ระบบตรวจจับโดรนด้วยการเรียนรู้ของเครื่องจากภาพ

DRONE DETECTION BY VISUAL MACHINE LEARNING

ธนภัทร แจ่มแจ้ง

มงคล สมานญา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

ระบบตรวจจับโดรนด้วยการเรียนรู้ของเครื่องจากภาพ

DRONE DETECTION BY VISUAL MACHINE LEARNING

ธนภัทร แจ่มแจ้ง
มงคล สมานญา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจจับโดรนด้วยการเรียนรู้ของเครื่องจากภาพ

DRONE DETECTION BY VISUAL MACHINE LEARNING

ผู้จัดทำ

1. นายธนกัทร แจ่มแจ้ง รหัสนักศึกษา 58010505
2. นายมงคล สมานญา รหัสนักศึกษา 58010998

อาจารย์ที่ปรึกษา

(ผศ. ดร.สุรินทร์ กิตติธรกุล)

ระบบตรวจจับโดรนด้วยการเรียนรู้ของเครื่องจากภาพ

นายธนกัทร แจ่มแจ้ง 58010505
นายมงคล สมานญา 58010998
ผศ. ดร.สุรินทร์ กิตติธรกุล อาจารย์ที่ปรึกษา
ปีการศึกษา 2561

บทคัดย่อ

โครงงานนี้จัดทำขึ้นเพื่อศึกษาและออกแบบการพัฒนาระบบตรวจจับโดรนโดยใช้เทคนิคการเรียนรู้ของเครื่อง เพื่อเป็นระบบที่ใช้ในการแก้ปัญหาโดรนบินรุกล้ำเข้าไปในพื้นที่หวงห้ามหรือสถานที่ที่ต้องการความปลอดภัยสูง เช่น สนามบิน ซึ่งอาจทำให้เกิดอันตรายต่อเครื่องบินได้ โดยโครงงานนี้ต้องการให้ระบบสามารถทำงานบน Raspberry Pi 3 ได้ เพราะต้องการให้ระบบติดตั้งง่าย และมีราคาถูก ทั้งนี้ระบบยังมีส่วนติดต่อกับผู้ใช้ที่สามารถตรวจดูภาพวีดีโอผลลัพธ์ที่ได้ทำการประมวลผลแล้ว และผู้ใช้จะได้รับการแจ้งเตือนหากมีการตรวจจับโดรนได้ ซึ่งระบบจะส่งรูปภาพและข้อมูลช่วงเวลาที่ตรวจจับได้

Drone Detection by Visual Machine Learning

Mr. Tanapat	Jamjang	58010505
Mr. Mongkhon	Samanya	58010998
Asst.Prof.Dr. Surin	Kittitornkun	Advisor

Academic Year 2018

ABSTRACT

This project is initiated to study and to develop Drone detection using visual machine learning techniques. The purpose of this system is to solve problems of Drone invasion into restricted areas or confined spaces; for example, airports because drones can be harmful for aircrafts. The system is able to work on Raspberry Pi 3 because it is low-priced and easy for installation. Furthermore, the system has a web interface which allows the users to access live streaming video for human inspections. Also, it will notify whenever the system detects a Drone's invasion with captured images in real time.

กิตติกรรมประกาศ

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

ขอขอบคุณอาจารย์และบุคลากรต่าง ๆ ในสาขาวิศวกรรมคอมพิวเตอร์ที่ให้ความรู้ คำแนะนำ และช่วยเหลือในการทำโครงงงานครั้งนี้มาโดยตลอด ขอขอบคุณรุ่นพี่และเพื่อน ๆ ทุกคนที่ให้คำปรึกษา คำแนะนำ และคอยแบ่งปันความรู้ต่าง ๆ ที่ทำให้โครงงานนี้เสร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ ขอขอบคุณ บิดา มารดา และครอบครัว ที่ได้เลี้ยงดู อบรม สั่งสอน และให้การสนับสนุน ในทุก ๆ ด้าน และคอยเป็นกำลังใจให้เสมอมา

ธนภัทร แจ่มแจ้ง

มงคล สมานญา

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตาราง หน้า

สารบัญรูป

สารบัญรูป (ต่อ)

รูป หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

อากาศยานไร้คนขับ (Unmanned aerial vehicle) หรือที่รู้จักกันในนามของ "โดรน" คือ อากาศยานที่ไม่มีผู้ขับอยู่บนตัวอากาศยาน สามารถควบคุมระยะไกล ผ่านโปรแกรมควบคุมการบิน หรือขับเคลื่อนอัตโนมัติด้วยระบบฝังตัวซึ่งจะเป็นระบบที่มีความซับซ้อนกว่า โดยในอดีตโดรนถูกนำมาใช้ในงานทางด้านการทหารเป็นหลัก สามารถทำภารกิจที่มีความเสี่ยงสูงได้ เพราะนักบินไม่ต้องเสี่ยงชีวิตเพื่อการทำภารกิจนั้น หรือยังสามารถใช้โดรนในการกิจสอดแนมหรือการโจมตีได้อีกด้วย ซึ่งในปัจจุบัน โดรน เริ่มนำมาใช้กันอย่างแพร่หลายในจุดประสงค์อื่นในภาคพลเรือน นอกจากทางด้านทหาร เช่น ค้นหาและกู้ภัย, การตรวจสภาพอากาศ, การตรวจการจราจร, ดับเพลิง ตลอดจนถึง โดรนส่วนตัว ที่ถูกออกแบบให้มีขนาดเล็ก น้ำหนักเบา ทำให้เป็นที่นิยมสำหรับคนทั่วไป สามารถนำไปใช้ประโยชน์ได้หลายรูปแบบ เช่น รับส่งพัสดุ บินสำรวจพื้นที่ หรือเพื่อความบันเทิง อย่างไรก็ตามการใช้งานอย่างแพร่หลายนี้จะมาพร้อมกับปัญหาต่าง ๆ ทั้งในเรื่องของ การสร้างปัญหารบกวน การละเมิดสิทธิส่วนบุคคลโดยการบินเข้าไปในพื้นที่ส่วนตัว การบินเข้าไปในพื้นที่หวงห้าม เช่น สถานที่ราชการ หน่วยงานเอกชน หรือการบินเข้าไปในสถานที่ที่ต้องการความปลอดภัยสูง เช่น สนามบิน อาจทำให้เกิดอันตรายต่อการเครื่องบินได้ และโดรนส่วนมากจะมีกล้องติดเอาไว้ สามารถถ่ายภาพมุมสูงเพื่อทำการสอดแนมได้ ซึ่งอาจเป็นภัยต่อความมั่นคงหากมีการบินเข้าไปในพื้นที่ราชการ หรือพื้นที่ที่เป็นความลับ

จากปัญหาที่กล่าวมา หน่วยงานที่ต้องการความปลอดภัยสูงอย่างกองทัพและสนามบิน จึงต้องมีการป้องกันและตรวจจับโดรนที่บินรุกล้ำเข้ามาในพื้นที่ตลอดเวลา ซึ่งระบบตรวจจับโดรนนี้จะช่วยในการจำแนกและตรวจจับโดรน จากภาพที่ได้จากกล้องวีดิโอที่ที่ติดตั้งไว้ โดยโครงงานนี้จะใช้บอร์ด Raspberry Pi 3 เป็นตัวประมวลผล เพราะต้องการให้ระบบสามารถติดตั้งได้ง่าย มีราคาถูกและมีประสิทธิภาพ นอกจากนี้ระบบยังมีส่วนแสดงผลสำหรับผู้ใช้ โดยผู้ใช้สามารถตรวจสอบภาพจากกล้องวีดีโอ และได้รับการแจ้งเตือนหากมีการตรวจจับโดรนได้ เป็นการลดภาระของเจ้าหน้าที่ทำให้ไม่ต้องตรวจสอบด้วยตัวเองตลอดเวลา

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาระบบตรวจจับโดรนอัตโนมัติที่สามารถจำแนกและตรวจจับโดรนที่บินเข้ามาในพื้นที่ได้
2) เพื่อพัฒนาระบบตรวจจับโดรนอัตโนมัติที่มีราคาถูก ติดตั้งได้ง่าย และสามารถนำไปใช้งานได้จริง
3) เพื่อศึกษาองค์ความรู้ในเรื่องของ Machine Learning ซึ่งเป็นเทคโนโลยีที่ใช้ในการจำแนกโดรน
4) เพื่อศึกษาหา Machine Learning Model ที่มีความเหมาะสมในการจำแนกโดรน

1.3 ขอบเขตของการทำงาน

1) จำแนกและตรวจจับโดรนเฉพาะรุ่นที่กำหนดคือ Skyhunter X8 จากภาพวีดีโอ
2) แสดงผลการตรวจจับผ่านทางส่วนติดต่อกับผู้ใช้ (User Interface)
3) แจ้งเตือนผู้ใช้เมื่อมีการตรวจจับโดรนได้

1.4 วิธีการดำเนินงาน

1) กำหนดวัตถุประสงค์และขอบเขตการทำงานของโครงงาน
2) ศึกษาเทคโนโลยีต่าง ๆ ที่ใช้ในการพัฒนาระบบ เช่น TensorFlow, Machine Learning, Convolution Neural Network, Raspberry Pi, Shinobi
3) ออกแบบและวิเคราะห์ระบบจากข้อมูล เนื้อหา และเอกสารที่เกี่ยวข้องที่ได้ทำการรวบรวมมา
4) ศึกษาและพัฒนาระบบตรวจจับโดรนโดยใช้ TensorFlow และแสดงผลผ่าน Shinobi

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้เกี่ยวกับการทำงานของ Machine Learning
2) ได้รับความรู้และทักษะในการใช้งาน TensorFlow
3) ได้รับความรู้ในการสอนและทดสอบโมเดลเพื่อทำงานด้าน Object detection
4) สามารถนำระบบตรวจจับโดรนด้วยการเรียนรู้ของเครื่องนี้ไปพัฒนาต่อเพื่อให้ใช้งานได้มีประสิทธิภาพมากขึ้น

1.6 ข้อจำกัดของโปรแกรม

1) สามารถจำแนกและตรวจจับได้เฉพาะโดรนรุ่นที่กำหนด คือ Skyhunter X8

2) คุณภาพของภาพในตอนกลางคืนไม่ดีพอ จึงทำให้มีโอกาสผิดพลาดในการจำแนกและตรวจจับโดรน
3) เนื่องจากใช้ Raspberry Pi 3ในการประมวลผล จึงอาจทำให้การประมวลผลไม่รวดเร็วพอ ทำให้เกิดการหน่วงเวลา (Delay) ในการตรวจจับ

1.7 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ ทำการแบ่งเนื้อหาทั้งหมดออกเป็น 5 บท ซึ่งได้แก่ บทนำ, ทฤษฎีและงานวิจัยเกี่ยวข้อง, การออกแบบและผลการพัฒนา, การทดลองและวิเคราะห์ผลการทดลอง และบทสรุปและข้อเสนอแนะ โดยในแต่ละบทมีรายละเอียด ดังนี้

บทที่ 1 บทนำ จะกล่าวถึง ที่มาและความสำคัญของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของการทำงาน วิธีการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับ ข้อจำกัดของโปรแกรม และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง จะกล่าวถึง ทฤษฎีและองค์ความรู้ต่าง ที่นำมาใช้ในการพัฒนาระบบ ได้แก่ Raspberry Pi, TensorFlow, Shinobi, Machine Learning, Deep Learning, Convolution Neural Network, Object detection model, Base network model และงานวิจัยที่เกี่ยวข้อง

บทที่ 3 การออกแบบและผลการพัฒนา จะกล่าวถึง ความต้องการของระบบ, Use case diagram, Sequence diagram, โครงสร้างของระบบ, การพัฒนาโปรแกรมตรวจจับโดรน และส่วนติดต่อกับผู้ใช้

บทที่ 4 การทดลองและวิเคราะห์ผลการทดลอง จะกล่าวถึง การทดลองเปรียบเทียบประสิทธิภาพของ Pre-trained model, การทดลองสตรีมภาพวีดีโอผลลัพธ์บน Web application และการทดลองระบบแจ้งเตือนผู้ใช้และระบบบันทึกไฟล์ผลลัพธ์ โดยจะกล่าวถึงรายละเอียดของแต่ละการทดลองคือวัตถุประสงค์, วิธีการทดลอง, ผลการทดลอง และการสรุปผลการทดลอง

บทที่ 5 บทสรุปและข้อเสนอแนะ จะกล่าวถึง บทสรุป, ปัญหาและอุปสรรค และแนวทางการพัฒนาต่อ

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 Raspberry Pi

รูป 2.1 โลโก้ของ Raspberry Pi

Raspberry Pi เป็นคอมพิวเตอร์ขนาดเล็ก ราคาถูก ขนาดเท่าบัตรเครดิต สามารถเชื่อมต่อกับจอคอมพิวเตอร์หรือทีวีที่รองรับ HDMI หรืออาจจะต่อผ่านสายสัญญาณวีดีโอปกติได้เช่นกัน และสามารถใช้งานร่วมกับเมาส์และคีบอร์ดพื้นฐาน Raspberry Pi สามารถทำเกือบจะทุกอย่างที่เดสก์ท็อปคอมพิวเตอร์ทำได้ เช่น การใช้งานอินเตอร์เน็ทและเล่นวีดีโอคุณภาพสูง ทำงานเอกสารต่าง ๆ หรือแม้แต่การเล่นเกม

Raspberry Pi ถูกพัฒนาในประเทศอังกฤษ โดย Raspberry Pi Foundation เพื่อการสอน Computer Science พื้นฐานในโรงเรียน โดยรุ่นดั้งเดิมได้รับความนิยมกว่าที่คาดการณ์ไว้มาก เนื่องจากมีราคาถูก ขนาดเล็ก และสามารถนำไปศึกษาการทำงานของคอมพิวเตอร์ พร้อมเขียนโปรแกรมอย่างง่ายได้ โดยระบบปฏิบัติการพื้นฐานของ Raspberry Pi มีชื่อว่า Raspbian ที่จะมีโปรแกรมและเกมบางส่วนให้ทดลองใช้งาน โดยสามารถเขียนโปรแกรมภาษา Python ง่าย ๆ ได้ทันที ทั้งนี้ Raspberry Pi ยังสามารถลงระบบปฏิบัติการอื่นได้ นอกจาก Raspbian ซึ่ง Raspberry Pi ในปัจจุบันมีหลายรุ่นสามารถแบ่งเป็นโมเดลหลัก ๆ ได้ 2 โมเดลคือ Model A และ Model B ซึ่งทั้ง 2 โมเดลมีคุณสมบัติทางเทคนิคที่ใกล้เคียงกัน แตกต่างกันเพียงบางส่วน โดย Raspberry Pi 3 Model B+ เป็นรุ่นล่าสุดในตระกูล Raspberry Pi 3 และเป็นรุ่นที่ใช้ในโครงงานนี้

