ดจากโปรแกรมไพทอน ให้ทำงานในสภาพแวดล้อมเสมือนบนแพลตฟอร์ม Google Colab ซึ่งการทำงานจำเป็นต้องติดตั้งไลบรารีจำนวน หลายตัว เพื่อให้โค้ดไพรทอรนสามารถงานได้อย่างสมบูรณ์ เช่น TensorFlow เป็นไลบรารี่ที่พัฒนา ด้วยบริษัทกูเกิลซึ่งใช้สำหรับการพัฒนาด้าน Machine Learning โดยเฉพาะ, OPENCV-Python เป็น ไลบรารี่ที่ใช้ทำงานกับรูปภาพและการทำงานแบบเรียวทาม, Matplotlib เป็นไลบรารี่ที่ใช้ในการสร้างกราฟ ดังรูปที่ 3.6 และรูปที่ 3.7

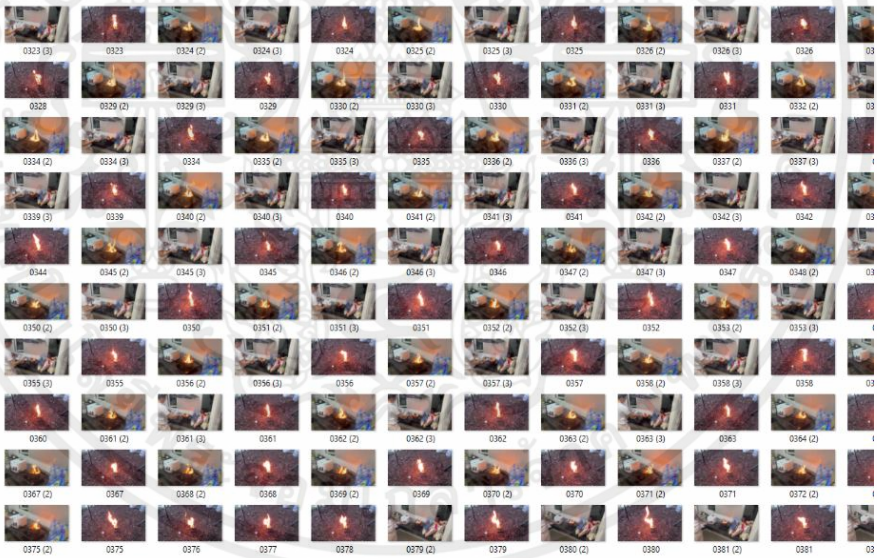
รูปที่ 3.6 คลิปแสดงการจำลองไฟไหม้ในห้องครัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 คลิปแสดงการจำลองไฟไหม้ในข้างนอกอาคาร

หลังจากที่โค้ดทำงานเรียบร้อยแล้วจะได้ภาพสำหรับเทรนในการทำโมเดล จำนวนภาพจะ มากหรือน้อยอยู่ที่ความยาวของวีดีโอคลิปและจำนวนคลิปที่ได้ถ่ายไว้ ดังรูปที่ 3.8



รูปที่ 3.8 ข้อมูลได้รับการรวบรวมข้อมูลภาพที่มีควันไฟและไฟไหม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 การออกแบบโมเดล

3.1.5.1 การ Label รูปภาพ

เมื่อได้ภาพที่ต้องการเทรนโมเดลแล้วจำเป็นต้องทำการระบุตำแหน่งของ Object ในภาพเพื่อใช้ในการเทรนภาพเข้าอัลกอริทึม โดยทางผู้วิจัยได้ใช้โปรแกรม LabelIMG เพื่ออำนวยความสะดวกในการทำงานด้านการระบุตำแหน่งของ Object สำหรับจำนวนภาพที่ใช้ในการเทรนข้อมูลยิ่งมีจำนวนภาพมากความแม่นยำของอัลกอริทึมย่อมมีมากไปด้วย ภาพใช้สำหรับตรวจสอบจำนวนภาพทั้งหมด 1000 ภาพซึ่งบางภาพมีความเบลอไม่ชัดเจน บางภาพไม่มีภาพไฟในภาพ หรือบางภาพมีภาพไฟใกล้กล้องจนเกินไป จึงไม่สามารถนำมาใช้ได้

โปรแกรม LabelIMG จะเป็นโปรแกรมที่ช่วยกำหนดตำแหน่งบนวัตถุบนภาพในรูปแบบแกน X และ Y โดยการตีกรอบบนภาพ ดังรูปที่ 3.9

รูปที่ 3.9 การทำ Image Labeling

เมื่อทำการ Image Labeling เสร็จแล้วและทำการ save เรียบร้อยแล้วจะได้ text file ที่เป็น label มา โดยที่ชื่อไฟล์ของ label จะต้องมีชื่อเดียวกับชื่อของรูปภาพ ดังรูปที่ 3.10

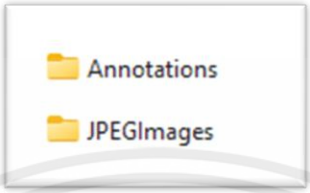
```
<annotation>
	<folder>fire-1</folder>
	<filename>microwave-oven-fire.jpg</filename>
	<path>C:\Users\USER\OneDrive\Desktop\fire kitchen\fire-1\microwave-oven-fire.jpg</path>
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>800</width>
		<height>533</height>
		<depth>3</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>fire</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>170</xmin>
			<ymin>112</ymin>
			<xmax>738</xmax>
			<ymax>526</ymax>
		</bndbox>
	</object>
</annotation>
```

รูปที่ 3.10 ตัวอย่างไฟล์ผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5.2 การเทรนภาพเข้าอัลกอริทึม YOLOv2

นำภาพที่ได้หลังจากการทำการตีกรอบ Object รูปภาพด้วยโปรแกรม LabelIMG แล้วมาแบ่งในสัดส่วนที่กำหนด โดยการนำภาพและไฟล์มาจัดให้อยู่ในรูปโฟลเดอร์ โดยที่โฟลเดอร์ Annotations เก็บไฟล์ที่ผ่านการ label เป็นไฟล์สกุล .xml และโฟลเดอร์ JPEGImages เก็บภาพที่นำไปทำการ label รูปภาพ ดังรูปที่ 3.10



รูปที่ 3.11 โครงสร้างโฟลเดอร์หลังจาก LabelIMG

อัพโหลดภาพทั้งหมดที่ได้เตรียมไว้ทั้หมดขึ้นกูเกิลไดรฟ์เพื่อใช้ในการเทรนโมเดล ด้วย Google Colab (Google Colaboration) ซึ่งเป็นบริการของกูเกิล ซึ่งเป็นบริการของบริษัทกูเกิล โดยทำการติดตั้งไลบารี่ TensorFlow V1 ซึ่งเป็นเทคโนโลยีสำหรับการสร้าง Artificial Intelligence (AI) ด้วย Deep Learning, imgaug v0.4 ใช้ในการปรับแต่งภาพ ติดตั้งไลบารี่ aXeleRate เพิ่มเติม และทำการต่อ path library ที่โหลดไปไว้ในไฟล์ระบบ ดังรูปที่ 3.12

```
# %tensorflow_version 1.x
!pip install --upgrade gdown
!pip uninstall -y imgaug && pip uninstall -
y albumentations && pip install imgaug==0.4 && pip install Keras-
Applications && pip install Keras-Preprocessing
!git clone https://github.com/apinuntong/aXeleRate-legacy-yolov2.git
import sys
sys.path.append('/content/aXeleRate-legacy-yolov2')
from axelerate import setup_training,setup_inference
```

รูปที่ 3.12 ติดตั้งไลบารี่ aXeleRate

aXeleRate เป็นไลบารี่ที่ถูกพัฒนาโดย AIWintermuteAI ด้วยโปรแกรมไพทอนโดยนำไลบารี่ Tensorflow มาใช้ในส่วนของ Machine Learning สามารถตรวจจับวัตถุ จัดหมวดหมู่รูปภาพ ด้วยอัลกอริทึม YOLOv2 ได้อย่างง่ายดาย ทั้งยังสามารถสร้างโมเดลเพื่อใช้งานได้อีกด้วย

ส่วนนี้เป็นการดาวโหลดชุดข้อมูล pascal-voc dataset ซึ่งเป็นชุดข้อมูลเพื่อนำมาใช้ในการใช้งาน จากนั้นทำการทดสอบแสดงผลของชุดข้อมูลของเรา โดยใช้จำนวนรูปเท่ากับ 10 รูปและขนาดของภาพเท่ากับ 224 พิกเซล ดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%matplotlib inline
!gdown https://drive.google.com/uc?id=1KVoipQ-VsmF3FDil2QiZgJD1_Nnf02XW  #pascal-voc dataset
!unzip --qq fire_smoke.zip

from axelerate.networks.common_utils.augment
import visualize_detection_dataset
visualize_detection_dataset(img_folder='/content/JPEGImages/', ann_folder='/content/Annotations/', num_imgs=10, img_size=224)
```

รูปที่ 3.13 ชุดคำสั่งสำหรับแสดงผล datasets

ขั้นต่อมาเป็นการย้าย datasets เอาไว้ใช้กับการ validation โดยทางผู้วิจัยจะใช้หลักของ Pereto โดยจะแบ่งจำนวน 80 เปอร์เซ็นต์เพื่อใช้ในการเทรนและอีก 20 เปอร์เซ็นต์เพื่อใช้ในการตรวจสอบของ datasets ทั้งหมด ดังรูปที่ 3.14

