

การตรวจจับวัตถุและประเมินตำแหน่งและมุมมองศาในรูปแบบสามมิติด้วย
เทคนิคแบบไฮบริดระหว่างการเรียนรู้เชิงลึกและอัลกอริทึม FAST-BRISK
สำหรับการประยุกต์ใช้งานประเภทหยิบวัตถุในภาชนะ

HYBRID DEEP LEARNING AND FAST-BRISK 3D OBJECT DETECTION
TECHNIQUE FOR BIN-PICKING APPLICATION



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัจฉริยะเชิงคำนวณ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2567

KMITL-2024-EN-M-407-246

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HYBRID DEEP LEARNING AND FAST-BRISK 3D OBJECT DETECTION
TECHNIQUE FOR BIN-PICKING APPLICATION



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ROBOTICS AND COMPUTATIONAL
INTELLIGENCE SYSTEMS
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2024

KMITL-2024-EN-M-407-246

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2024

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การตรวจจับวัตถุและประเมินตำแหน่งและมุมมองศาในรูปแบบสามมิติด้วยเทคนิคแบบไฮบริดระหว่างการเรียนรู้เชิงลึกและอัลกอริทึม FAST-BRISK สำหรับการประยุกต์ใช้งานประเภทหยิบวัตถุในภาชนะ
นักศึกษา	นายธนกฤต ทวีสุนทร
รหัสประจำตัว	64601059
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมหุ่นยนต์และระบบอัจฉริยะเชิงคำนวณ
พ.ศ.	2567
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.ภูมิ คงหัวรอบ

บทคัดย่อ

ปัจจุบันนี้การใช้งานหุ่นยนต์แขนกลได้รับความนิยมในอุตสาหกรรมต่าง ๆ เนื่องจากความแม่นยำและประสิทธิภาพในการทำงาน ในงานวิจัยนี้ได้นำเสนอวิธีการผสมผสาน (hybrid method) สำหรับงานหยิบวัตถุในภาชนะโดยใช้หุ่นยนต์ร่วมปฏิบัติงานหรือ cobot ร่วมกับโครงข่ายประสาทเทียมแบบคอนโวลูชันด้วยอัลกอริทึม You Only Look Once version 5 หรือ YOLOv5 สำหรับการตรวจจับวัตถุและประเมินตำแหน่ง ผสมผสานกับเทคนิคการตรวจจับลักษณะเด่นโดยใช้เทคนิค FAST และการอธิบายลักษณะเด่นโดยใช้เทคนิค BRISK และอัลกอริทึมการจับคู่ลักษณะเด่นเพื่อใช้ในการประเมินมุมมองศาของวัตถุ โดยการใช้งานอัลกอริทึมเหล่านี้ร่วมกับกล้องถ่ายภาพความลึกราคาต่ำที่ถูกใช้ในการถ่ายภาพสีและภาพความลึก ทำให้สามารถเพิ่มความเร็วในการตรวจจับวัตถุและการประเมินมุมมองศาของวัตถุได้อย่างแม่นยำในงานแบบเรียลไทม์ ทำให้หุ่นยนต์สามารถหยิบวัตถุได้มีประสิทธิภาพ นอกจากนี้ วิธีการที่นำเสนอข้างต้นถูกนำมาใช้ผ่านแพลตฟอร์ม ROS ซึ่งทำให้การทำงานร่วมกันระหว่างอุปกรณ์ต่าง ๆ รวมถึงอัลกอริทึมการคำนวณต่าง ๆ สามารถทำงานร่วมกันได้อย่างไร้รอยต่อ เราได้ทำการเปรียบเทียบผลลัพธ์ของงานวิจัยกับวิธีการอื่น ๆ โดยมุ่งเน้นที่ความแม่นยำในการตรวจจับวัตถุและความเร็วในการประมวลผลของวิธีการผสมผสานของเรา การบูรณาการเทคโนโลยีหุ่นยนต์ กล้อง และปัญญาประดิษฐ์ในงานวิจัยนี้ไม่เพียงแต่ช่วยพัฒนาหุ่นยนต์อุตสาหกรรม แต่ยังโอกาสใหม่ในการจัดการงานที่ซับซ้อนในระบบอัตโนมัติ พร้อมทั้งยกระดับประสิทธิภาพการดำเนินงานให้สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	HYBRID DEEP LEARNING AND FAST-BRISK 3D OBJECT DETECTION TECHNIQUE FOR BIN-PICKING APPLICATION
Student	Mr. Thanakrit Taweesoontorn
Student ID.	64601059
Degree	Master of Engineering
Program	Robotics and Computational Intelligence Systems
Year	2024
Thesis Advisor	Asst.Prof.Dr.Poom Konghuayrob

ABSTRACT

In the field of industrial robotics, the integration of robotic arms has significantly risen, driven by their precise functionality and operational efficiency. We here propose hybrid method for bin-picking tasks using a collaborative robot, or cobot combining the You Only Look Once version 5 (YOLOv5) convolutional neural networks (CNNs) model for object detection and pose estimation with traditional feature detection base on FAST technique, feature description utilized BRISK, and matching algorithms. By integrating these algorithms and utilizing a low-cost depth sensor camera for capturing depth and RGB images, the system enhances real-time object detection and pose estimation speed, facilitating accurate object manipulation by the robotic arm. Furthermore, the proposed method is implemented within the ROS framework to provide a seamless platform for robotic control and integration. We conduct a comparative of our results with other methodologies, highlighting the superior object detection accuracy and processing speed of our hybrid approach. This integration of robotic, camera, and AI technology contributes to the development of industrial robotics, opening up new possibilities for automating challenging tasks and improving overall operational efficiency.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยขอขอบพระคุณ ผศ.ดร.ภูมิ คงห้วยรอบ ที่ให้ความอนุเคราะห์รับเป็นอาจารย์ที่ปรึกษา ให้คำปรึกษา แนะนำ และให้การช่วยเหลือในการทำวิทยานิพนธ์ รวมถึงสอนวิธีการทำงานที่ดีทำให้ผู้วิจัยรู้จักพัฒนาตนเองในด้านการงานให้ดียิ่งขึ้น

ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ที่ให้คำแนะนำและแนวทางในการปรับปรุงวิทยานิพนธ์ให้มีความสมบูรณ์มากขึ้น ขอขอบคุณคณะอาจารย์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่สั่งสอนวิชาความรู้ ถ่ายทอดประสบการณ์ต่าง ๆ และนำความรู้เหล่านั้นมาใช้ในการทำวิจัยนี้

สำหรับคุณประโยชน์และคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้บิดามารดาผู้เป็นที่รักและเคารพยิ่ง ครูอาจารย์ที่ประสิทธิ์ประสาทวิชาความรู้ให้แก่ข้าพเจ้ามาตลอดจนถึงทุกวันนี้

ธนฤต ทวีสุนทร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและสำคัญ.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	3
1.3 สมมุติฐานของการศึกษา.....	4
1.3.1 You Only Look Once Version 5 หรือ YOLOv5.....	4
1.3.2 อัลกอริทึม Features from Accelerated Segment Test หรือ FAST และ อัลกอริทึม Binary Robust Invariant Scalable Keypoints หรือ BRISK.....	4
1.4 ขอบเขตของการศึกษา.....	5
1.5 ขั้นตอนของการศึกษา.....	6
1.6 แผนการดำเนินงาน.....	6
1.7 ขั้นตอนและระยะเวลาในการดำเนินงานจริง.....	8
1.8 คำจำกัดความที่ใช้ในการศึกษา.....	10
บทที่ 2 งานวิจัยที่เกี่ยวข้อง.....	11
2.1 การตรวจจับวัตถุจากภาพด้วยวิธีการหาคุณลักษณะในภาพ.....	11
2.2 การเรียนรู้เชิงลึกและโครงข่ายประสาทเทียมแบบคอนโวลูชัน.....	12
2.2.1 โครงข่ายประสาทเทียม.....	12
2.2.2 โครงข่ายประสาทเทียมคอนโวลูชัน.....	13
2.2.3 อัลกอริทึม You Only Look Once หรือ YOLO.....	14
2.3 ระบบปฏิบัติการหุ่นยนต์และโปรแกรมแสดงภาพ.....	15
บทที่ 3 การตรวจจับวัตถุและประเมินตำแหน่งและมุมมอง.....	17
3.1 อัลกอริทึมแรกสำหรับการตรวจจับวัตถุ (YOLOv5).....	17
3.2 อัลกอริทึมที่สองสำหรับการตรวจจับวัตถุ (FAST and BRISK).....	19

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 อัลกอริทึมการตรวจจับคุณลักษณะ FAST.....	19
3.2.2 อัลกอริทึมการจดจำคุณลักษณะ BRISK	21
3.2.3 อัลกอริทึมการจับคู่คุณลักษณะ	21
3.3 ขั้นตอนการทำงานของโปรแกรมและการใช้งานระบบปฏิบัติการหุ่นยนต์ ROS	22
3.3.1 ขั้นตอนการทำงานของระบบปฏิบัติการหุ่นยนต์ ROS.....	24
3.3.2 การแยกการทำงานเป็นโหนดสำหรับแต่ละโปรแกรมและอุปกรณ์	25
3.4 การคำนวณและการกำหนดท่าทางของหุ่นยนต์.....	29
บทที่ 4 วิธีการทดลองและผลการทดลอง	35
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	42
บรรณานุกรม.....	43
ภาคผนวก ก การผสมผสานเทคนิคการตรวจจับวัตถุเพื่อพัฒนาการทำงานของหุ่นยนต์และ กระบวนการโดยรวมในอุตสาหกรรม.....	45
ประวัติผู้เขียน.....	63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 แผนการดำเนินงานเดือน สิงหาคม พ.ศ.2566 ถึง ตุลาคม พ.ศ.2566 (ส่วนที่ 1)	6
ตารางที่ 1.2 แผนการดำเนินงานเดือน พฤศจิกายน พ.ศ.2566 ถึง มกราคม พ.ศ.2567	7
ตารางที่ 1.3 แผนการดำเนินงานเดือน กุมภาพันธ์ พ.ศ.2567 ถึง เมษายน พ.ศ.2567.....	8
ตารางที่ 1.4 การดำเนินงานจริงเดือน สิงหาคม พ.ศ.2566 ถึง ตุลาคม พ.ศ.2566 (ส่วนที่ 1).....	8
ตารางที่ 1.5 การดำเนินงานจริงเดือน พฤศจิกายน พ.ศ.2566 ถึง มกราคม พ.ศ.2567	9
ตารางที่ 1.6 การดำเนินงานจริงเดือน กุมภาพันธ์ พ.ศ.2567 ถึง เมษายน พ.ศ.2567.....	10
ตารางที่ 3.1 ผลลัพธ์การฝึกอบรมการตรวจจับวัตถุ.....	15
ตารางที่ 3.2 ความสามารถในการตรวจจับและระบุข้อมูลวัตถุ	29
ตารางที่ 4.1 ระยะเวลาในการตรวจจับวัตถุ.....	33
ตารางที่ 4.2 ระยะเวลาการทำงานในแต่ละขั้นตอนจากการทดลอง.....	34
ตารางที่ 4.3 การเปรียบเทียบผลลัพธ์ที่ได้จากการทดลองเทียบกับเทคนิคอื่น ๆ ก่อนหน้า	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
รูปที่ 1.1	หุ่นยนต์จริงและการจำลองหุ่นยนต์เชิงโครโนซ์แบบเรียลไทม์ใน Rviz พร้อมด้วย pointcloud ภาพความลึก และภาพวัตถุที่ตรวจพบ..... 4
รูปที่ 1.2	การแสดงผลแผนผังของกระบวนการทำงานของระบบ เริ่มตั้งแต่การรับข้อมูลจากกล้องไปยัง โหนดการตรวจจับและท้ายที่สุดไปยังหุ่นยนต์ โดยเน้นขั้นตอนต่าง ๆ ที่เกี่ยวข้องกับการตรวจจับวัตถุ การระบุตำแหน่งและมุมมอง และการคำนวณท่าทางหุ่นยนต์ในการหยิบวัตถุ..... 5
รูปที่ 2.1	ตรวจพบรูปภาพที่มีจุดสังเกต จุดสีเขียวแสดงมุม non-maximally suppressed [10]... 11
รูปที่ 2.2	โครงข่ายประสาทเทียมแบบหลายชั้นเพอร์เซปตรอน [13] 13
รูปที่ 2.3	ขั้นตอนการทำงานเบื้องต้นในการประมวลผลภาพด้วย YOLO [14] 15
รูปที่ 3.1	(a) ตำแหน่งมุมมองด้านบนของหุ่นยนต์ (b) ตำแหน่งของหุ่นยนต์ในขณะที่เคลื่อนที่เข้าใกล้ วัตถุที่เลือกมากขึ้น..... 17
รูปที่ 3.2	หลักทำงานเบื้องต้นของอัลกอริทึม CNN ที่เป็นเทคนิคที่ใช้ในการตรวจจับวัตถุของ อัลกอริทึม YOLOv5 18
รูปที่ 3.3	(a) ชุดข้อมูลการฝึกอบรมด้วยเทคนิคการเสริม (b) ผลลัพธ์การฝึกอบรมโดยใช้อัลกอริทึม YOLOv5..... 18
รูปที่ 3.4	แผนผังของจุดสำคัญที่กำลังประเมิน พร้อมด้วยพิกเซล 16 พิกเซลโดยรอบที่จัดเรียงเป็นรูป วงกลม..... 19
รูปที่ 3.5	(a) ภาพ RGB พร้อมการตรวจจับคุณสมบัติจาก FAST (b) ภาพความลึกแสดงเป็นแผนที่ ความร้อน (c) ลิงก์วัตถุที่มองเห็นใน Rviz..... 22
รูปที่ 3.6	ขั้นตอนกระบวนการทำงานของระบบในการถ่ายภาพวัตถุ การตรวจจับและเลือกวัตถุ คำนวณมุมมองของวัตถุและการหยิบวัตถุ..... 23
รูปที่ 3.7	โหนดการทำงานส่วนเซนเซอร์รับภาพสีและภาพความลึก ที่ใช้งานกล้อง Intel realsense D435i..... 26
รูปที่ 3.8	ขั้นตอนการส่งข้อมูลภาพวัตถุไปยังโหนด 2 โหนด ที่ใช้ข้อมูลนำไปประมวลผลภาพต่อไป 27
รูปที่ 3.9	ขั้นตอนการส่งข้อมูลต่าง ๆ ของโหนดหุ่นยนต์และโหนดการแสดงผล..... 28
รูปที่ 3.10	แผนผังการเชื่อมต่อและการรับส่งข้อมูลต่าง ๆ ของแต่ละโหนด 29
รูปที่ 3.11	(a) หุ่นยนต์สร้างภาพใน Rviz, (b) - (d) การหาตำแหน่งของวัตถุอย่างต่อเนื่องจากลิงค์ฐาน ของหุ่นยนต์..... 30
รูปที่ 3.12	อุปกรณ์ที่ใช้ในการติดตั้งกล้องเข้ากับปลายแขนของหุ่นยนต์ 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.13 รูปถ่ายของขั้นตอนการหยิบซึ่งแสดงทิศทางการหยิบในแนวตั้งที่แม่นยำและการเชื่อมต่อที่สมบูรณ์แบบของขอบถ้วยคู่กับวัตถุ.....	33
รูปที่ 4.1 Rviz เครื่องมือสร้างภาพ ROS แสดงลิงก์วัตถุ โมเดลหุ่นยนต์พร้อมลิงก์กล้อง พอยต์คลาวด์ ภาพความลึก และภาพวัตถุที่ตรวจพบ.....	35
รูปที่ 4.2 การทดลองที่ 1 วัตถุถูกจัดเรียงแบบเป็นระเบียบ, การทดลองที่ 2 วัตถุถูกจัดเรียงแบบกึ่งสุ่ม แนวของวัตถุยังคงหันไปในทิศเดียวกัน และการทดลองที่ 3 วัตถุทั้งหมดวางแบบสุ่มในทุกทิศทาง...	36
รูปที่ 4.3 การทำงานของหุ่นยนต์ที่แสดงผลผ่านโปรแกรม Rviz โดยแสดงตำแหน่งของวัตถุแบบ 3 มิติ พร้อมภาพการตรวจจับวัตถุและภาพความลึก	37
รูปที่ 4.4 โปรแกรม Rviz และส่วนประกอบต่าง ๆ สำหรับแสดงการทำงานของหุ่นยนต์	38
รูปที่ 4.5 ลำดับรูปภาพแสดงให้เห็นถึงการดำเนินการตามกระบวนการรับสินค้าที่ประสบความสำเร็จ (a) หุ่นยนต์ถูกวางตำแหน่งในการกำหนดค่ามุมมองด้านบน (b) หุ่นยนต์เคลื่อนที่เพื่อดูวัตถุเป้าหมายในระยะใกล้ (c) หุ่นยนต์เข้าใกล้วัตถุด้วย offset คงที่ (d) หุ่นยนต์เริ่มการเคลื่อนที่แบบหยิบจับ (e) หุ่นยนต์หยิบวัตถุได้สำเร็จ (f) หุ่นยนต์ขนย้ายวัตถุไปยังลังจัดเก็บ ภาพชุดนี้แสดงให้เห็นถึงการปฏิบัติงานหยิบจับโดยหุ่นยนต์อย่างเป็นระบบและแม่นยำ	40

บทที่ 1

บทนำ

1.1 ความเป็นมาและสำคัญ

ในปัจจุบันนี้การนำหุ่นยนต์มาใช้ในการอุตสาหกรรมต่าง ๆ มีความนิยมเพิ่มขึ้นมาก เนื่องจากความก้าวหน้าทางเทคโนโลยีในด้านหุ่นยนต์ที่เติบโตอย่างต่อเนื่อง นำมาซึ่งหุ่นยนต์ที่มีประสิทธิภาพขึ้นอย่างมากในระดับการผลิตและการบริการในหลายกลุ่มอุตสาหกรรม เช่น อุตสาหกรรมผลิตรถยนต์, อุตสาหกรรมอิเล็กทรอนิกส์, อุตสาหกรรมอาหารและเครื่องดื่ม, และอื่น ๆ ความสามารถของหุ่นยนต์ในการทำงานประเภททำซ้ำได้ดีได้รับการพัฒนาอย่างมีประสิทธิภาพ ซึ่งสามารถช่วยลดค่าใช้จ่ายในการผลิตและเพิ่มประสิทธิภาพในการทำงาน เช่น การใช้หุ่นยนต์ในกระบวนการผลิตที่ต้องการการทำงานที่ซ้ำซาก หรือในโกดังสินค้าต่าง ๆ ตลอดจนการใช้งานหุ่นยนต์ในการคัดแยกหรือจัดเรียงสินค้า

ในขณะเดียวกัน คอมพิวเตอร์วิทัศน์และปัญญาประดิษฐ์เป็นสองเทคโนโลยีที่มีบทบาทสำคัญและมีอิทธิพลอย่างมากต่อการพัฒนาและการใช้งานหุ่นยนต์ในหลากหลายสาขาอุตสาหกรรม ซึ่งเทคโนโลยีเหล่านี้มีความสามารถในการประมวลผลข้อมูลและเรียนรู้จากข้อมูลอย่างมีประสิทธิภาพ ช่วยให้หุ่นยนต์สามารถตรวจจับและตีความข้อมูลที่ได้รับจากภาพและวิดีโอได้อย่างแม่นยำ ทำให้สามารถรับรู้สภาพแวดล้อมในพื้นที่การทำงานซึ่งนำไปสู่การทำงานที่ซับซ้อนและมีความท้าทายได้ เทคโนโลยีปัญญาประดิษฐ์เสริมความสามารถในการเรียนรู้และการปรับตัวของหุ่นยนต์จากข้อมูลที่ได้รับ ช่วยให้หุ่นยนต์สามารถวิเคราะห์และตัดสินใจในสถานการณ์ที่ไม่แน่นอนหรือเปลี่ยนแปลงได้ เช่น การนำหุ่นยนต์มาใช้ในการคัดแยกสินค้าในสายการผลิต หรือการนำหุ่นยนต์มาใช้ในการจัดการคลังสินค้าโดยการวิเคราะห์และวางแผนเพื่อปรับปรุงการจัดเก็บสินค้าในคลังให้เป็นไปอย่างมีประสิทธิภาพที่สุด ดังนั้น ความร่วมมือระหว่างคอมพิวเตอร์วิทัศน์และปัญญาประดิษฐ์เป็นสิ่งสำคัญในการพัฒนาและใช้งานหุ่นยนต์ในปัจจุบันและในอนาคต หนึ่งในตัวอย่างการใช้งานคอมพิวเตอร์วิทัศน์และปัญญาประดิษฐ์ในหุ่นยนต์คือการตรวจจับและจดจำวัตถุ ซึ่งทำให้หุ่นยนต์สามารถรับรู้และระบุตำแหน่งวัตถุที่ต้องการได้ รวมทั้งประเมินสิ่งที่อยู่ในพื้นที่การทำงานได้เพื่อที่จะสามารถหยิบวัตถุได้อย่างถูกต้องและมีประสิทธิภาพสูงสุด การใช้เทคโนโลยีคอมพิวเตอร์วิทัศน์และปัญญาประดิษฐ์ในการประมวลผลภาพและการจดจำรูปแบบ เพื่อตรวจจับวัตถุและระบุรูปร่าง ลักษณะของวัตถุ ส่งผลให้หุ่นยนต์สามารถดำเนินงานที่ซับซ้อนและยืดหยุ่นได้มากขึ้น ตัวอย่างเช่น การหยิบและวางวัตถุในตำแหน่งที่กำหนดไว้ หรือการหยิบวัตถุในภาชนะที่มีรูปแบบการวางแบบสุ่ม หรือแม้แต่หยิบวัตถุที่มีตำแหน่งที่ไม่แน่นอนและสามารถเปลี่ยนแปลงได้

การใช้งานแขนกลในงานประเภทการหยิบวัตถุในภาชนะหรือที่เรียกว่า bin-picking นั้นถูกใช้งานในอุตสาหกรรมที่อย่างแพร่หลาย โดยเฉพาะในอุตสาหกรรมการผลิต, บรรจุภัณฑ์และระบบการส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนส่งสินค้า ซึ่งความแม่นยำและประสิทธิภาพและเวลาในการทำงานประเภทนี้นั้นมีผลต่อประสิทธิภาพโดยรวมในอุตสาหกรรม ซึ่งการทำงานประเภทหีบวัตฤในภาชนะนั้น มีขั้นตอนการทำงานพื้นฐานโดยเริ่มจากการตรวจจับวัตฤระบบหรือหุ่นยนต์จะใช้เซ็นเซอร์หรือกล้องเพื่อตรวจจับและระบุตำแหน่งของวัตฤภายในภาชนะ ต่อมาคือขั้นตอนการวางแผนการหีบวัตฤ โดยหลังจากที่ตรวจจับวัตฤได้แล้ว ระบบจะวางแผนวิธีการหีบวัตฤออกมาจากภาชนะให้เหมาะสม โดยพิจารณาจากตำแหน่งวัตฤรวมถึงความสามารถในการหีบวัตฤของอุปกรณ์ที่ถูกติดตั้งที่หุ่นยนต์ และขั้นตอนสุดท้ายคือการหีบวัตฤ โดยความท้าทายในงานประเภทนี้คือ การตรวจจับและระบุตำแหน่งของวัตฤภายในภาชนะอาจไม่แม่นยำ หรือการระบุมุมมองของวัตฤที่คลาดเคลื่อนอาจนำมาซึ่งความผิดพลาดในการหีบวัตฤ รวมถึงเวลาในการตรวจจับวัตฤหรือประมวลผลตำแหน่งและมุมมองของวัตฤตลอดจนความเร็วในการคำนวณวิธีการหีบวัตฤนั้น ๆ ซึ่งทั้งหมดที่กล่าวมานี้ส่งผลโดยตรงต่อประสิทธิภาพการทำงานโดยรวมของหุ่นยนต์ หนึ่งในปัจจัยที่สำคัญที่สุดสำหรับงานประเภทนี้คือ เซ็นเซอร์หรือกล้องที่ใช้ในการตรวจจับวัตฤ

