

Received: 28 เม.ย. 2568

Revised: 6 ม.ค. 2569

Accepted: 8 ม.ค. 2569

หุ่นยนต์ตรวจโรคพริกหวานอัตโนมัติโดยใช้ปัญญาประดิษฐ์และการประมวลผลภาพเชิงลึก:
กรณีศึกษา บริษัท ควอนต่าซินเนอร์จี้ จำกัด

Autonomous Bell Pepper Disease Detection Robot Using Artificial Intelligence
and Deep Image Processing: A Case Study
of Quanta Synergy Co., Ltd

กัญญาพัชร ยศสนิท¹, นิมมิตา สมริน¹, อัจฉริยะณัฐ ปะภูเต¹, ภาณุวัฒน์ เมฆะ¹, ปวีณ เชื้อนแก้ว¹ และ
พยุงค์ศักดิ์ เกษมสำราญ^{1*}

¹สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้เชียงใหม่ 50290 ประเทศไทย

Kanyaphat Kittham¹, Nimmita somrin¹, Oatchariyanut Patita¹, Panuwat Mekha¹,
Paween Khoenkaw¹ and Payungsak Kasemsumran^{1*}

¹Computer Science Department, Faculty of Science, Maejo University, Chaingmai,
Thailand, 50290

*Corresponding author: Payungsak.kae@gmail.com

Abstract

This research presents an autonomous robot for detecting bell pepper diseases using Artificial Intelligence (AI) and Deep Learning for plant image analysis and disease classification. The robot navigates autonomously in greenhouses using the strategy of Ko et al. (2015) and employs Deep Convolutional Neural Networks (CNNs) based on Picón et al. (2019) for classifying four diseases white spot, anthracnose, mosaic virus, and yellow leaf curl.

The robot is equipped with a Logitech B525 webcam, LiDAR (Robosense RS-LiDAR-16), and other sensors for disease and obstacle detection. The system achieves high accuracy in disease classification 90% for white spot, 100% for anthracnose, 100% for mosaic virus, 90% for yellow leaf curl, and 97.5% for healthy leaves. The robot is powered by an 882 Wh battery and has an average power consumption rate of 22.5 W.

The research results demonstrate that the robot helps reduce farmers' workload, shortens disease inspection time, improves disease control efficiency,

reduces chemical usage, and promotes smart agriculture. The results confirm that the robot can be effectively deployed in real agricultural environments and can significantly enhance the accuracy and efficiency of smart farming systems.

Keywords: *Plant Disease Detection Robot; Artificial Intelligence; Deep Learning; Bell pepper Diseases; Automated System*

บทคัดย่อ

งานวิจัยนี้นำเสนอหุ่นยนต์ตรวจโรคพริกหวานอัตโนมัติ โดยใช้ AI และ Deep Learning วิเคราะห์ภาพพืชผ่านโครงข่ายประสาทเทียมเชิงลึก (CNNs) ตามแนวทางของ Picón et al. (2019) หุ่นยนต์เคลื่อนที่ตามกลยุทธ์ของ Ko et al. (2015) เพื่อตรวจสอบโรค 4 ชนิด ได้แก่ ใบจุดสีขาว (*Cercospora capsici*), แอนแทรคโนส (*Colletotrichum* spp.), ไวรัสโมเสค (Potyvirus), และใบหงิกเหี่ยว (Begomovirus) ซึ่งเป็นโรคสำคัญที่สร้างความเสียหายทางเศรษฐกิจ ตัวหุ่นยนต์ติดตั้งกล้อง HD 720p @ 30fps, เลนส์ Fixed-focus, FOV 69 องศา และเทคโนโลยี RightLight™ 2 ช่วยปรับแสง พร้อมเซ็นเซอร์ Robosense RS-LiDAR-16 แบบหมุน 16 เลเซอร์ ตรวจจับได้ไกลสุด 150 เมตร ช่วยตรวจสอบโรคและหลบสิ่งกีดขวาง ระบบให้ความแม่นยำในการจำแนกโรคใบจุดสีขาว 90%, แอนแทรคโนส 100%, ไวรัสโมเสค 100%, ใบหงิกเหี่ยว 90%, และใบปกติ 97.5% หุ่นยนต์ใช้พลังงานจากแบตเตอรี่ 882 Wh อัตราการใช้พลังงานเฉลี่ย 22.5 W ช่วยลดภาระเกษตรกร ลดเวลาในการตรวจสอบโรค เพิ่มโอกาสในการควบคุมโรค ลดการใช้สารเคมีที่ไม่จำเป็น และส่งเสริมการเกษตรอัจฉริยะได้อย่างมีประสิทธิภาพ ผลลัพธ์ยืนยันว่าหุ่นยนต์สามารถนำไปใช้งานจริงในแปลงเกษตรได้อย่างมีประสิทธิภาพ และช่วยยกระดับระบบเกษตรอัจฉริยะให้มีความแม่นยำและรวดเร็วมากยิ่งขึ้น

คำสำคัญ: หุ่นยนต์ตรวจโรคพืช, ปัญญาประดิษฐ์, การเรียนรู้เชิงลึก, โรคพริกหวาน, ระบบอัตโนมัติ

1. บทนำ

ในยุคปัจจุบันการเกษตรกรรมมีความสำคัญอย่างยิ่งในการผลิตอาหารเพื่อสนับสนุนการเจริญเติบโตของประชากรโลก การเกษตรในยุคปัจจุบันต้องเผชิญกับความท้าทายมากมาย ไม่ว่าจะเป็นการเปลี่ยนแปลงสภาพภูมิอากาศ การแพร่ระบาดของศัตรูพืชและโรคพืช และการขาดแคลนแรงงานเกษตร ปัญหาที่เกษตรกรประสบอยู่เสมอคือการแพร่ระบาดของโรคพืช โรคพืชนั้นสามารถทำลายผลผลิตของพืชได้อย่างมากทำให้ผลผลิตลดลง คุณภาพของผลผลิตต่ำลง เช่น ขนาด สี กลิ่น

และ รสชาติ นอกจากนี้ยังสามารถแพร่ไปยังพืชชนิดอื่นได้อย่างรวดเร็ว โรคพืชบางชนิดสามารถทำลายพันธุ์พืชดั้งเดิมที่มีความสำคัญทางเศรษฐกิจ ความยากในการตรวจสอบและระบุโรค การตรวจสอบและระบุโรคพืชนั้นต้องการนักวิชาการหรือแรงงานที่มีความเชี่ยวชาญ ความรู้และความชำนาญเฉพาะทาง เกษตรกรบางคนอาจไม่มีความรู้หรือเครื่องมือในการตรวจสอบโรคพืชอย่างแม่นยำ หุ่นยนต์ตรวจโรคพริกหวานอัตโนมัติจึงถูกพัฒนาขึ้นมาเพื่อช่วยเพิ่มประสิทธิภาพในการตรวจสอบโรคพริกหวานที่รวดเร็วและแม่นยำมากยิ่งขึ้นลดความเสียหายจากโรคพริกหวานและการแก้ไขปัญหาได้อย่างรวดเร็วและมีประสิทธิภาพส่งผลให้ผลผลิตเพิ่มขึ้น ลดเวลาและแรงงานที่ใช้ในการตรวจสอบโรคพริกหวาน ด้วยตนเอง ช่วยให้เกษตรกรสามารถใช้สารเคมีในการควบคุมโรคเฉพาะจุด ลดการใช้สารเคมีเกินจำเป็น ซึ่งส่งผลดีต่อสิ่งแวดล้อมและสุขภาพของผู้บริโภคการมีเทคโนโลยีที่สามารถช่วยในการตรวจสอบและแก้ไขปัญหาโรคพริกอย่างรวดเร็วและมีประสิทธิภาพช่วยให้เกษตรกรมีความมั่นใจในการจัดการฟาร์มและเพิ่มความสามารถในการแข่งขันในตลาด การแพร่ระบาดของโรคพืชเป็นปัญหาสำคัญทางการเกษตรที่ส่งผลกระทบต่อผลผลิตและคุณภาพของพืชเศรษฐกิจ โดยเฉพาะพริกหวานซึ่งเป็นพืชที่มีความไวต่อโรคหลายชนิด การตรวจสอบโรคด้วยแรงงานคนต้องอาศัยผู้เชี่ยวชาญ ใช้เวลามาก และมีโอกาสเกิดความคลาดเคลื่อนสูง งานวิจัยด้านปัญญาประดิษฐ์และการประมวลผลภาพ ได้พิสูจน์แล้วว่าสามารถช่วยเพิ่มประสิทธิภาพการจำแนกโรคพืชได้อย่างแม่นยำ โดยงานวิจัยของ Picón et al. (2019) และงานอื่น ๆ ในด้าน CNN สำหรับโรคพืชแสดงศักยภาพสูง อย่างไรก็ตามยังมีช่องว่างด้านการบูรณาการระบบ AI เข้ากับแพลตฟอร์มหุ่นยนต์เคลื่อนที่อัตโนมัติ ที่สามารถใช้งานในพื้นที่เกษตรจริงได้ งานวิจัยนี้จึงมุ่งเน้นพัฒนาแพลตฟอร์มหุ่นยนต์เคลื่อนที่นำทางและ AI เพื่อให้เกิดระบบตรวจโรคพริกหวานแบบครบวงจรที่ใช้ได้จริงในแปลงเกษตร

