



ระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน

(Adaptive Cruise Control)

นายเฉลิมพล	ชมภูพู่	รหัส	58362124
นางสาวณัฐนรี	นรพงษ์	รหัส	58362247
นายณัฐปคัลภ์	ฟูพงษ์	รหัส	58362254

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2561



ใบรับรองโครงการ

หัวข้อโครงการ	ระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน		
	Adaptive Cruise Control		
ผู้ดำเนินโครงการ	นายเฉลิมพล	ชมภูพู่ รหัสนิสิต	58362124
	นางสาวณัฐนรี	นรพงษ์ รหัสนิสิต	58362247
	นายณัฐปคัลภ์	ฟูพงษ์ รหัสนิสิต	58362254
ที่ปรึกษาโครงการ	อาจารย์ชูพงศ์	ช่วยเทัญ	
ภาควิชา	วิศวกรรมเครื่องกล		
ปีการศึกษา	2561		

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ให้โครงการวิจัยฉบับนี้เป็นส่วนหนึ่งของการศึกษาดม
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล

คณะกรรมการสอบโครงการ

ที่ปรึกษาโครงการ

(ผู้ช่วยศาสตราจารย์ ดร. ชูพงศ์ ช่วยเทัญ)

กรรมการ

(ดร. ปองพันธ์ โอทกานนท์)

กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. อนันตชัย อยู่แก้ว)

ชื่อหัวข้อโครงการ	: ระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน		
ผู้ดำเนินโครงการ	: นายเฉลิมพล ชมภูพู่ รหัสนสิต	58362124	
	: นางสาวณัฐนรี นรพงษ์ รหัสนสิต	58362247	
	: นายณัฐปคัลภ์ พูพงษ์ รหัสนสิต	58362254	
ที่ปรึกษาโครงการ	: อาจารย์ชูพงศ์ ช่วยเพ็ญ		
สาขาวิชา	: วิศวกรรมเครื่องกล		
ภาควิชา	: วิศวกรรมเครื่องกล		
ปีการศึกษา	: 2561		

บทคัดย่อ

โครงการฉบับนี้จัดทำเพื่อการศึกษาและสร้างอัลกอริทึมสำหรับระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน โดยมีวัตถุประสงค์เพื่อเสริมสร้างและพัฒนาาระบบขับเคลื่อนอัตโนมัติในรถยนต์ต้นแบบของคณะวิศวกรรมศาสตร์ ภาควิศวกรรมเครื่องกล เพื่อกำหนดรูปแบบการตรวจจับรถยนต์ด้วยกล้องเว็บแคม รวมทั้งวิเคราะห์หาระยะห่างของยานพาหนะด้วยเซนเซอร์LiDAR และออกแบบอัลกอริทึมสำหรับควบคุมความเร็วอัตโนมัติแบบแปรผัน โดยคณะผู้จัดทำได้ศึกษาข้อมูลเกี่ยวกับโมเดลต่าง ๆ ที่ใช้สำหรับการตรวจจับวัตถุ ไปจนถึงการสร้างโมเดลการตรวจจับยานพาหนะพร้อมทั้งสร้างจุดศูนย์กลางของรถยนต์ภายในภาพ เพื่อนำมาวิเคราะห์หาระยะห่างของยานพาหนะในภาพ จากสมการความสัมพันธ์ของมุมเสมือนเซนเซอร์ลิดาร์ กับมุมจากเซนเซอร์ลิดาร์ ที่เกิดขึ้นจากสอบเทียบระหว่างกล้องและเซนเซอร์ลิดาร์ คณะผู้จัดทำได้ทำการทดลองโดยเก็บข้อมูลรูปภาพจากกล้องแคมและข้อมูลจากเซนเซอร์ลิดาร์ในสภาวะแวดล้อมเดียวกันที่ระยะห่างจริง 200, 408, 525, 862 และ 1200 เซนติเมตร ณ ความสูงของลิดาร์ที่ 96 เซนติเมตร เพื่อประเมินประสิทธิภาพของอัลกอริทึมการตรวจจับยานพาหนะพร้อมแสดงระยะห่าง จากการทดลอง พบว่าอัลกอริทึมที่ออกแบบให้ใช้สมการความสัมพันธ์จากการสอบเทียบมุมในภาพของกล้องแคมกับมุมจากเซนเซอร์ลิดาร์มีประสิทธิภาพ สามารถตรวจจับยานพาหนะ และหาระยะห่างของยานพาหนะในภาพได้ใกล้เคียงกับระยะจริงที่วัดได้

Project Title : Adaptive Cruice Control

Name : Mr. Chalernpol Chompupoo Code 58362124
Mrs. Natnaree Norapong Code 58362247
Mr. Nutpakan Foopong Code 58362254

Project Advisor : Chupong Chulpen

Major : Mechanical Engineering

Department : Mechanical Engineering

Academic year : 2018

Abstract

The project is made for studying and creating an algorithm for Adaptive Cruise Control (ACC) with the objective of enhancing and developing automatic driving systems in T-MAC. The vehicles are being developed by Development and Research of Innovative Vehicle Engineering (DRIVE) at Faculty of Engineering. Ours objective are determine the vehicle detection model with a web camera, and to analyze the distance of the vehicle with the LiDAR sensor and design algorithms for Adaptive Cruise Control. The organizing team studied information about detection models used for object detection including to create a vehicle detection model and creating the center of the vehicle within the image. The equation the relation of the assume angle from the sensor LIDA and the angle from the sensor LIDA. Created from calibrating between cameras and sensors LIDA. We are conducted experiments by collecting image data from camera and data from the sensor in the same environment at the actual distance of 200, 408, 525, 862 and 1200 centimeters at the height of the LIDA at 96 centimeters. Results to evaluate the efficiency of vehicle detection algorithms with distance. From the experiment, the algorithms can detect vehicles and show the distance. The organizers expect this project is useful in developing the algorithm for Adaptive Cruise Control.

กิตติกรรมประกาศ

โครงการระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน คณะวิศวกรรมเครื่องกลฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ทางคณะผู้ดำเนินงานต้องขอขอบพระคุณ อาจารย์ชูพงศ์ ช่วยเพ็ญ ผู้ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการที่กรุณาให้คำปรึกษา และชี้แนะแนวทางการแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นในระหว่างการดำเนินโครงการ ตลอดจนติดตาม ผลการดำเนินโครงการมาโดยตลอด ทางคณะผู้จัดทำโครงการขอขอบคุณท่านอาจารย์เป็นอย่างสูง มา ณ ที่นี้ ขอขอบพระคุณอาจารย์ทุกท่านที่อบรมสั่งสอน และให้ความรู้แก่ผู้ดำเนินงานขอขอบคุณฝ่ายเลขานุการภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ช่วยอำนวยความสะดวกในการดำเนินโครงการ สุดท้ายนี้ผู้ดำเนินงานขอขอบคุณผู้มีพระคุณทุกท่านที่ทำให้โครงการนี้สำเร็จลุล่วงไปด้วยดี หากเกิดข้อผิดพลาดประการใดจากโครงการนี้ ผู้ดำเนินการต้องขอกราบอภัยมา ณ ที่นี้ด้วย

นายเฉลิมพล

นางสาวณัฐนรี

นายณัฐปคัลภ์

ชมพู่

นรพงษ์

พูนพงษ์

สารบัญ

	หน้า
ใบรับรองโครงงาน	ก
บทคัดย่อ	ข
Abstract	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูปภาพ	ซ
ลำดับสัญลักษณ์	ณ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	3
1.3 ขอบเขตของโครงงาน	3
1.4 ขั้นตอนและแผนการดำเนินงาน	4
1.5 ผลที่คาดว่าจะได้รับ	4
1.6 งบประมาณที่ใช้	4
บทที่ 2 ทฤษฎีและวรรณกรรมปริทัศน์	5
2.1 ทฤษฎี	5
2.1.1 การเรียนรู้เชิงลึก (Deep learning)	5
2.1.2 โครงข่ายประสาทแบบคอนโวลูชันตามภูมิภาค (Region Based Convolutional Neural Network : R-CNN)	6
2.1.3 โครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค (Faster Region Based Convolutional Neural Network : Faster R-CNN)	8
2.1.4 YOLO	9
2.1.5 LiDAR	10
2.1.6 RPLIDAR A1M8	10
2.2 วรรณกรรมปริทัศน์	11
2.2.1 ความหมายของระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน	11

สารบัญ(ต่อ)

	หน้า
2.2.2 การพัฒนาจากระบบควบคุมความเร็วอัตโนมัติ	12
2.2.3 องค์ประกอบพื้นฐานของระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน	13
บทที่ 3 วิธีการดำเนินงาน	15
3.1. ศึกษาข้อมูลที่เกี่ยวข้อง	15
3.2 ตรวจจับยานพาหนะ (Vehicle Detection)	15
3.3 สร้างจุดศูนย์กลางของรถยนต์ภายในภาพ (Centroid)	24
3.4 การหาความสัมพันธ์ระหว่างมุมของวัตถุในภาพกับเซนเซอร์ลิดาร์	25
3.5 การหาระยะห่างจริงจากมุมเสมือนลิดาร์	29
3.6 แผนภาพแสดงลำดับขั้นตอนการทำงานของ Algorithm	30
บทที่ 4 ผลการทดลองและวิเคราะห์ผลการทดลอง	32
4.1 ผลการทดลอง	32
4.2 วิเคราะห์ผลการทดลอง	35
บทที่ 5 สรุปผลโครงการและข้อเสนอแนะ	37
5.1 สรุปผลการทดลอง	37
5.2 ข้อเสนอแนะ	38
5.2.1 ปัญหาที่พบ	38
5.2.2 ข้อเสนอแนะ	38
บรรณานุกรม	39
ประวัติผู้ดำเนินงาน	41

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 การจำลองหลักการทำงานของ Deep Learning	5
รูปที่ 2.2 ขั้นตอนการทำงานของ R-CNN	7
รูปที่ 2.3 ขั้นตอนการทำงานของ Faster R-CNN	8
รูปที่ 2.4 ขั้นตอนการทำงานของ YOLO	10
รูปที่ 2.5 อุปกรณ์ RPLIDAR A1M8	11
รูปที่ 2.6 การเปรียบเทียบระบบ Cruise Control กับ Adaptive Cruise Control	12
รูปที่ 2.7 Proposed System Architecture	13
รูปที่ 2.8 การทำงานของเซนเซอร์ LiDAR	14
รูปที่ 3.1 โค้ดดาวน์โหลดโปรแกรมผ่านการฝึกฝนการตรวจจับ	16
รูปที่ 3.2 โค้ดแยกไฟล์ข้อมูลรูปภาพยานพาหนะและใบนารีชุดข้อมูลยานพาหนะจริง	16
รูปที่ 3.3 โค้ดบรรจุข้อมูลในตาราง	17
รูปที่ 3.4 โค้ดแสดงผลภาพและทำความเข้าใจกับประเภทของภาพที่มีอยู่	17
รูปที่ 3.5 โค้ดแบ่งข้อมูลเพื่อนำไปฝึกฝนและประเมิน	18
รูปที่ 3.6 โค้ดกำหนดตัวเลือกสำหรับการฝึกฝน	18
รูปที่ 3.7 โค้ดการฝึกฝนโมเดล	19
รูปที่ 3.8 โค้ดตรวจจับยานพาหนะและสร้างกล่องขอบเขตยานพาหนะ	20
รูปที่ 3.9 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ของโมเดลโครงข่ายประสาทพัฒนาแบบคอนวูลูชัน	21
รูปที่ 3.10 โค้ดตรวจจับยานพาหนะและสร้างกล่องขอบเขตยานพาหนะ	22
รูปที่ 3.11 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ของโมเดลYOLOv2	23
รูปที่ 3.12 โค้ดสร้างจุดศูนย์กลางของรถยนต์ภายในภาพ	24
รูปที่ 3.13 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) สร้างจุดศูนย์กลางของรถยนต์ภายในภาพ	25
รูปที่ 3.14 กล่องเว็บแคมและเซนเซอร์ลิดาร์ที่ยึดติดกัน	26
รูปที่ 3.15 เตรียมวัตถุสำหรับเก็บข้อมูล	26
รูปที่ 3.16 วัตถุพิกเซลภาพระยะ a	27
รูปที่ 3.17 วัตถุพิกเซลภาพระยะ b	27
รูปที่ 3.18 สามเหลี่ยมมุมฉากตรีโกณมิติ	28
รูปที่ 3.19 โค้ดแสดงระยะห่างจริงจากมุมเสมือนเซนเซอร์ลิดาร์	29
รูปที่ 3.20 แผนผังการทำงานของอัลกอริทึม	31

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 4.1 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 200 เซนติเมตร	32
รูปที่ 4.2 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 408 เซนติเมตร	33
รูปที่ 4.3 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 525 เซนติเมตร	33
รูปที่ 4.4 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 862 เซนติเมตร	34
รูปที่ 4.5 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 1200 เซนติเมตร	34



สารบัญตาราง

		หน้า
ตารางที่ 1.1	ขั้นตอนการดำเนินงาน	3
ตารางที่ 3.1 เดียวกัน	แสดงข้อมูลระยะห่างพิกเซลภาพในภาพและข้อมูลมุมจากลิตาร์ในสภาพแวดล้อม	28
ตารางที่ 4.1	รายละเอียดของระยะห่างของยานพาหนะจากการวัดและผลจากอัลกอริทึม	35



