



รถจำลองอัตโนมัติสำหรับถนนจำลองพร้อมหลบหลีกและหยุดรถ
Auto Car Simulator For Road Simulator With Dodge and Stop

นายปวิภาณ แสงกาศ
นายวีรวุฒิ ลักษณะธาร
นายฐญา ปัญญาเคิ่ง

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา เชียงใหม่
ปีการศึกษา 2567

รถจำลองอัตโนมัติสำหรับถนนจำลองพร้อมหลบหลีกและหยุดรถ

นายปฏิภาณ แสงกาศ
นายวีรวุฒิ ลักขณาธร
นายฐญา ปัญญาเคิ่ง

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา เชียงใหม่
ปีการศึกษา 2567

AUTO CAR SIMULATOR FOR ROAD SIMULATOR WITH DODGE AND STOP

MR. PATIPARN SANGKAD

MR. WEERAWUT LUKKANATORN

MR. THAYA PANYAKERNG

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE BACHELOR DEGREE OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING FACULTY OF ENGINEERING
RAJAMANGLA UNIVERSITY OF TECHNOLOGY LANNA
ACADEMIC YEAR 2024

ปริญญานิพนธ์เรื่อง	รถจำลองอัตโนมัติสำหรับถนนจำลองพร้อมหลบหลีกและหยุดรถ
ชื่อนักศึกษา	นายวีรวุฒิ ลักขณาธร รหัส 65543206078-9
	นายฐญา ปัญญาเคิ่ง รหัส 65543206051-6
	นายปฏิภาณ แสงภาส รหัส 65543206068-0
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ขวัญชัย เอื้อวิริยานุกุล
ปีการศึกษา	2567

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา อนุมัติให้ปริญญานิพนธ์นี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรม
คอมพิวเตอร์

.....	รองคณบดีคณะวิศวกรรมศาสตร์
(ผู้ช่วยศาสตราจารย์กฤษดา ยิ่งขยัน)	
คณะกรรมการสอบโครงการวิศวกรรมคอมพิวเตอร์	

.....	ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ขวัญชัย เอื้อวิริยานุกุล)	

.....	กรรมการ
(ผู้ช่วยศาสตราจารย์ อนันท์ ทับเกิด)	

.....	กรรมการ
(อาจารย์ พิชิต ทนชัย)	

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

ปริญญานิพนธ์เรื่อง	รถจำลองอัตโนมัติสำหรับถนนจำลองพร้อมหลบหลีกและหยุดรถ
ชื่อนักศึกษา	นายวีรวุฒิ ลักขณารร รหัส 65543206078-9 นายฐญา ปัญญาเค็ง รหัส 65543206051-6 นายปฎิภาณ แสงกาศ รหัส 65543206068-0
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ขวัญชัย เอื้อวิริยานุกูล
ปีการศึกษา	2567

บทคัดย่อ

โครงการนี้มุ่งเน้นการพัฒนาารถจำลองอัตโนมัติสำหรับถนนจำลองที่สามารถหลบหลีกและหยุดรถได้ โดยใช้เทคโนโลยีปัญญาประดิษฐ์ (AI) เพื่อเพิ่มความปลอดภัยและลดอุบัติเหตุบนท้องถนน ระบบที่พัฒนาขึ้นประกอบด้วย Raspberry Pi ซึ่งทำหน้าที่ควบคุมการทำงานของกล้องเว็บแคมสองตัวและเซ็นเซอร์อัลตราโซนิก กล้องเว็บแคมถูกใช้ในการตรวจจับเส้นทางและวัตถุด้านหน้า โดยอาศัยโมเดล MobileNet และ YOLO เพื่อวิเคราะห์ข้อมูลอย่างแม่นยำ ระบบยังสามารถประเมินระยะทางจากเซ็นเซอร์อัลตราโซนิกเพื่อช่วยในการชะลอและหยุดรถเมื่อมีวัตถุขวางอยู่ นอกจากนี้ การดำเนินการยังครอบคลุมถึงการสร้างชุดข้อมูลเพื่อฝึกโมเดล AI และการทดสอบระบบในสถานการณ์จำลองหลากหลายรูปแบบ เพื่อให้มั่นใจในความเสถียรและประสิทธิภาพของระบบ

ผลลัพธ์ที่คาดว่าจะได้รับจากโครงการนี้ คือการส่งเสริมการศึกษาและพัฒนาทักษะด้านปัญญาประดิษฐ์ (AI) และเทคโนโลยี IoT พร้อมทั้งสร้างต้นแบบที่สามารถนำไปใช้ทดสอบและพัฒนา ระบบขับเคลื่อนอัตโนมัติในอนาคต โครงการนี้ยังมีศักยภาพในการนำไปต่อยอดเพื่อการวิจัยและพัฒนาาระบบรักษาความปลอดภัยสำหรับยานยนต์ไร้คนขับ อีกทั้งยังช่วยลดปัญหาการจราจรติดขัด และสร้างความปลอดภัยให้กับผู้ใช้รถใช้ถนนในอนาคต

คำสำคัญ : รถจำลองอัตโนมัติ, ปัญญาประดิษฐ์, YOLO, MobileNet, หลบหลีกและหยุดรถ, การตรวจจับวัตถุ

Project Title	Auto car simulator for road simulator with dodge and stop
Student	MR. Weerawut Lukkanatorn MR. Thaya Panyakerng MR. Patiparn Sangkad
Project Advisor	Asst. Prof. Kwanchai Ueawiriyananukul
Curriculum	Bachelor of Engineering
Major Field	Computer Engineering
Academic Year	2024

ABSTRACT

This project focuses on developing an autonomous car simulator for model roads that can avoid obstacles and stop as needed, utilizing artificial intelligence (AI) technology to enhance safety and reduce road accidents. The system incorporates a Raspberry Pi as the central controller, managing two webcams and an ultrasonic sensor. The webcams are used for detecting road patterns and objects in front, utilizing MobileNetV3 and YOLO V8 models for precise data analysis. Additionally, the ultrasonic sensor estimates distances to assist in deceleration and stopping when obstacles are detected. The project involves creating datasets to train AI models and testing the system under various simulated scenarios to ensure stability and efficiency.

The expected outcomes of this project include advancing education and skills in AI and IoT technologies and providing a prototype for testing and developing autonomous driving systems. Furthermore, this project has the potential for further application in research and development of security systems for autonomous vehicles. It aims to reduce traffic congestion and enhance road safety for future users.

Keywords : Autonomous car simulator, Artificial Intelligence (AI), YOLO, MobileNet, Obstacle avoidance and stopping system, Object detection

กิตติกรรมประกาศ

นิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ ผู้ช่วยศาสตราจารย์ ดร.ขวัญชัย เอื้อวิริยานุกูล อาจารย์ที่ปรึกษานิพนธ์ที่ได้ให้คำแนะนำ และข้อคิดเห็นต่าง ๆ ของการทำโครงการมาโดยตลอดจึงขอขอบพระคุณมา ณ ที่นี้ด้วย

ขอขอบคุณผู้ช่วยศาสตราจารย์ ดร.ขวัญชัย เอื้อวิริยานุกูล บุคคลที่ช่วยในการสืบค้นข้อมูล แลกเปลี่ยนความรู้ความคิด และให้กำลังใจในการศึกษาค้นคว้าตลอดมา

ท้ายนี้ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา มารดา ซึ่งสนับสนุนในด้านการเงินและให้กำลังใจแก่ผู้จัดทำเสมอมาจนสำเร็จการศึกษา

คณะผู้จัดทำ

นายปฏิภาณ แสงภาศ

นายวีรวุฒิ ลักขณาธร

นายฐญา ปัญญาเคิ่ง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญรูป	ฌ
สารบัญตารางและคำย่อ	ฎ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินโครงการ	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับโครงการ	5
2.1 ทฤษฎีที่เกี่ยวข้อง	5
2.2 งานวิจัยหรือโครงการที่เกี่ยวข้อง	12
บทที่ 3 ขั้นตอนการดำเนินงาน	16
3.1 วิธีการดำเนินโครงการ	16
3.2 หลักการแนวความคิด	17
3.3 การออกแบบซอฟต์แวร์	17
3.4 การออกแบบฮาร์ดแวร์	33
บทที่ 4 ผลการทดลอง	37
4.1 อุปกรณ์ที่ใช้ในการทดสอบ	37
4.2 ทดสอบการทำงานวิ่งบนถนนจำลอง	38
4.3 การทำงานของอุปกรณ์ตรวจจับโมเดลรถ	40

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปและข้อเสนอแนะ	50
5.1 สรุปผลตามของเขต	50
5.2 ปัญหาและอุปสรรค	50
5.3 ข้อเสนอแนะ	51
5.4 สรุปผลการทดลอง	51
บรรณานุกรม	52
ภาคผนวก	
Source Code	55
คู่มือการใช้งานรถจำลองอัตโนมัติตรวจจับโมเดลรถ	58
ประวัติผู้ทำปริญญานิพนธ์	61

สารบัญรูป

	หน้า
รูปที่ 2.1 Raspberry Pi	5
รูปที่ 2.2 YoLo	5
รูปที่ 2.3 Mobilenet	6
รูปที่ 2.4 Convolutional Neural Network	6
รูปที่ 2.5 Deep Learning	7
รูปที่ 2.6 Loss Function	7
รูปที่ 2.7 Overfitting	8
รูปที่ 2.8 Mean Absolute Error	9
รูปที่ 2.9 OpenCV	9
รูปที่ 2.10 Ultrasonic Sensor	10
รูปที่ 2.11 Google Colaboratory	10
รูปที่ 3.1 แสดงหลักการทำงานของอัลกอริทึมสำหรับถนนจำลองพร้อมหลบหลีกและหยุดรถ	17
รูปที่ 3.2 โครงสร้างโปรแกรมตรวจจับถนนและโมเดลรถ	18
รูปที่ 3.3 Code Data Collection Main	19
รูปที่ 3.4 Code DataCollectionModule	20
รูปที่ 3.5 Code JoyStickModule	21
รูปที่ 3.6 Code Test	22
รูปที่ 3.7 Code WebcamModule	23
รูปที่ 3.8 YB_PcB_car.py	24
รูปที่ 3.9 Dataset ถนนจำลอง	26
รูปที่ 3.10 ค่า Steering, Left_Speed, และ Right_Speed ในไฟล์ .csv	26
รูปที่ 3.11 Detect car model	27
รูปที่ 3.12 Dataset ถนนจำลอง	27
รูปที่ 3.13 โค้ดข้อมูลจากไฟล์ CSV ใน Google Colaboratory	28
รูปที่ 3.14 โค้ดเทรนใน Google Colaboratory ก	29
รูปที่ 3.16 โค้ดเทรนใน Google Colaboratory ข	29
รูปที่ 3.16 ความเร็วของมอเตอร์ที่ใช้เลี้ยวขวาและทางตรง	30
รูปที่ 3.17 กราฟการเลี้ยวซ้าย	31
รูปที่ 3.18 กราฟการเลี้ยวขวา	31

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.19 กราฟ Mse loss and Mae	32
รูปที่ 3.20 ขนาดความกว้างและความสูงของโมเดลต้นแบบรถจำลอง	33
รูปที่ 3.21 ส่วนประกอบต่างๆเมื่อนำมาประกอบเข้าด้วยกัน	34
รูปที่ 3.22 ดัดแปลงโมเดลต้นแบบ	34
รูปที่ 3.23 ออกแบบถนนจำลองสำหรับทดลองรถขับอัตโนมัติ	35
รูปที่ 3.24 Hardware Diagram	36
รูปที่ 4.1 Raspberry Pi 5	37
รูปที่ 4.2 Raspberry Pi AI HAT	37
รูปที่ 4.3 Webcam	38
รูปที่ 4.4 รถจำลองอัตโนมัติ	38
รูปที่ 4.5 ทางตรงของถนนจำลอง ก	38
รูปที่ 4.6 ทางตรงของถนนจำลอง ข	39
รูปที่ 4.7 การเลี้ยวขวาของถนนจำลอง ก	39
รูปที่ 4.8 การเลี้ยวขวาของถนนจำลอง ข	39
รูปที่ 4.9 การทำงานของอุปกรณ์ตรวจจับโมเดลรถ	40
รูปที่ 4.10 รถจำลองอัตโนมัติเจอโมเดลรถในระยะ 100 cm	40
รูปที่ 4.11 รถจำลองอัตโนมัติเจอโมเดลรถในระยะ 50 cm	41
รูปที่ 4.12 รถจำลองอัตโนมัติเจอโมเดลรถในระยะ 25 cm	41
รูปที่ 4.13 รถจำลองอัตโนมัติเจอโมเดลรถในระยะ 50 cm	42
รูปที่ 4.14 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 1	43
รูปที่ 4.15 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 3	44
รูปที่ 4.16 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 5	44
รูปที่ 4.17 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 1	45
รูปที่ 4.18 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 6	46
รูปที่ 4.19 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 7	46
รูปที่ 4.20 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 1	47
รูปที่ 4.21 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 2	48

สารบัญตาราง

	หน้า
ตารางที่ 1.1 แผนการดำเนินโครงการ	4
ตารางที่ 2.1 ตารางเปรียบเทียบสเปคของบอร์ดแต่ละชนิดทางวิศวกรรม	11
ตารางที่ 4.1 การทดสอบประสิทธิภาพโมเดล ในการตรวจจับและวิ่งอัตโนมัติทั้งหมด	42
ตารางที่ 4.2 การทดสอบประสิทธิภาพโมเดล ในการตรวจจับและวิ่งอัตโนมัติรอบที่ 1	43
ตารางที่ 4.3 การทดสอบประสิทธิภาพโมเดล ในการตรวจจับและวิ่งอัตโนมัติรอบที่ 2	45
ตารางที่ 4.4 การทดสอบประสิทธิภาพโมเดล ในการตรวจจับและวิ่งอัตโนมัติรอบที่ 3	47
ตารางที่ 4.5 สรุปความถูกต้องและประสิทธิภาพของโมเดล	48

ประมวลศัพท์และคำย่อ

AI	=	Artificial Intelligence
CNN	=	Convolutional Neural Network
YOLO	=	Yon Only Look One
PCB	=	Print Circuit Board
PTZ	=	Pan Tilt Zoom camera
GPIO	=	General Purpose input/output
I2C	=	Inter-Integrated Circuit
MQTT	=	Message Queuing Telemetry Transport

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยียานยนต์ที่ขับเคลื่อนด้วยพลังงานไฟฟ้า และยานยนต์ไร้คนขับ เป็นอีกหนึ่งนวัตกรรมที่กำลังได้รับความสนใจจากทั่วโลกด้วยความก้าวหน้าของวิทยาการเทคโนโลยีของปัญญาประดิษฐ์ หรือ Artificial Intelligence (AI) เพื่อให้รถยนต์อัตโนมัติสามารถรับรู้และตอบสนองต่อสิ่งแวดล้อมได้อย่างมีประสิทธิภาพ ทำให้ปัจจุบันการดำเนินชีวิตของมนุษย์มีความสะดวกสบาย และมีความปลอดภัยในการดำเนินชีวิต ลดการจราจรและอุบัติเหตุบนท้องถนน จะสังเกตเห็นได้ว่าการเปลี่ยนแปลงเยอะขึ้นมากในประเทศไทยจะเห็นรถยนต์บนท้องถนนมากขึ้น และคงหลีกเลี่ยงไม่ได้ว่าในอนาคตยานยนต์จะมีระบบขับเคลื่อนอัตโนมัติในระดับที่สูงขึ้น และอาจเปลี่ยนรูปแบบการเดินทางไปอย่างสิ้นเชิง

ปัญหบนท้องถนนนั้นมีการเกิดอุบัติเหตุที่เกิดจากไม่ระวังตัวของผู้ขับทำให้เกิดความเสียหายขึ้นเดือดร้อนต่อผู้ร่วมทางบนท้องถนน สาเหตุใหญ่มาจากคนขับที่ซบเซาเร็ว ขาดความระมัดระวัง เรื่องความเมา และความง่วง, ความเหนื่อย ของผู้ขับรถ ทางผู้พัฒนาได้เล็งเห็นโอกาสของรถจำลองอัตโนมัติสำหรับจำลองถนนพร้อมหลบหลีกและหยุดรถ ที่จะได้ลดอุบัติเหตุบนท้องถนน อีกทั้งยังสามารถแบ่งเบาภาระในชีวิตประจำวันของมนุษย์ได้อีกทางหนึ่ง โดยเฉพาะผู้สูงอายุ ซึ่งมักมีอุปสรรคในการขับรถเนื่องจากข้อจำกัดทางร่างกาย และทำให้รถยนต์ขับเคลื่อนได้ตามกฎจราจร การสัญจรสะดวก การจราจรไม่ติดขัด เพราะรถทุกคันสามารถขับเคลื่อนได้อย่างเป็นระเบียบด้วยระบบไร้คนขับ ทั้งนี้ยังช่วยให้คนพิการ และยังสามารถนำไปต่อยอดเพิ่มเติมได้

ทางผู้จัดทำได้เห็นถึงปัญหาดังกล่าวจึงได้คิดพัฒนา แบบจำลองรถพร้อมหลบหลีกและหยุดรถ โดยใช้วิทยาการเทคโนโลยีของปัญญาประดิษฐ์ หรือ Artificial Intelligence (AI) เพื่อพัฒนาระบบตรวจจับวัตถุ (Object Detection) โดยใช้กล้อง Webcam 2ตัว ในการบันทึกวิดีโอเพื่อสร้าง Dataset ขึ้นมาในการตรวจจับที่แม่นยำ โดย Webcam ตัวที่1 ใช้ในบันทึกตรวจจับเส้นทาง Webcam ตัวที่2 ใช้ในการตรวจจับรถด้านหน้า และมี Ultrasonic Sensor ใช้ในการ ตรวจจับวัตถุที่อยู่ด้านหน้า และวัดระยะห่างของรถด้านหน้า และการตรวจจับเส้นทางบนถนน โดยใช้ MobileNetV3 เป็นโมเดล และการตรวจจับวัตถุจะใช้ YOLO V8 สามารถหยุดรถเมื่อเจอวัตถุขวางอยู่หน้ารถหรือหลบวัตถุอื่นๆได้

1.1 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อพัฒนารถจำลองที่สามารถขับเคลื่อนอัตโนมัติบนถนนจำลองได้
- 1.2.2 เพื่อสร้างระบบที่สามารถหลีกเลี่ยงการชนและช่วยลดความเสี่ยงต่ออุบัติเหตุบนท้องถนน
- 1.2.3 เพื่อพัฒนาการตรวจจับและตอบสนองต่อสิ่งกีดขวางและรถคันหน้า โดยมีการหลบหลีกหรือหยุดรถ
- 1.2.4 เพื่อพัฒนาชุดข้อมูลสำหรับการตรวจจับวัตถุให้แม่นยำ โดยใช้กล้องเว็บแคมในการบันทึกวิดีโอ

1.3 ขอบเขตของโครงการ

โครงการ รถจำลองอัตโนมัติสำหรับจำลองถนนพร้อมหลบหลีกและหยุดรถ รถจำลอง จะติดตั้งเว็บแคมและเซ็นเซอร์อัลตราโซนิกและใช้ Raspberry Pi ในการควบคุมหลัก โดยใช้โมเดล AI Yolo V5 และ Mobilenet V2 สำหรับการตรวจจับรถยนต์ที่อยู่ด้านหน้าและวิเคราะห์รูปแบบถนน