2. วิธีดำเนินการ

การพัฒนาหุ่นยนต์ตรวจจับโรคพริกหวานอัตโนมัติเป็นนวัตกรรมที่มีศักยภาพในการยกระดับประสิทธิภาพภาคการเกษตร โดยสามารถตรวจจับโรคพืชได้อย่างรวดเร็ว และแม่นยำ ช่วยลดความสูญเสียทางเศรษฐกิจ และลดการใช้สารเคมีที่ไม่จำเป็น เพื่อให้โครงการนี้บรรลุผลสำเร็จ จำเป็นต้องมีการศึกษาค้นคว้า ในหลายด้าน ซึ่งรวมถึงเทคโนโลยีการประมวลผลภาพ ปัญญาประดิษฐ์ ระบบเซ็นเซอร์ กลไกการทำงานของหุ่นยนต์ และการศึกษาเกี่ยวกับพริกหวาน และโรคที่พบในพริกหวาน โดยมีรายละเอียดการศึกษาดังนี้

2.1 ข้อมูลทั่วไปเกี่ยวกับพริกหวานและโรคที่เกี่ยวข้อง

พริกหวานเป็นพืชในวงศ์ Solanaceae ซึ่งมีความหลากหลายทั้งในด้านพันธุกรรม และลักษณะของผล พริกหวานมีหลายสี เช่น เขียว แดง เหลือง และส้ม ซึ่งสีของผลจะขึ้นอยู่กับชนิดของพืช และระยะเวลาการสุก พริกหวานสามารถปลูกได้ในหลากหลายสภาพอากาศ ตั้งแต่เขตร้อนจนถึง

เขตที่มีอากาศอบอุ่น พริกหวานมีสารอาหารที่สำคัญโดยเฉพาะ วิตามิน C ซึ่งเป็นสารต้านอนุมูลอิสระที่ดี นอกจากนี้ยังมี วิตามิน A B6 โฟเลต และสารต้านอนุมูลอิสระอื่น ๆ ที่มีประโยชน์ต่อสุขภาพ

2.1.1 ลักษณะของต้นพริกหวาน (Bell Pepper) แสดงภาพต้นพริกหวานและผลพริกหวาน ดังภาพที่ 1 และภาพที่ 2



ภาพที่ 1 ต้นพริกหวาน



ภาพที่ 2 ผลพริกหวาน

ที่มา: <https://hkm.hrdi.or.th/knowledge/detail/52>

2.1.2 โรคที่พบบ่อยในพริกหวาน

1. โรคใบจุดสีเทา (Cercospora Leaf Spot) สาเหตุ : เชื้อรา *Cercospora capsica*

ลักษณะอาการ : ใบมีจุดสีเทาหรือเหลืองเป็นวงกลม

การป้องกันและรักษา : ใช้สารป้องกันเชื้อรา ปรับปรุงการระบายอากาศ ดังภาพที่ 3



ภาพที่ 3 โรคริ่บจุดสีขาว

2. โรคนแอนแทรคโนส (Anthracnose)

สาเหตุ : เชื้อรา *Colletotrichum* spp.

ลักษณะอาการ : ผลพริกเป็นแผลจุดดำ ลูกกลมจนเน่า

การป้องกันและรักษา : คัดเลือกเมล็ดพันธุ์ที่ปลอดโรค หลีกเลี่ยงพื้นที่ชื้น ดังภาพที่ 4



ภาพที่ 4 โรคนแอนแทรคโนส

3. โรควัไรส์โมเสค (Mosaic Virus)

สาเหตุ : วัไรส์ เช่น Pepper Mild Mottle Virus (PMMoV)

ลักษณะอาการ : ใบ บิดเบี้ยว ต้นแคระแกร็น

การป้องกันและรักษา : ควบคุมแมลงพาหะ เช่น เพลี้ยอ่อน ดังภาพที่ 5



ภาพที่ 5 โรคไวรัสโมเสค

4. โรคใบเหลือง (Yellow Leaf Curl Virus)

สาเหตุ : ไวรัส Tomato Yellow Leaf Curl Virus (TYLCV)

ลักษณะอาการ : ใบหงิกงอเหลือง การเจริญเติบโตของต้นลดลง

การป้องกันและรักษา : ใช้วิธีควบคุมแมลงพาหะ ลดการแพร่กระจายของไวรัส ดังภาพที่ 6



ภาพที่ 6 โรคใบเหลือง

2.2 วิเคราะห์บทบาทหน้าที่

ระบบหุ่นยนต์ตรวจจับโรคพริกอัตโนมัติโดยมีผู้ใช้ระบบ คือ ผู้พัฒนา (Developer) และเกษตรกรหรือผู้ใช้งาน (User) ซึ่งมีบทบาทในการทำงานที่เกี่ยวข้องกับระบบ ดังนี้

2.2.1 ผู้พัฒนา (Developer) มีหน้าที่ดังนี้

1. สามารถเก็บข้อมูลแผนที่
2. สามารถเก็บข้อมูลต้นพริกหวาน
3. สามารถวิเคราะห์ภาพโดยใช้ AI
4. ทดสอบซอฟต์แวร์

5. สามารถควบคุมตัวหุ่นยนต์ได้

6. สามารถทำงานผ่านการควบคุมระยะไกล (remote control) ตามข้อจำกัดที่กำหนดไว้

7. การบำรุงรักษาและอัปเดตระบบ

2.2.2 ผู้ใช้งาน (User) มีหน้าที่ดังนี้

1. สามารถดูข้อมูลเชิงลึกเกี่ยวกับพริกหวานได้

2. สามารถดูภาพจากกล้อง

3. ตรวจสอบการประมวลผลภาพ

1) ผู้ใช้งานสามารถตรวจสอบกระบวนการประมวลผลภาพที่เกิดขึ้นบนหุ่นยนต์ได้ โดยดูข้อมูลที่ ถูก "Publish" และ "Subscribe" ระหว่าง "Camera Node" และ "Image Processing Node"

4. สามารถควบคุมตัวหุ่นยนต์ได้

5. สามารถทำงานผ่านการควบคุมระยะไกลตามข้อจำกัดที่กำหนดไว้

2.2.3 ข้อจำกัด

1. อุปกรณ์บางส่วนไม่ได้ออกแบบไว้เพื่อกันน้ำ

2. วัตถุที่ต่ำหรือสูงเกินไปอาจจะทำให้ตัวเซนเซอร์มองไม่เห็น

3. ต้องให้ผู้เชี่ยวชาญเฉพาะด้านเป็นผู้ติดตั้ง

2.3 ขั้นตอนการทำงานของระบบหุ่นยนต์

1. เริ่มต้น ระบบเริ่มต้นการทำงาน

2. หุ่นยนต์เริ่มทำงาน หุ่นยนต์เปิดระบบและเริ่มกระบวนการทำงานการเคลื่อนที่

3. เก็บข้อมูลแผนที่ หุ่นยนต์ทำการสร้างแผนที่ของพื้นที่ทำงาน โดยใช้การทำงานของ เซ็นเซอร์ LiDAR ,กล้อง และ ROS ในการสร้างแผนที่