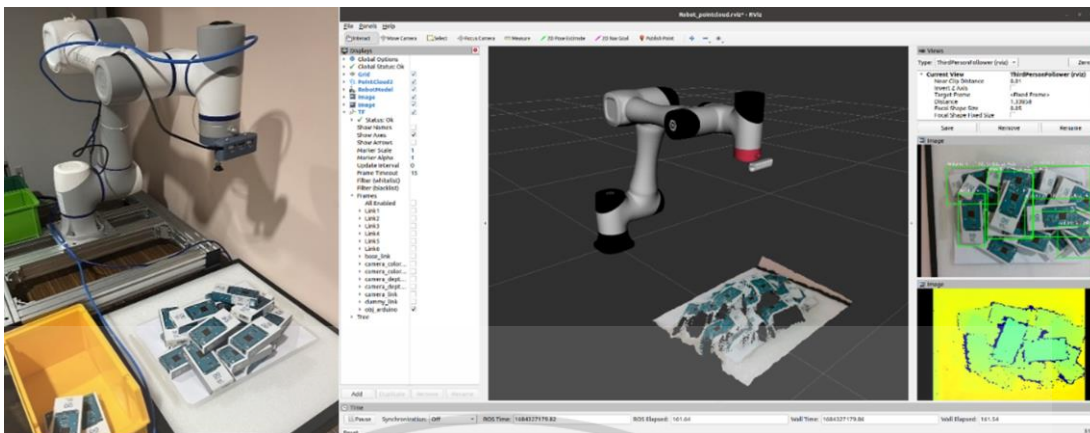
การศึกษาหลายชิ้นรวมถึงการศึกษาที่อ้างถึงในอ้างอิงที่ [1] ได้นำเสนอความก้าวหน้าในด้านการทำงานของหุ่นยนต์ร่วมกลับกล้องและอัลกอริทึมการตรวจจับวัตฤ ในงานวิจัยของ Zhuang และคณะ [1] ได้แสดงการทำงานประเภทหีบวัตฤในภาชนะโดยที่สำคัญได้ทำการสาธิตการทำงานในสภาพแวดล้อมแบบอุตสาหกรรมจริง อย่างไรก็ตาม สิ่งสำคัญคือต้องตระหนักว่าเทคโนโลยีเซ็นเซอร์หรือกล้องที่ใช้ในการศึกษาเหล่านี้ เช่น เครื่องสแกน 3 มิติหรือกล้องเซ็นเซอร์ความลึกขนาดใหญ่ อาจมีต้นทุนสูง โดยเฉพาะอย่างยิ่งในการใช้งานหีบวัตฤในภาชนะที่ใช้ในอุตสาหกรรมซึ่งใช้เซ็นเซอร์ราคาแพงนั้นพบเห็นได้บ่อยเนื่องจากความสามารถในการถ่ายภาพสีและภาพความลึก อีกทั้งการใช้งานเซ็นเซอร์หรือกล้องราคาต่ำมักเผชิญกับข้อจำกัดในแง่ของความละเอียดของภาพและระยะในการจับภาพ ซึ่งประสิทธิภาพนั้นไม่เพียงพอต่อการจัดการงานในพื้นที่ทำงานขนาดใหญ่ ข้อจำกัดเหล่านี้ถูกเปรียบเทียบให้เห็นอย่างชัดเจนในงานศึกษาที่อธิบายไว้ในอ้างอิงที่ [2] มีงานวิจัยหลายงานวิจัยในอดีตที่ได้ศึกษาวิธีการตรวจจับวัตฤและระบุมุมมองโดยที่ใช้เซ็นเซอร์หรือกล้องขนาดเล็กนี้ในการทำงานหีบวัตฤในภาชนะที่วางตัวแบบสุ่ม หนึ่งในนั้นคืองานวิจัยของ Torres และคณะ [3] ซึ่งมุ่งเน้นไปที่การใช้กล้องขนาดเล็กร่วมกับอัลกอริทึมการเรียนรู้เชิงลึกของปัญญาประดิษฐ์ในการตรวจหาวัตฤเฉพาะ เช่น ชิ้นส่วนข้อต่ออากาศยานยนต์ และทำการระบุตำแหน่งและมุมมองของวัตฤแบบ 4 มิติ (ตำแหน่งแกน x , y , z และองศาการหมุนรอบแกน z) อย่างไรก็ตาม แม้ว่าการตรวจจับวัตฤจะแม่นยำ แต่เวลาประมวลผลของอัลกอริทึมในตรวจหาวัตฤและการประเมินมุมมองของวัตฤยังคงใช้เวลานาน อีกหนึ่งงานวิจัยที่น่าสนใจคืองานวิจัยของ Zhuang และคณะ [4] ซึ่งในงานวิจัยนี้แสดงให้เห็นถึงความเร็วของอัลกอริทึมการตรวจหาวัตฤและการประเมินมุมมองของวัตฤที่ประมวลผลได้รวดเร็ว แต่จะมีความแม่นยำในการตรวจจับวัตฤที่ต่ำกว่า ในทางกลับกัน งานวิจัยของ Wong และคณะ [5] ได้ศึกษาอัลกอริทึมการตรวจจับวัตฤและประเมินมุมมองและได้นำเสนอเทคนิคที่มีความแม่นยำในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจจับวัตถุที่สูง แต่ยังพบข้อจำกัดในการประมาณค่าท่าทางของวัตถุที่ ซึ่งยังมีความคลาดเคลื่อนที่สูง เป็นที่น่าสังเกตว่างานศึกษาทั้งหมดที่กล่าวมานี้ ใช้ภาพมุมมองเดียวสำหรับการตรวจจับวัตถุและการประมาณมุมของวัตถุ ซึ่งการประมวลผลเพื่อตรวจหาและระบุตำแหน่งโดยการถ่ายภาพวัตถุในมุมมองเดียวนั้นถือเป็นเรื่องท้าทายเนื่องจากระยะห่างของกล้องจากวัตถุ ส่งผลให้ได้ภาพวัตถุที่มีความละเอียดต่ำนำมาซึ่งความแม่นยำลดลงในตรวจหาวัตถุ ระบุตำแหน่งและประมาณมุมองศาของวัตถุ เพื่อที่จะแก้ไขปัญหานี้ในงานวิจัยฉบับนี้ได้มีการศึกษาเทคนิคที่เรียกว่า eye-in-hand ซึ่งเป็นการติดกล้องที่ตำแหน่งปลายแขนของหุ่นยนต์ การติดตั้งแบบนี้ช่วยให้สามารถเคลื่อนย้ายกล้องได้อย่างอิสระจากการควบคุมแขนของหุ่นยนต์ ช่วยให้สามารถถ่ายภาพจากมุมมองระยะใกล้และไกลได้อย่างไม่มีข้อจำกัด ในงานวิจัยของ Monica และคณะ [6] นำเสนอให้เห็นอย่างชัดเจนถึงข้อดีของการใช้เทคนิค eye-in-hand ซึ่งเพิ่มความสามารถและความแม่นยำในการถ่ายภาพเพื่อใช้วัดความลึกจากเซ็นเซอร์ได้ อย่างไรก็ตามอัลกอริทึมที่ใช้ในการตรวจจับวัตถุในงานวิจัยนี้นั้นยังคงใช้เวลาในการประมวลผลที่ค่อนข้างกินเวลา ดังนั้น ในงานวิจัยฉบับนี้จึงมีวัตถุประสงค์ในการนำเสนอวิธีการตรวจจับวัตถุและระบุตำแหน่งพร้อมทั้งประเมินมุมมององศาของวัตถุแบบ 6 มิติ โดยที่มีความแม่นยำที่สูงและระยะเวลาในการประมวลผลที่รวดเร็ว โดยใช้งานกล้องขนาดเล็กในการถ่ายภาพสีและภาพความลึก เพื่อตอบสนองต่อความต้องการในภาคอุตสาหกรรม ซึ่งไม่เพียงแต่ลดค่าใช้จ่าย แต่ยังทำให้ระบบของที่นำเสนอนี้สามารถใช้งานได้จริงมากขึ้น ซึ่งท้ายที่สุดแล้วจะเป็นการเพิ่มประสิทธิภาพในการทำงานในอุตสาหกรรมอัตโนมัติ โดยผลงานวิจัยนี้จะช่วยให้การจัดการหยิบวัตถุในภาชนะโดยหุ่นยนต์เป็นเทคโนโลยีที่มีความแม่นยำมากยิ่งขึ้น มีความเชื่อถือได้ในการปฏิบัติงาน ลดเวลาที่สูญเสียไปกับการประมวลผลและเพิ่มประสิทธิภาพการทำงานสูงสุด

1.2 วัตถุประสงค์ของการศึกษา

การศึกษานี้มุ่งเน้นที่การแก้ไขปัญหาก็เกี่ยวข้องกับการหยิบวัตถุในภาชนะหรือที่เรียกว่า "bin-picking" เพื่อพัฒนาเทคโนโลยีในการตรวจจับและหยิบวัตถุ โดยเพิ่มความเร็วในการประมวลผลเพิ่มประสิทธิภาพและความแม่นยำของกระบวนการตรวจจับวัตถุและการ ประเมินมุมมององศาของวัตถุ ซึ่งจะทำให้หุ่นยนต์สามารถหยิบวัตถุได้อย่างแม่นยำในตำแหน่งและมุมมององศาที่ถูกต้อง โดยเฉพาะอย่างยิ่งในสภาพแวดล้อมที่ซับซ้อนและเปลี่ยนแปลงได้ เช่น วัตถุที่การวางในลักษณะที่ไม่เป็นระเบียบ มีการวางตำแหน่งและมุมที่ไม่แน่นอน รวมถึงการตรวจจับวัตถุในสภาพแวดล้อมที่เป็นอุปสรรค เช่น สภาพแวดล้อมที่มีแสงน้อยหรือมากเกินไป งานวิจัยจะทำการเปรียบเทียบกับงานวิจัยที่ผ่านมา ในหัวข้อความแม่นยำในการตรวจจับวัตถุ ความแม่นยำในการระบุตำแหน่งและมุมมององศาของวัตถุ และสุดท้ายคือระยะเวลาในการประมวลผลของอัลกอริทึม การเพิ่มประสิทธิภาพของกระบวนการตรวจจับวัตถุเหล่านี้จะช่วยลดความผิดพลาดในการหยิบและวางวัตถุของหุ่นยนต์ นำมาซึ่งประสิทธิภาพโดยรวมของกระบวนการทำงานในอุตสาหกรรมที่สูงขึ้น



รูปที่ 1.1 หุ่นยนต์จริงและการจำลองหุ่นยนต์เชิงโครโมสแบบเรียลไทม์ใน Rviz พร้อมด้วย pointcloud ภาพความลึก และภาพวัตถุที่ตรวจพบ

1.3 สมมุติฐานของการศึกษา

ในงานประเภทการหยิบวัตถุในลักษณะนั้น การที่จะเพิ่มประสิทธิภาพการตรวจจับและระบุตำแหน่งและมุมมองของวัตถุนั้นจำเป็นต้องคำนึงถึงอัลกอริทึมที่ใช้ในการตรวจจับวัตถุและระบุมุมมองของวัตถุ หากทำการแบ่งงานนี้ออกเป็น 2 ส่วน โดยส่วนแรกจะใช้อัลกอริทึมการเรียนรู้เชิงลึกที่มีความยืดหยุ่นและแม่นยำในการตรวจจับวัตถุสูง และส่วนที่สองจะใช้อัลกอริทึมแบบดั้งเดิมที่จำเป็นต้องใช้ภาพวัตถุในระยะใกล้แต่ผลลัพธ์จะสามารถระบุตำแหน่งและมุมมองได้อย่างแม่นยำ เพื่อให้บรรลุเป้าหมายทั้งหมดนี้จึงเลือกใช้วิธีการตรวจจับและประเมินมุมมองของวัตถุสองวิธีต่อไปนี้

1.3.1 You Only Look Once Version 5 หรือ YOLOv5

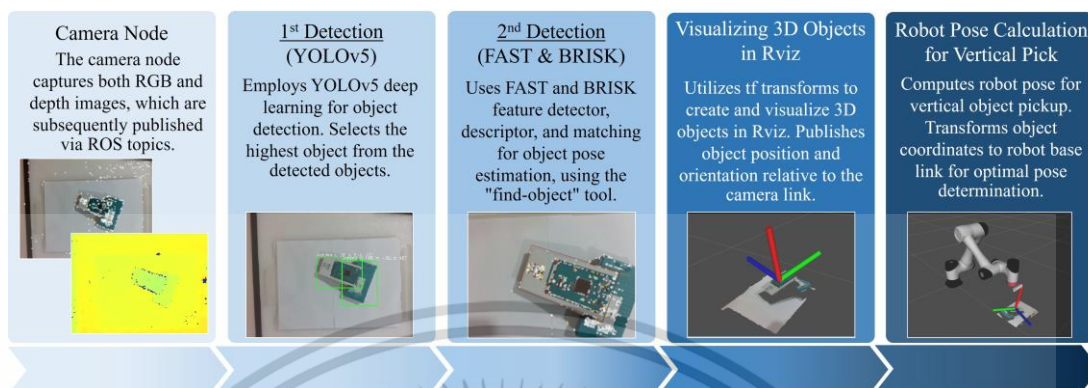
YOLOv5 เป็นอัลกอริทึมการเรียนรู้เชิงลึกที่ทันสมัยสำหรับการตรวจจับวัตถุนั้นเหมาะสำหรับใช้ในการตรวจจับวัตถุในมุมมองกว้างเนื่องจากความสามารถของอัลกอริทึมที่สามารถตรวจจับวัตถุได้ทุกระยะภายในพื้นที่การทำงานของหุ่นยนต์และตรวจจับวัตถุได้ทุกมุมมอง ด้วยการใช้ประโยชน์จาก YOLOv5 วัตถุสามารถตรวจจับได้อย่างรวดเร็ว และหุ่นยนต์สามารถเคลื่อนเข้าใกล้วัตถุเป้าหมายได้อย่างแม่นยำ

1.3.2 อัลกอริทึม Features from Accelerated Segment Test หรือ FAST และอัลกอริทึม Binary Robust Invariant Scalable Keypoints หรือ BRISK

อัลกอริทึม Features from Accelerated Segment Test หรือ FAST และอัลกอริทึม Binary Robust Invariant Scalable Keypoints หรือ BRISK เป็น การตรวจจับคุณลักษณะคำอธิบาย และอัลกอริทึมการจับคู่ ที่ถูกนำมาใช้โดยเมื่อหุ่นยนต์เข้าใกล้วัตถุในระยะที่กำหนด กล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะถ่ายภาพสีและภาพความลึกของวัตถุที่มีความละเอียดที่สูง ภาพความละเอียดสูงนี้เหมาะสำหรับการใช้อัลกอริทึม FAST และ BRISK เพื่อประเมินตำแหน่งและมุมมองของวัตถุอย่างรวดเร็ว



รูปที่ 1.2 การแสดงแผนผังของกระบวนการทำงานของระบบ เริ่มตั้งแต่การรับข้อมูลจากกล้องไปยังขั้นตอนการตรวจจับและท้ายที่สุดไปยังหุ่นยนต์ โดยเน้นขั้นตอนต่าง ๆ ที่เกี่ยวข้องกับการตรวจจับวัตถุ การระบุตำแหน่งและมุมมอง และการคำนวณท่าทางหุ่นยนต์ในการหยิบวัตถุ

1.4 ขอบเขตของการศึกษา

ในงานวิจัยนี้จะใช้งานหุ่นยนต์ร่วมปฏิบัติงาน Dobot CR5 ร่วมกับกล้องเซ็นเซอร์ความลึกราคาประหยัด อย่าง Intel RealSense D435i ร่วมกลับเทคนิค eye-in-hand เพื่อที่จะสามารถถ่ายภาพวัตถุทั้งมุมมองด้านบนที่ครอบคลุมพื้นที่การทำงานทั้งหมด และมุมมองระยะใกล้ของวัตถุที่สนใจ และใช้งานอัลกอริทึม เพื่อให้บรรลุเป้าหมายในการการตรวจจับวัตถุและการประมาณมุมมองของวัตถุอย่างแม่นยำและมีประสิทธิภาพ โดยในส่วนของการทำงานตรวจจับวัตถุด้วยอัลกอริทึม YOLOv5 จะใช้ชุดข้อมูลในการฝึกฝนซึ่งประกอบด้วย

- 1) ชุดข้อมูลภาพถ่ายวัตถุ (กล้อง Arduino) 1 คลาส จำนวน 1431 รูป
- 2) ฝึกฝนด้วยโมเดล YOLOv5l รอบการฝึกฝนทั้งหมด 100 epochs ขนาด batch 16 ต่อรอบการทำงาน

พร้อมทั้งทำการทดลองและเปรียบเทียบผลลัพธ์กับงานวิจัยอื่น ๆ ในอดีต โดยได้ทำการเปรียบเทียบประสิทธิภาพของเทคนิคการตรวจจับทั้งในด้านของความเร็วในการประมวลผล ในด้านของความแม่นยำในการตรวจจับข้อมูล โดยใช้ค่า mAP50 และ mAP50:95 ในการเปรียบเทียบ รวมถึงเปรียบเทียบความแม่นยำในการระบุมุมมองของวัตถุโดยใช้เกณฑ์ความคลาดเคลื่อนที่ต่ำกว่า 5 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนของการศึกษา

ขั้นตอนในการศึกษาเพื่อบรรลุวัตถุประสงค์ที่กล่าวไว้ เริ่มต้นด้วยการศึกษาวิธีการตรวจจับวัตถุด้วยอัลกอริทึมปัญญาประดิษฐ์ต่าง ๆ ซึ่งอาจเป็นการเรียนรู้เชิงลึกหรือการตรวจจับหาคุณลักษณะจากภาพในแบบปัญญาประดิษฐ์แบบดั้งเดิม เพื่อทราบถึงข้อดีและข้อเสีย รวมถึงข้อจำกัดต่างๆ ในการใช้งานและความสามารถหรือผลลัพธ์ที่ได้จากอัลกอริทึมเหล่านั้น เพื่อให้เลือกใช้งานอัลกอริทึมแต่ละตัวให้เหมาะสมกับงานมากที่สุด ต่อมาคือการศึกษาการทำงานของแขนหุ่นยนต์ ไม่ว่าจะเป็นการเคลื่อนที่ ระบบที่ใช้ควบคุม การใช้งานอุปกรณ์เสริมที่ติดตั้งและทำงานร่วมกับหุ่นยนต์จนนำไปสู่การติดตั้งกล่องให้เพื่อที่จะทำงานร่วมกันกับแขนหุ่นยนต์ได้อย่างไร้รอยต่อ

ขั้นตอนถัดไปคือการเชื่อมต่อการทำงานของอัลกอริทึมต่าง ๆ เข้าด้วยกัน ซึ่งจะมีการรับส่งข้อมูลจากกล่องมายังระบบ การใช้อัลกอริทึมการเรียนรู้เชิงลึกในการตรวจจับวัตถุ การตรวจจับหาคุณลักษณะจากภาพในแบบปัญญาประดิษฐ์แบบดั้งเดิมเพื่อระบุมุมมองของวัตถุ และการคำนวณการเคลื่อนที่และควบคุมหุ่นยนต์ ทั้งหมดนี้ทำงานบนระบบปฏิบัติการหุ่นยนต์ Robot Operating System หรือ ROS การศึกษาและประยุกต์ใช้งานระบบปฏิบัติการหุ่นยนต์ ROS เป็นหนึ่งในส่วนสำคัญที่สุดในขั้นตอนการศึกษาของงานวิจัยนี้ เพื่อให้สามารถนำเสนอผลลัพธ์ที่มีความเป็นจริงและมีประสิทธิภาพในการใช้งานในสถานการณ์จริงได้

1.6 แผนการดำเนินงาน

แผนการดำเนินงานคาดการณ์เวลาในการทำวิจัยทั้งสิ้น 29 สัปดาห์ ดังที่นำเสนอในตาราง

ตารางที่ 1.1 แผนการดำเนินงานเดือน สิงหาคม พ.ศ.2566 ถึง ตุลาคม พ.ศ.2566 (ส่วนที่ 1)

ขั้นตอนการดำเนินงาน	สิงหาคม พ.ศ.2566				กันยายน พ.ศ.2566				ตุลาคม พ.ศ.2566			
1. ตั้งขอบเขตเรื่องที่ศึกษาและศึกษาการใช้งานระบบอัตโนมัติในอุตสาหกรรม												
2. เริ่มต้นศึกษางานวิจัยต่าง ๆ ที่เกี่ยวข้องกับหุ่นยนต์ในอุตสาหกรรม												
3. ศึกษาปัญหาและความเป็นไปได้ในการทำการวิจัยเพื่อให้บรรลุวัตถุประสงค์												

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.1 แผนการดำเนินงานเดือน สิงหาคม พ.ศ.2566 ถึง ตุลาคม พ.ศ.2566 (ส่วนที่ 2)

ขั้นตอนการดำเนินงาน	สิงหาคม พ.ศ.2566				กันยายน พ.ศ.2566				ตุลาคม พ.ศ.2566			
4. เริ่มต้นศึกษาเทคนิคต่าง ๆ ที่ถูกใช้งานในอดีต รวมทั้งงานวิจัยในอดีต												
5. ศึกษาและเริ่มต้นใช้งานอุปกรณ์กล้องและแขนหุ่นยนต์												
6. ทดลองหลาย ๆ เทคนิคเพื่อหาเทคนิคที่เหมาะสมที่สุด												
7. คิดค้นวิธีการใหม่และทดลองเพื่อนำผลลัพธ์ที่ได้ไปปรับปรุงแก้ไข												

ตารางที่ 1.2 แผนการดำเนินงานเดือน พฤศจิกายน พ.ศ.2566 ถึง มกราคม พ.ศ.2567

ขั้นตอนการดำเนินงาน	พฤศจิกายน พ.ศ.2566				ธันวาคม พ.ศ.2566				มกราคม พ.ศ.2567			
7. คิดค้นวิธีการใหม่และทดลองเพื่อนำผลลัพธ์ที่ได้ไปปรับปรุงแก้ไข												
8. ปรับปรุงแก้ไขวิธีการที่นำเสนอพร้อมทั้งทำการประเมินผลการทดลอง												
9. เขียนและเรียบเรียงวารสารวิชาการจากวิธีการและเทคนิคที่นำเสนอ												
10. ปรับปรุงแก้ไขบทความวารสารเชิงวิชาการและดำเนินการส่งตีพิมพ์												
11. เขียนและเรียบเรียงวิทยานิพนธ์												

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.3 แผนการดำเนินงานเดือน กุมภาพันธ์ พ.ศ.2567 ถึง เมษายน พ.ศ.2567

ขั้นตอนการดำเนินงาน	กุมภาพันธ์ พ.ศ.2567				มีนาคม พ.ศ.2567				เมษายน พ.ศ.2567			
12. บทความวารสารวิชาการที่นำเสนอได้รับการอนุมัติให้ตีพิมพ์												
13. ปรับแก้เนื้อหาในเล่มวิทยานิพนธ์												
14. บทความวารสารวิชาการถูกตีพิมพ์												
15. ยื่นสอบวิทยานิพนธ์												

1.7 ขั้นตอนและระยะเวลาในการดำเนินงานจริง

ขั้นตอนและระยะเวลาในการดำเนินงานจริงใช้เวลาทั้งสิ้น 36 สัปดาห์ ซึ่งใช้เวลามากกว่าแผนการดำเนินงานที่ประเมินไว้ทั้งหมด 7 สัปดาห์ โดยใช้เวลาในส่วนของงานปรับแก้เนื้อหาในบทความวิชาการที่ส่งตีพิมพ์มากกว่าที่คาดการณ์ไว้ ดังที่นำเสนอในตาราง

ตารางที่ 1.4 การดำเนินงานจริงเดือน สิงหาคม พ.ศ.2566 ถึง ตุลาคม พ.ศ.2566 (ส่วนที่ 1)

ขั้นตอนการดำเนินงาน	สิงหาคม พ.ศ.2566				กันยายน พ.ศ.2566				ตุลาคม พ.ศ.2566			
1. ตั้งขอบเขตเรื่องที่ศึกษาและศึกษาการใช้งานระบบอัตโนมัติในอุตสาหกรรม												
2. เริ่มต้นศึกษางานวิจัยต่าง ๆ ที่เกี่ยวข้องกับหุ่นยนต์ในอุตสาหกรรม												
3. ศึกษาปัญหาและความเป็นไปได้ในการทำการวิจัยเพื่อให้บรรลุวัตถุประสงค์												
4. เริ่มต้นศึกษาเทคนิคต่าง ๆ ที่ถูกใช้งานในอดีตรวมทั้งงานวิจัยในอดีต												

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.4 การดำเนินงานจริงเดือน สิงหาคม พ.ศ.2566 ถึง ตุลาคม พ.ศ.2566 (ส่วนที่ 2)

ขั้นตอนการดำเนินงาน	สิงหาคม พ.ศ.2566				กันยายน พ.ศ.2566				ตุลาคม พ.ศ.2566			
5. ศึกษาและเริ่มต้นใช้งานอุปกรณ์กล้องและแขนหุ่นยนต์												
6. ทดลองหลาย ๆ เทคนิคเพื่อหาเทคนิคที่เหมาะสมที่สุด												

ตารางที่ 1.5 การดำเนินงานจริงเดือน พฤศจิกายน พ.ศ.2566 ถึง มกราคม พ.ศ.2567

ขั้นตอนการดำเนินงาน	พฤศจิกายน พ.ศ.2566				ธันวาคม พ.ศ.2566				มกราคม พ.ศ.2567			
6. ทดลองหลาย ๆ เทคนิคเพื่อหาเทคนิคที่เหมาะสมที่สุด												
7. คิดค้นวิธีการใหม่และทดลองเพื่อนำผลลัพธ์ที่ได้ไปปรับปรุงแก้ไข												
8. ปรับปรุงแก้ไขวิธีการที่นำเสนอพร้อมทั้งทำการประเมินผลการทดลอง												
9. เขียนและเรียบเรียงวารสารเชิงวิชาการจากวิธีการและเทคนิคที่นำเสนอ												
10. ปรับปรุงแก้ไขบทความวารสารเชิงวิชาการและดำเนินการส่งตีพิมพ์												
11. ปรับแก้ภาษาและเพิ่มส่วนเปรียบเทียบผลลัพธ์กับงานวิจัยต่าง ๆ ในอดีต												

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.6 การดำเนินงานจริงเดือน กุมภาพันธ์ พ.ศ.2567 ถึง เมษายน พ.ศ.2567

ขั้นตอนการดำเนินงาน	กุมภาพันธ์ พ.ศ.2567				มีนาคม พ.ศ.2567				เมษายน พ.ศ.2567			
12. เขียนและเรียบเรียงวิทยานิพนธ์												
13. บทความวารสารวิชาการที่นำเสนอได้รับการอนุมัติให้ตีพิมพ์												
14. ปรับแก้และเพิ่มเติมเนื้อหาในเล่มวิทยานิพนธ์												
15. บทความวารสารวิชาการถูกตีพิมพ์												
16. ยื่นสอบวิทยานิพนธ์												

1.8 คำจำกัดความที่ใช้ในการศึกษา

1. ปัญญาประดิษฐ์แบบดั้งเดิม หมายถึงเทคโนโลยีปัญญาประดิษฐ์ก่อนที่จะมีการใช้งานปัญญาประดิษฐ์ประเภทการเรียนรู้เชิงลึกหรือ deep learning
2. ตรวจจับวัตถุและประเมินมุมมองของวัตถุ หมายถึงกระบวนการระบุวัตถุที่สนใจจากภาพโดยใช้งานอัลกอริทึมเฉพาะ และทำการระบุตำแหน่งในแกน x, y และ z และการหมุนรอบแกน x, y และ z ของวัตถุซึ่งจะได้ผลลัพธ์เป็นตำแหน่ง 3 มิติ, 4 มิติ หรือ 6 มิติ ขึ้นอยู่กับระบบและอัลกอริทึมที่ใช้งาน
3. การฝึกฝนโมเดล หมายถึง การป้อนชุดข้อมูลต่าง ๆ เพื่อให้คอมพิวเตอร์เรียนรู้และปรับปรุงโมเดลเพื่อเพิ่มประสิทธิภาพในการตรวจจับวัตถุต่าง ๆ เป็นต้น และเพื่อให้มีความแม่นยำมากยิ่งขึ้น

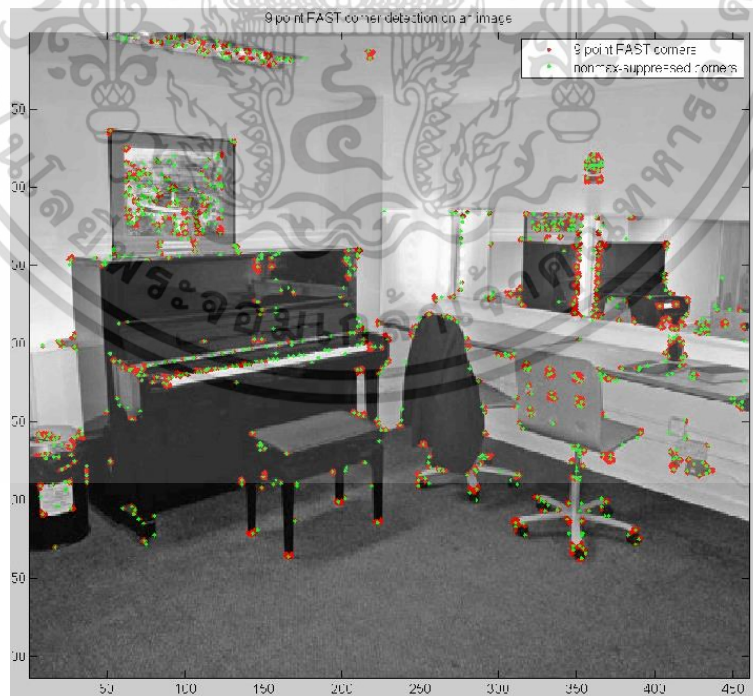
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

งานวิจัยที่เกี่ยวข้อง

2.1 การตรวจจับวัตถุจากภาพด้วยวิธีการหาคุณลักษณะในภาพ

ในด้านการประมวลผลภาพด้วยคอมพิวเตอร์ (computer vision) มีงานวิจัยมากมายที่นำเสนอการตรวจจับคุณลักษณะในภาพและระบุลักษณะเฉพาะของจุดสำคัญในภาพด้วยวิธีต่าง ๆ และอัลกอริทึมในการจับคู่ข้อมูล (feature detection, description, and matching algorithms) หนึ่งในวิธีการที่ได้รับความนิยมมากในอดีตคือ Harris corner detector [7] ซึ่งสามารถระบุมุมของวัตถุในภาพได้โดยการวิเคราะห์ความชันของความเข้มสีในบริเวณใกล้เคียง เทคนิค Scale invariant feature transform หรือ SIFT [8] และ speeded up robust features หรือ SURF [9] เป็นอัลกอริทึมที่โดดเด่นในการตรวจจับจุดสังเกตที่สามารถใช้งานกับภาพและวัตถุที่มีขนาดการหมุนและมุมมองที่ต่างออกไปได้ รวมถึงสามารถระบุลักษณะเฉพาะของจุดสังเกต (descriptor) เพื่อใช้เปรียบเทียบและตรวจจับภาพและวัตถุที่มีขนาดและมุมมองที่ต่างจากภาพอ้างอิงได้ ในทำนองเดียวกัน อัลกอริทึม FAST [10] และ BRISK [11] ได้รับการออกแบบมาเพื่อการตรวจจับคุณลักษณะที่มีประสิทธิภาพ โดย FAST เป็นวิธีการตรวจจับพิกเซลมุมในภาพ ซึ่งสามารถใช้จับจุดเด่นของภาพได้ตามในงานวิจัยที่อ้างอิง [12]



รูปที่ 2.1 ตรวจพบรูปภาพที่มีจุดสังเกต จุดสีเขียวแสดงมุม non-maximally suppressed [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