```
import os, random, shutil
files = os.listdir("/content/JPEGImages/")
random.shuffle(files)
files = files[0:200]
for file in files:
    imgSrc = "/content/JPEGImages/" + file
    imgTarget = "/content/images_v/" + file
    annoSrc = "/content/Annotations/" + os.path.splitext(file)[0] + "xml"
    annoTarget = "/content/annotations_v/" + os.path.splitext(file)[0] + ".xml"
    shutil.move(imgSrc,imgTarget)
    shutil.move(annoSrc,annoTarget)
```

รูปที่ 3.14 ชุดคำสั่งที่ทำการแบ่งชุดข้อมูลเอาไว้ใช้ validation ร้อยละ 20 ของชุดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนี้เป็นการกำหนด config ในการเทรนโมเดล ดังรูปที่ 3.15 ซึ่งมีส่วนที่สำคัญดังนี้

```
config = {
    "model":{
        "type":                 "Detector",
        "architecture":         "MobileNet7_5",
        "input_size":           (224,224),
        "anchors":              [0.1382,0.24, 0.4381,0.71, 0.8749,1.40, 1.6260,2.41, 2.9835,3.9032],
        "labels":               ["fire","smoke"],
        "coord_scale" :     1.0,
        "class_scale" :     1.0,
        "object_scale" :    5.0,
        "no_object_scale" :   1.0
    },
    "weights" : {
        "full":             "",
        "backend":          "imagenet"
    },
    "train" : {
        "actual_epoch":         100,
        "train_image_folder":   "/content/JPEGImages/",
        "train_annot_folder":   "/content/Annotations/",
        "train_times":          5,
        "valid_image_folder":   "/content/images_v/",
        "valid_annot_folder":   "/content/annotations_v/",
        "valid_times":          1,
        "valid_metric":         "mAP",
        "batch_size":           8,
        "learning_rate":        1e-3,
        "saved_folder":       F"/content/M",
        "first_trainable_layer": "",
        "augumentation":        True,
        "is_only_detect" :    False
    },
    "converter" : {
        "type":         ["k210"]
    }
}
```

รูปที่ 3.15 ชุดคำสั่งกำหนด config ในการเทรนโมเดล

- type คือ ชนิดที่จะเปลี่ยนได้ 3 แบบ Classifier Detector SegNet (จำแนกภาพ, ตรวจจับ(แบบสี่เหลี่ยม), ตรวจจับแบ่งขอบเขตวัตถุ)
- architecture คือ โครงสร้างของโมเดล โดยบอร์ด Corgi Dude จะเลือกใช้ MobileNet
- input_size คือ ขนาดของรูปที่เราต้องการนำเข้าไปเทรน ถ้าเป็น MobileNet จะเป็น 224,160,128 ให้เลือกตามการใช้งาน
- anchors คือ ค่ากรอบสี่เหลี่ยมที่จะเกิดขึ้นตอนตรวจจับได้
- labels คือ ค่าของวัตถุที่ต้องการตรวจจับที่จะเป็นตัวจับคู่ระหว่างตัวเลขและชื่อที่เราสามารถอ่านได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- full ค่าตรงนี้เป็นการบอกให้โมเดลไปโหลดข้อมูล weight จากโมเดลอื่นมาใส่ ไม่มีจะทำการสุ่มขึ้นใหม่
- actual_epoch เทรนจำนวนกี่รอบ
- train_image_folder ที่อยู่รูปที่จะเอาไว้เทรน
- train_annot_folder ที่อยู่เฉลยที่จะเอาไว้เทรน
- valid_image_folder ที่อยู่รูปที่จะเอาไว้ทดสอบ
- valid_annot_folder ที่อยู่เฉลยที่จะเอาไว้ทดสอบ
- batch_size ในการเทรนภาพจะถูกโหลดเข้าไปในการ์ดจอ จำนวนนี้บอกว่าจะเอากี่รูป ถ้ามีแรมที่เยอะสามารถเพิ่มได้และทำให้เทรนได้เร็วขึ้น
- learning_rate คืออัตราการเรียนรู้ ถ้ามีค่ามากโมเดลจะทำงานได้เร็ว แต่ผลลัพธ์จะแกว่งไปมา ถ้าค่าน้อยจะตรงกันข้าม
- saved_folder คือ ที่อยู่ในการบันทึกไฟล์ h5 และ kmodel
- first_trainable_layer คือ ให้กำหนด layer ไหนที่จะเริ่มเทรน
- type คือ ต้องการบันทึกไฟล์อะไรบ้าง

ในขั้นตอนนี้อาจจะใช้เวลาพอสมควรขึ้นอยู่กับจำนวนภาพและ num_experiments ที่ได้ระบุไว้ ควรเตรียมอุปกรณ์คอมพิวเตอร์ให้พร้อม ควรตั้งค่า Power & Sleep ให้เป็น Never ป้องกันเครื่องคอมพิวเตอร์ดับขณะทำงาน หลังจากทำการเทรนโมเดลเรียบร้อยแล้ว ต้องทำการประเมินความแม่นยำของโมเดลที่ได้แต่ละชุดว่าชุดใดมีความแม่นยำมากที่สุดเพื่อสามารถจะได้นำไปใช้ได้อย่างแม่นยำ

โค้ดด้านบนนี้ จะเป็นการทดสอบดูว่าเครื่องที่เรารันโค้ดอยู่มีการ์ดจอเชื่อมต่อหรือไม่ ดังรูปที่ 3.16

```
from tensorflow.python.client import device_lib
device_lib.list_local_devices()
```

รูปที่ 3.16 ชุดคำสั่งทำการเช็คว่าเครื่องเชื่อมต่อกับการ์ดจอ

ชุดคำสั่งส่วนนี้เป็นต้นไปโปรแกรมจะเริ่มเทรนโมเดลให้ตรวจจับตามข้อมูลที่เราตั้งค่าไว้ และจะทำการบันทึกรอบที่สามารถตรวจจับได้ดีที่สุดลงไปเรื่อย ๆ จนกว่าจะเทรนครบจำนวนที่เราตั้งไว้ พอรันจบ โปรแกรมจะเอาไฟล์ที่ดีที่สุดมาแปลงเป็น kmodel เพื่อให้เราสามารถใช้งานได้ต่อไป ดังรูปที่ 3.17

```
from keras import backend as K
K.clear_session()
model_path = setup_training(config_dict=config)
```

รูปที่ 3.17 ชุดคำสั่งสำหรับการเทรนโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งส่วนนี้ใช้สำหรับทดสอบรูปดูว่าได้ผลลัพธ์การเทรนกับรูปภาพว่าสามารถได้ผลลัพธ์ที่ตรงหรือไม่ ดังรูปที่ 3.17