4. เก็บข้อมูลต้นพริกหวาน หุ่นยนต์เคลื่อนที่ไปยังต้นพริกหวานและทำการเก็บข้อมูลภาพ เกี่ยวข้องกับต้นพริกหวาน

5. วิเคราะห์ภาพโดยใช้ AI ระบบปัญญาประดิษฐ์ (AI) จะทำการวิเคราะห์ภาพหรือข้อมูลที่ เก็บมา เพื่อตรวจหาความผิดปกติหรือโรคในต้นพริกหวาน

6. ตรวจสอบการประมวลผลข้อมูล

1) ถ้าไม่สามารถประมวลผลข้อมูลต้นพริกหวานได้ ระบบจะกลับไปเก็บข้อมูลต้นพริก หวานใหม่อีกครั้ง

2) ถ้าตรวจจับและระบุโรคได้ ระบบจะดำเนินการในขั้นตอนต่อไป

7. ส่งข้อมูลไปยังเว็บเบราว์เซอร์ ระบบจะส่งข้อมูลผลการวิเคราะห์และระบุโรคไปยังเว็บเบราว์เซอร์ เพื่อให้ผู้ใช้สามารถดูผลลัพธ์ได้

8. สิ้นสุด ระบบสิ้นสุดการทำงาน

2.4 การพัฒนาการควบคุมหุ่นยนต์และการประมวลผล

การพัฒนาโปรเจกต์หุ่นยนต์ตรวจโรคพริกหวานอัตโนมัติสามารถแบ่งออกเป็น 2 ส่วนหลักๆ ดังนี้

2.4.1 ขอบเขตการพัฒนาระบบหุ่นยนต์

1. การออกแบบและสร้างหุ่นยนต์

1) ออกแบบโครงสร้างของหุ่นยนต์ให้เหมาะสมกับการทำงานในแปลงพริกหวาน โดยตัวโครงหุ่นยนต์นั้นใช้กล่องพลาสติกกันน้ำ ABS

2) เลือกและติดตั้งอุปกรณ์ฮาร์ดแวร์ เช่น กล้อง LiDAR เซ็นเซอร์ต่างๆ, มอเตอร์, และแบตเตอรี่ขนาด 29.4 โวลต์ 30 แอมป์

2. การพัฒนาระบบควบคุม

1) พัฒนาระบบควบคุมการเคลื่อนที่ของหุ่นยนต์ โดยใช้ ROS (Robot Operating System)

2) พัฒนาระบบนำทางและหลีกเลี่ยงสิ่งกีดขวาง โดยใช้ข้อมูลจาก LiDAR และเซ็นเซอร์อื่นๆ

3. การจำลองการทำงาน

1) ใช้โปรแกรม Gazebo ในการจำลองการทำงานของหุ่นยนต์ในสภาพแวดล้อมเสมือนจริง

2) ใช้โปรแกรม RVIZ ในการแสดงผลข้อมูลจากเซ็นเซอร์และควบคุมการทำงานของหุ่นยนต์และสร้างแผนที่

2.4.2 ขั้นตอนการพัฒนาระบบหุ่นยนต์

1. สร้าง URDF robot (Create URDF robot)

1) ขั้นตอนนี้เป็นการสร้างแบบจำลองหุ่นยนต์ในรูปแบบ URDF (Unified Robot Description Format) ซึ่งเป็นรูปแบบไฟล์ XML ที่ใช้อธิบายโครงสร้างทางกายภาพของหุ่นยนต์ เช่น ขนาดรูปร่าง และข้อต่อต่างๆ

2) แบบจำลอง URDF นี้จะถูกใช้ในขั้นตอนต่อไป เช่น การจำลองใน Gazebo และการควบคุมหุ่นยนต์

2. Gazebo simulation & Lidar

1) ขั้นตอนนี้เป็นการทำงานการทำงานของหุ่นยนต์ในสภาพแวดล้อมเสมือนจริงโดยใช้โปรแกรม Gazebo

2) มีการจำลองการทำงานของเซ็นเซอร์ LiDAR เพื่อทดสอบการรับรู้สภาพแวดล้อมและการหลีกเลี่ยงสิ่งกีดขวาง

3) Build moter controller (สร้างตัวควบคุมมอเตอร์)

4) Simulation & Control (การจำลองและการควบคุม)

5) Slam tool box

5.1) ขั้นตอนนี้เป็นการใช้เครื่องมือ SLAM (Simultaneous Localization and Mapping) เพื่อสร้างแผนที่ของสภาพแวดล้อมและระบุตำแหน่งของหุ่นยนต์ในแผนที่นั้น

5.2) SLAM เป็นเทคนิคที่สำคัญสำหรับการนำทางของหุ่นยนต์ในสภาพแวดล้อมที่ไม่รู้จัก

6) Mapping and navigation (การทำแผนที่และการนำทาง)

6.1) ขั้นตอนนี้เป็นการใช้ข้อมูลจาก SLAM และเซ็นเซอร์อื่นๆ เพื่อสร้างแผนที่ของสภาพแวดล้อมและวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์เป็นการทำให้หุ่นยนต์สามารถ เคลื่อนที่ไปยังจุดหมายปลายทางได้อย่างอัตโนมัติ

7) webcam streaming (การสตรีมภาพจากเว็บแคม)

7.1) ขั้นตอนนี้เป็นการทำงานให้หุ่นยนต์สามารถส่งภาพจากเว็บแคมไปยังคอมพิวเตอร์หรืออุปกรณ์อื่นๆ ได้

7.2) ภาพจากเว็บแคมสามารถนำไปใช้ในการตรวจจับวัตถุหรือการควบคุมหุ่นยนต์จากระยะไกล

2.4.3 ขอบเขตการพัฒนาการประมวลผลภาพ

1. การเตรียมข้อมูลภาพ

1) รวบรวมข้อมูลภาพถ่ายโรคพริกหวาน 4 ชนิด (โรคใบจุดสีขาว, โรคแอนแทรคโนส, โรคไวรัสโมเสก, และโรคใบเหลือง) พร้อมกับข้อมูลของต้นพริกหวานที่ปกติ

2) ใช้ OpenCV ในการประมวลผลภาพเบื้องต้น เช่น การปรับขนาดภาพ, การปรับปรุงคุณภาพ, และการ normalize ภาพ

2. การพัฒนาโมเดล Deep Learning

1) ใช้ TensorFlow ในการสร้างและฝึกฝนโมเดลสำหรับจำแนกโรคพริกหวาน

2) แบ่งข้อมูลภาพถ่ายเป็นชุด train, validation, และ test เพื่อประเมินประสิทธิภาพของโมเดล

3. การนำโมเดลไปใช้งานบนหุ่นยนต์

1) แปลงโมเดล TensorFlow ให้เป็นรูปแบบ TensorFlow Lite เพื่อลดขนาดไฟล์และเพิ่มประสิทธิภาพการทำงานบนหุ่นยนต์

2) เขียนโปรแกรม Python เพื่อดึงภาพจากเว็บแคม, ประมวลผลภาพ, และใช้โมเดล TensorFlow Lite ในการจำแนกโรค

2.4.4 ขั้นตอนการพัฒนาการประมวลผลภาพ

1. เตรียมระบบคอมพิวเตอร์ (Prepare the computing system) ขั้นตอนนี้เป็นการเตรียมความพร้อมของระบบคอมพิวเตอร์ที่ใช้ในการพัฒนาการประมวลผลภาพ

2. ติดตั้งระบบปฏิบัติการ เฟรมเวิร์ก และไลบรารี (Install the operating system, frameworks, and libraries)

1) ขั้นตอนนี้เป็นการติดตั้งระบบปฏิบัติการ Ubuntu 20.04 LTS

2) ติดตั้งเฟรมเวิร์กที่จำเป็น เช่น TensorFlow สำหรับการพัฒนาโมเดล Deep Learning

3) ติดตั้งไลบรารีที่จำเป็น เช่น OpenCV สำหรับการประมวลผลภาพ

3. เตรียมชุดข้อมูล (Prepare the dataset)

1) ขั้นตอนนี้เป็นการรวบรวมและเตรียมชุดข้อมูลภาพถ่ายต้นพริกหวานที่ปกติ และ โรคพริกหวาน 4 ชนิด

2) มีการปรับขนาดภาพ, การปรับปรุงคุณภาพภาพ, และการ normalize ภาพเพื่อให้ข้อมูลเหมาะสมกับการเทรนโมเดล