มีคุณลักษณะ ความสามารถ ดังต่อไปนี้

- 1.3.1 รถจำลองที่สามารถขับเคลื่อนบนถนนจำลองได้อย่างอัตโนมัติโดยใช้ AI
 - 1.3.1.1 หลบหลีก แสง โมเดลรถที่อยู่ด้านหน้าหากมี 1 คัน
 - 1.3.1.2 ชะลอและหยุด เมื่อมีโมเดลรถอยู่ด้านหน้าข้างกัน 2 คัน
- 1.3.2 ติดตั้งกล้องเว็บแคมสองตัวและเซ็นเซอร์อัลตราโซนิก
 - 1.3.2.1 เว็บแคมตัวที่ 1 ใช้ตรวจจับเส้นทาง
 - 1.3.2.2 เว็บแคมตัวที่ 2 ใช้ตรวจจับรถยนต์ด้านหน้า
 - 1.3.2.3 เซ็นเซอร์อัลตราโซนิกเพื่อวัดระยะห่างและตรวจจับวัตถุด้านหน้า เมื่อรถจำลองจะเริ่มลดความเร็วเมื่อเจอโมเดลรถอยู่ข้างหน้าในระยะ 100 cm และจะชะลอในระยะ 50 cm. และจะหยุดรถในระยะ 25 cm
- 1.3.3 Raspberry Pi เป็นตัวควบคุมหลักสำหรับระบบทั้งหมด
 - 1.3.3.1 ควบคุมเว็บแคมทั้ง 2 อัน
 - 1.3.3.2 ควบคุมเซ็นเซอร์อัลตราโซนิก
 - 1.3.3.3 ควบคุมมอเตอร์ทั้ง 4 อัน
 - 1.3.3.4 ใช้โมเดล YOLO V8s และ ใช้โมเดล MobileNetV2 ในการตรวจจับ
- 1.3.4 ใช้โมเดล YOLO V8s และ ใช้โมเดล MobileNet V2
- 1.3.5 YOLO V8s ใช้ในการตรวจจับรถยนต์ที่อยู่ด้านหน้า
- 1.3.6 MobileNet V2 ใช้ในการตรวจจับและวิเคราะห์รูปแบบถนน และให้รถจำลองทำการโค้งตามรูปแบบของถนน

1.4 วิธีการดำเนินโครงการ

ในการดำเนินโครงการ "รถจำลองอัตโนมัติสำหรับจำลองถนนพร้อมหลบหลีกและหยุดรถ" ทีมนักพัฒนามุ่งเน้นที่การสร้างระบบจำลองรถอัตโนมัติที่สามารถตรวจจับและตอบสนองต่อวัตถุและสภาพแวดล้อมบนถนนได้อย่างแม่นยำ โดยอาศัยการใช้ปัญญาประดิษฐ์ (AI) รวมถึงการวิเคราะห์ข้อมูลจากเซ็นเซอร์และกล้อง 2 ตัว กระบวนการพัฒนาเริ่มต้นตั้งแต่การศึกษาและออกแบบระบบการทำงานของรถ การติดตั้งกล้องและเซ็นเซอร์ การพัฒนาโมเดลการตรวจจับวัตถุ

1.4.1 การวางแผนโครงการ

- 1.4.1.1. กำหนดวัตถุประสงค์ของโครงการ และ เขตของโครงการ
- 1.4.1.2. กำหนดรายละเอียดของโครงการ
- 1.4.1.3. กำหนดอุปกรณ์ และเครื่องมือ ที่ต้องใช้

1.4.2 ออกแบบโครงสร้างงาน

- 1.4.2.1. ออกแบบระบบตรวจสอบวัตถุ โดย YOLO V8s
- 1.4.2.2. ออกแบบการตรวจจับถนน โดย MobileNet V2
- 1.4.2.3. ออกแบบระบบเซ็นเซอร์อัลตราโซนิก
- 1.4.2.4. ออกแบบการขับอัตโนมัติของรถจำลอง

1.4.3 พัฒนาระบบ

- 1.4.3.1. พัฒนาการตรวจจับวัตถุโดย AI
- 1.4.3.2. พัฒนาการตรวจจับถนนโดย AI
- 1.4.3.3. พัฒนาระบบการขับอัตโนมัติของรถจำลอง
- 1.4.3.4. นำระบบทุกส่วนรวมเข้าด้วยกัน

1.4.4 ทดสอบและแก้ไข

- 1.4.4.1. ทดสอบการทำงานของระบบด้วยสถานการณ์ต่างๆ
- 1.4.4.2. ทำการทดสอบการทำงานของระบบเมื่อใช้งานร่วมกันระหว่างอัลตราโซนิก และ

Webcam

- 1.4.4.3. ปรับปรุงประสิทธิภาพการทำงานและแก้ไข

1.4.5 จัดทำรูปเล่มปริญญานิพนธ์

- 1.4.5.1. ทำรูปเล่มส่งมอบโครงการ

1.4.2 ตารางแผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	เดือน / พ.ศ. 2567 - 2568							
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. การวางแผนโครงการ	↔							
2. ออกแบบโครงสร้างงาน		↔	↔					
3. พัฒนาระบบ				↔	↔			
4. ทดสอบและแก้ไข				↔	↔		↔	
5. จัดทำรูปเล่มปริญญานิพนธ์					↔	↔	↔	↔

1.5 ประโยชน์ที่คาดว่าจะได้รับ

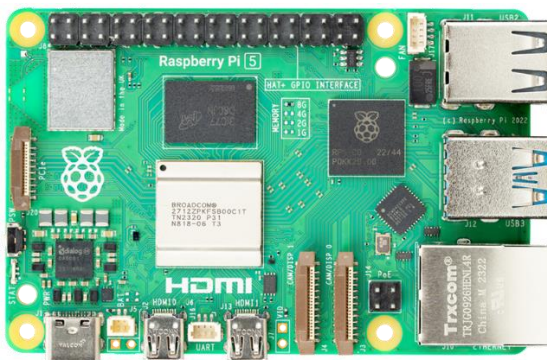
- 1.5.1. ส่งเสริมการศึกษาและพัฒนาทักษะทางด้านหุ่นยนต์และเทคโนโลยี IoT
- 1.5.2. นักศึกษาสามารถใช้โครงการนี้สำหรับการศึกษาการใช้งานปัญญาประดิษฐ์ (AI)
- 1.5.3. สามารถนำไปใช้เพื่อทดสอบและพัฒนาระบบรักษาความปลอดภัยสำหรับรถยนต์อัตโนมัติ

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับโครงการ

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Raspberry Pi

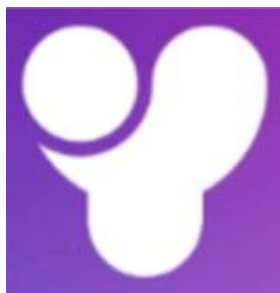


รูปที่ 2.1 Raspberry Pi 5

(ที่มา : <https://www.raspberrypi.com/products/raspberry-pi-5/>)

จากรูปที่ 2.1 Raspberry Pi 5 เป็นคอมพิวเตอร์ขนาดเล็กและขนาดเท่ากับบัตรเครดิตหนึ่งใบที่จัดอยู่ในกลุ่มคอมพิวเตอร์แบบฝังตัว (Embedded Computer) ที่มีประสิทธิภาพในการใช้งานสูง มีส่วนประกอบของอุปกรณ์หลักๆคือ หน่วยประมวลผลกลาง (CPU), หน่วยประมวลผลกราฟิก (GPU), หน่วยความจำ (RAM), ตัวรับสัญญาณไวไฟ (Wi-Fi Receiver), พอร์ตต่าง ๆ ได้แก่ HDMI, USB และ LAN ในตัว ได้ถูกนำมาใช้งานหลากหลาย ได้แก่ การใช้ตรวจจับใบหน้า การใช้งานแบบระบบ IoT และควบคุมอุปกรณ์แบบอัตโนมัติ หรือแม้กระทั่งใช้เป็น Web Server ได้

2.1.2 YoLo หรือ You Only Look Once



รูปที่ 2.2 YoLo

(ที่มา : <https://shorturl.asia/91UOD>)

จากรูปที่ 2.2 Realtime Object Detection Model เป็นโมเดลอัลกอริทึมในการตรวจจับวัตถุ ซึ่งจะเป็นวิธีการที่สามารถตรวจจับวัตถุได้ จากการส่งผ่านรูปภาพเข้าไปในระบบเพียงครั้งเดียว โดยใช้หลักการ โครงข่ายประสาทแบบคอนโวลูชัน ที่มีความโดดเด่นเรื่องความเร็วและความถูกต้อง โดยโครงสร้าง นี้จะมีหลายจำนวนชั้น (Layers) ทำให้ลดเวลาและลดการใช้ทรัพยากรในการประมวลผลลงได้อย่างมาก ในอัลกอริทึม YOLO V8 การดำเนินการจำแนกวัตถุว่าเป็นชนิดอะไร (Classification) และดำเนินการหาตำแหน่งของวัตถุ (Localization) โดยใช้กรอบล้อมวัตถุ (Bounding Box)

2.1.3 MobilenetV2

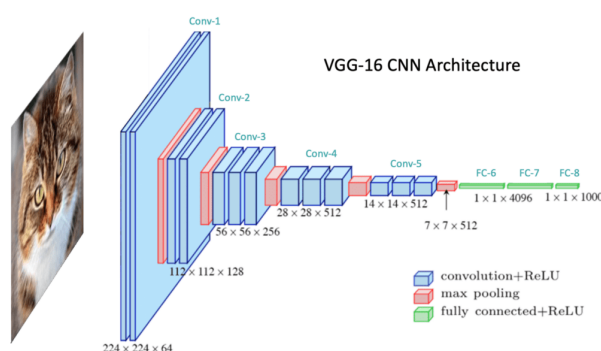
MobileNet

รูปที่ 2.3 Mobilenet

(ที่มา : <https://keras.io/api/applications/mobilenet/>)

จากรูปที่ 2.3 การนำ MobileNetV3 มาใช้ตรวจจับถนน ทำได้โดยการฝึกโมเดลให้เรียนรู้ลักษณะเฉพาะของถนนจากชุดข้อมูลขนาดใหญ่ที่ประกอบด้วยภาพถนนหลากหลายรูปแบบ เช่น ถนนในเมือง ถนนชนบท ถนนที่มีสภาพแตกต่างกัน และภายใต้สภาพแสงที่หลากหลาย เมื่อฝึกโมเดลสำเร็จแล้ว โมเดลจะสามารถนำไปใช้ในการวิเคราะห์ภาพใหม่ๆ และระบุตำแหน่งของถนนได้อย่างแม่นยำ

2.1.4 Convolutional Neural Network (CNN)

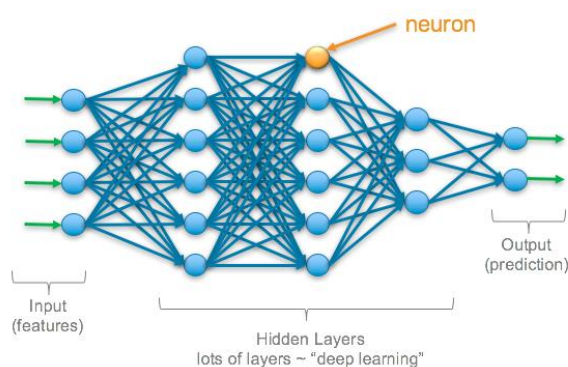


รูปที่ 2.4 Convolutional Neural Network

(ที่มา : <https://www.ibm.com/think/topics/convolutional-neural-networks>)

จากรูปที่ 2.4 หรือ โครงข่ายประสาทแบบคอนโวลูชัน CNN จะใช้ Convolution Layer มาประกอบกับ Layer ชนิดอื่น เช่น Pooling Layer แล้วนำกลุ่ม Layer ดังกล่าวมาซ้อนต่อกัน เพื่อดูว่าสิ่งที่เห็นอยู่เป็นอะไรกันแน่โดยที่ Convolution layer ซึ่งทำหน้าที่สกัดเอาส่วนต่างๆ ของภาพออกมา เช่น เส้นขอบของวัตถุต่างๆ เพื่อให้โมเดลสามารถเรียนรู้ลักษณะของภาพได้อย่างมีประสิทธิภาพและแม่นยำ

2.1.5 Deep Learning

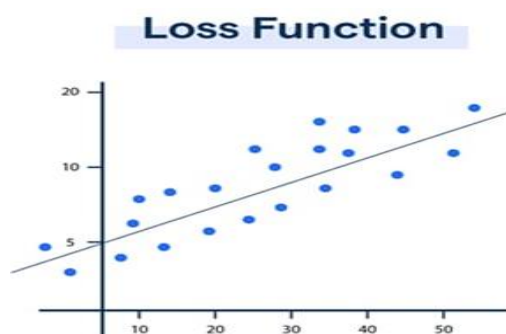


รูปที่ 2.5 Deep Learning

(ที่มา : <https://shorturl.asia/mDdbC>)

จากรูปที่ 2.5 วิธีการเรียนรู้แบบอัตโนมัติด้วยการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์โดยนำระบบโครงข่ายประสาทเทียม มาซ้อนกันหลายชั้น และทำการเรียนรู้ข้อมูลตัวอย่าง ซึ่งข้อมูลจะถูกนำไปใช้ในการตรวจจับรูปแบบหรือจัดหมวดหมู่ข้อมูล โดย Deep Learning การที่เครื่องจะสามารถเข้าใจสิ่งต่าง ๆ ได้จำเป็นที่จะต้องมีความรู้ก่อน จากนั้นจะประเมินชุดข้อมูลและนำเสนอหรือแทนองค์ความรู้นั้น

2.1.6 Loss Function



รูปที่ 2.6 Loss Function

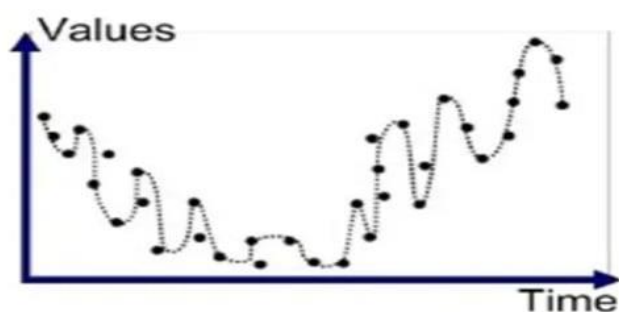
(ที่มา : <https://shorturl.asia/7KWwy>)

จากรูปที่ 2.6 เป็นองค์ประกอบที่สำคัญในกระบวนการเรียนรู้ของโมเดล Machine Learning และ Deep Learning ซึ่งใช้ในการวัดความผิดพลาดระหว่างค่าพยากรณ์ (Prediction) ของโมเดลกับค่าจริง (Actual Value) บทความนี้จะอธิบายพื้นฐานของ Loss Function ประเภทต่างๆ และการประยุกต์ใช้งาน พื้นฐานของ Loss Function เป็นฟังก์ชันที่นำค่าพยากรณ์ของโมเดลและค่าจริงมาเปรียบเทียบกับกัน แล้วส่งกลับค่าความผิดพลาด ซึ่งโมเดลจะใช้ค่าความผิดพลาดนี้ในการปรับปรุงน้ำหนักของโครงข่ายประสาทเทียมผ่านกระบวนการเรียนรู้แบบปรับปรุงอย่างค่อยเป็นค่อยไป (Gradient Descent) วัตถุประสงค์หลักของ Loss Function

2.1.6.1 วัดความผิดพลาด วัดความแตกต่างระหว่างค่าพยากรณ์กับค่าจริง

2.1.6.2 นำทางการเรียนรู้ ช่วยให้โมเดลสามารถปรับปรุงพารามิเตอร์เพื่อให้ได้ผลลัพธ์ที่ดียิ่งขึ้น

2.1.7 Overfitting



รูปที่ 2.7 Overfitting

(ที่มา : <https://shorturl.asia/LICX9>)

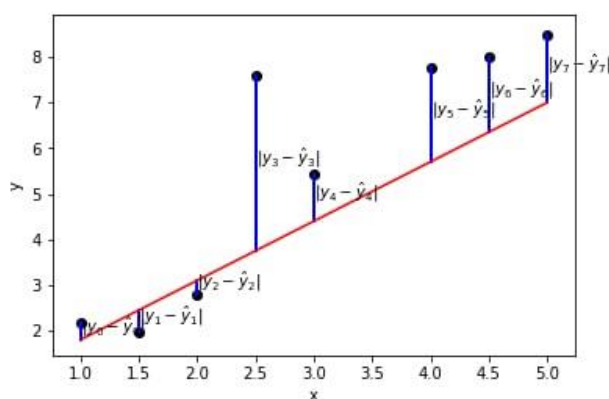
จากรูปที่ 2.7 หมายถึงสถานการณ์ที่โมเดลเรียนรู้ข้อมูลฝึก (Training Data) ได้ดีเกินไปจนสามารถจดจำรายละเอียดหรือรูปแบบเฉพาะเจาะจงในข้อมูลฝึกได้ทั้งหมด รวมถึง "Noise" หรือ "ข้อมูลที่ไม่สำคัญ" ซึ่งอาจไม่ใช่ลักษณะทั่วไปของข้อมูลในโลกความเป็นจริง (Generalization Problem) เมื่อเกิด Overfitting โมเดลจะทำงานได้ดีบนข้อมูลฝึก แต่มีประสิทธิภาพที่ต่ำเมื่อใช้งานกับข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน (Test Data) หรือในสถานการณ์จริง

2.1.7.1 Bias-Variance Tradeoff Overfitting เป็นผลมาจากการที่โมเดลมี Bias ต่ำ (สามารถเรียนรู้รายละเอียดในข้อมูลได้ดีมาก) แต่มี Variance สูง (ไวต่อความเปลี่ยนแปลงในข้อมูล) การเรียนรู้ที่ดีควรหาสมดุลระหว่าง Bias และ Variance เพื่อให้โมเดลมีความสามารถในการทำนายข้อมูลใหม่ได้

2.1.7.2 Capacity ของโมเดล โมเดลที่มีพารามิเตอร์จำนวนมาก (High Capacity) เช่น Neural Networks ที่มีชั้น (Layers) และหน่วยประมวลผล (Neurons) จำนวนมาก มีโอกาสเกิด Overfitting ได้ง่าย เนื่องจากโมเดลสามารถปรับให้เหมาะสมกับข้อมูลฝึกได้ทุกรูปแบบ แม้แต่ส่วนที่ไม่จำเป็น

2.1.7.3 Error Function การวัดความผิดพลาดในโมเดลมักใช้ Training Error และ Testing Error: ใน Overfitting ค่า Training Error ต่ำมาก แต่ Testing Error สูงความแตกต่างนี้แสดงให้เห็นว่าโมเดลไม่สามารถ Generalize กับข้อมูลใหม่ได้

2.1.8 Mean Absolute Error



รูปที่ 2.8 Mean Absolute Error

(ที่มา : <https://shorturl.asia/eBSYO>)

จากรูปที่ 2.8 Mean Absolute Error (MAE) คือ ค่าเฉลี่ยของผลต่างสัมบูรณ์ระหว่างผลเฉลยกับค่าที่เกิดจากการทำนายของ Model โดย Model ที่ใช้ เป็น LossFunction จะทนต่อ Data ที่มี Outlier ได้มากกว่า Model ที่ ใช้ MSE เป็น Loss Function

2.1.9 OpenCV



รูปที่ 2.9 OpenCV

(ที่มา : <https://opencv.org/>)

จากรูปที่ 2.9 คือ โลโก้ ฟังก์ชันการเขียนโปรแกรมโดยส่วนใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ โดย OpenCV สามารถประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่ง

และภาพเคลื่อนไหว เช่น จากภาพ วิดีโอ หรือไฟล์วิดีโอ โดยไม่ยึดติดทางด้านฮาร์ดแวร์ทำให้ OpenCV สามารถ พัฒนาโปรแกรมได้หลายภาษา OpenCV ถูกเขียนขึ้นด้วยภาษา C++ ด้วยให้มีการรองรับ ภาษา Python ,Java เป็นต้น

2.1.10 Ultrasonic Sensor



รูปที่ 2.10 Ultrasonic Sensor

(ที่มา : <https://www.omi.co.th/th/article/ultrasonic-sensor>)

จากรูปที่ 2.1.10 คือ เซ็นเซอร์ที่ใช้สำหรับตรวจจับวัตถุต่างๆ โดยอาศัยหลักการสะท้อนของคลื่นความถี่เสียง และ คำนวณหาค่าระยะทางได้จากการเดินทางของคลื่นและนำมาเทียบกับเวลา

2.1.11 Google Colaboratory หรือ Google Colab



รูปที่ 2.11 Google Colaboratory

(ที่มา : <https://colab.google/>)

จากรูปที่ 2.1.11 คือบริการคลาวด์จากทีม Google Research เป็น IDE ที่อนุญาตให้ผู้ใช้เขียนโค้ดด้วยภาษา Python และรันโมเดล Machine Learning ผ่านเบราว์เซอร์ได้โดยไม่ต้องติดตั้งซอฟต์แวร์บนเครื่องคอมพิวเตอร์ของเราเอง และเนื่องจากการเขียนโค้ดและการรันโมเดลเกิดขึ้นบนคลาวด์ของ Google Colab การรันโมเดลที่ต้องใช้ข้อมูลมหาศาลอย่างการเรียนรู้เชิงลึก DeepLearning ในหลายครั้งจะต้องใช้ Computing Power ค่อนข้างสูง ซึ่งอาจจะทำให้ต้องใช้