```
%matplotlib inline
from keras import backend as K
K.clear_session()
setup_inference(config, model_path)
```

รูปที่ 3.18 ชุดคำสั่งสำหรับทดสอบผลลัพธ์

3.1.6 การพัฒนาโปรแกรมเทคนิคการตรวจจับ

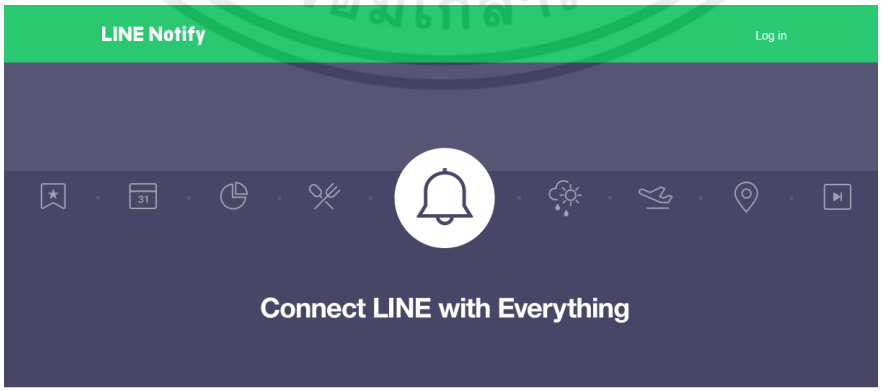
ในการพัฒนาแอพพลิเคชั่นได้ใช้โปรแกรม MaixPy IDE ใช้ภาษาของสคริปต์ Micro python ดังนั้นจึงไม่จำเป็นต้องคอมไพล์เหมือนภาษา C ในความเป็นจริงสามารถใช้งานได้ดีโดยไม่ต้องใช้ IDE ใช้เครื่องมือเทอร์มินัลอนุกรมซึ่งได้รับการติดตั้งมาก่อน การใช้ IDE จะอำนวยความสะดวกในการแก้ไขสคริปต์แบบเรียลไทม์บนคอมพิวเตอร์และอัปโหลดไปยังบอร์ดพัฒนา รันสคริปต์โดยตรงบนบอร์ดพัฒนา และดูภาพจากกล้องบนคอมพิวเตอร์แบบเรียลไทม์ และสามารถบันทึกไฟล์ไปยังบอร์ดพัฒนาได้อีกด้วย

ขั้นตอนการทำการงานของโปรแกรมตรวจนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านโมดูลกล้อง เมื่ออุปกรณ์เริ่มทำงานจะให้ตรวจจับไฟไหม้ด้วยโมดูลกล้องตามที่ได้โหลดโมเดลไว้ด้วยโปรแกรม MaixPyIDE ในบอร์ด Corgi Dude จากนั้นจะทำการตรวจจับ เข้าอัลกอริทึม YOLOv2 เพื่อทำการตรวจจับไฟไหม้และกรอบที่ตีล้อมไฟไหม้ที่พบ จากนั้นโปรแกรมจะทำการส่งข้อมูล กับภาพถ่ายที่ได้ทำการตีกรอบไฟที่ตรวจพบไปทางโปรแกรมไลน์ ที่ได้ลงทะเบียนไว้ หลังจากที่ส่งข้อมูผ่านทางโปรแกรมไลน์

3.1.7 การออกแบบการส่งแจ้งเตือนเพลิงไหม้ผ่านแอพพลิเคชั่นไลน์

3.1.7.1 การใช้งานบริการ LINE Notify

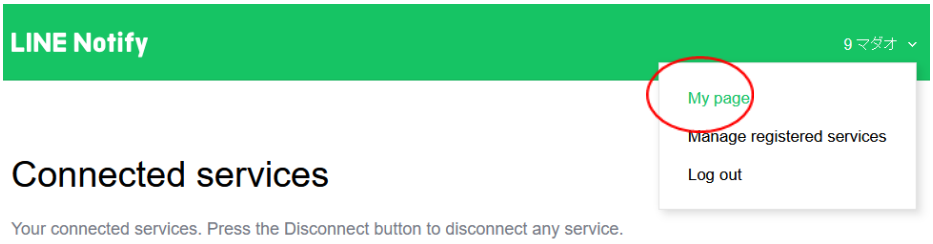
1) เข้าไปที่ https://notify-bot.line.me แล้วเข้าสู่ระบบด้วยอีเมลที่ผูกไว้กับไลน์ดังรูปที่ 3.19



รูปที่ 3.19 เข้าสู่ระบบแอพพลิเคชันไลน์

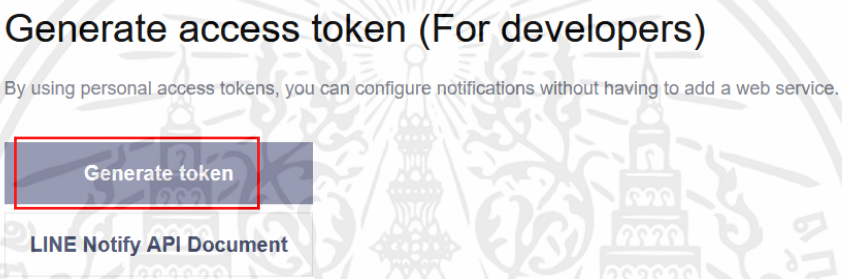
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) หลังจากเข้าสู่ระบบแล้วให้เข้าเมนู “My page”ดังรูปที่ 3.20



รูปที่ 3.20 เข้าเมนู “My page”

3) กดที่ปุ่ม Generate token เพื่อทำการออก Token สำหรับนำ API มาเชื่อมต่อในโปรแกรม Python ดังรูปที่ 3.21



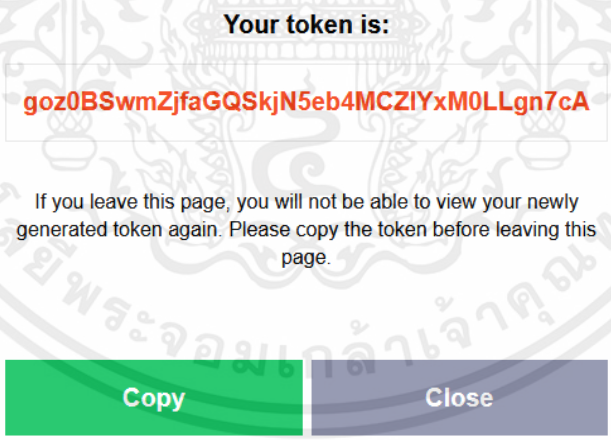
รูปที่ 3.21 ทำการออก Token สำหรับนำ API มาเชื่อมต่อ

4) ทำการตั้งชื่อการแจ้งเตือนที่จะแสดง (1) และ เลือกผู้ใช้งาน หรือกลุ่มLine ที่ต้องการให้ระบบส่งการแจ้งเตือนไปให้ (2) จากนั้นกดปุ่ม “ออก Token” (3) ดังรูปที่ 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.22 ตั้งชื่อการแจ้งเตือนที่จะแสดงและเลือกผู้ใช้งานที่ต้องการแจ้งเตือนไปยังผู้ใช้

5. คัดลอก Token เพื่อนำไปใส่ในหน้า Line API ในโปรแกรม Python ต่อไป ดังรูปที่ 3.23



รูปที่ 3.23 คัดลอก token เพื่อไปใช้ในโปรแกรม Python

3.1.7.2 ชุดคำสั่งที่ใช้ในการทำระบบส่งแจ้งเตือน

1) ข้อมูล API ของแอพพลิเคชันไลน์

การเขียนชุดคำสั่งของ MicroPython เชื่อมต่อกับแอพพลิเคชันไลน์เพื่อทำการส่งข้อความแจ้งเตือนนั้น จำเป็นที่จะต้องมีข้อมูล API ในส่วน Notification ของแอพพลิเคชันไลน์ โดยสามารถเข้าถึงได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“LINE Notify API Document” ดังแสดงในรูปที่ 3.24 ซึ่งเมื่อกดเข้ามาในส่วนของ Notification จะได้ข้อมูล API ที่ต้องการ

POST https://notify-api.line.me/api/notify

Sends notifications to users or groups that are related to an access token.
If this API receives a status code 401 when called, the access token will be deactivated on LINE Notify (disabled by the user in most cases). Connected services will also delete the connection information.
Requests use POST method with application/x-www-form-urlencoded (Identical to the default HTML form transfer type).

Expected use cases

When a connected service has an event that needs to send a notification to LINE

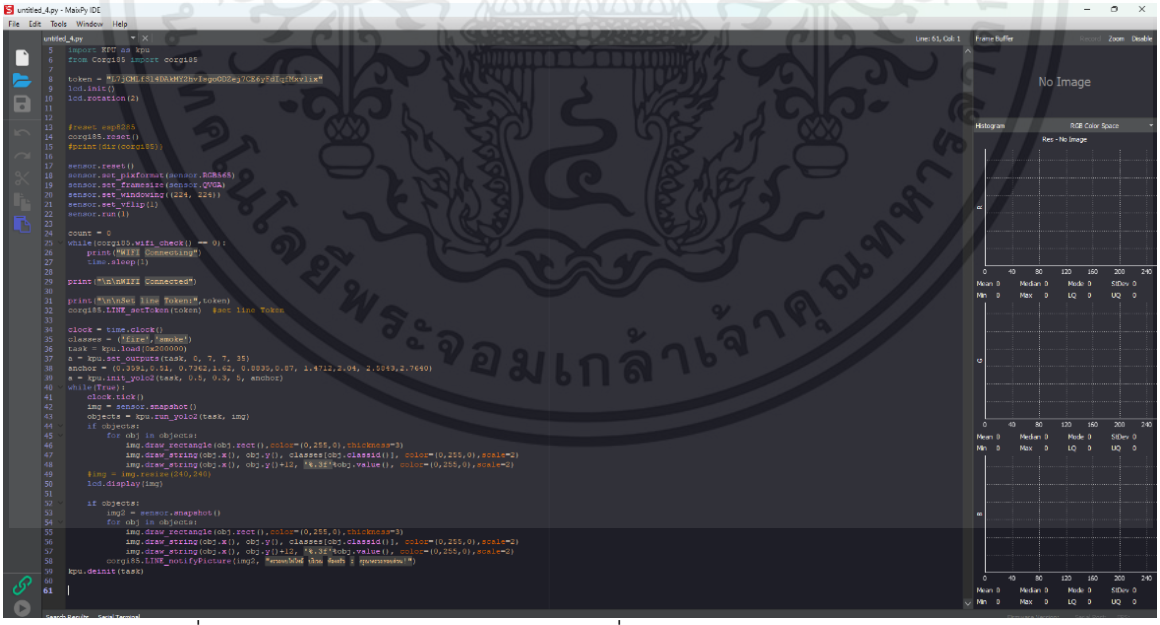
Request method

Request methods/headers	Value
Method	POST
Content-Type	application/x-www-form-urlencoded OR multipart/form-data
Authorization	Bearer <access_token>

รูปที่ 3.24 ข้อมูล API ในส่วน Notification ของแอพพลิเคชันไลน์

2) ชุดคำสั่งที่ออกแบบภายในโปรแกรม MaixPy IDE

นำข้อมูล API และ Token ที่ได้มาใส่ในโค้ดที่ออกแบบไว้ เพื่อทดลองให้สามารถส่งข้อความ “Fire Detected” ไปยังแอพพลิเคชันไลน์ได้ ดังรูปที่ 3.25



รูปที่ 3.25 ตัวอย่างโค้ด Micro Python เพื่อส่งข้อความไปยังแอพพลิเคชันไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.8 การออกแบบอุปกรณ์

ทำการประกอบโครงสร้างและวงจรเข้าด้วยกันทุกส่วน หลังที่กำหนดโครงสร้างและโปรแกรมเรียบร้อยแล้ว ดังรูปที่ 3.26

รูปที่ 3.26 อุปกรณ์ตรวจจับเพลิงไหม้

3.2 เครื่องมือที่ใช้ในการทดลอง

ในปริญญานิพนธ์นี้มีอุปกรณ์และเครื่องมือในการทดสอบดังนี้

3.2.1 บอร์ด Corgi Dude

เป็นบอร์ดไมโครคอนโทรลเลอร์ที่สามารถประมวลผลคล้ายกับการ์ดจอ มีระบบสมองกลฝังตัวอยู่ภายใน เช่น ตรวจจับใบหน้า รู้จำใบหน้า จำแนกและค้นหาวัตถุ ใช้ชิปประมวลผล Kendryte K210 ความเร็วในการประมวลผล 1 TOPS คือ 108MHz + Flash 16Mbyte (Tera Operations Per Second) ใช้กำลังไฟฟ้า 0.3 วัตต์ ใช้สถาปัตยกรรม RISC-V Dual Core 64 บิต มีขาอินพุต/เอาต์พุตใช้งานทั่วไป และพอร์ตต่อกล้องกับหน้าจอ

3.2.2 โมดูลกล้อง OV2640 2MP

โมดูลกล้อง OV2640 มีความละเอียดสูงสุด 2MP ส่วนประกอบในตัวของกล้อง มี เซ็นเซอร์ภาพ CMOS ที่ให้ภาพ VGA พร้อมหน่วยประมวลผลภาพ แหล่งจ่ายไฟ และเลนส์ สามารถส่งออกภาพนามสกุล JPEG หรือภาพวิดีโอและอินเทอร์เฟซ YCbCr 8/10 บิต แบบ ดิจิทัล มีขนาดเล็กสะดวกต่อการใช้งาน

3.2.3 โมดูลจอแสดงผลแบบสี ขนาด 1.3 นิ้ว ความละเอียด 240x240