4. ดำเนินการเรียนรู้ถ่ายโอนโมเดลโดยใช้ชุดข้อมูลที่เตรียมไว้ (Perform model transfer learning using the prepared dataset)

1) ขั้นตอนนี้เป็นการใช้เทคนิคการเรียนรู้ถ่ายโอน (transfer learning) เพื่อฝึกฝนโมเดล Deep Learning ด้วยชุดข้อมูลที่เตรียมไว้

2) เป็นการปรับแต่งโมเดลที่ได้รับการฝึกฝนมาแล้ว (pre-trained model) ให้สามารถจำแนกโรค พริกหวานได้อย่างแม่นยำ

5. ประเมินประสิทธิภาพของโมเดล (Evaluate the model's performance)

1) ขั้นตอนนี้เป็นการประเมินประสิทธิภาพของโมเดล Deep Learning ที่ฝึกฝนแล้ว

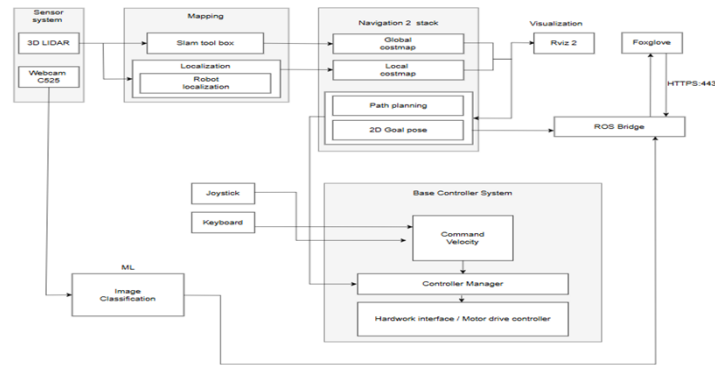
2) มีการใช้ชุดข้อมูลทดสอบ (test dataset) เพื่อวัดความแม่นยำ (accuracy)

6. ผสานรวมโมเดลเข้ากับ ROS 2 (Integrate the model with ROS 2)

1) ขั้นตอนนี้เป็นการนำโมเดล Deep Learning ที่ฝึกฝนแล้วไปใช้งานบนหุ่นยนต์โดย ผสานรวมโมเดลเข้ากับระบบ ROS 2

2) มีการแปลงโมเดล TensorFlow ให้เป็นรูปแบบ TensorFlow Lite เพื่อลดขนาดไฟล์ และเพิ่ม ประสิทธิภาพการทำงานบนหุ่นยนต์

2.5 Flow การทำงานโดยรวมของทั้งสอง



ภาพที่ 7 Flow การทำงานของระบบ

1. ระบบเซ็นเซอร์ (Sensor System)

1) 3D LiDAR เซ็นเซอร์ LiDAR 3 มิติ ทำหน้าที่ตรวจจับสภาพแวดล้อมรอบๆ หุ่นยนต์ เพื่อสร้างแผนที่และระบุตำแหน่งของหุ่นยนต์

2) Webcam C525 กล้องเว็บแคม C525 ทำหน้าที่ถ่ายภาพเพื่อนำไปประมวลผล

2. การทำแผนที่ (Mapping)

1) Slam Tool Box ใช้เครื่องมือ SLAM (Simultaneous Localization and Mapping) เพื่อสร้างแผนที่ของสภาพแวดล้อม

2) Localization ระบุตำแหน่งของหุ่นยนต์ในแผนที่ที่สร้างขึ้น

3) Robot Localization ปรับปรุงความแม่นยำของตำแหน่งหุ่นยนต์

3. ระบบนำทาง (Navigation 2 Stack)

1) Global Costmap สร้างแผนที่ (costmap) เพื่อวางแผนเส้นทางระยะยาว

2) Local Costmap สร้างแผนที่ เพื่อหลีกเลี่ยงสิ่งกีดขวางในระยะใกล้

3) Path Planning วางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์

4) 2D Goal Pose กำหนดจุดหมายปลายทางสองมิติให้หุ่นยนต์

4. ระบบควบคุมพื้นฐาน (Base Controller System)

1) Joystick/Keyboard ผู้ใช้สามารถควบคุมหุ่นยนต์โดยใช้จอยสติ๊กหรือคีย์บอร์ด

2) Command Velocity ส่งคำสั่งความเร็วไปยังตัวควบคุมมอเตอร์

3) Controller Manager จัดการการทำงานของตัวควบคุมมอเตอร์

4) Hardwork Interface/Motor Drive Controller ควบคุมการทำงานของมอเตอร์
ขับเคลื่อน

5. การประมวลผลภาพ (Image Classification)

1) Image Classification ทำหน้าที่จำแนกภาพที่ได้จากเว็บแคม

6. การแสดงผล (Visualization)

1) Rviz แสดงผลข้อมูลจากเซ็นเซอร์ แผนที่ และสถานะของหุ่นยนต์ในรูปแบบสามมิติ

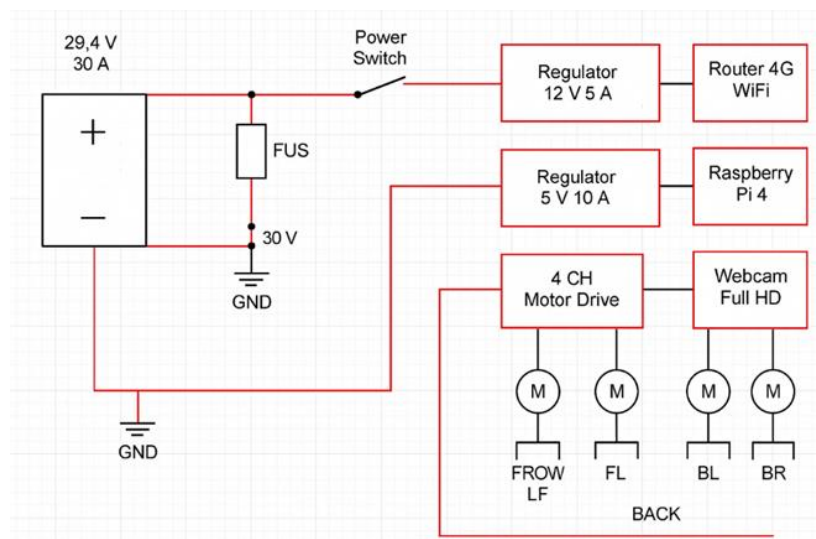
2) Foxglove แสดงผลข้อมูลจากระบบ ROS ผ่านเว็บเบราว์เซอร์ (HTTPS:443) โดยใช้

ROS Bridge

7. การเชื่อมต่อ (ROS Bridge)

1) ROS Bridge เชื่อมต่อระบบ ROS กับเว็บเบราว์เซอร์ ทำให้สามารถแสดงผลข้อมูลผ่าน
Foxglove ได้

2.6 แผนผังการเชื่อมต่อระบบไฟฟ้าและอิเล็กทรอนิกส์



ภาพที่ 8 แผนผังการเชื่อมต่อระบบไฟฟ้าและอิเล็กทรอนิกส์

2.6.1 แหล่งจ่ายไฟหลัก

1) แบตเตอรี่ 29.4V 30A ให้พลังงานกับทั้งระบบ ผ่านฟิวส์ (FUSE) เพื่อป้องกันกระแส
เกิน

2) FUSE 30A และ Power Switch ใช้เพื่อความปลอดภัยและควบคุมการจ่ายไฟหลัก
เข้าไปยังระบบ

2.6.2 ตัวแปลงแรงดัน (Regulator)

1) 12V 5A Regulator จ่ายไฟให้กับอุปกรณ์ที่ต้องการแรงดัน 12V เช่น Router 4G WiFi และ Lidar 3D