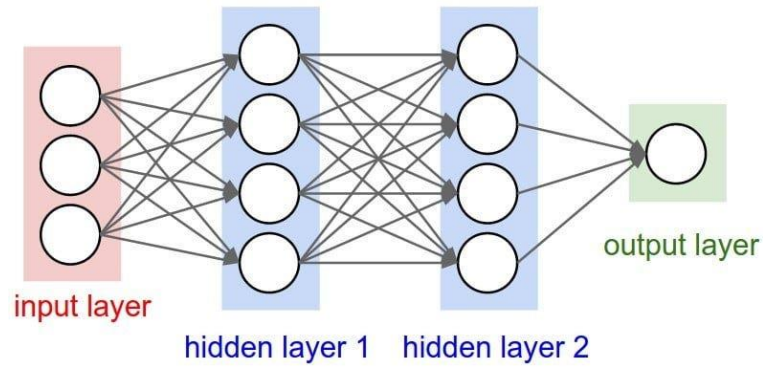
แม้ว่าวิธีการดั้งเดิมเหล่านี้ได้รับการพิสูจน์แล้วว่าประสิทธิภาพ แต่ก็อาศัยคุณสมบัติที่สร้างขึ้นโดยมนุษย์ผู้ใช้งานจึงอาจมองข้ามข้อมูลสำคัญในภาพได้ อีกทั้งยังเผชิญกับข้อจำกัดในการจดจำวัตถุและท่าทางของวัตถุภายใต้สภาวะที่หลากหลายไม่ว่าจะเป็นระยะของวัตถุที่อยู่ห่างมากเกินไปหรืออยู่ใกล้เกินไป หรือวางในมุมมองที่ยากในการตรวจจับเมื่ออยู่ในระยะที่ไม่เหมาะสม เนื่องจากอัลกอริทึมเหล่านี้ต้องใช้ความละเอียดของภาพวัตถุเพื่อให้อยู่ในแนวเดียวกันกับภาพที่ถูกใช้อ้างอิงในการจับคู่คุณลักษณะเพื่อตรวจจับวัตถุ เพื่อให้สามารถตรวจจับวัตถุได้อย่างแม่นยำ อัลกอริทึมการเรียนรู้เชิงลึกได้กลายเป็นเครื่องมือที่เหมาะสมอย่างมากที่ใช้ในการเรียนรู้คุณสมบัติโดยตรงจากข้อมูลช่วยให้การตรวจจับวัตถุมีประสิทธิภาพมากขึ้น แม้ในขนาด มุมการวางของวัตถุ และมุมมองที่หลากหลายก็ตาม

2.2 การเรียนรู้เชิงลึกและโครงข่ายประสาทเทียมแบบคอนโวลูชัน

การเรียนรู้เชิงลึกได้เพิ่มขีดความสามารถของการประมวลผลข้อมูลภาพด้วยอัลกอริทึมคอมพิวเตอร์อย่างมาก ไม่ว่าจะเป็นความสามารถในการตรวจจับวัตถุ หรือความสามารถในการระบุตำแหน่งของวัตถุแบบ 2 มิติ หนึ่งในอัลกอริทึมที่ได้รับความนิยมมากที่สุดคือ โครงข่ายประสาทเทียมแบบคอนโวลูชันหรือ CNN

2.2.1 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม [13] โครงข่ายประสาทเทียม ประกอบด้วยชั้นการดำเนินการหลัก 3 ชั้น ได้แก่ ชั้นข้อมูลนำเข้า (input layer) ชั้นการประมวลผลซ่อน (hidden layer) และชั้นผลลัพธ์ (output layer) โดยในชั้น ข้อมูลนำเข้ามีโหนดรับค่าข้อมูลที่จะทำการประมวลผล ชั้นการประมวลผลซ่อน ประกอบไปด้วยการ เรียงตัวของเพอร์เซปตรอนหลายเพอร์เซปตรอน โดยแต่ละตัวจะเชื่อมกับทุกโหนดในชั้นข้อมูลนำเข้า และเชื่อมกับทุกเพอร์เซปตรอนในชั้นการประมวลผลซ่อนชั้นถัดไป โดยในโครงข่ายประสาทเทียมจะมี ชั้นการประมวลผลซ่อนหลายชั้นได้ขึ้นอยู่กับความต้องการประมวลผล ชั้นผลลัพธ์จะเป็นชั้นสุดท้าย ของโครงข่ายประสาทเทียมโดยจะมีจำนวนโหนดตามรูปแบบของปัญหาที่ต้องการประมวลผล โดย แต่ละโหนดจะเชื่อมกับทุกโหนดในชั้นการประมวลผลซ่อนสุดท้าย แสดงการเชื่อมต่อของโครงข่าย ประสาทเทียมดังภาพที่ 2.2



รูปที่ 2.2 โครงข่ายประสาทเทียมแบบหลายชั้นเพอร์เซปตรอน [13]

2.2.2 โครงข่ายประสาทเทียมคอนโวลูชัน

โครงข่ายประสาทเทียมคอนโวลูชันเป็นหนึ่งในประเภทของโครงข่ายประสาทเทียมที่ใช้วิธีการประมวลผลทางคณิตศาสตร์ที่เรียกว่า คอนโวลูชัน เพื่อประมวลผลข้อมูลที่มีความสัมพันธ์เชิงพื้นที่แบบ grid ได้ดี โครงข่ายประสาทเทียมชนิดนี้ประกอบด้วยเพอร์เซปตรอนหลายตัวที่เรียงต่อกันในหลายชั้น และมีการเชื่อมต่อภายในระหว่างเพอร์เซปตรอนของชั้นที่อยู่ติดกันด้วย โครงสร้างพื้นฐานของโครงข่ายประสาทเทียมคอนโวลูชันประกอบไปด้วยชั้นดำเนินการต่าง ๆ ดังนี้

1. Convolution Layer เป็นชั้นที่มีการดำเนินการคอนโวลูชันประกอบด้วยเมทริกภาพ (Input image) และเมทริกถ่วงน้ำหนัก (weight) หรือที่รู้จักทั่วไปในชื่อ kernel หรือ filter การคอนโวลูชันเริ่มต้นด้วยการนำเมทริกถ่วงน้ำหนักมาวางทับเหนือเมทริกข้อมูลภาพ ค่าข้อมูลในช่องของเมทริกที่ทับกันจะถูกคูณกัน จากนั้นผลคูณแต่ละช่องจะถูกรวมกันเข้าด้วยกันโดยค่าผลรวมที่ได้จะถูกนำไปวางลงในเมทริกใหม่ที่ตำแหน่งกึ่งกลางของเมทริกถ่วงน้ำหนัก เมทริกถ่วงน้ำหนักจะเลื่อนไปยังตำแหน่งพิกเซลถัดไป กระบวนการนี้จะทำซ้ำไปเรื่อย ๆ จนกว่าพิกเซลทั้งหมดในภาพจะได้รับการประมวลผล
2. Activation layer เป็นชั้นดำเนินการของฟังก์ชันกระตุ้น มีบทบาทเสมือนประตูทางผ่านของสัญญาณระหว่างชั้นก่อนหน้าและชั้นถัดไป การควบคุมในชั้นนี้ขึ้นอยู่กับฟังก์ชันทางคณิตศาสตร์ที่ผู้ใช้งานกำหนด เช่น ฟังก์ชัน ReLU (Rectified Linear Unit) ถ้าค่านำเข้ามากกว่าศูนย์จะให้ผลลัพธ์เป็นค่าเดิม แต่ถ้าน้อยกว่าศูนย์จะให้ผลลัพธ์เป็นศูนย์ นี่คือการบวนการที่ช่วยให้โมเดลเรียนรู้และปรับปรุงการทำงานของโครงข่ายได้อย่างเหมาะสมในแต่ละชั้นของมัน
3. Pooling or Sub-sampling layer เป็นชั้นดำเนินการลดขนาดเชิงพื้นที่ด้วยวิธีการสุ่มตัวอย่าง เพื่อลดจำนวนพารามิเตอร์และการคำนวณภายในโครงข่าย รูปแบบการดำเนินการของชั้น Pooling ใช้วิธีการวางเมทริกสุ่มตัวอย่าง เช่น max pooling

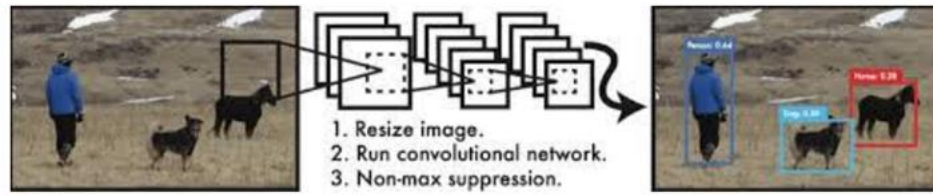
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ average pooling เป็นต้น โดยใน max pooling ค่าที่มากที่สุดในแต่ละเมทริกจะถูกเลือกเป็นตัวแทนข้อมูล ส่วนใน average pooling ค่าเฉลี่ยของข้อมูลในแต่ละเมทริกจะถูกใช้เป็นตัวแทนข้อมูล ขนาดของเมทริกที่สุ่มตัวอย่างมีหลายขนาด เช่น ขนาด 3×3 และมี stride เท่ากับ 2 หรือ ขนาด 2×2 โดยใช้ stride เท่ากับ 2 เป็นต้น การใช้ Pooling ช่วยลดขนาดของข้อมูลที่เข้ามาในชั้นถัดไปในโครงข่าย ซึ่งช่วยลดการคำนวณและช่วยในการควบคุมการเรียนรู้ของโมเดลได้อย่างมีประสิทธิภาพ

4. Fully Connected layer เป็นชั้นที่มาก่อนชั้นสุดท้ายในโครงข่าย มันเชื่อมต่อกับฟังก์ชันกระตุ้นทั้งหมดในชั้นก่อนหน้า และถูกเชื่อมต่อกับชั้นสุดท้ายด้วยชั้น Softmax เพื่อแสดงผลความน่าจะเป็นของข้อมูลขาออก และขยายค่าข้อมูลผลลัพธ์จากการดำเนินการคอนโวลูชันให้มีขนาดเท่ากับขนาดภาพนำเข้าเริ่มต้น และใช้ชั้น Softmax Activation Function ในชั้นสุดท้ายของโครงข่ายเพื่อทำนายความน่าจะเป็นของแต่ละพิกเซลตามจำนวนคลาสหรือช่องที่ต้องการ Batch Normalization layer เป็นชั้นที่มีการดำเนินการทางคณิตศาสตร์เพื่อปรับข้อมูลให้มีมาตรฐาน โดยการปรับจะรักษาค่าเฉลี่ยของข้อมูลขาออกให้เข้าใกล้ 0 และค่าเบี่ยงเบนมาตรฐานของข้อมูลขาออกให้เข้าใกล้ 1 ซึ่งช่วยให้การเรียนรู้ของโมเดลเป็นไปอย่างมีประสิทธิภาพและเสถียรมากขึ้น
5. Merging layer เป็นชั้นที่ดำเนินการผสมผลลัพธ์ระหว่างชั้นดำเนินการสองชั้นหรือมากกว่าเข้าด้วยกัน โดยรับข้อมูลนำเข้าในรูปแบบรายการของ tensor ที่มีรูปร่างเหมือนกันทั้งหมด และส่งคืนผลลัพธ์เป็นค่า tensor ตัวเดียวที่เป็นการผสมกันของข้อมูลที่ถูกนำเข้ามาผสมกันทั้งหมด โดยใช้วิธีการดำเนินการของชั้นต่าง ๆ เช่น Concatenate layer เป็นชั้นการเชื่อมข้อมูลนำเข้าในรูปแบบรายการของ tensor, Add layer เป็นชั้นการรวมข้อมูลนำเข้าในรูปแบบรายการของ tensor, Multiply layer เป็นชั้นการคูณ (element-wise) ข้อมูลนำเข้าในรูปแบบรายการของ tensor เป็นต้น

2.2.3 อัลกอริทึม You Only Look Once หรือ YOLO

YOLO หรือ You Only Look Once [14] เป็นตระกูลของแบบจำลองการตรวจจับวัตถุตามเวลาจริง แนวคิด YOLO เปิดตัวครั้งแรกในปี 2016 และเป็นที่ยอมรับเป็นอย่างมากเนื่องจากความสามารถในการตรวจจับที่รวดเร็วและแม่นยำกว่าอัลกอริทึมการตรวจจับวัตถุที่มีอยู่ในขณะนั้น



รูปที่ 2.3 ขั้นตอนการทำงานเบื้องต้นในการประมวลผลภาพด้วย YOLO [14]

แนวคิดพื้นฐานที่อัลกอริทึม YOLO เสนอคือการใช้โครงข่ายประสาทแบบ end-to-end โดยใช้ bounding box & class probability เพื่อคาดการณ์แบบเรียลไทม์ YOLO แตกต่างจากโมเดลการตรวจจับวัตถุก่อนหน้านี้ในแง่แนวทางที่แตกต่างกันในการตรวจจับวัตถุ โดยการนำตัวแยกประเภทกลับมาใช้ใหม่ เมื่อเปรียบเทียบกับอัลกอริทึมของ YOLO การตรวจจับวัตถุในตอนนั้นใช้เทคนิคแบบ heat map เพื่อตรวจจับตำแหน่งที่น่าสนใจ กระบวนการทำงานนั้นดำเนินการในแต่ละจุดแยกกัน ด้วยเหตุนี้โมเดลเหล่านี้จึงมีความซ้ำซ้อนในการตรวจจับหลายครั้งในภาพเดียวกัน จึงทำให้ขาดความแม่นยำและใช้เวลาในการดำเนินการสูงขึ้น ในทางกลับกันอัลกอริทึม YOLO จะทำการตรวจจับในครั้งเดียวด้วยโมเดลที่ผ่านการฝึกฝนมาก่อนหน้านี้แล้ว จึงมีความแม่นยำที่เหนือกว่า และเวลาในการดำเนินงานที่ต่ำกว่า

2.3 ระบบปฏิบัติการหุ่นยนต์และโปรแกรมแสดงผลภาพ

ระบบปฏิบัติการหุ่นยนต์ หรือ ROS มีส่วนสำคัญอย่างมากในเทคโนโลยีด้านวิทยาการหุ่นยนต์โดยการระบบของ ROS นั้นอำนวยความสะดวกในการสื่อสารและการทำงานร่วมกันระหว่างอุปกรณ์หรือโปรแกรมการทำงานต่าง ๆ ในการศึกษาที่นำมาอ้างอิง [15] ได้นำเสนอหลักการพื้นฐานและการทำงานของระบบ ROS และเปิดให้ผู้ใช้สามารถใช้งานระบบ ROS อย่างอิสระเป็นแบบ open source ทำให้ระบบปฏิบัติการหุ่นยนต์ ROS ถูกนำไปใช้อย่างกว้างขวางและมีการพัฒนาต่อยอดอย่างก้าวหน้าทั้งในวงการวิทยาการหุ่นยนต์และวงการด้านเทคโนโลยีอื่น ๆ ในศึกษานี้ได้ใช้ประโยชน์จากความสามารถของระบบปฏิบัติการหุ่นยนต์ ROS เพื่อรวมการทำงานของกล้องเข้ากับส่วนประมวลผลบนคอมพิวเตอร์และการควบคุมหุ่นยนต์เข้าด้วยกัน ช่วยให้สามารถศึกษาและพัฒนาการประยุกต์ใช้หุ่นยนต์ได้อย่างสะดวกรวดเร็ว โปรแกรมแสดงผลภาพบนแพลตฟอร์มของระบบ ROS หรือที่เรียกว่า Rviz มีความโดดเด่นในแง่ของโปรแกรมสร้างและแสดงผลภาพ 3 มิติ ซึ่งโปรแกรม Rviz นั้นมีประสิทธิภาพในการสร้างภาพ, โมเดลหุ่นยนต์, ภาพ 3 มิติหรือ pointcloud รวมถึงแสดงผลภาพและโมเดลหุ่นยนต์และกล้องได้แบบเรียลไทม์ ในงานศึกษาของ Kam และคณะ [16] ได้นำเสนอการประยุกต์ใช้งานของโปรแกรม Rviz ในการใช้งานร่วมกับหุ่นยนต์ที่ปฏิบัติงานและแสดงผลโมเดลอย่างต่อเนื่อง งานศึกษาของพวกเขาแสดงให้เห็นถึงบทบาทในการสร้างภาพหุ่นยนต์ 3 มิติและเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผลแบบเรียลไทม์ที่มีความต่อเนื่องและใช้งานได้จริง ในวิทยานิพนธ์ฉบับนี้ได้นำเสนอการใช้งานโปรแกรม Rviz เพื่อแสดงการเคลื่อนที่ของหุ่นยนต์แบบต่อเนื่องตามแผนหุ่นยนต์จริง รวมทั้งแสดงภาพ 3 มิติของสภาพแวดล้อมในพื้นที่การทำงานของหุ่นยนต์ที่สามารถถ่ายได้ด้วยกล้องถ่ายภาพสีและความลึก รวมถึงแสดงภาพการตรวจจับวัตถุจากอัลกอริทึมการตรวจจับ และแสดงตำแหน่งและมุมมองของวัตถุแบบ 6 มิติ โดยอ้างอิงตามตำแหน่งของหุ่นยนต์และกล้องที่ติดตั้งที่หุ่นยนต์ นำเสนอการทำงานของหุ่นยนต์ในได้อย่างชัดเจน แม่นยำและมีประสิทธิภาพสูงสุด

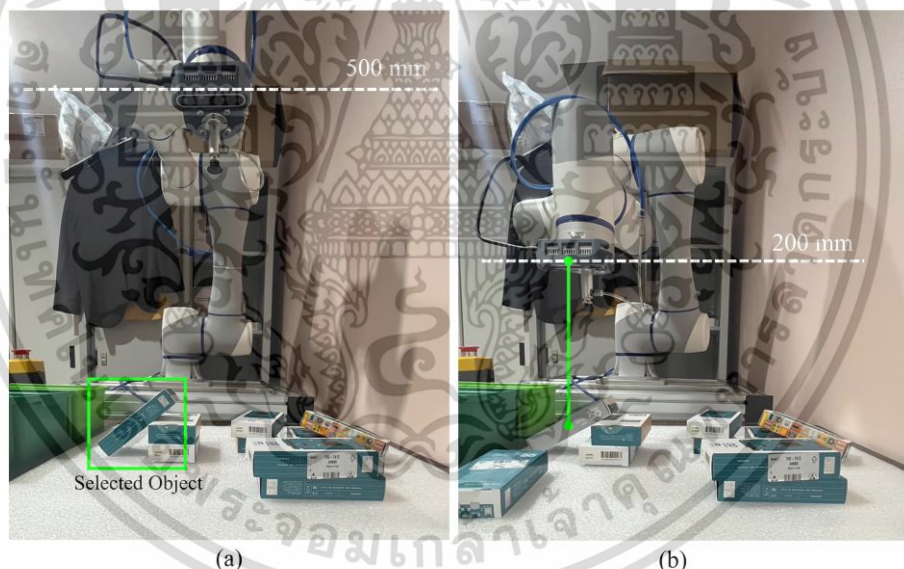


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การตรวจจับวัตถุและประเมินตำแหน่งและมุมมองศา

ในการตรวจจับวัตถุและประเมินท่าทาง ระบบจะใช้เทคนิคการตรวจจับสองแบบ ดังแสดงในรูปที่ 1.2 ประการแรก หุ่นยนต์วางตำแหน่งตัวเองในมุมมองด้านบน ช่วยให้กล้องสามารถจับภาพพื้นที่ทำงานทั้งหมดเพื่อความครอบคลุมที่ครอบคลุม ภาพมุมมองด้านบนนี้ได้รับการประมวลผลโดยโหนดการตรวจจับแรก ซึ่งใช้โมเดล YOLOv5 ที่ได้รับการฝึกอบรมสำหรับการตรวจจับวัตถุ โดยการเปรียบเทียบตำแหน่งของวัตถุที่ตรวจพบทั้งหมด อัลกอริทึมจะเลือกวัตถุที่มีตำแหน่งแนวตั้งสูงสุดต่อนั้น หุ่นยนต์จะเคลื่อนเข้าใกล้วัตถุที่เลือกมากขึ้นในขณะที่รักษาระยะห่างคงที่ไว้ที่ 200 มม. เหนือวัตถุ ดังแสดงในรูปที่ 3.1 ในระยะนี้ช่วยให้สามารถถ่ายภาพที่ได้ข้อมูลที่มีความรายละเอียด จากนั้นภาพระยะใกล้ของวัตถุจะถูกส่งต่อไปยังโหนดการตรวจจับที่สอง โหนดนี้ใช้อัลกอริทึมในการตรวจจับวัตถุและประเมินตำแหน่งตามแกน x, y และ z และมุมมองศาในแนวแกน x, y และ z ในพื้นที่การทำงานโดยอ้างอิงตามตำแหน่งของกล้อง

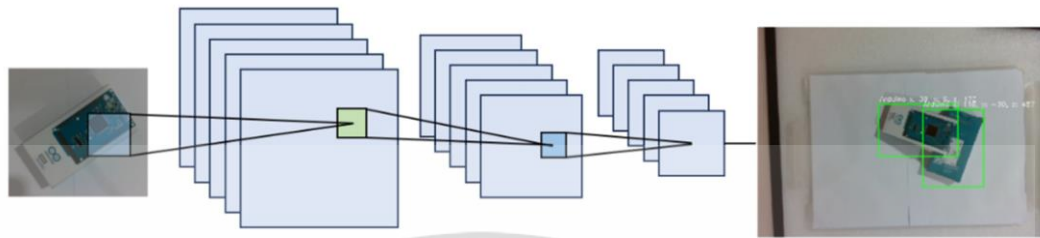


รูปที่ 3.1 (a) ตำแหน่งมุมมองด้านบนของหุ่นยนต์ (b) ตำแหน่งของหุ่นยนต์ในขณะที่เคลื่อนที่เข้าใกล้วัตถุที่เลือกมากขึ้น

3.1 อัลกอริทึมแรกสำหรับการตรวจจับวัตถุ (YOLOv5)

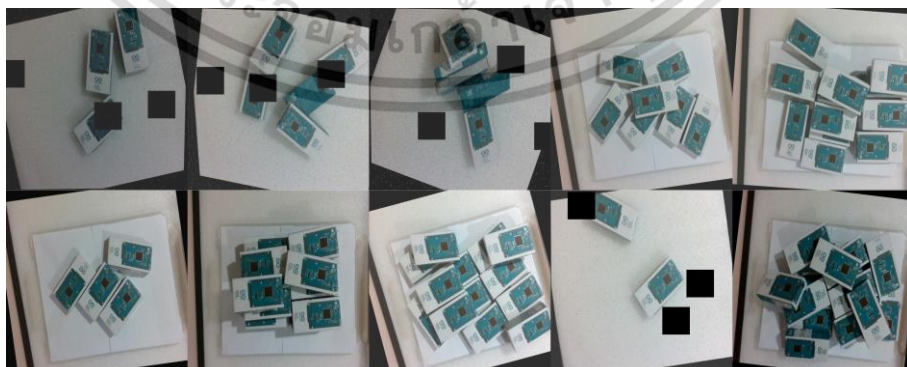
เทคนิคแรก YOLOv5 คือ อัลกอริทึมที่ใช้ในการตรวจจับวัตถุที่โดยใช้งานโครงข่ายประสาทเทียมแบบคอนโวลูชันหรือ CNN การเรียนรู้เชิงลึก YOLOv5 มีตัวอย่างขั้นตอนการทำงานถูกแสดงในรูปที่ 3.2 โมเดลของ YOLOv5 นั้นได้รับการยอมรับอย่างกว้างขวางว่ามีความแม่นยำสูงและเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพการตรวจจับแบบเรียลไทม์ ด้วยการทำนายและระบุตำแหน่งของวัตถุ พร้อมทั้งประเมินความน่าจะเป็นของคลาสสำหรับวัตถุหลายชิ้นภายในรูปภาพที่ตรวจจับไปพร้อม ๆ กัน YOLOv5 จึงสามารถตรวจจับวัตถุได้อย่างมีประสิทธิภาพและเหมาะสำหรับการใช้งาน



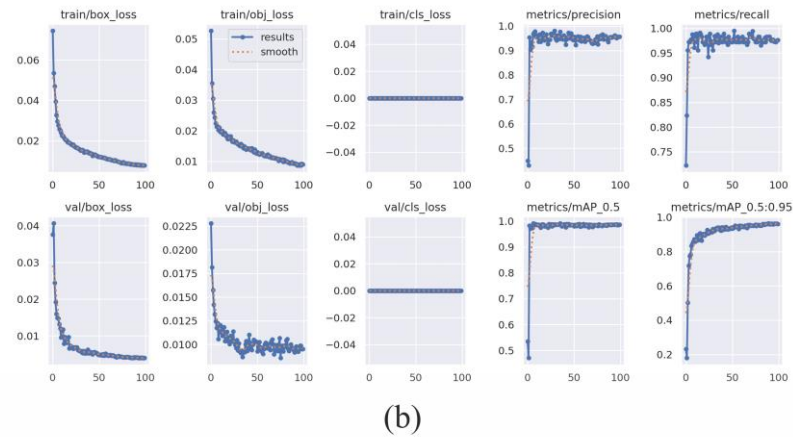
รูปที่ 3.2 หลักทำงานเบื้องต้นของอัลกอริทึม CNN ที่เป็นเทคนิคที่ใช้ในการตรวจจับวัตถุของอัลกอริทึม YOLOv5

เพื่อฝึกฝนโมเดล YOLOv5 ผู้วิจัยจะทำการเตรียมชุดข้อมูลที่ใช้ในการฝึกฝนโมเดลด้วยโปรแกรม Roboflow การใช้งานโปรแกรมนี้จะใช้สำหรับการ label ข้อมูลสำหรับฝึกฝนทั้งหมด ต่อมาจะทำการเพิ่มความหลากหลายให้กับข้อมูล เพื่อให้โมเดลมีความยืดหยุ่นมากยิ่งขึ้นเมื่อนำไปใช้งานจริง โดยการเพิ่มความหลากหลายของข้อมูลนั้นจะใช้เทคนิค argumentation จากโปรแกรม Roboflow โดยผ่านอัลกอริทึมสำหรับการเตรียมชุดข้อมูล ผลลัพธ์ที่ได้จากการเพิ่มความหลากหลายของข้อมูลนี้ถูกนำเสนอผ่านรูปที่ 3.3 (a) ทำให้โมเดลที่ได้จากการฝึกฝนด้วยชุดข้อมูลนี้มีคุณภาพมากยิ่งขึ้น ในกระบวนการฝึกฝนโมเดล YOLOv5l จะทำการฝึกฝนโมเดลวนซ้ำไปเรื่อย ๆ ทั้งหมด 100 ครั้ง หรือ 100 epoch โดยแต่ละครั้งจะใช้ขนาดของ batch เท่ากับ 16 ซึ่งชุดข้อมูลที่ใช้ในการฝึกฝนโมเดลจะประกอบด้วยภาพทั้งหมด 1431 ภาพ ตัวอย่างชุดข้อมูลการฝึกด้วยเทคนิคเสริม และผลลัพธ์ของการฝึกแสดงในรูปที่ 3.3 (b)



(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 (a) ชุดข้อมูลการฝึกอบรมด้วยเทคนิคการเสริม (b) ผลลัพธ์การฝึกอบรมโดยใช้อัลกอริทึม YOLOv5L

ตารางที่ 3.1 ผลลัพธ์การฝึกอบรมการตรวจจับวัตถุ

อัลกอริทึม	Precision	recall	mAP50	mAP50-95
YOLOv5L	0.955	0.983	0.986	0.965

ตารางที่ 3.1 แสดงผลลัพธ์ที่ได้จากการฝึกฝนโมเดลตรวจจับวัตถุด้วยอัลกอริทึม YOLOv5L ซึ่งแสดงตัวชี้วัดค่าความแม่นยำแบบต่าง ๆ ซึ่งผลลัพธ์ที่ได้นั้นแสดงให้เห็นถึงความสามารถในการตรวจจับวัตถุที่แม่นยำ รวมถึงดัชนีความแม่นยำ mAP50 และ mAP50-95 ที่น่าพอใจ ตัวชี้วัดเหล่านี้สะท้อนถึงความสามารถการตรวจจับวัตถุทั้งหมดภายในพื้นที่ทำงานเมื่อหุ่นยนต์อยู่ในตำแหน่งมุมมองด้านบน ในสถานการณ์ที่ตรวจพบวัตถุหลายชิ้นระบบจะเลือกวัตถุที่มีตำแหน่งอยู่สูงที่สุด โดยพิจารณาจากข้อมูลภาพความลึกที่ได้จากกล้อง หลังจากนั้นหุ่นยนต์จะถูกสั่งให้เคลื่อนที่เห็นวัตถุที่สนใจ โดยรักษาระยะห่างระหว่างกล้องและวัตถุแบบคงที่ จากนั้นจึงใช้เทคนิคการตรวจจับที่สองเพื่อประเมินตำแหน่งและมุมมองของวัตถุโดยใช้ภาพถ่ายวัตถุในระยะใกล้