เวลามากขึ้นในการวิเคราะห์ข้อมูลในกรณีที่คอมพิวเตอร์มีสเปคไม่สูงนัก Google Colab ก็จะช่วยประหยัดเวลาได้เพราะมี GPU และ TPU หรือ Tensor Processing Unit ที่ช่วยให้การทำงานกับข้อมูลขนาดมหาศาลและการฝึกโมเดลให้เร็วขึ้น

2.2 ตารางเปรียบเทียบสเปคของบอร์ดแต่ละชนิดทางวิศวกรรม

ตารางที่ 2.1 ตารางเปรียบเทียบสเปคของบอร์ดแต่ละชนิดทางวิศวกรรม

คุณสมบัติ	Arduino Uno R4	Raspberry Pi 5	Nvidia Jetson Orin Nano
ประเภทบอร์ด	ไมโครคอนโทรลเลอร์ (MCU)	คอมพิวเตอร์ขนาดเล็ก (Single-board Computer)	คอมพิวเตอร์ AI/ML สำหรับงานขนานสูง
CPU	Renesas RA4M1 (32-bit Arm Cortex-M4) @ 48 MHz	Quad-core Arm Cortex-A76 @ 2.4 GHz	6-core Arm Cortex-A78AE @ 1.5 GHz
GPU	-	VideoCore VII	NVIDIA Ampere (1024 CUDA cores + 32 Tensor cores)
การประมวลผล AI	-	ใช้ได้ระดับเบื้องต้น	40 TOPS (Tera Operations Per Second)
หน่วยความจำ (RAM)	32 KB RAM	4GB หรือ 8GB LPDDR4	8GB LPDDR5
หน่วยความจำ (Flash)	256 KB Flash	MicroSD / PCIe SSD	MicroSD / M.2 NVMe
GPIO	14 Digital, 6 Analog	40 GPIO pins	GPIO ผ่าน I2C, SPI, UART
พอร์ตการเชื่อมต่อ	USB-C, I2C, SPI, UART	USB 3.0, USB-C, HDMI, Ethernet, Wi-Fi 6	USB 3.2, Ethernet, PCIe, I2C, SPI, UART
การประมวลผลภาพ (Vision)	ไม่รองรับ	รองรับระดับเบื้องต้น	รองรับการประมวลผลภาพขั้นสูง (Computer Vision)
ระบบปฏิบัติการ	-	Linux-based OS (เช่น Raspberry Pi OS)	NVIDIA JetPack SDK (Linux-based)
การจ่ายไฟให้บอร์ด	5V (ขั้นต่ำ)	5V USB-C	7V ถึง 20V 5A
การใช้งานที่เหมาะสม	งานควบคุมระบบฝังตัว เช่น PWM, เซนเซอร์, มอเตอร์	งาน IoT, ระบบคอมพิวเตอร์ขนาดเล็ก, กล้อง, เซิร์ฟเวอร์	งาน AI/ML, การประมวลผลภาพ, หุ่นยนต์, การวิเคราะห์ข้อมูล AI
การประมวลผลแบบเรียลไทม์	รองรับ (Real-time control)	ไม่เหมาะกับการประมวลผลเรียลไทม์ขั้นสูง	รองรับการประมวลผลเรียลไทม์ในงาน AI/ML
ราคา	1,280 บาท	3,080 บาท	21,700 บาท

2.2.1. การเปรียบเทียบการใช้งาน

Arduino UNO R4 เหมาะสำหรับงานฝังตัว (Embedded Systems): Arduino UNO R4 มีการพัฒนาให้รองรับการเชื่อมต่อกับเซ็นเซอร์และอุปกรณ์ต่างๆ ได้ง่าย สามารถควบคุมอุปกรณ์ด้วยการเขียนโปรแกรมได้ง่ายๆ ผ่าน MQTT และอนาล็อก ทำให้เหมาะกับโปรเจกต์การควบคุมอุปกรณ์เชิงฟิสิกส์ เช่น มอเตอร์, หลอดไฟ, และ สวิตช์(eTechnophiles).การพัฒนา ระบบ IoT ขนาดเล็ก: รุ่น R4 WiFi มาพร้อมกับการเชื่อมต่อ Wi-Fi และ Bluetooth ทำให้เหมาะสำหรับการสร้าง

ระบบ IoT ที่เชื่อมต่ออินเทอร์เน็ตเพื่อส่งข้อมูลจากเซ็นเซอร์หรือควบคุมอุปกรณ์จากระยะไกล(Tom's Hardware).

Raspberry Pi 5 รองรับการใช้งานทั่วไป เช่น การประมวลผลการทำงาน AI เบื้องต้น การเล่นเกม 4K, การทำงานออฟฟิศ, การพัฒนาเว็บไซต์ขนาดเล็ก และการเขียนโปรแกรม พัฒนาแอปพลิเคชันทั่วไป การใช้งานที่เหมาะสม: เหมาะกับการสร้างคอมพิวเตอร์ราคาถูก.

Jetson Orin Nano เหมาะสำหรับการใช้งานด้าน AI: Jetson Nano Orin มี GPU แบบ Ampere พร้อม CUDA และ Tensor Cores ซึ่งทำให้รองรับการประมวลผล AI อย่างเช่น Machine Learning (ML), Deep Learning, และ Computer Vision ได้ดี เหมาะสำหรับโปรเจกต์ที่ต้องการการคำนวณที่ซับซ้อนและประมวลผลข้อมูลจำนวนมากแบบเรียลไทม์(NVIDIA Developer)(NVIDIA Newsroom). การวิเคราะห์วิดีโอและหุ่นยนต์: Orin สามารถทำงานกับการวิเคราะห์วิดีโอ เช่น การตรวจจับวัตถุในกล้อง, การประมวลผลภาพจากโดรน หรือระบบหุ่นยนต์อัจฉริยะที่ต้องใช้ AI ในการตัดสินใจแบบเรียลไทม์.ทำให้เหมาะสำหรับการสร้างระบบ IoT ที่เชื่อมต่ออินเทอร์เน็ตเพื่อส่งข้อมูลจากเซ็นเซอร์หรือควบคุมอุปกรณ์จากระยะไกล(Tom's Hardware).

2.2 งานวิจัยหรือโครงการที่เกี่ยวข้อง

พุดพิงศ์ ขุนทรง1 , ประณัฐ วิสุวรรณ2 และ วิจิต ฉัตรรัตน์กุลชัย (2554) ได้ทำการพัฒนาการการ หลบหลีกสิ่งกีดขวางเป็นปัจจัยสำคัญเพื่อความปลอดภัย โดยมีการใช้เทคโนโลยีต่าง ๆ เช่น Laserscanner สำหรับตรวจจับระยะห่างและสิ่งกีดขวาง รวมถึงการใช้กล้องวิดีโอในการตรวจจับวัตถุ และคำนวณระยะห่างผ่านเทคนิคการมองเห็นแบบภาพสองตา (Stereo Vision) ซึ่งช่วยเพิ่มความแม่นยำในการแยกแยะวัตถุจากสิ่งแวดล้อม แต่การตรวจจับวัตถุด้วยกล้องและระบบกลจักรวิทัศน์ยังมีความคลาดเคลื่อนอยู่ โดยเฉพาะเมื่อแสงหรือความสว่างของสิ่งแวดล้อมเปลี่ยนไป ทำให้การตรวจจับวัตถุและระยะห่างมีความแม่นยำน้อยลง [1].

ร้อยตรี ตรีรัตน์ เมตต์การุณจิต (2551) งานวิจัยที่เกี่ยวข้องกับการพัฒนารถอัจฉริยะไร้คนขับ โดยมุ่งเน้นการประมวลผลภาพและการนำทางผ่าน GPS รวมถึงการควบคุมการเคลื่อนที่ของรถด้วย เซนเซอร์และมอเตอร์ ข้อมูลจากงานวิจัย Stanford University ที่ใช้ระบบ Laser และกล้องวิดีโอในการตรวจจับถนนและสิ่งกีดขวาง รวมถึงโครงการศึกษาารถอัจฉริยะในประเทศไทย ที่นำระบบกล้องคู่ และการวัดระยะด้วยเลเซอร์มาประยุกต์ใช้ ระบบการหลบหลีกสิ่งกีดขวางและการควบคุมรถแบบอัตโนมัติถูกพัฒนาให้มีความแม่นยำและรวดเร็ว เช่น การใช้ Fuzzy Logic แต่ความแม่นยำในการตรวจจับถนนและสิ่งกีดขวางที่อาจลดลงเมื่อเจอสภาวะแวดล้อมที่ซับซ้อนหรือสภาพแสงที่ไม่แน่นอน

อีกทั้งระบบกล้องคู่ที่ใช้ในการประมวลผลภาพยังมีปัญหาความแม่นยำลดลงเมื่อวัตถุอยู่ไกล นอกจากนี้ การใช้งาน GPS ก็อาจได้รับผลกระทบจากสิ่งปลูกสร้างหรือสภาพแวดล้อมที่ทำให้สัญญาณถูกรบกวน ทำให้การนำทางไม่แม่นยำเท่าที่ควร [2].

Gurjashan Singh Pannu, Mohammad Dawud, Pritha Gupta (2558) ได้ทำการวิจัยเกี่ยวกับการพัฒนาต้นแบบรถยนต์ไร้คนขับด้วยการใช้กล้องและเซ็นเซอร์อัลตราโซนิกเพื่อรับข้อมูลจากสิ่งแวดลอม และประมวลผลด้วย Raspberry Pi เพื่อให้รถสามารถขับเคลื่อนไปยังจุดหมายได้อย่างปลอดภัยโดยหลีกเลี่ยงอุปสรรค มีการใช้เทคนิคต่างๆ เช่น การตรวจจับเลนและการตรวจจับสิ่งกีดขวาง อย่างไรก็ตาม ข้อเสียหลักของระบบนี้คือ ความซับซ้อนในการคำนวณและประมวลผลภาพจากกล้อง ซึ่งอาจทำให้การตอบสนองช้าในสภาพถนนที่ไม่แน่นอน และ Raspberry Pi เป็นเวอร์ชันเก่าอาจทำให้การประมวลผลล่าช้า [3].

Yash Sudhir Shirke, Udayraj Sambhaji Gawade, Karan Kamlakar Jadhav, Mr. M.A. Hawre (2563) ได้ทำการพัฒนารถไร้คนขับโดยใช้ Raspberry Pi และการเรียนรู้ของเครื่อง ซึ่งใช้ Pi Camera ในการฝึกฝนและตรวจจับภาพ, L298 Motor Driver ในการควบคุมมอเตอร์, และ Ultrasonic Sensor ในการคำนวณระยะห่างจากวัตถุ ระบบตรวจจับเลนของถนนโดยอาศัยทั้งเทคนิค Feature-Based และ Model-Based ซึ่งช่วยให้ตรวจจับเลนได้แม่นยำขึ้นและทนทานต่อเสียงรบกวน อย่างไรก็ตาม ข้อเสียของระบบคือ การพึ่งพาสัญญาณภาพและการประมวลผลจากกล้องเพียงอย่างเดียวอาจไม่เสถียรเมื่อสภาพแสงไม่ดี เช่น พื้นที่ที่มีเงาหรือแสงน้อย ซึ่งอาจทำให้การตรวจจับถนนหรือสิ่งกีดขวางผิดพลาดได้ [4].

ณัฐภูมิรินทร์ เขมะทาสี ยุทธนา พิมเสน (2561) พัฒนารถจำลองอัตโนมัติที่ใช้เทคโนโลยี CNN และ Raspberry Pi เพื่อควบคุมการหลบหลีกและหยุดรถ พบว่าการประมวลผลภาพด้วย CNN ช่วยเพิ่มความแม่นยำในการตรวจจับถนนและสิ่งกีดขวาง รวมถึงการใช้ Raspberry Pi ในการควบคุมระบบทำให้สามารถบูรณาการกับเซ็นเซอร์ต่าง ๆ ได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม ข้อจำกัดที่พบคือ ประสิทธิภาพของ Raspberry Pi อาจไม่รองรับการประมวลผลภาพความละเอียดสูงได้ในเวลาเรียลไทม์ ทำให้เกิดความล่าช้าในการตอบสนองและควบคุมรถเมื่อมีข้อมูลเข้ามามากเกินไป นอกจากนี้ สภาพแวดล้อมที่มีแสงน้อยหรือการรบกวนสัญญาณจากอุปกรณ์ต่าง ๆ อาจทำให้ระบบทำงานผิดพลาดได้ [5].

Minghong HU, Hui GUO, Xuyuan JI, (2018) งานวิจัยนี้พัฒนาโมเดล MobileNet-V2 ด้วยแนวคิดการเรียนรู้แบบถ่ายโอน (Transfer Learning) เพื่อใช้ในระบบขับเคลื่อนอัตโนมัติ (Autopilot) บนอุปกรณ์ฝังตัว เช่น Raspberry Pi โดยปรับโครงสร้างโมเดลให้เหมาะกับการทำนายมุมเลี้ยว ลดขนาดและความซับซ้อนของโมเดล ผลการทดลองแสดงว่าโมเดลสามารถทำงานที่ความเร็ว 10 เฟรม

ต่อวินาทีด้วย CPU เท่านั้น มีข้อผิดพลาดเฉลี่ย 5 องศา และสามารถขับเคลื่อนในเลนได้อย่างแม่นยำ ระบบนี้เหมาะสำหรับงานในโรงงานที่ต้องการตรวจสอบแบบเรียลไทม์ เช่น รถขนส่งในสายการผลิต ช่วยลดแรงงานและเพิ่มประสิทธิภาพการทำงานในอุตสาหกรรม [6].

Raj Shirolkar, Rohan Datar, Anushka Dhongade, Gayatri Behere, (2019) งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบรถยนต์ไร้คนขับต้นแบบขนาดเล็กโดยใช้ Raspberry Pi ซึ่งสามารถควบคุมด้วยคำสั่งเสียงหรือผ่านแอปพลิเคชันเว็บ ระบบนี้ประกอบด้วยเซ็นเซอร์อัลตราโซนิกสำหรับตรวจจับสิ่งกีดขวาง กล้อง Pi สำหรับตรวจจับเส้นทางและวัตถุ และไมโครโฟนสำหรับคำสั่งเสียง โดยใช้ฐานข้อมูล SQLite เพื่อเก็บข้อมูลแผนที่ ระบบยังสนับสนุนการหลีกเลี่ยงสิ่งกีดขวางแบบเรียลไทม์ และมีต้นทุนต่ำกว่าโซลูชันอื่น ๆ เหมาะสำหรับการใช้งานทั่วไปในอนาคตของรถยนต์ไร้คนขับ [7].

Yuanqing Suo, Song Chen*, Mao Zheng (2020) งานวิจัยนี้นำเสนอการพัฒนาระบบขับเคลื่อนอัตโนมัติสำหรับรถหุ่นยนต์โดยใช้ Raspberry Pi และโมเดล CNN แบบ Transfer Learning (VGG16) เพื่อทำนายคำสั่งบังคับเลี้ยวจากภาพถนน โมเดลถูกฝึกด้วยข้อมูลที่รวบรวมจากกล้อง Pi Camera พร้อมการปรับแต่งภาพ เช่น การตัดส่วนที่ไม่เกี่ยวข้องและแปลงเป็นขาวดำ ระบบได้รับการทดสอบทั้งในสภาพแวดล้อมออนไลน์และในโลกจริง โดยใช้โครงสร้างพื้นฐานราคาย่อมเยา ความแม่นยำในการทำนายอยู่ที่ประมาณ 80% อย่างไรก็ตาม ระบบยังต้องปรับปรุงเพื่อรองรับอุปสรรคและสภาพแวดล้อมที่หลากหลายมากขึ้นในอนาคต [8].

Abhishek Sarda, Dr.Shubhra Dixit, Dr. Anupama Bhan (2021) งานวิจัยนี้มุ่งเน้นการตรวจจับวัตถุสำหรับการขับอัตโนมัติ โดยใช้ YOLO (You Only Look Once) ซึ่งเป็นอัลกอริธึมการเรียนรู้เชิงลึกสำหรับการตรวจจับวัตถุที่รวดเร็วและแม่นยำ ระบบได้รับการปรับแต่งให้ตรวจจับวัตถุที่เกี่ยวข้องกับถนน เช่น รถยนต์ คน ป้ายจราจร และอาคาร โดยใช้ชุดข้อมูล Open Images Dataset และเพิ่มความแม่นยำผ่านการฝึกอบรมแบบกำหนดเอง ผลการทดลองชี้ให้เห็นว่า YOLO สามารถตรวจจับวัตถุได้ด้วยค่า mAP 74.6% พร้อมปรับปรุงประสิทธิภาพให้เหมาะสมกับการใช้งานจริงในรถยนต์อัตโนมัติ [9].

Khan, S.A.; Lee, H.J. Lim (2023) งานวิจัยนี้เสนอวิธีการแบบผสมผสานในการตรวจจับวัตถุสำหรับรถยนต์ขับเคลื่อนอัตโนมัติ โดยรวมข้อดีของโมเดล YOLO และ Faster R-CNN เข้าด้วยกัน เพื่อเพิ่มความแม่นยำในการจำแนกประเภทและขอบเขตของวัตถุ พร้อมทั้งลดเวลาในการประมวลผล โมเดลนี้ถูกฝึกด้วยภาพถ่ายการจราจร 10,000 ภาพ ซึ่งผลลัพธ์แสดงให้เห็นว่าโมเดลที่

นำเสนอมีความแม่นยำสูงขึ้นประมาณ 5-7% เมื่อเทียบกับ YOLO รุ่นมาตรฐาน นอกจากนี้ โมเดลนี้ยังสามารถใช้งานได้จริงในเวลาประมวลผลแบบเรียลไทม์สำหรับรถยนต์ขับเคลื่อนอัตโนมัติ [10].

ธวัชชัย 1สุรินแก้ว, นนทชัย กายายา, และ รัชตพล บุตรศรีชา โครงการนี้มีวัตถุประสงค์เพื่อพัฒนารถจำลองขับเคลื่อนที่สามารถตรวจจับและปฏิบัติตามสัญญาณไฟจราจรได้ โดยใช้กล้องตรวจจับสัญญาณและโมเดลคนจำลอง จากนั้นประมวลผลเพื่อตัดสินใจควบคุมการเคลื่อนที่ของรถ รถจะหยุดเมื่อเจอไฟแดงหรือคนจำลอง จะล่อเมื่อเจอไฟเหลือง และเคลื่อนที่ต่อเมื่อเจอไฟเขียว ระบบถูกทดสอบในสนามจำลอง และประเมินประสิทธิภาพด้วยค่า Autonomy สูงถึง 99.755% รวมถึงการวัดความแม่นยำของการตรวจจับวัตถุด้วยค่า mAP ที่ 0.710 และ 0.589 จากวิธีการคำนวณที่แตกต่างกัน [11].

