ระบบการแจ้งเตือนการชนในรถจักรยานยนต์

Motorcycle Warning System

นายณัฏฐนันท์ พุฒซ้อน รหัสนิสิต 60361163

นายณัฏฐพัชร จันทร รหัสนิสิต 60361170

นางสาวอมรรัตน์ สิงหรา รหัสนิสิต 60365710

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2563

มหาวิทยาลัยนเรศวร

ระบบการแจ้งเตือนการชนในรถจักรยานยนต์

Motorcycle Warning System

นายณัฏฐนันท์ พุฒซ้อน รหัสนิสิต 60361163

นายณัฏฐพัชร จันทร รหัสนิสิต 60361170

นางสาวอมรรัตน์ สิงหรา รหัสนิสิต 60365710

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2563

ใบรับรองโครงงาน

หัวข้อโครงงาน : ระบบการแจ้งเตือนการชนในรถจักรยานยนต์

ผู้ดำเนินโครงงาน : นายณัฏฐนันท์ พุฒซ้อน รหัสนิสิต 60361163
นายณัฏฐพัชร จันทร รหัสนิสิต 60361170
นางสาวอมรรัตน์ สิงหรา รหัสนิสิต 60365710

อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ชูพงศ์ ช่วยเพ็ญ

ภาควิชา : วิศวกรรมเครื่องกล

ปีการศึกษา : 2563

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานวิจัยฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล

คณะกรรมการสอบโครงงาน

.....ที่ปรึกษาโครงงาน

(ผู้ช่วยศาสตราจารย์ชูพงศ์ ช่วยเพ็ญ)

.....กรรมการ

(ดร.ปองพันธ์ โอทกานนท์)

.....กรรมการ

(ผศ.ดร.อนันต์ชัย อยู่แก้ว)

หัวข้อโครงงาน : ระบบการแจ้งเตือนการชนในรถจักรยานยนต์
ผู้ดำเนินโครงงาน : นายณัฏฐนันท์ พุฒซ้อน รหัสนิสิต 60361163
นายณัฏฐพัชร จันทร รหัสนิสิต 60361170
นางสาวอมรรัตน์ สิงหรา รหัสนิสิต 60365710
อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ชูพงศ์ ช่วยเพ็ญ
ภาควิชา : วิศวกรรมเครื่องกล
ปีการศึกษา : 2563

บทคัดย่อ

โครงงานฉบับนี้ถูกจัดทำขึ้นเพื่อการศึกษาและสร้างอัลกอริทึมสำหรับระบบการแจ้งเตือนการชนในรถจักรยานยนต์ โดยมีวัตถุประสงค์เพื่อลดปัญหาการเกิดอุบัติเหตุในพื้นที่ชุมชนและบนถนน เพื่อพัฒนาระบบการแจ้งเตือนการชนในรถจักรยานยนต์ และเพื่อวิเคราะห์ระยะห่างวัตถุด้วยเซนเซอร์ lidar ทางคณะผู้จัดทำได้ทำการศึกษาข้อมูลที่มีความเกี่ยวข้องกับระบบควบคุมความเร็วอัตโนมัติ และการทำงานของหุ่นยนต์อัตโนมัติที่มีการใช้ระบบตรวจจับสิ่งกีดขวาง โดยที่มีการทำงานคล้ายคลึงกับระบบควบคุมความเร็วอัตโนมัติ การศึกษาข้อมูลที่ได้กล่าวมานี้สามารถนำไปพัฒนาและประยุกต์ให้เข้ากับการสร้างระบบการแจ้งเตือนการชนในรถจักรยานยนต์ โดยใช้อุปกรณ์ RPLIDAR รุ่น A1M8 ในการตรวจจับและหาระยะห่างของยานพาหนะที่อยู่ด้านหน้าของอุปกรณ์ และใช้กล้อง Logitech c922 pro สำหรับการตรวจจับและจำแนกชนิดของยานพาหนะหรือวัตถุบนท้องถนน ทางคณะผู้จัดทำได้ทำการใช้ภาษา python เพื่อสร้างและพัฒนาอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ โดยการนำโมเดล MobileNet เข้ามาช่วยในการจำแนกประเภทของวัตถุ และมีการปรับปรุง MobileNet ให้สามารถได้ผลลัพธ์การทำงานออกมาเป็นมุมของภาพโดยนำสมการที่ได้จากการสอบเทียบการหาความสัมพันธ์ที่เกิดขึ้นระหว่างมุมวัตถุภายในภาพที่ได้รับข้อมูลมาจากกล้องและมุมของเซนเซอร์ลิดาร์ที่ได้เก็บข้อมูลในขณะนั้นเข้ามาช่วย และมีการนำโมเดลสำเร็จรูปของ RPLIDAR มาปรับปรุงเพื่อให้สามารถนำข้อมูลระยะห่างที่ต้องการมาใช้ให้สามารถทำงานร่วมกับ MobileNet เมื่อนำทั้งสองมารวมกันจะทำให้ได้ระยะห่างของวัตถุที่ตรวจจับได้ และสามารถจำแนกชนิดของวัตถุนั้นได้ ทางคณะผู้จัดทำได้ทำการทดสอบอัลกอริทึมในสภาพแวดล้อมจริงโดยกำหนดระยะห่างไว้ที่ 2000, 3000 และ 4000 มิลลิเมตร พบว่าอัลกอริทึมสามารถให้ข้อมูลระยะห่างของวัตถุที่มีค่าใกล้เคียงกับระยะห่างจริงได้ ทางคณะผู้จัดทำจึงทำการพัฒนาให้อัลกอริทึมสามารถมีการแจ้งเตือนออกมาได้โดยเป็นการแจ้งเตือนในรูปแบบของเสียง โดยอัลกอริทึมจะเริ่มต้นการทำงานเมื่อตรวจจับวัตถุที่มีระยะห่างไม่เกิน 3 เมตร เมื่อนำอัลกอริทึมสำหรับระบบการแจ้งเตือนการชนในรถจักรยานยนต์ไปทำการทดสอบ ผลการทดสอบพบว่าระบบการแจ้งเตือนการชนในรถจักรยานยนต์

