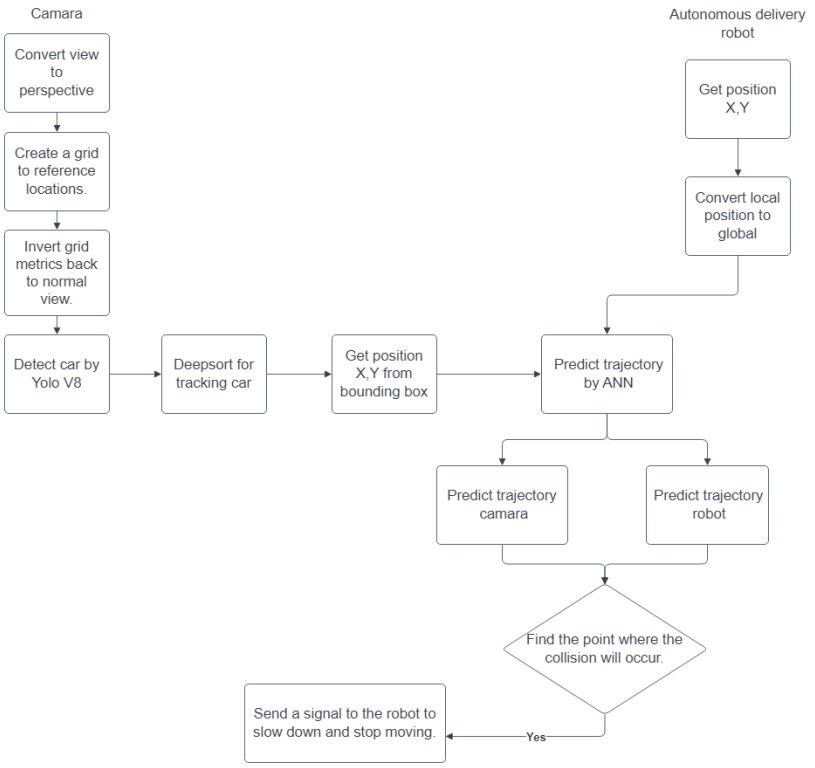
บทที่ 3
วิธีดำเนินงานวิจัย

วิจัยนี้จะสำรวจและอธิบายขั้นตอนต่าง ๆ ที่ผ่านการออกแบบและการทดลองเพื่อสร้างระบบการทำนายและป้องกันการชนระหว่างหุ่นยนต์ส่งของอัตโนมัติกับยานพาหนะในบริเวณทางแยกที่ไม่มีระบบช่วยเหลือหรือเซนเซอร์ประกอบการทำงาน 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 ซึ่งประกอบด้วยการคาดการณ์วิถีของหุ่นยนต์และยานยนต์ในบริเวณเดียวกันเพื่อระบุตำแหน่งที่อาจเกิดการชน 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 ภายใต้

Robot Operating System (ROS) เพื่อพัฒนาระบบที่มีความสามารถในการตอบสนองต่อเหตุการณ์ต่าง ๆ ได้อย่างรวดเร็วและแม่นยำ

3.1.2 การเตรียมพื้นที่ที่จะทำการทดสอบ

พื้นที่ทดสอบที่เลือกสำหรับการทดลองนี้เป็นพื้นที่ภายในที่ปิด ซึ่งจำลองเป็นบริเวณทางแยกเพื่อจำลองสภาพแวดล้อมจริงที่รถอัตโนมัติอาจจะเจอ พื้นที่นี้มีลักษณะเป็นพื้นผิวแบนที่กว้างขวาง ช่วยให้สามารถทำการทดสอบการเคลื่อนที่ของรถยนต์ได้โดยไม่มีข้อจำกัด และเหมาะสมกับการติดตั้งกล้องเพื่อการตรวจจับและการติดตามตำแหน่ง การที่พื้นที่นี้อยู่ภายในยังช่วยลดปัจจัยรบกวนจากภายนอก เช่น แสงแดดที่แปรปรวนหรือสภาพอากาศ ซึ่งอาจมีผลต่อประสิทธิภาพของเซ็นเซอร์และระบบการทำงานของรถอัตโนมัติ

รูปที่ 3.2 ภาพแสดงพื้นที่จำลองในการทำงาน

ในภาพมีรถยนต์ขนาดเล็กที่วางอยู่บนพื้นที่ทดสอบ ซึ่งใช้เป็นตัวแทนในการทดสอบ และกล้องที่ติดตั้งบนขาตั้งที่มีมุมมองที่เหมาะสมเพื่อจับภาพการเคลื่อนไหวของรถยนต์ การจัดตั้งพื้นที่ดังกล่าวช่วยให้การทดลองสามารถควบคุมได้อย่างแม่นยำและเก็บข้อมูลที่จำเป็นได้อย่างสมบูรณ์ สำหรับการวิเคราะห์และการปรับปรุงระบบ

