# การตรวจจับโดรนด้วยกล้อง Fish Eye Lens และ Raspberry Pi 4

# DRONE DETECTION BY FISH EYE LENS AND RASPBERRY PI 4


**ปิยพัทธ์ ฤกษ์การันต์**

**พงศกร กุลดิลก**

**มหิศร วิชาธรรม**


**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ปีการศึกษา 2563**


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# DRONE DETECTION BY FISH EYE LENS AND RASPBERRY PI 4

**Piyapat Ruekkaran**

**Pongsakorn Kuldilok**

**Mahisorn Wichatham**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF**

**THE REQUIREMENT FOR THE DEGREE OF**

**BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING**

**SCHOOL OF ENGINEERING**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2020**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ปริญญานิพนธ์ปี การศึกษา 2563

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การตรวจจับโดรนด้วยกล้อง Fish Eye Lens และ Raspberry Pi 4

DRONE DETECTION BY FISH EYE LENS AND RASPBERRY PI 4


ผู้จัดทำ

1. นายปิยพัทธ์ ฤกษ์การันต์ รหัสนักศึกษา 60010617
2. นายนายพงศกร กุลดิลก รหัสนักศึกษา 60010634
3. นายมหิศร วิชาธรรม รหัสนักศึกษา 60010829

อาจารย์ที่ปรึกษา

(ผศ. ดร.สุรินทร์ กิตติธรกุล)


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

# การตรวจจับโดรนด้วยกล้อง Fish Eye Lens และ Raspberry Pi 4


| | | |
|---|---|---|
| นายปิยพัทธ์ | ฤกษ์การันต์ | 60010617 |
| นายพงศกร | กุลดิลก | 60010634 |
| นายมหิศร | วิชาธรรม | 60010829 |
| ผศ. ดร.สุรินทร์ | กิตติธรกุล | อาจารย์ที่ปรึกษา |

ปีการศึกษา 2563



**บทคัดย่อ**

โครงงานนี้จัดทำขึ้นเพื่อศึกษาและออกแบบการพัฒนาระบบตรวจจับโดรนขนาดกลางโดยใช้เทคนิคการเรียนรู้ของเครื่อง เพื่อเป็นระบบที่ใช้ในการแก้ปัญหาโดรนบินรุกล้ำ เข้าไปในพื้นที่หวงห้ามหรือสถานที่ต้องการความปลอดภัยสูง เช่น สนามบิน ซึ่งอาจเกิดอันตรายต่อเครื่องบินได้ โดยโครงงานนี้ต้องการให้ระบบสามารถทำงานบน Raspberry Pi 4 ได้ เพราะต้องการให้ระบบติดตั้งง่าย และมีราคาถูก ทั้งนี้ระบบยังมีส่วนติดต่อกับผู้ใช้ที่สามารถตรวจดูภาพวีดีโอผลลัพธ์ที่ได้ทำการประมวลผลแล้ว และผู้ใช้จะ ได้รับการแจ้งเตือนหากมีการตรวจจับโดรนได้ ซึ่งระบบจะส่งรูปภาพและข้อมูลช่วงเวลาที่ตรวจจับได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

# Drone Detection by Fish Eye Lens and Raspberry Pi 4


| | | |
|---|---|---|
| Mr. Piyapat | Ruekkaran | 60010617 |
| Mr. Pongsakorn | Kuldilok | 60010634 |
| Mr. Mahisorn | Wichatham | 60010829 |
| Asst.Prof.Dr. Surin | Kittitornkun | Advisor |

Academic Year 2020



## ABSTRACT

This project is initiated to study and to develop Drone detection using vision-based machine learning techniques. The purpose of this system is to solve problems of drone invasion into restricted or confined areas; for example, airports because drones can be harmful for aircrafts. The system is so light-weight and able to exploit on Raspberry Pi 4 because it is low-priced and easy for installation. Furthermore, the user interface is web-based which allows the users to access live streaming video for human inspections. Also, it will notify whenever the system detects a drone invasion with captured images in real time.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

# กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาช่วยเหลือ แนะนำ ให้คำปรึกษา ตลอดจนการตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างดียิ่งจาก ผศ. ดร.สุรินทร์ กิตติธรกุล อาจารย์ที่ปรึกษาโครงงานนี้

ขอขอบคุณอาจารย์และบุคลากรต่าง ๆ ในสาขาวิศวกรรมคอมพิวเตอร์ที่ให้ความรู้ คำแนะนำ และช่วยเหลือในการทำโครงงานครั้งนี้มาโดยตลอด ขอขอบคุณรุ่นพี่และเพื่อน ๆ ทุกคน ที่ให้คำปรึกษา คำแนะนำ และคอยแบ่งปันความรู้ต่าง ๆ ที่ทำให้โครงงานนี้เสร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอขอบคุณ บิดา มารดา และครอบครัว ที่ได้เลี้ยงดู อบรม สั่งสอน และให้การสนับสนุนในทุก ๆ ด้าน และคอยเป็นกำลังใจให้เสมอมา

ปิยพัทธ์ ฤกษ์การันต์
พงศกร กุลดิลก
มหิศร วิชาธรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# สารบัญ

หน้า


บทคัดย่อภาษาไทย ..... I

บทคัดย่อภาษาอังกฤษ ..... II

กิตติกรรมประกาศ ..... III

สารบัญ ..... IV

สารบัญตาราง ..... VII

สารบัญรูป ..... VIII

บทที่ 1 บทนำ ..... 1

1.1 ความเป็นมาของปัญหา ..... 1

1.2 วัตถุประสงค์ของโครงงาน ..... 2

1.3 ประโยชน์ที่คาดว่าจะได้รับ ..... 2

1.4 ขอบเขตของการทำงาน ..... 2

1.5 วิธีการดำเนินงาน ..... 2

1.6 ข้อจำกัดของโครงงาน ..... 3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง ..... 4

2.1 งานวิจัยที่เกี่ยวข้อง (Related Works) ..... 4

2.2 Fisheye Lens ..... 5

2.3 Library TensorFlow ..... 8

2.4 Shinobi ..... 9

2.5 Machine Learning ..... 10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

# สารบัญ (ต่อ)

หน้า

2.6 Deep Learning ..... 11

2.7 Convolutional Neural Network ..... 14

2.8 Object Detection Model ..... 16

2.9 Raspberry Pi ..... 19

บทที่ 3 การออกแบบและพัฒนา ..... 22

3.1 โครงสร้างของระบบ ..... 22

3.2 ผังงาน (Flowchart) ของระบบ ..... 23

3.3 การพัฒนาโปรแกรมตรวจจับโดรน ..... 24

3.4 การพัฒนาส่วนติดต่อกับผู้ใช้ (User Interface) ..... 27

3.5 การพัฒนาระบบแจ้งเตือนผู้ใช้ ..... 27

บทที่ 4 การทดลองและวิเคราะห์ผลการทดลอง ..... 29

4.1 การทดลองเปรียบเทียบประสิทธิภาพของ Pre-trained model ..... 29

4.2 การทดลองสตรีมภาพวีดีโอผลลัพธ์บน Web Application ..... 37

4.3 การทดลองระบบแจ้งเตือนและระบบบันทึกไฟล์ผลลัพธ์ ..... 38

บทที่ 5 บทสรุปและข้อเสนอแนะ ..... 40

5.1 บทสรุป ..... 40

5.2 ปัญหาและอุปสรรค ..... 40

5.3 แนวทางการพัฒนาต่อ ..... 40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

# สารบัญ (ต่อ)

หน้า

บรรณานุกรม ..... 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# สารบัญตาราง

**ตาราง** **หน้า**

2.1 Features and Specs of VSTARCAM C61S .....5
2.2 คำอธิบายส่วนประกอบของนิวรอน (Neuron).....13
2.3 Features and Specs of Raspberry Pi 4 Model B.....20
4.1 คำอธิบายคำที่ใช้ในผลการทดลอง.....30
4.2 ผลการทดสอบเปรียบเทียบประสิทธิภาพ.....30
4.3 Confusion Matrix ของ SSD MobileNet v2 320x320.....35
4.4 Confusion Matrix ของ SSD MobileNet v2 FPNLite 640x640.....35
4.5 Confusion Matrix ของ SSD ResNet50 v1 FPN 640x640.....35
4.6 Confusion Matrix ของ EfficientDet D0.....35
4.7 ผลการคำนวณประสิทธิภาพโมเดลแบบแยกคลาส .....36
4.8 ผลการคำนวณประสิทธิภาพโมเดลด้วย Macro-average Method.....36
4.9 ผลการคำนวณประสิทธิภาพโมเดลด้วย Micro-average Method.....36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# สารบัญรูป

**รูป** **หน้า**

2.1 ภาพจากเลนส์ประเภท Circular Fisheye....7
2.2 สัญลักษณ์ TensorFlow....8
2.3 หน้าต่างแสดงผล TensorBoard....9
2.4 สัญลักษณ์ Shinobi....9
2.5 ประเภทของการเรียนรู้ของ Machine Learning เบื้องต้น....10
2.6 โครงสร้างของ Artificial neural network....12
2.7 โครงสร้างของนิวรอน (Neuron)....12
2.8 การเคลื่อนลงตามความชันไปหาจุดต่ำสุด....14
2.9 เปรียบเทียบโครงสร้างของ Simple Neural Network และ Deep Neural Network....14
2.10 โครงสร้างของ CNN....15
2.11 ตัวอย่างผลลัพธ์การทำ Object Detection แบบ Multiple Objects....16
2.12 สถาปัตยกรรมของ SSD....17
2.13 สถาปัตยกรรมของ MobileNet....17
2.14 สถาปัตยกรรมของ ResNet....18
2.15 ลำดับการทำงานของ FPN....18
2.16 สถาปัตยกรรมของ EfficientDet....19
2.17 RASPBERRY PI 4 MODEL B, CORTEX-A72, 8GB RAM....20
3.1 โครงสร้างของระบบตรวจจับโดรนด้วยกล้อง Fish Eye Lens และ Raspberry Pi 4....22
3.2 Flowchart แสดงการทำงานในภาพรวมของระบบ....23
3.3 ตัวอย่างรูปจาก Dataset....24
3.4 โปรแกรม LabelImg....25
3.5 TensorFlow 2 Object Detection Model....25
3.6 หน้าต่างแสดงผล TensorBoard....26
3.7 หน้าต่างตั้งค่าการเชื่อมต่อสำหรับรับข้อมูลจากโปรแกรมตรวจจับโดรน....27
3.8 LINE Notify API....28
4.1 ผลการทดลองสตรีมภาพวีดีโอผลลัพธ์บน Web application....37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# สารบัญรูป (ต่อ)

**รูป** **หน้า**

4.2 ผลการแจ้งเตือนผู้ใช้ผ่าน Line Application ..... 38
4.3 ผลการบันทึกไฟล์ภาพที่ตรวจจับได้บน File Server ..... 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาของปัญหา

อากาศยานไร้คนขับ (Unmanned aerial vehicle) หรือที่รู้จักกันอย่างแพร่หลายในนามของ “โดรน” คือ อากาศยานที่ไม่มีผู้ขับอยู่บนตัวอากาศยาน สามารถควบคุมระยะไกลผ่านโปรแกรมควบคุมการบิน หรือขับเคลื่อนอัตโนมัติด้วยระบบฝังตัวซึ่งจะเป็นระบบที่มีความซับซ้อนมากกว่า ในอดีตโดรนถูกนำมาใช้ในงานทางด้านการทหารเป็นหลัก สามารถทำภารกิจที่มีความเสี่ยงสูงได้ โดยที่นักบินไม่ต้องเสี่ยงชีวิตในการทำภารกิจนั้น ทั้งยังสามารถใช้โดรนในภารกิจสอดแนมหรือโจมตีได้อีกด้วย ซึ่งในปัจจุบัน โดรน เริ่มนำมาใช้กันอย่างแพร่หลายในจุดประสงค์อื่นสำหรับภาคพลเรือน นอกจากทางด้านทหาร เช่น การค้นหา, ช่วยเหลือผู้ประสบภัย, การตรวจสภาพอากาศ, การตรวจสภาพจราจร ตลอดจนถึง โดรนส่วนตัว ที่ถูกออกแบบให้มีขนาดเล็ก มีความคล่องตัว และน้ำหนักเบา ทำให้เป็นที่นิยมสำหรับคนทั่วไป สามารถใช้ประโยชน์ได้หลายรูปแบบ เช่น รับส่งพัสดุขนาดเล็ก, สำรวจสภาพพื้นที่ หรือใช้เพื่อสื่อความบันเทิงอย่างการถ่ายภาพ และวิดีโอ อย่างไรก็ตามการใช้งานอย่างแพร่หลายเหล่านี้จะมาพร้อมกับปัญหาต่าง ๆ ทั้งในเรื่องของการสร้างปัญหาเสียงรบกวนจากเครื่องยนต์ การสร้างปัญหาฝุ่น และสิ่งของกระเด็นจากแรงลมใบพัด การละเมิดสิทธิส่วนบุคคลโดยการบินเข้าไปในพื้นที่ส่วนตัว หรือในพื้นที่หวงห้าม เช่น สถานที่ราชการ หน่วยงานเอกชน การบินเข้าไปในสถานที่ต้องการความปลอดภัยสูง เช่น สนามบิน อาจทำให้เกิดอันตรายต่อเครื่องบิน และผู้โดยสารได้ นอกจากนี้โดรนส่วนมากจะติดตั้งกล้องบันทึกภาพ โดยสามารถถ่ายภาพมุมสูงเพื่อการสอดแนมได้ ซึ่งอาจเป็นภัยต่อความมั่นคงหากมีการบินเข้าไปในพื้นที่ราชการ หรือพื้นที่ที่เป็นความลับ

จากปัญหาดังกล่าว หน่วยงานที่ต้องการความปลอดภัยสูงอย่างกองทัพ และสนามบินจำเป็นต้องมีการตรวจจับ พร้อมทั้งป้องกันโดรนที่บินรุกล้ำเข้ามาในพื้นที่ตลอดเวลา ซึ่งระบบตรวจจับโดรนนี้จะสามารถตรวจจับ และจำแนกโดรนจากภาพที่ได้จากกล้องวีดิโอ โดยโครงงานนี้จะใช้บอร์ด Raspberry Pi 4 เป็นตัวประมวลผล เพราะต้องการให้ระบบสามารถติดตั้งได้ง่าย ราคาถูก และประสิทธิภาพสูง นอกจากนี้ระบบยังมีส่วนแสดงผลสำหรับผู้ใช้ ให้ผู้ใช้สามารถตรวจสอบภาพจากกล้องวีดีโอ และได้รับการแจ้งเตือนหากพบการตรวจจับโดรน เป็นการลดภาระของเจ้าหน้าที่ ทำให้ไม่จำเป็นต้องตรวจสอบด้วยตนเองตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

## 1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาระบบตรวจจับโดรนอัตโนมัติที่สามารถตรวจจับ และจำแนกโดรนที่บินเข้ามาในพื้นที่ว่าเป็นรุ่น Sky hunter X8 (4 ใบพัด) หรือรุ่น TYH scout (6 ใบพัด) ได้
2) เพื่อพัฒนาระบบตรวจจับโดรนอัตโนมัติที่ติดตั้งได้ง่าย ราคาถูก และสามารถนำไปใช้งานได้จริง
3) เพื่อศึกษาองค์ความรู้ในเรื่องของ Artificial Intelligence ซึ่งเป็นเทคโนโลยีที่ใช้ในการจำแนกโดรน
4) เพื่อศึกษาหา Machine Learning Model ที่มีความเหมาะสมในการจำแนกโดรน