รูป 2.2 บอร์ด Raspberry Pi 3 โมเดล B+

ตาราง 2.1 คุณลักษณะของบอร์ด Raspberry Pi 3 โมเดล B+

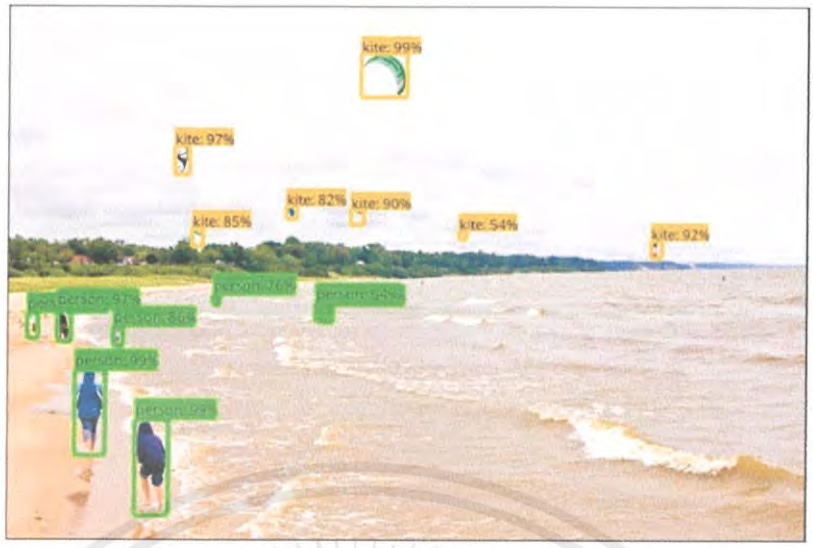
Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory	1GB LPDDR2 SDRAM
Connectivity	■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE ■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) ■ 4 × USB 2.0 ports
Access	Extended 40-pin GPIO header
Video & sound	■ 1 × full size HDMI ■ MIPI DSI display port ■ MIPI CSI camera port ■ 4 pole stereo output and composite video port
Multimedia	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support	Micro SD format for loading operating system and data storage
Input power	■ 5V/2.5A DC via micro USB connector ■ 5V DC via GPIO header ■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment	Operating temperature, 0–50°C

2.2 ไลบรารี TensorFlow

TensorFlow เป็น open source software library สำหรับการคำนวณเชิงตัวเลขประสิทธิภาพสูง ด้วยสถาปัตยกรรมที่ยืดหยุ่นทำให้ง่ายต่อการใช้งานในหลายแพลตฟอร์ม (CPUs, GPUs, TPUs) โดย เริ่มแรกถูกพัฒนาโดยนักวิจัยและวิศวกรจากทีม Google Brain ในองค์กร Google's AI แล้ว TensorFlow ก็กลายมาเป็นตัวช่วยที่มีประสิทธิภาพในการทำงานด้าน Machine Learning และ Deep learning หรืองานด้านคำนวณทางวิทยาศาสตร์อื่น ๆ โดยมี APIs ที่มีเสถียรภาพทั้ง Python API และ C APIs สำหรับผู้เริ่มต้นจนถึงผู้เชี่ยวชาญในการพัฒนาสำหรับเดสก์ท็อป, มือถือ, เว็บ และ cloud

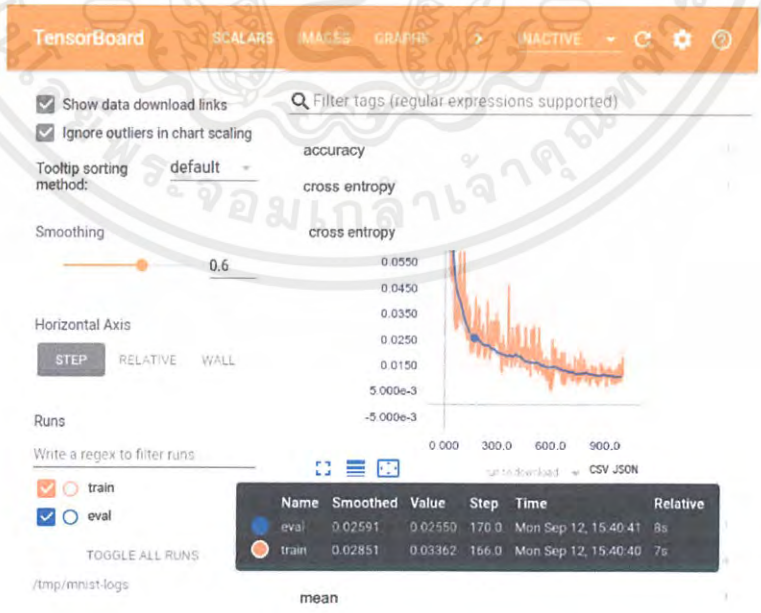
TensorFlow ให้ผู้พัฒนาสามารถสร้าง dataflow graph ซึ่งเป็นโครงสร้างที่อธิบายได้ว่าข้อมูลมี การเคลื่อนที่หรือเปลี่ยนแปลงไปมาอย่างไรผ่านกราฟ หรือชุดของ Node ที่ประมวลผล โดยแต่ละ Node ในกราฟแสดงถึง การดำเนินการทางคณิตศาสตร์ และเส้นเชื่อมระหว่าง Node คือ อาเรย์ของ ข้อมูลจำนวนหลายมิติ (multidimensional data) หรือ tensor ซึ่งทั้งหมดนี้ TensorFlow ทำในภาษา Python เพราะเป็นภาษาที่เรียนรู้ได้ง่าย โดย Node และ Tensors ใน TensorFlow คือ Python Objects และตัวแอพพลิเคชั่น TensorFlow ก็เป็น Python Application ประโยชน์ของ TensorFlow คือ เรา สามารถทำงานด้าน Machine learning โดยไม่ต้องรู้ถึงเบื้องหลังการทำงานที่ซับซ้อนซึ่งเป็นหน้าที่ ของ TensorFlow ที่จะคอยจัดการให้ นอกจากนี้ทาง Google ก็ยังได้พัฒนาโปรแกรม TensorBoard ซึ่งจะจำลองการทำงานของกระบวนการเรียนรู้ของ TensorFlow และยังมี TensorFlow Lite เฉพาะ สำหรับ Android พัฒนา TensorFlow Lite ให้สามารถใช้ได้กับ Android Oreo 8.1 ขึ้นไป โดยไฟล์ จะมีขนาดเล็ก สามารถรันได้เร็ว ซึ่งจะลดขนาดลงจาก TensorFlow ปกติ

API ของ TensorFlow ที่ใช้ในโครงงานนี้คือ TensorFlow Object Detection API โดยเป็น open source framework ในการทำงาน Machine Learning ที่เน้นในเรื่องของการ Classification และ Localizing วัตถุ โดยมี Model ต่าง ๆ มากมาย ทั้งที่เป็นทางการ และ Model ที่กำลังทำการวิจัยโดยทีม ของ TensorFlow อยู่ ซึ่งเราสามารถใช้ Model ที่ทาง TensorFlow ให้มา มาประยุกต์ใช้กับงานของเรา โดยเราสามารถทำ Transfer learning คือเทคนิคที่นำบางส่วนของ Model ที่มีการสอนและปรับค่า น้ำหนักพารามิเตอร์ต่าง ๆ ไว้แล้วมาสร้างเป็น Model ใหม่ เพราะการที่จะต้องเริ่มสอน Model ใหม่ จากศูนย์ต้องใช้จำนวนข้อมูลในการสอนที่มาก ใช้เวลานาน และการประมวลผลที่มากขึ้น



รูป 2.3 ผลลัพธ์ของการทำ Object detection ด้วย TensorFlow

TensorBoard เป็นเว็บแอพพลิเคชั่นสำหรับตรวจสอบ, แสดงผล (Visualization) และทำให้เข้าใจการทำงานของ TensorFlow ที่เรากำลังทำอยู่ เพราะในเรื่องของ Neural network นั้นมีความซับซ้อน เพื่อให้ง่ายต่อการทำความเข้าใจแก้ปัญหาและเพิ่มประสิทธิภาพโปรแกรม TensorFlow ทาง Google จึงได้พัฒนาชุดเครื่องมือนี้ โดยสามารถใช้ TensorBoard เพื่อสร้างภาพกราฟ TensorFlow ที่สร้างขึ้น วางแผนแปลงข้อมูลเมตริกเชิงปริมาณเกี่ยวกับการเรียกใช้กราฟของผู้ใช้งานและแสดงข้อมูลเพิ่มเติมเช่นภาพที่ส่งผ่าน ทำให้เราสามารถมีความง่ายในการตรวจสอบผลการเทรนว่าเป็นเช่นไรบ้าง



รูป 2.4 หน้าต่างแสดงผล TensorBoard

2.3 Shinobi

รูป 2.5 โลโก้ของ Shinobi

Shinobi คือ Open Source CCTV Solution เขียนด้วยเทคโนโลยี Node.js ออกแบบให้รองรับระบบบัญชีผู้ใช้ที่หลากหลายระดับ สตรีมด้วย Web socket และ บันทึกเป็น WebM โดย Shinobi สามารถบันทึกวีดีโอจาก IP Camera และ กล้องในเครื่องหลายๆตัวพร้อมกันได้ มีจุดเด่นดังนี้

- Time-lapse Viewer (ดูวิดีโอที่น่าสนใจนับไม่ถ้วนภายในไม่กี่นาที)
- 2-Factor Authentication
- เอาชนะขีดจำกัดของสตรีมที่กำหนดโดยเบราว์เซอร์
- Base64 (Stream Type)
- JPEG Mode (Option)
- บันทึกจาก IP Camera และกล้องภายในเครื่อง
- สตรีมด้วย WebSocket, HLS(include audio) และ MJPEG
- บันทึกเป็น WebM และ MP4
- สามารถบันทึกอัตโนมัติ
- ถ้าวีดีโอสตรีมสำเร็จจะสตรีมบนหน้าจอโดยไม่ต้องรีเฟรช
- 1 process สำหรับกล้องแต่ละตัวทั้งการ บันทึกและสตรีม
- กำหนดโซนในการตรวจจับการเคลื่อนไหว
- ลำดับเหตุการณ์ในการดูกิจกรรมการเคลื่อนไหวและวิดีโอ

- การมอนิเตอร์
- แก้ไขการมอนิเตอร์
- ลบวีดีโอ
- แบ่ง API key สำหรับบัญชีย่อย
- การมอนิเตอร์แบบกลุ่ม
- สามารถบันทึกรูปภาพจากสตรีมได้
- ควบคุมกล้องจากหน้าจอ interface
- Cron Filter สามารถตั้งค่าได้จาก Super Admin

2.4 การเรียนรู้ของเครื่อง (Machine Learning)

ในการจะกล่าวถึงความหมายของ Machine Learning เราควรรู้ความหมายของคำว่า ปัญญาประดิษฐ์ หรือ Artificial Intelligent ก่อน ซึ่งมีคำอธิบายต่างกันไป แต่ความหมายโดยรวมของปัญญาประดิษฐ์ คือ สิ่งที่มนุษย์สร้างขึ้นมาเพื่อให้สามารถคิดและทำอะไรบางอย่างแทนมนุษย์ได้ ตัวอย่างของปัญญาประดิษฐ์ เช่น ให้โปรแกรมทำนายว่าวันนี้หิมะจะตกหรือเปล่า โดยการป้อนข้อมูล ความชื้น อุณหภูมิ แรงลม หรือ โปรแกรมเล่นเกมอัตโนมัติที่เป็นการเขียนโปรแกรมรับข้อมูลรูปแบบสถานการณ์ในขณะนั้น และนำไปวิเคราะห์ว่า หากมีรูปแบบดังกล่าว ต้องตอบสนองอย่างไร ซึ่งในการสร้างปัญญาประดิษฐ์แบบดั้งเดิม ผู้เขียนโปรแกรมจะเป็นคนกำหนดเงื่อนไขทุกอย่างไว้ ว่าถ้าเจอข้อมูลแบบนี้ต้องตอบสนองอย่างไร แต่ปัญญาประดิษฐ์ลักษณะนี้จะไม่สามารถทำอะไรนอกเหนือจากเงื่อนไขที่ผู้เขียนโปรแกรมกำหนดไว้ได้ ซึ่งจะเป็นการดี หากมันสามารถเรียนรู้เองได้ เมื่อเจอรูปแบบที่ไม่เคยเจอมาก่อน มันก็จะสามารถทำนายได้ว่าควรตอบสนองอย่างไร จากการเรียนรู้ของมัน จึงเป็นที่มาของ การเรียนรู้ด้วยเครื่อง

Machine Learning หรือ การเรียนรู้ด้วยเครื่อง เป็นเทคนิคหนึ่งที่ใช้ในการสร้างปัญญาประดิษฐ์ ซึ่งปัญญาประดิษฐ์ไม่จำเป็นต้องใช้การเรียนรู้ด้วยเครื่องเสมอไป แต่หากใช้การเรียนรู้ของเครื่องในการสร้างก็จะทำให้ปัญญาประดิษฐ์มีประสิทธิภาพมากขึ้น

2.4.1 เทคนิคของ Machine Learning

2.4.1.1 Supervised Learning (การเรียนรู้โดยมีผู้สอน)

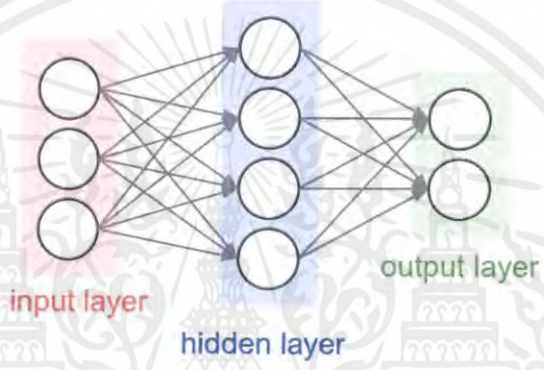
การเรียนรู้แบบนี้ เราจะป้อนข้อมูลขาเข้า คือตัวแปรต่าง ๆ พร้อมกับป้อนข้อมูลขาออก คือผลลัพธ์ที่ได้มาจากตัวแปรต้น และโปรแกรมก็จะไปทำการเรียนรู้สร้างความเชื่อมโยงกับข้อมูลที่ป้อนเข้าไป ทำให้สามารถทำนายได้ว่าหากมีข้อมูลขาเข้าเป็นเช่นนี้ จะมีข้อมูลขาออกเป็นอย่างไร

2.4.1.2 Unsupervised Learning (การเรียนรู้โดยไม่มีผู้สอน)

การเรียนรู้แบบไม่มีผู้สอน เราจะป้อนข้อมูลขาเข้า เท่านั้น ไม่ป้อนข้อมูลขาออก โดยโปรแกรมจะทำการเรียนรู้ด้วยตัวเอง ตัวอย่างเช่น การแบ่งกลุ่มสีของเม็ดสีน้ำเงินและสีแดงตามค่าสี หากใช้เทคนิคการเรียนรู้แบบนี้ โปรแกรมจะสามารถแบ่งกลุ่มสีออกมาได้เป็นสองกลุ่มคือกลุ่มของสีน้ำเงินและกลุ่มของสีแดง ซึ่งหากเราหยิบเม็ดสีแดงมาหนึ่งเม็ดให้โปรแกรมทายว่าเม็ดสีนี้เป็นสีอะไร โปรแกรมจะตอบได้เพียงว่า เม็ดสีนี้อยู่ในกลุ่มที่สองคือกลุ่มของสีแดง แต่ไม่สามารถบอกได้ว่า สีนี้เป็นสีแดงเพราะไม่ได้สอน วิธีการแบ่งกลุ่มข้อมูลด้วยการเรียนรู้แบบไม่มีผู้สอนนั้นมีชื่อเรียกว่า การแบ่งกระจุกข้อมูล (data clustering)