รูปที่ 3.3 ภาพแสดงการจำลองสภาพแวดล้อมจำลองของรถตามท้องถนนทั่วไป

รูปที่ 3.4 ภาพแสดงการจำลองสภาพแวดล้อมจำลองของรถอัตโนมัติ

จากภาพแสดงให้เห็นถึงการจำลองสภาพแวดล้อมที่มุมตึกที่เป็นจุดอับสายตาของระบบอัตโนมัติ

รูปที่ 3.5 พื้นที่ทดสอบที่มีการทำกริดสำหรับทดสอบการนำทางอัตโนมัติ

ภาพที่แสดงมีการทำเครื่องหมายบนพื้นเพื่อสร้างกริดโดยแต่ละจุดห่างกัน 50 เซนติเมตร การตั้งกริดนี้เป็นส่วนสำคัญในการทดสอบเพราะช่วยให้สามารถวัดและติดตามการเคลื่อนที่ของวัตถุในพื้นที่ได้แม่นยำ แต่ละช่องในกริดทำหน้าที่เป็นจุดอ้างอิงสำหรับการตรวจจับและการทำนายตำแหน่ง ซึ่งจำเป็นสำหรับการทดสอบระบบตรวจจับและการทำนายการทำเครื่องหมายเหล่านี้บนพื้นถือเป็นการจำลองสภาพแวดล้อมที่ควบคุมได้ซึ่งเหมาะสมสำหรับการทดสอบระบบการนำทางอัตโนมัติและการทำงานของเซนเซอร์ มันยังช่วยให้สามารถทดสอบอัลกอริทึมต่าง ๆ เช่น YOLO และ Deep Sort ในสภาพแวดล้อมที่จำกัดและเข้าใจได้ง่ายและยังช่วยให้สามารถทำการทดสอบซ้ำได้โดยคงไว้ซึ่งความสม่ำเสมอของตำแหน่งและระยะทางในแต่ละการทดลอง เพื่อเก็บข้อมูลที่สามารถนำไปวิเคราะห์ได้อย่างถูกต้อง

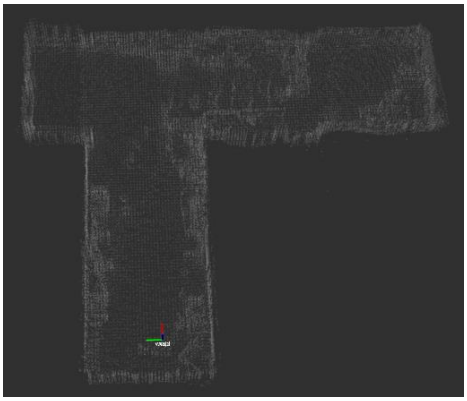
3.1.3 ทดสอบการทำงานของกล้อง

การทดสอบการทำงานของกล้องด้วย OpenCV เป็นกระบวนการที่สำคัญในการพัฒนาระบบวิชันคอมพิวเตอร์เพื่อการตรวจจับและติดตามวัตถุภายในภาพที่ได้จากกล้อง การทดสอบนี้มุ่งเน้นไปที่การประเมินความสามารถของกล้องในการจับภาพที่ชัดเจนและการแปลงภาพที่ได้ให้อยู่ในรูปแบบที่เหมาะสมสำหรับการประมวลผลต่อไป

รูปที่ 3.6 ภาพการทดสอบเปิดกล้องโดยใช้ Python และ ไลบรารี่ OpenCV

3.1.4 ทดสอบการทำงานของแผนที่ความละเอียดสูง

การทดสอบการทำงานของแผนที่ความละเอียดสูงใน rviz เป็นกระบวนการสำคัญที่ช่วยให้นักวิจัยและนักพัฒนาสามารถตรวจสอบและประเมินความเหมาะสมของแผนที่ที่จะใช้ในการนำทางอัตโนมัติสำหรับหุ่นยนต์ภายใน ROS แผนที่ที่มีความละเอียดสูงจำเป็นต้องถูกตรวจสอบอย่างละเอียดเพื่อให้มั่นใจว่ามีความแม่นยำสูง และปราศจากข้อผิดพลาดที่อาจส่งผลกระทบต่อการนำทางของหุ่นยนต์