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ระบบขับเคลื่อนอัตโนมัติ (Self-Driving Car) หรือเรียกอีกชื่อหนึ่งว่า รถยนต์ไร้คนขับ เป็นเทคโนโลยีที่พัฒนาเพื่อเพิ่มความปลอดภัยบนท้องถนน เพิ่มประสิทธิภาพในการใช้เชื้อเพลิง และประหยัดระยะเวลาในการเดินทางบนท้องถนน ในอดีตที่ผ่านมาการคมนาคมทางบกได้มีการพัฒนาอย่างต่อเนื่อง จนเข้าสู่ยุคปัจจุบันที่ได้มีการพัฒนามาสู่ระบบขับเคลื่อนอัตโนมัติ ได้แบ่งระดับการขับเคลื่อนอัตโนมัติตามความสามารถเริ่มจากระดับเริ่มต้นคือการใช้เทคโนโลยีเป็นผู้ช่วยในการขับขี่ ประมวลผลเพื่อช่วยในการตัดสินใจ ไปจนถึงระดับที่รถยนต์ปราศจากผู้ขับ กล่าวคือรถยนต์สามารถขับเคลื่อนอัตโนมัติไปยังสถานที่ต่าง ๆ โดยผู้ขับไม่จำเป็นต้องทราบเส้นทาง หรือพวงมาลัยซึ่งเป็นอุปกรณ์สำคัญที่ผู้ขับใช้ควบคุมทิศทางของรถยนต์ไม่มีความจำเป็นอีกต่อไป แม้ว่าระบบการคมนาคมทางบกจะต้องใช้เวลาอีกหลายปีจนกว่าจะเริ่มเข้าสู่รถยนต์ขับเคลื่อนอัตโนมัติอย่างเต็มตัว แต่รถยนต์ที่เราใช้งานในทุกๆวันก็ถูกนับว่าเป็นรถยนต์ที่มีระบบขับเคลื่อน ‘กึ่งอัตโนมัติ’ (Semi-Autonomous) และมีเทคโนโลยีช่วยเหลือผู้ขับในตัว (Driver Assist) ซึ่งเทคโนโลยีดังกล่าวถูกออกแบบมาเพื่อช่วยตรวจจับและวิเคราะห์อาการของรถที่มีความไม่สมบูรณ์ก่อนที่ผู้ขับจะรู้ว่าต้องการการช่วยเหลือเสียอีก เป็นที่แน่ชัดว่าเทคโนโลยีช่วยขับในปัจจุบันเหล่านี้ฉลาดพอที่จะคิดแทนและตัดสินใจชะลอการขับรถของผู้ขับได้ในบางสถานการณ์

แม้ปัจจุบันอาจจะยังมีความคิดเห็นว่าการช่วยเหลือผู้ขับขึ้นนั้นยังเป็นสิ่งไม่จำเป็น เป็นการติดตั้งเพียงเพื่อสร้างความน่าสนใจในความแปลกใหม่เท่านั้น แต่แท้จริงแล้วเทคโนโลยีช่วยเหลือผู้ขับนั้นสามารถช่วยลดอุบัติเหตุบนท้องถนนได้จริง และยังมีการพัฒนาแก้ไขจุดบกพร่องสร้างจุดแข็งต่อไปอย่างไม่มีที่สิ้นสุด ระบบควบคุมความเร็วอัตโนมัติ หรือ Cruise Control เป็นระบบควบคุมความเร็วที่มีจุดประสงค์สำหรับการเดินทางไกลที่มีระยะเวลาการเดินทางที่นาน เดิมทีนั้นการเดินทางระยะไกลจะสร้างความเมื่อยล้าต่อไปหาได้ง่าย เนื่องจากผู้ขับขี่จะต้องกดคันเร่งตลอดเวลาการเดินทาง การใช้คันเร่งด้วยความเร็วที่คงที่นั้นสามารถช่วยให้รถสามารถประหยัดพลังงานและเชื้อเพลิงได้มากขึ้นได้ อันเนื่องมาจากเครื่องยนต์

สามารถส่งจ่ายน้ำมันได้คงที่ตลอดเวลา การใช้งานระบบสามารถทำได้โดยการตั้งค่าระบบการเปิด-ปิด โดยผู้ขับเอง ระบบไม่สามารถทำงานได้เองโดยอัตโนมัติ ตามคำแนะนำนั้นระบบจะสามารถใช้ได้ผลดีที่สุด ก็ต่อเมื่อผู้ขับเปิดการใช้งานระบบในช่วงที่การจราจรไม่คับคั่ง กล่าวคือรถยนต์สามารถวิ่งต่อเนื่องในระยะ ยาวได้ ระบบนั้นไม่สามารถลดความเร็วได้เองแต่จะควบคุมความเร็วตามที่ผู้ขับได้ตั้งค่าไว้เท่านั้น ทั้งนี้ ระบบเป็นระบบที่มีความน่าสนใจและสามารถที่จะนำไปพัฒนาเพิ่มศักยภาพให้ดียิ่งขึ้นอีกด้วย ระบบได้ถูก พัฒนาต่อจนกลายเป็นระบบควบคุมความเร็วในการขับขี่ให้สอดคล้องกับสภาพของการจราจรอย่าง อัตโนมัติ (Adaptive Cruise Control) โดยอาศัยข้อมูลจากตรวจจับสภาพการจราจรจาก เซนเซอร์ เรดาร์ หรือภาพจากกล้อง และนำมาวิเคราะห์ เพื่อควบคุมความเร็วของรถให้เหมาะสมและปลอดภัย ซึ่งมี กระบวนการการทำงานดังนี้

1. กำหนดความเร็วสูงสุดในการขับขี่
2. ตรวจจับระยะห่างจากรถยนต์คันด้านหน้า ด้วยเซนเซอร์ต่าง ๆ
3. ควบคุมความเร็วอัตโนมัติ จากการวิเคราะห์ระยะห่างจากรถคันด้านหน้า

ปัจจุบันทางมหาวิทยาลัยนครสวรรค์มีโครงการพัฒนารถไฟฟ้าขับเคลื่อนอัตโนมัติของศูนย์วิจัยและพัฒนา วิศวกรรมยานยนต์ (Development and Research of Innovative Vehicle Engineering : DRIVE) คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมเครื่องกล เป็นโครงการพัฒนารถยนต์ต้นแบบ T-MEC ในการ ขับเคลื่อนอัตโนมัติ โดยใช้พลังงานไฟฟ้าในการขับเคลื่อน ซึ่งอยู่ในระหว่างการพัฒนา จึงเป็นเหตุผลใน การทำวิจัยเพื่อควบคุมการขับเคลื่อนของรถ T-MEC ให้สอดคล้องกับสภาพการจราจร ซึ่งอยู่ในระบบ ขับเคลื่อนอัตโนมัติ

ทั้งหมดที่กล่าวมานั้นเป็นที่มาและความสำคัญของโครงการ เพื่อสร้างสรรค์และพัฒนาระบบ ขับเคลื่อนอัตโนมัติ ด้วยการใช้กระบวนการประมวลผลภาพและการประมวลผลด้วย LiDAR โดยการ วิเคราะห์วัตถุในช่องทางเดินรถ ขนาดและระยะห่างของรถกับวัตถุในช่องทางเดินรถ เพื่อควบคุมการ ทำงานของเบรก และคันเร่งอัตโนมัติให้เหมาะสม และปลอดภัย ทั้งยังสอดคล้องกับข้อมูลที่นำมา ประมวลผล

1.2 วัตถุประสงค์ของการศึกษา

- 1) เพื่อสร้างเสริมสร้าง และพัฒนาระบบขับเคลื่อนอัตโนมัติ (Self-Driving) ในรถยนต์ต้นแบบของคณะวิศวกรรมศาสตร์ ภาควิศวกรรมเครื่องกล มหาวิทยาลัยนเรศวร
- 2) เพื่อกำหนดรูปแบบการตรวจจ็ปรถยนต์ด้วยกล้อง และวิเคราะห์ระยะห่างวัตถุด้วยเซนเซอร์ LiDAR
- 3) เพื่อให้ได้อัลกอริทึมต้นแบบสำหรับควบคุมความเร็วอัตโนมัติแบบแปรผัน (Adaptive cruise control)

1.3 ขอบเขตของโครงการ

- 1) ใช้ภาพและเซนเซอร์ LiDAR ในการตรวจจ็ปรถยนต์และวิเคราะห์วัตถุที่ต้องการตรวจจ็ปร
- 2) เป็นการควบคุมความเร็วอัตโนมัติแบบแปรผันในสถานการณ์จำลองที่กำหนด
- 3) ระยะเวลาในการดำเนินการวิจัย ตั้งแต่เดือน สิงหาคม พ.ศ.2561 ถึง พฤษภาคม พ.ศ. 2562

1.4 ขั้นตอนและแผนการดำเนินงาน

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

ลำดับ	กิจกรรม	2561					2562					
		ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.
1	ศึกษาทฤษฎีรวมถึงข้อมูลรายละเอียดต่างๆ ของงานวิจัยที่เกี่ยวข้อง											
2	ออกแบบและพัฒนาการตรวจจ็ปรถยนต์ด้วยเซนเซอร์ LiDAR											
3	สร้างอัลกอริทึมในการวิเคราะห์ และควบคุมความเร็วอัตโนมัติแบบแปรผันของรถยนต์ต้นแบบ											
4	ทดสอบ และประเมินผลประสิทธิภาพของระบบ											
5	วิเคราะห์และสรุปผลการดำเนินงาน											
6	จัดทำรูปเล่มและนำเสนอโครงการ											

1.5 ผลที่คาดว่าจะได้รับ

- 1) อัลกอริทึมสามารถตรวจจับ และวิเคราะห์การเคลื่อนที่ของรถยนต์ ด้านหน้าด้วยการประมวลผลด้วยภาพ และการประมวลผลด้วยเลเซอร์ LiDAR เพื่อควบคุมความเร็วอัตโนมัติในรถยนต์ต้นแบบ
- 2) เป็นข้อมูลและแนวทางการนำไปใช้เบื้องต้นของระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน
- 3) ผลการวิจัยนี้เป็นแนวทางในการพัฒนาระบบการขับเคลื่อนอัตโนมัติ ซึ่งอยู่รถยนต์ไฟฟ้าต้นแบบของคณะวิศวกรรมศาสตร์ ภาควิศวกรรมเครื่องกล มหาวิทยาลัยนเรศวร

1.6 งบประมาณที่ใช้

- 1) กล้องเว็บแคม 3,290 บาท
- 2) เซนเซอร์ลิดาร์ RPLiDAR 3,290 บาท
- 3) จัดทำรูปเล่ม 500 บาท
- 4) อุปกรณ์อื่นๆ 500 บาท



บทที่ 2

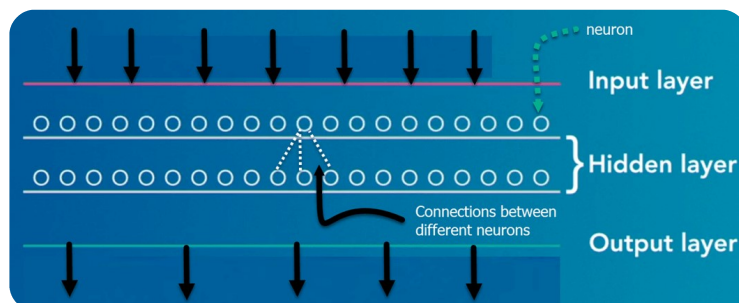
หลักการและทฤษฎี

2.1 ทฤษฎี

2.1.1 การเรียนรู้เชิงลึก (Deep learning)

Deep learning คือ ซอฟต์แวร์คอมพิวเตอร์ที่เลียนแบบการทำงานของระบบโครงข่ายประสาท หรือ Neurons ในสมองมนุษย์ โดยอัลกอริทึม (Algorithm) จะถูกสร้างขึ้นเพื่อการเรียนรู้ของเครื่องจักรหรือคอมพิวเตอร์ โดยชุดคำสั่งจะทำให้เครื่องจักรสามารถประมวลผลข้อมูลจำนวนมากด้วยการจำลองโครงข่ายประสาทแบบเดียวกันกับการทำงานของสมองมนุษย์ จากการนำเอาโครงข่ายประสาทเทียม (Neural Network) หลายๆ เลเยอร์ (Layer) มาต่อกัน เลเยอร์ลำดับแรกจะทำหน้าที่ในการรับข้อมูล (Input Layer) ในขณะที่เลเยอร์สุดท้ายจะทำหน้าที่ส่งผลลัพธ์การประมวลผลออกมา (Output Layer) ส่วนเลเยอร์ที่อยู่บริเวณระหว่างเลเยอร์แรกสุด และเลเยอร์สุดท้าย จะถูกเรียกว่า Hidden Layer ซึ่งนับได้ว่าการเรียนรู้เชิงลึกถือเป็นขั้นเซตของการเรียนรู้ของเครื่อง (Machine Learning)

คำว่า Deep Learning มีที่มาจากการใช้เลเยอร์ของโครงข่ายประสาทเทียมหลายอันมาต่อกัน กล่าวได้ว่ามี Hidden layer มากกว่า 2 เลเยอร์ขึ้นไป เนื่องจากเลเยอร์เหล่านี้เป็นโครงสร้างที่จัดเก็บแบบเป็นกองซ้อน (Stack) จึงเปรียบได้ว่าเลเยอร์จำนวนเยอะๆ ก็จะทำให้มีโครงสร้างที่ลึก (Deep) ยิ่งขึ้นนั่นเอง



รูปที่ 2.1 การจำลองหลักการการทำงานของ Deep Learning

(ที่มา : <https://www.thaiprogrammer.org/2018/12/deep-learning-คืออะไร/> , 2556)

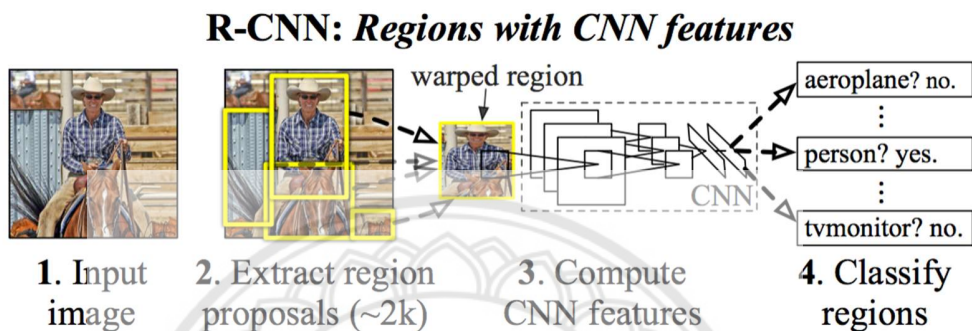
โดย Hidden layer ของแต่ละเลเยอร์จะเปรียบเสมือนว่าประกอบด้วย เซลล์ประสาท (Neural) จำนวนมาก ซึ่งมีหน้าที่ในการประมวลผล รับข้อมูลจากเลเยอร์ ที่อยู่เหนือกว่า และส่งข้อมูลที่ประมวลผลเสร็จแล้วไปยังเลเยอร์ที่อยู่ล่างกว่า ข้อดีของการส่งข้อมูลแบบนี้ก็คือแต่ละเลเยอร์สามารถที่จะมีค่าถ่วงน้ำหนัก (Weight) ค่าความเอนเอียงของข้อมูล (Bias) และ วิธีการประมวลผลทางคณิตศาสตร์ (Activation Function) ที่เป็นอิสระต่อกันได้ ยิ่งเราป้อนข้อมูลให้กับโมเดลมากเท่าไร เลเยอร์แต่ละเลเยอร์ก็จะสามารถสกัดลักษณะที่มีความซับซ้อนมากยิ่งขึ้น

