

การใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันชนิด YOLOv8 ร่วมกับ หุ่นยนต์แบบคาร์ทีเซียนสำหรับการตรวจหาโรคและรดน้ำอัตโนมัติ

ดำรงศักดิ์ กิจเดช* และ วีระพันธ์ ด้วงทองสุข

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเอเชียอาคเนย์

19/1 ถนนเพชรเกษม แขวงหนองค้างพลู เขตหนองแขม กรุงเทพฯ 10160

รับบทความ 6 มกราคม 2568 แก้ไขบทความ 2 กรกฎาคม 2568 ตอรับบทความ 4 กรกฎาคม 2568

บทคัดย่อ

ปัจจุบันสมาร์ตฟาร์มได้มีอิทธิพลกับงานเกษตรกรรมมากขึ้น เนื่องจากช่วยลดต้นทุนและเพิ่มประสิทธิภาพ อย่างไรก็ตามจากการศึกษาวิจัยพบว่าโรคที่เกิดขึ้นในพืชยังคงจำเป็นต้องใช้การสังเกตโดยคน งานวิจัยนี้จึงนำเสนอโครงข่ายประสาทเทียมแบบคอนโวลูชันชนิด YOLOv8 ร่วมกับหุ่นยนต์แบบคาร์ทีเซียนสำหรับการรดน้ำและการตรวจหาโรคอัตโนมัติ ซึ่งงานวิจัยนี้ใช้แคคตัสเป็นตัวอย่างในการทดลอง โดยหุ่นยนต์จะเคลื่อนที่ไปตำแหน่งต่างๆที่กำหนดเพื่อรดน้ำและถ่ายภาพเพื่อส่งให้ปัญญาประดิษฐ์ตรวจหาโรคเมื่อความชื้นในดินต่ำกว่าที่กำหนด หากพบโรค หุ่นยนต์จะทำการส่งภาพโรคที่เกิดขึ้นทางแอปพลิเคชันพร้อมเคลื่อนที่หัวฉีดไปยังตำแหน่งดังกล่าวเพื่อฉีดยากำจัดโรค ขั้นตอนการวิจัยเริ่มด้วยหุ่นยนต์เคลื่อนที่ไปยังตำแหน่งต่างๆพร้อมถ่ายภาพแคคตัส ต่อจากนั้นปัญญาประดิษฐ์จะทำการตรวจหาโรคพร้อมจับเวลาที่ใช้ จากนั้นนำภาพก่อนและหลังการตรวจหาโรคมาตรวจดูความถูกต้อง ผลที่ได้คือใช้เวลาในการประมวลผลแต่ละภาพโดยเฉลี่ย 0.57 วินาที ค่าความถูกต้องอยู่ที่ 90 เปอร์เซ็นต์

คำสำคัญ : โครงข่ายประสาทเทียม; คอนโวลูชัน; แคคตัส; คาร์ทีเซียน; ปัญญาประดิษฐ์

* ผู้นิพนธ์ประสานงาน โทร: +66 2807 4500-27 ต่อ 404, ไปรษณีย์อิเล็กทรอนิกส์: dumrongsakk@sau.ac.th

Using the Convolution Neural Network (YOLOv8) with Cartesian Robot for Automatic Disease Detection and Watering

Dumrongsak Kijdech* and Weerapun Duangthongsuk

Faculty of Engineering, Southeast Asia University

19/1 Phetkasem Road, Nongkangplu, Nongkham, Bangkok, 10160

Received 6 January 2025; Revised 2 July 2025; Accepted 4 July 2025

Abstract

Currently, smart farm has influence to more agriculture. Due to production costs have decreased and productivity has increased. However, the study found that diseases occurring in plants still require human observation. A convolutional neural network (YOLOv8) with a cartesian robot for automatic disease detection and watering is proposed in this research. Which species of cactus are used as experimental model in this research. When the soil moisture falls below the target level, the robot is moved to the target location to watering the plants and collect images for disease detection using artificial intelligence. In case a disease is detected, the robot displays an alert on the LINE application and moves the injector to target position to spray disease eradication. During the research process, the robot moves to each target location and collects images of cactuses. After that, artificial intelligence is used for disease detection and measuring the usage time. Subsequently, the disease detection results from the before and after images are validated. The results indicated that the average prediction time was approximately 0.57 seconds and the accuracy was 90 percentage.

Keywords : AI; Cactus; Cartesian Robot; Convolution; Neural Networks

* *Corresponding Author. Tel.: +66 2807 4500-27 ext. 404, E-mail Address: dumrongsakk@sau.ac.th*

1. บทนำ

แคคตัสเป็นไม้ดอกไม้ประดับซึ่งเป็นที่นิยมมากในปัจจุบัน จึงทำให้มีราคาสูง การดูแลทำได้ค่อนข้างยากในกรณีที่แคคตัสต้นปริแตก เป็นแผล เนื่องจากแรงดันภายในต้น ที่เกิดจากการได้รับปุ๋ยและน้ำมากเกินไป หรือถูกแมงกินบางส่วน ในขณะที่ต้นยังอ่อนแออยู่ จนเกิดแผล หรือเป็นต้นที่ทำการกราฟมา ไม่ว่าจะมาเป็นแผลจากการฉีกได้ก็ตาม หากใช้หัวสปริงเกอร์ในการรดจะเกิดการฟุ้งกระจายของน้ำจนอาจโดนแผล เป็นเหตุให้แคคตัสเน่าได้ และยังมีประสิทธิภาพที่ต่ำ การรดน้ำควรรดบริเวณของโคนแคคตัสที่ละต้นทำให้ใช้เวลาในการรดน้ำต้นไม้ค่อนข้างใช้เวลามากยิ่งในกรณีในเชิงพาณิชย์ยิ่งใช้เวลามากขึ้น เกิดโรคได้ง่าย เช่น เพลี้ยแป้ง ทำให้แคคตัสตายอย่างรวดเร็ว งานวิจัยนี้จึงใช้แคคตัสเป็นตัวอย่างในการทดลอง โดยจากเหตุผลข้างต้นการใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันชนิด YOLOv8 ร่วมกับหุ่นยนต์แบบคาร์ทีเซียนสำหรับการตรวจหาโรคและรดน้ำอัตโนมัติ โดยงานวิจัยนี้ทำการศึกษาการใช้ปัญญาประดิษฐ์ในการตรวจหาเพลี้ยเท่านั้น ในงานวิจัยนี้แบ่งการทบทวนวรรณกรรมเป็น 2 ส่วน ส่วนแรกคือ การหาปัญญาประดิษฐ์ที่เหมาะสมกับการนำมาใช้ในงานวิจัยนี้ ส่วนที่สองคือการหาวิธีการรดน้ำและการใช้อินเทอร์เน็ตของสรรพสิ่ง (IoT) ที่เหมาะสมกับงานวิจัยนี้

ส่วนแรก ปัญญาประดิษฐ์มีหลายชนิดที่ใช้กันอย่างแพร่หลาย จึงจำเป็นต้องศึกษาว่าปัญญาประดิษฐ์ชนิดไหนเหมาะสมที่สุดโดยมีผลดังนี้ การเปรียบเทียบความเร็วในการฝึกข้อมูลและความเร็วในการตรวจจับวัตถุ โดยที่ A. Kuznetsova et al. [1] เป็นการเปรียบเทียบระหว่าง YOLO V3 กับ YOLO V5 ซึ่งผลที่ได้คือ YOLO V5 มีความเร็วที่มากกว่าในส่วนของการฝึกข้อมูล ส่วนของความเร็วในการตรวจจับวัตถุและระบุตำแหน่งทำให้เร็วกว่าเมื่อใช้โครงสร้างของโครงข่ายประสาทเทียมขนาดเล็กสุด ในขณะที่ใช้โครงข่ายประสาทเทียมขนาดใหญ่จะได้ความเร็วในการระบุ

ตำแหน่งวัตถุที่เท่ากัน และ B. Benjdira et al. [2] ทำการเปรียบเทียบ Fast R-CNN กับ YOLO V3 ซึ่ง YOLO V3 มีประสิทธิภาพที่มากกว่า ที่ผ่านมามีการใช้ YOLOv3 YOLOv4 และ YOLOv5 ในงานต่างๆ มากมายดังแสดงตัวอย่างใน [3]-[5] ตามลำดับ โดยที่ Y. Tian et al. [3] ได้ YOLOv3 ในการตรวจหาผลแอปเปิ้ลที่มีระดับความสุกที่เหมาะสมเพื่อให้หุ่นยนต์ทำการเก็บ Z. Jiang et al. [4] ทำการลดขนาดแบบจำลองของ YOLOv4 เพื่อให้สามารถทำงานแบบเรียลไทม์ได้ และ G. Yang et al. [5] ใช้ YOLOv5 ในการตรวจหาใบหน้าคน อย่างไรก็ตามได้มีบางงานวิจัยเช่นใน J. Du [6] ได้ทำการเปรียบเทียบทั้งฝั่งของ R-CNN, และฝั่งของ YOLO ผลที่ได้จากงานวิจัยนี้ YOLOv2 ซึ่งเป็นเวอร์ชันใหม่สุดในเวลานั้นสามารถตรวจจับวัตถุได้เร็วกว่า ยิ่งไปกว่านั้นได้มีการปรับปรุงปัญญาประดิษฐ์แบบต่างๆ ให้สามารถตรวจจับวัตถุในงานเฉพาะทางได้ดีขึ้นดังใน [7], [8] มีการใช้งาน YOLOv6 ใน Model ขนาดต่างๆในงานอุตสาหกรรม และใน C.-Y. Wang et al. [9] ได้นำเสนอการเปลี่ยนแปลงสถาปัตยกรรมบางส่วนของ YOLOv7 ซึ่งช่วยเพิ่มความแม่นยำและความเร็วได้มากกว่า YOLO รุ่นเก่า, ในขณะที่ J. Terven et al. [10] มีการใช้ YOLOv8 ซึ่งเป็น YOLO รุ่นใหม่สุดออกมาในปี ค.ศ. 2023 มีพื้นฐานมาจาก YOLOv5 โดยที่ YOLOv8 มี Model หลากหลายชนิดและหนึ่งในนั้นคือให้ mask ออกมาเหมือน Mask R-CNN จึงทำให้ปัจจุบันมีปัญหาประดิษฐ์ชนิดใหม่เพิ่มขึ้นตลอดเวลา หากมองหางานวิจัยที่นำปัญญาประดิษฐ์มาประยุกต์ใช้ในด้านเกษตรคือ [11]-[13] โดยที่ D. Kijdech [11] ใช้ Convolution Neural Network ในการคัดแยกชนิดของวัชพืช โดยสามารถทำการคัดแยกได้อย่างมีประสิทธิภาพ ขณะที่ D. Kijdech et al. [12] ใช้การควบคุมหุ่นยนต์แบบคาร์ทีเซียนเพื่อรดน้ำแคคตัสโดยกำหนดจุดรดน้ำโดยอัตโนมัติ และ K. P. Ferentinos

