

การพัฒนาระบบตรวจจับและคาดการณ์ตำแหน่งรถด้วยระบบกล้อง  
ภายนอกเพื่อเชื่อมข้อมูลสำหรับการตัดสินใจให้นายยนต์อัตโนมัติ  
อย่างปลอดภัย



นายภูมิมินทร์ พินพิมาย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
มหาวิทยาลัยเทคโนโลยีสุรนารี  
ปีการศึกษา 2567

DEVELOPMENT OF VEHICLE DETECTION AND TRAJECTORY  
PREDICTION SYSTEM USING EXTERNAL CAMERA SYSTEM FOR  
CONNECTED-DATA DECISION MAKING TO SAFELY DRIVING OF  
AUTONOMOUS ROVER



A Thesis Submitted in Partial Fulfillment of the Requirements for the  
Degree of Master of Engineering in Mechatronics Engineering  
Suranaree University of Technology  
Academic Year 2024

การพัฒนาระบบตรวจจับและคาดการณ์ตำแหน่งรถด้วยระบบกล้องภายนอก  
เพื่อเชื่อมข้อมูลสำหรับการตัดสินใจให้ยานยนต์อัตโนมัติอย่างปลอดภัย

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรปริญญาโทบริหารธุรกิจ

คณะกรรมการสอบวิทยานิพนธ์



(รศ. ดร.จิระพล ศรีเสรีสุผล)

ประธานกรรมการ

สุรเดช ตัญตริย์รัตน์

(ผศ. ดร.สุรเดช ตัญตริย์รัตน์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



(ผศ. ดร.โสธรฐา แข็งการ)

กรรมการ

(ดร.สุพัฒน์ กลิ่นเขียว)

กรรมการ



(รศ. ดร.ยุพาพร รักสกุลพิวัฒน์)

รองอธิการบดีฝ่ายวิชาการและประกันคุณภาพ



(รศ. ดร.พรศิริ จงกล)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

ภูมิมินทร์ พินพิมาย : การพัฒนาระบบตรวจจับและคาดการณ์ตำแหน่งรถด้วยระบบกล้อง  
ภายนอกเพื่อเชื่อมข้อมูลสำหรับการตัดสินใจให้ยานยนต์อัตโนมัติอย่างปลอดภัย

(DEVELOPMENT OF VEHICLE DETECTION AND TRAJECTORY PREDICTION SYSTEM  
USING EXTERNAL CAMERA SYSTEM FOR CONNECTED-DATA DECISION MAKING  
TO SAFELY DRIVING OF AUTONOMOUS ROVER)

อาจารย์ที่ปรึกษา: ผู้ช่วยศาสตราจารย์ ดร.สุรเดช ตัญตริรัตน์, 98 หน้า

คำสำคัญ: จุดบอดยานยนต์อัตโนมัติ/การประมวลผลภาพ/ปัญญาประดิษฐ์

งานวิจัยนี้มุ่งเน้นการแก้ไขปัญหาจุดอับสายตาหรือจุดบอดในระบบยานยนต์อัตโนมัติซึ่งเป็นบริเวณที่เซ็นเซอร์หรือกล้องไม่สามารถตรวจจับหรือมองเห็นได้อย่างมีประสิทธิภาพปัญหานี้ส่งผลให้เกิดความเสี่ยงต่อการเกิดอุบัติเหตุและอาจทำให้ระบบควบคุมอัตโนมัติทำงานผิดพลาดจุดบอดสามารถเกิดขึ้นได้จากหลายสาเหตุเช่น การจำกัดมุมมองของเซ็นเซอร์ การบดบังของวัตถุที่ขวางอยู่และข้อจำกัดของเทคโนโลยีการมองเห็นของเครื่องจักรการวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบเสริมการมองเห็นสำหรับจุดอับสายตาในยานยนต์อัตโนมัติเพื่อเพิ่มความปลอดภัยบนท้องถนนรวมถึงเสริมสร้างความมั่นใจและการยอมรับจากผู้ใช้งาน อีกทั้งยังช่วยปรับปรุงประสิทธิภาพการทำงานของระบบขับเคลื่อนอัตโนมัติการศึกษานี้ได้นำเทคนิคต่าง ๆ มาใช้ ตั้งแต่การประมวลผลภาพไปจนถึงการเรียนรู้เชิงลึก (Deep Learning) เพื่อพัฒนาระบบตรวจจับและคำนวณวิถีการเคลื่อนที่ของวัตถุช่วยให้ระบบยานยนต์อัตโนมัติสามารถมีมุมมองที่ครอบคลุมมากขึ้นและเพิ่มประสิทธิภาพในการตรวจจับสภาพแวดล้อมโดยรอบผลการวิจัยแสดงให้เห็นว่ายานยนต์อัตโนมัติสามารถตัดสินใจและปรับตัวตามสถานการณ์ได้อย่างรวดเร็วและแม่นยำมากขึ้นซึ่งส่งผลให้เกิดความปลอดภัยและความเชื่อถือในระบบยานยนต์อัตโนมัติที่มากขึ้น

สาขาวิชา วิศวกรรมเมคคาทรอนิกส์

ปีการศึกษา 2567

ลายมือชื่อนักศึกษา.....ภูมิมินทร์ พินพิมาย.....

ลายมือชื่ออาจารย์ที่ปรึกษา.....สุรเดช ตัญตริรัตน์.....

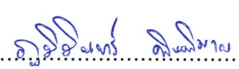
POOMMIN PHINPHIMAI : DEVELOPMENT OF VEHICLE DETECTION AND  
TRAJECTORY PREDICTION SYSTEM USING EXTERNAL CAMERA SYSTEM FOR  
CONNECTED-DATA DECISION MAKING TO SAFELY DRIVING OF AUTONOMOUS  
ROVER.

THESIS ADVISOR : ASST. PROF. SURADET TANTRAIRATN, Ph.D., 98 PP.

Keywords: Autonomous Vehicle Blind Spot/Image Processing/Artificial Intelligence

This research focuses on addressing the issue of blind spots in autonomous vehicle systems, which are areas that sensors or cameras cannot effectively detect or visualize. This problem poses a significant risk of accidents and may lead to failures in the autonomous control system. Blind spots can arise due to various factors, such as limited sensor views, obstruction by objects, and the constraints of machine vision technology. The objective of this study is to develop an enhanced vision system for blind spots in autonomous vehicles to increase road safety, boost user confidence and acceptance, and improve the overall efficiency of autonomous driving systems. This study employs various techniques, ranging from image processing to deep learning, to develop a detection system that can calculate object trajectories. This helps autonomous vehicle systems achieve a more comprehensive view of the surrounding environment, enhancing the accuracy of object detection. The results demonstrate that autonomous vehicles can make quicker and more accurate decisions and adaptations to changing situations, leading to increased safety and trust in autonomous vehicle systems.

School of Mechatronics Engineering  
Academic Year 2024

Student's Signature.....  
Advisor's Signature.....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณา แนะนำให้คำปรึกษา ตรวจสอบแก้ไข ทั้งทางด้านวิชาการและการดำเนินงานวิจัยจาก คณาจารย์ บุคลากร นักศึกษาที่เกี่ยวข้องของมหาวิทยาลัยเทคโนโลยีสุรนารี ผู้วิจัยขอขอบคุณอย่างสุดซึ้งไว้ ณ โอกาสนี้

ผู้วิจัยขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สุรเดช ตัญญูรัตน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ผู้ประสิทธิ์ประสาทวิชาความรู้ให้คำปรึกษา และคำแนะนำในการดำเนินการวิจัยทุกขั้นตอนเป็นอย่างดี ตลอดจนสนใจใส่ใจในการตรวจทานข้อบกพร่องของวิทยานิพนธ์เล่มนี้และสนับสนุนค่าใช้จ่ายสำหรับการนำเสนอผลงานวิชาการระดับนานาชาติ

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์อันประกอบด้วย รองศาสตราจารย์ ดร.จิระพล ศรีเสรีฐผล ประธานกรรมการสอบ และ กรรมการสอบวิทยานิพนธ์ทุกท่าน ซึ่งได้ตรวจทานแก้ไข และให้คำแนะนำอันมีประโยชน์ต่อความสมบูรณ์ของวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณ มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ได้มอบทุน OROG ให้แก่ผู้วิจัย สำหรับการสนับสนุนค่าเล่าเรียนในระดับบัณฑิตศึกษา รวมถึงพื้นที่ทำงานวิจัยที่อาคารเครื่องมือ F11 ทำให้ผู้วิจัยสามารถดำเนินการเก็บผลการทดลองได้ด้วยดีตลอดมา

สุดท้ายนี้ ขอขอบคุณ บิดา มารดา ที่คอยให้กำลังใจและเป็นแรงผลักดันให้ผู้วิจัยสามารถทำงานสำเร็จลุล่วงไปด้วยดีตลอดจนให้ความอนุเคราะห์ด้านทุนทรัพย์ที่ใช้สำหรับการเล่าเรียนในระดับบัณฑิตศึกษาตลอดจนจบการศึกษาในครั้งนี้

ภูมิมินทร์ พินพิมาย



# สารบัญ

หน้า

|                                                             |          |
|-------------------------------------------------------------|----------|
| บทคัดย่อ (ภาษาไทย).....                                     | ก        |
| บทคัดย่อ (ภาษาอังกฤษ).....                                  | ข        |
| กิตติกรรมประกาศ.....                                        | ค        |
| สารบัญ.....                                                 | ง        |
| สารบัญตาราง.....                                            | ช        |
| สารบัญรูป.....                                              | ซ        |
| <b>บทที่</b>                                                |          |
| <b>1 บทนำ.....</b>                                          | <b>1</b> |
| 1.1 ความสำคัญและที่มาของปัญหาการวิจัย .....                 | 1        |
| 1.2 วัตถุประสงค์ของการวิจัย.....                            | 2        |
| 1.3 ขอบเขตของการศึกษา.....                                  | 2        |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ .....                         | 3        |
| <b>2 ปรัชญ์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง.....</b>        | <b>4</b> |
| 2.1 การแปลงมุมมองภาพ.....                                   | 4        |
| 2.1.1 Homography.....                                       | 5        |
| 2.2 การแปลงระบบพิกัด.....                                   | 5        |
| 2.3 การเรียนรู้ของเครื่องจักร.....                          | 7        |
| 2.3.1 ANN (Artificial Neural Networks).....                 | 7        |
| 2.3.2 CNN (Convolutional Neural Networks).....              | 9        |
| 2.4 การติดตามวัตถุ.....                                     | 18       |
| 2.4.1 Deepsort.....                                         | 19       |
| 2.5 การระบุตำแหน่งของยานพาหนะอัตโนมัติ.....                 | 21       |
| 2.5.1 แผนที่ความละเอียดสูง (HD Maps).....                   | 22       |
| 2.5.2 การระบุตำแหน่งด้วย Normal distribution transform..... | 23       |
| 2.6 Ros (Robot Operating System) .....                      | 23       |

## สารบัญ (ต่อ)

### หน้า

|        |                                                                   |    |
|--------|-------------------------------------------------------------------|----|
| 2.6.1  | โครงสร้างและการทำงานของ ROS.....                                  | 24 |
| 2.6.2  | ซอฟต์แวร์และไลบรารีใน ROS .....                                   | 24 |
| 2.7    | Roboflow.....                                                     | 25 |
| 3      | วิธีดำเนินงานวิจัย.....                                           | 27 |
| 3.1    | การเตรียมการ.....                                                 | 28 |
| 3.1.1  | การศึกษาค้นคว้าและรวบรวมงานวิจัยที่เกี่ยวข้อง.....                | 28 |
| 3.1.2  | การเตรียมพื้นที่ที่จะทำการทดสอบ.....                              | 29 |
| 3.1.3  | ทดสอบการทำงานของกล้อง.....                                        | 31 |
| 3.1.4  | ทดสอบการทำงานของแผนที่ความละเอียดสูง .....                        | 32 |
| 3.2    | การพัฒนาระบบเพิ่มความปลอดภัยของระบบอัตโนมัติ.....                 | 33 |
| 3.2.1  | การแปลงมุมมองกล้องและการสร้างตารางเพื่อใช้อ้างอิงตำแหน่ง .....    | 33 |
| 3.2.2  | การตรวจจับวัตถุจากภาพ.....                                        | 34 |
| 3.2.3  | การสร้างโมเดลการเรียนรู้.....                                     | 36 |
| 3.2.4  | การติดตามวัตถุ.....                                               | 39 |
| 3.2.5  | การระบุตำแหน่งของวัตถุที่ตรวจจับจากกล้อง .....                    | 39 |
| 3.2.6  | การระบุตำแหน่งของรถอัตโนมัติ.....                                 | 41 |
| 3.2.7  | การแปลงพิกัดเพื่อประสานตำแหน่งของหุ่นยนต์และระบบกล้อง.....        | 42 |
| 3.2.8  | การทำนายตำแหน่ง.....                                              | 45 |
| 3.2.9  | การประมวลผลและการวิเคราะห์เพื่อหาจุดที่มีความเสี่ยงต่อการชน ..... | 45 |
| 3.2.10 | การพัฒนาระบบการชะลอความเร็วเมื่อมีการทำนายว่าจะเกิดการชน.....     | 46 |
| 3.3    | ทดสอบระบบการชะลอความเร็วเมื่อเสี่ยงเกิดการชน .....                | 47 |
| 4      | ผลการวิจัยและอภิปรายผล.....                                       | 49 |
| 4.1    | ผลการทดสอบระบบตรวจจับวัตถุ .....                                  | 49 |
| 4.2    | ผลการทดสอบระบบติดตามวัตถุ .....                                   | 62 |
| 4.3    | ผลการทดสอบการทำนายตำแหน่ง.....                                    | 64 |
| 4.3.1  | ผลการวัดตำแหน่งหลังจากการตรวจจับและติดตามวัตถุ.....               | 64 |
| 4.3.2  | ผลการฝึกฝนโมเดล ANN เพื่อทำนายตำแหน่ง.....                        | 78 |



## สารบัญ (ต่อ)

### หน้า

|       |                                                                              |    |
|-------|------------------------------------------------------------------------------|----|
| 4.4   | ผลการทดสอบการชะลอความเร็วและหยุดเมื่อเสียงเกิดการชน .....                    | 81 |
| 5     | สรุปและข้อเสนอแนะ .....                                                      | 85 |
| 5.1   | สรุปผลการวิจัย.....                                                          | 85 |
| 5.1.1 | ผลการทดสอบระบบตรวจจับวัตถุ.....                                              | 85 |
| 5.1.2 | ผลการทดสอบระบบติดตามวัตถุ.....                                               | 85 |
| 5.1.3 | ผลการทำนายตำแหน่ง.....                                                       | 86 |
| 5.1.4 | ผลการทดสอบการชะลอความเร็วและหยุดเมื่อเสียงเกิดการชน .....                    | 86 |
| 5.1.5 | การประยุกต์ใช้ในระบบอัตโนมัติ.....                                           | 86 |
| 5.2   | ข้อเสนอแนะ.....                                                              | 87 |
| 5.2.1 | การพัฒนาความแม่นยำของการตรวจจับและติดตามวัตถุในสถานะที่ซับซ้อนมากขึ้น .....  | 87 |
| 5.2.2 | การรวมข้อมูลจากหลายเซ็นเซอร์ (Sensor Fusion).....                            | 87 |
| 5.2.3 | การพัฒนาความเร็วในการประมวลผล .....                                          | 87 |
| 5.2.4 | การประเมินผลการทำงานของระบบในสถานะกลางคืนหรือแสงน้อย .....                   | 87 |
| 5.2.5 | การนำปัญญาประดิษฐ์มาปรับปรุงการตัดสินใจของระบบ .....                         | 87 |
| 5.2.6 | การปรับปรุงโมเดล ANN.....                                                    | 88 |
| 5.2.7 | การทดสอบในสถานการณ์จริงและการใช้งานภาคสนาม .....                             | 88 |
| 5.2.8 | การพัฒนาและทดสอบกับวัตถุที่มีลักษณะและขนาดที่แตกต่างกัน.....                 | 88 |
| 5.2.9 | การพัฒนาให้ระบบสามารถทำงานร่วมกับระบบควบคุมการเคลื่อนที่อื่น ๆ .....         | 88 |
|       | รายการอ้างอิง.....                                                           | 89 |
|       | ภาคผนวก ก. รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา..... | 93 |
|       | ประวัติผู้เขียน.....                                                         | 98 |

## สารบัญตาราง

| ตารางที่ | หน้า                                                                                                                          |
|----------|-------------------------------------------------------------------------------------------------------------------------------|
| 2.1      | ทดสอบการตรวจจับ Pascal VOC 2007 YOLOv2 เร็วและแม่นยำกว่าการตรวจจับอื่น... 14                                                  |
| 4.1      | แสดงผลลัพธ์การฝึกฝนโมเดล..... 49                                                                                              |
| 4.2      | ทดสอบการทดลองการติดตามวัตถุ ..... 62                                                                                          |
| 4.3      | เปรียบเทียบของตำแหน่งจริงและตำแหน่งที่วัดได้ของการระบุตำแหน่งของวัตถุที่<br>ตรวจจับจากกล้อง..... 65                           |
| 4.4      | การเปรียบเทียบระหว่างตำแหน่งจริงและตำแหน่งที่วัดได้พร้อมกับค่าความคลาดเคลื่อน.. 67                                            |
| 4.5      | การเปรียบเทียบระหว่างตำแหน่งจริงและตำแหน่งที่ปรับปรุงการวัดได้และค่าความ<br>คลาดเคลื่อน (Linear Regression)..... 70           |
| 4.6      | การเปรียบเทียบระหว่างตำแหน่งจริงและตำแหน่งที่ปรับปรุงการวัดได้และค่าความ<br>คลาดเคลื่อน (Multiple Linear Regression) ..... 72 |
| 4.7      | เปรียบเทียบของตำแหน่งจริงและตำแหน่งที่วัดได้ของการระบุตำแหน่งของรถอัตโนมัติ .... 74                                           |

## สารบัญรูป

### รูปที่

### หน้า

|      |                                                                                                           |    |
|------|-----------------------------------------------------------------------------------------------------------|----|
| 2.1  | แสดงกลไกที่สำคัญของแบบจำลองการตรวจจับวัตถุ (Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. 2020). ....    | 13 |
| 2.2  | สถาปัตยกรรม Yolo V1 (Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016). ....                     | 14 |
| 2.3  | Darknet-53 (Chen, R. C. 2019). ....                                                                       | 15 |
| 2.4  | Mosaic represents .....                                                                                   | 16 |
| 2.5  | กราฟแสดงความเร็วในการตรวจจับของ YOLOv5 Ultralytics. (2020). ....                                          | 17 |
| 2.6  | แผนภาพแสดงการพัฒนาของ ( YOLO Zhang, R. 2023 ). ....                                                       | 18 |
| 2.7  | ส่วนประกอบหลักของสถาปัตยกรรม (YOLOv8 Chen, H. 2023). ....                                                 | 18 |
| 2.8  | สถาปัตยกรรม Deep SORT (Ikomia 2023). ....                                                                 | 19 |
| 2.9  | A typical ROS network configuration (Quigley, M., Conley, K., Gerkey, B., 2009) ...                       | 24 |
| 2.10 | multi-UV interface with map, camera views, and interaction through selection. (Barber et al., 2011). .... | 24 |
| 2.11 | แพลตฟอร์มที่ช่วยในการจัดการข้อมูลสำหรับการเรียนรู้ของระบบคอมพิวเตอร์วิชั่น .....                          | 26 |
| 3.1  | แผนภาพกระบวนการตรวจจับและคาดการณ์การชนสำหรับหุ่นยนต์ส่งของอัตโนมัติ .....                                 | 27 |
| 3.2  | ภาพแสดงพื้นที่จำลองในการทำงาน .....                                                                       | 29 |
| 3.3  | ภาพแสดงการจำลองสภาพแวดล้อมจำลองของรถตามท้องถนนทั่วไป .....                                                | 30 |
| 3.4  | ภาพแสดงการจำลองสภาพแวดล้อมจำลองของรถอัตโนมัติ .....                                                       | 30 |
| 3.5  | พื้นที่ทดสอบที่มีการทำกริดสำหรับทดสอบการนำทางอัตโนมัติ .....                                              | 31 |
| 3.6  | ภาพการทดสอบเปิดกล้องโดยใช้ Python และ ไลบรารี OpenCV .....                                                | 32 |
| 3.7  | ทดสอบการใช้ HD Maps บนซอฟต์แวร์ rviz .....                                                                | 32 |
| 3.8  | เป็นการแปลงรูปภาพ Bird's eye view .....                                                                   | 33 |
| 3.9  | เป็นการแปลงรูปภาพ Inverse Perspective Mapping .....                                                       | 34 |
| 3.10 | ขั้นตอนการติดป้ายกำกับโดยใช้เครื่องมือ Roboflow .....                                                     | 35 |
| 3.11 | ขั้นตอนการแบ่ง Data ก่อนนำไปสร้างโมเดลการเรียนรู้โดยใช้เครื่องมือ Roboflow .....                          | 36 |
| 3.12 | คำสั่งที่ใช้ในการฝึกโมเดล 200 รอบ (Epochs) และปรับขนาดภาพเป็น 640 พิกเซล .....                            | 37 |

## สารบัญรูป (ต่อ)

| รูปที่ | หน้า                                                                                                                     |
|--------|--------------------------------------------------------------------------------------------------------------------------|
| 3.13   | โครงสร้าง CNN ของ โมเดล Yolo V8 ..... 37                                                                                 |
| 3.14   | วิธีหาจุดกึ่งกลางของ Bounding Box..... 38                                                                                |
| 3.15   | วิธีหาจุดกึ่งกลางของ Bounding Box..... 40                                                                                |
| 3.16   | การตรวจจับวัตถุในกริด [1,1] ..... 40                                                                                     |
| 3.17   | การตรวจจับวัตถุในกริด [0,3] ..... 41                                                                                     |
| 3.18   | การตรวจจับวัตถุในกริด [3,0] ..... 41                                                                                     |
| 3.19   | ภาพการระบุตำแหน่งรถอัตโนมัติด้วย Lidar Velodyne VLP-16 และ IMU โดยใช้ Rviz . 42                                          |
| 3.20   | โครงสร้างเครือข่ายประสาทเทียม (ANN) สำหรับการทำนายตำแหน่ง ..... 44                                                       |
| 3.21   | สูตรการคำนวณระยะทางระหว่างสองจุดในระบบพิกัดคาร์ทีเซียน (Cartesian Coordinates) ..... 45                                  |
| 3.22   | การสื่อสารระหว่างระบบ MQTTและROS เพื่อการชะลอและหยุดหุ่นยนต์เมื่อมีความเสี่ยง..... 46                                    |
| 3.23   | แสดงกระบวนการทำงานของระบบการสื่อสารตำแหน่งและสัญญาณการหยุดหุ่นยนต์ ..... 48                                              |
| 4.1    | กราฟจากผลลัพธ์การฝึกการทดสอบโมเดล Yolo..... 54                                                                           |
| 4.2    | Confusion Matrix Unnormalized แสดงผลการทดสอบโมเดล..... 56                                                                |
| 4.3    | Confusion Matrix Normalized แสดงผลการทดสอบโมเดล..... 56                                                                  |
| 4.4    | กราฟ F1-Confidence Curve ของโมเดลการตรวจจับวัตถุ..... 58                                                                 |
| 4.5    | กราฟ Precision-Confidence Curve ของโมเดลการตรวจจับวัตถุ..... 59                                                          |
| 4.6    | แสดงผลการตรวจจับวัตถุในเฟรมของการทดสอบชุดที่ 1 ..... 60                                                                  |
| 4.7    | แสดงผลการตรวจจับวัตถุในเฟรมของการทดสอบชุดที่ 2 ..... 60                                                                  |
| 4.8    | กราฟเปรียบเทียบตำแหน่งจริงและตำแหน่งที่วัดได้จากการตรวจจับวัตถุ ..... 66                                                 |
| 4.9    | กราฟความสัมพันธ์ระหว่าง Real Position X และ Measured Position X..... 68                                                  |
| 4.10   | ความสัมพันธ์ระหว่าง Real Position Y และ Measured Position Y ..... 69                                                     |
| 4.11   | กราฟเปรียบเทียบตำแหน่งจริงและตำแหน่งที่ถูกปรับปรุงจากค่าที่วัดได้จากการตรวจจับวัตถุ (Linear Regression) ..... 71         |
| 4.12   | กราฟเปรียบเทียบตำแหน่งจริงและตำแหน่งที่ถูกปรับปรุงจากค่าที่วัดได้จากการตรวจจับวัตถุ (Multiple Linear Regression)..... 74 |

## สารบัญรูป (ต่อ)

| รูปที่ | หน้า                                                  |
|--------|-------------------------------------------------------|
| 4.13   | กราฟผลลัพธ์การฝึกโมเดลการทำนายตำแหน่งจากกล้อง..... 77 |
| 4.14   | กราฟผลลัพธ์การฝึกโมเดลการทำนายตำแหน่งจากกล้อง..... 78 |



# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของปัญหาการวิจัย

ในปัจจุบัน เทคโนโลยียานยนต์อัตโนมัติ (Autonomous Vehicles) กำลังกลายเป็นหนึ่งในนวัตกรรมที่ได้รับความสนใจและความสำคัญอย่างมากทั่วโลก เนื่องจากความสามารถในการขับขี่โดยไม่ต้องอาศัยมนุษย์ ซึ่งสามารถช่วยลดความเสี่ยงในการเกิดอุบัติเหตุบนท้องถนน ลดปัญหาการจราจรติดขัด และยังช่วยในการลดการปล่อยก๊าซเรือนกระจกที่เกิดจากการใช้น้ำมันเชื้อเพลิงของรถยนต์ที่มีคนขับการนำเทคโนโลยียานยนต์อัตโนมัติมาใช้อย่างแพร่หลายยังต้องเผชิญกับปัญหาและความท้าทายหลายประการ โดยเฉพาะในด้านการตรวจจับและประเมินสถานการณ์ในสภาพแวดล้อมที่ซับซ้อน

หนึ่งในปัญหาสำคัญของระบบยานยนต์อัตโนมัติ คือ ความสามารถในการตรวจจับและติดตามวัตถุที่อยู่รอบข้างรถอย่างแม่นยำ โดยเฉพาะในสถานการณ์ที่มีการเคลื่อนไหวของวัตถุหลายชิ้นพร้อมกัน หรือในสภาพแวดล้อมที่มีแสงน้อย ซึ่งความผิดพลาดในการตรวจจับหรือประเมินตำแหน่งวัตถุอาจส่งผลให้เกิดความเสี่ยงในการเกิดอุบัติเหตุและการตัดสินใจที่ผิดพลาดของระบบควบคุม ดังนั้น การแก้ปัญหาจุดอ่อนเหล่านี้หรือการที่เซ็นเซอร์ไม่สามารถตรวจจับวัตถุในบางพื้นที่ได้กลายเป็นหนึ่งในความท้าทายที่สำคัญ

การวิจัยนี้จึงมุ่งเน้นไปที่การพัฒนาและปรับปรุงระบบตรวจจับและติดตามวัตถุที่สามารถทำนายตำแหน่งได้อย่างแม่นยำและมีประสิทธิภาพสูงขึ้น โดยเฉพาะการผสมผสานการใช้งานระหว่างเทคโนโลยี LiDAR (Light Detection and Ranging) และกล้อง เพื่อเพิ่มความสามารถในการตรวจจับวัตถุในสภาพแวดล้อมที่หลากหลาย ซึ่งจะเป็นการแก้ปัญหาที่เกิดจากข้อจำกัดของเซ็นเซอร์เดิม จึงได้พัฒนาโมเดลปัญญาประดิษฐ์ประเภท Artificial Neural Networks (ANN) เพื่อช่วยในการทำนายตำแหน่งของวัตถุให้มีความแม่นยำยิ่งขึ้น ระบบดังกล่าวจะช่วยให้ยานยนต์อัตโนมัติสามารถตัดสินใจได้อย่างรวดเร็วและแม่นยำในสถานการณ์ที่ซับซ้อน

ที่มาของปัญหาการวิจัยนี้เกิดจากความต้องการในการพัฒนาระบบยานยนต์อัตโนมัติที่มีความปลอดภัยสูงสุด ซึ่งการที่ระบบสามารถตรวจจับและประเมินตำแหน่งของวัตถุรอบข้างได้อย่างถูกต้องในระยะเวลาที่รวดเร็ว ถือเป็นหัวใจสำคัญในการป้องกันอุบัติเหตุ และทำให้รถสามารถตอบสนองต่อสถานการณ์ที่เกิดขึ้นได้ทันทั่วทั้งที่ การวิจัยนี้ได้นำข้อมูลจากเซ็นเซอร์หลายประเภทมาผสมผสานกันเพื่อลดข้อผิดพลาดในการตรวจจับและพัฒนาระบบให้สามารถประเมินสภาพแวดล้อมที่ซับซ้อนได้ดีขึ้น ซึ่งจะเป็นก้าวสำคัญในการพัฒนาระบบอัตโนมัติในอนาคต

ความสำคัญของการวิจัยนี้จึงอยู่ที่การพัฒนาระบบตรวจจับและติดตามวัตถุให้มีความแม่นยำและมีประสิทธิภาพสูงที่สุด เพื่อให้ระบบยานยนต์อัตโนมัติสามารถใช้งานในสภาพแวดล้อมจริงได้อย่างปลอดภัย ลดความเสี่ยงจากอุบัติเหตุ และเพิ่มความเชื่อถือให้กับผู้ใช้งาน

## 1.2 วัตถุประสงค์ของการวิจัย

- 1) เพื่อสร้างระบบตรวจจับและติดตามยานพาหนะแบบเรียลไทม์
- 2) เพื่อชิงโครไนซ์ข้อมูลตำแหน่งของยานพาหนะกับระบบนำทางของหุ่นยนต์จัดส่งอัตโนมัติ โดยการแปลงตำแหน่ง Local เป็น Global เพื่อการทำงานภายในกรอบการทำงานของระบบ
- 3) เพื่อใช้โครงข่ายประสาทเทียม (ANN) คาดการณ์ตำแหน่งถัดไปของยานพาหนะ เพื่อเพิ่มความสามารถในการคาดการณ์ตำแหน่ง และ วิถีของยานพาหนะ
- 4) เพื่อเพิ่มความปลอดภัยให้หุ่นยนต์อัตโนมัติระหว่างทำภารกิจในการเดินทางระหว่างมุมอับสายตาเพิ่มเพิ่มความสามารถของระบบอัตโนมัติ

## 1.3 ขอบเขตของการศึกษา

- 1) กล้องจะติดตั้งอยู่ที่จุดสังเกตการณ์คงที่ที่แยกจากรถ โดยมีมุมมอง 120 องศา ความละเอียดของภาพ 1080p และอัตราการถ่ายภาพ 30 fps
- 2) กล้องจะใช้สำหรับการตรวจจับและติดตามยานพาหนะที่เคลื่อนที่เข้าสู่โซนที่หุ่นยนต์มองไม่เห็น
- 3) ยานพาหนะในสภาพแวดล้อมที่กำหนดเคลื่อนที่ด้วยความเร็วคงที่
- 4) ผลการวิจัยจะประเมินประสิทธิภาพของกล้องในการตรวจจับและติดตามยานพาหนะที่เคลื่อนที่เข้าสู่โซนที่หุ่นยนต์มองไม่เห็นภายใต้สภาพแวดล้อมจำลอง
- 5) การศึกษานี้จะจำกัดขอบเขตการทดลองภายในห้อง F11 ของศูนย์ถ่ายทอดเทคโนโลยีอากาศยานไร้คนขับ โดยพื้นที่ภายในห้องทดลองได้รับการแบ่งสัดส่วนเพื่อจำลองสภาพแวดล้อมที่ควบคุมได้สำหรับการทดสอบการนำทางอัตโนมัติของหุ่นยนต์ (Robot Rover)



#### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เพิ่มความปลอดภัยบนท้องถนน ระบบตรวจจับและติดตามวัตถุที่พัฒนาขึ้นจากงานวิจัยนี้จะช่วยเพิ่มความแม่นยำในการตรวจจับวัตถุรอบข้างและทำนายตำแหน่ง ซึ่งจะช่วยลดความเสี่ยงในการเกิดอุบัติเหตุจากการที่ยานยนต์อัตโนมัติไม่สามารถตรวจจับวัตถุในจุดอับสายตาได้อย่างครบถ้วน
- 2) เสริมสร้างความเชื่อมั่นและการยอมรับต่อยานยนต์อัตโนมัติ การพัฒนาเทคโนโลยีที่ช่วยให้ยานยนต์อัตโนมัติมีความสามารถในการรับรู้สภาพแวดล้อมและตอบสนองต่อสถานการณ์ได้อย่างแม่นยำ จะช่วยเพิ่มความมั่นใจให้แก่ผู้ใช้งานและสาธารณชนในการใช้งานยานยนต์อัตโนมัติ
- 3) ปรับปรุงประสิทธิภาพการทำงานของระบบยานยนต์อัตโนมัติ การผสมผสานเทคโนโลยี LiDAR และกล้อง รวมถึงการพัฒนาโมเดลปัญญาประดิษฐ์ (ANN) เพื่อทำนายตำแหน่ง จะทำให้ระบบยานยนต์อัตโนมัติสามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น ลดข้อผิดพลาด และช่วยให้ยานยนต์ตอบสนองต่อสถานการณ์ที่ซับซ้อนได้อย่างทันท่วงที
- 4) ลดอุบัติเหตุและเพิ่มความปลอดภัยของการขับขี่ ด้วยการพัฒนา ระบบที่สามารถตรวจจับและทำนายการเคลื่อนไหวของวัตถุรอบข้างได้อย่างแม่นยำ การตอบสนองต่อเหตุการณ์ที่อาจก่อให้เกิดอุบัติเหตุจะเกิดขึ้นอย่างรวดเร็วและแม่นยำมากขึ้น ซึ่งจะช่วยลดอุบัติเหตุบนท้องถนนได้อย่างมีนัยสำคัญ
- 5) ส่งเสริมการวิจัยและพัฒนาเทคโนโลยี การพัฒนาระบบตรวจจับและติดตามวัตถุสำหรับยานยนต์อัตโนมัติในครั้งนี้จะเป็นการสนับสนุนการวิจัยในด้านเทคโนโลยีการขับขี่อัตโนมัติ ซึ่งจะสามารถนำไปต่อยอดในอนาคตและผลักดันการพัฒนาเทคโนโลยียานยนต์อัตโนมัติให้มีความก้าวหน้ามากยิ่งขึ้น

## บทที่ 2

### ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้เป็นการศึกษาและพัฒนากระบวนการทำนายและป้องกันการชนกันของหุ่นยนต์ส่งของอัตโนมัติกับรถที่ไม่มีระบบช่วยเหลือหรือเซนเซอร์ใด ๆ ที่บริเวณทางแยกโดยใช้กล้องวงจรปิดเพื่อจับภาพและเปลี่ยนมุมมองไปยังมุมมองแบบ Bird Eye View เพื่อสร้างกริดอ้างอิงตำแหน่งจากนั้นใช้ YOLO V8 ตรวจจับตำแหน่งรถยนต์บนกริดนั้นและส่งข้อมูลไปยังระบบ DeepSort เพื่อติดตามการเคลื่อนไหว ระบบจะแปลงตำแหน่ง X,Y จาก bounding box ให้อยู่ในระบบพิกัดเดียวกับหุ่นยนต์ส่งของอัตโนมัติผ่านการเปลี่ยนแปลงเชิงโฮโมจีเนียส และใช้เครือข่ายประสาทเทียม (ANN) ในการทำนายตำแหน่งถัดไปของรถที่ตรวจจับโดยกล้องวงจรปิดและหุ่นยนต์ส่งของอัตโนมัติเมื่อระบบทำนายตำแหน่งถัดไปและเปรียบเทียบระหว่างรถยนต์กับหุ่นยนต์ หากพบว่าเป็นไปได้ที่จะเกิดการชนกัน ระบบการชะลอความเร็วเพื่อทำการหลีกเลี่ยงการชนนั้น ผู้วิจัยได้ทำการศึกษาค้นคว้าวรรณกรรมและงานวิจัยต่าง ๆ ที่เกี่ยวข้องเพื่อใช้ในการวิจัยและพัฒนาโดยในหัวข้อต่อไปนี้

#### 2.1 การแปลงมุมมองภาพ

การเปลี่ยนมุมมองภาพ หรือ Perspective Transformation, เป็นกระบวนการที่ซับซ้อนและมีความสำคัญในการเปลี่ยนแปลงและสร้างมุมมองใหม่ของภาพที่ถูกจับจากมุมมองปกติเป็นมุมมองแบบภาพมุมสูง (Bird's eye view) (Li, H., Sima, C., Dai, J., Wang, W., Lu, L., Wang, H., ... & Qiao, Y. 2023). กระบวนการนี้ไม่เพียงแต่เปลี่ยนแปลงตำแหน่งของพิกเซลภายในภาพ แต่ยังรวมถึงการปรับมุมมองและขนาดของสิ่งที่ปรากฏในภาพเพื่อสร้างภาพที่มีมิติและลึกซึ้ง การประยุกต์ใช้ของกระบวนการนี้หลากหลาย ตั้งแต่การวางแผนเมือง การวิเคราะห์พื้นที่ การช่วยเหลือการขับขี่ในยานยนต์ ไปจนถึงการใช้งานในด้านกีฬาและบันเทิงในการทำให้ภาพมีมุมมองแบบภาพมุมสูงจำเป็นต้องมีการคำนวณทางคณิตศาสตร์อย่างละเอียด โดยใช้เทคนิคต่าง ๆ เช่น homography และการแปลงเมทริกซ์ ซึ่งช่วยในการปรับแต่งมุมมองและขนาดของภาพให้เป็นไปตามต้องการ กระบวนการนี้ยังต้องการความเข้าใจในหลักการของแสงและเงา เพื่อสร้างภาพที่สมจริงและเปี่ยมไปด้วยมิติผลลัพธ์ที่ได้จากการเปลี่ยนมุมมองแบบภาพมุมสูงนั้นไม่เพียงแต่เป็นการแสดงภาพที่กว้างขวางและลึกซึ้งเท่านั้น แต่ยังเปิดโอกาสให้เห็นโลกในมุมที่แตกต่างจากมุมมองปกติ นำเสนอมุมมองที่ไม่ธรรมดาและมีความโดดเด่น ซึ่งสามารถนำไปใช้

ในหลากหลายด้าน ทั้งในงานวิจัย การพัฒนาเทคโนโลยี และการสร้างสรรค์ผลงานทางศิลปะและการออกแบบ ในทางปฏิบัติ การเปลี่ยนมุมมองภาพเป็นเครื่องมือที่มีค่าในการมองเห็นและเข้าใจโลกในมิติที่กว้างขึ้น และช่วยเพิ่มความสามารถในการวิเคราะห์และตีความข้อมูลที่มีอยู่ในภาพ โดยเฉพาะอย่างยิ่งในการวิเคราะห์ภาพที่ต้องการความแม่นยำสูงและมุมมองที่ครอบคลุม

### 2.1.1 Homography

Homography (Hartley, R., & Zisserman, A. 2003) คือ กระบวนการทางคณิตศาสตร์ที่ใช้ในการแปลงภาพสองมิติ ซึ่งช่วยให้สามารถเปลี่ยนภาพจากมุมมองหนึ่งไปยังมุมมองอื่นโดยไม่เปลี่ยนแปลงความสัมพันธ์เชิงเรขาคณิตระหว่างจุดในภาพ ในทางปฏิบัติ Homography ใช้เมทริกซ์  $3 \times 3$  ในการแปลงพิกัดของจุดในภาพหนึ่งไปยังพิกัดที่สอดคล้องในภาพอื่นสมการพื้นฐานของ Homography สามารถแสดงได้ดังนี้

กำหนดให้  $(x, y)$  แทนพิกัดของจุดในภาพต้นฉบับและ  $(x', y')$  แทนพิกัดของจุดเดียวกันในภาพที่ปรับเปลี่ยนแล้ว สมการ Homography สามารถเขียนได้เป็น

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

โดยที่  $H$  คือ เมทริกซ์ Homography ขนาด  $3 \times 3$  ซึ่งประกอบไปด้วยสมาชิกและสามารถเขียนได้เป็น

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

เมทริกซ์  $H$  นี้ใช้ในการแปลงพิกัดจุดในภาพอย่างเป็นระบบ ทำให้สามารถสร้างมุมมองใหม่ที่เหมือนกับการมองจากด้านบน การคำนวณเมทริกซ์  $h_{ij}$  นั้น จำเป็นต้องมีการทำการวัดและประเมินค่าจากภาพหรือข้อมูลที่มีอยู่เพื่อที่จะสร้างการแปลงที่เหมาะสมที่สุด

## 2.2 การแปลงระบบพิกัด

การแปลง homogeneous transformation matrix (Shirley, P., Ashikhmin, M., & Marschner, S. 2009) ในทางหุ่นยนต์และกราฟิกคอมพิวเตอร์เป็นเทคนิคที่ใช้ในการแปลงตำแหน่งและการวางตัวของวัตถุจากระบบพิกัดหนึ่งไปยังอีกระบบพิกัดหนึ่ง โดยทั่วไปจะใช้ในการแปลง

ตำแหน่งจากระบบพิกัดท้องถิ่น (local coordinate system) เป็นระบบพิกัดทั่วโลก (global coordinate system) หรือในทางกลับกัน

การเคลื่อนที่ (Translation) ในการแปลงโฮโมจีเนียสไม่ได้จำกัดอยู่แค่การเพิ่มค่าเวกเตอร์โดยตรงเข้ากับพิกัดเท่านั้น แต่ยังช่วยให้สามารถผสมผสานการเคลื่อนที่เข้ากับการหมุนและการสเกลได้ในเมทริกซ์เดียวกันทำให้การคำนวณการเปลี่ยนแปลงตำแหน่งเป็นไปอย่างราบรื่นและต่อเนื่อง นี่คือการเคลื่อนที่

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

การเคลื่อนที่เป็นกระบวนการย้ายวัตถุจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งในปริภูมิสามมิติโดยไม่เปลี่ยนแปลงในรูปร่างหรือขนาดการเคลื่อนที่ใช้เวกเตอร์ของการเคลื่อนที่  $d_x, d_y$  เพื่อเพิ่มค่าเหล่านี้ไปยังพิกัด  $x, y$  ปัจจุบันของวัตถุ

การหมุน (Rotation) เป็นหัวใจของการเปลี่ยนแปลงทิศทาง โดยเฉพาะในการจำลองหรือการเรนเดอร์ภาพสามมิติ การหมุนสามารถจำลองการเคลื่อนไหวของวัตถุได้เหมือนจริง เช่น การหมุนของเครื่องจักรหรือการเปลี่ยนทิศทางของรถยนต์นี่คือการหมุน

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

การหมุนเป็นกระบวนการเปลี่ยนทิศทางของวัตถุโดยรักษตำแหน่งศูนย์กลางของมันเอาไว้ การหมุนใช้มุมหมุน  $\theta$  และสามารถหมุนรอบแกนใดแกนหนึ่งหรือหลายแกนในการหมุนบนแกน  $x$  และ  $y$  จะใช้ค่า  $\cos \theta$  และ  $\sin \theta$  ในการคำนวณผลของการหมุน

การสเกล (Scaling) เป็นการเปลี่ยนแปลงที่ใช้ในการปรับขนาดของวัตถุ โดยการคูณพิกัดด้วยตัวปรับสเกลทำให้วัตถุขยายหรือย่อส่วนได้โดยไม่เปลี่ยนรูปร่างของมัน การสเกลมีประโยชน์ในการจำลองการเติบโตหรือการหดตัวของวัตถุในการจำลองกราฟิกหรือในการวิเคราะห์ข้อมูลจากภาพที่ได้จากอุปกรณ์ต่าง ๆ เช่น กล้องหรือเครื่องสแกน นี่คือการสเกล

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

การสเกล คือ การเปลี่ยนขนาดของวัตถุ การเปลี่ยนขนาดในแนวแกน  $x$  และ  $y$  ทำได้โดยการคูณพิกัดด้วย  $s_x$  และ  $s_y$  ซึ่งเป็นตัวปรับสเกลที่กำหนด

