

หุ่นยนต์กำจัดวัชพืชตรวจจับด้วยการประมวลผลภาพ

ROBOTIC WEED CONTROL SYSTEM

USING IMAGE PROCESSING



กนกวรรณ สัจวารินทร์

ณัฐมน บุญประเสริฐ

ธีรदनย์ จันทร์หอม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ปรอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์กำจัดวัชพืชตรวจจับด้วยการประมวลผลภาพ

ROBOTIC WEED CONTROL SYSTEM USING IMAGE PROCESSING

ผู้จัดทำ

1. นางสาวกนกวรรณ สังวารภรณ์ รหัสนักศึกษา 62010010
2. นางสาวณัฐมน บุญประเสริฐ รหัสนักศึกษา 62010285
3. นายธีรดนัย จันทร์หอม รหัสนักศึกษา 62010436



อาจารย์ที่ปรึกษา

(ดร.วัชระ นัตริวิริยะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์กำจัดวัชพืชตรวจจับด้วยการประมวลผลภาพ

นางสาวกนกวรรณ	สังวราภรณ์	62010010
นางสาวณัฐมน	บุญประเสริฐ	62010285
นายธีรคนย์	จันทร์หอม	62010436
ดร.วัชร	นัครวิริยะ	อาจารย์ที่ปรึกษา
ปีการศึกษา 2565		

บทคัดย่อ

เกษตรกรต้องเผชิญกับปัญหาเรื่องวัชพืชที่ส่งผลกระทบต่อผลผลิตทางการเกษตรเป็นจำนวนมาก โดยแนวทางที่เกษตรกรนิยมใช้แก้ปัญหาในปัจจุบันคือการพ่นสารเคมี ซึ่งเป็นวิธีที่ก่อให้เกิดอันตรายต่อสภาพแวดล้อม ส่งผลกระทบต่อเนื้อที่ทั้งตัวเกษตรกรและผู้บริโภค

ปัญหานี้จึงนำมาเสนอการพัฒนาหุ่นยนต์กำจัดวัชพืช ซึ่งเป็นหุ่นยนต์คาร์ทีเซียนเคลื่อนที่เป็นเส้นตรงสามแกนขนาด 100 x 120 x 150 เซนติเมตร โดยใช้ Raspberry Pi 4 เป็นหน่วยประมวลผลหลักที่ติดตั้ง ใช้ Arduino Mega 2560 และส่วนต่อขยาย RAMPS 1.6 ในการขับสเต็ปมอเตอร์ NEMA 17 ทั้งหมด 4 ตัวเพื่อเคลื่อนหุ่นยนต์ ใช้กล้อง Web Camera ในการถ่ายภาพแล้วนำไปใช้ตรวจหาวัชพืชโดยการนำภาพที่ได้มาทำการแบ่งภาพ (Segmentation) ด้วย VGG 16-Unet เพื่อทำการจำแนกกระหว่างดินกับพืช จากนั้นนำภาพที่ได้จากการแบ่งภาพมาทำการลดสัญญาณรบกวนด้วยการใช้ Morphology operation ทำการติกรอบเพื่อระบุตำแหน่ง และนำจำแนกพืชและวัชพืช (Classification) โดยใช้โมเดล VGG16 แล้วจึงทำการส่งตำแหน่งวัชพืชไปยังหุ่นยนต์ เพื่อให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งที่ได้รับ และสามารถใส่โมดูลเลเซอร์ในการกำจัดวัชพืชได้

ซึ่งผลการดำเนินงานคือหุ่นยนต์สามารถขับไปถ่ายภาพ และหาตำแหน่งวัชพืชและทำการเคลื่อนที่ไปยังตำแหน่งและสามารถใช้เลเซอร์เผาทำลายได้สำเร็จ

Robotic Weed Control System using Image Processing

Ms. Kanokvan Sangvaraporn 62010010

Ms. Nathamon Boonprasert 62010285

Mr. Teeradon Chanhom 62010436

Dr. Watchara Chatwiriya Advisor

Academic Year 2022

ABSTRACT

This thesis presents the development of a three-axis Cartesian robot with dimensions of 1.0 x 1.2 x 1.5 meters, which is powered by a Raspberry Pi 4 as the main processor and controlled by the ROS operating system. The robot uses an Arduino Mega 2560 with RAMPS 1.6 to drive NEMA 17 stepper motors for its movement. A web camera is used to capture images, which are then processed using VGG16-Unet for image segmentation to distinguish between soil and plants. The resulting images are then processed using morphology operations to reduce noise and to determine the position of the plants and weeds using VGG16 for classification. The robot is then directed to move towards the detected weed position and can use a laser module to eliminate the weed.

The results show that the robot can move towards the weed position, capture images, identify and eliminate the weed successfully.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จตามวัตถุประสงค์ด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษา ดร. วิชระ ฉัตรวิริยะ ที่ให้ความรู้ คำปรึกษา คำแนะนำ และแนวทางในการแก้ปัญหา รวมถึงการสนับสนุนด้านอุปกรณ์เครื่องมือและสถานที่ในการทำวิจัย ตลอดจนความช่วยเหลือในทุก ๆ ด้าน จนทำให้การวิจัยในครั้งนี้สำเร็จลุล่วงไปด้วยดี ขอขอบพระคุณเป็นอย่างสูงสำหรับโอกาสและความปรารถนาดีเสมอมา

ขอขอบพระคุณนายสนชัย ภูมิภาค นางสาวกานดา รงค์สวัสดิ์ นางสาวณัฐรุจา วัฒนสุข และเพื่อน ๆ พี่ ๆ น้อง ๆ ในห้องปฏิบัติการสมองกลฝังตัว (Embedded System Laboratory) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับการให้คำปรึกษาและคำแนะนำเพิ่มเติมในการวิเคราะห์และแก้ปัญหา รวมถึงคำแนะนำในการจัดทำเล่มปริญญานิพนธ์ฉบับนี้

ขอขอบพระคุณเพื่อน ๆ และน้อง ๆ ทุกคนในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การสนับสนุนในการจัดหาอุปกรณ์และเป็นกำลังใจเสมอมา

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดา และครอบครัวของคณะผู้จัดทำที่คอยให้การสนับสนุนและเป็นกำลังใจเสมอมา ทำให้ปริญญานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี

กนกวรรณ สัจจวราภรณ์
ณัฐมน บุญประเสริฐ
ธีรคนย์ จันทร์หอม

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 แผนการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 งานวิจัยที่เกี่ยวข้อง.....	3
2.2 ทฤษฎีที่เกี่ยวข้อง.....	5
บทที่ 3 การออกแบบและการพัฒนา.....	13
3.1 การวิเคราะห์ความต้องการ.....	13
3.2 การออกแบบ.....	13
3.3 การประเมินผลภาพ.....	25
บทที่ 4 ผลการดำเนินงาน.....	28
4.1 ผลการพัฒนาหุ่นยนต์.....	28
4.2 ผลการพัฒนาระบบโมดูลจำแนกระหว่างพืชและวัชพืช.....	36
บทที่ 5 สรุปผล และข้อเสนอแนะ.....	39
5.1 สรุปผลการพัฒนาโครงการ.....	39
5.2 ข้อเสนอแนะและแนวทางการพัฒนาต่อ.....	39
บรรณานุกรม.....	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
ภาคผนวก.....	42
ภาคผนวก ก การคำนวณการเคลื่อนที่ของมอเตอร์.....	43
ภาคผนวก ข รายการวัสดุ.....	44



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
1.1 แผนการดำเนินงาน.....	2
3.1 สถานะการเก็บข้อมูลบน Logger.....	21
3.2 Protocol การส่งข้อมูลจากบอร์ด Controller ไปยัง Arduino Mega.....	22
3.2 Protocol การรับข้อมูลจาก Arduino Mega 2560 ไปยังบอร์ด Controller.....	23
4.1 การทดสอบการเคลื่อนที่ของหุ่นยนต์.....	33
4.2 การทดสอบจับเวลาที่ใช้ในการเคลื่อนที่.....	34



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
2.1 แผนภาพกระบวนการจำแนกพืชและวัชพืช.....	4
2.2 แผนภาพกระบวนการตรวจจับวัชพืช.....	4
2.3 โครงสร้างของ Neural Networks.....	6
2.4 Convolutional Neural Networks.....	7
2.5 การแบ่งภาพออกเป็น ส่วน ๆ ตามวัตถุ (Image Segmentation).....	7
2.6 ลักษณะหุ่นยนต์แต่ละประเภท.....	9
2.7 ระบบปฏิบัติการหุ่นยนต์ (Robot Operating System).....	10
2.8 ลักษณะของ Stator และ Rotor.....	11
2.9 ขั้นตอนการทำงานของมอเตอร์ในโหมด Full step.....	11
2.10 ขั้นตอนการทำงานของมอเตอร์ในโหมด Half step.....	12
3.1 หุ่นยนต์ FarmBot Genesis.....	14
3.2 การออกแบบโครงสร้างและขนาดของหุ่นยนต์.....	14
3.3 Raspberry Pi 4 Model B.....	15
3.4 Arduino Mega 2560.....	15
3.5 RAMPS 1.6.....	16
3.6 NEMA 17 Stepper Motor.....	16
3.7 DRV8825.....	17
3.8 EGA Webcam USB Camera.....	17
3.9 Endstop Switch Module.....	18
3.10 Laser Module.....	18
3.11 Relay 5V 1 Channel.....	19
3.12 Power Supply.....	19
3.13 แผนภาพการเชื่อมต่ออุปกรณ์.....	20
3.14 การออกแบบการเชื่อมต่อระบบควบคุมหุ่นยนต์.....	21
3.15 การออกแบบการสื่อสารระหว่าง ROS Node และ Topic.....	22
3.16 Weed Detection Process Diagram.....	25
3.17 ภาพที่เก็บจากแปลงปลูกพืช.....	26
3.18 ตัวอย่างชุดข้อมูลที่นำมาใช้.....	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา VII นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูป	หน้า
3.19 ตัวอย่างชุดข้อมูลที่นำมาทำการเพิ่มชุดข้อมูล.....	27
3.20 ตัวอย่างภาพที่ได้จากการเพิ่มชุดข้อมูล.....	27
4.1 ผลลัพธ์หุ่นยนต์กำจัดวัชพืช.....	28
4.2 โครงสร้างไฟล์แพ็คเกจ weeder_robot.....	29
4.3 ผลการทำงานของ launch file.....	29
4.4 ทดสอบภาพที่ได้ในแต่ละความละเอียดของกล้อง.....	30
4.5 การวิเคราะห์สัดส่วนจริงต่อสัดส่วนของภาพ.....	31
4.6 การวางแผนลำดับการเคลื่อนที่เพื่อเก็บภาพ.....	32
4.7 การทดสอบ Communication Protocol.....	33
4.8 การวิเคราะห์การเคลื่อนที่ของหุ่นยนต์.....	34
4.9 กราฟแสดงค่า IoU ระหว่างการเทรนนิ่ง.....	36
4.10 ผลลัพธ์จากการเรียกใช้โมเดลการแบ่งส่วนภาพ.....	37
4.11 ภาพนำเข้าและผลลัพธ์จากการลดสัญญาณรบกวนในภาพ.....	37
4.12 ภาพที่ได้ทำการวาดกล่องรอบวัตถุ.....	38
4.13 ภาพที่ได้ทำการจำแนกข้อมูลและคืนค่าตำแหน่งของวัชพืช.....	38

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันเกษตรกรต้องเผชิญกับปัญหาโรคระบาดด้านภายใต้ข้อจำกัดต่าง ๆ หนึ่งในปัญหาสำคัญที่เกิดขึ้นผลกระทบต่อผลผลิตทางการเกษตรอย่างเลี่ยงไม่ได้คือ ปัญหาเรื่องวัชพืชที่ส่งผลกระทบต่อทั้งการก่อกำเนิดโรคพืช แย่งอาหารและน้ำ อีกทั้งยังเป็นแหล่งพักตัวของแมลงศัตรูพืช โดยทั่วไปในการจัดการวัชพืชของเกษตรกร มีแนวทางดูแลป้องกันอยู่หลายแนวทาง หนึ่งในแนวทางที่เกษตรกรนิยมใช้คือ การพ่นสารเคมีเพื่อป้องกันหรือกำจัด ซึ่งเป็นวิธีที่ทำได้เร็ว มีประสิทธิภาพและใช้แรงงานคนน้อย แต่อย่างไรก็ตามการใช้สารเคมีนั้นนำไปสู่การเกิดปัญหาด้านสภาพแวดล้อมในอนาคต ทำให้ดินและน้ำมีการปนเปื้อน เป็นอันตรายต่อสภาพแวดล้อม และเกิดผลกระทบต่อเนื้อที่ตัวเกษตรกรและผู้บริโภค

เพื่อลดการเกิดผลเสียดังกล่าว ปรินญาณิพนธ์ฉบับนี้ จึงนำเสนอการนำเทคโนโลยีการประมวลผลภาพ และหุ่นยนต์มาใช้ในการกำจัดวัชพืชเชิงกายภาพ โดยใช้กล้องจับภาพ นำมาประมวลผลหาตำแหน่งของวัชพืช แล้วให้หุ่นยนต์ทำการกำจัดวัชพืชด้วยการเผาทำลาย มาใช้รับมือกับปัญหาวัชพืช หลีกเลี่ยงและลดปริมาณการใช้สารเคมีเกินความจำเป็น และลดแรงงานคนที่ใช้ในการกำจัดวัชพืช

1.2 วัตถุประสงค์

- 1) เพื่อประยุกต์ใช้เทคโนโลยีการประมวลผลภาพ ในการตรวจจับและระบุตำแหน่งของวัชพืชในแปลงทดสอบ
- 2) เพื่อพัฒนาหุ่นยนต์ในการเคลื่อนที่และกำจัดวัชพืชบนแปลงทดสอบ

1.3 ขอบเขตของโครงการ

ระบบตัวเคลื่อนที่ของหุ่นยนต์

- 1) หุ่นยนต์ทำงานบนรางเลื่อน สามารถเคลื่อนที่ได้ 3 แกน
- 2) หุ่นยนต์ทำงานในห้องทดลองที่สามารถควบคุมแสงและดินได้ บนแปลงที่มีพื้นที่จำกัด

ระบบการประมวลผลภาพ

- 1) สามารถจำแนกพืชหลักและวัชพืชได้ โดยพืชหลักที่ใช้ ได้แก่ โหระพา
- 2) สามารถระบุตำแหน่งวัชพืชได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) ต้นพีชมีระยะห่างระหว่างต้น 30 เซนติเมตร
- 4) ภาพที่ใช้ประมวลผล เป็นภาพสี RGB มีขนาด 640 x 480 พิกเซล

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) สามารถใช้การประมวลผลภาพในการตรวจจับและระบุตำแหน่งวัชพีชได้
- 2) สามารถสร้างหุ่นยนต์ที่สามารถตรวจจับและกำจัดวัชพีชได้

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

No	ขั้นตอนการดำเนินงาน	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1	กำหนดขอบเขต								
2	ศึกษาและออกแบบ								
3	จัดหาอุปกรณ์								
4	พัฒนาระบบ								
5	ทดสอบและปรับปรุง								
6	จัดทำเล่มรายงานและนำเสนอ								

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

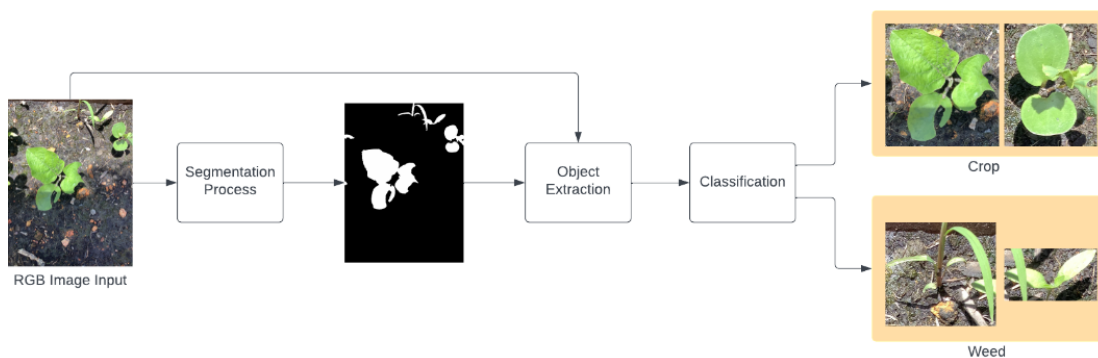
ในบทนี้จะเป็นการอธิบายถึงงานวิจัยและทฤษฎีที่นำมาใช้ ประกอบด้วย 2 หัวข้อหลัก คือ 1) งานวิจัยที่เกี่ยวข้อง ในที่นี้นำมาทั้งหมด 3 งานวิจัยได้แก่ การใช้เทคนิคการเรียนรู้เชิงลึกในการจำแนกพืชและวัชพืช และการใช้ระบบปฏิบัติการหุ่นยนต์ในงานอุตสาหกรรม และ 2) ทฤษฎีที่เกี่ยวข้อง โดยในทฤษฎีที่เกี่ยวข้อง ประกอบด้วยรายละเอียดใน 2 ส่วนหลัก คือ พื้นฐานการเพาะปลูก และเทคโนโลยีที่ใช้ในการพัฒนา โดยในส่วนพื้นฐานการเพาะปลูกจะอธิบายถึงความหมายของวัชพืชและการกำจัดวัชพืช ส่วนเทคโนโลยีที่ใช้พัฒนาจะอธิบายถึง เทคนิคการจำแนกข้อมูล เทคโนโลยีการประมวลผลภาพ หุ่นยนต์ในงานอุตสาหกรรม และเทคโนโลยีการควบคุมและการเคลื่อนที่ ดังรายละเอียดต่อไปนี้