กระบวนการทำงานของการเรียนรู้เชิงลึกโดยโมเดลให้ความแม่นยำ (Accuracy) ที่สูงในหลายๆปัญหา ตั้งแต่การตรวจจับวัตถุ (Object Detection) ไปจนถึงการรู้จำเสียงพูด (Speech Recognition) โดยที่ผู้ใช้งานไม่จำเป็นต้องให้ความรู้พื้นฐานใด ๆ ไว้ก่อนล่วงหน้าเลย ผู้ใช้งานเพียงให้ข้อมูลตัวอย่าง (Input data) ระบบก็จะทำการเรียนรู้จากข้อมูลและสังเคราะห์สู่องค์ความรู้ออกมาได้อย่างอัตโนมัติ อาทิเช่น ระบบสแกนใบหน้าหรือ Face ID ของ Apple โดยใช้วิธียิงจุดอินฟราเรดออกไปเพื่อใช้ในการตรวจสอบรูปของหน้า แม้ลักษณะทางใบหน้าเปลี่ยนแปลงเล็กน้อยหรือการเสริมอุปกรณ์ตกแต่งร่างกาย ระบบสแกนใบหน้ายังคงสามารถตรวจสอบได้ นั่นเป็นเพราะซอฟต์แวร์การเรียนรู้เชิงลึกเรียนรู้รูปแบบลักษณะหน้าที่แตกต่างกันจำนวนมาก และสร้างโมเดลให้คอมพิวเตอร์เรียนรู้ หรือระบบแนะนำหนังที่ผู้ใช้อาจมีความสนใจโดย Netflix โดยระบบจะการค้นหาและเชื่อมโยงคอนเทนต์ที่ผู้ใช้งานอาจมีความสนใจ เพื่อพยายามชักจูงให้ผู้ใช้ตัดสินใจชมคอนเทนต์นั้น ๆ โดยเฉพาะกับเนื้อเรื่องที่เราอาจไม่ค่อยคุ้น โดยนำเสนอด้วยภาพกราฟิกที่เหมาะสมกับแต่ละคน เช่น ภาพฉากการกระทำของตัวละครที่ผู้ใช้อาจมีความสนใจ , ภาพนักแสดงที่ผู้ใช้คุ้นเคย เป็นต้น ทั้งนี้การเรียนรู้เชิงลึกสามารถปรับใช้ได้กับระบบที่ต้องการพัฒนาให้มีความฉลาดมากยิ่งขึ้นได้ทุกรูปแบบ แม้แต่เทคโนโลยีใกล้ตัวดังเช่นโทรศัพท์มือถือก็มีการใช้การเรียนรู้เชิงลึกเป็นส่วนหนึ่งของระบบปฏิบัติการเช่นกัน [6][7][8]

2.1.2 โครงข่ายประสาทแบบคอนโวลูชันตามภูมิภาค (ภาพ)

โครงข่ายประสาทแบบคอนโวลูชันตามภูมิภาค หรือตัวย่อภาษาอังกฤษว่า R-CNN เป็นโครงข่ายประสาทเทียมหนึ่ง โดยพัฒนามาจากโครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network) เพื่อจะจำลองการมองเห็นของมนุษย์ที่มองพื้นที่เป็นที่ย่อย ๆ ในลำดับแรกและนำกลุ่มของพื้นที่ย่อย ๆ มาผสานกันในลำดับต่อไป เพื่อให้ได้คำตอบที่ว่าสิ่งที่เราเห็นนั้นคือวัตถุอะไร และการรวมผลจากการตรวจจับหลายภูมิภาคให้คงเหลือเพียงสิ่งที่ควรจะตอบเท่านั้น โดยใช้อัลกอริทึมที่ชื่อว่า Non-Maximum Suppression (NMS) โดยพิจารณาจากค่าความน่าจะเป็นวัตถุนั้น ๆ ร่วมกับค่า IOU (Intersection Over Union) กับทุกคำตอบ เพื่อเลือกผลลัพธ์ที่ดีที่สุดออกมาซึ่งอัลกอริทึมนี้ใช้กับการตรวจจับวัตถุ ที่กาลเวลาทำให้เทคโนโลยีต่าง ๆ มีการพัฒนาถึงที่มีอัลกอริทึมที่สามารถเสนอบริเวณที่อาจจะมีวัตถุอยู่ โดยอัลกอริทึมเหล่านี้ใช้พลังประมวลผลไม่มาก

นัก มีความเร็วและความคลาดเคลื่อนที่อยู่ในระดับที่รับได้ เมื่อมีอุปกรณ์และเทคโนโลยีที่พัฒนามาเหมาะสมกับการพัฒนา ทำให้โครงข่ายประสาทแบบคอนโวลูชันถูกพัฒนามาสู่โครงข่ายประสาทแบบคอนโวลูชันตามภูมิภาค R-CNN มีขั้นตอนการทำงานไม่ค่อยซับซ้อนนัก แต่ระบบต้องใช้เวลาระดับหนึ่งในการประมวลผล โดยมีหลักการทำงานดังนี้



รูปที่ 2.2 ขั้นตอนการทำงานของ R-CNN

(ที่มา : <https://medium.com/@sapjunior/object-detection-part-1-37bf39dc557f/> , 2561)

1) ทำการเรียกวัตถุ (Object Proposal) โดยใช้การเลือกค้นหา(Selective Search) ว่า บริเวณภายในภาพบริเวณไหนที่มีวัตถุที่เราต้องการ

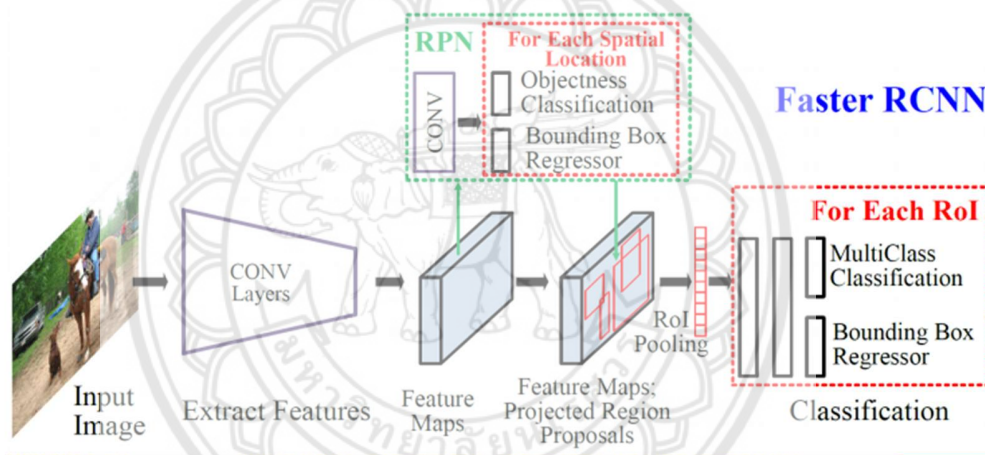
2) ปรับขนาดหรือนำบางส่วนของบริเวณภาพที่ถูกคัดเลือกมา(Region) ที่คาดว่าอาจจะมีวัตถุอยู่โดยไม่ต้องคำนึงถึงอัตราส่วนของภาพ โดยปรับขนาดย่อหรือขยายบริเวณภาพที่คาดว่าจะมีวัตถุที่เราสนใจให้มีขนาดเท่าที่ส่วนคัดกรอง(Classifier)ต้องการ แล้วส่งข้อมูลตามส่วนคัดกรองครั้งละบริเวณภาพที่ถูกคัดเลือกมา ว่าส่วนนั้นคืออะไร โดยใช้โครงข่ายประสาทแบบคอนโวลูชันเป็นตัวแยกคุณลักษณะเพื่อสร้างคุณลักษณะสำคัญ จากนั้นจึงใช้อัลกอริทึมซัพพอร์ทเวกเตอร์แมชชีน (Support Vector Machine : SVM) เพื่อทำการจัดหมวดหมู่อีกครั้งหนึ่ง

3) เนื่องจากการเรียกวัตถุบางครั้งเกิดการคลาดเคลื่อน ทำให้ได้บริเวณภาพที่ถูกคัดเลือก อาจไม่มีวัตถุอยู่ จึงทำให้มีพัฒนาอัลกอริทึมที่ทำการสอนเพื่อปรับแก้พิกัดของวัตถุที่จะตอบออกไป

เนื่องจากการเลือกค้นหาทำงานไม่เร็วนัก ในส่วนของการจัดหมวดหมู่ก็การทำงานที่ใช้เวลาอย่างสิ้นเปลืองและใช้พลังในการประมวลผลที่มาก และโครงข่ายประสาทแบบคอนโวลูชันตามภูมิภาคเป็นอัลกอริทึมที่เป็นกรรมสิทธิ์ สามารถใช้งานได้เฉพาะบนโปรแกรม MATLAB เท่านั้นจึงได้มีการพัฒนาอัลกอริทึมต่อไป

2.1.3 โครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค (Faster Region Based Convolutional Neural Network : Faster R-CNN) [11][1]

Faster R-CNN คือ การนำเอา Selective Search มารวมไว้ในก้อน Neural Networks (NN) ก่อน NN นี้มีส่วนประกอบหลัก 3 ส่วนคือ 1. ส่วนที่ทำหน้าที่คัดเลือก feature 2. ส่วน Region Proposal Network (RPN) ที่ทำหน้าที่คัดเลือกเอาบริเวณที่ “น่าจะ” เป็นวัตถุจาก feature map และ 3. ส่วนจำแนกประเภท ที่นำเอา feature map และ region ที่ได้จาก RPN มาประมวลผลโดยทำ RoI pooling เพื่อตอบว่าบริเวณใดของภาพมีวัตถุอะไรอยู่บ้างและวัตถุที่ต้องการอยู่ตรงส่วนไหน ทำให้การทำงานของ Faster R-CNN เร็วกว่า และยิ่งใกล้เคียงกับคำว่า Realtime อีกด้วย



รูปที่ 2.3 ขั้นตอนการทำงานของ Faster R-CNN

(ที่มา : [https://blog.datawow.io/ย้อนรอย-object-detection-และเจาะลึก-retinanet-ea1407ad7325 /](https://blog.datawow.io/ย้อนรอย-object-detection-และเจาะลึก-retinanet-ea1407ad7325/), 2561)

โดย Faster R-CNN ประกอบด้วยส่วนแรก process รูปและส่งออกไปส่วนที่สอง feature map หรือ NN และส่งต่อไปยังส่วนสุดท้าย Classification โดยต้องผ่าน RoI Pooling จาก feature maps ที่ได้ก่อน

- 1) ภาพนำเข้าก็ยังไม่ผ่าน Selective Search เพื่อหา proposed regions เหมือนเดิม
- 2) ภาพนำเข้าทั้งภาพ ไปผ่าน CNN Barebone ตัวที่เลือกไว้ จะ VGG-16, Resnet, Inception หรืออะไรก็ได้ แต่ network ที่เลือกมานั้น จะตัด fully connected layer ทิ้งไปทั้งหมด จะใช้ผล feature map จาก convolution layer สุดท้ายเท่านั้น

3) แต่ละ proposed region ที่ได้จาก Selective Search ที่เป็นพิกัดตำแหน่งบนภาพ นำเข้า จะถูกนำมาหาพิกัดตำแหน่งที่สัมพันธ์กัน กับขนาดของ feature map จาก convolution layer สุดท้าย (เพราะผ่าน CNN มา feature mapจะมีขนาดเล็กลง)

4) จากพิกัดตำแหน่งบน feature map convolution layer สุดท้ายที่ได้ นำ region ตรงนั้นไปผ่าน ROI Pooling Layer เพื่อให้ได้ feature vector ที่มีขนาดคงที่ ซึ่งจะถูกส่งไปที่ fully connected layer กับ softmax เพื่อทำ classification ว่า proposed region feature นั้นจะตอบว่า เป็นคลาสอะไร (สังเกตว่าไม่ได้ใช้ SVM มาเป็น classifier แล้ว ต่างจาก R-CNN เพราะ SVM train ยาก และธรรมชาติของมันเองไม่ได้เป็น multiclass classifier)

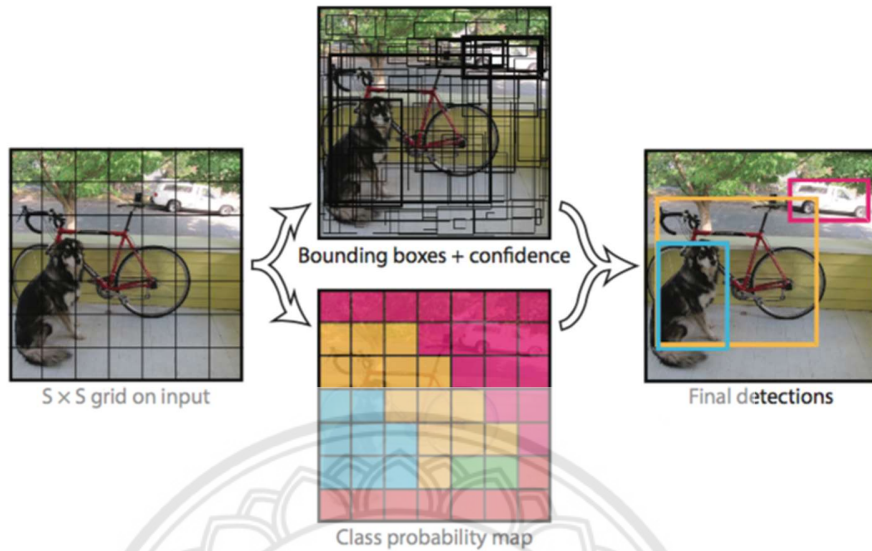
5) จากพิกัดตำแหน่ง บนภาพนำเข้าจริงของ proposed region ก็จะถูกส่งเข้าไปที่ regressor network ร่วมกับ feature vector ที่ได้จาก ROI Pooling เพื่อปรับแก้พิกัดของวัตถุที่จะตอบ ออกไป [12]