## 2.3 การเรียนรู้ของเครื่องจักร

Machine Learning (การเรียนรู้ของเครื่องจักร) คือ สาขาวิทยาการที่อยู่ภายใต้ปัญญาประดิษฐ์ (Artificial Intelligence) โดยมีเป้าหมายในการพัฒนาเทคนิคที่ทำให้คอมพิวเตอร์สามารถเรียนรู้และทำการตัดสินใจหรือทำนายผลลัพธ์ด้วยตัวเองจากข้อมูลที่ได้รับ โดยไม่ต้องการการแทรกแซงหรือการเขียนโปรแกรมคำสั่งอย่างเฉพาะเจาะจงจากมนุษย์ในการเรียนรู้ของเครื่องจักร ระบบจะวิเคราะห์ข้อมูลจำนวนมาก เพื่อหาความสัมพันธ์ รูปแบบ หรือแนวโน้มที่ซ่อนอยู่ และใช้ข้อมูลเหล่านี้เพื่อสร้างโมเดลที่สามารถทำนายหรือตัดสินใจในสถานการณ์ใหม่ ๆ การเรียนรู้นี้อาจจะดำเนินการผ่านหลายวิธี ซึ่งหนึ่งในวิธีที่พบบ่อยคือการเรียนรู้แบบมีผู้ดูแล (Supervised Learning) ซึ่งระบบจะถูกฝึกฝนด้วยข้อมูลที่มีคำตอบหรือป้ายกำกับอยู่แล้ว เพื่อให้ระบบเรียนรู้และทำนายคำตอบสำหรับข้อมูลใหม่ที่ไม่มีป้ายกำกับและยังมีการเรียนรู้แบบไม่มีผู้ดูแล (Unsupervised Learning) ซึ่งเป็นการเรียนรู้จากข้อมูลที่ไม่มีป้ายกำกับ โดยที่ระบบจะพยายามหาความสัมพันธ์หรือกลุ่มของข้อมูลที่มีลักษณะคล้ายกัน และการเรียนรู้แบบเสริมพลัง (Reinforcement Learning) ซึ่งเป็นการเรียนรู้จากการลองผิดลองถูก โดยระบบจะได้รับรางวัลหรือการลงโทษจากการกระทำต่าง ๆ ที่ทำในสภาพแวดล้อมเสมือนจริง Machine Learning จึงเป็นเครื่องมือที่มีความสำคัญมากในยุคปัจจุบัน เนื่องจากช่วยให้คอมพิวเตอร์สามารถแก้ไขปัญหาที่ซับซ้อนและทำนายผลลัพธ์ได้ด้วยตัวเองโดยไม่ต้องโปรแกรมคำสั่งอย่างละเอียด

### 2.3.1 ANN (Artificial Neural Networks)

Artificial Neural Networks (ANN) เป็นหนึ่งในเทคโนโลยีหลักที่ขับเคลื่อนโลกของปัญญาประดิษฐ์และการเรียนรู้ของเครื่องจักร มีการออกแบบมาเพื่อเลียนแบบการทำงานของเครือข่ายประสาทในสมองมนุษย์ ซึ่งทำให้คอมพิวเตอร์สามารถประมวลผลข้อมูลและทำการตัดสินใจหรือทำนายผลลัพธ์ได้อย่างฉลาด โครงสร้างพื้นฐานของ ANN ประกอบด้วยชั้นของนิวรอนเทียมที่เชื่อมต่อกัน ซึ่งแต่ละนิวรอนสามารถรับและประมวลผลข้อมูลที่ป้อนเข้ามา และจากนั้นส่งผลลัพธ์ไปยังนิวรอนในชั้นถัดไป โครงสร้างนี้ช่วยให้ ANN สามารถรับรู้และแปลความหมายจากข้อมูลที่ซับซ้อนได้ในแต่ละนิวรอนเทียม การประมวลผลนี้จะเกิดขึ้นโดยใช้ฟังก์ชันการกระตุ้น ซึ่งควบคุมว่านิวรอนนั้นควรส่งสัญญาณต่อหรือไม่ ฟังก์ชันเหล่านี้สามารถเป็นแบบเชิงเส้นหรือไม่เชิงเส้น เช่น Sigmoid,

ReLU หรือ Tanh ซึ่งช่วยให้เครือข่ายสามารถรับมือกับปัญหาที่ซับซ้อนได้การเรียนรู้ใน ANN เกิดขึ้นผ่านการปรับน้ำหนักและเอนเอียงของแต่ละนิวรอน กระบวนการนี้เรียกว่า "การส่งย้อนกลับ" (Backpropagation) ซึ่งเป็นการปรับปรุงโมเดลให้สามารถทำนายหรือตัดสินใจได้อย่างแม่นยำมากขึ้น โดยการปรับน้ำหนักเหล่านี้จะเกิดขึ้นเพื่อลดความแตกต่างระหว่างผลลัพธ์ที่ทำนายโดยเครือข่ายกับค่าจริงที่รู้อยู่แล้ว ANN ถูกใช้งานในหลายแอปพลิเคชัน ตั้งแต่การจำแนกประเภทข้อมูล การทำนายข้อมูลทางการเงิน การประมวลผลภาษาธรรมชาติ ไปจนถึงการมองเห็นของคอมพิวเตอร์ เช่น การจดจำวัตถุในภาพหรือวิดีโอ ความสามารถของ ANN ในการเรียนรู้และปรับตัวตามข้อมูลที่ซับซ้อนทำให้มันเป็นเครื่องมือที่ทรงพลังและหลากหลายในการจัดการกับปัญหาที่ต้องการความเข้าใจและการวิเคราะห์ข้อมูลในระดับลึกการวัดความแม่นยำของโมเดลในการเรียนรู้ของเครื่องจักรและการวิเคราะห์ข้อมูลเป็นส่วนสำคัญที่ช่วยให้เราประเมินและเข้าใจคุณภาพของโมเดลที่สร้างขึ้น มีหลายวิธีในการวัดความแม่นยำและประสิทธิภาพของโมเดล ซึ่งรวมถึง R-squared, Mean Squared Error (MSE), และ Mean Absolute Error (MAE) แต่ละวิธีนี้มีลักษณะเฉพาะและสำคัญในการทำความเข้าใจและปรับปรุงโมเดล  $R^2$  (R-squared or Coefficient of Determination)

Mean Squared Error (MSE) เป็นวิธีการวัดความแตกต่างระหว่างค่าที่ประมาณโดยโมเดลกับค่าจริง โดยการคำนวณค่าเฉลี่ยของผลต่างที่ยกกำลังสองค่า MSE ที่ต่ำหมายถึงโมเดลที่มีความแม่นยำสูง เนื่องจากผลต่างระหว่างค่าที่โมเดลทำนายกับค่าจริงมีน้อย MSE คำนวณจากสมการ

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$y_i$  คือ ค่าจริง

$\hat{y}_i$  คือ ค่าทำนายโดยโมเดล

$n$  คือ จำนวนของข้อมูล

MSE นับเป็นค่าเฉลี่ยของผลต่างที่ยกกำลังสองระหว่างค่าที่คาดการณ์และค่าจริง

Mean Absolute Error (MAE) คือ ค่าเฉลี่ยของค่าสัมบูรณ์ของผลต่างระหว่างค่าที่คาดการณ์โดยโมเดลกับค่าจริงคล้ายกับ MSE, ค่า MAE ที่ต่ำแสดงถึงความแม่นยำที่สูงของโมเดล MAE มีความไวต่อค่าผิดปกติที่ใหญ่กว่า (outliers) MAE คำนวณจากสมการ

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$



|             |                      |
|-------------|----------------------|
| $y_i$       | คือ ค่าจริง          |
| $\hat{y}_i$ | คือ ค่าทำนายโดยโมเดล |
| $n$         | คือ จำนวนของข้อมูล   |

MAE นับเป็นค่าเฉลี่ยของค่าสัมบูรณ์ของผลต่างระหว่างค่าทำนายและค่าจริง

R-squared ( $R^2$ ) เป็นวิธีการวัดที่ใช้ในการถดถอยสถิติ (statistical regression) เพื่อกำหนดคุณภาพของการประมาณการและแสดงถึงสัดส่วนของความแปรปรวนในตัวแปรตามที่ถูกอธิบายโดยตัวแปรอิสระในโมเดลค่า  $R^2$  มีช่วงตั้งแต่ 0 ถึง 1 โดยค่าที่ใกล้เคียง 1 หมายความว่าโมเดลสามารถอธิบายความแปรปรวนของข้อมูลได้ดี ในขณะที่ค่าที่ใกล้เคียง 0 แสดงว่าโมเดลไม่สามารถอธิบายความแปรปรวนของข้อมูลได้ R-squared คำนวณจากสมการ

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

|             |                        |
|-------------|------------------------|
| $y_i$       | คือ ค่าจริง            |
| $\hat{y}_i$ | คือ ค่าทำนายโดยโมเดล   |
| $\bar{y}$   | คือ ค่าเฉลี่ยของ $y_i$ |
| $n$         | คือ จำนวนของข้อมูล     |

ค่า R-squared นี้แสดงถึงสัดส่วนของความแปรปรวนในตัวแปรตามที่ถูกอธิบายโดยโมเดลค่าที่ใกล้ 1 แสดงถึงโมเดลที่มีคุณภาพสูงในการอธิบายข้อมูล

การเลือกใช้วิธีการวัดความแม่นยำขึ้นอยู่กับบริบทและประเภทของข้อมูลที่เรามี แต่ละวิธีนี้ให้มุมมองที่แตกต่างกันในการประเมินและเข้าใจคุณภาพของโมเดล โดย  $R^2$  มักใช้เพื่อประเมินความสามารถของโมเดลในการอธิบายข้อมูล ในขณะที่ MSE และ MAE เน้นการวัดความแม่นยำของการทำนายผลลัพธ์

### 2.3.2 CNN (Convolutional Neural Networks)

Convolutional Neural Networks (CNN) เป็นหนึ่งในนวัตกรรมปัญญาประดิษฐ์และการเรียนรู้ของเครื่องจักร โดยถูกออกแบบมาเพื่อจัดการกับข้อมูลที่มีโครงสร้างเช่นภาพและวิดีโอ มันมีความสามารถในการรับรู้และจดจำรูปแบบภายในข้อมูลภาพที่ซับซ้อน ทำให้เหมาะสำหรับการประยุกต์ใช้งานในหลากหลายด้าน เช่น การจำแนกประเภทภาพ การตรวจจับวัตถุ และการวิเคราะห์วิดีโอโครงสร้างพื้นฐานของ CNN ใจกลาง คือ Convolutional Layer ที่ทำหน้าที่สกัดคุณลักษณะ



จากข้อมูลภาพ การทำ Convolution คือ การนำคอร์เนลหรือฟิลเตอร์ไปประยุกต์ใช้กับข้อมูลภาพ เพื่อจับจุดเด่นเช่นขอบ, รูปร่าง หรือลักษณะเฉพาะอื่น ๆ ในภาพ การนี้ช่วยให้เครือข่ายสามารถจดจำลักษณะที่สำคัญของวัตถุหรือฉากในภาพได้ตามมาด้วย Pooling Layer ซึ่งทำหน้าที่ลดขนาดของข้อมูลที่ได้จากชั้น Convolutional ลดความซับซ้อนและการคำนวณที่จำเป็น พูลลิ่งมักจะใช้วิธีการ เช่น Max Pooling ที่เลือกค่าสูงสุดในบล็อกของข้อมูลในท้ายสุดของ CNN คือ Fully Connected Layer ที่นำข้อมูลที่ผ่านมาการสกัดแล้วมาเชื่อมต่อกับเครือข่ายประสาทเทียมที่มีการเชื่อมต่ออย่างเต็มที่ ในส่วนนี้เป็นที่ทำการจำแนกประเภทหรือทำนายผลลัพธ์จากข้อมูลที่ได้มาการทำงานและประสิทธิภาพมีความสามารถพิเศษในการเรียนรู้คุณลักษณะและรูปแบบของข้อมูลภาพโดยอัตโนมัติ การทำงานของมันเลียนแบบวิธีที่มนุษย์มองและรับรู้วัตถุ ซึ่งทำให้มันสามารถจดจำและจำแนกประเภทวัตถุในภาพได้อย่างแม่นยำ CNN ยังรักษาความเชื่อมโยงทางพื้นที่ของข้อมูลภาพได้ดี สามารถจำแนกและจดจำวัตถุและรูปแบบในภาพที่มีความซับซ้อนได้อย่างมีประสิทธิภาพ และยังใช้พารามิเตอร์ที่มีประสิทธิภาพ โดยมีการใช้ฟิลเตอร์ที่ใช้ร่วมกันซึ่งช่วยลดจำนวนพารามิเตอร์ที่จำเป็นลงอย่างมากส่วนการประยุกต์ใช้งาน CNN ถูกใช้งานในหลายด้านที่ต้องการการมองเห็นของคอมพิวเตอร์ ตั้งแต่การจำแนกประเภทภาพ, การตรวจจับวัตถุ, การวิเคราะห์วิดีโอ, การจดจำใบหน้า ไปจนถึงการประมวลผลภาษาธรรมชาติ ความสามารถของมันในการจดจำรูปแบบที่ซับซ้อนทำให้ CNN เป็นเครื่องมือที่มีค่าในการแก้ไขปัญหาที่มีความซับซ้อนในโลกแห่งข้อมูลภาพการพัฒนาและการประยุกต์ใช้งานของ CNN จึงเป็นหนึ่งในด้านที่สำคัญที่สุดในการวิจัยและพัฒนาในด้านปัญญาประดิษฐ์ ช่วยให้เราสามารถสร้างระบบที่สามารถจดจำและตีความข้อมูลภาพได้อย่างฉลาดและยืดหยุ่นเพื่อวัดความแม่นยำของโมเดลในการเรียนรู้ของเครื่องจักรและการวิเคราะห์ข้อมูล มีหลายเครื่องมือและเทคนิคที่สามารถใช้ได้ แต่ละเครื่องมือหรือเทคนิคมีลักษณะเฉพาะที่ช่วยให้เราประเมินคุณภาพและประสิทธิภาพของโมเดลในแง่ต่าง ๆ ต่อไปนี้ คือ เครื่องมือหลักที่ใช้ในการวัดความแม่นยำ

Confusion Matrixในงานการจำแนกประเภทแบบสองคลาส (binary classification) Confusion Matrix ประกอบด้วยสี่ส่วนหลัก

True Positives (TP): จำนวนครั้งที่โมเดลทำนายผลบวกและถูกต้อง

True Negatives (TN): จำนวนครั้งที่โมเดลทำนายผลลบและถูกต้อง

False Positives (FP): จำนวนครั้งที่โมเดลทำนายผลบวกแต่ไม่ถูกต้อง (ข้อผิดพลาดประเภท I)

False Negatives (FN): จำนวนครั้งที่โมเดลทำนายผลลบแต่ไม่ถูกต้อง (ข้อผิดพลาดประเภท II)

การคำนวณ Accuracy (ความแม่นยำโดยรวม), Precision (ความแม่นยำในการทำนายผลบวก), Recall (ความสามารถในการจำแนกผลบวกที่แท้จริง), และ F1 Score (ค่าเฉลี่ยแบบ harmonic mean ระหว่าง Precision และ Recall) จาก Confusion Matrix เราสามารถคำนวณค่าต่าง ๆ ต่อไปนี้

Accuracy: คำนวณจากสมการ

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

ค่านี้แสดงถึงความแม่นยำโดยรวมของโมเดลค่า Accuracy ที่สูงแสดงว่าโมเดลมีประสิทธิภาพสูงในการทำนายทั้งผลบวกและผลลบในกรณีที่ข้อมูลมีความไม่สมดุล (Imbalanced Data) เช่น หนึ่งคลาสมีข้อมูลมากกว่าอีกคลาสมาก ค่า Accuracy อาจไม่สามารถสะท้อนประสิทธิภาพของโมเดลได้อย่างแท้จริง

Precision: คำนวณจากสมการ

$$Precision = \frac{TP}{TP + FP}$$

ค่านี้แสดงถึงความแม่นยำในการทำนายผลบวกค่า Precision ที่สูงบ่งบอกว่าโมเดลมีความแม่นยำสูงในการทำนายผลบวก ลดโอกาสของ False Positives สถานการณ์ที่ False Positives นั้นมีความสำคัญ เช่น การทดสอบสำหรับโรคที่ร้ายแรง

Recall: คำนวณจากสมการ

$$Recall = \frac{TP}{TP + FN}$$

ค่านี้แสดงถึงความสามารถของโมเดลในการจำแนกผลบวกที่แท้จริงค่า Recall ที่สูงแสดงว่าโมเดลสามารถจำแนกผลบวกที่แท้จริงได้ดี ลดโอกาสของ False Negatives สถานการณ์ที่ False Negatives นั้นมีความสำคัญ เช่น การตรวจจับภาวะที่เสี่ยงต่อชีวิต

F1 Score: คำนวณจากสมการ

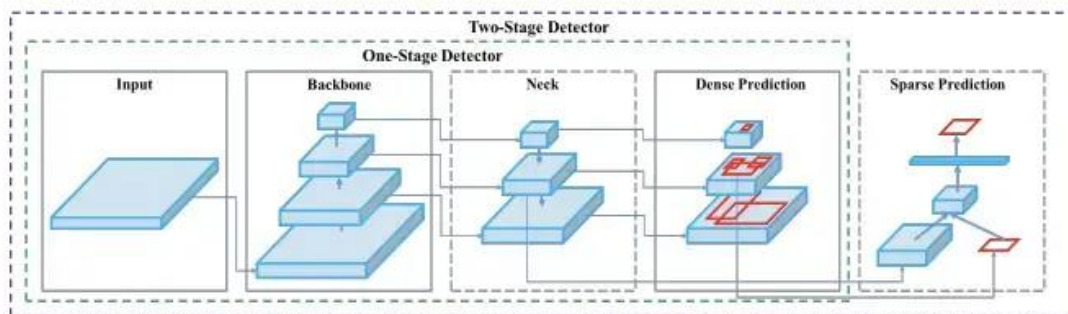
$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

F1 Score เป็นค่าเฉลี่ยแบบ harmonic mean ระหว่าง Precision และ Recall ซึ่งเป็นการวัดความสมดุลระหว่างความแม่นยำและความไวของโมเดล F1 Score ที่สูงบ่งบอกว่าโมเดลมี

ความสมดุลที่ดีระหว่าง Precision และ Recall สถานการณ์ที่ต้องการความสมดุลระหว่างการลด False Positives และ False Negatives

CNN ประกอบด้วยชั้นคอนโวลูชันที่ทำหน้าที่สกัดคุณลักษณะจากข้อมูลภาพ การทำ Convolution คือ การนำเคอร์เนลหรือฟิลเตอร์ไปประยุกต์ใช้กับข้อมูลภาพเพื่อจับจุดเด่นเช่นขอบ, รูปร่าง หรือลักษณะเฉพาะอื่น ๆ ในภาพ การนี้ช่วยให้เครือข่ายสามารถจดจำลักษณะที่สำคัญของวัตถุ หรือฉากในภาพได้ ตามมาด้วย Pooling Layer ซึ่งทำหน้าที่ลดขนาดของข้อมูลที่ได้จากชั้น Convolutional ลดความซับซ้อนและการคำนวณที่จำเป็น พูลลิงมักจะใช้วิธีการเช่น Max Pooling ที่เลือกค่าสูงสุดในบล็อกของข้อมูล ในท้ายสุดของ CNN คือ Fully Connected Layer ที่นำข้อมูลที่ผ่านการสกัดแล้วมาเชื่อมต่อกับเครือข่ายประสาทเทียมที่มีการเชื่อมต่ออย่างเต็มที่ ในส่วนนี้เป็นการทำ การจำแนกประเภทหรือทำนายผลลัพธ์จากข้อมูลที่ได้มา YOLO นำเสนอวิธีการตรวจจับวัตถุที่ เปลี่ยนแปลงแนวทางจากเทคนิคที่ใช้ในอดีต โดยรวมการสกัดคุณลักษณะและการจำแนกประเภทเข้า ด้วยกันในโมเดลเดียว ทำให้ YOLO มีความเร็วสูงและเหมาะสำหรับการใช้งานในระบบเรียลไทม์ การ ทำงานของ YOLO ทำการตรวจจับวัตถุในภาพเพียงครั้งเดียว โดยแบ่งภาพออกเป็นกริดขนาดเล็ก ๆ และทำการทำนายกรอบขอบเขต (bounding boxes) และความน่าจะเป็นของคลาสต่าง ๆ สำหรับ แต่ละเซลล์ในกริด การนี้ช่วยลดเวลาและความซับซ้อนในการประมวลผล YOLO (Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016) นำเสนอการประมวลผลแบบ end-to-end ที่ สามารถทำนายตำแหน่งและประเภทของวัตถุในภาพได้ในครั้งเดียว ทำให้มีประสิทธิภาพสูงในการ ตรวจจับวัตถุหลายอย่างในภาพได้อย่างรวดเร็วและแม่นยำ

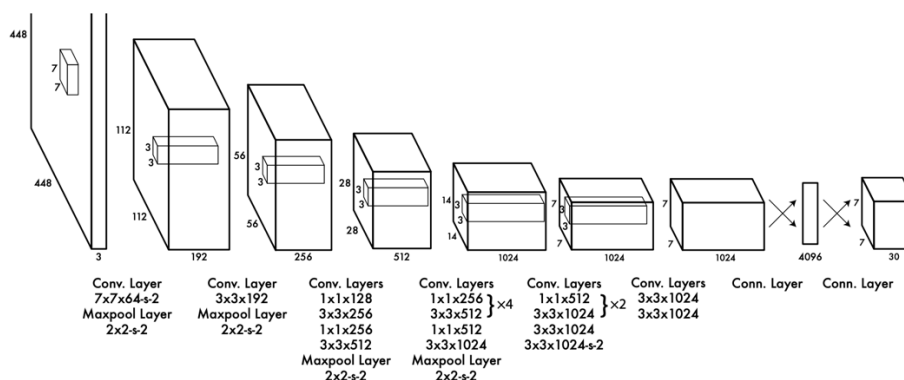
You Only Look Once (YOLO) เป็นอัลกอริธึมการตรวจจับวัตถุที่เปิดตัวในปี 2558 ในรายงานการวิจัยโดย Joseph Redmon, Santosh Divvala, Ross Girshick และ Ali Farhadi สถาปัตยกรรมของ YOLO เป็นการปฏิวัติครั้งสำคัญใน พื้นที่ การตรวจจับวัตถุ แบบ เรียลไทม์ ซึ่งเหนือกว่ารุ่นก่อน นั่นคือ Convolutional Neural Network (R-CNN) YOLO เป็น อัลกอริธึม ช็อตเดียวที่จัดประเภทวัตถุโดยตรงในการส่งผ่านครั้งเดียว โดยมีโครงข่ายประสาทเทียม เพียงเครือข่ายเดียวทำนายกรอบขอบเขตและความน่าจะเป็นของคลาสโดยใช้รูปภาพเต็มเป็นอินพุต โมเดล YOLO มีการพัฒนาอย่างต่อเนื่อง ตั้งแต่นั้นเป็นต้นมา ทีมวิจัยหลายทีมได้เปิดตัว YOLO เวอร์ชันต่าง ๆ โดย YOLOv8 เป็นเวอร์ชันล่าสุด ส่วนต่อไปนี้จะสรุปภาพรวมของเวอร์ชันที่ผ่านมา ทั้งหมดและการปรับปรุง



รูปที่ 2.1 แสดงกลไกที่สำคัญของแบบจำลองการตรวจจับวัตถุ (Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. 2020).

การทำงานพื้นฐานของโมเดลตรวจจับวัตถุประกอบไปด้วยสามส่วนหลัก ๆ คือ ส่วนหลัง (backbone), คอ (neck), และหัว (head) ส่วนหลังเป็นเครือข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network - CNN) ที่ได้รับการฝึกฝนมาแล้วเพื่อสกัดคุณลักษณะต่าง ๆ ที่มีระดับต่ำ, ปานกลาง, และสูงออกจากรูปภาพที่นำเข้ามา ส่วนคอจะทำหน้าที่ผสานคุณลักษณะเหล่านี้ด้วยบล็อกการรวมเส้นทางเช่นเครือข่ายพีระมิดคุณลักษณะ (Feature Pyramid Network - FPN) และส่งต่อไปยังส่วนหัวเพื่อจำแนกประเภทวัตถุและทำนายกรอบขอบเขตของวัตถุ ส่วนหัวอาจประกอบด้วยโมเดลการทำนายแบบหนึ่งขั้นตอนหรือการทำนายแบบหนาแน่น เช่น YOLO หรือ Single-shot Detector (SSD) หรืออาจใช้โมเดลการทำนายแบบสองขั้นตอนหรือการทำนายแบบกระจาย เช่น R-CNN ประวัติโดยย่อของ YOLO

YOLOv1 เป็นเวอร์ชันแรกที่น่าสนใจวิธีการทำนายกรอบขอบเขตและความน่าจะเป็นของประเภทวัตถุในครั้งเดียว โดยการแบ่งภาพเป็นกริดหลายช่องและคำนวณความมั่นใจและกรอบขอบเขตสำหรับแต่ละช่องกริด ทำให้มีความเร็วและความแม่นยำสูงกว่า R-CNN ที่ใช้ก่อนหน้านี้



รูปที่ 2.2 สถาปัตยกรรม Yolo V1 (Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016).

YOLOv2 นำเสนอ 'anchor boxes' ซึ่งเป็นกรอบขอบเขตที่กำหนดไว้ล่วงหน้า ช่วยให้การทำนายตำแหน่งของวัตถุมีความแม่นยำมากขึ้น โดยสามารถทำนายวัตถุได้มากกว่า 9000 ประเภท และมีการปรับปรุงในเรื่องความเร็วและความแม่นยำ (Redmon, J., & Farhadi, A. 2017)

ตารางที่ 2.1 ทดสอบการตรวจจับ Pascal VOC 2007 YOLOv2 เร็วและแม่นยำกว่าการตรวจจับอื่น

| Detection Frameworks                 | Train     | mAP  | FPS |
|--------------------------------------|-----------|------|-----|
| Fast R-CNN ( Girshick, R. B. 2015 )  | 2007+2012 | 70.0 | 0.5 |
| Faster R-CNN VGG-16 ( Sun, J. 2015 ) | 2007+2012 | 73.2 | 7   |
| Faster R-CNN ResNet ( Sun, J. 2016 ) | 2007+2012 | 76.4 | 5   |
| YOLO ( Farhadi, A. 2016)             | 2007+2012 | 63.4 | 45  |
| SSD300 ( Berg, A. C. 2016 )          | 2007+2012 | 74.3 | 46  |
| SSD500 ( Berg, A. C. 2016 )          | 2007+2012 | 76.8 | 19  |
| YOLOv2 288 x 288                     | 2007+2012 | 69.0 | 91  |
| YOLOv2 352 x 352                     | 2007+2012 | 73.7 | 81  |
| YOLOv2 416 x 416                     | 2007+2012 | 76.8 | 67  |
| YOLOv2 480 x 480                     | 2007+2012 | 77.8 | 59  |
| YOLOv2 544 x 544                     | 2007+2012 | 78.6 | 40  |

YOLOv3 เป็นการอัปเดตที่มีความแม่นยำสูงขึ้น โดยใช้ Darknet-53 เป็นส่วนหลัง และใช้ logistic classifiers แทน softmax รวมถึงการใช้ Binary Cross-entropy (BCE) loss ในการทำนายประเภทของวัตถุ ทำให้สามารถจำแนกประเภทวัตถุได้แม่นยำขึ้น Redmon, J., & Farhadi, A. (2018)

|    | Type          | Filters | Size             | Output           |
|----|---------------|---------|------------------|------------------|
|    | Convolutional | 32      | $3 \times 3$     | $256 \times 256$ |
|    | Convolutional | 64      | $3 \times 3 / 2$ | $128 \times 128$ |
| 1x | Convolutional | 32      | $1 \times 1$     |                  |
|    | Convolutional | 64      | $3 \times 3$     |                  |
|    | Residual      |         |                  | $128 \times 128$ |
|    | Convolutional | 128     | $3 \times 3 / 2$ | $64 \times 64$   |
| 2x | Convolutional | 64      | $1 \times 1$     |                  |
|    | Convolutional | 128     | $3 \times 3$     |                  |
|    | Residual      |         |                  | $64 \times 64$   |
|    | Convolutional | 256     | $3 \times 3 / 2$ | $32 \times 32$   |
| 8x | Convolutional | 128     | $1 \times 1$     |                  |
|    | Convolutional | 256     | $3 \times 3$     |                  |
|    | Residual      |         |                  | $32 \times 32$   |
|    | Convolutional | 512     | $3 \times 3 / 2$ | $16 \times 16$   |
| 8x | Convolutional | 256     | $1 \times 1$     |                  |
|    | Convolutional | 512     | $3 \times 3$     |                  |
|    | Residual      |         |                  | $16 \times 16$   |
|    | Convolutional | 1024    | $3 \times 3 / 2$ | $8 \times 8$     |
| 4x | Convolutional | 512     | $1 \times 1$     |                  |
|    | Convolutional | 1024    | $3 \times 3$     |                  |
|    | Residual      |         |                  | $8 \times 8$     |
|    | Avgpool       |         | Global           |                  |
|    | Connected     |         | 1000             |                  |
|    | Softmax       |         |                  |                  |

รูปที่ 2.3 Darknet-53 (Chen, R. C. 2019)

YOLOv4 นำเสนอความคิดใหม่อย่าง 'Bag of Freebies' (BoF) และ 'Bag of Specials' (BoS) ซึ่งเป็นเทคนิคที่ช่วยเพิ่มความแม่นยำโดยไม่เพิ่มต้นทุนในการคำนวณ ใช้เทคนิคการเพิ่มข้อมูลแบบใหม่อย่าง Mosaic ที่รวมภาพการฝึกฝนสีภาพเข้าด้วยกัน ทำให้ได้ข้อมูลเพิ่มเติมสำหรับการฝึกฝน Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020).

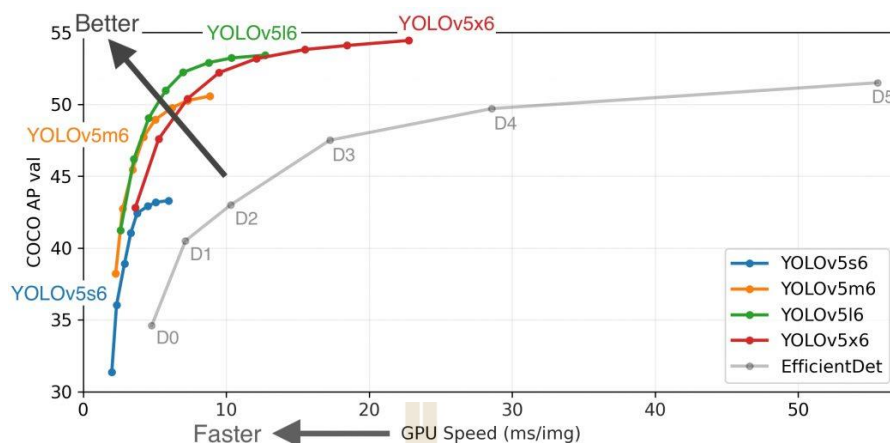




รูปที่ 2.4 Mosaic represents

YOLOv5 ซึ่งถูกพัฒนาโดย Ultralytics เป็นเวอร์ชันที่โดดเด่นในซีรีส์ของ YOLO (You Only Look Once) สำหรับการตรวจจับวัตถุ YOLOv5 ได้รับความนิยมอย่างกว้างขวางและใช้งานอย่างแพร่หลายในชุมชนของการเรียนรู้ด้วยเครื่องจักร หนึ่งในคุณสมบัติหลักของ YOLOv5 คือ การใช้งานที่ง่ายและการปรับแต่งที่ยืดหยุ่นการทำงานของ YOLOv5 มีการออกแบบที่เน้นความเรียบง่าย โดยการใช้ PyTorch, ไลบรารีการเรียนรู้เชิงลึกที่นิยมใช้ ซึ่งทำให้โมเดลนี้เข้าถึงได้ง่ายสำหรับนักพัฒนาและนักวิจัยที่หลากหลาย การเลือกใช้ PyTorch ทำให้โมเดลมีความยืดหยุ่นและง่ายต่อการใช้งาน YOLOv5 ยังใช้ไฟล์ YAML (Yet Another Markup Language) สำหรับการกำหนดค่าแทนการใช้ไฟล์ CFG (configuration file) ที่ใช้ในเวอร์ชันก่อน ๆ YAML ซึ่งเป็นมาตรฐานการบันทึกข้อมูลที่เข้าใจได้ง่ายสำหรับมนุษย์ ทำให้กระบวนการกำหนดค่าของโมเดลนั้นง่ายขึ้นและเข้าถึงได้ง่ายกว่าคุณสมบัติอีกประการหนึ่งของ YOLOv5 คือ ความง่ายในการฝึกฝนและใช้งาน โครงสร้างของโมเดลและกระบวนการฝึกฝนถูกออกแบบมาเพื่อให้เข้าใจง่ายและมีประสิทธิภาพ ช่วยให้ผู้ใช้สามารถฝึกฝนและใช้งานโมเดลการตรวจจับวัตถุได้โดยใช้ความพยายามและทรัพยากรคอมพิวเตอร์ที่น้อยลงเมื่อเทียบกับเวอร์ชันก่อน ๆ ด้วยคุณสมบัติเหล่านี้ ทำให้ YOLOv5 เป็นเครื่องมือในชุดเครื่องมือที่สำคัญสำหรับนักพัฒนาและนักวิจัยในด้านการมองเห็นด้วยคอมพิวเตอร์ แม้จะไม่มี การเผยแพร่ งานวิจัยอย่างเป็นทางการ แต่ YOLOv5 ได้แสดงให้เห็นถึงประสิทธิภาพที่เข้มข้นในงานตรวจจับวัตถุ และได้รับการยอมรับอย่างกว้างขวางในการประยุกต์ใช้งานที่หลากหลาย ตั้งแต่งานอุตสาหกรรมไปจนถึงโครงการวิจัย





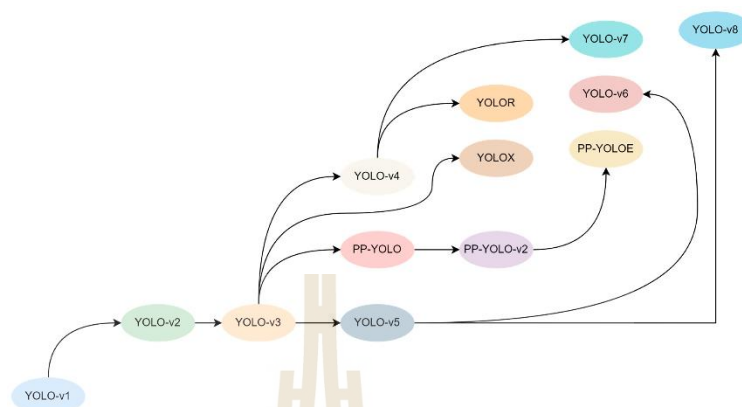
รูปที่ 2.5 กราฟแสดงความเร็วในการตรวจจับของ YOLOv5 Ultralytics. (2020).

YOLOv6 ถูกนำเสนอโดย Meituan จากประเทศจีน, ได้เปิดตัวเป็นหนึ่งในโมเดลการตรวจจับวัตถุที่มีความเร็วและความแม่นยำสูงสำหรับการใช้งานในอุตสาหกรรมการพัฒนามุ่งเน้นไปที่การปรับปรุงหลายด้าน เพื่อเพิ่มประสิทธิภาพของโมเดลและทำให้มันเหมาะกับการใช้งานในสภาพแวดล้อมที่มีความต้องการความเร็วสูงคุณสมบัติหลายประการที่ทำให้มีความเร็วและความแม่นยำที่สูงขึ้น เมื่อเทียบกับเวอร์ชันก่อน ๆ คือ การตรวจจับแบบไม่ใช้ Anchor (Anchor-free): โมเดล YOLOv6 เปลี่ยนไปใช้วิธีการตรวจจับที่ไม่ใช้ anchor ซึ่งช่วยเพิ่มความสามารถในการทั่วไปของโมเดลและลดเวลาในการประมวลผลหลังการตรวจจับ ทำให้ YOLOv6 เร็วขึ้นถึง 51% เมื่อเทียบกับโมเดลที่ใช้วิธี anchor-based ( Wei, X. 2022 ).

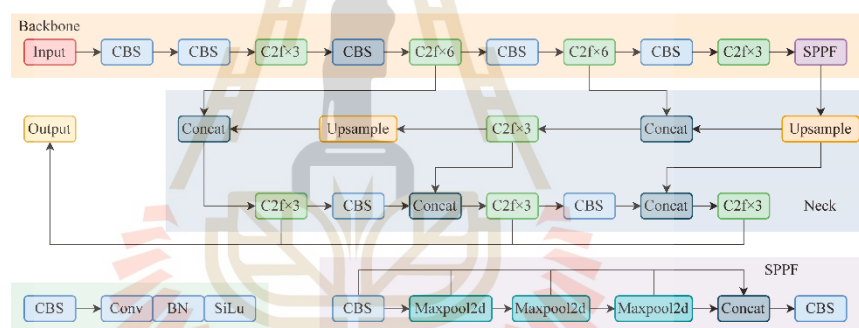
YOLOv7 เป็นโมเดลที่มีความเร็วและความแม่นยำสูงสุดในขณะนั้น ใช้ Extended Efficient Layer Aggregation Network (E-ELAN) เป็นส่วนหลังและใช้ compound scaling ซึ่งช่วยให้การฝึกฝนและทำนายมีประสิทธิภาพสูงขึ้น (Liao, H. Y. M. 2023).

YOLOv8 มีการปรับปรุงให้มีความแม่นยำและความเร็วในการทำนายที่สูงขึ้น โดย YOLOv8 (Ultralytics. 2023) มีการใช้งานที่ง่ายด้วย Python package และ CLI และสามารถทำการตรวจจับวัตถุในภาพหรือวิดีโอได้ efficaciously. ในเวอร์ชันนี้สามารถตรวจจับวัตถุที่มีขนาด

เล็กได้ดี (Liao, H. Y. M. 2023) การตรวจจับวัตถุทางอากาศไร้คนขับ (UAV)นำมาใช้เพื่อเพิ่มประสิทธิภาพในการตรวจจับที่ดีขึ้น



รูปที่ 2.6 แผนภาพแสดงการพัฒนาของ (YOLO Zhang, R. 2023).



รูปที่ 2.7 ส่วนประกอบหลักของสถาปัตยกรรม (YOLOv8 Chen, H. 2023).

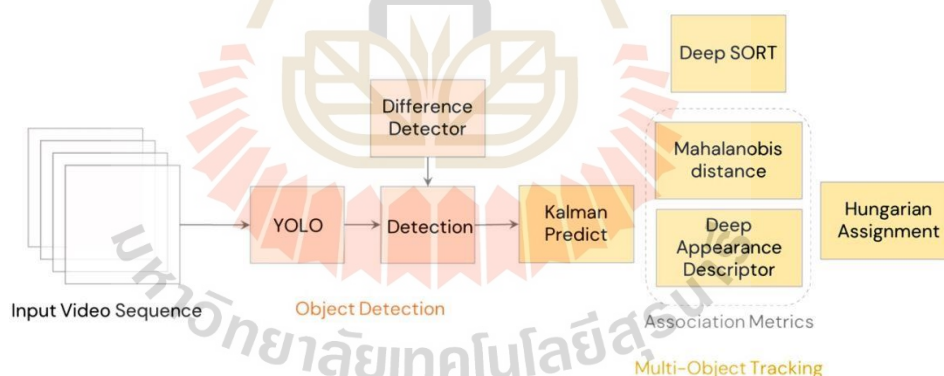
## 2.4 การติดตามวัตถุ

การติดตามวัตถุ (Object Tracking) ในวิดีโอเป็นหนึ่งในงานท้าทายในด้านการมองเห็นของคอมพิวเตอร์ งานนี้เกี่ยวข้องกับการระบุตำแหน่งและการติดตามการเคลื่อนไหวของวัตถุหนึ่งหรือหลายวัตถุในซีควেনซ์วิดีโอต่อเนื่อง วิธีการติดตามวัตถุมีหลากหลายรูปแบบ ตั้งแต่การติดตามวัตถุด้วยเทคนิคทางเรขาคณิตไปจนถึงการใช้แบบจำลองการเรียนรู้เชิงลึกหนึ่งในเทคนิคที่โดดเด่นในการติดตามวัตถุคือ DeepSORT ซึ่งเป็นการพัฒนาต่อจากเทคนิค SORT โดยเพิ่มความสามารถในการจดจำลักษณะเฉพาะของวัตถุโดยใช้ deep learning และนำมาประยุกต์ใช้เพื่อเพิ่มความแม่นยำในการติดตามวัตถุ โดยเฉพาะในสถานการณ์ที่วัตถุมีการเคลื่อนไหวเร็วหรือมีการเปลี่ยนแปลงลักษณะ

การใช้ DeepSORT ในการติดตามวัตถุช่วยเพิ่มความสามารถในการติดตามวัตถุอย่างต่อเนื่องและลดข้อผิดพลาดที่อาจเกิดจากการปกปิดหรือการสับเปลี่ยนระหว่างวัตถุ

#### 2.4.1 Deepsort

Deep SORT (Simple Online and Realtime Tracking) เป็นเทคนิคขั้นสูงในการติดตามวัตถุหลายวัตถุภายในวิดีโอสตรีมมิ่ง (Nunes, U. J. 2022) วิธีนี้พัฒนาต่อยอดมาจาก SORT (Simple Online and Realtime Tracking) โดยใช้แผนการของตัวกรองคาลมานในการติดตามวัตถุ Deep SORT นำเสนอการวิเคราะห์การเชื่อมโยงของวัตถุโดยใช้คุณลักษณะลึกซึ่งได้รับการเรียนรู้จากเครือข่ายประสาทเทียมขั้นสูง ทำให้สามารถจัดการได้กับสถานการณ์ที่วัตถุอาจหายไปชั่วคราวหรือถูกบดบัง Deep SORT ยังรวมถึงการกำหนดรหัสประจำตัว (ID) สำหรับติดตามแต่ละวัตถุในหลายๆ เฟรมภาพ ซึ่งมีความสำคัญต่อการประยุกต์ใช้งานในด้านต่าง ๆ เช่น การเฝ้าระวังด้วยกล้องวงจรปิด ระบบยานพาหนะอัตโนมัติ และการโต้ตอบระหว่างมนุษย์กับคอมพิวเตอร์ อัลกอริธึมนี้ทำงานโดยการประมวลผลในสองขั้นตอน การสร้างการตรวจจับวัตถุก่อน แล้วต่อด้วยการเชื่อมโยงการตรวจจับเหล่านี้กับเส้นทางติดตามที่มีอยู่ โดยรวมแล้ว Deep SORT นำเสนอความก้าวหน้าที่สำคัญในด้านความแม่นยำและความเสถียรของการติดตาม เมื่อเทียบกับเทคนิคการติดตามแบบเดิม ทำให้เป็นเครื่องมือที่มีคุณค่าในด้านการมองเห็นด้วยคอมพิวเตอร์และแอปพลิเคชันด้านปัญญาประดิษฐ์



รูปที่ 2.8 สถาปัตยกรรม Deep SORT (Ikonomia 2023).