2.1 งานวิจัยที่เกี่ยวข้อง

2.1.1 Crop and Weeds Classification for Precision Agriculture Using Context-Independent Pixel-Wise Segmentation

งานวิจัยนี้นำเสนอในปี 2019 โดย Mulham Fawakherji และคณะ ได้ทำการพัฒนาระบบสำหรับหุ่นยนต์การเกษตรเพื่อแก้ปัญหาวัชพืช โดยทำการพัฒนาการจำแนกพืชและวัชพืชโดยใช้เทคนิคการเรียนรู้เชิงลึก ซึ่งผู้เขียนพัฒนาระบบ 2 ส่วนด้วยกัน ได้แก่ ส่วนของการลบพื้นหลังภาพเพื่อทำการแยกแยะระหว่างดินกับพืช และส่วนของการจำแนกพืชและวัชพืช

งานวิจัยนี้ได้แบ่งขั้นตอนในการพัฒนาเป็น 3 ขั้นตอน ได้แก่ 1) ส่วนของการแบ่งส่วนภาพ ในขั้นตอนนี้ใช้โครงสร้างของ VGG16-UNet มาใช้ในการแบ่งส่วนภาพ โดยนำ VGG-16 มาใช้ในการ Encoder เพื่อทำการแยกแยะระหว่างพืชและดิน 2) Blob extraction ในการทำหาคูณลักษณะภาพที่สนใจจากผลลัพธ์ที่ได้จากการทำการแบ่งส่วนภาพ 3) กระบวนการจำแนกข้อมูล เป็นขั้นตอนในการจำแนกพืชและวัชพืชโดยใช้ VGG-16 ในการพัฒนา

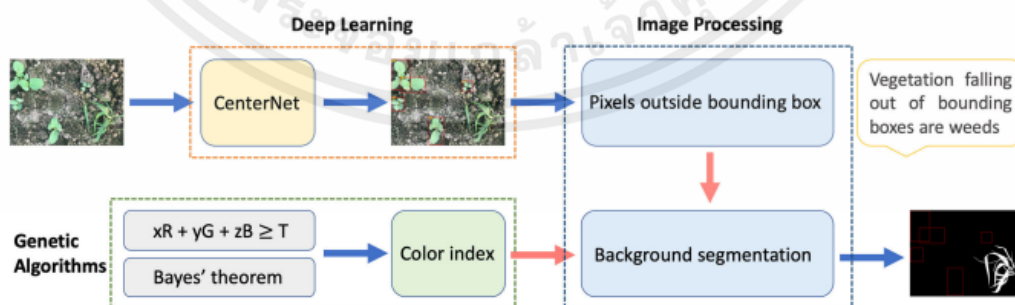


รูปที่ 2.1 แผนภาพกระบวนการจำแนกพืชและวัชพืช

2.1.2 Weed Identification Using Deep Learning and Image Processing in Vegetable

Plantation

งานวิจัยนี้นำเสนอในปี 2021 โดย Xiaojun Jin, Jun Che และ Yong Chen งานวิจัยศึกษาการระบุวัชพืชในสวนผักโดยใช้การเรียนรู้เชิงลึกและการประมวลผลภาพด้วยการดำเนินการ 2 ขั้นตอน ขั้นตอนแรกทำการรวบรวมภาพผักกวางตุ้ง เพื่อเป็นชุดข้อมูล (Dataset) ที่ใช้สำหรับฝึกสอนโมเดล โดยการฝึกสอนโมเดลนั้น ได้ใช้โมเดล CenterNet เพื่อทำการตรวจจับ และทำการวาดขอบเขตของผักกวางตุ้ง จากนั้นในขั้นตอนที่สอง ทำการคำนวณหาค่าดัชนีสีของผักกวางตุ้ง โดยใช้ทฤษฎีของเบย์ (Bayes' Theorem) และทำการประเมินโดยใช้วิธีเชิงพันธุกรรม (Genetic Algorithms) จากนั้นจะทำการแบ่งภาพ (Segmentation) โดยหากพบเจอวัตถุสีเขียวนอกขอบเขตที่วาดไว้ จะทำการระบุว่าวัตถุนั้นคือวัชพืช ซึ่งจากการทดลองขั้นตอนแรกทำการปรับขนาดภาพผักกาดที่นำเข้ามาโมเดลเป็น 512×512 พิกเซล และขนาดแบทช์ถูกกำหนดเป็น 4 หน่วย ใช้ Optimizer ที่มีชื่อว่า Adam และการกำหนดค่าเทรซโซลด์เท่ากับ 0.6 เป็นค่าที่เหมาะสมที่สุด โดยมีค่าความแม่นยำ (Precision) อยู่ที่ร้อยละ 0.956



รูปที่ 2.2 แผนภาพกระบวนการตรวจจับวัชพืช

ที่มา : <https://ieeexplore.ieee.org/document/9317797>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 Management of a Multi-robots System for Industrial Material Handling

งานวิจัยนี้นำเสนอในปี 2019 โดย พงศกร ชาญชัยชูจิต และ พุทธิกร สมิตไมตรี งานวิจัยนี้ใช้ระบบปฏิบัติการหุ่นยนต์ หรือ ROS (Robot Operating System) ในการเพิ่มศักยภาพการใช้หุ่นยนต์ในโรงงานอุตสาหกรรม โดยการลดความซับซ้อนของระบบการควบคุมหุ่นยนต์ที่มีหลายชนิด ลดทรัพยากรที่ใช้ควบคุมหุ่นยนต์ และลดเวลาการทำงานของหุ่น AGV ที่ใช้ในโรงงานอุตสาหกรรม

โดยเหตุผลที่นักวิจัยใช้ ROS ในการแก้ปัญหาเนื่องจากตัวระบบมีความยืดหยุ่นในการควบคุมหุ่นยนต์ที่มีความหลากหลายพร้อมกันได้ สามารถใช้งานได้ฟรี และมีแนวโน้มที่จะนำมาใช้ในโรงงานอุตสาหกรรมมากขึ้นในปัจจุบัน

สิ่งที่นำมาใช้คือการสร้างการควบคุมการเคลื่อนที่ของหุ่นยนต์แต่ละชนิด ที่มีตัวแปรในการควบคุมที่มีการซับซ้อนที่ต่างกัน และใช้อัลกอริทึม Dijkstra ที่ใช้สำหรับหาเส้นทางที่ใกล้ที่สุดมาใช้ควบคุมหุ่นยนต์โรงงานพร้อมกันที่ละหลายตัวเพราะ ROS สามารถทำงานแบบ Multitasking ที่สามารถสั่งการหุ่นยนต์แต่ละตัวอย่างอิสระต่อกันได้

ผลการวิจัยระบุว่าระบบที่ใช้ ROS ควบคุมหุ่นยนต์ควบคุมหุ่นยนต์หลายชนิดพร้อม ๆ กันสามารถทำงานได้เร็วขึ้นเป็นเท่าตัว สามารถรับงานพร้อมกันได้มากกว่า 100 งาน แต่เมื่อเพิ่มจำนวนชนิดหุ่นยนต์ที่ใช้งานด้วยได้จนถึงระดับหนึ่งจึงจะช้าลง เนื่องจากจะต้องประมวลการทำงานของหุ่นยนต์ที่ต่างกัน รวมถึงระยะทางที่หุ่นยนต์หลาย ๆ ตัวเคลื่อนที่ให้ไม่ทับกับหุ่นตัวอื่นนั้นจะใช้เวลาในการคิดมากยิ่งขึ้น

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 ความหมายของวัชพืชและการกำจัดวัชพืช

วัชพืช หมายถึง พืชที่ขึ้นเองตามธรรมชาติ ส่งผลต่อระบบการผลิตทางการเกษตรในทางลบ มักสร้างความเสียหายโดยแทรกแซงการเจริญเติบโตของพืชที่ปลูก โดยการกำจัดหรือการควบคุมเพื่อลดปริมาณวัชพืชหรือให้วัชพืชหมดไป การกำจัดวัชพืชมีหลายวิธีดังนี้

1) การกำจัดวัชพืชด้วยวิธีเขตกรรม เป็นวิธีการสมัยโบราณ ซึ่งยังมีการใช้มาถึงปัจจุบัน เนื่องจากเป็นวิธีที่มีประสิทธิภาพ วิธีนี้มีหลายอย่าง เช่น การถอน การใช้เครื่องมือทุ่นแรงในการเกษตร การใช้เครื่องตัดหญ้าหรือกรรไกรตัดหญ้า การใช้ความร้อนหรือเผา เป็นต้น

2) การกำจัดวัชพืชโดยสิ่งมีชีวิต เป็นวิธีที่ได้จากประสบการณ์จากธรรมชาติ หรือก็คือวัชพืชจะถูกกำจัดโดยศัตรูธรรมชาติ ได้แก่ สัตว์ พืช แมลงและเชื้อโรค แต่วิธีการนี้ไม่สามารถกำจัดวัชพืชให้หมดได้ ทำได้เพียงลดปริมาณวัชพืช เช่น อเมริกาใช้พะยูนแมนนาตี (Manatee) ซึ่งเป็นสัตว์น้ำชนิดหนึ่งในการกำจัดผักตบชวาและวัชพืชน้ำอื่น ๆ สัตว์ชนิดนี้กินพืชเป็นอาหาร เป็นต้น

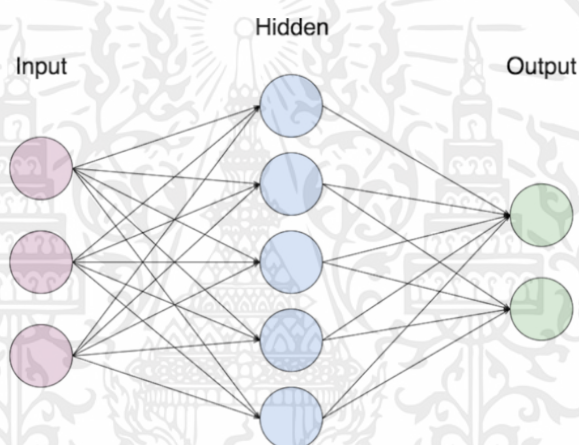
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การกำจัดโดยใช้สารกำจัดวัชพืช คือ การใช้สารเคมีในการฆ่าทำลายหรือยับยั้งการเจริญเติบโตของวัชพืช เป็นวิธีการที่นิยมในปัจจุบัน เนื่องจากเป็นวิธีที่ได้ผลเร็วกว่าการใช้แรงงานคนและสิ้นเปลืองน้อยกว่า แต่การใช้สารเคมีมักมีผลกระทบต่อผู้ใช้ และสิ่งแวดล้อม รวมถึงผู้บริโภค จึงควรมีการควบคุมและระมัดระวังในการใช้

2.2.2 เทคนิคการจำแนกข้อมูล

2.2.2.1 โครงข่ายประสาทเทียม (Neural Network: NN)

โครงข่ายประสาทเทียมเป็นหนึ่งในเทคนิคการจำแนกข้อมูล มีแนวคิดเริ่มต้นจากการจำลองพฤติกรรมเซลล์สมองของมนุษย์ ทำให้สามารถเรียนรู้จากข้อมูลจำนวนมากได้ แม้ว่าโครงข่ายประสาทจะมีเพียงชั้นเดียว ก็ยังคงสามารถคาดการณ์ได้ แต่หากมีชั้นที่ซ่อนอยู่ (Hidden Layer) เพิ่มเติม ก็จะสามารถช่วยในการเพิ่มประสิทธิภาพและเพิ่มความแม่นยำในการคาดการณ์ได้

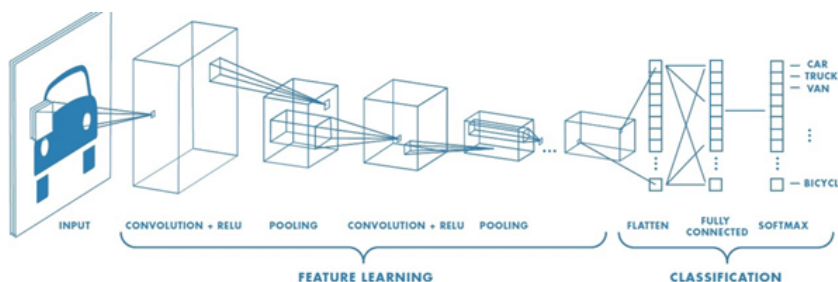


รูปที่ 2.3 โครงสร้างของ Neural Networks

ที่มา : <https://towardsdatascience.com/classical-neural-network-what-really-are-nodes-and-layers-ec51c6122e09>

2.2.2.2 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Networks: CNN)

โครงข่ายประสาทแบบคอนโวลูชันเป็นโครงข่ายประสาทเทียมหลายชั้นที่มีสถาปัตยกรรมที่มีความเฉพาะตัว ซึ่งถูกออกแบบมาเพื่อเพิ่มความสามารถในการแยกคุณลักษณะที่มีความซับซ้อนมากขึ้นของข้อมูลในแต่ละชั้น โดยโครงข่ายประสาทแบบคอนโวลูชันเหมาะสมสำหรับงานประเภทการรับรู้ (Perceptual Tasks) เป็นอย่างมาก โครงข่ายประสาทแบบคอนโวลูชันนั้นส่วนมากถูกใช้เมื่อข้อมูลที่ไม่มีเป็นลักษณะเชิงโครงสร้าง (Unstructured Data) เช่น รูปภาพ บทความข่าว ข้อมูลสื่อสังคมออนไลน์ เป็นต้น



รูปที่ 2.4 Convolutional Neural Networks

ที่มา : <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

2.2.3 เทคนิคการประมวลผลภาพ

การประมวลผลภาพ (Image Processing) เป็นกระบวนการจัดการและวิเคราะห์รูปภาพให้เป็นข้อมูลในรูปแบบดิจิทัล ด้วยการประมวลผลภาพด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่ต้องการทั้งในเชิงคุณภาพและปริมาณ (ขนาดและรูปร่าง) สำหรับโครงการนี้ เทคนิคการประมวลผลภาพที่นำมาใช้ มี ดังนี้

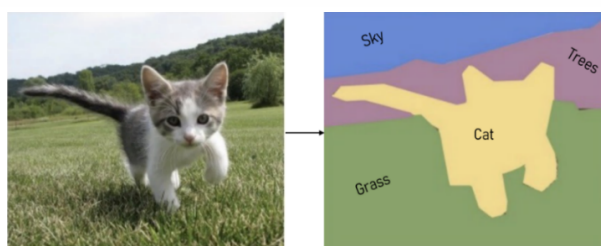
2.2.3.1 การแบ่งส่วนภาพ (Image Segmentation)

คือ การแบ่งภาพออกเป็นส่วนๆ เช่น การแบ่งแยกวัตถุที่สนใจ ออกจากพื้นหลัง เป็นต้น โดยสามารถแบ่งวิธีการแบ่งส่วนภาพ (Image Segmentation) ออกเป็น 2 วิธีการหลัก ได้แก่

1) Complete Segmentation เป็นการแบ่งภาพออกเป็นส่วน ๆ ตามวัตถุบนโลกจริง หรือตามสิ่งที่มองเห็น เช่น เรือ สะพาน ภูเขา ท้องฟ้า เป็นต้น โดยไม่มีส่วนที่ซ้อนทับกัน ต้องใช้ข้อมูลที่เฉพาะเจาะจงในการแบ่งแยกวัตถุ