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้เกี่ยวกับการทำงานของ Artificial Intelligence
2) ได้รับความรู้ และทักษะในการใช้งาน TensorFlow
3) ได้รับความรู้ในการสอน และทดสอบโมเดลเพื่อทำงานด้าน Object detection
4) สามารถนำระบบตรวจจับโดรนไปพัฒนาต่อเพื่อให้ใช้งานได้มีประสิทธิภาพมากขึ้น

## 1.4 ขอบเขตของการทำงาน

1) จำแนก และตรวจจับโดรนรุ่น Sky hunter X8 (4 ใบพัด) และรุ่น TYH scout (6 ใบพัด) จากภาพวีดีโอ
2) แสดงผลการตรวจจับผ่าน Web Application
3) แจ้งเตือนผู้ใช้เมื่อพบการตรวจจับโดรนด้วย Line Chat Bot

## 1.5 วิธีการดำเนินงาน

1) กำหนดวัตถุประสงค์และขอบเขตการทำงานของโครงงาน
2) ศึกษาเทคโนโลยีต่าง ๆ ที่ใช้ในการพัฒนาระบบ เช่น TensorFlow, Convolution Neural Network, Raspberry Pi, Shinobi
3) ออกแบบและวิเคราะห์ระบบจากข้อมูล เนื้อหา และเอกสารที่เกี่ยวข้องที่ได้ทำการรวบรวม
4) ศึกษาและพัฒนาระบบตรวจจับโดรนโดยใช้ TensorFlow แสดงผลผ่าน Shinobi และแจ้งเตือนผู้ใช้เมื่อพบการตรวจจับโดรนด้วย Line Chat Bot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

## 1.6 ข้อจำกัดของโครงงาน

1) สามารถจำแนกและตรวจจับได้เฉพาะโดรนรุ่นที่กำหนด คือ Sky hunter X8 (4 ใบพัด) และรุ่น TYH scout (6 ใบพัด)
2) สามารถตรวจจับโดรนที่กำลังบินด้วยความสูงไม่เกิน 10 เมตร
3) คุณภาพของภาพในตอนกลางคืนไม่ดีพอ จึงทำให้มีโอกาสผิดพลาดในการตรวจจับและจำแนกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# บทที่ 2
# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

## 2.1 งานวิจัยที่เกี่ยวข้อง (Related Works)

### 2.1.1 Using Deep Networks for Drone Detection

เป็นงานวิจัยในการตรวจจับโดรนโดยใช้ Deep learning model คือ YOLO V2 โดยใช้ Dataset ที่มีความหลากหลาย และจำนวนที่มากพอ แล้วทำการทดลองเทรนโมเดล และบันทึกผล เพื่อวัด ประสิทธิภาพของโมเดลที่ได้ สำหรับในอนาคตจะทำการพิจารณาในเรื่องของเวลา ให้ใช้เวลาใน การประมวลผลน้อยลง [ปีที่พิมพ์: 2017, ผู้เขียน: Cemal Aker, Sinan Kalkan]

### 2.1.2 Image Processing Based Proposed Drone for Detecting and Controlling Street Crimes

วัตถุประสงค์ของงานวิจัยนี้ คือ การลดความเสี่ยงในการเกิดเหตุอาชญากรรม โดยการใช้โดรนเพื่อ บินสำรวจและใช้เทคโนโลยี Image Processing ในการตรวจหาวัตถุที่คาดว่าจะเป็นอันตราย โดย ใช้ shape detection algorithm ในการตรวจหาอาวุธ และใช้ SVM ในการ Classification [ปีที่ พิมพ์: 2017, ผู้เขียน: Shahid Karim, Ye Zhang, Asif Ali Laghari, Muhammad Rizwan Asif]

### 2.1.3 Panoramic UAV Surveillance and Recycling System Based on Structure-Free Camera Array

งานวิจัยนี้สร้างระบบเฝ้าระวังโดรน แบบพาโนรามาที่ใช้กล้องหลายตัวแบบไม่มีรูปแบบการจัดวาง ออกแบบอัลกอริธึมการปรับเทียบด้วยแหล่งกำเนิดแสงเลเซอร์แบบไดนามิก ระบุตำแหน่งโดรนบนพิกัด 3 มิติ และอัลกอริธึมการรีไซเคิลอัตโนมัติโดยใช้กลุ่มกล้อง fisheye [ปีที่ พิมพ์: 2019, ผู้เขียน: Tao Yang, Zhi Li, Fangbing Zhang, Bolin Xie, Jing Li, Linfeng Liu]

### 2.1.4 Investigation of Fish-Eye Lenses for Small-UAV Aerial Photography

นำเสนอการใช้โดรนกับเลนส์ fisheye สำหรับการถ่ายภาพทางอากาศและการทำแผนที่ อธิบายรายละเอียดตั้งแต่การบิดโค้งภาพของเลนส์ fisheye ไปจนถึงการบันทึกภาพ ระบบดังกล่าวทดสอบเปรียบเทียบประสิทธิภาพกับแพลตฟอร์มโดรนสองขนาด [ปีที่ พิมพ์: 2008, ผู้เขียน: Alex Gurtner, Duncan G. Greer, Richard Glassock, Luis Mejias, Rodney A. Walker, Wageeh W. Boles]

### 2.1.5 Comparison and Calibration of Mobile Phone Fisheye Lens and Regular Fisheye Lens via Equidistant Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

วัตถุประสงค์หลักเพื่อทดสอบเปรียบเทียบการปรับเทียบมาตรฐานของเลนส์ Fisheye ที่ใช้กับโทรศัพท์มือถือ iPhone 4S และเลนส์ fisheye ที่ใช้กับกล้อง Nikon โดยพิจารณาจากรุ่นที่มี

ระยะเท่ากัน ผลการศึกษานี้ชี้ให้เห็นกระบวนการปรับเทียบล่วงหน้าของฮาร์ดแวร์ประเภทนี้ในการทดสอบ [ปีที่ พิมพ์: 2016, ผู้เขียน: Cumhur Sahin]

**2.1.6 Drone Detection by Visual Machine Learning**

ออกแบบและพัฒนาระบบตรวจจับโดรนโดยใช้เทคนิคการเรียนรู้ของเครื่อง โดยให้ระบบทำงานบน Raspberry Pi 3 มีส่วนติดต่อกับผู้ใช้ที่สามารถตรวจดูภาพวีดีโอผลลัพธ์ที่ได้ทำการประมวลผลแล้ว และมีการแจ้งเตือนส่งรูปภาพและข้อมูลช่วงเวลาที่ตรวจจับได้หากมีการตรวจจับพบโดรน [ปีที่ พิมพ์: 2018, ผู้เขียน: ธนภัทร แจ่มแจ้ง, มงคล สมานญา]

## 2.2 Fisheye Lens

เลนส์ตาปลา (Fisheye Lens) เป็นเลนส์ที่มีลักษณะคล้ายตาของปลาที่ว่ายอยู่ในน้ำ สามารถเก็บภาพได้ กว้างมากถึง 180 องศา มากกว่าเลนส์ทุกชนิด เลนส์ Fisheye เป็นเลนส์มุมกว้างพิเศษ มุมมองของเลนส์ Fisheye มักจะอยู่ระหว่าง 100 และ 180 องศา หรือบางเลนส์ก็ออกแบบมาให้สามารถเก็บภาพมุมกว้าง ได้มากกว่า 180 องศา เช่น เลนส์ Fisheye ขนาดทางยาวโฟกัส 8 มิลลิเมตร สามารถนำไปถ่ายภาพท้องฟ้าได้ ทั่วทั้งท้องฟ้า แทนที่จะสร้างภาพด้วยมุมมองเส้นตรง เลนส์ฟิชอายใช้การทำแผนที่พิเศษ ซึ่งภาพที่ได้จะมี ลักษณะบิดเบือนโค้งงอนูนเป็นวงกลม

Fisheyes lens ในปัจจุบันแบ่งออกเป็น 2 ประเภทหลัก ๆ ได้แก่ • Circular Fisheye เป็นเลนส์ที่ให้ภาพเป็นวงกลม ซึ่งมุมภาพทั้ง 4 จะเป็นสีดำ เพราะถูกตัด ด้วยเฟรมของกล้องบริเวณหน้าระนาบฟิล์ม โดยมีความกว้าง 180 องศา • Full Frame Fisheye เป็นเลนส์ที่ให้ภาพแบบเต็ม ๆ ไม่มีการถูกตัดขอบแบบ Circular Fisheye จึงได้รับความนิยมใช้มากกว่า เพราะสามารถถ่ายภาพในพื้นที่จำกัดให้ออกมากว้าง ได้หลักการของเลนส์ฟิชอายชนิดนี้คือ เมื่อไม่มีการถูกตัดขอบแบบ Circular Fisheye ความ กว้าง 180 องศาจึงอยู่ในแนวทแยง ส่วนความกว้างของด้านยาวรับได้160 องศา และด้าน กว้าง 100 องศาเท่านั้น ดังนั้นเมื่อเปรียบเทียบเลนส์ฟิชอาย 2 ชนิดดังกล่าว สามารถสรุปได้ว่า Circular Fisheye ให้มุม ได้กว้างกว่า แต่ Full Frame Fisheye เป็นที่นิยมมากกว่า เพราะไม่มีการตัดขอบเหมือน Circular Fisheye ซึ่งทำให้ภาพที่ได้นั้น ดูเหมือนมีมุมลึกมากกว่า

**ตาราง 2.1 Features and Specs of VSTARCAM C61S**

| | | |
|---|---|---|
| System | Operation system | Embedded Linux OS |
| | Online visitor | Support 4 visitors at the same time |
| Image Capture | Image sensor | 1/2.4 inch 1536p Progressive Scan CMOS |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 2.1 Features and Specs of VSTARCAM C61S (ต่อ)**

| | | |
|---|---|---|
| Image Capture | Minimum Illumination | 0.3~0.5Lux (Color mode) ,0Lux (B8W Mode) |
| | Lens | 2.3mm@F24 |
| | View Angle | 180° |
| | Nightvision | Dual Fiter,IR-Cut, 3pcs 850nm Infrared array LED; IR irradiation distance:5~10m |
| Video | Encode format | H.264 main profile@levels:4.0/Motion-JPEG |
| | Three Stream | Main Stream:1536P (1536x1536) @15fps |
| | Bit Rate | 128-~4096kbps |
| | Maximum frame rate | 15fps |
| | Image adjustment | Contrast, Brightness, Saturation, Hue |
| Audio | Input | Built-in -38db microphone |
| | Output | Built-in (8Ω1W) speaker |
| | Sampling frequency / width | 8KHz/16bit |
| | Compression standard | ADPCM/32kbps |
| Network | Ethernet | 10Base-T / 100Base-TX, RJ45 connector |
| | Protocols | TCP/IP,HTTP,TCP,UDP,DHCP,DNS,NTP,RTSP,P2P etc |
| | Wireless LAN | IEEE802.11b/g/n |
| | Wireless LAN Frequency | 2.4~ 2.4835GHz |
| | Wireless LAN security | 64/128-bit WEP/WPA-PSK/WPA2-PSK data encryption |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 2.1 Features and Specs of VSTARCAM C61S (ต่อ)**

| | | |
|---|---|---|
| Memory | Socket | TF Push-push socket,Maximum support 128GB; Cloud Storage |
| Alarm | Alarm Trigger | Motion detection |
| Physical Environmental | Rated voltage | DC12V±5% |
| | Power consumption | Related power:6.24W(IR on) |
| | Operating condition | Temperature:-10~50 C°,humidity:90% |
| | Weight | Gross: THD g(Note: in kind prevail) |
| | Package size | 196X183X110mm (L'W'H) |

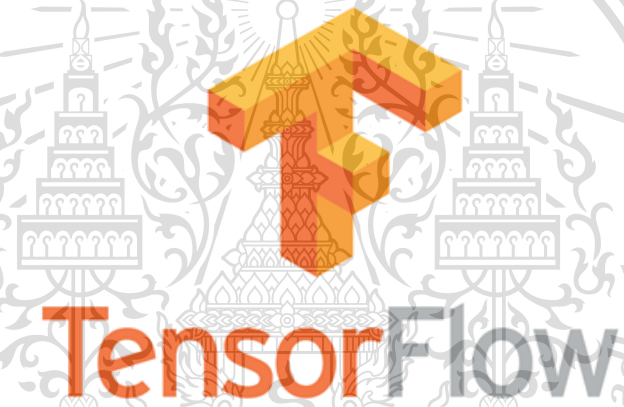
**รูป 2.1 ภาพจากเลนส์ประเภท Circular Fisheye**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

## 2.3 Library TensorFlow

TensorFlow เป็น Python Library แบบ Open-Source ที่ถูกพัฒนาขึ้นและร่วมมือกันระหว่าง Brain Team ของ Google TensorFlow ถูกใช้สำหรับเขียน Algorithms ใหม่ที่เกี่ยวข้องกับ Tensor Operations จำนวนมาก เนื่องจาก Neural Network สามารถแสดงเป็น Computational Graphs ได้ง่าย จึงสามารถนำไปใช้เป็น ชุดของการ Operation บน Tensors โดยใช้ TensorFlow ยิ่งไปกว่านั้น Tensors เหล่านี้คือ N-Dimensional Matrices ที่จะใช้สำหรับอธิบาย/แสดงข้อมูลของคุณ

TensorFlow มีความแตกต่างจาก Numpy และ Libraries อื่น ๆ โดยจะช่วยให้เห็น Graph หลายๆ ส่วนได้ง่ายมากขึ้น ซึ่งหากคุณใช้ Numpy หรือ SciKit ก็จะไม่เจอกับ Option นี้ มันสามารถช่วย Train CPU ได้เป็นอย่างดี รวมถึง GPU ของ Distributed Computing นอกจากนี้ TensorFlow ยังถูกใช้ใน Application ส่วนมากของ Google สำหรับการทำ Machine Learning



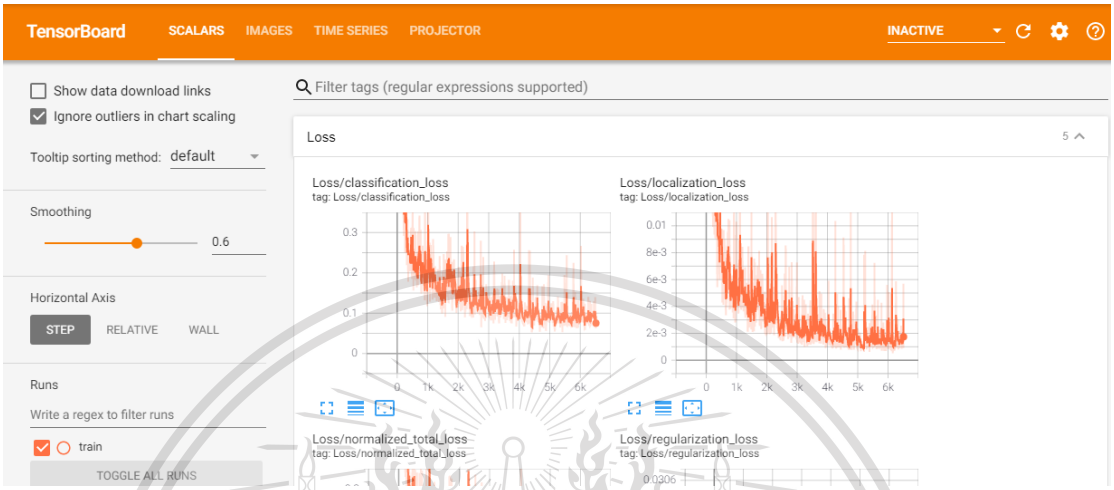

**รูป 2.2 สัญลักษณ์ TensorFlow**

API ของ TensorFlow ที่ใช้ในโครงงานนี้คือ TensorFlow Object Detection API โดยเป็น open source framework ในการทำงาน Machine Learning ที่เน้นในเรื่องของการ Classification และ Localizing วัตถุโดยมี Model ต่าง ๆ มากมายทั้งที่เป็นทางการและ Model ที่กำลังทำการวิจัยโดยทีมของ TensorFlow อยู่ซึ่งเราสามารถใช้ Model ที่ทาง TensorFlow มีมาประยุกต์ใช้กับงานของเรา โดยเราสามารถทำ Transfer learning คือเทคนิคที่นำบางส่วนของ Model ที่มีการสอนและปรับค่าน้ำหนักพารามิเตอร์ต่าง ๆ ไว้แล้วมาสร้างเป็น Model ใหม่ เพราะการที่จะต้องเริ่มสอน Model ใหม่ จากศูนย์ต้องใช้จำนวนข้อมูลในการสอนที่มาก ใช้เวลานาน และการประมวลผลที่มากขึ้น