สามารถใช้งานได้ โดยที่อัลกอริทึมสามารถตรวจจับยานพาหนะ จำแนกชนิดยานพาหนะ และทำการแจ้งเตือนในรูปแบบของเสียงเมื่อตรวจพบยานพาหนะที่มีระยะห่างไม่เกิน 3 เมตร

Project title : Motorcycle Warning System

Name :. Mr. Natthanan Phutsorn ID. 60361163

: Mr. Nattapat Janthorn ID. 60361170

: Miss. Amonrat Singhara ID. 60365710

Project advisor : Asst. Prof. Choopong Chuaypen

Major : Mechanical Engineering

Department : Mechanical Engineering

Academic year : 2020

Abstract

This project was developed to study and create algorithms for the motorcycle warning system. The objective is to reduce the risk of accidents in the community area and on the road. To develop a Motorcycle Warning System and analyze object distance with a lidar sensor, the organizer's team has studied the data related to the automatic speed control system and how an automatic robot which uses obstacle detection systems work. Therefore, It works similarly to the automatic speed control system. The studied data can be developed and applied to the inventing of a Motorcycle Warning System using the RPLIDAR model A1M8 to detect and locate the distance of vehicles in front of the device and the Logitech c922 pro camera for detecting and classifying vehicles or objects on the road. In addition, the organizers' team has been using python language to create and develop algorithms for the Motorcycle Warning System by adopting MobileNet models to assist in the classification of objects. Moreover, MobileNet has been improved to achieve an angle of view performance by using the equation obtained from the calibration, the correlation between the angle of the object within the image obtained by the camera, and the Lidar sensor's angle which helps collect information at that time. The RPLIDAR ready-made model has been improved to enable the required distance

data to be used with MobileNet. Hence, when the two are combined, the distance between objects will be possible to detect and able to be classified the type. The organizer's team has tested the algorithm with spacing 2000, 3000, and 4000 millimeters in real environments. Consequently, It found that the algorithm was able to inform the distance of an object which is close to the actual distance. Thus, the organizer's team decided to develop an algorithm to be able to warn in the form of a sound. The algorithm is triggered when it detects an object with a distance of up to 3 meters. When the algorithm for the Motorcycle Warning System is tested, the result shows that the Motorcycle Warning System can work practically as the algorithm can detect the vehicle, classify the vehicle, and warn in the form of a sound alert when it detects a vehicle within 3 meters.