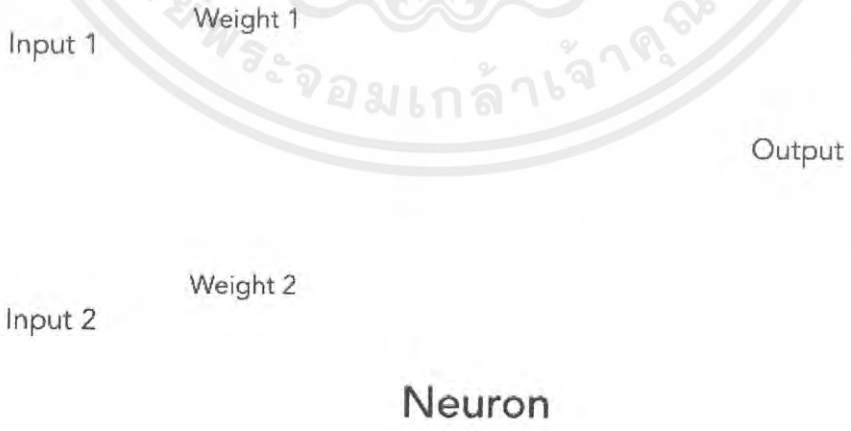
2.5 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก หรือ Deep Learning เป็นสาขาหนึ่งของการเรียนรู้ด้วยเครื่อง โดยเลียนแบบมาจาก เซลล์ประสาทของมนุษย์ เทคนิคนี้มีพื้นฐานมาจากเพอร์เซปตรอน โดยนำเซลล์ประสาทหลาย ๆ เซลล์มาเรียงต่อกันเป็นชั้น ๆ จำนวนมาก เป็นโครงข่ายประสาทเทียม หรือ Artificial neural network (ANN) ตัวอย่างของ ANN ง่าย ๆ คือ เมื่อตาเราเห็นสุนัข รูปสุนัขนี้จะถูกนำไปประมวลผล และได้คำตอบมาว่า สิ่งนี้คือสุนัข ถ้าเราจำลอง ANN จะได้ 3 ส่วน คือ input คือสิ่งที่ตาเราเห็น ในที่นี้คือสุนัข, Hidden layer เปรียบได้กับสิ่งที่ระบบเซลล์ประสาทในสมองเราประมวลผล และ Output layer คือ ผลลัพธ์ที่สรุปออกมาให้เรารู้ว่า สิ่งนี้เป็นสุนัข ซึ่งทั้งสามส่วนนี้ จะเชื่อมโยงกัน



รูป 2.6 โครงสร้างของ Artificial neural network

ส่วนที่เล็กที่สุดของ Neural Network คือ Neuron ซึ่งทำหน้าที่คำนวน Input ที่เข้ามา เพื่อให้ได้ผลลัพธ์ออกไป โดยมีส่วนประกอบสำคัญดังนี้

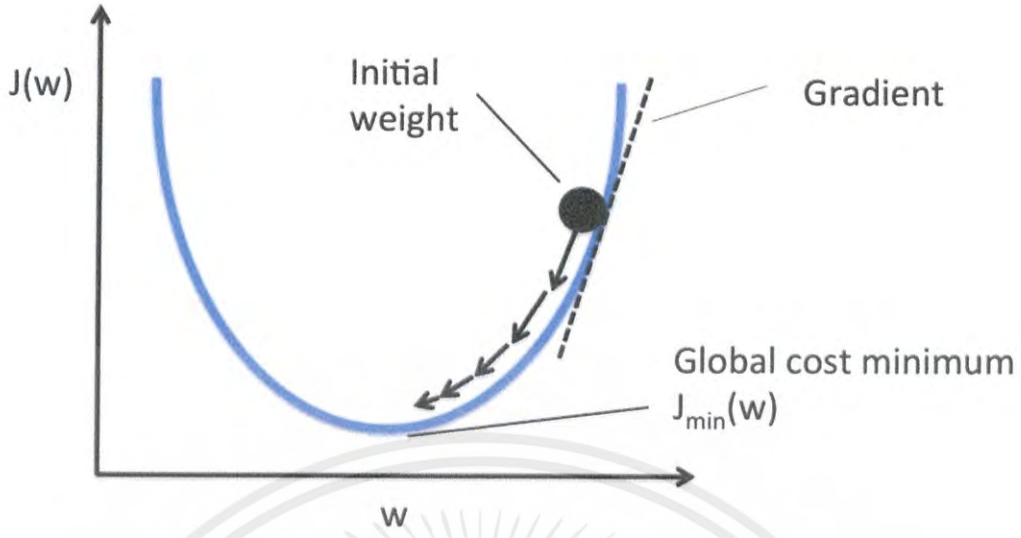


รูป 2.7 โครงสร้างของนิวรอน (Neuron)

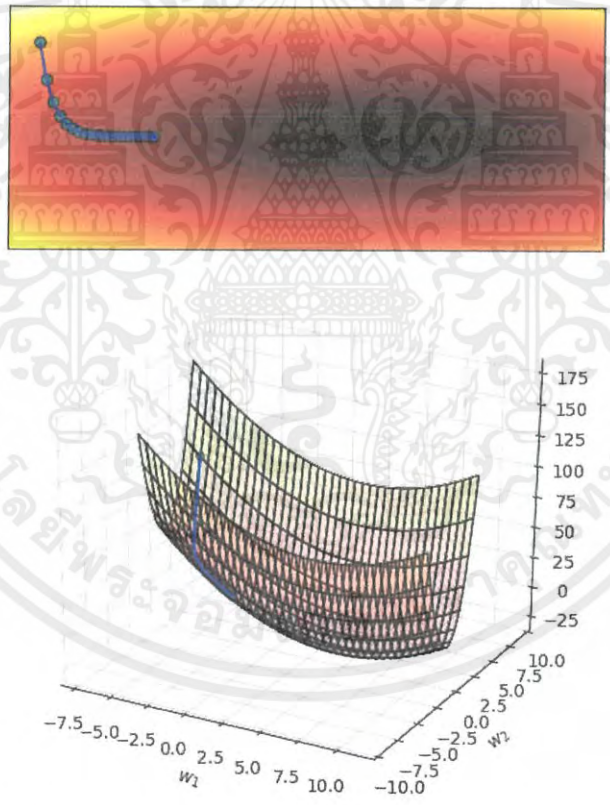
- Input คือ ค่าที่ส่งเข้ามาที่ Neuron สามารถมีขาเข้าได้หลายขา
- Weight เป็นการให้น้ำหนักของขาแต่ละที่ส่งเข้ามา โดยมีค่าระหว่าง 0-1 เมื่อเริ่มต้นจะเป็นการสุ่ม ขึ้นมา จากนั้นตัว Neuron เมื่อทำการเรียนรู้เรื่อย ๆ จะมีการปรับ weight เพื่อให้ได้ Output ใกล้เคียงกับคำตอบมากที่สุด
- Bias คือค่าที่จะช่วยทำให้ค่าที่เข้ามาอยู่ในระหว่าง 0 - 1 ได้ โดยจะเป็นเลขสุ่มและปรับไปเรื่อย ๆ ทุกครั้งที่เรียนรู้
- Output คือผลลัพธ์

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

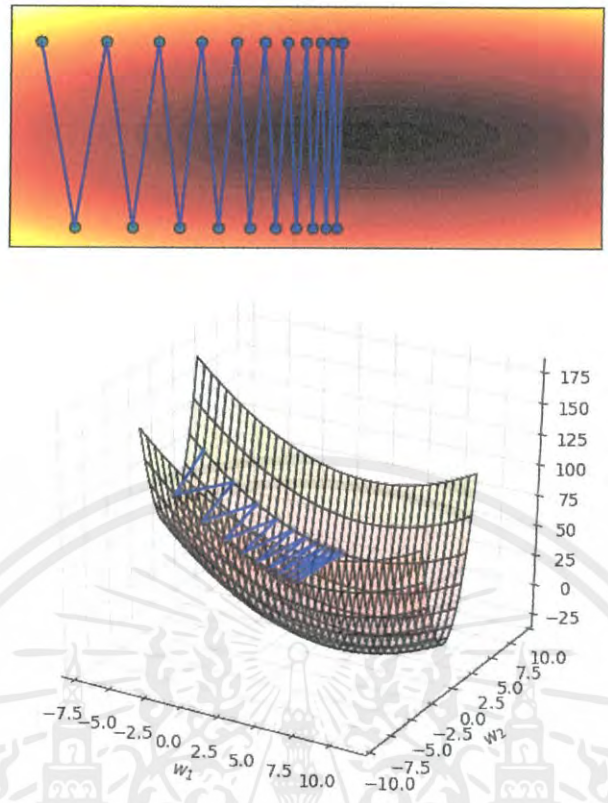
ในการลด Cost function เราจะใช้เทคนิคที่เรียกว่า Gradient Descent หรือการเคลื่อนลงตามความชัน ที่จะใช้ในการหาค่าต่ำสุดของฟังก์ชัน ในที่นี้คือ Cost function โดยจะทยอยปรับค่า Weight จาก Cost function โดยการหาอนุพันธ์ของ cost function และเราจะเห็นว่ามันเคลื่อนที่ไปในทิศทางใด การเคลื่อนที่ครั้งหนึ่งจะเคลื่อนไปเท่าไหร่นั้น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



รูป 2.8 การเคลื่อนลงตามความชันไปหาจุดต่ำสุด

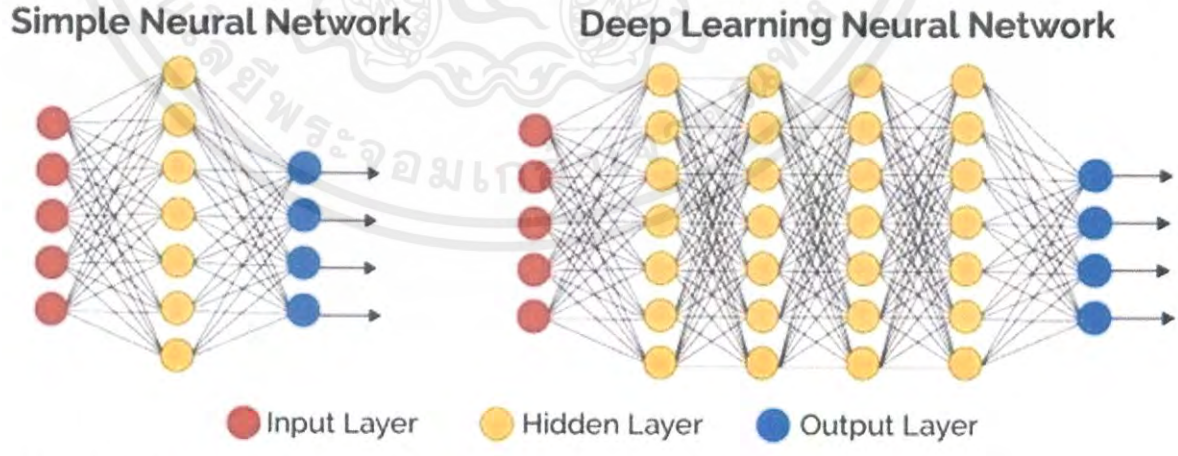


รูป 2.9 การกำหนด Learning rate ที่มีขนาดน้อยเกินไป



รูป 2.10 การกำหนด Learning rate ที่มีขนาดมากเกินไป

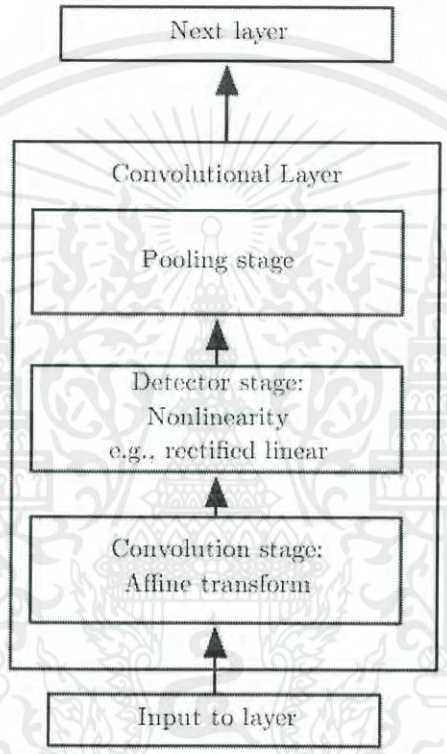
หากเรามี Neuron หรือ เซลล์ที่ทำหน้าที่วิเคราะห์และประมวลมาก ก็จะทำให้สามารถวิเคราะห์อะไรที่ซับซ้อนได้ ซึ่งความแตกต่างระหว่าง ANN และ Deep learning คือ ระดับ hidden layer ใน Deep learning จะมีมากกว่า



รูป 2.11 เปรียบเทียบโครงสร้างของ Simple Neural Network และ Deep Neural Network

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) หรือ โครงข่ายประสาทแบบคอนโวลูชัน เป็นโครงข่ายประสาทเทียมชนิดหนึ่ง ซึ่งจะต่างจาก โครงข่ายประสาทเทียมแบบธรรมดา คือ มีการใช้เทคนิค Convolution ที่จะจำลองการมองเห็นของมนุษย์ที่มองพื้นที่เป็นที่ย่อย ๆ และนำกลุ่มของพื้นที่ย่อย ๆ นั้นมาผสานกัน เพื่อวิเคราะห์ว่าสิ่งที่กำลังเห็นอยู่คืออะไร จึงทำให้ CNN ถูกนำไปใช้งานในการแก้ปัญหา Classification ที่เกี่ยวกับรูปภาพ ได้เป็นอย่างดี

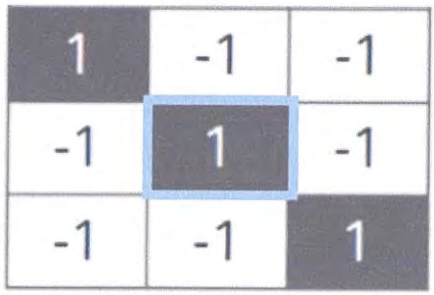


รูป 2.12 ส่วนประกอบหลักของ CNN ใน 1 Layer

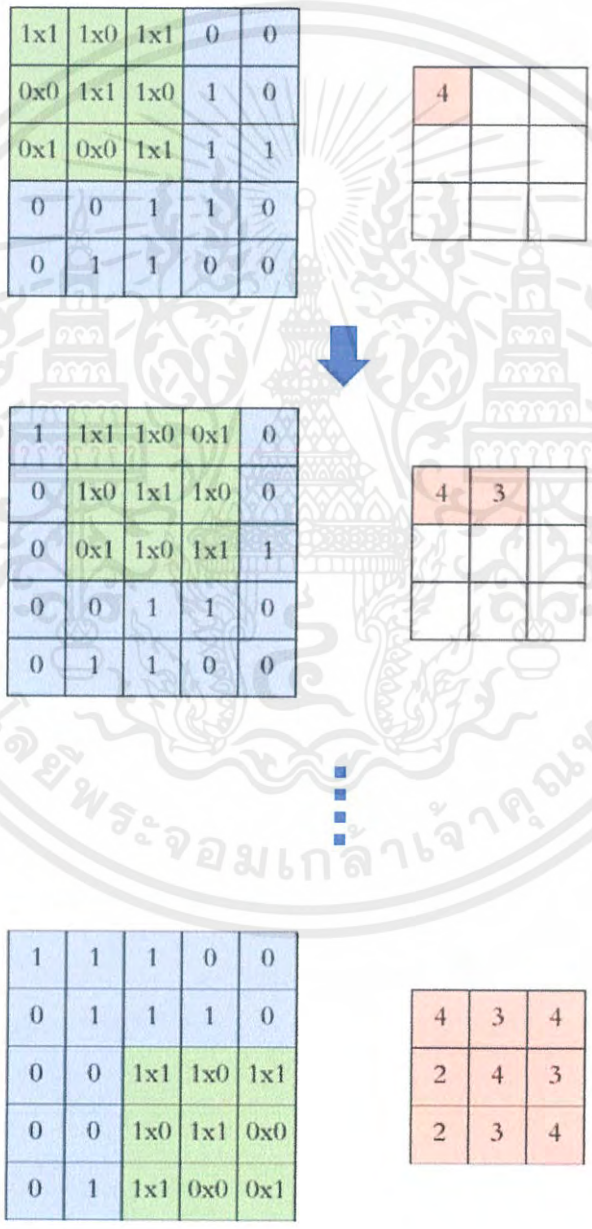
ในหนึ่ง convolutional layer ประกอบไปด้วย 3 ส่วนหลัก ๆ ได้แก่