TensorBoard เป็นเว็บแอพพลิเคชั่นสำหรับตรวจสอบ แสดงผล และทำให้เข้าใจการทำงานของ TensorFlow ที่กำลังทำอยู่ เพราะในเรื่องของ Neural Network นั้นมีความซับซ้อน เพื่อให้ง่ายต่อการทำความเข้าใจแก้ปัญหาและเพิ่มประสิทธิภาพโปรแกรม TensorFlow ทาง Google จึงได้พัฒนาชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

เครื่องมือนี้โดยสามารถใช้ TensorBoard เพื่อสร้างภาพกราฟ TensorFlow ที่สร้างขึ้นวางแผนแปลงข้อมูลเมตริกเชิงปริมาณเกี่ยวกับการเรียกใช้กราฟของผู้ใช้งานและแสดงข้อมูลเพิ่มเติม เช่น ภาพที่ส่งผ่าน ทำให้ง่ายต่อการตรวจสอบผลการเทรน




**รูป 2.3 หน้าต่างแสดงผล TensorBoard**

## 2.4 Shinobi

Shinobi เป็น Open-Source CCTV Solution ที่ถูกเขียนด้วย Node.JS และถูกคิดค้นออกแบบให้สามารถรองรับการใช้งานพร้อมกันได้หลายบัญชี โดยสามารถที่จะทำการติดต่อสื่อสารระหว่าง Web server กับ Client ได้แบบ Real Time และ Shinobi สามารถบันทึกวีดิโอจาก IP Cameras หรือจากกล้องทั่วไปได้

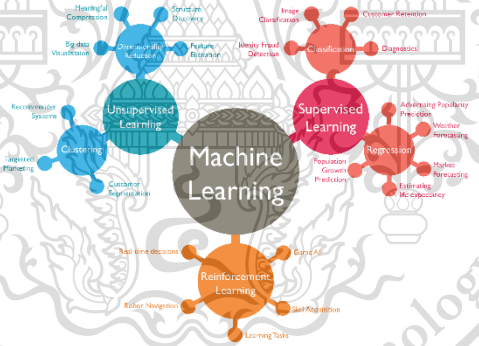
**รูป 2.4 สัญลักษณ์ Shinobi**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกา ตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

## 2.5 Machine Learning

การเรียนรู้ของเครื่อง (Machine earning - ML) เป็นการศึกษาอัลกอริทึมของคอมพิวเตอร์ที่มีการพัฒนา การเรียนรู้ของเครื่องถูกมองว่าเป็นส่วนหนึ่งของปัญญาประดิษฐ์ โดยอัลกอริทึมสร้างแบบจำลองทางคณิตศาสตร์จากข้อมูลตัวอย่าง เพื่อที่จะคาดการณ์หรือตัดสินใจได้อย่างชัดเจน เช่น ให้โปรแกรม ทำนายว่าวันนี้หิมะจะตกหรือเปล่า โดยการป้อนข้อมูล ความชื้น อุณหภูมิ แรงลม และนำไปวิเคราะห์ว่า หากมีรูปแบบดังกล่าว ต้องตอบสนองอย่างไร ซึ่งในการสร้างปัญญาประดิษฐ์แบบดั้งเดิม ผู้เขียนโปรแกรมจะเป็นคนกำหนดเงื่อนไขทุก อย่างไว้ ว่าถ้าเจอข้อมูลแบบนี้ต้องตอบสนองอย่างไร แต่ปัญญาประดิษฐ์ลักษณะนี้จะไม่สามารถทำอะไร นอกเหนือจากเงื่อนไขที่ผู้เขียนโปรแกรมกำหนดไว้ได้ ซึ่งจะเป็นการดี หากมันสามารถเรียนรู้เองได้ เมื่อเจอรูปแบบ ที่ไม่เคยเจอมาก่อน มันก็จะสามารถทำนายได้ว่าควรตอบสนองอย่างไร จากการเรียนรู้ของมัน จึงเป็นที่มาของ การ เรียนรู้ด้วยเครื่อง

Machine Learning หรือ การเรียนรู้ด้วยเครื่อง เป็นเทคนิคหนึ่งที่ใช้ในการสร้างปัญญาประดิษฐ์ ซึ่ง ปัญญาประดิษฐ์ไม่จำเป็นต้องใช้การเรียนรู้ด้วยเครื่องเสมอไป แต่หากใช้การเรียนรู้ของเครื่องในการสร้างก็จะทำให้ ปัญญาประดิษฐ์มีประสิทธิภาพมากขึ้น




**รูป 2.5 ประเภทของการเรียนรู้ของ Machine Learning เบื้องต้น**

### 2.5.1 เทคนิคของ Machine Learning

#### 2.5.1.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)

คือ การที่เราต้องคอยแยกประเภทหรือบอกผลลัพธ์ที่ควรจะเป็นไว้ล่วงหน้า จากนั้นใช้ข้อมูลตัวอย่างในการฝึกสอนนำไปผ่านอัลกอริทึมสำหรับสร้างโมเดลที่ไว้สำหรับคาดการณ์ผลลัพธ์ต่าง ๆ ที่จะเกิดขึ้น เมื่อมีการเรียนรู้เกิดขึ้นแล้ว ให้นำข้อมูลใหม่ที่เครื่องมือไม่เคยเห็นทำการ input เข้าไป เครื่องมือ Machine Learning จะสามารถคาดการณ์ผลลัพธ์ที่เกิดขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**2.5.1.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)**

การเรียนรู้แบบนี้เราจะป้อนข้อมูลขาเข้า เท่านั้น ไม่ ป้อนข้อมูลขาออก โดยโปรแกรมจะทำการเรียนรู้ด้วยตัวเอง ตัวอย่างเช่น การแบ่งกลุ่มสีของเม็ดสีน้ำเงินและสีแดง ตามค่าสี หากใช้เทคนิคการเรียนรู้แบบนี้ โปรแกรมจะสามารถแบ่งกลุ่มสีออกมาได้เป็นสองกลุ่มคือกลุ่มของสีน้ำเงินและกลุ่มของสีแดง ซึ่งหากเราหยิบเม็ดสีแดงมาหนึ่งเม็ดให้โปรแกรมทายว่าเม็ดสีนี้เป็นสีอะไร โปรแกรมจะตอบได้เพียงว่า เม็ดสีนี้อยู่ในกลุ่มที่สองคือกลุ่มของสีแดง แต่ไม่สามารถบอกได้ว่า สีนี้เป็นสีแดงเพราะไม่ได้สอน วิธีการแบ่งกลุ่มข้อมูลด้วยการเรียนรู้แบบไม่มีผู้สอนนั้นมีชื่อเรียกว่า การแบ่งกระจุกข้อมูล (data clustering)

**2.5.1.3 การเรียนรู้แบบเสริมแรง (Reinforcement Learning)**

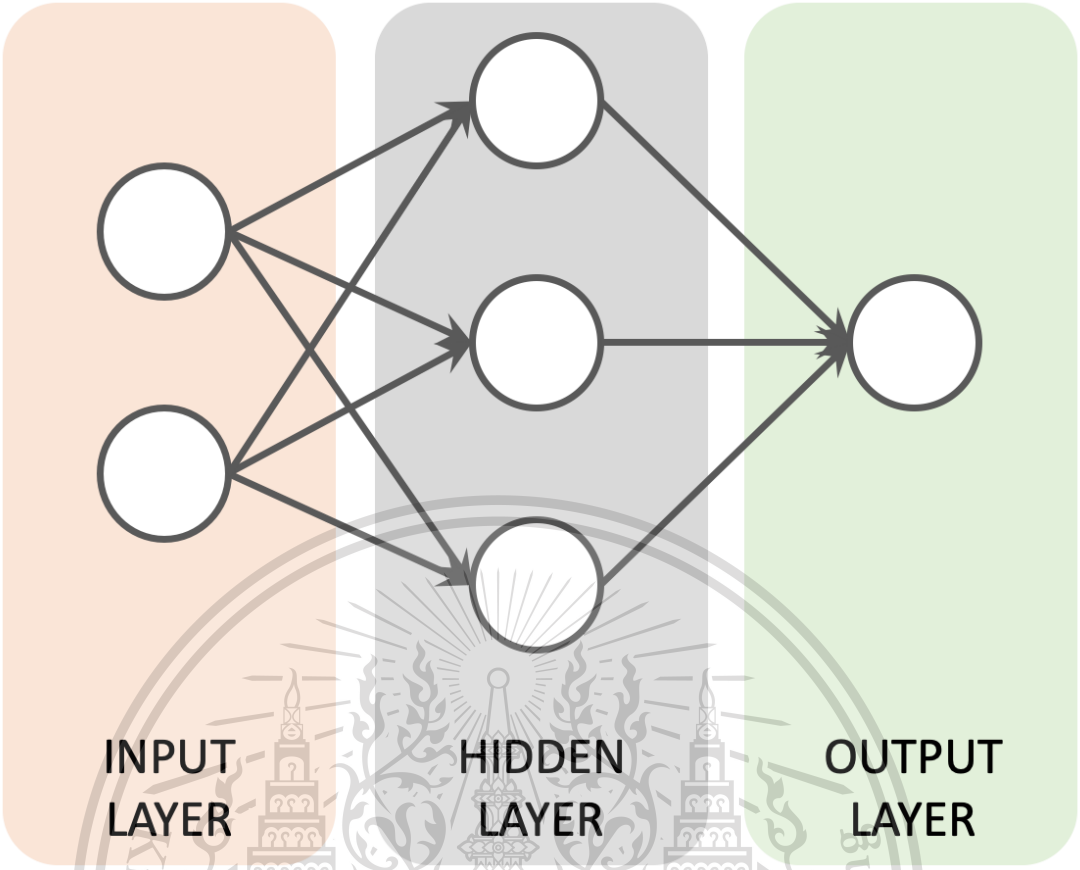
คือ เครื่องมือรูปแบบนี้เป็น Artificial Intelligence (AI) มากที่สุด เพราะจะเรียนรู้และเปลี่ยนไปตามสิ่งแวดล้อมตัวอย่างกรณีเป็นข่าวไปทั่วโลกคือ ระบบ AI ชื่อว่า“Alpha Go”ที่พัฒนาโดยบริษัท Deep Mind ของ Google นั้น โดย Alpha Go ได้ทำการพัฒนาและศึกษารูปแบบการเรียนของเซียนโกะทั่วโลก เพื่อพัฒนาเป็นรูปแบบการเล่นของตัวเอง ฝึกฝนเพื่อเพิ่มความสามารถ โดยฝึกกับตัวเองในรูปแบบการเล่นต่าง ๆ นับล้านครั้ง ทำให้สามารถวิเคราะห์เกมและเล่นเก่งมากขึ้นจนสามารถชนะเซียนโกะระดับโลกที่มีฝีมือขั้นสูงสุดในการเล่นโกะได้

## 2.6 Deep Learning

การเรียนรู้เชิงลึก (Deep Learning -DL) เป็นสาขาหนึ่งของการเรียนรู้ด้วยเครื่อง โดยเลียนแบบมาจาก เซลล์ประสาทของมนุษย์ เทคนิคนี้มีพื้นฐานมาจากเพอร์เซปตรอน โดยนำเซลล์ประสาทหลาย ๆ เซลล์มาเรียงต่อกันเป็นชั้น ๆ จำนวนมาก เป็นโครงข่ายประสาทเทียม หรือ Artificial neural network (ANN) ตัวอย่างของ ANN ง่าย ๆ คือ เมื่อตาเราเห็นสุนัข รูปสุนัขนี้จะถูกนำไปประมวลผลและได้คำตอบมาว่า สิ่งนี้คือสุนัข

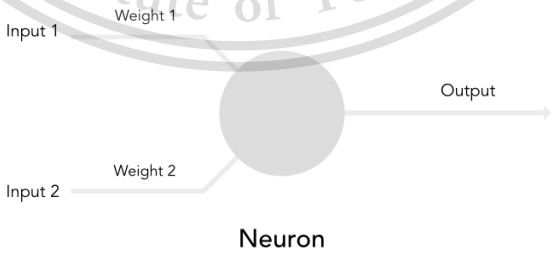
ถ้าเราจำลอง ANN จะได้ 3 ส่วน ส่วนแรกเป็น input หมายถึงสิ่งที่ตาเราเห็น คือ สุนัข ต่อมาเป็น Hidden layer เปรียบได้กับสิ่งที่ระบบเซลล์ประสาทในสมองเราประมวลผล และ Output layer คือ ผลลัพธ์ที่สรุปออกมาให้เรารู้ว่า สิ่งนี้เป็นสุนัข ซึ่งทั้งสามส่วนนี้ จะเชื่อมโยงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.