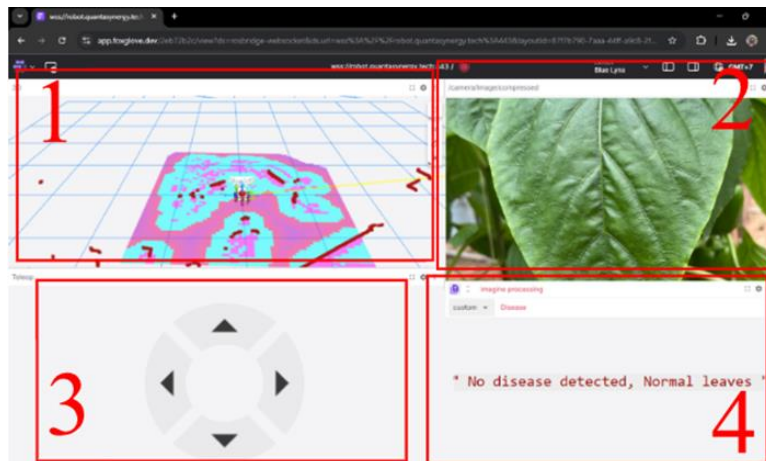
2) 5V 10A Regulator จ่ายไฟให้กับอุปกรณ์ 5V เช่น Raspberry Pi 4 และ Webcam Full HD

2.6.3 ระบบควบคุมการขับเคลื่อน 4 CH Motor Drive ไดรฟ์ขับเคลื่อนมอเตอร์ 4 ช่อง (4 ช่อง หมายถึงควบคุมมอเตอร์ได้ 4 ตัวอย่างอิสระ) เชื่อมต่อกับ มอเตอร์ 4 ตัว

- 1) FL (Front Left) – มอเตอร์หน้าซ้าย
- 2) FR (Front Right) – มอเตอร์หน้าขวา
- 3) BL (Back Left) – มอเตอร์หลังซ้าย
- 4) BR (Back Right) – มอเตอร์หลังขวา

2.7 หน้าจอแสดงผลการส่งข้อมูลการตรวจสอบและวิเคราะห์โรคพริกหวาน

หน้าจอนี้แสดงผลข้อมูลที่ได้จากระบบ ROS (Robot Operating System) ของหุ่นยนต์ ผู้ใช้สามารถดูแผนที่ 3 มิติของ สภาพแวดล้อม ภาพจากเว็บแคม และควบคุมการเคลื่อนที่ของหุ่นยนต์ได้ผ่านหน้าจอนี้ และแสดงการวิเคราะห์โรคของใบพริกหวาน โดยระบุว่า “No disease detected, Normal leaves” ซึ่งหมายความว่าไม่พบโรค และใบพริกหวานมีลักษณะปกติ ดังภาพที่ 9



ภาพที่ 9 หน้าจอแสดงผลการส่งข้อมูลการตรวจสอบและวิเคราะห์โรคพริกหวาน จากภาพแสดงส่วนของการแสดงแผนที่ 3 มิติ, ภาพจากเว็บแคม, ส่วนควบคุมการเคลื่อนที่ของหุ่นยนต์ และ ส่วนแสดงการวิเคราะห์โรคของใบพริกหวาน ตามลำดับ

3. ผลการศึกษา

3.1 การทดสอบการตรวจจับภาพ (Image Detection Test)

วัตถุประสงค์

เพื่อประเมินความสามารถของเซ็นเซอร์กล้องในการจับภาพใบพริกหวานภายใต้สภาพแสงด้วยกล้อง Webcam Logitech รุ่น B525 ความละเอียดวิดีโอสูงสุด HD 720p (1280 × 720) 30fps, เลนส์ Fixed-focus, มุมมอง (FOV) 69 องศา และรองรับเทคโนโลยี RightLight™ 2 เครื่องมือที่ใช้

1. หุ่นยนต์ตรวจสอบโรค
2. โปรแกรมวิเคราะห์ภาพ
3. แหล่งกำเนิดแสงต่าง ๆ (กลางวันและกลางคืน)
4. TES-1333 Solar Power Meter สำหรับแบ่งระหว่างกลางวันและกลางคืน

ขั้นตอนการทดสอบ

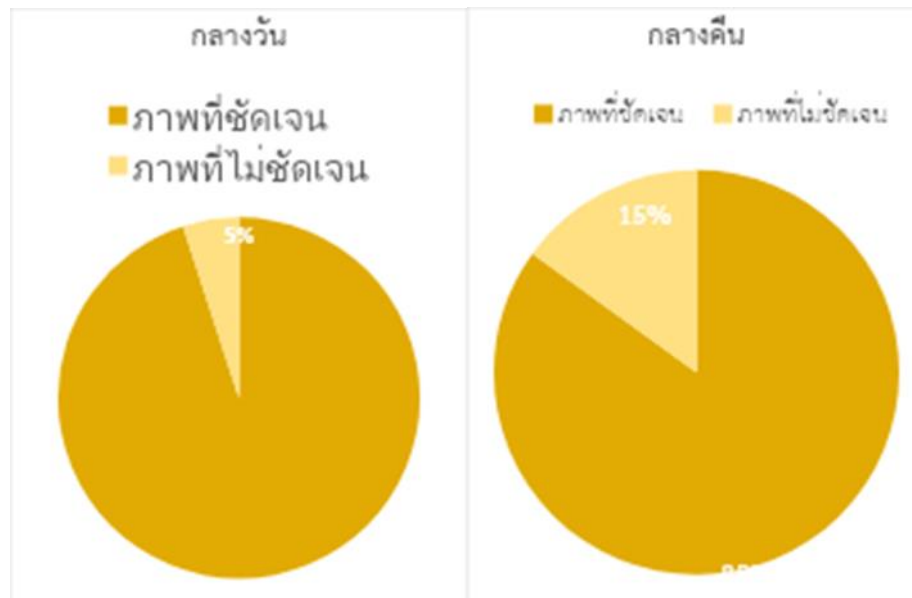
1. จับภาพในสภาพแสงต่าง ๆ
2. วิเคราะห์คุณภาพโดยพิจารณาจากความคมชัดและรายละเอียดที่สามารถตรวจจับโรคได้เพื่อให้การทดสอบการตรวจจับภาพมีความแม่นยำและตรงกับสภาพแสงที่แท้จริงค่าความเข้มแสงที่ใช้ในการ กำหนดช่วงกลางวันและกลางคืนมีดังนี้
 - 1) กลางวัน (Daytime) ค่าความเข้มแสงที่ 100 W/m^2 ขึ้นไปจะถือว่าเป็นช่วงกลางวัน โดยสามารถเป็นช่วงเวลาที่มีแสงแดดส่องถึงหุ่นยนต์และระบบตรวจจับในกรณีที่แสงแดดไม่เต็มที่
 - 2) กลางคืน (Nighttime) ค่าความเข้มแสง $\leq 0 \text{ W/m}^2$ หรือค่าที่ต่ำกว่าจะถือว่าเป็นช่วงกลางคืน ซึ่งในกรณีนี้จะไม่มีการตรวจจับจากดวงอาทิตย์และแสงจากแหล่งกำเนิดอื่น ๆ จะเป็นตัวเสริมแสงให้การทดสอบ

ตารางที่ 1 ผลการทดสอบการตรวจจับภาพ

สภาพแสง	จำนวนภาพที่ถ่าย (ภาพ)	ภาพชัดเจน (%)	ภาพไม่ชัดเจน (%)
95-150(กลางวัน)	100	95	5
≤ 0 (กลางคืน)	100	85	15

จากข้อมูลตารางที่ 1 แสดงผลการทดสอบประสิทธิภาพการจับภาพภายใต้สภาวะแสงแตกต่างกัน จำนวนภาพที่ทำการทดสอบเท่ากับ 100 ภาพในแต่ละช่วงแสง พบว่า ภายใต้สภาพแสง

กลางวัน (95–150 ลักซ์) ภาพที่ได้มีความชัดเจนร้อยละ 95 และไม่ชัดเจนร้อยละ 5 ขณะที่สภาพแสงกลางคืน (≤ 0 ลักซ์) ให้ภาพชัดเจนร้อยละ 85 และไม่ชัดเจนร้อยละ 15 ทั้งนี้ สภาพแสงมีผลต่อความชัดเจนของภาพที่ได้ อย่างมีนัยสำคัญ จากกราฟแสดงผล Image Detecion Test ดังภาพที่ 10