กิตติกรรมประกาศ

โครงงานระบบแจ้งเตือนการชนในรถจักรยานยนต์ คณะวิศวกรรมเครื่องกลฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ทางคณะผู้ดำเนินงานต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ชูพงศ์ ช่วยเพ็ญ ผู้ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานที่กรุณาให้คำปรึกษา ชี้แนะแนวทางและวิธีในการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นในระหว่างการดำเนินโครงงาน ตลอดจนติดตามผลการดำเนินโครงงานด้วยความเอาใจใส่ทุกขั้นตอนเพื่อให้การดำเนินงานผ่านพ้นไปได้ด้วยดี ทางคณะผู้จัดทำโครงงานขอขอบพระคุณท่านอาจารย์เป็นอย่างสูงมา ณ ที่นี้ ขอขอบพระคุณบรรดาอาจารย์ทุกท่านที่อบรมสั่งสอน และให้ความรู้แด่ผู้จัดทำจนสามารถนำความรู้ที่ได้รับมาพัฒนาและต่อยอดกับโครงงาน ขอขอบพระคุณฝ่ายเลขานุการภาควิชาวิศวกรรมเครื่องกลที่ช่วยอำนวยความสะดวกสบาย และแจ้งข้อมูลข่าวสารที่จำเป็นต่อการดำเนินการทำโครงงาน สุดท้ายนี้ผู้ดำเนินงานขอขอบพระคุณครอบครัวของผู้ดำเนินงาน สมาชิกผู้จัดทำโครงงาน มิตรสหายของผู้ดำเนินงาน ตลอดจนผู้มีพระคุณทุกท่านที่คอยให้กำลังใจและให้การช่วยเหลือทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี หากเกิดข้อผิดพลาดประการใดจากโครงงานนี้ ผู้ดำเนินงานต้องขอกราบอภัยมา ณ ที่นี้ด้วย

นายณัฏฐนันท์ พุฒซ้อน
นายณัฏฐพัชร จันทร
นางสาวอมรรัตน์ สิงหรา

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

สารบัญรูปภาพ (ต่อ)

หน้า

สารบัญรูปภาพ (ต่อ)

หน้า

สารบัญรูปภาพ (ต่อ)

หน้า

บทที่ 1
บทนำ

1.1. ที่มาและความสำคัญของโครงงาน

ในปัจจุบันนี้สาเหตุที่ทำให้เกิดการเสียชีวิตรองจากโรคต่าง ๆ นั่นคือ อุบัติเหตุที่เกิดบนทางถนน เนื่องจากอุบัติเหตุทางถนนเป็นปัญหาระดับโลกจากการรายงานขององค์กรอนามัยโลก (World Health Organization: WHO) ในปี 2018 พบว่า ปัจจุบันจำนวนผู้เสียชีวิตจากอุบัติเหตุทางถนนทั่วโลกเพิ่มขึ้นจาก 1.25 ล้านคน เป็น 1.35 ล้านคนในช่วงเพียงสามปีที่ผ่านมา ซึ่งเท่ากับวันละ 3,700 คน มากกว่าครึ่งหนึ่งของ ผู้เสียชีวิตจากการชนบนถนนเป็นผู้ขับขี่รถจักรยานยนต์ รถจักรยานและคนเดินถนน (Vulnerable Road Users) [1] สำหรับอุบัติเหตุในประเทศไทย จากรายงานการวิเคราะห์สถานการณ์ทางอุบัติเหตุทางถนนของกระทรวงคมนาคม พ.ศ. 2560 พบว่ามีการเกิดอุบัติเหตุทางถนนของประเทศไทยมีจำนวน 85,949 ครั้ง มีจำนวนผู้เสียชีวิต 8,746 ราย มีผู้บาดเจ็บ 3,785 ราย โดยส่วนใหญ่เกิดกับรถจักรยานยนต์คิดเป็นร้อยละ 37.38 โดยมูลเหตุสันนิษฐานที่ทำให้เกิดอุบัติเหตุมากที่สุดสามอันดับ โดยอันดับแรกคือ ตัดหน้ากระชั้นชิดเกิดขึ้น 9,885 ครั้ง อันดับสองคือ ขับรถเร็วเกิดขึ้น 8,773 ครั้ง และอันดับสามคือ ขับรถตามกระชั้นชิดเกิดขึ้น 8,220 ครั้ง และมีแนวโน้มเพิ่มขึ้นทุกปี [2] จากการเกิดอุบัติเหตุสำหรับรถจักรยานยนต์ที่มีแนวโน้มเพิ่มขึ้น จึงทำให้เกิดการพัฒนาระบบเบรกในรถจักรยานยนต์ที่มีชื่อเรียกว่า ระบบป้องกันล้อล็อกตาย หรือ ABS (Anti-Lock Brake System : ABS) และนำไปสู่การใช้ระบบ ACC (Adaptive Cruise Control : ACC) ที่มีในรถยนต์ มาประยุกต์ใช้ในรถจักรยานยนต์