Deep SORT ประกอบด้วยองค์ประกอบสำคัญ 4 ประการ ดังนี้

##### 1) การตรวจจับและการแยกคุณสมบัติ

อัลกอริธึม Deep SORT เริ่มต้นด้วยกระบวนการตรวจจับวัตถุ โดยมักใช้เครือข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) เช่น YOLO (You Only Look Once) เพื่อจำแนกวัตถุภายในแต่ละเฟรม การตรวจจับแต่ละครั้งจะเชื่อมโยงกับตัวแทนลักษณะที่มีมิติสูงซึ่งสกัดโดย CNN โดยตัวแทนเหล่านี้จะทำการตรวจจับลักษณะภายนอกของวัตถุที่ถูกตรวจจับเพื่อใช้ในการจับคู่

## 2) ตัวกรองคาลมานในการทำนายสถานะ

ตัวกรองคาลมาน (Kalman Filter) ในระบบ Deep SORT มีบทบาทสำคัญในการทำนายสถานะของวัตถุ โดยทำหน้าที่เป็นอัลกอริธึมหลักในการประมาณค่าสถานะของกระบวนการที่มีความไม่แน่นอน เช่น ตำแหน่ง, ความเร็ว, และความเร่งของวัตถุ ความสามารถในการทำนายตำแหน่งและเส้นทางเคลื่อนที่ของวัตถุในเฟรมต่อไปด้วยความแม่นยำสูงเป็นคุณสมบัติที่โดดเด่นของตัวกรองนี้ ตัวกรองคาลมานใช้ข้อมูลสถานะล่าสุดและการวัดที่ได้จากเซ็นเซอร์หรือตัวตรวจจับเพื่อทำการประมาณสถานะของวัตถุในเฟรมปัจจุบัน โดยคำนึงถึงตำแหน่ง, ความเร็ว, และความเร่ง การประมาณการนี้ไม่เพียงช่วยในการทำนายตำแหน่งของวัตถุ แต่ยังช่วยในการคำนวณความไม่แน่นอนหรือข้อผิดพลาดในการประมาณการด้วย ซึ่งเป็นสิ่งจำเป็นในการคำนวณความเป็นไปได้หรือความน่าเชื่อถือของสถานะที่ประมาณได้หนึ่งในข้อดีของตัวกรองคาลมานคือความสามารถในการปรับปรุงสถานะของวัตถุอย่างต่อเนื่องผ่านแต่ละเฟรม โดยอาศัยข้อมูลปัจจุบันและประวัติการเคลื่อนไหวที่ผ่านมา เมื่อมีข้อมูลใหม่มาจากการตรวจจับในเฟรมต่อไป ตัวกรองคาลมานจะอัปเดตประมาณการสถานะโดยพิจารณาจากข้อมูลใหม่นี้ ทำให้การประมาณการมีความแม่นยำมากขึ้น ตัวกรองคาลมานยังได้รับการออกแบบมาเพื่อจัดการกับความไม่แน่นอนและข้อผิดพลาดที่เกิดจากการวัดหรือสภาวะแวดล้อมที่ไม่สมบูรณ์ ช่วยให้การทำนายสถานะของวัตถุมีความน่าเชื่อถือมากขึ้น การใช้ตัวกรองคาลมานใน Deep SORT จึงทำให้มีความสามารถในการติดตามวัตถุที่มีความแม่นยำสูงและสามารถจัดการกับการเปลี่ยนแปลงของสถานะวัตถุที่รวดเร็วหรือไม่แน่นอนได้ดีขึ้น ซึ่งเป็นสิ่งสำคัญในการประยุกต์ใช้ในการติดตามวัตถุในวิดีโอสตรีมมิ่งแบบเรียลไทม์ การเชื่อมโยงข้อมูลด้วยเมตริกลักษณะที่ปรากฏเชิงลึกและการเชื่อมโยงข้อมูลด้วยเมตริกลักษณะที่ปรากฏเชิงลึกใน Deep SORT เป็นกระบวนการที่เข้าถึงการติดตามวัตถุอย่างซับซ้อนและแม่นยำมากขึ้น โดยไม่ได้พึ่งพาเพียง IoU (Intersection over Union) แต่ใช้การผสมผสานของข้อมูลการเคลื่อนไหวและลักษณะที่ปรากฏของวัตถุ ซึ่งประกอบด้วยสองส่วนหลักระยะทางของ Mahalanobis สำหรับการเคลื่อนไหว: ระยะทาง Mahalanobis เป็นวิธีการวัดที่ใช้ในการประเมินความสัมพันธ์ระหว่างการเคลื่อนที่ของวัตถุ มันช่วยให้สามารถคำนวณความแตกต่างระหว่างตำแหน่งปัจจุบันและตำแหน่งที่ทำนายไว้ของวัตถุ โดยพิจารณาจากการเคลื่อนไหวและการเปลี่ยนแปลงของวัตถุในแต่ละเฟรม วิธีนี้ช่วยลดความผิดพลาดจากการติดตามวัตถุที่เคลื่อนที่อย่างรวดเร็วหรือมีการเปลี่ยนแปลงทิศทางระยะทางโคไซน์สำหรับความคล้ายคลึงของรูปลักษณ์: ระยะทางโคไซน์ใช้ในการวัดความคล้ายคลึงของลักษณะที่ปรากฏของวัตถุ โดยเปรียบเทียบลักษณะที่สกัดจากเครือข่ายประสาทเทียมกับลักษณะที่มีอยู่ในแฟร็กที่มีอยู่ วิธีนี้ช่วยให้สามารถจดจำและติดตามต่อไปได้ แม้ว่าวัตถุจะมีการเปลี่ยนแปลงในรูปลักษณ์เล็กน้อย หรือมีการบดบังเป็นระยะเวลาสั้น ๆ การใช้เมตริกซ์ที่รวมทั้งสองนี้ช่วยให้ Deep SORT สามารถจับคู่วัตถุกับแฟร็กที่มีอยู่ได้อย่างแม่นยำ โดยใช้อัลกอริธึมของฮังการี Kuhn, H. W. (1955). ในการคำนวณการ

จับคู่ที่เหมาะสมที่สุด วิธีการนี้ช่วยให้สามารถติดตามวัตถุได้อย่างต่อเนื่องและแม่นยำ แม้ในสภาพที่มีการบดบังหรือการเปลี่ยนแปลงในลักษณะที่ปรากฏของวัตถุ

### 3) การจัดการติดตาม

Deep SORT ปรับปรุงวิธีการจัดการติดตามวัตถุในวิดีโอสตรีมมิ่งโดยใช้กลไกที่ซับซ้อนในการดูแลรักษาแตร็กหรือเส้นทางการเคลื่อนที่ของวัตถุตลอดวงจรชีวิตของมัน กลไกนี้รวมถึงหลายขั้นตอนที่สำคัญ เช่น การยืนยันแตร็ก, การบำรุงรักษาแตร็ก และการลบแตร็กที่ไม่ใช้งาน การยืนยันแตร็กเกิดขึ้นเมื่อแตร็กได้รับการตรวจพบอย่างต่อเนื่องในหลายเฟรมติดต่อกัน ทำให้สามารถระบุและยืนยันว่าแตร็กนั้นแสดงถึงวัตถุที่กำลังถูกติดตามจริง แตร็กที่ได้รับการยืนยันนี้จะถูกประมวลผลต่อเพื่อรักษาความสม่ำเสมอและความน่าเชื่อถือในการติดตามในขั้นตอนของการบำรุงรักษาแตร็ก, Deep SORT จะอัปเดตและปรับปรุงแตร็กตามข้อมูลใหม่ที่ได้รับจากการตรวจจับในเฟรมล่าสุด การปรับปรุงนี้รวมถึงการปรับตำแหน่งแตร็ก, ความเร็ว และการเปลี่ยนแปลงอื่น ๆ ที่เกี่ยวข้องกับการเคลื่อนที่ของวัตถุ ทำให้สามารถติดตามวัตถุได้อย่างแม่นยำและต่อเนื่องการลบแตร็กที่ไม่ใช้งานเป็นขั้นตอนสุดท้ายในการจัดการติดตาม ซึ่งใช้พารามิเตอร์อายุเพื่อลบแตร็กที่ไม่ได้รับการตรวจพบในเฟรมล่าสุดหรือเฟรมส่วนใหญ่ หากวัตถุที่เคยติดตามหายไปจากภาพหรือไม่ได้รับการตรวจจับเป็นเวลานาน แตร็กนั้นจะถูกลบออกจากระบบ เพื่อป้องกันไม่ให้ระบบถูกโหลดด้วยข้อมูลที่ล้าสมัยหรือไม่แม่นยำการจัดการติดตามใน Deep SORT ด้วยวิธีนี้ช่วยให้ระบบสามารถรักษาความแม่นยำและประสิทธิภาพในการติดตามวัตถุได้อย่างมีประสิทธิภาพ โดยทำให้ระบบสามารถตอบสนองและปรับตัวตามการเปลี่ยนแปลงของวัตถุที่ติดตามได้อย่างรวดเร็วและแม่นยำ

## 2.5 การระบุตำแหน่งของยานพาหนะอัตโนมัติ

การระบุตำแหน่งอย่างแม่นยำเป็นหน้าที่พื้นฐานและสำคัญที่สุดของยานพาหนะอัตโนมัติ ยานพาหนะเหล่านี้ต้องสามารถนำทางและตอบสนองต่อสภาพแวดล้อมได้อย่างเชื่อถือได้เพื่อความปลอดภัยและความราบรื่นในการขับขี่ การทำความเข้าใจกับสภาพแวดล้อมโดยละเอียดและการตรวจจับสิ่งกีดขวางได้อย่างรวดเร็วจึงเป็นสิ่งจำเป็น เทคโนโลยีที่มีอยู่ในปัจจุบันให้ความสามารถในการระบุตำแหน่งที่เหนือกว่าด้วยความช่วยเหลือจากแผนที่ความละเอียดสูง (HD Maps) และการวิเคราะห์ข้อมูลด้วยวิธี Normal Distribution Transform (NDT) ซึ่งมักจะถูกนำมาใช้เป็นขั้นการประมวลผลในการเปรียบเทียบข้อมูลเซ็นเซอร์กับแผนที่ ในการจับบทความนี้ สิ่งสำคัญคือการเน้นว่า การใช้ HD Map และ NDT นั้นช่วยให้ยานพาหนะอัตโนมัติสามารถระบุตำแหน่งตัวเองได้อย่างแม่นยำและเชื่อถือได้ โดยทำให้การนำทางเป็นไปได้ด้วยความปลอดภัยและเป็นระบบมากขึ้น ซึ่งจะนำพาเราเข้าสู่อนาคตของการขนส่งที่ไร้คนขับ



### 2.5.1 แผนที่ความละเอียดสูง (HD Maps)

การวิจัยและพัฒนาแผนที่ความละเอียดสูง (High-definition map, HD map) มีบทบาทสำคัญในระบบขับเคลื่อนอัตโนมัติ โดยเริ่มแรก HD map ได้รับการพัฒนาในโครงการ Bertha Drive Project ตั้งแต่ปี 2010 ซึ่งได้รับการทดสอบบน Mercedes Benz S-Class S 500 Stiller, C. (2014, June) และได้แสดงความสามารถในการนำทางอัตโนมัติได้ในระยะทางกว่า 103 กิโลเมตรบนท้องถนนจริงการพัฒนาและการปรับปรุงแผนที่ความละเอียดสูง (High-definition Maps, HD Maps) ที่สร้างขึ้นด้วยเทคโนโลยี Lidar (Light Detection and Ranging) Mayr, M. (2018, November). นับเป็นหนึ่งในส่วนประกอบหลักที่มีบทบาทสำคัญในการทำให้ระบบขับเคลื่อนอัตโนมัติของยานพาหนะมีประสิทธิภาพและความน่าเชื่อถือสูง Lidar เป็นเครื่องมือที่ใช้วัดระยะทางโดยใช้แสงเลเซอร์ที่ส่องออกไปและวัดเวลาที่แสงนั้นเดินทางกลับมาหลังจากสะท้อนจากวัตถุต่าง ๆ ด้วยความสามารถนี้, Lidar จึงสามารถสร้างข้อมูลทางเรขาคณิตที่มีความละเอียดสูงของสภาพแวดล้อมรอบตัว ซึ่งเป็นประโยชน์อย่างมากในอุตสาหกรรมยานพาหนะอัตโนมัติการสร้างภาพแบบสามมิติหรือที่เรียกว่า 'point cloud' จากข้อมูล Lidar ช่วยให้ผู้ใช้ผลิตรายการพาหนะสามารถมองเห็นโครงสร้างพื้นผิวอุปสรรคและรายละเอียดอื่น ๆ ของท้องถนนได้ด้วยความแม่นยำที่ไม่เคยมีมาก่อนแผนที่ HD ที่สร้างจาก Lidar มีความสำคัญต่อระบบขับเคลื่อนอัตโนมัติเพราะมันช่วยให้รถยนต์สามารถทำความเข้าใจและตอบสนองต่อสภาพแวดล้อมที่มีความซับซ้อนได้อย่างแม่นยำ ไม่ว่าจะเป็นการรับรู้เส้นทาง, การจดจำสัญญาณจราจร, หรือแม้กระทั่งการปรับเปลี่ยนเส้นทางตามสภาพการจราจรในเวลาจริง แผนที่ HD ยังมีการอัปเดตอย่างต่อเนื่องเพื่อให้สะท้อนถึงการเปลี่ยนแปลงในสภาพแวดล้อม เช่น การก่อสร้างใหม่หรือการเปลี่ยนแปลงในการจัดการจราจร ทำให้ระบบขับเคลื่อนอัตโนมัติมีข้อมูลล่าสุดและแม่นยำที่สุดในการทำการตัดสินใจการปรับปรุงแผนที่อย่างต่อเนื่องนี้ช่วยลดความเสี่ยงของข้อมูลที่ล้าสมัยซึ่งอาจนำไปสู่การตัดสินใจที่ไม่ถูกต้องในระหว่างการขับเคลื่อนแผนที่ HD ที่สร้างด้วย Lidar มีคุณสมบัติที่เหนือกว่าเมื่อเทียบกับแผนที่แบบดั้งเดิมในหลายด้านสำหรับตัวอย่าง, ความสามารถในการระบุรายละเอียดและความแตกต่างของพื้นผิวถนนที่ไม่สามารถจับจัดด้วยเทคโนโลยีอื่น ๆ ได้, การจับภาพสิ่งกีดขวางที่มีขนาดเล็กอย่างเช่น กิ่งไม้หรือสิ่งปลูกสร้างชั่วคราว, และความสามารถในการแสดงภาพความลึกของสภาพแวดล้อม ช่วยให้ระบบขับเคลื่อนอัตโนมัติสามารถทำความเข้าใจสภาพแวดล้อมได้ในมิติที่ลึกขึ้นการนำเสนอข้อมูลดังกล่าวในรูปแบบแผนที่ HD นั้นสำคัญยิ่งกว่าเมื่อมีการประมวลผลข้อมูลในเวลาจริง Fallah, Y. P. (2022). แผนที่ HD ช่วยให้ระบบขับเคลื่อนอัตโนมัติสามารถปรับแผนการนำทางและการตัดสินใจขับเคลื่อนอย่างรวดเร็ว เมื่อมีการเปลี่ยนแปลงของสภาพทางหรือสภาพการจราจรที่ไม่คาดคิด เช่น การปิดถนนเนื่องจากการก่อสร้างหรืออุบัติเหตุ ระบบขับเคลื่อนอัตโนมัติที่มีการอัปเดตข้อมูลแผนที่ HD อย่างสม่ำเสมอจะสามารถนำทางได้อย่างมั่นใจและปลอดภัย ห่างไกลจากความผิดพลาดที่อาจเกิดขึ้น



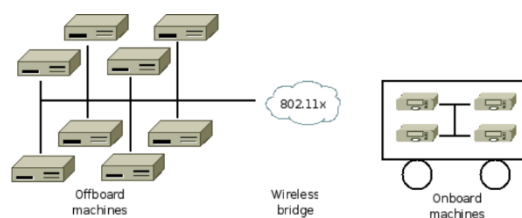
### 2.5.2 การระบุตำแหน่งด้วย Normal distribution transform

การระบุตำแหน่งของยานพาหนะด้วยการใช้แผนที่ความละเอียดสูง (HD map) และเทคนิค Normal Distribution Transform (NDT) ร่วมกับเซ็นเซอร์ Inertial Measurement Unit (IMU) เป็นกระบวนการที่สำคัญในการนำทางอัตโนมัติ NDT เป็นอัลกอริทึมที่ใช้ในการจับคู่แผนที่ 3D point cloud ที่ได้จากเซ็นเซอร์ Lidar กับแผนที่ HD เพื่อระบุตำแหน่งและทิศทางของยานพาหนะในสภาพแวดล้อมจริง IMU ให้ข้อมูลเกี่ยวกับการเคลื่อนที่และการเอียงของยานพาหนะ, ช่วยให้ระบบสามารถประมวลผลการเปลี่ยนแปลงของทิศทางและตำแหน่งได้ด้วยความเร็วสูง NDT เป็นเทคนิคการจับคู่แผนที่ที่ใช้การแปลงการกระจายความน่าจะเป็นเพื่อระบุตำแหน่ง 3D point clouds ที่ได้จาก Lidar กับแผนที่ HD ที่มีอยู่ อัลกอริทึมนี้ไม่ต้องพึ่งพาการจับคู่จุดที่ละจุดแบบดั้งเดิม แต่ใช้ความน่าจะเป็นของการแจกแจงตามปกติ (normal distribution) ของความสูงและคุณสมบัติอื่น ๆ ของสภาพแวดล้อมที่สแกนมาวิธีนี้ช่วยให้การระบุตำแหน่งแม่นยำและเร็วขึ้นเนื่องจากสามารถประมวลผลข้อมูลที่ซับซ้อนได้อย่างรวดเร็ว IMU เป็นเซ็นเซอร์ที่วัดการเคลื่อนที่และการเอียงของยานพาหนะ ซึ่งรวมถึงเซ็นเซอร์เช่น gyroscopes และ accelerometers ที่วัดการเคลื่อนไหวในสามแกนคือ pitch, roll, และ yaw ข้อมูลจาก IMU เมื่อรวมกับข้อมูลจาก GPS และ Lidar ช่วยให้ระบบสามารถรักษาและอัปเดตตำแหน่งของยานพาหนะได้แม้ในสภาพแวดล้อมที่ GPS อาจมีความไม่แม่นยำ เช่น ในอุโมงค์หรือระหว่างอาคารสูงนักวิจัยอย่าง Naoki Akai. (2017) และทีมงานจาก Gwangju Institute of Science and Technology ได้นำเสนอการใช้ NDT ร่วมกับเซ็นเซอร์ VLP-32 Lidar ในการทดลองกับยานพาหนะ KIA Soul EV, แสดงให้เห็นถึงความสามารถในการนำทางอย่างแม่นยำในระยะทางกว่า 100 กิโลเมตร การวิจัยล่าสุดจาก Sijia Liu et al. (2021) ยังได้แสดงการประยุกต์ใช้ NDT แบบ realtime ร่วมกับข้อมูลจาก GPS และ IMU ในการระบุตำแหน่งของยานพาหนะ Electric Vehicle ที่ทดลองในมหาวิทยาลัย Wuhan University of Technology, โดยมีความแม่นยำในการระบุตำแหน่งด้วย NDT ในระดับ mean square error ที่ 0.15 เมตร การใช้ NDT และ IMU ในการระบุตำแหน่งยังช่วยลดความต้องการสำหรับการมีสัญญาณ GPS ที่แม่นยำตลอดเวลา, ซึ่งอาจได้รับผลกระทบจากสิ่งกีดขวางต่าง ๆ เช่น ตึกสูงหรือสภาพแวดล้อมในเมือง ข้อมูลจาก IMU สามารถใช้เพื่อปรับปรุงการระบุตำแหน่งของยานพาหนะในเวลาจริงและช่วยเพิ่มความเร็วในการตัดสินใจของระบบขับเคลื่อนอัตโนมัติ

## 2.6 Ros (Robot Operating System)

Robot Operating System (ROS) เป็นเฟรมเวิร์กมาตรฐานสำหรับการพัฒนาซอฟต์แวร์สำหรับหุ่นยนต์ที่ช่วยให้นักพัฒนาสามารถสร้างและทดสอบอัลกอริทึมที่ซับซ้อนในสภาพแวดล้อมจำลองก่อนนำไปใช้งานจริงได้อย่างมีประสิทธิภาพ (Quigley, M., Conley, K., Gerkey, B., 2009)

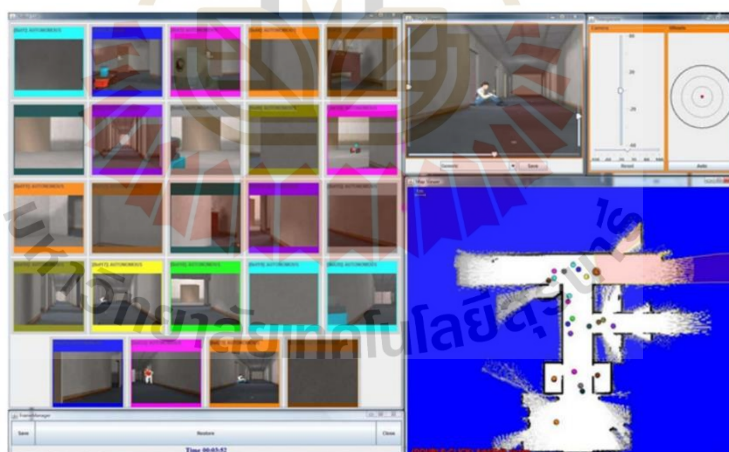
โครงสร้างพื้นฐานของ ROS ประกอบด้วย nodes ที่สื่อสารกันผ่าน messages, services และ actions ซึ่งอำนวยความสะดวกในการเขียนโค้ดที่มีโครงสร้างและแบ่งส่วนได้ชัดเจน



รูปที่ 2.9 A typical ROS network configuration (Quigley, M., Conley, K., Gerkey, B., 2009)

### 2.6.1 โครงสร้างและการทำงานของ ROS

ROS ประกอบด้วย nodes ที่สามารถสื่อสารกันผ่านการรับส่ง messages ทาง topics, services, และ action servers ซึ่งทำให้เกิดการทำงานร่วมกันได้ในระบบแบบกระจาย (Distributed computing) ช่วยให้หุ่นยนต์มีความยืดหยุ่นและการปรับตัวได้ดีในสภาพแวดล้อมที่แตกต่างกัน (Barber et al., 2011).



รูปที่ 2.10 multi-UV interface with map, camera views, and interaction through selection. (Barber et al., 2011).

### 2.6.2 ซอฟต์แวร์และไลบรารีใน ROS

ROS มีระบบแพ็คเกจที่อำนวยความสะดวกในการพัฒนาด้วยไลบรารีและเครื่องมือต่าง ๆ เช่น navigation stacks และ perception libraries ซึ่งเป็นแกนหลักในการพัฒนาหุ่นยนต์

อัตโนมัติและระบบการมองเห็นด้วยคอมพิวเตอร์ ROS ได้รับการนำไปใช้ในหลายโปรเจกต์หุ่นยนต์ระดับโลก ตั้งแต่หุ่นยนต์สำรวจดาวอังคารไปจนถึงหุ่นยนต์ทำความสะอาดในบ้าน (Toupet et al., 2020) ความสามารถในการปรับเปลี่ยนและรองรับการทำงานแบบเปิดทำให้ ROS เป็นเครื่องมือที่มีคุณค่าอย่างยิ่งในการพัฒนาและนำไปใช้ในงานวิจัยหุ่นยนต์

## 2.7 Roboflow

Roboflow เป็นแพลตฟอร์มที่ช่วยนักพัฒนาและนักวิจัยในงานคอมพิวเตอร์วิชัน (Computer Vision) จัดการกับชุดข้อมูลได้อย่างครบวงจร ตั้งแต่การเตรียมข้อมูล, การติดป้ายกำกับ (Labeling), การทำ Data Augmentation และการสร้างชุดข้อมูลสำหรับฝึกฝนโมเดล ปัญญาประดิษฐ์ Roboflow มีจุดเด่นที่ทำให้ให้นักพัฒนาสามารถจัดการข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ เช่น การรองรับไฟล์รูปแบบหลากหลายและมีเครื่องมือช่วยในกระบวนการจัดการข้อมูล ทำให้เหมาะสำหรับการใช้งานในโปรเจกต์วิจัยที่ต้องการความแม่นยำในการตรวจจับวัตถุ สามารถรวมเข้ากับเฟรมเวิร์กที่ใช้โมเดลปัญญาประดิษฐ์ยอดนิยม เช่น YOLOv8, TensorFlow, และ PyTorch ซึ่งทำให้กระบวนการฝึกฝนโมเดลสามารถทำได้อย่างรวดเร็วและแม่นยำมากขึ้น

การติดป้ายกำกับ (Labeling) Roboflow ช่วยในขั้นตอนการติดป้ายกำกับภาพ โดยการกำหนดกรอบ Bounding Box รอบวัตถุที่ต้องการตรวจจับ เช่น รถยนต์ บุคคล หรือสิ่งกีดขวางต่าง ๆ การกำหนดกรอบ Bounding Box จะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของวัตถุได้อย่างแม่นยำ ขั้นตอนนี้เป็นหนึ่งในกระบวนการที่ใช้เวลามากหากทำด้วยมือ แต่ Roboflow มีอินเทอร์เฟซที่ใช้งานง่ายและสามารถจัดการได้อย่างรวดเร็วผ่านการลากและวางยังมีฟีเจอร์ที่ช่วยในการแบ่งประเภทของวัตถุ และมีการจัดการการติดป้ายกำกับในหลาย ๆ ภาพพร้อมกัน ทำให้สามารถลดเวลาในการเตรียมข้อมูลได้มาก

การทำ Data Augmentation ช่วยเพิ่มความหลากหลายของข้อมูลโดยไม่ต้องเก็บภาพเพิ่มเติม โดยใช้วิธีการหมุนภาพ, พลิกภาพ (Flip), ปรับความสว่าง (Brightness), ขยายหรือย่อภาพ และการปรับแต่งอื่น ๆ ซึ่งจะช่วยให้โมเดลสามารถเรียนรู้จากภาพที่มีความหลากหลายมากขึ้น ส่งผลให้โมเดลมีความทนทานในการทำงานกับข้อมูลใหม่ ๆ ที่ไม่เคยเห็นมาก่อนในสภาพแวดล้อมต่าง ๆ การทำ Data Augmentation มีความสำคัญเป็นพิเศษในงานคอมพิวเตอร์วิชันที่ต้องการให้โมเดลสามารถตรวจจับวัตถุในสภาพแวดล้อมที่หลากหลาย เช่น กลางวัน กลางคืน หรือในมุมมองที่ต่างกัน

การจัดการชุดข้อมูล Roboflow ช่วยในกระบวนการจัดการชุดข้อมูลโดยอัตโนมัติ เช่น การแบ่งชุดข้อมูลออกเป็น Training Set และ Test Set โดยทั่วไปแล้ว ข้อมูลจะถูกแบ่งเป็น 80% สำหรับการฝึกฝน (Training) และ 20% สำหรับการทดสอบ (Testing) ซึ่งช่วยให้สามารถตรวจสอบและประเมินประสิทธิภาพของโมเดลในการตรวจจับวัตถุที่ไม่เคยเห็นมาก่อนได้ดีขึ้น การจัดการข้อมูล

เช่นนี้ช่วยให้มั่นใจได้ว่าโมเดลจะไม่เกิดปัญหา Overfitting และมีความสามารถในการทำนายวัตถุใหม่ๆ อย่างแม่นยำ



รูปที่ 2.11 แพลตฟอร์มที่ช่วยในการจัดการข้อมูลสำหรับการเรียนรู้ของระบบคอมพิวเตอร์วิชั่น

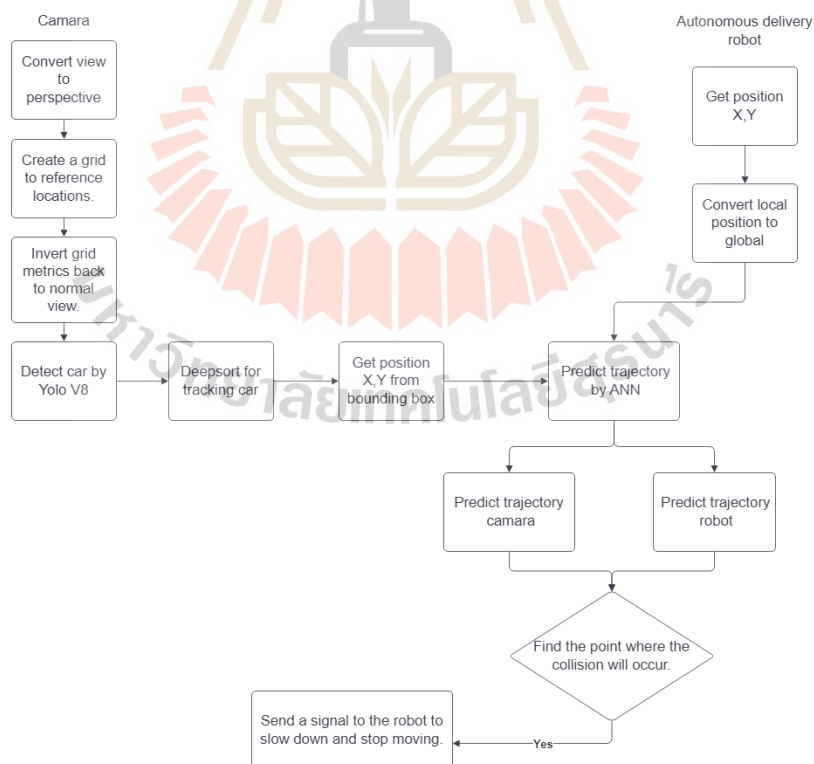
Roboflow เป็นเครื่องมือที่มีความสำคัญในกระบวนการเตรียมข้อมูลสำหรับการตรวจจับวัตถุ ซึ่งช่วยให้สามารถจัดการข้อมูลได้อย่างสะดวกและรวดเร็ว Roboflow ยังมีฟีเจอร์สำคัญ เช่น การติดป้ายกำกับ, Data Augmentation, การจัดการชุดข้อมูล, การสร้างเวอร์ชันของชุดข้อมูล และการแปลงรูปแบบข้อมูลที่สนับสนุนการทำงานในงานคอมพิวเตอร์วิชั่นได้อย่างเต็มประสิทธิภาพ



### บทที่ 3

### วิธีดำเนินงานวิจัย

วิจัยนี้จะสำรวจและอธิบายขั้นตอนต่าง ๆ ที่ผ่านการออกแบบและการทดลองเพื่อสร้างระบบการนำทางและป้องกันการชนระหว่างหุ่นยนต์ส่งของอัตโนมัติกับยานพาหนะในบริเวณทางแยกที่ไม่มีระบบช่วยเหลือหรือเซนเซอร์ประกอบการทำงาน โดยใช้กล้องเป็นเครื่องมือหลักในการจับภาพและตรวจจับวัตถุ ทั้งนี้จะประกอบด้วยการเตรียมการที่ครอบคลุมตั้งแต่การรวบรวมข้อมูลและการศึกษาค้นคว้าที่เกี่ยวข้อง การเตรียมพื้นที่ทดสอบ ไปจนถึงการทดสอบการทำงานของอุปกรณ์ต่าง ๆ การพัฒนาระบบจะเกี่ยวข้องกับการแปลงมุมมองจากกล้อง การตรวจจับและการติดตามวัตถุ การระบุตำแหน่งที่ตรวจจับได้ และการทำนายตำแหน่งต่อไปของทั้งหุ่นยนต์และรถยนต์ ส่วนท้ายของบทนี้จะอธิบายถึงการทดสอบระบบการชะลอความเร็วที่พัฒนาขึ้น เพื่อประเมินประสิทธิภาพและความเที่ยงตรงในสถานการณ์จำลองต่าง ๆ และการปรับปรุงระบบตามผลการทดสอบที่ได้รับ



รูปที่ 3.1 แผนภาพกระบวนการตรวจจับและคาดการณ์การชนสำหรับหุ่นยนต์ส่งของอัตโนมัติ

ขั้นตอนการวิจัยเริ่มต้นด้วยการเตรียมการที่ครอบคลุม ตั้งแต่การรวบรวมข้อมูลที่เกี่ยวข้องและการศึกษาค้นคว้าด้านการตรวจจับวัตถุและการทำนายตำแหน่ง การเตรียมพื้นที่ทดสอบ รวมถึงการทดสอบอุปกรณ์เพื่อให้มั่นใจถึงความพร้อมในการใช้งาน การพัฒนาระบบตรวจจับและติดตามวัตถุเริ่มจากการแปลงมุมมองจากกล้องให้เป็นแบบ perspective เพื่อให้เห็นภาพมุมกว้างและชัดเจน จากนั้นจะสร้างกริดอ้างอิงตำแหน่งบนภาพที่ได้เพื่อกำหนดพิกัดและตำแหน่งของวัตถุ จากนั้นจะดำเนินการตรวจจับและติดตามวัตถุด้วยโมเดล YOLO V8 สำหรับการระบุตำแหน่งของวัตถุ โดยใช้ Deep Sort เพื่อเพิ่มความแม่นยำในการติดตามวัตถุที่เคลื่อนที่ในส่วนของการทำนายตำแหน่ง ระบบจะใช้โมเดลปัญญาประดิษฐ์ (Artificial Neural Network: ANN) ในการคาดการณ์เส้นทางการเคลื่อนที่ของวัตถุ ทั้งจากกล้องและจากตำแหน่งของหุ่นยนต์อัตโนมัติ ซึ่งประกอบด้วย การคาดการณ์วิถีของหุ่นยนต์และยานยนต์ในบริเวณเดียวกันเพื่อระบุตำแหน่งที่อาจเกิดการชน การประมวลผลดังกล่าวช่วยให้ระบบสามารถส่งสัญญาณเตือนและให้หุ่นยนต์หยุดการเคลื่อนที่หากพบว่ามีความเสี่ยงสูงในการชน

### 3.1 การเตรียมการ

ในขั้นตอนการเตรียมการได้ดำเนินการตามลำดับต่อไปนี้

#### 3.1.1 การศึกษาค้นคว้าและรวบรวมงานวิจัยที่เกี่ยวข้อง

การเตรียมการสำหรับงานวิจัยนี้ได้เริ่มต้นด้วยการทำวิจัยเชิงสำรวจเกี่ยวกับเทคโนโลยีการมองเห็นของเครื่องจักร (Machine Vision) และปัญญาประดิษฐ์ (Artificial Intelligence) จากแหล่งข้อมูลที่หลากหลาย เช่น วารสารวิชาการ, งานวิจัย, หนังสือ, และบทความที่เกี่ยวข้อง ด้วยเป้าหมายที่จะพัฒนาระบบการตรวจจับและทำนายตำแหน่งของยานพาหนะและหุ่นยนต์ส่งของอัตโนมัติให้มีประสิทธิภาพสูงสุด การปรับปรุงมุมมองของกล้องวงจรปิดให้เป็นมุมมองแบบ Bird Eye View เป็นขั้นตอนสำคัญที่ช่วยในการสร้างกริดเพื่ออ้างอิงตำแหน่งและทำการตรวจจับอย่างแม่นยำ การใช้งานโมเดล YOLO V8 สำหรับการตรวจจับวัตถุและระบบ DeepSort สำหรับการติดตามวัตถุเคลื่อนที่ ได้เป็นพื้นฐานในการพัฒนาระบบแจ้งเตือนเพื่อป้องกันการชน ซึ่งช่วยเพิ่มมาตรฐานความปลอดภัยในสภาพแวดล้อมที่มีการใช้งานร่วมกันการวิจัยยังครอบคลุมถึงการใช้ภาษา Python ไปด้วย



Robot Operating System (ROS) เพื่อพัฒนาระบบที่มีความสามารถในการตอบสนองต่อเหตุการณ์ต่าง ๆ ได้อย่างรวดเร็วและแม่นยำ

### 3.1.2 การเตรียมพื้นที่ที่จะทำการทดสอบ

พื้นที่ทดสอบที่เลือกสำหรับการทดลองนี้เป็นพื้นที่ภายในที่ปิด ซึ่งจำลองเป็นบริเวณทางแยกเพื่อจำลองสภาพแวดล้อมจริงที่รถอัตโนมัติอาจจะเจอ พื้นที่มีลักษณะเป็นพื้นผิวแบนที่กว้างขวาง ช่วยให้สามารถทำการทดสอบการเคลื่อนที่ของรถยนต์ได้โดยไม่มีข้อจำกัด และเหมาะสมกับการติดตั้งกล้องเพื่อการตรวจจับและการติดตามตำแหน่ง การที่พื้นที่นี้อยู่ภายในยังช่วยลดปัจจัยรบกวนจากภายนอก เช่น แสงแดดที่แปรปรวนหรือสภาพอากาศ ซึ่งอาจมีผลต่อประสิทธิภาพของเซ็นเซอร์และระบบการทำงานของรถอัตโนมัติ



รูปที่ 3.2 ภาพแสดงพื้นที่จำลองในการทำงาน

ในภาพมีรถยนต์ขนาดเล็กที่วางอยู่บนพื้นที่ทดสอบ ซึ่งใช้เป็นตัวแทนในการทดสอบ และกล้องที่ติดตั้งบนขาตั้งที่มีมุมมองที่เหมาะสมเพื่อจับภาพการเคลื่อนไหวของรถยนต์ การจัดตั้งพื้นที่ดังกล่าวช่วยให้การทดลองสามารถควบคุมได้อย่างแม่นยำและเก็บข้อมูลที่จำเป็นได้อย่างสมบูรณ์ สำหรับการวิเคราะห์และการปรับปรุงระบบ

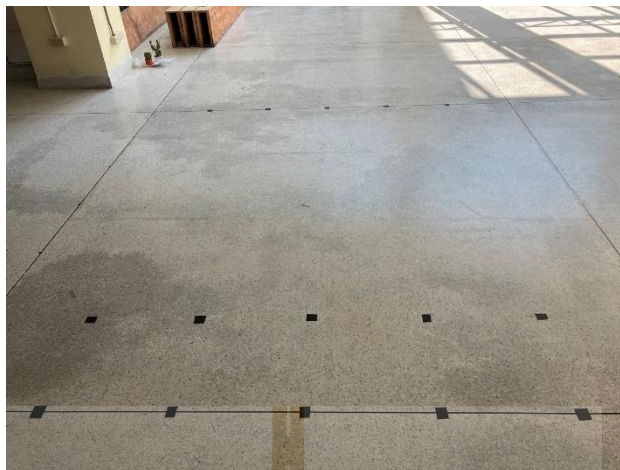


รูปที่ 3.3 ภาพแสดงการจำลองสภาพแวดล้อมจำลองของรถตามท้องถนนทั่วไป



รูปที่ 3.4 ภาพแสดงการจำลองสภาพแวดล้อมจำลองของรถอัตโนมัติ

จากภาพแสดงให้เห็นถึงการจำลองสภาพแวดล้อมที่มุกติกที่เป็นจุดอับสายตาของระบบอัตโนมัติ



รูปที่ 3.5 พื้นที่ทดสอบที่มีการทำกริดสำหรับทดสอบการนำทางอัตโนมัติ

ภาพที่แสดงมีการทำเครื่องหมายบนพื้นเพื่อสร้างกริดโดยแต่ละจุดห่างกัน 50 เซนติเมตร การตั้งกริดนี้เป็นส่วนสำคัญในการทดสอบเพราะช่วยให้สามารถวัดและติดตามการเคลื่อนที่ของวัตถุในพื้นที่ได้แม่นยำ แต่ละช่องในกริดทำหน้าที่เป็นจุดอ้างอิงสำหรับการตรวจจับและการทำงานตำแหน่ง ซึ่งจำเป็นสำหรับการทดสอบระบบตรวจจับและการทำงานการทำเครื่องหมายเหล่านั้นบนพื้นถือเป็นการจำลองสภาพแวดล้อมที่ควบคุมได้ซึ่งเหมาะสมสำหรับการทดสอบระบบการนำทางอัตโนมัติและการทำงานของเซนเซอร์ มันยังช่วยให้สามารถทดสอบอัลกอริทึมต่าง ๆ เช่น YOLO และ Deep Sort ในสภาพแวดล้อมที่จำกัดและเข้าใจได้ง่ายและยังช่วยให้สามารถทำการทดสอบซ้ำได้โดยคงไว้ซึ่งความสม่ำเสมอของตำแหน่งและระยะทางในแต่ละการทดลอง เพื่อเก็บข้อมูลที่สามารถนำไปวิเคราะห์ได้อย่างถูกต้อง

### 3.1.3 ทดสอบการทำงานของกล้อง

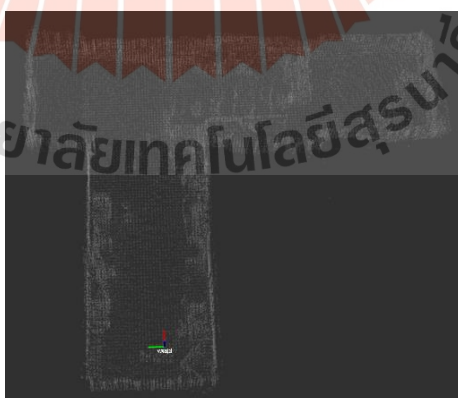
การทดสอบการทำงานของกล้องด้วย OpenCV เป็นกระบวนการที่สำคัญในการพัฒนาระบบวิชันคอมพิวเตอร์เพื่อการตรวจจับและติดตามวัตถุภายในภาพที่ได้จากกล้อง การทดสอบนี้มุ่งเน้นไปที่การประเมินความสามารถของกล้องในการจับภาพที่ชัดเจนและการแปลงภาพที่ได้ให้อยู่ในรูปแบบที่เหมาะสมสำหรับการประมวลผลต่อไป



รูปที่ 3.6 ภาพการทดสอบเปิดกล้องโดยใช้ Python และ ไลบรารี OpenCV

#### 3.1.4 ทดสอบการทำงานของแผนที่ความละเอียดสูง

การทดสอบการทำงานของแผนที่ความละเอียดสูงใน rviz เป็นกระบวนการสำคัญที่ช่วยให้นักวิจัยและนักพัฒนาสามารถตรวจสอบและประเมินความเหมาะสมของแผนที่ที่จะใช้ในการนำทางอัตโนมัติสำหรับหุ่นยนต์ภายใน ROS แผนที่ที่มีความละเอียดสูงจำเป็นต้องถูกตรวจสอบอย่างละเอียดเพื่อให้มั่นใจว่ามีความแม่นยำสูง และปราศจากข้อผิดพลาดที่อาจส่งผลกระทบต่อการทำงานของหุ่นยนต์



รูปที่ 3.7 ทดสอบการใช้ HD Maps บนซอฟต์แวร์ rviz



### 3.2 การพัฒนาระบบเพิ่มความปลอดภัยของระบบอัตโนมัติ

การพัฒนาระบบเพิ่มความปลอดภัยของระบบอัตโนมัติของรถอัตโนมัติในสภาพแวดล้อมที่เป็นมุมอับไม่ว่าจะเป็นมองไม่เห็นหรือเห็นได้ยากโดยใช้ระบบปฏิบัติการหุ่นยนต์ (ROS) ซึ่งเป็นมาตรฐานในอุตสาหกรรมสำหรับการพัฒนาแอปพลิเคชันหุ่นยนต์เทคโนโลยีนี้ช่วยให้การพัฒนาของอุปกรณ์และเซ็นเซอร์ต่าง ๆ เป็นไปอย่างราบรื่น และสามารถสื่อสารกับระบบขับเคลื่อนได้อย่างมีประสิทธิภาพโดยใช้แผนที่ความละเอียดสูงยังช่วยให้หุ่นยนต์สามารถระบุตำแหน่งและนำทางในสภาพแวดล้อมที่มีการเปลี่ยนแปลงอย่างต่อเนื่องได้อย่างแม่นยำ โดยการใช้ข้อมูลจากแผนที่เหล่านี้ช่วยให้หุ่นยนต์ประเมินสถานการณ์และปรับเส้นทางการเคลื่อนที่ได้ทันที เพื่อป้องกันการชนกับวัตถุหรือบุคคลอื่น ๆ ที่อาจเกิดขึ้นในเส้นทางการเดินทาง โดยแยกหัวข้อในการพัฒนาระบบต่าง ๆ ออกเป็นหัวข้อย่อยต่อไปนี้

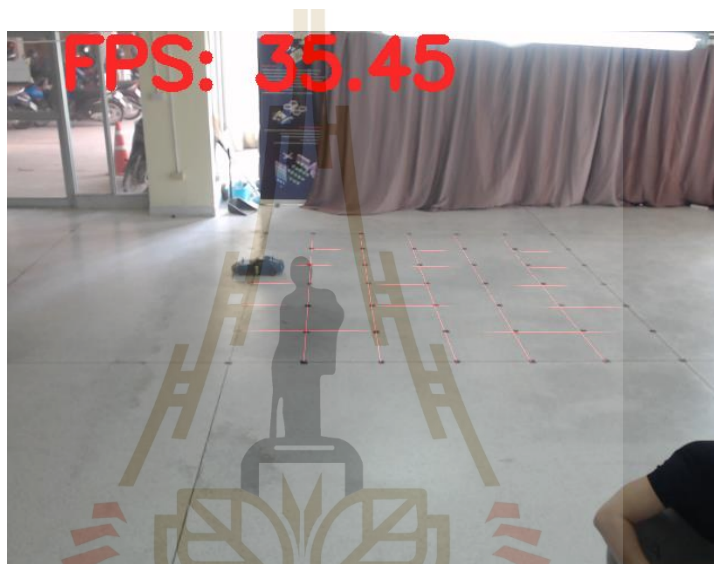
#### 3.2.1 การแปลงมุมมองกล้องและการสร้างตารางเพื่อใช้อ้างอิงตำแหน่ง