2) Partial Segmentation เป็นการแบ่งภาพออกเป็นส่วนๆ ตามคุณสมบัติที่ผู้พัฒนา กำหนดขึ้นมาเอง เช่น สี หรือความสว่าง เป็นต้น

โดยเทคนิคที่ใช้ในโครงการนี้ จะเป็นการทำการแบ่งส่วนภาพ (Image Segmentation) แบบ Partial Segmentation โดยการแบ่งภาพตามสี (Color detection) คือ การแบ่งแยกวัตถุที่เรานำสนใจออกจากพื้นหลังโดยเลือกเฉพาะบริเวณที่มีสีเดียวกับวัตถุ



รูปที่ 2.5 การแบ่งภาพออกเป็นส่วน ๆ ตามวัตถุ (Image Segmentation)

ที่มา : <https://www.mygreatlearning.com/blog/fcn-fully-convolutional-network-semantic-segmentation/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 หุ่นยนต์อุตสาหกรรม

หุ่นยนต์อุตสาหกรรมที่นำเข้ามาใช้ในการเพิ่มผลผลิตหรือควบคุมคุณภาพและมาตรฐานนั้นมีความหลากหลายตามหน้าที่การใช้งานหรือความซับซ้อนสำหรับงานนั้น ๆ จึงจำเป็นต้องเข้าใจหน้าที่ วิธีการทำงานและความแตกต่างของหุ่นยนต์แต่ละประเภท โดยทั่วไปสามารถแบ่งประเภทของหุ่นยนต์ตามการใช้งานได้ 5 ประเภทดังต่อไปนี้

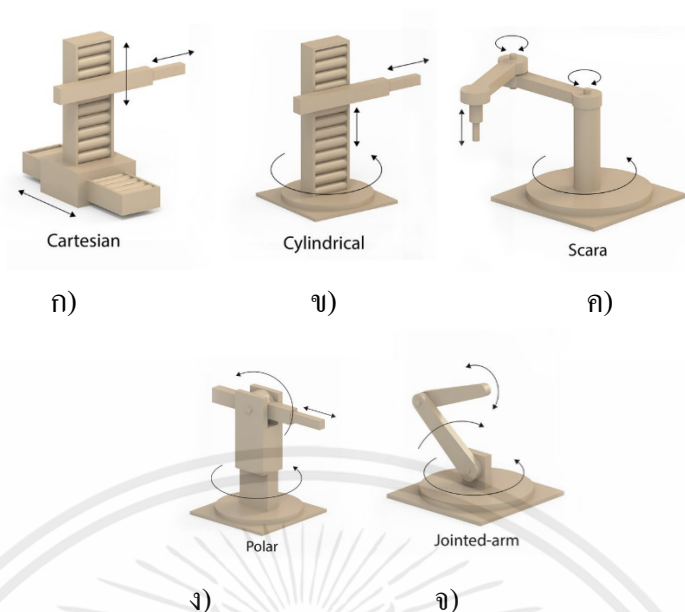
2.2.4.1 Cartesian Robot เป็นหุ่นยนต์ที่ทั้ง 3 แกนเคลื่อนที่เป็นเส้นตรง แสดงในรูปที่ 2.7 ก) โครงสร้างมีความแข็งแรง รับน้ำหนักได้สูงที่สุดเทียบกับหุ่นยนต์ประเภทอื่น นิยมใช้ในการหยิบจับเพื่อการประกอบชิ้นส่วนอิเล็กทรอนิกส์หรือยานยนต์

2.2.4.2 Cylindrical Robot จะมีฐานที่ใช้การหมุนแทนการเคลื่อนเป็นเส้นตรง แสดงในรูปที่ 2.7 ข) ใช้ในการหยิบขนย้าย และงานที่มีความซับซ้อนต่ำ

2.2.4.3 SCARA (Selective Compliance Assembly Robot Arm) เป็นหุ่นยนต์ที่มีการเคลื่อนที่แบบหมุน 2 จุด แสดงในรูปที่ 2.7 ค) มีความแม่นยำสูง สามารถเคลื่อนที่ได้รวดเร็วในแนวราบ ยกชิ้นส่วนที่มีน้ำหนักมากไม่ได้ นิยมใช้กับงานประกอบชิ้นส่วนอิเล็กทรอนิกส์

2.2.4.4 Polar Robot (หรือ Spherical Robot) เป็นหุ่นยนต์ที่มีการเคลื่อนที่แบบหมุน 2 จุด แสดงในรูปที่ 2.7 ง) มีส่วนมือจับที่สามารถยืดหดได้ เหมาะกับงานที่เคลื่อนไหวแนวตั้งไม่มาก ใช้ในงานประเภทหยิบจับรวมถึงงานเชื่อมรูปแบบต่าง ๆ

2.2.4.5 Jointed Arm เป็นหุ่นยนต์ที่เคลื่อนที่แบบหมุนทุกแกนคล้ายแขนคน แสดงในรูปที่ 2.7 จ) สามารถใช้งานได้หลายประเภทเพราะสามารถเข้าถึงตำแหน่งต่าง ๆ ได้ดี เช่นงานเชื่อมงานตัด รวมถึงงานที่มีการเคลื่อนที่ยาก ๆ เช่น งานทากาว หรืองานพ่นสี



รูปที่ 2.6 ลักษณะหุ่นยนต์แต่ละประเภท

ก) Cartesian Robot

ข) Cylindrical Robot

ค) SCARA

ง) Polar Robot

จ) Jointed Arm

ที่มา : <https://www.sumipol.com/knowledge/types-of-industrial-robot/>

2.2.5 ระบบปฏิบัติการหุ่นยนต์

ระบบปฏิบัติการหุ่นยนต์ (Robot Operating System: ROS) เป็นระบบปฏิบัติการโอเพ่นซอร์ส (Open source) ที่นิยมนำมาใช้แก้ปัญหาการพัฒนาหุ่นยนต์ที่มีความซับซ้อน ให้สามารถแยกการทำงานออกเป็น ส่วน ๆ แล้วนำมารวมกันได้ โดยจะแบ่งองค์ประกอบในการทำงานออกเป็น โหนด (Node) แต่ละโหนดสามารถทำงานอิสระจากกันทำให้สามารถทำงานที่หลากหลายได้ในเวลาเดียวกัน (Multi-tasking) และสื่อสารระหว่างโหนดแบบ Peer-to-Peer ซึ่งมีหลายวิธี ได้แก่ Topics Services และ Actions โดยมีลักษณะดังนี้

1) Topics (หัวข้อ) เป็นการสื่อสารแบบผู้ส่ง (Publisher) กับผู้รับ (Subscriber) เป็นลักษณะการสื่อสารทางเดียว (One-way-Communication) โดยการส่งข้อมูล (Message) ขึ้นไปบนหัวข้อนั้น ๆ แล้วผู้ที่ตกลงรับหัวข้อนั้น ๆ ใดก็จะได้รับข้อมูล เป็นการส่งทั้งแบบ One-to-One One-to-Many Many-to-One และ Many-to-Many

2) Services (บริการ) เป็นการสื่อสารแบบสองทาง (Two-way-Communication) เรียกว่า Call and Response โดยจะมีสองฝั่งคือ ลูกข่าย (Client) กับแม่ข่าย (Server) โดยฝั่งลูกข่ายจะทำการร้องขอข้อมูล (Request) จากฝั่งแม่ข่าย ซึ่งจะส่งเป็นผลตอบรับ (Response) กลับมาให้กับลูกข่ายที่ทำการขอข้อมูลมาเท่านั้น โดยใน 1 บริการจะมีเพียง 1 แม่ข่ายเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Actions (แอกชัน) เป็นการสื่อสารที่ออกแบบสำหรับใช้ส่งข้อมูลต่อเนื่องในช่วงเวลาหนึ่งหรือเงื่อนไขใด ๆ (Long run) ประกอบไปด้วย เป้าหมาย (Goal) ผลตอบรับ (Feedback) และผลลัพธ์ (Result) โดยฝั่งลูกข่ายจะทำการส่งเป้าหมายไปที่ลูกข่าย แล้วจึงส่งคำขอผลลัพธ์ที่ต้องการ (Result Request) ไปยังแม่ข่ายเพื่อรับผลรับในหัวข้อนั้น ๆ มาเรื่อย ๆ จนกว่าจะบรรลุเป้าหมายที่ตั้งไป และส่งผลตอบรับตอบกลับ (Result Response) กลับมายังลูกข่าย



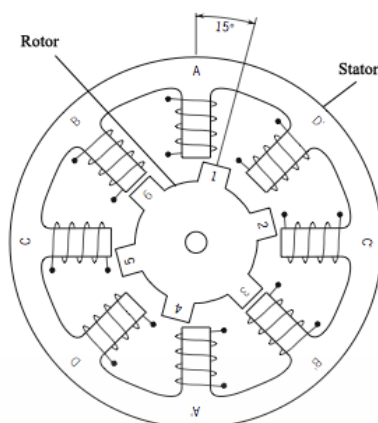
รูปที่ 2.7 ระบบปฏิบัติการหุ่นยนต์ (Robot Operating System)

ที่มา : https://en.wikipedia.org/wiki/Robot_Operating_System

2.2.6 การทำงานของสเต็ปมอเตอร์

สเต็ปมอเตอร์นั้นต่างจากมอเตอร์แบบธรรมดาทั่วไปตรงที่มอเตอร์ทั่วไปนั้นจะใช้กับงานที่ต้องหมุนต่อเนื่อง ไม่เน้นความละเอียดหรือความแม่นยำมากนัก ต่างจากสเต็ปมอเตอร์ที่เหมาะสมกับงานที่ต้องใช้ความละเอียด แม่นยำในการทำงาน โดยเกิดจากการที่มอเตอร์ทำการแบ่งการหมุนเต็มรอบเป็นการหมุนทีละส่วนเล็ก ๆ เท่ากัน ทำให้มอเตอร์สามารถหมุนด้วยองศาที่แม่นยำ แต่ไม่เหมาะกับการนำมาใช้ในการหมุนที่ต่อเนื่องเป็นเวลานานเหมือนมอเตอร์ทั่วไป

ส่วนประกอบที่สำคัญในการเคลื่อนที่ของสเต็ปมอเตอร์ได้แก่ Stator และ Rotor มีลักษณะดังภาพที่ 2.8 โดย Stator เป็นขั้วแม่เหล็กที่ติดอยู่กับตัวถังมอเตอร์ ซึ่งในสเต็ปมอเตอร์จะมีซี่ฟันที่พันด้วยขดลวดเหนี่ยวนำให้เกิดสนามแม่เหล็กขึ้น และ Rotor เป็นก้อนแม่เหล็กถาวรที่ติดอยู่กับแกนหมุน มีลักษณะคล้ายเฟืองเป็นฟัน ทำหน้าที่คู่กับซี่ฟันของ Stator ตามตำแหน่งต่าง ๆ ที่จ่ายไฟ

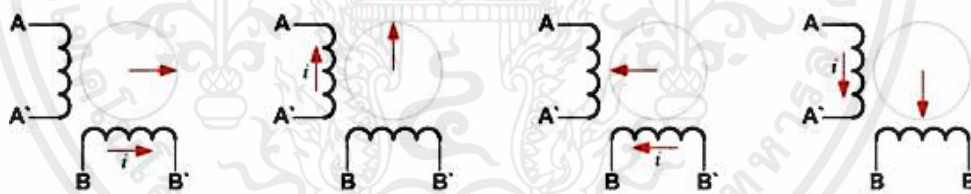


รูปที่ 2.8 ลักษณะของ Stator และ Rotor

ที่มา : <https://circuitdigest.com/tutorial/what-is-stepper-motor-and-how-it-works>

โดยในการใช้งานจริงคือการจ่ายไฟฟ้าเข้าไปในขดลวดเพื่อเปลี่ยนแปลงสนามแม่เหล็ก ให้ Rotor ติดกับขดลวดทำให้มอเตอร์เคลื่อนที่ 1 ขั้น (step) ซึ่งการจ่ายไฟนั้นสามารถแบ่งการทำงานได้ออกเป็น 2 โหมดได้แก่

1. โหมด Full Step เป็นการจ่ายไฟไปยังสองขด ขดละ 1 ขดลวดพร้อมกัน ทำให้ Rotor ถูกดูดไปในทิศทางที่ต้องการ วิธีนี้ทำแรงบิดของมอเตอร์สูงจากการจ่ายไฟให้ขดลวดดูด Rotor พร้อมกัน โดยแสดงขั้นตอนการหมุนดังภาพด้านล่าง

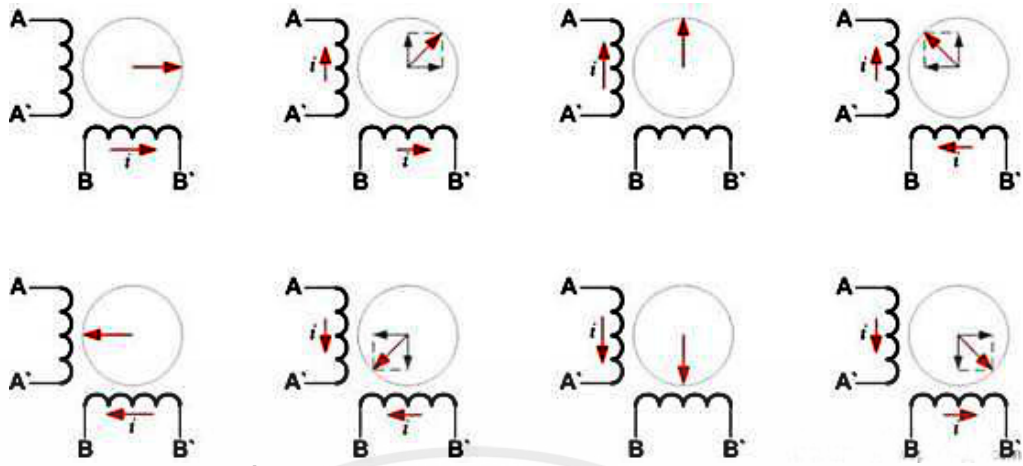


รูปที่ 2.9 ขั้นตอนการทำงานของมอเตอร์ในโหมด Full step

ที่มา : <https://www.automate.org/case-studies/what-is-the-difference-between-full-stepping-the-half-stepping-and-the-micro-drive>

2. โหมด Half Step เป็นการใช้แม่เหล็กหนึ่งขั้วนำขั้วก่อนหน้า ก่อนที่จะปิดขั้วเดิมทีละขั้น ดังเช่นภาพที่ 2.10 ทำให้ Rotor หมุนทีละครึ่งสเต็ป ส่งผลให้ได้ความละเอียดในการหมุนเพิ่มขึ้นจากการจ่ายไฟแบบแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ขั้นตอนการทำงานของมอเตอร์ไนโหมด Half step

ที่มา : <https://www.automate.org/case-studies/what-is-the-difference-between-full-stepping-the-half-stepping-and-the-micro-drive>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการพัฒนา

3.1 การวิเคราะห์ความต้องการ

หุ่นยนต์กำจัดวัชพืชตรวจจับด้วยการประมวลผลภาพนั้น มีความสามารถกำจัดวัชพืชในแปลงปลูกได้อย่างแม่นยำ ดังนั้นจึงสามารถแบ่งความต้องการตามการทำงานของระบบ ได้ดังนี้

- 1) การเคลื่อนที่ไปยังตำแหน่งเก็บภาพ
- 2) การเก็บภาพจากแปลงปลูกพืชด้วยกล้อง
- 3) การจำแนกระหว่างพืชและวัชพืช และระบุตำแหน่งวัชพืช
- 4) กำจัดวัชพืชด้วยหุ่นยนต์

3.2 การออกแบบ

ในการออกแบบหุ่นยนต์กำจัดวัชพืชนั้นสามารถแบ่งการออกแบบระบบได้เป็น 3 ส่วน ได้แก่ 1) โครงสร้างของหุ่นยนต์ 2) ระบบควบคุมหุ่นยนต์ และ 3) โมดูลสำหรับการจำแนกระหว่างพืชและวัชพืช มีรายละเอียดดังนี้