ระบบเบรกสำหรับรถจักรยานยนต์ในปัจจุบันนี้ได้ถูกพัฒนาให้มีประสิทธิภาพมากขึ้น เพื่อลดการเกิดอุบัติเหตุบนทางถนน โดยเรียกว่าระบบ ABS มีหน้าที่ในการป้องกันการจับตัวของปั้มเบรกและดิสก์เบรก โดยจะมีเซนเซอร์วัดแรงบีบจากก้านเบรกหน้า หรือแป้นเบรกหลัง โดยจะทำการจับแรงบีบที่มีตามค่ากำหนดแล้วแต่รถแต่ล่ะคันจะเซตค่ามาจากโรงงาน หากตรวจพบแรงบีบจำนวนที่เกินกว่าค่าที่กำหนดไว้ตัวเซนเซอร์จะทำการคลายปั้มเบรก

และกดปั้มเบรกใหม่ในอีกจังหวะต่อมา เพื่อป้องกันการล็อกของล้อและลดแรงเสียดทานที่เกิดขึ้นกับหน้ายางและพื้นผิวของถนน ซึ่งจะทำให้ไม่เกิดอาการท้ายยกในเวลาที่ผู้ขับขี่ทำการเบรกหน้าหนัก นอกจากระบบเบรกที่ช่วยเรื่องความปลอดภัยของผู้ขับขี่แล้ว ผู้ขับขี่ควรคำนึงถึงระยะการเบรกที่ปลอดภัย โดยส่วนมากแล้ว ระยะเบรกจะแปรผันตามความเร็ว เช่น ที่ความเร็ว 30 mph หรือ 48 km/hr ระยะเบรก อาจเป็น 14 เมตร แต่ถาเพิ่มความเร็วเป็น 70 mph หรือ 112 km/hr ระยะเบรกอาจเป็น 75 เมตร ความเร็วเพิ่มขึ้นสองเท่าแต่ระยะเบรกอาจเพิ่มขึ้นถึงห้าเท่า [3] เนื่องจากรถยนต์ในปัจจุบันนี้มีระบบที่ช่วยควบคุมความเร็วในการขับขี่ให้สอดคล้องกับสภาพของการจราจรอย่างอัตโนมัตินั่นคือ ระบบ ACC โดยมีลักษณะการทำงาน คือ อาศัยข้อมูลจากเรดาร์ หรือเลเซอร์ จากนั้นทำการตั้งค่าความเร็วและระยะห่าง ระบบจะตรวจสอบระยะห่างไม่ว่าจะเป็นเรดาร์ หรือเลเซอร์ จะทำการวัดระยะห่างของรถคันหน้า เพื่อเพิ่ม-ลดความเร็วให้สอดคล้องกับรถคันหน้า และรักษาระยะห่างให้อยู่ในระดับที่ปลอดภัย รวมไปถึงกล้องเพื่อประมวลผลความเร็วการขับขี่อย่างเหมาะสมและปลอดภัย หลังจากที่เรดาร์สามารถตรวจจับความเร็วและระยะห่างของรถคันหน้าได้แล้ว ข้อมูลเหล่านี้ จะถูกส่งไปให้กับระบบประมวลผล หลังจากนั้นจะส่งสัญญาณเพื่อไปควบคุมการทำงานของเครื่องยนต์ เพื่อให้สามารถเพิ่ม-ลดความเร็วได้อย่างอัตโนมัติ นอกจากนั้น ข้อมูลเหล่านี้ยังถูกส่งไปที่ระบบเกียร์ เพื่อให้เกียร์สามารถปรับเปลี่ยนอัตราทดให้เหมาะสมกับการขับขี่ หากเกิดเหตุการณ์รถคันหน้าเบรกกะทันหัน ระบบประมวลผลจะสั่งการให้ระบบเบรกทำงานอย่างอัตโนมัติเพื่อลดความเร็วแบบฉับพลัน ซึ่งจะช่วยหลีกเลี่ยงการชนได้อย่างมีประสิทธิภาพ [4] แม้กระทั่งหุ่นยนต์อัตโนมัติยังมีการใช้ระบบตรวจจับสิ่งกีดขวาง ที่มีลักษณะการทำงานที่คล้ายคลึงกับระบบ ACC ในด้านของการตรวจจับ โดยที่หุ่นยนต์อัตโนมัติจะมีกล้องไว้สำหรับแสกนหาสิ่งกีดขวาง มักใช้สำหรับการเคลื่อนย้ายสินค้าและการจัดเก็บสินค้า การพัฒนาของหุ่นยนต์อัตโนมัติจึงส่งผลให้ลดการเสียค่าใช้จ่ายในการจ้างพนักงานและเพิ่มประสิทธิภาพในกระบวนการเคลื่อนย้ายและจัดเก็บสินค้า