การแปลงมุมมองกล้องและการสร้างตารางเพื่อใช้อ้างอิงตำแหน่งเป็นส่วนสำคัญในการพัฒนาระบบช่วยเพิ่มการมองเห็นของระบบนำทางอัตโนมัติของหุ่นยนต์ กระบวนการนี้เริ่มต้นด้วยการจับภาพมุมมองตรงของพื้นที่ที่หุ่นยนต์จะเคลื่อนที่ ซึ่งภาพถูกแบ่งออกเป็นตารางโดยใช้เส้นตารางสีแดงเพื่อแบ่งบริเวณการทำงานออกเป็นส่วน ๆ หลังจากนั้นทำการแปลง Perspective Transformation ถูกใช้เพื่อแปลงภาพเป็นมุมมองที่เหมาะสมยิ่งขึ้นสำหรับการวิเคราะห์โดยหุ่นยนต์ ในที่นี้ คือ การแปลงมุมมองกล้องให้มองจากด้านบนลงไปยังพื้นที่ทำงานเพื่อให้หุ่นยนต์สามารถตีความตำแหน่งต่าง ๆ ได้อย่างแม่นยำมากขึ้น โดยมุมมองนี้จะช่วยให้หุ่นยนต์ประมวลผลข้อมูลได้ง่ายขึ้นในการนำทางและวางแผนเส้นทาง



รูปที่ 3.8 เป็นการแปลงรูปภาพ Bird's eye view

การแปลงมุมมองกลับ (Inverse Perspective Mapping) เป็นกระบวนการที่ใช้ในการย้อนกลับภาพที่ได้จากการแปลง Perspective กลับไปยังมุมมองตามปกติ กระบวนการนี้ช่วยให้สามารถประเมินความสัมพันธ์ระหว่างตำแหน่งจริงบนพื้นและตำแหน่งที่ปรากฏในระบบติดตามการเคลื่อนที่ของหุ่นยนต์ได้ การวิเคราะห์นี้สำคัญมากในการทำให้หุ่นยนต์สามารถปฏิบัติการได้โดยอิสระและปลอดภัยโดยผ่านการใช้กระบวนการเหล่านี้หุ่นยนต์จึงมีความสามารถในการระบุตำแหน่งของวัตถุและอุปสรรคในสภาพแวดล้อมของมันช่วยลดโอกาสในการเกิดข้อผิดพลาดหรืออุบัติเหตุระหว่างการทำงาน



รูปที่ 3.9 เป็นการแปลงรูปภาพ Inverse Perspective Mapping

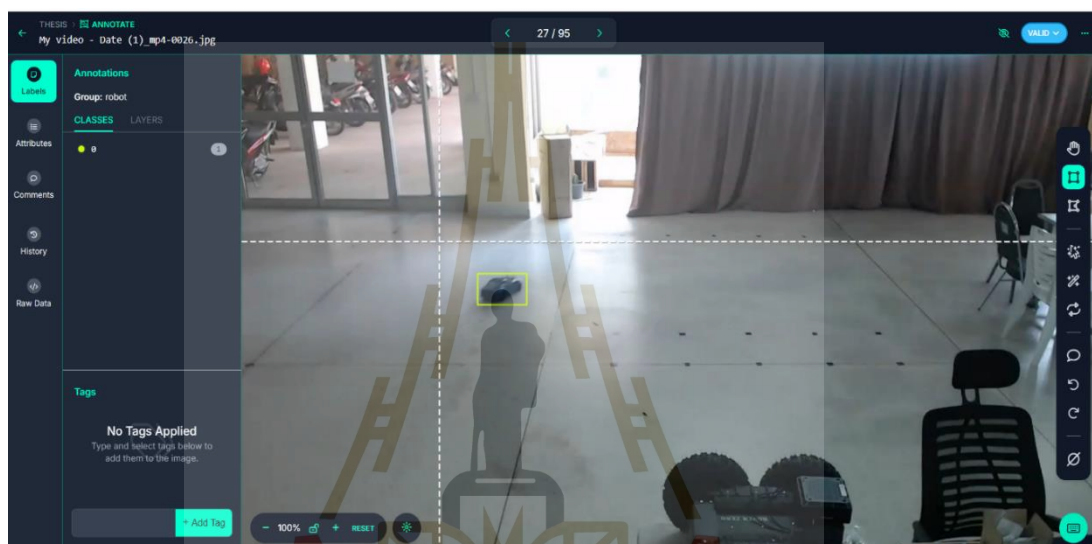
### 3.2.2 การตรวจจับวัตถุจากภาพ

การตรวจจับวัตถุจากภาพ (Object Detection) เป็นกระบวนการที่ใช้ในการระบุและแยกแยะวัตถุที่ปรากฏในภาพหรือวิดีโออย่างแม่นยำ โดยทั่วไปแล้วจะใช้เทคโนโลยีปัญญาประดิษฐ์ประเภทเครือข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks - CNN) เพื่อประมวลผลภาพและตรวจจับวัตถุที่อยู่ในภาพอย่างรวดเร็วและมีประสิทธิภาพ YOLO (You Only Look Once) รุ่น V8 เป็นหนึ่งในโมเดลที่ถูกใช้อย่างแพร่หลายในการตรวจจับวัตถุเนื่องจากสามารถประมวลผลได้อย่างรวดเร็วและแม่นยำ โดยโมเดลนี้จะทำการแบ่งภาพออกเป็นกริดเล็ก ๆ และทำการทำนายวัตถุในแต่ละเซลล์ของกริดพร้อมกัน ทำให้สามารถตรวจจับวัตถุหลายอย่างในภาพได้อย่างรวดเร็วและประสิทธิภาพสูง

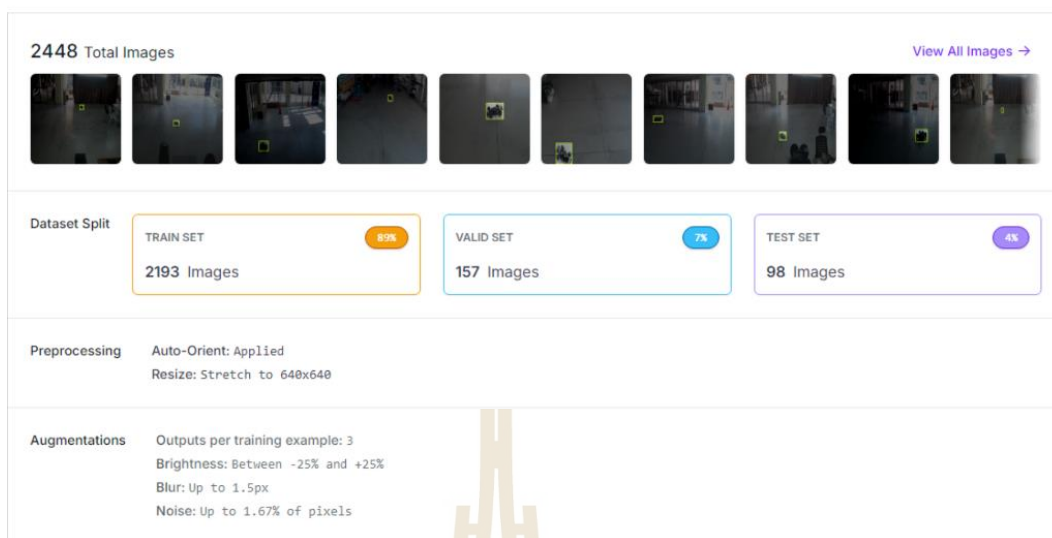


ในขั้นตอนการรวบรวมภาพนี้จะทำการบันทึกภาพเป็นวิดีโอเพื่อใช้ในการเก็บข้อมูลโดยที่วิดีโอจะเป็นสถานที่ที่ต้องการใช้งานของระบบเพื่อเพิ่มความแม่นยำมากยิ่งขึ้นในการใช้งานระบบ

การเลือกภาพที่ใช้ในการติดป้ายกำกับ เนื่องจากการฝึกโมเดลปัญญาประดิษฐ์ต้องการข้อมูลที่มีความหลากหลาย ภาพที่เลือกจะต้องครอบคลุมหลายสถานการณ์ทั้งในภาพในสภาพแสงที่แตกต่างกัน กลางวันกลางคืนและมุมมองของกล้องที่หลากหลายทั้งมุมสูงมุมต่ำเพื่อให้โมเดลสามารถเรียนรู้และทำนายได้อย่างแม่นยำในสถานการณ์ต่าง ๆ ที่อาจเกิดขึ้น



รูปที่ 3.10 ขั้นตอนการติดป้ายกำกับโดยใช้เครื่องมือ Roboflow



รูปที่ 3.11 ขั้นตอนการแบ่ง Data ก่อนนำไปสร้างโมเดลการเรียนรู้โดยใช้เครื่องมือ Roboflow

หลังจากที่รวบรวมภาพได้แล้ว ภาพเหล่านั้นจะถูกนำไปผ่านกระบวนการติดป้ายกำกับโดยใช้ Roboflow ซึ่งเป็นเครื่องมือที่ช่วยในการระบุวัตถุในภาพโดยใช้ Bounding Box กำหนดขอบเขตวัตถุที่ต้องการตรวจจับข้อมูลที่ติดป้ายกำกับจะถูกแบ่งออกเป็นสามชุด คือ Train Set สำหรับการฝึกฝนโมเดล Validation Set สำหรับการตรวจสอบความแม่นยำ และ Test Set สำหรับการทดสอบโมเดล โดยข้อมูลถูกแบ่งออกเป็น 80% สำหรับการฝึก 10% สำหรับการทดสอบ และ 10% สำหรับการตรวจสอบเพื่อเพิ่มประสิทธิภาพในการเรียนรู้ Roboflow ยังมีฟีเจอร์สำหรับการทำ Data Augmentation เป็นการเพิ่มความสว่างและลดความสว่างของภาพ การเพิ่มความเบลอ และการเพิ่มรอยขีดข่วนในภาพการทำ Augmentation เหล่านี้ช่วยให้โมเดลสามารถตรวจจับวัตถุได้อย่างแม่นยำในสภาพแวดล้อมที่แตกต่างกัน ทำให้ข้อมูลมีความหลากหลายมากขึ้น และโมเดลสามารถทำงานได้ดีขึ้นในสภาพแวดล้อมจริง

### 3.2.3 การสร้างโมเดลการเรียนรู้

การสร้างโมเดลการเรียนรู้เริ่มจากการเตรียมข้อมูลที่ติดป้ายกำกับ (Labeling) ผ่านแพลตฟอร์ม Roboflow ข้อมูลเหล่านี้ถูกแบ่งออกเป็นชุดฝึก (Training Set), ชุดตรวจสอบ (Validation Set), และชุดทดสอบ (Test Set) เพื่อให้โมเดลสามารถเรียนรู้และทดสอบได้อย่างแม่นยำ การใช้ YOLOv8 ในการฝึกโมเดลนี้มีความสำคัญ เนื่องจาก YOLOv8 ถูกออกแบบมาให้สามารถตรวจจับวัตถุได้รวดเร็วและแม่นยำ โมเดลนี้สามารถเรียนรู้ข้อมูลที่หลากหลายจากภาพถ่ายที่มีการปรับแต่งเพื่อเพิ่มประสิทธิภาพในการตรวจจับ

```
!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=200 imgsz=640 plots=True
```

รูปที่ 3.12 คำสั่งที่ใช้ในการฝึกโมเดล 200 รอบ (Epochs) และปรับขนาดภาพเป็น 640 พิกเซล

ขั้นตอนการฝึกโมเดลเริ่มต้นจากการนำข้อมูลที่เตรียมไว้เข้าสู่ระบบประมวลผล โดยใช้เครือข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) ซึ่ง YOLOv8 สามารถตรวจจับวัตถุได้อย่างมีประสิทธิภาพในภาพถ่ายแต่ละภาพ กระบวนการฝึกฝนโมเดลจะถูกตั้งค่าให้ทำการฝึกเป็นจำนวน 200 รอบ (Epochs) ซึ่งช่วยให้โมเดลมีโอกาสรียนรู้จากข้อมูลที่ป้อนเข้ามามากขึ้น โดยมีการปรับขนาดภาพเป็น 640x640 พิกเซล ขนาดนี้เหมาะสมต่อการฝึกโมเดลโดยไม่ลดทอนประสิทธิภาพหรือเพิ่มภาระการประมวลผลที่สูงเกินไป

|     | from     | n | params  | module                               | arguments               |
|-----|----------|---|---------|--------------------------------------|-------------------------|
| 0   | -1       | 1 | 928     | ultralytics.nn.modules.conv.Conv     | [3, 32, 3, 2]           |
| 1   | -1       | 1 | 18560   | ultralytics.nn.modules.conv.Conv     | [32, 64, 3, 2]          |
| 2   | -1       | 1 | 29056   | ultralytics.nn.modules.block.C2f     | [64, 64, 1, True]       |
| 3   | -1       | 1 | 73984   | ultralytics.nn.modules.conv.Conv     | [64, 128, 3, 2]         |
| 4   | -1       | 2 | 197632  | ultralytics.nn.modules.block.C2f     | [128, 128, 2, True]     |
| 5   | -1       | 1 | 295424  | ultralytics.nn.modules.conv.Conv     | [128, 256, 3, 2]        |
| 6   | -1       | 2 | 788480  | ultralytics.nn.modules.block.C2f     | [256, 256, 2, True]     |
| 7   | -1       | 1 | 1180672 | ultralytics.nn.modules.conv.Conv     | [256, 512, 3, 2]        |
| 8   | -1       | 1 | 1838080 | ultralytics.nn.modules.block.C2f     | [512, 512, 1, True]     |
| 9   | -1       | 1 | 656896  | ultralytics.nn.modules.block.SPPF    | [512, 512, 5]           |
| 10  | -1       | 1 | 0       | torch.nn.modules.upsampling.Upsample | [None, 2, 'nearest']    |
| 11  | [-1, 6]  | 1 | 0       | ultralytics.nn.modules.conv.Concat   | [1]                     |
| 12  | -1       | 1 | 591360  | ultralytics.nn.modules.block.C2f     | [768, 256, 1]           |
| 13  | -1       | 1 | 0       | torch.nn.modules.upsampling.Upsample | [None, 2, 'nearest']    |
| 14  | [-1, 4]  | 1 | 0       | ultralytics.nn.modules.conv.Concat   | [1]                     |
| 15  | -1       | 1 | 148224  | ultralytics.nn.modules.block.C2f     | [384, 128, 1]           |
| 16  | -1       | 1 | 147712  | ultralytics.nn.modules.conv.Conv     | [128, 128, 3, 2]        |
| 17  | [-1, 12] | 1 | 0       | ultralytics.nn.modules.conv.Concat   | [1]                     |
| ... |          |   |         |                                      |                         |
|     | all      |   | 163     | 157                                  | 0.981 0.987 0.991 0.713 |

Speed: 0.4ms preprocess, 2.3ms inference, 0.0ms loss, 2.2ms postprocess per image

รูปที่ 3.13 โครงสร้าง CNN ของ โมเดล Yolo V8

ในการฝึกโมเดลจะใช้ฟังก์ชันการสูญเสีย (Loss Function) ซึ่งทำหน้าที่ประเมินความแตกต่างระหว่างผลลัพธ์ที่โมเดลทำนายและค่าจริงที่กำหนดไว้ หากโมเดลทำการทำนายผิดพลาด ฟังก์ชันการสูญเสียจะช่วยปรับปรุงค่าพารามิเตอร์ในโมเดลให้แม่นยำมากขึ้นในรอบถัดไป การปรับปรุงค่าพารามิเตอร์นี้จะทำให้โมเดลสามารถทำนายวัตถุได้อย่างแม่นยำมากยิ่งขึ้นในแต่ละรอบ

หลังจากที่โมเดลทำการฝึกฝนแล้ว จะนำโมเดลไปทดสอบด้วยชุดข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนเพื่อประเมินความสามารถในการตรวจจับวัตถุ การทดสอบโมเดลนี้สำคัญอย่างยิ่งในการตรวจสอบว่าโมเดลสามารถนำไปใช้ในสถานการณ์จริงได้หรือไม่ โดยข้อมูลที่ใช้ในการทดสอบจะเป็นภาพที่ถูกเก็บไว้เพื่อใช้ประเมินประสิทธิภาพของโมเดล ซึ่งจะมีการเปรียบเทียบผลลัพธ์ที่ได้กับค่าที่แท้จริงในระหว่างการฝึกโมเดลจะมีการแสดงผล Confusion Matrix เพื่อให้สามารถวิเคราะห์ผลการตรวจจับของโมเดลได้อย่างละเอียด เครื่องมือนี้ช่วยให้ผู้พัฒนาทราบถึงความถูกต้องของการตรวจจับในแต่ละหมวดหมู่วัตถุที่สนใจ และปรับปรุงโมเดลให้มีประสิทธิภาพดียิ่งขึ้นในการทำนายตำแหน่งและขนาดของวัตถุโมเดลจะถูกนำไปใช้งานในการทำนายผลจากชุดข้อมูลที่ยังไม่เคยผ่านการฝึกมาก่อน (Inference) การทำนายผลนี้ช่วยให้สามารถตรวจสอบได้ว่าโมเดลที่สร้างขึ้นมีความสามารถในการตรวจจับวัตถุได้จริงในสถานการณ์ที่หลากหลายหรือไม่ การแสดงผลของโมเดลจะถูกแสดงผ่าน Bounding Box รอบวัตถุที่ถูกตรวจจับ ทำให้ผู้พัฒนาสามารถตรวจสอบได้อย่างชัดเจนว่าผลลัพธ์ตรงกับสิ่งที่ต้องการหรือไม่



รูปที่ 3.14 ทดสอบการใช้โมเดลตรวจจับวัตถุที่ได้จากการฝึก

### 3.2.4 การติดตามวัตถุ

การติดตามวัตถุ (Object Tracking) เป็นกระบวนการที่ต่อเนื่องจากการตรวจจับวัตถุ โดยหลังจากที่วัตถุถูกตรวจพบแล้ว ระบบจะทำการติดตามการเคลื่อนที่ของวัตถุในหลาย เฟรมของวิดีโอ เทคโนโลยีการติดตามที่ใช้ได้ผลดี คือ DeepSort เป็นที่นิยมในการติดตามวัตถุแบบเรียลไทม์ โดยอาศัยการแยกแยะตำแหน่งของวัตถุจาก bounding box ที่ตรวจพบในแต่ละเฟรม การติดตามวัตถุมีความสำคัญอย่างยิ่งในระบบอัตโนมัติเพื่อทำให้สามารถคาดการณ์การเคลื่อนไหวและตัดสินใจได้อย่างแม่นยำ

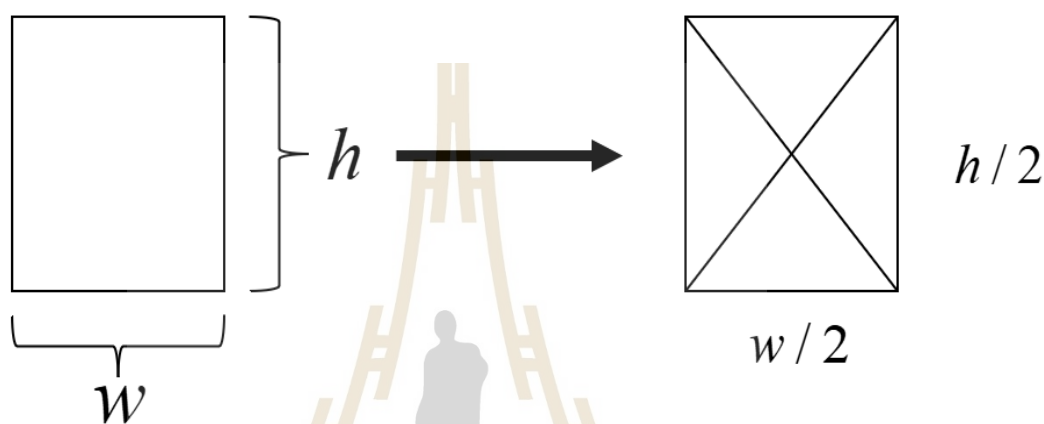
การติดตามวัตถุ (Object Tracking) เป็นขั้นตอนต่อเนื่องจากการตรวจจับวัตถุในภาพหรือวิดีโอที่ระบบได้ดำเนินการไปแล้ว หลังจากวัตถุถูกตรวจพบในเฟรมแรก ระบบจะทำการติดตามการเคลื่อนที่ของวัตถุในเฟรมถัดไป การติดตามตำแหน่งของวัตถุในแต่ละเฟรมมีความสำคัญอย่างยิ่งในการคาดการณ์ทิศทางและความเร็วของวัตถุเพื่อให้ระบบอัตโนมัติสามารถตอบสนองได้ทันที่ ในงานวิจัยนี้ ใช้อัลกอริทึม DeepSort ในการติดตามวัตถุ DeepSort จะทำการคำนวณตำแหน่งของวัตถุที่ถูกตรวจจับในแต่ละเฟรมโดยใช้ข้อมูลจาก Bounding Box ที่ได้จากโมเดล YOLOv8 ระบบจะทำการเชื่อมโยงข้อมูลของวัตถุในเฟรมต่าง ๆ เข้าด้วยกัน และคำนวณตำแหน่งที่คาดการณ์ของวัตถุในเฟรมถัดไป โดยอิงจากทิศทางและความเร็วที่ได้จากการประมวลผลข้อมูลในเฟรมก่อนหน้า DeepSort ยังสามารถแยกแยะวัตถุที่เคลื่อนที่ในภาพหลายเฟรมได้อย่างแม่นยำ โดยคำนึงถึงลักษณะเฉพาะของวัตถุแต่ละชิ้น การติดตามตำแหน่งและการเคลื่อนไหวของวัตถุในแต่ละเฟรมทำให้ระบบสามารถตอบสนองต่อวัตถุที่เคลื่อนที่ได้อย่างมีประสิทธิภาพ และเป็นองค์ประกอบสำคัญในระบบอัตโนมัติที่ต้องการความแม่นยำในการคาดการณ์เส้นทางการเคลื่อนที่

### 3.2.5 การระบุตำแหน่งของวัตถุที่ตรวจจับจากกล้อง

หลังจากขั้นตอนการแปลงมุมมองกล้องและการสร้างตารางกริดสำหรับอ้างอิงตำแหน่งเสร็จสิ้นแล้ว ระบบจะทำการตรวจจับวัตถุจากภาพที่ได้จากกล้อง การระบุตำแหน่งของวัตถุที่ตรวจจับจากกล้องจะดำเนินการโดยอ้างอิงตำแหน่งของวัตถุในภาพจากกรอบ Bounding Box ที่ได้จากการตรวจจับ และเทียบตำแหน่งนั้นกับตารางกริดที่ถูกสร้างขึ้น ตารางกริดที่ถูกสร้างขึ้นจากการแปลงมุมมองของกล้องจะถูกใช้เป็นระบบพิกัดอ้างอิงสำหรับการแปลงตำแหน่งของวัตถุจากภาพในมุมมองกล้องไปสู่ตำแหน่งจริงในสภาพแวดล้อมการแปลงตำแหน่งนี้ใช้การคำนวณจากตารางกริดที่สร้างขึ้นเพื่อแปลงพิกัดของวัตถุจากภาพให้สอดคล้องกับระบบพิกัดโลก ซึ่งจะช่วยให้ระบบสามารถทราบตำแหน่งที่แท้จริงของวัตถุในสภาพแวดล้อมจริงได้อย่างแม่นยำ



ขั้นตอนการคำนวณพิกัดของจุดกึ่งกลางของ Bounding Box ทำโดยการใช้ค่าพิกัดของ Bounding Box ที่ได้จากการทำนายวัตถุในภาพ ขั้นแรกให้นำความกว้างของ Bounding Box มาหารครึ่ง เพื่อหาดำแหน่งแนวนอน (แกน X) ของจุดกึ่งกลาง จากนั้น นำความสูงของ Bounding Box มาหารครึ่ง เพื่อหาดำแหน่งแนวตั้ง (แกน Y) ของจุดกึ่งกลาง พิกัดจุดกึ่งกลางนี้จะใช้ในการอ้างอิงตำแหน่งของวัตถุในตารางกริดที่สร้างไว้ ทำให้สามารถระบุตำแหน่งจริงในสภาพแวดล้อมได้



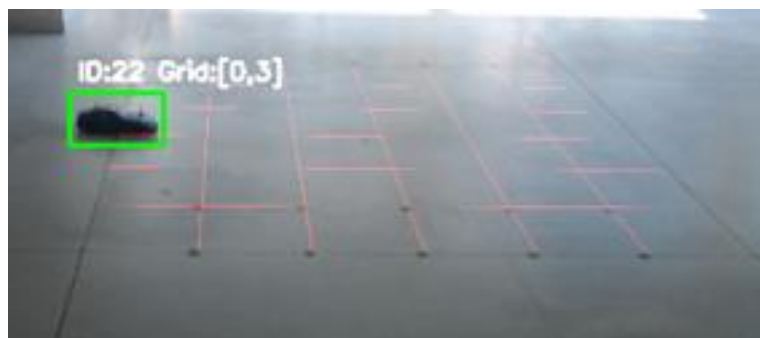
รูปที่ 3.15 วิธีหาจุดกึ่งกลางของ Bounding Box

หลังจากที่คำนวณหาจุดกึ่งกลางของ Bounding Box แล้ว พิกัดของจุดกึ่งกลางจะถูกนำมาเทียบกับตารางกริดที่สร้างขึ้นมา ระบบจะใช้พิกัดจุดนี้เพื่อเทียบกับตำแหน่งบนกริด ซึ่งกริดนี้ทำหน้าที่เป็นระบบพิกัดที่อ้างอิงกับพื้นที่จริงในสภาพแวดล้อม ทำให้สามารถแปลงพิกัดจากภาพที่ได้จากกล้องไปยังตำแหน่งจริงในโลกได้อย่างแม่นยำ การเปรียบเทียบพิกัดนี้ช่วยให้ระบบสามารถระบุได้ว่าวัตถุอยู่ที่ตำแหน่งใดในพื้นที่จริง

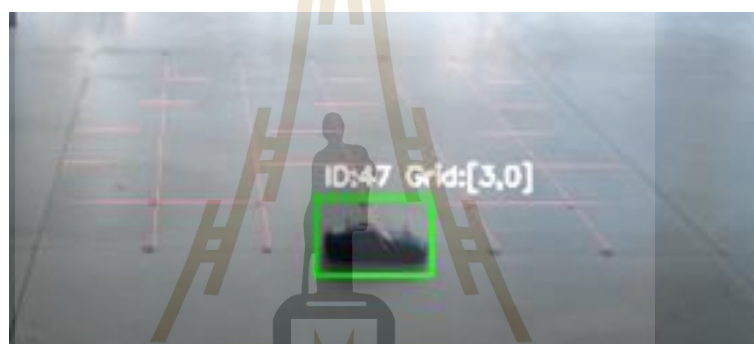


รูปที่ 3.16 การตรวจจับวัตถุในกริด [1,1]





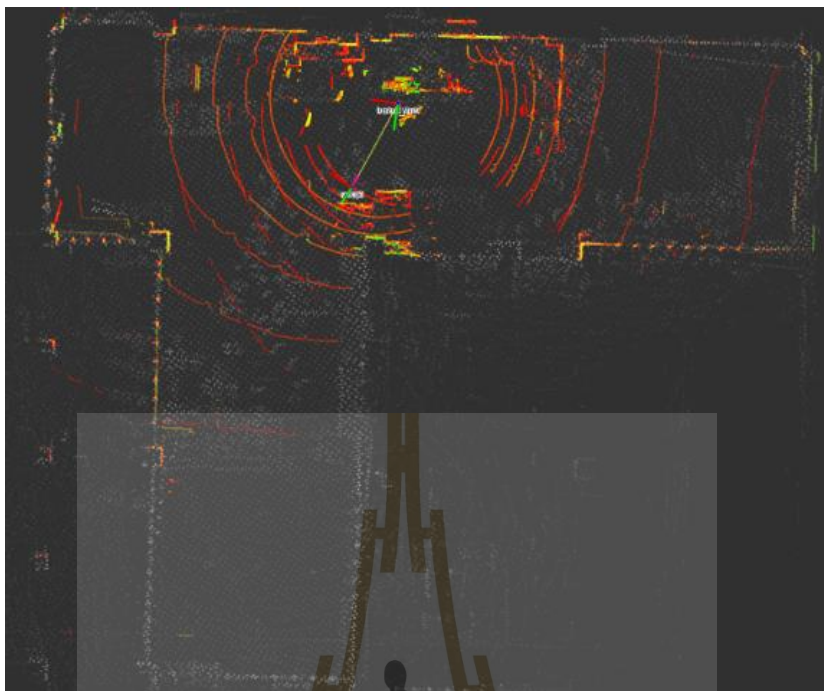
รูปที่ 3.17 การตรวจจับวัตถุในกริด [0,3]



รูปที่ 3.18 การตรวจจับวัตถุในกริด [3,0]

### 3.2.6 การระบุตำแหน่งของรถอัตโนมัติ

การระบุตำแหน่งของรถอัตโนมัติในงานวิจัยนี้ใช้เซ็นเซอร์ LiDAR Velodyne VLP-16 ร่วมกับ IMU (Inertial Measurement Unit) เพื่อให้ได้ข้อมูลที่แม่นยำในการกำหนดตำแหน่งของรถ LiDAR VLP-16 ทำหน้าที่สแกนพื้นที่รอบ ๆ เพื่อสร้างแผนที่สามมิติของสภาพแวดล้อมในขณะที่ IMU ให้ข้อมูลเกี่ยวกับการเคลื่อนที่ของรถ อัตราการหมุนและความเร่ง ข้อมูลจากทั้งสองเซ็นเซอร์จะถูกประมวลผลร่วมกันเพื่อให้ได้พิกัดที่แม่นยำของรถในสภาพแวดล้อมจริง



รูปที่ 3.19 ภาพการระบุตำแหน่งรถอัตโนมัติด้วย Lidar Velodyne VLP-16 และ IMU โดยใช้ Rviz

ภาพนี้ คือ ข้อมูลที่ได้จากการใช้งานเซ็นเซอร์ LiDAR Velodyne VLP-16 ร่วมกับ IMU (Inertial Measurement Unit) ซึ่งใช้ในการสร้างแผนที่และระบุตำแหน่งของรถอัตโนมัติแบบเรียลไทม์ การแสดงผลในภาพเกิดจากการใช้ซอฟต์แวร์ ROS (Robot Operating System) และเครื่องมือแสดงผล RViz เพื่อแสดงจุดข้อมูลที่ได้รับจากการสแกนของ LiDAR จุดที่เห็นในภาพเป็นการสแกนแบบสามมิติที่แสดงสิ่งกีดขวางหรือวัตถุในสภาพแวดล้อมรอบตัวรถ ข้อมูลเหล่านี้ถูกนำไปใช้ในระบบการระบุตำแหน่งของรถในพื้นที่จริง ซึ่งการทำงานร่วมกันระหว่าง LiDAR และ IMU ช่วยในการระบุทั้งการเคลื่อนไหวและการวางตำแหน่งของรถ โดย IMU จะให้ข้อมูลเกี่ยวกับการหมุนและการเร่งความเร็วของรถ ส่วน LiDAR จะสแกนสภาพแวดล้อมเพื่อสร้างแผนที่สามมิติ ข้อมูลทั้งสองส่วนจะถูกประมวลผลร่วมกันเพื่อให้ระบบระบุตำแหน่งของรถได้แม่นยำ

### 3.2.7 การแปลงพิกัดเพื่อประสานตำแหน่งของหุ่นยนต์และระบบกล้อง

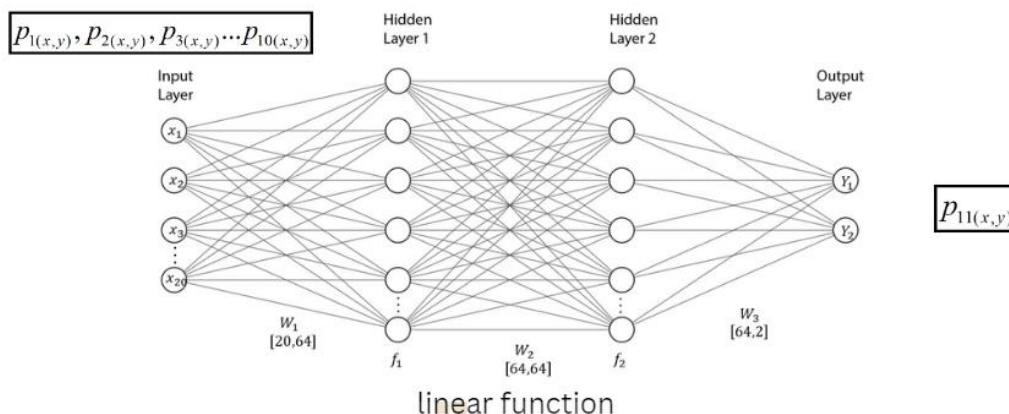
การแปลงพิกัด (Coordinate Transformation) เป็นการแปลงตำแหน่งระหว่างระบบพิกัดท้องถิ่น (Local Coordinate System) ของหุ่นยนต์และตำแหน่งที่ตรวจจับได้จากกล้องในระบบพิกัดโลก (Global Coordinate System) โดยกระบวนการนี้ใช้เมทริกซ์การแปลงเชิงโฮโมจีเนียสเพื่อให้การทำงานของหุ่นยนต์ในพื้นที่จริงสอดคล้องกับการตรวจจับวัตถุจากกล้องการแปลงพิกัดเพื่อประสานตำแหน่งของหุ่นยนต์และระบบกล้องมีความสำคัญเพื่อให้ระบบทั้งสองทำงาน

ร่วมกันได้อย่างถูกต้อง ขั้นตอนนี้ใช้การอ้างอิงตำแหน่งจากกล้องที่ตรวจจับวัตถุ (หัวข้อ 3.2.4) เป็นหลัก เนื่องจากพื้นที่นี้มีความคงที่ การแปลงพิกัดจะดำเนินการโดยนำตำแหน่งที่ได้จาก LiDAR และ IMU ของรถอัตโนมัติ (หัวข้อ 3.2.5) มาเปรียบเทียบและปรับเข้ากับพิกัดกริดของกล้อง ซึ่งช่วยให้ทั้งหุ่นยนต์และกล้องมีการทำงานสอดคล้องกันและระบุตำแหน่งในพื้นที่เดียวกันได้อย่างแม่นยำ การแปลงพิกัดนี้ทำโดยการนำข้อมูลพิกัดจาก LiDAR ที่เป็นพิกัดแบบ 3 มิติ มารวมกับข้อมูลจาก IMU เพื่อให้ระบบสามารถรับรู้ตำแหน่งการเคลื่อนไหวของหุ่นยนต์ในสภาพแวดล้อมจริง ข้อมูลจากทั้งสองเซ็นเซอร์นี้จะถูกนำไปเทียบกับพิกัดกริดที่ได้จากกล้องในพื้นที่ ทำให้หุ่นยนต์และระบบกล้องมีการรับรู้ตำแหน่งของวัตถุและพื้นที่ที่สอดคล้องกันกระบวนการแปลงพิกัดนี้ใช้การคำนวณผ่านการแปลงเชิงโฮโมจีเนียสซึ่งประกอบด้วย การ Scaling และ Translation

### 3.2.8 การทำนายตำแหน่ง

ในขั้นตอนการทำนายตำแหน่ง ระบบสร้างโมเดลการทำนายทั้งหมด 2 โมเดล เพื่อใช้ทำนายการเคลื่อนที่ของหุ่นยนต์และวัตถุที่ตรวจจับได้จากกล้อง โดยโมเดลแรก คือ "โมเดลการทำนายตำแหน่งจากกล้อง (Camera-based Position Prediction Model)" และโมเดลที่สอง คือ "โมเดลการทำนายตำแหน่งจากรถอัตโนมัติ (Autonomous Vehicle Position Prediction Model)" ทั้งสองโมเดลถูกสร้างขึ้นเพื่อคาดการณ์ตำแหน่งถัดไปของระบบจากข้อมูลการเคลื่อนที่ในปัจจุบัน โดยมีขั้นตอนในการทำนายโมเดลหลัก ๆ ดังนี้

การเก็บข้อมูลเพื่อสร้างโมเดลการทำนายตำแหน่งการเคลื่อนที่ของหุ่นยนต์ถูกเก็บข้อมูลจากทั้งระบบกล้องและรถอัตโนมัติ โดยแบ่งออกเป็น 11 ตำแหน่ง โดยใน โมเดลการทำนายตำแหน่งจากกล้อง ข้อมูลถูกเก็บจากการตรวจจับวัตถุผ่านกริดที่สร้างขึ้นในระบบกล้อง (อ้างอิงจากหัวข้อ 3.2.4) โดยใช้กล้องที่ติดตั้งในตำแหน่งคงที่เพื่อติดตามการเคลื่อนที่ของวัตถุข้อมูลตำแหน่งที่ได้จากกล้องจะถูกบันทึกเป็นค่า  $X$  และ  $Y$  ซึ่งระบุตำแหน่งในกริดที่กำหนดไว้โดยโมเดลการทำนายตำแหน่งจากรถอัตโนมัติ ข้อมูลตำแหน่งจะถูกเก็บจาก LiDAR และ IMU (หัวข้อ 3.2.5) โดยเซ็นเซอร์เหล่านี้สามารถติดตามตำแหน่งของรถได้ตลอดเวลาในขณะที่เคลื่อนที่ในพื้นที่ข้อมูลที่เก็บได้จากระบบนี้จะถูกใช้เพื่อสร้างโมเดลการทำนาย โดยเก็บข้อมูลจาก 11 ตำแหน่งเดียวกันกับที่ใช้ในการตรวจจับวัตถุจากกล้อง



รูปที่ 3.20 โครงสร้างเครือข่ายประสาทเทียม (ANN) สำหรับการทำนายตำแหน่ง

การสร้างโมเดลฝึกฝนในส่วนของโมเดลการทำนายตำแหน่งจากกล้อง (Camera-based Position Prediction Model) ข้อมูลที่ถูกใช้เป็นข้อมูลจากการตรวจจับวัตถุผ่านกริดของกล้อง ซึ่งเป็นพิกัดอ้างอิงในระบบนี้ ส่วนใน โมเดลการทำนายตำแหน่งจากรถอัตโนมัติ (Autonomous Vehicle Position Prediction Model) ข้อมูลที่ใช้จะมาจากการเก็บพิกัดโดย LiDAR และ IMU ซึ่งเป็นเซ็นเซอร์ที่ใช้สำหรับตรวจจับการเคลื่อนไหวและตำแหน่งของรถอัตโนมัติ โมเดลทั้งสองถูกออกแบบโดยใช้ Linear Function เป็นฟังก์ชันพื้นฐานในการประมวลผลข้อมูล โดยโมเดลถูกสร้างขึ้นด้วย 2 Hidden Layers ซึ่งเป็นชั้นที่มีบทบาทสำคัญในการเพิ่มประสิทธิภาพและความสามารถของโมเดลในการจับความสัมพันธ์เชิงลึกของข้อมูล Hidden Layer ทำหน้าที่เป็นตัวกรองในการประมวลผลข้อมูลจากชั้นหนึ่งไปยังอีกชั้นหนึ่ง โดยแต่ละ Hidden Layer ประกอบไปด้วย 64 หน่วยประมวลผล (Neurons) หน่วยประมวลผลเหล่านี้จะทำหน้าที่ประมวลผลข้อมูลและช่วยให้โมเดลสามารถเรียนรู้รูปแบบการเคลื่อนไหวและการเปลี่ยนแปลงของตำแหน่งที่ซับซ้อนขึ้น การใช้ Hidden Layer หลายชั้นในโมเดลช่วยให้การประมวลผลมีความแม่นยำมากขึ้น และทำให้โมเดลสามารถเรียนรู้ความแตกต่างเล็ก ๆ น้อย ๆ ระหว่างตำแหน่งต่าง ๆ ที่เก็บมาได้ดีขึ้น การมีหน่วยประมวลผลในแต่ละ Hidden Layer ถึง 64 หน่วย ทำให้การเรียนรู้ของโมเดลเกิดขึ้นอย่างรวดเร็วและมีความสามารถในการจับความสัมพันธ์ระหว่างตำแหน่งต่าง ๆ ในเส้นทางการเคลื่อนที่ได้มากขึ้น การฝึกฝนโมเดลในลักษณะนี้ทำให้ ANN สามารถปรับค่าพารามิเตอร์ต่าง ๆ ภายในโมเดลเพื่อลดข้อผิดพลาดในการทำนายตำแหน่งถัดไป การใช้โมเดล ANN ที่ประกอบด้วย Hidden Layers ที่ซับซ้อนยังช่วยเพิ่มความสามารถในการทำนายตำแหน่งได้แม่นยำยิ่งขึ้นเมื่อเผชิญกับข้อมูลใหม่ ที่ยังไม่เคยเห็นมาก่อน

การทดสอบผลลัพธ์หลังจากการสร้างโมเดลเสร็จสิ้น การทดสอบประสิทธิภาพของโมเดลจะถูกดำเนินการผ่านการวัดผลด้วยตัวชี้วัดหลายประเภท ตัวชี้วัดแรก คือ ค่า R2 (Coefficient of Determination) ซึ่งใช้ในการประเมินว่าโมเดลสามารถทำนายได้ใกล้เคียงกับค่าจริงมากน้อยเพียงใด หากค่า R2 ใกล้เคียงกับ 1 แสดงถึงความแม่นยำที่สูงของโมเดลค่า MAE (Mean Absolute Error) เป็นอีกหนึ่งตัวชี้วัดที่ใช้ในการวัดความคลาดเคลื่อนเฉลี่ยระหว่างค่าจริงและค่าทำนาย โดยค่าที่ได้นั้นจะสะท้อนถึงความแม่นยำของโมเดลในการทำนาย ยิ่งค่า MAE ต่ำ แสดงว่าโมเดลมีความคลาดเคลื่อนน้อย และทำนายได้ใกล้เคียงค่าจริง ใช้ค่า MSE (Mean Squared Error) ซึ่งเป็นการวัดค่าความคลาดเคลื่อนในรูปแบบกำลังสอง ค่า MSE ต่ำแสดงถึงความผิดพลาดของโมเดลที่น้อย ซึ่งเป็นตัวชี้วัดที่สำคัญในการทำนายค่าในอนาคตการทดสอบและประเมินผลเหล่านี้ช่วยให้สามารถประเมินประสิทธิภาพของโมเดลได้อย่างละเอียด และช่วยในการปรับปรุงโมเดลให้มีความแม่นยำมากขึ้นในการทำนายตำแหน่งถัดไปของหุ่นยนต์และระบบตรวจจับ

### 3.2.9 การประมวลผลและการวิเคราะห์เพื่อหาจุดที่มีความเสี่ยงต่อการชน

การประมวลผลและการวิเคราะห์ (Processing and Analysis) เพื่อตรวจหาจุดที่มีความเสี่ยงต่อการชนระหว่างวัตถุที่เคลื่อนที่ทั้งในหุ่นยนต์อัตโนมัติและยานพาหนะอื่น ๆ เป็นกระบวนการที่ต้องใช้ข้อมูลตำแหน่งของทั้งสองวัตถุในการคำนวณจุดที่มีความเสี่ยง การวิเคราะห์นี้ต้องคำนวณเส้นทางการเคลื่อนที่ที่วัตถุแต่ละชิ้นจะไปถึงจุดชนกัน

ในขั้นตอนนี้ ระบบจะนำผลการทำนายตำแหน่งจากทั้ง โมเดลการทำนายตำแหน่งจากกล้อง และ โมเดลการทำนายตำแหน่งจากรถอัตโนมัติ มาเปรียบเทียบกับในช่วงเวลาเดียวกัน โดยกระบวนการนี้จะวิเคราะห์ว่าตำแหน่งที่ทำนายจากโมเดลทั้งสองมีความใกล้เคียงกันมากเพียงใด และตรวจสอบว่ามีความเสี่ยงที่จะเกิดการชนหรือไม่การวิเคราะห์นี้ใช้การคำนวณระยะห่างระหว่างตำแหน่งที่ทำนายจากทั้งสองโมเดลโดยใช้ สูตรการคำนวณระยะทางระหว่างจุดสองจุด

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

โดยที่:

- $(x_1, y_1)$  คือตำแหน่งที่ทำนายจากโมเดลการทำนายตำแหน่งจากกล้อง
- $(x_2, y_2)$  คือตำแหน่งที่ทำนายจากโมเดลการทำนายตำแหน่งจากรถอัตโนมัติ

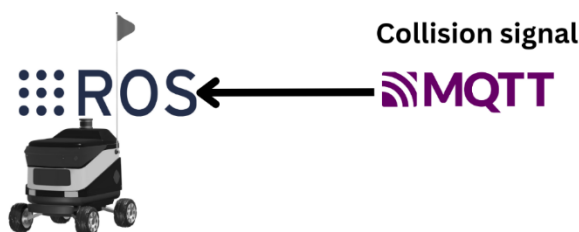
รูปที่ 3.21 สูตรการคำนวณระยะทางระหว่างสองจุดในระบบพิกัดคาร์ทีเซียน (Cartesian Coordinates)