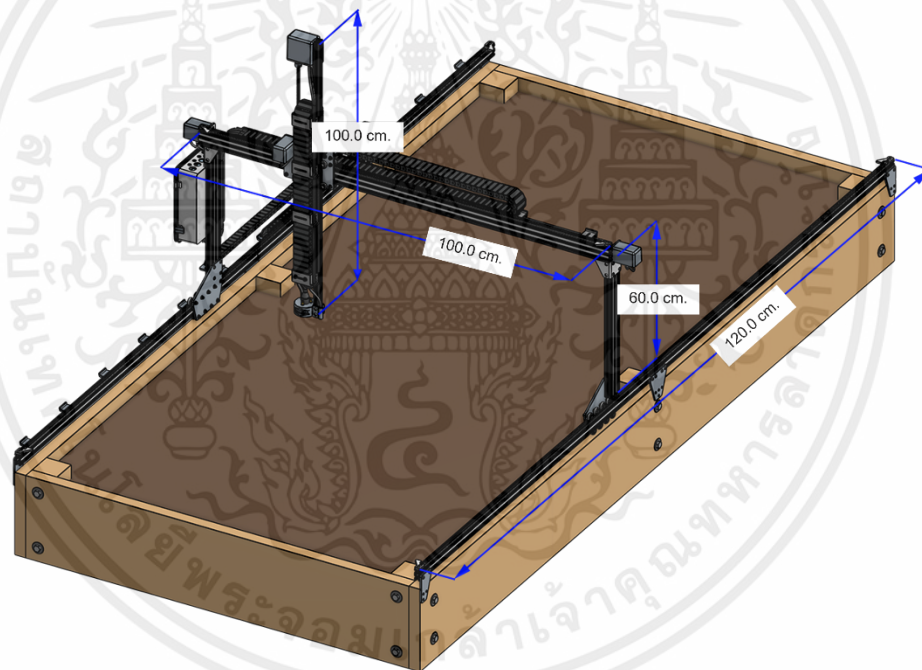
3.2.1 การออกแบบโครงสร้างหุ่นยนต์

หุ่นยนต์ที่นำมาใช้ในโครงการนี้นำต้นแบบมาจากหุ่นยนต์ FarmBot Genesis ซึ่งเป็นหุ่นยนต์เคลื่อนที่ 3 แกนดังรูปที่ 3.1 โดยนำแนวคิดในการทำโครงด้วยอลูมิเนียมโปรไฟล์ ใช้ล้อเลื่อน V-Slot สายพาน ไทม์มิ่ง และลีดสกรูในการเคลื่อนที่แนวและแนวตั้งมาประยุกต์ใช้ โดยปรับขนาดหุ่นยนต์ ใช้ลิมิตสวิตช์ในการตรวจจับการชนเพื่อจำกัดระยะการเคลื่อนที่ ใช้กล้องเว็บแคมและเลเซอร์กำจัดวัชพืชติดที่แกนตั้ง เพื่อให้ได้หุ่นยนต์ที่ขนาดกว้าง 100 เซนติเมตร ยาว 120 เซนติเมตร และสูง 150 เซนติเมตรดังรูปที่ 3.2



รูปที่ 3.1 หุ่นยนต์ FarmBot Genesis

ที่มา : <https://farm.bot/pages/genesis>

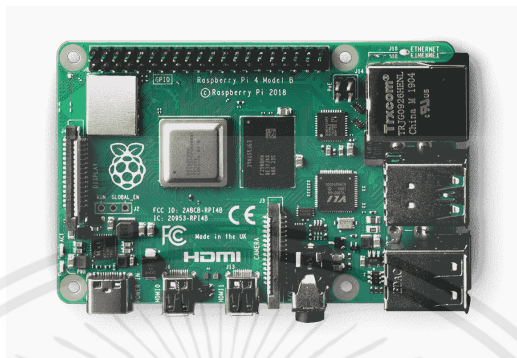


รูปที่ 3.2 การออกแบบโครงสร้างและขนาดของหุ่นยนต์

เพื่อให้ได้ผลลัพธ์ตามการออกแบบในรูปที่ 3.2 นั้นทำให้เราสามารถเลือกอุปกรณ์ที่ต้องการนำมาใช้เป็นรายการดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

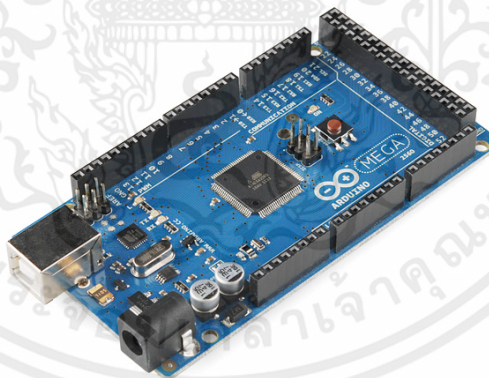
1. Raspberry Pi 4 Model B 4GB ทำหน้าที่เป็นหน่วยประมวลผลหลัก ใช้ในการควบคุมระบบการทำงานของหุ่นยนต์ ทั้งการติดต่อกับผู้ใช้งาน การสั่งการถ่ายภาพ ควบคุมการทำงานของเลเซอร์ รวมถึงการสื่อสารกับบอร์ด Arduino Mega 2560 เพื่อสั่งการเคลื่อนที่สเต็ปมอเตอร์



รูปที่ 3.3 Raspberry Pi 4 Model B

ที่มา : <https://drakemall.com/products/raspberry-pi-4-model-b-4gb>

2. Arduino Mega 2560 เป็นหน่วยประมวลผลขนาดเล็ก สามารถเชื่อมต่อกับส่วนขยายต่าง ๆ ได้หลายแบบ ในที่นี้นำมาใช้ควบคู่กับส่วนขยาย RAMPS เพื่อใช้ในการขับเคลื่อนมอเตอร์ เชื่อมต่อสื่อสารกับ Raspberry Pi ผ่านสาย USB Serial

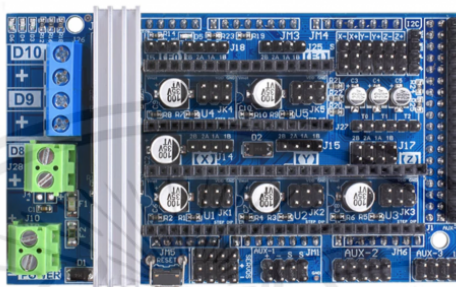


รูปที่ 3.4 Arduino Mega 2560

ที่มา : <https://www.arduitronics.com/product/5/arduino-mega2560-rev3-free-usb-cable>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. RAMPS (RepRap Arduino Mega Pololu Shield) รุ่น 1.6 เป็นส่วนขยายสำหรับบอร์ด Arduino Mega 2560 ใช้ขับสเต็ปมอเตอร์ที่ได้รับความนิยมในการทำเครื่องพิมพ์สามมิติ สามารถขับสเต็ปมอเตอร์พร้อมกันได้สูงสุด 6 ตัว พร้อมยังมีพอร์ตสำหรับใช้งานอื่น ๆ เช่น Endstop Switch max min ทั้ง 3 แกน พัดลม ตัวควบคุมอุณหภูมิ และอื่น ๆ รับไฟ 12V จากพาวเวอร์ซัพพลายในการใช้งานร่วมกับบอร์ด Arduino Mega 2560



รูปที่ 3.5 RAMPS 1.6

ที่มา : <https://uge-one.com/3d-printer-accessories-ramps-1-6-r6-control-board.html>

4. Nema 17 เป็นสเต็ปมอเตอร์ความละเอียด 200 ขั้นต่อรอบ (Steps per revolution) มีความละเอียดที่ 1.8 องศา สามารถหมุนด้วยความเร็วสูงสุดที่ 600 รอบต่อนาที ใช้ในการเคลื่อนที่หุ่นยนต์โดยการต่อเข้ากับเพลาเพื่อขับสายพานหรือลีดสกรูให้หมุนด้วยองศาที่แม่นยำ ต่อเข้ากับบอร์ด

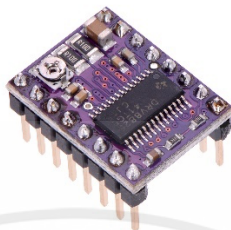


รูปที่ 3.6 NEMA 17 Stepper Motor

ที่มา : <https://www.hwlibre.com/th/nema-17>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. DRV8825 บอร์ดขับเคลื่อนมอเตอร์ให้กับสเต็ปมอเตอร์ ติดตั้งบนบอร์ด RAMPS 1.6 สามารถทำไมโครสเต็ปเพื่อเพิ่มความละเอียดให้กับสเต็ปมอเตอร์ได้ตั้งแต่ 1/2 เท่า (0.9 องศา) จนถึง 1/32 เท่า (0.05625 องศา) ทำให้หุ่นยนต์สามารถเคลื่อนที่ในองศาที่แม่นยำและมีแรงขับที่สูงขึ้นได้



รูปที่ 3.7 DRV8825

ที่มา : <https://www.pololu.com/product/2982>

6. EGA Webcam USB Camera ใช้ในการถ่ายภาพ มีความละเอียดสูงสุด 1920 x 1080 พิกเซล ความเร็วในการถ่ายภาพอยู่ที่ 30 เฟรมต่อวินาที โฟกัสแบบอัตโนมัติ มีองศารับภาพ 70 องศาในแนวทแยง มีช่องใส่ไมโครโฟน M5 ทำการยึดกับบอลูมิเนียมโปรไฟล์ได้ ใช้สำหรับถ่ายภาพ



รูปที่ 3.8 EGA Webcam USB Camera

ที่มา : <https://www.mercular.com/ega-type-w1-webcam>

7. ลิมิตสวิตช์ ใช้ตรวจสอบการชนของแกนต่าง ๆ เพื่อปรับเทียบตำแหน่งของหุ่นยนต์ ใช้หาตำแหน่งเริ่มต้นหรือสิ้นสุดของหุ่นยนต์ เพื่อให้มั่นใจว่าหุ่นยนต์ยังอยู่ในระยะการทำงานที่ถูกต้อง



รูปที่ 3.9 Endstop Switch Module

ที่มา : <https://electropeak.com/mech-endstop-v1-2>

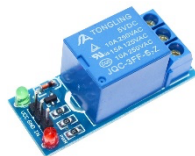
8. โมดูลเลเซอร์ LD3070LA เป็นโมดูลเลเซอร์ที่ใช้ในงานเลเซอร์คัท มีความยาวคลื่น 450nm มีบอร์ดควบคุมเลเซอร์ ใช้สัญญาณ PWM ในการทำงาน โดยจ่ายไฟผ่านรีเลย์ที่เชื่อมกับ Raspberry Pi ใช้ในการเผาทำลายวัชพืช รับไฟ 12V และมีพัดลมระบายอากาศในตัว



รูปที่ 3.10 Laser Module

ที่มา : <https://produto.mercadolivre.com.br>

9. โมดูลรีเลย์ 5V 1 ช่อง ใช้ในการควบคุมการจ่ายไฟให้กับอุปกรณ์ไฟฟ้าแรงดันสูง ในที่นี้ นำมาใช้สำหรับจ่ายไฟให้โมดูลเลเซอร์ที่ใช้กำจัดวัชพืช สังกานด้วย GPIO ของ Raspberry Pi



รูปที่ 3.11 Relay 5V 1 Channel

ที่มา : <https://www.spmicrotech.com/>

10. Power Supply 12V/30A ทำหน้าที่แปลงไฟ 220V กระแสสลับให้เป็น 12V กระแสตรง เพื่อจ่ายกระแสไฟให้กับบอร์ด Arduino Mega 2560 และ โมดูลเลเซอร์

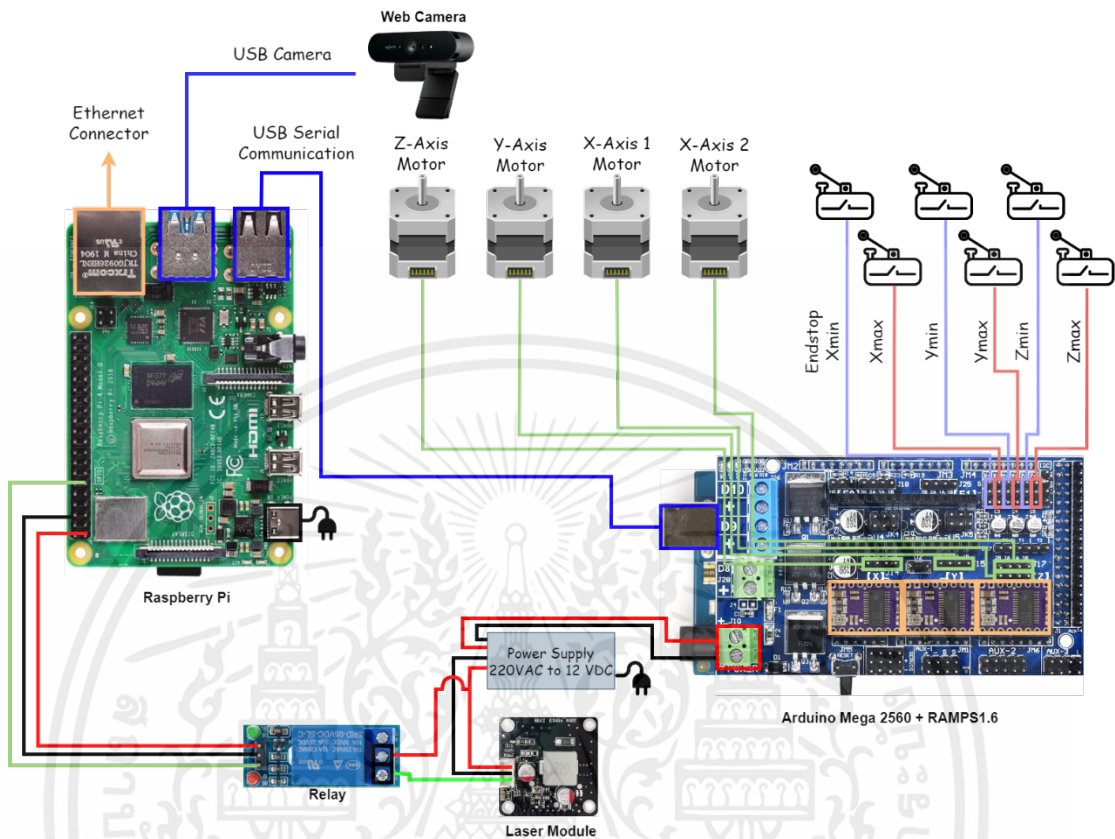


รูปที่ 3.12 Power Supply

ที่มา : https://www.m-v-com.com/products_detail/view/7574457

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจากอุปกรณ์ดังรายการข้างต้นทำให้ได้ออกแบบแผนภาพในการเชื่อมต่ออุปกรณ์ได้ดังนี้



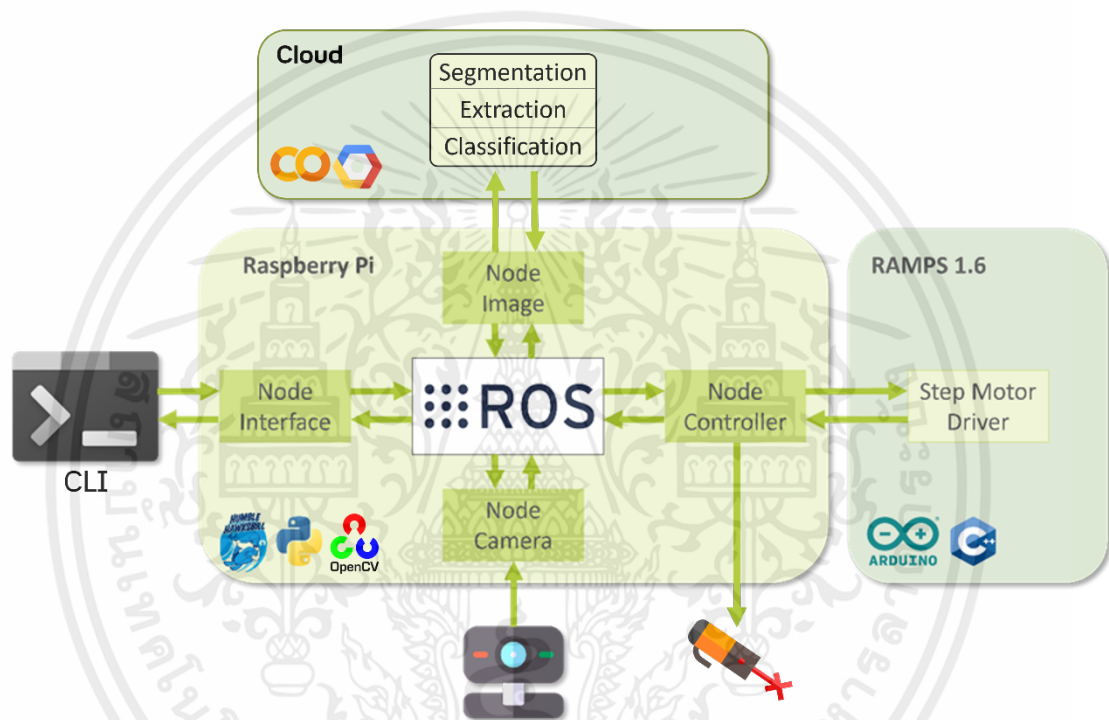
รูปที่ 3.13 แผนภาพการเชื่อมต่ออุปกรณ์

โดยมีชิ้นส่วนอื่น ๆ ที่ใช้ในการเชื่อมต่อส่วนต่าง ๆ ของหุ่นยนต์เข้าด้วยกันได้แก่ลুমินีม โปรไฟล์ 20x20, 20x40, 20x60 รวมถึงชิ้นส่วนที่หาซื้อไม่ได้ ก็ได้จัดทำด้วยกันสองวิธีคือ หากเป็นชิ้นส่วนที่รับน้ำหนักไม่มากจะผลิตด้วยเครื่องพิมพ์สามมิติ และส่วนที่มีผลกับการเคลื่อนที่จะใช้การกลึงและมิลลิ่งลุมินีมซึ่งสั่งทำขึ้นมาแทน โดยสามารถดูชิ้นส่วนและราคาทั้งหมดได้ที่ภาคผนวก ข และ ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การออกแบบระบบควบคุมหุ่นยนต์