1. Convolution stage

ขั้นตอนนี้จะสร้าง Filter หรือ Kernel มาสแกนรูปภาพที่เป็น input เพื่อทำ Feature map เพื่อแยกคุณลักษณะของรูป เช่น ขอบ รูปทรง สี โดยจะมี Stride เป็นตัวกำหนดว่าเราจะเลื่อน Kernel ไปด้วย Step เท่าไหร่

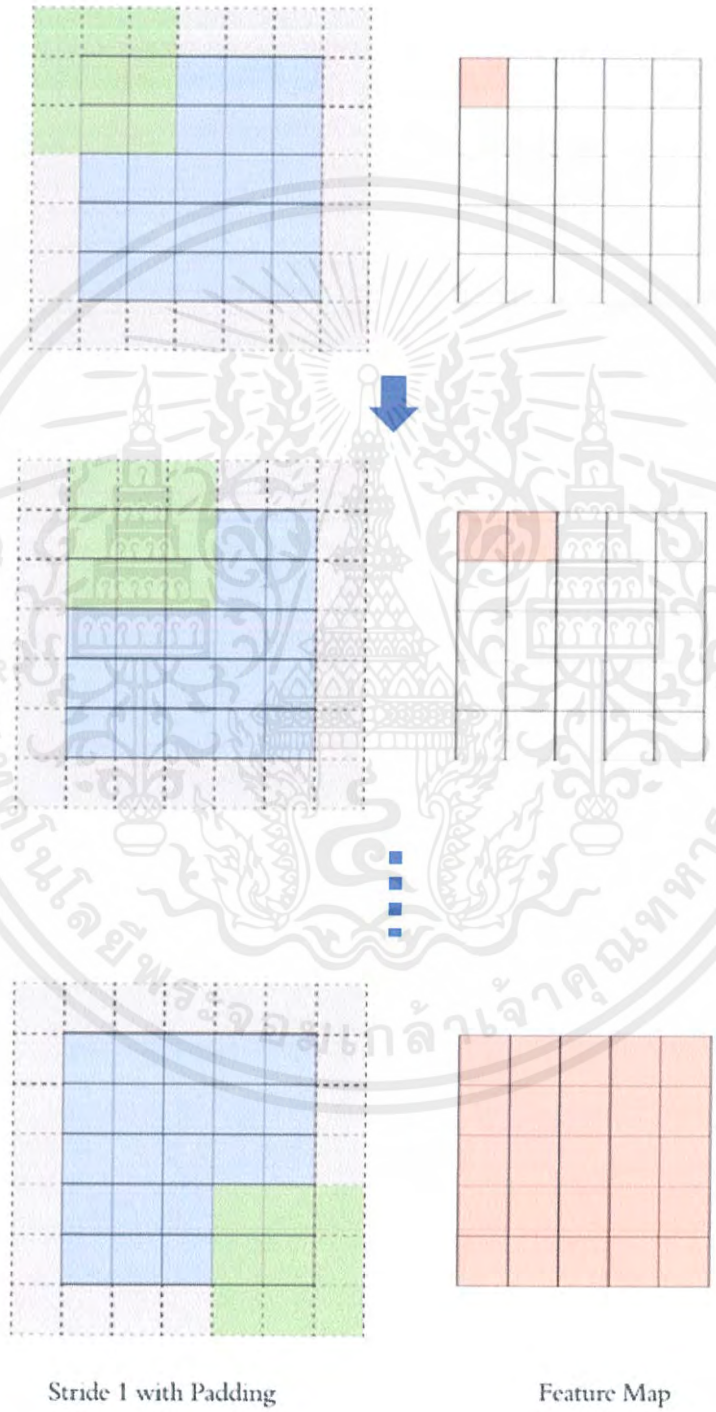


รูป 2.13 Kernel สำหรับหาเส้นตรงทะแยง



รูป 2.14 Feature map ด้วย Kernel 3x3 Stride 1

ในการเลื่อน Kernel ตามค่า Stride จะมีบางบริเวณที่ไม่สามารถเลื่อนได้ เพราะเกินขนาดของรูป โดยเฉพาะบริเวณขอบของรูป ซึ่งอาจทำให้ไม่สามารถเก็บรายละเอียดของภาพที่สำคัญได้ จึงต้องใช้ เทคนิค Padding ในการเพิ่มข้อมูลเช่นศูนย์หรือค่าต่าง ๆ เพื่อเพิ่มพื้นที่รูปบริเวณขอบ



รูป 2.15 วิธีการ Padding และ Stride 1

2. Detector stage

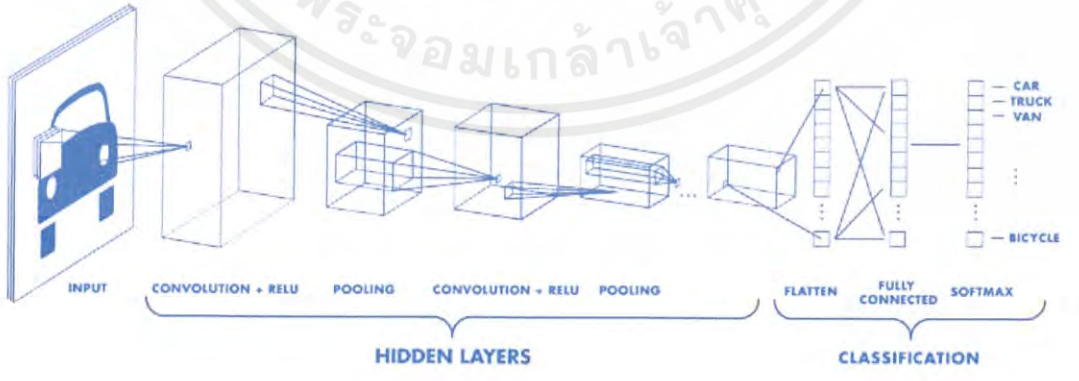
ขั้นนี้จะทำการนำ Output จาก Convolution stage มาแปลงให้อยู่ในรูปของ Non-linear โดนใช้ Activation เช่น Rectified linear Units (ReLU) เพื่อให้สะดวกในการคำนวณ

3. Pooling stage

Pooling เป็นการลดขนาดข้อมูลให้เล็กลง โดยที่เก็บรายละเอียดที่สำคัญของ Input เพราะในความเป็นจริง เรามีข้อมูลที่ต้องดำเนินการเยอะมาก หากเราไม่ทำการลดขนาดข้อมูล จะทำให้เสียเวลาและเปลืองทรัพยากรในการดำเนินการเกินความจำเป็น และยังเป็นการแก้ปัญหา Overfitting ได้ด้วย ซึ่งอัลกอริธึมที่นิยมใช้คือ L2 pooling และ Max pooling

รูป 2.16 Max pooling ขนาด 2x2 และ Stride 2

ใน layer ถัดไป ก็อาจทำซ้ำเดิม แต่ Layer สุดท้าย ต้องเป็น Fully-connected Neural Network เพื่อทำการ Classification ให้ได้ผลลัพธ์ออกมา



รูป 2.17 โครงสร้างของ CNN

2.7 Object detection model

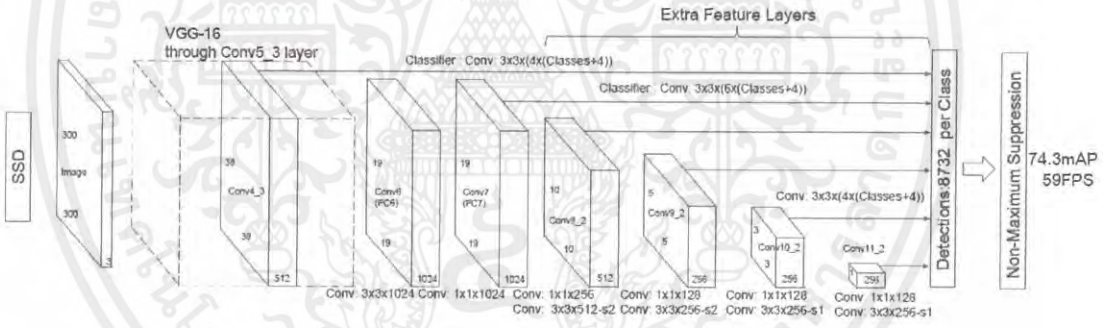
เป็นโมเดลที่ใช้ในงานด้าน Object detection โดยเฉพาะในการ Classification ว่าในรูปมีวัตถุใดบ้าง และ Localizing วัตถุเหล่านั้นว่าอยู่ในตำแหน่งใด โดยโมเดลต่อไปนี้จะมีอยู่ใน TensorFlow และโครงงานนี้เลือกใช้ในการทดลอง

2.7.1 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



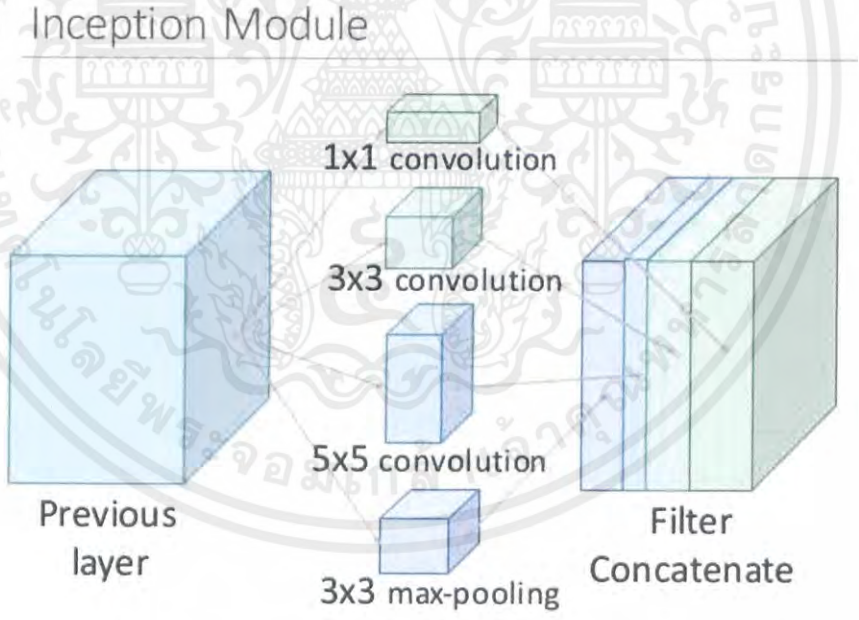
รูป 2.18 สถาปัตยกรรมของ SSD

2.8 Base Network Model

ในงานด้าน Object detection จะใช้ Model เหล่านี้ในการทำ Feature extraction หรือ การสกัดคุณลักษณะของวัตถุต่าง ๆ ในรูป แล้วนำ Feature map ที่ได้นี้ ไปทำงานต่อร่วมกับ Object detection model ซึ่ง โมเดลเหล่านี้มีอยู่ใน TensorFlow และได้รับการเทรนมาก่อนแล้ว สามารถนำไปทำ Transfer learning กับ Dataset ของเราได้ ซึ่ง Pre-trained model เหล่านี้ เป็นโมเดลที่โครงงานนี้ทดลอง

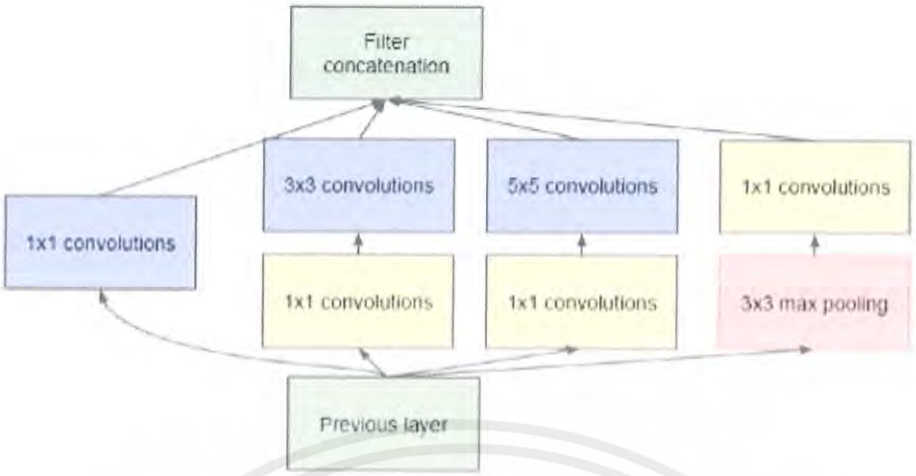
2.8.1 Inception

เป้าหมายหลักของ Inception คือ จะทำอย่างให้ในการขยาย Neural Network โดยไม่เพิ่ม Computational cost ซึ่งในงานวิจัยต้นฉบับได้มุ่งเน้นในการสร้าง Block สำหรับ deep network มีชื่อว่า Inception module โดยมีหลักการ คือ ในการทำ Convolutional ทำไมเราจะต้องเลือกเองว่าจะใช้ Transformation รูปแบบใด ทำไมไม่ให้ Model เป็นคนเลือก ดังนั้นตัว Inception module นี้จึงทำทั้ง 5x5 convolutional, 3x3 convolutional, 1x1 convolutional และ Max-pool นำมา stack กัน ได้ Output จะเป็น Input ใน Layer ต่อไป ซึ่งในชั้นต่อไป Model จะเป็นคนเลือกเองว่าจะใช้ Transformation รูปแบบใด



รูป 2.19 โครงสร้างของ Inception Module

แต่ปัญหาของแนวคิดนี้ คือ จะมี Computational cost ที่สูงมาก เพราะต้องทำ Transformation ทุกแบบ จึงแก้ปัญหาด้วยการทำ 1x1 convolution เพื่อลดมิติของข้อมูลก่อน แล้วจึงทำ Transformation รูปแบบอื่น ๆ เหมือนเดิม ซึ่งวิธีนี้จะช่วยลดการคำนวณได้ถึงประมาณ 10 เท่า



รูป 2.20 Inception Module ที่ทำการลดขนาดมิติข้อมูลก่อน

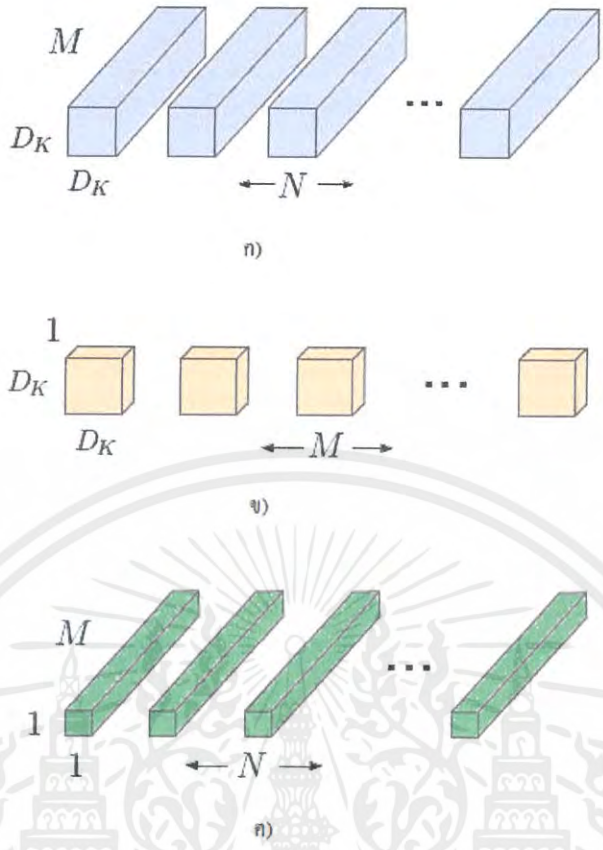
2.8.2 MobileNet

เป็นโมเดลที่ถูกออกแบบมาให้เหมาะกับการใช้งานบน Mobile และอุปกรณ์ขนาดเล็กที่มีกำลังในการประมวลผลไม่สูงมาก โดยจุดเด่นหลัก คือ การใช้ Depthwise separable convolution ที่ช่วยในการลดจำนวนพารามิเตอร์ที่ต้องคำนวณเมื่อเทียบกับการทำ convolution แบบปกติ ทำให้ MobileNet เป็น Deep neural network ขนาดเล็ก