2.1.4 YOLO

โมเดล YOLO มีระบบและขั้นตอนการทำงานคล้ายกับ Faster R-CNN แต่ความเร็วในการประมวลผลนั้นเร็วกว่ามาก แตกต่างกันตรงที่โมเดลจะทำการทำนายตำแหน่ง ขนาดของกล่องขอบเขต และความน่าจะเป็นที่จะมีกล่องล้อมรอบขอบเขตวัตถุคือ

1) จากการทายกล่องขอบเขตแล้วจึงส่งไปคัดกรอง แต่โมเดลYOLOนั้นทำการทำนายทั้งกล่องขอบเขตและความน่าจะเป็นของคลาสต่าง ๆ ออกมาพร้อมกัน

2) จากการทำนายค่าต่าง ๆ จากภาพนั้น จะทำการแบ่งภาพออกเป็นส่วน ๆ แทน สำหรับแต่ละส่วนจะยังคงทำนายกล่องขอบเขตและคลาส ซึ่งเราสามารถนำมารวมกันเพื่อเลือกคู่คลาสและกล่องขอบเขตที่คะแนนสูงสุดได้ [10]



รูปที่ 2.4 ขั้นตอนการทำงานของ YOLO

(ที่มา : [https://medium.com/@sanparithmarukatat/โลกหมุนไป-งานวิจัยก็หมุนตาม-46ae76d4e195 /](https://medium.com/@sanparithmarukatat/โลกหมุนไป-งานวิจัยก็หมุนตาม-46ae76d4e195/) ,)

2.1.5 LIDAR

เทคโนโลยีไลดาร์ หรือ ลิดาร์ (LIDAR) ย่อมาจาก Light Detection and Ranging ทำงานโดยการยิงลำแสงเลเซอร์ไปยังพื้นผิวที่ต้องการเก็บข้อมูล โดยข้อมูลที่ได้จะเป็นข้อมูลจากการวัดระยะเวลาในการเดินทางของลำแสงเลเซอร์ที่สะท้อนกลับมายังเซนเซอร์เริ่มต้น ซึ่งสามารถนำข้อมูลที่ได้มาสร้างภาพเพื่อในไปใช้ประโยชน์ รวมทั้งยังได้ภาพของพื้นผิวที่มีรายละเอียดมากกว่าภาพถ่ายดาวเทียม ลิดาร์มีหลักการคล้ายกับเรดาร์ แต่มีความแตกต่างกันที่ เทคโนโลยีเรดาร์ใช้คลื่นวิทยุเป็นเครื่องมือในการระบุระยะห่างของเรดาร์กับพื้นที่ที่สนใจ [3][4]

2.1.6 RPLIDAR A1M8

RPLIDAR รุ่น A1M8 คือลิดาร์ที่สามารถสแกนรอบทิศทาง 360 องศา มีความถี่ในการสแกนอยู่ที่ 5.5 ถึง 10 เฮิร์ตซ์ และมีความถี่ในการสุ่มตัวอย่าง 4,000 – 8,000 เฮิร์ตซ์ โดยระยะในการตรวจสอบสูงสุดที่ 12 เมตร อาร์พีลิดาร์ เอ-วัน (RPLIDAR A1) เป็นอุปกรณ์เลเซอร์สแกนเนอร์ชนิด 2 มิติ ถูกพัฒนาโดยบริษัท SLAMTEC โดยข้อมูลที่ได้สามารถนำไปสร้างแผนภาพจำลองสภาพแวดล้อมและวัตถุ มีระบบการวัดเลเซอร์แบบสามเหลี่ยม สามารถทำงานในสภาพแวดล้อมในร่มและกลางแจ้งทุกประเภท โดยไม่จำเป็นต้องถูกแสงแดดโดยตรง อาร์พีลิดาร์ เอ-วัน ประกอบด้วยระบบสแกนและระบบมอเตอร์ เมื่อเปิดการใช้งานอาร์พีลิดาร์

เอ-วันจะเริ่มหมุนและสแกนตามเข็มนาฬิกาอาร์ฟิไลดาร์ เอ-วัน อยู่บนพื้นฐานของหลักการเลเซอร์ที่หลากหลาย และใช้การได้มาซึ่งการมองเห็นความสูง [15]



รูปที่ 2.5 อุปกรณ์ RPLIDAR A1M8

2.2 วรรณกรรมปริทัศน์

การวิจัยเรื่อง “ระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน” คณะผู้วิจัยได้ทำการศึกษาเอกสาร และงานวิจัยที่เกี่ยวข้องกับรูปแบบการเรียนรู้ โดยนำเสนอตามลำดับ ดังนี้

2.2.1 ความหมายของระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน

Ritesh Kanjee et al. (2013) ให้คำนิยามระบบควบคุมความเร็วอัตโนมัติแบบแปรผันเป็นการเคลื่อนที่แบบอัตโนมัติที่ใช้เซ็นเซอร์และระบบควบคุม ช่วยให้ผู้ใช้ขับขี่รักษาความเร็วในการขับขี่ที่ปลอดภัย และระยะห่างสัมพันธ์กับรถด้านหน้า ระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน ให้การควบคุมระยะทางในช่องจราจรดังต่อไปนี้:

- 1) ค้นหาหนทางเป้าหมาย
- 2) กำหนดตำแหน่งที่รถเป้าหมายอยู่ในช่องจราจร
- 3) วัดระยะทางไปยังยานพาหนะเป้าหมาย
- 4) ใช้คันเร่งหรือเบรกที่จำเป็น เพื่อรักษาระยะห่างที่ปลอดภัยจากรถเป้าหมาย

ถ้าไม่พบรถเป้าหมายอยู่ในช่องจราจร ระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน จะกลับสู่

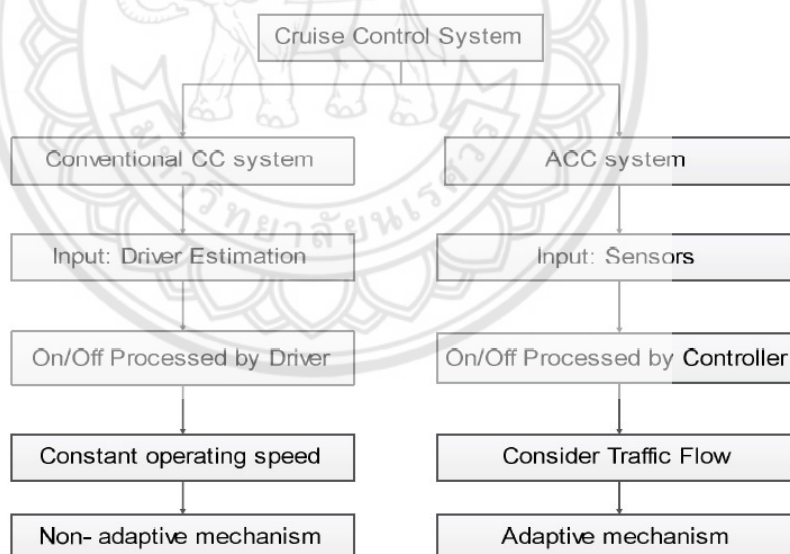
- 5) ความเร็วที่กำหนดไว้ล่วงหน้า

จากข้อความที่กล่าวมาข้างต้นการทำงานของระบบ กล่าวได้คือ ระบบจะรักษาความเร็วอัตโนมัติแปรผันตามสภาพการจราจร หมายความว่า ระบบจะชะลอ หรือเบรกเพื่อหยุดเองโดยอัตโนมัติ ในขณะที่ผู้ใช้ขับขี่ไม่ต้องควบคุมเบรกหรือคันเร่ง โดยระบบจะรักษาระยะห่างจากรถคันด้าน

2.2.2 การพัฒนาจากระบบควบคุมความเร็วอัตโนมัติ

Ibrahim Hussein et al. (2017) ในกรณีของระบบควบคุมความเร็วอัตโนมัติแบบเดิม มีความเสี่ยงอันตรายต่อการใช้งาน ด้วยเหตุผลหลายประการ เช่น เวลาตอบสนองของผู้ขับขี่ ในการควบคุม ยานพาหนะแตกต่างกัน เวลาที่ต่างกันนี้อาจส่งผลทำให้เกิดอุบัติเหตุขึ้น เป็นเหตุผลในการพัฒนาระบบ ควบคุมความเร็วอัตโนมัติขึ้น จากรูปที่ 1 แสดงถึงการเปรียบเทียบระบบ Cruise Control กับ Adaptive Cruise Control การเปิดใช้งาน ขึ้นอยู่กับผู้ขับขี่ เมื่อผู้ขับขี่เร่งความเร็วจนถึงความเร็วที่ต้องการ ทำการ เปิดระบบควบคุมความเร็วอัตโนมัติความเร็วขณะนี้ถูกล็อกไว้ โดยระบบจะไม่คำนึงถึงสภาพการจราจร ผู้ ขี่สามารถทำการปิดระบบเมื่อใดก็ได้

จากข้อความที่กล่าวมาข้างต้น ระบบควบคุมความเร็วอัตโนมัติแบบเดิม ผู้ขับขี่จำเป็นต้องควบคุม ยานพาหนะให้เคลื่อนที่อย่างปลอดภัย เช่นนั้นในระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน ไม่ จำเป็นต้องใช้การควบคุมจากผู้ขับขี่ เซนเซอร์จะทำการวัดระยะและเก็บข้อมูลมาประมวลผลควบคุม ยานพาหนะ



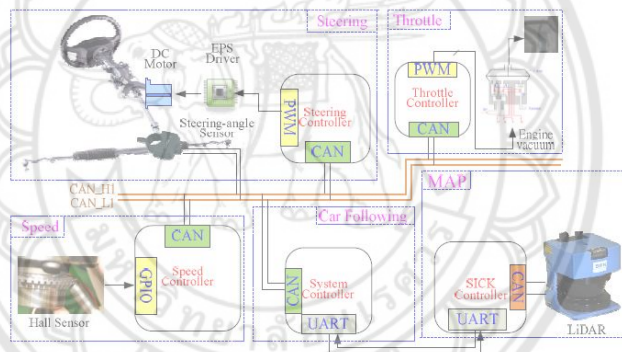
รูปที่ 2.6 การเปรียบเทียบระบบ Cruise Control กับ Adaptive Cruise Control

(ที่มา : <https://www.semanticscholar.org/paper/Modeling-and-Simulation-of-Intelligent-Cruise-Hussein-Hourani/b494ed35af943564b56f6548b9f9cd088f2b5790>, 2560)

2.2.3 องค์ประกอบพื้นฐานของระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน

1) แผนภาพการทำงานของระบบ

Chan Wei Hsu et ls. (2012) แผนภาพการทำงานของระบบแสดงดังรูปที่ 1 เป็นรูปแบบแนวคิดที่กำหนดโครงสร้างพฤติกรรมและมุมมองเพิ่มเติมของระบบ ฐานข้อมูลทั้งหมดรองรับด้วยไมโครคอนโทรลเลอร์เป็นหลัก ที่การพัฒนาขั้นตอนวิธีการเข้าถึงข้อมูลนำเข้าและโปรแกรมข้อมูลนำออกเพื่อให้บรรลุเป้าหมายการใช้งาน ข้อมูลรูปภาพ รวมทั้งข้อมูล LiDAR ในส่วนของตัวควบคุมมีการตอบสนอง เพื่อสื่อสารกับตัวควบคุม ไมโครคอนโทรลเลอร์แต่ละตัว จะส่งคำสั่งไปยังอุปกรณ์ปลายทางหรือได้รับสัญญาณ ตอบรับผ่านบัส CAN (CAN bus) ในขณะเดียวกัน ไมโครคอนโทรลเลอร์แต่ละตัวเปิดใช้งานโดยคำสั่งเริ่มต้นของระบบ ข้อมูลที่ได้รับจะมีขั้นตอนการควบคุมเวลา โดยการแบ่งความถี่ ด้วยเหตุนี้ ข้อมูลทั้งหมดในการตรวจจับ จะเข้าถึงได้โดยการเลือกพอร์ต

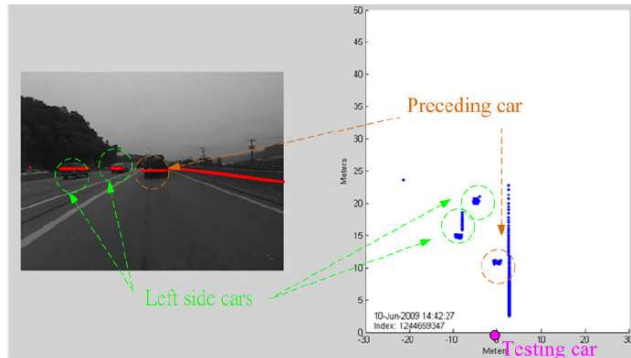


รูปที่ 2.7 Proposed System Architecture

(ที่มา : https://www.artc.org.tw/chinese/03_service/03_02detail.aspx?pid=1448,2555)

2) LiDAR

Chan Wei Hsu et ls. (2012) เนื่องจากความสามารถในการสร้างแผนภาพตรวจจับสภาพแวดล้อมทางกายภาพ LiDAR ได้กลายเป็นเทคโนโลยีที่โดดเด่นในเรื่องรูปทรงและขนาด เป็นเซ็นเซอร์ที่ใช้เพื่อให้ข้อมูลเกี่ยวกับสภาพแวดล้อมโดยรอบ ใช้ในการหลีกเลี่ยงอุปสรรคและการทำแผนที่จากรูปที่ แสดงการทำงานของเซ็นเซอร์ LiDAR โดยเซ็นเซอร์ให้ข้อมูลนำออกเป็นค่าระยะห่างกับสภาพแวดล้อมทางกายภาพ แสดงเป็นแผนภาพ