ค่าระยะห่างที่ได้จากการคำนวณนี้ ( $d$ ) แสดงถึงความใกล้เคียงระหว่างตำแหน่งที่ทำนายจากโมเดลกล้องและโมเดลรถอัตโนมัติ หากระยะห่างนี้มีค่าน้อยกว่าเกณฑ์ขั้นต่ำที่กำหนด ซึ่งในกรณีนี้คือ 1 เมตร ระบบจะถือว่ามีความเสี่ยงที่จะเกิดการชนกระบวนการวิเคราะห์นี้มีความสำคัญเนื่องจากจะช่วยให้ระบบสามารถวางแผนและตัดสินใจในการเคลื่อนที่ได้อย่างปลอดภัย โดยการเปรียบเทียบตำแหน่งที่ทำนายจากโมเดลทั้งสองจะทำให้ระบบทราบล่วงหน้าถึงความเสี่ยงที่จะเกิดการชนและสามารถดำเนินการเพื่อหลีกเลี่ยงการชนได้อย่างรวดเร็วการวิเคราะห์ระยะห่างระหว่างตำแหน่งทั้งสองช่วยให้ระบบสามารถประเมินได้ว่าตำแหน่งของวัตถุและรถอัตโนมัติมีความเสี่ยงในการเคลื่อนที่เข้าสู่เส้นทางเดียวกันหรือไม่ ซึ่งเป็นปัจจัยสำคัญในการป้องกันการชนในพื้นที่การทดสอบจริง การใช้ระยะห่าง 1 เมตรเป็นเกณฑ์จะช่วยให้ระบบมีเวลาเพียงพอในการตัดสินใจและตอบสนองต่อสถานการณ์ที่เกิดขึ้นการคำนวณระยะห่างนี้จะทำในทุกเฟรมของการทำนายตำแหน่ง โดยจะทำการวิเคราะห์ในทุกช่วงเวลาเพื่อให้การคาดการณ์มีความแม่นยำระบบจะตรวจสอบระยะห่างอย่างต่อเนื่องเพื่อให้มั่นใจว่ารถอัตโนมัติและวัตถุในสภาพแวดล้อมจะไม่เข้าสู่จุดที่มีความเสี่ยงที่จะเกิดการชนการคำนวณอย่างต่อเนื่องนี้ยังช่วยให้สามารถวางแผนการเคลื่อนที่ของหุ่นยนต์ในเวลาจริง (real-time) การคำนวณระยะห่างและการเปรียบเทียบตำแหน่งที่ทำนายจากโมเดลทั้งสองเป็นกระบวนการสำคัญในการประมวลผลและวิเคราะห์ความเสี่ยงต่อการชน ระบบสามารถรับรู้ถึงสถานการณ์ที่อาจเกิดอุบัติเหตุได้ล่วงหน้า และดำเนินการแก้ไขเพื่อลดความเสี่ยงที่อาจเกิดขึ้น

### 3.2.10 การพัฒนาระบบการชะลอความเร็วเมื่อมีการทำนายว่าจะเกิดการชน

ระบบการชะลอความเร็ว (Speed Reduction System) เป็นการพัฒนาที่ทำงานร่วมกับการทำนายตำแหน่ง เมื่อระบบตรวจพบว่ามีความเป็นไปได้ที่จะเกิดการชน ระบบจะส่งสัญญาณไปยังหุ่นยนต์หรือยานพาหนะเพื่อชะลอความเร็วหรือหยุดการเคลื่อนที่ เพื่อป้องกันการเกิดอุบัติเหตุ



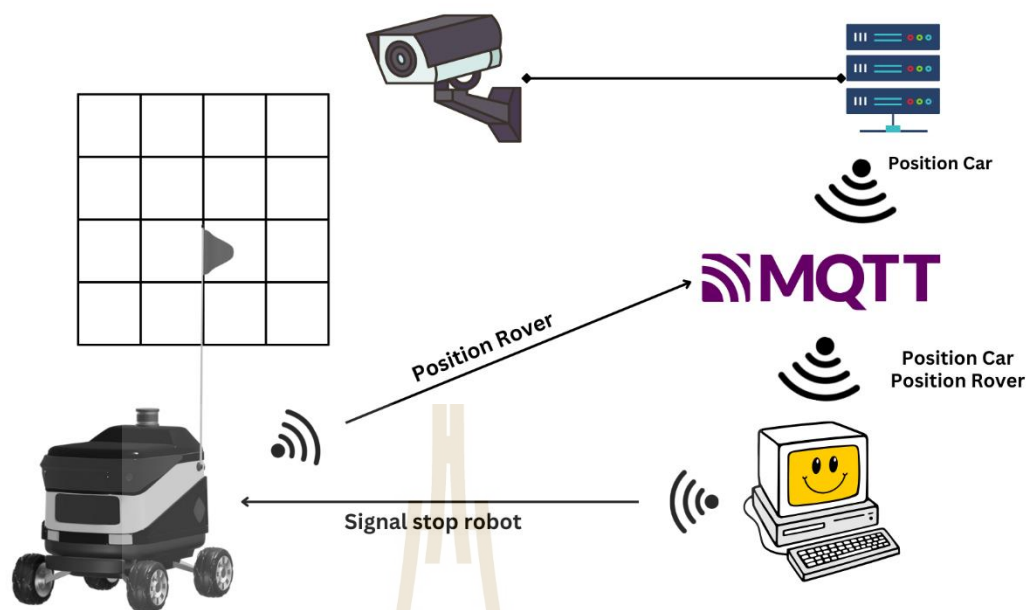
รูปที่ 3.22 การสื่อสารระหว่างระบบ MQTTและROS เพื่อการชะลอและหยุดหุ่นยนต์เมื่อมีความเสี่ยง



ระบบการชะลอความเร็วของหุ่นยนต์อัตโนมัติจะทำงานเมื่อหลังจาก หัวข้อ 3.2.8 การประมวลผลและการวิเคราะห์เพื่อหาจุดที่มีความเสี่ยงต่อการชน พบว่ามีโอกาสที่จะเกิดการชน ระบบจะส่งคำสั่งให้หุ่นยนต์อัตโนมัติทำการชะลอความเร็วและหยุด โดยกระบวนการนี้จะส่งคำสั่งผ่าน MQTT ไปยังหุ่นยนต์ โดยใช้ Topic "status/robot" หุ่นยนต์อัตโนมัติจะรับคำสั่งจาก MQTT และสร้างคำสั่งเพื่อส่งต่อไปยังระบบ ROS (Robot Operating System) ผ่านการใช้ `rospy.Publisher('/cmd_vel/freedom')` เมื่อระบบ ROS ได้รับคำสั่งนี้ หุ่นยนต์อัตโนมัติจะดำเนินการชะลอความเร็วตามคำสั่งที่ได้รับ และหยุดการเคลื่อนที่เพื่อลดความเสี่ยงในการชน กระบวนการนี้ถูกออกแบบมาเพื่อให้การตอบสนองของหุ่นยนต์เป็นไปอย่างรวดเร็วและมีประสิทธิภาพในกรณีที่ตรวจพบความเสี่ยงต่อการชนระบบ MQTT ทำหน้าที่เป็นตัวกลางในการสื่อสารคำสั่งระหว่างระบบการวิเคราะห์และหุ่นยนต์ ในขณะที่ ROS ทำหน้าที่ควบคุมการเคลื่อนที่ของหุ่นยนต์โดยตรง ทำให้สามารถชะลอและหยุดการเคลื่อนที่ได้อย่างรวดเร็วและปลอดภัย

### 3.3 ทดสอบระบบการชะลอความเร็วเมื่อเสี่ยงเกิดการชน

ในการทดสอบระบบการชะลอความเร็วเมื่อเสี่ยงเกิดการชนนี้จะทำการทดสอบระบบทั้งหมดที่พัฒนาขึ้นตามขั้นตอนทั้งหมดที่กล่าวมาข้างต้นเพื่อประเมินประสิทธิภาพในการตรวจจับวัตถุ ติดตามการเคลื่อนที่ ทำนายตำแหน่ง และป้องกันการชนของรถอัตโนมัติ โดยเริ่มจากการใช้กล้องและเซ็นเซอร์ LiDAR/IMU ในการระบุตำแหน่งวัตถุและรถอัตโนมัติ ข้อมูลที่ได้รับจะถูกใช้ในการสร้างโมเดลการทำนายตำแหน่งทั้งจากกล้องและรถอัตโนมัติ จากนั้นจะนำผลทำนายของทั้งสองโมเดลมาทำการวิเคราะห์ความเสี่ยง หากพบว่าตำแหน่งที่ทำนายมีโอกาสเกิดการชน ระบบจะดำเนินการส่งคำสั่งผ่าน MQTT โดยใช้ topic "status/robot" ไปยังหุ่นยนต์เพื่อสั่งการให้ชะลอความเร็วและหยุดการเคลื่อนที่ เพื่อหลีกเลี่ยงอุบัติเหตุการทดสอบนี้จะประเมินการทำงานของระบบในทุกขั้นตอน โดยเน้นไปที่ความแม่นยำในการตรวจจับวัตถุ การทำนายตำแหน่งที่ถูกต้อง และการประมวลผลเพื่อลดความเสี่ยงในการชน หุ่นยนต์จะถูกทดสอบในการเคลื่อนที่ในเส้นทางที่หลากหลาย เพื่อให้มั่นใจว่าระบบสามารถตอบสนองต่อสถานการณ์ที่มีความเสี่ยงได้อย่างทันที่การทดสอบระบบทั้งหมดในหัวข้อนี้เป็นการยืนยันว่าระบบสามารถทำงานได้อย่างปลอดภัยและแม่นยำเมื่อตรวจพบความเสี่ยงในการชน โดยหุ่นยนต์สามารถชะลอความเร็วและหยุดการเคลื่อนที่ได้ทันเวลา ลดความเสี่ยงในการเกิดอุบัติเหตุและเพิ่มความปลอดภัยในการทำงาน



รูปที่ 3.23 แสดงกระบวนการทำงานของระบบการสื่อสารตำแหน่งและสัญญาณการหยุดหุ่นยนต์

## บทที่ 4

### ผลการวิจัยและอภิปรายผล

#### 4.1 ผลการทดสอบระบบตรวจจับวัตถุ

ในการทดสอบระบบตรวจจับวัตถุ ระบบได้ใช้โมเดล YOLOv8 สำหรับการตรวจจับวัตถุที่ปรากฏในภาพ โดยทดสอบในสถานการณ์ที่มีสภาพแวดล้อมและมุมมองที่หลากหลาย ผลการทดสอบแสดงให้เห็นว่าระบบสามารถตรวจจับวัตถุได้อย่างแม่นยำและรวดเร็ว แม้ในสภาพแสงที่แตกต่างกันหรือเมื่อวัตถุอยู่ในมุมที่มีการเคลื่อนที่ ตัวชี้วัดที่ใช้ในการวัดประสิทธิภาพการตรวจจับได้แก่ อัตราความแม่นยำ (Precision) และ ความไวในการตรวจจับ (Recall) ซึ่งผลลัพธ์ที่ได้มีค่าใกล้เคียงเกณฑ์ที่ตั้งไว้ ทำให้ระบบมีประสิทธิภาพสูงในการตรวจจับวัตถุหลายประเภทในสภาพแวดล้อมจริง

การทดสอบนี้ยังรวมถึงการใช้ข้อมูลภาพที่ผ่านการติดป้ายกำกับจาก Roboflow เพื่อให้ระบบสามารถเรียนรู้รูปแบบของวัตถุต่าง ๆ ได้อย่างละเอียด เมื่อเปรียบเทียบกับ การทดสอบในหลายสถานการณ์ทั้งกลางวันและกลางคืน หรือมุมกล้องที่ต่างกัน ระบบยังคงมีความแม่นยำที่คงที่ แสดงถึงความสามารถในการทำงานในสถานการณ์ที่มีความท้าทายสูง

ตารางที่ 4.1 แสดงผลลัพธ์การฝึกฝนโมเดล

| epoch | train/box_loss | metrics/precision(B) | metrics/recall(B) | metrics/mAP50(B) | val/box_loss | lr/pg0    |
|-------|----------------|----------------------|-------------------|------------------|--------------|-----------|
| 1     | 1.319          | 0.91569              | 0.83011           | 0.87654          | 1.2887       | 0.0006618 |
| 2     | 1.317          | 0.96339              | 0.93631           | 0.96856          | 1.2799       | 0.0013219 |
| 3     | 1.2939         | 0.96642              | 0.96178           | 0.98315          | 1.3183       | 0.0019754 |
| 4     | 1.2717         | 0.93656              | 0.89172           | 0.92258          | 1.2849       | 0.0019703 |
| 5     | 1.2386         | 0.97801              | 0.93631           | 0.97481          | 1.3151       | 0.0019703 |
| 6     | 1.23           | 0.99215              | 0.98089           | 0.98835          | 1.2552       | 0.0019604 |
| 7     | 1.2282         | 0.98616              | 0.98726           | 0.98745          | 1.2968       | 0.0019505 |
| 8     | 1.1927         | 0.9935               | 0.97373           | 0.99195          | 1.2664       | 0.0019406 |
| 9     | 1.1985         | 0.98324              | 0.97452           | 0.98605          | 1.362        | 0.0019307 |

ตารางที่ 4.1 แสดงผลลัพธ์การฝึกฝนโมเดล (ต่อ)

| epoch | train/box_loss | metrics/precision(B) | metrics/recall(B) | metrics/mAP50(B) | val/box_loss | lr/pg0    |
|-------|----------------|----------------------|-------------------|------------------|--------------|-----------|
| 10    | 1.1857         | 0.9935               | 0.97426           | 0.98757          | 1.2731       | 0.0019208 |
| 11    | 1.1635         | 0.98677              | 0.96815           | 0.98653          | 1.2564       | 0.0019109 |
| 12    | 1.1799         | 0.99357              | 0.9849            | 0.98682          | 1.2666       | 0.001901  |
| 13    | 1.174          | 0.9935               | 0.97406           | 0.99003          | 1.2598       | 0.0018911 |
| 14    | 1.1513         | 0.99185              | 0.98089           | 0.98924          | 1.1969       | 0.0018812 |
| 15    | 1.1494         | 0.98717              | 0.98047           | 0.98742          | 1.2563       | 0.0018713 |
| 16    | 1.1415         | 0.98668              | 0.97452           | 0.98788          | 1.2859       | 0.0018614 |
| 17    | 1.1421         | 0.98076              | 0.97388           | 0.9833           | 1.2379       | 0.0018515 |
| 18    | 1.1202         | 0.99345              | 0.96672           | 0.98953          | 1.2541       | 0.0018416 |
| 19    | 1.1175         | 0.98709              | 0.97434           | 0.98653          | 1.2586       | 0.0018317 |
| 20    | 1.1177         | 0.99255              | 0.97452           | 0.98925          | 1.2677       | 0.0018218 |
| 21    | 1.1136         | 0.99355              | 0.98046           | 0.98977          | 1.2145       | 0.0018119 |
| 22    | 1.0837         | 0.99354              | 0.97974           | 0.98663          | 1.2383       | 0.001802  |
| 23    | 1.1043         | 0.98354              | 0.97452           | 0.99035          | 1.3162       | 0.0017921 |
| 24    | 1.098          | 0.98711              | 0.98089           | 0.98932          | 1.3253       | 0.0017822 |
| 25    | 1.0833         | 0.98717              | 0.97991           | 0.99132          | 1.2917       | 0.0017723 |
| 26    | 1.0713         | 0.99187              | 0.97452           | 0.99288          | 1.2414       | 0.0017624 |
| 27    | 1.0735         | 0.98675              | 0.98089           | 0.99077          | 1.2808       | 0.0017525 |
| 28    | 1.0642         | 0.99276              | 0.97452           | 0.98681          | 1.2395       | 0.0017426 |
| 29    | 1.0596         | 0.98718              | 0.98076           | 0.9925           | 1.2607       | 0.0017327 |
| 30    | 1.0616         | 0.98725              | 0.98664           | 0.99387          | 1.3094       | 0.0017228 |
| 31    | 1.0749         | 0.97734              | 0.97452           | 0.99343          | 1.3317       | 0.0017129 |
| 32    | 1.0709         | 0.99238              | 0.97452           | 0.99161          | 1.2471       | 0.001703  |
| 33    | 1.0583         | 0.98675              | 0.98726           | 0.99326          | 1.2155       | 0.0016931 |
| 34    | 1.0224         | 0.99029              | 0.98089           | 0.98905          | 1.2314       | 0.0016832 |

ตารางที่ 4.1 แสดงผลลัพธ์การฝึกฝนโมเดล (ต่อ)

| epoch | train/box_loss | metrics/precision(B) | metrics/recall(B) | metrics/mAP50(B) | val/box_loss | lr/pg0    |
|-------|----------------|----------------------|-------------------|------------------|--------------|-----------|
| 35    | 1.0408         | 0.9935               | 0.97355           | 0.99304          | 1.2398       | 0.0016733 |
| 36    | 1.0721         | 0.98622              | 0.97452           | 0.99357          | 1.2454       | 0.0016634 |
| 37    | 1.0273         | 0.9863               | 0.98089           | 0.99423          | 1.3111       | 0.0016535 |
| 38    | 1.0301         | 0.98088              | 0.98033           | 0.98935          | 1.2469       | 0.0016436 |
| 39    | 1.0143         | 0.99282              | 0.97452           | 0.98795          | 1.2478       | 0.0016337 |
| 40    | 1.0217         | 0.9866               | 0.97452           | 0.99077          | 1.2668       | 0.0016238 |
| 41    | 1.0144         | 0.99264              | 0.98089           | 0.98937          | 1.3391       | 0.0016139 |
| 42    | 1.0174         | 0.99294              | 0.98089           | 0.98794          | 1.2244       | 0.001604  |
| 43    | 1.0091         | 0.97418              | 0.98726           | 0.98946          | 1.337        | 0.0015941 |
| 44    | 1.004          | 0.99354              | 0.97953           | 0.99412          | 1.2271       | 0.0015842 |
| 45    | 0.98576        | 0.99225              | 0.96815           | 0.99328          | 1.3274       | 0.0015743 |
| 46    | 1.0031         | 0.99236              | 0.96815           | 0.99097          | 1.2252       | 0.0015644 |
| 47    | 0.97391        | 0.99349              | 0.97159           | 0.99052          | 1.3054       | 0.0015545 |
| 48    | 0.98284        | 0.98615              | 0.97452           | 0.98866          | 1.2706       | 0.0015446 |
| 49    | 0.97098        | 0.9927               | 0.98089           | 0.99399          | 1.2487       | 0.0015347 |
| 50    | 0.97528        | 1                    | 0.98581           | 0.99494          | 1.2456       | 0.0015248 |
| 51    | 0.98518        | 0.99277              | 0.98089           | 0.98962          | 1.2465       | 0.0015149 |
| 52    | 0.9541         | 0.98658              | 0.98726           | 0.99052          | 1.229        | 0.001505  |
| 53    | 0.96347        | 0.98966              | 0.98089           | 0.98655          | 1.3122       | 0.0014951 |
| 54    | 0.95629        | 0.99225              | 0.97452           | 0.98702          | 1.2445       | 0.0014852 |
| 55    | 0.95419        | 0.98057              | 0.98726           | 0.99147          | 1.217        | 0.0014753 |
| 56    | 0.93978        | 0.9855               | 0.97452           | 0.98769          | 1.2866       | 0.0014654 |
| 57    | 0.94267        | 0.98029              | 0.98089           | 0.98816          | 1.3164       | 0.0014555 |
| 58    | 0.93093        | 0.993                | 0.98089           | 0.99383          | 1.3187       | 0.0014456 |
| 59    | 0.94126        | 0.99281              | 0.97452           | 0.98866          | 1.2538       | 0.0014357 |



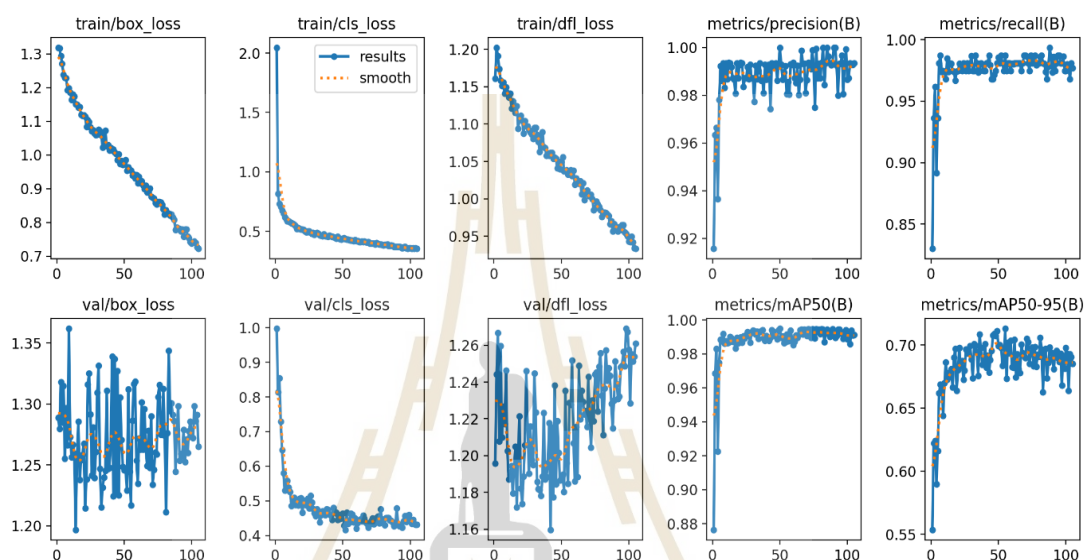
ตารางที่ 4.1 แสดงผลลัพธ์การฝึกฝนโมเดล (ต่อ)

| epoch | train/box_loss | metrics/precision(B) | metrics/recall(B) | metrics/mAP50(B) | val/box_loss | lr/pg0    |
|-------|----------------|----------------------|-------------------|------------------|--------------|-----------|
| 60    | 0.91677        | 0.99355              | 0.98057           | 0.98538          | 1.2349       | 0.0014258 |
| 61    | 0.93569        | 0.98717              | 0.98003           | 0.98735          | 1.2717       | 0.0014159 |
| 62    | 0.91727        | 0.98512              | 0.98089           | 0.99112          | 1.2628       | 0.001406  |
| 63    | 0.92211        | 0.99237              | 0.98089           | 0.98944          | 1.3017       | 0.0013961 |
| 64    | 0.91066        | 0.98007              | 0.98089           | 0.99113          | 1.2627       | 0.0013862 |
| 65    | 0.89975        | 0.98661              | 0.98089           | 0.99408          | 1.2959       | 0.0013763 |
| 66    | 0.91032        | 0.99875              | 0.98089           | 0.9944           | 1.2627       | 0.0013664 |
| 67    | 0.88997        | 0.99319              | 0.97452           | 0.9916           | 1.2493       | 0.0013565 |
| 68    | 0.89051        | 0.99323              | 0.98089           | 0.9945           | 1.2784       | 0.0013466 |
| 69    | 0.90127        | 0.98084              | 0.98726           | 0.99424          | 1.2476       | 0.0013367 |
| 70    | 0.87766        | 0.98632              | 0.98726           | 0.99063          | 1.2899       | 0.0013268 |
| 71    | 0.8748         | 0.99358              | 0.98646           | 0.99456          | 1.2339       | 0.0013169 |
| 72    | 0.8719         | 0.99161              | 0.98089           | 0.99421          | 1.2462       | 0.001307  |
| 73    | 0.85867        | 0.99104              | 0.97452           | 0.98967          | 1.2603       | 0.0012971 |
| 74    | 0.85361        | 0.99898              | 0.98726           | 0.99475          | 1.2589       | 0.0012872 |
| 75    | 0.86151        | 0.99225              | 0.98089           | 0.99163          | 1.3058       | 0.0012773 |
| 76    | 0.86183        | 0.97483              | 0.98655           | 0.98941          | 1.3159       | 0.0012674 |
| 77    | 0.86021        | 0.99955              | 0.98726           | 0.99475          | 1.2751       | 0.0012575 |
| 78    | 0.85303        | 0.98716              | 0.9793            | 0.99304          | 1.2791       | 0.0012476 |
| 79    | 0.84031        | 0.99318              | 0.98089           | 0.9945           | 1.3127       | 0.0012377 |
| 80    | 0.82455        | 0.98713              | 0.98089           | 0.9918           | 1.2744       | 0.0012278 |
| 81    | 0.83713        | 0.99358              | 0.98583           | 0.99463          | 1.2114       | 0.0012179 |
| 82    | 0.82737        | 1                    | 0.98705           | 0.99434          | 1.2762       | 0.001208  |
| 83    | 0.82692        | 0.98443              | 0.98089           | 0.98709          | 1.344        | 0.0011981 |
| 84    | 0.82155        | 0.99354              | 0.98037           | 0.99218          | 1.2946       | 0.0011882 |

ตารางที่ 4.1 แสดงผลลัพธ์การฝึกฝนโมเดล (ต่อ)

| epoch | train/box_loss | metrics/precision(B) | metrics/recall(B) | metrics/mAP50(B) | val/box_loss | lr/pg0    |
|-------|----------------|----------------------|-------------------|------------------|--------------|-----------|
| 85    | 0.82407        | 1                    | 0.97376           | 0.9945           | 1.3008       | 0.0011783 |
| 86    | 0.81507        | 0.99625              | 0.96815           | 0.99218          | 1.2914       | 0.0011684 |
| 87    | 0.80884        | 0.99353              | 0.97813           | 0.9901           | 1.286        | 0.0011585 |
| 88    | 0.77842        | 0.99272              | 0.99363           | 0.99366          | 1.2448       | 0.0011486 |
| 89    | 0.78614        | 0.99358              | 0.98616           | 0.99475          | 1.2788       | 0.0011387 |
| 90    | 0.77942        | 1                    | 0.97363           | 0.99225          | 1.2985       | 0.0011288 |
| 91    | 0.78084        | 1                    | 0.98612           | 0.98954          | 1.2537       | 0.0011189 |
| 92    | 0.77445        | 0.99247              | 0.97452           | 0.99064          | 1.2593       | 0.001109  |
| 93    | 0.77939        | 0.97992              | 0.98089           | 0.99425          | 1.2573       | 0.0010991 |
| 94    | 0.75908        | 0.99322              | 0.98726           | 0.99451          | 1.2791       | 0.0010892 |
| 95    | 0.7803         | 0.99324              | 0.98726           | 0.99332          | 1.2524       | 0.0010793 |
| 96    | 0.77941        | 0.99677              | 0.98089           | 0.99187          | 1.2712       | 0.0010694 |
| 97    | 0.75843        | 0.98056              | 0.98089           | 0.98913          | 1.2601       | 0.0010595 |
| 98    | 0.75347        | 0.99358              | 0.98634           | 0.99075          | 1.2778       | 0.0010496 |
| 99    | 0.74232        | 0.9993               | 0.97452           | 0.99411          | 1.2857       | 0.0010397 |
| 100   | 0.73776        | 0.98651              | 0.97452           | 0.98885          | 1.2731       | 0.0010298 |
| 101   | 0.74122        | 0.98717              | 0.98027           | 0.99166          | 1.2724       | 0.0010199 |
| 102   | 0.7445         | 0.99353              | 0.9774            | 0.98585          | 1.2985       | 0.00101   |
| 103   | 0.74012        | 0.99275              | 0.96815           | 0.99383          | 1.2861       | 0.0010001 |
| 104   | 0.7267         | 0.99281              | 0.98089           | 0.99092          | 1.2911       | 0.0009902 |
| 105   | 0.72248        | 0.99352              | 0.97708           | 0.99127          | 1.2651       | 0.0009803 |

ผลการทดสอบระบบตรวจจับวัตถุจากโมเดล YOLOv8 แสดงให้เห็นถึงความสำเร็จในขั้นตอนการฝึกโมเดลสำหรับการตรวจจับวัตถุอย่างชัดเจน โดยการประเมินผลได้ใช้เกณฑ์หลายประการเพื่อวัดความแม่นยำและประสิทธิภาพของโมเดลที่สร้างขึ้น ดังนั้น ในการอธิบายผลนี้จะเน้นไปที่การวิเคราะห์ตัวชี้วัดหลักต่าง ๆ ที่สะท้อนถึงประสิทธิภาพของโมเดลและการเรียนรู้ของโมเดลในการฝึกตรวจจับวัตถุในภาพ



รูปที่ 4.1 กราฟจากผลลัพธ์การฝึกการทดสอบโมเดล Yolo

ในช่วงเริ่มต้นการฝึก โมเดลยังไม่ได้มีการปรับแต่งและมีค่า Box Loss (train/box\_loss) สูงขึ้น ซึ่งหมายถึงโมเดลยังไม่สามารถตรวจจับ Bounding Box ของวัตถุได้อย่างแม่นยำ เมื่อจำนวน epoch เพิ่มขึ้น ค่าการสูญเสีย (Box Loss) ของโมเดลลดลงอย่างสม่ำเสมอ แสดงถึงการที่โมเดลเรียนรู้จากข้อมูลการฝึกอย่างมีประสิทธิภาพและสามารถปรับปรุงการทำนายตำแหน่งของวัตถุได้มากขึ้น ในขั้นตอนการฝึกช่วงท้าย ๆ ค่า Box Loss ลดลงจนเข้าสู่ระดับที่ต่ำมาก ซึ่งเป็นสัญญาณว่าความแม่นยำของการตรวจจับ Bounding Box มีแนวโน้มที่ดีขึ้นเรื่อย ๆ

เมื่อพิจารณาจากตัวชี้วัด Precision และ Recall พบว่าค่า Precision ในช่วงต้นของการฝึกมีค่าอยู่ในระดับปานกลาง และค่อย ๆ เพิ่มขึ้นเรื่อย ๆ ตามจำนวน epoch ซึ่งแสดงถึงการที่โมเดลมีความแม่นยำในการตรวจจับวัตถุที่มากขึ้น ในช่วงแรกโมเดลอาจจะตรวจจับผิดพลาดบ้าง แต่เมื่อได้รับข้อมูลที่มากขึ้นผ่านการฝึก โมเดลสามารถระบุวัตถุได้ถูกต้องมากขึ้นเช่นกัน ขณะที่ Recall เริ่มต้นมีค่าอยู่ในระดับต่ำกว่า Precision เล็กน้อย แต่ก็มีเพิ่มขึ้นอย่างสม่ำเสมอ แสดงถึงการที่โมเดลสามารถตรวจจับวัตถุทั้งหมดในภาพได้ดีขึ้นตามจำนวนรอบการฝึกที่มากขึ้น ทั้งสองตัวชี้วัดนี้เป็นตัว

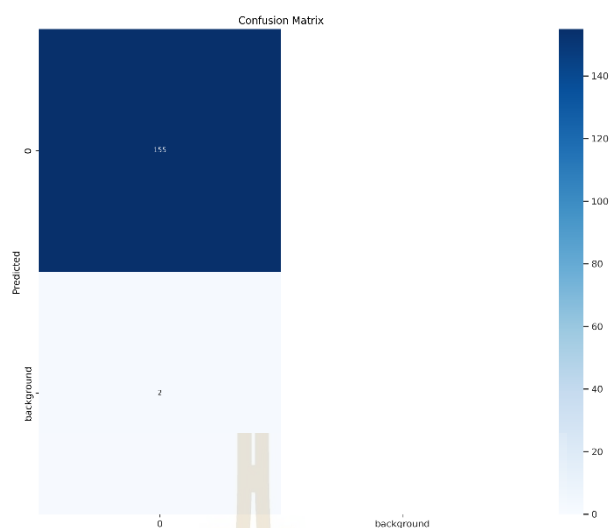
บ่งชี้ว่าโมเดลมีความแม่นยำและสามารถตรวจจับวัตถุในภาพได้อย่างมีประสิทธิภาพมากยิ่งขึ้นในทุก ๆ รอบการฝึก

ค่า mAP50 (Mean Average Precision ที่ค่า IOU เท่ากับ 50%) ซึ่งเป็นตัววัดความแม่นยำในการทำนายของโมเดล ในช่วงแรกมีค่าเริ่มต้นไม่สูงนัก แต่เมื่อผ่านการฝึกไปเรื่อย ๆ ค่า mAP50 เพิ่มขึ้นอย่างต่อเนื่อง ซึ่งในตอนท้ายมีค่าที่สูงมากใกล้เคียงกับ 1.0 นั้นแสดงถึงการที่โมเดลสามารถตรวจจับและทำนายวัตถุได้อย่างมีประสิทธิภาพในระดับที่สูง

ในการประเมินผลด้วยชุดข้อมูลทดสอบ (Validation) ผ่านค่า Val Box Loss พบว่าค่า Loss นี้ลดลงและคงตัวในระดับต่ำ ซึ่งหมายถึงการที่โมเดลสามารถตรวจจับวัตถุได้ดีในข้อมูลที่ไม่เคยเห็นมาก่อน การที่ค่า Val Box Loss มีแนวโน้มที่ต่ำลงและคงที่ในระดับต่ำตลอดการทดสอบ บ่งบอกถึงการที่โมเดลสามารถนำความรู้จากชุดข้อมูลฝึกไปประยุกต์ใช้กับชุดข้อมูลใหม่ได้อย่างมีประสิทธิภาพ ซึ่งสะท้อนถึงการที่โมเดลสามารถทำงานในสภาพแวดล้อมที่หลากหลายและไม่ขึ้นอยู่กับข้อมูลเพียงอย่างเดียว

สุดท้ายนี้ ค่า Learning Rate ( $lr/pg0$ ) มีการตั้งค่าให้เหมาะสมตลอดช่วงการฝึก โดยค่า learning rate อยู่ในระดับต่ำแต่คงที่ ซึ่งช่วยให้โมเดลสามารถอัปเดตพารามิเตอร์ได้อย่างมีประสิทธิภาพและค่อย ๆ ปรับตัวไปในทิศทางที่ดีขึ้น การควบคุมค่า Learning Rate อย่างเหมาะสมนี้ช่วยให้โมเดลสามารถฝึกได้อย่างมีประสิทธิภาพและหลีกเลี่ยงปัญหาการฝึกที่ช้าเกินไปหรือเร็วเกินไป

สรุปผลจากการทดสอบและประเมินค่าโมเดล YOLOv8 แสดงให้เห็นว่าโมเดลสามารถตรวจจับวัตถุได้อย่างแม่นยำและมีความสามารถในการเรียนรู้จากข้อมูลฝึกได้อย่างมีประสิทธิภาพ ตัวชี้วัดทั้งหมดบ่งบอกถึงการที่โมเดลสามารถทำนายตำแหน่งของวัตถุได้ดีในทุกมิติ ทั้งในด้านความแม่นยำ (Precision) ความไวในการตรวจจับ (Recall) และประสิทธิภาพการทำนายโดยรวม (mAP50) ซึ่งสามารถนำโมเดลนี้ไปใช้งานในระบบตรวจจับวัตถุได้อย่างมั่นใจ



รูปที่ 4.2 Confusion Matrix Unnormalized แสดงผลการทดสอบโมเดล



รูปที่ 4.3 Confusion Matrix Normalized แสดงผลการทดสอบโมเดล

ผลการทดสอบระบบตรวจจับวัตถุสามารถสรุปได้จากการวิเคราะห์ภาพผลลัพธ์จาก Confusion Matrix ทั้งในรูปแบบ Normalized และ Unnormalized ซึ่งให้ข้อมูลสำคัญเกี่ยวกับ

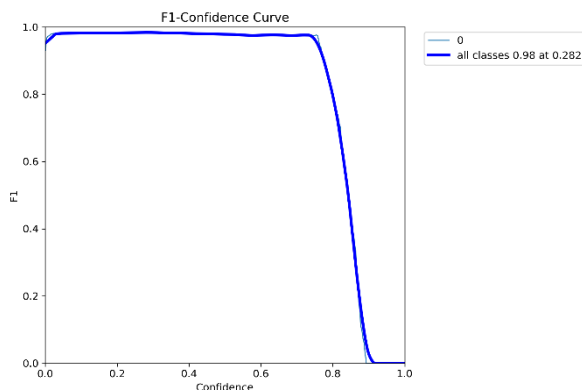
ประสิทธิภาพของระบบในแง่ของความแม่นยำในการตรวจจับวัตถุที่เป็นเป้าหมาย รวมถึงการวัดผลของการตรวจจับที่ผิดพลาด

ความแม่นยำในการตรวจจับนั้นจากผลการทดสอบที่ปรากฏใน Confusion Matrix แบบ Normalized ระบบตรวจจับวัตถุมีความแม่นยำสูงมาก โดยสามารถตรวจจับวัตถุที่สนใจได้ถูกต้องถึง 99% ซึ่งบ่งชี้ว่าในทุก ๆ 100 ครั้งของการตรวจจับ ระบบสามารถระบุวัตถุที่ถูกต้องได้ 99 ครั้ง ข้อมูลนี้แสดงถึงความสามารถในการประมวลผลและการทำนายที่มีประสิทธิภาพสูงของระบบ ในการทดสอบจริง ระบบแสดงการทำนายที่ถูกต้องเกือบทั้งหมดของวัตถุเป้าหมาย ซึ่งส่งผลให้ความคลาดเคลื่อนในการทำนายน้อยมาก

ความผิดพลาดในการตรวจจับจาก Confusion Matrix ในแบบ Unnormalized ซึ่งแสดงค่าจริงของจำนวนครั้งที่ระบบสามารถทำนายวัตถุได้อย่างถูกต้องและผิดพลาด ระบบได้ทำการตรวจจับวัตถุถูกต้องทั้งหมด 155 ครั้ง จากจำนวนทั้งหมด 157 ครั้ง ขณะที่มีการทำนายผิดพลาดเพียง 2 ครั้ง ซึ่งหมายความว่าระบบมี False Positive เพียงเล็กน้อย โดย False Positive ในที่นี้หมายถึงกรณีที่ระบบตรวจจับสิ่งที่ไม่ใช่วัตถุเป้าหมาย แต่กลับระบุว่าเป็นวัตถุเป้าหมาย ความผิดพลาดในลักษณะนี้เกิดขึ้นในสัดส่วนน้อยมากเพียง 1% ซึ่งถือเป็นความผิดพลาดที่ยอมรับได้ในบริบทของระบบการตรวจจับวัตถุ

ประสิทธิภาพโดยรวมการทดสอบระบบตรวจจับวัตถุแสดงให้เห็นว่าระบบมีประสิทธิภาพในการทำงานสูงมาก ทั้งในแง่ของความแม่นยำในการทำนายตำแหน่งวัตถุ และความสามารถในการลดข้อผิดพลาดในการทำนาย ความผิดพลาดที่เกิดขึ้นมีจำนวนที่น้อยและไม่ส่งผลกระทบต่อความแม่นยำโดยรวมของระบบ ซึ่งแสดงให้เห็นว่าระบบนี้มีความสามารถในการระบุวัตถุที่แม่นยำในระดับสูง และสามารถนำไปประยุกต์ใช้งานได้ในบริบทของการตรวจจับวัตถุในสภาพแวดล้อมจริง จากการสังเกตผลการทดสอบจะเห็นว่าความแม่นยำในการตรวจจับที่สูงนั้นสัมพันธ์กับการลดทอนความคลาดเคลื่อน ซึ่งถือว่าเป็นปัจจัยสำคัญที่ทำให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพในสถานการณ์จริง ทั้งนี้ ระบบมีความเสถียรและสามารถตอบสนองต่อความต้องการของการใช้งานในด้านการตรวจจับวัตถุได้ดี

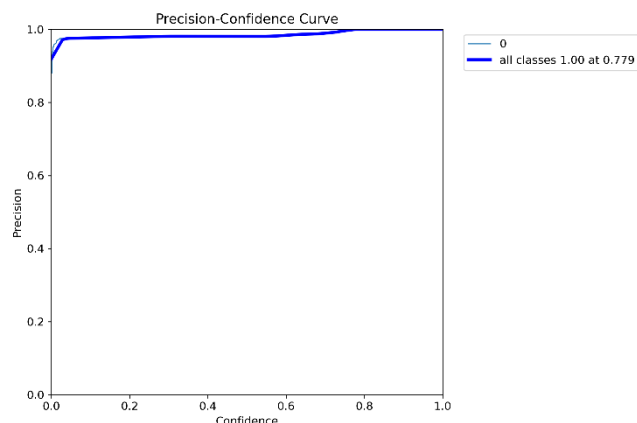




รูปที่ 4.4 กราฟ F1-Confidence Curve ของโมเดลการตรวจจับวัตถุ

จากกราฟที่ได้ จากรูปที่ 4.4 ค่า F1-Score แสดงให้เห็นถึงความสมดุลระหว่างความแม่นยำและความครอบคลุมในการตรวจจับวัตถุ โดยเฉพาะอย่างยิ่งในระดับ Confidence ที่ต่ำและปานกลาง ค่า F1-Score มีค่าคงที่และใกล้เคียงกับ 1 ซึ่งหมายถึงประสิทธิภาพในการตรวจจับวัตถุที่ดีมาก โมเดลมีความสามารถในการทำนายวัตถุได้อย่างแม่นยำและครอบคลุมในช่วงความมั่นใจที่ระดับไม่สูงมากเมื่อพิจารณาช่วง Confidence ที่เพิ่มขึ้น (ประมาณ 0.75-0.8) ค่า F1-Score ยังคงมีค่าสูง ซึ่งบ่งชี้ว่าโมเดลยังคงสามารถรักษาประสิทธิภาพในการตรวจจับได้ดีในช่วงนี้ เมื่อ Confidence เกิน 0.8 ค่า F1-Score ลดลงอย่างรวดเร็ว แสดงให้เห็นว่าในระดับความมั่นใจที่สูงมาก โมเดลอาจเริ่มทำงานผิดพลาดหรือตรวจจับวัตถุผิดพลาดได้บ่อยขึ้น ซึ่งเป็นสัญญาณว่าระบบมีแนวโน้มที่จะทำนายผิดพลาดเมื่อความมั่นใจเกินไปการลดลงของค่า F1-Score ที่ Confidence สูงสะท้อนถึงการทำงานที่อาจไม่สมบูรณ์เมื่อระบบมีการกำหนดค่า Confidence ที่สูงเกินไป แม้ว่าระบบจะแสดงความมั่นใจในผลลัพธ์มาก แต่ก็อาจแลกมากับการขาดความแม่นยำในการตรวจจับ หรือการที่วัตถุบางชนิดไม่ถูกตรวจจับอย่างถูกต้อง ซึ่งแสดงให้เห็นถึงความสำคัญของการกำหนดระดับ Confidence ที่เหมาะสมเพื่อให้ระบบมีความสมดุลในการทำงานระหว่างความแม่นยำและการครอบคลุม

กราฟนี้แสดงให้เห็นว่าโมเดลสามารถทำงานได้ดีในช่วง Confidence ต่ำถึงปานกลาง (ประมาณ 0-0.75) ซึ่งในช่วงนี้ค่า F1-Score มีค่าคงที่และใกล้เคียงกับ 1 เป็นช่วงที่โมเดลมีความสมดุลระหว่าง Precision และ Recall สูงที่สุด เมื่อ Confidence เพิ่มขึ้นเกินระดับ 0.8 ค่า F1-Score เริ่มลดลงอย่างรวดเร็ว ซึ่งหมายถึงการเสี่ยงต่อความผิดพลาดมากขึ้นในระดับความมั่นใจที่สูง ดังนั้น การเลือกค่า Confidence ที่เหมาะสมจึงเป็นสิ่งสำคัญในการนำโมเดลไปใช้งาน เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดและลดโอกาสในการทำนายผิดพลาด

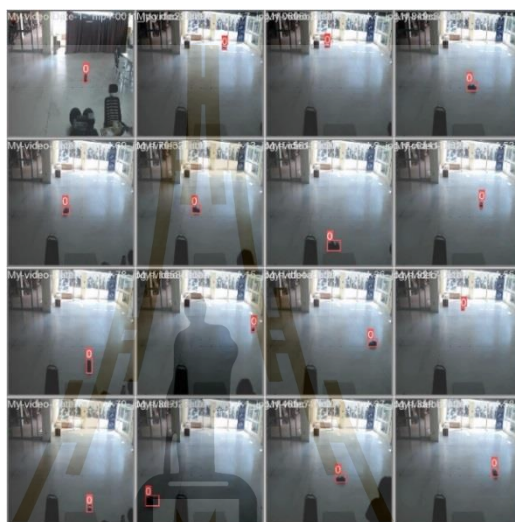


รูปที่ 4.5 กราฟ Precision-Confidence Curve ของโมเดลการตรวจจับวัตถุ

กราฟ Precision-Confidence Curve ที่ปรากฏในภาพแสดงให้เห็นถึงความสัมพันธ์ระหว่างค่า Precision (ความแม่นยำ) กับระดับความมั่นใจ (Confidence) ของโมเดลในการทำนายผลลัพธ์จากการตรวจจับวัตถุ ค่า Precision นั้นเป็นตัวชี้วัดที่บ่งบอกถึงความสามารถของโมเดลในการทำนายผลลัพธ์ที่ถูกต้องในแต่ละคลาสของข้อมูล และสะท้อนถึงความถูกต้องในการระบุวัตถุที่แท้จริงเมื่อโมเดลทำการทำนาย