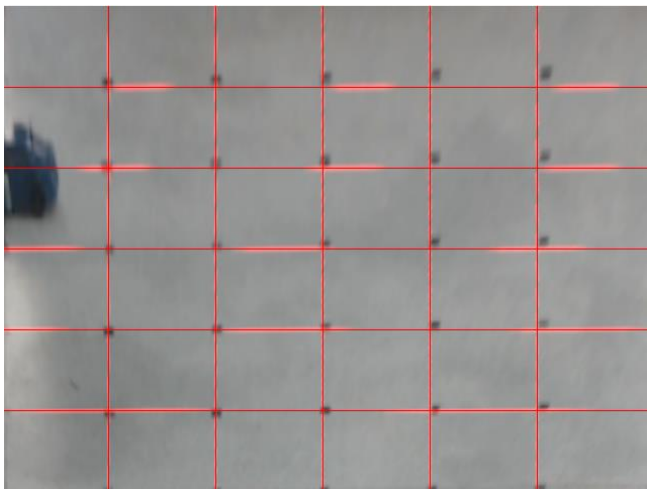
รูปที่ 3.7 ทดสอบการใช้ HD Maps บนซอฟต์แวร์ rviz

3.2 การพัฒนาระบบเพิ่มความปลอดภัยของระบบอัตโนมัติ

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

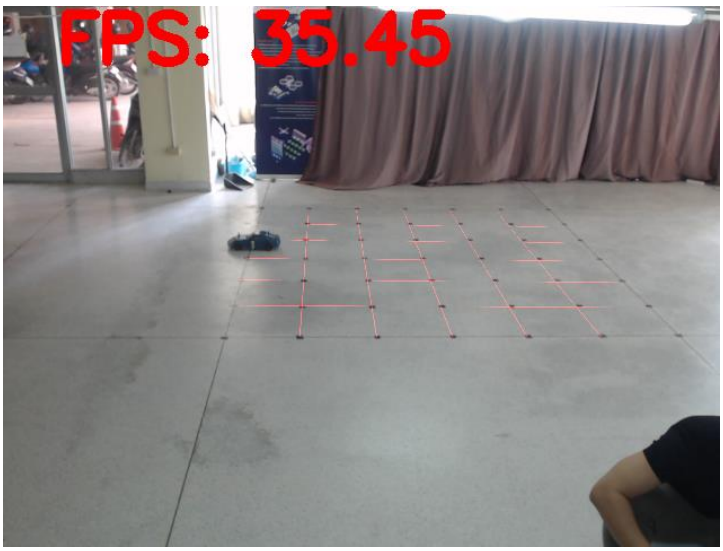
3.2.1 การแปลงมุมมองกล้องและการสร้างตารางเพื่อใช้อ้างอิงตำแหน่ง

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



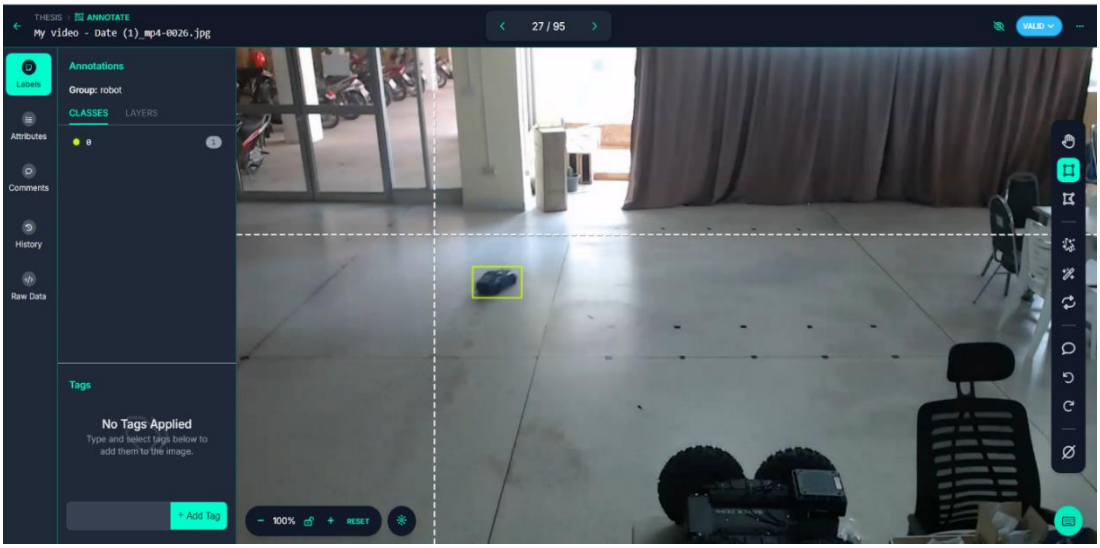
รูปที่ 3.9 เป็นการแปลงรูปภาพ Inverse Perspective Mapping