[13] ใช้ปัญญาประดิษฐ์ทั้งหมด 5 ชนิดมาเปรียบเทียบกันในการตรวจหาโรคที่เกิดขึ้นที่ใบพืช 25 ชนิดพืชและ 58 กลุ่มโรค ซึ่งได้ผลความแม่นยำที่ 37 เปอร์เซ็นต์ในสภาพแวดล้อมจริง เนื่องจากไม่ได้นำภาพจากสภาพแวดล้อมจริงเข้าฝึกด้วย อย่างไรก็ตามงานวิจัยเหล่านี้เป็น YOLO เวอร์ชันเก่าที่มีความช้ากว่าและไม่สามารถติดตามพื้นที่ที่สนใจได้ทำได้แค่ติกรอบสี่เหลี่ยมและตัวงานวิจัยเป็นแค่การจำแนกชนิดของแคคตัสมีได้นำมาตรวจหาโรค จากที่กล่าวมาทั้งหมดข้างต้นพบว่า ปัญญาประดิษฐ์ทางฝั่งของ YOLO มีความเร็วมากกว่า โดยที่ได้ผลลัพธ์เหมือนกัน หากมีการทำ YOLO เวอร์ชันใหม่กว่าออกมาจะมีความเร็วมากกว่าเสมอและมีโมเดลที่หลากหลายมากขึ้น งานวิจัยนี้จึงเลือกใช้ YOLOv8 ซึ่งเป็นเวอร์ชันล่าสุดขณะนั้นมาใช้งาน

ส่วนที่สอง ได้ทำการศึกษาหาชนิดของหุ่นยนต์และชนิดของเซ็นเซอร์ที่เหมาะสมกับงานวิจัยนี้ ในส่วนของ การรดน้ำต้นไม้ [14], [15] ได้ทำระบบรดน้ำต้นไม้เช่นกัน แต่เป็นการรับค่าความชื้นในดิน อุณหภูมิอากาศ และทำการสั่งหัวรดน้ำที่เป็นแบบไม่สามารถควบคุมไม่ให้ยอดต้นไม้โดนน้ำได้จึงเหมาะสมสำหรับต้นไม้ทั่วไป แต่ไม่ใช่สำหรับแคคตัส แต่มีการส่งข้อมูลขึ้นออนไลน์เพื่อให้เจ้าของดูแบบออนไลน์ได้ ในขณะที่ Y. Huang et al. [16] ใช้เซ็นเซอร์วัดความชื้นในดิน อุณหภูมิ และความชื้นในอากาศเพื่อทำการรดน้ำอย่างเหมาะสม ส่วน N. Hema et al. [17] ใช้ Mobile Robot ช่วยในการรดน้ำโดยมี RFID Tag สำหรับระบุตัวของต้นไม้และกระถาง ซึ่งมีเซ็นเซอร์ต่างๆติดอยู่ที่แต่ละกระถางและส่งสัญญาณแบบไร้สายไปที่ตัว Mobile Robot โดยเส้นทางการเคลื่อนที่จะได้จากการติดตามเส้นสีดำที่พื้น ในขณะที่ P. Tangtisanon [18] ใช้สปริงเกอร์ในการรดน้ำต้นไม้โดยพิจารณาความชื้นในดินและใช้หุ่นยนต์เคลื่อนที่ในการกำจัดวัชพืชที่อยู่ในบริเวณการควบคุมจะ ใช้การควบคุมแบบ Fuzzy Logic จากงานวิจัยต่างๆ

ข้างต้นจะพบว่ามีการใช้ IoT หรือ Internet of thing ร่วมกับหุ่นยนต์ในรูปแบบต่างๆ เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดและเหมาะสมกับงานนั้นๆ โดยงานวิจัยนี้เลือกใช้หุ่นยนต์แบบคาร์ทีเซียนแนวตั้งเนื่องจากปลูกภายในอาคารช่วยประหยัดพื้นที่และต้นทุนในการทำต่ำกว่าการใช้ Mobile Robot มาก

2. ระเบียบวิธีวิจัย

ตัวแปรที่จะศึกษาคือความถูกต้องแม่นยำในการตรวจโรค ความสามารถในการรักษาโรค ปริมาณน้ำที่ใช้ในการรดน้ำแคคตัสที่ค่าความชื้นในดินเดียวกัน ในส่วนของค่าความแม่นยำในการตรวจโรคจะทำการเปรียบเทียบระหว่างจำนวนต้นที่เป็นโรคทั้งหมดเทียบกับจำนวนต้นที่พบโรค ส่วนของความสามารถในการรักษาโรคจะทำการเปรียบเทียบระหว่างจำนวนต้นที่ตรวจพบโรคกับจำนวนต้นที่ได้รับการฉีดยากำจัดโรค โดยระบบอัตโนมัติแล้วหาย ส่วนของปริมาณน้ำที่ใช้ในการรดแคคตัสที่ค่าความชื้นในดินเท่ากันเป็นการเปรียบเทียบระหว่างการรดแบบปกติที่ใช้หัวสปริงเกอร์แบบละเอียด (สปริงเกอร์ 1 ตัว ต่อแคคตัส 4 ต้น) เทียบกับการรดแบบใช้หุ่นยนต์รดน้ำของงานวิจัยนี้

ระเบียบวิธีวิจัยถูกแบ่งเป็น 4 คือ อุปกรณ์การทดลอง การพัฒนาระบบควบคุม การเตรียมข้อมูลและการฝึกข้อมูล และขั้นตอนนมนการทดลอง โดยมีรายละเอียดดังต่อไปนี้

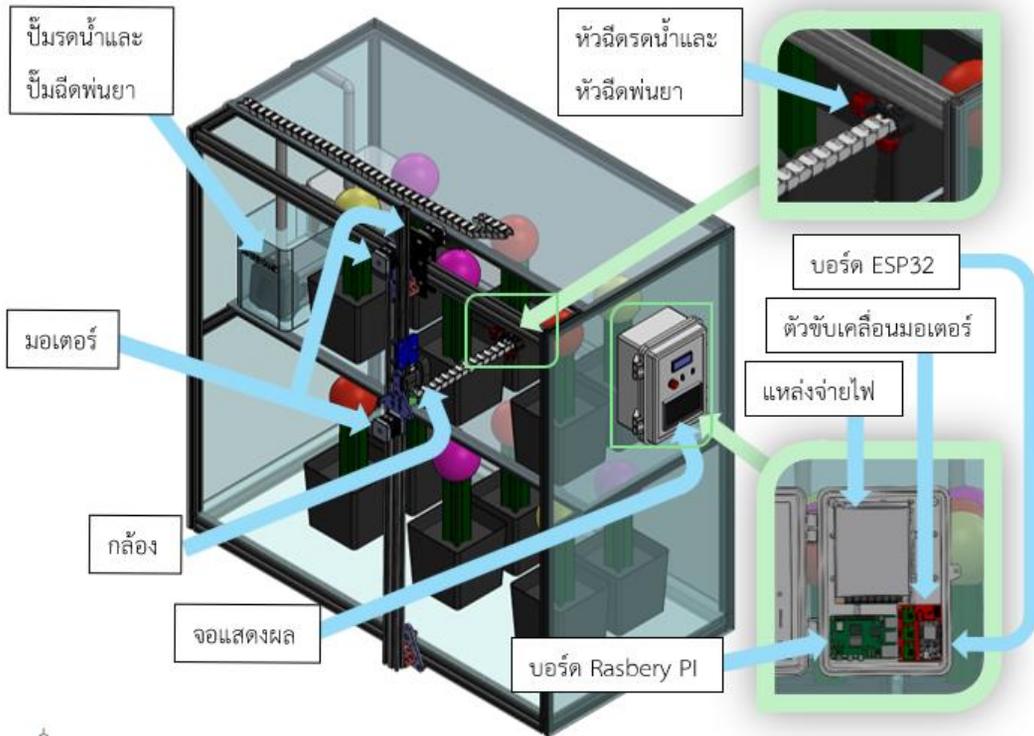
2.1 อุปกรณ์การทดลอง

อุปกรณ์ที่ใช้ในการทดลองมีส่วนประกอบหลักคือ กล้อง Webcam Logitech C525 ความละเอียด 720p อัตราภาพ 30 fps ใช้ในการเก็บรวบรวมภาพสำหรับฝึกปัญญาประดิษฐ์ชนิด YOLOv8 และสำหรับตรวจจับโรคแบบเรียลไทม์ ส่วนต่อมาเป็นบอร์ด Rasberry Pi 5 ทำหน้าที่เป็นคอมพิวเตอร์ขนาดเล็ก ส่วนถัดมาคือ จอแสดงผลแบบจอสัมผัส WAVESHARE