จอภาพขนาด 1.3 นิ้ว ความละเอียด 240×240 Pixel จอสีแบบ IPS ถูกออกมาให้ ใช้งานร่วมกับไมโครคอนโทรลเลอร์เชื่อมต่อผ่านทางพอร์ต SPI ใช้ชิพ ST7789 ท างานที่ แรงดัน 3.3 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 โมดูลบัซเซอร์

โมดูลบัซเซอร์คือ อุปกรณ์ซึ่งทำหน้าที่ส่งสัญญาณเสียงที่ติดตั้งรวมมากับวงจร กำเนิดความถี่เพื่อทำให้เกิดเสียง ภายในตัวบัซเซอร์ประกอบด้วยขดลวดต่ออนุกรมกับแผ่น สั่นสะเทือนซึ่งทำหน้าที่เป็นหน้าสัมผัสตัดต่อวงจรและใช้การดึง และดีดกลับซ้ำไปมาทำให้เกิดเสียงดังต่อเนื่อง

3.2.5 อุปกรณ์แปลงแรงดันไฟฟ้า

ในปริญญานิพนธ์นี้ผู้จัดทำได้เลือกใช้อุปกรณ์แปลงแรงดันไฟฟ้า ที่ให้แรงดันไฟฟ้าขาออกขนาด 6 โวลต์ และกระแสไฟฟ้า 2 แอมป์ เป็นแหลงจ่ายไฟให้กับ บอร์ด Corgi Dude แสดงดังรูปที่ 3.27

รูปที่ 3.27 อุปกรณ์แปลงแรงดันไฟฟ้า

คอมพิวเตอร์พกพา (Laptop) สำหรับการใช้งานโปรแกรมการทำงาน รายละเอียดข้อมูลจำเพาะและ ดังรูปที่ 3.28 ถึง 3.29

รูปที่ 3.28 Laptop Acer Aspire 3 A315-5122

คอมพิวเตอร์พกพา (Laptop) Acer Aspire 3 A315-5122 รายละเอียดข้อมูลจำเพาะ ดังนี้

1) CPU : Intel Core i5-8250U (1.60 GHz, 6 MB L3 Cache up to 3.40 GHz)

2) GPU : NVIDIA GeForce MX130 (2GB GDDR5)

3) RAM : 8 GB DDR4 2133Mhz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.29 Laptop Lenovo YOGA 530

คอมพิวเตอร์พกพา (Laptop) Lenovo YOGA 530 รายละเอียดข้อมูลจำเพาะ ดังนี้

1) CPU : Intel Core i5-8250U (1.60 GHz, 6 MB L3 Cache up to 3.40 GHz)

2) GPU : NVIDIA GeForce MX130 (2GB GDDR5)

3) RAM : 8 GB DDR4 2400 Mhz

3.3 การจัดเก็บผลการทดสอบ

3.3.1 การทดสอบการทำงานของโมเดลแบบจำลอง

ทำการทดสอบการเทรนโมเดลแบบจำลองของ YOLOv2 โดยนำข้อมูลภาพเพลิงไหม้ที่บันทึกได้มาทำการเทรน แบบจำลองเพื่อใช้ในการสร้างแบบจำลองสำหรับการตรวจจับวัตถุเพลิงไหม้ โดยการใช้ Google Colab จากนั้นทำการสังเกตผลค่าที่ได้รับ

3.3.2 การทดสอบจากแบบจำลองที่ผ่านการเรียนรู้

ทำการนำข้อมูลภาพ มาใช้กับแบบจำลองจากโมเดลที่ได้สร้างขึ้น โดยทำการทดลองกับไฟล์ภาพและไฟล์วิดีโอ และนำมาทำการตรวจจับวัตถุเพลิงไหม้ จากแบบจำลองที่ผ่านการเรียนรู้ 100 รอบ โดยทำการตรวจจับ และทำการเก็บผลการทดสอบ

3.3.3 การทดสอบการทำงานร่วมกับบอร์ด Corgi Dude

ทำการนำโมเดลที่ได้ออกแบบไว้มาทำการใช้งานร่วมกับบอร์ด Corgi Dude ให้ทำการตรวจจับภาพที่มีวัตถุเพลิงไหม้ ณ เวลานั้น และแสดงผลออกมาทางจอภาพ

3.3.4 การทดสอบการทำงานร่วมกับแอพพลิเคชันไลน์

ทดสอบการทำงานแสดงผลการทำงานของโมเดล ร่วมกับบอร์ด Corgi Dude ให้ทำการตรวจจับวัตถุที่มีเพลิงไหม้เกิดขึ้นและส่งข้อความแจ้งเตือนไปยังแอพพลิเคชันไลน์

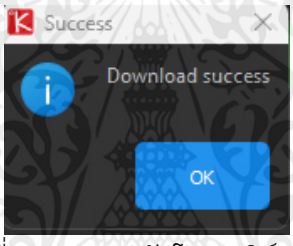
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

จากการศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องเพื่อเป็นความรู้สำหรับการศึกษาในการวิจัยครั้งนี้ ผู้วิจัยได้จำลองการวิจัยเพื่อศึกษาเกี่ยวกับเทคนิคการตรวจจับไฟไหม้ด้วยการนำโมเดลที่ผู้วิจัยได้ทำการเทรนโมเดลขึ้น มาทำการทดลองความแม่นยำจากการตรวจจับไฟไหม้ โดยได้เก็บผลการทดลอง นำข้อมูลที่ได้รับมาสรุป และนำข้อมูลมาวิเคราะห์เพิ่มเติม

4.1 ผลการทดสอบการติดตั้งเฟิร์มแวร์ ของ Corgi Dude

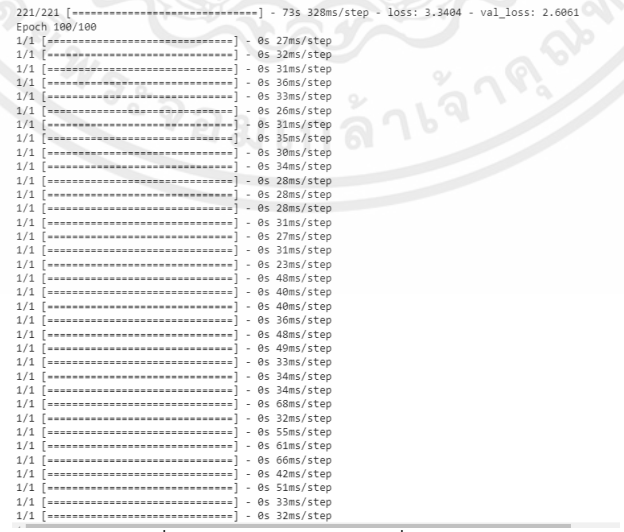
ในการทำงานทำงานของบอร์ด Corgi Dude จะต้องใช้โปรแกรม kflash เพื่อใช้ในการอัปโหลดเฟิร์มแวร์ให้กับ Corgi Dude เพื่อใช้ในการใช้งานโปรแกรม MaixPy และใช้งานโมเดล เมื่อสามารถเชื่อเมื่อสามารถทำการอัปโหลดข้อมูลเสร็จ จะแสดงหน้าต่างที่มีข้อความ “Download success” ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 ผลการอัปโหลดเฟิร์มแวร์

4.2 ผลการทดสอบการเทรนโมเดล

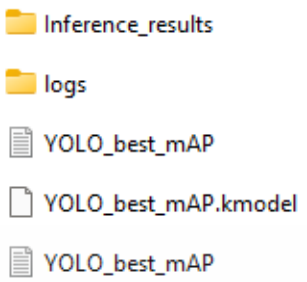
ภาพที่ได้มาทำ Labeling และเทรนโมเดล ด้วยอัลกอริทึม Yolov2 หลังจากเทรนโมเดลแล้ว จะได้โมเดลต้นแบบจำนวนหลายโมเดล ต้องทำการประเมินความแม่นยำเพื่อหาโมเดลที่แม่นยำที่สุด เพื่อนำมาใช้ในการตรวจจับ ดังรูปที่ 4.2



รูปที่ 4.2 แสดงการเทรนนิ่งโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดโมเดลที่ได้หลังจากการเทรนโมเดลด้วยภาพถ่ายที่ได้เตรียมไว้ ดังรูปที่ 4.3



รูปที่ 4.3 ชุดโมเดลหลังจากทำการเทรน

ผลการทดสอบการเทรนโมเดลของรูปภาพไฟไหม้ หลังจากผ่านการเทรนทั้งหมด 100 รอบจะแสดงตัวอย่างผลการทดสอบที่ทำการเทรนได้สูงที่สุดจำนวน 100 รอบ ผลที่ได้จากการตรวจสอบความแม่นยำของโมเดล โดยจะเลือกชุดโมเดลที่มีค่า mAP (Mean Average Precision) มากที่สุดจากชุดโมเดลทั้งหมดที่ได้เทรนออกมา ดังรูปที่ 4.4

รูปที่ 4.4 ผลลัพธ์จากการตรวจสอบความแม่นยำสูงที่สุดของโมเดล

ผลลัพธ์จากการทดสอบการทำงานของโมเดลกับภาพข้อมูลไฟไหม้จำนวน 100 รอบ แสดงให้เห็นว่าโมเดลตรวจจับไฟไหม้มีค่าความถูกต้อง ดังตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ค่าความถูกต้องของการเทรนโมเดล

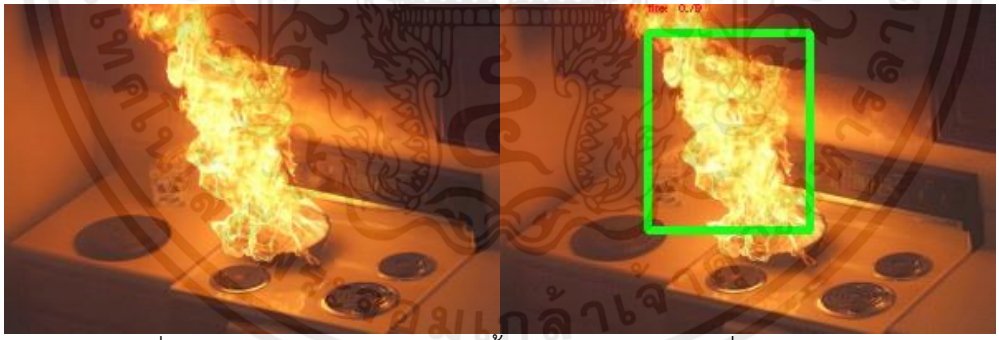
จำนวนรอบ	ค่าความแม่นยำเฉลี่ย	ค่าความแม่นยำสูงสุด
100	0.9012	0.9881

จากตารางที่ 4.1 ตารางแสดงค่าความถูกต้องของการเทรนโมเดล มีค่าเฉลี่ยความถูกต้อง มากกว่า 0.9 หรือ 90 เปอร์เซ็นต์ ซึ่งอยู่ในเกณฑ์ที่ดี จากผลการทดสอบเป็นการแสดงให้เห็นว่าระบบตรวจจับไฟไหม้สามารถตรวจจับได้อย่างแม่นยำ

4.3 ผลการทดสอบจากโมเดลที่ผ่านการเทรน

จากข้อมูลภาพเพลิงไหม้ บันทึกได้นั้น มีองค์ประกอบที่แตกต่างกัน มากเช่น สถานที่ มุมมองของกล้อง เวลา เป็นต้น ทำให้โมเดลที่ได้มีการใช้งานแตกต่างออกไปตาม สถานที่และมุมมองของกล้อง ซึ่งถ้าหากนำแบบจำลองที่ผ่านการเทรนมาใช้ในสถานที่ซึ่ง องค์ประกอบไม่สอดคล้องกัน อาจทำให้ผลลัพธ์มีความคลาดเคลื่อนเป็นอย่างมากและไม่สามารถ นำมาใช้งานได้จริง อีกทั้งการนำข้อมูลภาพ มาใช้กับแบบจำลองดั้งเดิมที่ไม่ได้มีการปรับปรุงแก้ไข และพัฒนาให้เหมาะกับการตรวจจับเพลิงไหม้นั้น จะไม่สามารถระบุประเภทของวัตถุและตรวจจับวัตถุได้