บทที่ 3

ขั้นตอนการดำเนินงาน

3.1 วิธีการดำเนินโครงการ

ในการดำเนินโครงการ "รถจำลองอัตโนมัติสำหรับจำลองถนนพร้อมหลบหลีกและหยุดรถ" ทีมนักพัฒนามุ่งเน้นการสร้างระบบจำลองรถอัตโนมัติที่สามารถตรวจจับและตอบสนองต่อวัตถุและสภาพแวดล้อมบนถนนได้อย่างแม่นยำ โดยอาศัยการใช้ปัญญาประดิษฐ์ (AI) รวมถึงการวิเคราะห์ข้อมูลจากเซ็นเซอร์และกล้อง 2 ตัว

3.1.1 การวางแผนโครงการ

- 3.1.1.1. กำหนดวัตถุประสงค์ของโครงการ และ เขตของโครงการ
- 3.1.1.2. กำหนดรายละเอียดของโครงการ
- 3.1.1.3. กำหนดอุปกรณ์ และเครื่องมือ ที่ต้องใช้

3.1.2 ออกแบบโครงสร้างงาน

- 3.1.2.1. ออกแบบระบบตรวจสอบวัตถุ โดย YOLO V8s
- 3.1.2.2. ออกแบบการตรวจจับถนน โดย MobileNet V2
- 3.1.2.3. ออกแบบระบบเซ็นเซอร์อัลตราโซนิก
- 3.1.2.4. ออกแบบการขับอัตโนมัติของรถจำลอง

3.1.3 พัฒนาระบบ

- 3.1.3.1. พัฒนาการตรวจจับวัตถุโดย AI
- 3.1.3.2. พัฒนาการตรวจจับถนนโดย AI
- 3.1.3.3. พัฒนาระบบการขับอัตโนมัติของรถจำลอง
- 3.1.3.4. นำระบบทุกส่วนรวมเข้าด้วยกัน

3.1.4 ทดสอบและแก้ไข

- 3.1.4.1. ทดสอบการทำงานของระบบด้วยสถานการณ์ต่างๆ
- 3.1.4.2. ทำการทดสอบการทำงานของระบบเมื่อใช้งานร่วมกันระหว่างอัลตราโซนิก และ

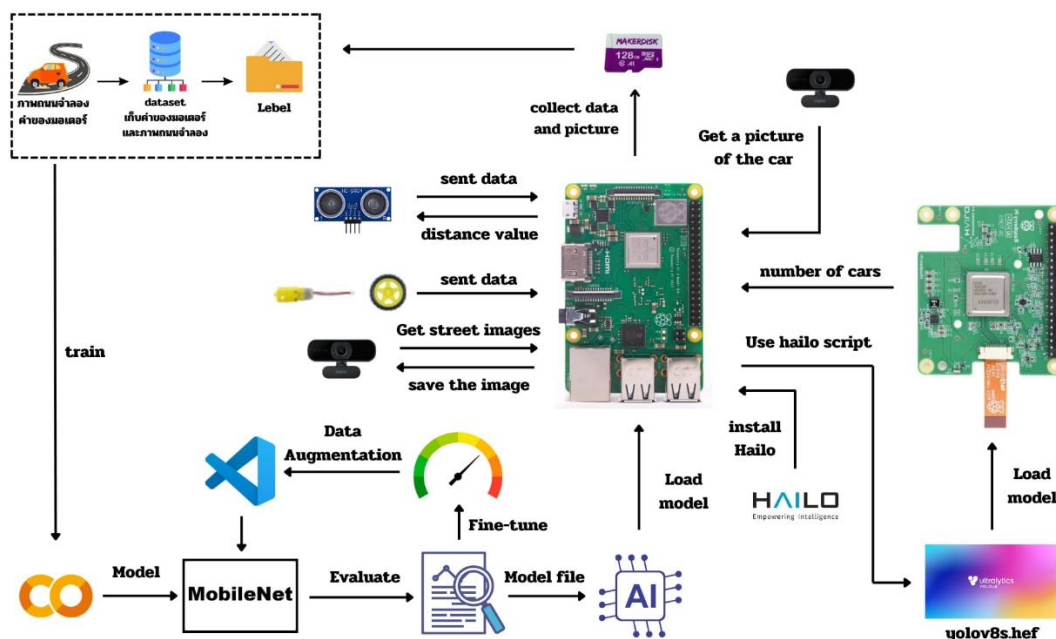
Webcam

- 3.1.4.3 ปรับปรุงประสิทธิภาพการทำงานและแก้ไข

จัดทำรูปเล่มปริญญานิพนธ์

3.1.5. ทำรูปเล่มส่งมอบโครงการ

3.2 หลักการแนวความคิด



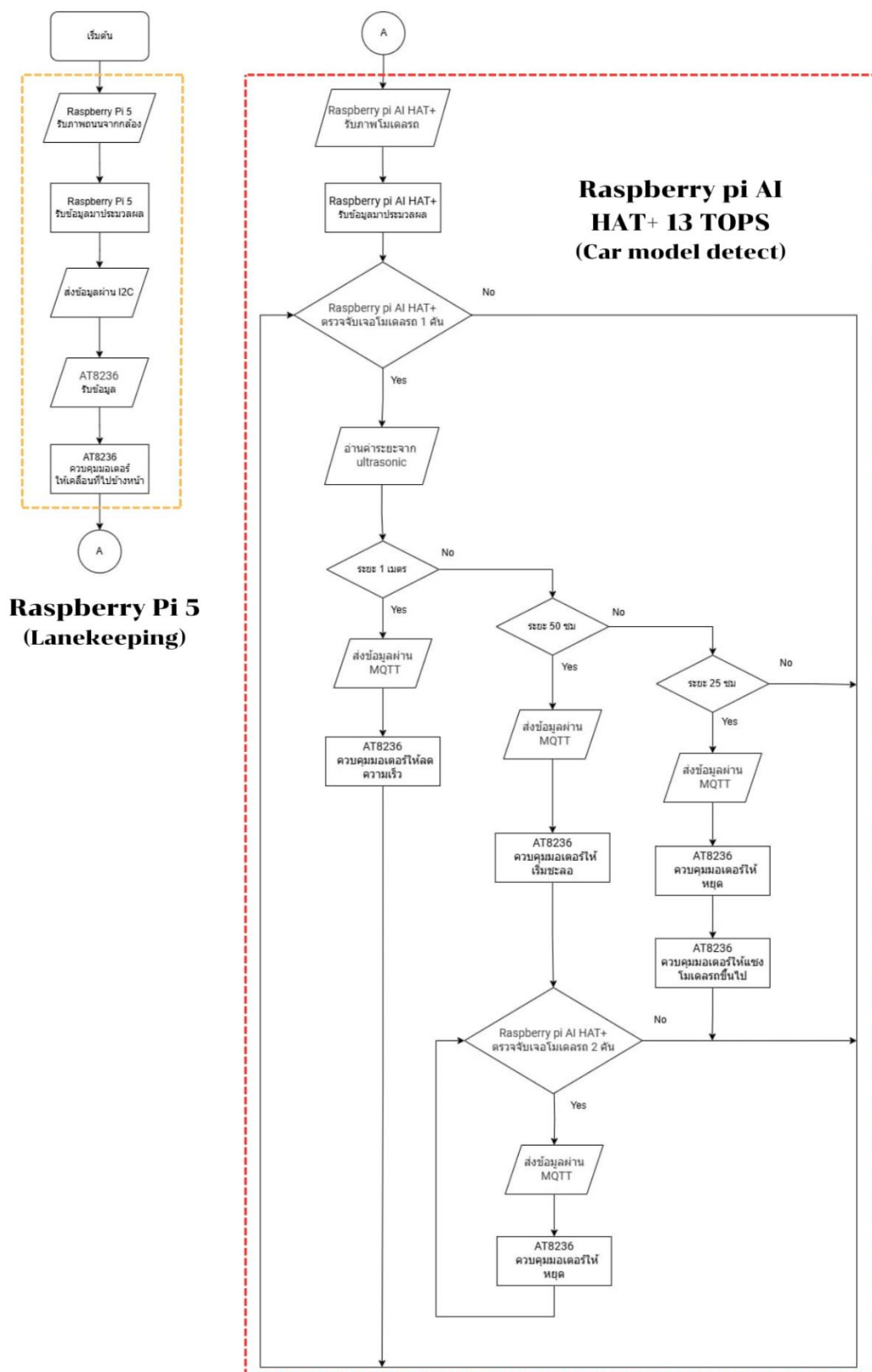
รูปที่ 3.1 แสดงหลักการรถจำลองอัตโนมัติสำหรับถนนจำลองพร้อมหลบหลีกและหยุดรถ

จากรูปที่ 3.1 แสดงหลักการทำงานของระบบ รถจำลองอัตโนมัติสำหรับถนนจำลองพร้อมหลบหลีกและหยุดรถ ระบบนี้ใช้ Raspberry Pi เป็นตัวควบคุมหลัก ซึ่งทำหน้าที่ประมวลผลข้อมูลจากกล้องเว็บแคมโดยมีกล้องสองตัวที่ทำงานร่วมกัน หนึ่งตัวใช้ในการตรวจจับเส้นทาง และอีกตัวสำหรับการตรวจจับรถที่อยู่ด้านหน้า โดยเว็บแคมตัวที่ 1 ใช้ YOLO V8s เพื่อทำการตรวจจับโมเดลรถ ช่วยให้รถจำลองสามารถหลีกเลี่ยงได้อย่างแม่นยำ ในขณะที่ MobileNet V2 จะถูกใช้ในการวิเคราะห์และระบุรูปแบบถนนและเซ็นเซอร์อัลตราโซนิกเมื่อรถจำลองจะเริ่มลดความเร็วเมื่อเจอโมเดลรถอยู่ข้างหน้าในระยะ 1 ม. และจะเริ่มชะลอในระยะ 50 ซม. และจะหยุดรถในระยะ 25 ซม.

3.3 การออกแบบซอฟต์แวร์

3.3.1 การออกแบบโมเดลตรวจถนน

การออกแบบโมเดลตรวจถนนเริ่มจากการเก็บข้อมูลที่ครอบคลุม โดยใช้จอยสติ๊กเชื่อมกับ Raspberry Pi เพื่อบันทึกค่ามอเตอร์รถจำลองและภาพถนน ชุดข้อมูลครอบคลุมทางตรง ทางโค้ง และสภาพแวดล้อมที่ต่างกัน เช่น แสงสว่างน้อยหรือกลางแจ้ง จากนั้นแบ่งข้อมูลเป็นชุดฝึกและทดสอบอย่างเหมาะสม ตรวจสอบคุณภาพข้อมูล เช่น ความคมชัดและความสมดุลของประเภทข้อมูล ป้องกันการเรียนรู้ที่ลำเอียงและเพิ่มประสิทธิภาพโมเดล โดยใช้ภาพและคำอธิบายที่มีชื่อไฟล์ตรงกัน



รูปที่ 3.2 โครงสร้างโปรแกรมตรวจจับถนนและโมเดลรถ

จากรูปที่ 3.2 แสดงกระบวนการทำงานของตรวจจับทำงานของรถจำลองอัตโนมัติโดยเริ่มต้นมา Raspberry Pi 5 จะรับภาพมาจากถนนโดยใช้กล้อง Webcam ตัวที่ 1 ส่งข้อมูลผ่าน MQTT และตัวมอเตอร์ AT8236 จะรับข้อมูลมาและสั่งควบคุมมอเตอร์ โดยจะตรวจสอบว่ามีโมเดลรถอยู่หรือไม่ หากมีโมเดลรถอยู่ด้านหน้าตัวรถจำลอง Raspberry Pi 5 รับภาพมา ส่งข้อมูลผ่าน GPIO และตัวมอเตอร์ ที่ถูกควบคุมโดย AT8236 จะทำให้รถจำลองชะลอและหยุด และควบคุมมอเตอร์ให้แซงโมเดลรถ แต่ถ้าหาก ไม่ AT8236 ก็จะควบคุมมอเตอร์ให้ขับต่อไปตามถนนจำลอง Raspberry Pi 5 รับภาพมา หากมีโมเดลในเลนทั้ง 2 เลนจะส่งข้อมูลผ่าน MQTT เพื่อให้ AT8236 ควบคุมมอเตอร์ให้หยุด

3.3.2 การออกแบบการเก็บ Dataset สำหรับ MobileNet V2

3.3.2.1. Data Collection Main

```
import WebcamModule as wM
import DataCollectionModule as dcM
import JoyStickModule as jsM
import YB_Pcb_Car
from time import sleep

# Motor and joystick configuration
car = YB_Pcb_Car.YB_Pcb_Car()
maxThrottle = 175 # Maximum speed
throttle_sensitivity = 0.5
steering_sensitivity = 0.3 # Reduced steering sensitivity for better control
record = False # Start in recording mode off

while True:
    joyVal = jsM.getJS() # Get joystick values
    steering = joyVal['axis3'] # Steering control
    throttle = -joyVal['axis2'] * maxThrottle * throttle_sensitivity # Inverted throttle for correct direction

    # Toggle recording with 'share' button
    if joyVal['share'] == 1:
        record = not record
        print('Recording Started' if record else 'Recording Stopped')
```

รูปที่ 3.3 Code Data Collection Main

จากรูปที่ 3.3 Data Collection Main รับค่าจากจอยสติ๊กเพื่อนำไปคำนวณค่า Throttle และ Steering สำหรับควบคุมรถบังคับแบบอัตโนมัติ จากนั้นใช้ค่าที่คำนวณได้กำหนดความเร็วของล้อซ้ายและขวาแล้วส่งไปยังมอเตอร์ผ่าน Car.Control_Car() หากรถเคลื่อนที่ ระบบจะบันทึกภาพจากกล้องและค่าการควบคุมลงฐานข้อมูลโดยใช้ DcM.SaveData() นอกจากนี้สามารถกดปุ่ม 'Share' เพื่อเริ่มหรือหยุดการบันทึกข้อมูล และกดปุ่ม 'Window' เพื่อบันทึก Log ไฟล์และหยุด

การบันทึกข้อมูลทั้งหมด ระบบทำงานในลูปต่อเนื่องโดยมีการหน่วงเวลาเล็กน้อยในแต่ละรอบเพื่อให้การควบคุมราบรื่น

3.3.2.2. DataCollectionModule

```
import pandas as pd
import os
import cv2
from datetime import datetime

global imgList, steeringList, leftSpeedList, rightSpeedList, imgCounter
countFolder = 0
imgList = []
steeringList = []
leftSpeedList = []
rightSpeedList = []
imgCounter = 0 # Image counter for ordered naming

# GET CURRENT DIRECTORY PATH
myDirectory = os.path.join(os.getcwd(), 'DataCollected')

# CREATE A NEW FOLDER BASED ON THE PREVIOUS FOLDER COUNT
while os.path.exists(os.path.join(myDirectory, f'IMG{str(countFolder)}')):
    countFolder += 1
newPath = os.path.join(myDirectory, f"IMG{countFolder}")
os.makedirs(newPath, exist_ok=True)

# SAVE IMAGES IN THE FOLDER
def saveData(img, steering, left_speed, right_speed):
    global imgList, steeringList, leftSpeedList, rightSpeedList, imgCounter
    if img is None:
        print("Error: Image capture failed, not saving data.")
        return
    # Create ordered file name based on imgCounter
    fileName = os.path.join(newPath, f'Image_{imgCounter:04d}.jpg')
    cv2.imwrite(fileName, img)
    imgList.append(fileName)
    steeringList.append(steering)
    leftSpeedList.append(left_speed)
    rightSpeedList.append(right_speed)
    imgCounter += 1 # Increment the image counter for the next image
```

รูปที่ 3.4 Code DataCollectionModule

จากรูปที่ 3.4 Code DataCollectionModule ใช้บันทึกข้อมูลภาพและค่าการควบคุมรถบังคับ โดยสร้างโฟลเดอร์ใหม่สำหรับเก็บภาพและ Log ข้อมูลทุกครั้งที่มีการบันทึก เมื่อรับภาพจากกล้องสำเร็จจะบันทึกภาพลงในโฟลเดอร์พร้อมเก็บค่า Steering และความเร็วของล้อซ้าย-ขวา ข้อมูลทั้งหมดถูกจัดเก็บในรายการและบันทึกลงไฟล์ CSV เมื่อลูปการทำงานสิ้นสุด หากกล้องไม่

สามารถเปิดหรือจับภาพได้ ระบบจะแสดงข้อความผิดพลาดและหยุดการทำงาน ผู้ใช้สามารถกด 'q' เพื่อออกจากโปรแกรมและบันทึก Log สุดท้ายก่อนปิดกล้องและหน้าต่างแสดงผล

3.3.2.3. JoyStickModule

```
import pygame
from time import sleep

pygame.init()
controller = pygame.joystick.Joystick(0)
controller.init()
buttons = {'A': 0, 'B': 0, 'X': 0, 'Y': 0,
           'L1': 0, 'R1': 0, 'window': 0, 'share': 0,
           'share': 0, 'options': 0,
           'axis1': 0., 'axis2': 0., 'axis3': 0., 'axis4': 0.}
axiss = [0., 0., 0., 0., 0., 0.]

def getJS(name=''):
    global buttons
    # retrieve any events ...
    for event in pygame.event.get(): # Analog Sticks
        if event.type == pygame.JOYAXISMOTION:
            axiss[event.axis] = round(event.value, 2)
        elif event.type == pygame.JOYBUTTONDOWN: # When button pressed
            # print(event.dict, event.joy, event.button, 'PRESSED')
            for x, (key, val) in enumerate(buttons.items()):
                if x < 10:
                    if controller.get_button(x): buttons[key] = 1
        elif event.type == pygame.JOYBUTTONUP: # When button released
            # print(event.dict, event.joy, event.button, 'released')
            for x, (key, val) in enumerate(buttons.items()):
                if x < 10:
                    if event.button == x: buttons[key] = 0

    # to remove element 2 since axis numbers are 0 1 3 4
    buttons['axis1'], buttons['axis2'], buttons['axis3'], buttons['axis4'] =
    if name == '':
        return buttons
    else:
        return buttons[name]
```

รูปที่ 3.5 Code JoyStickModule

จากรูปที่ 3.5 Pygame เพื่ออ่านค่าจากจอยสติ๊กและแปลงข้อมูลปุ่มและแกนอนาล็อกเป็นพิกัดขั้วกรม โดยเริ่มต้นด้วยการกำหนดค่าปุ่มและแกนทั้งหมดเป็นศูนย์ ฟังก์ชัน getJS()

จะดึงข้อมูลเหตุการณ์จากจอยสติ๊ก เช่น การกดปุ่มหรือการขยับแกนอนาล็อก แล้วอัปเดตค่าตามลำดับ เมื่อรันโค้ดจะเรียกใช้ main() ในลูปหลักเพื่อพิมพ์ค่าการควบคุมที่ได้รับทุก ๆ 0.05 วินาที

3.3.2.4. test

```
import tf.lite_runtime.interpreter as tflite
import cv2
import numpy as np
from YB_Pcb_Car import YB_Pcb_Car # นำเข้าโมดูลควบคุมรถ

# โหลดโมเดล TensorFlow Lite
model_path = '/home/pi/Downloads/final_model1.tflite' # เปลี่ยนเป็นไฟล์ .tflite
interpreter = tflite.Interpreter(model_path=model_path)
interpreter.allocate_tensors()

# ขนาดภาพที่ใช้กับโมเดล
IMG_SIZE = 224

# ฟังก์ชันสำหรับประมวลผลภาพ
def preprocess_image(frame):
    img = cv2.resize(frame, (IMG_SIZE, IMG_SIZE)) # ปรับขนาดภาพให้ตรงกับโมเดล
    img = img / 255.0 # ปรับภาพให้เป็น [0, 1] (ถ้าจำเป็น)
    return np.expand_dims(img, axis=0) # เพิ่ม batch dimension

# ฟังก์ชันควบคุมรถ
def control_car(steering, left_speed, right_speed, car):
    """
    แปลงค่าจากโมเดลเป็นคำสั่งควบคุม Yahboom Car
    """
    try:
        # กำหนดทิศทางและความเร็วตาม output ของโมเดล
        left_direction = 1 if left_speed >= 0 else 0
        right_direction = 1 if right_speed >= 0 else 0

        # ส่งคำสั่งไปยังรถ
        car.Ctrl_Car(
            left_direction, int(abs(left_speed)),
            right_direction, int(abs(right_speed))
        )
    except Exception as e:
        print(f"Error in control_car: {e}")
```

รูปที่ 3.6 Code test

จากรูปที่ 3.6 TensorFlow Lite บน Raspberry Pi เพื่อควบคุมรถบังคับโดยอัตโนมัติผ่านการประมวลผลภาพจากกล้อง โมเดลที่โหลดมาจะรับภาพจากกล้อง ขนาด 224x224 พิกเซล แล้วปรับให้อยู่ในช่วง 0-1 ก่อนป้อนเข้าโมเดล โมเดลจะพยากรณ์ค่าการบังคับรถ ได้แก่ การ

เลี้ยว ความเร็วล้อซ้าย และขวา จากนั้นค่าที่ได้จะถูกส่งไปควบคุมรถ Yahboom ผ่านฟังก์ชัน Control_Car() รูปหลักทำงานอย่างต่อเนื่องจนกว่าผู้ใช้จะกด 'q' เพื่อหยุด จากนั้นระบบจะปิดกล้อง และหยุดรถโดยอัตโนมัติ

3.3.2.5. WebcamModule

```
import cv2
import sys
import os

os.environ["SDL_VIDEODRIVER"] = "dummy"

cap = cv2.VideoCapture(0)
if not cap.isOpened():
    print("Error: Could not open camera.")
    sys.exit()

def getImg(display=False, size=[480, 240]):
    ret, img = cap.read()
    if not ret:
        print("Error: Failed to capture image.")
        return None
    img = cv2.resize(img, (size[0], size[1]))
    if display:
        cv2.imshow('IMG', img)
    return img

if __name__ == '__main__':
    while True:
        img = getImg(True)
        if img is None:
            break
    cap.release()
    cv2.destroyAllWindows()
```