ขนาด 7 นิ้ว ใช้ในการแสดงผลภาพแคคตัสที่ตรวจพบโรค (เพลี้ย) ในระบบรดน้ำและพ่นยาจะมีหัวฉีดที่ฉีดเป็นฝอยละออง 2 หัวและมีปั้มน้ำแยกกัน

ส่วนชุดหุ่นยนต์ขับเคลื่อนคาร์ทีเซียนขนาด 50 ซม. x 100 ซม. x 100 ซม. ภายในมีสเต็ปมอเตอร์ Nema 17 3 ตัว ใช้ในการขับเคลื่อนหุ่นยนต์ใน 3 แกน

เซ็นเซอร์วัดความชื้นในดินรุ่น YL-69 และใช้บอร์ด ESP32 ในการควบคุมหุ่นยนต์และรับส่งข้อมูล ส่วนของตัวขับเคลื่อนมอเตอร์เป็น A4988 Driver ร่วมกับ CNC Shields V3 ส่วนแหล่งจ่ายไฟใช้เป็น Power supply 12V 30A โดยที่ภาพรวมโครงสร้างของหุ่นยนต์ถูกแสดงในรูปที่ 1



รูปที่ 1 ส่วนประกอบหลักของหุ่นยนต์ที่ใช้ในการทดลอง

2.2 การพัฒนาระบบควบคุม

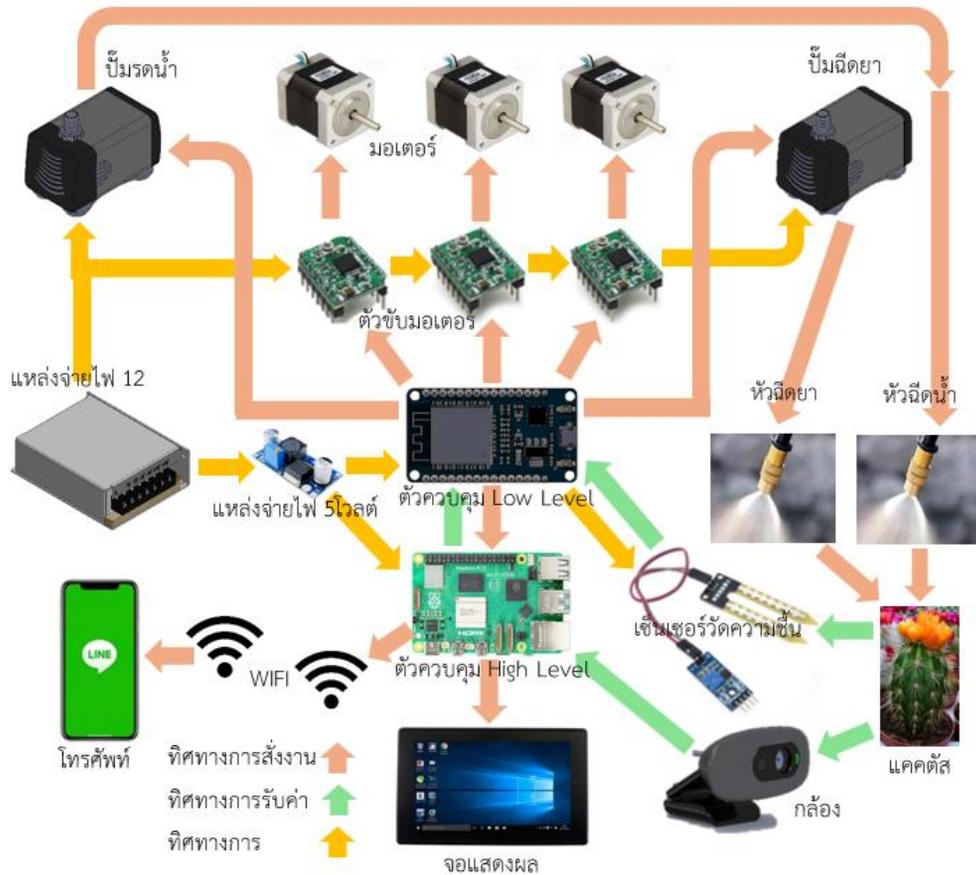
ภาพรวมของระบบรดน้ำอัตโนมัติและตรวจโรคในแคคตัสแสดงในรูปที่ 2 ระบบการทำงานเริ่มจากเซ็นเซอร์วัดความชื้นในดินถูกปิดไว้แล้วละ 1 ตัว เมื่อค่าความชื้นในดินต่ำกว่า 40% หุ่นยนต์จะทำการเคลื่อนไปยังตำแหน่งที่กำหนดซึ่งถูกแสดงดังรูปที่ 3 โดยหุ่นยนต์จะเคลื่อนจากทางด้านล่างซ้ายไปขวาจากนั้นเคลื่อนที่ขึ้นไปยังชั้นที่ 2 และเคลื่อนที่จากขวาไปซ้าย ตามลำดับตัวเลขที่แสดงในรูปที่ 3 แต่ละตำแหน่งตัวหุ่นยนต์จะเคลื่อนที่หัวฉีดรดน้ำเข้าไปด้านในเพื่อรดน้ำต้นแคคตัส

ด้านในด้วย โดยที่น้ำถูกเก็บอยู่ในถังและมีปั้มน้ำใช้ในการรดน้ำขึ้นมา เมื่อถึงจุดที่กำหนดบอร์ด ESP32 (Low Level) จะทำการสั่งให้ปั้มน้ำทำงานผ่านรีเลย์ในปริมาณน้ำที่กำหนด

ในส่วนของการตรวจจับเพลี้ยในแคคตัส เริ่มจากกล้องจะทำการเก็บภาพใน 6 จุด ที่กำหนด ซึ่งการเก็บภาพจะเกิดขึ้นเมื่อความชื้นในดินต่ำกว่าที่กำหนด ก่อนการรดน้ำของแต่ละแถว จะทำการการถ่ายภาพ 1 ครั้งโดยสามารถมองเห็นแคคตัสได้ทั้งแถว และส่งไปยังบอร์ด Rasbery Pi5 ปัญญาประดิษฐ์ชนิด YOLOv8 ภายใน

บอร์ดจะทำการตรวจหาโรคผ่านภาพที่ส่งมา หากพบเชื้อในแคคตัสระบบภายในบอร์ด Rasbery Pi5 (High Level) จะทำการส่งค่าตำแหน่งที่เกิดโรคไปยังบอร์ด ESP32 เพื่อควบคุมหุ่นยนต์ส่งงานปั๊มเพื่อฉีดน้ำยากำจัดเชื้อหลังจาก

รดน้ำแล้ว จากนั้นส่งรูปต้นแคคตัสที่เกิดโรคไปยังแอปพลิเคชันเพื่อแสดงผลและแจ้งเตือน ในส่วนของจอแสดงผลที่เชื่อมต่อกับบอร์ด Rasbery Pi5 จะทำการแสดงผลการตรวจหาโรคแบบเรียลไทม์

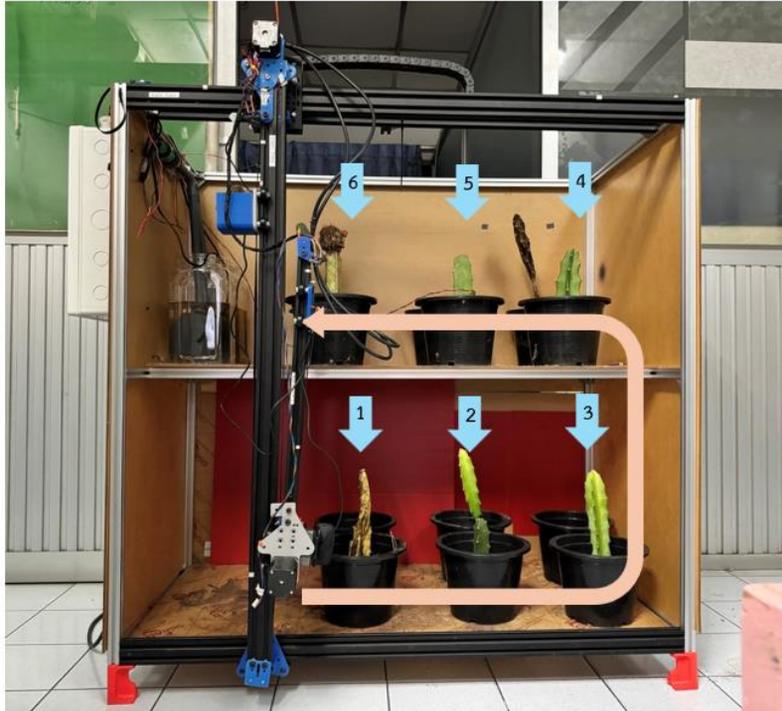


รูปที่ 2 ภาพรวมของระบบรดน้ำอัตโนมัติและตรวจโรคในแคคตัส

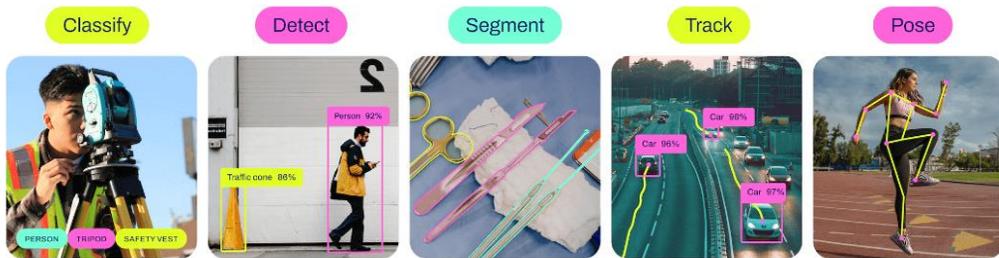
2.3 การเตรียมข้อมูล (Dataset) และการฝึกข้อมูล (Data Training)