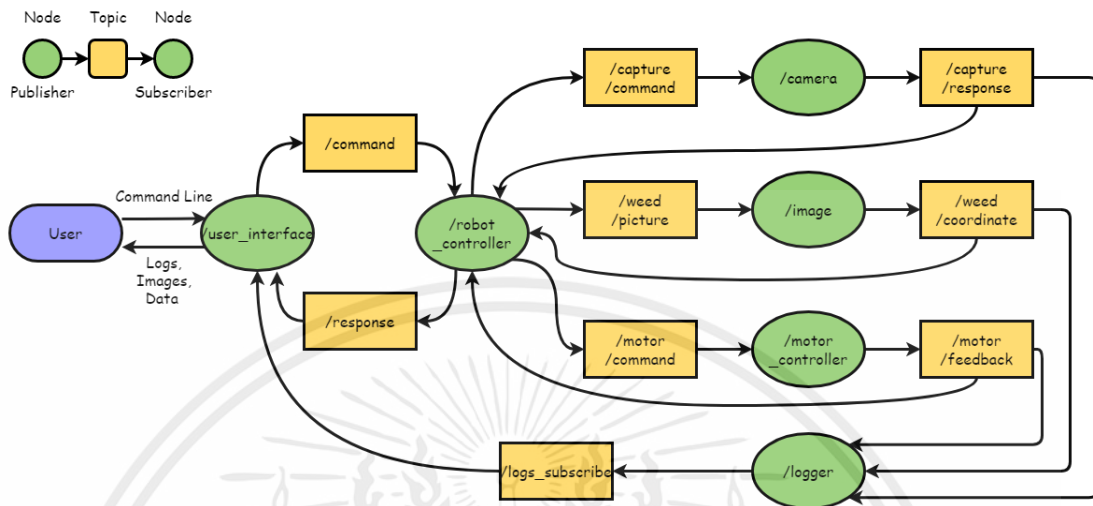
การทำงานของหุ่นยนต์ที่ควบคุมด้วย Raspberry Pi จะใช้ระบบปฏิบัติการ ROS ในการควบคุม โดยเวอร์ชันที่ใช้คือ ROS 2 Humble Hawksbill Release 1 (2022-11-23) ซึ่งติดตั้งบน Raspberry Pi 4 ที่ใช้ระบบปฏิบัติการ Ubuntu 22.04.1 Desktop (Jammy Jellyfish) โดยมีการออกแบบส่วนการเชื่อมต่อการทำงานตั้งแต่การรับคำสั่งจากผู้ใช้งานจนถึงการสื่อสารระหว่าง Raspberry Pi กับ กล้องเว็บแคม เลเซอร์ ตัวขับเคลื่อนมอเตอร์ และ โมดูลการจำแนกพืชและวัชพืชบนคลาวด์โดยมี ROS เป็นตัวเชื่อมต่อระบบการทำงานทั้งหมดได้เป็นแผนภาพดังรูปที่ 3.14 นี้



รูปที่ 3.14 การออกแบบการเชื่อมต่อระบบควบคุมหุ่นยนต์

อธิบายจากรูปด้านบนเพิ่มเติมคือ ในแนวคิดการพัฒนาระบบนี้เราจะใช้ ROS เป็นตัวกลางสร้างโหนดเพื่อจัดการทำงานกับส่วนต่าง ๆ แยกกันโดยแบ่งออกเป็นส่วนตัวติดต่อกับผู้ใช้งาน ส่วนควบคุมถ่ายภาพ ส่วนรับส่งภาพไปยังคลาวด์เพื่อหาตำแหน่งวัชพืช และส่วนควบคุมมอเตอร์และเลเซอร์ ซึ่งซอฟต์แวร์ไลบรารีที่นำมาใช้ในการพัฒนาบน ROS ได้แก่ Python และ OpenCV บนส่วนของการควบคุมมอเตอร์ได้แก่ Arduino และ C++ และในส่วนการแยกพืชและวัชพืชจะอธิบายรายละเอียดเพิ่มเติมในหัวข้อที่ 3.2.3

จากการออกแบบแผนภาพในรูปที่ 3.14 ทำให้สามารถออกแบบโหนดและหัวข้อการสื่อสารได้ดังนี้



รูปที่ 3.15 การออกแบบการสื่อสารระหว่าง ROS Node และ Topic

จากแผนภาพด้านบนมีทั้งหมด 6 โหนด โดยมีรายละเอียดชื่อโหนดและหน้าที่การทำงานดังนี้

- 1) User_Interface ใช้ในการรับค่าคำสั่งจากผู้ใช้ในการทำงานของหุ่นยนต์ และส่งผลการทำงานรวมถึง log ต่าง ๆ ไปแสดงผลผ่าน Command line ให้กับผู้ใช้งาน
- 2) Robot_Controller ใช้ในการแปลงคำสั่งจากผู้ใช้ออกเป็นขั้นตอนควบคุมการทำงานบนโหนดต่าง ๆ พร้อมทั้งส่งข้อมูลผลลัพธ์ที่ผู้ใช้ต้องการกลับไป
- 3) Logger ใช้เก็บข้อมูลการทำงานต่าง ๆ ที่เกิดขึ้นจากในระบบ โดยแยกสถานะที่เกิดขึ้น มีลักษณะเป็น [ตัวย่อ][โหนด]{รายละเอียด} โดยแบ่งออกเป็น 3 หัวข้อดังตารางต่อไปนี้

ตารางที่ 3.1 สถานะการเก็บข้อมูลบน Logger

สถานะ	ตัวย่อ	คำอธิบาย	ตัวอย่าง
Information	I	แสดงข้อความการทำงานทั่วไปของหุ่นยนต์	[I][robot_contoller] Send findHome command to /motor/command
Warning	W	แสดงข้อความแจ้งเตือนสิ่งนี้อาจทำให้การทำงานของหุ่นยนต์ผิดพลาดได้	[W][robot_controller] Robot hasn't done calibrated for too long.
Error	E	แสดงข้อความแจ้งเตือนข้อผิดพลาดของหุ่นยนต์ที่เกิดขึ้น	[E][camera] Can't open camera by index

- 4) โหนด Camera ใช้ในการเรียกเก็บภาพจาก USB Camera ผ่านไลบรารี OpenCV ซึ่งภาพที่ได้จะมีขนาด 640 x 480 พิกเซล แล้วทำการส่งตำแหน่งของภาพที่เก็บได้กลับไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) โหนด Image ใช้ในการส่งตำแหน่งของภาพไปยังส่วนระบบประมวลผลภาพเพื่อตรวจหาวัตถุ แล้วส่งค่าตำแหน่งของวัตถุไปที่โหนด robot_controller

6) โหนด Motor_Controller ใช้ในการสร้างชุดข้อมูลสำหรับสื่อสารกับบอร์ด Arduino Mega 2560 เพื่อควบคุม RAMPS 1.6 ที่เป็นตัวขับเคลื่อนให้หุ่นยนต์มีการเคลื่อนที่ไปยังตำแหน่งที่ต้องการ มีการรับส่งค่าผ่าน UART ด้วยโปรโตคอลที่กำหนดขึ้นเพื่อลดความผิดพลาดที่เกิดจากการสื่อสารระหว่างอุปกรณ์ได้แก่ Tx Data Package (Transmission) และ Rx Data Package (Receiving) ดังนี้

ตารางที่ 3.2 Protocol การส่งข้อมูลจากโหนด Controller ไปยัง Arduino Mega 2560

Tx Data	Start	Function Type	Payload Length	Data	Stop
Size (Byte)	1	1	1	0-2	1
Data	0x7E	0xF1 = Get Position	0x00	-	0x7F
		0xF2 = Move	0x02	[Axis][Pos]	
		0xF3 = Find Home	0x00	-	
		0xF4 = Exterminate	0x00	-	

โดยในโปรโตคอลส่งข้อมูลนี้เป็นชุดข้อมูลเลขฐานสิบหก ประกอบไปด้วยไบต์เริ่ม (Start) ไบต์ฟังก์ชัน (Function Type) ไบต์ขนาดข้อมูล (Payload Length) ไบต์ข้อมูล (Data) และไบต์สิ้นสุด (Stop) โดยมีรายละเอียดดังนี้

Start: 0x7E ขนาด 1 ไบต์ ใช้บอกจุดเริ่มต้นของชุดข้อมูล

Function Type: บอกชนิดคำสั่งที่ส่งไปยังตัวขับเคลื่อน โดยมีคำสั่งดังนี้

0xF1: รับข้อมูลตำแหน่งของหุ่นยนต์ในปัจจุบัน

0xF2: สั่งให้หุ่นยนต์เคลื่อนที่

0xF3: สั่งให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งเริ่มต้น เพื่อปรับเทียบและเตรียมพร้อมสำหรับการทำงานครั้งถัดไป

0xF4: สั่งให้หุ่นยนต์ทำการกำจัดวัตถุ

Payload Length: ขนาดของข้อมูลนับเป็นไบต์

Data: ข้อมูลตำแหน่งที่ต้องการเคลื่อนที่ ใช้ในคำสั่งเคลื่อนที่ (0xF2) เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Axis] คือแกนหุ่นยนต์ที่ต้องการเคลื่อนที่ มีขนาด 4 บิต โดย

0x1 = แกน X, 0x2 = แกน Y, 0x3 = แกน Z

[Pos] คือตำแหน่งที่ต้องการเคลื่อนที่ไป มีขนาด 12 bit (0 - 4095 มิลลิเมตร)

Stop: 0x7F ขนาด 1 ไบต์ ใช้บอกจุดสิ้นสุดของชุดข้อมูล

ตารางที่ 3.3 Protocol การรับข้อมูลจาก Arduino Mega 2560 ไปยังโหนด Controller

Rx Data	Start	Status	Payload Length	Message	Stop
Size (Byte)	1	1	1	n	1
Data	0x7E	0x0A = Success	n	[Position]	0x7F
		0x0F = Error		[Error Message]	

โดยในโปรโตคอลส่งข้อมูลนี้เป็นชุดข้อมูลเลขฐานสิบหก ประกอบไปด้วยไบต์เริ่ม (Start) ไบต์สถานะ (Status) ไบต์ขนาดข้อมูล (Payload Length) ไบต์ข้อความ (Message) และไบต์สิ้นสุด (Stop) มีรายละเอียดดังนี้

Start: 0x7E ขนาด 1 ไบต์ ใช้บอกจุดเริ่มต้นของชุดข้อมูล

Status: บอกชนิดข้อมูลที่ส่งกลับมาจากตัวขับเคลื่อน

0x0A: ได้รับคำสั่งถูกต้อง และดำเนินการสำเร็จ

0x0F: เกิดข้อผิดพลาดขึ้น

Payload Length: ขนาดข้อมูลของข้อความนับเป็นไบต์

Message: ข้อมูลตอบกลับจากตัวขับเคลื่อนสูงสุด 2^8-1 ไบต์

[Position]: ตำแหน่งของหุ่นยนต์ขณะปัจจุบัน

[Error Message]: ข้อผิดพลาดที่เกิดขึ้น

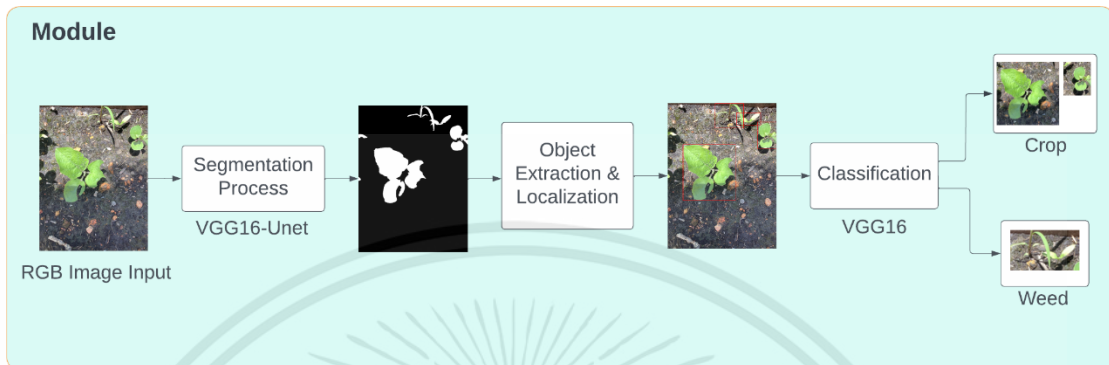
Stop: 0x7F ขนาด 1 ไบต์ ใช้บอกจุดสิ้นสุดของชุดข้อมูล

โดยการพูดคุยระหว่างโหนดทั้ง 6 ที่กล่าวมาจะใช้ชนิดข้อมูลแบบ String ที่มาจากไลบรารี std_msgs.msg ซึ่งเป็นไลบรารีของ ROS ซึ่งเป็นข้อมูลชนิดอ็อบเจกต์ที่เก็บข้อมูล data ที่เป็นชนิด string ไว้ข้างใน ส่งข้อมูลผ่าน topic โดยกำหนด publisher และ subscriber ของแต่ละโหนดให้ตรงกับ diagram การเชื่อมต่อผ่าน topic ตามแผนภาพรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การออกแบบระบบโมดูลสำหรับการจำแนกระหว่างพืชและวัชพืช

โมดูลนี้ใช้สำหรับการจำแนกระหว่างพืชและวัชพืช และทำการระบุตำแหน่งของวัชพืชเพื่อส่งกลับไปยังหุ่นยนต์ให้ทำการกำจัดต่อไป โดยมีหลักการทำงานดังนี้



รูปที่ 3.16 Weed Detection Process Diagram

หลักการทำงานของโมดูล มีดังนี้

- 1) ทำการแปลงภาพ RGB ที่ได้รับให้เป็นภาพไบนารี (Binary image) โดยการเรียกใช้โมเดลการแบ่งส่วนภาพ
- 2) นำภาพไบนารี (Binary image) มาเพิ่มขอบภาพ (edge) โดยใช้ Dilation Operator ให้ตัวภาพของพืชนั้นมีความหนามากขึ้น
- 3) ทำการระบุตำแหน่ง (Localization) และตีกรอบ (Bounding box) พื้นที่ที่สนใจในภาพ (ROI)
- 4) นำภาพที่ได้ทำการตีกรอบ (Bounding box) พื้นที่ที่สนใจ ซึ่งในที่นี้คือ พืชทั้งหมดในแปลง ไปทำต่อในกระบวนการจำแนกข้อมูล (Classification) เพื่อจำแนกพืชหลักและวัชพืช
- 5) ทำการส่งตำแหน่งของวัชพืชไปยังหุ่นยนต์

3.3 การประมวลผลภาพ

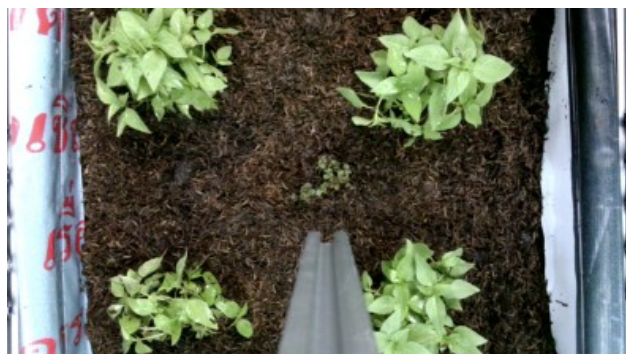
3.3.1 การเตรียมชุดข้อมูล

3.3.1.1 การเก็บชุดข้อมูล

โดยชุดข้อมูลที่นำมาใช้ในการฝึกสอน โมเดล แบ่งออกเป็น 2 แบบ คือ 1) ชุดข้อมูลที่เก็บเองจากแปลงปลูกพืช และ 2) ชุดข้อมูลออนไลน์ ดังนี้