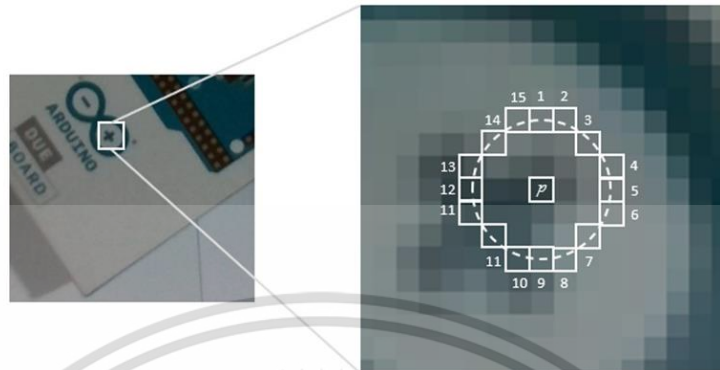
3.2 อัลกอริทึมที่สองสำหรับการตรวจจับวัตถุ (FAST and BRISK)

นอกจากอัลกอริทึม YOLOv5 แล้ว เทคนิคที่สองที่ใช้ในการวิจัยนี้คือการผสมผสานระหว่าง FAST เป็นตัวตรวจจับคุณลักษณะ และ BRISK เป็นตัวอธิบายคุณลักษณะ โดยจะใช้ทั้ง 2 เทคนิคนี้สำหรับการระบุตำแหน่งของวัตถุและการประเมินมุมมองของวัตถุ

3.2.1 อัลกอริทึมการตรวจจับคุณลักษณะ FAST

การตรวจจับคุณลักษณะเป็นเทคนิคการตรวจจับวัตถุด้วยคอมพิวเตอร์ขั้นที่จะระบุจุดที่โดดเด่นในภาพเพื่อการวิเคราะห์เพิ่มเติม เทคนิค FAST ใช้อัลกอริทึมที่ออกแบบมาเพื่อตรวจหาพิกเซลมุมจากวัตถุหรือภาพ โดยเทคนิค FAST จะเปรียบเทียบความเข้มของพิกเซลในบริเวณใกล้เคียงโดยจะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างเป็นวงกลมรอบๆ จุดที่พิจารณา โดยจะเปรียบเทียบว่าพิกเซลที่ใกล้เคียงนั้นมีความเข้มของแสงแตกต่างกันเพียงพอหรือไม่



รูปที่ 3.4 แผนผังของจุดสำคัญที่กำลังประเมิน พร้อมด้วยพิกเซล 16 พิกเซลโดยรอบที่จัดเรียงเป็นรูปวงกลม

อัลกอริทึม FAST จะสร้างวงกลมรอบจุดที่กำลังพิจารณาที่ประกอบด้วย 16 พิกเซล โดยกำกับตั้งแต่ 1 ถึง 16 ดังแสดงในรูปที่ 3.4 และทำการเปรียบเทียบความเข้มของพิกเซลรอบข้างกับพิกเซลที่พิจารณา โดยค่าความเข้มแสงนี้จะถูกแทนด้วยตัวแปร I ค่าเกณฑ์ความแตกต่างของความเข้มแสงที่กำหนดจะแทนด้วย t ซึ่งอัลกอริทึม FAST จะใช้เงื่อนไขต่อไปนี้สำหรับการจัดพิจารณาหาพิกเซลมุมดังนี้

- เงื่อนไขที่ 1: หากชุดของ (N) พิกเซลที่อยู่ติดกัน (S) ภายในวงกลมเป็นไปตามเงื่อนไข
- $$\forall x \in S : I_x > I_p + t$$
- พิกเซลที่สนใจนี้จะถูกจัดเป็นจุดสังเกต
- เงื่อนไขที่ 2: หากชุดของ (N) พิกเซลที่อยู่ติดกัน (S) ภายในวงกลมเป็นไปตามเงื่อนไข
- $$\forall x \in S : I_x < I_p - t$$
- พิกเซลที่สนใจนี้จะถูกจัดเป็นจุดสังเกต

การเลือกค่าที่เหมาะสมสำหรับค่าพารามิเตอร์ N และเกณฑ์ (t) ที่ใช้เปรียบเทียบค่าความเข้มแสง จะแปรผันตรงกับจำนวนจุดมุมที่ตรวจพบและประสิทธิภาพของเทคนิค โดยทั่วไปแล้วค่าพารามิเตอร์ N จะถูกตั้งค่าเป็น 12 นอกจากนี้ยังมีเทคนิคอื่น ๆ ที่จะช่วยเพิ่มความเร็วการตรวจจับวัตถุของเทคนิคนี้อีกด้วย ซึ่งจะช่วยปรับปรุงประสิทธิภาพโดยรวมของเทคนิคให้ดียิ่งขึ้น กระบวนการเปรียบเทียบความเข้มแสงที่รวดเร็วของ FAST ช่วยให้สามารถตรวจจับจุดสำคัญที่มุมของวัตถุแบบเรียลไทม์ได้อย่างแม่นยำและมีประสิทธิภาพสูง

3.2.2 อัลกอริทึมการจดจำคุณลักษณะ BRISK

เมื่อตรวจพบจุดสำคัญบนภาพแล้ว จำเป็นต้องเลือกจดจำคุณลักษณะของจุดนั้น ๆ ด้วยเทคนิคที่มีประโยชน์สำหรับการตรวจจับวัตถุ อัลกอริทึมการจดจำคุณลักษณะจะเข้ามามีบทบาทในขั้นตอนนี้ ในการวิจัยนี้ใช้งานอัลกอริทึม BRISK ในการจดจำคุณลักษณะ อัลกอริทึมนี้จะระบุคุณลักษณะเฉพาะของจุดสำคัญโดยการเข้ารหัสข้อมูลลักษณะ พื้นผิว และรูปร่าง ในรูปแบบไบนารีขนาดเล็ก ข้อมูลแบบไบนารีนี้มีความทนทานต่อการแปลงรูปภาพเมื่อวัตถุในภาพอยู่ในตำแหน่งหรือมุมมองที่ต่างไปจากเดิม เพิ่มความยืดหยุ่นในการตรวจจับวัตถุ และยังช่วยให้การจับคู่จุดสำคัญมีประสิทธิภาพมากขึ้น ซึ่งทำให้ความแม่นยำและความน่าเชื่อถือโดยรวมของเทคนิคการตรวจจับคุณลักษณะนั้นสูงยิ่งขึ้น ส่งผลให้กระบวนการระบุตำแหน่งและการประเมินค่ามุมมองของวัตถุมีความแม่นยำมากยิ่งขึ้น ลดความผิดพลาดของการปฏิบัติงานของหุ่นยนต์

3.2.3 อัลกอริทึมการจับคู่คุณลักษณะ

เทคนิคการจับคู่คุณสมบัตินี้ใช้เพื่อสร้างความสอดคล้องระหว่างจุดสำคัญในรูปภาพอ้างอิงและรูปภาพที่ใช้ค้นหาวัตถุ โดยจะใช้งานอัลกอริทึม nearest neighbor เพื่อช่วยในการจับคู่ โดยจะมีการเปรียบเทียบคุณลักษณะเพื่อค้นหาคู่ที่เหมาะสม จากนั้นจึงใช้เทคนิคการประมาณค่าแบบโฮโมกราฟีเพื่อกำหนดการเปลี่ยนแปลงทางของมุมมองของวัตถุระหว่างทั้ง 2 ภาพ การคำนวณนี้จะคำนวณตำแหน่งของวัตถุได้อย่างแม่นยำ รวมถึงสามารถประเมินมุมมองของวัตถุได้ด้วย

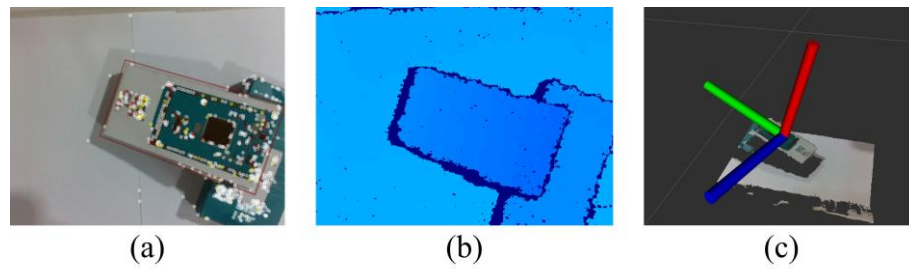
ด้วยการผสมรวมเทคนิค FAST และ BRISK กระบวนการตรวจจับวัตถุนี้จะสามารถระบุตำแหน่งของวัตถุและประเมินมุมมองของวัตถุได้อย่างถูกต้อง การใช้งานโปรแกรม find object [17] ซึ่งภายในโปรแกรมนั้นประกอบด้วยอัลกอริทึมการตรวจจับคุณลักษณะ เช่น SIFT, FAST, ORB และอัลกอริทึมการจดจำคุณลักษณะ ส่งผลให้สามารถใช้งานอัลกอริทึมในการตรวจจับวัตถุได้อย่างมีประสิทธิภาพในภาพ และตรวจจับวัตถุได้ด้วยความเร็วแบบเรียลไทม์ โปรแกรม find object มีการใช้งานเทคนิค เช่น RANSAC และฟังก์ชันโฮโมกราฟี เพื่อช่วยในการระบุตำแหน่งของวัตถุและคุณมุมมองที่เปลี่ยนไปของวัตถุจากภาพ แม้ในสถานการณ์ที่วัตถุมีการเปลี่ยนแปลงของมุมมองที่แตกต่างจากภาพอ้างอิงอย่างมากก็ตาม

ด้วยการใช้เทคนิคเหล่านี้ กระบวนการตรวจจับวัตถุจะสามารถบรรลุเป้าหมายตามวัตถุประสงค์ การใช้งานโปรแกรมและอัลกอริทึมเหล่านี้ร่วมกันได้อย่างราบรื่นนั้น จะทำการใช้งานทุกโปรแกรมภายในระบบปฏิบัติการหุ่นยนต์ ROS เพื่อที่จะมั่นใจได้ว่าการเชื่อมต่อของข้อมูลระหว่างโปรแกรมจะเป็นไปอย่างราบรื่นและรวดเร็ว ทำให้ระบบการตรวจจับวัตถุและประเมินมุมมองของวัตถุทำงานร่วมกับหุ่นยนต์นั้นจะสามารถบรรลุเป้าหมายได้ นอกจากนั้นยังสามารถแสดงผลการทำงานของตรวจจับวัตถุ โดยแสดงถึงกวัตตแบบ 6 มิติ ดังรูปที่ 3.5 ที่แสดงผ่านโปรแกรม Rviz ซึ่งตำแหน่งของวัตถุจะถูกใช้สำหรับการการอ้างอิงตำแหน่งของหุ่นยนต์ที่ใช้ในการหยิบวัตถุ โดยรับส่งข้อมูลภายใน

ระบบปฏิบัติการหุ่นยนต์ ROS

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

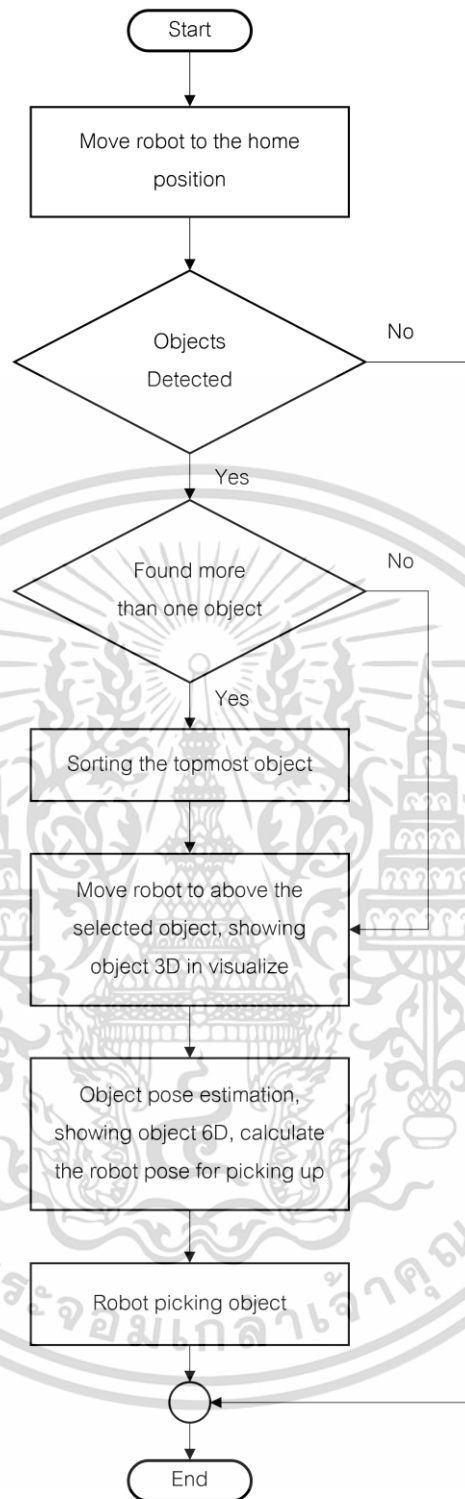
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 (a) ภาพ RGB พร้อมการตรวจจับคุณสมบัติจาก FAST (b) ภาพความถี่แสดงเป็นแผนที่ความร้อน (c) ลิงก์วัตถุที่มองเห็นใน Rviz

3.3 ขั้นตอนการทำงานของโปรแกรมและการใช้งานระบบปฏิบัติการหุ่นยนต์ ROS

การศึกษานี้ใช้ประโยชน์จากระบบปฏิบัติการหุ่นยนต์ ROS เพื่อรวมการทำงานของอุปกรณ์ต่าง ๆ เข้าด้วยกันตั้งแต่ กล้อง ส่วนประมวลผลบนคอมพิวเตอร์ ตลอดจนเซนเซอร์หุ่นยนต์เข้าด้วยกัน ทำให้สามารถรับส่งข้อมูลและคำสั่งได้อย่างสะดวกรวดเร็ว และทำการแสดงผลกับผู้ใช้งานด้วยโปรแกรมแสดงภาพแบบเรียลไทม์ ซึ่งกระบวนการทำงานหลักของระบบนั้นถูกนำเสนอผ่านแผนภาพการทำงานในรูปที่ 3.6 โดยเริ่มตั้งแต่การถ่ายภาพวัตถุ ตรวจจับและคำนวณ ตลอดจนการหยิบวัตถุด้วยเซนเซอร์หุ่นยนต์



รูปที่ 3.6 ขั้นตอนกระบวนการทำงานของระบบในการถ่ายภาพวัตถุ การตรวจจับและเลือกวัตถุ
คำนวณมุมมองของวัตถุและการหยิบวัตถุ

การทำงานในแต่ละขั้นตอนนั้นแบ่งออกเป็น 8 ขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ขั้นตอนที่ 1 หุ่นยนต์จะเคลื่อนที่ไปประจำตำแหน่งที่ตำแหน่งโฮม หรือเป็นตำแหน่งที่สามารถมองเห็นวัตถุได้ครอบคลุมทั้งหมด
2. ขั้นตอนที่ 2 จะใช้งานเซนเซอร์กล้องและเซนเซอร์วัดความลึกที่อยู่ในกล้อง ในการถ่ายภาพสีและภาพความลึก และส่งข้อมูลดังกล่าวไปประมวลผลเพื่ออยู่ในรูปของข้อมูลที่สามารถใช้งานอัลกอริทึมในการตรวจจับวัตถุจากรูปภาพได้ และทำการส่งข้อมูลไปยังส่วนประมวลผลเพื่อทำการตรวจจับวัตถุด้วยอัลกอริทึม YOLOv5 เมื่อทำการใช้งานอัลกอริทึมแล้วจะตรวจสอบว่าพบวัตถุหรือไม่
3. ขั้นตอนที่ 3 เมื่อตรวจจับวัตถุพบแล้วจะทำการตรวจสอบว่าพบจำนวนวัตถุมากกว่าหนึ่งหรือไม่
4. ขั้นตอนที่ 4 หากพบวัตถุมากกว่าหนึ่งวัตถุในภาพ จะทำการเปรียบเทียบระยะวัตถุของแต่ละวัตถุเพื่อระบุวัตถุที่อยู่ในระดับที่สูงที่สุดแล้วทำงานเลือกวัตถุนั้นเพื่อใช้เป็นวัตถุเป้าหมาย
5. ขั้นตอนที่ 5 เมื่อได้วัตถุเป้าหมายแล้วจะทำการคำนวณตำแหน่งของหุ่นยนต์เพื่อที่จะเข้าไปถ่ายภาพวัตถุในระยะใกล้ เมื่อคำนวณตำแหน่งของหุ่นยนต์แล้วจะทำการส่งคำสั่งไปยังแขนหุ่นยนต์เพื่อเคลื่อนที่ไปยังตำแหน่งดังกล่าว
6. ขั้นตอนที่ 6 เมื่อหุ่นยนต์เคลื่อนที่ไปยังตำแหน่งที่ต้องการโดยที่กล้องอยู่เหนือวัตถุตามระยะที่กำหนด ระบบจะทำการถ่ายภาพวัตถุและส่งข้อมูลภาพไปประมวลผลเพื่อทำงานประเมินตำแหน่งและมุมมองของวัตถุ จากนั้นจะนำข้อมูลของวัตถุที่ได้เพื่อใช้ในการคำนวณท่าทางของหุ่นยนต์ในการหยิบวัตถุนั้น
7. ขั้นตอนที่ 7 ข้อมูลท่าทางของหุ่นยนต์และขั้นตอนในการหยิบวัตถุนั้นจะถูกส่งไปยังแขนหุ่นยนต์ตามลำดับ เพื่อที่จะทำการหยิบวัตถุเป้าหมายและนำวัตถุไปวางยังตำแหน่งที่ระบุไว้
8. ขั้นตอนที่ 8 เมื่อเสร็จสิ้นขั้นตอนการหยิบวัตถุและนำวัตถุไปวางยังตำแหน่งที่ต้องการเรียบร้อยแล้ว ระบบจะทำการส่งคำสั่งไปยังแขนหุ่นยนต์เพื่อที่จะเคลื่อนที่กลับสู่ตำแหน่งโฮม เพื่อทำการวนซ้ำขั้นตอนที่ 1 ใหม่ไปเรื่อย ๆ จนกว่าจะหยิบวัตถุทั้งหมดในพื้นที่การทำงาน

3.3.1 ขั้นตอนการทำงานของระบบปฏิบัติการหุ่นยนต์ ROS

เนื่องจากระบบทั้งหมดในทุกขั้นตอนนี้ถูกสร้างขึ้นและใช้งานบนระบบปฏิบัติการหุ่นยนต์ ROS ดังนั้นมาตรฐานการรับส่งข้อมูล การใช้งานโปรแกรม หรือการเรียกใช้คำสั่งต่าง ๆ จะเป็นมาตรฐานของระบบปฏิบัติการหุ่นยนต์ ROS ทั้งสิ้น ระบบปฏิบัติการหุ่นยนต์ ROS มีความยืดหยุ่นและมีความสามารถในการแบ่งงานออกเป็นโมดูลหรือโหนดที่ทำงานแยกกัน โดยแต่ละโหนดจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่และการทำงานของตัวเองโดยอิสระ และสามารถสื่อสารกับโหนดอื่นผ่านทาง Topics หรือ Services โดยเป็นระบบการส่งข้อมูลแบบไม่ต่อเนื่อง โดยมีองค์ประกอบหลักดังนี้

1. Nodes หรือ โหนด คือ กระบวนการทำงานใน ROS แต่ละกระบวนการ ซึ่งเป็นหน่วยพื้นฐานของการทำงาน แต่ละโหนดมีความสามารถในการประมวลผลข้อมูล สร้างข้อมูล หรือสื่อสารกับโหนดอื่น ๆ ผ่านการใช้ Topics และ Services
2. Messages คือ ข้อมูลที่ถูกส่งผ่านระหว่างโหนด เป็นรูปแบบข้อความที่ถูกแพ็คไปใน Message ซึ่งมีโครงสร้างที่กำหนดไว้ล่วงหน้า แต่ละข้อความมักจะมีโครงสร้างที่เข้าใจง่าย เช่น ข้อความเพื่อส่งค่าตัวเลข, ข้อความสำหรับข้อมูลเซ็นเซอร์, หรือข้อความสำหรับคำสั่งควบคุม
3. Topics คือ ระบบเพื่อให้โหนดสามารถสื่อสารกันได้โดยตรง Topics เป็นช่องทางสำหรับส่งข้อมูลที่ไม่ต้องการการตอบกลับ ซึ่งโหนดที่สนใจสามารถเข้ารับข้อมูลที่ถูกส่งผ่าน Topics นั้น
4. Services คือ เป็นวิธีการสื่อสารที่ใช้สำหรับการส่งคำขอและการตอบกลับ บริการมักจะถูกใช้เมื่อโหนดต้องการข้อมูลหรือการกระทำที่ต้องการให้โหนดอื่นทำ
5. Parameter Server คือ เซิร์ฟเวอร์ที่ใช้สำหรับจัดเก็บค่าพารามิเตอร์ ซึ่งเป็นข้อมูลที่สามารถเปลี่ยนแปลงได้ในระหว่างการทำงานของโปรแกรม พารามิเตอร์เหล่านี้สามารถใช้ในการกำหนดค่าพารามิเตอร์สำหรับโหนดต่าง ๆ ในระหว่างการทำงาน
6. Tools คือ เครื่องมือที่ช่วยในการพัฒนา ทดสอบ และปรับแต่งโปรแกรม เช่น Rviz เป็นโปรแกรมที่ใช้สำหรับการแสดงผล

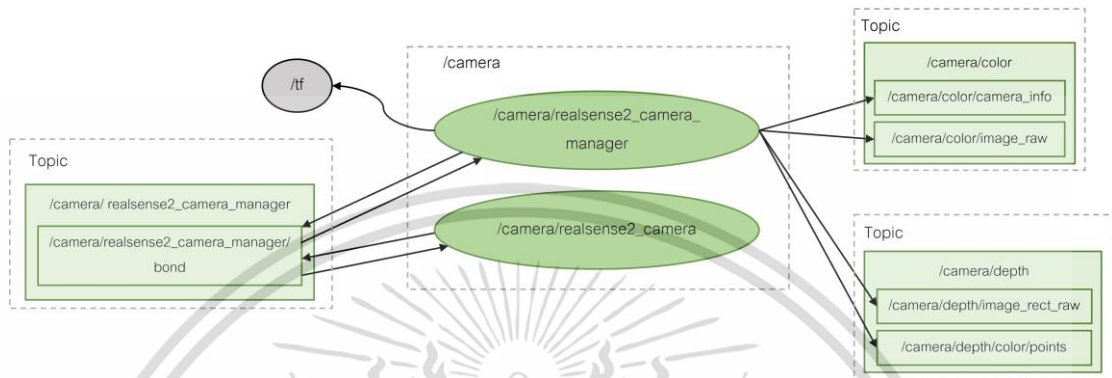
การใช้งานระบบปฏิบัติการ ROS จะช่วยลดความซับซ้อนและเพิ่มประสิทธิภาพในการทำงานกับหลายอุปกรณ์ที่แตกต่างกัน และยังช่วยในการจัดการการสื่อสารและการเชื่อมต่อระหว่างโปรแกรมของหุ่นยนต์ได้โดยง่ายซึ่งในงานศึกษานี้จะแยกโหนดการทำงานของอุปกรณ์ออกเป็น 3 ส่วนหลัก ประกอบด้วย ส่วนของการทำงานของกล้อง ส่วนของโปรแกรมประมวลผลหลักและการตรวจจับวัตถุ และส่วนของการควบคุมการทำงานของหุ่นยนต์

3.3.2 การแยกการทำงานเป็นโหนดสำหรับแต่ละโปรแกรมและอุปกรณ์

ในการศึกษานี้ได้ทำการแบ่งส่วนการทำงานออกเป็น 3 ส่วน ประกอบด้วย ส่วนของการทำงานของกล้อง ส่วนของโปรแกรมประมวลผลหลักและการตรวจจับวัตถุ และส่วนของการควบคุมการทำงานของหุ่นยนต์ โดยแต่ละส่วนจะประกอบด้วยโปรแกรมในการประมวลผลหลายโปรแกรม ซึ่งแต่ละโปรแกรมจะทำงานร่วมกันทั้งภายในส่วนการทำงานของตนเองและทำงานร่วมกับส่วนอื่น ๆ โดยจะรับและส่งข้อมูลต่าง ๆ รวมถึงเรียกใช้คำสั่งต่าง ๆ มาตรฐานระบบปฏิบัติการหุ่นยนต์ ROS ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแรกสุดของระบบนี้คือส่วนของเซนเซอร์รับภาพสีและภาพความลึกที่ใช้งานกล้อง Intel realsense D435i โดยในส่วนนี้จะประกอบด้วยอุปกรณ์ชิ้นเดียว และโปรแกรมควบคุม ประมวลผล ภาพจากเซนเซอร์ทั้ง 2 ชนิด แปลงและส่งข้อมูลภาพในมาตรฐานระบบปฏิบัติการหุ่นยนต์ ROS ทั้งหมดแบ่งได้เป็น 2 โหนดดังรูปที่ 3.7



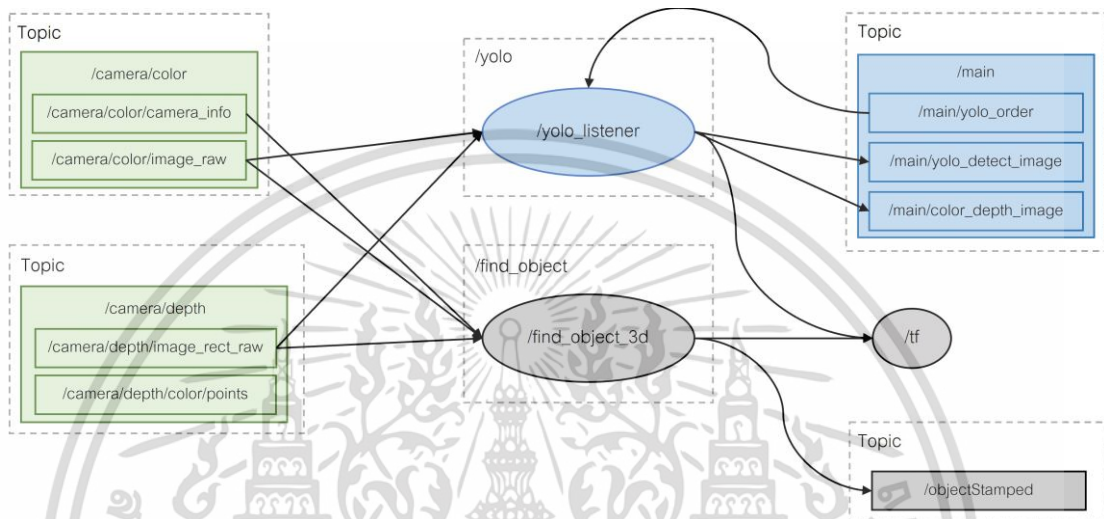
รูปที่ 3.7 โหนดการทำงานส่วนเซนเซอร์รับภาพสีและภาพความลึก ที่ใช้งานกล้อง Intel realsense D435i