ทดสอบภาพจากโมเดลที่ผ่านการเทรน โดยทำการทดสอบโดยการนำไฟล์ภาพมา 1 ภาพ และนำมาทำการตรวจจับวัตถุเพลิงไหม้ จากโมเดลที่ผ่านการเทรน 100 รอบ โดยทำการและแสดงผลการตรวจจับวัตถุเพลิงไหม้ โดยดูจากการเปรียบเทียบระหว่างภาพดั้งเดิมกับภาพที่ผ่านโมเดลที่ผ่านการเทรน ดังรูปที่ 4.5



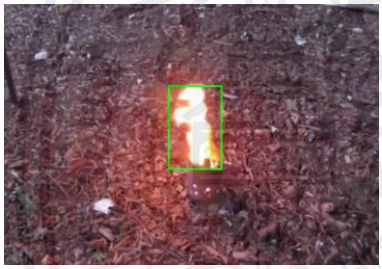
รูปที่ 4.5 ภาพเปรียบระหว่างภาพดั้งเดิมกับแบบจำลองที่ผ่านการเทรน

4.4 ผลการนำโมเดลมาใช้งานจริงในการตรวจจับ

นำโมเดลที่ได้ในการเทรนทดสอบการตรวจจับไฟไหม้เพื่อทดสอบความแม่นยำของอัลกอริทึมที่ได้จัดทำขึ้น ดังตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 แสดงรูปภาพกับโมเดลที่ทำการตรวจจับ (Detection)

ภาพที่ / การดำเนินการ	ภาพที่ 1	ภาพที่ 2	ภาพที่ 3
ภาพเหตุการณ์เพลิงไหม้			
ภาพเหตุการณ์เพลิงไหม้ (ผ่านการตรวจจับ)			
ภาพที่ระบบไม่สามารถตรวจจับได้			

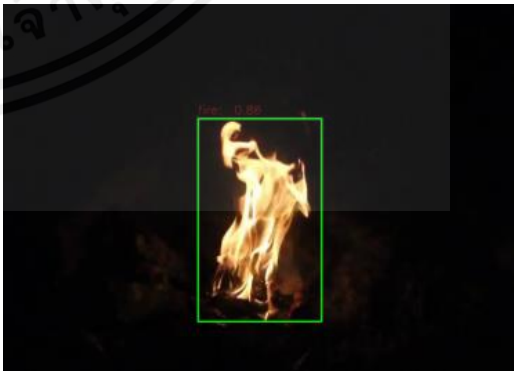
จากตารางที่ 4.2 จะเห็นได้ว่ามี 2 กรณีที่น่าสนใจ กรณีที่ 1 อาจจะไม่เห็นไฟไหม้ทั้งจอ ภาพที่ระบบตรวจสอบได้ในกรณีนี้ ภาพที่เห็นจะต้องเห็นเปลวไฟโดยสมควร อ้างอิงจากเหตุการณ์เพลิงไหม้ (ผ่านการตรวจจับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2 และหากเห็นเปลวไฟน้อยเกินไปจะจับไม่ได้ดังที่ระบบไม่สามารถตรวจจับได้ ภาพที่ 2 และในกรณีที่ 2 เห็นภาพมากหรือน้อยเกินไป ระบบจะสามารถตรวจสอบได้ในกรณีนี้ ภาพที่เห็นจะต้องมีขนาดที่เหมาะสม อ้างอิงจากภาพเหตุการณ์เพลิงไหม้ (ผ่านการตรวจจับ) ภาพที่ 1 และ ภาพที่ 3 และจะตรวจจับได้ยากในกรณีที่ภาพเปลวไฟมีขนาดใหญ่หรือเล็กเกินไปจนระบบไม่แน่ใจว่าใช่ไฟไหม้หรือไม่ ดังภาพที่ระบบไม่สามารถตรวจจับได้ ภาพที่ 1 และภาพที่ 3

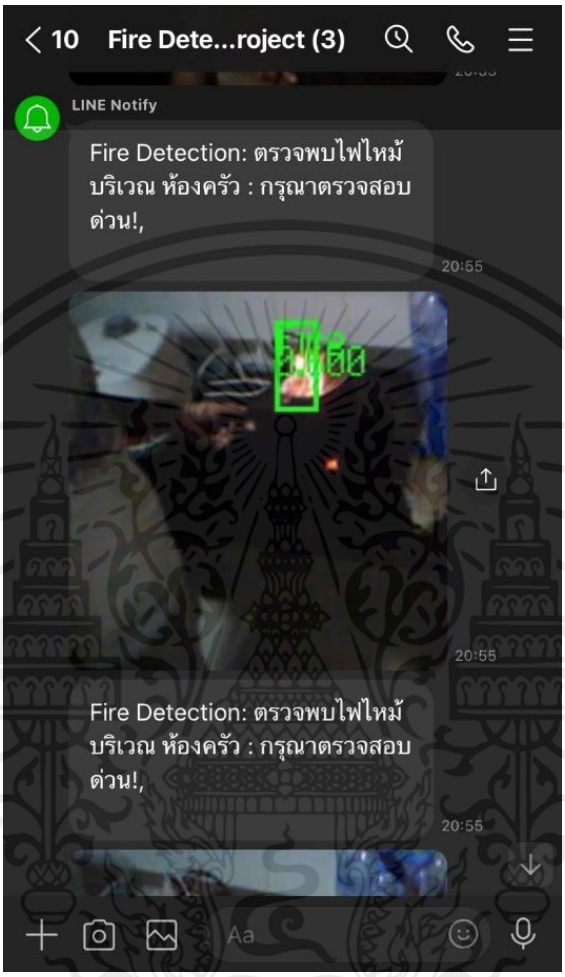
ผลการถ่ายภาพไฟไหม้ด้วยอุปกรณ์ตรวจจับไฟไหม้ตามสภาพแวดล้อมที่แตกต่างกัน โดยนำเปรียบเทียบตารางที่ 4.3

ตารางที่ 4.3 ถ่ายภาพการตรวจจับในสภาวะที่แตกต่างกัน

สภาพแวดล้อม	**ภาพถ่าย**	**ภาพหลังผ่านการตรวจจับ**
ภาพถ่ายสภาพแวดล้อมที่มีแสงสว่างมาก		
ภาพถ่ายสภาพแวดล้อมที่มีแสงสว่างน้อย		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการส่งข้อความและภาพที่ได้ทำการตรวจพบไฟไหม้ ผ่านทางโปแกรมแอปพลิเคชันไลน์ โดยจะเป็นข้อความผลการตรวจจับไฟไหม้ 1 บรรทัดและภาพที่ได้ทำการตีกรอบที่ ตรวจพบไฟไหม้ อีกจำนวน 1 ภาพ ดังรูปที่ 4.6



รูปที่ 4.6 การส่งข้อมูลระดับไฟไหม้ และภาพทางโปรแกรมแอปพลิเคชันไลน์

ระบบตรวจจับเพลิงไหม้ สามารถตรวจจับไฟไหม้ในห้องครัวหรือในอาคารได้เป็นอย่างดี จะเห็นได้ จากผลการทดลอง ที่มีการใช้ภาพหลากหลายรูปแบบที่ได้มา จากการถ่ายคลิปวิดีโอ ภาพเหล่านั้นจะให้ข้อมูลคุณลักษณะรูปแบบของไฟไหม้แก่ระบบ ซึ่งจะทำให้ระบบได้รู้ถึงรูปแบบของเปลวเพลิงมีลักษณะเป็นอย่างไร โดยข้อมูลที่นำใช้เทรนมีค่าความแม่นยำมากกว่า 90 เปอร์เซ็นต์ จากการทดลองส่งผลให้การเตือนของระบบตรวจจับไฟไหม้ นั้นมีประสิทธิภาพที่ดีตามไปด้วยและเมื่อทำการตรวจจับได้จะส่งข้อมูลและรูปภาพไปยังแอปพลิเคชันไลน์ และอาจจะได้เห็นว่ายังมีบางกรณีที่ ระบบไม่สามารถตรวจจับไฟไหม้ได้ ซึ่งเกิดจากข้อมูลที่ถูกฝึกฝนให้แก่ระบบมีไม่ครอบคลุมที่จะใช้ในการตรวจจับเมื่อเปลวไฟมีขนาดใหญ่หรือเล็กจนเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและข้อเสนอแนะ

การศึกษาและวิจัยเรื่องประสิทธิภาพของระบบตรวจจับเพลิงไหม้บริเวณภายในห้องครัวนี้ ได้มีการนำเสนอข้อสรุปของปริญญานิพนธ์ตามลำดับดังนี้ คือ สรุปผลของงานวิจัย ปัญหาและอุปสรรค และข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้มีเป้าหมายเพื่อศึกษาและทำการออกแบบระบบตรวจจับและแจ้งเตือนเพลิงไหม้บริเวณภายในห้องครัวซึ่งเป็นพื้นที่ภายในที่พักอาศัยที่มีความเสี่ยงสูงต่อการเกิดเพลิงไหม้ ซึ่งสามารถตรวจจับได้ทั้งเปลวไฟและกลุ่มควันเนื่องจากการลุกไหม้ โดยอาศัยเทคโนโลยี AI (Artificial Intelligence) ซึ่งถูกสร้างและเรียนรู้ผ่านนวัตกรรม YOLO (You Only Look Once) ประมวลผลด้วยบอร์ด Corgidude ทำงานได้เมื่อเมื่อมีไฟเลี้ยงเข้ามาในระบบ เมื่อระบบตรวจพบเพลิงไหม้จะทำการส่งแจ้งเตือนให้ผู้ใช้รู้ผ่านเสียงเตือนและแอพพลิเคชันไลน์ โดยมีการดำเนินงานซึ่งสามารถสรุปได้ ดังนี้

1) ออกแบบโครงสร้างเครื่องตรวจจับไฟไหม้โดยการใช้บอร์ดคอนโทรลเลอร์ Corgi Dude เพื่อทำการเขียนคำสั่งใช้งานลงบนบอร์ดโดยใช้งานร่วมกับโมดูลกล้องและทำการส่งข้อมูลไปยังหน้าจอแสดงภาพ

2) การนำเข้าชุดข้อมูลโดยการทำการอัดคลิปวิดีโอ และทำการแปลงเป็นรูปภาพรูปภาพที่มีควันไฟและไฟไหม้ รวมถึงกรองภาพที่มีความละเอียดต่ำออก หลังจากได้ภาพแล้วจากนั้นทำการเตรียมการกำหนดผลลัพธ์ในแต่ละรูป (Data Labeling) สำหรับการทำ Image Detection

3) นำเข้าข้อมูลจากชุดข้อมูลมาจัดเตรียมสำหรับโมเดลการตรวจจับไฟไหม้ โดยการทำ Image Detection โดยใช้วิธี YOLO โดยที่ YOLO คือสถาปัตยกรรมที่ทาง ultralytics ได้ออกแบบไว้เพื่อทำ Image Detection ได้อย่างรวดเร็วและมีประสิทธิภาพ

4) ทำการ Train โมเดลสำหรับการทำ image Detection นำข้อมูลจากไฟล์ภาพข้อมูลมาทำการสร้างแบบจำลองตรวจจับไฟไหม้ มาประมวลผลและทดสอบร่วมกับข้อมูลชุดรูปภาพที่มีอยู่

5) นำโมเดลการตรวจจับไฟไหม้ที่ทำการสร้างขึ้นจาก YOLO โดยการใช้บอร์ด และทำการตรวจจับภาพไปยังจอแสดงผล

6) ทำการส่งข้อมูลแสดงผลและทำการแจ้งเตือนไปยังแอพพลิเคชันไลน์

จาการทดสอบภาพจากโมเดลที่ผ่านการเทรน โดยทำการทดสอบโดยการนำไฟล์ภาพมา 1 ภาพ นำมาทำการตรวจจับวัตถุเพลิงไหม้ จากโมเดลที่ผ่านการเทรน 100 รอบ โดยทำการและแสดงผลการตรวจจับวัตถุเพลิงไหม้ เมื่อเปรียบเทียบระหว่างภาพดั้งเดิมกับภาพที่ผ่านโมเดลที่ผ่านการเทรนพบว่าโมเดลสามารถตรวจจับภาพที่มีเปลวไฟได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาและอุปสรรค

1) จำนวนของรูปภาพที่ตรงตามเงื่อนไขเพลิงไหม้บริเวณภายในห้องครัวที่จะนำมาใช้เป็นข้อมูลเพื่อนำไปสร้างโมเดลตรวจจับเปลวไฟและควันไฟมีให้ใช้น้อยกว่าที่ควรจะเป็น ทำให้อาจตรวจไฟไหม้ที่มีขนาดที่ใหญ่หรือเล็กมากได้

2) ประสิทธิภาพของคอมพิวเตอร์ที่ใช้ในการเทรนโมเดลและประมวลผลชุดคำสั่งมีไม่สูงนักเมื่อเทียบกับจำนวนข้อมูลที่ใช้ ทำให้การรันคำสั่งและการเทรนโมเดลแต่ละรอบใช้เวลานาน และยิ่งนานขึ้นไปอีกเมื่อมีข้อผิดพลาดภายในชุดคำสั่งหรือโมเดลที่ได้ไม่สามารถสร้างผลลัพธ์ที่น่าพึงพอใจแก่ผู้วิจัย

5.3 ข้อเสนอแนะ

1) การสร้างโมเดลการตวจจับด้วยข้อมูลรูปภาพในโปรแกรม LabelImg ให้ได้ผลลัพธ์ที่แม่นยำ ควรมีจำนวนข้อมูลแต่ละคลาสตั้งแต่ 1500 รูปขึ้นไป และมีองค์ประกอบอื่นๆภายในภาพที่หลากหลาย

2) ก่อนการเทรน (Train) AI ในโปรแกรมไพธอน (Python) ควรตรวจสอบข้อมูลรูปภาพให้ถูกต้องทั้งจำนวน นามสกุลไฟล์ และชื่อของรูปภาพที่ตรงกันของข้อมูลก่อนและหลังทำ Label รวมไปถึงจำนวนรอบในการเทรนควรมากกว่า 100 รอบขึ้นไป

3) ในขั้นตอนการเบิร์นโมเดลลงไปในบอร์ด Corgidude ควรตรวจสอบตำแหน่งให้แน่ใจว่าถูกต้องแล้ว เนื่องจากหากตำแหน่งผิดพลาดอาจทำให้บอร์ดเสียหายได้

4) ตรวจสอบวิธีการต่อโมดูลหรืออุปกรณ์ทางอิเล็กทรอนิกส์ต่างๆรวมถึงค่าทางไฟฟ้าเข้ากับบอร์ด Corgidude ให้ถูกต้อง หากผิดพลาดอาจก่อให้เกิดความเสียหายต่อโมดูลหรือบอร์ดได้

5) การติดตั้งอุปกรณ์ตรวจจับและแจ้งเตือนเพลิงไหม้นี้ ควรติดตั้งในตำแหน่งที่สูงเหนือหัว มุมกล้องครอบคลุมบริเวณภายในครัวให้ได้มากที่สุด เพื่อสอดส่องบริเวณที่มีการใช้ไฟ แก๊สหุงต้ม และอุปกรณ์ไฟฟ้าที่ใช้ในการทำอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] กองเผยแพร่และประชาสัมพันธ์. “ปภ.เผยสถิติสาธารณภัยปี 2565” [Online]. Available : http://relation.disaster.go.th/

[2] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” CoRR[Online]. Available: http://arxiv.org/abs/1804.02767. [Accessed: April 8, 2018]

[3] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 27 - 30, 2016, pp. 779-788.

[4] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July21 - 26 2017, pp. 6517–6525.

[5] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, et al., “Microsoft COCO: Common Objects in Context,” 2014. [Online]. Available:https://arxiv.org/abs/1405.0312. [Accessed: May 1, 2014].

[6] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyamaand Y. Sekimoto, “Transfer Learning-based Road Damage Detection for Multiple Countries,” [Online]. Available: https://arxiv.org/abs/2008.13101. [Accessed: August 30, 2020].

[7] TT Smart Vision. “Homeระบบตรวจจับไฟไหม้ Early Fire Detection” [Online]. Available : https://www.ttsmartvision.com/

[8] P.CCTV Network Engineering. “ประโยชน์ของกล้องวงจรปิดที่มี Motion Detection”. [Online]. Available : https://www.pi-tech.biz/17200327/ประโยชน์ของกล้องวงจรปิดที่มี-Motion-Detection

[9] M. Armbrust et al. “A view of cloud computing”. In: Commun. ACM 53.4 (2010), pp. 50–58.

[10] M. Armbrust et al. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28. Available from: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf. University of California at Berkeley, Electrical Engineering and Computer Sciences, 2009, p. 23.

[11] P. Mell and T. Grance. The NIST Definition of Cloud Computing v15. Version 15 available from : http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[12] L. M. Vaquero et al. “A break in the clouds: towards a cloud definition”. In: SIGCOMM Comput. Commun. Rev. 39.1 (2009), pp. 50–55. issn: 0146-4833. doi: http://doi.acm.org/10.1145/1496091.1496100.

[13] N. McKeown et al. “OpenFlow: enabling innovation in campus networks”. In: SIGCOMM Comput. Commun. Rev. 38.2 (2008), pp. 69–74. issn: 0146-4833. doi: http://doi.acm.org/10.1145/1355734.1355746.

[14] N. Gude et al. “NOX: towards an operating system for networks”. In:SIGCOMM Comput. Commun. Rev. 38.3 (2008), pp. 105–110. issn: 0146-4833. doi: http://doi.acm.org/10.1145/1384609.1384625.

[15] Nuttakan Chuntra. “OpenCV ”. [Online]. Available : https://medium.com/@nut.ch40/opencv-คืออะไร

[16] Tutor4dev. “ เขียนโปรแกรมส่งอีเมลด้วยภาษา Python โปรแกรมที่เขียนง่าย”. [Online]. Available : https://www.tutor4dev.com/article/2019-03-30-send-email-using-python-and-smtplib

[17] burntoDev. “พื้นฐานการใช้ NumPy ใน Python 3 NumPy คืออะไร ?”. [Online]. Available : https://www.borntodev.com/2020/04/16/พื้นฐานการใช้-NumPy-ใน-Python-3/

[18] mindphp.com “AI (เอไอ) คืออะไร - ปัญญา ประดิษฐ์ วิธีการทำให้คอมพิวเตอร์มีความสามารถคล้ายมนุษย์”. [Online]. Available : https://www.mindphp.com/คู่มือ/73-คืออะไร/4025-what-is-ai.html

[19] Matana Wiboonyasake. “Machine Learning คืออะไร?”. [Online]. Available : https://www.aware.co.th/machine-learning-คืออะไร/

[20] องค์ประกอบของการลุกติดไฟ : เรียนรู้พื้นฐานของการเกี่ยวกับไฟ. [Online]. Available : https://www.safesiri.com/fire-component/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
โค้ดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดโปรแกรมสำหรับการเรียนรู้ของระบบและโปรแกรมตรวจจับเพลิงไหม้