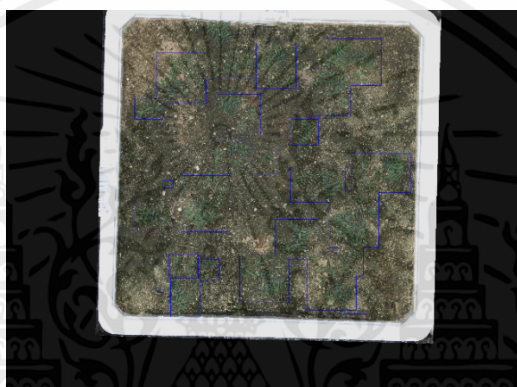
- 1) ชุดข้อมูลที่เก็บเองจากแปลงปลูกพืช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 ภาพที่เก็บจากแปลงปลูกพืช

2) ชุดข้อมูลออนไลน์



รูปที่ 3.18 ตัวอย่างชุดข้อมูลออนไลน์ที่นำมาใช้

(2020. **Plant semantic segmentation** [Online]. Available :

<https://www.kaggle.com/datasets/humansintheloop/plant-semantic-segmentation>)

3.3.1.2 การคัดเลือกชุดข้อมูล

นำภาพที่ได้จากการเก็บและภาพจากออนไลน์มาทำการคัดเลือก โดยทำการแยกภาพที่ไม่มีวัชพืชหรือวัชพืชในภาพออกไป เพื่อไม่ให้โมเดลเกิดความสับสน และสามารถจำแนกพืชได้อย่างแม่นยำ

3.3.1.3 การเพิ่มชุดข้อมูล (Image augmentation)

นำข้อมูลจำนวน 100 ภาพ ที่จะนำมาใช้สำหรับการเทรนมาทำการเพิ่มชุดข้อมูลโดยการครอบภาพแบบสุ่มตำแหน่ง เพื่อลดขนาดของภาพให้เหลือเพียง 128x128 พิกเซล และกำหนดค่าสำหรับการทำ augmentation ดังนี้

ทำการปรับ Horizontal flip ความน่าจะเป็นในการปรับคือ 0.5

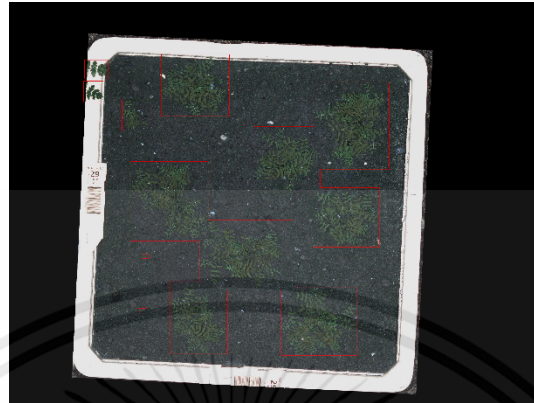
ทำการปรับ Gaussian noise ความน่าจะเป็นในการปรับคือ 0.2

ทำการปรับ Perspective ความน่าจะเป็นในการปรับคือ 0.5

ทำการปรับ Random brightness หรือ Random gamma ความน่าจะเป็นในการปรับคือ 0.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการปรับ Sharpen หรือ Blur ความน่าจะเป็นในการปรับคือ 0.9
ทำการปรับ Brightness contrast โดยทำการปรับเสมอ



รูปที่ 3.19 ตัวอย่างชุดข้อมูลที่นำมาทำการเพิ่มชุดข้อมูล



ก)

ข)

รูปที่ 3.20 ตัวอย่างภาพที่ได้จากการเพิ่มชุดข้อมูล

ก) ภาพที่มีฟิชประกอบอยู่

ข) ภาพที่ไม่มีฟิชประกอบอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงาน

ในบทนี้จะแสดงผลของการดำเนินงาน โดยแบ่งเนื้อหาออกเป็น 2 หัวข้อ ได้แก่ ผลการพัฒนาหุ่นยนต์ และผลการพัฒนาระบบ โมดูลจำแนกระหว่างพืชและวัชพืช ซึ่งมีรายละเอียดดังต่อไปนี้

4.1 ผลการพัฒนาหุ่นยนต์

4.1.1 ผลการพัฒนาโครงหุ่นยนต์

การติดตั้งและประกอบโครงสร้างของหุ่นยนต์ที่ใช้ในการเคลื่อนที่อุปกรณ์กำจัดวัชพืชไปยังตำแหน่งของวัชพืช โดยมีโครงสร้างจาก Farmbot เป็นต้นแบบ ที่ได้เพิ่มโมดูลเลเซอร์ และเปลี่ยนกล้องที่ใช้ โดยหลังจากที่ศึกษาและทำความเข้าใจชิ้นส่วนอุปกรณ์ต่าง ๆ ที่จำเป็นต้องใช้แล้ว จึงดำเนินการสั่งซื้อและประกอบชิ้นส่วนต่าง ๆ เข้าด้วยกันตามแผนงาน โดยในระหว่างการพัฒนา ก็ได้มีปัญหาการใช้ชิ้นส่วนข้อต่อที่ผลิตจากเครื่องพิมพ์สามมิติเกิดขึ้นซึ่งก็คือ แม้จะใช้ infill จำนวนมาก แต่ด้วยน้ำหนักของหุ่นยนต์และการใช้งาน ทำให้ข้อต่อบางส่วนงอ และบางส่วนก็เกิดการปริแตก ทำให้ต้องเปลี่ยนชิ้นส่วนเป็นอลูมิเนียมซึ่งมีน้ำหนักเบา และแข็งแรง ได้ผลลัพธ์ดังภาพที่ 4.1



รูปที่ 4.1 ผลลัพธ์หุ่นยนต์กำจัดวัชพืช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการพัฒนากระบวนการสื่อสารและการเคลื่อนที่

ดาวน์โหลดและติดตั้งระบบปฏิบัติการ Ubuntu 22.04.1 Desktop ลงบน Raspberry Pi ติดตั้ง XRDP เพื่อทำการรีโมทเข้ามาทำงานโดยไม่ต้องแกะเครื่องออกมาต่อจอได้ จากนั้นทำการติดตั้ง ROS 2 Humble Hawksbill บน Raspberry Pi ทำการสร้างเวิร์กสเปซ (workspace) สำหรับเก็บไฟล์เดออร์แพ็คเกจ (package) ที่ชื่อว่า weeder_robot ที่ใช้ไลบรารี python ในการสร้าง โดยสร้างโหนดตาม ROS Diagram ดังรูปที่ 3.11 หลังจากสร้าง (build) แพ็คเกจแล้วจะได้หน้าตาโครงสร้างโพลเดอร์ของแพ็คเกจดังนี้

```
workspace/src/weeder_robot
├── launch
│   ├── __pycache__
│   │   └── weeder_robot_launch.cpython-310.pyc
│   └── weeder_robot_launch.py
├── package.xml
├── resource
│   └── weeder_robot
├── setup.cfg
├── setup.py
├── test
│   ├── test_copyright.py
│   ├── test_flake8.py
│   └── test_pep257.py
└── weeder_robot
    ├── camera.py
    ├── image.py
    ├── __init__.py
    ├── logger.py
    ├── motor_controller.py
    ├── robot_controller.py
    └── user_interface.py
```

รูปที่ 4.2 โครงสร้างไฟล์แพ็คเกจ weeder_robot

อธิบายโครงสร้างจากรูปด้านบน จะมีโพลเดอร์ /resource, /test, และ /weeder_robot ที่เป็นส่วนที่ได้จากการสร้างและใช้งานแพ็คเกจ ซึ่งจากการทดลองการใช้งานปกติจะต้องสั่งรันโปรแกรมแต่ละโหนดแยกกันทั้งหมด 6 โหนดจนครบ ทำให้มีขั้นตอนที่ซ้ำซ้อน จึงทำการเพิ่มไฟล์ launch เข้ามาเป็นระบบที่ช่วยตั้งค่าการทำงานโปรแกรม ช่วยให้เราสามารถเรียกใช้โหนดหรือบริการทั้งหมดในคำสั่งเดียวได้ พร้อมทั้งสามารถตรวจสอบการทำงานของแต่ละโหนดได้ดังในรูปที่ 4.3

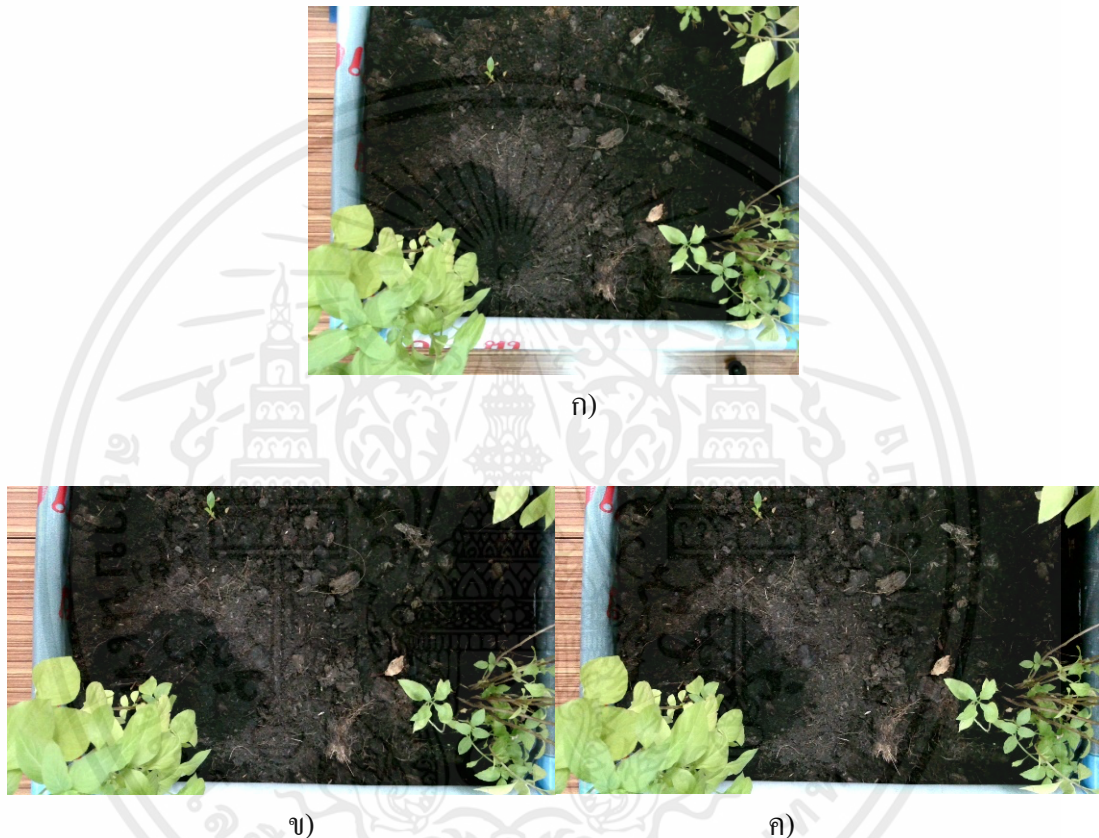
```
font@font-desktop:~/ROS/workspace/src/weeder_robot/launch$ ros2 launch weeder_robot_launch.py
[INFO] [launch]: All log files can be found below /home/font/.ros/log/2023-04-19-18-27-49-734965-font-desktop-904607
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [user_interface-1]: process started with pid [904640]
[INFO] [robot_controller-2]: process started with pid [904642]
[INFO] [camera-3]: process started with pid [904644]
[INFO] [motor_controller-4]: process started with pid [904646]
[INFO] [logger-5]: process started with pid [904648]
```

รูปที่ 4.3 ผลการทำงานของ launch file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อทำการกำหนดจุดในการถ่ายภาพของหุ่นยนต์ทั่วทั้งแปลงได้ เราจำเป็นต้องรู้ขอบเขตของภาพที่กล้องสามารถหาได้ และคำนวณหาระยะทางจริงเทียบกับพิกเซลที่ได้จากภาพ โดยเริ่มต้นได้ทำการวัดระยะจากเลนส์กล้องถึงพื้นและขอบแปลง โดยได้ผลลัพธ์ดังนี้

ทำการถ่ายภาพจากกล้องด้วยความสูง 420 มิลลิเมตร ในมุมมองเดียวกันด้วยลักษณะภาพที่ต่างกัน จากไลบรารี OpenCV ได้ผลลัพธ์จากการถ่ายภาพเป็นดังนี้



รูปที่ 4.4 ทดสอบภาพที่ได้ในแต่ละความละเอียดของกล้อง

ก) ความละเอียด 640 x 480 พิกเซล

ข) ความละเอียด 1280 x 720 พิกเซล

ค) ความละเอียด 1280 x 720 พิกเซล

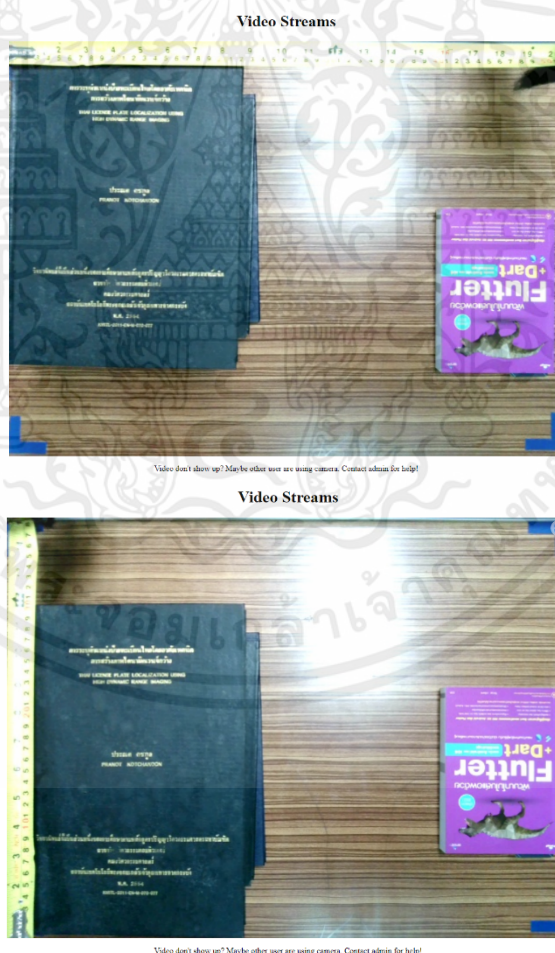
โดยภาพที่จับได้สามารถถ่ายได้ความกว้างจริงที่มีขนาดเท่ากัน แต่ภาพที่ความละเอียด 640 x 480 พิกเซลนั้นมีอัตราส่วนของภาพเป็น 4:3 ซึ่งมีความกว้างของภาพที่แคบกว่าภาพที่มีความละเอียด 1280 x 720 หรือ 1920 x 1080 ที่มีอัตราส่วนของภาพเป็น 16:9 ทำให้เมื่อได้ขนาดความกว้างของภาพที่เท่ากันแล้วจะได้ส่วนของภาพด้านแนวตั้งที่มากกว่าชนิดความละเอียด 1280 x 720 พิกเซล และ 1920 x 1080 พิกเซล

จากสามความละเอียดที่ได้ทดสอบ เราได้เลือกใช้ภาพความละเอียด 640 x 480 พิกเซล เนื่องจากภาพที่ได้มีพื้นที่ของภาพที่มากกว่า แกรมยังมีขนาดเล็กกว่า แต่ยังมีความละเอียดพอที่จะใช้ตรวจจับวัชพืชได้ จะทำให้โมเดลสามารถทำงานได้ไวกว่า

จากนั้นทำการวัดพื้นที่จริงได้จากขอบแปลงปลูกสูง 18 เซนติเมตร ได้ความยาวจริง 510 +/- 5 มิลลิเมตร และกว้าง 377.5 +/- 2.5 มิลลิเมตร จะได้ว่าในความยาว 1 พิกเซลวัดได้ 0.80 มิลลิเมตร และในความกว้าง 1 พิกเซลวัดได้ 0.79 มิลลิเมตร