การส่งออกข้อมูลจะใช้รูปแบบการรับส่งข้อมูลแบบ Publisher และ Subscriber โดยโหนดที่ส่งข้อความจะเรียกว่า Publisher และข้อมูลที่ส่งนั้นจะถูกส่งไปยัง Topic โดยชื่อที่ระบุใน Topic จะใช้เพื่อระบุเนื้อหาของข้อความ โหนดที่สนใจในข้อมูลบางประเภทจะทำการ Subscribe หัวข้อที่ต้องการ อาจมีการ Publisher และ Subscriber หลายรายพร้อมกันสำหรับหัวข้อเดียวและโหนดเดียวอาจมี Publisher และ Subscriber ได้หลาย Topics เช่นเดียวกัน โดยทั่วไป Publisher และ Subscriber จะไม่ได้รู้ถึงการมีอยู่ของกันและกัน หรือก็คือจะแยกกันผลิตข้อมูลหรือรับข้อมูลตามหน้าที่ของตนเอง ในส่วนของโหนดกล้องนี้ จะประกอบด้วย 2 โหนด คือ /camera/realsense2_camer_manager และ /camera/realsense2_camera ซึ่ง โหนด /camera/realsense2_camera จะทำหน้าที่ในการส่งออกข้อมูลภาพสีและข้อมูลภาพความลึก หรือก็คือจะทำหน้าที่เป็น Publisher ในการส่งข้อมูลเข้ายัง Topic โดย Topic ทั้งหมดที่เกี่ยวข้องได้แก่ 4 Topics ต่อไปนี้

1. /camera/color/camera_info คือข้อมูลสถานะของกล้อง
2. /camera/color/image_raw คือข้อมูลภาพสีที่ถ่ายได้จากกล้อง
3. /camera/depth/image_rect_raw คือข้อมูลภาพความลึกที่ถ่ายได้จากกล้อง
4. /camera/depth/color/points คือข้อมูลภาพสีแบบ 3 มิติ ที่เกิดจากการรวมกันของข้อมูลภาพสีและภาพความลึก โดยจะแสงเป็นจุด RGB ที่เรียกว่า pointcloud

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งข้อมูลเหล่านี้จะถูกนำไปใช้ต่อในส่วนของการประมวลผลเพื่อเข้าสู่กระบวนการตรวจจับวัตถุจากภาพ โดยโหนดที่รับข้อมูลเหล่านี้ไปใช้งานต่อหรือ Subscriber จะประกอบด้วย 2 โหนดหลักคือโหนด /yolo_listener และ /find_object_3d ซึ่งขั้นตอนและการทำงานต่าง ๆ ของโปรแกรมเหล่านี้จะถูกควบคุมด้วยโหนดประมวลผลหลักที่จะกล่าวถึงต่อไป ในส่วนนี้จะกล่าวถึงส่วนของการรับส่งข้อมูลภาพเพื่อใช้ในการตรวจจับวัตถุเป็นหลักซึ่งถูกแสดงดังรูปที่ 3.8



รูปที่ 3.8 ขั้นตอนการส่งข้อมูลภาพวัตถุไปยังโหนด 2 โหนด ที่ใช้ข้อมูลนำไปประมวลผลภาพต่อไป

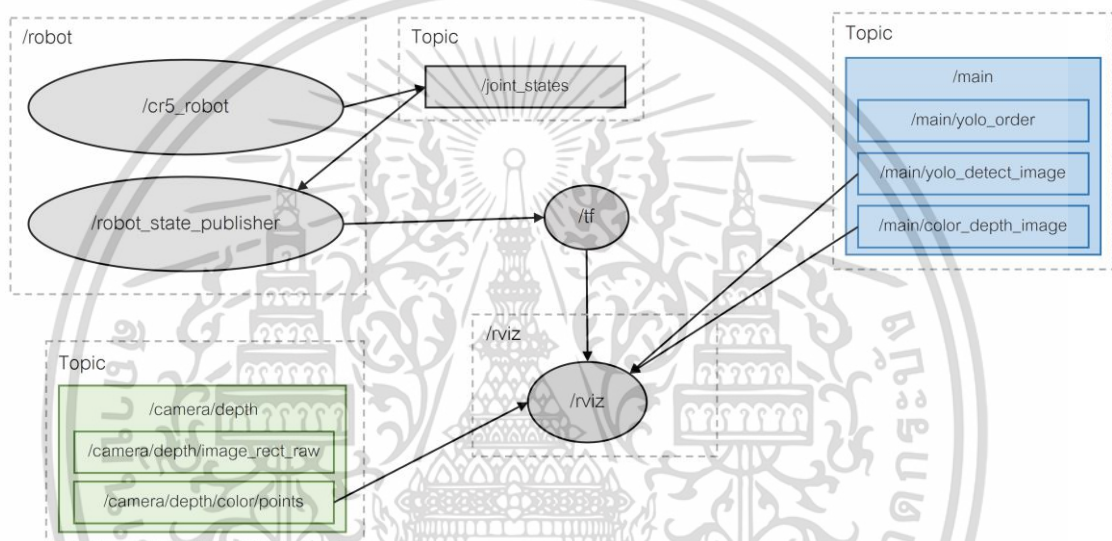
เมื่อโปรแกรมประมวลผลหลักสั่งงานให้โหนด 2 ทั้ง ทำงานตามขั้นตอน โหนด /yolo_listener ซึ่งเป็นโหนด Subscriber ที่รับข้อมูลจาก /camera/color/image_raw หรือข้อมูลภาพสีที่ถ่ายได้จากกล้อง และ /camera/depth/image_rect_raw หรือข้อมูลภาพความลึกที่ถ่ายได้จากกล้อง โดยข้อมูลภาพจะถูกใช้ในกระบวนการตรวจจับวัตถุด้วยอัลกอริทึม YOLOv5 จากโมเดลที่ผ่านการฝึกฝนมาแล้ว เมื่อได้ผลลัพธ์จากอัลกอริทึมการตรวจจับวัตถุแล้ว โหนด /yolo_listener จะนำข้อมูลที่ส่งเข้าไปยัง Topic ต่าง ๆ ซึ่งในส่วนนี้โหนด /yolo_listener จะทำหน้าที่เป็น Publisher โดยข้อมูลที่จะถูกส่งออกไปได้แก่ ตำแหน่งของวัตถุแบบ 3 มิติ ที่จะถูกส่งไปยัง Topic ที่ชื่อว่า /main/yolo_detect_image และ plugin ที่ชื่อว่า /tf ส่วนข้อมูลภาพความลึกจะถูกประมวลผลใหม่ใหม่ด้วยอัลกอริทึมการใส่สีให้กับภาพความลึก โดยผลลัพธ์ที่ได้จะเป็นภาพความลึกที่มีสีโดยสีจะบ่งบอกถึงระยะตำแหน่งนั้น ๆ ของภาพ โดยจะถูกส่งไปยัง Topic ที่ชื่อว่า /main/color_depth_image อีกโหนดประมวลผลภาพวัตถุคือ โหนด /find_object_3d ทำหน้าที่เป็น Subscriber คล้ายกันกับโหนดก่อนหน้านี้ โดยจะรับข้อมูลจาก Topic ที่ชื่อว่า /camera/color/image_raw หรือข้อมูลภาพสี และ /camera/depth/image_rect_raw หรือข้อมูลภาพความลึก และประมวลผลข้อมูลและตรวจจับวัตถุด้วยเทคนิค feature detection,

feature description และ feature matching เมื่อได้ผลลัพธ์เป็นตำแหน่งของวัตถุ และมุมมองเสาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของวัตถุแล้ว จะนำข้อมูลที่ได้ออกไปยัง /tf เพื่อนำค่าของวัตถุแบบ 6 มิติไปใช้ในการคำนวณการหุบของหุ่นยนต์ต่อไป

ต่อมาในส่วนของโหนดการควบคุมหุ่นยนต์นั้นจะประกอบด้วย 2 โหนด โดยโหนดแรกจะทำหน้าที่ส่งคำสั่งไปควบคุมหุ่นยนต์และอีกโหนดหนึ่งจะทำหน้าที่อ่านค่าของ joint ต่าง ๆ ของหุ่นยนต์ เพื่อทำการคำนวณท่าทางและตำแหน่งของหุ่นยนต์ รวมทั้งทำการส่งข้อมูลท่าทางของหุ่นยนต์ไปยัง /tf เพื่อใช้เป็นตำแหน่งอ้างอิงและแสดงผลโมเดลหุ่นยนต์ ส่วนโหนดการแสดงผลประกอบด้วย 1 โหนด คือ /rviz โดยจะรับข้อมูลต่าง ๆ ที่ถูกส่งมายัง Topic และแสดงผลผ่านโปรแกรม RVIZ โดยแผนผังการรับส่งข้อมูลของ 2 องค์ประกอบนี้ถูกแสดงดังรูปที่ 3.9

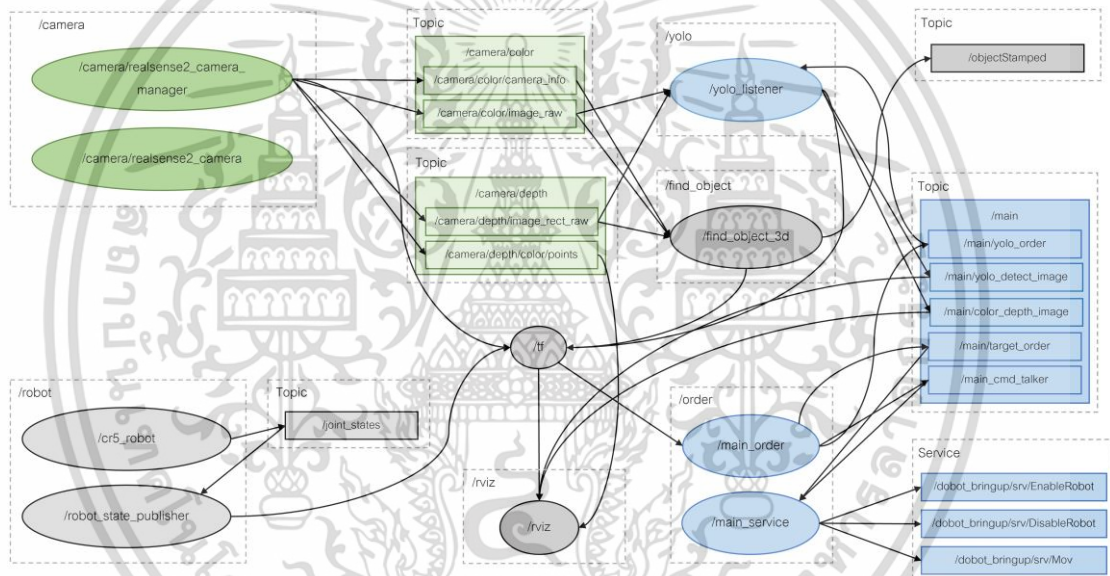


รูปที่ 3.9 ขั้นตอนการส่งข้อมูลต่าง ๆ ของโหนดหุ่นยนต์และโหนดการแสดงผล

โดยโมเดล 3 มิติของหุ่นยนต์ กล้อง และวัตถุที่แสดงผ่านโปรแกรม RVIZ จะถูกแสดงอ้างอิงตามตำแหน่งจาก /tf ซึ่งจะมีข้อมูลของลิงค์ต่าง ๆ อ้างอิงจากไฟล์ URDF รวมถึงตำแหน่งของลิงค์ทั้ง 3 แกนและมุมมองการหมุนรอบแกนทั้ง 3 โดยจะปรับเปลี่ยนไปตามตำแหน่งจริงของลิงค์นั้น ๆ จากข้อมูลที่ถูกส่งมายัง /tf แบบเรียลไทม์ นอกจากนี้การแสดงผลของพื้นที่การทำงานทั้งหมดด้วยการแสดง pointcloud นั้น ตำแหน่งของจุดต่าง ๆ แบบ 3 มิติของ pointcloud นี้จะถูกอ้างอิงตามตำแหน่งลิงค์ของกล้องที่ถูกติดตั้งอยู่ที่ส่วนปลายแขนของหุ่นยนต์ การแสดงผลภาพสีจากกล้องที่ผ่านการตรวจจับวัตถุ และนำมาตีกรอบรอบวัตถุที่ถูกระบุจับได้จะถูกนำเสนอภาพผ่าน Topic ที่ชื่อว่า /main/yolo_detect_image ในส่วนภาพความลึกที่มีสีตามระยะความลึกจะถูกแสดงผ่าน Topic ที่ชื่อว่า /main/color_depth_image โดยทั้ง 2 ภาพนี้จะแสดงอยู่ในโปรแกรม RVIZ เพื่อให้ผู้ใช้งานสามารถทราบถึงการทำงานของอัลกอริทึม รวมทั้งทราบถึงวัตถุที่สามารถตรวจจับได้จากอัลกอริทึมผ่านภาพในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อที่จะให้ทุกโหนดและโปรแกรมทำงานร่วมกันได้อย่างถูกต้อง ระบบจะมีโหนดหนึ่งทำหน้าที่ควบคุมขั้นตอนการทำงานทั้งหมดของระบบนี้ โดยหน้าที่หลักคือการทำงานให้กระบวนการทำงานของระบบเป็นไปตามกระบวนการดังรูปที่ 3.3.1 โดยจะควบคุมการรับข้อมูลและใช้งานข้อมูลของโหนดต่าง ๆ ซึ่งโหนดที่ว่านี้คือ /main_order โดยควบคุมการทำงานของโหนด yolo ผ่าน Topic รวมทั้งจะใช้ข้อมูลต่าง ๆ ของ /tf ตามเงื่อนไขต่าง ๆ ที่ถูกกำหนดไว้ และสุดท้ายจะทำการควบคุมหุ่นยนต์ผ่านการ publish คำสั่งไปยังอีกโหนดหนึ่งที่สร้างขึ้นมาเพื่อทำการควบคุมหุ่นยนต์โดยเฉพาะคือ โหนด /main_service โดยโหนดนี้จะทำการออกคำสั่งควบคุมหุ่นยนต์ผ่านการเรียกใช้ ROS Service ต่าง ๆ โดยคำสั่งที่ใช้ในการควบคุมหุ่นยนต์เช่นการ EnableRobot ตลอดจนการเคลื่อนที่ต่าง ๆ ของหุ่นยนต์ทั้ง 6 แกน และคำสั่งสำหรับอุปกรณ์ที่ปลายแขนของหุ่นยนต์ที่จะดูหรือปล่อยวัตถุอีกด้วย



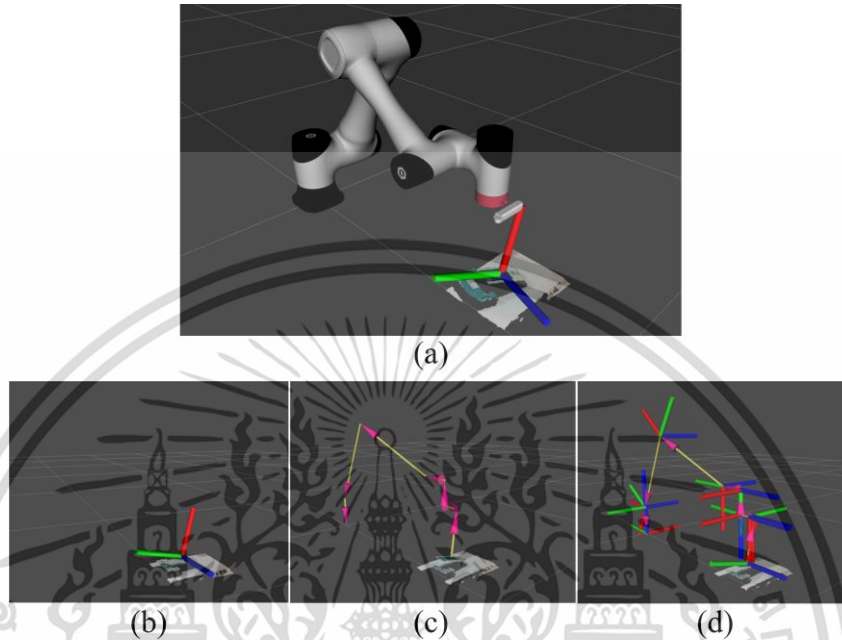
รูปที่ 3.10 แผนผังการเชื่อมต่อและการรับส่งข้อมูลต่าง ๆ ของแต่ละโหนด

เมื่อรวมการทำงานของทุกโหนดเข้าด้วยกันและจะได้แผนผังการเชื่อมต่อและการรับส่งข้อมูลต่าง ๆ ของแต่ละโหนดดังรูปที่ 3.10 โดยทุกโหนดจะทำงานร่วมกันอย่างราบรื่นไปตามกระบวนการที่ถูกออกแบบไว้เพื่อที่จะได้ผลลัพธ์ตามวัตถุประสงค์ที่กำหนด

3.4 การคำนวณและการกำหนดท่าทางของหุ่นยนต์

เพื่อเปิดใช้งานการจัดการวัตถุแนวตั้งที่แม่นยำ ท่าหยิบของหุ่นยนต์ได้รับการคำนวณด้วย offset คงที่จากวัตถุเป้าหมาย การคำนวณท่าทางเกี่ยวข้องกับการคำนวณตามเมทริกซ์ โดยอ้างอิงกรอบหุ่นยนต์เพื่อการวางตำแหน่งที่แม่นยำภายในกรอบงาน ROS ด้วยการวางแนวตัวเองกับวัตถุใน

ลักษณะที่มีการควบคุมและมีประสิทธิภาพ หุ่นยนต์จึงสามารถเพิ่มประสิทธิภาพโดยรวมและความน่าเชื่อถือของระบบหยาบขยับได้



รูปที่ 3.11 (a) หุ่นยนต์สร้างภาพใน Rviz, (b) - (d) การหาตำแหน่งของวัตถุอย่างต่อเนื่องจากลิงค์ฐานของหุ่นยนต์

ภายในระบบปฏิบัติการหุ่นยนต์ ROS ลิงค์ของวัตถุถูกสร้างขึ้นเพื่ออ้างอิงจากลิงค์ของกล้องที่ถูกอ้างอิงมาจากแขนของหุ่นยนต์ ซึ่งข้อมูลของลิงค์กล้องนั้นรวมอยู่ในไฟล์ URDF ของหุ่นยนต์ ซึ่งเป็นหนึ่งในรูปแบบการกำหนดโมเดลหุ่นยนต์อย่างหนึ่ง โมเดลหุ่นยนต์หรือที่เรียกว่า Robot Description นั้นสามารถพัฒนาขึ้นเป็นรูปแบบ URDF (Unified Robot Description Format) โดยภาษา XML กล่าวคือ URDF มีหน้าที่อธิบายลักษณะการติดตั้งเชิงกลหรือ Body Layout ซึ่งข้อมูลต่าง ๆ ภายในไฟล์ URDF นั้นจะมีการกำหนดเฟรมอ้างอิงของหุ่นยนต์หรือที่เรียกว่า Reference Frame ทำให้ทราบข้อมูลของ ข้อต่อของหุ่นยนต์ (Joint) , ชิ้นส่วนต่าง ๆ ของหุ่นยนต์หรือลิงค์ (Link) รวมถึงเซ็นเซอร์ต่าง ๆ ที่ถูกเคลื่อนย้าย โดยสถานะของการเปลี่ยนแปลงเหล่านี้จะปรากฏที่ State Publisher ในทางปฏิบัติจริงนั้นจะทำการติดตั้งอุปกรณ์ชิ้นหนึ่งเพื่อทำการติดตั้งกล้องเข้ากับที่ปลายแขนของหุ่นยนต์ ซึ่งจำทำให้กล้องนั้นสามารถเคลื่อนที่ไปพร้อมกับปลายแขนหุ่นยนต์ โดยตำแหน่งของกล้องจะยื่นออกมาด้านข้างของแขนหุ่นยนต์ โดยจะมีการไขนอตยึดติดกับที่หลายแขนหุ่นยนต์ และเว้นช่องว่างสำหรับการต่อสายไปยังกล้องที่ถูกติดตั้งอยู่ รูปของอุปกรณ์นี้จะถูกแสดงในรูป 3.12 พร้อมทั้งภาพเมื่อติดตั้งอุปกรณ์ชิ้นนี้กับแขนหุ่นยนต์จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 อุปกรณ์ที่ใช้ในการติดตั้งกล่องเข้ากับปลายแขนของหุ่นยนต์

การกำหนดลิค์เหล่านี้จะทำให้ได้รับตำแหน่งของวัตถุอย่างต่อเนื่องโดยอ้างอิงได้จากตำแหน่งหุ่นยนต์ ทำให้สามารถแสดงภาพของทั้งหุ่นยนต์และวัตถุด้วยโปรแกรมแสดงผลดังรูปที่ 3.11 การใช้งานนี้ทำให้การนำแสดงโมเดลหุ่นยนต์และตำแหน่งวัตถุทั้งหมดในพื้นที่การทำงานเป็นไปอย่างราบรื่นและมีความต่อเนื่อง เมื่อหุ่นยนต์และกล่องเคลื่อนที่ไปมาในขณะที่แสดงภาพนั้นจะสามารถแสดงผลตำแหน่งของวัตถุได้อย่างถูกต้องและมีความต่อเนื่อง

มุมการหยิบวัตถุนั้นจะถูกคำนวณโดยสมการ ซึ่งแสดงผ่านตัวแปร $Robot_{xyz}$ โดยค่าของตัวแปรนี้จะถูกกำหนดโดยใช้การแปลงเมทริกตามตำแหน่งของวัตถุ Obj_x, Obj_y, Obj_z และมุมมองของวัตถุ $Obj_{rx}, Obj_{ry}, Obj_{rz}$ การคำนวณตำแหน่งการหยิบของหุ่นยนต์ในแต่ละแกน $Robot_x, Robot_y, Robot_z$ จะถูกคำนวณโดยใช้สมการต่อไปนี้

$$Robot_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(R_y) & -\sin(R_y) \\ 0 & \sin(R_y) & \cos(R_y) \end{bmatrix} \quad (3.1)$$

$$Robot_y = \begin{bmatrix} \cos(R_x) & 0 & \sin(R_x) \\ 0 & 1 & 0 \\ -\sin(R_x) & 0 & \cos(R_x) \end{bmatrix} \quad (3.2)$$

$$Robot_z = \begin{bmatrix} \cos(R_z) & -\sin(R_z) & 0 \\ \sin(R_z) & \cos(R_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

สมการ (3.1), (3.2) และ (3.3) แสดงถึงการคำนวณสำหรับตำแหน่งหยิบของหุ่นยนต์ ซึ่งจะถูกนำมาคำนวณต่อเพื่อเพิ่มระยะ offset ระหว่างแขนหุ่นยนต์สำหรับการกำหนดระยะของปลายแขน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ก่อนดำเนินการหยิบวัตถุที่ตำแหน่งที่ระบุ โดย offset นี้จะเป็นระยะห่างระหว่างปลายแขน หุ่นยนต์กับวัตถุ ระยะนี้จะแทนด้วยเมทริก $Robot_{off}$ ดังแสดงในสมการต่อไปนี้

$$Robot_{off} = [0 \quad 0 \quad -d] \quad (3.4)$$

การคำนวณต่อมาสำหรับตำแหน่งในแนวแกน x, y และ z การหยิบวัตถุของหุ่นยนต์จะแสดงด้วยเมทริก $Robot_{xyz}$ ดังสมการต่อไปนี้

$$Robot_{xyz} = Robot_x \cdot Robot_y \cdot Robot_z \cdot Robot_{off} + [Obj_x \quad Obj_y \quad Obj_z] \quad (3.5)$$

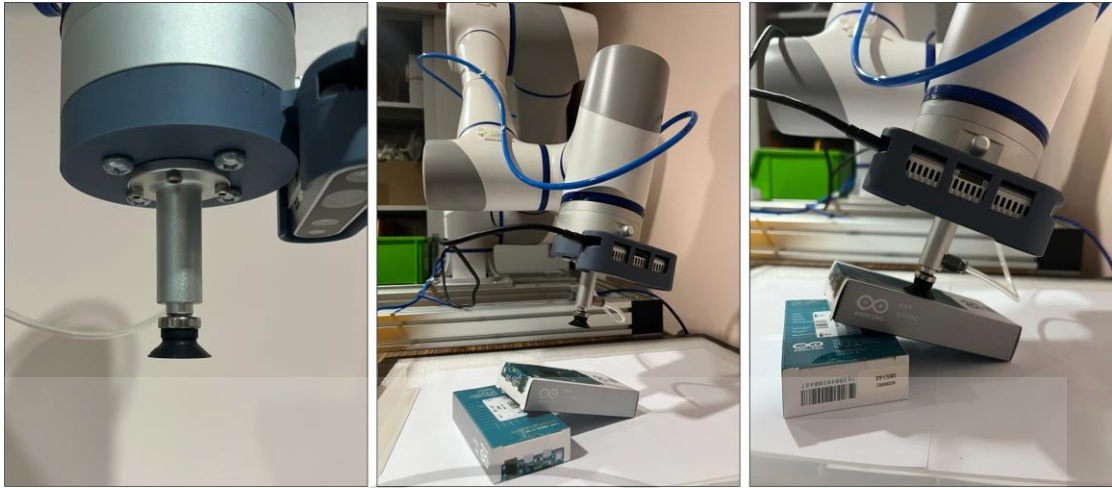
ในสมการ (3.5) ตำแหน่ง $Robot_{xyz}$ จะคำนวณต่อโดยจะเพิ่มค่า offset และเมทริก $[Obj_x \quad Obj_y \quad Obj_z]$ เข้าด้วยกัน เพื่อกำหนดตำแหน่งการหยิบวัตถุตามที่ต้องการ ผลลัพธ์ที่ได้คือ หุ่นยนต์จะอยู่ห่างจากวัตถุในระยะ d มม. ก่อนที่จะเคลื่อนที่แบบตั้งฉากกับวัตถุภายในระนาบไปหาวัตถุ การวางตำแหน่งของปลายแขนของหุ่นยนต์สามารถคำนวณได้ตามสมการต่อไปนี้

$$Robot_{rx} = Obj_{rx} + 180 \quad (3.6)$$

$$Robot_{ry} = -(Obj_{ry} - 180) \quad (3.7)$$

$$Robot_{rz} = Obj_{rz} + 180 \quad (3.8)$$

สมการ (3.6) ถึง (3.8) เป็นการคำนวณเพื่อกำหนดค่าของ $Robot_{rx}$, $Robot_{ry}$ และ $Robot_{rz}$ ซึ่งแสดงถึงการวางแนวของส่วนปลายแขนของหุ่นยนต์ ด้วยการรวมการคำนวณเหล่านี้ หุ่นยนต์จึงสามารถบรรลุท่าหยิบที่ต้องการได้ วิธีการนี้ช่วยให้แน่ใจว่าหุ่นยนต์อยู่ในตำแหน่งที่ถูกต้องและกำหนดทิศทางเพื่อการจัดการวัตถุที่มีประสิทธิภาพและความคม ดังแสดงในรูปที่ 3.13



รูปที่ 3.13 รูปถ่ายของขั้นตอนการหยิบซึ่งแสดงทิศทางการหยิบในแนวตั้งที่แม่นยำและการเชื่อมต่อที่สมบูรณ์แบบของขบถด้วยคูดักบัววัตถุ

วิธีการแบบไฮบริดที่เรานำเสนอแสดงให้เห็นถึงความสามารถที่โดดเด่นในแง่ของการตรวจจับวัตถุและการประมาณค่าท่าทาง จากการเปรียบเทียบในตารางที่ 3.2 วิธีการแบบไฮบริดนั้นเหนือกว่าวิธีการแบบดั้งเดิมในการตรวจจับวัตถุในทุกระยะห่างและทิศทาง นอกจากนี้ยังให้การประมาณค่าท่าทาง 6D ที่แม่นยำ พร้อมด้วยข้อกำหนดพิกัด x , y และ z ที่แม่นยำ

ตารางที่ 3.2 แสดงความสามารถในการตรวจจับและระบุข้อมูลวัตถุ

ความสามารถในการตรวจจับ	วิธีแบบไฮบริด	YOLOv5	FAST & BRISK	Torres et Wong et al. [3]	al. [5]
สามารถตรวจจับวัตถุได้ทุกระยะและตำแหน่งในพื้นที่การทำงาน	✓	✓		✓	✓
สามารถตรวจจับวัตถุได้ทุกมุมมองการวางของวัตถุ	✓	✓		✓	✓
ระบุวัตถุและมุมมองของวัตถุได้ 6 แกน	✓		✓		✓
ระบุตำแหน่งวัตถุในแนวแกน x และ y	✓	✓	✓	✓	✓
ระบุตำแหน่งวัตถุในแนวแกน z	✓		✓	✓	✓