รูปที่ 3.7 Code Webcam Module

จากรูปที่ 3.7 OpenCV เพื่อเปิดกล้องและจับภาพวิดีโอแบบเรียลไทม์ โดยกำหนดตัวแปร cap เพื่อเปิดใช้งานกล้อง หากเปิดไม่สำเร็จจะหยุดโปรแกรม ฟังก์ชัน GetImg() ใช้สำหรับจับ

ภาพจากกล้อง ปรับขนาด และเลือกแสดงผลภาพตามต้องการ เมื่อรันโค้ดจะเข้าสู่ลูปที่เรียกใช้ GetImg(True) อย่างต่อเนื่องจนกว่าการจับภาพจะล้มเหลวหรือผู้ใช้ปิดโปรแกรม สุดท้ายระบบจะปล่อยทรัพยากรของกล้องและปิดหน้าต่างแสดงผล

3.3.2.6. YB_PcB_car.py

สร้างคลาส YB_Pcb_Car เพื่อควบคุมการเคลื่อนที่ของรถผ่าน I2C โดยสามารถสั่งให้รถวิ่ง เดินหน้า ถอยหลัง เลี้ยว หมุน และควบคุมมอเตอร์

```
#!/usr/bin/env python
# coding: utf-8
import smbus
import time
import math
class YB_Pcb_Car(object):

    def get_i2c_device(self, address, i2c_bus):
        self._addr = address
        if i2c_bus is None:
            return smbus.SMBus(1)
        else:
            return smbus.SMBus(i2c_bus)

    def __init__(self):
        # Create I2C device.
        self._device = self.get_i2c_device(0x16, 1)

    def write_u8(self, reg, data):
        try:
            self._device.write_byte_data(self._addr, reg, data)
        except:
            print ('write_u8 I2C error')

    def write_reg(self, reg):
        try:
            self._device.write_byte(self._addr, reg)
        except:
            print ('write_u8 I2C error')

    def write_array(self, reg, data):
        try:
            # self._device.write_block_data(self._addr, reg, data)
            self._device.write_i2c_block_data(self._addr, reg, data)
        except:
            print ('write_array I2C error')
```

รูปที่ 3.8 YB_PcB_car.py

จากรูปที่ 3.8 โปรแกรมที่พัฒนาขึ้นนี้เป็นโปรแกรมควบคุมรถผ่านอินเทอร์เฟซ I2C โดยใช้ภาษา Python และโมดูล smbus เพื่อส่งคำสั่งไปยังบอร์ดควบคุมของรถยนต์ โดยสามารถสั่งให้รถเคลื่อนที่ไปในทิศทางต่างๆ รวมถึงควบคุมเซอร์โวมอเตอร์ได้

เพื่อพัฒนาโปรแกรมควบคุมรถผ่าน I2C เพื่อให้สามารถควบคุมการเคลื่อนที่ของรถได้ เช่น เดินหน้า ถอยหลัง เลี้ยวซ้าย เลี้ยวขวา และหมุน เพื่อควบคุมเซอร์โวมอเตอร์ให้สามารถหมุนไปยังมุมที่ต้องการได้ หลักการทำงานของโปรแกรม โปรแกรมนี้ใช้ I2C ในการสื่อสารกับบอร์ดควบคุมรถผ่านคลาส YB_Pcb_Car ซึ่งมีฟังก์ชันหลักดังนี้

ฟังก์ชัน get_i2c_device(address, i2c_bus): ใช้เปิดการเชื่อมต่อกับอุปกรณ์ I2C ที่อยู่บนบัสที่กำหนด

1) การเขียนค่าลงทะเบียน I2C

write_u8(reg, data): เขียนข้อมูลลงทะเบียน I2C

write_reg(reg): เขียนค่าลงทะเบียนโดยไม่ต้องกำหนดข้อมูลเพิ่มเติม

write_array(reg, data): ส่งข้อมูลเป็นชุดไปยังอุปกรณ์ I2C

2) การควบคุมการเคลื่อนที่ของรถ

Ctrl_Car(l_dir, l_speed, r_dir, r_speed): กำหนดทิศทางและความเร็วของล้อซ้ายและขวา

Control_Car(speed1, speed2): คำนวณทิศทางและส่งคำสั่งไปที่ Ctrl_Car

Car_Run(speed1, speed2): เดินหน้าด้วยความเร็วที่กำหนด

Car_Back(speed1, speed2): ถอยหลัง

Car_Left(speed1, speed2), Car_Right(speed1, speed2): เลี้ยวซ้ายและขวา

Car_Spin_Left(speed1, speed2), Car_Spin_Right(speed1, speed2): หมุนซ้ายและขวา

Car_Stop(): หยุดการเคลื่อนที่ของรถ

3) การควบคุมเซอร์โวมอเตอร์

Ctrl_Servo(id, angle): ควบคุมเซอร์โวมอเตอร์ให้หมุนไปยังองศาที่ต้องการ

โดยจำกัดค่าระหว่าง 0 ถึง 180 องศา

3.3.2 ฝึก AI ด้วยดาต้าเซตสำหรับรถขับอัตโนมัติโดย MobileNet V2



รูปที่ 3.9 Dataset ถนนจำลอง

จากรูปที่ 3.9 Dataset รูปถนนจำลองที่ใช้ในการ เทรน MobileNet V2 โดยมีจำนวน 16,861 ภาพ โดยที่ขนาด 240*120

Index	Image	Steering	Left_Speed	Right_Speed
2500	Image_2500.jpg	0.33	152	102
2501	Image_2501.jpg	0.32	151	103
2502	Image_2502.jpg	0.32	151	103
2503	Image_2503.jpg	0.32	151	103
2504	Image_2504.jpg	0.32	151	103
2505	Image_2505.jpg	0.32	151	103
2506	Image_2506.jpg	0.32	151	103
2507	Image_2507.jpg	0.32	151	103
2508	Image_2508.jpg	0.32	151	103
2509	Image_2509.jpg	0.32	151	103
2510	Image_2510.jpg	0.32	151	103
2511	Image_2511.jpg	0.32	151	103
2512	Image_2512.jpg	0.32	151	103
2513	Image_2513.jpg	0.32	151	103
2514	Image_2514.jpg	0.32	151	103
2515	Image_2515.jpg	0.32	151	103
2516	Image_2516.jpg	0.41	158	96
2517	Image_2517.jpg	0.41	158	96
2518	Image_2518.jpg	0.41	158	96
2519	Image_2519.jpg	0.43	160	94
2520	Image_2520.jpg	0.43	160	94
2521	Image_2521.jpg	0.43	160	94
2522	Image_2522.jpg	0.44	161	93
2523	Image_2523.jpg	0.49	164	90
2524	Image_2524.jpg	1.0	204	51

รูปที่ 3.10 ค่า Steering, Left_Speed, และ Right_Speed ในไฟล์ .csv

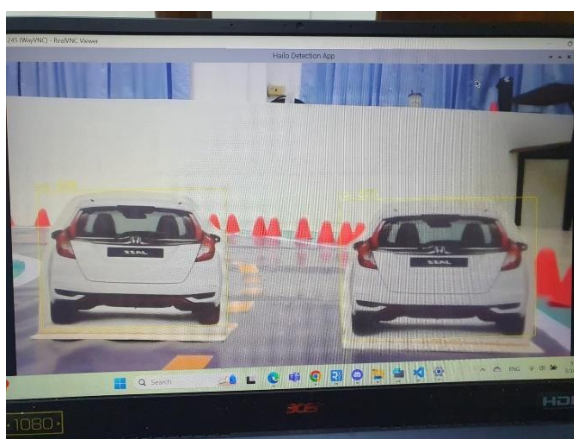
จากรูปที่ 3.10 ภาพนี้เป็นตารางข้อมูลที่ประกอบด้วยคอลัมน์หลัก ได้แก่ index, Image, Steering, Left_Speed, และ Right_Speed ซึ่งแสดงค่าการควบคุมความเร็วล้อซ้าย-ขวา (Left_Speed, Right_Speed)

3.3.3 Detect Car Model



รูปที่ 3.11 Detect car model

จากรูปที่ 3.11 Detect car model เป็นการตรวจสอบโมเดลโดย Hailo Model Zoo ทำงานร่วมกับไฟล์ TFRecord ซึ่งจัดเก็บรูปภาพและป้ายกำกับของชุดข้อมูลสำหรับการประเมินและการปรับเทียบคำแนะนำเกี่ยวกับวิธีการสร้างไฟล์ TFRecord ตามค่าเริ่มต้น ชุดข้อมูลจะถูกเก็บไว้ในเส้นทาง ชุดข้อมูลภายในที่มีภาพ 5000 ภาพภาพ RGB ขนาด 1080x1920x3 การปรับขนาดรูปภาพเป็น 640x640x3 เกิดขึ้นบนชิป



รูปที่ 3.12 การตรวจจับยานพาหนะ

จากรูปที่ 3.12 เครือข่ายการตรวจจับยานพาหนะ (YOLOv8s_Vehicles) ของ Hailo

ใช้ YOLOv8s และได้รับการฝึกอบรมภายในองค์กรด้วยคลาสเดียว สามารถทำงานภายใต้สภาพอากาศและสภาพแสงที่หลากหลาย และมุมมองที่หลากหลาย

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import os
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')

# กำหนด path ของไฟล์และโฟลเดอร์
base_path = '/content/drive/MyDrive/log/lane'
csv_file = os.path.join(base_path, 'evening2.csv')
image_folder = os.path.join(base_path, 'evening2')

# โหลดข้อมูลจาก CSV
try:
    df = pd.read_csv(csv_file)
    print("ข้อมูลใน CSV file:")
    print(df.head())
    print("\nColumns in CSV:", df.columns.tolist())
except FileNotFoundError:
    print(f"ไม่พบไฟล์ CSV ที่ path: {csv_file}")
except Exception as e:
    print(f"เกิดข้อผิดพลาดในการอ่านไฟล์ CSV: {str(e)}")

# ตรวจสอบและแสดงรูปภาพจากโฟลเดอร์
def display_images(folder_path, num_images=5):
    try:
        # ตรวจสอบว่าโฟลเดอร์มีอยู่จริง
        if not os.path.exists(folder_path):
            print(f"ไม่พบโฟลเดอร์รูปภาพที่ path: {folder_path}")
            return

        # รับรายการไฟล์ภาพทั้งหมด
        image_files = [f for f in os.listdir(folder_path)
                        if f.lower().endswith(('.png', '.jpg', '.jpeg'))]
```

รูปที่ 3.13 โค้ดข้อมูลจากไฟล์ CSV ใน Google Colab

จากรูปที่ 3.13 โค้ดนี้โหลดข้อมูลจากไฟล์ CSV และแสดงรูปภาพจากโฟลเดอร์ใน Google Drive โดยใช้ Google Colab ตรวจสอบไฟล์ CSV และแสดงตัวอย่างข้อมูล จากนั้นดึงรูปภาพที่เกี่ยวข้องและใช้ Matplotlib แสดงตัวอย่าง 5 รูปแรก พร้อมจัดการข้อผิดพลาดหากไฟล์ไม่พบหรือโหลดไม่ได้

```

import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img, img_to_array, ImageDataGenerator
from tensorflow.keras.layers import (GlobalAveragePooling2D, Dense, Dropout, Input,
                                     DepthwiseConv2D, BatchNormalization, ReLU)
from tensorflow.keras.models import Model
from tensorflow.keras.regularizers import l2
from tensorflow.keras.optimizers import Adam, Adamax, AdamW
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt

# Parameters
IMG_SIZE = 224
BATCH_SIZE = 32 # Updated batch size
EPOCHS = 50
LEARNING_RATE = 0.0001

# Function to preprocess images
def load_and_preprocess_image(image_path):
    img = load_img(image_path, target_size=(IMG_SIZE, IMG_SIZE))
    img_array = img_to_array(img)
    img_array = tf.keras.applications.mobilenet_v2.preprocess_input(img_array)
    return img_array

# Function to create a modified MobileNetV2 model
def create_custom_mobilenetv2_model():
    base_model = tf.keras.applications.MobileNetV2(
        input_shape=(IMG_SIZE, IMG_SIZE, 3),
        include_top=False,
        weights="imagenet"
    )
    base_model.trainable = True

    for layer in base_model.layers[:-20]:
        layer.trainable = False

```

รูปที่ 3.14 โค้ดเทรนใน Google Colab ก

```

# Function to plot training history
def plot_training_history(history):
    plt.figure(figsize=(12, 6))
    plt.subplot(1, 2, 1)
    plt.plot(history.history['loss'], label='Train Loss')
    plt.plot(history.history['val_loss'], label='Validation Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.title('Loss Over Epochs')
    plt.legend()