3.2.2 การตรวจจับวัตถุจากภาพ

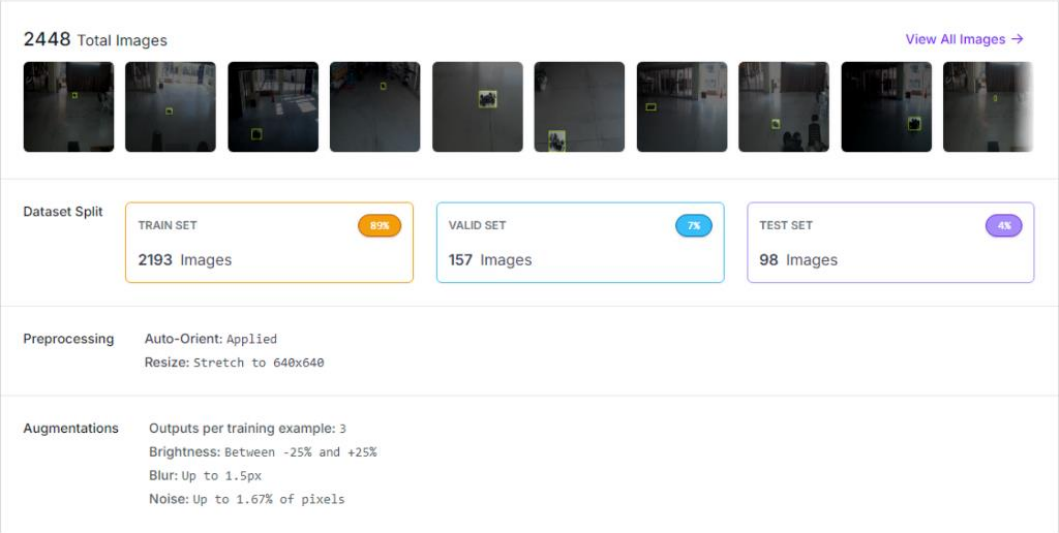
การตรวจจับวัตถุจากภาพ (Object Detection) เป็นกระบวนการที่ใช้ในการระบุและแยกแยะวัตถุที่ปรากฏในภาพหรือวิดีโออย่างแม่นยำ โดยทั่วไปแล้วจะใช้เทคโนโลยีปัญญาประดิษฐ์ประเภทเครือข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks - CNN) เพื่อประมวลผลภาพและตรวจจับวัตถุที่อยู่ในภาพอย่างรวดเร็วและมีประสิทธิภาพ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

ในตอนการรวบรวมภาพนี้จะทำการบันทึกภาพเป็นวีดีโอเพื่อใช้ในการเก็บข้อมูลโดยที่วีดีโอจะเป็นสถานที่ที่ต้องการใช้งานของระบบเพื่อเพิ่มความแม่นยำมากยิ่งขึ้นในการใช้งานระบบ

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 และการเพิ่มนอยซ์ลงในภาพการทำ 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) ซึ่งช่วยให้โมเดลมีโอกาสเรียนรู้จากข้อมูลที่ป้อนเข้ามามากขึ้น 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

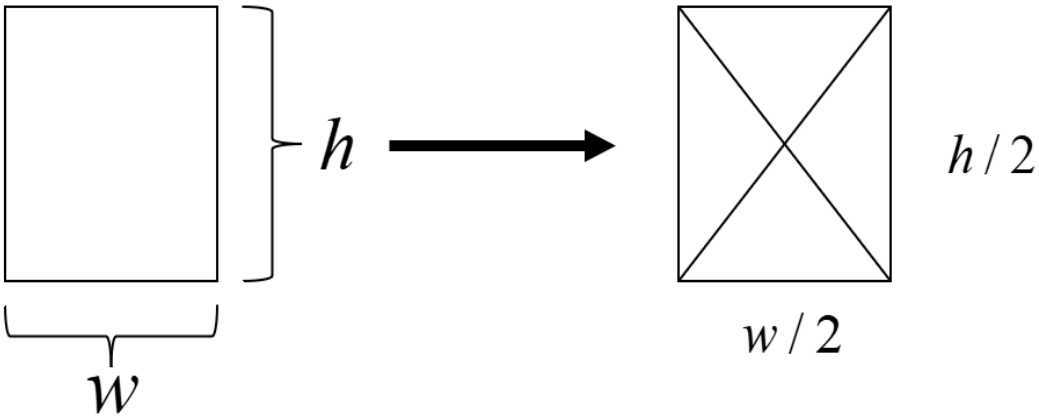
การติดตามวัตถุ (Object Tracking) เป็นกระบวนการที่ต่อเนื่องจากการตรวจจับวัตถุ โดยหลังจากที่วัตถุถูกตรวจพบแล้ว ระบบจะทำการติดตามการเคลื่อนที่ของวัตถุในหลาย ๆ เฟรมของวิดีโอ เทคโนโลยีการติดตามที่ใช้ได้ผลดี คือ DeepSort เป็นที่นิยมในการติดตามวัตถุแบบเรียลไทม์ โดยอาศัยการแยกแยะตำแหน่งของวัตถุจาก bounding box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