ทั้งหมดที่กล่าวมานั้นเป็นที่มาและความสำคัญในการจัดทำโครงงาน เพื่อที่จะประยุกต์ใช้ระบบการตรวจจับวัตถุที่มีทั้งในรถยนต์หรือหุ่นยนต์อัตโนมัติ มาปรับใช้กับรถจักรยานยนต์ โดยที่ให้มีระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (Motorcycles Warning System : MWS) ด้วยการนำ lidar เข้ามาช่วยในการตรวจจับสิ่งกีดขวางเพื่อควบคุมการทำงานของสัญญาณเตือน หากมีระบบ MWS ในรถจักรยานยนต์จะสามารถช่วยลดอัตราการเกิดอุบัติเหตุบนทางถนนหรือในพื้นที่ชุมชน และช่วยให้ผู้ขับขี่รถจักรยานยนต์มีความปลอดภัยมากขึ้น

1.2. วัตถุประสงค์ของการศึกษา

1) เพื่อลดปัญหาการเกิดอุบัติเหตุในพื้นที่ชุมชนและบนทางถนน
2) เพื่อพัฒนาระบบการแจ้งเตือนการชนในรถจักรยานยนต์

3) เพื่อวิเคราะห์ระยะห่างวัตถุด้วยเซนเซอร์ lidar

1.3. ขอบเขตของโครงงาน

1) พัฒนาอัลกอริทึมในการตรวจจับและวิเคราะห์การเคลื่อนที่ของสิ่งกีดขวางสำหรับใช้ในรถจักรยานยนต์
2) พัฒนาอัลกอริทึมและโปรแกรมตรวจจับและวิเคราะห์ ฯ โดยใช้ข้อมูลจากกล้องวีดีโอและ lidar sensor
3) พัฒนาชุดอุปกรณ์ที่ใช้อัลกอริทึมและโปรแกรมข้างต้นเพื่อใช้ติดตั้งบนรถจักรยานยนต์
4) โปรแกรมและชุดอุปกรณ์ที่พัฒนาสามารถตรวจจับและวิเคราะห์สิ่งกีดขวางได้ในระยะ 9 เมตร และสำหรับใช้รถจักรยานยนต์ที่มีความเร็วไม่เกิน 30 กิโลเมตรต่อชั่วโมง
5) โปรแกรมและชุดอุปกรณ์สามารถแจ้งเตือนผู้ขับขี่ได้ในระยะเวลาน้อยกว่า 2.5 วินาที

1.4. ขั้นตอนและแผนการดำเนินงาน

ตารางที่ 1 แผนการดำเนินงาน

ลำดับ	กิจกรรม	2562	2563			2564	
		ธ.ค.	ม.ค.-ก.พ.	มี.ค.-เม.ย.	พ.ค.-ธ.ค.	ม.ค.-มี.ค.	เม.ย.
1	ศึกษาทฤษฎีรวมถึงข้อมูลรายละเอียดต่างๆ ของงานวิจัยที่เกี่ยวข้อง						
2	ออกแบบและพัฒนาการตรวจจับด้วยเซนเซอร์ lidar						
3	สร้างอัลกอริทึมในการวิเคราะห์และตรวจจับวัตถุของรถจักรยานยนต์ต้นแบบ						
4	ทดสอบ และประเมินผลประสิทธิภาพของระบบ						
5	วิเคราะห์และสรุปผลการดำเนินงาน						
6	จัดทำรูปเล่มและสื่อนำเสนอโครงงาน						