Depthwise separable convolution จะทำการแยกจาก Convolution แบบปกติเป็น 2 layer คือ Depthwise convolution และ Pointwise convolution ที่เป็น 1x1 Convolutional filters

สำหรับการ Convolution แบบปกติ ถ้ามี input เป็น D_F, D_F, M และเราต้องการ N feature maps ด้วย Kernel ที่มีขนาด D_K, D_K, M จำนวน N ตัว เราจะได้ผลลัพธ์ที่มีขนาดเป็น D_G, D_G, N

สำหรับ Depthwise separable convolution ถ้ามี input เป็น D_F, D_F, M และเราต้องการ N feature เราจะทำ Depthwise convolution ก่อน คือ Kernel มีขนาดเป็น D_K, D_K, 1 จำนวน M ตัว จะได้ผลลัพธ์ที่มีขนาดเป็น D_G, D_G, M แล้วจากนั้นจึงทำ Pointwise convolution ด้วย Kernel ที่มีขนาด 1, 1, M จำนวน N ตัว เราจะได้ผลลัพธ์สุดท้ายที่มีขนาดเป็น D_G, D_G,N



รูป 2.21 การทำงานของ Depthwise Separable Convolution

ก) Convolutional Filter แบบปกติ ขนาด D_K, D_K, M จำนวน N ตัว

ข) Depthwise Convolutional Filter ขนาด D_K, D_K, 1 จำนวน M ตัว

ค) Pointwise Convolutional Filter ขนาด 1, 1, M จำนวน N ตัว

2.9 งานวิจัยที่เกี่ยวข้อง

2.9.1 Using Deep Networks for Drone Detection

เป็นงานวิจัยในการตรวจจับโดรนโดยใช้ Deep learning model คือ YOLO V2 โดยใช้ Dataset ที่มีความหลากหลาย และจำนวนที่มากพอ แล้วทำการทดลองเทรนโมเดล และบันทึกผลเพื่อวัดประสิทธิภาพของโมเดลที่ได้ สำหรับในอนาคตจะทำการพิจารณาในเรื่องของเวลา ให้ใช้เวลาในการประมวลผลน้อยลง [ปีที่พิมพ์: 2017, ผู้เขียน: Shahid Karim, Ye Zhang, Asif Ali Laghari, Muhammad Rizwan Asif]

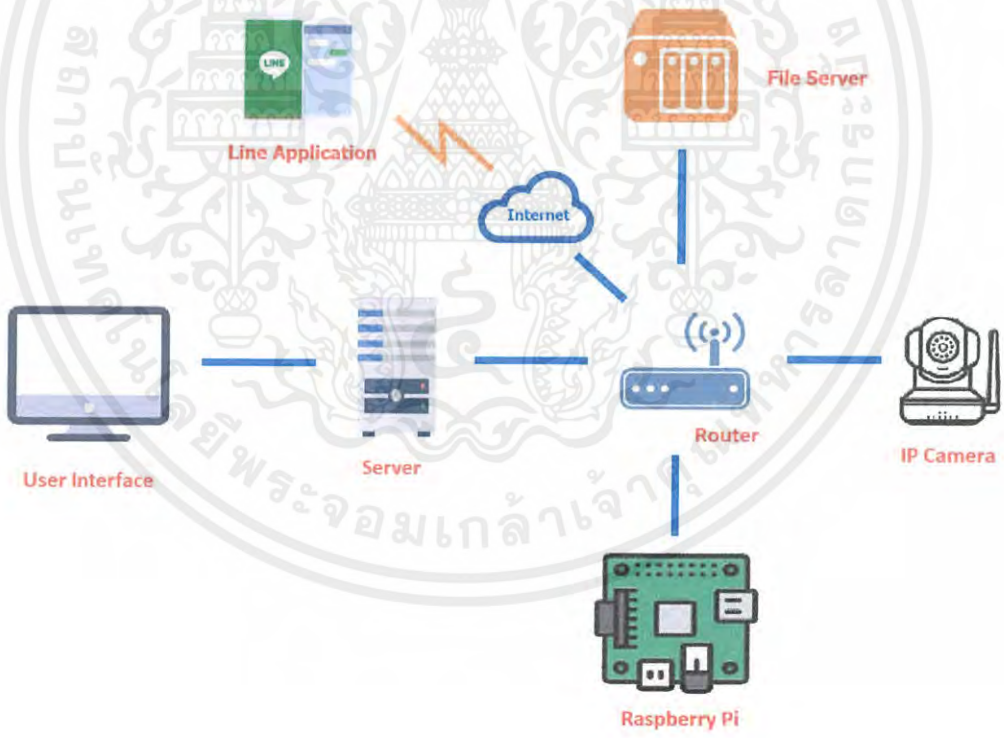
2.9.2 Image Processing Based Proposed Drone for Detecting and Controlling Street Crimes

วัตถุประสงค์ของงานวิจัยนี้ คือ การลดความเสี่ยงในการเกิดเหตุอาชญากรรม โดยการใช้โดรนเพื่อบินสำรวจและใช้เทคโนโลยี Image Processing ในการตรวจหาวัตถุที่คาดว่าจะเป็นอันตราย โดยใช้ shape detection algorithm ในการตรวจหาอาวุธ และใช้ SVM ในการ Classification [ปีที่พิมพ์: 2017, ผู้เขียน: Cemal Aker, Sinan Kalkan]

บทที่ 3
การออกแบบและผลการพัฒนา

3.1 โครงสร้างของระบบ

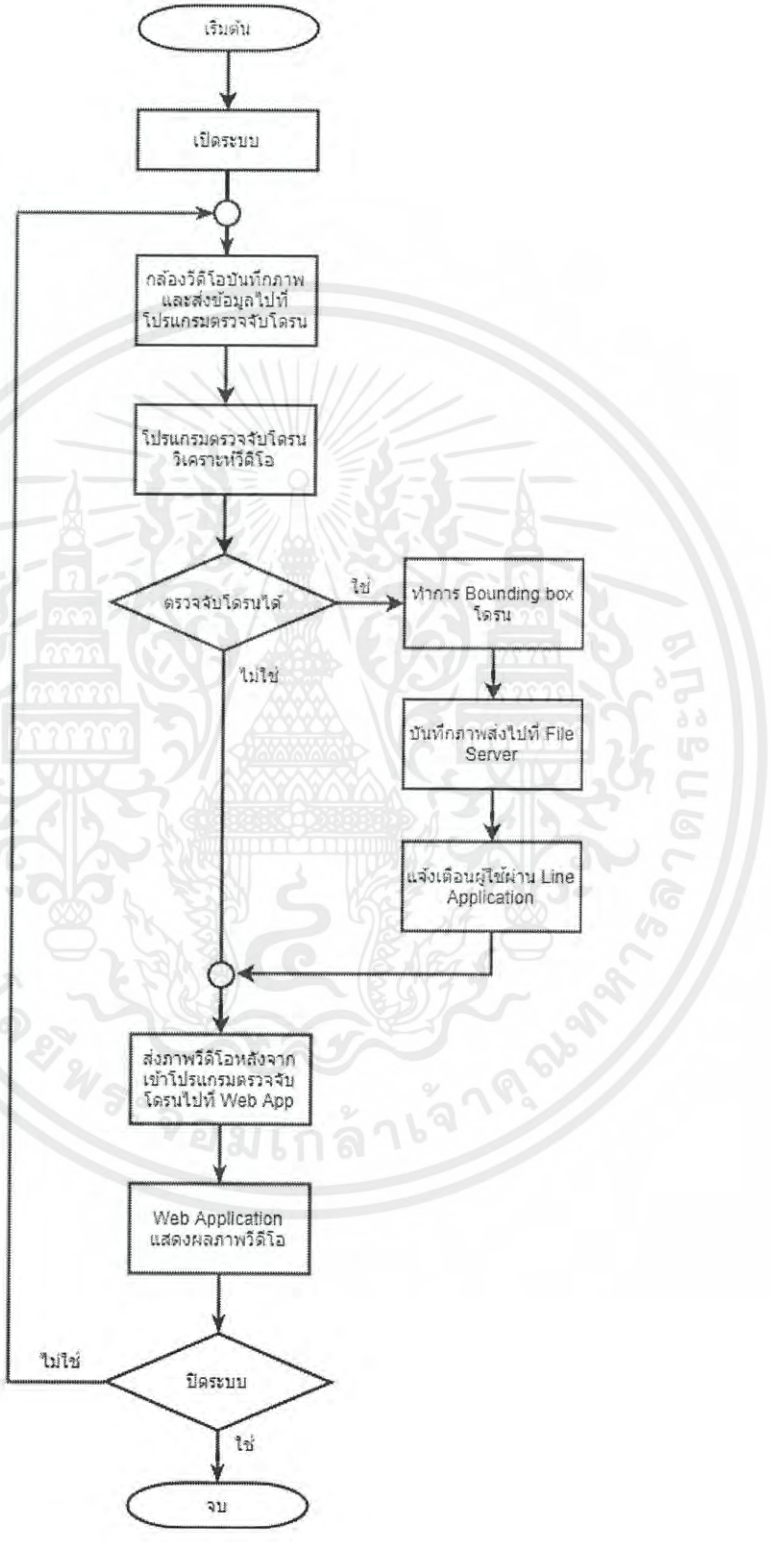
เริ่มต้นโปรแกรมตรวจจับโดรนที่ทำงานอยู่บนส่วนประมวลผล คือ Raspberry Pi 3 จะทำการรับข้อมูลภาพวีดีโอจากกล้อง IP และทำการประมวลผล ผลลัพธ์ที่ได้จะเป็นภาพวีดีโอต้นฉบับ ซึ่งหากมีการตรวจพบโดรนจะทำการ Bounding Box หรือตีกรอบรอบวัตถุ(โดรน) และทำการแจ้งเตือนผู้ใช้ โดยจะส่งภาพของโดรนพร้อมทั้งวันที่และเวลาที่ตรวจจับได้ไปให้ผู้ใช้ผ่านทาง Line Application พร้อมทั้งบันทึกรูปของโดรนที่ถูกตรวจจับได้ไปที่ File server เพื่อให้ผู้ใช้สามารถตรวจสอบได้ในภายหลัง จากนั้นจะทำการสตรีมวีดีโอที่ได้ไปที่ User Interface ที่เป็น Web Application เพื่อแสดงภาพวีดีโอที่ได้หลังจากเข้าสู่โปรแกรมตรวจจับโดรน



รูป 3.1 โครงสร้างของระบบ

3.2 โฟลวชาร์ต (Flowchart) การทำงาน

Flowchart แสดงการทำงานในภาพรวมของระบบ



รูป 3.2 Flowchart

3.3 ความต้องการของระบบ

3.3.1 input/output specification

3.3.1.1 Input specifications

- ภาพวีดีโอจากกล้อง IP camera ส่งข้อมูลไปยัง Raspberry Pi 3

3.3.1.2 Output specifications

- ทำการ Bounding Box โดรน หากมีการตรวจจับโดรนได้
- แสดงสตรีมมิ่งภาพวีดีโอบน User Interface
- แจ้งเตือนผู้ใช้หากมีตรวจจับโดรนได้
- บันทึกภาพโดรนที่ถูกตรวจจับได้ไว้ที่ File Server

3.3.2 User requirement

3.3.2.1 ส่วนรับข้อมูล (Input Device)

1. กล้องสามารถใช้งานได้ตลอด 24 ชั่วโมงทั้งกลางวันและกลางคืน
2. กล้องสามารถส่งภาพวีดีโอไปยังหน่วยประมวลผล คือ Raspberry Pi 3 ได้

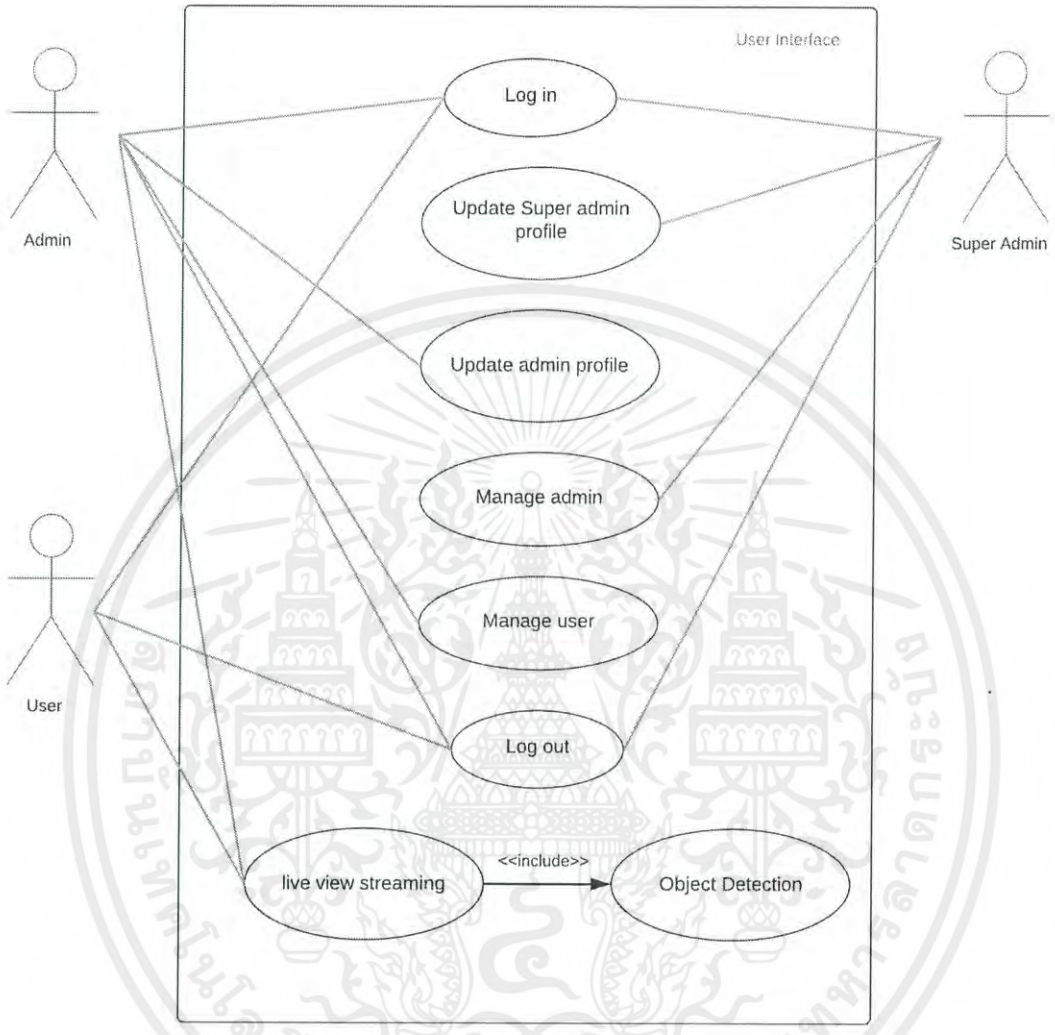
3.3.2.2 ส่วนประมวลผล (Processing Device)

1. ส่วนประมวลผล Raspberry Pi 3 สามารถรับข้อมูลภาพวีดีโอจากกล้อง
2. โปรแกรมตรวจจับโดรนจากภาพวีดีโอที่รับเข้ามาพร้อมทั้งทำ Bounding Box โดรน
3. โปรแกรมตรวจจับโดรนแจ้งเตือนผู้ใช้หากมีการตรวจจับโดรนได้
4. ส่วนประมวลผล สามารถส่งข้อมูลภาพวีดีโอ ที่เข้าสู่โปรแกรมตรวจจับโดรนแล้วไปยัง Server เพื่อทำการแสดงผลกับผู้ใช้
5. ระบบสามารถบันทึกภาพเมื่อตรวจจับโดรนได้ เพื่อให้ผู้ใช้ตรวจสอบภายหลัง