ภาพที่ 10 กราฟแสดงผล Image Detecion Test

จากภาพที่ 10 พบว่า ในสภาวะแสงกลางวัน ภาพที่ได้มีความชัดเจนร้อยละ 95 และไม่ชัดเจนร้อยละ 5 ในขณะที่สภาวะแสงกลางคืน ภาพที่ชัดเจนลดลงเหลือร้อยละ 85 และไม่ชัดเจนเพิ่มขึ้นเป็นร้อยละ 15 ผลการวิเคราะห์ชี้ให้เห็นว่าสภาพแสงส่งผลต่อคุณภาพของภาพที่ได้ อย่างชัดเจน

3.2 การทดสอบการประมวลผลภาพเพื่อระบุโรค (Image Processing Disease Detection Test)

วัตถุประสงค์

ประเมินประสิทธิภาพของระบบในการจำแนกโรคพริกหวาน 4 ชนิด ได้แก่ โรคจุดสีขาว โรคแอน โรคน้ำแตรคโนส โรคไวรัสโมเสคและโรคใบเหลือง วิเคราะห์ความแม่นยำของระบบในการจำแนกโรคแต่ละชนิด เปรียบเทียบผลการจำแนกของระบบกับการจำแนกโดยผู้เชี่ยวชาญ

ขั้นตอนการทดสอบ

1. นำชุดข้อมูลทดสอบมาป้อนเข้าสู่โมเดลที่ฝึกสอนแล้วโดยแบ่งภาพออกเป็น 2 ชุด คือ ชุดฝึกสอน (Training Set) 80% ชุดทดสอบ (Testing Set) 20%
2. บันทึกผลลัพธ์การจำแนกของโมเดล

ผลการทดสอบ

ตารางที่ 2 เมทริกซ์ความสับสน (Confusion Matrix)

จริง/ทำนาย	โรคใบจุดสี ขาว	โรคแอนแทรก โนส	โรคไวรัสโม เสค	โรค ใบเหลือง	ใบที่ไม่เป็น โรค
โรคใบจุดสีขาว	80	0	0	10	10
โรคแอนแทรกโนส	0	100	0	0	0
โรคไวรัสโมเสค	0	0	100	0	0
โรคใบเหลือง	0	0	0	60	40
ใบที่ไม่เป็นโรค	10	0	0	0	90

คำอธิบาย

TN : True Negative (ทำนายว่าไม่เป็นโรค และถูกต้อง)

TP : True Positive (ทำนายว่าไม่เป็นโรค แต่จริงๆ เป็นโรค)

FP : False Positive (ทำนายว่าเป็นโรค แต่จริงๆ ไม่เป็นโรค)

FN : False Negative (ทำนายว่าไม่เป็นโรค แต่จริงๆ เป็นโรค)

1. โรคใบจุดสีขาว

TP = 80 ทำนายถูกว่าเป็นโรคใบจุดสีขาว FP = 10 ทำนายผิดว่าเป็นโรคใบจุดสีขาว แต่จริง ๆ แล้วเป็นใบที่ไม่เป็นโรค FN = 20 ทำนายว่าไม่เป็นโรคใบจุดสีขาว แต่จริง ๆ แล้วเป็น (10 ใบเหลือง + 10 ใบไม่เป็นโรค) TN = 390 ทำนายว่าไม่เป็นโรคใบจุดสีขาว และจริง ๆ ก็ไม่เป็น
สรุป มีความคลาดเคลื่อนทั้ง FP และ FN ซึ่งอาจเกิดจากลักษณะใบที่คล้ายกัน เช่น ใบจุดสีขาวกับ ใบเหลือง

2. โรคแอนแทรกโนส

TP = 100 FP = 0 FN = 0 TN = 400

สรุป โมเดลสามารถพยากรณ์โรคแอนแทรกโนสได้ถูกต้อง 100% ไม่มีข้อผิดพลาด

3. โรคไวรัสโมเสค

TP = 100 FP = 0 FN = 0 TN = 400

สรุป โมเดลสามารถพยากรณ์โรคไวรัสโมเสคได้แม่นยำเช่นกัน ไม่มีข้อผิดพลาด

4. โรคใบเหลือง

TP = 60 FP = 10 ทำนายผิดว่าเป็นโรคใบเหลือง แต่จริง ๆ เป็นโรคใบจุดสีขาว

FN = 40 ทำนายว่าไม่เป็นโรคใบเหลือง แต่จริง ๆ เป็น TN = 390

สรุป โมเดลยังมีความสับสนระหว่างโรคใบเหลืองกับใบที่ไม่เป็นโรค และใบจุดสีขาว

5. ใบที่ไม่เป็นโรค

TP = 0 ไม่มี TP เพราะ "ไม่เป็นโรค" ถือเป็นคลาสลบ FP = 10 ทำนายผิดว่าเป็นโรค (ใบจุดสีขาว) แต่ จริง ๆ แล้วไม่เป็นโรค

FN = 0 ไม่มีกรณีที่ใบที่ไม่เป็นโรคถูกทำนายว่าไม่เป็นโรค (FN) TN = 90

สรุป โมเดลมีความแม่นยำค่อนข้างดีในการพยากรณ์ใบไม่เป็นโรค แต่ยังมี FP จากการทำนายผิดเป็นโรคใบจุดสีขาว

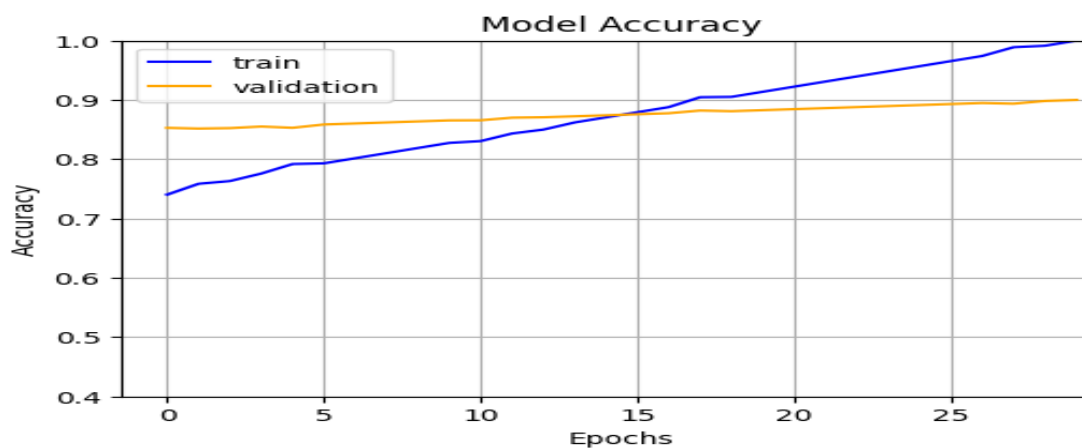
สรุปผลการคำนวณ Accuracy

1. โรคใบจุดสีขาว: 90%

2. โรคแอนแทรคโนส: 100%

3. โรคไวรัสโมเสค: 100%

4. โรคใบเหลือง: 90% 5. ใบที่ไม่เป็นโรค: 97.5% ดังกราฟแสดงผล Model Accuracy



ภาพที่ 11 กราฟแสดงผล Accuracy

จากภาพที่ 11 แสดงกราฟความแม่นยำของโมเดล (Model Accuracy) เปรียบเทียบระหว่างชุดข้อมูลฝึก (train) และชุดข้อมูลตรวจสอบ (validation) ตลอดช่วง 30 รอบการฝึก (epochs) โดยผลการทดสอบพบว่าความแม่นยำของชุดฝึกเพิ่มขึ้นอย่างต่อเนื่องจนถึงระดับ 1.0 ในขณะที่ชุดตรวจสอบมีค่าความแม่นยำคงที่อยู่ที่ประมาณ 0.9 แสดงให้เห็นว่าโมเดลสามารถเรียนรู้ข้อมูลฝึกได้ดี แต่เริ่มมีแนวโน้มเกิดการ overfitting หลังจากจำนวน epoch เพิ่มขึ้น

3.3 การทดสอบการเคลื่อนที่ของหุ่นยนต์ (Robot Navigation Test)