รูปที่ 2.8 การทำงานของเซนเซอร์ LiDAR

(ที่มา : https://www.artc.org.tw/upfiles/ADUupload/knowledge/tw_knowledge_408149883.pdf,

2555)

3) กล้อง

Chan Wei Hsu et ls. (2012) ในการใช้ประโยชน์ การทำงานของกล้อง ช่วยในการแสดงผล สภาพแวดล้อมรอบๆ ของพื้นที่การใช้งาน จากนั้นส่งข้อมูลที่ได้ สู่การวางแผนในระบบ เพื่อให้ภาพที่ผ่านการประมวลผลเกิดขึ้น นอกจากนั้นยังนำมาพิจารณาและเชื่อมโยงกับการตรวจจับมุมของ LiDAR โดยประมาณ

4) อุปกรณ์ควบคุมการทำงาน (actuator devices)

Chan Wei Hsu et ls. (2012) อุปกรณ์ขับเคลื่อนแบ่งออกเป็น 3 ส่วน คือ ลื่นเร่ง, พวงมาลัยและการประเมิณความเร็วล้อ ลื่นเร่งจะถูกควบคุมด้วยเครื่องยนต์สูญญากาศ (engine vacuum) เพื่อปรับความดันของโพรงเครื่องยนต์ให้สัมพันธ์กับข้อมูลที่ได้จากเซนเซอร์ต่าง ๆ และการเปลี่ยนแปลงมุมมองของพวงมาลัยทำได้โดยการควบคุมมอเตอร์

บทที่ 3

วิธีการดำเนินงาน

3.1. ศึกษาข้อมูลที่เกี่ยวข้อง

ศึกษาและค้นหาข้อมูลเกี่ยวกับอัลกอริทึมที่ใช้สำหรับการตรวจจับยานพาหนะ เพื่อค้นหาวิธีการเชื่อมต่อระหว่างเซนเซอร์ลิดาร์กับโปรแกรม Matlab เนื่องจากคณะผู้จัดทำได้ใช้โปรแกรม Matlab ในการพัฒนาอัลกอริทึม รวมทั้งศึกษาหาข้อมูลและสร้างความสัมพันธ์ระหว่างกล้องและลิดาร์เพื่อสอบเทียบเครื่องมือวัด(Calibrate) หากเมื่อสามารถสอบเทียบได้สำเร็จนั้นจะสามารถแสดงระยะห่างจริงที่เกิดขึ้นระหว่างเซนเซอร์ลิดาร์และยานพาหนะได้ เพื่อนำไปพัฒนาอัลกอริทึมสำหรับระบบขับเคลื่อนอัตโนมัติแบบแปรผันต่อไป

3.2 ตรวจจับยานพาหนะ (Vehicle Detection)

เนื่องจากปัจจุบันมีความก้าวหน้าทางเทคโนโลยีที่สามารถส่งเสริมและเป็นอุปกรณ์เพื่อพัฒนางานวิจัยต่าง ๆ จากในอดีตให้สามารถเกิดขึ้นได้จริง ยกตัวอย่างเช่นการพัฒนาเติบโตของอัลกอริทึมการตรวจจับวัตถุ ในประวัติศาสตร์ เนื่องจากเครื่องมือที่ใช้ในงานประเภทนี้ยังไม่ได้มีความพร้อมมากนักเมื่อเปรียบเทียบกับปัจจุบันที่ความก้าวหน้าของเทคโนโลยีสามารถสร้างสรรค์ให้เกิดการพัฒนาสิ่งต่าง ๆ ที่ก่อให้เกิดประโยชน์ ความน่าสนใจหรือเพียงเพิ่มความสะดวกสบายในการใช้ชีวิตประจำวัน

โมเดลที่ใช้สำหรับการตรวจจับวัตถุในปัจจุบันหลากหลายโมเดลที่มีความน่าสนใจนั้นก็ใช้ซอฟต์แวร์การเรียนรู้เชิงลึกเพื่อใช้ในการพัฒนา เช่น โครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค , YOLOv2 , MobileNet , Robot Surgery Segmentation และ 3D Face Reconstruction from a Single Image เป็นต้น ซึ่งทางคณะผู้จัดทำได้ตัดสินใจใช้โมเดลโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค และโมเดล YOLOv2 ในการพัฒนาอัลกอริทึมในโครงการฉบับนี้ ทั้งนี้เอกสารออนไลน์ของโปรแกรมนั้นมีการฝึกฝนโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาคเพื่อตรวจจับรถยนต์แบบสำเร็จรูปให้สามารถเรียกใช้งานได้ หรือในอีกทางหนึ่งคือการฝึกฝนโมเดลด้วยตนเอง โดยในหัวถัดไปคณะผู้จัดทำได้มีความตั้งใจในการนำเสนอการฝึกฝนโมเดลด้วยตนเองอย่างละเอียด

ขั้นตอนการฝึกฝนโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค สามารถฝึกฝนได้โดยปฏิบัติตามตัวอย่างการฝึกฝนของเอกสารออนไลน์จากบริษัท MatWorks [13] มีขั้นตอนดังรูปภาพที่ 3.2 ถึง รูปภาพที่ 3.7 ดังต่อไปนี้

```
doTrainingAndEval = false;
if ~doTrainingAndEval && ~exist('fasterRCNNResNet50VehicleExample.mat','file')
    % Download pretrained detector.
    disp('Downloading pretrained detector (118 MB)...');
    pretrainedURL =
'https://www.mathworks.com/supportfiles/vision/data/fasterRCNNResNet50VehicleExample.mat';
    websave('fasterRCNNResNet50VehicleExample.mat', pretrainedURL);
end
```

รูปที่ 3.1 โค้ดดาวน์โหลดโปรแกรมที่ผ่านการฝึกฝนการตรวจจับ

รูปภาพที่ 3.1 การเรียกดาวน์โหลดตัวอย่างการฝึกฝนโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาคให้ตรวจจับยานพาหนะ หากแหล่งเก็บข้อมูล (Folder) ที่ใช้สำหรับการทำงาน ไม่มีไฟล์นามสกุล mat (.mat) ที่อยู่ในรูปแบบการเก็บข้อมูลไบนารีที่ใช้โปรแกรม MATLAB ขยาย ใช้สำหรับการการฝึกฝนโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค โค้ดคำสั่งให้ดาวน์โหลดข้อมูลมา จากนั้นจึงเป็นส่วนของการเตรียมข้อมูล ดังนี้

```
% Unzip vehicle dataset images.
unzip vehicleDatasetImages.zip

% Load vehicle dataset ground truth.

data = load('vehicleDatasetGroundTruth.mat');

vehicleDataset = data.vehicleDataset;
```

รูปที่ 3.2 โค้ดแยกไฟล์ข้อมูลรูปภาพยานพาหนะและไบนารีชุดข้อมูลยานพาหนะจริง

รูปภาพที่ 3.2 โปรแกรมให้ทำการแยกไฟล์บีบอัดข้อมูลรูปภาพยานพาหนะที่ถูกบีบอัดรวมไว้เป็นกลุ่มเดียวกันให้กระจายออกมาแยกแต่ละไฟล์ จากนั้นจึงดาวน์โหลดไบนารีชุดข้อมูลยานพาหนะในความเป็นจริง

```
% Display first few rows of the data set.  
vehicleDataset(1:4,:)
```

รูปที่ 3.3 โค้ดบรรจุข้อมูลในตาราง

รูปภาพที่ 3.3 ข้อมูลยานพาหนะในความเป็นจริงจะถูกเก็บให้อยู่ในตาราง โดยคอลัมภ์แรกคือส่วนของไฟล์ภาพ และคอลัมภ์อื่นๆคือป้ายสำหรับยานพาหนะ

```
vehicleDataset.imageFilename = fullfile(pwd, vehicleDataset.imageFilename);  
  
% Read one of the images.  
I = imread(vehicleDataset.imageFilename{10});  
  
% Insert the ROI labels.  
I = insertShape(I, 'Rectangle', vehicleDataset.vehicle{10});  
  
% Resize and display image.  
I = imresize(I,3);  
  
figure  
imshow(I)
```

รูปที่ 3.4 โค้ดแสดงภาพและทำความเข้าใจกับประเภทของภาพที่มีอยู่

รูปภาพที่ 3.4 เพิ่มข้อมูลฉบับเต็มลงไปไฟล์เตอร์ข้อมูลยานพาหนะ จากนั้นอ่านข้อมูลรูปภาพเพื่อเพิ่มป้าย ROI และขนาดของภาพ ขั้นตอนนี้เพื่อแสดงหนึ่งในภาพจากชุดข้อมูลเพื่อทำความเข้าใจกับประเภทของภาพที่มีอยู่

```

% Set random seed to ensure example training reproducibility.

rng(0);

% Randomly split data into a training and test set.

shuffledIdx = randperm(height(vehicleDataset));

idx = floor(0.6 * height(vehicleDataset));

trainingData = vehicleDataset(shuffledIdx(1:idx),:);

testData = vehicleDataset(shuffledIdx(idx+1:end),:);

```

รูปที่ 3.5 โค้ดแบ่งข้อมูลเพื่อนำไปฝึกฝนและประเมิน

รูปภาพที่ 3.5 แบ่งชุดข้อมูลเพื่อนำไปเข้าสู่การฝึกฝนการตรวจจับด้วยชุดข้อมูลที่ใช้ฝึกฝนและชุดทดสอบเพื่อประเมินการตรวจจับ โดยใช้ข้อมูลจำนวนร้อยละ 60 เพื่อนำไปฝึกฝนและข้อมูลอีกร้อยละ 40 หรือข้อมูลที่เหลือใช้สำหรับการประเมิน

```

% Options for step 1.

options = trainingOptions('sgdm', ...

'MaxEpochs', 5, ...

'MiniBatchSize', 1, ...

'InitialLearnRate', 1e-3, ...

'CheckpointPath', tempdir);

```

รูปที่ 3.6 โค้ดกำหนดค่าตัวเลือกสำหรับการฝึกฝน

รูปภาพที่ 3.6 ทำการกำหนดค่าตัวเลือกของการฝึกฝน หลังจากขั้นตอนนี้จึงสามารถฝึกฝนโมเดลด้วยโปรแกรมดังนี้

```

if doTrainingAndEval

    % Train Faster R-CNN detector.

    % * Use 'resnet50' as the feature extraction network.

    % * Adjust the NegativeOverlapRange and PositiveOverlapRange to ensure
    %   training samples tightly overlap with ground truth.

    [detector, info] = trainFasterRCNNObjectDetector(trainingData, 'resnet50', options, ...
        'NegativeOverlapRange', [0 0.3], ...
        'PositiveOverlapRange', [0.6 1]);

else

    % Load pretrained detector for the example.

    pretrained = load('fasterRCNNResNet50VehicleExample.mat');

    detector = pretrained.detector;

end

```

รูปที่ 3.7 โค้ดการฝึกฝนโมเดล

รูปภาพที่ 3.7 โปรแกรมนี้ใช้เพื่อการฝึกฝนโมเดลโดยใช้ Resnet50 เช่นเดียวกับเครือข่ายการแยกคุณสมบัติ และปรับช่วงการซ้อนทับเชิงลบและการซ้อนทับเชิงบวกเพื่อให้แน่ใจว่าตัวอย่างการฝึกอบรมจะทับซ้อนใกล้เคียงกับความเป็นจริง

ผลลัพธ์หลังจากการฝึกฝนโมเดลสำเร็จจะได้ชุดข้อมูลที่ฝึกฝนเรียบร้อยแล้ว สามารถนำไปใช้ในการตรวจจับยานพาหนะได้ผลลัพธ์ดังที่กล่าวไว้ในเนื้อหาถัดไป และในส่วนของขั้นตอนการฝึกฝน YOLOv2 ให้ตรวจจับยานพาหนะสามารถฝึกฝนได้โดยปฏิบัติตามตัวอย่างการฝึกฝนของเอกสารออนไลน์ของบริษัท MatWorks [14] ได้เช่นกัน

หลังจากที่ฝึกฝนโมเดลสำเร็จ เราจะสามารถนำโมเดลการตรวจจับยานพาหนะมาใช้ในการตรวจจับยานพาหนะโดยใช้โปรแกรมได้ดังนี้ โดยจะกล่าวถึงการตรวจจับยานพาหนะโดยโมเดลโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาคก่อน ดังเนื้อหาที่ปรากฏต่อไปนี้

```

clear

camera = webcam;

fasterRCNN = vehicleDetectorFasterRCNN('full-view');

while true

picture = camera.snapshot;

[boxes,scores] = detect(fasterRCNN,picture);

if    not(isempty(boxes)) && not(isempty(scores))

    I = insertObjectAnnotation(picture,'rectangle',boxes,scores);

else

    I = picture;

end

image(I);

end

```

รูปที่ 3.8 โค้ดตรวจจับยานพาหนะและสร้างกล่องขอบเขตยานพาหนะ

รูปภาพที่ 3.8 โปรแกรมสั่งให้ล้างข้อมูลตัวแปรที่เก็บค่าไว้ในหน้าต่าง Workspace ของโปรแกรม จากนั้นจะทำการเรียกใช้กล่องเว็บแคมและเรียกใช้ฟังก์ชันการตรวจจับรถยนต์ของโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค ณ ที่นี้คณะผู้จัดทำได้เรียกใช้ฟังก์ชันสำเร็จรูป หากต้องการใช้โมเดลที่ฝึกฝนสามารถเปลี่ยนโค้ดเพื่อเรียกใช้ได้ จากนั้นสร้างลูปรูปการทำซ้ำของกระบวนการต่อไป เริ่มจากการจับภาพครั้งหนึ่งเพื่อตรวจหารถยนต์ภายในภาพและสร้างกล่องขอบเขต (Bounding Box) ให้กับรถยนต์ที่สามารถตรวจจับได้ พร้อมการให้คะแนนจากนั้นจึงแสดงผล หากสามารถตรวจจับรถยนต์ภายในภาพได้ จะปรากฏกรอบสี่เหลี่ยมขึ้นมาล้อมรอบบริเวณที่มีรถยนต์ หากไม่สามารถตรวจจับรถยนต์ได้กล่าวคือไม่ปรากฏกรอบสี่เหลี่ยมนั้น โค้ดคำสั่งให้แสดงภาพที่ป้อนข้อมูลเข้ามาแสดงผลแทน