**รูป 2.6 โครงสร้างของ Artificial neural network**

ส่วนที่เล็กที่สุดของ Neural Network คือ Neuron ทำหน้าที่คำนวนผลลัพธ์ โดยมีส่วนประกอบสำคัญดังนี้




**รูป 2.7 โครงสร้างของนิวรอน (Neuron)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

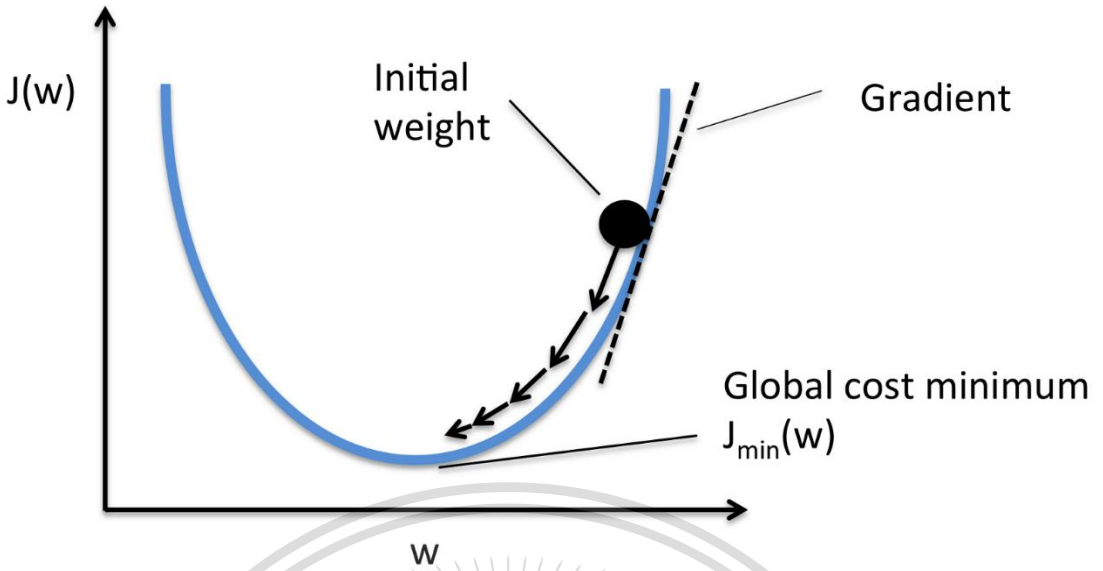
**ตาราง 2.2 คำอธิบายส่วนประกอบของนิวรอน (Neuron)**

| | |
|---|---|
| Input | ค่าที่ส่งเข้ามาที่ Neuron สามารถมีขาเข้าได้หลายขา |
| Weight | เป็นการให้น้ำหนักของขาแต่ละที่ส่งเข้ามา โดยมีค่าระหว่าง 0-1 จากนั้นตัว Neuron เมื่อทำการเรียนรู้เรื่อย ๆ จะมีการปรับ weight เพื่อให้ได้ Output ใกล้เคียงกับคำตอบมากที่สุด |
| Bias | ค่าที่จะช่วยทำให้ค่าที่เข้ามาอยู่ในระหว่าง 0 - 1 ได้ โดยจะเป็นเลขสุ่มและปรับไปเรื่อย ๆ ทุกครั้งที่เรียนรู้ |
| Output | ผลลัพธ์ |

เมื่อสร้างตัว Neuron ขึ้น ก็จะมีการกำหนด Weight เริ่มต้นให้กับแต่ละขา Input โดยการสุ่ม และเมื่อเราป้อน Input และ Output ตัว Neuron จะทำการหาผลรวมของ Input คูณกับ Weight ในแต่ละขา จากนั้นบวกค่าอื่น ๆ เช่น Bias แล้วนำไปเข้าฟังก์ชันที่ตัว Neuron นั้นกำหนดไว้ เรียกว่า Activation Function เช่น Sigmoid, Hyperbolic Tangent, Hard Limit, Rectified Linear Unit เป็นต้น ก็จะได้ Output ซึ่งค่า Output ที่คำนวณมาได้นี้ จะไม่ตรงกับ Output จริง ๆ ที่เราป้อนเข้าไป มีความคลาดเคลื่อน หรือเกิด Error ที่สามารถหาได้ด้วย Cost Function โดยเราจะใช้เทคนิคที่เรียกว่า Back Propagation คือการนำค่า Error นั้นมาปรับค่า Weight และ Bias ใหม่ เพื่อให้ได้ค่า Output ใกล้เคียงกับผลลัพธ์ที่เราต้องการมากที่สุด

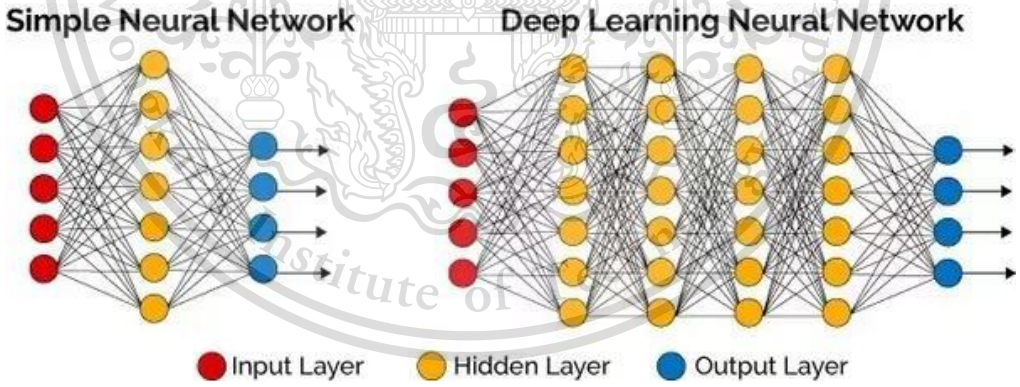
ในการลด Cost function เราจะใช้เทคนิคที่เรียกว่า Gradient Descent หรือการเคลื่อนลงตามความชัน ที่จะใช้ในการหาค่าต่ำสุดของฟังก์ชัน ในที่นี้คือ Cost function โดยจะทยอยปรับค่า Weight จาก Cost function โดยการหาอนุพันธ์ของ cost function และเราจะเห็นว่ามันเคลื่อนที่ไปในทิศทางใด การเคลื่อนที่ครั้งหนึ่งจะเคลื่อนไปเท่าไหร่นั้น ขึ้นอยู่กับอัตราการเรียนรู้ (learning rate) ถ้ากำหนดให้สูงไปแทนที่จะได้เคลื่อนลงไปหาจุดต่ำสุด ก็อาจจะกลายเป็นกระโดดข้ามไปมา แต่หากกำหนด learning rate น้อยเกินไป ก็จะทำให้การเคลื่อนที่ช้ามาก ทำให้เสียเวลามาก กว่าจะเคลื่อนที่ไปถึงจุดต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.




**รูป 2.8 การเคลื่อนลงตามความชันไปหาจุดต่ำสุด**

หากเรามี Neuron หรือ เซลล์ที่ทำหน้าที่วิเคราะห์และประมวลมาก ก็จะทำให้สามารถวิเคราะห์อะไรที่ซับซ้อนได้ ซึ่งความแตกต่างระหว่าง ANN และ Deep learning คือ ระดับ hidden layer ใน Deep learning จะมีมากกว่า




**รูป 2.9 เปรียบเทียบโครงสร้างของ Simple Neural Network และ Deep Neural Network**

## 2.7 Convolutional Neural Network

Convolutional Neural Network (CNN) หรือ โครงข่ายประสาทแบบ Convolution เป็นโครงข่ายประสาทเทียมชนิดหนึ่ง ซึ่งจะต่างจาก โครงข่ายประสาทเทียมแบบธรรมดา คือ มีการใช้เทคนิค Convolution ที่จะจำลองการมองเห็นของมนุษย์ที่มองพื้นที่เป็นที่ย่อย ๆ และนำกลุ่มของพื้นที่ย่อย ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

นั้นมาผสานกัน เพื่อวิเคราะห์ว่าสิ่งที่กำลังเห็นอยู่คืออะไร จึงทำให้ CNN ถูกนำไปใช้งานในการแก้ปัญหา Classification ที่เกี่ยวกับรูปภาพ ได้เป็นอย่างดี

**2.7.1 Convolutional Layer ประกอบไปด้วย 3 ส่วนหลัก ๆ ได้แก่**

**2.7.1.1 Convolution stage**

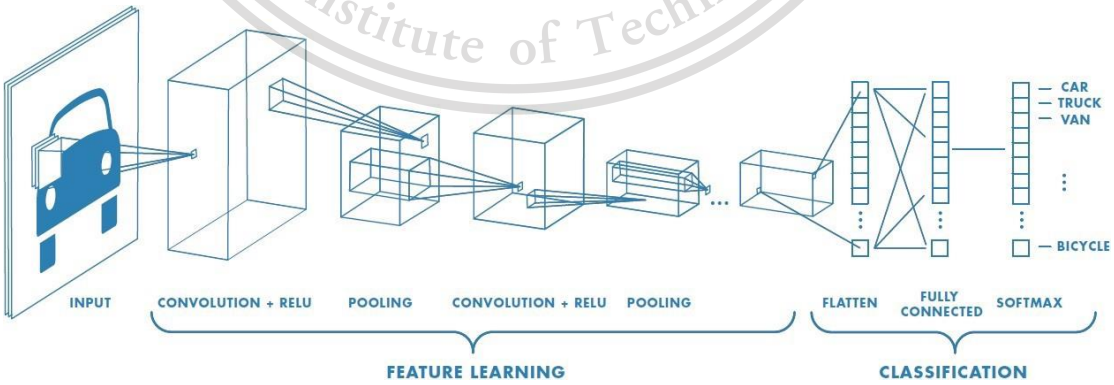
ขั้นตอนนี้จะสร้าง Filter หรือ Kernel มาสแกนรูปภาพที่เป็น input ทำ Feature map เพื่อแยกคุณลักษณะของรูป เช่น ขอบ รูปทรง สี โดยจะมี Stride เป็นตัวกำหนดว่าเราจะเลื่อน Kernel ไปด้วย Step เท่าไหร่ ในการเลื่อน Kernel ตามค่า Stride จะมีบางบริเวณที่ไม่สามารถเลื่อนได้ เพราะเกินขนาดของรูป โดยเฉพาะบริเวณขอบของรูป ซึ่งอาจทำให้ไม่สามารถเก็บรายละเอียดของภาพที่สำคัญได้ จึงต้องใช้ เทคนิค Padding ในการเพิ่มข้อมูลเช่นศูนย์หรือค่าต่าง ๆ เพื่อเพิ่มพื้นที่รูปบริเวณขอบ

**2.7.1.2 Detector stage**

ขั้นนี้จะทำการนำ Output จาก Convolution stage มาแปลงให้อยู่ในรูปของ Non-linear โดนใช้ Activation เช่น Rectified linear Units (ReLU) เพื่อให้สะดวกในการคำนวณ

**2.7.1.3 Pooling stage Pooling**

เป็นการลดขนาดข้อมูลให้เล็กลง โดยที่เก็บรายละเอียดที่สำคัญของ Input เพราะในความเป็นจริง เรามีข้อมูลที่ต้องดำเนินการเยอะมาก หากเราไม่ทำการลดขนาดข้อมูล จะทำให้เสียเวลาและเปลืองทรัพยากรในการดำเนินการเกินความจำเป็น และยังเป็นการแก้ปัญหา Overfitting ได้ด้วย ซึ่งอัลกอริทึมที่นิยมใช้คือ L2 pooling และ Max pooling ใน layer ถัดไป ก็อาจทำซ้ำเดิม แต่ Layer สุดท้าย ต้องเป็น Fully-connected Neural Network เพื่อทำการ Classification ให้ได้ผลลัพธ์ออกมา



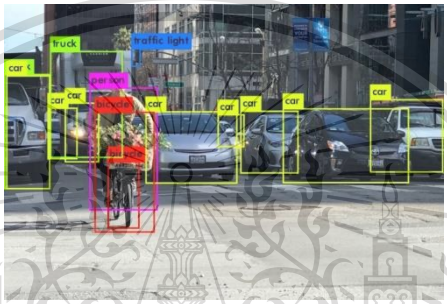

**รูป 2.10 โครงสร้างของ CNN**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

## 2.8 Object Detection Model

Object Detection การตรวจจับวัตถุ คือ เทคโนโลยีในทางคอมพิวเตอร์ หลักการที่เกี่ยวกับ Computer Vision และ Image Processing ที่ใช้ในงาน AI ตรวจจับวัตถุชนิดที่กำหนด เช่น มนุษย์ รถยนต์ อาคาร ที่อยู่ในรูปภาพ หรือวิดีโอ

งาน Object Detection การตรวจจับวัตถุในรูปภาพ สามารถเจาะลึกลงไปได้อีกหลายแขนง เช่น การทำ Face Detection ตรวจจับหน้าคน Pedestrian Detection ตรวจจับคนเดินถนน สามารถประยุกต์ใช้ได้หลากหลาย เช่น ใช้ในงานรักษาความปลอดภัย และรถยนต์ไร้คนขับ เป็นต้น




**รูป 2.11 ตัวอย่างผลลัพธ์การทำ Object Detection แบบ Multiple Objects**

Object Detection Model เป็นโมเดลที่ใช้ในงานด้าน Object detection โดยเฉพาะในการ Classification ว่าในรูปมีวัตถุใดบ้าง และ Localizing วัตถุเหล่านั้นว่าอยู่ในตำแหน่งใด โดยโมเดลต่อไปนี้จะมีอยู่ใน TensorFlow และโครงงานนี้เลือกใช้ในการทดลอง

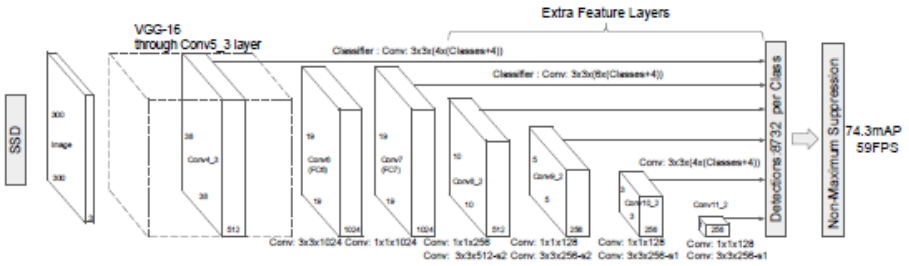
### 2.8.1 Single Shot multi-Box Detector (SSD)

SSD ถูกออกแบบมาเพื่อทำงานด้าน Object detection แบบ Real-time เพราะมีขนาดที่เล็ก และมีประสิทธิภาพในระดับหนึ่ง

เหตุที่ทำให้ SSD ใช้เวลาในการประมวลผลน้อย คือ การกำจัด bounding box proposal ออก และสามารถแชร์ Feature ระหว่าง layer ได้ โดยจะใช้ Predictor (Filter) ใน Aspect Ratio ที่แตกต่างกัน เพื่อทำ Detection ที่หลากหลาย Scale แทนการทำ Object Proposal หรือการหาพื้นที่ ที่คาดว่าจะมีวัตถุอยู่ออก จะช่วยให้ทำงานกับภาพที่มีวัตถุ scale ต่างกันมากได้

สำหรับ SSDLite จะมีข้อแตกต่างจาก SSD รูปแบบเดิมคือ จะทำการเปลี่ยน convolution ธรรมดาใน SSD layer รวมถึง Layer สุดท้ายคือ box และ class prediction เป็น Depthwise separable convolution ที่ช่วยในการลดจำนวน Parameter ที่ต้องคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