3.3.2.3 ส่วนแสดงผล (User interface)

1. ระบบสามารถแสดงภาพวีดีโอที่ทำการเข้าสู่โปรแกรมตรวจจับโดรนจากส่วนประมวลผลบน User Interface ได้
2. ระบบสามารถแจ้งเตือนผู้ใช้ผ่านทาง Line Application หากมีการตรวจจับโดรนได้

3.4 Use Case diagram



รูป 3.3 Use case diagram

3.4.1 รายละเอียด Use Case

ตาราง 3.1 ถึงตาราง 3.7 ต่อไปนี้จะอธิบายรายละเอียดของแต่ละ Use case

ตาราง 3.1 Login use case

Use Case	Log in
Actor	Admin, Super admin, User
Use case purpose	เป็นส่วนที่ให้ผู้ใช้เพื่อเข้าสู่ระบบ
Main Flow	1. กรอก ชื่อผู้ใช้ และ รหัส 2. กดปุ่ม Login 3. เข้าสู่ระบบเสร็จสิ้น
Exceptional Flow	1. ใส่ ชื่อผู้ใช้ หรือ รหัส ไม่ถูกต้อง ระบบแจ้งผู้ใช้ว่า ชื่อผู้ใช้หรือรหัสผ่านไม่ถูกต้อง 2. ไม่มีชื่อผู้ใช้อยู่ในระบบ ระบบแจ้งผู้ใช้ว่า ไม่มีชื่อผู้ใช้อยู่ในระบบ

ตาราง 3.2 Update Super admin use case

Use Case	Update Super admin
Actor	Super admin
Use case purpose	เป็นส่วนที่ให้ Super admin ทำการอัพเดท ข้อมูลของตัวเอง
Main Flow	1. ตั้งค่าชื่อหรือรหัสผ่านใหม่ 2. ยืนยัน
Exceptional Flow	-

ตาราง 3.3 Update admin profile use case

Use Case	Update admin profile
Actor	Admin
Use case purpose	เป็นส่วนที่ให้ Admin ทำการอัพเดทข้อมูลของตัวเอง
Main Flow	1. ตั้งค่าชื่อหรือรหัสผ่านใหม่ 2. ยืนยัน
Exceptional Flow	-

ตาราง 3.4 Manage Admin use case

Use Case	Manage Admin
Actor	Super admin
Use case purpose	เป็นส่วนที่ให้ Super Admin ทำเพิ่มหรือลบแอคเคาท์ของ Admin
Main Flow	1. เพิ่ม/ลบ แอคเคาท์ Admin 2. ยืนยัน
Exceptional Flow	1. ผู้ใช้ Admin ซ้ำ ระบบแจ้งว่า ชื่อผู้ใช้ซ้ำ 2. ไม่มีผู้ใช้ Admin ที่จะลบอยู่ในระบบ ระบบแจ้งว่า ไม่มีชื่อผู้ใช้นี้อยู่ในระบบ

ตาราง 3.5 Manage user use case

Use Case	Manage User
Actor	Admin
Use case purpose	เป็นส่วนที่ให้ Admin ทำเพิ่มหรือลบแอคเคาท์ของ User
Main Flow	1. เพิ่ม/ลบ แอคเคาท์ของ User 2. ยืนยัน
Exceptional Flow	1. ผู้ใช้ซ้ำ ระบบแจ้งว่า ชื่อผู้ใช้ซ้ำ 2. ไม่มีผู้ใช้ที่จะลบอยู่ในระบบ ระบบแจ้งว่า ไม่มีชื่อผู้ใช้นี้อยู่ในระบบ

ตาราง 3.6 Log out use case

Use Case	Log out
Actor	Admin, Super admin, User
Use case purpose	เป็นส่วนที่ให้ผู้ใช้ออกจากระบบ
Main Flow	1. กดปุ่ม Log out 2. ออกจากระบบเสร็จสิ้น
Exceptional Flow	-

ตาราง 3.7 live view streaming use case

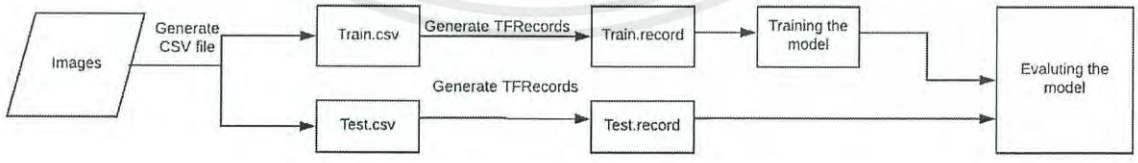
Use Case	live view streaming
Actor	Admin, User
Use case purpose	เป็นส่วนที่ให้ผู้ใช้สามารถดูภาพวีดีโอหลังจากเข้าสู่โปรแกรมตรวจจับโดรนที่ส่งมาจาก Raspberry Pi 3
Main Flow	1. แสดงภาพวีดีโอ
Exceptional Flow	1. ระบบภายในผิดพลาด

3.5 การพัฒนาโปรแกรมตรวจจับโดรน

โปรแกรมตรวจจับโดรนเขียนด้วย open source software library คือ TensorFlow โดยใช้ TensorFlow Object Detection API ด้วยภาษา Python ในการทำ Machine Learning โดยเราต้องสร้าง Model แล้วทำการเทรน เพื่อใช้ในการ Classification วัตถุที่เราต้องการ ซึ่งในที่นี้คือโดรน โดยทาง TensorFlow มี Model ที่ทำการเทรนและปรับค่าน้ำหนักของพารามิเตอร์ต่าง ๆ ไว้แล้ว โดยเราจะนำ Pre-trained model นั้นมาทำการ Transfer learning คือการนำ Pre-trained model นั้นมาเทรนต่อจาก Check Point เดิมหรือจุดที่โมเดลนั้นถูกเทรนไว้ล่าสุดกับ Dataset ใหม่ของเรา ซึ่ง Pre-trained model เป็น model ที่มีการปรับค่าน้ำหนักด้วยการสอนจาก Dataset จำนวนมาก

เมื่อเราเลือก Pre-trained model ที่จะนำมาใช้เทรนต่อได้แล้ว เราต้องเตรียม Dataset ของเรา โดยแบ่งข้อมูลออกเป็น 2 ส่วนคือ ข้อมูลสำหรับการเทรนประมาณ 80% และข้อมูลสำหรับทดสอบประมาณ 20% โดยการเตรียมข้อมูลเพื่อนำไปใช้สอนโมเดลนั้นต้องทำให้อยู่ในรูปแบบที่ TensorFlow เข้าใจ เริ่มแรกให้ Label ตำแหน่งของโดรนด้วยโปรแกรม LabelImg ซึ่งจะเป็นการเก็บพิกัดของตำแหน่งโดรนในรูปภาพ ได้ผลลัพธ์ออกมาเป็นไฟล์นามสกุล XML แล้วเราต้องแปลงไฟล์นามสกุล XML นี้เป็น ไฟล์นามสกุล CSV เพื่อนำไปแปลงต่อให้เป็น TFrecord ซึ่งเป็น File Format ของ TensorFlow

เมื่อเตรียมไฟล์ config และ label map เรียบร้อยแล้วก็เริ่มทำการเทรนโมเดลได้ ในระหว่างการเทรนเราสามารถตรวจสอบค่า loss ต่าง ๆ เช่น Classification loss, localization loss และ Total loss ที่เป็นค่าความผิดพลาดต่าง ๆ ได้ เพื่อใช้ดูว่าการเทรนโมเดลของเรานั้นเป็นไปในทิศทางใด รวมทั้งสามารถ evaluate โมเดลเราขณะเทรน ด้วย TensorBoard โดยเราจะเทรนจนกว่าค่า Total loss น้อยที่สุดและเริ่มไม่มีการเปลี่ยนแปลงค่า Total loss แล้ว เมื่อเทรนเสร็จแล้ว เราต้องทำการ save model หรือเรียกว่าการ freezing graph คือการบันทึกค่าพารามิเตอร์สำคัญที่ถูกปรับค่าจากการเทรน เพื่อนำโมเดลนี้ไปใช้ต่อไป



รูป 3.4 ภาพรวมการเทรนโมเดล

3.5.1 การเตรียม Dataset

Dataset ที่เรานำมาใช้สอนโมเดล เราใช้รูปโดรน SkyHunter X8 ที่เราเตรียมไว้เองส่วนหนึ่ง และอีกส่วนหนึ่งนำมาจากคลิปวีดีโอ Youtube แล้วนำมาทำการแยกออกมาเป็นภาพนิ่ง และทำการเลือกรูปที่มีความหลากหลาย ทั้งขนาดและมุมของโดรนในภาพ โดยเราทำการแบ่งรูปที่ไว้ใช้เทรน 800 รูป และสำหรับทดสอบ 200 รูป รวม Dataset ทั้งหมด 1,000 รูป

เมื่อทำแบ่งข้อมูลสำหรับการเทรนและทดสอบแล้ว จึงทำการ Label ตำแหน่งของโดรนทุกรูปด้วยโปรแกรม LabelImg โดยจะได้เป็นไฟล์นามสกุล XML เก็บพิกัดตำแหน่งของโดรนในรูป และทำการแปลงข้อมูลนามสกุล XML เป็น ไฟล์นามสกุล CSV เมื่อได้ข้อมูลพิกัดตำแหน่งของโดรนในรูปแบบไฟล์ CSV แล้ว ให้แปลงข้อมูลนั้นเป็น File Format ของ TensorFlow คือ TFrecord โดยจะไฟล์ Output คือ train.record สำหรับข้อมูลเทรน และ test.record สำหรับข้อมูลทดสอบ

รูป 3.5 ตัวอย่างรูปจาก Dataset (1)

รูป 3.6 ตัวอย่างรูปจาก Dataset (2)

รูป 3.7 ตัวอย่างรูปจาก Dataset (3)

รูป 3.8 ตัวอย่างรูปจาก Dataset (4)

รูป 3.9 โปรแกรม LabelImg

3.5.2 การเลือก Pre-trained model

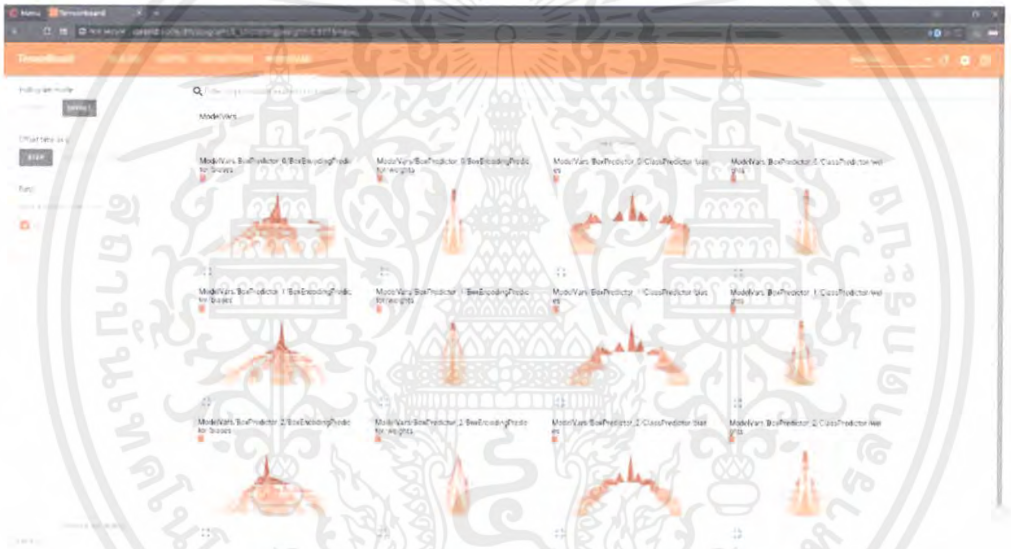
TensorFlow มี Pre-trained model สำหรับงานด้าน Object Detection ที่เทรนมาแล้วกับ Dataset มาตราฐานต่าง ๆ เช่น COCO dataset, Kitti dataset และ Open Images dataset ไว้ให้ใช้มากมาย และเราจะนำ Pre-trained model นั้น ไปทำ Transfer learning กับ Dataset ของเรา โดยเราเลือกโมเดลที่คาดว่าเหมาะกับงานเรามาทั้งหมด 4 โมเดลคือ SSD Inception V2, SSD MobileNet V2, SSD Lite MobileNet V2 และ Faster RCNN Inception V2 มาเปรียบทั้งความเร็วในการประมวลและความแม่นยำในการทำนาย โดยเรานำโมเดลทั้งสี่มาเทรนด้วย Dataset เดียวกัน เทรนในจำนวน Global step ที่ใกล้เคียงกัน (Global step ใน TensorFlow คือจำนวน Batches ที่เข้าสู่การเทรนจนถึงปัจจุบัน) และปรับค่า config ในการเทรนโมเดลเป็นค่าเดียวกัน เช่น Batch size และ learning rate

เมื่อทั้งสี่โมเดลถูกเทรนเรียบร้อยแล้ว ก็นำมาทดสอบประสิทธิภาพด้วยชุดข้อมูลทดสอบเดียวกัน และทำการคำนวณค่า Precision, Recall, Accuracy และ F score พร้อมทั้งทดสอบความเร็วในการประมวลต่อหนึ่งเฟรม (FPS) แล้วนำมาเปรียบเทียบกัน

3.5.3 การเทรนโมเดล

เมื่อเลือก Pre-trained model ที่จะนำมาทำ Transfer learning ได้แล้วก็ทำการแก้ไขค่าต่างๆ ในไฟล์ config สำหรับโมเดลนั้น ๆ ค่าที่ต้องมีการปรับแก้ คือ Batch size, learning rate และ Path ของโฟลเดอร์ Dataset รวมถึงโฟลเดอร์ที่เก็บค่าพิกัดตำแหน่งของโดรนในรูปซึ่งอยู่ในรูปแบบ TFrecord และสร้างไฟล์สำหรับทำ label map คือ การระบุคลาสที่มีทั้งหมด สำหรับโครงงานนี้มีเพียง 1 คลาส คือ โดรน ไฟล์นี้จะเป็นไฟล์นามสกุล pbtxt เมื่อทำการแก้ไขไฟล์ config และสร้างไฟล์ label map เรียบร้อยแล้วก็ทำการเริ่มเทรนโมเดลนั้น และคอยตรวจสอบการเทรนผ่าน Tensorboard เพื่อดูค่า loss ต่าง ๆ เช่น Classification loss, localization loss และ Total loss ที่เป็นค่าความผิดพลาดต่าง ๆ และทำการ evaluate โมเดลขณะเทรนได้ด้วย เพื่อดูว่าตอนนี้โมเดลเราสามารถ classification รูปทดสอบตัวอย่างได้ดีแค่ไหน

รูป 3.10 หน้าต่างแสดงผล Tensorboard แสดงค่า Loss ต่าง ๆ



รูป 3.11 หน้าต่างแสดงผล Tensorboard แสดงค่า Histogram ของพารามิเตอร์ต่าง ๆ

3.5.4 การบันทึกโมเดล

เมื่อเทรนโมเดลเสร็จเรียบร้อยแล้ว เราจะทำบันทึกโมเดลนั้น หรือที่เรียกว่า Freezing the graph เพื่อบันทึกค่าน้ำหนักพารามิเตอร์ต่าง ๆ ที่โมเดลได้มีการปรับค่า เป็น TensorFlow Graph