ทั้งนี้การที่ 1 พิกเซลมีขนาดความกว้างหรือความยาวไม่เท่ากันนั้น เกิดจากเลนส์รับภาพที่มีมุมรับภาพ 70 องศาในแนวทแยง ทำให้รูปกับความยาวจริงที่วัดได้ของทั้งสองด้านมีขนาดไม่เท่ากัน

จากผลการทดสอบทำให้เลือกใช้ความยาวจากกรณีที่สอง เพราะการหาตำแหน่งของวัชพืชจะต้องวัดจากความสูงของแปลงปลูก หากใช้ความสูงจากกรณีที่หนึ่งจะได้ค่าที่ไกลกว่าความเป็นจริง



รูปที่ 4.5 การวัดหาสัดส่วนจริงต่อสัดส่วนของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ผลการทดสอบการทำงานของหุ่นยนต์

1) ทดสอบการ Decode/Encode Payload ที่รับส่งระหว่าง Raspberry Pi และ Arduino Mega 2560

เขียน Unit test ด้วยภาษาซีโดยทำการทดสอบ Valid test และ Boundary Test ทั้งหมด 21 Test Cases โดยมีวิธีการ Decode Package ด้วยการทำ Switch Stage ให้ตรวจสอบสถานะของ Byte ที่รับเข้ามาว่าอยู่ในขั้นตอนที่เท่าไร หากผิดพลาดก็จะทำการ Flush ข้อมูลที่จนกว่าจะถึง Start Byte ของชุดข้อมูลใหม่ จึงจะเริ่มกลับเข้าสู่ Stage Decode อีกครั้ง

ปัจจุบันทดสอบด้วยการส่ง Byte Payload ติดกันที่หลาย ๆ ชุด ประกอบไปด้วยชุดที่ถูกต้อง ติดกัน, ชุดแรกถูกต้อง ชุดที่สองผิดพลาด, ชุดแรกผิดพลาด ชุดที่สองถูกต้อง และผิดพลาดหลายชุด ติดกัน ผลคือสามารถ Decode ได้ตรงตามเคสที่ตั้งไว้ และ Raspberry Pi สามารถส่งค่าเพื่อให้ Arduino Mega 2560 ทำการจับมอเตอร์ได้ถูกต้อง

```

Case 01 : Move X 162 = Function F2, 1, 162
x_pos : 162

Case 02 : Move Y 230 = Function F2, 2, 230
y_pos : 230

Case 03 : Move Z 100 = Function F2, 3, 100
z_pos : 100

Case 04 : Get Pos = Function F1, 3, 100
Case 05 : Find Home = Function F3, 3, 100
x, y, z : 0 0 100

Case 06 : Dig = Function F4, 3, 100
Case 07 : Packet Loss = Invalid LengthByte
Case 08 : Invalid packetLen = Invalid LengthByte Invalid StartByte
Case 09 : Invalid Start Byte = Invalid StartByte Invalid StartByte Invalid StartByte Invalid StartByte
Case 10 : Invalid Stop Byte = Invalid StopByte
Case 11 : Invalid Function = Invalid FunctionByte Invalid StartByte Invalid StartByte
Case 12 : Invalid Function = Invalid FunctionByte Invalid StartByte Invalid StartByte
Case 13 : Invalid Stop Byte = Invalid StopByte
Case 14 : Invalid Axis = Invalid MovingAxis Invalid StartByte
Case 15 : Invalid Axis = Invalid MovingAxis Invalid StartByte
Case 16 : Invalid Axis = Invalid MovingAxis Invalid StartByte
Case 17 : 2 Get Pos Command = Function F1, 7, 1892Function F1, 7, 1892
Case 18 : 2 Get Pos with 2 error within = Function F1, 7, 1892Invalid StartByte Invalid StartByte Function F1, 7, 1892
Case 19 : Get Pos After Error byte = Invalid StartByte Invalid StartByte Function F1, 7, 1892
Case 20 : Encode OK 20 20 20 = 7E 0A 06 32 30 32 30 32 30 7F
Case 21 : Encode Error InvalidAxis = 7E 0F 0B 49 6E 76 61 6C 69 64 41 78 69 73 7F

```

รูปที่ 4.7 การทดสอบ Communication Protocol

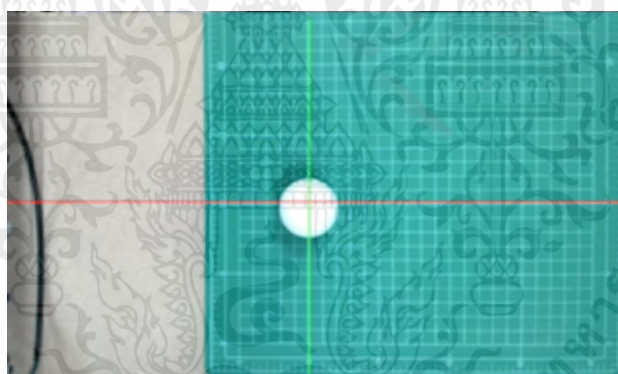
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ทดสอบการเคลื่อนที่ของหุ่นยนต์

ทดสอบโดยการสั่งให้หุ่นขยับจากจุดหนึ่งไปอีกจุดหนึ่ง แล้ววัดค่าด้วยกล้อง web cam โดยหาจุดกึ่งกลางของภาพเพื่อใช้ระยะของหุ่นยนต์ขณะปัจจุบันเทียบกับแผ่นยางรองตัด แล้ววัดว่าเมื่อเคลื่อนที่ไปอีกจุดหนึ่งนั้นมีตำแหน่งตรงกับที่สั่งให้เคลื่อนที่หรือไม่ ได้ผลการทดลองดังตารางด้านล่าง

ตารางที่ 4.1 การทดสอบการเคลื่อนที่ของหุ่นยนต์

การทดลองที่	แกน	จุดเริ่มต้น (มม.)	จุดสิ้นสุด (มม.)	ค่าที่ได้ (มม.)	ผลลัพธ์
1	X	0	100	100	ถูกต้อง
2	X	100	800	800	ถูกต้อง
3	Y	0	50	50	ถูกต้อง
4	Y	50	500	500	ถูกต้อง
5	Z	0	100	100	ถูกต้อง
6	Z	100	250	250	ถูกต้อง



รูปที่ 4.8 การวัดระยะการเคลื่อนที่ของหุ่นยนต์

จากการทดลองทั้ง 6 ครั้งในแกน X Y และ Z สรุปผลได้ว่าหุ่นยนต์สามารถเคลื่อนที่ได้ อย่างแม่นยำในระดับมิลลิเมตร สิ่งนี้อาจส่งผลกับการเคลื่อนที่ได้แก่น้ำหนักของแกน ความเร็วในการเคลื่อนที่ และความพอดีของระยะล้อเลื่อน หลังจากปรับปรุงระยะห่างของล้อเลื่อนให้พอดีแล้ว หุ่นยนต์ก็ไม่เกิดการกระตุกและสามารถเคลื่อนที่ในแต่ละการทดลองได้อย่างแม่นยำ

3) ทดสอบเวลาที่ใช้ในการเคลื่อนที่ของหุ่นยนต์

ทดสอบจับเวลาในการเคลื่อนที่จากจุดหนึ่งไปอีกจุดหนึ่งโดยใช้นาฬิกาจับเวลาเพื่อตรวจสอบเวลาที่ใช้ในการทำงานทั้งหมด ในที่นี้ใช้ไมโครสแต็ปเป็น 1/1 และใช้ความหน่วงต่อหนึ่งคำสั่งที่ 1 มิลลิวินาที ได้ผลลัพธ์การทดลองดังนี้

ตารางที่ 4.2 การทดสอบจับเวลาที่ใช้ในการเคลื่อนที่

การทดลองที่	แกน	ระยะทาง (มม.)	เวลาที่ใช้ (วินาที)	เวลาเฉลี่ย (มม./วินาที)
1	X	100	1.02	98.0
2	X	700	7.08	98.9
3	Y	50	0.47	106.4
4	Y	450	4.54	99.1
5	Z	100	5.06	19.8
6	Z	150	7.55	19.9

จากผลการทดลอง เวลาเฉลี่ยของแกน X, Y, Z ได้แก่ 98.45, 102.75 และ 19.85 มิลลิเมตรต่อวินาทีตามลำดับ โดยถ้าเฉลี่ยตามตัวเคลื่อนที่จะได้ค่าเฉลี่ยของสายพานอยู่ที่ 100.6 วินาที และของลีดสกรูอยู่ที่ 19.85 มิลลิเมตรต่อวินาที โดยเวลาที่ใช้สามารถลดเฉลี่ยให้อยู่ในระดับไมโครสแต็ปได้ แต่ทำให้หุ่นยนต์เกิดการสั่นและฝืด ส่งผลให้หุ่นยนต์เคลื่อนที่ได้ไม่แม่นยำ ทำให้ผู้จัดทำเลือกใช้ความหน่วงต่อคำสั่งที่ 1 มิลลิวินาที

4.2 ผลการพัฒนาระบบโมดูลจำแนกระหว่างพืชและวัชพืช

การพัฒนาระบบโมดูลจำแนกระหว่างพืชและวัชพืช แบ่งได้ทั้งหมด 3 ส่วน ได้แก่ 1) ส่วนการแบ่งส่วนภาพ 2) ส่วนการแยกลักษณะเฉพาะของภาพและการระบุตำแหน่ง 3) ส่วนการจำแนกข้อมูล

4.2.1 ส่วนการแบ่งส่วนภาพ

ขั้นแรกจะทำการจำแนกพืชและดินโดยการทำการแบ่งส่วนภาพ (Image segmentation) ซึ่งทำการพัฒนาโดยใช้โครงสร้าง U-Net และ กำหนด Backbone เป็น VGG16 จากนั้นได้ทำการเทรน โดยใช้ชุดข้อมูลที่ได้ทำการเพิ่มชุดข้อมูล และทำการใช้พารามิเตอร์ที่ได้จากการปรับค่าพารามิเตอร์ของโมเดล (Hyperparameter tuning) โดยใช้ GridSearch ซึ่งเป็นเทคนิคที่ใช้ในการปรับค่าพารามิเตอร์ของโมเดล โดยการใช้การทำให้แบบ cross-validation เพื่อหาค่าพารามิเตอร์ที่ดีที่สุดสำหรับโมเดล โดยจะทำการทดสอบค่าของพารามิเตอร์ที่เรากำหนดไว้ในลักษณะของกริด (Grid) จากนั้นจึงเลือกค่าที่ให้ผลลัพธ์ที่ดีที่สุด โดยได้กำหนดค่าของพารามิเตอร์ที่จะใช้ปรับแต่ละครั้งดังนี้

Optimizer ได้แก่ SGD, RMSprop, Adadelta, Adam

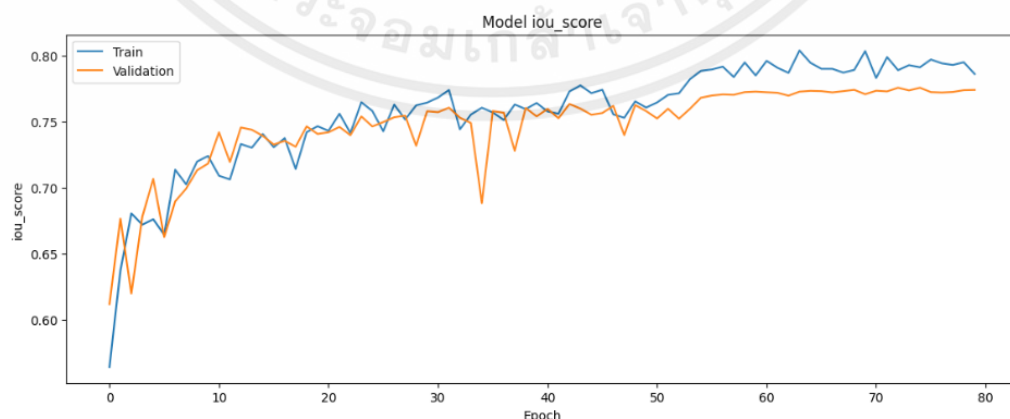
จำนวน Epochs ได้แก่ 20, 30, 40

Batch size ได้แก่ 2, 4, 8

ค่า Learning rate ได้แก่ 0.0001, 0.001, 0.1

โดยผลลัพธ์ที่ได้จากการใช้ GridSearch ค่าของพารามิเตอร์ที่ดีที่สุดที่ได้ นั่นคือ Optimizer คือ Adam จำนวน Epochs 40 epochs Batch size เท่ากับ 8 และ ค่า Learning rate เท่ากับ 0.0001

เมื่อนำค่าของพารามิเตอร์ที่ได้ไปทำการเทรนโมเดล และใช้ชุดข้อมูลสำหรับการทดสอบ มาทำการเรียกใช้โมเดล จะได้ค่า Intersection over Union (IoU) เฉลี่ยนั้นมีค่าเท่ากับ 0.7694 และผลลัพธ์ที่ได้จากการเรียกใช้โมเดลการแบ่งส่วนภาพ แสดงดังรูปที่ 4.10 (ข)



รูปที่ 4.9 กราฟแสดงค่า IoU ระหว่างการเทรนนิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ผลลัพธ์จากการเรียกใช้โมเดลการแบ่งส่วนภาพ

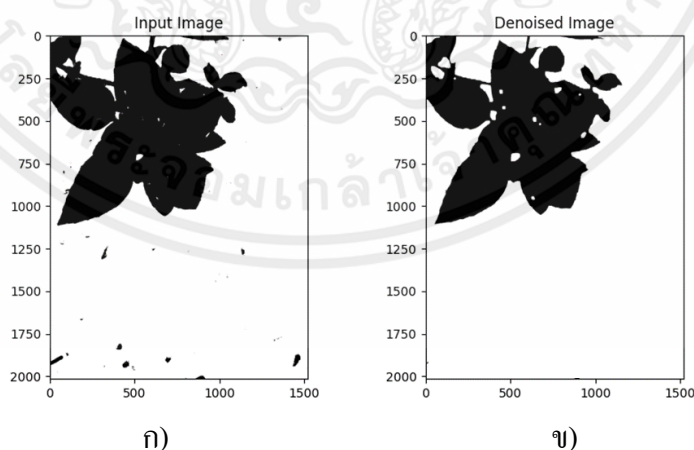
ก) ภาพที่ได้จากกล้อง

ข) ภาพไบนารีที่ได้จากการใช้โมเดลการแบ่งส่วนภาพ

4.2.2 ส่วนการแยกลักษณะเฉพาะของภาพและการระบุตำแหน่ง

ในส่วนของกระบวนการนี้ จะนำภาพไบนารีที่ได้จากการใช้โมเดลการแบ่งส่วนภาพ มาทำการแยกลักษณะเฉพาะและระบุตำแหน่งของพืช โดยกระบวนการแยกลักษณะเฉพาะและระบุตำแหน่งของพืชในภาพนั้นสามารถแบ่งออกได้เป็น 2 ขั้นตอน ดังนี้

1) การลดสัญญาณรบกวนในภาพ (noise) เพื่อปรับปรุงคุณภาพและทำให้ง่ายต่อการนำไปประมวลผลต่อไป ซึ่งวิธีที่นำมาใช้ในการลดสัญญาณรบกวนคือ การเปลี่ยนแปลงลักษณะรูปร่างของภาพ (Morphological Operation) โดยการดึงองค์ประกอบและคุณลักษณะต่างๆของภาพที่ต้องการ มาทำการกร่อน (Erosion) และขยายภาพ (Dilation) เพื่อลดสัญญาณรบกวนของภาพในขณะที่ยังคงคุณลักษณะสำคัญของพืช



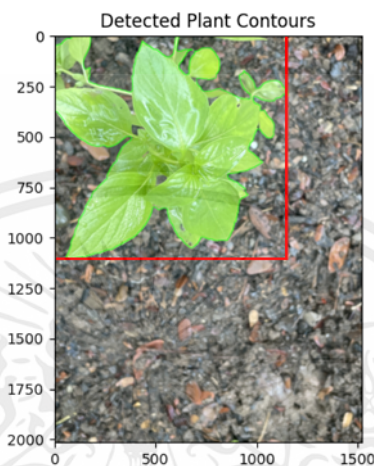
รูปที่ 4.11 ภาพนำเข้าและผลลัพธ์จากการลดสัญญาณรบกวนในภาพ

ก) Binary Image ที่ได้จากโมเดลการแบ่งส่วนภาพ

ข) Denoised Image ที่ได้จากการใช้การเปลี่ยนแปลงลักษณะรูปร่างของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