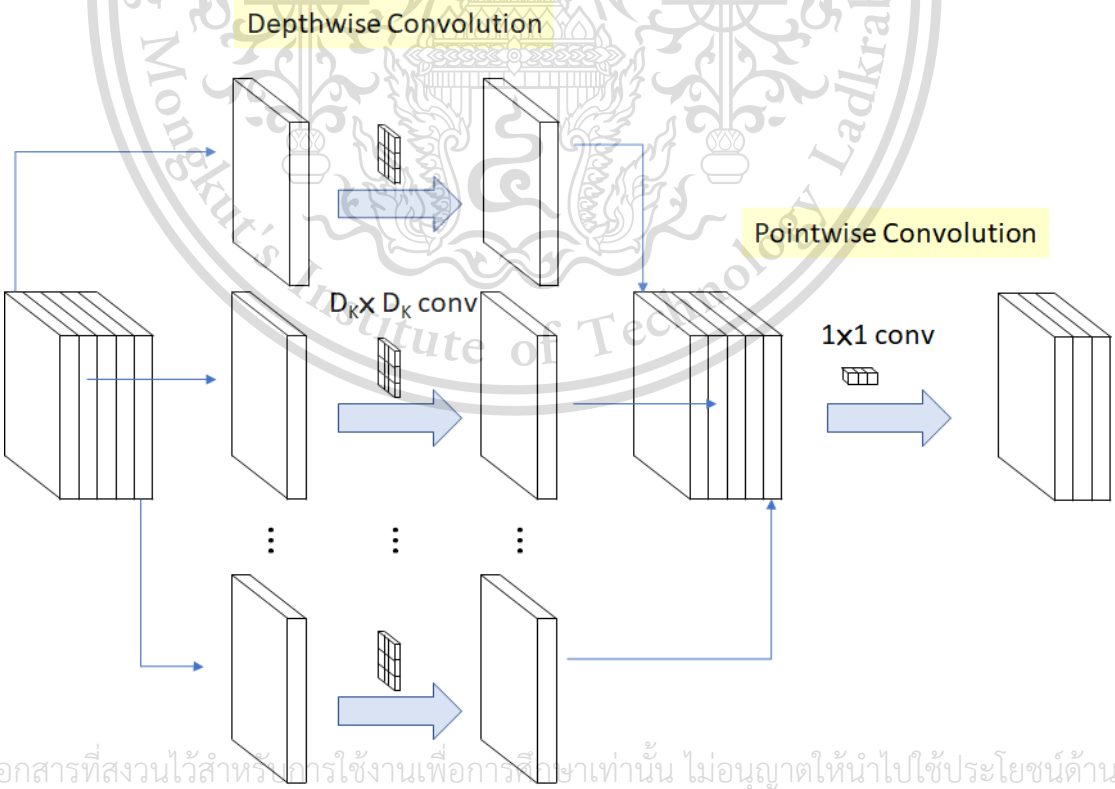



**รูป 2.12 สถาปัตยกรรมของ SSD**

### 2.8.2 MobileNet

MobileNet เป็นโมเดลที่ถูกออกแบบมาให้เหมาะกับการใช้งานบน Mobile และอุปกรณ์ขนาดเล็กที่มีกำลังในการประมวลผลไม่สูงมากโดยจุดเด่นหลักคือการใช้ Depthwise Separable Convolution ที่ ช่วยในการลดจำนวนพารามิเตอร์ที่ต้องคำนวณเมื่อเทียบกับการทำ Convolution แบบปกติ ทำให้ MobileNet เป็น Deep Neural Network ขนาดเล็ก

Depthwise Separable Convolution จะทำการแยกจาก Convolution แบบปกติเป็น 2 Layerคือ Depthwise Convolution และ Pointwise Convolution ที่เป็น 1x1 Convolutional Filters

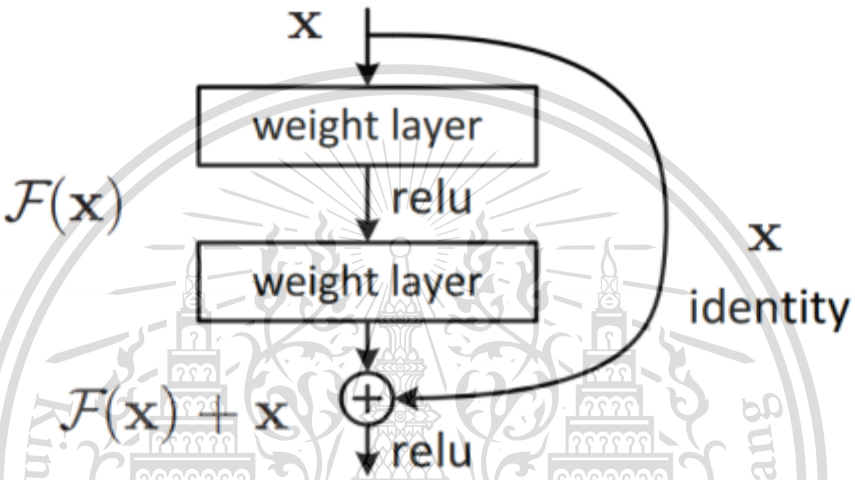



**รูป 2.13 สถาปัตยกรรมของ MobileNet**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

### 2.8.3 Deep Residual Network (ResNet)

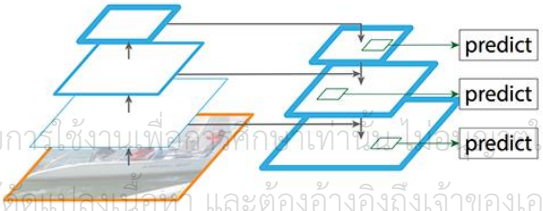
ResNet ใช้การเรียนรู้แก้ปัญหาเรื่อง Vanishing Gradient ซึ่งเกิดขึ้นกับโครงข่ายที่มีความลึกค่อนข้างมาก ด้วยการใส่ทางลัด (Shortcut) ลงในโครงข่าย โดยปกติผลลัพธ์จาก Layer แบบ Convolution ก่อนหน้าจะเริ่มเลือนหาย ติดตามได้ยากว่าการเปลี่ยนค่าพารามิเตอร์ใน Layer ก่อนหน้าจะให้ผลอะไร เพราะถูกกลืนไปกับ Layer ถัดมา ResNet แก้ปัญหานี้ได้ด้วยการส่งผ่านผลลัพธ์ลัดชั้นขึ้นมา แล้วเอาไปบวกกับผล จาก Layer ที่สาม ดังภาพข้างล่างนี้




**รูป 2.14 สถาปัตยกรรมของ ResNet**

### 2.8.4 Feature Pyramid Network (FPN)

FPN เป็นตัวแยก Feature จากหลาย Scale ประกอบด้วยทิศทางข้อมูลจาก Scale ใหญ่ไปเล็ก(ล่างขึ้นบน) และจากเล็กกลับมาใหญ่(บนลงล่าง) เส้นทางจากล่างขึ้นบนเป็นเครือข่าย Convolutional ตามปกติสำหรับการแยกคุณลักษณะ เมื่อทิศทางสูงขึ้นไปความละเอียดเชิงพื้นที่จะลดลง แต่เมื่อตรวจพบโครงสร้างระดับสูงมากขึ้น Semantic Value (ค่าความหมาย) สำหรับแต่ละชั้นจะเพิ่มขึ้น นอกจากนี้ยังมีการส่งข้อมูลข้ามใน Scale เดียวกันอีกด้วย ทำให้ FPN สามารถดึงข้อมูลที่มี High Semantic Value จากทุก ๆ Resolution ได้

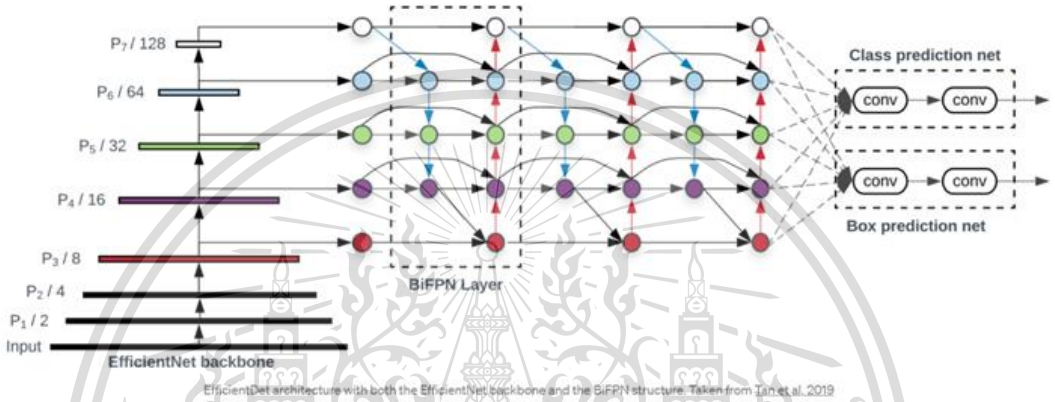



**รูป 2.15 ลำดับการทำงานของ FPN**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

### 2.8.5 EfficientDet

EfficientDet นั้นมี EfficientNet ซึ่งเป็น CNN รูปแบบหนึ่งที่เพิ่ม Efficiency และ Accuracy ด้วยการ Scale up ทั้ง Depth, Width และ Resolution ของ Network (Compound Scaling) เป็น backbone ของโมเดลทำหน้าที่สกัดรูปภาพใน Scale ต่างๆออกมา จากนั้นจึงจะถูกป้อนเข้าไปใน BiFPN ที่ทำหน้าที่เป็น Feature Fusion Network และท้ายสุดข้อมูลที่สกัดจาก BiFPN ก็จะถูกป้อนเข้า Class/Box Prediction Network เพื่อทำนาย Object ต่างๆ เป็นการเสร็จสิ้นกระบวนการ




**รูป 2.16 สถาปัตยกรรมของ EfficientDet**

## 2.9 Raspberry Pi

Raspberry Pi ถูกคิดค้นขึ้นในคริสต์ศักราช 2006 ที่มหาวิทยาลัยเคมบริดจ์ ประเทศอังกฤษ โดยผู้คิดค้นทั้งสี่ (Upton, Rob Mullins, Jack Lang และ Alan Mycroft) มีจุดประสงค์ที่จะให้ Raspberry Pi เป็นคอมพิวเตอร์ราคาย่อมเยาที่ให้ทุกคนสามารถครอบครองได้ และสามารถศึกษาการทำงานของคอมพิวเตอร์พร้อมทั้งเขียนโปรแกรมง่ายๆ ได้ทันที

Raspberry Pi เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอภาพที่มีการรองรับพอร์ต HDMI และยังสามารถเชื่อมต่อร่วมกับแป้นพิมพ์ เมาส์ได้อีกด้วย Raspberry Pi เปรียบเสมือนคอมพิวเตอร์ที่มีประสิทธิภาพสูงในราคาที่ประหยัด รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น จึงทำให้ Raspberry Pi เป็นอีกอุปกรณ์หนึ่งที่สามารถนำมาใช้ประโยชน์ได้มากมายหลากหลายด้าน ไม่ว่าจะเป็นการใช้งานด้านทั่วไป เช่น ใช้งานอินเทอร์เน็ต รับชมวิดีโอ เป็นต้น รวมถึงการประยุกต์ใช้ด้านการเรียนรู้ เพื่อใช้เป็นส่วนประกอบในการพัฒนาโครงงาน เช่น นำไปใช้เขียนโปรแกรม หรือใช้เป็นบอร์ดไมโครคอนโทรเลอร์ เพื่อควบคุมระบบและการทำงานต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**รูป 2.17 RASPBERRY PI 4 MODEL B, CORTEX-A72, 8GB RAM**

**ตาราง 2.3 Features and Specs of Raspberry Pi 4 Model B**

| Features/Specs | Raspberry Pi 4 Model B |
|---|---|
| Release Date | 24th June 2019 |
| Processor | Broadcom BCM2711 |
| Core Type | Cortex-A72 (ARM v8) 64-bit |
| No. of Cores | Quad core |
| GPU | VideoCore VI |
| Multimedia | H.265 (4kp60 decode)<br>H264 (1080p60 decode, 1080p30 encode)<br>OpenGL ES 3.0 graphics |
| CPU Clock | 1.5GHz |
| Memory/OS storage | MicroSD |
| RAM | LPDDR: 1GB, 2GB, 4GB and 8GB options |
| Ethernet | True Gigabit Ethernet |
| USB Port | 2 × USB 3.0 ports.<br>2 × USB 2.0 ports. |
| HDMI | 2 × micro-HDMI ports (up to 4kp60 supported |
| WIFI | 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless |
| Bluetooth | Bluetooth 5.0 + BLE (Shielded) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 2.3 Features and Specs of Raspberry Pi 4 Model B (ต่อ)**

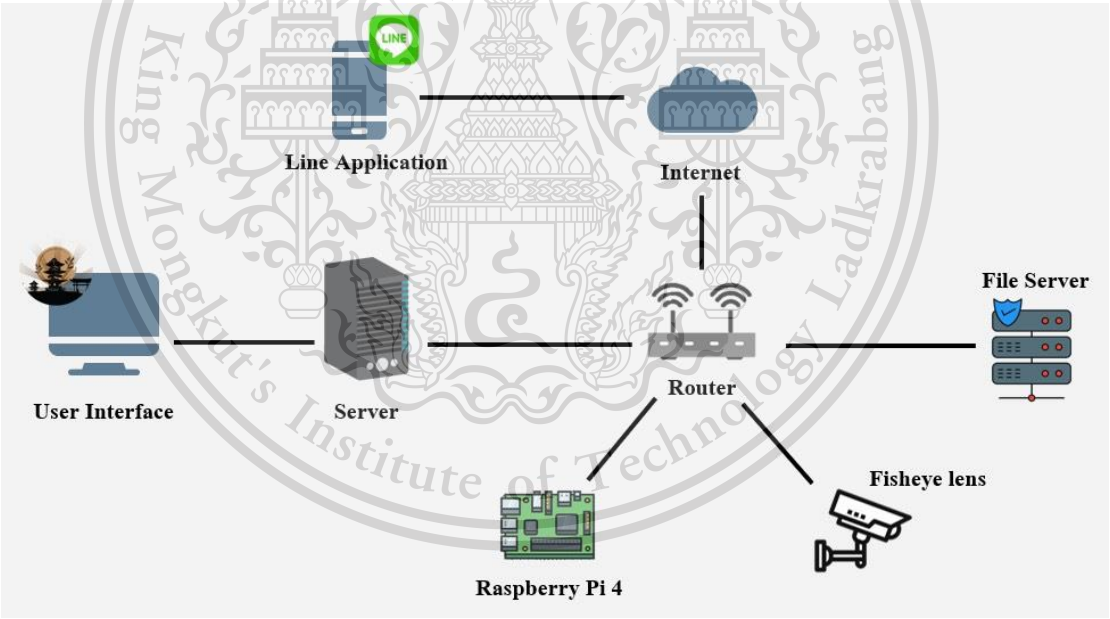
| | |
|---|---|
| Antenna | PCB Antenna (Similar to Rpi Zero W) |
| GPIO | 40 pins (fully backwards compatible with previous boards) |
| Operating System | Raspbian (>24 June 2019) |
| Dimension | 85mm × 56mm |
| Power Input | 5V DC via USB-C connector (minimum 3A*)<br>5V DC via GPIO header (minimum 3A*)<br>Power over Ethernet (PoE) enabled (requires separate PoE HAT) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# บทที่ 3
# การออกแบบและพัฒนา