จากตารางที่ 3.2 จะแสดงให้เห็นว่าวิธีการแบบไฮบริดที่งานวิจัยนี้ได้นำเสนอ นั้นมีความสามารถทั้งในการตรวจจับวัตถุได้ทุกระยะและตำแหน่งในพื้นที่การทำงาน เหนือกว่าการใช้งานอัลกอริทึม FAST & BRISK ที่สามารถตรวจจับวัตถุได้เพียงแคในระบะที่จำกัด วิธีการแบบไฮบริดนั้นสามารถตรวจจับวัตถุได้ทุกมุมมองการวางของวัตถุและระบุวัตถุและมุมมองของวัตถุได้ 6 แกนได้ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีความเหนือกว่าการใช้งาน YOLOv5 เพียงอย่างเดียวที่ไม่สามารถระบุมุมมองของวัตถุได้ รวมถึงวิธีการนี้สามารถบอกตำแหน่งวัตถุในแกน x, y และ z ได้ จะเห็นว่าความสามารถนี้นั้นเหมือนกับงานศึกษาบางงาน หรืองานศึกษาของ Wong และคณะ ที่สังเกตได้จากตารางนี้ แต่ผลลัพธ์จากการเปรียบเทียบในการทดลองต่อไปจะแสดงให้เห็นว่าวิธีการแบบไฮบริดที่งานวิจัยนี้นั้นมีความสามารถที่สูงกว่า

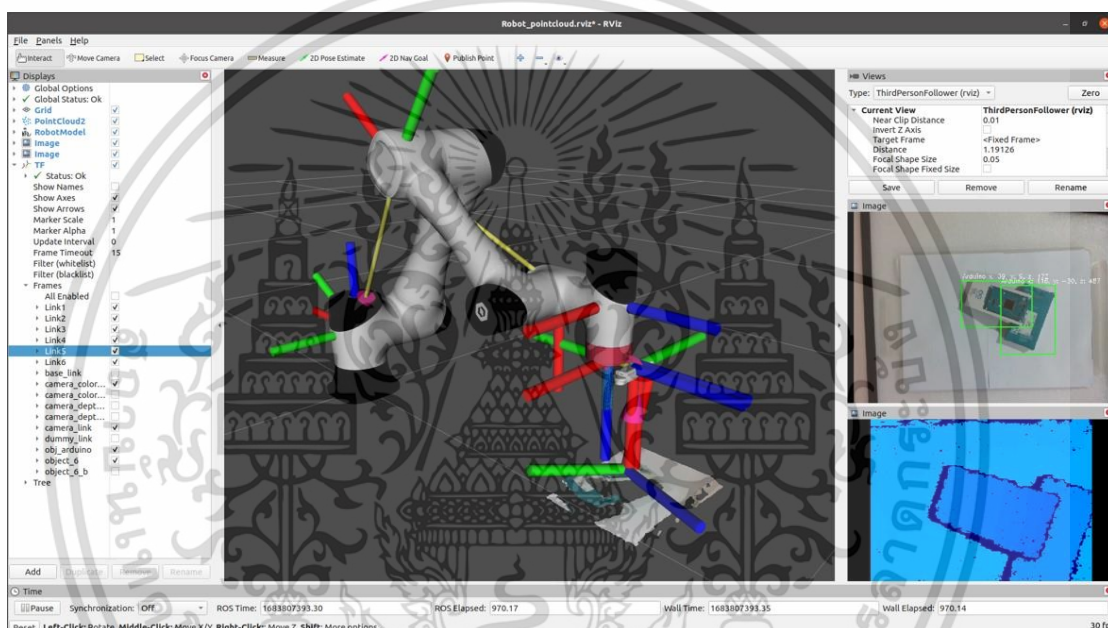


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

วิธีการทดลองและผลการทดลอง

ในการทดลองนี้จะทดสอบความสามารถของระบบตั้งแต่ความสามารถของอัลกอริทึมที่นำไปใช้ตรวจจับวัตถุเป้าหมาย การเลือกวัตถุได้อย่างถูกต้องและระบุตำแหน่งและมุมมองของวัตถุอย่างถูกต้อง รวมถึงการใช้งานโปรแกรม Rviz ในการแสดงผลของหุ่นยนต์ กล้อง และวัตถุเป็นไปอย่างถูกต้อง แสดงจุด pointcloud และตำแหน่งของภาพได้ตรงตามความเป็นจริง โดยทุกองค์ประกอบจะต้องทำงานร่วมกันอย่างราบรื่นและได้ผลลัพธ์ตามวัตถุประสงค์ที่กำหนดไว้

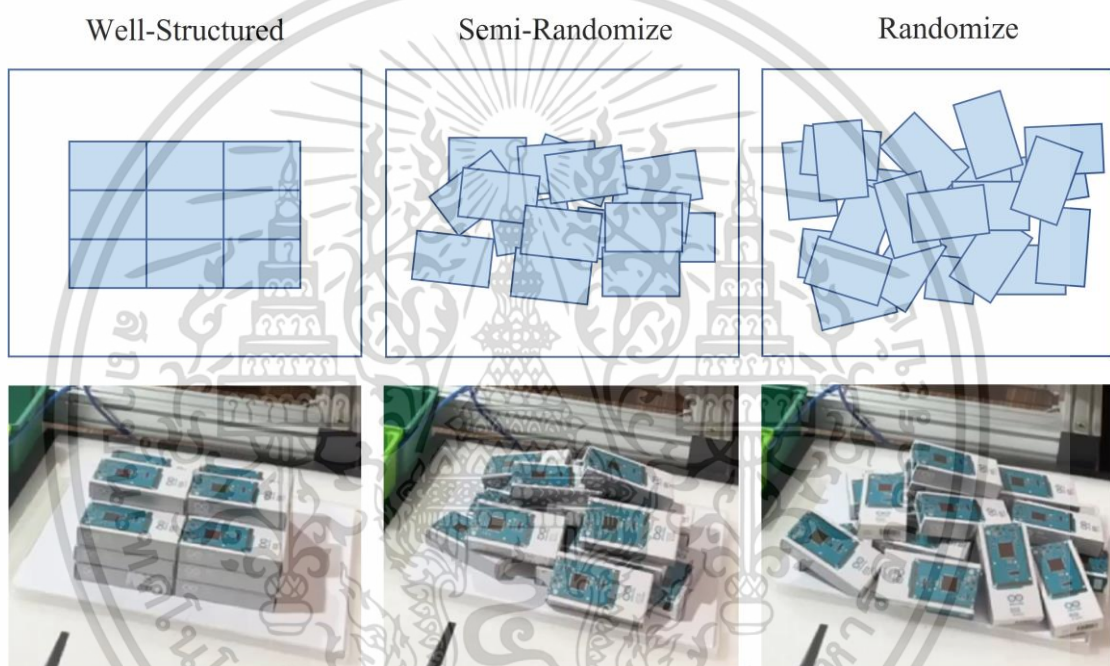


รูปที่ 4.1 Rviz เครื่องมือสร้างภาพ ROS แสดงลิงก์วัตถุ โมเดลหุ่นยนต์พร้อมลิงก์กล้อง พอยต์คลาวด์ ภาพความลึก และภาพวัตถุที่ตรวจพบ

การทดสอบความสามารถของระบบการทำงานประเภทการหิยวัตถุในภาชนะนั้น เพื่อที่จะทำการประเมินผลการทดสอบอย่างครอบคลุมทุกสถานะการณ์ในการทำงานจริงไม่ว่าจะเป็น การหิยวัตถุจากในภาชนะโดยที่วัตถุจะถูกจัดเรียงอย่างเป็นระเบียบ หรือวัตถุทั้งหมดวางแบบสุ่มในทุกตำแหน่งทุกทิศทาง ในการทดสอบนี้จึงได้แบ่งการทดลองออกเป็น 3 การทดลอง โดยมีวัตถุประสงค์ที่ท้าทายต่างกันคือความสามารถในการตรวจจับวัตถุเมื่อสถานะการณ์แตกต่างกันออกไป วัตถุมีความซับซ้อนมากขึ้น ความสามารถในการระบุตำแหน่งและประเมินมุมมองของวัตถุ และความสามารถในการหิยวัตถุ โดยทุกการทดลองนี้จะแสดงผลภาพแบบหุ่นยนต์และวัตถุแบบเรียลไทม์ดังรูปที่ 4.1 ทั้ง 3 การทดลองจะใช้วัตถุที่นำมาทดลองเหมือนกันแต่จะมีการวางตัวของวัตถุที่นำมาใช้ในการทดลองแตกต่างกันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การทดลองที่ 1 วัตถุจะถูกจัดเรียงอย่างเป็นระเบียบภายในภาชนะโดยมี เพื่อทดสอบความแม่นยำของระบบในการตรวจจับและประเมินตำแหน่งและมุมมองของวัตถุที่ไม่ซับซ้อน
2. การทดลองที่ 2 วัตถุถูกจัดเรียงแบบกึ่งสุ่มแนวของวัตถุยังคงหันไปในทิศเดียวกันภายในภาชนะซึ่งมีความซับซ้อนในการจัดการวัตถุตั้งแต่ขั้นตอนการตรวจจับ, ประเมินตำแหน่งและมุมมอง รวมถึงการปรับมุมมองการหยิบวัตถุของหุ่นยนต์ได้อย่างถูกต้อง
3. การทดลองที่ 3 วัตถุทั้งหมดวางแบบสุ่มในทุกตำแหน่งทุกทิศทางภายในภาชนะ ซึ่งมีความท้าทายในแง่ของการตรวจจับวัตถุและการประเมินตำแหน่งและมุมมองอย่างถูกต้องที่จะนำไปสู่การหยิบวัตถุได้อย่างถูกต้อง



รูปที่ 4.2 การทดลองที่ 1 วัตถุถูกจัดเรียงแบบเป็นระเบียบ, การทดลองที่ 2 วัตถุถูกจัดเรียงแบบกึ่งสุ่มแนวของวัตถุยังคงหันไปในทิศเดียวกัน และการทดลองที่ 3 วัตถุทั้งหมดวางแบบสุ่มในทุกทิศทาง

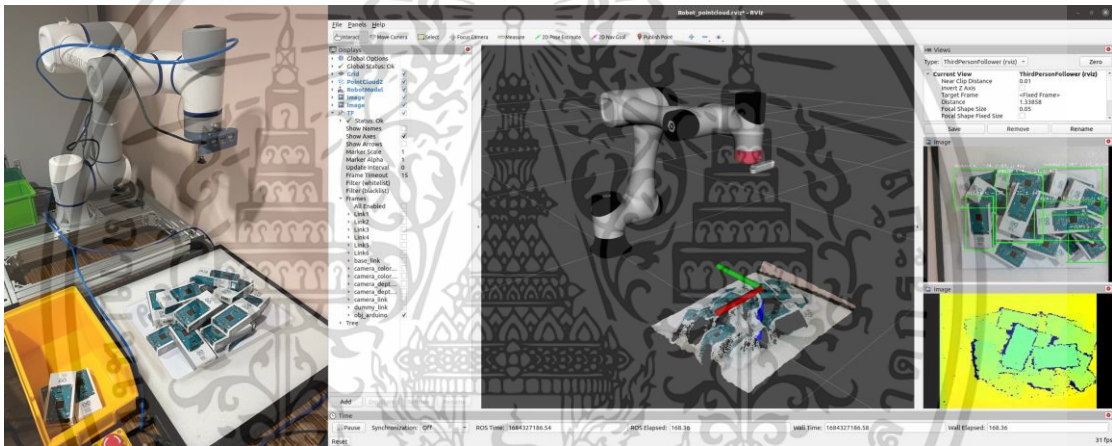
เทคนิคแบบไฮบริดจะเพิ่มระยะของการการตรวจจับวัตถุให้สูงขึ้น โดยการใช้ประโยชน์จากความสามารถในการตรวจจับวัตถุของอัลกอริทึม YOLOv5 ที่สามารถตรวจจับวัตถุในระยะไกลได้อย่างแม่นยำ ส่งผลให้ระยะการตรวจจับหรือพื้นที่การทำงานมีขนาดใหญ่ขึ้น ระยะการตรวจจับหรือพื้นที่การทำงานนี้ถูกนำเสนอผ่านตารางที่ 4.1 และเปรียบเทียบกับเทคนิคแบบดั้งเดิมเพื่อที่จะเน้นย้ำถึงความสามารถที่มากกว่าของวิธีการตรวจจับวัตถุแบบไฮบริด ยิ่งไปกว่านั้นค่าตัวเลขระยะการตรวจจับด้วยวิธีการแบบไฮบริดที่วัดได้นี้ คือตำแหน่งสูงสุดของแขนหุ่นยนต์ที่จะสามารถเคลื่อนออกห่างจากวัตถุได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงระยะในการตรวจจับวัตถุ

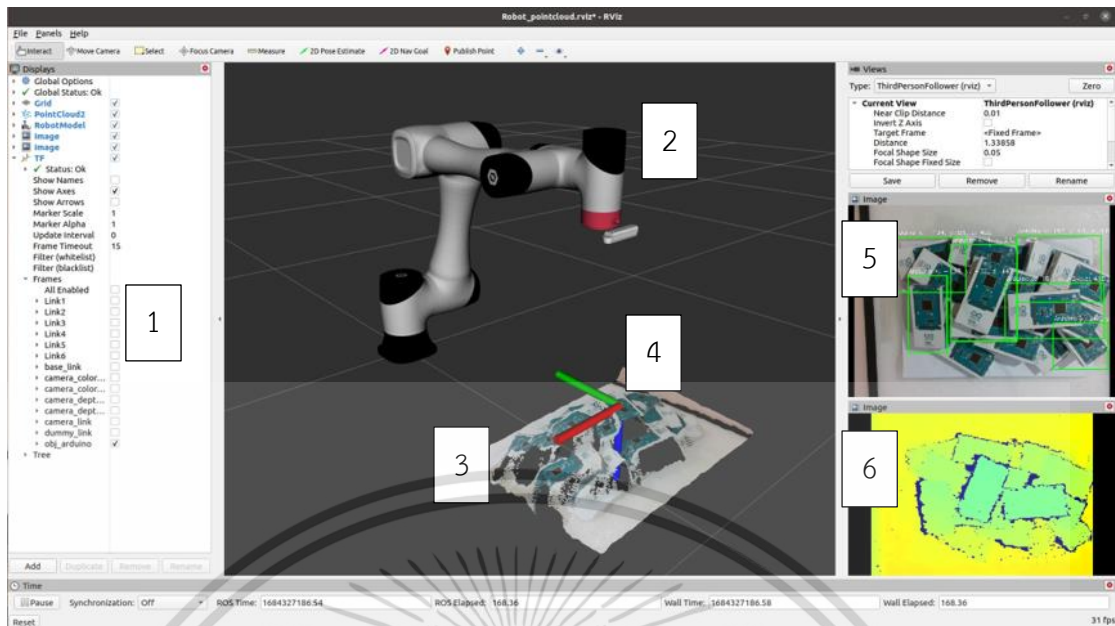
เทคนิคที่ใช้ในการตรวจจับวัตถุ	ระยะในการตรวจจับ (มม.)
FAST & BRISK	150 - 280
YOLOv5 ร่วมกับ FAST & BRISK	150 - 730

กระบวนการตรวจจับวัตถุในระบบการทำงานนี้นั้นสามารถตรวจจับวัตถุได้สำเร็จตามวัตถุประสงค์ด้วยวิธีการตรวจจับแบบไฮบริด รวมถึงสามารถระบุตำแหน่งและมุมมองได้อย่างถูกต้อง การใช้งานโปรแกรม Rviz ในการแสดงผลของหุ่นยนต์และลิงค์วัตถุนี้เป็นไปอย่างถูกต้องและมีประสิทธิภาพ รวมถึงแสดงจุด pointcloud และตำแหน่งของภาพและวัตถุได้ตรงตามความเป็นจริง ดังที่แสดงในรูปที่ 4.3



รูปที่ 4.3 การทำงานของหุ่นยนต์ที่แสดงผลผ่านโปรแกรม Rviz โดยแสดงตำแหน่งของวัตถุแบบ 3 มิติ พร้อมภาพการตรวจจับวัตถุและภาพความลึก

การสร้างภาพข้อมูลนี้ช่วยให้สามารถติดตามตำแหน่งและทิศทางของวัตถุได้อย่างแม่นยำ ช่วยให้เห็นใจได้ถึงการจัดการและการโต้ตอบที่แม่นยำ การผสานรวมหุ่นยนต์และ Rviz ได้อย่างราบรื่นช่วยให้การตรวจสอบและควบคุมการตรวจจับวัตถุและกระบวนการประมาณค่าเป็นไปอย่างราบรื่น เป็นการยืนยันถึงประสิทธิภาพและความน่าเชื่อถือของระบบ



รูปที่ 4.4 โปรแกรม Rviz และส่วนประกอบต่าง ๆ สำหรับแสดงการทำงานของหุ่นยนต์

จากรูปที่ 4.4 ได้แสดงถึงส่วนแสดงผลต่าง ๆ ภายในโปรแกรม Rviz โดยแต่ละส่วนแสดงผล จะทำหน้าที่ดังนี้อ้างอิงตามตัวเลขที่กำกับในรูป

1. ส่วนนี้จะทำหน้าที่แสดงรายการส่วนประกอบต่าง ๆ ที่แสดงผลภายในหน้าต่างโปรแกรม โดยที่ผู้ใช้งานจะสามารถเลือกเปิดหรือปิดการแสดงผลของหน้าต่างโปรแกรมนั้น ๆ ได้ พร้อมทั้งเลือกปรับแต่งอุปกรณ์ต่าง ๆ ตลอดจน Topic ที่นำมาแสดงผล
2. ส่วนแสดงผลโมเดลหุ่นยนต์และกล้อง ภายในหน้าต่างนี้ผู้ใช้งานสามารถปรับมุมมองต่าง ๆ ที่ในพื้นที่การทำงาน รวมถึงสามารถขยายและลดขนาดจากมุมมองได้
3. ส่วนแสดงผล pointcloud แสดงพื้นที่แวดล้อมแบบ 3 มิติที่ถ่ายได้จากกล้อง
4. ส่วนแสดงลิงค์ของวัตถุ แสดงตำแหน่งวัตถุและมุมมอง
5. หน้าต่างแสดงภาพสีและตีกรอบรอบวัตถุที่ตรวจจับได้
6. หน้าต่างแสดงภาพความลึกโดยแทนค่าความลึกด้วยสีที่แตกต่างกัน

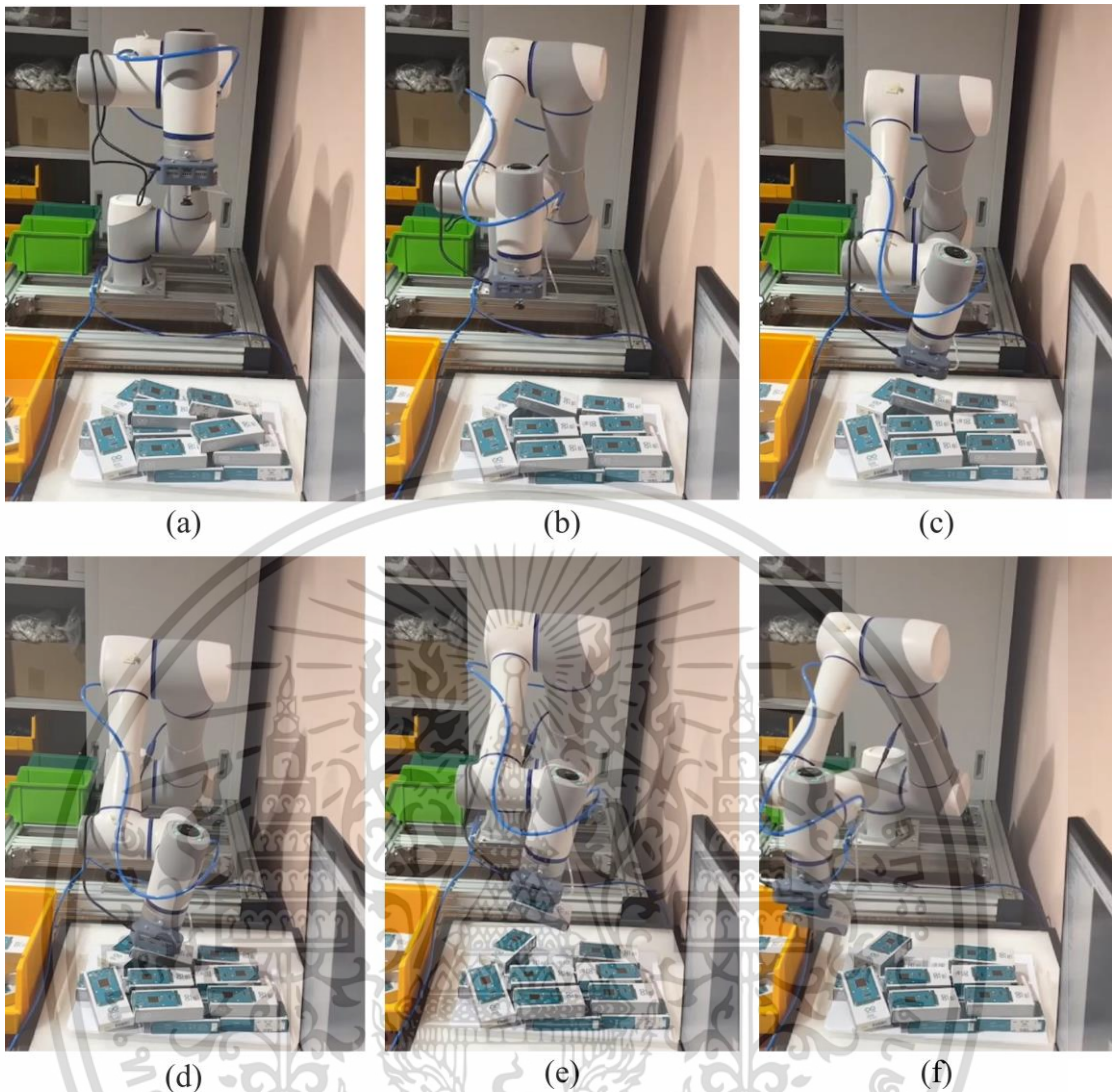
ในด้านผลการทดลอง ระบบการหยิบวัตถุในภาชนะด้วยวิธีการแบบไฮบริดนั้น มีอัตราความสำเร็จจากการพยายามหยิบครั้งเดียวอยู่ที่ 100% ในการทดลองแรก และมีอัตราความสำเร็จลดลงมาเล็กน้อยอยู่ที่ 97.5% ในการทดลองที่ 2 อย่างไรก็ตาม อัตราความสำเร็จลดลงเป็น 86.67% ในการทดลองที่ 3 ตารางที่ 4.2 แสดงระยะเวลาการดำเนินงานของแต่ละขั้นตอนจากการทดลอง โดยเริ่มจากเวลาการตรวจจับวัตถุในขั้นตอนแรก, เปรียบเทียบตำแหน่งและเลือกวัตถุเป้าหมาย เวลาที่หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งเหนือวัตถุ, เวลาในการประเมินมุมมองของวัตถุแบบ 6 แขน และเวลาการคำนวณท่าทางของหุ่นยนต์และหยิบวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 แสดงระยะเวลาการทำงานในแต่ละขั้นตอนจากการทดลอง

ขั้นตอนการทำงาน	เวลาเฉลี่ยจากการทดลองที่ 1 (วินาที)	เวลาเฉลี่ยจากการทดลองที่ 2 (วินาที)	เวลาเฉลี่ยจากการทดลองที่ 3 (วินาที)
การตรวจจับวัตถุในขั้นตอนแรก, เปรียบเทียบตำแหน่งและเลือกวัตถุเป้าหมาย	0.12	0.12	0.12
หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งเหนือวัตถุ	5.30	5.84	5.79
การประเมินมุมมองคาของวัตถุแบบ 6 แกน	0.20	0.20	0.20
การคำนวณท่าทางของหุ่นยนต์และดำเนินการหยิบวัตถุ	11.24	12.60	16.84
รวม	16.86	18.76	22.95

จากตารางนี้จะพบว่าใน 3 การทดลองนี้ การทดลองที่ 3 นั้นมีระยะเวลาการดำเนินการเฉลี่ยสูงที่สุด เนื่องจากความซับซ้อนของงานและการวางแผนของวัตถุที่ทำนายทำให้หุ่นยนต์จำเป็นต้องปรับมุมมองคาของแขนมากกว่าการทดลองที่ 1 และ 2 ทำให้กินระยะเวลาในขั้นตอนการหยิบที่สูง ส่วนในขั้นตอนของการตรวจจับวัตถุ และการประเมินตำแหน่งและมุมมองคาของวัตถุนั้น อัลกอริทึมที่ใช้งานผ่านวิธีการตรวจจับแบบไฮบริดนั้นใช้ระยะเวลาในการทำงานเท่ากันหมดทั้ง 3 การทดลอง



รูปที่ 4.5 ลำดับรูปภาพแสดงให้เห็นถึงการดำเนินการตามกระบวนการรับสินค้าที่ประสบความสำเร็จ (a) หุ่นยนต์ถูกวางตำแหน่งในการกำหนดค่ามุมมองด้านบน (b) หุ่นยนต์เคลื่อนที่เพื่อดูวัตถุเป้าหมายในระยะใกล้ (c) หุ่นยนต์เข้าใกล้วัตถุด้วย offset คงที่ (d) หุ่นยนต์เริ่มการเคลื่อนที่แบบหีบจับ (e) หุ่นยนต์หีบวัตถุได้สำเร็จ (f) หุ่นยนต์ขนย้ายวัตถุไปยังลังจัดเก็บ ภาพชุดนี้แสดงให้เห็นถึงการปฏิบัติงานหีบจับโดยหุ่นยนต์อย่างเป็นระบบและแม่นยำ

ภาพของหุ่นยนต์จริงขณะทำการทดลองถูกแสดงผ่านรูปที่ 4.4 โดยถ่ายภาพของหุ่นยนต์ในระหว่างทดลองในขั้นตอนต่าง ๆ เริ่มตั้งแต่แขนของหุ่นยนต์อยู่ที่ตำแหน่งโฮมและกล้องที่ติดอยู่ที่ปลายแขนหุ่นยนต์ถ่ายภาพรวมของพื้นที่การทำงานเพื่อที่จะหาวัตถุที่สนใจ และดำเนินการตามขั้นตอนในระบบไปจนกระทั่งหีบและวางวัตถุอย่างถูกต้อง จากนั้นจะกลับไปอยู่ตำแหน่งโฮมเหมือนเดิมและทำการหาวัตถุต่อไปอีกครั้ง วนซ้ำไปเรื่อย ๆ จนกว่าจะไม่พบวัตถุในพื้นที่การทำงานหุ่นยนต์ถึงจะหยุดรอจนกว่าจะตรวจจับวัตถุพบอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงการเปรียบเทียบผลลัพธ์ที่ได้จากการทดลองเทียบกับเทคนิคในอดีต

อัลกอริทึม	ชนิดของวัตถุ ที่ใช้ในการ ทดลอง	เซอเซอร์	ความแม่นยำในการ ตรวจจับวัตถุ		การประเมิน มุมมองศิววัตถุ คลากเคลื่อน ต่ำกว่า 5°	ความเร็วใน การ ประมวลผล อัลกอริทึม (วินาที)
			AP ₅₀	AP _{50:95}		
Torres et al. ^[3]	ชิ้นส่วนข้อต่อ	ขนาดเล็ก	98 %	-	-	7.4
Zhuang et al. ^[4]	กล่องกระดาษ ทิชชู	ขนาดเล็ก	92.7 %	-	95 %	-
Wong et al. ^[5]	กล่องกระดาษ	ขนาดเล็ก	98.9 %	-	87.9 %	-
Le and Lin ^[18]	แพ็ค USB	ขนาดเล็ก	100 %	91.2 %	-	0.862
วิธีแบบ ไฮบริด	กล่อง Arduino	ขนาดเล็ก	98.6 %	96.5 %	94.7 %	0.32