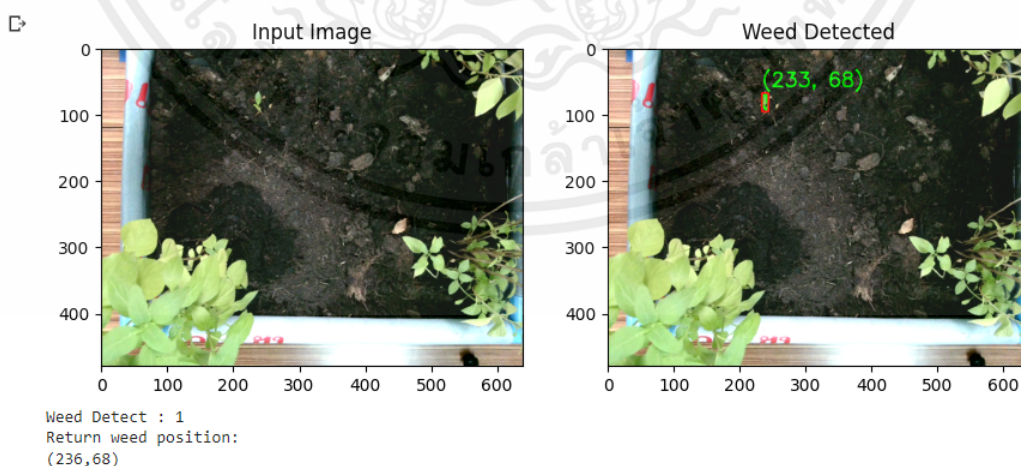
2) ทำการหาเส้นขอบของพืชและคำนวณพื้นที่ของกล่องรอบวัตถุ (Bounding Box) ที่ล้อมรอบพืช โดยมีการกำหนดเกณฑ์ขั้นต่ำของพื้นที่ของกล่องรอบวัตถุ เพื่อลดความผิดพลาดและปรับปรุงประสิทธิภาพ หลังจากนั้นทำการวาดกล่องรอบวัตถุลงบนวัตถุที่สนใจ และทำการจัดเก็บตำแหน่งของวัตถุ



รูปที่ 4.12 ภาพที่ได้ทำการวาดกล่องรอบวัตถุ

4.2.3 ส่วนการจำแนกข้อมูล

ขั้นตอนในการจำแนกข้อมูลนำภาพที่ได้ทำการวาดกล่องรอบวัตถุมาทำการจำแนกข้อมูล โดยในการพัฒนาใช้โมเดล VGG16 และใช้ชุดข้อมูลสำหรับการทดสอบมาทำการเรียกใช้โมเดล จำแนกว่าเป็นวัชพืช และผลลัพธ์ที่ได้จากการเรียกใช้โมเดลการจำแนกข้อมูล แสดงดังรูปที่ 4.12



รูปที่ 4.13 ภาพที่ได้ทำการจำแนกข้อมูลและคืนค่าตำแหน่งของวัชพืช

บทที่ 5

สรุปผล และข้อเสนอแนะ

ในบทนี้จะกล่าวถึงการสรุปผลการทำโครงการหุ่นยนต์กำจัดวัชพืชด้วยการประมวลผลภาพ และข้อเสนอแนะเพื่อเป็นแนวทางการพัฒนาโครงการต่อไป โดยมีรายละเอียดดังต่อไปนี้

5.1 สรุปผลการพัฒนาโครงการ

ในการพัฒนาโครงการทางคณะผู้จัดทำได้แบ่งการพัฒนาออกเป็น 2 ส่วนหลัก ได้แก่ ส่วนการพัฒนาหุ่นยนต์กำจัดวัชพืช และส่วนการจำแนกพืชและวัชพืช โดยส่วนการพัฒนาหุ่นยนต์กำจัดวัชพืชสามารถประกอบโครงสร้างส่วนการควบคุมการเคลื่อนที่ขึ้นมาได้ สามารถใช้ ROS ในการควบคุมการทำงานของกล้อง การควบคุมมอเตอร์ และการสื่อสารกับโมดูลประมวลผลร่วมกันได้ หุ่นยนต์สามารถเคลื่อนที่ได้โดยมีความละเอียด 1 มิลลิเมตร สามารถทำการถ่ายภาพจากกล้องเว็บแคมด้วยไลบรารี OpenCV ได้ และทำการส่งภาพที่ได้ไปทำการจำแนกพืชและดิน โดยการทำการ Image segmentation ซึ่งทำการพัฒนาโดยใช้โมเดล U-Net และ VGG-16 encoder โดยวัดค่า Intersection over Union (IoU) มีค่าเฉลี่ยเท่ากับ 0.769 นำภาพที่ได้ทำการระบุตำแหน่งของพืช ด้วยการใช่มอร์โฟโลยี Morphology Operation เพื่อลดสัญญาณรบกวนในภาพ (noise) และระบุตำแหน่งของพืชทั้งหมดในภาพโดยการวาดกล่องรอบวัตถุ (Bounding box) เพื่อนำไปจำแนกว่าเป็นพืชหรือวัชพืชด้วยโมเดล VGG16 โดยในการเรียกใช้โมเดลจำแนกว่าเป็นวัชพืช หลังจากนั้นการส่งคืนตำแหน่งของวัชพืชไปยังหุ่นยนต์ และหุ่นยนต์สามารถเคลื่อนที่ไปตามพิกัดที่ส่งมาและสั่งเปิดเลเซอร์เพื่อกำจัดวัชพืชตามตำแหน่งที่ได้รับมาได้อย่างถูกต้อง

5.2 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

การพัฒนาโครงการในครั้งนี้ถูกจัดทำขึ้นในห้องทดลองที่มีสภาพแวดล้อมจำกัด ได้แก่ อุณหภูมิ ความชื้น ทิศทางของแสง และพื้นที่ในการเคลื่อนที่ของหุ่นยนต์ที่จำกัดเท่านั้น หากจะนำไปใช้บนพื้นที่สภาพแวดล้อมจริงนั้นอาจจะได้ผลลัพธ์ความถูกต้องในการทำงานไม่เท่ากับในห้องทดลอง เนื่องจากองค์ประกอบทางสภาพแวดล้อมหลาย ๆ อย่างที่มีส่วน เช่น แสงและเงาของภาพที่ถ่ายได้ หรือพื้นผิวที่หุ่นยนต์เคลื่อนที่ อาจทำให้การตรวจจับหรือความแม่นยำในการเคลื่อนที่ของหุ่นยนต์ไปยังวัชพืชเพื่อกำจัดนั้นคลาดเคลื่อนจากตำแหน่งจริงได้ และในปัจจุบันตำแหน่งของวัชพืชที่ต้องกำจัดนั้น ไม่ได้เป็นค่าที่เรียงลำดับจากตำแหน่งที่มีประสิทธิภาพสูงสุด ควรนำค่าที่ได้ไปหาระยะทางที่สั้นหรือมีประสิทธิภาพที่สุดจากการเคลื่อนที่ก่อน โดยการเรียงลำดับระยะทางที่ใกล้ที่สุด หรือตรวจสอบว่าจุดที่จะไปกำจัดนั้น เป็นจุดเดิมกับที่เคยกำจัดมาแล้วหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการพัฒนาโครงการต่อนั้น ส่วนหุ่นยนต์ควรมีการเพิ่มระบบควบคุมแบบป้อนกลับ (Feedback control) เพื่อตรวจสอบหรือเปรียบเทียบผลลัพธ์การทำงานที่เกิดขึ้นให้มีความถูกต้อง ซึ่งจะทำให้การทำงานของหุ่นยนต์มีประสิทธิภาพและแม่นยำมากยิ่งขึ้น สามารถนำไปใช้กับการแสดงผล เช่นการพัฒนาส่วนของผู้ใช้ให้มีความทันสมัยมากยิ่งขึ้น สามารถควบคุมหุ่นยนต์ได้อย่างเห็นภาพหรือตำแหน่งได้ชัดเจนยิ่งขึ้น ในส่วนจำแนกพืชและวัชพืชควรเพิ่มจำนวนชุดข้อมูลที่ใช้ให้มีความหลากหลายมากขึ้น ทั้ง ภาพพืชที่ถ่ายในสภาพแสงและเงาที่ต่างกัน รวมถึงภาพพืชหรือวัชพืชชนิดอื่น ๆ เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- Xiaojun J., Jun C. and Yong C. 2021. **Weed Identification Using Deep Learning and Image Processing in Vegetable Plantation** [Online]. Available :
<https://ieeexplore.ieee.org/document/9317797>
- Mulham Fawakherji, Ali Youssef, Domenico Bloisi, Alberto Pretto, Daniele Nardi. 2019. **Crop and Weeds Classification for Precision Agriculture Using Context-Independent Pixel-Wise Segmentation** [Online]. Available :
<https://ieeexplore.ieee.org/document/8675654>
- พงศกร ชาญชัยชูจิต และพฤทธิกร สมิตไมตรี. 2018. **Management of a Multi-robots System for Industrial Material Handling**. วารสารวิชาการพระจอมเกล้าพระนครเหนือ. 29(3), 454-464.
- Michael Chan. 2020. **Classical Neural Network: What really are Nodes and Layers?** [Online]. Available : <https://towardsdatascience.com/classical-neural-network-what-really-are-nodes-and-layers-ec51c6122e09>
2020. **รู้จักกับประเภทของ หุ่นยนต์อุตสาหกรรม และการใช้งานที่เกี่ยวข้อง**. [Online]. Available : <https://www.sumipol.com/knowledge/types-of-industrial-robot/>
- TullyFoote. 2022. **ROS 2 Documentation** [Online]. Available : <http://wiki.ros.org/Documentation>
- Aswirth Raj. 2018. **What is Stepper Motor and How it Works**. [Online]. Available : <https://circuitdigest.com/tutorial/what-is-stepper-motor-and-how-it-works>
2019. **What is the difference between full-stepping, the half-stepping, and the micro-drive?** [Online]. Available : <https://www.automate.org/case-studies/what-is-the-difference-between-full-stepping-the-half-stepping-and-the-micro-drive>
- Daniel Johnson. 2022. **Deep Learning Tutorial for Beginners: Neural Network Basics**. [Online]. Available : <https://www.guru99.com/deep-learning-tutorial.html>



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

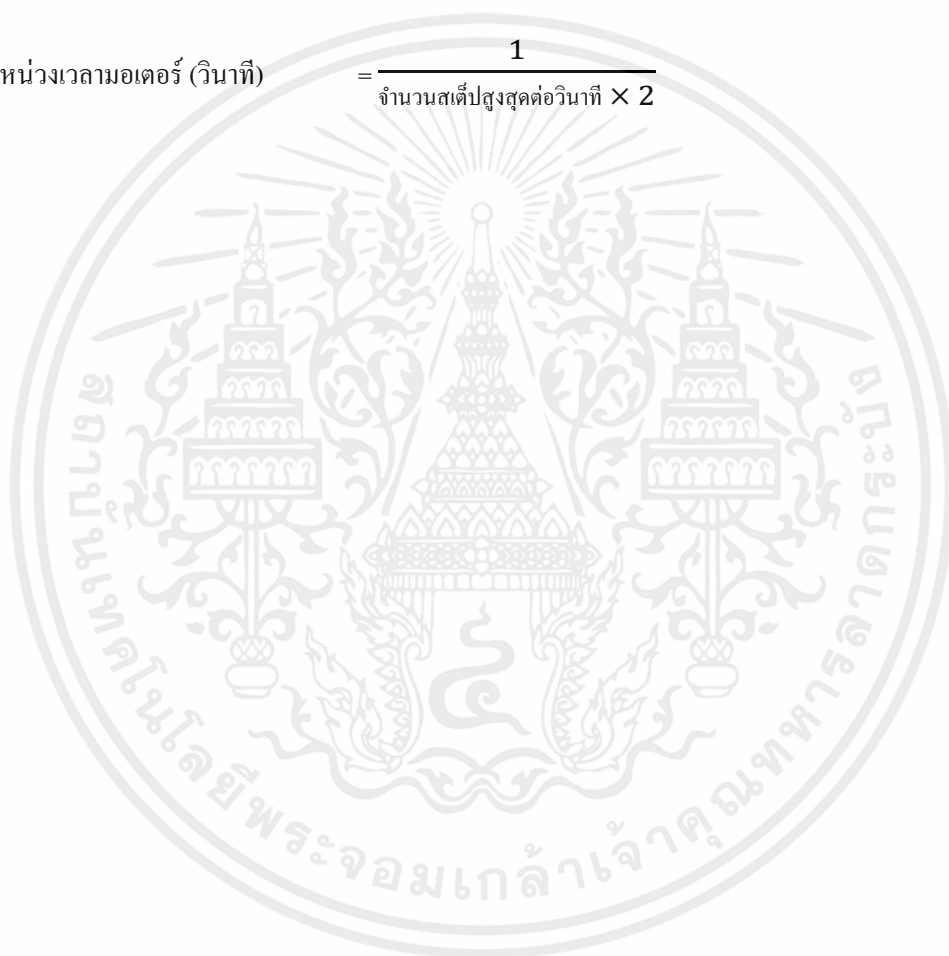
ภาคผนวก ก

การคำนวณการเคลื่อนที่ของมอเตอร์

$$\text{การเคลื่อนที่ของสายพาน (สเต็ป/มม.)} = \frac{\text{ความละเอียดของมอเตอร์ (สเต็ป/รอบ)}}{\text{จำนวนไมโครสเต็ป} \times \text{จำนวนฟันลูกกรอก (มม.)} \times \text{ขนาดพิทช์ของสายพาน (มม.)}}$$

$$\text{การเคลื่อนที่ของลิคตกรู (สเต็ป/มม.)} = \frac{\text{ความละเอียดของมอเตอร์ (สเต็ป/รอบ)}}{\text{จำนวนไมโครสเต็ป} \times \text{ขนาดพิทช์ของลิคตกรู (มม.)}}$$

$$\text{ค่าหน่วยเวลามอเตอร์ (วินาที)} = \frac{1}{\text{จำนวนสเต็ปสูงสุดต่อวินาที} \times 2}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รายการวัสดุ

ลำดับ ที่	รายการ	ราคา/หน่วย	หน่วย	จำนวน	รวม
1	Aluminium Profile V-Slot (20x20 mm)	128.00	เมตร	5	640.00
2	Aluminium Profile V-Slot (20x40 mm)	240.00	เมตร	3.40	816.00
3	Aluminium Profile V-Slot (20x60 mm)	425.00	เมตร	1	425.00
4	Lead Screw 8mm 700mm	300.00	ชิ้น	1	300.00
5	Nut Block (Lead 8mm)	45.00	ชิ้น	1	45.00
6	5 x 8 mm Flexible Shaft Coupling	85.00	ชิ้น	1	85.00
7	2GT Width 6mm, 20 Teeth Timing Pulley	30.00	ชิ้น	3	90.00
8	Solid v wheel for v-slot (set)	44.00	ชุด	30	1,320.00
9	Aluminium Spacer Isolation Column (10pcs)	30.00	ชุด	2	60.00
10	Eccentric column	7.00	ชิ้น	13	91.00
11	M5x25mm Low Profile Screws (10pcs)	30.00	ชุด	3	90.00
12	2 core x 0.75 mm ² Shielded Wire Signal	35.00	เมตร	2	70.00
13	Limit Switch + Connector	30.00	ชิ้น	3	90.00
14	10 x 20 mm Cable Drag Chain R28mm 1M and End Connectors (Bridge open outside)	290.00	ชิ้น	3	870.00
15	2GT-6mm Dustproof Neoprene Timing Belt	60.00	เมตร	5	300.00
16	Z-Axis Motor Mount	490.00	ชิ้น	1	490.00
17	AL5083 Cross-Slide Plate	850.00	ชิ้น	1	850.00
18	AL1100 Gantry Corner Bracket	900.00	ชิ้น	2	900.00
19	AL1100 Gantry Wheel Plate	750.00	ชิ้น	2	750.00
20	NEMA 17 Stepper motor	270.00	ชิ้น	4	1,080.00
21	EGA Type W1 Stream Webcam	990.00	ชิ้น	1	990.00
22	LD3070LA Laser Module 450nm 40W	3,000.00	ชิ้น	1	3,000.00
23	Arduino Mega 2560 + RAMPS 1.6	650.00	ชุด	1	650.00
24	DRV8825	25.00	ชิ้น	5	125.00
25	Raspberry Pi 4 Model B 4GB	2,000.00	ชิ้น	1	2,000.00
26	Power Supply 12V 30A	350.00	ชิ้น	1	350.00
27	Electronic Plastic Enclosure	245.00	ชิ้น	1	245.00
รวมสุทธิ = 16,722 บาท					

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้