## 3.1 โครงสร้างของระบบ

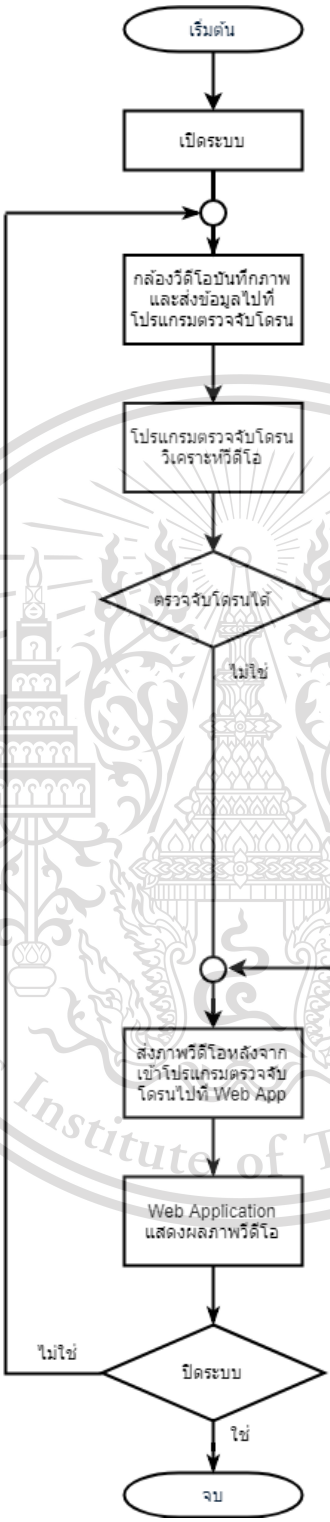
เริ่มต้นโปรแกรมตรวจจับโดรนที่ทำงานอยู่บนส่วนประมวลผลคือ Raspberry Pi 4 จะทำการรับข้อมูลภาพวีดีโอจากกล้อง และทำการประมวลผล ผลลัพธ์ที่ได้จะเป็นภาพวีดีโอต้นฉบับ ซึ่งหากมีการตรวจพบโดรนจะทำหาร Bounding Box หรือตีกรอบรอบโดรน และทำการแจ้งเตือนผู้ใช้โดยจะส่งภาพของโดรนพร้อมทั้งวันที่และเวลาที่ตรวจจับได้ไปให้ผู้ใช้ผ่านทาง Line Application พร้อมทั้งบันทึกรูปของโดรนที่ถูกตรวจจับได้ไปที่ File server เพื่อให้ผู้ใช้สามารถตรวจสอบได้ในภายหลัง จากนั้นจะทำการสตรีมวีดีโอที่ได้ไปที่ User Interface ที่เป็น Web Application เพื่อแสดงภาพวิดีโอที่ได้หลังจากเข้าสู่โปรแกรมตรวจจับโดรน




**รูป 3.1 โครงสร้างของระบบตรวจจับโดรนด้วยกล้อง Fish Eye Lens และ Raspberry Pi 4**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

### 3.2 ผังงาน (Flowchart) ของระบบ




**รูป 3.2 Flowchart แสดงการทำงานในภาพรวมของระบบ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

## 3.3 การพัฒนาโปรแกรมตรวจจับโดรน

### 3.3.1 การเตรียม Dataset

Dataset ที่เรานำมาใช้สอนโมเดล เราใช้รูปโดรนจากคลิปวีดีโอที่บันทึกภาพขณะโดรนบินด้วยความสูงไม่เกิน 10 เมตรด้วยกล้องวงจรปิด C61S ที่วางอยู่บนพื้น แล้วนำมาทำการแยกออกมาเป็นภาพนิ่ง และทำการเลือกรูปที่มีความหลากหลาย ของมุมมองโดรนในภาพ โดยเราทำการแบ่งรูปที่ไว้ใช้เทรน 800 รูป และสำหรับทดสอบ 200 รูป รวม Dataset ทั้งหมด 1,000 รูป

**รูป 3.3 ตัวอย่างรูปจาก Dataset**

เมื่อแบ่งข้อมูลสำหรับการเทรนและทดสอบแล้ว จึงทำการ Label ตำแหน่งของโดรนทุกรูปด้วยโปรแกรม LabelImg โดยจะได้เป็นไฟล์นามสกุล XML เก็บพิกัดตำแหน่งของโดรนในรูป และทำการแปลงข้อมูลนามสกุล XML เป็นไฟล์นามสกุล CSV เมื่อได้ข้อมูลพิกัดตำแหน่งของโดรนในรูปแบบไฟล์ CSV แล้ว ให้แปลงข้อมูลนั้นเป็น File Format ของ TensorFlow คือ TFrecord โดยจะเป็นไฟล์ Output คือ train.record สำหรับข้อมูลเทรน และ test.record สำหรับข้อมูลทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**รูป 3.4 โปรแกรม LabelImg**

### 3.3.2 การเลือก Pre-trained model

## TensorFlow 2 Detection Model Zoo

TensorFlow 2.2 Python 3.6

We provide a collection of detection models pre-trained on the COCO 2017 dataset. These models can be useful for out-of-the-box inference if you are interested in categories already in those datasets. You can try it in our inference colab

They are also useful for initializing your models when training on novel datasets. You can try this out on our few-shot training colab.

Please look at this guide for mobile inference.

Finally, if you would like to train these models from scratch, you can find the model configs in this directory (also in the linked `tar.gz` s).

| Model name | Speed (ms) | COCO mAP | Outputs |
|---|---|---|---|
| CenterNet HourGlass104 512x512 | 70 | 41.9 | Boxes |
| CenterNet HourGlass104 Keypoints 512x512 | 76 | 40.0/61.4 | Boxes/Keypoints |
| CenterNet HourGlass104 1024x1024 | 197 | 44.5 | Boxes |
| CenterNet HourGlass104 Keypoints 1024x1024 | 211 | 42.8/64.5 | Boxes/Keypoints |
| CenterNet Resnet50 V1 FPN 512x512 | 27 | 31.2 | Boxes |
| CenterNet Resnet50 V1 FPN Keypoints 512x512 | 30 | 29.3/50.7 | Boxes/Keypoints |
| CenterNet Resnet101 V1 FPN 512x512 | 34 | 34.2 | Boxes |
| CenterNet Resnet50 V2 512x512 | 27 | 29.5 | Boxes |

**รูป 3.5 TensorFlow 2 Object Detection Model**

TensorFlow มี Pre-trained model สำหรับงานด้าน Object Detection ที่เทรนมาแล้วกับ Dataset มาตราฐานต่าง ๆ เช่น COCO dataset, Kitti dataset และ Open Images dataset ไว้ให้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

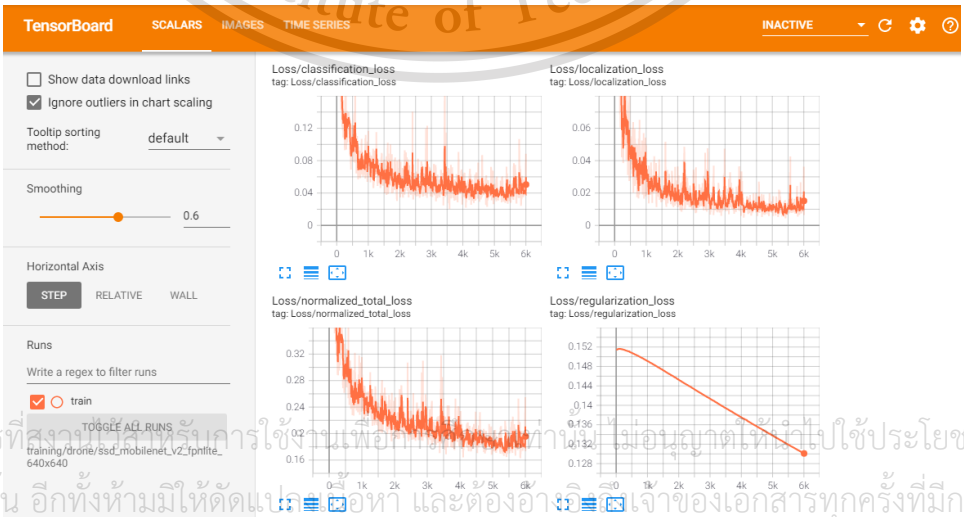
มากมาย และเราจะนำ Pre-trained model นั้น ไปทำ Transfer learning กับ Dataset ของเรา โดยเราเลือกโมเดลที่คาดว่าเหมาะกับงานเรามาทั้งหมด 4 โมเดลคือ SSD MobileNet V2, SSD MobileNet V2 FPNLite, SSD ResNet50 V1 FPN และ EfficientDet D0 มาเปรียบทั้งความเร็วในการประมวลและความแม่นยำในการทำนาย โดยเรานำโมเดลทั้งสี่มาเทรนด้วย Dataset เดียวกัน เทรนในจำนวน Global step ที่ใกล้เคียงกัน (Global step ใน TensorFlow คือจำนวน Batches ที่เข้าสู่การเทรนจนถึงปัจจุบัน) และปรับค่า config ในการเทรนโมเดลเป็นค่าเดียวกัน เช่น Batch Size และ Learning Rate

เมื่อทั้ง 4 โมเดลถูกเทรนเรียบร้อยแล้ว ก็นำมาทดสอบประสิทธิภาพด้วยชุดข้อมูลทดสอบเดียวกัน และทำการคำนวณค่า Precision, Recall, Accuracy และ F score พร้อมทั้งทดสอบความเร็วในการประมวลต่อหนึ่งเฟรม (FPS) แล้วนำมาเปรียบเทียบกัน

### 3.3.3 การเทรนโมเดล

เมื่อเลือก Pre-trained model ที่จะนำมาทำ Transfer learning ได้แล้วก็ทำการแก้ไขค่าต่าง ๆ ในไฟล์ config สำหรับโมเดลนั้น ๆ ค่าที่ต้องมีการปรับแก้ คือ Batch size, learning rate และ Path ของโฟลเดอร์ Dataset รวมถึงโฟลเดอร์ที่เก็บค่าพิกัดตำแหน่งของโดรนในรูปซึ่งอยู่ในรูปแบบ TFrecord และสร้างไฟล์สำหรับทำ label map คือ การระบุคลาสที่มีทั้งหมด สำหรับโครงงานนี้มี 2 คลาส คือ โดรนรุ่น Sky hunter X8 (4 ใบพัด) และรุ่น TYH scout (6 ใบพัด) ไฟล์นี้จะเป็นไฟล์นามสกุล pbtxt เมื่อทำการแก้ไขไฟล์ config และสร้างไฟล์ label map เรียบร้อยแล้วจึงเริ่มเทรนโมเดลนั้น และคอยตรวจสอบการเทรนผ่าน Tensorboard เพื่อดูค่า loss ต่าง ๆ เช่น Classification Loss, Localization Loss และ Total Loss ที่เป็นค่าความผิดพลาดต่าง ๆ และทำการ ฎvaluate โมเดลขณะเทรนได้ด้วย เพื่อดูว่าตอนนี้โมเดลเราสามารถ Classification รูปทดสอบตัวอย่างได้ดีแค่ไหน

เมื่อเทรนโมเดลเสร็จเรียบร้อยแล้ว เราจะทำบันทึกโมเดลนั้น หรือที่เรียกว่า Freezing the Graph เพื่อบันทึกค่าน้ำหนักพารามิเตอร์ต่าง ๆ ที่โมเดลได้มีการปรับค่าเป็น TensorFlow Graph




**รูป 3.6 หน้าต่างแสดงผล TensorBoard**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

### 3.3.4 การนำโมเดลไปใช้งาน

เมื่อเราได้โมเดลที่เทรนเรียบร้อยแล้ว ก็นำไปใช้งาน โดยเขียนโปรแกรมที่รับข้อมูลภาพวีดีโอจากกล้อง IP แล้วนำข้อมูลภาพวีดีโอนั้นเข้าสู่โมเดลเพื่อทำการ Classification และ Localization หากตรวจพบโดรน ทีละเฟรม ซึ่งหากมีการตรวจพบโดรนโปรแกรมจะทำการ Bounding Box หรือตีกรอบสี่เหลี่ยมรอบโดรน ข้อมูลภาพวีดีโอหลังจากเข้าสู่โมเดลที่ทำการ Classification และ Localization แล้วนั้น จะถูกส่งไปที่ Server เพื่อทำการแสดงภาพวีดีโอผลลัพธ์ที่ได้จากการประมวลผลของโมเดล

## 3.4 การพัฒนาส่วนติดต่อกับผู้ใช้ (User Interface)

ส่วนติดต่อกับผู้ใช้ที่เป็น Web Application จะเป็นส่วนแสดงผลภาพวีดีโอที่เข้าสู่โปรแกรมตรวจจับโดรนที่ทำงานบน Raspberry Pi 4 แล้วส่งข้อมูลนั้นต่อมายังส่วนติดต่อกับผู้ใช้นี้ โดยโครงงานนี้เลือกใช้ Open Source CCTV Solution คือ Shinobi ในการแสดงผล

เมื่อผู้ใช้ต้องการใช้งาน ผู้ใช้จะต้องเข้าสู่ระบบ และเข้าไปตั้งค่าการเชื่อมต่อ เพื่อรับข้อมูลภาพวีดีโอที่ส่งมาหลังจากทำการประมวลผล

**รูป 3.7 หน้าต่างตั้งค่าการเชื่อมต่อสำหรับรับข้อมูลจากโปรแกรมตรวจจับโดรน**

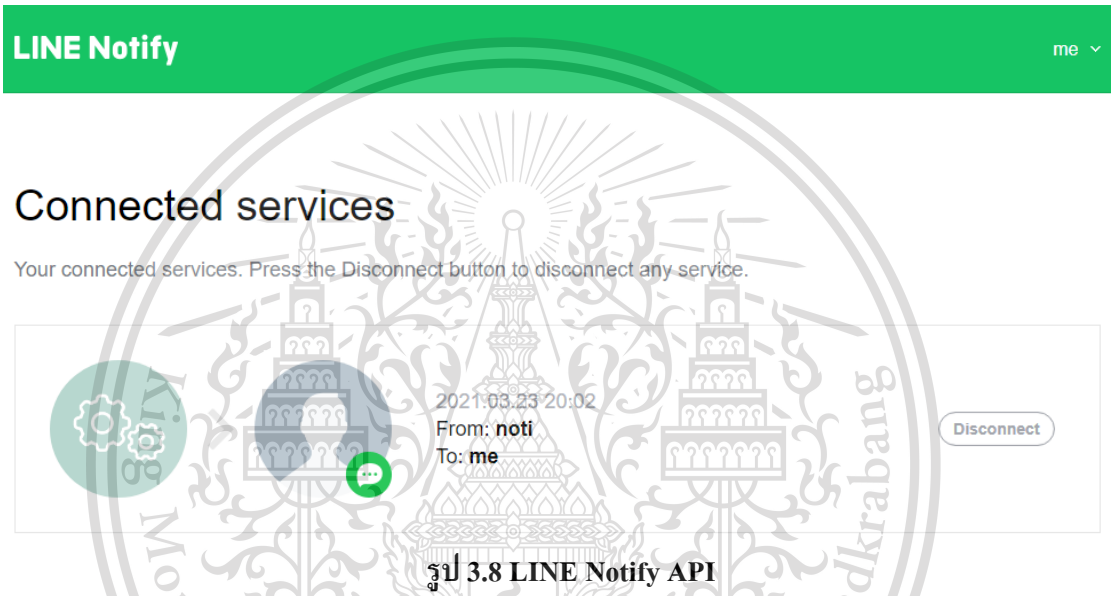
## 3.5 การพัฒนาระบบแจ้งเตือนผู้ใช้

ระบบจะทำการแจ้งเตือนผู้ใช้หากมีการตรวจจับโดรนได้ ผ่านทาง Line Application เพราะเป็น Platform ที่สะดวกกับผู้ใช้ สำหรับการพัฒนาระบบการแจ้งเตือนด้วย Line Bot นี้ เราจะทำการลงทะเบียนกับ Line Notify ซึ่งเป็นบริการที่ทาง Line ได้เตรียมไว้ให้ในรูปแบบของ API เพื่อให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สามารถพัฒนาโปรเจค ที่มีความต้องการส่งข้อความเพื่อแจ้งเตือนเข้าไปยัง กลุ่มหรือบัญชีที่เราต้องการได้ เมื่อเราได้ทำการลงทะเบียนแล้ว เราจะได้ Access Token เพื่อสร้าง Line Bot ที่เราจะนำไปใช้งาน โดยเราจะทำการเพิ่ม Line bot ไปที่ Group Chat ที่สร้างขึ้นมา ซึ่งผู้ใช้สามารถเพิ่มบัญชี Line ของตัวเองเข้าไปใน Group Chat เพื่อรับการแจ้งเตือนจาก Line bot ดังกล่าว