ในผลการทดลองและการเปรียบเทียบ วิธีการแบบไฮบริดของเราแสดงให้เห็นประสิทธิภาพที่น่าประทับใจ ตามที่สรุปไว้ในตารางที่ 4.3 ในแง่ของการตรวจจับวัตถุ วิธีการของเราได้รับคะแนน AP50:95 ที่ 96.5% ซึ่งตอกย้ำความสามารถในการตรวจจับวัตถุได้อย่างแม่นยำในทุกทิศทาง และ มีความแม่นยำในการประมาณค่าแบบก่อให้เกิด 94.7% ภายในความคลาดเคลื่อนของมุม 5° นอกจากนี้ วิธีการแบบไฮบริดของเรายังแสดงให้เห็นถึงการประมวลผลที่มีประสิทธิภาพ โดยมีเวลาในการตรวจจับและประมาณค่าน้อยที่สุดเพียง 0.32 วินาที ซึ่งบ่งบอกถึงความเหมาะสมสำหรับการใช้งานแบบเรียลไทม์ รวมถึงการแสดงภาพ 3 มิติแบบเรียลไทม์ของการซิงโครไนซ์หุ่นยนต์กับวัตถุผลลัพธ์นี้แสดงให้เห็นถึงการมีส่วนร่วมที่สำคัญของงานของเราในการตรวจจับวัตถุและการประมาณค่าในหุ่นยนต์อุตสาหกรรม ซึ่งปูทางไปสู่ระบบอัตโนมัติที่ได้รับการปรับปรุงในการใช้งานในโลกแห่งความเป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

ในวิทยานิพนธ์ฉบับนี้ได้นำเสนอการแก้ไขปัญหาก็เกี่ยวกับงานประเภทการหยิบวัตถุใน
ภาชนะ โดยการประยุกต์ใช้งานกล้องจับความลึกราคาต่ำ ร่วมกับระบบการตรวจจับวัตถุ การประเมิน
มุมมองของวัตถุ และการใช้งานหุ่นยนต์ รวมถึงการใช้งานเทคนิค eye-in-hand สำหรับกล้องและ
หุ่นยนต์ โดยที่โปรแกรมการประมวลผลและอุปกรณ์ทุกอุปกรณ์ถูกเชื่อมต่อและใช้งานบน
ระบบปฏิบัติการ ROS ในด้านของการประมวลผลภาพเพื่อตรวจจับวัตถุได้ทำการบูรณาการ
อัลกอริทึม YOLOv5, FAST และ BRISK ทำให้ระบบสามารถบรรลุเป้าหมายในการตรวจจับและ
ประเมินตำแหน่งและมุมมองของวัตถุได้อย่างแม่นยำและมีประสิทธิภาพ แม้ในสถานการณ์ที่ซับซ้อน
และท้าทายต่อการตรวจจับวัตถุ นอกจากนี้ในวิทยานิพนธ์นี้ได้นำเสนอการแสดงผลภาพและโมเดลการ
เคลื่อนไหวของหุ่นยนต์แบบ 3 มิติ และเรียลไทม์ จากผลการทดลองสามารถสรุปได้ว่าวิธีการแบบ
ไฮบริดที่ได้นำเสนอในวิทยานิพนธ์ฉบับนี้นั้นมีผลลัพธ์ที่น่าพึงพอใจเป็นอย่างมาก การเปรียบเทียบกับ
เทคนิคและวิธีการต่าง ๆ ที่ผ่านมานั้นในอดีต ได้แสดงให้เห็นถึงความแม่นยำในการตรวจจับด้วยวิธี
ไฮบริดนั้นสูงกว่า และความเร็วในการประมวลผลเหนือกว่าเทคนิคและวิธีการตรวจจับแบบอื่น ๆ
ความก้าวหน้าจากงานศึกษานี้มีความสำคัญในระบบอัตโนมัติ โดยเฉพาะอย่างยิ่งในภาคการผลิตและ
ระบบการขนส่งสินค้า เนื่องจากการใช้กล้องถ่ายภาพเชิงลึกราคานั้นไม่เพียงแต่ลดค่าใช้จ่าย แต่ยัง
ทำให้ระบบการทำงานนี้สามารถใช้งานได้จริงมากขึ้นในหลากหลายมิติ ซึ่งจะเป็นการเพิ่มความคุ้มค่า
ในด้านต่าง ๆ ในกระบวนการผลิตรวมถึงเพิ่มประสิทธิภาพในขั้นตอนการผลิตด้วย จากความสำเร็จ
และประสบการณ์ที่ได้รับจากงานวิจัยฉบับนี้ ข้อเสนอแนะสำหรับงานศึกษานี้คือควรพัฒนาเทคนิค
และวิธีการต่อไปอย่างต่อเนื่อง เพื่อเพิ่มประสิทธิภาพและความแม่นยำในการตรวจจับ รวมทั้งพัฒนา
และปรับปรุงระบบการทำงานของหุ่นยนต์ ให้มีความเร็วและความสามารถในการใช้งานใน
สถานการณ์ที่ซับซ้อนและท้าทายมากขึ้นในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] CH. Zhuang, SH. Li, and H. Ding: **Rob. Comput. Integr. Manuf.** 82 (2023) 102541. [Online]. Available: <https://doi.org/10.1016/j.rcim.2023.102541>
- [2] P. Dolezel, J. Pidanic, T. Zalabsky, and M. Dvorak: **Proc. 2019 20th International Carpathian Control Conf.** (ICCC, 2019) 1–6.
- [3] P. Torres, J. Arents, H. Marques, and P. Marques: **Electronics** 11 (2022) 476. [Online]. Available: <https://doi.org/10.3390/electronics11030476>
- [4] CH. Zhuang, ZH. Wang, H. Zhao, and H. Ding: **Rob. Comput. Integr. Manuf.** 68 (2021) 102086. [Online]. Available: <https://doi.org/10.1016/j.rcim.2020.102086>
- [5] CH. Wong, CH. Tsai, R. Chen, SH. Chien, Y. Yang, SH. Wong, and CH. Yeh: **IEEE Access** 10 (2022) 65257. [Online]. Available: <https://doi.org/10.1109/ACCESS.2022.3182114>
- [6] R. Monica, A. Saccuti, J. Aleotti, and M. Lippi: **Proc. 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA, 2022)** 1–8.
- [7] C. G. Harris and M. J. Stephens: **In Alvey Vision Conf.** (1988). [Online]. Available: <https://api.semanticscholar.org/CorpusID:1694378>
- [8] D. G. Lowe: **Int. J. Comput. Vision**, 60(2) (2004) 91. [Online]. Available: <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- [9] H. Bay, T. Tuytelaars, and L. G. Van: **Proc. 2006 European Conference on Computer Vision (ECCV, 2006)** 404–417.
- [10] Features from Accelerated Segment Test (FAST). [Online]. Available: <https://api.semanticscholar.org/CorpusID:17031649> (accessed 2011).
- [11] S. Leutenegger, M. Chli, and R. Y. Siegwart: **2011 International Conference on Computer Vision (ICCV, 2011)** 2548–2555.
- [12] E. Rosten, and T. Drummond: **Proc. 9th European Conference on Computer Vision (ECCV, 2006)** 430–443.
- [13] D. Wei: **Deep learning essentials: your hands-on guide to the fundamentals of deep learning and neural network modeling**, Bhardwaj A., W. Jianing (Packt Publishing), 2018.
- [14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: **Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR, 2016)** 779–788

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [15] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng: **Proc. 2009 IEEE International Conference on Robotics and Automation** (ICRA, 2009). [Online]. Available: <https://api.semanticscholar.org/CorpusID:6324125>
- [16] H.R. Kam, SH. Lee, T. Park, and CH. Kim: **Telecommun. Syst.** 60 (2015) 337. [Online]. Available: <https://doi.org/10.1007/s11235-015-0034-5>
- [17] Find-Object: <http://introlab.github.io/find-object> (accessed 2011).
- [18] T. Le and C. Lin: **Sensor** 19 (2019) 3602. [Online]. Available: <https://doi.org/10.3390/s19163602>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การผสมผสานเทคนิคการตรวจจับวัตถุเพื่อพัฒนาการทำงานของ
หุ่นยนต์และกระบวนการโดยรวมในอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การผสมผสานเทคนิคการตรวจจับวัตถุเพื่อพัฒนาการทำงานของ หุ่นยนต์และกระบวนการโดยรวมในอุตสาหกรรม

ในภายวิจัยนี้ได้นำเสนอวิธีแบบไฮบริดหรือการผสมผสานเทคนิคเพื่อพัฒนาการทำงานของหุ่นยนต์ในการหยิบวัตถุในภาชนะ โดยเฉพาะการนำเสนอวิธีการทำงานของกล้องร่วมกับหุ่นยนต์ และอัลกอริทึมปัญญาประดิษฐ์ วิธีการแบบไฮบริดนี้ช่วยเพิ่มความแม่นยำและความเร็วในการทำงานของหุ่นยนต์ในสถานการณ์แบบเรียลไทม์ เทคนิคที่นำมาผสมผสานนั้นคือการใช้เทคโนโลยีการตรวจจับลักษณะเด่นด้วยเทคนิค FAST และการอธิบายลักษณะเด่นด้วยเทคนิค BRISK เพื่อทำการประเมินมุมมองของวัตถุอย่างมีประสิทธิภาพและแม่นยำ

การทดลองในงานวิจัยนี้ได้ใช้กล้องถ่ายความลึกขนาดเล็กที่มีราคาต่ำกว่าเซนเซอร์ขนาดใหญ่อื่น ๆ เพื่อใช้ในการถ่ายภาพสีและภาพความลึก เพื่อเพิ่มประสิทธิภาพในการตรวจจับวัตถุและการประเมินมุมมองของวัตถุในเวลาจริง ผลการทดลองแสดงให้เห็นถึงความแม่นยำและความเร็วของวิธีการที่แบบไฮบริดนี้ โดยสามารถใช้งานผ่านระบบปฏิบัติการหุ่นยนต์ ROS ทำให้หุ่นยนต์และอุปกรณ์ต่าง ๆ สามารถทำงานร่วมกันได้อย่างต่อเนื่องและมีประสิทธิภาพ วิธีการแบบไฮบริดนี้เปิดโอกาสใหม่ในการใช้งานหุ่นยนต์ในระบบอัตโนมัติต่าง ๆ ที่ท้าทาย เพิ่มประสิทธิภาพของกระบวนการตรวจจับวัตถุ ช่วยลดความผิดพลาดในการหยิบวัตถุของหุ่นยนต์ นำมาซึ่งประสิทธิภาพโดยรวมของกระบวนการทำงานในอุตสาหกรรมที่สูงขึ้น

Hybrid Deep Learning and FAST–BRISK 3D Object Detection Technique for Bin-picking Application

Thanakrit Taweesoontorn, Sarucha Yanyong, and Poom Konghuayrob*

School of Engineering, Department of Robotics and AI Engineering, King Mongkut's Institute of Technology Ladkrabang, No. 1, Chalong-Kung, 1-Alley, Ladkrabang, Bangkok 10520, Thailand

(Received January 9, 2024; accepted February 28, 2024)

Keywords: computer vision, object detection and pose estimation, bin-picking application, robot operating system

In the field of industrial robotics, robotic arms have been significantly integrated, driven by their precise functionality and operational efficiency. We here propose a hybrid method for bin-picking tasks using a collaborative robot, or cobot combining the You Only Look Once version 5 (YOLOv5) convolutional neural network (CNN) model for object detection and pose estimation with traditional feature detection based on the features from accelerated segment test (FAST) technique, feature description using binary robust invariant scalable keypoints (BRISK) algorithms, and matching algorithms. By integrating these algorithms and utilizing a low-cost depth sensor camera for capturing depth and RGB images, the system enhances real-time object detection and pose estimation speed, facilitating accurate object manipulation by the robotic arm. Furthermore, the proposed method is implemented within the robot operating system (ROS) framework to provide a seamless platform for robotic control and integration. We compared our results with those of other methodologies, highlighting the superior object detection accuracy and processing speed of our hybrid approach. This integration of robotic arm, camera, and AI technology contributes to the development of industrial robotics, opening up new possibilities for automating challenging tasks and improving overall operational efficiency.

1. Introduction

In recent years, the robotics field has undergone significant advancements, resulting in transformations across various industries, including manufacturing and logistics. Notably, industrial robots have become prevalent owing to their capability to perform repetitive tasks with high accuracy, efficiency, and reliability. Concurrently, computer vision and artificial intelligence are two technologies that have particularly influenced the field of robotics. Computer vision facilitates the interpretation of visual information, such as images and videos, by machines, empowering them to execute complex tasks. In the same way, AI enables machines to learn from experience and adjust their behaviour to achieve specific goals. The integration of computer vision and AI empowers robots to perceive and interact with their environment or

*Corresponding author: e-mail: poom.ko@kmitl.ac.th

<https://doi.org/10.18494/SAM4840>

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์
ISSN 0914-4935 © MYU K.K.
<https://myukk.org/>

workspace. By utilizing cameras and AI algorithms, robots can detect and recognize objects. This makes it possible to perform complex and flexible tasks such as object detection, pick and place, and bin-picking.

Bin-picking is one of the significant applications in robotics and involves retrieving objects from a bin or container. In this application, the sensor or camera that captures the images used to identify and detect the object as well as estimate the pose of the object is very important. Several studies, including the one cited in Ref 1, have made advancements in the field of robot work with cameras. Zhuang *et al.*⁽¹⁾ demonstrated successful implementation in real-world industrial settings. However, it is important to realize that the sensor or camera technology employed in these studies, such as 3D scanners or large-depth sensor cameras, can be associated with high costs. Particularly in industrial bin-picking applications, the use of expensive sensors such as 3D scanners is common. This dependence on costly sensors is attributed to the limitations of alternative options, such as low-cost cameras. Low-cost cameras often face constraints in terms of image resolution and capturing distance, which make them insufficient for handling large workspace tasks. This limitation is highlighted in the comparison described in Ref 2.

Previous studies have explored the integration of low-cost cameras in various methods for bin-picking tasks. Torres *et al.*⁽³⁾ introduced an application utilizing low-cost cameras for picking up small objects such as car connectors, achieving 4D object detection and pose estimation. However, despite accurate object detection, the algorithm's processing time for recognizing objects and determining their pose/orientation remained lengthy. In contrast, Zhuang *et al.*⁽⁴⁾ demonstrated faster algorithmic speed, albeit with lower object detection accuracy. Conversely, Wong *et al.*⁽⁵⁾ achieved high object detection accuracy but observed limitations in object pose estimation accuracy. Remarkably, all these studies employed single-view approaches for object detection and pose estimation.

Performing object detection and pose estimation by capturing an image in a single view, such as the top view of the workspace, is challenging because of the camera's distance from the objects and results in low-resolution object images and lower accuracy in pose estimation. To overcome this issue, the eye-in-hand technique was adopted, which involves attaching the camera to the robot's end-effector. This configuration enables flexible camera movement by manipulating the robot, facilitating image capture from close view. Monica *et al.*⁽⁶⁾ clearly showed the advantage of the eye-in-hand, which allows high depth measurements from the sensor, but the detection time is too long.

The objective of this research is to enhance the speed of object detection and pose estimation within bin-picking applications while maintaining high accuracy. This is achieved through the utilization of the Dobot CR5 collaborative robot in conjunction with a low-cost depth sensor camera, specifically the Intel Realsense D435i camera. Employing the eye-in-hand technique, as shown in Fig. 1, enables the acquisition of both a comprehensive top view of the entire workspace and close-up views for specific objects, thereby enhancing object resolution. To achieve the goal of object detection and pose estimation, we utilized the following two detection approaches.

- Firstly, You Only Look Once Version 5 (YOLOv5), a state-of-the-art deep learning algorithm for object detection, was employed. By leveraging YOLOv5, objects could be detected in real-time streams, and the robot could be precisely moved closer to the targeted object.

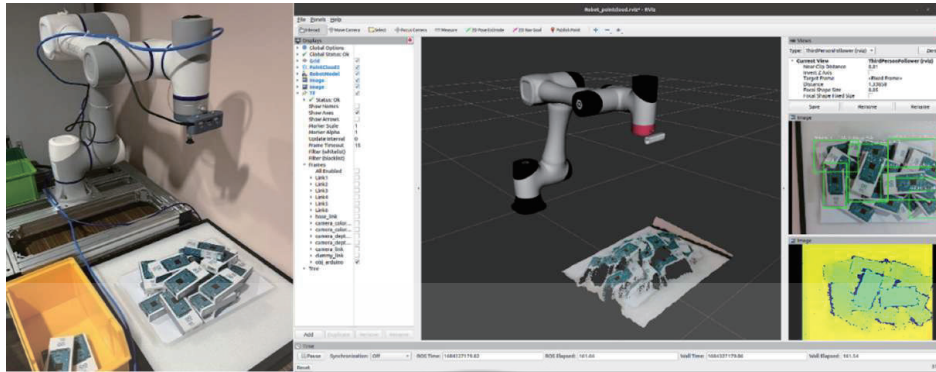


Fig. 1. (Color online) Actual robot and real-time synchronized robot simulation in Rviz with pointcloud, depth image, and detected object image.

- Secondly, the features from accelerated segment test (FAST) and binary robust invariant scalable keypoints (BRISK) algorithms, which are feature detection, description, and matching algorithms, were utilized. As the robot approached close to the object, a high-resolution object image was captured by the camera. This image was suitable for using the FAST and BRISK algorithms to rapidly estimate the object's pose. The combination of these techniques enabled high accuracy in picking up objects at the correct position and orientation. By integrating these methods, as shown in Fig. 2, our technology enhances bin-picking applications by optimizing object detection and pose estimation processes. This contributes to the overall automation capabilities of industrial robots, offering a cost-effective solution for efficient and accurate object manipulation.

2. Related Theory

2.1 Traditional feature detection, description, and matching

In the field of computer vision, significant research has been dedicated to traditional feature detection, description, and matching algorithms. The widely adopted Harris corner detector⁽⁷⁾ identifies corners by analyzing local intensity gradients. Scale invariant feature transform (SIFT)⁽⁸⁾ and speeded up robust features (SURF)⁽⁹⁾ algorithms excel in detecting scale-invariant keypoints and generating descriptors resilient to variations in scale, rotation, and viewpoint. Similarly, FAST⁽¹⁰⁾ and BRISK⁽¹¹⁾ algorithms are designed for efficient feature detection. FAST is a corner detection method that can be used to extract feature points.⁽¹²⁾ While these traditional methods have proven effective, they rely on hand-crafted features and may potentially overlook crucial information in an image. Consequently, they face limitations when it comes to recognizing objects and their poses under diverse conditions, since they require the object's image resolution to align closely with the reference image to accomplish accurate detection and recognition. In contrast, deep learning approaches have emerged as powerful tools that learn features directly from data, enabling more robust object detection, even in the presence of scale, rotation, and viewpoint variations.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

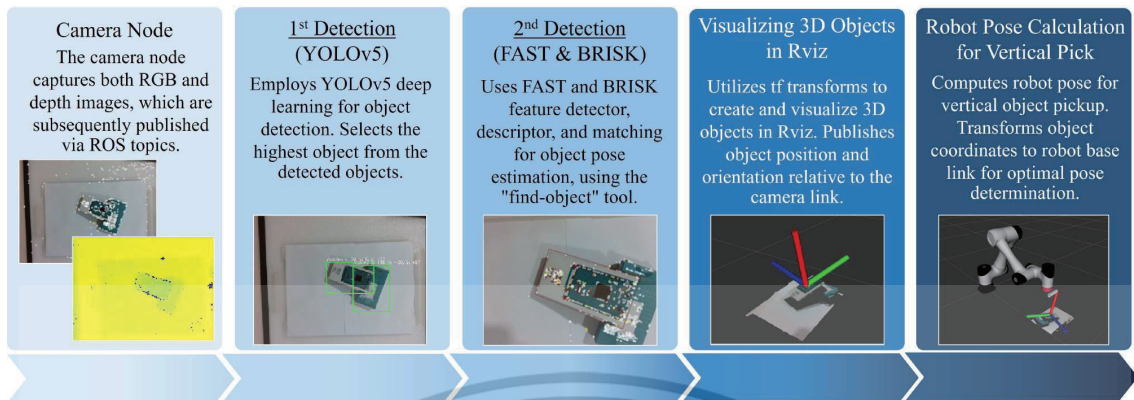


Fig. 2. (Color online) Schematic representation of the system workflow, from the camera to the detection nodes and ultimately to the robot node, highlighting the various stages involved in object detection, pose estimation, and robot pose computation.

2.2 Deep learning and convolutional neural network (CNN)

Deep learning has revolutionized various computer vision tasks, including object detection and recognition. CNNs have emerged as powerful models for analyzing image data, as they can automatically learn hierarchical representations from raw pixels.⁽¹³⁾ CNNs have demonstrated their effectiveness in object detection tasks, providing accurate bounding box predictions and advancing applications in robotics, surveillance, and autonomous vehicles. One popular CNN-based approach for object detection is YOLO.⁽¹⁴⁾ YOLO introduced a unified architecture that predicts object bounding boxes and class probabilities in a single pass, enabling real-time object detection.

2.3 Robot operating system (ROS) framework and visualization tool, Rviz

ROS has made significant contributions to the field of robotics by providing a versatile framework that facilitates seamless communication and collaboration among various components. In a foundational paper,⁽¹⁵⁾ the fundamental principles and architectural design of the ROS were discussed, and it was established as an open-source operating system. This influential work has played a pivotal role in the widespread adoption and advancement of ROS within the robotics community. Researchers leverage the capabilities of ROS to effortlessly integrate cameras with computers and robots, enabling innovative research and development in the field of robotics. Within the ROS framework, ROS visualization, or Rviz, stands out as an exceptional 3D visualization tool. Rviz offers real-time visualization capabilities for various data types, such as robot models, images, pointcloud, and trajectories. Kam *et al.*⁽¹⁶⁾ extensively explored the practical application of Rviz in constructing mobile robots. Their paper showcases Rviz's crucial role in the visualizing robot. In this study, we emphasize Rviz's significant contribution to the real-time visualization and evaluation functions of robotic systems, which further advances the capabilities of the ROS framework.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Object Detection and Pose Estimation

To detect objects and estimate their poses, the system employs two detection techniques, as shown in Fig. 2. Firstly, the robot positions itself in a top-view configuration, enabling the camera to capture the entire workspace for comprehensive coverage. This top-view image is processed by the first detection node, which utilizes a trained YOLOv5 model for object detection. By comparing the positions of all detected objects, the algorithm selects the object with the highest vertical position. Subsequently, the robot moves closer to the selected object while maintaining a fixed distance of 200 mm above the object, as shown in Fig. 3. This proximity allows for capturing detailed information. A close-up image of the object is then forwarded to the second detection node. This node employs algorithms to detect the object and estimate its pose, including its x , y , z coordinates and x , y , z orientation in 6D space.

3.1 First detection (YOLOv5)

The first technique (YOLOv5) is a state-of-the-art object detection framework based on deep learning CNNs. YOLOv5 (Fig. 4) is widely recognized for its high accuracy and real-time performance. By simultaneously predicting bounding boxes and class probabilities for multiple objects within an input image, YOLOv5 achieves efficient and precise object detection.

To train the YOLOv5 model, a dataset was prepared using Roboflow. The dataset was carefully labelled, and various augmentation techniques were applied to enhance its diversity and quality. During the training process, the YOLOv5l model was utilized with 100 epochs, batch size of 16, and the training dataset consisting of 1431 images. The example training dataset with augmentation techniques and training output results are shown in Fig. 5.

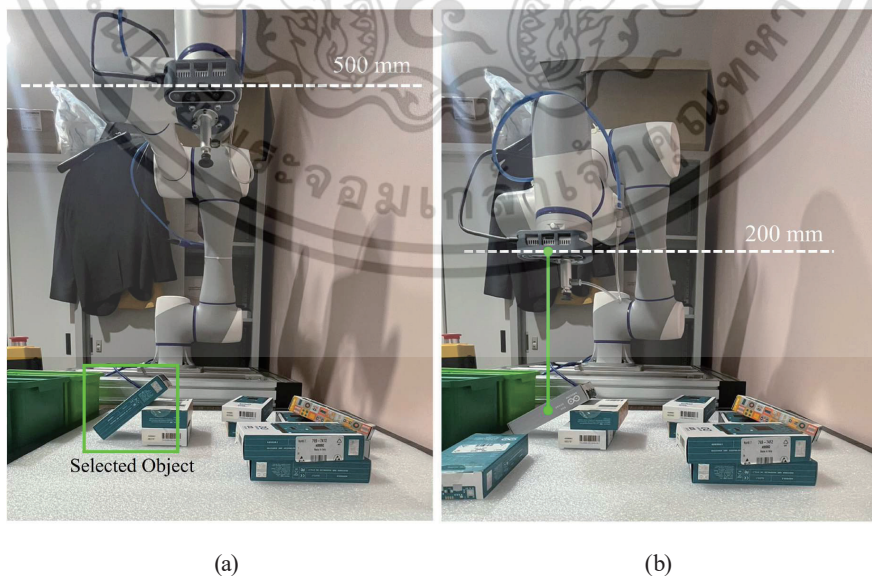


Fig. 3. (Color online) (a) Robot's top-view position and (b) robot's position as it moves closer to the selected object.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

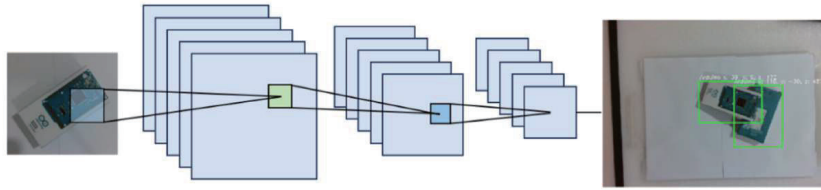


Fig. 4. (Color online) Schematic representation of a simple concept of CNNs.

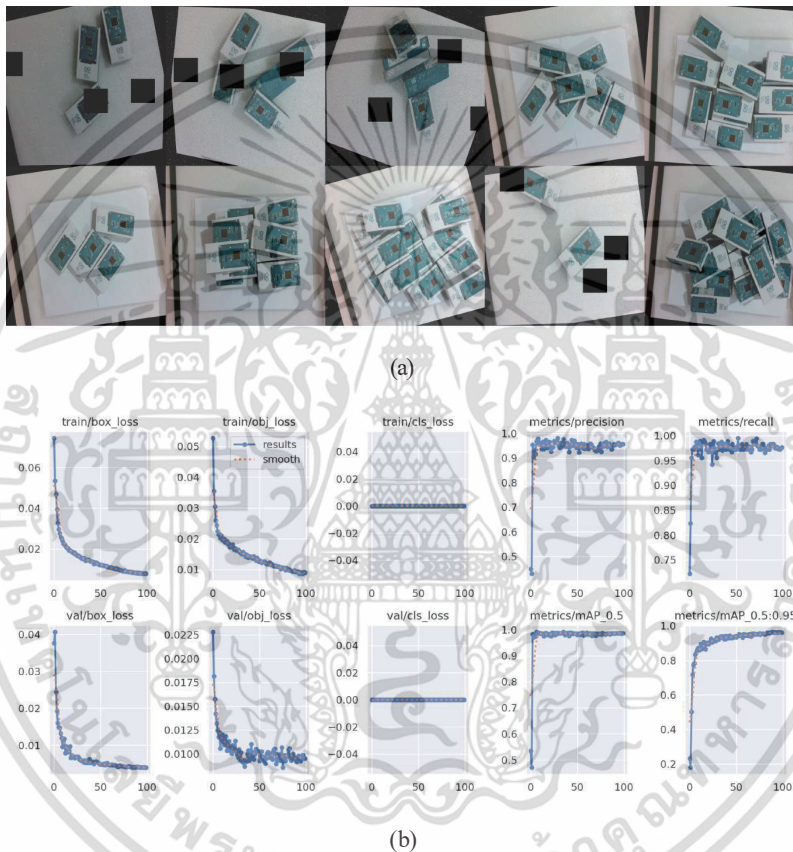


Fig. 5. (Color online) (a) Training dataset with augmentation techniques. (b) Training output results using the YOLOv5l algorithm.