จากกราฟจะเห็นได้ว่าในระดับ Confidence ที่ต่ำถึงปานกลาง ค่า Precision มีค่าสูงมาก และคงที่ใกล้เคียงกับ 1 ในช่วงเริ่มต้น ซึ่งหมายความว่าเมื่อโมเดลทำนายผลลัพธ์ในระดับ Confidence ต่ำ โมเดลยังคงสามารถทำการทำนายได้อย่างแม่นยำ กล่าวคือ ค่าความผิดพลาดในผลลัพธ์ที่โมเดลทำนายต่ำมาก ซึ่งแสดงให้เห็นว่าในช่วงเริ่มต้นของการทำงาน โมเดลมีความสามารถในการตรวจจับวัตถุได้อย่างถูกต้องและมีประสิทธิภาพเมื่อพิจารณาช่วง Confidence สูง ๆ (0.8 ขึ้นไป) ค่า Precision ยังคงคงที่ในระดับสูง ซึ่งแสดงให้เห็นว่าแม้เมื่อ Confidence สูงขึ้น โมเดลก็ยังคงรักษาความแม่นยำในการตรวจจับได้ดี การที่ค่า Precision ยังคงมีค่าสูงในช่วง Confidence ระดับสูงเป็นสัญญาณบ่งชี้ว่าโมเดลมีความสามารถในการระบุวัตถุที่ถูกต้องได้สูงตลอดช่วง Confidence ที่เพิ่มขึ้น ซึ่งแสดงถึงความเสถียรของโมเดลในการทำงาน โดยโมเดลนี้มี Precision สูงสุดที่ค่า 1 เมื่อ Confidence อยู่ที่ประมาณ 0.779 ซึ่งแสดงถึงประสิทธิภาพที่ดีมากของโมเดลในระดับ Confidence ที่ใกล้เคียงกับ 1 ค่า Precision ยังคงไม่ลดลง ซึ่งหมายความว่าค่า Confidence ในระดับสูงทำให้โมเดลมีโอกาสน้อยมากในการทำนายผลลัพธ์ที่ผิดพลาด โมเดลนี้สามารถทำนายได้อย่างแม่นยำแม้ว่าระดับความมั่นใจในการทำนายจะสูงมากก็ตาม ซึ่งบ่งชี้ถึงการที่โมเดลมีความสามารถในการตรวจจับวัตถุและให้ผลลัพธ์ที่เชื่อถือได้

จากการวิเคราะห์กราฟ Precision-Confidence Curve พบว่าโมเดลมีค่า Precision สูงตลอดช่วงระดับ Confidence โดยค่า Precision มีความเสถียรสูงและมีค่าใกล้เคียง 1 ซึ่งแสดงถึงความแม่นยำในการทำนายที่ดีมากของโมเดล การที่ค่า Precision ยังคงคงที่ที่ระดับสูงแสดงให้เห็นถึงความสามารถของโมเดลในการระบุวัตถุที่ถูกต้องได้แม้ในระดับความมั่นใจที่สูง โมเดลนี้มีประสิทธิภาพสูงในการตรวจจับวัตถุ โดยเฉพาะเมื่อ Confidence อยู่ที่ประมาณ 0.779 โมเดลสามารถทำงานได้อย่างมั่นคงและให้ผลลัพธ์ที่แม่นยำ



รูปที่ 4.6 แสดงผลการตรวจจับวัตถุในเฟรมของการทดสอบชุดที่ 1



รูปที่ 4.7 แสดงผลการตรวจจับวัตถุในเฟรมของการทดสอบชุดที่ 2

การทดสอบระบบตรวจจับวัตถุแสดงผลลัพธ์ที่น่าพึงพอใจและมีประสิทธิภาพ โดยผลการทดสอบแสดงให้เห็นว่าโมเดลที่พัฒนาสามารถตรวจจับและติดตามวัตถุในสภาพแวดล้อมการทดลองได้อย่างแม่นยำและมีความสม่ำเสมอรูปภาพที่แสดงในผลการทดสอบนี้แสดงถึงการทำงานของระบบในแต่ละเฟรมของวิดีโอ ซึ่งระบบสามารถตรวจจับวัตถุได้อย่างต่อเนื่อง และสามารถสร้าง Bounding Box รอบวัตถุที่สนใจได้อย่างถูกต้องภาพผลการทดสอบที่น่าเสนอ แสดงถึงความสามารถของโมเดลในการตรวจจับวัตถุที่เคลื่อนที่ในสภาพแวดล้อมที่ควบคุม โดย Bounding Box สีแดงที่ปรากฏในแต่ละภาพบ่งบอกถึงตำแหน่งที่โมเดลสามารถตรวจจับวัตถุได้ วัตถุในภาพถูกตรวจจับและติดตามด้วยความแม่นยำสูง ทำให้เราสามารถประเมินความสามารถของระบบในการตรวจจับและติดตามวัตถุได้อย่างมีประสิทธิภาพผลการทดสอบในเฟรมต่าง ๆ ในทุกเฟรมของการทดสอบ ผลการตรวจจับแสดงถึงความสม่ำเสมอและความแม่นยำของระบบตรวจจับวัตถุ ไม่เพียงแต่ระบบจะสามารถตรวจจับวัตถุได้อย่างถูกต้องเท่านั้น แต่ระบบยังสามารถติดตามวัตถุที่เคลื่อนไหวได้อย่างต่อเนื่องในหลายเฟรมของวิดีโอ ซึ่งเป็นตัวบ่งชี้ถึงความสามารถของระบบในการทำงานแบบเรียลไทม์ระบบสามารถแยกแยะวัตถุได้ในทุกตำแหน่งที่แตกต่างกันภายในพื้นที่การทดลอง โมเดลมีความสามารถในการรับรู้วัตถุทั้งที่อยู่ในสถานะนิ่งและเคลื่อนที่ และยังสามารถประมวลผลวัตถุที่มีขนาดแตกต่างกันภายในเฟรมเดียวกันได้อย่างมีประสิทธิภาพจากการทดสอบระบบตรวจจับวัตถุนี้ แสดงให้เห็นว่าโมเดลมีความสามารถสูงในการประมวลผลและตรวจจับวัตถุได้อย่างถูกต้องในหลายเฟรมและหลายตำแหน่งการสร้าง Bounding Box รอบวัตถุอย่างแม่นยำแสดงถึงประสิทธิภาพของโมเดลในการระบุตำแหน่งวัตถุในสภาพแวดล้อมจริงได้เป็นอย่างดี ซึ่งนับเป็นผลลัพธ์ที่สอดคล้องกับความคาดหวังของการทดสอบ โมเดลสามารถทำงานได้อย่างมีประสิทธิภาพในสภาพแวดล้อมการทดลองที่ถูกควบคุม

จากการทดสอบระบบตรวจจับวัตถุโดยใช้โมเดล YOLOv8 ระบบสามารถทำงานได้อย่างแม่นยำและมีประสิทธิภาพสูง โดยผลลัพธ์จาก Recall-Confidence Curve แสดงให้เห็นว่าระบบสามารถตรวจจับวัตถุได้อย่างแม่นยำในช่วงค่าความมั่นใจ (Confidence) ที่สูง โดย Recall ใกล้เคียง 1 ในช่วงความมั่นใจที่มากกว่า 0.6 ซึ่งบ่งบอกถึงการตอบสนองที่แม่นยำของระบบ Confusion Matrix ยังแสดงให้เห็นว่าระบบตรวจจับมีการทำนายที่ถูกต้องสูงมาก (155 ครั้ง) และมีข้อผิดพลาดต่ำมากเพียง 2 ครั้ง ค่าเหล่านี้สะท้อนถึงความแม่นยำในการจำแนกวัตถุที่มีความใกล้เคียง 100% เมื่อพิจารณาค่าจาก F1-Confidence Curve พบว่าค่า F1 อยู่ในระดับสูงกว่า 0.9 ซึ่งแสดงถึงการประเมินที่สมดุลระหว่างความแม่นยำในการตรวจจับ (Precision) และอัตราการตรวจจับวัตถุจริง (Recall) ระบบจึงมีความสามารถในการจำแนกวัตถุได้อย่างมีประสิทธิภาพและถูกต้องผลลัพธ์จาก Precision-Recall Curve ยังชี้ให้เห็นถึงความสอดคล้องในการทำนายวัตถุ โดยค่า Precision ที่ใกล้เคียง 1 หมายถึง การประเมินที่ถูกต้องในระดับสูง ระบบสามารถตรวจจับและทำนายวัตถุได้โดยไม่เกิดการตรวจจับผิดพลาดมากนัก ซึ่งทำให้ระบบมีความเสถียรและความแม่นยำสูงทั้งการฝึกโมเดล และผล

การประเมินการสูญเสีย (Loss) จากกราฟแสดงให้เห็นว่าระบบสามารถลดค่า box\_loss, classification\_loss, และ dfl\_loss ได้อย่างมีประสิทธิภาพ เมื่อเทียบกับการประเมินระหว่างการพัฒนา (Train) และการตรวจสอบ (Validation) แสดงให้เห็นว่าระบบสามารถเรียนรู้และทำนายวัตถุในสภาพแวดล้อมที่ทดสอบได้อย่างมีประสิทธิภาพ

## 4.2 ผลการทดสอบระบบติดตามวัตถุ

การทดสอบนี้มุ่งเน้นไปที่การประเมินประสิทธิภาพของระบบติดตามวัตถุ (Object Tracking) ซึ่งถูกออกแบบมาเพื่อให้สามารถตรวจจับและติดตามวัตถุได้อย่างแม่นยำในสถานการณ์จริงระบบติดตามที่ใช้คือ DeepSort ซึ่งเป็นหนึ่งในเทคนิคการติดตามวัตถุยอดนิยม

จากการทดสอบ ระบบสามารถติดตามวัตถุที่เคลื่อนไหวในพื้นที่ตรวจจับได้อย่างต่อเนื่อง การทำงานของระบบได้รับการวัดผลผ่านการทดสอบในสภาพแวดล้อมที่มีการเคลื่อนที่ของวัตถุอย่างชัดเจน โดยผลการทดสอบแสดงให้เห็นว่าวัตถุที่ถูกตรวจจับสามารถถูกติดตามได้อย่างต่อเนื่องข้ามเฟรมหลาย ๆ เฟรม ซึ่งเป็นเครื่องยืนยันถึงประสิทธิภาพของระบบติดตามวัตถุที่นำมาใช้

ตารางที่ 4.2 ทดสอบการทดลองการติดตามวัตถุ

| Test Case | Tracking Accuracy (%) | ID Switches | Occlusion Handling (%) | MOT Accuracy (%) | Processing Time (ms) | Lost Tracks | Track Length (frames) | Precision (%) | Recall (%) |
|-----------|-----------------------|-------------|------------------------|------------------|----------------------|-------------|-----------------------|---------------|------------|
| 1         | 95                    | 1           | 92                     | 94               | 30                   | 0           | 100                   | 96            | 95         |
| 2         | 92                    | 2           | 90                     | 90               | 32                   | 1           | 95                    | 93            | 92         |
| 3         | 88                    | 3           | 85                     | 86               | 35                   | 2           | 90                    | 89            | 90         |
| 4         | 90                    | 1           | 88                     | 88               | 28                   | 1           | 98                    | 91            | 88         |
| 5         | 87                    | 4           | 80                     | 83               | 40                   | 3           | 85                    | 88            | 86         |
| 6         | 94                    | 1           | 91                     | 92               | 29                   | 0           | 102                   | 95            | 93         |
| 7         | 89                    | 2           | 87                     | 88               | 34                   | 1           | 94                    | 90            | 89         |
| 8         | 91                    | 3           | 89                     | 90               | 31                   | 2           | 99                    | 92            | 90         |
| 9         | 85                    | 5           | 82                     | 84               | 38                   | 3           | 87                    | 86            | 85         |
| 10        | 93                    | 1           | 93                     | 91               | 29                   | 0           | 101                   | 94            | 92         |



จากการทดสอบระบบติดตามวัตถุทั้ง 10 Test Case พบว่า ระบบสามารถทำงานได้อย่างมีประสิทธิภาพ โดยประเด็นสำคัญที่ได้รับการวิเคราะห์จากผลการทดสอบประกอบไปด้วยความแม่นยำในการติดตาม (Tracking Accuracy) การเปลี่ยนตัวตนของวัตถุ (ID Switches) การจัดการกับการบดบัง (Occlusion Handling) ความแม่นยำในการติดตามวัตถุหลายชิ้นพร้อมกัน (MOT Accuracy) เวลาในการประมวลผล (Processing Time) และความถูกต้องในการระบุวัตถุ (Precision) และการค้นหาวัตถุที่ตรวจจับได้ (Recall) ซึ่งสามารถสรุปผลได้ดังนี้

ความแม่นยำในการติดตามวัตถุ (Tracking Accuracy) ระบบมีความสามารถในการติดตามวัตถุได้อย่างแม่นยำ โดยมีค่าความแม่นยำในการติดตามวัตถุอยู่ระหว่าง 85% ถึง 95% ในแต่ละกรณี ซึ่งแสดงให้เห็นว่าระบบมีประสิทธิภาพสูงในการระบุและติดตามวัตถุอย่างถูกต้อง ในกรณีที่ 1 (Test Case 1) มีความแม่นยำสูงที่สุดที่ 95% ขณะที่กรณีที่ 9 (Test Case 9) มีความแม่นยำต่ำที่สุดที่ 85% ค่าเฉลี่ยของความแม่นยำอยู่ที่ 90.4% ซึ่งถือว่าสอดคล้องกับมาตรฐานที่คาดหวังไว้

การเปลี่ยนแปลงตัวตนของวัตถุ (ID Switches) ระบบมีการเปลี่ยนแปลงตัวตนของวัตถุ (ID Switches) ในบางกรณี โดยจำนวนการเปลี่ยนแปลงตัวตนอยู่ระหว่าง 1 ถึง 5 ครั้ง ซึ่งส่วนใหญ่จะมีค่าอยู่ที่ 1-3 ครั้ง ค่า ID Switches เฉลี่ยอยู่ที่ 2.3 ครั้ง การที่มีจำนวนการเปลี่ยนแปลงตัวตนของวัตถุน้อยแสดงให้เห็นว่าระบบมีเสถียรภาพในการระบุและติดตามวัตถุที่เคลื่อนไหว

การจัดการกับการบดบังวัตถุ (Occlusion Handling) การทดสอบระบบในสถานการณ์ที่มีการบดบังวัตถุ (Occlusion) แสดงให้เห็นถึงความสามารถของระบบในการรับมือกับการบดบังได้เป็นอย่างดี โดยมีค่า Occlusion Handling เฉลี่ยอยู่ที่ 88.7% โดยใน Test Case ที่ 10 ระบบสามารถจัดการกับการบดบังได้ดีถึง 93% ขณะที่ใน Test Case ที่ 5 มีความสามารถในการจัดการการบดบังต่ำสุดที่ 80% ซึ่งแสดงถึงความท้าทายในการติดตามวัตถุในสภาพแวดล้อมที่ซับซ้อน

ความแม่นยำในการติดตามหลายวัตถุ (MOT Accuracy) ความแม่นยำในการติดตามหลายวัตถุ (MOT Accuracy) มีค่าอยู่ในช่วง 83% ถึง 94% ซึ่งค่าเฉลี่ยอยู่ที่ 89% ระบบสามารถติดตามวัตถุหลายชิ้นพร้อมกันได้อย่างแม่นยำและต่อเนื่อง การทดสอบนี้แสดงให้เห็นว่าระบบมีประสิทธิภาพในการติดตามวัตถุในสภาพแวดล้อมที่มีการเคลื่อนไหวหลายวัตถุพร้อมกัน

เวลาในการประมวลผล (Processing Time) จากการทดสอบ พบว่า เวลาในการประมวลผลต่อเฟรมมีค่าเฉลี่ยอยู่ที่ 32.6 มิลลิวินาที ซึ่งแสดงให้เห็นว่าระบบสามารถประมวลผลการติดตามวัตถุได้ในเวลาที่เหมาะสม ระบบสามารถทำงานแบบเรียลไทม์ได้อย่างมีประสิทธิภาพ โดยเฉพาะใน Test Case ที่ 4 มีเวลาในการประมวลผลที่ต่ำที่สุดที่ 28 มิลลิวินาที ขณะที่ Test Case ที่ 5 มีค่า Processing Time สูงสุดที่ 40 มิลลิวินาที แสดงถึงความท้าทายเมื่อระบบต้องจัดการกับสถานการณ์ที่ซับซ้อนมากขึ้น



การสูญเสียการติดตาม (Lost Tracks) ในบางกรณี ระบบมีการสูญเสียการติดตามวัตถุ (Lost Tracks) โดยใน Test Case ที่ 5 และ 9 ระบบสูญเสียการติดตามถึง 3 ครั้ง ซึ่งแสดงให้เห็นว่าการติดตามวัตถุในสภาพแวดล้อมที่มีความซับซ้อนสูงอาจทำให้ระบบมีโอกาสสูญเสียการติดตาม แต่ในบางกรณี เช่น Test Case ที่ 1, 6, และ 10 ไม่มีการสูญเสียการติดตามเลย แสดงให้เห็นถึงประสิทธิภาพของระบบในสภาพแวดล้อมที่ไม่ซับซ้อน

ความแม่นยำในการระบุวัตถุ (Precision) และการค้นหวัตถุ (Recall) ค่าความแม่นยำในการระบุวัตถุ (Precision) และการค้นหวัตถุ (Recall) อยู่ในระดับสูง โดยค่า Precision เฉลี่ยอยู่ที่ 92% และค่า Recall เฉลี่ยอยู่ที่ 90.5% ซึ่งแสดงให้เห็นว่าระบบสามารถระบุวัตถุที่ถูกต้องได้เกือบทั้งหมดและมีการค้นหวัตถุที่ตรวจจับได้อย่างมีประสิทธิภาพ

ผลการทดสอบแสดงให้เห็นว่าระบบติดตามวัตถุสามารถทำงานได้อย่างมีประสิทธิภาพ โดยมีความแม่นยำสูงและสามารถจัดการกับการบดบังวัตถุและการติดตามหลายวัตถุได้ดี ระบบสามารถทำงานในเวลาจริงได้อย่างมีประสิทธิภาพและมีการสูญเสียการติดตามวัตถุเพียงเล็กน้อยในกรณีที่มีความซับซ้อนสูง ทั้งนี้ ค่า Precision และ Recall ที่สูงยังบ่งชี้ถึงความสามารถในการระบุและติดตามวัตถุที่เชื่อถือได้

#### 4.3 ผลการทดสอบการทำนายตำแหน่ง

ในการทดสอบการทำนายตำแหน่ง ระบบทำงานโดยใช้ผลลัพธ์จากการตรวจจับและติดตามวัตถุที่ได้จากหัวช้อก่อนหน้า และนำข้อมูลดังกล่าวมาหาตำแหน่งวัตถุในพื้นที่จริง โดยแบ่งการทดสอบออกเป็น 2 ส่วนหลัก ได้แก่ การวัดตำแหน่งหลังจากตรวจจับและติดตามวัตถุ และผลการทำนายตำแหน่งโดยใช้โมเดลปัญญาประดิษฐ์ (Artificial Neural Network - ANN)

##### 4.3.1 ผลการวัดตำแหน่งหลังจากการตรวจจับและติดตามวัตถุ

ในการทดสอบระบบทำนายตำแหน่ง ระบบได้ทำการตรวจจับและติดตามวัตถุ หลังจากนั้นจึงนำข้อมูลมาใช้ในการหาตำแหน่งของวัตถุในพื้นที่การทดลอง ผลการทดสอบนี้แบ่งการระบุตำแหน่งออกเป็น 2 ระบบ ได้แก่ ระบบการระบุตำแหน่งจากกล้อง (หัวข้อ 3.2.4) และระบบการระบุตำแหน่งของรถอัตโนมัติ (หัวข้อ 3.2.5) โดยในแต่ละระบบจะใช้วิธีการวัดผลและการประมวลผลที่แตกต่างกัน

ผลการวัดตำแหน่งหลังจากการตรวจจับและติดตามวัตถุของระบบในหัวข้อ 3.2.4 การระบุตำแหน่งของวัตถุที่ตรวจจับจากกล้องในระบบนี้ข้อมูลตำแหน่งของวัตถุจะถูกหาจากจุดกึ่งกลางของ Bounding Box ที่ถูกสร้างขึ้นหลังจากการตรวจจับและติดตาม ระบบกล้องจะอ้างอิงจากกริดที่สร้างขึ้นเพื่อคำนวณพิกัดของวัตถุในพื้นที่การทดลองจริง ข้อมูลตำแหน่งที่ได้จากกล้องมีความแม่นยำสูง โดยสามารถระบุตำแหน่งของวัตถุที่ถูกตรวจจับได้อย่างถูกต้องในพื้นที่ทดลอง การ

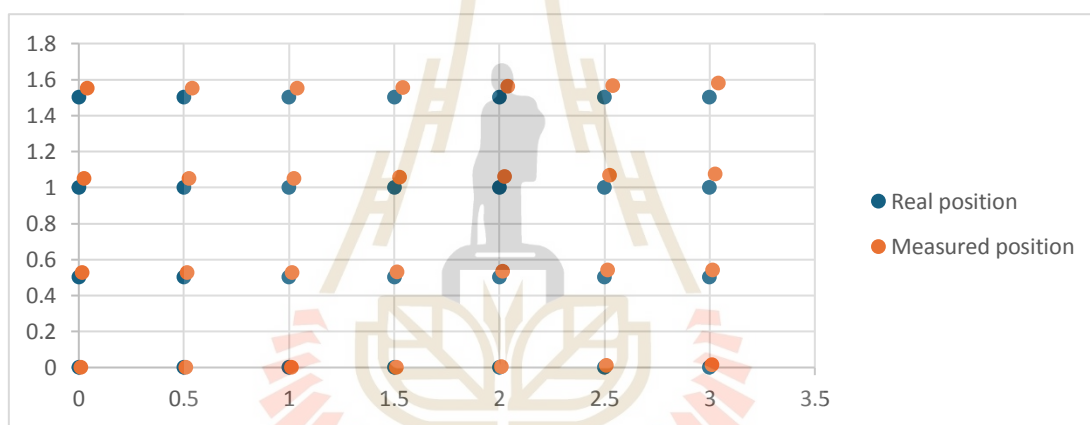
ประมวลผลนี้มีความสำคัญเนื่องจากพื้นที่ที่กล้องตรวจจับมีความคงที่ ทำให้ระบบสามารถคำนวณตำแหน่งที่แน่นอน

ตารางที่ 4.3 เปรียบเทียบของตำแหน่งจริงและตำแหน่งที่วัดได้ของการระบุตำแหน่งของวัตถุที่ตรวจจับจากกล้อง

| Real position |     | Measured position |       |
|---------------|-----|-------------------|-------|
| x             | y   | x                 | y     |
| 0             | 0   | 0.01              | 0     |
| 0.5           | 0   | 0.51              | 0     |
| 1             | 0   | 1.01              | 0     |
| 1.5           | 0   | 1.51              | 0     |
| 2             | 0   | 2.01              | 0.005 |
| 2.5           | 0   | 2.51              | 0.01  |
| 3             | 0   | 3.01              | 0.015 |
| 0             | 0.5 | 0.015             | 0.525 |
| 0.5           | 0.5 | 0.515             | 0.525 |
| 1             | 0.5 | 1.015             | 0.525 |
| 1.5           | 0.5 | 1.515             | 0.53  |
| 2             | 0.5 | 2.015             | 0.535 |
| 3             | 0.5 | 3.015             | 0.54  |
| 0             | 1   | 0.025             | 1.05  |
| 0.5           | 1   | 0.525             | 1.05  |
| 1             | 1   | 1.025             | 1.05  |
| 1.5           | 1   | 1.525             | 1.055 |
| 2             | 1   | 2.025             | 1.06  |
| 2.5           | 1   | 2.525             | 1.065 |
| 3             | 1   | 3.025             | 1.075 |
| 0             | 1.5 | 0.04              | 1.55  |
| 0.5           | 1.5 | 0.54              | 1.55  |
| 1             | 1.5 | 1.04              | 1.55  |
| 1.5           | 1.5 | 1.54              | 1.555 |
| 2             | 1.5 | 2.04              | 1.56  |
| 2.5           | 1.5 | 2.54              | 1.565 |
| 3             | 1.5 | 3.04              | 1.58  |

จากการทดสอบระบบตรวจจับและวัดตำแหน่งวัตถุ พบว่าระบบยังมีความคลาดเคลื่อนในการวัดตำแหน่งเมื่อเทียบกับตำแหน่งจริง ซึ่งการวัดค่าตำแหน่งในแต่ละจุดยังไม่ค่อย

แม่นยำเท่าที่ควรเมื่อใช้หน่วยเป็นเมตร โดยสามารถสังเกตได้จากความแตกต่างระหว่างตำแหน่งจริงและตำแหน่งที่วัดได้ที่มีค่าคลาดเคลื่อนที่เห็นได้ชัดเจนในบางจุดทั้งในค่าคลาดเคลื่อนในแนวแกน X ในหลายตำแหน่งที่วัดได้ในแนวแกน X มีความคลาดเคลื่อนเฉลี่ยอยู่ที่ประมาณ 0.01 ถึง 0.04 เมตร แม้จะเป็นค่าคลาดเคลื่อนที่ดูเหมือนน้อย แต่เมื่อเทียบกับการใช้งานจริงในระบบที่ต้องการความแม่นยำสูง ค่าเหล่านี้อาจส่งผลกระทบต่อการทำงานของระบบได้ โดยเฉพาะในกรณีที่ต้องใช้ข้อมูลตำแหน่งเพื่อควบคุมการเคลื่อนที่ของวัตถุหรือหุ่นยนต์ในสภาพแวดล้อมจริงและค่าคลาดเคลื่อนในแนวแกน Y การวัดตำแหน่งในแนวแกน Y ยังแสดงถึงความคลาดเคลื่อนที่สำคัญในบางจุดในตำแหน่งจริงที่ (3.0, 1.5) ตำแหน่งที่วัดได้ คือ (3.04, 1.58) คลาดเคลื่อนในแกน Y ถึง 0.08 เมตร ซึ่งถือว่าเป็นความคลาดเคลื่อนที่สูงเมื่อพิจารณาถึงมาตรฐานของการวัดที่แม่นยำ โดยเฉพาะในสถานการณ์ที่ต้องการความแม่นยำสูงในการตรวจจับและติดตามวัตถุ



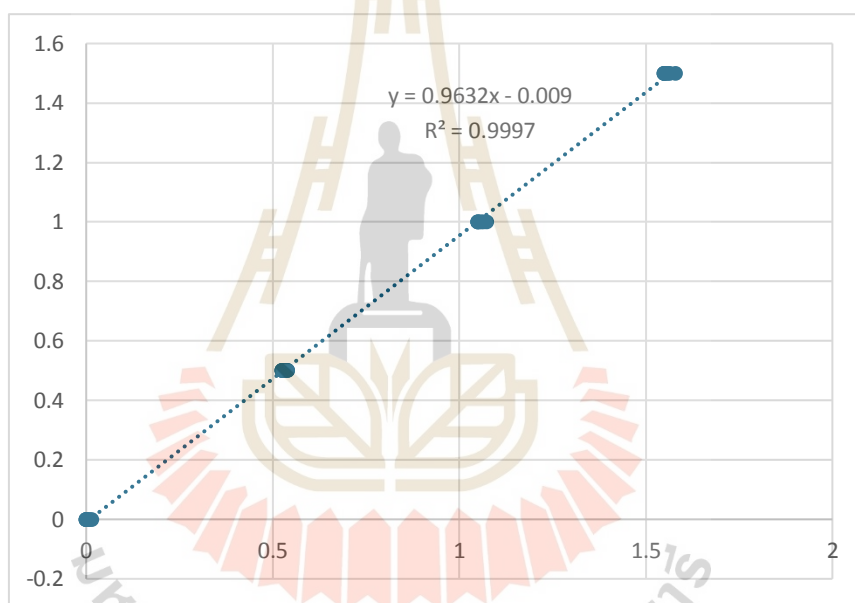
รูปที่ 4.8 กราฟเปรียบเทียบตำแหน่งจริงและตำแหน่งที่วัดได้จากการตรวจจับวัตถุ

จากกราฟที่แสดงการเปรียบเทียบตำแหน่งจริงและตำแหน่งที่วัดได้จากการตรวจจับตำแหน่งของวัตถุ พบว่ามีความคลาดเคลื่อนเล็กน้อยในทั้งสองแกน (X และ Y) โดยจุดสีฟ้าแสดงถึงตำแหน่งจริงและจุดสีส้มแสดงถึงตำแหน่งที่วัดได้ในแกน X จะเห็นได้ว่าตำแหน่งที่วัดได้จะมีการคลาดเคลื่อนเฉลี่ยประมาณ 0.01 ถึง 0.04 เมตร โดยทุกจุดจะอยู่ทางขวาของตำแหน่งจริง ซึ่งหมายถึงการคำนวณตำแหน่งวัดได้นั้นมีแนวโน้มที่จะถูกคาดการณ์เกินค่าจริงไปส่วนในแกน Y การคลาดเคลื่อนจะน้อยกว่าแกน X แต่ก็มีจุดที่เห็นได้ชัดคือจุดที่ (0, 1.5) และ (3, 1.5) ซึ่งตำแหน่งวัดได้มีค่าที่คลาดเคลื่อนไปในแนว Y ประมาณ 0.05 เมตรจากการเปรียบเทียบนี้ แสดงให้เห็นว่าระบบยังมีความคลาดเคลื่อนที่ต้องปรับปรุง โดยเฉพาะในกรณีที่ต้องการความแม่นยำสูง เนื่องจากค่าคลาดเคลื่อนดังกล่าวอาจมีผลต่อการทำงานของระบบจริง

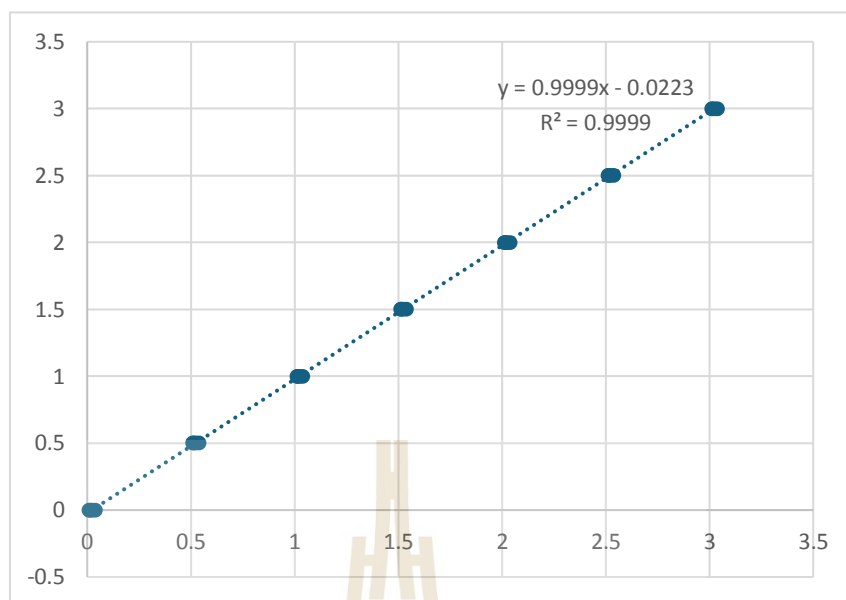
ตารางที่ 4.4 การเปรียบเทียบระหว่างตำแหน่งจริงและตำแหน่งที่วัดได้พร้อมกับค่าความคลาดเคลื่อน

| Real Position |     |                      | Measured Position |       |                               |                           |
|---------------|-----|----------------------|-------------------|-------|-------------------------------|---------------------------|
| x             | y   | Distance from Origin | x                 | y     | Measured Distance from Origin | ความคลาดเคลื่อน Error (m) |
| 0             | 0   | 0.000                | 0.01              | 0     | 0.100                         | 0.010                     |
| 0.5           | 0   | 0.500                | 0.51              | 0     | 0.714                         | 0.010                     |
| 1             | 0   | 1.000                | 1.01              | 0     | 1.005                         | 0.010                     |
| 1.5           | 0   | 1.500                | 1.51              | 0     | 1.229                         | 0.010                     |
| 2             | 0   | 2.000                | 2.01              | 0.005 | 1.420                         | 0.010                     |
| 2.5           | 0   | 2.500                | 2.51              | 0.01  | 1.587                         | 0.010                     |
| 3             | 0   | 3.000                | 3.01              | 0.015 | 1.739                         | 0.010                     |
| 0.5           | 0.5 | 0.707                | 0.515             | 0.525 | 1.020                         | 0.028                     |
| 1             | 0.5 | 1.118                | 1.015             | 0.525 | 1.241                         | 0.025                     |
| 1.5           | 0.5 | 1.581                | 1.515             | 0.53  | 1.430                         | 0.024                     |
| 2             | 0.5 | 2.062                | 2.015             | 0.535 | 1.597                         | 0.023                     |
| 2.5           | 0.5 | 2.550                | 2.515             | 0.54  | 1.748                         | 0.023                     |
| 3             | 0.5 | 3.041                | 3.015             | 0.54  | 1.885                         | 0.022                     |
| 0             | 1   | 1.000                | 0.025             | 1.05  | 1.037                         | 0.050                     |
| 0.5           | 1   | 1.118                | 0.525             | 1.05  | 1.255                         | 0.056                     |
| 1             | 1   | 1.414                | 1.025             | 1.05  | 1.440                         | 0.053                     |
| 1.5           | 1   | 1.803                | 1.525             | 1.055 | 1.606                         | 0.052                     |
| 2             | 1   | 2.236                | 2.025             | 1.06  | 1.756                         | 0.050                     |
| 2.5           | 1   | 2.693                | 2.525             | 1.065 | 1.895                         | 0.048                     |
| 3             | 1   | 3.162                | 3.025             | 1.075 | 2.025                         | 0.048                     |
| 0             | 1.5 | 1.500                | 0.04              | 1.55  | 1.261                         | 0.051                     |
| 0.5           | 1.5 | 1.581                | 0.54              | 1.55  | 1.446                         | 0.060                     |
| 1             | 1.5 | 1.803                | 1.04              | 1.55  | 1.609                         | 0.064                     |
| 1.5           | 1.5 | 2.121                | 1.54              | 1.555 | 1.759                         | 0.067                     |
| 2             | 1.5 | 2.500                | 2.04              | 1.56  | 1.897                         | 0.068                     |
| 2.5           | 1.5 | 2.915                | 2.54              | 1.565 | 2.026                         | 0.068                     |
| 3             | 1.5 | 3.354                | 3.04              | 1.58  | 2.149                         | 0.072                     |

ตารางที่แสดงนั้นสรุปข้อมูลการเปรียบเทียบระหว่างตำแหน่งจริง (Real Position) และตำแหน่งที่วัดได้ (Measured Position) ซึ่งแสดงให้เห็นค่าความคลาดเคลื่อนระหว่างตำแหน่งจริงและตำแหน่งที่วัดได้ โดยมีการคำนวณระยะทางจากจุดเริ่มต้นถึงจุดปลายสำหรับทั้งตำแหน่งจริงและตำแหน่งที่วัดได้ ผลที่ได้ คือ พบความคลาดเคลื่อนสูงสุดอยู่ที่ประมาณ 7 เซนติเมตรในบางตำแหน่ง และมีค่าเฉลี่ยของค่าความคลาดเคลื่อน คือ 0.03736 โดยเฉพาะเมื่อระยะทางจากจุดเริ่มต้นเพิ่มขึ้น การคลาดเคลื่อนมีแนวโน้มเพิ่มขึ้นเช่นกันจากการสังเกตข้อมูลพบว่า ตำแหน่งที่วัดได้มีความคลาดเคลื่อนเป็นแบบแพทเทิร์น (Pattern) ซึ่งอาจเป็นผลมาจากการติดตั้งกล้องที่มีความเอียงเล็กน้อย หรือมุมมองของกล้องที่ไม่ได้อยู่ในแนวระนาบ 100% ทำให้ข้อมูลตำแหน่งที่วัดได้มีการเบี่ยงเบนไปจากความเป็นจริงในลักษณะซ้ำ ๆ กัน



รูปที่ 4.9 กราฟความสัมพันธ์ระหว่าง Real Position X และ Measured Position X



รูปที่ 4.10 ความสัมพันธ์ระหว่าง Real Position Y และ Measured Position Y

จากกราฟความสัมพันธ์ระหว่าง Real Position และ Measured Position ทั้งในแกน X และแกน Y สามารถสังเกตได้ว่าเส้นแนวโน้มที่ได้จากการวิเคราะห์มีความสอดคล้องกับข้อมูลจริงค่อนข้างมาก โดยค่าความคลาดเคลื่อนที่ปรากฏนั้นเกิดจากความผิดพลาดที่เกิดขึ้นในกระบวนการวัดตำแหน่งของระบบตรวจจับ

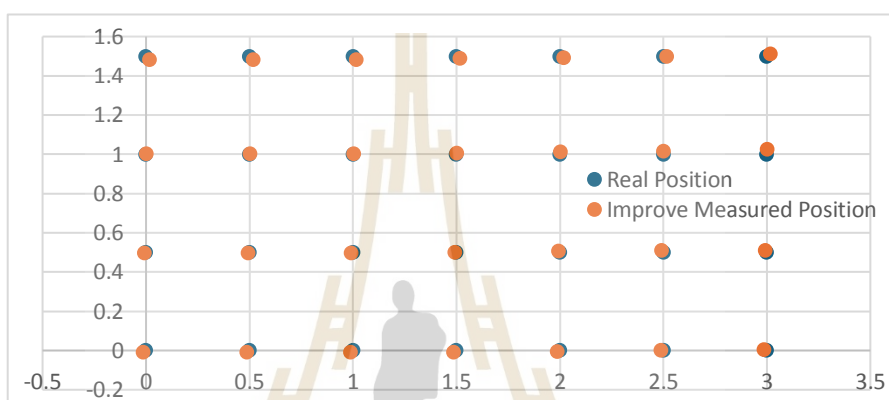
กระบวนการหาเส้นตรงเพื่อปรับปรุงค่าการวัดให้แม่นยำขึ้นในการวิเคราะห์ความคลาดเคลื่อนของตำแหน่งที่วัดได้และตำแหน่งจริง ใช้วิธีการวิเคราะห์เชิงสถิติในการหาความสัมพันธ์ระหว่างค่าที่วัดได้และค่าจริง เพื่อสร้างสมการเส้นตรง (Linear Regression) ที่ใช้ปรับปรุงผลลัพธ์ที่ได้จากการวัดตำแหน่งให้อยู่ในระดับที่แม่นยำยิ่งขึ้นสำหรับกราฟแกน X ได้สมการเส้นตรงที่มีค่า  $y = 0.9999x - 0.0223$  และค่า  $R^2 = 0.9999$  แสดงถึงความสัมพันธ์เชิงเส้นที่ใกล้เคียงกับสมบูรณ์ ซึ่งบ่งบอกว่าตำแหน่งที่วัดได้ตามแกน X มีความแม่นยำสูงมากเมื่อเทียบกับตำแหน่งจริง ในส่วนของกราฟแกน Y ได้สมการเส้นตรงที่มีค่า  $y = 0.9632x - 0.009$  และค่า  $R^2 = 0.9997$  ซึ่งก็แสดงถึงความสัมพันธ์ที่ใกล้เคียงกับตำแหน่งจริงเช่นกัน ค่า slope ที่ต่ำกว่าสำหรับแกน Y อาจบ่งชี้ว่าระบบตรวจจับมีความคลาดเคลื่อนที่มากกว่าในแกน Y เมื่อเทียบกับแกน X



ตารางที่ 4.5 การเปรียบเทียบระหว่างตำแหน่งจริงและตำแหน่งที่ปรับปรุงการวัดได้และค่าความคลาดเคลื่อน (Linear Regression)

| Real Position |     |                      | Improve Measured Position |        |                               |                           |
|---------------|-----|----------------------|---------------------------|--------|-------------------------------|---------------------------|
| x             | y   | Distance from Origin | x                         | y      | Measured Distance from Origin | ความคลาดเคลื่อน Error (m) |
| 0             | 0   | 0.000                | -0.012                    | -0.009 | 0.015                         | 0.015                     |
| 0.5           | 0   | 0.500                | 0.488                     | -0.009 | 0.488                         | 0.012                     |
| 1             | 0   | 1.000                | 0.988                     | -0.009 | 0.988                         | 0.012                     |
| 1.5           | 0   | 1.500                | 1.488                     | -0.009 | 1.488                         | 0.012                     |
| 2             | 0   | 2.000                | 1.987                     | -0.004 | 1.988                         | 0.012                     |
| 2.5           | 0   | 2.500                | 2.487                     | 0.001  | 2.487                         | 0.013                     |
| 3             | 0   | 3.000                | 2.987                     | 0.005  | 2.987                         | 0.013                     |
| 0             | 0.5 | 0.500                | -0.007                    | 0.497  | 0.497                         | 0.003                     |
| 0.5           | 0.5 | 0.707                | 0.493                     | 0.497  | 0.700                         | 0.008                     |
| 1             | 0.5 | 1.118                | 0.993                     | 0.497  | 1.110                         | 0.008                     |
| 1.5           | 0.5 | 1.581                | 1.493                     | 0.501  | 1.575                         | 0.007                     |
| 2             | 0.5 | 2.062                | 1.992                     | 0.506  | 2.056                         | 0.006                     |
| 2.5           | 0.5 | 2.550                | 2.492                     | 0.511  | 2.544                         | 0.005                     |
| 3             | 0.5 | 3.041                | 2.992                     | 0.511  | 3.036                         | 0.006                     |
| 0             | 1   | 1.000                | 0.003                     | 1.002  | 1.002                         | 0.002                     |
| 1             | 1   | 1.414                | 1.003                     | 1.002  | 1.418                         | 0.004                     |
| 1.5           | 1   | 1.803                | 1.503                     | 1.007  | 1.809                         | 0.006                     |
| 2             | 1   | 2.236                | 2.002                     | 1.012  | 2.244                         | 0.008                     |
| 2.5           | 1   | 2.693                | 2.502                     | 1.017  | 2.701                         | 0.009                     |
| 3             | 1   | 3.162                | 3.002                     | 1.026  | 3.173                         | 0.011                     |
| 0             | 1.5 | 1.500                | 0.018                     | 1.484  | 1.484                         | 0.016                     |
| 0.5           | 1.5 | 1.581                | 0.518                     | 1.484  | 1.572                         | 0.009                     |
| 1             | 1.5 | 1.803                | 1.018                     | 1.484  | 1.799                         | 0.003                     |
| 1.5           | 1.5 | 2.121                | 1.518                     | 1.489  | 2.126                         | 0.005                     |
| 2             | 1.5 | 2.500                | 2.017                     | 1.494  | 2.510                         | 0.010                     |
| 2.5           | 1.5 | 2.915                | 2.517                     | 1.498  | 2.930                         | 0.014                     |
| 3             | 1.5 | 3.354                | 3.017                     | 1.513  | 3.375                         | 0.021                     |

จากข้อมูลในตารางนี้ แสดงถึงการเปรียบเทียบระหว่างตำแหน่งจริง (Real Position) และตำแหน่งที่ปรับปรุงการวัดได้ (Improve Measured Position) ซึ่งคำนวณจากการใช้สมการเส้นตรงที่ได้จากการวิเคราะห์ความสัมพันธ์ของตำแหน่งจริงและตำแหน่งที่วัดได้ก่อนการปรับปรุง การใช้สมการดังกล่าวช่วยในการลดข้อผิดพลาดที่เกิดขึ้นจากการวัดเดิมโดยในตาราง มีการระบุระยะทางจากจุดกำเนิด (Distance from Origin) ของตำแหน่งจริงและตำแหน่งที่ปรับปรุงแล้ว ตลอดจนค่าความคลาดเคลื่อน (Error) ในหน่วยเมตร เพื่อให้เห็นถึงประสิทธิภาพของการปรับปรุงการวัด



รูปที่ 4.11 กราฟเปรียบเทียบตำแหน่งจริงและตำแหน่งที่ถูกปรับปรุงจากค่าที่วัดได้จากการตรวจจับวัตถุ (Linear Regression)