    plt.subplot(1, 2, 2)
    plt.plot(history.history['mae'], label='Train MAE')
    plt.plot(history.history['val_mae'], label='Validation MAE')
    plt.xlabel('Epochs')
    plt.ylabel('Mean Absolute Error')
    plt.title('MAE Over Epochs')
    plt.legend()

```

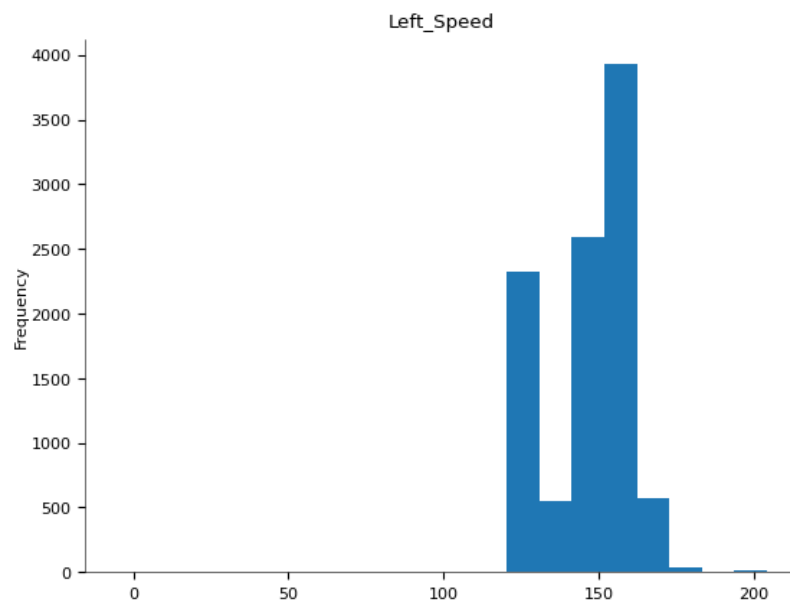
รูปที่ 3.15 โค้ดเทรนใน Google Colaboratory ข

จากรูปที่ 3.14 และ 3.15 โค้ดนี้ใช้ TensorFlow และ MobileNetV2 เพื่อสร้าง โมเดล Deep Learning สำหรับทำนาย ค่าพวงมาลัย (Steering) และความเร็วล้อซ้าย-ขวา ของรถจำลองขับอัตโนมัติ โดยใช้ ภาพจากกล้อง และ ข้อมูลจาก CSV เป็นอินพุตการตั้งค่าพารามิเตอร์ ขนาดภาพ (IMG_SIZE = 224), ขนาดแบตช์ (BATCH_SIZE = 32), จำนวนรอบ (EPOCHS = 50), อัตราการเรียนรู้ (LEARNING_RATE = 0.0001) โหลดและเตรียมข้อมูลภาพอ่านข้อมูลภาพจาก CSV (evening2.csv) ใช้ load_img และ img_to_array โหลดภาพและปรับขนาดใช้ train_test_split แบ่งข้อมูลเป็นชุด train และ validation (80:20) สร้างโมเดล Custom MobileNetV2 ใช้ MobileNetV2 (Pretrained) แต่ Freeze 20 Layer แรกเพิ่ม DepthwiseConv2D, BatchNormalization, ReLU, GlobalAveragePooling2D เพิ่ม Dense Layer 256 และ 128 units พร้อม Dropout เอาต์พุตเป็น Dense(3, activation='linear') เพื่อทำนายค่า [Steering, Left_Speed, Right_Speed] กำหนด Loss Function ใช้ Weighted Loss Function เพื่อให้ค่า Steering มีผลมากกว่าความเร็ว สร้าง Data Generator ใช้ ImageDataGenerator เพื่อเพิ่มข้อมูล ด้วย Rotation, Zoom, Brightness, Flip ฝึกโมเดลใช้ AdamW เป็น Optimizer ใช้ EarlyStopping เพื่อลด Overfitting ใช้ LearningRateScheduler ปรับค่า Learning Rate ทุก 10 Epoch บันทึกโมเดลทั้ง SavedModel Format และ .h5 ประเมินผลและทดสอบโมเดลใช้ model.evaluate() วัดค่า Loss และ Mean Absolute Error ใช้ model.predict() ทำนายค่า [Steering, Left_Speed, Right_Speed] จากภาพทดสอบ

164	90	127	127
164	90	127	127
163	91	127	127
163	91	127	127
164	90	127	127
161	93	127	127

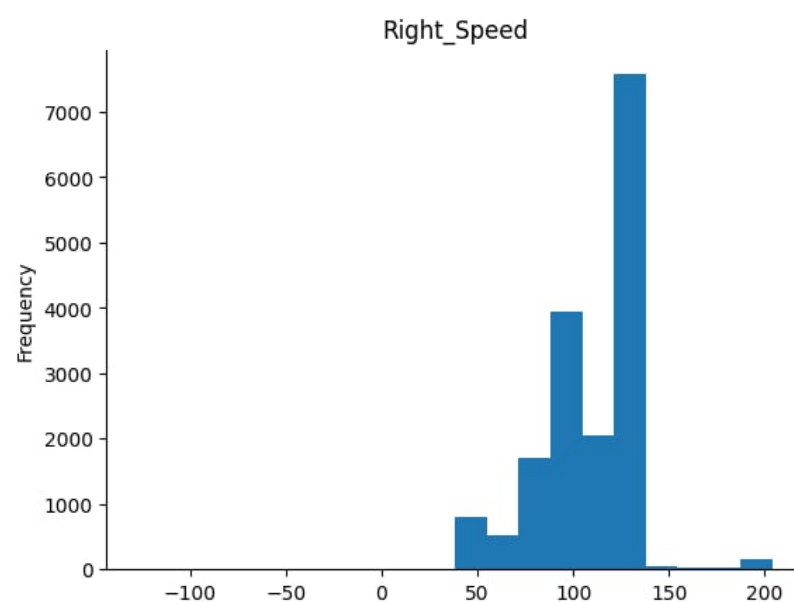
รูปที่ 3.16 ความเร็วของมอเตอร์ที่ใช้เลี้ยวขวาและทางตรง

จากรูปที่ 3.16 ทางโค้งไปจนถึงทางตรงอีกรอบโดยการเลี้ยวจะใช้ความเร็วรอบของมอเตอร์ฝั่งซ้ายที่เร็วกว่าฝั่งขวาหากกล้องเห็นวิเคราะห์เป็นทางตรงความเร็วของมอเตอร์จะเท่ากันทั้งซ้ายและขวา



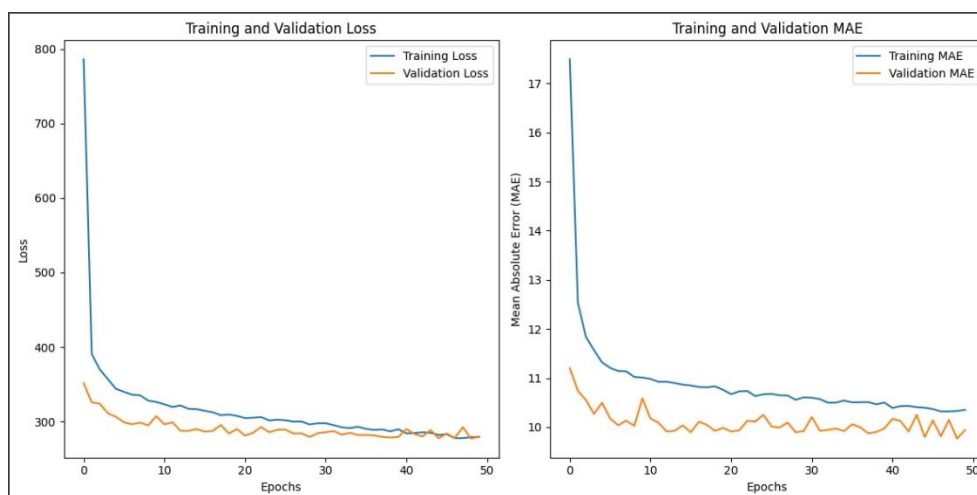
รูปที่ 3.17 กราฟการเลี้ยวซ้าย

จากรูปที่ 3.17 กราฟนี้เป็นฮิสโตแกรมที่แสดงการกระจายความถี่ของค่าตัวแปร **Left_Speed** แกน X ค่าความเร็ว Left_Speed มีค่าตั้งแต่ประมาณ 0 ถึง 200 ค่าที่พบบ่อยอยู่ในช่วงประมาณ 120-180 แกน Y (ความถี่ของข้อมูล) ค่าความถี่สูงสุดอยู่ที่ประมาณ 4000 ลักษณะของข้อมูลข้อมูลส่วนใหญ่อยู่ในช่วงแคบ (120-180)ข้อมูล **Left_Speed** มีการกระจายตัวไม่กว้างมาก ส่วนใหญ่อยู่ในช่วง 120-180 และมีค่าสูงสุดประมาณ 4000 ครั้งในบางช่วงของข้อมูล



รูปที่ 3.18 กราฟการเลี้ยวขวา

จากรูปที่ 3.18 กราฟนี้เป็นฮิสโตแกรมของตัวแปร **Right_Speed** ซึ่งแสดงการกระจายตัวของข้อมูลค่าความเร็วทางขวา โดยมีแกนนอนแทนค่า **Right_Speed** และแกนตั้งแทนความถี่ของแต่ละช่วงค่า ข้อมูลส่วนใหญ่อยู่ในช่วงประมาณ 50-150 โดยมีการกระจุกตัวสูงสุดที่ช่วง 100-150 นอกจากนี้ยังมีค่าความเร็วบางส่วนที่อยู่นอกช่วงนี้ รวมถึงค่าผิดปกติที่ต่ำกว่าศูนย์ ซึ่งอาจเกิดจากข้อผิดพลาดของข้อมูล

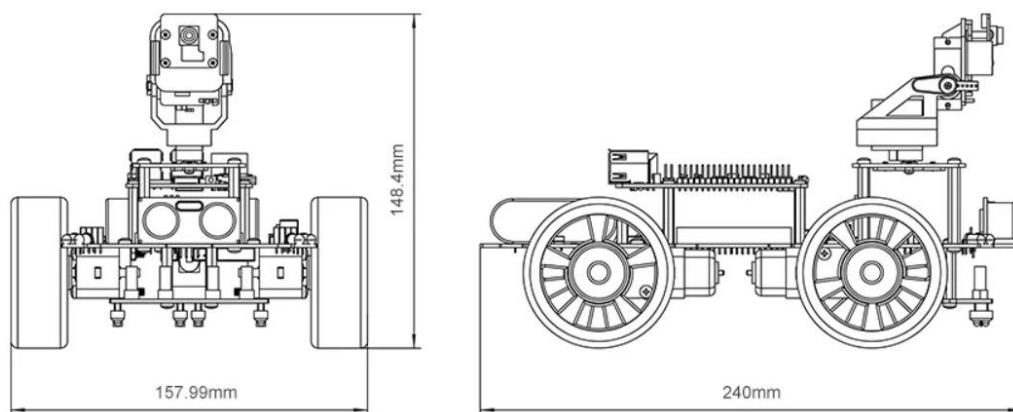


รูปที่ 3.19 กราฟ Mse loss and Mae

จากรูปที่ 3.19 กราฟนี้แสดงผลการเรียนรู้ของโมเดลในแง่ของ Loss และ Mean Absolute Error (MAE) บนชุดข้อมูลฝึก (Training) และชุดข้อมูลตรวจสอบ (Validation) ตามจำนวน Epochs กราฟซ้าย: แสดงค่า Loss ลดลงตาม Epochs ซึ่งหมายความว่าโมเดลกำลังเรียนรู้ได้ดี โดยค่า Validation Loss และ Training Loss มีแนวโน้มลดลงคล้ายกัน กราฟขวา: แสดงค่า MAE ซึ่งเป็นอีกตัวชี้วัดความผิดพลาดของโมเดล มีแนวโน้มลดลงในลักษณะเดียวกับ Loss ค่า Training Loss และ MAE สูงในช่วงแรก แต่ลดลงอย่างรวดเร็วในช่วงต้น ก่อนจะลดลงช้าลงเมื่อ Epochs เพิ่มขึ้น ค่า Validation Loss และ Validation MAE อยู่ต่ำกว่า Training เล็กน้อย อาจบ่งชี้ว่าโมเดลไม่ได้ Overfitting มากนัก ค่า Loss และ MAE เริ่มทรงตัวที่ค่าต่ำสุดช่วงปลายของการฝึก ซึ่งหมายถึงโมเดลอาจถึงจุดที่เรียนรู้ได้ดีที่สุดแล้ว

3.4 การออกแบบฮาร์ดแวร์

โมเดลต้นแบบรถจำลองจะมีความสูงอยู่ที่ 148.4 มิลลิเมตร และความกว้างของโมเดลต้นแบบรถจำลองจะมีความกว้างอยู่ที่ 157.99 มิลลิเมตร



รูปที่ 3.20 ขนาดความกว้างและความสูงของโมเดลต้นแบบรถจำลอง

3.4.1 ส่วนประกอบของโมเดลรถจำลอง

3.4.1.1 Raspberry Pi5 + AI Hat 13 Tops

3.4.1.2 Motor Fix Frame

3.4.1.3 Car expansion board

3.4.1.4 Platform PCB

3.4.1.5 Camera Fixed PCB

3.4.1.6 PTZ Component

3.4.1.7 Tire

3.4.1.8 Motor

3.4.1.9 Battery and Velco

3.4.1.10 Pin Cable 40

3.4.1.11 Camera and Cable

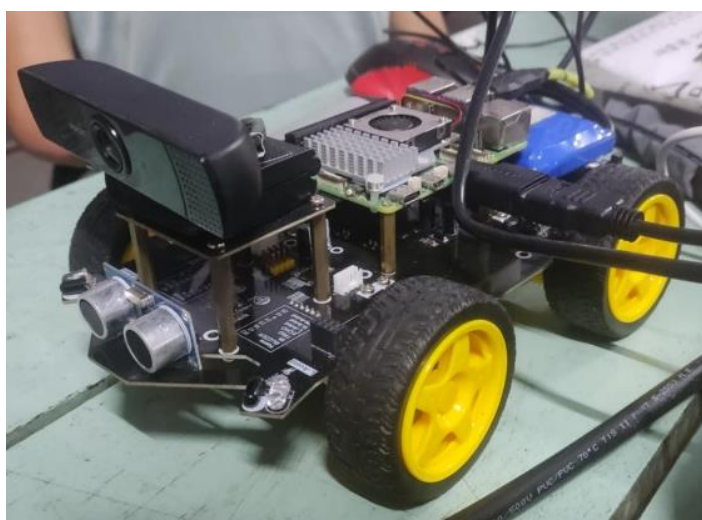
3.4.1.12 Ultrasonic Sensor



รูปที่ 3.21 ส่วนประกอบต่างๆเมื่อนำมาประกอบเข้าด้วยกัน

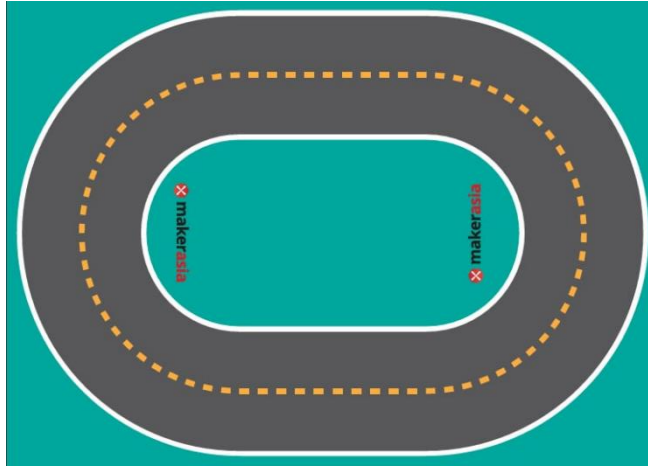
3.4.2 ดัดแปลงโมเดลต้นแบบรถจำลอง

โดยทางทีมผู้พัฒนาจะได้ทำการดัดแปลงโมเดลต้นแบบรถจำลองโดยที่นำตัว Camera Fixed PCB และตัว PTZ Component ออกไปและติดตั้งกล้อง Webcam เข้าไปที่ตัว Platform PCB แทนที่กล้องตัวเดิมเพื่อให้สามารถรับภาพที่คุณภาพดีกว่าเดิมและนำไปใช้ในการเทรนเพื่อการ Detection



รูปที่ 3.22 ดัดแปลงโมเดลต้นแบบ

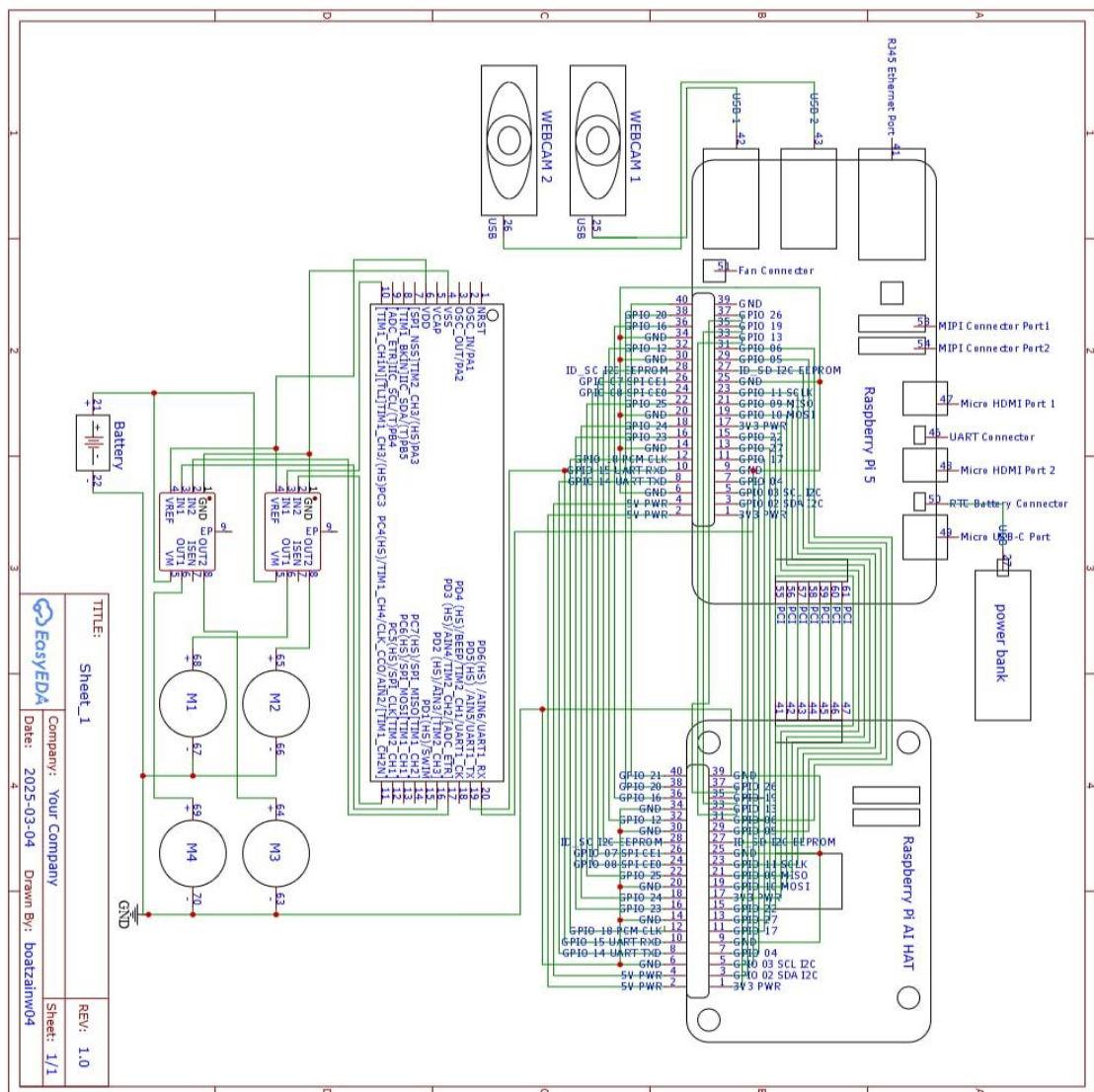
3.4.3 สร้างถนนจำลองสำหรับทดลองรถขับอัตโนมัติ



รูปที่ 3.23 ออกแบบถนนจำลองสำหรับทดลองรถขับอัตโนมัติ

จากรูปที่ 3.23 ได้ทำการออกแบบถนนจำลองให้มีลักษณะเป็นรูปตัว วงรี มีจุดโค้ง
ทั้งหมด 2 จุด ขนาด 300*420 cm

3.4.4 Hardware Diagram



รูปที่ 3.24 Hardware Diagram

จากรูปที่ 3.24 วงจรนี้ออกแบบโดย Raspberry Pi 5 โดยมีการเชื่อมต่อกับ Raspberry Pi AI HAT ผ่าน GPIO และกล้อง Webcam 1 และ Webcam 2 ที่เชื่อมต่อผ่านพอร์ต USB มอเตอร์ M1, M2, M3, M4 ถูกควบคุมผ่าน Motor Driver และใช้แหล่งจ่ายไฟจาก Battery และ Power Bank นอกจากนี้ GPIO ของ Raspberry Pi ยังถูกใช้ในการควบคุมและเชื่อมต่อกับอุปกรณ์อื่น ๆ เพื่อให้ทำงานได้อย่างสมบูรณ์

บทที่ 4

ผลการทดลอง

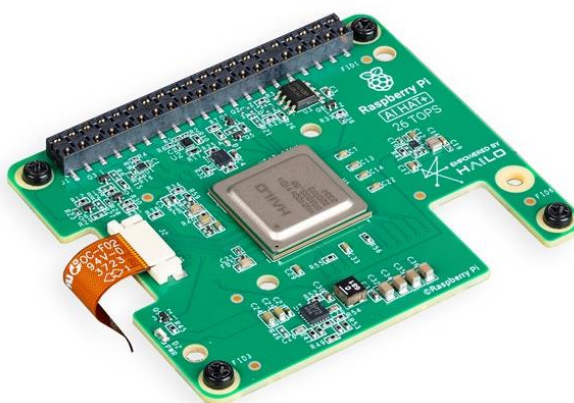
บทนี้เป็นระบบตรวจจับและการขับเคลื่อนของรถจำลองอัตโนมัติถนนจำลองพร้อมหลบหลีกและหยุดรถต้องอาศัยการทดสอบเกี่ยวกับวิธีการใช้งานของเครื่องและข้อจำกัดต่างๆ เพื่อเป็นข้อมูลเบื้องต้นสำหรับผู้สนใจจะนำไปศึกษาเพื่อสร้างใช้งานและปรับปรุงในเชิงอุตสาหกรรมหรือนำไปพัฒนาให้มีคุณสมบัติให้ดียิ่งขึ้นต่อไป งานวิจัยนี้แบ่งออกเป็น 2 ส่วนได้แก่ การทดลองการทำงานส่วนซอฟต์แวร์และส่วนฮาร์ดแวร์ ทำการทดสอบตามขอบเขตที่ได้ตั้งไว้ในโครงงาน

หลังจากที่ได้ดำเนินงานในขั้นตอนการออกแบบชิ้นงานและดำเนินการจัดทำชิ้นงานทำให้ได้รู้ว่า Raspberry Pi 5 ไม่สามารถที่จะใช้งานโมเดล MobileNet V2 และ YOLO V5 ได้พร้อมกันทางคณะผู้จัดทำจึงได้มาใช้ Raspberry Pi AI HAT ที่มีประสิทธิภาพในประมวลผล AI เพื่อให้รองรับการใช้งานโปรแกรมในการจับวัตถุ

4.1. อุปกรณ์ที่ใช้ในการทดสอบ



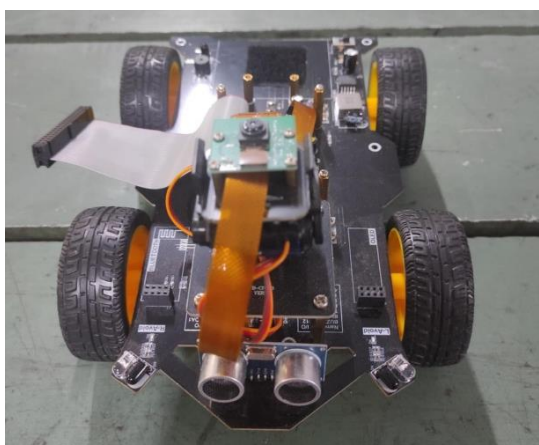
รูปที่ 4.1 Raspberry Pi 5



รูปที่ 4.2 Raspberry Pi AI HAT



รูปที่ 4.3 Webcam



รูปที่ 4.4 รถจำลองอัตโนมัติ

4.2 ทดสอบการทำงานวิ่งบนถนนจำลอง

เริ่มจากทดสอบจากการทำงานตามขอบเขตที่กำหนดเอาไว้โดยจะเริ่มทดสอบ ผลลัพธ์ให้เป็นไปตามขอบเขตลำดับต่อไปนี้

4.2.1 การทำงานของอุปกรณ์ตรวจจับถนนจำลอง

- ทางตรง



รูปที่ 4.5 ทางตรงของถนนจำลอง ก



รูปที่ 4.6 ทางตรงของถนนจำลอง ข

จากรูปที่ 4.5 ถึง 4.6 การทดลอง Mobilenet V2 ที่เกี่ยวกับวิเคราะห์ทางตรงของถนนจำลองซึ่งเป็นการแสดงการทำงานของระบบตรวจจับเมื่อมีกรวยและเส้นตัวรถจำลองอัตโนมัติจะเคลื่อนที่ไปตามทางตรงไปจนถึงทางโค้งโดยความเร็วของมอเตอร์จะเท่ากันทั้งซ้ายและขวา

- การเลี้ยวขวา



รูปที่ 4.7 การเลี้ยวขวาของถนนจำลอง ก

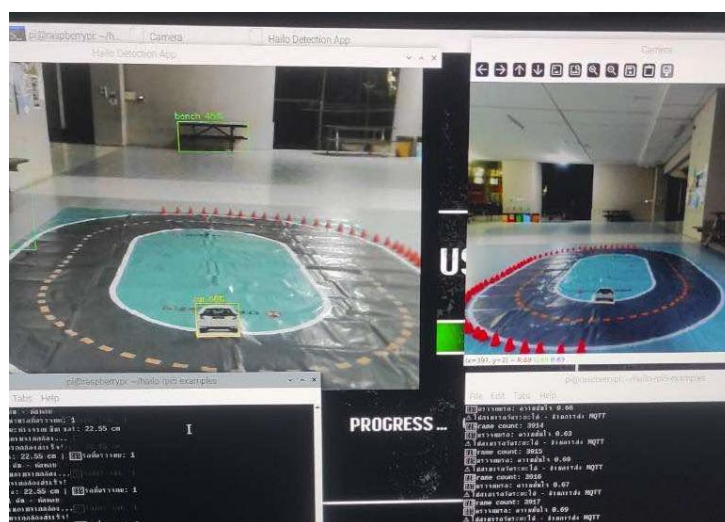


รูปที่ 4.8 การเลี้ยวขวาของถนนจำลอง ข

จากรูปที่ 4.7 ถึง 4.8 การทดลองที่เกี่ยวกับ Mobilenet V2 ที่เกี่ยวกับวิเคราะห์ ทางโค้งของถนนจำลองซึ่งเป็นการแสดงการทำงานของระบบตรวจจับเมื่อมีกรวยและเส้นตัวรถจำลองอัตโนมัติจะเคลื่อนที่ไปตามทางโค้งไปจนถึงทางตรงอีกรอบโดยการเลี้ยวจะใช้ความเร็วรอบของมอเตอร์ฝั่งซ้ายที่เร็วกว่าฝั่งขวาเพื่อให้ตัวรถจำลองอัตโนมัติไปทางขวา

4.3 การทำงานของอุปกรณ์ตรวจจับโมเดลรถ

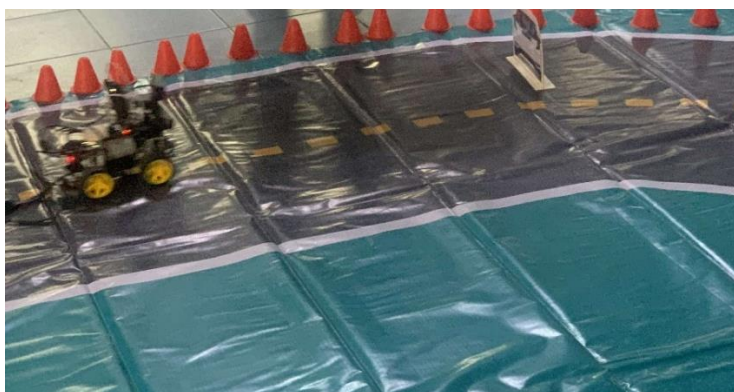
เป็นการทดลองร่วมกับอุปกรณ์ตรวจจับโมเดลรถเมื่อมีโมเดลรถอยู่ด้านหน้ารถจำลองอัตโนมัติตรวจจับโดยใช้ YOLOv8s ได้จะ แสดงข้อมูลต่างๆตามขอบเขตดังต่อไปนี้



รูปที่ 4.9 การทำงานของอุปกรณ์ตรวจจับโมเดลรถ

จากรูปที่ 4.9 การทำงานของอุปกรณ์ที่ใช้ YOLO V8s ตรวจจับโมเดลรถพร้อมกับการที่รัน MobilenetV2

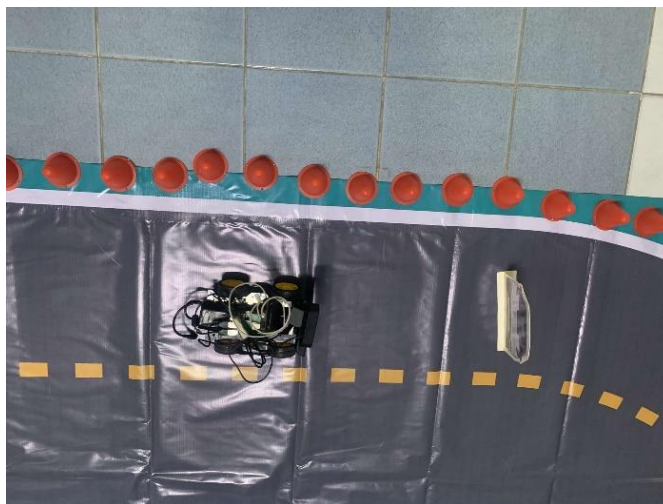
4.3.1 เจอโมเดลรถในระยะ 100 cm



รูปที่ 4.10 รถจำลองอัตโนมัติเจอโมเดลรถในระยะ 100 cm

จากรูปที่ 4.10 เมื่อ Webcam ตรวจจับเจอโมเดลในระยะ 100 cm ตัวรถจำลองอัตโนมัติ ความเร็วจะค่อยๆ ลดลง

4.3.2 เจอโมเดลรถในระยะ 50 cm



รูปที่ 4.11 รถจำลองอัตโนมัติเจอโมเดลรถในระยะ 50 cm