Table 1 presents the object detection training results for the YOLOv5l algorithm, showcasing high accuracy metrics including precision, recall, mAP50, and mAP50-95 indices. These evaluation metrics reflect the system’s capability to detect all objects within the workspace when the robot is positioned in the top view. In scenarios where multiple objects are detected, the system selects the object with the highest vertical position based on depth information from the camera node. Subsequently, the robot is directed to move above the selected object, maintaining a fixed distance. Subsequently, the second detection technique is employed to estimate the pose using a close-up image.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1
Object detection training result output.

Algorithm	Precision	Recall	mAP50	mAP50-95
YOLOv5l	0.955	0.983	0.986	0.965

3.2 Second detection (FAST and BRISK)

In addition to YOLOv5, the second technique employed in this research is a combination of FAST as the feature detector and BRISK as the feature descriptor.

3.2.1 Feature detector—FAST

Feature detection is a fundamental computer vision technique that identifies distinctive points within an image for further analysis. FAST, a highly efficient algorithm designed to detect corner points, was employed. FAST compares the intensities of pixels in a circular neighborhood surrounding a candidate point, evaluating whether enough contiguous pixels are either brighter or darker than the candidate pixel.

The algorithm utilizes a circle consisting of 16 pixels, labelled clockwise from 1 to 16, within which a candidate point is assessed, as shown in Fig. 6. By comparing the intensities of the candidate pixel (I) with a threshold value (t), FAST applies the following conditions for corner classification:

- Condition 1: If a set of (N) contiguous pixels (S) within the circle satisfies $\forall x \in S: I_x > I_p + t$, the candidate point is classified as a corner.
- Condition 2: If a set of (N) contiguous pixels (S) within the circle satisfies $\forall x \in S: I_x < I_p - t$, the candidate point is classified as a corner.

The selection of appropriate values for N and the threshold (t) involves consideration of the number of detected corner points and computational efficiency; typically, N is set to 12. Additionally, a high-speed test can be applied to exclude noncorner points, further improving the algorithm's overall performance. FAST's rapid intensity comparison process enables real-time corner keypoint detection with high accuracy and efficiency.

3.2.2 Feature descriptor—BRISK

Once keypoints are detected, it is essential to describe their visual attributes in a compact and distinctive technique. This is where feature descriptors come into play. In this research, the BRISK algorithm is utilized as the feature descriptor. BRISK captures the unique characteristics of keypoints by encoding their local appearance, texture, and shape information into a compact binary representation. This binary descriptor ensures robustness against image transformations and enables the efficient matching of keypoints, enhancing the accuracy and reliability of the object pose estimation process.

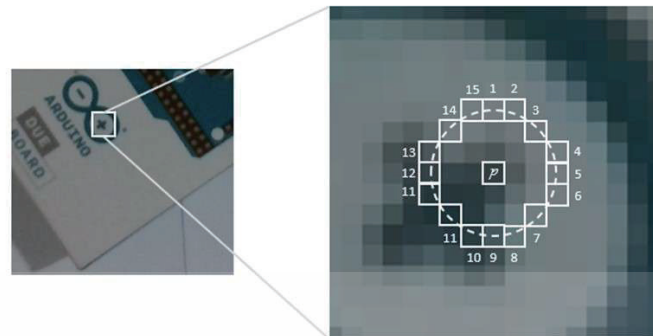


Fig. 6. (Color online) Schematic of the keypoint being evaluated, along with the surrounding 16 pixels arranged in a circular pattern.

3.2.3 Feature matching

Feature matching techniques are used to establish correspondence between keypoints in the reference and query images. The brute force strategy is applied for nearest neighbour matching, wherein the feature descriptors are compared to find suitable matches. Homography estimation techniques are then employed to determine the geometric transformation between the images. This estimation accurately calculates the object's pose, including its position and orientation.

By integrating FAST and BRISK, the system achieves robust and accurate object pose estimation. Leveraging the “find-object” tool,⁽¹⁷⁾ which incorporates feature detection algorithms such as SIFT, FAST, ORB, and feature description, results in effective object identification in images and real-time streams. The “find-object” tool utilizes advanced techniques such as RANSAC and homography functions to accurately estimate the pose of objects, even in challenging scenarios involving perspective transformations.

By sequentially employing these techniques, the system excels in object detection and pose estimation. The initial top-view detection provides comprehensive workspace coverage, while the subsequent close-up detection offers detailed information for precise pose estimation. This seamless integration within a ROS framework ensures efficient object detection and pose estimation, as depicted by the 6D object link shown in Fig. 7, serving as a reference for the object's position and orientation required for picking up.

3.3 Robot pose calculation and determination

To enable precise vertical object manipulation, the robot's picking-up pose was calculated with a fixed offset from the target object. The pose computation involved matrix-based calculations, referencing the robot frame for accurate positioning within the ROS framework. By aligning itself with the object in a controlled and efficient manner, the robot could enhance the overall performance and reliability of the bin-picking system.

Within the ROS framework, the object link was established to reference the camera link integrated into the robot's URDF file. This connection allowed for continuous derivation of the

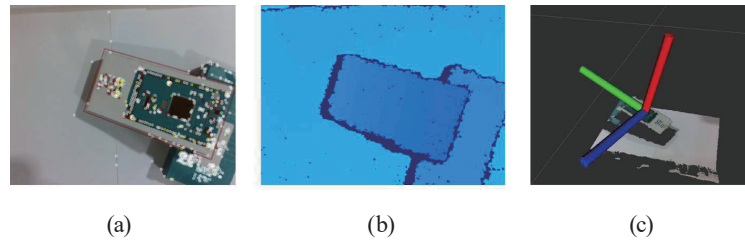


Fig. 7. (Color online) (a) RGB image with feature detection from FAST. (b) Depth image represented as a heat map. (c) Object link visualized in Rviz.

object's position from the robot's base link, enabling accurate visualization of both the robot and the object within the Rviz (Fig. 8). This integration provided a comprehensive and dynamic representation of the robot-object configuration for research purposes.

The picking pose, denoted as $Robot_{xyz}$, was determined using matrix transformations based on the object's position Obj_x, Obj_y, Obj_z and orientation $Obj_{rx}, Obj_{ry}, Obj_{rz}$. The calculations for the robot's picking-up position on each axis $Robot_x, Robot_y, Robot_z$ were computed using the following equations.

$$Robot_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(R_y) & -\sin(R_y) \\ 0 & \sin(R_y) & \cos(R_y) \end{bmatrix} \quad (1)$$

$$Robot_y = \begin{bmatrix} \cos(R_x) & 0 & \sin(R_x) \\ 0 & 1 & 0 \\ -\sin(R_x) & 0 & \cos(R_x) \end{bmatrix} \quad (2)$$

$$Robot_z = \begin{bmatrix} \cos(R_z) & -\sin(R_z) & 0 \\ \sin(R_z) & \cos(R_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Equations (1)–(3) represent the calculations for the robot's picking-up position on the respective axes. The offset in the picking-up position is denoted as $Robot_{off}$ and can be represented as

$$Robot_{off} = [0 \ 0 \ -d]. \quad (4)$$

The final calculation of the robot's picking position denoted as $Robot_{xyz}$ is given by

$$Robot_{xyz} = Robot_x \cdot Robot_y \cdot Robot_z \cdot Robot_{off} + [Obj_x \ Obj_y \ Obj_z]. \quad (5)$$

In Eq. (5), the position $Robot_{xyz}$ is added to the object position $[Obj_x \ Obj_y \ Obj_z]$ to determine the picking-up position with offset. As a result, the robot stays at a distance of d mm from the object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

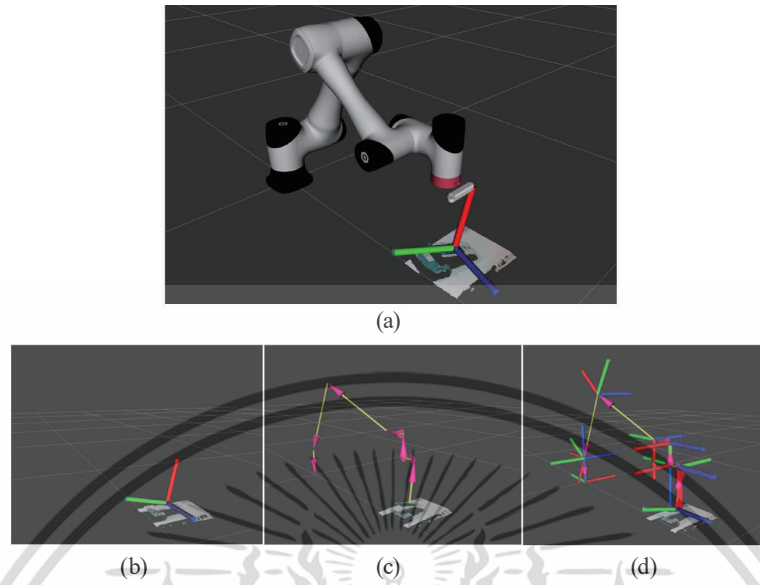


Fig. 8. (Color online) Robot pose visualization in Rviz (a). (b)–(d) Continuous derivation of the object's position from the robot's base link.

before moving directly towards it in a vertical manner within the object's plane. Additionally, the orientation of the robot's end-effector can be calculated as below.

$$Robot_{rx} = Obj_{rx} + 180 \quad (6)$$

$$Robot_{ry} = -(Obj_{ry} - 180) \quad (7)$$

$$Robot_{rz} = Obj_{rz} + 180 \quad (8)$$

Equations (6)–(8) provide the calculations for determining the values of $Robot_{rx}$, $Robot_{ry}$, and $Robot_{rz}$, which represent the orientation of the robot's end-effector. By incorporating these calculations, the robot can achieve the desired picking-up pose. This approach ensures that the robot is properly positioned and oriented for efficient and controlled object manipulation, as shown in Fig. 9.

In summary, our proposed hybrid method demonstrates remarkable capabilities in terms of object detection and pose estimation. As the comparison in Table 2, the hybrid method approach surpasses traditional methods in detecting objects across all distances and orientations. Moreover, it provides accurate 6D pose estimation, along with the precise specification of x , y , and z coordinates.

4. Experiments and Results

In the experiment, the implemented algorithm successfully detected the target object, and the object link was displayed in Rviz, along with the robot model, camera link, point cloud, depth image, and detected object image, as shown in Fig. 10.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

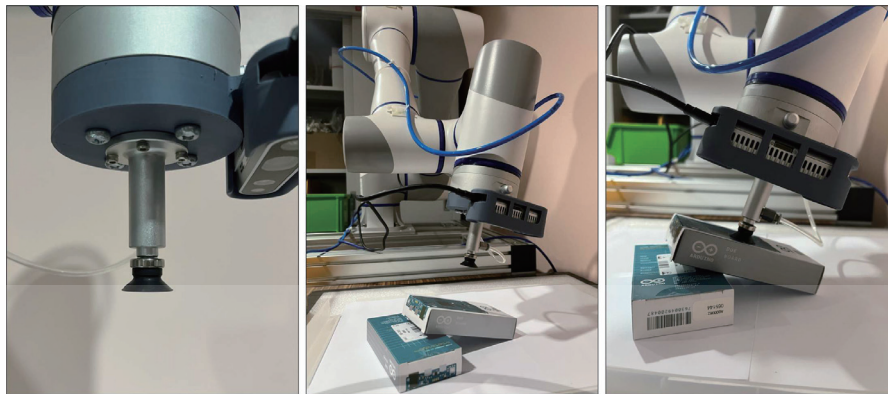


Fig. 9. (Color online) Photographs of the picking-up step, showing precise vertical pick-up direction and perfect articulation of the suction cup edge with the object.

Table 2
Detection capability.

	Our Hybrid Method	YOLOv5	FAST & BRISK	Torres <i>et al.</i> ⁽³⁾	Wong <i>et al.</i> ⁽⁵⁾
Detectable at any distance in workspace	✓	✓		✓	✓
Detectable at any orientation	✓	✓		✓	✓
6D pose estimation	✓		✓		✓
<i>x</i> and <i>y</i> coordinate specified	✓	✓	✓	✓	✓
<i>z</i> coordinate specified	✓		✓	✓	✓

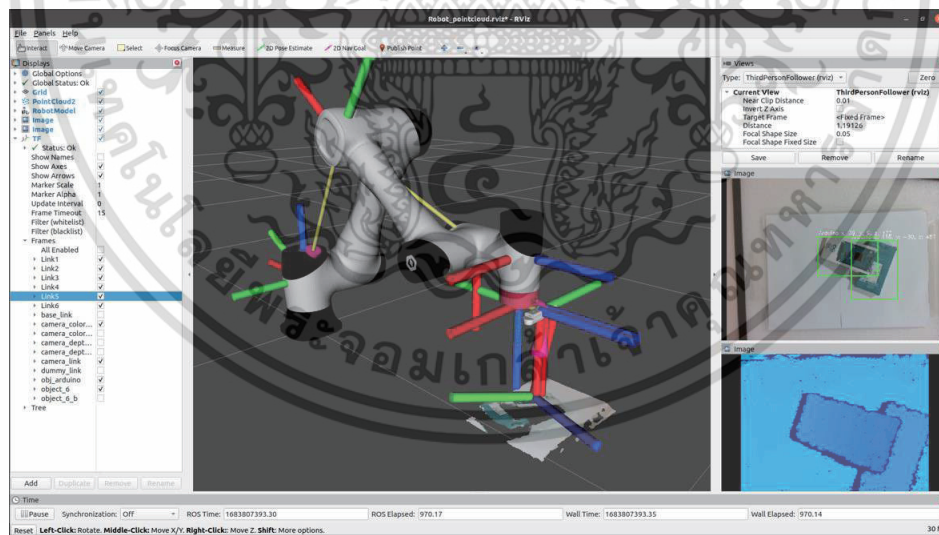


Fig. 10. (Color online) Rviz, the ROS visualization tool, displays the object link, robot model with the camera link, point cloud, depth image, and detected object image.

The performance of the bin-picking system was evaluated through an experiment that encompassed object detection, pose estimation, robot positioning for vertical pick-up, and real-time visualization of robot synchronization. The bin-picking task was divided into three

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

subtasks, as shown in Fig. 11, to assess different scenarios: well-structured object positions, semi-randomized objects with slight rotation, and randomized object positions and orientations.

- (1) In the first subtask, objects were neatly arranged in a well-structured position within the container with the aim of testing the system's accuracy in detecting and estimating object poses.
- (2) The second subtask introduced semi-randomized objects with slight rotation, challenging the system to handle variations in object orientation.
- (3) The third subtask involved randomly positioned and oriented objects, representing a more challenging scenario.

The hybrid technique considerably extends the object detection range, leveraging YOLOv5 capabilities to detect objects at a greater distance, resulting in a larger detectable area or workspace. This expanded detection area is evident in Table 3 and emphasizes the hybrid method's superiority over traditional techniques. Notably, our hybrid method achieves a detectable range that extends to the maximum reach of the robot arm.

The system successfully detected the object by the hybrid method, and the Rviz effectively depicted the actual robot and object link, showcasing the real-world implementation of the system (Fig. 12). This visualization enabled the precise tracking of the object's position and orientation, ensuring accurate manipulation and interaction. The seamless integration of the robot and Rviz facilitated smooth monitoring and control of the object detection and pose estimation process, confirming the system's effectiveness and reliability.

In terms of experimental outcomes, as visually depicted in each process showcased in Fig. 13, the system achieved a 100% single-attempt success rate in the first subtask, while maintaining a

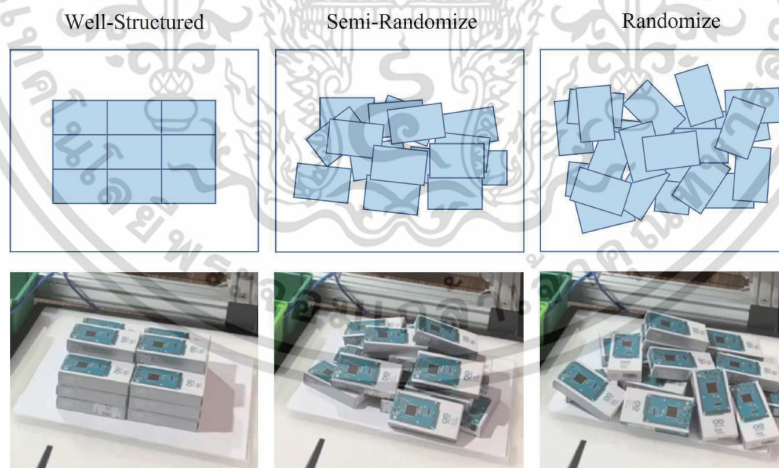


Fig. 11. (Color online) Bin-picking subtasks: well-structured, semi-randomized, and randomized arrangements.

Table 3

Detectable range for each technique.

Technique	Detectable Range (mm)
FAST & BRISK	150–280
YOLOv5 + FAST & BRISK	150–730

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

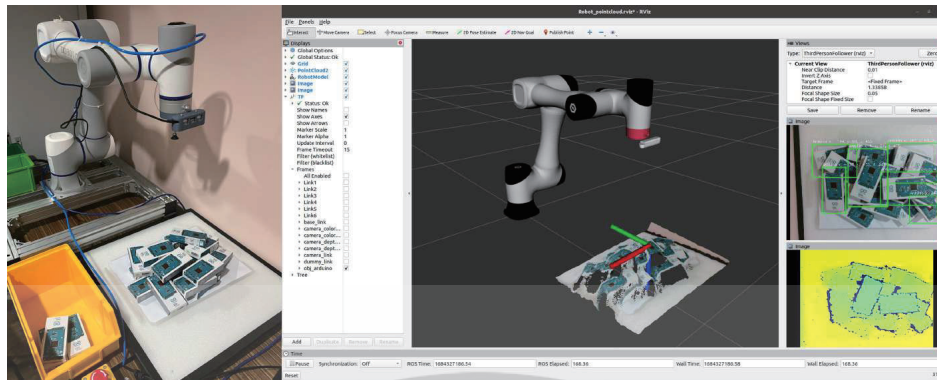


Fig. 12. (Color online) Actual robot and Rviz with target object link display, showcasing the system's real-world implementation and visualization.

Table 4
Our proposed hybrid method operation time.

	First Subtask average time (s)	Second Subtask average time (s)	Third Subtask average time (s)
Object detection, positioning and selecting	0.12	0.12	0.12
Robot moving above object	5.30	5.84	5.79
6D Pose / orientation estimation	0.20	0.20	0.20
Picking pose calculation and operation	11.24	12.60	16.84
Total	16.86	18.76	22.95

Table 5
Comparison between the previous methods.

Algorithms	Experimental object	Sensor	Object detection		Pose estimation (5°)	Detection and pose estimation time (s)
			AP ₅₀	AP _{50:95}		
Torres <i>et al.</i> ⁽³⁾	Connector part	Low-cost	98%	—	—	7.4
Zhuang <i>et al.</i> ⁽⁴⁾	Tissue Box	Low-cost	92.7%	—	95%	—
Wong <i>et al.</i> ⁽⁵⁾	Box	Low-cost	98.9%	—	87.9%	—
Le and Lin ⁽¹⁸⁾	USB Packs	Low-cost	100%	91.2%	—	0.862
Our Hybrid	Arduino Box	Low-cost	98.6%	96.5%	94.7%	0.32

97.5% success rate in the second subtask. However, the success rate decreased slightly to 86.67% in the last subtask. Table 4 provides a breakdown of the operation times for each subtask, including object detection, robot movement, pose estimation, picking pose calculation, and overall operation time.

In the experimental result and comparison, our hybrid method showed impressive performance, as summarized in Table 5. In terms of object detection, our approach achieved an AP_{50:95} score of 96.5%, underscoring its capabilities in accurately detecting objects at any orientation, and a pose estimation accuracy of 94.7% within angle error of 5°. Additionally, our hybrid method demonstrated efficient processing, with a minimal detection and pose estimation time of merely 0.32 s, indicating its suitability for real-time applications. The inclusion of real-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

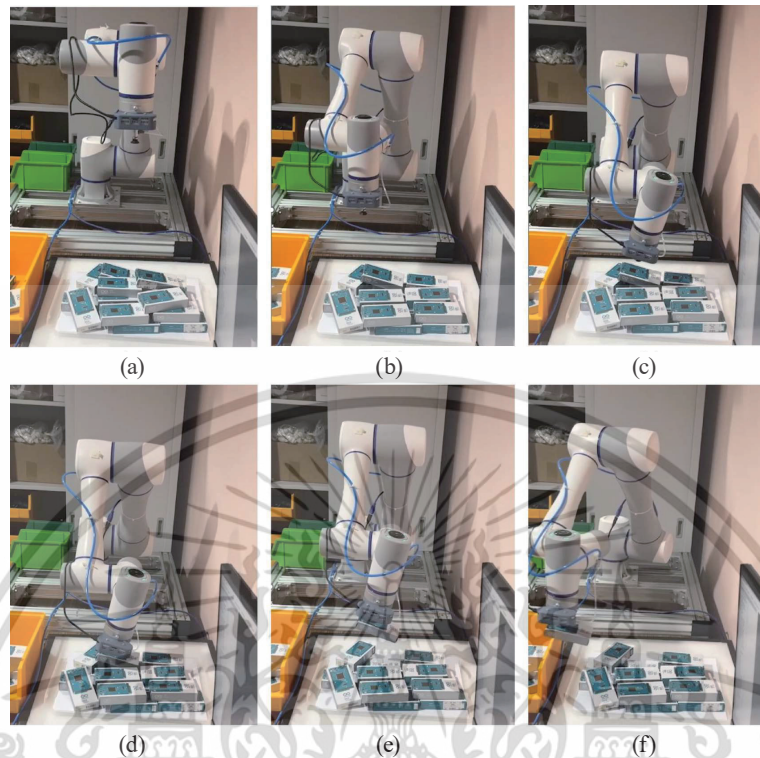


Fig. 13. (Color online) The sequence of pictures demonstrates the successful execution of the picking-up process. (a) The robot is positioned in the top-view configuration. (b) The robot moves for a close-up view of the target object. (c) The robot approaches the object with a fixed offset. (d) The robot initiates the picking-up motion. (e) The robot successfully picks up the object. (f) The robot transports the object to the storage crate. This series of images showcases the systematic and precise execution of the picking-up task by the robot.

time 3D visualization depicting the robot's synchronization with objects enhances the clarity and convenience of monitoring and controlling the system during operation. This result showing the significant contributions of our work in object detection and pose estimation in industrial robotics paves the way for enhanced automation across real-world applications.

5. Conclusion

We presented a comprehensive system for object detection, pose estimation with a low-cost depth camera, and robot positioning. The system was implemented using the eye-in-hand configuration in bin-picking tasks within the ROS framework. Through the integration of the YOLOv5, FAST, and BRISK algorithms, our system achieves accurate and efficient detection and pose estimation, even in challenging scenarios. Furthermore, our system offers real-time 3D visualization of robot movements, providing insights into its operations. The experimental results confirmed a remarkable improvement in both detection accuracy and processing speed over other methods. This advancement is significant in automation, particularly in the manufacturing and logistics sectors. By utilizing a low-cost depth camera, we not only reduce expenses but also make our system more practical for real-world use, ultimately boosting productivity and efficiency in industrial operations.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Acknowledgments

This research work is supported by the Department of Robotics and AI Engineering, School of Engineering, at King Mongkut's Institute of Technology Ladkrabang. The authors extend their appreciation to the members of the AIMs lab for their invaluable assistance and support in various aspects of this work.

References

- 1 C. H. Zhuang, S. H. Li, and H. Ding: *Rob. Comput. Integr. Manuf.* **82** (2023) 102541. <https://doi.org/10.1016/j.rcim.2023.102541>
- 2 P. Dolezel, J. Pidanic, T. Zalabsky, and M. Dvorak: *Proc. 2019 20th Int. Carpathian Control Conf. (ICCC, 2019)* 1–6.
- 3 P. Torres, J. Arents, H. Marques, and P. Marques: *Electronics* **11** (2022) 476. <https://doi.org/10.3390/electronics11030476>
- 4 C. H. Zhuang, Z. H. Wang, H. Zhao, and H. Ding: *Rob. Comput. Integr. Manuf.* **68** (2021) 102086. <https://doi.org/10.1016/j.rcim.2020.102086>
- 5 C. H. Wong, C. H. Tsai, R. Chen, S. H. Chien, Y. Yang, S. H. Wong, and C. H. Yeh: *IEEE Access* **10** (2022) 65257. <https://doi.org/10.1109/ACCESS.2022.3182114>
- 6 R. Monica, A. Saccuti, J. Aleotti, and M. Lippi: *Proc. 2022 IEEE 27th Int. Conf. Emerging Technologies and Factory Automation (ETFA, 2022)* 1–8.
- 7 C. G. Harris and M. J. Stephens: In *Alvey Vision Conf.* (1988). <https://api.semanticscholar.org/CorpusID:1694378>
- 8 D. G. Lowe: *Int. J. Comput. Vision* **60** (2004) 91. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- 9 H. Bay, T. Tuytelaars, and L. G. Van: *Proc. 2006 European Conf. Computer Vision (ECCV, 2006)* 404–417.
- 10 Features from Accelerated Segment Test (FAST): <https://api.semanticscholar.org/CorpusID:17031649> (accessed 2011).
- 11 S. Leutenegger, M. Chli, and R. Y. Siegwart: *2011 Int. Conf. Computer Vision (ICCV, 2011)* 2548–2555.
- 12 E. Rosten and T. Drummond: *Proc. 9th European Conf. Computer Vision (ECCV, 2006)* 430–443.
- 13 A. Krizhevsky, I. Sutskever, and G. E. Hinton: *Commun. ACM* **60** (2012) 84. <https://doi.org/10.1145/3065386>
- 14 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR, 2016)* 779–788.
- 15 M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng: *Proc. 2009 IEEE Int. Conf. Robotics and Automation (ICRA, 2009)* <https://api.semanticscholar.org/CorpusID:6324125>
- 16 H. R. Kam, S. H. Lee, T. Park, and C. H. Kim: *Telecommun. Syst.* **60** (2015) 337. <https://doi.org/10.1007/s11235-015-0034-5>
- 17 Find-Object: <http://introlab.github.io/find-object> (accessed 2011).
- 18 T. Le and C. Lin: *Sensor* **19** (2019) 3602. <https://doi.org/10.3390/s19163602>

About the Authors



Thanakrit Taweesoontorn received his B.S. degree from King Mongkut's Institute of Technology Ladkrabang, Thailand (KMITL), in 2021. From 2021 to 2023, he was a teacher's assistant at KMITL, Thailand. Since 2023, he has been a remote service engineer at E80 Group. His research interests are in industrial robots, machine vision, and artificial intelligence. (romthanakrit@gmail.com)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Sarucha Yanyong received his B.S. degree in 2012 and his M.S. and Ph.D. degrees in 2014 and 2023, respectively, from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand. Since 2023, he has been a professor at KMITL. His research interests are in mobile robotics, embedded systems, and artificial intelligence. (sarucha.ya@kmitl.ac.th)



Poom Konghuayrob received his B.S. degree in 2012 and his M.S. and Ph.D. degrees in 2013 and 2017, respectively, from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand. Since 2017, he has been a professor at KMITL. His research interests are in industrial robots, automation systems, manufacturing, adaptive control systems, and artificial intelligence. (poom.ko@kmitl.ac.th)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล นาย ธนกฤต ทวีสุนทร
วัน เดือน ปีเกิด 3 สิงหาคม 2541 ที่กรุงเทพมหานคร
ที่อยู่ บ้านเลขที่ 4/562 แขวงคลองกุ่ม เขตบึงกุ่ม กรุงเทพมหานคร 10240
ประวัติการศึกษา:
2563 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอัตโนมัติ (เกียรตินิยมอันดับ
2) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์การทำงาน
2566 วิศวกร บริษัท E80 SEA Co., Ltd.

ผลงานวิจัย:
2567 Sensors and Materials, Vol. 36, No. 4 (2024) 1389-1404



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้