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

3.2.5 การระบุตำแหน่งของวัตถุที่ตรวจจับจากกล้อง

หลังจากขั้นตอนการแปลงมุมมองกล้องและการสร้างตารางกริดสำหรับอ้างอิงตำแหน่งเสร็จสิ้นแล้ว ระบบจะทำการตรวจจับวัตถุจากภาพที่ได้จากกล้อง การระบุตำแหน่งของวัตถุที่ตรวจจับจากกล้องจะดำเนินการโดยอ้างอิงตำแหน่งของวัตถุในภาพจากกรอบ Bounding Box ที่ได้จากการตรวจจับ และเทียบตำแหน่งนั้นกับตารางกริดที่ถูกสร้างขึ้น ตารางกริดที่ถูกสร้างขึ้นจากการแปลงมุมมองของกล้องจะถูกใช้เป็นระบบพิกัดอ้างอิงสำหรับการแปลงตำแหน่งของวัตถุจากภาพในมุมมองกล้องไปสู่ตำแหน่งจริงในสภาพแวดล้อมการแปลงตำแหน่งนี้ใช้การคำนวณจากตารางกริดที่สร้างขึ้นเพื่อแปลงพิกัดของวัตถุจากภาพให้สอดคล้องกับระบบพิกัดโลก ซึ่งจะช่วยให้ระบบสามารถทราบตำแหน่งที่แท้จริงของวัตถุในสภาพแวดล้อมจริงได้อย่างแม่นยำ

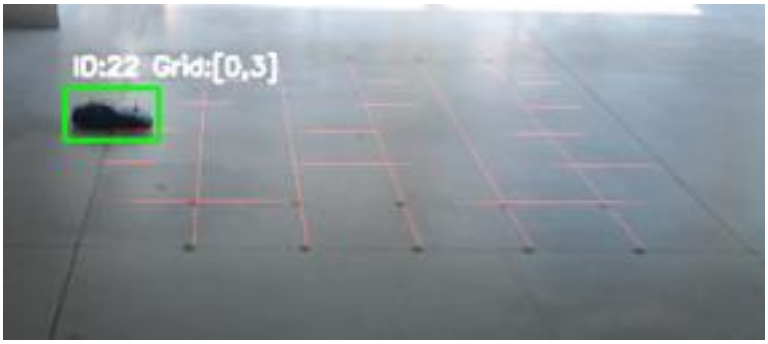
ขั้นตอนการคำนวณพิกัดของจุดกึ่งกลางของ Bounding Box ทำโดยการใช้ค่าพิกัดของ Bounding Box ที่ได้จากการทำนายวัตถุในภาพ ขั้นแรก ให้นำความกว้างของ Bounding Box มาหารครึ่ง เพื่อหาตำแหน่งแนวนอน (แกน X) ของจุดกึ่งกลาง จากนั้น นำความสูงของ Bounding Box มาหารครึ่ง เพื่อหาตำแหน่งแนวตั้ง (แกน Y) ของจุดกึ่งกลาง พิกัดจุดกึ่งกลางนี้จะใช้ในการอ้างอิงตำแหน่งของวัตถุในตารางกริดที่สร้างไว้ ทำให้สามารถระบุตำแหน่งจริงในสภาพแวดล้อมได้



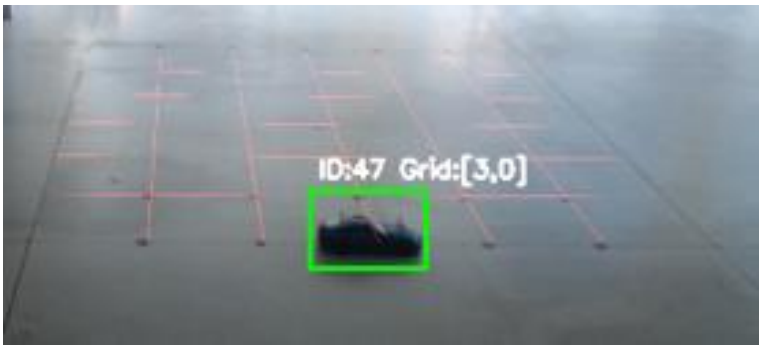
รูปที่ 3.15 วิธีหาจุดกึ่งกลางของ Bounding Box