วัตถุประสงค์

เพื่อทดสอบการเคลื่อนที่ของหุ่นยนต์ในแปลงพริกหวานที่มีสิ่งกีดขวาง

ขั้นตอนการทดสอบ

1. กำหนดเส้นทางให้หุ่นยนต์เคลื่อนที่ในแปลง
2. วางสิ่งกีดขวางและสังเกตการหลบหลีก

ตารางที่ 3 ผลการทดสอบการเคลื่อนที่ของหุ่นยนต์

ระยะทาง (เมตร)	ความเร็วเฉลี่ย (km/hr)	จำนวนการชน
10	1.8	0
10	1.44	0
10	1.08	1

จากตารางที่ 3 แสดงผลการทดลองการเคลื่อนที่ของหุ่นยนต์ โดยให้หุ่นยนต์เคลื่อนที่ในระยะทางเท่ากันคือ 10 เมตร พบว่าถ้าหุ่นยนต์วิ่งด้วยความเร็วเฉลี่ย 1.8 กม./ชม. และ 1.44 กม./ชม. จะไม่มีการชนเลย ซึ่งหมายความว่าหุ่นยนต์เคลื่อนที่ได้ค่อนข้างมั่นคงที่ความเร็วเหล่านี้ แต่เมื่อลดความเร็วลงเหลือ 1.08 กม./ชม. กลับมีการชนเกิดขึ้น 1 ครั้ง แสดงให้เห็นว่าความเร็วมีผลต่อการเคลื่อนที่ของหุ่นยนต์ ถ้าความเร็วไม่เหมาะสมก็อาจทำให้หุ่นยนต์ไม่เสถียรและชนได้

โดยเฉลี่ยแล้วมนุษย์เดินด้วยความเร็วเฉลี่ยที่ 4.3-5.4 km/hr และ สูงสุดอยู่ที่ 1 m/s 3.6 km/hr

สรุป หุ่นยนต์สามารถเคลื่อนที่ผ่านเส้นทางได้โดยมีการชนเพียงเล็กน้อยในระยะทางที่ยาวขึ้น

3.4 การทดสอบระยะตรวจจับสูงสุดของเซ็นเซอร์ตรวจจับสิ่งกีดขวาง (Maximum Detection Range Test)

วัตถุประสงค์

เพื่อประเมินระยะตรวจจับสูงสุดของเซ็นเซอร์ Robosense RS-LiDAR-16 ในการตรวจจับสิ่งกีดขวาง

ขั้นตอนการทดสอบ

1. วางสิ่งกีดขวางขนาดต่าง ๆ ในระยะต่าง ๆ จากหุ่นยนต์

2. บันทึกระยะที่เซ็นเซอร์สามารถตรวจจับได้

ตารางที่ 4 ผลการทดสอบระยะตรวจจับสูงสุดของเซ็นเซอร์ตรวจจับสิ่งกีดขวาง

ประเภทสิ่งกีดขวาง	ระยะตรวจจับสูงสุด (เมตร)	
	ระยะที่ใกล้ที่สุด	ระยะที่ไกลที่สุด
ต้นไม้	0.15	10.2 ± 0.1
โต๊ะ	0.15	11.4 ± 0.1
มนุษย์	0.15	12.0 ± 0.1

จากตารางที่ 4 สรุป เซ็นเซอร์สามารถตรวจจับสิ่งกีดขวางขนาดใหญ่ได้ดีในระยะไกล ขณะที่สิ่งกีดขวางขนาดเล็กมีระยะตรวจจับสั้นกว่า

3.5 การทดสอบความเสถียรของระบบพลังงาน (Power Stability Test)

วัตถุประสงค์

เพื่อทดสอบความเสถียรของระบบพลังงานของหุ่นยนต์เมื่อทำงานต่อเนื่อง โดยค่าความจุของแบตเตอรี่ (battery) = 882 Wh แรงดัน = 24 v

ขั้นตอนการทดสอบ

1. ให้หุ่นยนต์ทำงานต่อเนื่องในแปลงเป็นเวลา 2 ชั่วโมง
2. ตรวจสอบการทำงานของระบบพลังงาน

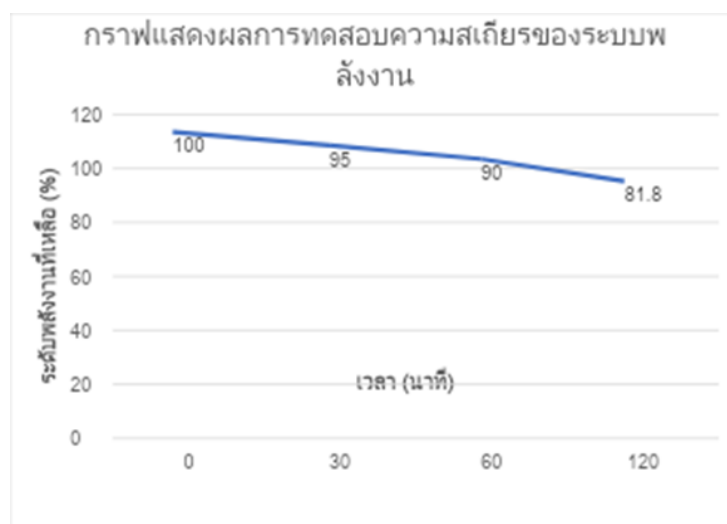
ตารางที่ 5 การทดสอบความเสถียรของระบบพลังงาน

เวลา (นาท)	ระดับพลังงานที่เหลือ (%)	ความเร็วในการเคลื่อนที่ (km/hr)	สถานะการทำงาน
30	95.46	1.8	ทำงานปกติ
60	90.9	1.8	ทำงานปกติ
120	81.8	1.8	ทำงานปกติ

จากตารางที่ 5 แสดงผลการทดสอบความเสถียรของระบบพลังงาน โดยสังเกตความสัมพันธ์ระหว่างเวลา ระดับพลังงานที่เหลืออยู่ และความเร็วในการเคลื่อนที่ของหุ่นยนต์ พบว่าเมื่อเวลาเพิ่มขึ้นจาก 30 นาที เป็น 120 นาที ระดับพลังงานลดลงอย่างต่อเนื่องจาก 95.46% เป็น 81.8% ในขณะที่ความเร็วในการเคลื่อนที่ยังคงที่ 1.8 km/hr และสถานะการทำงานยังคงเป็น "ทำงานปกติ" ซึ่งบ่งชี้ว่าระบบพลังงานสามารถจ่ายพลังงานได้อย่างเสถียรภายใต้ภาระการทำงานที่ความเร็วคงที่ในช่วงเวลาดังกล่าว

- โดยอัตราการบริโภคพลังงานของหุ่นยนต์โดยรวมอยู่ที่ 22.5 วัตต์
- ระยะเวลาใช้งานในกรณีที่ไม่มีเคลื่อนที่อยู่ที่ 39 ชั่วโมง 12 นาที
- ระยะเวลาใช้งานในกรณีที่มีการเคลื่อนที่ด้วยความเร็ว 1.8 กิโลเมตร/ชั่วโมงอยู่ที่ 11 ชั่วโมง 7 นาที

กราฟแสดงผล การทดสอบความเสถียรของระบบพลังงาน (Power Stability Test) ดังภาพที่ 12



ภาพที่ 12 กราฟแสดงผลการทดสอบความเสถียรของระบบพลังงาน

กราฟนี้แสดงผลการทดสอบความเสถียรของระบบพลังงาน โดยแกน X คือเวลา (นาที) และแกน Y คือระดับพลังงานที่เหลืออยู่ (%) เส้นกราฟแสดงให้เห็นว่าเมื่อเวลาผ่านไป ระดับพลังงานจะค่อยๆ ลดลง จากเริ่มต้นที่ประมาณ 100% ที่เวลา 0 นาที ลดลงเหลือ 95% ที่ 30 นาที 90% ที่ 60 นาที และ 81.8% ที่ 120 นาที ซึ่งบ่งชี้ถึงอัตราการใช้พลังงานของระบบเมื่อเวลาผ่านไป

สรุป จากการทดสอบเมื่อหุ่นยนต์ทำงานต่อเนื่องเป็นเวลา 2 ชั่วโมง โดยระดับพลังงานลดลง ซึ่งเป็นการคายประจุที่สม่ำเสมอ และหุ่นยนต์ยังคงทำงานได้ตามปกติที่ความเร็ว 1.8 km/hr จากผลระบบพลังงานมีความเสถียร และเพียงพอสำหรับการใช้งานในภาคสนาม โดยไม่พบความผิดปกติใด ๆ ระหว่างการทำงาน ซึ่งเหมาะสมกับการใช้งานต่อเนื่องในแปลงปลูก