เราจะทำการพัฒนาโปรแกรมที่จะส่งข้อมูล 2 อย่างได้แก่ รูปภาพ และวันที่เวลา ที่ตรวจจับได้ผ่านทาง Line bot ที่ได้ลงทะเบียนไว้ และส่งข้อมูลชุดเดียวกันนี้ไปที่ File server ผ่าน FTP ด้วย เพื่อให้ผู้ใช้สามารถตรวจสอบภาพการตรวจจับในภายหลังได้




**รูป 3.8 LINE Notify API**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# บทที่ 4
# การทดลองและวิเคราะห์ผลการทดลอง

## 4.1 การทดลองเปรียบเทียบประสิทธิภาพของ Pre-trained model

### 4.1.1 วัตถุประสงค์

เพื่อเปรียบเทียบประสิทธิภาพของ Pre-trained model ที่คาดว่าจะสามารถทำงานได้บน Raspberry Pi 4 เพื่อคัดเลือกโมเดลที่เหมาะสมที่สุดในการนำมาใช้ในการตรวจจับโดรน

### 4.1.2 วิธีการทดลอง

1) เลือก Pre-trained model ใน TensorFlow ที่จะทำการเปรียบเทียบ โดยคัดเลือกโมเดลที่จะนำมาเปรียบเทียบเบื้องต้นจากตารางแสดงค่าความแม่นยำ (mAP) และ Speed ของโมเดลจากเว็บไซต์ GitHub ของ TensorFlow ในที่นี้เลือก EfficientDet D0, SSD MobileNet v2 320x320, SSD MobileNet v2 FPNLite 640x640 และ SSD ResNet50 v1 FPN 640x640
2) เตรียม Dataset ที่จะใช้สอนโมเดลทั้งหมด 1,000 รูป โดยแบ่งสำหรับการสอนจำนวน 800 รูป และสำหรับทดสอบจำนวน 200 รูป
3) เริ่มเทรนโมเดลด้วยการตั้งค่าตัวแปรพื้นฐานให้เท่ากัน คือ Batch size, Learning rate และจำนวน step ที่ทำาการเทรน
4) เมื่อเทรนโมเดลเสร็จเรียบร้อย ให้ทำการทดสอบโมเดล ด้วยรูปชุดทดสอบจำนวน 100 รูป แบ่งเป็นรูปโดรนรุ่น Sky hunter X8 (4 ใบพัด) จำนวน 50 รูป และรุ่น TYH scout (6 ใบพัด) จำนวน 50 รูป และบันทึกผลการทดสอบ
5) ทดสอบค่า FPS หรือจำนวนเฟรมที่ประมวลผลได้ในหนึ่งวินาทีและทำการบันทึกผล
6) นำผลการทดสอบมาคำนวณเพื่อวัดประสิทธิภาพของโมเดล ได้แก่ Confusion matrix, Precision, Recall, Accuracy และ F-score
7) เปรียบเทียบผลการคำนวณของแต่ละโมเดลและคัดเลือกโมเดลที่ดีที่สุดที่เหมาะกับโครงงานนี้โดยดูจากค่า Precision, Recall, Accuracy และ F-score รวมถึง FPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

### 4.1.3 ผลการทดลอง

**ตาราง 4.1 คำอธิบายคำที่ใช้ในผลการทดลอง**

| คำย่อ | ความหมาย |
|---|---|
| MobileNet | SSD MobileNet v2 320x320 |
| FPNLite | SSD MobileNet v2 FPNLite 640x640 |
| ResNet50 | SSD ResNet50 v1 FPN 640x640 |
| EfficientDet | EfficientDet D0 |
| Scout | TYH scout |
| Sky | Sky hunter X8 |
| ไม่พบ | ไม่พบโดรนในภาพ |

**ตาราง 4.2 ผลการทดสอบเปรียบเทียบประสิทธิภาพ**

| Image no. | Actual | ผลการตรวจจับ | | | |
|---|---|---|---|---|---|
| | | MobileNet | FPNLite | ResNet50 | EfficientDet |
| 1 | Scout | Scout | Scout | Scout | ไม่พบ |
| 2 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 3 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 4 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 5 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 6 | Scout | Scout | Scout | Scout | ไม่พบ |
| 7 | Scout | Scout | Scout | Scout | ไม่พบ |
| 8 | Scout | Scout | Scout | Scout | ไม่พบ |
| 9 | Scout | Scout | Scout | Scout | ไม่พบ |
| 10 | Scout | Scout | Scout | Scout | ไม่พบ |
| 11 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 12 | Scout | Scout | Scout | Scout | ไม่พบ |
| 13 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 4.2 ผลการทดสอบเปรียบเทียบประสิทธิภาพ (ต่อ)**

| Image no. | Actual | ผลการตรวจจับ | | | |
|---|---|---|---|---|---|
| | | MobileNet | FPNLite | ResNet50 | EfficientDet |
| 14 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 15 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 16 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 17 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 18 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 19 | Scout | Scout | Scout | Scout | ไม่พบ |
| 20 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 21 | Scout | Scout | Scout | Scout | Sky |
| 22 | Scout | Scout | Scout | Scout | ไม่พบ |
| 23 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 24 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 25 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 26 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 27 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 28 | Scout | Sky | Scout | Sky | ไม่พบ |
| 29 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 30 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 31 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 32 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 33 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 34 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 35 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 36 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 37 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 38 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 4.2 ผลการทดสอบเปรียบเทียบประสิทธิภาพ (ต่อ)**

| Image no. | Actual | ผลการตรวจจับ | | | |
|---|---|---|---|---|---|
| | | MobileNet | FPNLite | ResNet50 | EfficientDet |
| 39 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 40 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 41 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 42 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 43 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 44 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 45 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 46 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 47 | Scout | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 48 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 49 | Scout | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 50 | Scout | Scout | Scout | ไม่พบ | ไม่พบ |
| 51 | Sky | Sky | Sky | Sky | ไม่พบ |
| 52 | Sky | Sky | Sky | ไม่พบ | ไม่พบ |
| 53 | Sky | ไม่พบ | Sky | ไม่พบ | ไม่พบ |
| 54 | Sky | Sky | Sky | Sky | ไม่พบ |
| 55 | Sky | Sky | Sky | Sky | ไม่พบ |
| 56 | Sky | Sky | Sky | Sky | ไม่พบ |
| 57 | Sky | Sky | Sky | Sky | ไม่พบ |
| 58 | Sky | Sky | Sky | Sky | ไม่พบ |
| 59 | Sky | Sky | Sky | Sky | ไม่พบ |
| 60 | Sky | Sky | Sky | Sky | ไม่พบ |
| 61 | Sky | Sky | Sky | Sky | ไม่พบ |
| 62 | Sky | Sky | Sky | Sky | ไม่พบ |
| 63 | Sky | Sky | Sky | Sky | ไม่พบ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 4.2 ผลการทดสอบเปรียบเทียบประสิทธิภาพ (ต่อ)**

| Image no. | Actual | ผลการตรวจจับ | | | |
|---|---|---|---|---|---|
| | | MobileNet | FPNLite | ResNet50 | EfficientDet |
| 64 | Sky | Sky | Sky | Sky | ไม่พบ |
| 65 | Sky | Sky | Sky | Sky | ไม่พบ |
| 66 | Sky | Sky | Sky | Sky | ไม่พบ |
| 67 | Sky | Sky | Sky | Sky | ไม่พบ |
| 68 | Sky | Sky | Sky | Sky | ไม่พบ |
| 69 | Sky | Sky | Sky | Sky | ไม่พบ |
| 70 | Sky | Sky | Sky | Sky | ไม่พบ |
| 71 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 72 | Sky | Sky | Sky | Sky | ไม่พบ |
| 73 | Sky | Sky | Sky | Sky | ไม่พบ |
| 74 | Sky | Sky | Sky | Sky | ไม่พบ |
| 75 | Sky | Sky | Sky | ไม่พบ | ไม่พบ |
| 76 | Sky | Sky | Scout | ไม่พบ | ไม่พบ |
| 77 | Sky | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 78 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 79 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 80 | Sky | Sky | Scout | Sky | ไม่พบ |
| 81 | Sky | Sky | Scout | Sky | ไม่พบ |
| 82 | Sky | Sky | Sky | Sky | ไม่พบ |
| 83 | Sky | Sky | Scout | Sky | ไม่พบ |
| 84 | Sky | ไม่พบ | Sky | Sky | ไม่พบ |
| 85 | Sky | ไม่พบ | Sky | ไม่พบ | ไม่พบ |
| 86 | Sky | Sky | Scout | ไม่พบ | ไม่พบ |
| 87 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 88 | Sky | Sky | Scout | ไม่พบ | ไม่พบ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 4.2 ผลการทดสอบเปรียบเทียบประสิทธิภาพ (ต่อ)**

| Image no. | Actual | ผลการตรวจจับ | | | |
|---|---|---|---|---|---|
| | | MobileNet | FPNLite | ResNet50 | EfficientDet |
| 89 | Sky | Sky | Sky | Sky | ไม่พบ |
| 90 | Sky | Sky | Scout | Sky | ไม่พบ |
| 91 | Sky | ไม่พบ | Sky | ไม่พบ | ไม่พบ |
| 92 | Sky | Sky | Scout | Sky | ไม่พบ |
| 93 | Sky | Sky | Sky | ไม่พบ | ไม่พบ |
| 94 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 95 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 96 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 97 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |
| 98 | Sky | ไม่พบ | Scout | ไม่พบ | ไม่พบ |
| 99 | Sky | ไม่พบ | Sky | ไม่พบ | ไม่พบ |
| 100 | Sky | ไม่พบ | ไม่พบ | ไม่พบ | ไม่พบ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 4.3 Confusion Matrix ของ SSD MobileNet v2 320x320**

| MobileNet | | ผลการตรวจจับ | | |
|---|---|---|---|---|
| | | Sky | Scout | ไม่พบ |
| Actual | Sky | 34 | 0 | 16 |
| | Scout | 1 | 20 | 29 |
| | ไม่พบ | 0 | 0 | 0 |

**ตาราง 4.4 Confusion Matrix ของ SSD MobileNet v2 FPNLite 640x640**

| FPNLite | | ผลการตรวจจับ | | |
|---|---|---|---|---|
| | | Sky | Scout | ไม่พบ |
| Actual | Sky | 31 | 10 | 9 |
| | Scout | 0 | 32 | 18 |
| | ไม่พบ | 0 | 0 | 0 |

**ตาราง 4.5 Confusion Matrix ของ SSD ResNet50 v1 FPN 640x640**

| ResNet50 | | ผลการตรวจจับ | | |
|---|---|---|---|---|
| | | Sky | Scout | ไม่พบ |
| Actual | Sky | 29 | 0 | 21 |
| | Scout | 1 | 10 | 39 |
| | ไม่พบ | 0 | 0 | 0 |

**ตาราง 4.6 Confusion Matrix ของ EfficientDet D0**

| EfficientDet | | ผลการตรวจจับ | | |
|---|---|---|---|---|
| | | Sky | Scout | ไม่พบ |
| Actual | Sky | 0 | 0 | 50 |
| | Scout | 1 | 0 | 49 |
| | ไม่พบ | 0 | 0 | 0 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**ตาราง 4.7 ผลการคำนวณประสิทธิภาพโมเดลแบบแยกคลาส**

| | MobileNet | | FPNLite | | ResNet50 | | EfficientDet | |
|---|---|---|---|---|---|---|---|---|
| | **Sky** | **Scout** | **Sky** | **Scout** | **Sky** | **Scout** | **Sky** | **Scout** |
| **TP** | 34 | 20 | 31 | 32 | 29 | 10 | 0 | 0 |
| **TN** | 49 | 50 | 50 | 40 | 49 | 50 | 49 | 50 |
| **FP** | 1 | 0 | 0 | 10 | 1 | 0 | 1 | 0 |
| **FN** | 16 | 30 | 19 | 18 | 21 | 40 | 50 | 50 |
| **Accuracy** | 0.83 | 0.70 | 0.81 | 0.72 | 0.78 | 0.60 | 0.49 | 0.50 |
| **Precision** | 0.97 | 1.00 | 1.00 | 0.76 | 0.97 | 1.00 | 0.00 | - |
| **Recall** | 0.68 | 0.40 | 0.62 | 0.64 | 0.58 | 0.20 | 0.00 | 0.00 |
| **F score** | 0.80 | 0.57 | 0.77 | 0.69 | 0.73 | 0.33 | - | - |
| **FPS** | 2 | | 2 | | 2 | | 1 | |

**ตาราง 4.8 ผลการคำนวณประสิทธิภาพโมเดลด้วย Macro-average Method**

| **Macro** | **MobileNet** | **FPNLite** | **ResNet50** | **EfficientDet** |
|---|---|---|---|---|
| Accuracy | 0.77 | 0.77 | 0.69 | 0.50 |
| Precision | 0.99 | 0.88 | 0.99 | - |
| Recall | 0.54 | 0.63 | 0.39 | 0.00 |
| F score | 0.70 | 0.73 | 0.56 | - |

**ตาราง 4.9 ผลการคำนวณประสิทธิภาพโมเดลด้วย Micro-average Method**

| **Micro** | **MobileNet** | **FPNLite** | **ResNet50** | **EfficientDet** |
|---|---|---|---|---|
| Accuracy | 0.77 | 0.77 | 0.69 | 0.50 |
| Precision | 0.98 | 0.86 | 0.98 | 0 |
| Recall | 0.54 | 0.63 | 0.39 | 0 |
| F score | 0.70 | 0.73 | 0.56 | - |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

**4.1.4 วิเคราะห์ผลการทดลอง**

จากตาราง 4.7 จะเห็นได้ว่าทั้งสี่โมเดลที่เรานำมาทดสอบ MobileNet, FPNLite และ ResNet50 มี FPS สูงที่สุดเท่ากันคือ 2 ขณะที่ EfficientDet มี FPS เพียง 1 นอกจากนี้ตาราง 4.8 และ 4.9 ทำให้เห็นว่าทั้งสี่โมเดลมีประสิทธิภาพที่แตกต่างกันอย่างชัดเจน โดย MobileNet และ FPNLite มี Accuracy สูงที่สุดเท่ากันคือ 0.77 ตามด้วย ResNet50 และ EfficientDet แต่ FPNLite มี F-Score สูงที่สุดคือ 0.73 ตามด้วย MobileNet และ ResNet50 ขณะที่ EfficientDet ไม่สามารถคำนวณได้

ดังนั้น โครงงานนี้จึงเลือก FPNLite หรือ SSD MobileNet v2 FPNLite 640x640 เป็นโมเดลที่จะนำไปใช้ เนื่องจากมี FPS, Accuracy และ F score ที่สูงที่สุด