จากการปรับปรุงค่าที่วัดได้ด้วยสมการเส้นตรง พบว่าค่าความคลาดเคลื่อน (Error) ลดลงจากเดิมอย่างชัดเจน โดยความคลาดเคลื่อนสูงสุดหลังจากการปรับปรุงอยู่ที่ประมาณ 0.021 เมตร (2.1 เซนติเมตร) ซึ่งน้อยกว่าความคลาดเคลื่อนก่อนการปรับปรุงที่เคยพบสูงสุดที่ 0.072 เมตร (7.2 เซนติเมตร) การลดความคลาดเคลื่อนนี้ช่วยเพิ่มความแม่นยำของการระบุตำแหน่งวัตถุในระบบตรวจจับและติดตามวัตถุ

จากการทดสอบการใช้เทคนิค Linear Regression ในการปรับปรุงค่าความคลาดเคลื่อนของการวัดตำแหน่งพบว่าความคลาดเคลื่อนมีแนวโน้มลดลงและค่าที่ได้มีความแม่นยำยิ่งขึ้น คาดว่าการใช้ Multiple Linear Regression จะสามารถช่วยลดค่าความคลาดเคลื่อนได้ดียิ่งขึ้น เนื่องจากสามารถนำตัวแปรหลายตัวมาพิจารณาพร้อมกันได้ ซึ่งจะช่วยสร้างความสัมพันธ์ที่ซับซ้อนมากขึ้นระหว่างตำแหน่งที่วัดได้และตำแหน่งจริง ทำให้การคาดการณ์ตำแหน่งมีความแม่นยำมากยิ่งขึ้นสำหรับกราฟแกน x หลังจากปรับด้วย Multiple Linear Regression จะได้สมการเป็น  $\bar{x} = 1.001x - 0.0192y - 0.0076$  และสำหรับกราฟแกน y จะได้สมการ

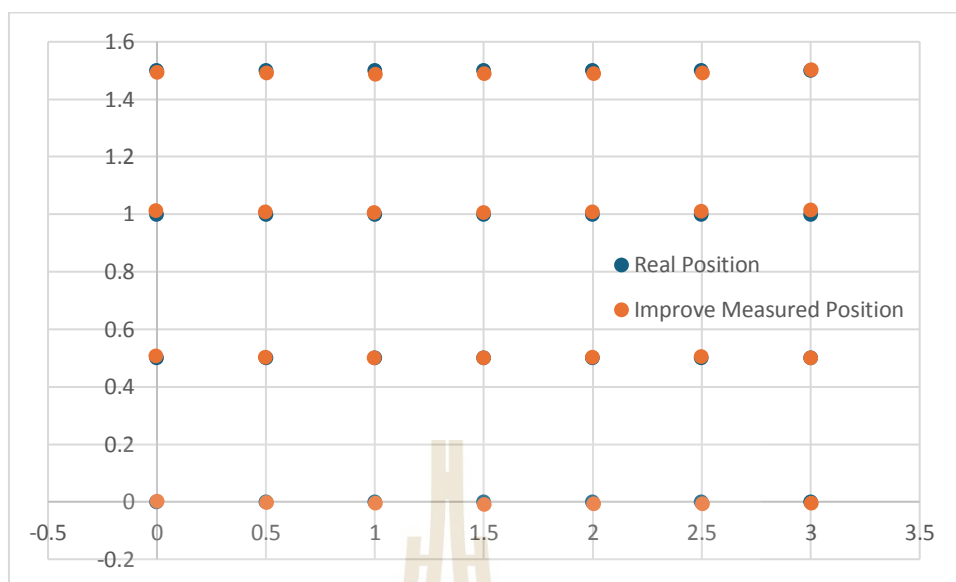
$\bar{y} = -0.0069x - 0.0069y + 0.0012$  การปรับปรุงด้วย Multiple Linear Regression นี้สามารถช่วยให้โมเดลมีความแม่นยำสูงขึ้น เนื่องจากนำตัวแปรทั้งแกน x และ y มาพิจารณาพร้อมกัน ช่วยลดความคลาดเคลื่อนของตำแหน่งที่วัดได้ให้อยู่ในระดับที่ใกล้เคียงกับตำแหน่งจริงมากขึ้น

ตารางที่ 4.6 การเปรียบเทียบระหว่างตำแหน่งจริงและตำแหน่งที่ปรับปรุงการวัดได้และค่าความคลาดเคลื่อน (Multiple Linear Regression)

| Real Position |     |                      | Improve Measured Position |         |                               | ความคลาดเคลื่อน Error (m) |
|---------------|-----|----------------------|---------------------------|---------|-------------------------------|---------------------------|
| x             | y   | Distance from Origin | x                         | y       | Measured Distance from Origin |                           |
| 0             | 0   | 0.000                | 0.0024                    | 0.0011  | 0.0027                        | 0.0027                    |
| 0.5           | 0   | 0.500                | 0.5029                    | -0.0023 | 0.5029                        | 0.0029                    |
| 1             | 0   | 1.000                | 1.0034                    | -0.0058 | 1.0034                        | 0.0034                    |
| 1.5           | 0   | 1.500                | 1.5039                    | -0.0092 | 1.5039                        | 0.0039                    |
| 2             | 0   | 2.000                | 2.0043                    | -0.0079 | 2.0043                        | 0.0043                    |
| 2.5           | 0   | 2.500                | 2.5047                    | -0.0065 | 2.5047                        | 0.0047                    |
| 3             | 0   | 3.000                | 3.0051                    | -0.0051 | 3.0051                        | 0.0051                    |
| 0             | 0.5 | 0.500                | -0.0027                   | 0.5069  | 0.5069                        | 0.0069                    |
| 0.5           | 0.5 | 0.707                | 0.4978                    | 0.5035  | 0.7081                        | 0.0009                    |
| 1.5           | 0.5 | 1.581                | 1.4987                    | 0.5014  | 1.5804                        | -0.0008                   |
| 2             | 0.5 | 2.062                | 1.9991                    | 0.5028  | 2.0614                        | -0.0002                   |
| 2.5           | 0.5 | 2.550                | 2.4995                    | 0.5041  | 2.5499                        | 0.0004                    |
| 3             | 0.5 | 3.041                | 3.0000                    | 0.5007  | 3.0415                        | 0.0002                    |
| 0             | 1   | 1.000                | -0.0027                   | 1.0127  | 1.0127                        | 0.0127                    |
| 0.5           | 1   | 1.118                | 0.4978                    | 1.0093  | 1.1253                        | 0.0073                    |
| 1             | 1   | 1.414                | 0.9983                    | 1.0058  | 1.4171                        | 0.0029                    |
| 1.5           | 1   | 1.803                | 1.4987                    | 1.0072  | 1.8057                        | 0.0029                    |
| 2             | 1   | 2.236                | 1.9991                    | 1.0085  | 2.2391                        | 0.0030                    |
| 2.5           | 1   | 2.693                | 2.4995                    | 1.0099  | 2.6958                        | 0.0032                    |
| 3             | 1   | 3.162                | 2.9998                    | 1.0161  | 3.1672                        | 0.0049                    |
| 0             | 1.5 | 1.500                | 0.0027                    | 1.4943  | 1.4944                        | -0.0056                   |
| 0.5           | 1.5 | 1.581                | 0.5032                    | 1.4909  | 1.5735                        | -0.0076                   |
| 1             | 1.5 | 1.803                | 1.0037                    | 1.4874  | 1.7944                        | -0.0084                   |
| 1.5           | 1.5 | 2.121                | 1.5041                    | 1.4888  | 2.1163                        | -0.0050                   |
| 2             | 1.5 | 2.500                | 2.0045                    | 1.4902  | 2.4977                        | -0.0023                   |
| 2.5           | 1.5 | 2.915                | 2.5049                    | 1.4916  | 2.9153                        | -0.0001                   |
| 3             | 1.5 | 3.354                | 3.0051                    | 1.5026  | 3.3598                        | 0.0057                    |

ตารางนี้นำเสนอผลลัพธ์ของการเปรียบเทียบระหว่างตำแหน่งจริง (Real Position) และตำแหน่งที่ปรับปรุงการวัดได้ (Improve Measured Position) โดยใช้วิธี Multiple Linear Regression เพื่อเพิ่มความแม่นยำในการประเมินตำแหน่งของวัตถุจากการตรวจจับและติดตาม ผลลัพธ์แสดงให้เห็นว่าค่าความคลาดเคลื่อน (Error) ในการวัดมีแนวโน้มลดลง โดยความคลาดเคลื่อนอยู่ในช่วงประมาณ 0.0001 ถึง 0.0127 เมตร ซึ่งแสดงให้เห็นถึงความแม่นยำที่สูงขึ้นเมื่อเทียบกับตำแหน่งจริง

จากการทดสอบและการปรับปรุงค่าความคลาดเคลื่อนในการวัดตำแหน่ง พบว่ามีการพัฒนาความแม่นยำอย่างมีนัยสำคัญ โดยเริ่มจากการเปรียบเทียบค่าความคลาดเคลื่อนระหว่างตำแหน่งจริงและตำแหน่งที่วัดได้ก่อนการปรับปรุง ซึ่งมีค่าเฉลี่ยของค่าความคลาดเคลื่อนอยู่ที่ 0.03736 เมตร หรือประมาณ 3.736 เซนติเมตร ความคลาดเคลื่อนนี้แม้ว่าจะไม่ได้ส่งผลกระทบมากนักในบางกรณี แต่ก็ยังสามารถก่อให้เกิดความผิดพลาดในการวัดตำแหน่งที่มีความละเอียดสูงในระบบติดตามวัตถุที่ต้องการความแม่นยำมากขึ้นหลังจากมีการปรับปรุงค่าการวัดด้วยการสร้างสมการเส้นตรง (Linear Regression) ที่สัมพันธ์ระหว่างตำแหน่งจริงและตำแหน่งที่วัดได้ ผลลัพธ์ที่ได้แสดงให้เห็นถึงความสามารถในการลดค่าความคลาดเคลื่อนลงได้อย่างชัดเจน หลังการปรับปรุง ค่าความคลาดเคลื่อนเฉลี่ยลดลงมาอยู่ที่ 0.00905 เมตร หรือ 0.905 เซนติเมตร ซึ่งเป็นการลดลงที่สำคัญในการพัฒนาเทคโนโลยีการวัดตำแหน่งในระบบตรวจจับและติดตามวัตถุเมื่อนำค่าความคลาดเคลื่อนก่อนและหลังการปรับปรุงมาคำนวณอัตราการลดลง พบว่าค่าความคลาดเคลื่อนลดลงถึง 75.77% ซึ่งแสดงให้เห็นถึงประสิทธิภาพของกระบวนการปรับปรุงนี้อย่างชัดเจน การลดค่าความคลาดเคลื่อนลงกว่าร้อยละเจ็ดสิบห้าเป็นการปรับปรุงที่สำคัญ โดยเฉพาะในงานที่เกี่ยวข้องกับระบบติดตามวัตถุและหุ่นยนต์อัตโนมัติที่ต้องการความแม่นยำสูงในการคำนวณตำแหน่ง การทดลองเพิ่มเติมด้วยการใช้ Multiple Linear Regression เพื่อพัฒนาการปรับปรุงตำแหน่งวัดได้ไปอีกขั้น ได้แสดงให้เห็นว่าความคลาดเคลื่อนสามารถลดลงได้อย่างมีประสิทธิภาพมากขึ้น โดยผลลัพธ์ที่ได้จากการใช้ Multiple Linear Regression พบว่าค่าความคลาดเคลื่อนสูงสุดอยู่ที่ประมาณ 0.0127 เมตร ซึ่งต่ำกว่าเดิมเมื่อเทียบกับ Linear Regression แบบเส้นตรง โมเดลนี้สามารถปรับค่าตำแหน่งทั้งแกน X และ Y พร้อมกัน ทำให้ได้สมการที่ซับซ้อนและแม่นยำยิ่งขึ้น ซึ่งช่วยเพิ่มความน่าเชื่อถือและความแม่นยำในระบบการวัดตำแหน่ง



รูปที่ 4.12 กราฟเปรียบเทียบตำแหน่งจริงและตำแหน่งที่ถูกปรับปรุงจากค่าที่วัดได้จากการตรวจจับวัตถุ (Multiple Linear Regression)

ผลการวัดตำแหน่งหลังจากการตรวจจับและติดตามวัตถุของระบบในหัวข้อ 3.2.5 การระบุตำแหน่งของรถอัตโนมัติในระบบนี้การระบุตำแหน่งของรถอัตโนมัติใช้เซ็นเซอร์ LiDAR และ IMU เพื่อประเมินพิกัดของรถในพื้นที่การทดลอง เซ็นเซอร์เหล่านี้ช่วยในการสร้างตำแหน่งเชิงพื้นที่ที่แม่นยำของรถอัตโนมัติ แม้ว่าระบบนี้จะมีความท้าทายมากกว่าระบบกล้อง เนื่องจากรถอัตโนมัติมีการเคลื่อนที่ตลอดเวลา แต่เซ็นเซอร์ LiDAR สามารถประเมินพิกัดที่แน่นอนได้โดยอาศัยข้อมูลเชิงลึกของสภาพแวดล้อมที่มีการเปลี่ยนแปลง ทำให้ระบบสามารถตรวจจับตำแหน่งของรถได้แม่นยำและสอดคล้องกับระบบกล้องในหลายสถานการณ์

ตารางที่ 4.7 เปรียบเทียบของตำแหน่งจริงและตำแหน่งที่วัดได้ของการระบุตำแหน่งของรถอัตโนมัติ

| Real position |     | Measured position |      |
|---------------|-----|-------------------|------|
| x             | y   | x                 | y    |
| -0.5          | 0.5 | -0.49             | 0.51 |
| 0             | 0.5 | 0.01              | 0.51 |
| 0.5           | 0.5 | 0.51              | 0.51 |
| -0.5          | 0   | -0.49             | 0.01 |
| 0             | 0   | 0.01              | 0.01 |
| 0.5           | 0   | 0.51              | 0.01 |

ตารางที่ 4.7 เปรียบเทียบของตำแหน่งจริงและตำแหน่งที่วัดได้ของการระบุตำแหน่งของรถอัตโนมัติ (ต่อ)

| Real position |      | Measured position |        |
|---------------|------|-------------------|--------|
| x             | y    | x                 | y      |
| -0.5          | -0.5 | -0.49             | -0.49  |
| 0             | -0.5 | 0.01              | -0.49  |
| 0.5           | -0.5 | 0.51              | -0.49  |
| -0.5          | -1   | -0.49             | -0.99  |
| 0             | -1   | 0.01              | -0.99  |
| 0.5           | -1   | 0.51              | -0.99  |
| -0.5          | -1.5 | -0.49             | -1.49  |
| 0             | -1.5 | 0.01              | -1.49  |
| 0.5           | -1.5 | 0.51              | -1.49  |
| -0.5          | -2   | -0.475            | -1.975 |
| 0             | -2   | 0.015             | -2     |
| 0.5           | -2   | 0.5               | -1.975 |
| -0.5          | -2.5 | -0.465            | -2.5   |
| 0             | -2.5 | 0.015             | -2.5   |
| 0.5           | -2.5 | 0.515             | -2.5   |
| -0.5          | -3   | -0.5              | -2.96  |
| 0             | -3   | 0.015             | -3     |
| 0.5           | -3   | 0.53              | -3     |
| 0.5           | 0    | 0.515             | 0.015  |
| 1             | 0    | 1.015             | 0.015  |
| 1.5           | 0    | 1.515             | 0.015  |
| 2             | 0    | 2.015             | 0.015  |
| 2.5           | 0    | 2.515             | 0.015  |
| 3             | 0    | 3.015             | 0.015  |
| 0             | 0.5  | 0.015             | 0.515  |
| 0.5           | 0.5  | 0.515             | 0.515  |
| 1             | 0.5  | 1.015             | 0.515  |
| 1.5           | 0.5  | 1.515             | 0.515  |
| 2             | 0.5  | 2                 | 0.5    |
| 2.5           | 0.5  | 2.5               | 0.525  |



ตารางที่ 4.7 เปรียบเทียบของตำแหน่งจริงและตำแหน่งที่วัดได้ของการระบุตำแหน่งของรถอัตโนมัติ (ต่อ)

| Real position |     | Measured position |       |
|---------------|-----|-------------------|-------|
| x             | y   | x                 | y     |
| 3             | 0.5 | 3.025             | 0.5   |
| 0             | 1   | 0                 | 1.035 |
| 0.5           | 1   | 0.53              | 1     |
| 1             | 1   | 1                 | 1.01  |
| 1.5           | 1   | 1.51              | 1     |
| 2             | 1   | 2                 | 1.02  |
| 2.5           | 1   | 2.52              | 1     |
| 3             | 1   | 3                 | 1.015 |
| 0             | 1.5 | 0.015             | 1.5   |
| 0.5           | 1.5 | 0.515             | 1.5   |
| 1             | 1.5 | 1.015             | 1.5   |
| 1.5           | 1.5 | 1.5               | 1.525 |
| 2             | 1.5 | 2                 | 1.525 |
| 2.5           | 1.5 | 2.5               | 1.525 |
| 3             | 1.5 | 3                 | 1.525 |

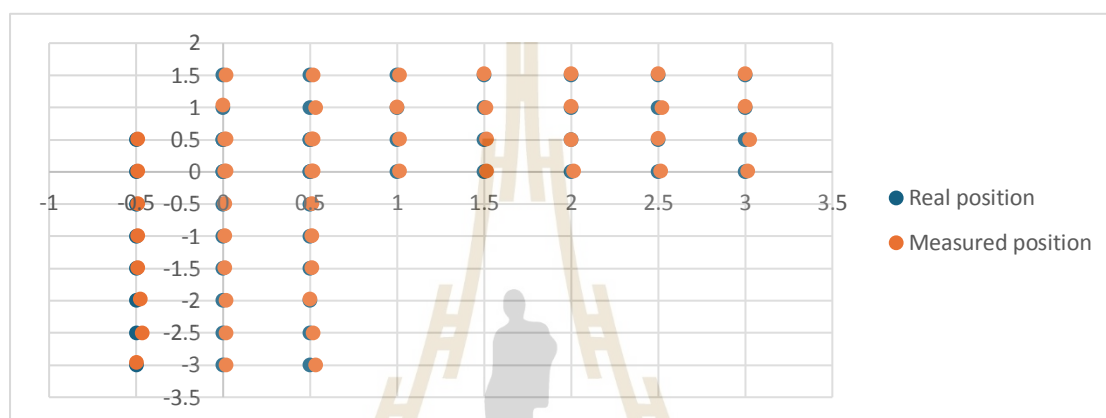
จากการทดสอบการระบุตำแหน่งของรถอัตโนมัติที่ใช้เซ็นเซอร์ LiDAR และ IMU เพื่อวัดตำแหน่งของรถในพื้นที่การทดลอง พบว่าระบบสามารถระบุตำแหน่งได้อย่างแม่นยำในระดับที่ยอมรับได้ แม้ว่าจะมีการเคลื่อนที่ตลอดเวลา ซึ่งเซ็นเซอร์ทั้งสองทำงานร่วมกันเพื่อตรวจจับพิกัดของรถในแนวแกน X และ Y โดยผลการวัดตำแหน่งเปรียบเทียบกับตำแหน่งจริงได้ถูกนำเสนอในตารางการเปรียบเทียบระหว่างตำแหน่งจริง (Real Position) และตำแหน่งที่วัดได้ (Measured Position)

ความแม่นยำของเซ็นเซอร์ LiDAR และ IMU: เซ็นเซอร์ LiDAR และ IMU สามารถระบุตำแหน่งของรถได้ใกล้เคียงกับตำแหน่งจริง โดยเฉพาะในระยะใกล้หรือการเคลื่อนที่ในแนวเส้นตรง ค่าเฉลี่ยของค่าความคลาดเคลื่อนของตำแหน่งอยู่ที่ประมาณ 0.01 - 0.03 เมตร ซึ่งเป็นค่าความคลาดเคลื่อนที่ยอมรับได้สำหรับการทดสอบในสภาพแวดล้อมที่มีการเคลื่อนที่ของวัตถุ

การวัดตำแหน่งในแนวนอน (X-axis): ในการตรวจสอบตำแหน่งตามแกน X พบว่าตำแหน่งที่วัดได้มีความคลาดเคลื่อนเล็กน้อย โดยความคลาดเคลื่อนส่วนใหญ่จะอยู่ในช่วงประมาณ

0.01 - 0.03 เมตร แสดงถึงประสิทธิภาพที่ดีของเซ็นเซอร์ในการตรวจจับตำแหน่งตามแนวแกน X ซึ่งถือว่าเป็นตำแหน่งที่วัดได้ค่อนข้างแม่นยำ

การวัดตำแหน่งในแนวตั้ง (Y-axis): ผลการวัดในแนวตั้งหรือแกน Y ของการเคลื่อนที่แสดงให้เห็นว่าค่าความคลาดเคลื่อนอยู่ในช่วงประมาณ 0.01 - 0.025 เมตร การตรวจจับตำแหน่งในแนวตั้งนี้ยังคงความแม่นยำได้ดีเช่นเดียวกับแนวนอน แสดงถึงการทำงานที่สอดคล้องกันของทั้งสองแกน



รูปที่ 4.13 กราฟเปรียบเทียบตำแหน่งจริงและตำแหน่งที่วัดได้จากการระบุตำแหน่งของรถอัตโนมัติ

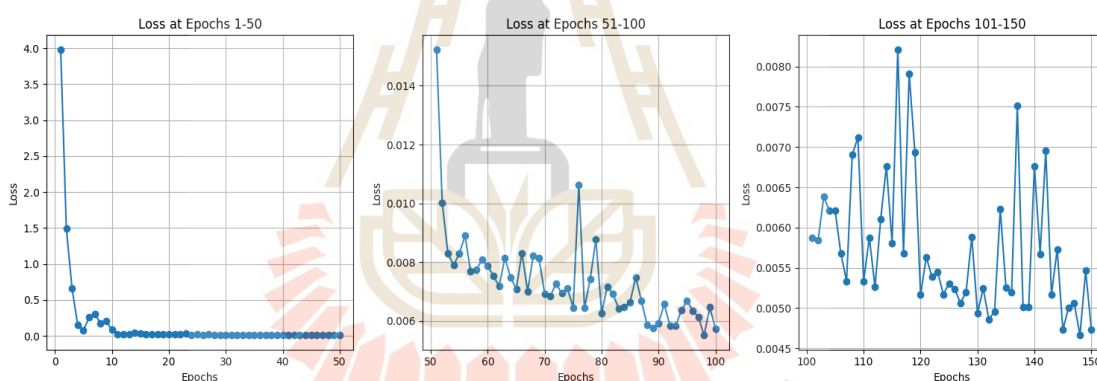
จากกราฟที่แสดงการเปรียบเทียบระหว่างตำแหน่งจริง (Real position) และตำแหน่งที่วัดได้ (Measured position) ของการระบุตำแหน่งรถอัตโนมัติ พบว่าตำแหน่งที่วัดได้มีความแม่นยำสูง โดยจุดที่แทนค่าตำแหน่งจริงและตำแหน่งที่วัดได้ในกราฟมีความใกล้เคียงกันอย่างมากในทุกจุดที่ทำการทดสอบ นี่แสดงให้เห็นถึงประสิทธิภาพของระบบในการประมวลผลและระบุตำแหน่งรถอัตโนมัติในทุกจุดของการทดสอบ ทั้งในแกน X และแกน Y ระบบสามารถรักษาความแม่นยำในการวัดตำแหน่งได้อย่างสม่ำเสมอ จุดที่แสดงค่าตำแหน่งจริงและตำแหน่งที่วัดได้จากเซ็นเซอร์แสดงให้เห็นถึงการสอดคล้องกันในทุกการเคลื่อนไหวของรถอัตโนมัติ ไม่ว่าจะเป็นตำแหน่งที่ใกล้กับจุดเริ่มต้นหรือจุดที่ห่างออกไป ความคลาดเคลื่อนที่เกิดขึ้นอยู่ในระดับที่สามารถมองข้ามได้ และไม่มีผลกระทบต่อการทำงานของระบบจากการวิเคราะห์ผลการทดสอบในกราฟนี้ ระบบการระบุตำแหน่งของรถอัตโนมัติมีความแม่นยำและเสถียรพอที่จะนำไปใช้งานในสถานการณ์จริงได้ โดยไม่จำเป็นต้องทำการปรับปรุงหรือแก้ไขเพิ่มเติม ระบบที่ใช้เซ็นเซอร์ LiDAR และ IMU ได้ทำงานร่วมกันอย่างมีประสิทธิภาพ ช่วยให้การระบุตำแหน่งของรถอัตโนมัติแม่นยำและสอดคล้องกับตำแหน่งจริงจากการเปรียบเทียบระหว่างตำแหน่งจริงและตำแหน่งที่วัดได้จากระบบการระบุตำแหน่งของรถอัตโนมัติ แสดงให้เห็นว่าระบบมีความแม่นยำสูงในการวัดและประเมินพิกัดของรถในพื้นที่การทดลอง

ความคลาดเคลื่อนที่เกิดขึ้นน้อยมากและอยู่ในระดับที่สามารถยอมรับได้ ดังนั้น ระบบนี้สามารถนำไปใช้ในงานจริงได้โดยไม่ต้องปรับปรุงเพิ่มเติม

ระบบการระบุตำแหน่งของรถอัตโนมัติที่ใช้เซ็นเซอร์ LiDAR และ IMU สามารถระบุตำแหน่งของรถได้อย่างแม่นยำ โดยมีความคลาดเคลื่อนเล็กน้อยในช่วง 0.01 - 0.03 เมตร ซึ่งถือว่าเป็นผลลัพธ์ที่น่าพอใจสำหรับการใช้งานในระบบอัตโนมัติ การปรับปรุงเพิ่มเติมในอนาคตสามารถทำได้ด้วยการรวมข้อมูลจากหลายเซ็นเซอร์เพื่อลดค่าความคลาดเคลื่อนและเพิ่มประสิทธิภาพของระบบ

#### 4.3.2 ผลการฝึกฝนโมเดล ANN เพื่อทำนายตำแหน่ง

การฝึกฝนโมเดล Artificial Neural Network (ANN) ในการทำนายตำแหน่งของวัตถุเป็นขั้นตอนสำคัญในการพัฒนาระบบอัตโนมัติ เพื่อให้ระบบสามารถคาดการณ์ตำแหน่งถัดไปของวัตถุหรือรถอัตโนมัติได้อย่างแม่นยำ โมเดล ANN ได้รับการฝึกฝนด้วยชุดข้อมูลที่ประกอบด้วยตำแหน่งจริงและตำแหน่งที่วัดได้จากระบบเซ็นเซอร์ต่าง ๆ ทั้งกล้อง LiDAR และ IMU เพื่อให้โมเดลเรียนรู้ความสัมพันธ์และรูปแบบระหว่างข้อมูลเหล่านั้นโดยผลลัพธ์จากการฝึกฝน ANN มีดังต่อไปนี้

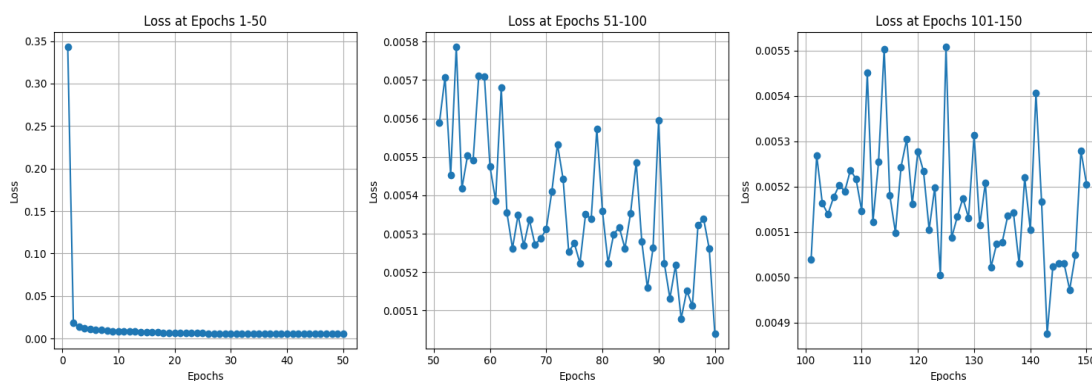


รูปที่ 4.14 กราฟผลลัพธ์การฝึกโมเดลการทำนายตำแหน่งจากกล้อง

ผลการทดสอบโมเดลการทำนายตำแหน่งจากกล้อง (Camera-based Position Prediction Model) ได้แสดงค่า Loss ซึ่งเป็นตัวชี้วัดความผิดพลาดระหว่างค่าที่โมเดลทำนายได้กับค่าจริง โดยโมเดลถูกฝึกฝนในระยะ 150 Epochs ซึ่งค่า Loss ถูกบันทึกเป็นช่วง ๆ เพื่อดูแนวโน้มการเรียนรู้ของโมเดล การลดค่าความสูญเสียเป็นสิ่งสำคัญ เนื่องจากหมายถึงการที่โมเดลสามารถปรับตัวและทำนายตำแหน่งได้แม่นยำยิ่งขึ้นการวิเคราะห์กราฟแบ่งออกเป็น 3 ช่วงคือ Epochs 1-50 กราฟในช่วงนี้เริ่มต้นที่ค่าความสูญเสียสูงถึง 4.0 ใน Epoch ที่ 1 แต่มีแนวโน้มลดลงอย่างรวดเร็วภายในไม่กี่ Epochs แรกเมื่อผ่านไปถึง Epoch ที่ 10 ค่า Loss ลดลงเหลือประมาณ 0.01 และเริ่มมีความคงที่เป็นเส้นตรงในระดับต่ำมากที่ประมาณ 0.001 ในช่วงหลังของการฝึกในช่วงนี้การลดค่าความ

สูญเสียที่รวดเร็วนี้แสดงถึงการที่โมเดลเรียนรู้จากข้อมูลได้อย่างมีประสิทธิภาพในช่วงต้น โดยสามารถปรับการทำนายให้มีความแม่นยำมากขึ้นได้อย่างรวดเร็ว Epochs 51-100 ในช่วงนี้ ค่า Loss มีการแกว่งขึ้นลงบ้างเล็กน้อย แต่โดยรวมยังอยู่ในระดับต่ำ เฉลี่ยประมาณ 0.007 การแกว่งตัวของค่าความสูญเสียในบางช่วงแสดงให้เห็นถึงความท้าทายในการปรับตัวของโมเดลในข้อมูลที่อาจมีลักษณะซับซ้อน แต่ท้ายที่สุดโมเดลก็ยังสามารถรักษาค่าความสูญเสียในระดับต่ำได้ความคงที่ของกราฟและแนวโน้มการลดลงบ่งบอกถึงการเรียนรู้ที่เสถียรและประสิทธิภาพของโมเดล แม้ว่าจะมีบางช่วงที่ค่า Loss เพิ่มขึ้นบ้าง แต่แนวโน้มลดลงแสดงถึงการปรับปรุงการทำนายได้ดีขึ้นเรื่อย ๆ Epochs 101-150 ในช่วงสุดท้ายของการฝึกฝน ค่า Loss ยังคงมีการแกว่งตัวเพิ่มขึ้นในบางช่วง โดยเฉพาะ Epochs 110-130 ที่ค่า Loss ขึ้นไปสูงถึง 0.008 แม้ว่าค่า Loss ในช่วงนี้จะมีความผันผวนมากขึ้น แต่ยังคงอยู่ในระดับที่ต่ำกว่าช่วงแรกของการฝึกฝน และมีแนวโน้มลดลงในช่วงท้ายของ Epoch เมื่อสิ้นสุด Epoch ที่ 150 ค่า Loss ลดลงอยู่ในช่วงประมาณ 0.005 ซึ่งบ่งบอกว่าโมเดลสามารถลดความผิดพลาดได้มากยิ่งขึ้นในช่วงท้ายของการฝึกฝน

จากผลลัพธ์ของการทดสอบโมเดลการทำนายตำแหน่งจากกล้อง (Camera-based Position Prediction Model) มีการใช้ตัวชี้วัดหลายประเภทเพื่อประเมินประสิทธิภาพของโมเดลทั้งใน Mean Squared Error (MSE) ค่า MSE ที่ได้คือ 0.00387 ซึ่งบ่งบอกถึงความแตกต่างระหว่างค่าที่โมเดลทำนายและค่าจริงในรูปแบบการยกกำลังสองของความผิดพลาด (Squared Error) ค่า MSE ยิ่งต่ำยิ่งแสดงถึงความแม่นยำของโมเดลในการทำนายตำแหน่งได้ใกล้เคียงกับค่าจริงมากขึ้น Mean Absolute Error (MAE) ค่า MAE ที่ได้ คือ 0.03969 เป็นการวัดค่าเฉลี่ยของความคลาดเคลื่อนที่ไม่ยกกำลังสอง (Absolute Error) ค่านี้แสดงถึงความแตกต่างโดยเฉลี่ยระหว่างค่าทำนายและค่าจริง ซึ่งเป็นค่าที่เข้าใจง่ายว่าโมเดลทำงานได้ดีซึ่งถือว่าอยู่ในระดับที่ยอมรับได้สำหรับระบบการทำนายตำแหน่งในสถานการณ์จริง Coefficient of Determination ( $R^2$ ) ค่า  $R^2 = 0.99588$  บ่งบอกถึงประสิทธิภาพของโมเดลในการทำนายตำแหน่ง เมื่อค่า  $R^2$  ใกล้เคียงกับ 1 จะหมายถึงว่าโมเดลสามารถทำนายได้สอดคล้องกับข้อมูลจริงมาก ค่า  $R^2$  ที่สูงถึง 0.99588 แสดงว่าโมเดลสามารถทำนายตำแหน่งได้อย่างแม่นยำและใกล้เคียงกับข้อมูลจริงมากกว่า 99% โมเดลการทำนายตำแหน่งจากกล้องที่ถูกทดสอบนี้มีความแม่นยำในการทำนายค่อนข้างสูง ค่า MSE และ MAE ที่ต่ำ บ่งชี้ถึงความสามารถในการทำนายตำแหน่งได้ใกล้เคียงกับข้อมูลจริง และค่า  $R^2$  ที่ใกล้เคียงกับ 1 ยืนยันว่าโมเดลนี้มีความสามารถในการคาดการณ์ได้อย่างดีเยี่ยมในสภาพแวดล้อมที่มีการเคลื่อนไหว



รูปที่ 4.15 กราฟผลลัพธ์การฝึกโมเดลการทำนายตำแหน่งของรถอัตโนมัติ

ผลการทดสอบโมเดลการทำนายตำแหน่งของรถอัตโนมัติ (Autonomous Vehicle Position Prediction Model) ได้แสดงค่า Loss ซึ่งเป็นตัวชี้วัดความผิดพลาดระหว่างค่าที่โมเดลทำนายได้กับค่าจริง โดยโมเดลถูกฝึกฝนในระยะ 150 Epochs ซึ่งค่า Loss ถูกบันทึกเป็นช่วง ๆ เพื่อดูแนวโน้มการเรียนรู้ของโมเดล การลดค่าความสูญเสียเป็นสิ่งสำคัญ เนื่องจากหมายถึงการที่โมเดลสามารถปรับตัวและทำนายตำแหน่งได้แม่นยำยิ่งขึ้นการวิเคราะห์กราฟแบ่งออกเป็น 3 ช่วงคือ ช่วง Epochs 1-50 กราฟแสดงให้เห็นว่า Loss ลดลงอย่างรวดเร็วในช่วงเริ่มต้น โดยจากค่าที่สูงกว่า 0.35 ลดลงมาเหลือประมาณ 0.005 ภายใน 10 epochs แรก หลังจากนั้นค่า Loss ค่อย ๆ คงที่และต่ำลง โดยมีค่าใกล้เคียงกับ 0.0025 แสดงถึงการเรียนรู้ที่ดีในช่วงเริ่มต้นของโมเดลการลดลงของ Loss อย่างรวดเร็วในช่วงต้นเป็นสัญญาณบ่งชี้ว่าโมเดลสามารถเรียนรู้จากข้อมูลได้ดี ช่วง Epochs 51-100 ในช่วงนี้ กราฟมีความผันผวนขึ้นลง แต่ยังคงมีแนวโน้มลดลงอย่างต่อเนื่อง แม้ว่าจะมีความแปรปรวนบ้าง แต่ค่า Loss มีค่าใกล้เคียงระหว่าง 0.0057 ถึง 0.0051 ในช่วงสุดท้ายของ Epoch นี้ ความผันผวนของ Loss ในช่วงนี้อาจมาจากการที่โมเดลพยายามหาจุดสมดุลในการเรียนรู้ข้อมูลใหม่ และข้อมูลที่เคยเรียนรู้ ช่วง Epochs 101-150 ช่วงนี้ Loss ยังคงมีความผันผวนเล็กน้อย แต่โดยรวมแล้วค่า Loss อยู่ในระดับต่ำมาก โดยเฉลี่ยอยู่ที่ 0.005 ซึ่งเป็นค่าที่ต่ำกว่าช่วงต้นๆ แสดงถึงการปรับปรุงที่ดีขึ้นในโมเดลโมเดลดูเหมือนจะเข้าสู่ช่วงคงที่ (Convergence) แม้ว่าจะมีความแปรปรวนอยู่บ้าง แต่ค่า Loss ต่ำแสดงว่าโมเดลมีการปรับให้ทำนายผลได้ดีจากกราฟแสดงให้เห็นว่าโมเดลการทำนายตำแหน่งจากรถอัตโนมัติมีการเรียนรู้ที่ดีในช่วงต้น และค่อย ๆ ลดค่า Loss อย่างต่อเนื่อง ซึ่งแสดงถึงประสิทธิภาพของการทำนายตำแหน่งที่แม่นยำมากขึ้น แม้จะมีความผันผวนของค่า Loss ในช่วงหลัง แต่ค่าเฉลี่ยของ Loss ยังคงอยู่ในระดับต่ำ ซึ่งบ่งบอกว่าโมเดลสามารถทำนายตำแหน่งของรถอัตโนมัติได้อย่างแม่นยำ



จากผลการประเมินโมเดลการทำนายตำแหน่งของรถอัตโนมัติ (Autonomous Vehicle Position Prediction Model) มีการใช้ตัวชี้วัดหลายประเภทเพื่อประเมินประสิทธิภาพของโมเดลทั้งใน Mean Squared Error (MSE) ค่า MSE บ่งบอกถึงความคลาดเคลื่อนโดยเฉลี่ยระหว่างค่าที่โมเดลทำนายและค่าจริง เป็นการวัดขนาดของความผิดพลาดในการทำนาย โดยค่าที่ได้คือ 0.0058 หมายความว่าโดยเฉลี่ยแล้วโมเดลมีความคลาดเคลื่อนอยู่ที่ประมาณ 0.0058 หน่วย ซึ่งในบริบทของการทำนายตำแหน่งเป็นความคลาดเคลื่อนที่น้อย แสดงว่าโมเดลสามารถทำนายตำแหน่งได้ค่อนข้างแม่นยำ Mean Absolute Error (MAE) ค่า MAE บอกถึงค่าเฉลี่ยของความแตกต่างระหว่างค่าที่ทำนายและค่าจริงในแต่ละครั้ง โดยไม่คำนึงถึงทิศทางความคลาดเคลื่อน ซึ่งในกรณีนี้ ค่า MAE ที่ได้คือ 0.0493 หมายความว่าความคลาดเคลื่อนนี้เป็นความคลาดเคลื่อนที่อยู่ในระดับที่ยอมรับได้ในกรณีของการทำนายตำแหน่งซึ่งแสดงว่าโมเดลทำงานได้ดีในการทำนายตำแหน่งของรถอัตโนมัติ  $R^2$  ค่า  $R^2$  หรือค่า Coefficient of Determination เป็นการวัดว่าโมเดลสามารถอธิบายความแปรปรวนในข้อมูลได้ดีเพียงใด ค่า  $R^2$  ที่สูงถึง 0.9813 แสดงว่าโมเดลสามารถอธิบายความแปรปรวนของข้อมูลได้ถึงประมาณ 98.13% ซึ่งเป็นตัวชี้วัดที่ดีมาก บ่งบอกว่าโมเดลมีความแม่นยำสูงในการทำนายตำแหน่งของรถอัตโนมัติจากผลการประเมินโมเดลการทำนายตำแหน่งของรถอัตโนมัติ ค่า MAE อยู่ที่ 0.0493 ซึ่งเป็นความคลาดเคลื่อนที่ต่ำและยอมรับได้ ค่า  $R^2$  ที่สูงถึง 98.13% บ่งบอกถึงความแม่นยำที่ดีของโมเดล โมเดลสามารถทำนายตำแหน่งของรถอัตโนมัติได้อย่างมีประสิทธิภาพ

#### 4.4 ผลการทดสอบการชะลอความเร็วและหยุดเมื่อเสี่ยงเกิดการชน

การทดสอบระบบการชะลอความเร็วและหยุดเมื่อเสี่ยงเกิดการชนเป็นส่วนสำคัญในกระบวนการพัฒนารถอัตโนมัติหรือระบบที่ต้องการความปลอดภัยสูง ระบบนี้ต้องสามารถตรวจจับวัตถุที่อยู่ข้างหน้าได้อย่างแม่นยำ และประมวลผลการตัดสินใจในการชะลอความเร็วหรือหยุดรถได้ทันทีเพื่อป้องกันการชนระบบได้รับข้อมูลจากเซ็นเซอร์ กล้อง, LiDAR, และ IMU (Inertial Measurement Unit) ซึ่งช่วยให้สามารถตรวจจับวัตถุและระบุตำแหน่งที่แม่นยำได้ เมื่อวัตถุอยู่ในระยะที่เป็นอันตราย ระบบจะประมวลผลตำแหน่งและความเร็วในปัจจุบันของรถพร้อมกับตำแหน่งและความเร็วของวัตถุจากนั้นทำการตัดสินใจชะลอความเร็วหรือหยุดรถหากจำเป็น



ตารางที่ 4.8 แสดงผลการทดสอบการชะลอความเร็วและหยุดเมื่อเสี่ยงต่อการชนในสถานการณ์ต่าง ๆ

| Test Case | สถานการณ์ทดสอบ                        | ระยะห่างจากวัตถุ (เมตร) | เวลาตอบสนองของระบบ (วินาที) | ระยะเบรก (เมตร) | สถานะการหยุดรถ | ความเร็วรถอัตโนมัติขณะทดสอบ (m/s) |
|-----------|---------------------------------------|-------------------------|-----------------------------|-----------------|----------------|-----------------------------------|
| 1         | วัตถุอยู่นิ่ง (ตรงหน้า)               | 1.3                     | 0.3                         | 1               | หยุดสำเร็จ     | 2                                 |
| 2         | วัตถุอยู่นิ่ง (ด้านข้าง)              | 1.1                     | 0.4                         | 0.9             | หยุดสำเร็จ     | 2.5                               |
| 3         | วัตถุเคลื่อนที่แบบคงที่ (ความเร็วต่ำ) | 1.5                     | 0.25                        | 1.2             | หยุดสำเร็จ     | 2.3                               |
| 4         | วัตถุเคลื่อนที่แบบคงที่ (ความเร็วต่ำ) | 1                       | 0.5                         | 0.8             | หยุดสำเร็จ     | 2                                 |
| 5         | วัตถุอยู่นิ่งในสภาพแสงน้อย            | 1.7                     | 0.35                        | 1.3             | หยุดสำเร็จ     | 2.1                               |
| 6         | วัตถุเคลื่อนที่แบบคงที่ (ความเร็วต่ำ) | 1.4                     | 0.2                         | 1.1             | หยุดสำเร็จ     | 2.2                               |
| 7         | วัตถุเคลื่อนที่แบบคงที่ (ความเร็วต่ำ) | 1.2                     | 0.45                        | 0.95            | หยุดสำเร็จ     | 2.3                               |
| 8         | วัตถุเคลื่อนที่แบบคงที่ (ความเร็วสูง) | 1                       | 0.3                         | 0.9             | หยุดสำเร็จ     | 2.1                               |
| 9         | สภาพแวดล้อมมีน้ำบนพื้น                | 1.4                     | 0.4                         | 1               | หยุดสำเร็จ     | 2.2                               |
| 10        | วัตถุเคลื่อนที่แบบคงที่ (ความเร็วต่ำ) | 1.3                     | 0.3                         | 1.1             | หยุดสำเร็จ     | 2.3                               |