1.5. ผลที่คาดว่าจะได้รับ

1) อัลกอริทึมและโปรแกรมที่สามารถตรวจจับและวิเคราะห์การเคลื่อนที่ของสิ่งกีดขวางที่ในระยะ 9 เมตร
2) ชุดอุปกรณ์ที่สามารถแจ้งเตือนถึงสิ่งกีดขวางแก่ผู้ขับขี่รถจักรยายนต์เพื่อลดโอกาสในการเกิดอุบัติเหตุ

1.6. งบประมาณที่ใช้

1) บอร์ด Raspberry Pi 3 Model B ราคา 1,235.58 บาท
2) เซนเซอร์ลิดาร์ RPLIDAR รุ่น A1M8 ราคา 3,290 บาท
3) กล้อง Logitech c922 pro ราคา 2,990 บาท

บทที่ 2
ทฤษฎีและหลักการทำงาน

2.1. ทฤษฎี

สำหรับการขับขี่รถจักรยานยนต์ในปัจจุบันมีการควบคุมความเร็วสำหรับการขับขี่ โดยยึดตามพระราชบัญญัติจราจรทางบก พ.ศ.2522 เป็นกฎหมายหลักที่ใช้ในการควบคุมการใช้ความเร็วในประเทศไทย โดยได้ให้อำนาจรัฐมนตรีในการกำหนดอัตราความเร็วออกมาในรูปแบบกฎกระทรวงซึ่งผลของกฎกระทรวงที่ประกาศออกมาจะเป็นการควบคุมความเร็วโดยทั่วไปทั้งประเทศ โดยมาตรา 67 แห่งพระราชบัญญัติจราจรทางบก พ.ศ.2522 ได้บัญญัติไว้สำหรับการควบคุมความเร็วทั่วไป รถยนต์และรถจักรยานยนต์ให้ขับในเขตกรุงเทพมหานคร เขตเมืองพัทยา หรือเขตเทศบาล ไม่เกินชั่วโมงละ 80 กิโลเมตร หรือนอกเขตดังกล่าวให้ขับไม่เกินชั่วโมงละ 90 กิโลเมตร [5] ดังรูปที่ 2.1 และจากการศึกษาการทดสอบด้วยการขับรถด้วยความเร็ว 50 กิโลเมตรต่อชั่วโมงในเขตชุมชนหรือเขตในเมืองพบว่าจะมีระยะเบรกที่ไม่เพียงพอจนสามารถทำให้เกิดอุบัติเหตุชนคนเสียชีวิตได้ จึงได้ทำการลดความเร็วในการขับขี่ลงมาที่ 30 กิโลเมตรต่อชั่วโมง ซึ่งช่วยให้ลดการเกิดอุบัติเหตุได้ แม้จะมีการควบคุมความเร็วแต่ยังไม่สามารถทำให้ลดการเกิดอุบัติเหตุได้ ทำให้เกิดความสนใจที่จะทำการพัฒนาระบบการแจ้งเตือนการชนในรถจักรยานยนต์ โดยที่ระบบการแจ้งเตือนการชนในรถจักรยานยนต์ คือ ระบบที่ถูกพัฒนาเพื่อใช้ในรถจักรยานยนต์สามารถทำการตรวจจับสิ่งกีดขวางข้างหน้าแล้วทำการแจ้งเตือนให้กับรถจักรยานยนต์เพื่อลดการเกิดอุบัติเหตุบนทางถนน ในระบบการแจ้งเตือนการชนของผู้ดำเนินโครงการจะประกอบด้วย lidar, ราสเบอร์รี่ พาย (Raspberry Pi) หรือบอร์ด NVIDIA Jetson Nano Developer Kit และ Open CV โดยที่ระบบการแจ้งเตือนการชนนี้จะสอดคล้องกับระยะการเบรกของรถจักรยานยนต์ที่ขึ้นกับความเร็ว ณ ขณะนั้น