## 4.2 การทดลองสตรีมภาพวีดีโอผลลัพธ์บน Web Application

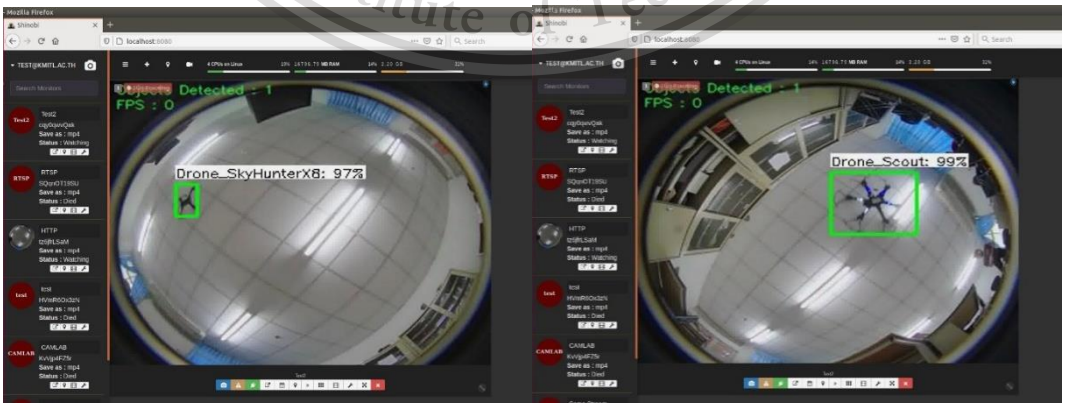
**4.2.1 วัตถุประสงค์**

เพื่อทดสอบโปรแกรมตรวจจับโดรนที่พัฒนาโดยใช้โมเดลที่ และทดสอบการแสดงวีดีโอผลลัพธ์บน Web Application Shinobi CCTV Solution

**4.2.2 วิธีการทดลอง**

1) ตั้งค่ากล้อง IP camera เพื่อเก็บภาพ
2) กำหนดให้โปรแกรมตรวจจับโดรนรับ Input เป็น ภาพวีดีโอจากกล้อง IP Camera
3) ตั้งค่าการเชื่อมต่อบน Web Application เพื่อให้รองรับการสตรีมมิ่งจากโปรแกรมตรวจจับโดรนที่ส่งมา
4) บันทึกผลการทดลอง

**4.2.3 ผลการทดลอง**




**รูป 4.1 ผลการทดลองสตรีมภาพวีดีโอผลลัพธ์บน Web application**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

### 4.2.4 วิเคราะห์ผลการทดลอง

จากผลการทดลอง สรุปได้ว่าโปรแกรมตรวจจับโดรนที่ใช้โมเดล SSD MobileNet v2 FPNLite 640x640 สามารถตรวจจับโดรนได้จากภาพวีดีโอที่ได้รับมาจาก IP camera และยังสามารถสตรีมภาพวีดีโอผลลัพธ์หลังเข้าโปรแกรมตรวจจับโดรนแล้วไปที่ Web Application เพื่อแสดงผลกับผู้ใช้ต่อไปได้

## 4.3 การทดลองระบบแจ้งเตือนและระบบบันทึกไฟล์ผลลัพธ์

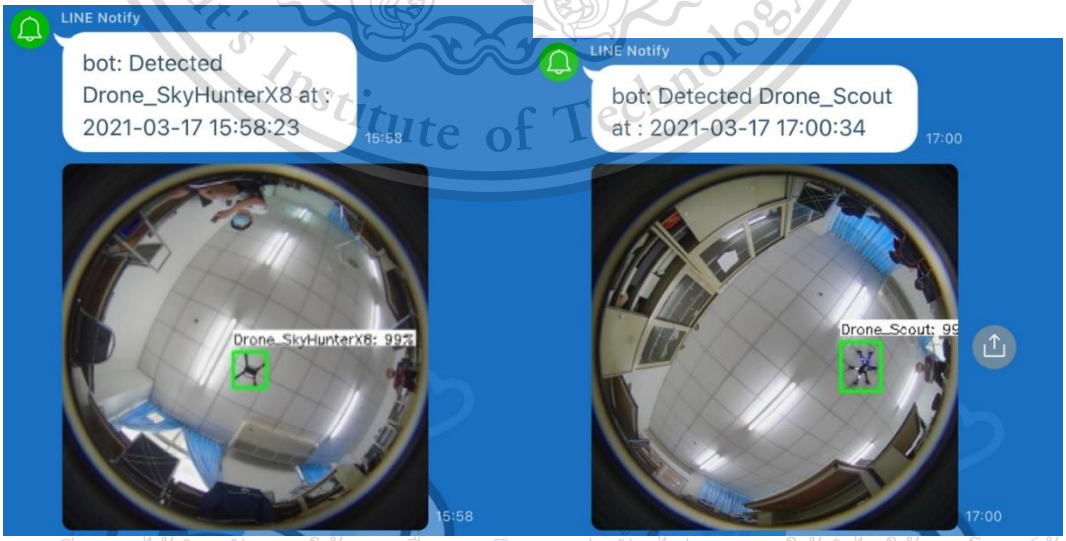
### 4.3.1 วัตถุประสงค์

เพื่อทดสอบระบบการแจ้งเตือนผู้ใช้ผ่าน Line Application และระบบบันทึกไฟล์ผลลัพธ์ที่ File Server ผ่าน FTP เมื่อตรวจจับโดรนได้

### 4.3.2 วิธีการทดลอง

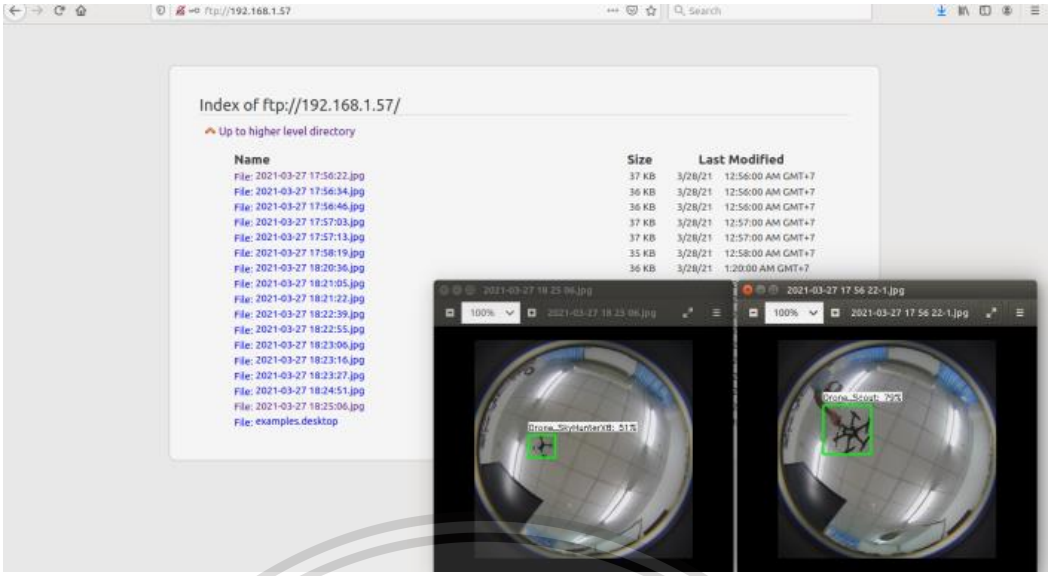
1) ตั้งค่าโปรแกรมตรวจจับโดรนในส่วนของการแจ้งเตือนผู้ใช้โดยกำหนด Access Token ให้ตรงกับ Line Bot ที่เราต้องการจะใช้
2) ตั้งค่าโปรแกรมตรวจจับโดรนในส่วนของส่งไฟล์ผลลัพธ์ผ่าน FTP โดยกำหนดให้ส่งข้อมูลไปที่ File Server
3) เริ่มโปรแกรมตรวจจับโดรน แล้วทดลองบินโดรนผ่านหน้ากล้อง
4) บันทึกผลการทดลอง

### 4.3.3 ผลการทดลอง




**รูป 4.2 ผลการแจ้งเตือนผู้ใช้ผ่าน Line Application**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.




**รูป 4.3 ผลการบันทึกไฟล์ภาพที่ตรวจจับได้บน File Server**

**4.2.4 วิเคราะห์ผลการทดลอง**

จากผลการทดลอง ระบบการแจ้งเตือนผู้ใช้ผ่าน Line Application สามารถทำงานได้ปกติเช่นเดียวกับระบบบันทึกไฟล์ผลลัพธ์บน File Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# บทที่ 5
# บทสรุปและข้อเสนอแนะ

## 5.1 บทสรุป

### 5.1.1 ส่วนประมวลบน Raspberry Pi

ระบบสามารถตรวจจับโดรนได้ 2 รุ่น ได้แก่รุ่น Sky hunter X8 (4 ใบพัด) และรุ่น TYH scout (6 ใบพัด) ด้วยโปรแกรมตรวจจับโดรนที่ประมวลผลอยู่บนRaspberry Pi 4 โดยใช้ Machine Learning Model ชื่อ “SSD MobileNet v2 FPNLite 640x640” ในการวิเคราะห์ภาพที่ได้จากกล้อง IP camera ซึ่งมีความแม่นยำของโมเดลอยู่ที่ประมาณ 77% สามารถส่งภาพวีดีโอผลลัพธ์ไปแสดงบน Web Application เพื่อแสดงผลแก่ผู้ใช้ สามารถแจ้งเตือนผู้ใช้ผ่าน Line Application และบันทึกรูปเมื่อตรวจจับโดรนได้ไว้ที่ File Server เพื่อให้ผู้ใช้ตรวจสอบในภายหลังได้

### 5.1.2 ส่วนติดต่อผู้ใช้

โครงงานนี้มี Web Application คือ Shinobi ซึ่งเป็น Open Source CCTV Solution ในการแสดงผลภาพวีดีโอผลลัพธ์ที่ได้จากโปรแกรมตรวจจับโดรนที่ประมวลอยู่บน Raspberry Pi 4 โดยผู้ใช้จะสามารถดูสตรีมภาพวิดีโอผ่านหน้าเว็บไซต์ได้อย่างสะดวก

## 5.2 ปัญหาและอุปสรรค

1) Object Detection Model ที่ใช้ในการประมวลผลยังมีความแม่นยำไม่สูงมาก
2) Object Detection Model ที่ใช้ในการประมวลผลยังไม่สามารถวิเคราะห์ภาพในตอนกลางคืนได้
3) Object Detection Model ที่ใช้ในการประมวลผลยังไม่สามารถตรวจจับโดรนที่อยู่ในระยะไกลมากได้
4) ระบบประมวลผลยังไม่สามารถทำงานได้แบบ Real-Time ยังมีการหน่วงเวลาอยู่

## 5.3 แนวทางการพัฒนาต่อ

### 5.3.1 ส่วนประมวลบน Raspberry Pi

1) เพิ่มประสิทธิภาพของ Object Detection Model ที่ใช้ในประมวลผลตรวจจับโดรน
2) ปรับปรุง Object Detection Model ที่ใช้ในการประมวลผลเพื่อให้สามารถวิเคราะห์โดรนรุ่นอื่นได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3) ปรับปรุงระยะเวลาที่ใช้ในการประมวลผลให้น้อยลง เพื่อให้ใกล้เคียง Real-Time มากที่สุด

**5.3.2 ส่วนติดต่อผู้ใช้**

1) เพิ่มฟังก์ชันการแจ้งเตือนบน Web Application
2) แสดงข้อมูลอื่น ๆ ที่เป็นประโยชน์ เช่น สถิติที่ตรวจจับโดรนได้ในแต่ละวัน




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

# บรรณานุกรม


ศุภฤกษ์ คฤหานนท์. 2560. **เลนส์ฟิชอาย (Fisheye Lens) กับการถ่ายภาพดาราศาสตร์**. [Online]. Available: https://mgronline.com/science/detail/9600000062219.

AiRAphz. 2020. **สอนให้โมเดลตรวจจับวัตถุด้วย Tensorflow Object Detection API บน Google Colab**. [Online]. Available: https://airaphz.medium.com/สอนให้โมเดลตรวจจับวัตถุด้วย-tensorflow-object-detection-api-บน-google-colab-d55153aca505.

Amy Unruh. 2017. **What is the TensorFlow machine intelligence platform?**. [Online]. Available: https://opensource.com/article/17/11/intro-tensorflow.

armaanpriyadarshan. 2020. **Object-Detection-on-Raspberry-Pi**. [Online]. Available: https://github.com/armaanpriyadarshan/Object-Detection-on-Raspberry-Pi.

armaanpriyadarshan. 2021. **Training-a-Custom-TensorFlow-2.X-Object-Detector**. [Online]. https://github.com/armaanpriyadarshan/Training-a-Custom-TensorFlow-2.x-Object-Detector.

eiu_admin. 2019. **machine Learning สำคัญกับอุตสาหกรรมอย่างไร**. [Online]. Available: http://eiu.thaieei.com/box/Research/40/บทความMachine%20Learning.pdf.

jakkcoder. 2020. **widows-object-detection-setup**. [Online]. Available: https://github.com/jakkcoder/widows-object-detection-setup.

Jonathan Hui. 2018. **SSD object detection: Single Shot MultiBox Detector for real-time processing**. [Online]. Available: https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06.

Jonathan Hui. 2018. **Understanding Feature Pyramid Networks for object detection (FPN)**. [Online]. Available: https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c.

Keng Surapong. 2020. **MobileNet คืออะไร สอน TensorFlow.js สร้าง Image Classification จำแนกรูปภาพ จาก MobileNet โมเดลสำเร็จรูป – tfjs ep.7**. [Online]. Available: https://www.bualabs.com/archives/3439/tensorflow-js-tutorial-build-image-classification-javascript-mobilenet-pretrained-model-tfjs-ep-7.

Moe Alam. 2021. **SHINOBIDOCS**. [Online]. Available: https://shinobi.video/docs/start.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use. Forbidden to modify the content, and cite the document when use.

Mr.P L. 2018. **เริ่มเรียน Machine Learning 0–100 (Introduction)**. [Online]. Available: https://medium.com/mmp-li/เริ่มเรียน-machine-learning-0-100-introduction-1c58e516bfcd.

Natthawat Phongchit. 2018. **Convolutional Neural Network (CNN) คืออะไร**. [Online]. Available: https://medium.com/@natthawatphongchit/มาลองดูวิธีการคิดของ-cnn-กัน-e3f5d73eebaa.

Natthawat Phongchit. 2020. **มาทำความรู้จัก ResNet กันดีกว่า**. [Online]. Available: https://medium.com/@natthawatphongchit/มาทำความรู้จั่ก-resnet-กันดีกว่า-aec3a8c10793.

Pakpoom Thaweesitthichat. 2018. **Deep Learning มันทำงานอย่างไรกันนะ?**. [Online]. Available: https://medium.com/@pakpoom.thawee/deep-learning-มันทำงานอย่างไรกันนะ-8eef57561d19.

Raspberry Pi Foundation. 2015. **ABOUT US**. [Online]. Available: https://www.raspberrypi.org/about/.

Raspberry Pi Trading Ltd. 2019. **Raspberry Pi 4 Computer Model B**. [Online]. Available: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf.

Serdar Yegulalp. 2018. **What is TensorFlow? The machine learning library explained**. [Online]. Available: https://www.infoworld.com/article/3278008/tensorflow/what-is-tensorflow-the-machine-learning-library-explained.html.

Shenzhen VStarcam Technology Co., Ltd. **Specifications Model : C61S**. [Online]. Available: https://www.vstarcam.com/support/wp-content/uploads/2021/01/C61S-SPEC_EN_20200805.pdf.

Sornpraram Xu. 2021. **รู้จักกับ EfficientDet หนึ่งในโมเดล Object Detection**. [Online]. Available: https://medium.com/super-ai-engineer/รู้จักกับ-efficientdet-หนึ่งในโมเดล-object-detection-cd0ac67f1f9b.

Subham Kapiswe. 2019. **5 Useful Python Libraries For Machine Learning Projects**. [Online]. Available: https://www.technotification.com/2019/08/5-useful-python-libraries-for-machine-learning-projects.html.

Yu-hui Chen. 2021. **TensorFlow 2 Detection Model Zoo**. [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.