4. สรุปผล และอภิปรายผล

โครงการนี้พัฒนาหุ่นยนต์ต้นแบบที่สามารถตรวจจับโรคพริกหวานได้อย่างอัตโนมัติ โดยมีการบูรณาการองค์ความรู้ทางด้านฮาร์ดแวร์และซอฟต์แวร์ ประกอบด้วย

1. ด้านฮาร์ดแวร์ หุ่นยนต์ถูกสร้างขึ้นด้วยโครงสร้างที่แข็งแรง สามารถเคลื่อนที่ได้คล่องตัว มีการติดตั้งเซ็นเซอร์ตรวจจับสิ่งกีดขวาง กล้องสำหรับบันทึกภาพและระบบพลังงานที่เสถียร

2. ด้านซอฟต์แวร์ มีการพัฒนาอัลกอริทึมสำหรับประมวลผลภาพวิเคราะห์และตรวจจับโรคพริกหวานจากภาพถ่าย รวมถึงระบบควบคุมการเคลื่อนที่ของหุ่นยนต์

จากการทดสอบพบว่าหุ่นยนต์สามารถทำงานได้ตามวัตถุประสงค์ สามารถตรวจจับโรคใน ใบพริกหวานได้อย่างแม่นยำ เคลื่อนที่หลบหลีกสิ่งกีดขวางในแปลงเกษตรได้ และระบบพลังงานมีความเสถียรเพียงพอต่อการใช้งาน

ข้อเสนอแนะ

ข้อเสนอแนะแนวทางการพัฒนาโครงการหุ่นยนต์ตรวจจับโรคพริกหวานอัตโนมัติเพื่อเพิ่มประสิทธิภาพและความสามารถในการใช้งานจริง มีดังนี้

1. เพิ่มประสิทธิภาพการตรวจจับโรค

- 1) ปรับปรุงอัลกอริทึมการประมวลผลภาพ เพื่อเพิ่มความแม่นยำในการตรวจจับโรค โดยเฉพาะในสภาพแสงน้อย อาจพิจารณาใช้เทคนิค Deep Learning หรือ Machine Learning เพื่อช่วยในการวิเคราะห์ภาพและ แยกแยะลักษณะของโรค ได้อย่างมีประสิทธิภาพมากขึ้น

- 2) ขยายฐานข้อมูลภาพเพิ่ม เพิ่มจำนวนความหลากหลายของภาพใบพริกหวานทั้งที่เป็นโรคและไม่เป็นโรคในฐานข้อมูล เพื่อใช้ในการฝึกฝนอัลกอริทึม และเพิ่มความแม่นยำในการตรวจจับ

- 3) เพิ่มโรคพริกหวาน เพิ่มความหลากหลายของโรคมกกว่านี้ เพื่อเพิ่มความหลากหลายในการตรวจจับ

2. เพิ่มฟังก์ชันการใช้งาน

ระบบบันทึกและวิเคราะห์ข้อมูล พัฒนาให้หุ่นยนต์สามารถบันทึกข้อมูลการแพร่ระบาดของโรคในแปลง วิเคราะห์แนวโน้มการเกิดโรค และแสดงผลในรูปแบบที่เข้าใจง่าย เช่น แผนที่แสดงตำแหน่งที่พบโรค กราฟแสดงสถิติการเกิดโรค เป็นต้น

3. การใช้งานจริง

ทดสอบในสภาพแวดล้อมที่หลากหลาย ทำการทดสอบในสภาพแวดล้อมและสภาพอากาศที่หลากหลาย เช่น สภาพอากาศร้อนจัด ฝนตกพื้นที่ที่มีความลาดชันเพื่อให้มั่นใจว่าหุ่นยนต์สามารถทำงานได้อย่างมีประสิทธิภาพในทุกสภาวะ

5. เอกสารอ้างอิง

- เพชรรัช ประระไทย, นฤพร เต็งไตรรัตน์ และ ชัชวาลย์ ชัยชนะ, การสร้างแบบจำลอง Tilapia Image Segmentation ด้วยวิธีการเรียนรู้อย่างลึกด้วยเครือข่ายคอนโวลูชันพีระมิดเชิงพื้นที่ โดยใช้ YOLOv8 (Tilapia Image Segmentation Modeling using Deep Learning with Spatial Pyramid Pooling Convolutional Networks based on YOLOv8), ปีที่ 10 ฉบับที่ 2 (พฤษภาคม - สิงหาคม 2567)
- เพชรรัช ประระไทย, นฤพร เต็งไตรรัตน์ และ Wai Lok Woo, การคาดการณ์พื้นที่น้ำท่วม บริเวณพื้นที่ลุ่มน้ำชุมชนเมืองจังหวัดเชียงใหม่ โดยใช้การสำรวจระยะไกลและการเรียนรู้การถดถอย (Flood Area Prediction in Urban Watershed of Chiang Mai Province Using Remote Sensing and Regression Learning), ปีที่ 11 ฉบับที่ 2 (พฤษภาคม - สิงหาคม 2568)
- ภาณุวัฒน์ เมฆะ และ ญัฐนิชา ตียะสุขเศรษฐ์, การเปรียบเทียบประสิทธิภาพของอัลกอริทึมจำแนกภาพสำหรับโรคตาในมนุษย์ (Comparison of Efficiency of Image Classification Algorithms for Human Eye Diseases), ปีที่ 10 ฉบับที่ 3 (กันยายน - ธันวาคม 2567)
- เลอศักดิ์ โพธิ์ทอง, ภาคนิ เล่ามูล, ภัททิยา โมรา และ อนุพงศ์ สุขประเสริฐ, การสร้างตัวแบบการเรียนรู้เชิงลึกเพื่อการวินิจฉัยโรคมาลาเรียอย่างรวดเร็วและแม่นยำในภาพถ่ายฟิล์มเลือด (Creating a Deep Learning Model for pid and Accurate Diagnosis of Malaria in Blood Film Images), ปีที่ 11 ฉบับที่ 2 (พฤษภาคม - สิงหาคม 2568)
- สรารุณี ภัทรภูวดล, สนิท สิทธิ, ภาณุวัฒน์ เมฆะ และ พยุงศักดิ์ เกษมสำราญ, การตรวจจับสิ่งปนเปื้อนขนาดเล็กผ่านการประมวลผลภาพโดยใช้ภาษา Python (Small Contaminant Detection via Image Processing using Python), ปีที่ 11 ฉบับที่ 1 (มกราคม - เมษายน 2568)
- Aoo Pattana-anurak. (2023, March 20). TensorFlow คืออะไร? Thai Config System & Digital Marketing. <https://thaiconfig.com/artificial-intelligence-ai/what-is-tensorflow/>
- Chris Lalancette. (2023, June 27). ROS installation options. ROS Wiki. <https://wiki.ros.org/ROS/Installation>
- Mandala AI. (2023, March 23). Artificial intelligence (AI) คืออะไร? เครื่องมือไหนบ้างที่ใช้. Mandala AI. <https://www.mandalasystem.com/blog/th/297/What-is-artificial-intelligence-AI>

- Minaphinat, V. (2018, February 28). **Machine Learning คืออะไร?** Medium.<https://medium.com/investic/machine-learning-คืออะไร-fa8bf6663c07>
- patda9. (2019, July 15). **ลองใช้ TensorFlow.js — Part 1.** Medium.<https://medium.com/@patdanai.duangwaropas/ลองใช้-tensorflow-js-part-1-b1c1c4c055e8>
- Phongchit, N. (2018, September 19). **Convolutional Neural Network (CNN) คืออะไร.** Medium. <https://medium.com/@natthawatphongchit>
- RomanS. (2024, August 27). **Foxglove Studio.** ROS Wiki.<https://wiki.ros.org/Foxglove/Studio>
- Sphinx. (2024). **Distributions.** ROS Documentation.<https://docs.ros.org/en/jazzy/Releases.html>
- Surapong Kanoktipsatharporn. (2020, January 12). **MobileNet คืออะไร: สอน TensorFlow.js สร้าง Image Classification ด้วย MobileNet รุ่นสำเร็จรูป – tfjs ep.7.** BUA Labs. <https://www.bualabs.com/archives/3439/tensorflow-js-tutorial-build-image-classification-javascript-mobilenet-pretrained-model-tfjs-ep-7/>