หลังจากที่คำนวณหาจุดกึ่งกลางของ Bounding Box แล้ว พิกัดของจุดกึ่งกลางจะถูกนำมาเทียบกับตารางกริดที่สร้างขึ้นมา ระบบจะใช้พิกัดจุดนี้เพื่อเทียบกับตำแหน่งบนกริด ซึ่งกริดนี้ทำหน้าที่เป็นระบบพิกัดที่อ้างอิงกับพื้นที่จริงในสภาพแวดล้อม ทำให้สามารถแปลงพิกัดจากภาพที่ได้จากกล้องไปยังตำแหน่งจริงในโลกได้อย่างแม่นยำ การเปรียบเทียบพิกัดนี้ช่วยให้ระบบสามารถระบุได้ว่าวัตถุอยู่ที่ตำแหน่งใดในพื้นที่จริง

รูปที่ 3.16 การตรวจจับวัตถุในกริด [1,1]



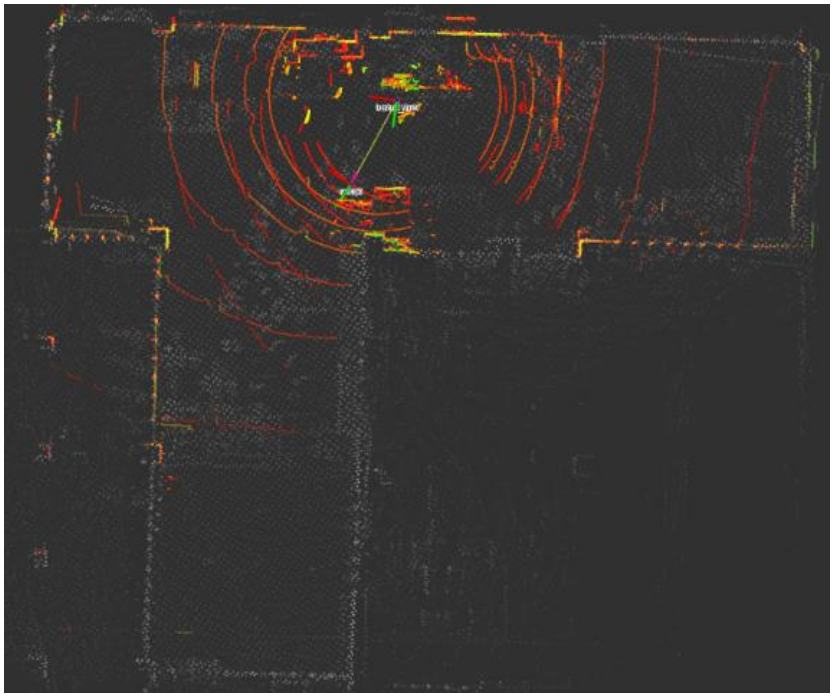
รูปที่ 3.17 การตรวจจับวัตถุในกริด [0,3]



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 ให้ข้อมูลเกี่ยวกับการเคลื่อนที่ของรถอัตราการหมุนและความเร่งข้อมูลจากทั้งสองเซ็นเซอร์จะถูกประมวลผลร่วมกันเพื่อให้ได้พิกัดที่แม่นยำของรถในสภาพแวดล้อมจริง



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 ข้อมูลเหล่านี้ถูกนำไปใช้ในกระบวนการระบุตำแหน่งของรถในพื้นที่จริง ซึ่งการทำงานร่วมกันระหว่าง LiDAR และ IMU ช่วยในการระบุทั้งการเคลื่อนไหวและการวางตำแหน่งของรถ โดย IMU จะให้ข้อมูลเกี่ยวกับการหมุนและการเร่งความเร็วของรถ ส่วน LiDAR จะสแกนสภาพแวดล้อมเพื่อสร้างแผนที่สามมิติ ข้อมูลทั้งสองส่วนจะถูกประมวลผลร่วมกันเพื่อให้ระบบระบุตำแหน่งของรถได้แม่นยำ