```
# %tensorflow_version 1.x
!pip install --upgrade gdown
!pip uninstall -y imgaug && pip uninstall -y albumentations && pip install imgaug==0.4 && pip install Keras-Applications && pip install Keras-Preprocessing
!git clone https://github.com/apinuntong/aXeleRate-legacy-yolov2.git
import sys
sys.path.append('/content/aXeleRate-legacy-yolov2')
from axelerate import setup_training,setup_inference


%matplotlib inline
!gdown https://drive.google.com/uc?id=1KVoipQ-VsmF3FDil2QiZgJD1_Nnf02XW  #pascal-voc dataset
!unzip --qq fire_smoke.zip

from axelerate.networks.common_utils.augment import visualize_detection_dataset
visualize_detection_dataset(img_folder='/content/JPEGImages/', ann_folder='/content/Annotations/', num_imgs=10, img_size=224)


import os

def createFolder(directory):
    try:
        if not os.path.exists(directory):
            os.makedirs(directory)
    except OSError:
        print ('Error: Creating directory. ' +  directory)

import os, random, shutil
files = os.listdir("/content/JPEGImages/")
random.shuffle(files)
files = files[0:88]
for file in files:
    imgSrc = "/content/JPEGImages/" + file
    imgTarget = "/content/images_v/" + file
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    annoSrc = "/content/Annotations/" + os.path.splitext(file)[0] + ".xml"
    annoTarget = "/content/annotations_v/" + os.path.splitext(file)[0] + ".xml"
    shutil.move(imgSrc,imgTarget)
    shutil.move(annoSrc,annoTarget)

import xml.etree.ElementTree as ET
import os
import numpy as np
import random
def parse_annotation(ann_dir, img_dir, labels=[]):
    all_imgs = []
    seen_labels = {}
    for ann in sorted(os.listdir(ann_dir)):
        img = {'object':[]}
        tree = ET.parse(ann_dir + ann)
        for elem in tree.iter():
            if 'filename' in elem.tag:
                img['filename'] = img_dir + elem.text
            if 'width' in elem.tag:
                img['width'] = int(elem.text)
            if 'height' in elem.tag:
                img['height'] = int(elem.text)
            if 'object' in elem.tag or 'part' in elem.tag:
                obj = {}
                for attr in list(elem):
                    if 'name' in attr.tag:
                        obj['name'] = attr.text

                        if obj['name'] in seen_labels:
                            seen_labels[obj['name']] += 1
                        else:
                            seen_labels[obj['name']] = 1

                        if len(labels) > 0 and obj['name'] not in labels:
                            break
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                else:
                    img['object'] += [obj]
            if 'bndbox' in attr.tag:
                for dim in list(attr):
                    if 'xmin' in dim.tag:
                        obj['xmin'] = int(round(float(dim.text)))
                    if 'ymin' in dim.tag:
                        obj['ymin'] = int(round(float(dim.text)))
                    if 'xmax' in dim.tag:
                        obj['xmax'] = int(round(float(dim.text)))
                    if 'ymax' in dim.tag:
                        obj['ymax'] = int(round(float(dim.text)))
        if len(img['object']) > 0:
            all_imgs += [img]
    return all_imgs, seen_labels
def IOU(ann, centroids):
    w, h = ann
    similarities = []

    for centroid in centroids:
        c_w, c_h = centroid

        if c_w >= w and c_h >= h:
            similarity = w*h/(c_w*c_h)
        elif c_w >= w and c_h <= h:
            similarity = w*c_h/(w*h + (c_w-w)*c_h)
        elif c_w <= w and c_h >= h:
            similarity = c_w*h/(w*h + c_w*(c_h-h))
        else: #means both w,h are bigger than c_w and c_h respectively
            similarity = (c_w*c_h)/(w*h)
        similarities.append(similarity) # will become (k,) shape

    return np.array(similarities)

def avg_IOU(anns, centroids):
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    n,d = anns.shape
    sum = 0.