3.5.5 การนำโมเดลไปใช้งาน

เมื่อเราได้โมเดลที่เทรนเรียบร้อยแล้ว ก็นำไปใช้งาน โดยเขียนโปรแกรมที่รับข้อมูลภาพวีดีโอจากกล้อง IP แล้วนำข้อมูลภาพวีดีโอนั้นเข้าสู่โมเดลเพื่อทำการ classification และ localization หากตรวจพบโดรน ทีละเฟรม ซึ่งหากมีการตรวจพบโดรนโปรแกรมจะทำการ bounding box หรือตีกรอบสี่เหลี่ยมรอบโดรน ข้อมูลภาพวีดีโอหลังจากเข้าสู่โมเดลที่ทำการ classification และ

localization แล้วนั้น จะถูกส่งไปที่ server เพื่อทำการแสดงภาพวีดีโอผลลัพธ์ที่ได้จากการประมวลผลของโมเดล

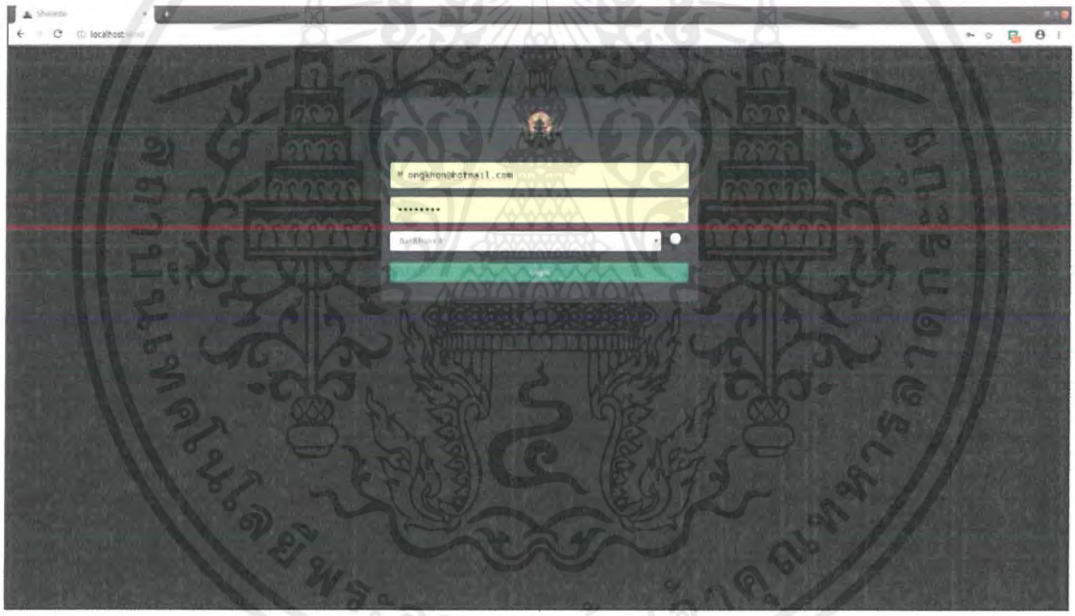
3.6 การพัฒนาส่วนติดต่อกับผู้ใช้ (User Interface)

ส่วนติดต่อกับผู้ใช้ที่เป็น Web Application จะเป็นส่วนแสดงผลภาพวีดีโอที่เข้าสู่โปรแกรมตรวจจับโดรนที่ทำงานบน Raspberry Pi 3 แล้วส่งข้อมูลนั้นต่อมายังส่วนติดต่อกับผู้ใช้นี้ โดยโครงงานนี้เลือกใช้ open source CCTV solution คือ Shinobi ในการแสดงผล

เมื่อผู้ใช้ต้องการใช้งาน ผู้ใช้จะต้องเข้าสู่ระบบ และเข้าไปตั้งค่าการเชื่อมต่อ เพื่อรับข้อมูลภาพวีดีโอที่ส่งมาหลังจากทำการประมวลผล

3.6.1 หน้าต่างลงชื่อเข้าใช้

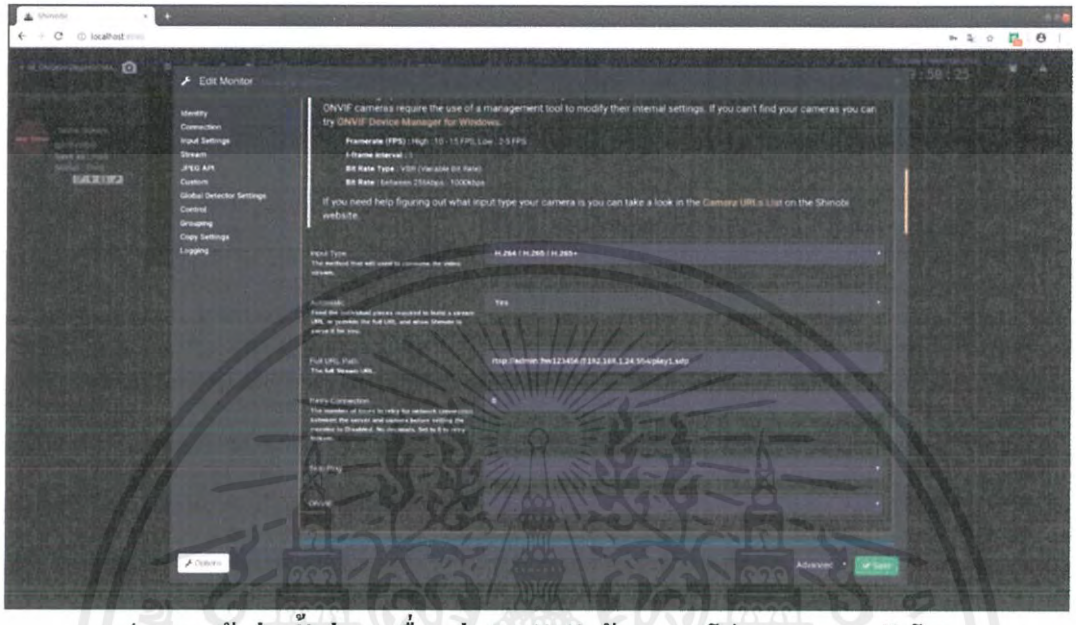
ผู้ใช้สามารถลงใช้เข้าใช้โดยใช้อีเมลและรหัสผ่าน



รูป 3.12 หน้าต่างลงชื่อเข้าใช้

3.6.2 หน้าต่างการตั้งค่าเชื่อมต่อ

เป็นหน้าสำหรับการตั้งค่า เพื่อให้ Shinobi รับข้อมูลภาพวีดีโอผลลัพธ์ที่ได้จากโปรแกรมตรวจจับโดรน



รูป 3.13 หน้าต่างตั้งค่าการเชื่อมต่อสำหรับรับข้อมูลจากโปรแกรมตรวจจับโดรน

3.6.3 หน้าต่างแสดงผลการสตรีม

ผู้ใช้สามารถดูภาพวีดีโอผลลัพธ์ที่ได้จากโปรแกรมตรวจจับโดรนที่ทำงานบน Raspberry Pi 3

รูป 3.14 การแสดงผลเมื่อการตรวจจับไม่พบโดรน

รูป 3.15 การแสดงผลการตรวจจับ เมื่อสามารถตรวจจับโดรนได้

3.7 การพัฒนาระบบแจ้งเตือนผู้ใช้

ระบบจะทำการแจ้งเตือนผู้ใช้หากมีการตรวจจับโดรนได้ ผ่านทาง Line Application เพราะเป็น Platform ที่สะดวกกับผู้ใช้ สำหรับการพัฒนาระบบการแจ้งเตือนด้วย Line Bot นี้ เราจะทำการลงทะเบียนกับ Line Notify ซึ่งเป็นบริการที่ทาง Line ได้เตรียมไว้ให้ในรูปแบบของ API เพื่อให้สามารถพัฒนาโปรเจค ที่มีความต้องการส่งข้อความในการแจ้งเตือนเข้าไปยัง กลุ่ม หรือบัญชีส่วนตัวของเราได้ โดยเมื่อเราได้ทำการลงทะเบียนแล้ว เราจะได้ Access Token เพื่อสร้าง Line Bot ที่เราจะนำไปใช้งาน โดยเราจะทำการเพิ่ม Line bot ไปที่ Group Chat ที่สร้างขึ้นมา ซึ่งผู้ใช้สามารถเพิ่มแอคเคาท์ Line ของตัวเองเข้าไปในกลุ่มนั้น ๆ เพื่อรับการแจ้งเตือนจาก Line bot ดังกล่าว

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

บทที่ 4
การทดลองและวิเคราะห์ผลการทดลอง

4.1 การทดลองเปรียบเทียบประสิทธิภาพของ Pre-trained model

4.1.1 วัตถุประสงค์

เพื่อเปรียบเทียบประสิทธิภาพของ Pre-trained model ที่คาดว่าจะสามารถทำงานได้บน Raspberry Pi 3 เพื่อคัดเลือกโมเดลที่เหมาะสมที่สุดในการนำมาใช้ในการตรวจจับโดรน

4.1.2 วิธีการทดลอง

1) เลือก Pre-trained model ใน TensorFlow ที่จะทำการเปรียบเทียบ โดยคัดเลือกโมเดลที่จะนำมาเปรียบเทียบเบื้องต้นจากตารางแสดงค่าความแม่นยำ (mAP) และ Speed ของโมเดลจากเว็บไซต์ GitHub ของ TensorFlow ในที่นี้เลือก SSD Inception V2, SSD MobileNet V1, SSD MobileNet V2 และ SSD Lite MobileNet V2 เนื่องจาก มีค่า mAP ที่อยู่ในระดับปานกลาง เช่นเดียวกับ Speed ที่คาดว่าเพียงพอจะสามารถทำงานบน Raspberry Pi 3 ได้
2) เตรียม Dataset ที่จะใช้สอนโมเดลทั้งหมด 1,000 รูป โดยแบ่งสำหรับการสอนจำนวน 800 รูป และ สำหรับทดสอบจำนวน 200 รูป
3) เริ่มเทรนโมเดลด้วยการตั้งค่าตัวแปรพื้นฐานให้เท่ากัน คือ Batch size, Learning rate และจำนวน step ที่ทำการเทรน
4) เมื่อเทรนโมเดลเสร็จเรียบร้อย ให้ทำการทดสอบโมเดล ด้วยรูปชุดทดสอบเดียวกันจำนวน 100 รูป แบ่งเป็นรูปที่มีโดรนจำนวน 50 รูป และรูปที่ไม่มีโดรนจำนวน 50 รูป และบันทึกผลการทดสอบ
5) ทดสอบค่า FPS หรือจำนวนเฟรมที่ประมวลผลได้ในหนึ่งวินาทีและทำการบันทึกผล
6) นำผลการทดสอบมาคำนวณเพื่อวัดประสิทธิภาพของโมเดล ได้แก่ Confusion matrix, Precision, Recall, Accuracy และ F-score
7) เปรียบเทียบผลการคำนวณของแต่ละโมเดล และคัดเลือกโมเดลที่ดีที่สุดที่เหมาะกับโครงงานนี้ โดยดูจากค่า Precision, Recall, Accuracy และ F-score รวมถึง FPS

4.1.3 ผลการทดลอง

ตาราง 4.1 ผลการทดสอบโมเดล

	ผลการทำนาย				
Image No	Actual	SSD Inception	SSD Mobilenet V1	SSD Mobilenet V2	SSD Lite Mobilenet V2
1	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
2	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
3	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
4	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
5	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
6	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
7	มีโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
8	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
9	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
10	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
11	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
12	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
13	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
14	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
15	มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ตรวจพบโดรน	ไม่พบโดรน
16	มีโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
17	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
18	มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
19	มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
20	มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
21	มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
22	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
23	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
24	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
25	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน

Image No	ผลการทำนาย				
	Actual	SSD Inception	SSD Mobilenet V1	SSD Mobilenet V2	SSD Lite Mobilenet V2
26	มีโดรน	ไม่พบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ตรวจพบโดรน
27	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
28	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
29	มีโดรน	ไม่พบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน
30	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
31	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
32	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
33	มีโดรน	ไม่พบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
34	มีโดรน	ไม่พบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
35	มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
36	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
37	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
38	มีโดรน	ตรวจพบโดรน	ไม่พบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
39	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
40	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
41	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
42	มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
43	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
44	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
45	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
46	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
47	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
48	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
49	มีโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
50	มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
51	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
52	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
53	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน

Image No	ผลการทำนาย				
	Actual	SSD Inception	SSD Mobilenet V1	SSD Mobilenet V2	SSD Lite Mobilenet V2
54	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
55	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
56	ไม่มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ตรวจพบโดรน
57	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
58	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
59	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
60	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
61	ไม่มีโดรน	ตรวจพบโดรน	ไม่พบโดรน	ตรวจพบโดรน	ไม่พบโดรน
62	ไม่มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
63	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
64	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
65	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
66	ไม่มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
67	ไม่มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
68	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
69	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
70	ไม่มีโดรน	ไม่พบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
71	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
72	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
73	ไม่มีโดรน	ไม่พบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
74	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
75	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
76	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
77	ไม่มีโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
78	ไม่มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
79	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
80	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
81	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน

Image No	ผลการทำนาย				
	Actual	SSD Inception	SSD Mobilenet V1	SSD Mobilenet V2	SSD Lite Mobilenet V2
82	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
83	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
84	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
85	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
86	ไม่มีโดรน	ตรวจพบโดรน	ไม่พบโดรน	ตรวจพบโดรน	ไม่พบโดรน
87	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
88	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
89	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
90	ไม่มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน
91	ไม่มีโดรน	ไม่พบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน
92	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
93	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
94	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
95	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
96	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน
97	ไม่มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
98	ไม่มีโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน	ตรวจพบโดรน
99	ไม่มีโดรน	ไม่พบโดรน	ตรวจพบโดรน	ไม่พบโดรน	ไม่พบโดรน
100	ไม่มีโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน	ไม่พบโดรน

ตาราง 4.2 ผลการคำนวณประสิทธิภาพโมเดล

	SSD Inception	SSD MobilenetV1	SSD MobilenetV2	SSD Lite MobilenetV2
TP	39%	39%	37%	36%
TN	39%	38%	42%	45%
FP	11%	12%	8%	5%
FN	11%	11%	13%	14%
Accuracy	0.78	0.77	0.79	0.81
Precision(Y)	0.78	0.76	0.82	0.88
Recall(Y)	0.78	0.78	0.74	0.72
F score	0.78	0.77	0.78	0.79
FPS	0.5	1	1	2

4.1.4 วิเคราะห์ผลการทดลอง

จากตาราง 4.2 จะเห็นได้ว่าทั้งสี่โมเดลที่เรานำมาทดสอบมีประสิทธิภาพที่ค่อนข้างใกล้เคียงกัน โดย SSD Lite Mobilenet V2 มี Accuracy และ F-Score ที่สูงที่สุด ตามด้วย SSD Inception และ SSD Mobilenet V2 ที่มีค่า F-score เท่ากันคือ 0.78 แต่ SSD Lite Mobilenet V2 มีค่า FPS สูงที่สุด คือ 2 ในขณะที่ SSD Mobilenet V1 และ SSD Mobilenet V2 มีค่า FPS คือ 1 และสุดท้าย SSD Inception มีค่า FPS เพียง 0.5 เท่านั้น

ดังนั้นเราจึงเลือก SSD Lite Mobilenet V2 เป็นโมเดลที่จะนำไปใช้ต่อ ถึงแม้จะมีค่า Accuracy และ F-Score ใกล้เคียงกับโมเดลอื่น แต่มีค่า FPS ที่สูงที่สุด ซึ่งเป็นปัจจัยสำคัญอย่างหนึ่งในการเลือกโมเดล