จากรูปที่ 4.11 เมื่อตรวจจับเจอโมเดลในระยะ 50 cm ตัวรถจำลองอัตโนมัติจะเริ่มชะลอ

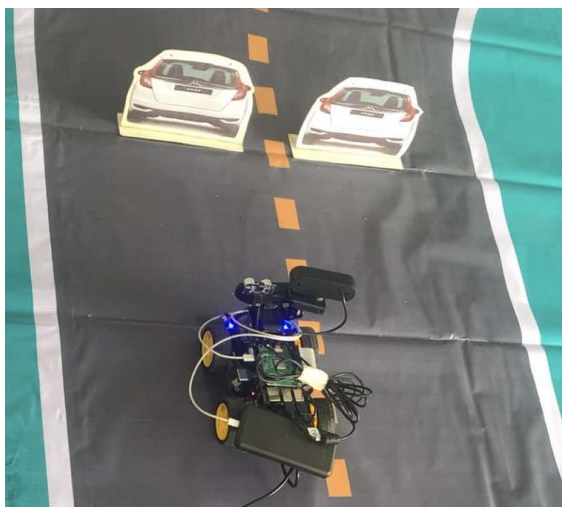
4.3.3 เจอโมเดลรถในระยะ 25 cm และ หยุดรถและหลบหลีก



รูปที่ 4.12 รถจำลองอัตโนมัติเจอโมเดลรถในระยะ 25 cm

จากรูปที่ 4.12 เมื่อ Webcam ตรวจจับเจอโมเดลในระยะ 25 cm ตัวรถจำลองอัตโนมัติ จะหยุด และหลบหลีกจากจะแซงขึ้นไป

4.3.4 เจอโมเดลรถ 2 คันในระยะ 50 cm และ หยุดรถ



รูปที่ 4.13 รถจำลองอัตโนมัติเจอโมเดลรถในระยะ 50 cm

จากรูปที่ 4.13 เมื่อ Webcam ตรวจจับเจอโมเดลในระยะ 50 cm ตัวรถจำลองอัตโนมัติจะหยุด และจะรอให้โมเดลรถเหลือ 1 คันเพื่อให้หลบหลีกจากจะแซงขึ้นไป

ตารางที่ 4.1 การทดสอบประสิทธิภาพโมเดล ในการตรวจจับและวิ่งอัตโนมัติทั้งหมด

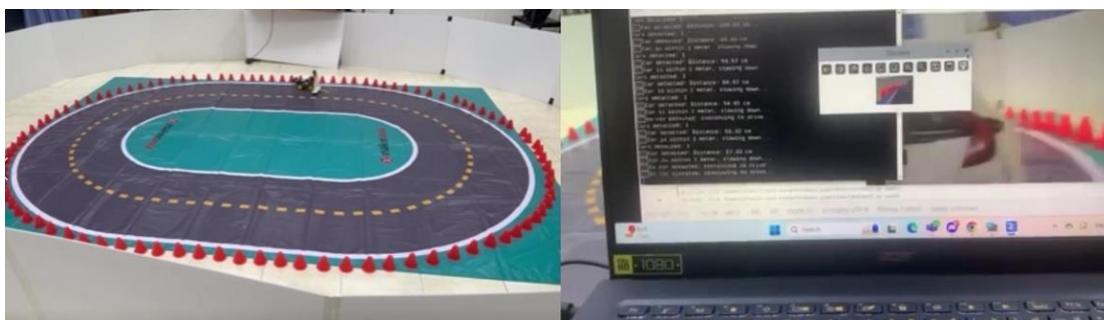
โมเดล	รอบที่	โมเดลรถ 1 คัน	โมเดลรถ 1 คันขยับ	โมเดลรถ 2 คัน
MobileNet V2 + YOLO V8S	1	ชน	ชน	ชน
	2	แซงได้	แซงได้	ชน
	3	ชน	แซงได้	แซงได้
	4	แซงได้	แซงได้	แซงได้
	5	ชน	แซงได้	แซงได้
	6	แซงได้	ชน	แซงได้
	7	แซงได้	ชน	แซงได้
	8	แซงได้	แซงได้	แซงได้
	9	แซงได้	แซงได้	แซงได้
	10	แซงได้	แซงได้	แซงได้

ตารางที่ 4.2 การทดสอบประสิทธิภาพโมเดล ในการตรวจจับและวิ่งอัตโนมัติรอบที่ 1

โมเดล	รอบที่	โมเดลรถ 1 คัน	หมายเหตุ
MobileNet V2 + YOLO V8S	1	ชน	เลี้ยวมาไม่ตรงเลนและอัลตราโซนิกไม่โดนวัตถุ
	2	แข่งได้	แข่งได้
	3	ชน	เลี้ยวมาไม่ตรงเลนและอัลตราโซนิกไม่โดนวัตถุ
	4	แข่งได้	แข่งได้
	5	ชน	เลี้ยวมาไม่ตรงเลนและอัลตราโซนิกไม่โดนวัตถุ
	6	แข่งได้	แข่งได้
	7	แข่งได้	แข่งได้
	8	แข่งได้	แข่งได้
	9	แข่งได้	แข่งได้
	10	แข่งได้	แข่งได้

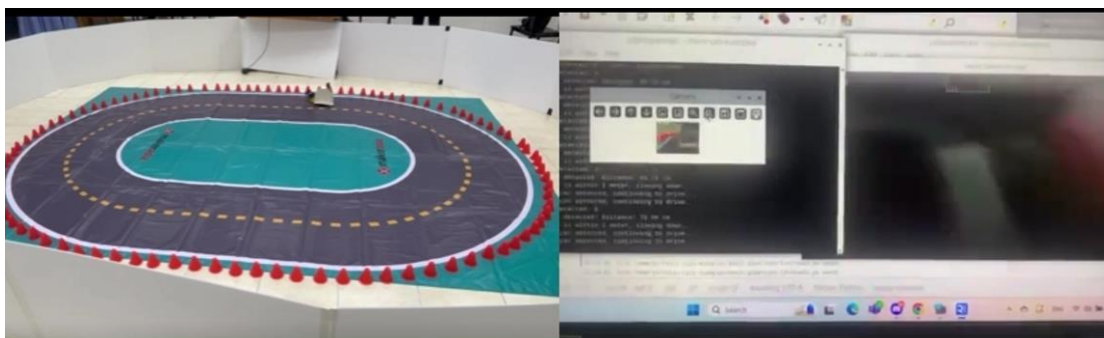
จากตารางที่ 4.2 โมเดล **MobileNet V2 + YOLO V8S** ทดสอบ 10 รอบหลบได้ 7 ครั้งและพบว่าการชน 3 ครั้ง เกิดข้อผิดพลาดเรื่องการเลี้ยวกลับเข้าเลนและอัลตราโซนิก

4.3.5 วิเคราะห์ความผิดพลาดของตารางที่ 4.2



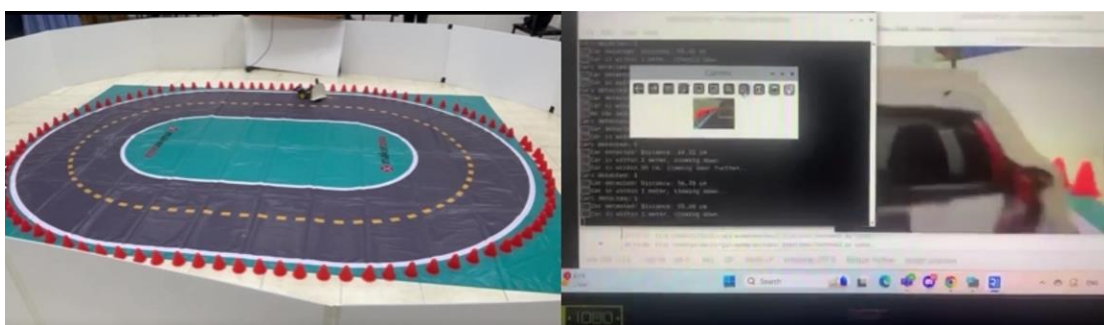
รูปที่ 4.14 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 1

จากรูปที่ 4.14 การวิ่งอัตโนมัติตรวจจับโมเดลรถในรอบที่ 1 นั้นที่ชน เกิดจากการที่รถวิ่งมาไม่ตรงเลนและออกซ้ายมากเกินไปทำให้ออกจากระยะของกล้องและอัลตราโซนิกในการไม่เจอวัตถุในตรวจจับ



รูปที่ 4.15 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 3

จากรูปที่ 4.15 การวิ่งอัตโนมัติตรวจจับโมเดลรถในรอบที่ 1 นั้นที่ชน เกิดจากการที่รถวิ่งมาไม่ตรงเลนและออกซ้ายมากเกินไปทำให้ออกจากกระยะของกล้องและอัลตราโซนิกในการไม่เจอวัตถุในตรวจจับ



รูปที่ 4.16 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 5

จากรูปที่ 4.16 การวิ่งอัตโนมัติตรวจจับโมเดลรถในรอบที่ 1 นั้นที่ชน เกิดจากการที่รถวิ่งมาไม่ตรงเลนและออกซ้ายมากเกินไปทำให้ออกจากกระยะของกล้องและอัลตราโซนิกในการไม่เจอวัตถุในตรวจจับ

ตารางที่ 4.3 การทดสอบประสิทธิภาพโมเดล ในการตรวจจับและวิ่งอัตโนมัติรอบที่ 2

โมเดล	รอบที่	โมเดลรถ 1 คั่นขยับ	หมายเหตุ
MobileNet V2 + YOLO V8S	1	ชน	เลี้ยวมาไม่ตรงเลนและอัลตราโซนิกไม่โดนวัตถุ
	2	แข่งได้	แข่งได้
	3	แข่งได้	แข่งได้
	4	แข่งได้	แข่งได้
	5	แข่งได้	แข่งได้
	6	ชน	เลี้ยวมาไม่ตรงเลน
	7	ชน	เลี้ยวมาไม่ตรงเลน
	8	แข่งได้	แข่งได้
	9	แข่งได้	แข่งได้
	10	แข่งได้	แข่งได้

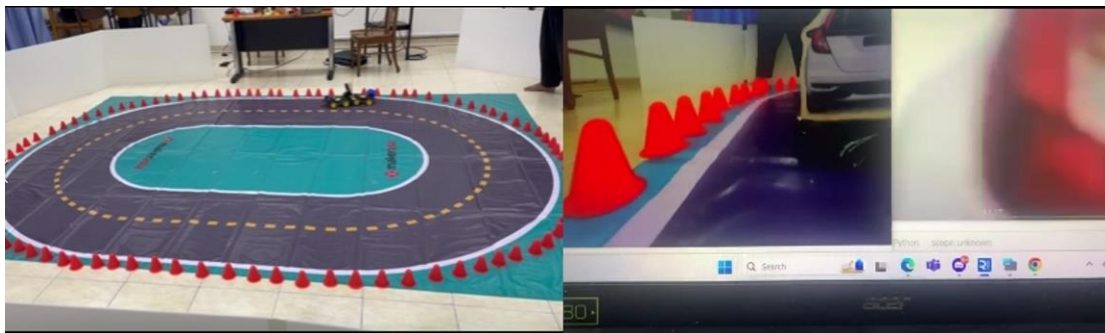
จากตารางที่ 4.3 โมเดล MobileNet V2 + YOLO V8S ทดสอบ 10 รอบหลบได้ 7 ครั้ง และว่าพบมีการชน 3 ครั้ง เกิดข้อผิดพลาดเรื่องการเลี้ยวกลับเข้าเลนและอัลตราโซนิก

4.3.6 วิเคราะห์ความผิดพลาดของตารางที่ 4.3



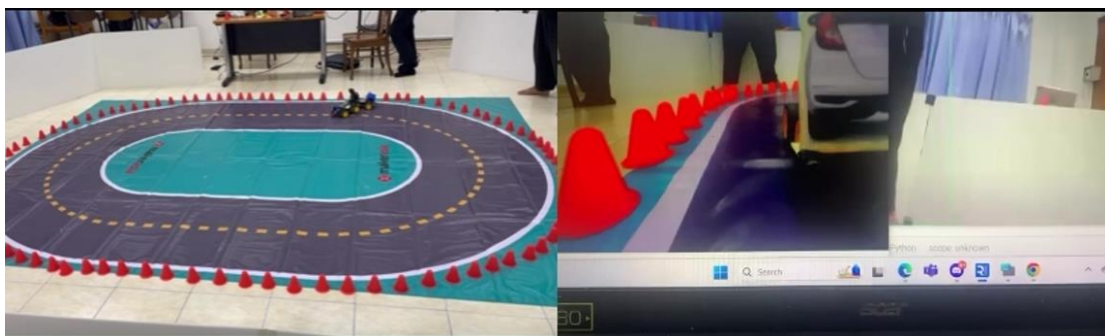
รูปที่ 4.17 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 1

จากรูปที่ 4.17 การวิ่งอัตโนมัติตรวจจับโมเดลรถในรอบที่ 1 นั้นที่ชน เกิดจากการที่รถวิ่งมาไม่ตรงเลนและออกซ้ายมากเกินไปทำให้ออกจากกระยะของกล้องและอัลตราโซนิกในการไม่เจอวัตถุในตรวจจับ



รูปที่ 4.18 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 6

จากรูปที่ 4.18 การวิ่งอัตโนมัติตรวจจับโมเดลรถในรอบที่ 6 นั้นที่ชน เกิดจากการที่รถวิ่งมาไม่ตรงเลนและเมื่อเจอโมเดลรถและอัลตราโซนิกในการตรวจจับแล้วแต่ตอนทำการเลี้ยวออกไม่ได้ องศาในการที่จะเร่งตรงเพื่อที่จะแข่งทำให้ชน



รูปที่ 4.19 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 7

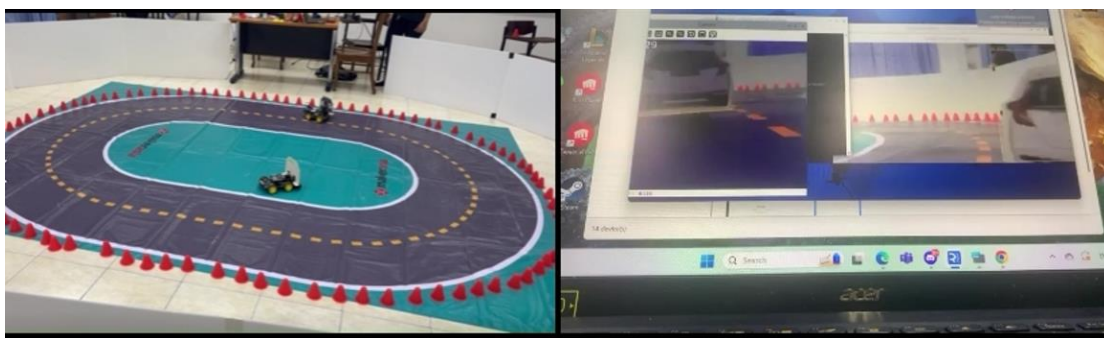
จากรูปที่ 4.19 การวิ่งอัตโนมัติตรวจจับโมเดลรถในรอบที่ 7 นั้นที่ชน เกิดจากการที่รถวิ่งมาไม่ตรงเลนและเมื่อเจอโมเดลรถและอัลตราโซนิกในการตรวจจับแล้วแต่ตอนทำการเลี้ยวออกไม่ได้ องศาในการที่จะเร่งตรงเพื่อที่จะแข่งทำให้ชน

ตารางที่ 4.4 การทดสอบประสิทธิภาพโมเดล ในการตรวจจับและวิ่งอัตโนมัติรอบที่ 3

โมเดล	รอบที่	โมเดลรถ 2 คัน	หมายเหตุ
MobileNet V2 + YOLO V8S	1	ชน	เลี้ยวมาไม่ตรงเลนและอัลตราโซนิกไม่โดนวัตถุ
	2	ชน	เลี้ยวมาไม่ตรงเลนและอัลตราโซนิกไม่โดนวัตถุ
	3	แข่งได้	แข่งได้
	4	แข่งได้	แข่งได้
	5	แข่งได้	แข่งได้
	6	แข่งได้	แข่งได้
	7	แข่งได้	แข่งได้
	8	แข่งได้	แข่งได้
	9	แข่งได้	แข่งได้
	10	แข่งได้	แข่งได้

จากตารางที่ 4.4 โมเดล MobileNet V2 + YOLO V8S ทดสอบกับรถ 2 คัน 10 รอบ
หลบได้ 8 ครั้ง และพบว่าชน 2 ครั้ง เกิดข้อผิดพลาดเรื่องการเลี้ยวกลับเข้าเลนและอัลตราโซนิก

4.3.7 วิเคราะห์ความผิดพลาดของตารางที่ 4.4



รูปที่ 4.20 ข้อผิดพลาดการทดสอบการตรวจจับและวิ่งอัตโนมัติในรอบที่ 1

จากรูปที่ 4.20 การวิ่งอัตโนมัติตรวจจับโมเดลรถในรอบที่ 1 นั้นที่ชน เกิดจากการที่รถ
วิ่งมาไม่ตรงเลนและออกซ้ายมากเกินไปทำให้ออกจากระยะของกล้องและอัลตราโซนิกในการไม่เจอ
วัตถุในตรวจจับ



รูปที่ 4.21 ข้อผิดพลาดการทดสอบการตรวจจับและวิ้งอัตโนมัติในรอบที่ 2

จากรูปที่ 4.21 การวิ้งอัตโนมัติตรวจจับโมเดลรถในรอบที่ 2 นั้นที่ชน เกิดจากการที่รถวิ้งมาไม่ตรงเลนและออกซ้ายมากเกินไปทำให้ออกจากระยะของกล้องและอัลตราโซนิกในการไม่เจอวัตถุในตรวจจับ

สรุปผลการทดลอง

โมเดล MobileNet V2 + YOLO V8S ถูกทดสอบในสองเงื่อนไข คือ รถ 1 คัน, รถ 1 คันขยับ และ รถ 2 คัน โดยทดสอบทั้งหมดเงื่อนไขละ 10 รอบ สำหรับรถ 1 คัน พบว่าชน 3 ครั้ง โดยสาเหตุหลักมาจากการเบี่ยงเลนไม่ตรงและอัลตราโซนิกตรวจจับไม่เจอวัตถุ ส่วนที่เหลืออีก 7 ครั้งสามารถหลบได้สำเร็จ สำหรับรถ 1 คันขยับ พบว่าชน 3 ครั้ง โดยสาเหตุหลักมาจากการเบี่ยงเลนไม่ตรงและอัลตราโซนิกตรวจจับไม่เจอวัตถุ 1 ครั้ง และอีก 2 ครั้งองศาหักหลบไม่พ้น ส่วนที่เหลืออีก 7 ครั้งสามารถหลบได้สำเร็จ สำหรับรถ 2 คัน พบว่าชน 2 ครั้ง โดยสาเหตุหลักมาจากการเบี่ยงเลนไม่ตรงและอัลตราโซนิกตรวจจับไม่เจอวัตถุ 1 ครั้ง และหักหลบใกล้เคียงเกินไป และสามารถหลบได้ 8 ครั้ง แสดงให้เห็นว่าโมเดลมีความสามารถในการหลบหลีกยังมีข้อผิดพลาดในบางกรณีที่ต้องปรับปรุง

ตารางที่ 4.5 สรุปความถูกต้องและประสิทธิภาพของโมเดล

การตรวจจับ	ความถูกต้อง
โมเดลรถ 1 คัน	70%
โมเดลรถ 1 คันขยับ	70%
โมเดลรถ 2 คัน	80%
ค่าเฉลี่ย	75%

จากตารางที่ 4.5 พบว่าโมเดล MobileNetV2 + YOLO V8s ถูกทดสอบกับจำนวนโมเดลรถที่แตกต่างกัน (1 คัน, 1 คันขยับ, และ 2 คัน) โดยการตรวจจับทั้งหมด 10 รอบ ผลลัพธ์แสดงให้เห็นว่าโมเดลที่มีรถ 2 คันมีความถูกต้องสูงสุดที่ 80% ในขณะที่โมเดลที่มีรถ 1 คัน และ 1 คันขยับมีความถูกต้องเท่ากันที่ 70% ส่งผลให้ค่าเฉลี่ยของความถูกต้องอยู่ที่ 75%

บทที่ 5

สรุปและข้อเสนอแนะ

ในส่วนนี้จะเป็นการสรุปผลของโครงการรถจำลองอัตโนมัติสำหรับถนนจำลองพร้อมหลบหลีกและหยุดรถ ได้ทดสอบและนำไปสู่การทดลองของโครงการวิจัย และได้บันทึกผลการทดลองแล้วนำมา สรุปผลลัพธ์ดังต่อไปนี้

5.1 สรุปผลตามของเขต

5.1.1 รถจำลองที่สามารถขับเคลื่อนบนถนนจำลองได้อย่างอัตโนมัติโดยใช้ AI ได้

5.1.1.1 สามารถหลบหลีก แชน โมเดลรถที่อยู่ด้านหน้าหากมี 1 คัน

5.1.1.2 สามารถหยุดหรือชะลอ เมื่อมีโมเดลรถอยู่ด้านหน้าข้างกัน 2 คัน

5.1.2 ติดตั้งกล้องเว็บแคมสองตัวและเซ็นเซอร์อัลตราโซนิก

5.1.2.1 เว็บแคมตัวที่ 1 สามารถตรวจจับเส้นทาง

5.1.2.2 เว็บแคมตัวที่ 2 สามารถตรวจจับรถยนต์ด้านหน้า

5.1.2.3 เซ็นเซอร์อัลตราโซนิกสามารถวัดระยะห่างและตรวจจับวัตถุด้านหน้าเมื่อรถจำลองระยะ 100 cm และจะเริ่มชะลอในระยะ 50 cm และจะหยุดรถในระยะ 25 cm

5.1.3 Raspberry Pi เป็นตัวควบคุมหลักสำหรับระบบทั้งหมด

5.1.3.1 สามารถควบคุมเว็บแคมทั้ง 2 อัน

5.1.3.2 สามารถควบคุมเซ็นเซอร์อัลตราโซนิก

5.1.3.3 สามารถควบคุมมอเตอร์ทั้ง 4 อัน

5.1.3.4 สามารถใช้โมเดล YOLO V8s และ ใช้โมเดล MobileNet V2 ในการตรวจจับ

5.1.4 สามารถใช้โมเดล YOLO V8s และ ใช้โมเดล MobileNet V2 ได้พร้อมกัน

5.1.5 YOLO V8 สามารถใช้ในการตรวจจับรถยนต์ที่อยู่ด้านหน้า

5.1.6 MobileNet V2 สามารถใช้ในการตรวจจับและวิเคราะห์รูปแบบถนน และให้รถจำลองทำการโค้งตามรูปแบบของถนน

5.2 ปัญหาและอุปสรรค

5.2.1 ปัญหาที่เกี่ยวข้องกับ Raspberry Pi 5 ได้แก่ข้อจำกัดด้านฮาร์ดแวร์ที่มีประสิทธิภาพต่ำ และปัญหาความร้อนสะสมที่เกิดขึ้นระหว่างการทำงาน ซึ่งส่งผลให้ประสิทธิภาพของโปรแกรมสำหรับการตรวจจับวัตถุลดลง

วิธีแก้ปัญหา ใช้ฮาร์ดแวร์ที่มีชื่อว่า Raspberry Pi AI HAT ที่มีประสิทธิภาพในประมวลผล AI เพื่อให้รองรับการใช้งานโปรแกรมในการจับวัตถุ

5.2.2 ปัญหาที่เกี่ยวกับภาพกระตุกและอัตราเฟรมเรทลดลง

วิธีแก้ปัญหา ลดประสิทธิภาพของภาพและโมดิฟายเเยอร์ในการปรับจูนโมเดล

5.2.3 ปัญหาที่เกี่ยวกับการเลี้ยวของรถจำลองอัตโนมัติ

วิธีแก้ปัญหา โมดิฟายเเยอร์ในการปรับจูนโมเดล และนำกรวยมาช่วยในการเลี้ยว