ผู้จัดทำได้จัดทำการศึกษาทดลองโค้ดคำสั่งการตรวจจับยานพาหนะโดยใช้โมเดลโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค โดยการนำภาพรถยนต์และภาพเคลื่อนไหวบนถนนที่ประกอบด้วยรถยนต์ เพื่อสังเกตผลลัพธ์ที่ได้โมเดล



รูปที่ 3.9 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ของโมเดลโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค

การนำเข้ารูปภาพที่ต้องการตรวจจับรูปที่ 3.9 (ก) เมื่อโปรแกรมสามารถตรวจจับยานพาหนะได้ จะสร้างกล่องขอบเขตตั้งรหัสคำสั่งที่กล่าวมาข้างต้น แสดงผลลัพธ์ดัง 3.9 (ข)

และรหัสคำสั่งการตรวจจับยานพาหนะของโมเดล YOLOv2 ที่ทำการฝึกฝนมาใช้ในการตรวจจับยานพาหนะโดยใช้รหัสในโปรแกรมได้ดังนี้

```

clear

camera = webcam;

vehicleDetector = load('yolov2VehicleDetector.mat','detector');

detector = vehicleDetector.detector;

while true

picture = camera.snapshot;

[bboxes,scores] = detect(detector,picture);

if    not(isempty(bboxes)) && not(isempty(scores))

    I = insertObjectAnnotation(picture,'rectangle',bboxes,scores);

else

    I = picture;

end

image(I);

end

```

รูปที่ 3.10 โค้ดตรวจจับยานพาหนะและสร้างกล่องขอบเขตยานพาหนะ

รูปที่ 3.10 รหัสจะสั่งให้ล้างข้อมูลตัวแปรที่เก็บค่าไว้ในหน้าต่าง Workspace ของโปรแกรม จากนั้นจะทำการเรียกใช้กล้องเว็บแคมและดาวน์โหลดข้อมูลไบนารีการตรวจจับยานพาหนะ YOLOv2 จากนั้นสร้างรูปการทำซ้ำของกระบวนการต่อไปนี เริ่มจากการจับภาพครั้งหนึ่งเพื่อตรวจหารถยนต์ภายในภาพและสร้างกล่องขอบเขต (Bounding Box) ให้กับรถยนต์ที่สามารถตรวจจับได้ พร้อมการให้คะแนน จากนั้นจึงแสดงผล หากสามารถตรวจจับรถยนต์ภายในภาพได้จะปรากฏกรอบสี่เหลี่ยมขึ้นมาล้อมรอบบริเวณที่มีรถยนต์ หากไม่สามารถตรวจจับรถยนต์ได้กล่าวคือไม่ปรากฏกรอบสี่เหลี่ยมนั้น รหัสจะสั่งให้แสดงภาพที่ป้อนข้อมูลเข้ามาแสดงผลแทน

ผู้จัดทำได้จัดทำทดลองโดยใช้รหัสคำสั่งการตรวจจับยานพาหนะโดยใช้โมเดล YOLOv2 ด้วยการนำภาพรถยนต์และภาพเคลื่อนไหวบนถนนที่ประกอบด้วยรถยนต์ในการทดลอง ดังนี้



(ก)

(ข)

รูปที่ 3.11 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ของโมเดลYOLOv2

การนำเข้ารูปภาพที่ต้องการตรวจจับรูปที่ 3.11 (ก) เมื่อรหัสคำสั่งสามารถตรวจจับยานพาหนะได้ จะสร้างกล่องขอบเขตตั้งรหัสคำสั่งที่กล่าวมาข้างต้น แสดงผลลัพธ์ดังรูปที่ 3.11 (ข)

ทั้งนี้ผลลัพธ์ระหว่างการใช้รหัสคำสั่งโมเดลโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาค และโมเดล YOLOv2 ในการตรวจจับยานพาหนะนั้น ด้านความสามารถในการตรวจจับรถยนต์นั้นมีความแตกต่างกัน คือ โมเดลโครงข่ายประสาทพัฒนาแบบคอนโวลูชันตามภูมิภาคนั้นสามารถตรวจจับได้มีคุณภาพมากกว่าโมเดล YOLOv2 สามารถแก้ไขได้โดยการเพิ่มข้อมูลในการฝึกฝนโมเดล แต่ด้านความไวในการตรวจจับนั้นโมเดล YOLOv2 สามารถทำงานได้ประสิทธิภาพกว่ามาก เนื่องจากผลลัพธ์จากการตรวจจับยานพาหนะนั้นนำไปใช้งานในขั้นตอนต่อไป ซึ่งเป็นการใช้ข้อมูลเพื่อการประมวลผลแบบต่อเนื่อง คณะผู้จัดทำจึงตัดสินใจในการใช้โมเดล YOLOv2 ในการตรวจจับยานพาหนะของโครงการฉบับนี้

3.3 สร้างจุดศูนย์กลางของรถยนต์ภายในภาพ (Centroid)

จากความสามารถตรวจจับยานพาหนะภายในภาพและสร้างกล่องขอบเขตวัตถุ (Bounding Box) หรือกรอบสี่เหลี่ยมล้อมรอบยานพาหนะ กล่าวคือสามารถระบุตำแหน่งของยานพาหนะภายในภาพได้ ส่งผลให้สามารถหาจุดศูนย์กลางซึ่งมีลักษณะข้อมูลเป็นคู่อันดับของพิกเซลภาพของกล่องขอบเขตที่เกิดขึ้นได้เป็นลำดับถัดมา ซึ่งจุดศูนย์กลางภาพนั้นเปรียบเสมือนจุดของภาพที่เป็นความกว้างเฉลี่ยของรถยนต์ที่ระนาบความสูงจากเซนเซอร์ลิดาร์ เพื่อนำจุดศูนย์กลางภาพนี้สู่การคำนวณระยะห่างในเนื้อหาที่จะกล่าวในหัวข้อต่อไป ซึ่งการคำนวณหาจุดศูนย์กลางของกล่องขอบเขตนั้นสามารถเขียนรหัสคำสั่งเสริมจากรหัสคำสั่งในการตรวจจับยานพาหนะและสร้างกล่องขอบเขตในหัวข้อที่กล่าวมาแล้วได้ ด้วยรหัสคำสั่งดังนี้

```
vehicleDetector = load('yolov2/vehicleDetector.mat','detector');  
detector = vehicleDetector.detector;  
[bboxes,scores] = detect(detector,picture);  
  
l = insertObjectAnnotation(picture,'rectangle',bboxes,scores);  
centers = bboxes(1:2) + (bboxes(3:4))/2;  
  
l = insertMarker(l,centers,'o');
```

รูปที่ 3.12 โค้ดสร้างจุดศูนย์กลางของรถยนต์ภายในภาพ

กล่องขอบเขตจากโค้ดคำสั่งรูปที่ 3.11 สามารถนำมาหาจุดกึ่งกลางภาพโดยใช้คำสั่งดังรูปที่ 3.12 โดย bboxes จะเก็บข้อมูลเป็นเมตริกขนาด 1x4 ซึ่งข้อมูลในคอลัมภ์แรกคือ ระยะห่างจากขอบภาพถึงขอบกล่องขอบเขตในระนาบแกน x คอลัมภ์สองคือระยะห่างจากขอบภาพถึงขอบกล่องขอบเขตในระนาบแกน y คอลัมภ์สามคือความสูงของกล่องขอบเขต และ คอลัมภ์สี่คือความกว้างของกล่องขอบเขต เมื่อนำข้อมูลในคอลัมภ์ที่ 1 รวมกับครึ่งหนึ่งของข้อมูลในคอลัมภ์ที่ 3 และข้อมูลในคอลัมภ์ที่ 2 รวมกับครึ่งหนึ่งของข้อมูลในคอลัมภ์ที่ 4 จะได้คู่อันดับพิกเซลของจุดกึ่งกลางกล่องขอบเขต



(ก)



(ข)

รูปที่ 3.13 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) สร้างจุดศูนย์กลางของรถยนต์ภายในภาพ

ตัวอย่างการทำงานของรหัสคำสั่ง รูปที่ 3.13 (ก) การนำเข้ารูปภาพที่ต้องการตรวจจับ เมื่อรหัสคำสั่งสามารถตรวจจับยานพาหนะได้จะแสดงผลดัง รูปที่ 3.13 (ข) จุดสีเขียวคือข้อมูลคู่อันดับของพิกเซลจุดศูนย์กลางภาพนั่นเอง

ซึ่งข้อมูลคู่อันดับของพิกเซลจุดศูนย์กลางภาพนี้ สามารถนำไปใช้ในการคำนวณมุมเสมือนของเซนเซอร์ลิตาร์ในภาพที่ได้จากกล้องแวปแคม เพื่อนำไปเปรียบเทียบมุมจริงของเซนเซอร์ลิตาร์ต่อไป ข้อมูลทั้งหลายเหล่านี้จะสามารถนำไปหาระยะห่างระหว่างยานพาหนะที่ตรวจเจอบนภาพกับเซนเซอร์ลิตาร์ได้

3.4 การหาความสัมพันธ์ระหว่างมุมของวัตถุในภาพกับเซนเซอร์ลิตาร์

ในการนำข้อมูลระยะห่างจากเซนเซอร์ลิตาร์มาใช้จำเป็นต้องทำการสอบเทียบ(Calibrate) อธิบายได้ว่าการหาความสัมพันธ์ที่เกิดขึ้นระหว่างมุมวัตถุภายในภาพที่ได้จากกล้องแวปแคมและมุมของเซนเซอร์ลิตาร์ที่เก็บข้อมูลในสภาพแวดล้อมเดียวกัน เพื่อนำค่าผลลัพธ์ที่ได้นั้นมาเปรียบเทียบกับมีความสัมพันธ์กันเป็นสมการใด จากนั้นจึงจะสามารถบอกระยะห่างจริงของวัตถุในภาพว่ามีระยะห่างเป็นเท่าไรจากเซนเซอร์ลิตาร์



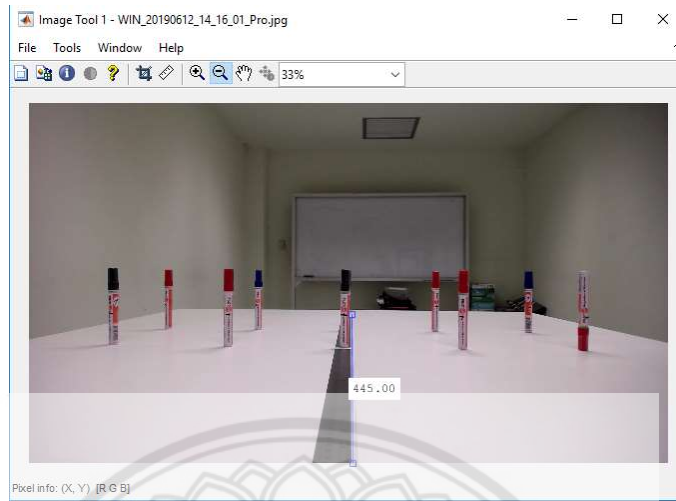
รูปที่ 3.14 กล้องเว็บคาเมราและเซนเซอร์ลิดาร์ที่ยึดติดกัน

นำกล้องและเซนเซอร์ลิดาร์ยึดติดกันเพื่อการสร้างสมการความความสัมพันธ์และสอบเทียบที่ถูกต้อง ดังรูปที่ 3.14



รูปที่ 3.15 เตรียมวัตถุสำหรับเก็บข้อมูล

ทำการเก็บข้อมูลโดยนำวัตถุวางไว้ที่ตรงกลางของภาพโดยมีระยะห่างจากเซนเซอร์ลิดาร์ 60 เซนติเมตร จากนั้นจึงวางต่อกันตามแถวกว้างและแถวลึกห่างกันจุดละ 20 เซนติเมตร ดังรูปที่ 3.15



รูปที่ 3.16 วัดพิกลเซลภาพระยะ a

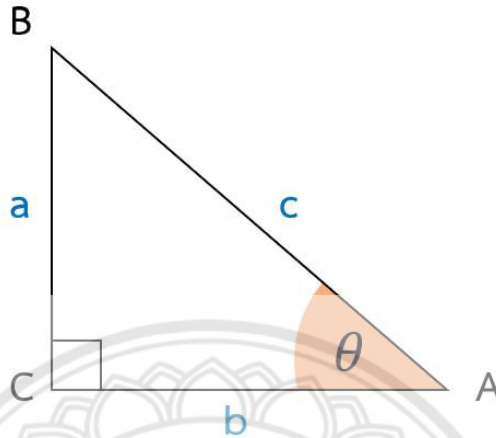
เก็บข้อมูลโดยใช้รหัสคำสั่ง `imtool()` ในโปรแกรม Matlab เพื่อวัดระยะพิกลเซลภายในภาพ
 ด้านล่างขอบภาพจนถึงระนาบการตรวจจับของเซนเซอร์ลิตาร์ดบนวัตถุบริเวณกลางภาพที่กึ่งกลางของภาพ
 ดังรูปที่ 3.16



รูปที่ 3.17 วัดพิกลเซลภาพระยะ b

เก็บข้อมูลโดยใช้รหัสคำสั่ง `imtool()` ในโปรแกรม Matlab เพื่อวัดระยะพิกลเซลระหว่างวัตถุ
 บริเวณกลางภาพด้านหน้าสุดและด้ามอื่นๆด้านหน้าบนระนาบความสูงที่เซนเซอร์ลิตาร์ดตรวจจับดังภาพรูป
 ที่ 3.17 จะได้ข้อมูลทั้งหมดจำนวน 5 ข้อมูลโดยเริ่มจากวัตถุด้านขวาจนถึงวัตถุด้านซ้าย

นำระยะที่เก็บข้อมูลมาคำนวณหามุมพิกเซลภายในภาพโดยอาศัยความรู้เรื่องตรีโกณมิติ



รูปที่ 3.18 สามเหลี่ยมมุมฉากตรีโกณมิติ

อัตราส่วนตรีโกณมิติ

$$\theta \text{ (rad)} = \tan^{-1}\left(\frac{a}{b}\right) \quad (3.1)$$

- เมื่อ a คือ ความยาวด้านตรงกันข้ามมุม
b คือ ความยาวด้านประชิดมุม

ตารางที่ 3.1 แสดงข้อมูลระยะห่างพิกเซลภายในภาพและข้อมูลมุมจากลิตาร์ในสภาพแวดล้อมเดียวกัน