4.2 การทดลองสตรีมภาพวีดีโอผลลัพธ์บน Web Application

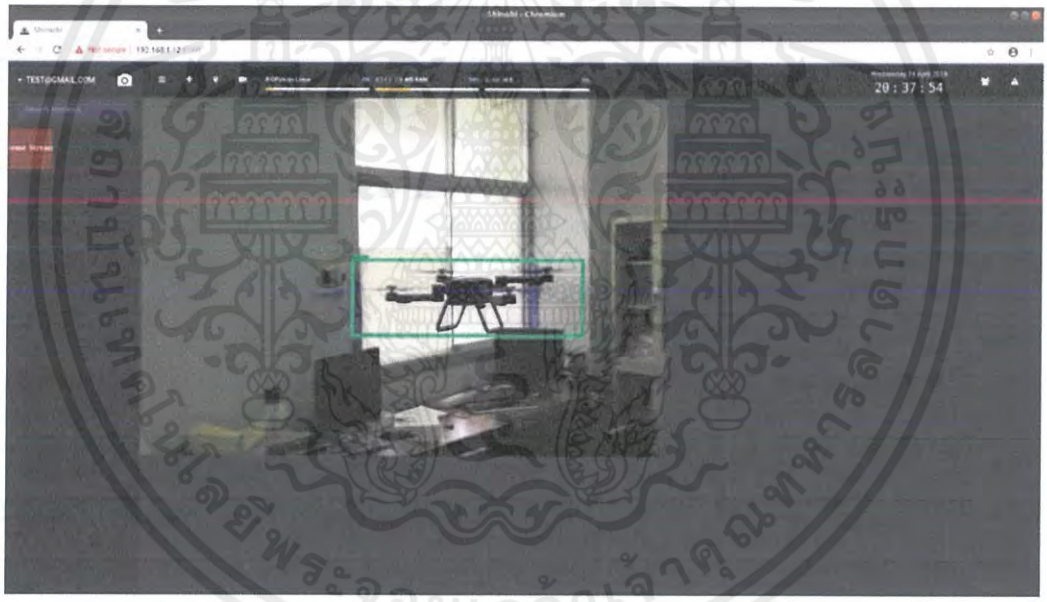
4.2.1 วัตถุประสงค์

เพื่อทดสอบโปรแกรมตรวจจับโดรนที่พัฒนาโดยใช้โมเดลที่เลือกนั่นคือ SSD Lite Mobilenet V2 และทดสอบการแสดงวีดีโอผลลัพธ์บน Web Application คือ Shinobi

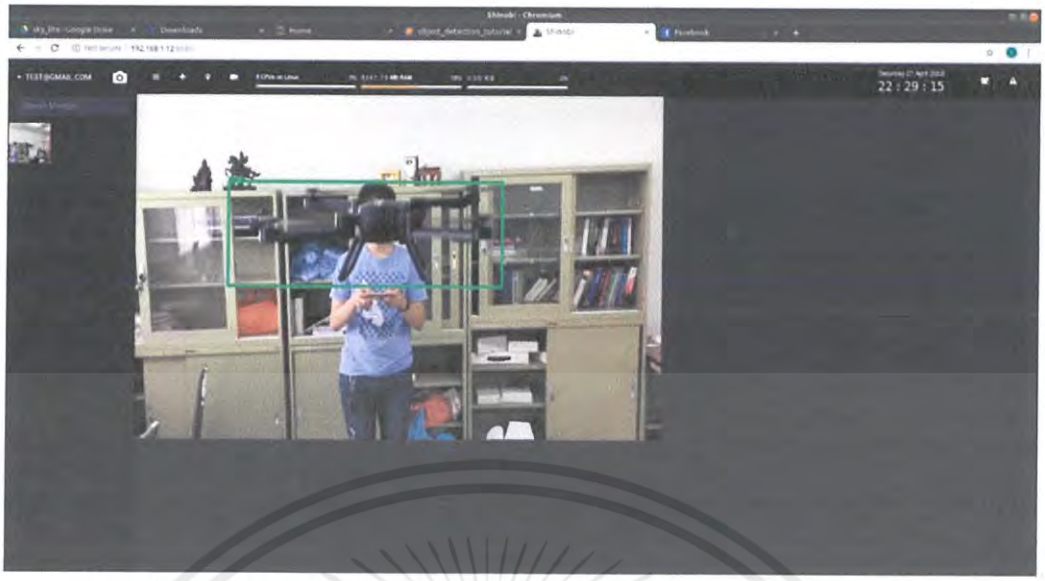
4.2.2 วิธีการทดลอง

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

4.2.3 ผลการทดลอง



รูป 4.1 ผลการทดลองสตรีมภาพวีดีโอผลลัพธ์บน Web application (1)



รูป 4.2 ผลการทดลองสตรีมภาพวีดีโอผลลัพธ์บน Web application (2)

4.2.4 วิเคราะห์ผลการทดลอง

จากผลการทดลอง สรุปได้ว่าโปรแกรมตรวจจับโดรนที่ใช้โมเดล SSD Lite Mobilenet V2 สามารถตรวจจับโดรนได้จากภาพวีดีโอที่ได้รับมาจาก IP camera และยังสามารถสตรีมภาพวีดีโอผลลัพธ์หลังเข้าโปรแกรมตรวจจับโดรนแล้วไปที่ Web Application เพื่อแสดงผลกับผู้ใช้ต่อไปได้

4.3 การทดลองระบบแจ้งเตือนผู้ใช้และระบบบันทึกไฟล์ผลลัพธ์

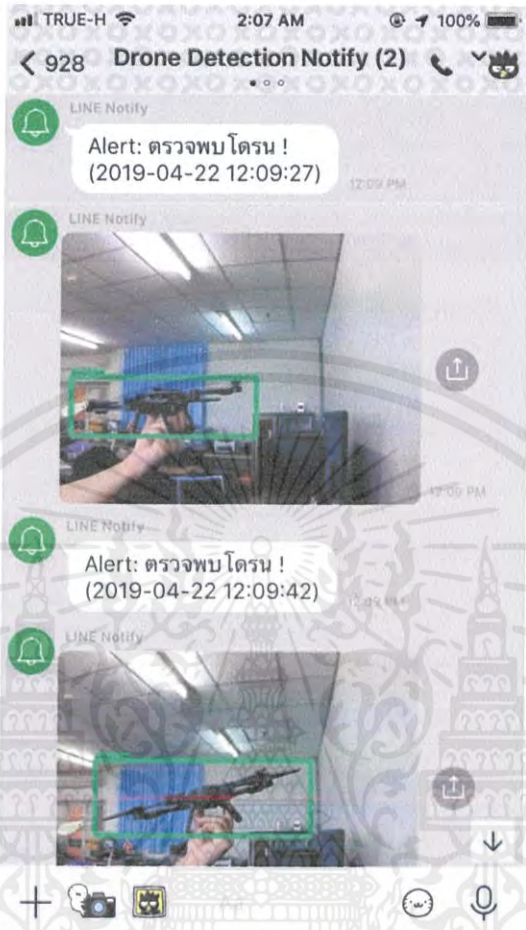
4.3.1 วัตถุประสงค์

เพื่อทดสอบระบบการแจ้งเตือนผู้ใช้ผ่าน Line Application และระบบบันทึกไฟล์ผลลัพธ์ที่ File Server ผ่าน FTP เมื่อตรวจจับโดรนได้

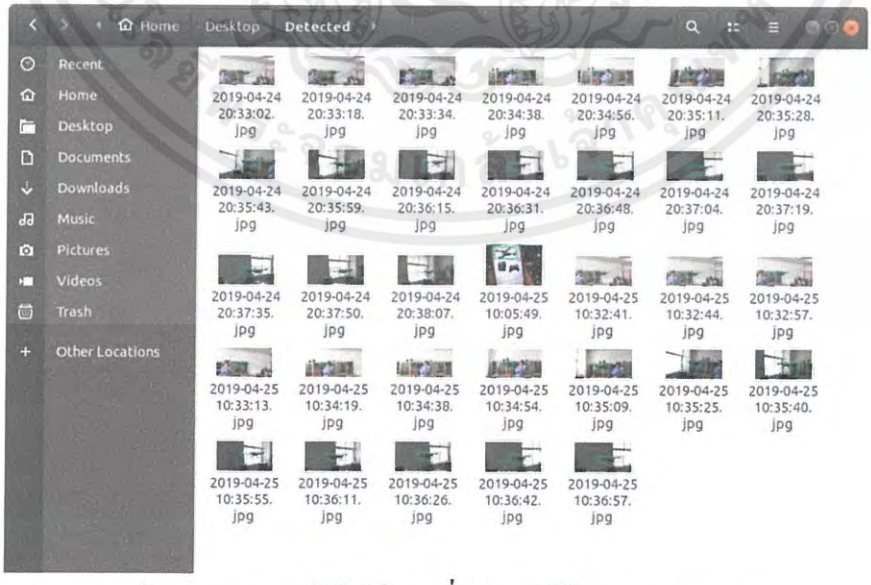
4.3.2 วิธีการทดลอง

1) ตั้งค่าโปรแกรมตรวจจับโดรนในส่วนของการแจ้งเตือนผู้ใช้ โดยกำหนด Access Token ให้ตรงกับ Line Bot ที่เราต้องการจะใช้
2) ตั้งค่าโปรแกรมตรวจจับโดรนในส่วนของส่งไฟล์ผลลัพธ์ผ่าน FTP โดยกำหนดให้ส่งข้อมูลไปที่ File Server
3) ทำให้โปรแกรมตรวจจับโดรน จับโดรนให้ได้
4) บันทึกผลการทดลอง

4.3.3 ผลการทดลอง



รูป 4.3 ผลการแจ้งเตือนผู้ใช้ผ่าน Line Application



รูป 4.4 ผลการบันทึกไฟล์ภาพที่ตรวจจับได้บน File Server

4.3.4 วิเคราะห์ผลการทดลอง

ระบบการแจ้งเตือนผู้ใช้ผ่าน Line Application สามารถทำงานได้ปกติ เช่นเดียวกับระบบบันทึกไฟล์ผลลัพธ์บน File Server

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

5.1.1 ส่วนประมวลบน Raspberry Pi 3

ระบบสามารถตรวจจับโดรนได้ด้วยโปรแกรมตรวจจับโดรนที่ประมวลผลอยู่บน Raspberry Pi 3 โดยใช้ Machine Learning Model คือ SSD Lite Mobilenet V2 ในการวิเคราะห์ภาพที่ได้จากกล้อง IP camera ซึ่งมีความแม่นยำของโมเดลนี้อยู่ที่ประมาณ 79% และสามารถส่งภาพวีดีโอผลลัพธ์ไปแสดงบน Web Application เพื่อแสดงแก่ผู้ใช้ต่อไป และสามารถแจ้งเตือนผู้ใช้ผ่าน Line Application และบันทึกรูปเมื่อตรวจจับโดรนได้ไว้ที่ File Server เพื่อให้ผู้ใช้ตรวจสอบในภายหลังได้

5.1.2 ส่วนติดต่อกับผู้ใช้

โครงงานนี้มี Web Application คือ Shinobi ซึ่งเป็น Open Source CCTV Solution ในการแสดงผลภาพวีดีโอผลลัพธ์ที่ได้จากโปรแกรมตรวจจับโดรนที่ประมวลอยู่บน Raspberry Pi 3 โดยผู้ใช้จะสามารถดูสตรีมภาพวีดีโอผ่านหน้าเว็บไซต์ได้อย่างสะดวก

5.2 ปัญหาและอุปสรรค

1) Machine Learning Model ที่ใช้ในการประมวลผลยังมีความแม่นยำไม่สูงมาก
2) Machine Learning Model ที่ใช้ในการประมวลผลยังไม่สามารถวิเคราะห์ภาพในตอนกลางคืนได้
3) Machine Learning Model ที่ใช้ในการประมวลผลยังไม่สามารถตรวจจับโดรนที่อยู่ในระยะไกล ๆ ได้
4) ระบบประมวลผลยังไม่สามารถทำงานได้แบบ Real-Time ยังมีการหน่วงเวลาอยู่ที่ประมาณ 6 วินาที

5.3 แนวทางการพัฒนาต่อ

5.3.1 ส่วนประมวลผลบน Raspberry Pi 3

1) เพิ่มประสิทธิภาพของ Machine Learning Model ที่ใช้ในประมวลผลตรวจจับโดรน
2) ปรับปรุง Machine Learning Model ที่ใช้ในการประมวลผลให้สามารถวิเคราะห์โดรนรุ่นอื่น ๆ ได้ด้วย

3) ปรับปรุงระยะเวลาที่ใช้ในการประมวลผลให้น้อยลง เพื่อให้ใกล้เคียง Real-Time มากที่สุด

5.3.2 ส่วนติดต่อกับผู้ใช้

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

บรรณานุกรม

Raspberry Pi Foundation. 2014. **WHAT IS A RASPBERRY PI?.** [Online]. Available: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

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

ThaiEasyElec. 2014. **บทความการพัฒนาโปรแกรมบน Raspberry Pi ด้วย Qt.** [Online]. Available: https://thaieasyelec.com/article-wiki/embedded-electronics-application/บทความการพัฒนาโปรแกรมบน-raspberry-pi-ด้วย-qt.html

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

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

R Studio. 2015. **TensorFlow Mechanics 101.** [Online]. Available: https://tensorflow.rstudio.com/tensorflow/articles/tutorial_tensorflow_mechanics.html

TensorFlow. 2018. **How to Retrain an Image Classifier for New Categories.** [Online]. Available: https://www.tensorflow.org/hub/tutorials/image_retraining

Bibhu Pala. 2018. **Saving, Freezing, Optimizing for inference, Restoring of tensorflow models.** [Online]. Available: https://medium.com/@prasadpal107/saving-freezing-optimizing-for-inference-restoring-of-tensorflow-models-b4146deb21b5

Pkulzc. 2018. **Tensorflow detection model zoo.** [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

KHARI JOHNSON. 2017. **Google launches TensorBoard API to enhance machine learning visualizations**. [Online]. Available: https://venturebeat.com/2017/09/11/google-launches-tensorboard-api-to-enhance-machine-learning-visualizations/

moeiscool. 2018. **Shinobi CE**. [Online]. Available: https://github.com/moeiscool/Shinobi/blob/master/README.md

Chatchawan Niyomthum. 2018. **Neural Network 101 : CNN with TensorFlow**. [Online]. Available: https://medium.com/@thebear19/neural-network-101-cnn-with-tensorflow-fd5d515e979b

Athiwat. 2017. **Deep Learning คืออะไร**. [Online]. Available: https://medium.com/@athivvat/deep-leaning-คืออะไร-785e16d01773

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

Coladev. 2017. **สรุปแนวคิด Neural Network แบบไม่มี Math.** [Online]. Available: https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic

Natthawat Phongchit. 2018. **มาลองดูวิธีการคิดของ CNN กัน !!!**. [Online]. Available: https://medium.com/convolab/มาลองดูวิธีการคิดของ-cnn-กัน-e3f5d73eebaa

Rutger Ruizendaal. 2017. **Deep Learning #3: More on CNNs & Handling Overfitting**. [Online]. Available: https://towardsdatascience.com/deep-learning-3-more-on-cnns-handling-overfitting-2bd5d99abe5d

Joyce Xu. 2017. **An Intuitive Guide to Deep Network Architectures**. [Online]. Available: https://towardsdatascience.com/an-intuitive-guide-to-deep-network-architectures-65fdc477db41

Joyce Xu. 2017. **Deep Learning for Object Detection: A Comprehensive Review**. [Online].

Available: https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9

Sanparith Marukatat. 2018. **โลกหมุนไป งานวิจัยก็หมุนตาม.** [Online]. Available: https://medium.com/@sanparithmarukatat/โลกหมุนไป-งานวิจัยก็หมุนตาม-46ae76d4e195

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

Jonathan Hui. 2018. **What do we learn from region based object detectors (Faster R-CNN, R-FCN, FPN)?**. [Online]. Available: https://medium.com/@jonathan_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9