3.2.7 การแปลงพิกัดเพื่อประสานตำแหน่งของหุ่นยนต์และระบบกล้อง

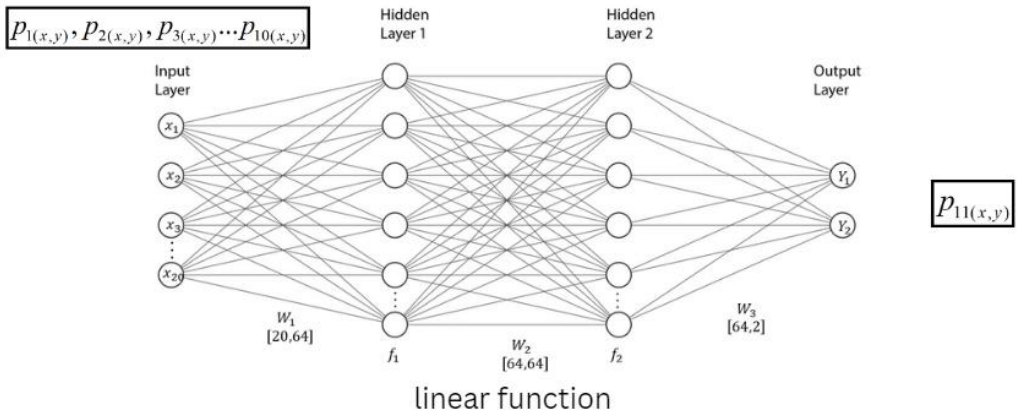
การแปลงพิกัด (Coordinate Transformation) เป็นการแปลงตำแหน่งระหว่างระบบพิกัดท้องถิ่น (Local Coordinate System) ของหุ่นยนต์และตำแหน่งที่ตรวจจับได้จากกล้องในระบบพิกัดโลก (Global Coordinate System) โดยกระบวนการนี้ใช้เมทริกซ์การแปลงเชิงโฮโมจีเนียสเพื่อให้การทำงานของหุ่นยนต์ในพื้นที่จริงสอดคล้องกับการตรวจจับวัตถุจากกล้องการแปลงพิกัดเพื่อประสานตำแหน่งของหุ่นยนต์และระบบ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

ร่วมกันได้อย่างถูกต้อง ขั้นตอนนี้ใช้การอ้างอิงตำแหน่งจากกล้องที่ตรวจจับวัตถุ (หัวข้อ 3.2.4) เป็นพิกัดหลัก เนื่องจากพื้นที่นี้มีความคงที่ การแปลงพิกัดจะดำเนินการโดยนำตำแหน่งที่ได้จาก LiDAR และ IMU ของรถอัตโนมัติ (หัวข้อ 3.2.5) มาเปรียบเทียบและปรับเข้ากับพิกัดกริดของกล้อง ซึ่งช่วยให้ทั้งหุ่นยนต์และกล้องมีการทำงานสอดคล้องกันและระบุตำแหน่งในพื้นที่เดียวกันได้อย่างแม่นยำ การแปลงพิกัดนี้ทำโดยการนำข้อมูลพิกัดจาก LiDAR ที่เป็นพิกัดแบบ 3 มิติ มารวมกับข้อมูลจาก IMU เพื่อให้ระบบสามารถรับรู้ตำแหน่งการเคลื่อนไหวของหุ่นยนต์ในสภาพแวดล้อมจริง ข้อมูลจากทั้งสองเซ็นเซอร์นี้จะถูกนำไปเทียบกับพิกัดกริดที่ได้จากกล้องในพื้นที่ ทำให้หุ่นยนต์และระบบกล้องมีการรับรู้ตำแหน่งของวัตถุและพื้นที่ที่สอดคล้องกันกระบวนการแปลงพิกัดนี้ใช้การคำนวณผ่านการแปลงเชิงโฮโมจีเนียสซึ่งประกอบด้วยการ 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) หน่วยประมวลผลเหล่านี้จะทำหน้าที่ประมวลผลข้อมูลและช่วยให้โมเดลสามารถเรียนรู้รูปแบบการเคลื่อนไหวและการเปลี่ยนแปลงของตำแหน่งที่ซับซ้อนขึ้นการใช้ 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 มาเปรียบเทียบกันในช่วงเวลาเดียวกัน โดยกระบวนการนี้จะวิเคราะห์ว่าตำแหน่งที่ทำนายจากโมเดลทั้งสองมีความใกล้เคียงกันมากเพียงใด และตรวจสอบว่ามีความเสี่ยงที่จะเกิดการชนหรือไม่การวิเคราะห์นี้ใช้การคำนวณระยะห่างระหว่างตำแหน่งที่ทำนายจากทั้งสองโมเดลโดยใช้ สูตรการคำนวณระยะทางระหว่างจุดสองจุด

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

โดยที่:

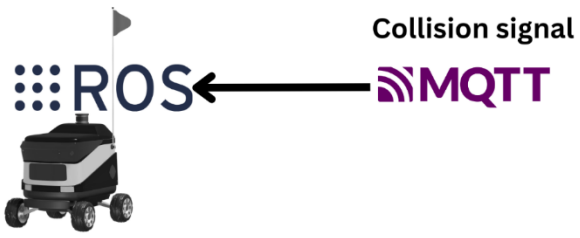
- (x_1, y_1) **คือตำแหน่งที่ทำนายจากโมเดลการทำนายตำแหน่งจากกล้อง**
- (x_2, y_2) **คือตำแหน่งที่ทำนายจากโมเดลการทำนายตำแหน่งจากรถอัตโนมัติ**

รูปที่ 3.21 สูตรการคำนวณระยะทางระหว่างสองจุดในระบบพิกัดคาร์ทีเซียน (Cartesian Coordinates)

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

3.2.10 การพัฒนาระบบการชะลอความเร็วเมื่อมีการทำนายว่าจะเกิดการชน

ระบบการชะลอความเร็ว (Speed Reduction System) เป็นการพัฒนาที่ทำงานร่วมกับการทำนายตำแหน่ง เมื่อระบบตรวจพบว่ามีความเป็นไปได้ที่จะเกิดการชน ระบบจะส่งสัญญาณไปยังหุ่นยนต์หรือยานพาหนะเพื่อชะลอความเร็วหรือหยุดการเคลื่อนที่ เพื่อป้องกันการเกิดอุบัติเหตุ

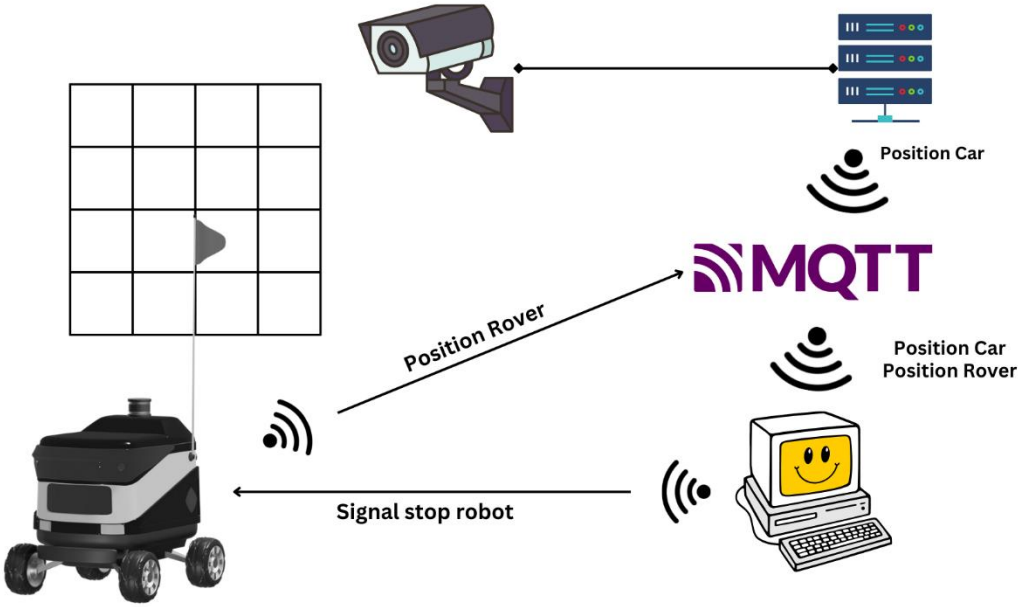


รูปที่ 3.22 การสื่อสารระหว่างระบบ MQTTและROS เพื่อการชะลอและหยุดหุ่นยนต์เมื่อมีความเสี่ยงชน

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 ทำให้สามารถชะลอและหยุดการเคลื่อนที่ได้อย่างราบรื่นและปลอดภัย

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 เพื่อให้มั่นใจว่าระบบสามารถตอบสนองต่อสถานการณ์ที่มีความเสี่ยงได้อย่างทันทีการทดสอบระบบทั้งหมดในหัวข้อนี้เป็นการยืนยันว่าระบบสามารถทำงานได้อย่างปลอดภัยและแม่นยำเมื่อตรวจพบความเสี่ยงในการชน โดยหุ่นยนต์สามารถชะลอความเร็วและหยุดการเคลื่อนที่ได้ทันเวลา ลดความเสี่ยงในการเกิดอุบัติเหตุและเพิ่มความปลอดภัยในการทำงาน



รูปที่ 3.23 แสดงกระบวนการทำงานของระบบการสื่อสารตำแหน่งและสัญญาณการหยุดหุ่นยนต์