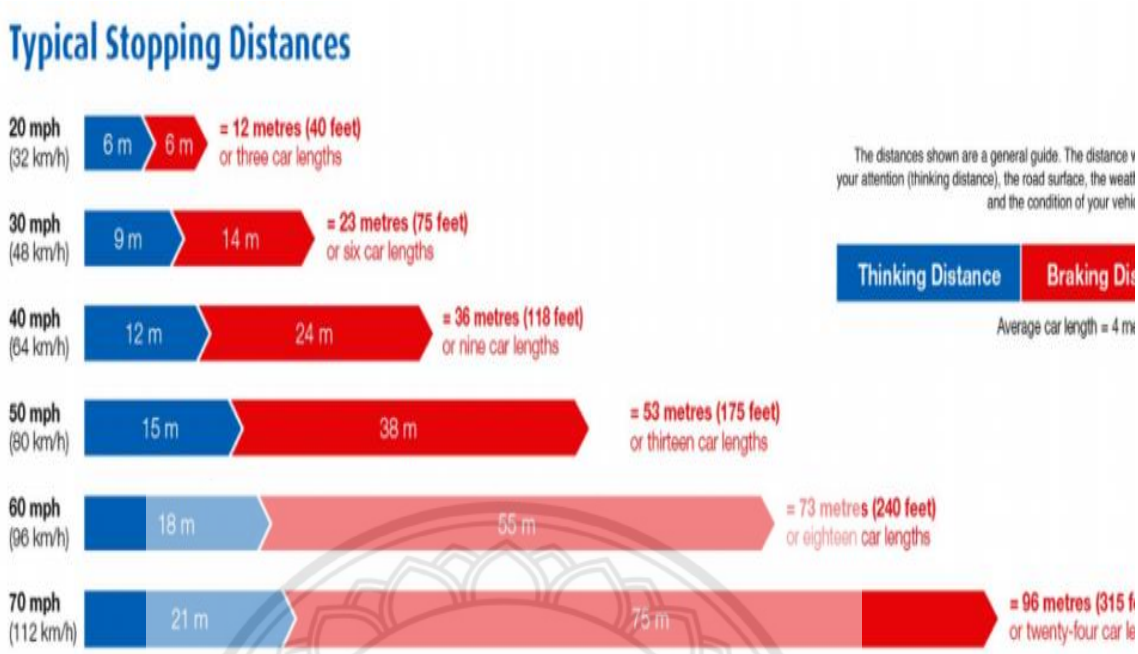
รูปภาพที่ 2.1 การควบคุมความเร็วทั้งทั่วไปและความเร็วสำหรับเฉพาะที่
(ที่มา : https://www.tqm.co.th/blog/ความเร็วเท่าไหร่ไม่โดนปรับ/,2561)

สำหรับข้อมูลที่มีความสำคัญในการดำเนินโครงงานครั้งนี้ ในด้านของข้อมูลทางเทคนิค นั่นคือข้อมูลของระยะการเบรกที่มีความเกี่ยวข้องกับความเร็ว และอุปกรณ์ที่ใช้ในการดำเนินโครงงาน โดยมีรายละเอียดดังต่อไปนี้

ระยะการเบรกนั้นมีความสำคัญในการจัดทำโครงงานเนื่องจาก ระบบการแจ้งเตือนการชนในรถจักรยานยนต์นั้นมีความสอดคล้องกับการคำนวณระยะเบรก หากไม่ทราบการคำนวณระยะเบรกจะทำให้ตัวเซนเซอร์ไม่สามารถคำนวณระยะที่รถจักรยานยนต์อยู่ห่างจากสิ่งกีดขวางข้างหน้าได้

2.1.1) ระยะการเบรก

ระยะเบรก คือ ระยะทางที่ยานพาหนะใช้ในการเคลื่อนที่ไปตั้งแต่เริ่มทำการเบรกจนกระทั่งยานพาหนะอยู่ในสภาวะหยุดนิ่ง ณ ขณะนั้น โดยระยะทางที่ใช้ในการเบรกจะขึ้นอยู่กับความเร็วตอนที่เริ่มทำการเบรกซึ่งเมื่อความเร็วก่อนการเบรกสูงระยะทางใช้ในการเบรกจะสูงด้วยเช่นกัน



รูปภาพที่ 2.2 ระยะทางที่ใช้ในการตัดสินใจทำการเบรกและระยะทางที่ใช้ในการเบรก (ที่มา : http://begin-motorcycling.co.uk/the-5-elements-of-cbt/element-c/braking/,2560)