5.2.4 ปัญหาที่เกี่ยวกับไฟในการเลี้ยงอุปกรณ์ไม่พอ

วิธีแก้ปัญหา ใช้ 2 แหล่งจ่าย แหล่งจ่ายที่ 1 จ่ายให้ PCB Car แหล่งจ่ายที่ 2 จ่ายให้

Raspberry Pi

5.3 ข้อเสนอแนะ

5.3.1 ใช้งาน Raspberry Pi 5 ที่มีประสิทธิภาพสูงขึ้นและมีระบบระบายความร้อนออกจากฮาร์ดแวร์เพื่อให้มีการใช้งานที่ดีขึ้นและยาวนานขึ้น

5.3.2 เพิ่มประสิทธิภาพของโมเดลในการตรวจจับวัตถุให้มีความแม่นยำมากขึ้น

5.3.3 กล้องที่เชื่อมต่อฮาร์ดแวร์โดยตรงเพื่อลดการตีเลย์ของภาพ

5.4 สรุปผลการทดลอง

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนาระบบรถจำลองอัตโนมัติ หลบหลีกและหยุดรถ เพื่อช่วยให้ผู้ใช้งานบนถนนได้รับความปลอดภัย โดยระบบใช้เทคโนโลยีปัญญาประดิษฐ์ (AI) และ Deep Learning ผ่านสถาปัตยกรรม Convolutional Neural Network (CNN) ทำการฝึกโมเดลด้วย MobileNet V2 จะเป็นการตรวจจับรูปแบบของถนนจำลอง และ YOLO V8s เป็นการตรวจจับรถโมเดล

บรรณานุกรม

- [1] ณัฐภูมิษฐ์ เขมะทาสี, และ ยุทธนา พิมเสน. (2561). **รถจำลองขับเคลื่อนอัตโนมัติ สาขาวิชาวิศวกรรมคอมพิวเตอร์, มหาวิทยาลัยสยาม.**
 สืบค้นจาก : <https://eresearch.slam.edu/wpcontent/uploads/2020/06/engIneerIng-computer-engineerIng-2018-project-Automatically-Car.pdf>
- [2] นายรัชชัย สุรินแก้ว, นายนนทชัย กากายา, และ นายรัชตพล บุตรศรีชา. (2566). **รถจำลองขับเคลื่อนอัตโนมัติตามแผนที่ปฏิบัติตามสัญญาณไฟจราจรได้, มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา เชียงใหม่, คณะวิศวกรรมศาสตร์, สาขาวิชาวิศวกรรมคอมพิวเตอร์.**
 สืบค้นจาก : https://drive.google.com/file/d/1s__AC8FFejzrS_TGw2LnV0bSRm-Hp/vlew
- [3] พุฒิพงศ์ ขุนทรง, ประณัฐ วิสุวรรณ, และ วิฑิต ฉัตรรัตน์กุลชัย. (2554) **การหลบหลีกสิ่งกีดขวางของรถอัจฉริยะบนพื้นราบ, มหาวิทยาลัยเกษตรศาสตร์/กรุงเทพฯ.**
 สืบค้นจาก : <https://dric.nrct.go.th/index.php?/Search/SearchDetail/263064>
- [4] ตรีรัตน์ เมตต์การุณจิต (2551) **การสร้างและทดสอบรถอัจฉริยะไร้คนขับ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น, สถาบันเทคโนโลยีไทย-ญี่ปุ่น.**
 สืบค้นจาก : <http://library.tni.ac.th/thesis/upload/files/RES%20260663/Tiratana%20Metkarunchit%20Res%20IT%202008.pdf>
- [5] Sarda, A., Dixit, S., & Bhan, A. (2021). **Object Detection for Autonomous Driving using YOLO [You Only Look Once] algorithm, Intelligent Communication Technologies and Virtual Mobile Networks**
 doi :10.1109/ICICV50876.2021.9388577
 Retrieved from : <https://www.semanticscholar.org/paper/Object-Detection-for-Autonomous-Driving-using-YOLO-Sarda-Dixit/b72f75a1ba8cac95e0000f97f998daec8a17c9d7>

- [6] Pannu, G. S., Dawud, M., & Gupta, P. (2015). **Design and Implementation of Autonomous Car using Raspberry Pi**. *International Journal of Computer Applications*, International Journal of Computer Applications 113(9). Retrieved from : https://www.researchgate.net/publication/276027654_Design_and_Implementation_of_Autonomous_Car_using_Raspberry_Pi
- [7] Khan, S. A., Lee, H. J., & Lim. (2023). **Comparison of the Efficacy and Safety of Left Atrial Appendage Closure and Direct Oral Anticoagulants for Atrial Fibrillation: A Meta-Analysis of Randomized Control Trials and Observational Studies**, Wei et al, *Cureus* 15(12), DOI :10.7759/cureus.49827. Retrieved from : <https://www.semanticscholar.org/paper/Comparison-of-the-Efficacy-and-Safety-of-Left-and-A-Wei-Lim/ae4fa47e0d72627053e18054dd8403b0deaea6d8>
- [8] Hu, M., Guo, H., & Ji, X. (2018). **Automatic Driving of End-to-end Convolutional Neural Network Based on MobileNet-V2 Migration Learning** school of mechanical power East China university of science and technology China. doi :10.1145/3356422.3356458. Retrieved from : https://www.researchgate.net/publication/337362394_Automatic_Driving_of_End-to-end_Convolutional_Neural_Network_Based_on_MobileNet-V2_Migration_Learning

- [9] Shirolkar, R., Datar, R., Dhongade, A., & Behere, G. (2019). **Self-Driving Autonomous Car using Raspberry Pi**, *International Journal of Engineering Research & Technology (IJERT)* International Journal of Engineering Research & Technology, 8(5). doi :10.17577/IJERTV8IS050100.
Retrieved from : <https://www.ijert.org/self-driving-autonomous-car-using-raspberry-pi>
- [10] Shirke, Y. S., Gawade, U. S., Jadhav, K. K., & Hawre, M. A. (2020). **Implementation of SDC: Self-Driving Car based on Raspberry Pi**, *INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY*, 7(3).
Retrieved from : <https://www.semanticscholar.org/paper/Implementation-of-SDC-A-Self-Driving-Car-based-on-Pi-Shirke-Gawade/29ec00ed7446c228515b705e707ab22e9c6b8492>
- [11] Suo, Y., Chen, S., & Zheng, M. (2020). **Developing an Autonomous Driving Model Based on Raspberry Pi**, Department of Mathematics and Statistics University of Wisconsin-La Crosse.
Retrieved from : <https://www.semanticscholar.org/paper/Developing-an-Autonomous-Driving-Model-Based-on-Pi-Suo-Chen/0c2be6d425acf99cda65dda6e55d424d9247e436>

ภาคผนวก ก

Source Code

hailo_model_zoo



https://github.com/hailo-ai/hailo_model_zoo/tree/master

Dataset



<https://github.com/ThayaPanyakerng/ThayaPanyakerng/tree/main/outleft>

Colab Notebooks สำหรับเทรนให้รถวิ่ง



<https://github.com/ThayaPanyakerng/ThayaPanyakerng/blob/main/train2.ipynb>

import.py สำหรับจูนให้รถวิ่งดีขึ้น



<https://github.com/ThayaPanyakerng/ThayaPanyakerng/blob/main/import.py>

ภาคผนวก ข

คู่มือการใช้งานรถจำลองอัตโนมัติตรวจจับโมเตลรถ และ วิเคราะห์รูปแบบถนน

คู่มือการใช้งานรถจำลองอัตโนมัติตรวจจับวิเคราะห์รูปแบบถนน

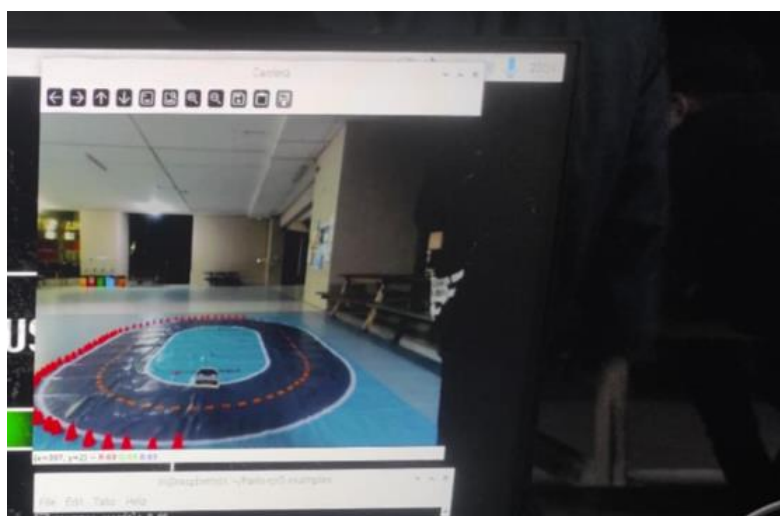
1. การติดตั้งฮาร์ดแวร์

นำอุปกรณ์ Raspberry Pi 5 ไปติดตั้งบนรถจำลองอัตโนมัติและติดตั้ง Raspberry Pi AI HAT บน Raspberry Pi 5 และ ติดตั้งกล้องที่ต้องการจะเชื่อมต่อเข้ากับตัว Raspberry Pi 5 จะสามารถเชื่อมต่อกับ Raspberry Pi 5 ได้อยู่ที่ 2 ตัว โดยกล้องที่เชื่อมต่อ Raspberry Pi 5 ตัวที่ 1 จะใช้เป็นกล้องที่วิเคราะห์รูปแบบถนน และตัวที่ 2 ใช้ตรวจจับโมเดลรถบนถนนและอัลตราโซนิกเพื่อวัดระยะห่างและตรวจจับวัตถุด้านหน้า



2. เปิดเครื่องและดูความถูกต้อง

ทำการเปิดเครื่องและตรวจสอบอุปกรณ์ว่าสามารถเชื่อมต่อกับกล้องได้หรือไม่



คู่มือการใช้งานรถจำลองอัตโนมัติตรวจจับโมเดลรถ

1. การติดตั้งฮาร์ดแวร์

นำอุปกรณ์ Raspberry Pi 5 ไปติดตั้งบนรถจำลองอัตโนมัติและติดตั้ง Raspberry Pi AI HAT บน Raspberry Pi 5 และ ติดตั้งกล่องที่ต้องการจะเชื่อมต่อเข้ากับตัว Raspberry Pi 5 จะสามารถเชื่อมต่อกับ Raspberry Pi 5 ได้อยู่ที่ 2 ตัว โดยกล่องที่เชื่อมต่อ Raspberry Pi 5 ตัวที่ 2 ใช้ตรวจจับโมเดลรถบนถนนและอัลตราโซนิกเพื่อวัดระยะห่างและตรวจจับวัตถุด้านหน้า



2. เปิดเครื่องและดูความถูกต้อง

ทำการเปิดเครื่องและตรวจสอบอุปกรณ์ว่าสามารถเชื่อมต่อกับกล่องได้หรือไม่



ประวัติผู้จัดทำโครงการ



ชื่อ - นามสกุล (ภาษาไทย)	นายวีรวุฒิ ลักขณาธร
ชื่อ - นามสกุล (ภาษาอังกฤษ)	Mr.Weerawut Lukkanatorn
รหัสนักศึกษา	65543206078-9
สถานที่ติดต่อ	98 ถ.ป่าขาม2 ต.พระบาท อ.เมือง จ.ลำปาง 52000
เบอร์โทรศัพท์	065-119-0012
E-mail	gearmcc2545@gmail.com

ประวัติการศึกษา

พ.ศ. 2563	ประกาศนียบัตรวิชาชีพ(ปวช.) สาขาอิเล็กทรอนิกส์ วิทยาลัยเทคนิคลำปาง
พ.ศ. 2565	ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.) สาขาอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา จ.เชียงใหม่
พ.ศ. 2569	คณะวิศวกรรมศาสตร์ หลักสูตรวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา จ.เชียงใหม่

ประวัติผู้จัดทำโครงการ



ชื่อ - นามสกุล (ภาษาไทย)	นายปฎิภาณ แสงกาศ
ชื่อ - นามสกุล (ภาษาอังกฤษ)	Mr.Patiparn sangkad
รหัสนักศึกษา	65543206068-0
สถานที่ติดต่อ	209 ม.5 ต.ท่าทุ่งหลวง อ.แม่ทา จ.ลำพูน 51170
เบอร์โทรศัพท์	065-9453828
E-mail	jino1234qaz@gmail.com

ประวัติการศึกษา

พ.ศ. 2563	ประกาศนียบัตรวิชาชีพ(ปวช.) สาขาอิเล็กทรอนิกส์ วิทยาลัยเทคนิคลำพูน
พ.ศ. 2565	ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.) สาขาอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา จ.เชียงใหม่
พ.ศ. 2569	คณะวิศวกรรมศาสตร์ หลักสูตรวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา จ.เชียงใหม่

ประวัติผู้จัดทำโครงการ



ชื่อ - นามสกุล (ภาษาไทย)	นายฐญา ปัญญาเค็ง
ชื่อ - นามสกุล (ภาษาอังกฤษ)	Mr.Thaya Panyakerng
รหัสนักศึกษา	65543206051-6
สถานที่ติดต่อ	33 หมู่ 6 ต.บ้านกลาง อ.เมือง จ.ลำพูน 51000
เบอร์โทรศัพท์	061-2712923
E-mail	boatzainw04@gmail.com
ประวัติการศึกษา	
พ.ศ. 2563	ประกาศนียบัตรวิชาชีพ(ปวช.) สาขาอิเล็กทรอนิกส์ วิทยาลัยเทคนิคลำพูน
พ.ศ. 2565	ประกาศนียบัตรวิชาชีพชั้นสูง(ปวส.) สาขาอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา จ.เชียงใหม่
พ.ศ. 2569	คณะวิศวกรรมศาสตร์ หลักสูตรวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา จ.เชียงใหม่