งานวิจัยนี้ใช้ปัญญาประดิษฐ์ชนิด YOLOv8 (You Only Look Once Version 8) ในการตรวจหาเชื้อในแคคตัส โดยที่ YOLOv8 มีพื้นฐานมาจากเทคนิคโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Networks) โดย Ultralytics

[20] แสดงให้เห็นว่า YOLOv8 ถูกแบ่งออกเป็น 5 แบบหลัก ได้แก่ (1) การจำแนกประเภทวัตถุ (Classify), (2) การตรวจจับตำแหน่งของวัตถุ (Detect), (3) การแยกขอบเขตของวัตถุ (Segment), (4) การติดตามการเคลื่อนไหวของวัตถุ (Track) และ (5) การระบุท่าทางของวัตถุ (Pose) ดังแสดงในรูปที่ 4 โดยงานวิจัยนี้เลือกการทำ Segment ซึ่งมีความเหมาะสมกับงานมากที่สุด



รูปที่ 3 ตำแหน่งที่กำหนดและทิศทางทางการเคลื่อนที่ของหุ่นยนต์



รูปที่ 4 รูปแบบในการทำปัญหาประดิษฐ์ของ YOLOv8 (Ultralytics [20])

ปัจจุบันเป็นเวอร์ชันล่าสุดที่นิยมมากในการนำมาใช้งานร่วมกับกล้อง เนื่องจากโครงข่ายประสาทเทียมชนิดนี้มีตัวเลือกการใช้งานที่หลากหลาย ใช้งานง่ายและใช้เวลาในการฝึกข้อมูลน้อยเมื่อเทียบกับโครงข่ายประสาทเทียมชนิดอื่น โครงข่ายประสาทเทียม (Neural Networks) คือการจำลองโครงข่ายประสาทในสมองของมนุษย์ซึ่งช่วยให้สามารถตัดสินใจงานที่มีความซับซ้อนได้ เช่น พื้นหลัง แสงและเงา รวมไปถึงรูปร่างของ

ของวัตถุที่เปลี่ยนไปมา YOLOv8 มีความเร็วในการประมวลผลสูง ส่วนของความแม่นยำขึ้นอยู่กับจำนวนของภาพและความหลากหลายของภาพที่นำมาฝึกปัญหาประดิษฐ์

การฝึกปัญหาประดิษฐ์ ภาพที่ใช้ในการฝึกปัญหาประดิษฐ์จะใช้ภาพที่ได้จากกล้อง Webcam ถูกลดความละเอียดลงเหลือ 640 x 640 pixel โดยภาพที่ใช้ในการฝึกจะมีพื้นหลังที่หลากหลายและมีรูปร่างของ

โรคที่แตกต่างกันออกไป ภาพที่ใช้ในการฝึกข้อมูล (Dataset) อยู่ที่ 96 ภาพ ซึ่งภาพที่ใช้ในการฝึกทั้งหมดมาจากการถ่ายภาพแคดคัสที่ใช้ในการทดลองจริงและโหลดมาจากเว็บไซต์ออนไลน์ การทำ Label ให้กับ Dataset ทั้งหมดถูกทำบนเว็บไซต์ Roboflow ซึ่งภาพที่ถูกทำ Label แล้ว ถูกแสดงดังรูปที่ 5 และได้ทำการใช้กระบวนการทางภาพ ช่วยทำภาพเพิ่มเติม (Augmentations) ทำให้มีภาพที่ใช้ในการฝึกข้อมูลทั้งหมดเพิ่มจาก 96 ภาพ เป็น 240 ภาพ ภาพที่เพิ่มเข้ามาได้จากการกลับภาพแนวนอน (Flip), การหมุนภาพ 90 องศาในทิศทางทวนเข็มนาฬิกาและตามเข็มนาฬิกา (Rotate), การตัดภาพบางส่วน ชุม 26% (Crop), การทำภาพเอียง ± 14 องศา (Shear) และการทำภาพเบลอ (Blur) เป็นต้น โดยภาพทั้งหมดเป็นภาพที่ใช้ในการฝึกปัญญาประดิษฐ์ (Training) 216 ภาพ ภาพที่ใช้ในการประเมินผลการฝึก (Valid) 13 ภาพ และภาพที่ใช้ในการทดสอบ 11 ภาพ ค่าตัวแปรต่าง ๆ ที่ใช้ในการฝึกปัญญาประดิษฐ์มีดังนี้ จำนวนชั้นในการฝึก (Layer) ทั้งหมด 261 เลเยอร์, มีพารามิเตอร์ทั้งหมด 11,790,870 พารามิเตอร์, Optimizer เป็น Adam, อัตราการเรียนรู้ (Learning Rate) อยู่ที่ 0.001667, Momentum อยู่ที่ 0.9, Batch size ใช้เป็น 60% ของ CUDA Memory จำนวนรอบในการฝึก (Epoch) อยู่ที่ 100 รอบ

โครงสร้างภายในของแบบจำลอง YOLOv8 เป็นการผสมกันหลายชั้นของ Convolution Layer, BN (Batch Normalization) Layer และ Leaky ReLU (Rectified Linear Unit) เพื่อช่วยให้โครงข่ายประสาทเทียมสามารถตรวจจับและตีเส้นขอบได้ดียิ่งขึ้น โดยที่ Convolution Layer คือการนำภาพมาใส่ตัวกรอง (Filter) แบบต่างๆ ส่วน Batch Normalization จะช่วยให้แต่ละ Layer ในโครงข่ายประสาทเทียมสามารถเรียนรู้ได้ด้วยตัวเองอย่างอิสระและลดการผูกติดกับ Layer อื่นๆ ส่วน Leaky ReLU คือการเปลี่ยนค่า

ต่างๆให้เป็นค่าบวกทั้งหมดเนื่องจากการกรองภาพโดยใช้ Convolution Layer จะทำให้ค่าที่ได้บางค่ามีค่าติดลบทำให้ไม่สามารถแสดงภาพได้ ตัวอย่างโครงสร้างภายในช่วงขาเข้า (Input) และขาออก (Output) ของ YOLOv8 ถูกแสดงดังรูปที่ 6

การฝึกปัญญาประดิษฐ์จำเป็นต้องใช้คอมพิวเตอร์ที่มีการ์ดจอที่มีความสามารถในการประมวลผลที่สูงมาก สำหรับการฝึกปัญญาประดิษฐ์งานวิจัยนี้ทำการฝึกบน Google Collab ซึ่งเป็นบริการประมวลผลแบบออนไลน์ โดยใช้การ์ดจอ NVIDIA Tesla T4 ใช้ Python เวอร์ชัน 3.10.12 ในการเขียนโปรแกรมฝึกและใช้ Pytorch เวอร์ชัน 2.3.0 เป็น Backbone ส่วนโปรแกรมที่ใช้ควบคุมการทำงานการ์ดจอเป็น cu121 และใช้ CPU (Intel Xeon 2.20GHz) และเวลาที่ใช้ในการฝึกข้อมูลทั้งหมด 0.284 ชั่วโมง หลังจากทำการฝึกเสร็จจะได้แบบจำลอง (Model) สำหรับการตรวจหาเปลี้ยและระบุตำแหน่งโดยการตีเส้นขอบวัตถุ