Object	a	b	Anglmg(rad)	AngLi(rad)
1	445	706	1.009	0.60105
2	445	343	0.6566	0.33952
3	445	0	0	0.05209
4	445	360	-0.682	-0.2387
5	445	711	-1.012	-0.5089

จากตารางที่ 3.1 คอลัมป์ a เป็นตัวแปรของข้อมูลระยะพิกเซลระหว่างด้านล่างขอบภาพจนถึงระนาบการตรวจจับของเซนเซอร์ลิตาร์บนวัตถุบริเวณกลางภาพ คอลัมป์ b เป็นตัวแปรของข้อมูลระยะพิกเซลระหว่างวัตถุบริเวณกลางภาพด้านหน้าสุดถึงวัตถุติดกันด้านหน้าบนระนาบความสูงการตรวจจับ

ของเซนเซอร์ลิตาร์ จากอัตราส่วนตรีโกณมิติทำให้ทราบมุมพิกเซลภายในภาพโดยใช้ความรู้อัตราส่วนตรีโกณมิติดังสมการ 3.1 ผลลัพธ์ดังคอลัมภ์ที่ Anglmg หน่วยคือ rad และคอลัมภ์ AngLi แสดงข้อมูลมุมของวัตถุทั้ง 5 ที่เซนเซอร์สามารถตรวจจับได้ ทำให้เมื่อนำขนาดมุมพิกเซลภายในภาพที่ได้จากการคำนวณเปรียบเทียบกับข้อมูลระยะห่างจากเซนเซอร์ลิตาร์ที่เก็บข้อมูลในสภาพแวดล้อมเดียวกันจะได้ความสัมพันธ์ที่เกิดขึ้น แสดงดังสมการ 3.2

$$y = -0.0317x^4 + 0.2046x^3 + 0.0279x^2 + 0.3402x + 0.0521 \quad (3.2)$$

เมื่อ x = ขนาดมุมพิกเซลในภาพ

y = ขนาดมุมเสมือนเซนเซอร์ลิตาร์

หากนำขนาดมุมเสมือนเซนเซอร์ลิตาร์เปรียบเทียบกับมุมจริงของเซนเซอร์ลิตาร์ จะทำให้ทราบระยะห่างระหว่างยานพาหนะและเซนเซอร์ลิตาร์ซึ่งเปรียบเสมือนระยะห่างจริงได้

3.5 การหาระยะห่างจริงจากมุมเสมือนเซนเซอร์ลิตาร์

นำขนาดมุมเสมือนเซนเซอร์ลิตาร์จากสมการที่ 3.2 เปรียบเทียบกับข้อมูลขนาดมุมที่ได้จากการเก็บข้อมูลของเซนเซอร์ลิตาร์โดย หากข้อมูลทั้งสองมีขนาดมุมใกล้เคียงกัน จะทำให้ทราบค่าระยะห่างของวัตถุที่อยู่ในมุมนั้น ๆ ของเซนเซอร์ สามารถเขียนอัลกอริทึมได้ดังนี้

```

anglmg = atan((centers(1)-960)/(centers(2)-1080));
angli = -0.0295*anglmg^4 + 0.2033*anglmg^3 + 0.0255*anglmg^2 + 0.3417*anglmg + 0.0521;
for i = 1:360
    if allangles(i) - 0.02 <= imageAngle && imageAngle <= allangles(i) + 0.02
        end
    end
if i >= 360
    count = 0; alldistances = []; allangles = [];
end
count = count+1;

```

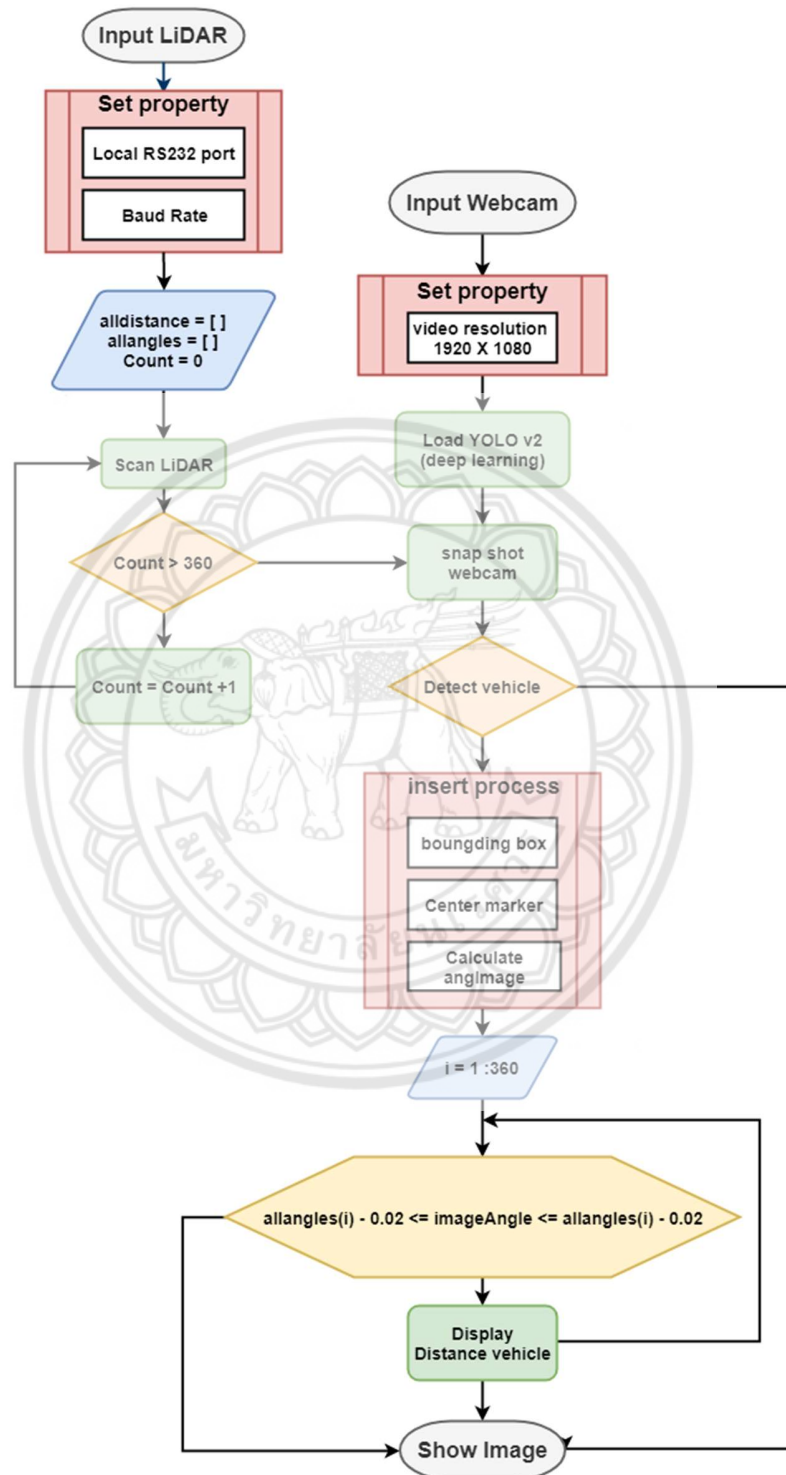
รูปที่ 3.19 โค้ดแสดงระยะห่างจริงจากมุมเสมือนเซนเซอร์ลิตาร์

จากรูปที่ 3.19 รับค่ามุมพิกเซลในภาพและคำนวณขนาดมุมเสมือนเซนเซอร์ลิตาร์ จากนั้นเริ่มลูป (loop) การทำซ้ำรับข้อมูล 360 ครั้ง กล่าวคือข้อมูลจากเซนเซอร์ลิตาร์ที่ทำงานครบรอบ 360 องศา หากข้อมูลในชุดข้อมูลนี้มีมุมใกล้เคียงกับขนาดมุมเสมือนเซนเซอร์ลิตาร์ข้างต้น คำสั่งให้แสดงค่าระยะห่างที่ขนาดมุนั้นๆ โดยคณะผู้จัดทำได้กำหนดผลต่างของข้อมูลที่จัดให้นำมาแสดงที่ 0.02 เรเดียนหรือ 1.15 องศา

3.6 แผนภาพแสดงลำดับขั้นตอนการทำงานของ Algorithm (Flowchart Alogorithm)

แผนภาพแสดงทำงานของอัลกอริทึมการตรวจจับยานพาหนะพร้อมระยะห่างโดยนำได้คำสั่งจากหัวข้อ การตรวจจับยานพาหนะ การสร้างกล่องขอบเขตล้อมรอบยานพาหนะ การหาความสัมพันธ์ระหว่างขนาดมุมพิกเซลภาพและขนาดมุมจากเซนเซอร์ลิตาร์ มาสร้างได้คำสั่งการใช้งานร่วมกัน ขั้นตอนวิธีการประมวลผลข้อมูล และแสดงผลลัพธ์ระยะของยานพาหนะที่เกิดขึ้นบนภาพจากกล้องเว็บแคมเราได้





รูปที่ 3.20 แผนผังการทำงานของอัลกอริทึม

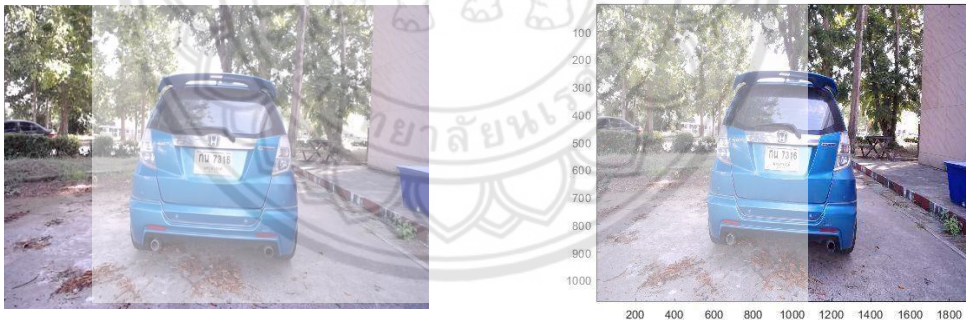
บทที่ 4

ผลการทดลองและวิเคราะห์ผลการทดลอง

4.1 ผลการทดลอง

นำกล้องและเซนเซอร์ลิดาร์เก็บข้อมูลยานพาหนะที่ระยะห่างแตกต่างกันเพื่อใช้อัลกอริทึมหาระยะห่างที่เกิดขึ้น และเปรียบเทียบขนาดระยะห่างที่ได้จาก algorithm กับระยะห่างที่เกิดขึ้นจริงของยานพาหนะ ณ ระยะต่างๆ ที่ความสูงของเซนเซอร์ลิดาร์เท่ากับ 96 ซม.

ณ ระยะห่าง 200 เซนติเมตร



(ก)

(ข)

รูปที่ 4.1 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 200 เซนติเมตร

ที่ระยะห่าง 200 เซนติเมตร พบว่าโมเดลตรวจจับยานพาหนะไม่สามารถตรวจจับได้ คาดคะเนว่าอาจเป็นเพราะระยะที่ภาพที่ใกล้ ทำให้ไม่สามารถตรวจจับได้ ดังรูปที่ 4.1

ณ ระยะห่าง 408 เซนติเมตร



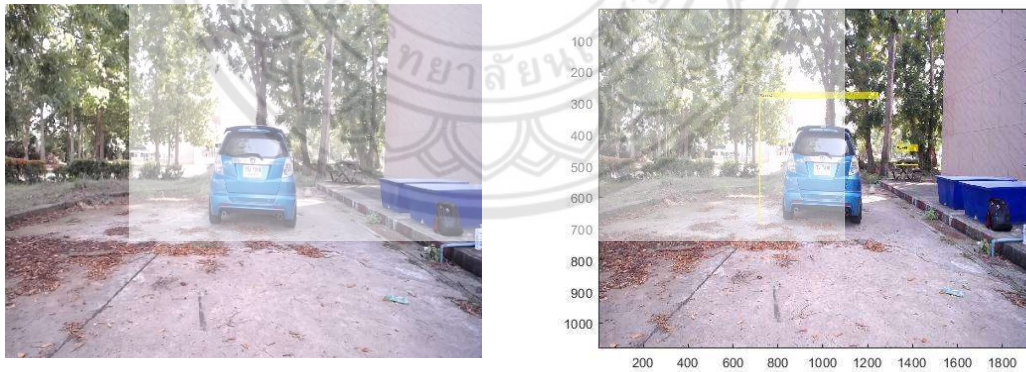
(ก)

(ข)

รูปที่ 4.2 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 408 เซนติเมตร

ที่ระยะห่าง 400 เซนติเมตร พบว่าโมเดลตรวจจับยานพาหนะสามารถตรวจจับได้ที่คะแนน 0.9857 โดยแสดงผลระยะที่ได้จากอัลกอริทึมที่ 418.05 ถึง 419.22 แสดงดังรูปที่ 4.2

ณ ระยะห่าง 525 เซนติเมตร



(ก)

(ข)

รูปที่ 4.3 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 525 เซนติเมตร

ที่ระยะห่าง 525 เซนติเมตร พบว่าโมเดลตรวจจับยานพาหนะสามารถตรวจจับได้ที่คะแนน 0.9996 โดยแสดงผลระยะที่ได้จากอัลกอริทึมที่ 520.10 ถึง 531.20 แสดง ดังรูปที่ 4.3