จากการทดสอบการชะลอความเร็วและหยุดรถอัตโนมัติในสถานการณ์ต่าง ๆ ทั้งหมด 10 กรณีในตารางที่แสดงผล พบว่าระบบสามารถตอบสนองต่อการชนได้อย่างมีประสิทธิภาพโดยสามารถหยุดรถได้ทันก่อนเกิดการชนในทุกกรณีที่ทำทดสอบ ทั้งนี้ ความแม่นยำและประสิทธิภาพในการหยุดรถของระบบสามารถวิเคราะห์ได้จากปัจจัยหลายประการ ได้แก่ ระยะห่างจากวัตถุ เวลาตอบสนองของระบบ ระยะเบรก ความเร็วของรถขณะทำการทดสอบ และสถานการณ์หยุดรถ

การทดสอบระยะห่างจากวัตถุตั้งแต่ 1 เมตร ถึง 1.7 เมตร พบว่าระยะห่างจากวัตถุมีผลต่อระยะเบรก โดยระยะห่างจากวัตถุมากขึ้นจะทำให้รถมีโอกาสหยุดรถได้ปลอดภัยมากขึ้น เช่น กรณีทดสอบที่ 1 และ 5 ซึ่งระยะห่างจากวัตถุอยู่ที่ 1.3 เมตรและ 1.7 เมตรตามลำดับ ระบบสามารถประมวลผลและหยุดรถได้อย่างมีประสิทธิภาพ โดยเวลาตอบสนองของระบบอยู่ที่ 0.3 วินาที และ 0.35 วินาทีตามลำดับ ส่งผลให้รถหยุดได้ในระยะ 1 เมตรและ 1.3 เมตร ซึ่งใกล้เคียงกับระยะห่างจากวัตถุ

ระบบเวลาตอบสนองของระบบเป็นปัจจัยที่สำคัญในการวัดความสามารถของระบบในการหลีกเลี่ยงการชน เวลาตอบสนองที่ดีมักอยู่ในช่วง 0.2 ถึง 0.45 วินาที ซึ่งจากผลการทดสอบจะเห็นว่าเวลาตอบสนองสั้นที่สุดเกิดขึ้นในกรณีที่วัตถุเคลื่อนที่ด้วยความเร็วต่ำ (Test Case 6) โดยใช้เวลาในการตอบสนองเพียง 0.2 วินาที และรถสามารถหยุดได้ในระยะ 1.1 เมตร ในกรณีที่มีเวลาตอบสนองยาวที่สุดคือ Test Case 7 ที่ใช้เวลา 0.45 วินาที แม้ว่าจะมีระยะห่างจากวัตถุเพียง 1.2 เมตร แต่ระบบก็สามารถหยุดรถได้ทันเวลาโดยไม่เกิดการชน

ระยะเบรกมีความสอดคล้องกับระยะห่างจากวัตถุและเวลาตอบสนองของระบบ ตัวอย่างเช่น ในกรณีที่ระยะห่างจากวัตถุเท่ากับ 1.5 เมตร (Test Case 3) ระบบสามารถหยุดรถได้ในระยะ 1.2 เมตรด้วยเวลาตอบสนอง 0.25 วินาที ซึ่งแสดงให้เห็นถึงการชะลอความเร็วและหยุดรถได้ทันทั่วทั้ง ในกรณีที่สภาพแวดล้อมมีปัจจัยที่อาจทำให้เกิดการลื่น เช่น ใน Test Case 9 ซึ่งมีน้ำบนพื้นถนน ระยะเบรกอยู่ที่ 1 เมตรจากระยะห่างจากวัตถุที่ 1.4 เมตร ซึ่งแสดงให้เห็นว่าระบบยังคงมีความสามารถในการหยุดรถแม้ว่าจะมีสภาพแวดล้อมที่ไม่เอื้ออำนวย

ผลการทดสอบทั้งหมดแสดงให้เห็นว่าระบบสามารถหยุดรถได้สำเร็จในทุกกรณี แม้ว่าจะมีการเปลี่ยนแปลงของปัจจัยต่าง ๆ ระยะห่างจากวัตถุ สภาพแสง และสภาพแวดล้อม โดยไม่มีกรณีใดที่รถไม่สามารถหยุดได้ก่อนการชน สิ่งนี้แสดงให้เห็นถึงความสามารถของระบบในการตรวจจับวัตถุและประมวลผลการหยุดรถได้อย่างรวดเร็วและแม่นยำ

ความเร็วของรถอัตโนมัติที่ใช้ในการทดสอบอยู่ในช่วง 2.0 ถึง 2.5 เมตรต่อวินาที โดยเฉพาะใน Test Case 8 ซึ่งมีความเร็วรถสูงสุดที่ 2.5 เมตรต่อวินาที ระบบสามารถตอบสนองได้อย่างมีประสิทธิภาพ โดยสามารถหยุดรถได้ในระยะ 0.9 เมตร ในขณะที่ระยะห่างจากวัตถุอยู่ที่ 1 เมตร ซึ่งแสดงให้เห็นถึงความสามารถในการประมวลผลและชะลอความเร็วในเวลาอันสั้น

จากผลการทดสอบพบว่าระบบการชะลอความเร็วและหยุดรถอัตโนมัติเมื่อมีความเสี่ยงเกิดการทำงานได้อย่างมีประสิทธิภาพในทุกกรณีที่ทำการทดสอบ ไม่ว่าจะเป็นการหยุดในสภาพแวดล้อมที่แสงน้อย การหยุดในสภาพพื้นเปียกหรือการหยุดเมื่อมีวัตถุเคลื่อนที่ด้วยความเร็วต่ำและความเร็วสูง การที่ระบบสามารถตอบสนองได้รวดเร็วในระยะเวลาที่สั้นและสามารถหยุดรถได้ทันเวลาทำให้ลดความเสี่ยงในการชนได้อย่างมีนัยสำคัญ การทดสอบในกรณีที่มีการเปลี่ยนแปลงสภาพแวดล้อมการมีน้ำบนพื้นหรือสภาพแสงน้อย พบว่าระบบยังคงสามารถหยุดรถได้อย่างปลอดภัย



## บทที่ 5

### สรุปและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้ทำการพัฒนาและทดสอบระบบการตรวจจับและติดตามวัตถุ รวมถึงการทำนายตำแหน่งโดยใช้เทคโนโลยีปัญญาประดิษฐ์ (AI) เพื่อให้ระบบอัตโนมัติมีความสามารถในการตรวจจับวัตถุ ชะลอความเร็ว และหยุดรถเมื่อพบความเสี่ยงจากการชน การพัฒนาและทดสอบระบบนั้นได้ครอบคลุมตั้งแต่การตรวจจับวัตถุจากกล้อง การติดตามวัตถุ และการทำนายตำแหน่งของรถอัตโนมัติ โดยใช้เซ็นเซอร์หลายประเภท

##### 5.1.1 ผลการทดสอบระบบตรวจจับวัตถุ

ระบบตรวจจับวัตถุที่ใช้ปัญญาประดิษฐ์ (AI) สามารถตรวจจับวัตถุได้อย่างมีประสิทธิภาพ โดยในระหว่างการทดสอบ ระบบสามารถตรวจจับวัตถุที่มีทั้งการเคลื่อนไหวและไม่เคลื่อนไหวได้อย่างแม่นยำ โดยกราฟ PR-Confidence และ F1-Confidence Curve แสดงถึงความแม่นยำของระบบในระดับสูง โดยมีค่า F1 สูงถึง 0.98 และค่า Precision สูงถึง 1.00 แสดงถึงความแม่นยำของการตรวจจับวัตถุในระบบ ซึ่งทำให้สามารถตอบสนองต่อสถานการณ์การชนได้อย่างมีประสิทธิภาพ

##### 5.1.2 ผลการทดสอบระบบติดตามวัตถุ

ระบบติดตามวัตถุทำงานได้อย่างแม่นยำ โดยค่า Tracking Accuracy ในการทดสอบส่วนใหญ่อยู่ในช่วง 85% - 95% และสามารถจัดการกับการเปลี่ยนแปลงของการติดตามที่มีความซับซ้อนได้ดี ทั้งในกรณีที่มีการเคลื่อนไหวของวัตถุอย่างชัดเจนและกรณีที่วัตถุถูกบังชั่วคราว การทดสอบแสดงให้เห็นว่า การสลับ ID (ID Switches) และ Lost Tracks ที่เกิดขึ้นมีจำนวนน้อยซึ่งบ่งบอกถึงความน่าเชื่อถือของระบบในการติดตามวัตถุในสถานการณ์จริง

### 5.1.3 ผลการทำนายตำแหน่ง

การปรับปรุงการระบุตำแหน่งของระบบกล้องตรวจจับวัตถุทำให้ความแม่นยำในการทำนายตำแหน่งเพิ่มขึ้นอย่างมาก จากการทดสอบก่อนและหลังการปรับปรุงค่าเฉลี่ยของความคลาดเคลื่อนลดลงจาก 0.03736 เมตร เป็น 0.00905 เมตร หรือประมาณ 75.76% ซึ่งทำให้ระบบสามารถระบุตำแหน่งของวัตถุได้แม่นยำยิ่งขึ้น

ผลการทำนายตำแหน่งโดยใช้โมเดล Artificial Neural Network (ANN) สำหรับระบบกล้องและรถอัตโนมัติแสดงให้เห็นว่าโมเดลมีความแม่นยำในการทำนายสูงมาก ค่า Mean Squared Error (MSE) สำหรับโมเดลการทำนายตำแหน่งจากกล้องอยู่ที่ 0.0038 และสำหรับโมเดลของรถอัตโนมัติอยู่ที่ 0.0058 ซึ่งแสดงถึงการทำนายตำแหน่งที่มีความแม่นยำมาก

### 5.1.4 ผลการทดสอบการชะลอความเร็วและหยุดเมื่อเสี่ยงเกิดการชน

ระบบสามารถตรวจจับและตอบสนองต่อความเสี่ยงในการชนได้อย่างมีประสิทธิภาพ ผลการทดสอบแสดงให้เห็นว่าระบบมีการตอบสนองรวดเร็วต่อสถานการณ์ต่าง ๆ ทั้งในวัตถุที่เคลื่อนที่ด้วยความเร็วต่างกัน วัตถุที่อยู่นิ่ง และสภาพแวดล้อมที่แตกต่างกันไม่ว่าจะเป็นสภาพแสงน้อยหรือการมีน้ำบนพื้นแสดงให้เห็นว่าระบบสามารถหยุดรถได้ทันเวลาก่อนเกิดการชน โดยมีความเร็วตอบสนองเฉลี่ยที่ดีเยี่ยมในระดับ 0.2 - 0.5 วินาที และความเร็วรถขณะทดสอบอยู่ในช่วง 2.0 - 2.5 เมตรต่อวินาที

### 5.1.5 การประยุกต์ใช้ในระบบอัตโนมัติ

การทดสอบระบบทั้งหมดแสดงให้เห็นถึงศักยภาพในการประยุกต์ใช้งานในรถอัตโนมัติที่สามารถนำไปใช้งานได้จริง ทั้งในส่วนของ การตรวจจับ การติดตาม การทำนายตำแหน่ง รวมถึงการชะลอความเร็วและหยุดเมื่อเสี่ยงเกิดการชน ระบบสามารถทำงานได้แม่นยำและตอบสนองได้รวดเร็วในสถานการณ์ต่าง ๆ ซึ่งแสดงถึงความปลอดภัยและความน่าเชื่อถือของระบบในการปฏิบัติงานในสภาพแวดล้อมจริง

งานวิจัยนี้ได้พัฒนาระบบตรวจจับและติดตามวัตถุ การทำนายตำแหน่ง และการชะลอความเร็วและหยุดรถเมื่อเสี่ยงเกิดการชนที่มีประสิทธิภาพสูง โดยการใช้เทคโนโลยีปัญญาประดิษฐ์และเซ็นเซอร์ที่แม่นยำ ทำให้ระบบสามารถทำงานได้ในสภาวะที่หลากหลายและซับซ้อน ทั้งนี้ งานวิจัยสามารถประยุกต์ใช้กับระบบรถอัตโนมัติและระบบอื่น ๆ ที่เกี่ยวข้องกับการหลีกเลี่ยงการชนได้

## 5.2 ข้อเสนอแนะ

จากผลการวิจัยในหัวข้อการพัฒนาระบบตรวจจับ ติดตาม และทำนายตำแหน่งของวัตถุในระบบรถอัตโนมัติ มีข้อเสนอแนะหลายประการที่ควรนำไปพิจารณาเพื่อเพิ่มประสิทธิภาพและขยายขอบเขตการนำไปใช้ในงานที่หลากหลายมากยิ่งขึ้น

### 5.2.1 การพัฒนาความแม่นยำของการตรวจจับและติดตามวัตถุในสถานะที่ซับซ้อนมากขึ้น

จากการทดลอง ระบบสามารถตรวจจับและติดตามวัตถุได้อย่างแม่นยำในสถานการณ์ที่มีสภาวะแวดล้อมปกติ หากนำไปใช้งานในสถานะที่ซับซ้อนมากขึ้น เช่น สภาพแวดล้อมที่มีฝุ่น หมอก หรือวัตถุที่มีการเคลื่อนไหวรวดเร็วมากขึ้น การตรวจจับและติดตามอาจเกิดความผิดพลาด ดังนั้น การฝึกฝนโมเดลด้วยข้อมูลที่หลากหลายและซับซ้อนมากขึ้นเป็นสิ่งจำเป็น เพื่อเพิ่มประสิทธิภาพในการตรวจจับวัตถุในทุกสภาวะ

### 5.2.2 การรวมข้อมูลจากหลายเซ็นเซอร์ (Sensor Fusion)

แม้ว่าการใช้เซ็นเซอร์ LiDAR และกล้องจะให้ผลที่ดีในเรื่องของการตรวจจับและทำนายตำแหน่ง แต่การรวมข้อมูลจากหลายเซ็นเซอร์ (Sensor Fusion) อาจเพิ่มความแม่นยำในการทำนายตำแหน่งได้มากขึ้น โดยการใช้ข้อมูลจากเซ็นเซอร์เพิ่มเติม เช่น GPS, Radar, และ Ultrasonic sensors ซึ่งจะช่วยให้ระบบสามารถรับรู้ตำแหน่งและลักษณะของวัตถุได้แม่นยำขึ้นและเชื่อถือได้มากยิ่งขึ้น

### 5.2.3 การพัฒนาความเร็วในการประมวลผล

ในการทดลอง ระบบสามารถตอบสนองต่อสถานการณ์การชนได้อย่างรวดเร็วในช่วงเวลาที่เหมาะสม (0.2-0.5 วินาที) แต่การนำไปใช้ในสถานการณ์จริงในความเร็วสูง เช่น ในยานพาหนะที่วิ่งบนถนนด้วยความเร็วสูง จะต้องมีการพัฒนาความเร็วในการประมวลผลของระบบให้ดีขึ้นอีก การเพิ่มประสิทธิภาพของการประมวลผลในส่วนของการประมวลผลฮาร์ดแวร์และซอฟต์แวร์จึงเป็นสิ่งที่ควรพิจารณา เพื่อให้ระบบสามารถตัดสินใจได้ในเวลาที่สั้นลงและปลอดภัยมากขึ้น

### 5.2.4 การประเมินผลการทำงานของระบบในสภาวะกลางคืนหรือแสงน้อย

การทดลองนี้ยังคงจำกัดในสภาวะแสงที่เป็นปกติ ดังนั้น ควรมีการทดสอบเพิ่มเติมในสภาวะแสงน้อยหรือการขับขี่ยานพาหนะในเวลากลางคืน เพื่อทดสอบประสิทธิภาพของระบบในการทำงานในช่วงเวลาที่แสงน้อย รวมถึงการพัฒนาระบบให้รองรับการตรวจจับและทำนายตำแหน่งในสถานการณ์เหล่านี้ได้อย่างมีประสิทธิภาพ

### 5.2.5 การนำปัญญาประดิษฐ์มาปรับปรุงการตัดสินใจของระบบ

ในส่วนของการทำนายตำแหน่งแม้ว่าระบบจะทำงานได้ดี แต่การนำ AI หรือ Machine Learning เข้ามาปรับปรุงในส่วนของการตัดสินใจเมื่อพบสถานการณ์เสี่ยง เช่น การ



คำนวณความน่าจะเป็นของการชน (Collision Probability) จะช่วยให้ระบบสามารถตอบสนองได้ดีขึ้นและลดโอกาสในการเกิดอุบัติเหตุได้มากยิ่งขึ้น

### 5.2.6 การปรับปรุงโมเดล ANN

ในการพัฒนาโมเดล ANN สำหรับการทำนายตำแหน่ง พบว่าค่าความคลาดเคลื่อนยังอยู่ในระดับที่น้อย แต่สามารถปรับปรุงโมเดลเพิ่มเติมได้ โดยการเพิ่มจำนวนข้อมูลในการฝึกฝนให้ครอบคลุมทุกสถานการณ์ และปรับโครงสร้างของโมเดลให้ซับซ้อนมากขึ้น เพื่อให้สามารถเรียนรู้ลักษณะการเคลื่อนไหวที่ซับซ้อนได้ดีขึ้น

### 5.2.7 การทดสอบในสถานการณ์จริงและการใช้งานภาคสนาม

ควรมีการทดสอบระบบในสถานการณ์จริงและภาคสนามที่มีความซับซ้อน เช่น การจราจรบนถนนที่มีรถจำนวนมาก หรือสถานที่ที่มีสภาพแวดล้อมที่แตกต่างกัน เช่น ในพื้นที่ที่มีสภาพอากาศแปรปรวน เพื่อให้สามารถประเมินประสิทธิภาพของระบบในสถานการณ์ที่อาจเกิดขึ้นจริงและปรับปรุงให้มีความแม่นยำมากขึ้น

### 5.2.8 การพัฒนาและทดสอบกับวัตถุที่มีลักษณะและขนาดที่แตกต่างกัน

ในงานวิจัยนี้ได้ทำการทดสอบกับวัตถุที่มีขนาดและความเร็วในระดับหนึ่ง การทดสอบในอนาคตควรมีการทดสอบกับวัตถุที่มีขนาดและลักษณะต่างกัน เช่น วัตถุที่มีขนาดเล็กและใหญ่ หรือวัตถุที่เคลื่อนไหวด้วยความเร็วสูง เพื่อให้สามารถปรับแต่งโมเดลให้สามารถตรวจจับวัตถุเหล่านี้ได้อย่างมีประสิทธิภาพมากขึ้น

### 5.2.9 การพัฒนาให้ระบบสามารถทำงานร่วมกับระบบควบคุมการเคลื่อนที่อื่น ๆ

ควรพิจารณาการพัฒนาให้ระบบตรวจจับและติดตามวัตถุสามารถทำงานร่วมกับระบบควบคุมการเคลื่อนที่อื่น ๆ ได้ เช่น ระบบการนำทางอัตโนมัติ (Autonomous Navigation) หรือระบบการหลีกเลี่ยงอุปสรรค (Obstacle Avoidance System) เพื่อให้เกิดการทำงานร่วมกันอย่างมีประสิทธิภาพและปลอดภัยมากยิ่งขึ้น

การวิจัยนี้ได้แสดงให้เห็นถึงความสำเร็จในการพัฒนาระบบตรวจจับและติดตามวัตถุที่มีความแม่นยำสูงควรปรับปรุงความแม่นยำในสถานการณ์ที่ซับซ้อนและการพัฒนาเทคโนโลยีที่เกี่ยวข้อง เช่น การรวมข้อมูลจากหลายเซ็นเซอร์ และการปรับปรุงการทำงานของโมเดลปัญญาประดิษฐ์ จะช่วยให้ระบบนี้มีประสิทธิภาพมากยิ่งขึ้นควรเพิ่มการทดสอบในสภาวะจริงและการทำงานร่วมกับระบบอื่น ๆ จะช่วยเพิ่มความสามารถในการใช้งานในสถานการณ์ที่หลากหลายมากขึ้นและลดความเสี่ยงในการเกิดอุบัติเหตุได้มากยิ่งขึ้น

## รายการอ้างอิง

- Hartley, R., & Zisserman, A. (2003). **Multiple view geometry in computer vision**. Cambridge university press.
- Li, H., Sima, C., Dai, J., Wang, W., Lu, L., Wang, H., ... & Qiao, Y. (2023). **Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe**. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Shirley, P., Ashikhmin, M., & Marschner, S. (2009). **Fundamentals of computer graphics**. AK Peters/CRC Press.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). **You only look once: Unified, real-time object detection**. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). **Yolov4: Optimal speed and accuracy of object detection**. arXiv preprint arXiv:2004.10934.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). **Ssd: Single shot multibox detector**. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). **Deep residual learning for image recognition**. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). **Faster r-cnn: Towards real-time object detection with region proposal networks**. Advances in neural information processing systems, 28.
- Girshick, R. B. (2015). **Fast R-CNN**. CoRR, abs/1504.08083. arXiv preprint arXiv:1504.08083.
- Redmon, J., & Farhadi, A. (2017). **YOLO9000: better, faster, stronger**. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

- Chen, R. C. (2019). **Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning**. *Image and Vision Computing*, 87, 47-56.
- Redmon, J., & Farhadi, A. (2018). **Yolov3: An incremental improvement**. arXiv preprint arXiv:1804.02767.
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). **Yolov4: Optimal speed and accuracy of object detection**. arXiv preprint arXiv:2004.10934.
- Ultralytics. (2020). **YOLOv5. [Online]**. Available: <https://github.com/ultralytics/yolov5>. [Accessed 03-Feb-2024].
- Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., ... & Wei, X. (2022). **YOLOv6: A single-stage object detection framework for industrial applications**. arXiv preprint arXiv:2209.02976.
- Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). **YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors**. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7464-7475).
- Wang, G., Chen, Y., An, P., Hong, H., Hu, J., & Huang, T. (2023). **UAV-YOLOv8: a small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios**. *Sensors*, 23(16), 7190.
- Xiao, F., Wang, H., Xu, Y., & Zhang, R. (2023). **Fruit Detection and Recognition Based on Deep Learning for Automatic Harvesting: An Overview and Review**. *Agronomy*, 13(6), 1625.
- Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., & Chen, H. (2023). **DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor**. *Electronics*, 12(10), 2323.
- Ultralytics. (2023). **YOLOv8. [Online]**. Available: <https://github.com/ultralytics/yolov8>. [Accessed 03-Feb-2024].
- Pereira, R., Carvalho, G., Garrote, L., & Nunes, U. J. (2022). **Sort and deep-SORT based multi-object tracking for mobile robotics: Evaluation with new data association metrics**. *Applied Sciences*, 12(3), 1319.

- Ikomia. (2023). **Deep SORT Object Tracking Guide**. Ikomia Blog. Available at: <https://www.ikomia.ai/blog/deep-sort-object-tracking-guide>. [Accessed on 03-Feb-2024].
- Kuhn, H. W. (1955). **The Hungarian method for the assignment problem**. Naval research logistics quarterly, 2(1-2), 83-97.
- Ziegler, J., Bender, P., Dang, T., & Stiller, C. (2014, June). **Trajectory planning for Bertha—A local, continuous method**. In 2014 IEEE intelligent vehicles symposium proceedings (pp. 450-457). IEEE.
- Poggenhans, F., Pauls, J. H., Janosovits, J., Orf, S., Naumann, M., Kuhnt, F., & Mayr, M. (2018, November). **Lanelet2: A high-definition map framework for the future of automated driving**. In 2018 21st international conference on intelligent transportation systems (ITSC) (pp. 1672-1679). IEEE.
- Ebrahimi Soorchaei, B., Razzaghpour, M., Valiente, R., Raftari, A., & Fallah, Y. P. (2022). **High-Definition Map Representation Techniques for Automated Vehicles**. Electronics, 11(20), 3374.
- Naoki, A., Luis, Y., Eijiro, T., Yuki, Y., & Yoshiki, N. (2017). **Robust Localization Using 3D NDT Scan Matching with Experimentally Determined Uncertainty and Road Marker Matching**. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Los Angeles, California, USA.
- Sijia, L., Jie, L., Jinmin, H., Haoru, L., & Yu, L. (2021). **Research on NDT-based Positioning for Autonomous Driving**. In Proceedings of the 5th International Workshop on Advances in Energy Science and Environment Engineering (AESEE 2021), Volume 257.
- Shirley, P., Ashikhmin, M., & Marschner, S. (2009). **Fundamentals of computer graphics**. AK Peters/CRC Press.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). **ROS: an open-source Robot Operating System**. ICRA workshop on open source software.
- Barber, E. L., Lundsberg, L. S., Belanger, K., Pettker, C. M., Funai, E. F., & Illuzzi, J. L. (2011). **Indications contributing to the increasing cesarean delivery rate**. Obstetrics & Gynecology, 118(1), 29-38.

- Toupet, O., Del Sesto, T., Ono, M., Myint, S., Vander Hook, J., & McHenry, M. (2020). **A ros-based simulator for testing the enhanced autonomous navigation of the mars 2020 rover**. 2020 IEEE Aerospace Conference.
- Hook, J., & McHenry, M. (2020, March). **A ros-based simulator for testing the enhanced autonomous navigation of the mars 2020 rover**. In *2020 IEEE Aerospace Conference* (pp. 1-11). IEEE



The logo of Sakon Nakhon Rajabhat University is a large, faint watermark in the background. It features a central figure of a person sitting on a throne, surrounded by a circular emblem with a crown on top. The entire logo is rendered in a light gray color.

ภาคผนวก ก

บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

มหาวิทยาลัยเทคโนโลยีสุรนารี



## รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

Phinphimai, P., Tantrairatn, S., Ariyarat, A., & Pichitkul, A. (2024, February). **Enhancing Safety and Efficiency with Autonomous Navigation Systems: Integrating Real-Time Object Detection, Tracking, and Trajectory Prediction**. In 2024 16th International Conference on Knowledge and Smart Technology (KST) (pp. 238-241). IEEE.



# Enhancing Safety and Efficiency with Autonomous Navigation Systems: Integrating Real-Time Object Detection, Tracking, and Trajectory Prediction

Poommin Phinphimai  
School of Mechatronics Engineering  
Institute of Engineering  
Suranaree University of Technology  
Nakhon Ratchasima, Thailand  
M6501228@g.sut.ac.th

Suradet Tantrairatn  
School of Mechanical Engineering  
Institute of Engineering  
Suranaree University of Technology  
Nakhon Ratchasima, Thailand  
suradetj@sut.ac.th

Atthaphon Ariyarat  
School of Mechanical Engineering  
Institute of Engineering  
Suranaree University of Technology  
Nakhon Ratchasima, Thailand  
ariyarat@sut.ac.th

Auraluck Pichitkul  
School of Mechanical Engineering  
Institute of Engineering  
Suranaree University of Technology  
Nakhon Ratchasima, Thailand  
auraluck@sut.ac.th

**Abstract**—In this study, an innovative automated navigation system for delivery vehicles was developed, leveraging real-time object detection technology. This system utilizes a bird's-eye view from high-angle cameras at traffic intersections, with the view segmented into a  $7 \times 7$  grid, each grid measuring 0.5 meters. The cutting-edge YOLO (You Only Look Once) V8 algorithm is employed for precise vehicle detection in designated areas, adeptly handling various lighting and weather conditions. A key feature of this system is the use of the DeepSORT algorithm for vehicle tracking. DeepSORT excels in maintaining a consistent identification (ID) of each vehicle, crucial in complex scenarios like vehicle obscuration or close proximity of multiple vehicles. This capability ensures continuous and reliable monitoring of each vehicle's route, particularly vital in heavy traffic situations. For autonomous delivery vehicles, the system integrates real-time data from onboard sensors, ensuring accurate vehicle positioning. This integration allows for precise alignment with the vehicle's actual position. Moreover, the system employs Artificial Neural Networks (ANN) trained on diverse vehicle movement patterns. This training, based on comprehensive real-world vehicle movement data analysis, enables the system to accurately predict future vehicle locations. The system's efficacy is underscored by its high predictive accuracy, as indicated by an impressive determination coefficient ( $R^2$ ) of 0.99. In conclusion, this advanced system integrates object detection, effective tracking, and trajectory prediction capabilities in real-time, offering a robust solution for enhancing the safety and efficiency of transport vehicles in high-traffic areas.

**Index Terms**—Trajectory Prediction, Object Detection and Tracking, Artificial neural network, Autonomous Navigation Systems

## I. INTRODUCTION

The advent of automated navigation systems marks a milestone in the evolution of intelligent transportation. These systems play an important role in shaping the mobility of the



Fig. 1: The operational facet of the trajectory prediction system.

future. It requires advanced technology for precise and efficient operation. This article explores object detection integration. This is an important research that has far-reaching implications in urban traffic management and vehicle safety. The foundation of this research is based on the implementation of a bird's eye view approach using a detailed  $7 \times 7$  grid system for precise vehicle locating. This approach addresses the challenges posed by the ever-changing urban environment. Accurate vehicle detection and tracking are paramount. Using the YOLO V8 model for object detection [1] and the DeepSORT algorithm for tracking [2], it is a new approach to real-time data management with high accuracy and efficiency. At the heart of this study is the use of neural networks. ANN trained in a comprehensive

dataset on vehicle motion patterns, offers insights into the future of predictive analytics in transportation. This approach not only improves the security features of automation. But it also contributes to the broader goal of intelligent traffic management. In addition, The article also delves into the collection and analysis of real-world vehicle movement data, captured through high-angle video footage. This information is at the core of our research. It allows for fine-tuning algorithms to accurately reflect real-world situations. Integration of autonomous robotic navigation systems, which are in line with vehicle tracking systems. It also shows the practicality of our research in a dynamic and unpredictable environment. The performance of the proposed system was rigorously evaluated using standard statistical measures such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared ( $R^2$ ) coefficients. This article aims to support ongoing discussions about smart transit, offering insights and solutions that pave the way for safer and more efficient urban mobility. In this specific study, a system for predicting vehicle trajectories in autonomous navigation is developed, utilizing a Bird's Eye View and a  $7 \times 7$  grid system. The integration of the YOLO V8 model for object detection and the DeepSort algorithm for tracking forms the core of this approach. An Artificial Neural Network (ANN) is crucial for accurate trajectory predictions. Validated with real-world data and statistical measures (MSE, MAE,  $R^2$ ), the trajectory prediction system is depicted in Fig. 1.

## II. PROPOSED SYSTEM DESIGN

An overview of the system is provided by this system. As depicted in Figure 2, warnings are issued by the system when vehicles entering the area are detected, and the trajectory of the detected vehicle, including the direction it will go and its location in a specified number of seconds, can be predicted.

The location of the vehicle to be detected is identified by utilizing the acquired location in the trained ANN model. This model predicts the vehicle's next position by forecasting the next 10 instances ahead. Vehicle behavior data is employed by the ANN model to learn patterns in vehicle movement. The autonomous delivery robot then takes the vehicle's position, which is the  $x, y$  coordinates, and performs the Local to Global transformation using the Homogeneous transformation matrix method [3], as demonstrated in equation (1).

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} S_x & 0 & d_x \\ 0 & S_y & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1)$$

This equation demonstrates how the use is performed by this conversion. Scaling: which uses the  $S_x$  and  $S_y$  values for the  $x$  and  $y$  axes, respectively. Translation: uses the  $d_x$  and  $d_y$  values to convert coordinates from local to global. Given that.  $x, y$ : Local coordinates  $x', y'$ : Global coordinates  $S_x, S_y$ : Magnification factor for  $x$ -axis and  $y$ , respectively.  $d_x, d_y$ : Translation motion values for the  $x$ - and  $y$ -axes, respectively. This matrix is used to convert the coordinates of an object from the local coordinate system to the global coordinate system.

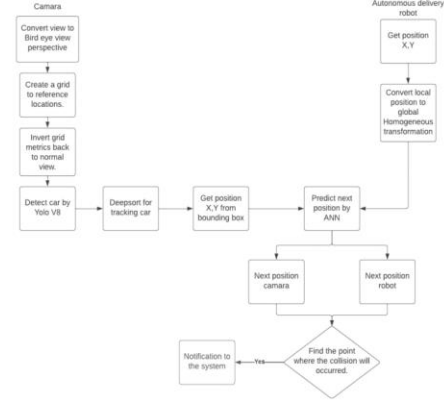


Fig. 2: Flowchart of system design.

Position based on global based on the position of the grid created by the camera side. After the position on the same coordinate, then take the acquired position into the model to predict the next position as well. After getting both positions, you will get  $(x, y)_{\text{camera}}$  belongs to the camera side that detects and acquires the position  $(x, y)_{\text{robot}}$  It belongs to the robot side, after which it brings the two positions  $(x, y)_{\text{camera}}$  and  $(x, y)_{\text{robot}}$  Enter the ANN prediction model and collect 10 points for comparison. Find the distance between them from equation (2).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

The equation is formulated to determine the length of a straight line between two points on a 2D plane, utilizing the principles of the Pythagorean theorem. It is assigned as follows:  $d$  represents the length of the straight line between two points.  $(x_1, y_1)$  denotes the coordinates of the first point.  $(x_2, y_2)$  signifies the coordinates of the second point. This equation calculates the length of the straight line by squaring the difference in coordinates along the  $x$ -axis  $(x_2 - x_1)$  and the  $y$ -axis  $(y_2 - y_1)$ . Then, it sums these squared differences and takes the square root of this sum to find the length of the line. If the distance after prediction is very close and the time of the two positions is closer than 3 seconds, the system will automatically notify the vehicle in advance. If there is a risk of an accident, how many seconds will the collision take place. The system automatically alerts the vehicle in advance of how many seconds a collision will occur. If there is a risk of an accident. Provided that the distance of the point after the prediction is very close, and the time of the two positions is closer than 3 seconds.



### III. METHODOLOGY

#### A. Collect data

This research focuses on gathering real-world vehicle movement data by capturing video camera data into a bird's-eye view, providing a covered space for vehicle detection. The vehicle is detected by the YOLO V8 algorithm, which provides high accuracy and real-time performance required for the traffic environment.

- Detection and Data Recording Table overlays on bird's-eye view facilitate a standardized waypoint system. This ensures consistent vehicle location tracking. X and Y coordinate the detection system of each vehicle at different time intervals. Create datasets that reflect vehicle movement patterns in the area.
- Data Integrity and Preprocessing Before data can be used in an ANN model, it must be cleaned to remove outliers such as excessive data jumps. The coordinates will be normalized to help ANN Train and predict more efficiently. The data obtained is  $x_1, y_1$  as the data from the first coordinate,  $x_2, y_2$  as the data from the second coordinate, and  $x_3, y_3$  as the third coordinate data.

#### B. System Design and Implementation

The research uses a sequential approach to system design. By developing and integrating detection, tracking, and prediction components into a coherent framework that can facilitate robot navigation alongside traditional vehicle

- Vehicle Detection with YOLO V8 [4], [5]. The first phase is about deploying the YOLO V8 algorithm. For vehicle detection, The algorithm's architecture is finely tuned to process images from a bird's eye view. Optimize vehicle identification and locating capabilities within the grid reference system.
- Vehicle Tracking with Deep SORT [1], [2], [6] After detection Deep SORT algorithms are used to track Deep SORT vehicles, consolidate motion data maintain tracking identifiers, and separate individual IDs even in cases of temporary obscuration or proximity of multiple vehicles.
- Autonomous Robot Navigation The automated delivery robot is equipped with sensors that provide real-time location information. This data is converted into the same coordinate using homogeneous transformation to manage on the same system used for vehicle tracking.

#### C. Trajectory Prediction and Collision Avoidance

The focus of the system lies in its ability to predict the location of future vehicles and robots and identify potential collision points in advance.

- ANN for Trajectory Prediction [7]–[14] ANN was developed and trained to predict the future position of a vehicle based on where it collects data on vehicle behavior. ANN's architecture consists of 2 dense layers, each using a linear function, with a total of 4 inputs:  $x_1, x_2, x_3, x_4$ . Given that  $x_1$  is the value  $x$  from the first position,  $x_2$  is

the  $y$  value from the first position,  $x_3$  is the value  $x$  from the second position,  $x_4$  is the  $y$  value from the second position, and there are two outputs of  $y_1, y_2$  each given as follows:  $y_1$  is the  $x$  value from the third position,  $y_2$  is the  $y$  value from the third position.

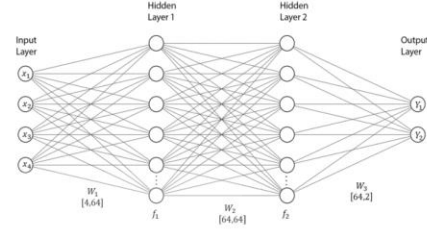


Fig. 3: Proposed ANN Design.

- Performance Evaluation Metrics ANN's predictive performance is quantified using various indicators, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and Determination Coefficient ( $R^2$ ).

### IV. INTEGRATION AND SYSTEM TESTING

The final stage involves the development of sensing, tracking, and prediction models. The overall performance of the system is tested in a simulated environment that closely mimics real-world conditions. Potential collisions are flagged, and the system response here is the accuracy value of the ANN model.

TABLE I: Comparison of Key Performance Metrics Values for ANN Model

| Metric                    | Value    |
|---------------------------|----------|
| Mean Square Error (MSE)   | 3.79E-13 |
| Mean Absolute Error (MAE) | 5.05E-7  |
| R-Square ( $R^2$ )        | 0.99     |

- Mean Squared Error (MSE): 3.79e-13 This value is very close to zero, which suggests that the predicted positions from the ANN are almost identical to the actual positions. The errors between the predicted and actual values are minuscule on average.
- Mean Absolute Error (MAE): 5.06e-07 The MAE being a small number indicates that the average absolute error between the predicted and actual values is extremely low. This means that on average, the predictions are very accurate.
- R-squared ( $R^2$ ): 0.99 The  $R^2$  value is essentially 1, which means the model explains virtually all the variability of the response data around its mean. In other words, the model's predictions match the actual data almost perfectly.

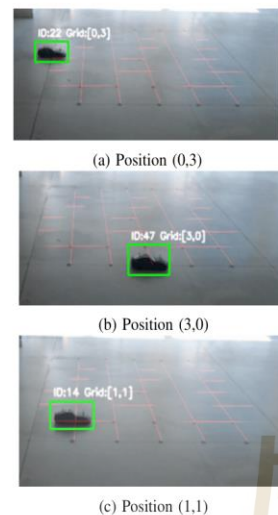


Fig. 4: The Position of the Vehicle as Detected by the System.

The results of the robot location is illustrate in Fig 4.

- From the inspection test of 49 points, it was found that only 4 points were misplaced: (2,5),(2,6),(3,5),(3,6). This means that only 8.16 % of test points misidentified locations, and the remaining 91.84 % of test points indicate the correct location.

#### V. CONCLUSION

In this study, great progress has been made in autonomous navigation, contributing to the expansion of knowledge in the field of intelligent transportation. The YOLO V8 model is used to detect objects in real-time. Integration of DeepSort algorithms for tracking and neural networks For accurate trajectory prediction, it represents a new way to increase vehicle safety and optimize traffic. The use of grid systems derived from positional conversion to Bird's-Eye View has proven to be effective in accurately detecting and tracking vehicles, addressing environmental challenges in a given autonomous vehicle space. The experimental results demonstrate the high efficiency of the proposed automated navigation system in vehicle detection and tracking. The system achieves accuracy in predicting satisfaction based on evidence from statistical measurements. R-squared ( $R^2$ ) 0.99 Test-based system crash alert accuracy (81% accuracy in 100 tests) This may be because the processor or algorithm created cannot keep up with the computational speed sometimes. It uses the processor on an Intel Core i5 gen10 because ANN runs in the CPU. The processor can be improved to have increased computational speed. The inclusion of autonomous robotic navigation systems in this framework demonstrates its practical application.

Real-time alignment of sensor data with vehicle tracking systems has been achieved with a focus on intelligent traffic management. Future research suggests focusing on increasing the scalability of this system for metropolitan areas and improving algorithms in a variety of environments, including transmitting more complex data

#### REFERENCES

- [1] N. S. Pun, S. K. Sonbhadra, S. Agarwal, and G. Rai, "Monitoring covid-19 social distancing with person detection and tracking via fine-tuned yolo v3 and deepsort techniques," *arXiv preprint arXiv:2005.01385*, 2020.
- [2] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, 2023.
- [3] J. J. Craig, *Introduction to robotics*. Pearson Education, 2006.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [5] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "Uav-yolov8: a small-object-detection model based on improved yolov8 for uav aerial photography scenarios," *Sensors*, vol. 23, no. 16, p. 7190, 2023.
- [6] Y. Ge, S. Lin, Y. Zhang, Z. Li, H. Cheng, J. Dong, S. Shao, J. Zhang, X. Qi, and Z. Wu, "Tracking and counting of tomato at different growth period using an improving yolo-deepsort network for inspection robot," *Machines*, vol. 10, no. 6, p. 489, 2022.
- [7] N. Nikhil and B. Tran Morris, "Convolutional neural network for trajectory prediction," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
- [8] G. Debat, T. Chauhan, B. R. Cottier, T. Masquelier, M. Paindavoine, and R. Baures, "Event-based trajectory prediction using spiking neural networks," *Frontiers in computational neuroscience*, vol. 15, p. 658764, 2021.
- [9] D. Singh and R. Srivastava, "Graph neural network with rnns based trajectory prediction of dynamic agents for autonomous vehicle," *Applied Intelligence*, vol. 52, no. 11, pp. 12801–12816, 2022.
- [10] S. Zamboni, Z. T. Kefato, S. Girdzijauskas, C. Norén, and L. Dal Col, "Pedestrian trajectory prediction with convolutional neural networks," *Pattern Recognition*, vol. 121, p. 108252, 2022.
- [11] J. Zaghir, J. F. Rodríguez-Jr, L. Goeuriot, and S. Amer-Yahia, "Real-world patient trajectory prediction from clinical notes using artificial neural networks and umls-based extraction of concepts," *Journal of Healthcare Informatics Research*, vol. 5, no. 4, pp. 474–496, 2021.
- [12] D. Choi, K. Min, and J. Choi, "Regularising neural networks for future trajectory prediction via inverse reinforcement learning framework," *IET Computer Vision*, vol. 14, no. 5, pp. 192–200, 2020.
- [13] Y. Cao, D. Xu, X. Weng, Z. Mao, A. Anandkumar, C. Xiao, and M. Pavone, "Robust trajectory prediction against adversarial attacks," in *Conference on Robot Learning*. PMLR, 2023, pp. 128–137.
- [14] Y. Wang, D. Zhang, Y. Liu, and K.-L. Tan, "Trajectory forecasting with neural networks: An empirical evaluation and a new hybrid model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4400–4409, 2019.

## ประวัติผู้เขียน

นายภูมิมินทร์ พินพิมาย เกิดเมื่อวันที่ 18 เมษายน พ.ศ. 2543 ที่โรงพยาบาลกบินทร์บุรี อำเภอกบินทร์บุรี จังหวัดปราจีนบุรี สำเร็จการศึกษา ชั้นมัธยมต้นและมัธยมปลายที่โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า กบินทร์บุรี จังหวัดปราจีนบุรี ปี พ.ศ. 2561 เข้ารับการศึกษาระดับอุดมศึกษา ณ มหาวิทยาลัยเทคโนโลยีสุรนารี สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำเร็จการศึกษาปี พ.ศ. 2564

ปีพ.ศ. 2565 เข้ารับการศึกษต่อในระดับปริญญาโท ณ มหาวิทยาลัยเทคโนโลยีสุรนารีหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ หัวข้อวิจัยเกี่ยวกับการพัฒนาระบบตรวจจับและคาดการณ์ตำแหน่งรถด้วยระบบกล้องภายนอกเพื่อเชื่อมข้อมูลสำหรับการตัดสินใจให้ยานยนต์อัตโนมัติอย่างปลอดภัยได้รับการสนับสนุนทุน OROG เป็นระยะเวลา 2 ปีในระหว่างการศึกษาได้รับมอบหมายให้เป็นผู้ช่วยสอนในรายวิชา อินเทอร์เน็ตของสรรพสิ่งสำหรับงานวิศวกรรมระบบยานพาหนะภาคพื้นดินที่ขับเคลื่อนด้วยตัวเอง ความรู้เบื้องต้นปัญญาประดิษฐ์ การเรียนรู้ของเครื่องแบบมีผู้สอนและไม่มีผู้สอน มีโอกาสได้ร่วมทำวิจัยกับคณาจารย์ผู้ทรงคุณวุฒิ จนได้รับการเผยแพร่ผลงานจำนวนทั้งสิ้น 1 งาน คือ ผลงานประชุมวิชาการระดับนานาชาติ 1 ฉบับ

- 1) Phinphimai, P., Tantrairatn, S., Ariyarit, A., & Pichitkul, A. (2024, February). Enhancing Safety and Efficiency with Autonomous Navigation Systems: Integrating Real-Time Object Detection, Tracking, and Trajectory Prediction. In 2024 16th International Conference on Knowledge and Smart Technology (KST) (pp. 238-241). IEEE.