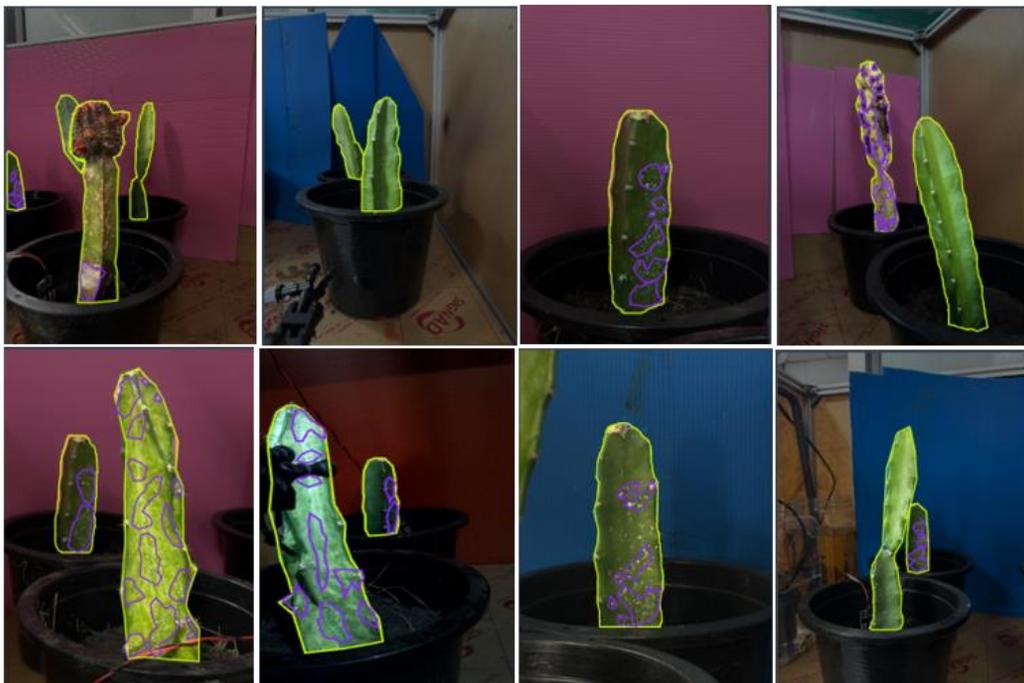
ในส่วนของ อัตราการเรียนรู้ (Learning Rate) ไม่มีค่าที่ตายตัว ขึ้นอยู่กับหลายปัจจัย ซึ่งอัตราการเรียนรู้ควรมีค่าน้อยที่สุดเท่าที่จะทำได้เพื่อป้องกันไม่ให้เกิดการ Overfitting (การที่ไม่สามารถตรวจหาโรคจากรูปที่ไม่ได้ถูกนำมาฝึกได้) แน่นนอนว่าหากมีค่าน้อยเกินไปจะทำให้จำเป็นต้องใช้เวลาในการฝึกข้อมูลมากขึ้น อย่างไรก็ตามมีค่าที่แนะนำคือหากรูปที่ใช้ในการฝึกข้อมูลน้อยกว่า 1,000 รูป ควรใช้ค่าอัตราการเรียนรู้ในช่วง 0.001 - 0.005 โดยอ้างอิงจาก Smith et al. [19] ในส่วนของจำนวนรอบในการฝึก (Epoch) ไม่มีค่าตายตัวเช่นเดียวกับอัตราการเรียนรู้ หากใช้มากเกินไปเวลาในการฝึกข้อมูลจะมากขึ้นตามไปด้วย โดยงานวิจัยนี้ใช้ 100 รอบ หากมองที่ผลของการฝึกข้อมูลดังแสดงในรูปที่ 8 พบว่าสามารถฝึกข้อมูลเพิ่มได้อีก อย่างไรก็ตามจากการทดลองพบว่าแบบจำลองที่ได้มีความแม่นยำเพียงพอแล้ว เนื่องจากการทดลองระบบปิด

2.4 ขั้นตอนในการทดลอง

ปกติหุ่นยนต์จะทำงานตาม Flow Chart ในรูปที่ 7 ในกรณีที่ความชื้นต่ำกว่า 40% ระบบจะเริ่มต้นอีกครั้งและพ่นยาซ้ำจนกว่าพืชจะตายหมด

โดยการทดลองถูกแบ่งเป็น 2 ส่วน คือ การเก็บผลปริมาณน้ำที่สูญเสียการรดน้ำแบบปกติ นั่นคือการเปิดสปริงเกอร์รดน้ำจนกว่าค่าความชื้นในดินจะถึงจุดที่กำหนดเปรียบเทียบกับการรดเป็นรายต้นของงานวิจัยนี้ และอีกส่วนคือการนำแบบจำลองที่ได้จาก YOLOv8 มาทำการทดสอบกับภาพที่ไม่ได้ถูกนำมาทดลองเพื่อหาค่าความถูกต้องของปัญญาประดิษฐ์

ส่วนแรก จะทำการเก็บผลโดยการเปิดโซลินอยด์วาล์วจนกว่าค่าความชื้นในดินจะถึงจุดที่กำหนด (โดยในโหมดนี้ระบบทำการรดทุก 1 ชั่วโมง) จึงปิดโซลินอยด์วาล์วและจับเวลา จากนั้นนำหัวสปริงเกอร์มาใส่ในขวดและทำการเปิดและปิดโซลินอยด์วาล์วเท่ากับเวลาที่ได้จากการทดสอบ จากนั้นทำการชั่งน้ำหนักหาปริมาณน้ำที่ใช้ โดยแบ่งเป็นการรดแบบปกติคือสปริงเกอร์ 1 ตัว รดน้ำต้นไม้ 4 ต้น กับการรดน้ำเป็นรายต้นของงานวิจัยนี้ ทำการทดสอบทั้งหมด 3 ครั้ง (หยุดรดน้ำโดยการรับค่าจากเซ็นเซอร์วัดความชื้นระบบตัดโซลินอยด์วาล์วอัตโนมัติ)



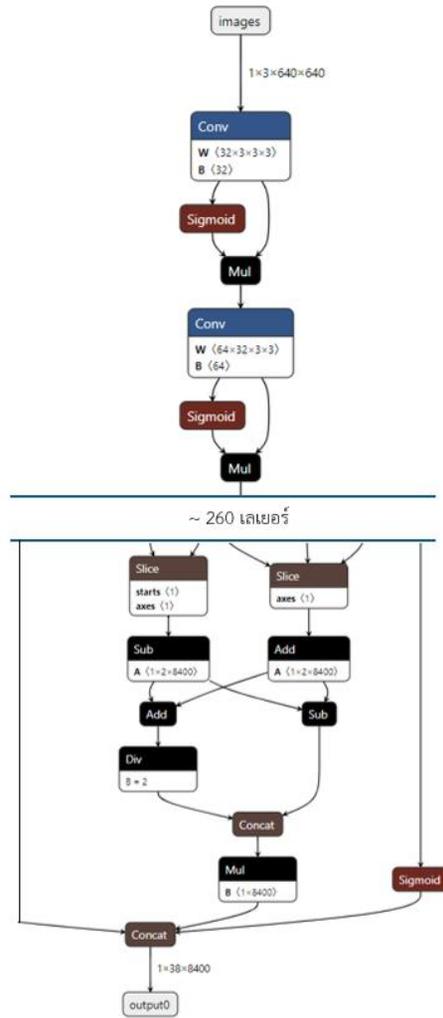
รูปที่ 5 แสดงตัวอย่าง Dataset ที่ถูกใช้ในการทดลองนี้ โดยสีม่วงเป็นตำแหน่งที่เกิดเพรี้ยและสีเหลืองคือต้นแคคตัส

ส่วนที่สอง ทำการเก็บผล 2 แบบ แบบแรก คือการเก็บผลการฝึกข้อมูล โดยผลที่ได้ถูกแสดงเป็นกราฟ แกน Y มีค่าเป็น 0 – 1 เป็นตัวแทนของ 0 – 100% ขณะที่แกน X เป็นจำนวนครั้งในการฝึกข้อมูล ค่าที่แสดงมี 2 ส่วน คือ ค่า Loss จากการฝึกข้อมูลและค่า Loss จากการประเมินผล ดังแสดงในรูปที่ 8 โดยที่ค่ายิ่งน้อยยิ่งมีความแม่นยำมาก

Box loss เป็นการฝึกและประเมินผลเพื่อให้ดีกรอบสี่เหลี่ยมวัตถุที่สนใจให้ถูกต้องและแม่นยำ, Seg (Segmentation) Loss เป็นการฝึกและประเมินผลเพื่อให้ดีกรอบตามรูปร่างวัตถุที่สนใจให้ถูกต้องและแม่นยำ, Cls (Classification) Loss เป็นการฝึกและประเมินผลเพื่อคัดแยกชนิดของวัตถุที่สนใจให้ถูกต้อง, Dfl (Distributed

Focal Loss) Loss เป็น Loss ที่เพิ่มขึ้นใหม่ใน YOLOv8 ถูกออกแบบมาเพื่อปรับปรุงความแม่นยำในการตรวจจับวัตถุโดยมุ่งเน้นไปที่ตัวอย่างที่ตรวจพบได้ยาก ใช้การกำหนดค่าน้ำหนักที่ท้าทายมากขึ้น ทำให้เชื่อถือได้และแม่นยำมากขึ้นในสภาพแวดล้อมที่มีความหลากหลาย ในขณะที่รูปที่ 9 แสดงผลการฝึกข้อมูลในส่วนของค่า Precision คือความแม่นยำเป็นการวัดปริมาณสัดส่วนของผลในทางบวกที่แท้จริง จากการคาดการณ์เชิงบวกทั้งหมด โดยประเมินความสามารถของแบบจำลองเพื่อหลีกเลี่ยงผลบวกหลง ในขณะที่ค่า Recall คือการคำนวณสัดส่วนของผลบวกจริงเทียบกับผลทางบวกจริงทั้งหมด โดยวัดความสามารถของแบบจำลองที่ใช้กับตัวอย่างทั้งหมด,

mAP50 (mean Average Precision) คือความแม่นยำเฉลี่ยที่คำนวณที่มีความทับซ้อนกันระหว่างพื้นที่ (IOU, Intersection Over Union) ที่ 0.50 เป็นการวัดความแม่นยำของแบบจำลองโดยพิจารณาเฉพาะการตรวจจับที่ง่ายเท่านั้น ขณะที่ mAP50-95 คือความแม่นยำเฉลี่ยที่คำนวณที่มีความทับซ้อนกันระหว่างพื้นที่ ตั้งแต่ 0.50 ถึง 0.95 โดยครอบคลุมเกี่ยวกับประสิทธิภาพของแบบจำลองในระดับยาก แบบสอง เป็นการนำภาพที่ไม่ได้อยู่ในการฝึกข้อมูลและการประเมินผลข้อมูล ซึ่งเป็นภาพจริงที่ใช้ในการทดลอง โดยแบ่งเป็นภาพที่มีเปรียบเทียบกับภาพที่ไม่มีเปรียบมาอย่างละ 10 ภาพ จากนั้นทำการทำนายแล้วเทียบกับเฉลยและเวลาที่ใช้เฉลี่ย



รูปที่ 6 ตัวอย่างโครงสร้างภายในของแบบจำลอง YOLOv8

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = \left(\frac{\text{Recall}^{-1} + \text{Precision}^{-1}}{2} \right)^{-1} \tag{3}$$