ณ ระยะห่าง 862 เซนติเมตร



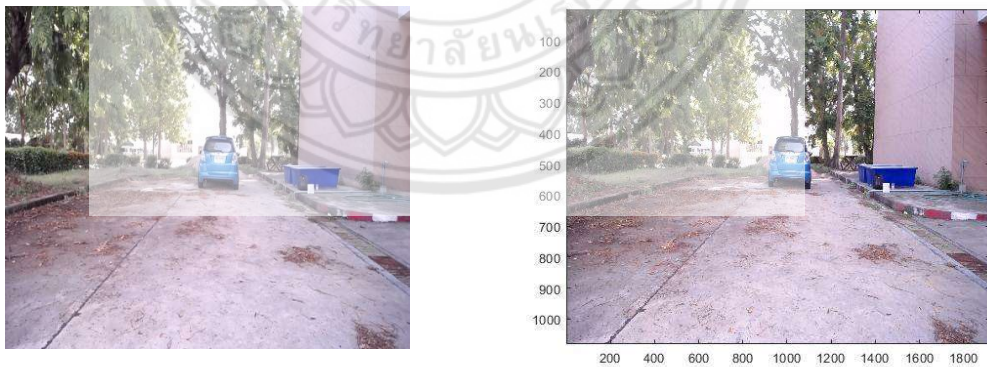
(ก)

(ข)

รูปที่ 4.4 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 862 เซนติเมตร

ที่ระยะห่าง 862 เซนติเมตร พบว่าโมเดลตรวจจบบานพาหนะสามารถตรวจจับได้ที่คะแนน 0.8490 แต่ไม่มีข้อมูลระยะจากเซนเซอร์ลิดาร์เนื่องจากเซนเซอร์ลิดาร์ยิงเลเซอร์สูงๆจะทำให้เกิดการดูดซับลำแสงไป แสดงดังรูปที่ 4.4

ณ ระยะห่าง 1200 เซนติเมตร



(ก)

(ข)

รูปที่ 4.5 ภาพต้นแบบ (ก) และผลการทำงานของโปรแกรม (ข) ที่ระยะห่าง 1200 เซนติเมตร

ที่ระยะห่าง 1200 เซนติเมตร พบว่าโมเดลตรวจจบบานพาหนะไม่สามารถตรวจจับได้ คาดคะเนว่าอาจเป็นเพราะระยะที่ภาพที่ไกล ทำให้ไม่สามารถตรวจจับได้ ดังรูปที่ 4.5

ตารางที่ 4.1 รายละเอียดของระยะห่างของยานพาหนะจากการวัดและผลจากอัลกอริทึม

ระยะห่างของยานพาหนะจากการวัด (เซนติเมตร)	algorithm		
	ระยะห่างของยานพาหนะ (เซนติเมตร)	มุมจากการคำนวณ (rad)	มุมของลิตาร์ (rad)
200	ไม่พบ	-	-
408	418.05 – 419.22	-3.0802	-3.0789 ถึง -3.08
525	520.10 – 531.20	3.124	3.1244 ถึง 3.1149
862	0	-3.0791	-3.0671 ถึง -3.0911
1200	ไม่พบ	-	-

การทดลองอัลกอริทึมพบว่าโมเดลการตรวจจับยานพาหนะสามารถตรวจจับได้ที่ระยะห่าง 408 เซนติเมตร , 525 เซนติเมตร และ 862 เซนติเมตรแต่ไม่สามารถตรวจจับที่ระยะ 200 เซนติเมตรและ 1200 เซนติเมตรได้ ในส่วนของระยะห่างที่แสดงผลนั้น เนื่องจากที่ระยะ 200 เซนติเมตรและ 1200 เซนติเมตรไม่สามารถตรวจจับยานพาหนะได้ทำให้ไม่ปรากฏขนาดของระยะห่างที่ได้จากอัลกอริทึม ในขณะที่ระยะห่าง 408 เซนติเมตร ได้ผลลัพธ์ของอัลกอริทึมที่ขนาด 418.05-418.05 เซนติเมตร เปอร์เซ็นต์ความผิดพลาด (The percent of error) ที่ 0.28676 % ที่ระยะห่าง 525 เซนติเมตร ได้ผลลัพธ์ของอัลกอริทึมที่ขนาด 520.01-531.20 เซนติเมตร เปอร์เซ็นต์ความผิดพลาดที่ 2.13143 % และที่ระยะห่าง 1200 เซนติเมตรไม่สามารถทราบระยะห่างได้เนื่องจากเลเซอร์ยิงสู่กระจกทำให้เกิดการดูดซับลำแสงเกิดขึ้น

4.2 วิเคราะห์ผลการทดลอง

จากผลการทดลอง พบว่าคู่อันดับจุดศูนย์กลางของกล่องขอบเขตที่นำมาแทนค่าในสมการความสัมพันธ์ที่ 3.2 สามารถหาระยะห่างของยานพาหนะในภาพได้ ซึ่งเมื่อสังเกตข้อมูลในตารางผลการทดลองที่ 4.1 จะพบว่า ค่าระยะห่างของยานพาหนะจากอัลกอริทึมมีค่าใกล้เคียงกับระยะที่วัดได้จริง และมุมจากการคำนวณผ่านอัลกอริทึมสามารถนำมาใช้เปรียบเทียบกับมุมที่เก็บจากเซนเซอร์ลิตาร์ได้ จะถือได้ว่าสมการความสัมพันธ์ที่ 3.2 นี้สามารถนำมาใช้หาระยะห่างของยานพาหนะในภาพได้จริง แต่มีข้อจำกัดต่างๆ ประการแรกคือ โมเดลตรวจจับยานพาหนะ ใช้ชุดข้อมูลตัวอย่างในการฝึกฝนโมเดลเป็นชุดข้อมูลที่ไม่ได้เก็บจากสภาพแวดล้อมภายนอกมหาวิทยาลัย ทำให้การตรวจจับไม่มีประสิทธิภาพเท่าที่ควร ประการต่อมา เซนเซอร์ลิตาร์ใช้การวัดระยะวัตถุด้วยการยิงลำแสงเลเซอร์ ที่มีข้อจำกัดดังนี้

1. พื้นที่ผิวมีการหักเหมาก ทำให้ค่าผิดพลาด
2. ในสภาพแวดล้อมที่มีแสงมากจะทำให้ระยะที่วัดได้น้อยกว่า การวัดในที่มืด

ดังนั้น ปริยญาพนธ์เล่มนี้เป็นถือพื้นฐานในการพัฒนาระบบควบคุมความเร็วอัตโนมัติ ซึ่งผลการดำเนินงานเป็นการตรวจจ้ยานพาหนะ และหาความสัมพันธ์ของภาพกับเซนเซอร์ลิตาร์ในการวัดระยะของยานพาหนะ ทั้งนี้อาจต้องพัฒนาด้านโมเดลการตรวจจ้ยานพาหนะ และการออกแบบอัลกอริทึมเพื่อลดข้อจำกัดต่างๆที่ได้กล่าวมาข้างต้น



บทที่ 5

สรุปผลโครงการและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการศึกษาข้อมูลเกี่ยวกับระบบควบคุมความเร็วอัตโนมัติแบบแปรผัน (Adaptive Cruise Control) ระบบขับเคลื่อนอัตโนมัติ (Self-Driving Car) เซนเซอร์ LiDAR และการเรียนรู้เชิงลึก เพื่อนำข้อมูลที่ได้มาจัดทำอัลกอริทึมในโปรแกรมแมทแล็บ (MATLAB) เพื่อตรวจจับและแสดงระยะห่างระหว่างรถยนต์คันข้างหน้ากับขับเคลื่อนอยู่บนท้องถนน โดยใช้ RPLIDAR รุ่น A1M8 ในการตรวจจับ โดยเราสามารถตรวจจับรถยนต์ที่อยู่ในขอบเขตการค้นหาของเซนเซอร์ LiDAR รวมถึงขอบระยะห่างระหว่างรถยนต์และตัวเซนเซอร์ได้ โดยการหาความสัมพันธ์ของขนาดมุมพิกเซลในภาพและขนาดมุมเซนเซอร์ลิดาร์เพื่อแสดงระยะทางจากการวิเคราะห์ภาพได้ และอัลกอริทึมนี้สามารถนำไปพัฒนาเพื่อพัฒนาระบบควบคุมความเร็วอัตโนมัติแบบแปรผันที่จะนำไปใช้ในรถ T-MAC ของศูนย์ได้

จากผลการทดลอง พบว่าสมการความสัมพันธ์ที่ 3.2 สามารถนำมาใช้ได้จากการคำนวณ มาเปรียบเทียบกับข้อมูลของเซนเซอร์ลิดาร์ผ่านอัลกอริทึมได้ ทั้งยังแสดงค่าระยะห่างของยานพาหนะในภาพเป็นไปตามระยะที่วัดจริง ดังตารางผลการทดลองที่ 4.1 แต่โมเดลการตรวจจับรถยนต์ยังมีความผิดพลาด คือประสิทธิภาพในการตรวจจับยังไม่สามารถตรวจจับยานพาหนะได้ทั้งหมด เพราะชุดข้อมูลในการเรียนรู้ไม่ได้มาจากสภาพแวดล้อมที่ใช้ในการทดลอง และเซนเซอร์ลิดาร์มีข้อจำกัดในการใช้งานที่ผิววัตถุ และสภาพแวดล้อมต่างๆ

5.2 ข้อเสนอแนะ

5.2.1 ปัญหาที่พบ

- 1) ประสิทธิภาพของ คอมพิวเตอร์ ที่ใช้ในการประมวลผลนั้นยังมีประสิทธิภาพไม่สูงพอ จึงทำให้เกิดการค้าง หรือล่าช้าในการใช้งาน
- 2) มีปัญหาในการหาข้อมูล และ การ train เนื่องจากตัวโปรแกรม MATLAB นั้นต้องใช้ภาษา c ในการเขียน แต่ข้อมูลที่หาได้นั้นเป็นของ โปรแกรมอื่น
- 3) การกำหนดคุณภาพของกล้องมีผลต่อการทำงานของอัลกอริทึม เพราะในความละเอียดสูง ทำให้ตรวจจับยานพาหนะได้ง่าย แต่มีผลทำให้การตอบสนองของภาพไม่ต่อเนื่องล่าช้า ทำให้ประสิทธิภาพในการใช้งานลดลง
- 4) ในการใช้งานเซนเซอร์ลิตาร์บนโปรแกรม MATLAB ไม่สามารถปรับเร็วในการตรวจจับรับค่าได้ เนื่องจากไม่มีชุดคำสั่งในการแก้ไขคุณสมบัติ ทำให้เซนเซอร์ลิตาร์ทำงานไม่เต็มประสิทธิภาพ ที่ความถี่ 10 Hz ตามอุปกรณ์ระบุไว้
- 5) ข้อจำกัดของเซนเซอร์ลิตาร์ การหักเหบางพื้นผิวของวัตถุทำให้ค่าที่ได้ไม่เป็นจริง และการใช้งานสภาพแวดล้อมที่มีแสงมากค่าที่ได้ จะมีค่าน้อยกว่าในที่มืดที่สภาพแวดล้อมเดียวกัน

5.2.2 ข้อเสนอแนะ

- 1) ควรซื้อเครื่องมือ และอุปกรณ์ที่ใช้สำหรับทดลองที่มีประสิทธิภาพมากกว่านี้ โดยการเพิ่มสเปคของเครื่องคอมพิวเตอร์ ที่ใช้ในการทดสอบ ให้สูงพอที่จะใช้งาน
- 2) ควรศึกษาพื้นฐานการเขียน code และทำความเข้าใจในการเขียน code สำหรับ โปรแกรม MATLAB ให้มากขึ้น โดยเริ่มค้นคว้าให้เข้าใจในระดับหนึ่งก่อนจะเริ่มทำการทดสอบ
- 3) เพิ่มเวลาในการค้นหาข้อมูลให้มากขึ้นรวมถึงศึกษา code ที่ใกล้เคียงและสามารถนำมาใช้ในโปรแกรม MATLAB ได้

บรรณานุกรม

1. Natthawat Phongchit . (ไม่ทราบ) . Object Detection. สืบค้นเมื่อ 30 พฤษภาคม 2562, จาก ; <https://blog.datawow.io/ย้อนรอย-object-detection-และเจาะลึก-retinanet-ea1407ad7325>.
2. mc.ai. 2561. มาลองดูวิธีการคิดของ CNN กัน. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://mc.ai/มาลองดูวิธีการคิดของ-cnn-ก/>
3. IEEE . 2559 . LIDAR (light detection and ranging). สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://www.blognone.com/node/84463>
4. NGThai. 2561. เทคโนโลยีไลดาร์ ช่วยให้ภาพที่ตึกกว่าเดิม. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://ngthai.com/science/7677/lidar-technology-2/>.
5. Comdet Phaudphut . 2561 . AI & Machine Learning 101. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://medium.com/@comdetphaudphut/ai-machine-learning-101-ด้วย-python-ตอนที่-3-ว่าด้วยเรื่องราวของ-ai-f5d2ac4b8554>.
6. nessessence. 2561 . Deep learning. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://www.thaiprogrammer.org/2018/12/deep-learning-คืออะไร/>.
7. Vithan Minaphinant . 2561 . Deep learning. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://blog.finnomena.com/deep-learning-คืออะไร-อาชีพไหนจะตกงานบ้าง-499c250784a1>
8. Jen Namjatturas . 2562. Machine learning&Deep learning. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก; <https://techsauce.co/tech-and-biz/ai-machine-learning-deep-learning-differences/>.
9. Anonymous. 2560. Neural Network. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic> .
10. Sanparith Marukatat . 2561. Machine Learning. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก; <https://medium.com/@sanparithmarukatat/โลกหมุนไป-งานวิจัยก็หมุนตาม-46ae76d4e195>.
11. Thananop Kobchaisawat . 2561 . Object Detection (Part 1). สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://medium.com/@sapjunior/object-detection-part-1-37bf39dc557f>.

12. Thananop Kobchaisawat . 2561 . Object Detection (Part 2). สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://medium.com/@sapjunior/object-detection-part-2-7c41fc11011f>

13. MathWorks. (ไม่ทราบ) . Object Detection Using Faster R-CNN Deep Learning. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://ww2.mathworks.cn/help/vision/examples/object-detection-using-faster-r-cnn-deep-learning.html>.

14. MathWorks. (ไม่ทราบ) . Object Detection Using YOLO v2 Deep Learning. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <https://ww2.mathworks.cn/help/deeplearning/examples/object-detection-using-yolo-v2.html>.

15. SLAMTEC. (ไม่ทราบ). RPLIDAR. สืบค้นเมื่อ 30 พฤษภาคม 2562 , จาก ; <http://www.slamtec.com/en/lidar/a1>