    for i in range(anns.shape[0]):
        sum+= max(IOU(anns[i], centroids))

    return sum/n

def print_anchors(centroids):
    anchors = centroids.copy()
    widths = anchors[:, 0]
    sorted_indices = np.argsort(widths)

    r = "anchors: ["
    for i in sorted_indices[:-1]:
        r += '%0.4f,%0.2f, ' % (anchors[i,0], anchors[i,1])

    #there should not be comma after last anchor, that's why
    r += '%0.4f,%0.4f' % (anchors[sorted_indices[-1:],0], anchors[sorted_indices[-1:],1])
    r += "]"

    print(r)

def run_kmeans(ann_dims, anchor_num):
    ann_num = ann_dims.shape[0]
    iterations = 0
    prev_assignments = np.ones(ann_num)*(-1)
    iteration = 0
    old_distances = np.zeros((ann_num, anchor_num))

    indices = [random.randrange(ann_dims.shape[0]) for i in range(anchor_num)]
    centroids = ann_dims[indices]
    anchor_dim = ann_dims.shape[1]

    while True:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        distances = []
        iteration += 1
        for i in range(ann_num):
            d = 1 - IOU(ann_dims[i], centroids)
            distances.append(d)
        distances = np.array(distances) # distances.shape = (ann_num, anchor_num)

        print("iteration {}: dists = {}".format(iteration, np.sum(np.abs(old_distances-distances))))

        #assign samples to centroids
        assignments = np.argmin(distances,axis=1)

        if (assignments == prev_assignments).all() :
            return centroids

        #calculate new centroids
        centroid_sums=np.zeros((anchor_num, anchor_dim), np.float)
        for i in range(ann_num):
            centroid_sums[assignments[i]]+=ann_dims[i]
        for j in range(anchor_num):
            centroids[j] = centroid_sums[j]/(np.sum(assignments==j) + 1e-6)

        prev_assignments = assignments.copy()
        old_distances = distances.copy()

def gen_anchor(anno_path,image_path,num_anchors,labels,input_size):
    train_imgs, train_labels = parse_annotation(anno_path,image_path,labels)

    grid_w = input_size[0]/32
    grid_h = input_size[1]/32

    # run k_mean to find the anchors
    annotation_dims = []
    for image in train_imgs:
        cell_w = image['width']/grid_w
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cell_h = image['height']/grid_h

        for obj in image['object']:
            relative_w = (float(obj['xmax']) - float(obj['xmin']))/cell_w
            relatice_h = (float(obj["ymax"]) - float(obj['ymin']))/cell_h
            annotation_dims.append(tuple(map(float, (relative_w,relatice_h))))

    annotation_dims = np.array(annotation_dims)
    centroids = run_kmeans(annotation_dims, num_anchors)

    # write anchors to file
    print('\naverage IOU for', num_anchors, 'anchors:', '%0.2f' % avg_IOU(annotation_dims, centroids))
    print_anchors(centroids)

labels = ["fire","smoke"]
input_size = [224,224]
gen_anchor("/content/Annotations/","/content/JPEGImages/",5,labels,input_size)

config = {
        "model":{
            "type":                 "Detector",
            "architecture":         "MobileNet7_5",
            "input_size":           (224,224),
            "anchors":              [0.3591,0.51, 0.7362,1.62, 0.8835,0.87, 1.4712,2.04, 2.5843,2.7640],
            "labels":               ["fire","smoke"],
            "coord_scale" :     1.0,
            "class_scale" :     1.0,
            "object_scale" :    5.0,
            "no_object_scale" :   1.0
        },
        "weights" : {
            "full":             "",
            "backend":          "imagenet"
        },
        "train" : {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        "actual_epoch":          100,
        "train_image_folder":   "/content/JPEGImages/",
        "train_annot_folder":   "/content/Annotations/",
        "train_times":           5,
        "valid_image_folder":   "/content/images_v/",
        "valid_annot_folder":   "/content/annotations_v/",
        "valid_times":           1,
        "valid_metric":         "mAP",
        "batch_size":            8,
        "learning_rate":         1e-3,
        "saved_folder":       F"/content/M",
        "first_trainable_layer": "",
        "augumentation":        True,
        "is_only_detect" :    False
    },
    "converter" : {
        "type":          ["k210"]
    }
}


from tensorflow.python.client import device_lib
device_lib.list_local_devices()



from keras import backend as K
K.clear_session()
model_path = setup_training(config_dict=config)


%matplotlib inline
from keras import backend as K
K.clear_session()
setup_inference(config, model_path)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import sensor
import image
import lcd
import time
import KPU as kpu
from Corgi85 import corgi85

token = "L7jCMLfSl4DAkMY2hvIsgoODZej7CE6yFdIqfMxv1ix
lcd.init()
lcd.rotation(2)



#reset esp8285
corgi85.reset()
#print(dir(corgi85))

sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.set_windowing((224, 224))
sensor.set_vflip(1)
sensor.run(1)

count = 0
while(corgi85.wifi_check() == 0):
    print("WIFI Connecting")
    time.sleep(1)

print("\n\nWIFI Connected")

print("\n\nSet line Token:",token)
corgi85.LINE_setToken(token)  #set line Token

clock = time.clock()
classes = ('fire','smoke')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
task = kpu.load(0x200000)
a = kpu.set_outputs(task, 0, 7, 7, 35)
anchor = (0.3591,0.51, 0.7362,1.62, 0.8835,0.87, 1.4712,2.04, 2.5843,2.7640)
a = kpu.init_yolo2(task, 0.5, 0.3, 5, anchor)
while(True):
    clock.tick()
    img = sensor.snapshot()
    objects = kpu.run_yolo2(task, img)
    if objects:
        for obj in objects:
            img.draw_rectangle(obj.rect(),color=(0,255,0),thickness=3)
            img.draw_string(obj.x(), obj.y(), classes[obj.classid()], color=(0,255,0),scale=2)
            img.draw_string(obj.x(), obj.y()+12, '%.3f'%obj.value(), color=(0,255,0),scale=2)
    #img = img.resize(240,240)
    lcd.display(img)

    if objects:
        img2 = sensor.snapshot()
        for obj in objects:
            img.draw_rectangle(obj.rect(),color=(0,255,0),thickness=3)
            img.draw_string(obj.x(), obj.y(), classes[obj.classid()], color=(0,255,0),scale=2)
            img.draw_string(obj.x(), obj.y()+12, '%.3f'%obj.value(), color=(0,255,0),scale=2)
        corgi85.LINE_notifyPicture(img2, "ตรวจพบไฟไหม้ บริเวณ ห้องครัว : กรุณาตรวจสอบด่วน!")
kpu.deinit(task)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดโปรแกรมการแจ้งเตือนผ่านแอพพลิเคชั่นไลน์

```
import sensor
import image
import lcd
import time
import KPU as kpu
from Corgi85 import corgi85

token = "L7jCMLfSl4DAkMY2hvIsgoODZej7CE6yFdIqfMxv1ix
lcd.init()
lcd.rotation(2)



#reset esp8285
corgi85.reset()
#print(dir(corgi85))

sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.set_windowing((224, 224))
sensor.set_vflip(1)
sensor.run(1)

count = 0
while(corgi85.wifi_check() == 0):
    print("WIFI Connecting")
    time.sleep(1)

print("\n\nWIFI Connected")

print("\n\nSet line Token:",token)
corgi85.LINE_setToken(token)  #set line Token

clock = time.clock()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
classes = ('fire','smoke')
task = kpu.load(0x200000)
a = kpu.set_outputs(task, 0, 7, 7, 35)
anchor = (0.3591,0.51, 0.7362,1.62, 0.8835,0.87, 1.4712,2.04, 2.5843,2.7640)
a = kpu.init_yolo2(task, 0.5, 0.3, 5, anchor)
while(True):
    clock.tick()
    img = sensor.snapshot()
    objects = kpu.run_yolo2(task, img)
    if objects:
        for obj in objects:
            img.draw_rectangle(obj.rect(),color=(0,255,0),thickness=3)
            img.draw_string(obj.x(), obj.y(), classes[obj.classid()], color=(0,255,0),scale=2)
            img.draw_string(obj.x(), obj.y()+12, '%.3f'%obj.value(), color=(0,255,0),scale=2)
    #img = img.resize(240,240)
    lcd.display(img)

    if objects:
        img2 = sensor.snapshot()
        for obj in objects:
            img.draw_rectangle(obj.rect(),color=(0,255,0),thickness=3)
            img.draw_string(obj.x(), obj.y(), classes[obj.classid()], color=(0,255,0),scale=2)
            img.draw_string(obj.x(), obj.y()+12, '%.3f'%obj.value(), color=(0,255,0),scale=2)
        corgi85.LINE_notifyPicture(img2, "ตรวจพบไฟไหม้ บริเวณ ห้องครัว : กรุณาตรวจสอบด่วน!")
kpu.deinit(task)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้