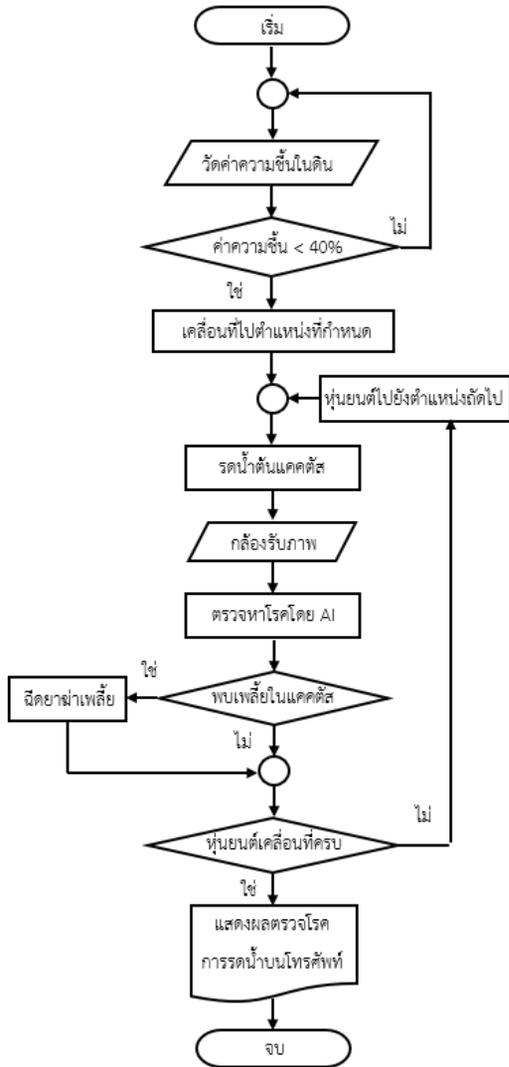
สมการที่ (1) แสดงการคำนวณค่า Precision และ สมการที่ (2) แสดงการคำนวณค่า Recall โดยค่า TP (True Positive) คือกรณีที่แบบจำลองทำนายว่ามี class และมี class จริง ส่วน FP (False Positive) คือ กรณีที่แบบจำลองทำนายว่ามี class แต่จริงๆแล้วไม่มี class (บางครั้งเรียก Type I error) และ FN (False Negative) คือ กรณีที่แบบจำลองทำนายว่าไม่มี class แต่จริงๆ แล้วมี class อยู่ (บางครั้งเรียก Type II error) ขณะที่การคำนวณหาค่า F1 Score โดยใช้สมการที่ (3) เป็นค่าเฉลี่ยของ Precision และ Recall ซึ่งจะได้การประเมินที่สมดุลของประสิทธิภาพของแบบจำลอง

3. ผลการศึกษาและอภิปรายผล

จากรูปที่ 8 แสดงผลการฝึกข้อมูลและประเมินผล ข้อมูล ในส่วนของ Bounding Box, Segmentation, Classification และ Distributed Focal Loss มีค่า Loss ต่ำมากอยู่ในระดับที่น่าพอใจ และในรูปที่ 9 แสดงความแม่นยำในการตอบถูก (Precision) ทั้งในส่วนของ Box และ Mask อยู่ที่ประมาณ 75% ขณะที่ค่า Recall ของ Box และ Mask อยู่ที่ประมาณ 55% และ 45% ตามลำดับ, ค่า mAP50 ของ Box และ Mask อยู่ที่ประมาณ 60% และ 50% ตามลำดับ และ ค่า mAP50-95 ของ Box และ Mask อยู่ที่ประมาณ 30% และ 25% ตามลำดับ

รูปที่ 10 แสดงผลการตรวจหาโรคที่ถูกส่งไปยัง แอปพลิเคชันไลน์ ในกรณีที่ตรวจพบโรค โดยภาพที่ได้จะเป็นภาพจากกล้องที่ติดอยู่ที่ตัวหุ่นยนต์ เมื่อมีสัญญาณจากบอร์ดควบคุมที่เครื่องแจ้งมา ระบบปัญญาประดิษฐ์ที่คอมพิวเตอร์จะทำการตรวจจับโรคและหากพบจะทำการส่งไปยังไลน์

ตารางที่ 1 แสดงปริมาณน้ำที่ใช้ในการรดน้ำแบบ ปกติเปรียบเทียบกับแบบรดน้ำรายต้นของงานวิจัยนี้โดย



รูปที่ 7 แสดงการทำงานของระบบบรรดน้ำอัตโนมัติ และตรวจโรคในแคคตัส

ทำการทดลอง 3 ครั้ง ค่าเฉลี่ยปริมาณน้ำที่ใช้ในการรดน้ำแบบปกติอยู่ที่ 267 ลูกบาศก์เซนติเมตร ขณะที่ค่าเฉลี่ยปริมาณน้ำที่ใช้ในการรดน้ำแบบรายต้นคือ 82 ลูกบาศก์เซนติเมตร ประหยัดน้ำไป 185 ลูกบาศก์เซนติเมตร ต่อการรดน้ำ 1 ต้น หรือ 69.29%

ตารางที่ 2 แสดงเวลาที่ใช้ในการการหาตำแหน่งและจำแนกเพลี้ยแป้งออกจากแคคตัสโดยเป็นภาพที่ไม่มีเพลี้ยแป้งอยู่ เวลาที่ใช้เฉลี่ย 437.3 มิลลิวินาที อัตราการหายถูกคือ 70% โดยผลการทดลองใน ตารางที่ 2 และ ตารางที่ 3 ใช้ภาพที่ถ่ายในตำแหน่งจริง แต่นำมาทดลองแยกเพื่อลด Error จากปัจจัยอื่น ๆ

ขณะที่ตารางที่ 3 แสดงเวลาที่ใช้ในการหาตำแหน่งและจำแนกเพลี้ยแป้งออกจากแคคตัสโดยเป็นภาพที่มีเพลี้ยแป้งอยู่ เวลาที่ใช้เฉลี่ย 570.1 มิลลิวินาที อัตราการหายถูกคือ 90%

Precision สำหรับการทดลองจริงคือ 0.75 ส่วนค่า Recall สำหรับการทดลองจริงคือ 0.9 และค่า F1 Score สำหรับการทดลองจริงคือ 0.81

ในส่วนของ Confusion matrix แสดงดังรูปที่ 13 ได้แสดงให้เห็นว่าหากทำการตรวจหา 3 ประเภท คือการรวมพื้นหลังเข้าไปด้วย ทำให้สามารถตรวจหาแคคตัสได้ 78% เพลี้ยแป้ง 50%

ตารางที่ 1 แสดงปริมาณน้ำที่ใช้ในการรดน้ำแบบปกติเปรียบเทียบกับแบบรดน้ำรายต้น

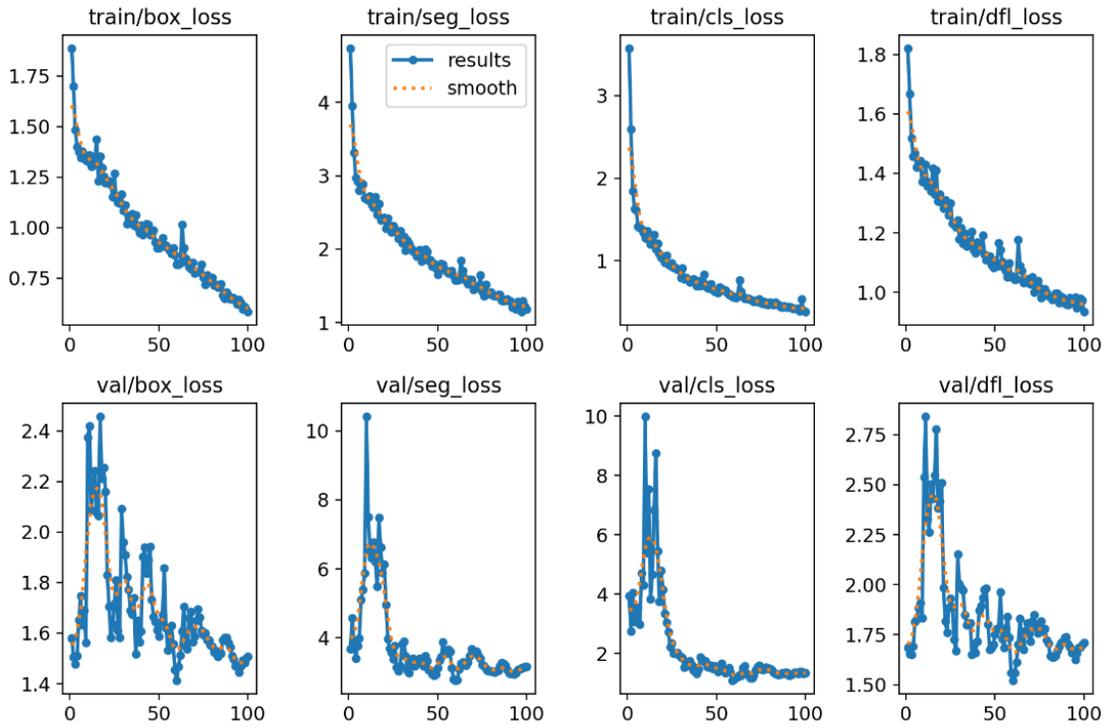
ครั้งที่	ปริมาณน้ำที่ใช้แต่ละต้น	
	แบบปกติ (ลูกบาศก์ เซนติเมตร)	จากการทดลอง (ลูกบาศก์เซนติเมตร)
1	115	80
2	114	83
3	116	83
เฉลี่ย	267	82

ตารางที่ 2 แสดงเวลาที่ใช้ในการหาตำแหน่งและจำแนกเพลี้ยแป้งออกจากแคคตัสโดยเป็นภาพที่ไม่มีเพลี้ยแป้งอยู่

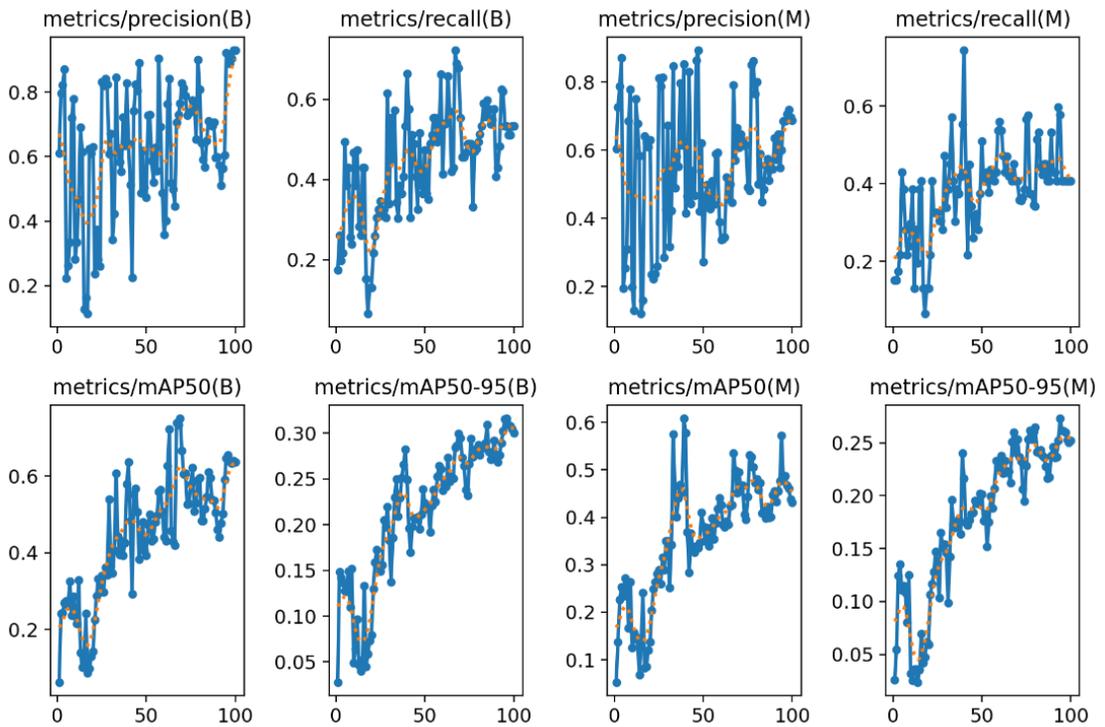
ภาพที่	ภาพที่ไม่มีเพลี้ยแป้ง	เวลาที่ใช้ (มิลลิวินาที)
1	ไม่พบ	583.5
2	ไม่พบ	516.1
3	ไม่พบ	461.6
4	ไม่พบ	419.5
5	พบ	660.3
6	ไม่พบ	583.1
7	พบ	365.8
8	พบ	346.7
9	ไม่พบ	225.3
10	ไม่พบ	211.5
เวลาเฉลี่ย		437.3

ตารางที่ 3 แสดงเวลาที่ใช้ในการหาตำแหน่งและจำแนกเพลี้ยแป้งออกจากแคคตัสโดยเป็นภาพที่มีเพลี้ยแป้งอยู่

ภาพที่	ภาพที่มีเพลี้ยแป้ง	เวลาที่ใช้ (มิลลิวินาที)
1	พบ	620.1
2	พบ	723.9
3	พบ	614.9
4	พบ	384.4
5	พบ	696.2
6	ไม่พบ	535.6
7	พบ	489.2
8	พบ	584.6
9	พบ	516.0
10	พบ	536.0
เวลาเฉลี่ย		570.1



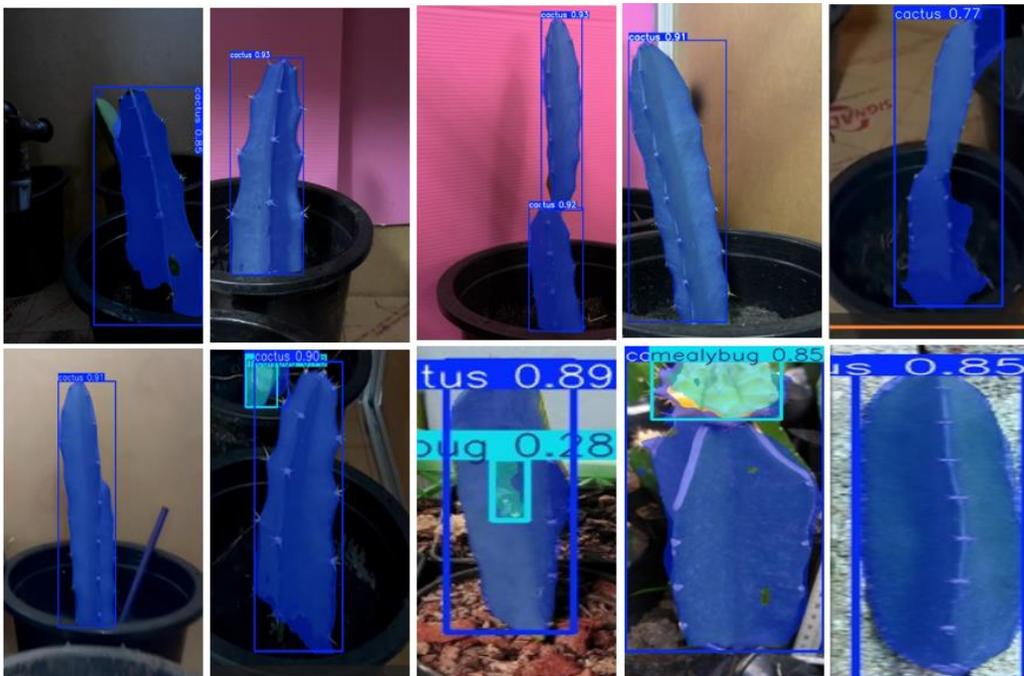
รูปที่ 8 ผลการฝึกข้อมูลและประเมินผลข้อมูล



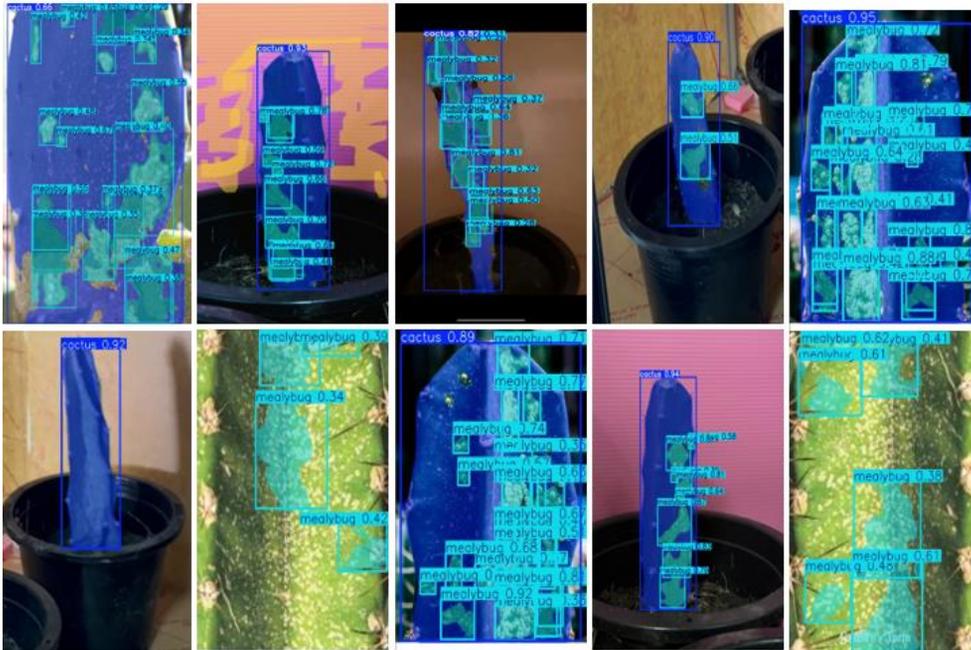
รูปที่ 9 ผลการประเมินผลข้อมูลโดยที่ (B) คือ Box และ (M) คือ Mask



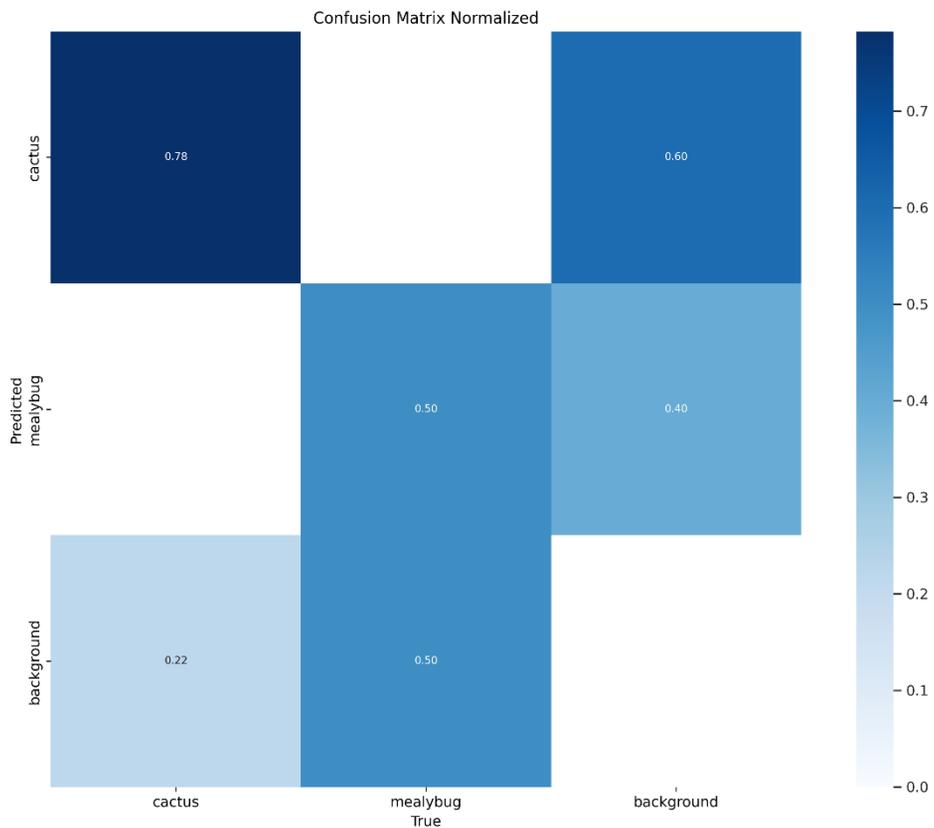
รูปที่ 10 ผลการตรวจหาถูกส่งไปยังแอปพลิเคชันไลน์



รูปที่ 11 แสดงผลการตรวจหาตำแหน่งและจำแนกเพลี้ยแบ่งออกจากแคคตัสโดยเป็นภาพที่ไม่มีเพลี้ยแบ่งอยู่



รูปที่ 12 แสดงผลการตรวจหาตำแหน่งและจำแนกเพลี้ยแป้งออกจากแคคตัสโดยเป็นภาพที่มีเพลี้ยแป้งอยู่



รูปที่ 13 แสดงผล Confusion matrix แบบ แบบ Normalized ของการตรวจหา แคคตัส เพลี้ยแป้ง และพื้นหลัง

4. สรุป

จากผลการทดลองในรูปที่ 8 และ รูปที่ 9 แสดงให้เห็นว่าลักษณะความชันของกราฟที่ยังคงลดลง ทำให้เห็นว่าแบบจำลองยังมีศักยภาพในการปรับปรุงเพิ่มเติม เพื่อเพิ่มความแม่นยำได้อีก หากมีเพิ่มจำนวนครั้งในการฝึกข้อมูล ขณะที่รูปที่ 9 แสดงให้เห็นว่าค่า Precision, Recall, mAP50 และ mAP50-95 ในส่วนของ Mask มีค่าน้อยกว่า Box เนื่องจาก Mask คือการตีเส้นขอบรอบวัตถุที่ต้องการหาซึ่งทำได้ยากกว่าการตีกรอบสี่เหลี่ยมของ Box

ในส่วนของการใช้ปัญญาประดิษฐ์ตรวจหาเพลี้ยแป้งและหากเจอให้ส่งรูปไปยังแอปพลิเคชันไลน์ทำได้ดี และแอปพลิเคชันไลน์เป็นแอปพลิเคชันที่ผู้คนส่วนใหญ่เปิดอยู่เป็นประจำและมีระบบแจ้งเตือนทำให้สะดวกต่อบุคคลทั่วไปหรือธุรกิจขนาดเล็ก โดยจะช่วยให้สามารถพบปัญหาก่อนที่จะรุนแรงเกินกว่าจะแก้ไขได้

ผลการทดลองส่วนของการรดน้ำต้นไม้พบว่าการรดเป็นรายต้นจะช่วยประหยัดน้ำได้มากกว่าถึง 69.29% ซึ่งเป็นปริมาณที่มาก และการฟุ้งกระจายของน้ำยังต่ำเนื่องจากรดน้ำระยะใกล้ทำให้โอกาสเกิดโรคที่เกิดจากปริมาณความชื้นในอากาศที่มากเกินไปลดลงมาก

ส่วนของการตรวจหาโรคทำได้ดีมาก อย่างไรก็ตาม ผลจากตารางที่ 2 ที่ไม่ค่อยดีนัก เมื่อตรวจสอบพบว่าเกิดจากที่ปริมาณข้อมูลที่ใช้ในการฝึกลดลง ทำให้เข้าใจผิดว่าดินที่มีสีขาวเป็นเพลี้ยและต้นแคคตัสที่มีสีเหลืองซีดเป็นเพลี้ยดังแสดงในรูปที่ 12

จากสรุปผลการทดลองข้างต้นสามารถสรุปเป็นผลการทดลองรวมดังนี้ คือจำเป็นต้องเพิ่มภาพที่ใช้ในการฝึกข้อมูลให้มากกว่านี้ รวมไปถึงจำนวนครั้งในการฝึกข้อมูล เพื่อให้สามารถตรวจหาโรคได้อย่างมีประสิทธิภาพ ในส่วนของภาพรวมการทำงานของทุกระบบร่วมกันยังมีข้อผิดพลาดอยู่บ้างเป็นบางครั้ง จำเป็นต้องมีการปรับปรุงประสิทธิภาพต่อไปในอนาคต

5. กิตติกรรมประกาศ

ขอบคุณสาขาวิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเอเชียอาคเนย์ ที่เอื้อเพื่อ สถานที่ในการทดลอง และขอบคุณทุนวิจัยจาก มหาวิทยาลัยเอเชียอาคเนย์

6. เอกสารอ้างอิง

- [1] A. Kuznetsova, T. Maleva, and V. Soloviev, "Detecting Apples in Orchards Using YOLOv3 and YOLOv5 in General and Close-Up Images," *Advances in Neural Networks – ISNN*, pp. 233-243, 2020.
- [2] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3," in *Proceedings of the 1st International Conference on Unmanned Vehicle Systems (UVS)*, Muscat, Oman, 2019.
- [3] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Computers and Electronics in Agriculture*, vol. 157, pp. 417-426, Feb. 2019.
- [4] Z. Jiang, L. Zhao, S. Li and Y. Jia, "Real-time object detection method based on improved YOLOv4-tiny," in *Proceedings of Computer Vision and Pattern Recognition*, USA, 2020.
- [5] G. Yang, W. Feng, J. Jin, Q. Lei, X. Li, G. Gui, and W. Wang, "Face Mask Recognition System with YOLOV5 Based on Image

- Recognition,” in *Proceedings of IEEE 6th International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2020, pp. 1398-1404.
- [6] J. Du, “Understanding of Object Detection Based on CNN Family and YOLO,” in *Proceedings of 2nd International Conference on Machine Vision and Information Technology (CMVIT 2018)*, Hong Kong, 2018.
- [7] Y. J. Sang, Z. Wu, P. Guo, H. Hu, H. Xiang, Q. Zhang, and B. Cai, “An Improved YOLOv2 for Vehicle Detection,” *Sensors*, vol. 18, no.12, 2018.
- [8] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, “YOLOv6: A single-stage object detection framework for industrial applications,” in *Proceedings of Computer Vision and Pattern Recognition, USA, 2022*, pp. 9387-9396.
- [9] C.-Y. Wang, A. Bochkovskiy, and H.-Y.M. Liao, “YOLOv7: Train able bag-of-freebies sets new state-of-the-art for real-time object detectors,” in *Proceedings of 2022 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464-7475.
- [10] J. Terven, D. M. C. Esparza, and J. A. R. González, “A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and YOLO-NAS,” *Machine learning and knowledge extraction*, vol. 5, no. 4, pp. 1680-1716, Nov. 2023.
- [11] D. Kijdech, “Weed Classification by Using Convolution Neural Network for Studying and Weed Eliminate Robot,” in *Proceedings of the 7th SAU National Interdisciplinary Conference 2020*, Bangkok, Thailand, 2020, pp. 203-209.
- [12] D. Kijdech and S. Wongbunyong, “Artificial Intelligence in Localization and Classification of Cactus for Automatic Watering Works,” in *Proceedings of The 8th SAU National Interdisciplinary Conference 2021*, Bangkok, Thailand, 2021, pp. 814-821.
- [13] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311-318, Feb. 2018.
- [14] D. Divani, P. Patil, and S.K. Punjabi, “Automated plant Watering system,” in *Proceedings of 2016 International Conference on Computation of Power, Energy Information and Communication (ICCPEIC)*, Melmaruvathur, India, 2016, pp. 180-182.
- [15] M. S. Munir, I.S. Bajwa, and S. M. Cheema, “An intelligent and secure smart watering system using fuzzy logic and blockchain,” *Computer & Electrical Engineering*, vol. 77, pp. 109-119, Jul. 2019.
- [16] Y. Huang, Z. Liu, and Y. Dong, “Design of Remote Wireless Automatic Watering Robot Control System,” in *Proceedings of International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019*, China, 2019, pp. 839-846.

- [17] N. Hema, R. Aswani, and M. Malik, "Plant Watering Autonomous Mobile Robot," *IAES International Journal of Robotics and Automation*, vol.1, no. 3, pp. 152, Sep. 2012.
- [18] P. Tangtisanon, "Small Gardening Robot with Decision-making Watering System," *Sensors and Materials*, vol. 31, no. 6, pp. 1905–1916, 2019.
- [19] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proceedings of 2017 IEEE winter conference on applications of computer vision (WACV)*, USA, 2017, pp. 464-472.
- [20] Ultralytics, "YOLOv8 Docs – Modes: Detect, Segment, Classify, Pose, Track," [Online]. Available: <https://docs.ultralytics.com/modes/>. [Accessed: Jun. 6, 2025].