จากรูปที่ 2.2 แสดงให้เห็นถึงระยะทางที่ใช้ในการคิดก่อนการเบรกและระยะทางที่ใช้เบรกในและช่วงความเร็วโดยที่สีฟ้าจะแสดงถึงระยะทางที่ใช้ในการคิดก่อนทำการเบรกและสีแดงคือระยะทางที่ใช้ในการเบรกซึ่งเมื่อนำระยะทางทั้งสองมารวมกันจะได้เป็นระยะทางที่ใช้ในการเบรกทั้งหมด เช่น ที่ความเร็ว 20 mph (32 km/h) จะใช้เวลาในการคิด 6 m และระยะทางที่ใช้เบรกคือ 6 m ดังนั้นระยะทางที่ใช้ในการเบรกที่ความเร็ว 20 mph (32 km/h) เท่ากับ 12 m

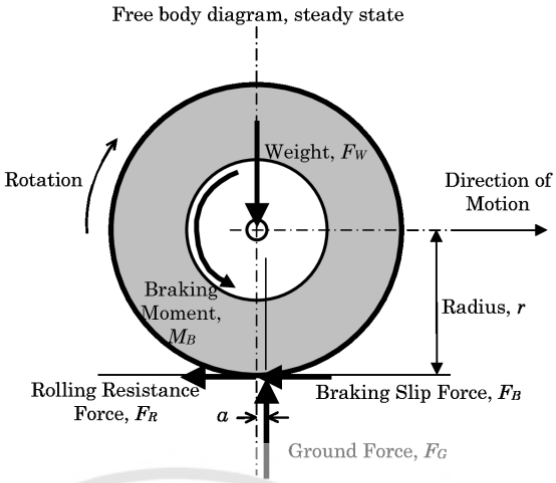


Figure 10. Forces and moments of a constant-braked wheel on a bare, **dry paved surface** (Andresen and Wambold, 1999).

รูปภาพที่ 2.3 แรงที่กระทำกับล้อรถขณะเบรก
(ที่มา : https://www.nap.edu/read/23038/chapter/5#23)

จากรูปที่ 2.3 แสดงให้เห็นถึงแรงที่เกิดขึ้นบนล้อของรถเมื่อทำการเบรก โดยจะมีแรงต้านทานการหมุน (F_R) แรงในการเบรก (F_B) โมเมนต์ของการเบรก (M_B) แรงจากน้ำหนักของรถ (F_W) และแรงที่พื้นกระทำกับล้อ (F_G) เมื่อยานพาหนะทำการเบรกจะส่งผลให้เกิดแรงเสียดทานการเบรก F ซึ่งสามารถหาแรงดังกล่าวได้จากสมการ

$$F = \mu_k m_R g \tag{2.1}$$

และจากกฎข้อที่ 2 ของนิวตัน

$$F = m_{total} a \tag{2.2}$$

นำสมการที่ (2.1) มาเท่ากับ (2.2)

จะได้ $$\mu_k m_R g = m_{total} a$$

จัดรูปสมการ

$$a = \frac{\mu_k m_R g}{m_{total}} \tag{2.3}$$

เมื่อ μ_k คือ สัมประสิทธิ์แรงเสียดทาน (หาได้จากตารางที่ 2.1)

m_R คือ มวลที่กดลงบนล้อที่ทำการเบรก

g คือ ความเร่งเนื่องจากแรงโน้มถ่วงของโลก

m_{total} คือ มวลทั้งหมดของรถ

a คือ ความเร่งของรถเมื่อทำการเบรก

ตารางที่ 2.1 สัมประสิทธิ์แรงเสียดทานสถิต (μ_s) และสัมประสิทธิ์แรงเสียดทานจลน์ (μ_k)

Surfaces	μ_s (static)	μ_k (kinetic)
Steel on steel	0.74	0.57
Glass on glass	0.94	0.40
Metal on Metal (lubricated)	0.15	0.06
Ice on ice	0.10	0.03
Teflon on Teflon	0.04	0.04
Tire on concrete	1.00	0.80
Tire on wet road	0.60	0.40
Tire on snow	0.30	0.20

(ที่มา : http://ffden-2.phys.uaf.edu/211_fall2002.web.dir/ben_townsend /staticandkineticfriction.htm)

จากสมการที่ (2.3) จะสามารถนำค่าความเร่งที่คิดได้มาคำนวณหาระยะทางที่ในการเบรกของรถโดยจะสามารถหาได้จากสมการการเคลื่อนที่ คือ

$$s = \frac{u^2 - v^2}{2a} \qquad (2.4)$$

เมื่อ s คือ ระยะทางในการเบรก

u คือ ความเร็วเมื่อรถหยุดนิ่ง (มีค่า = 0)

v คือ ความเร็วเมื่อรถเริ่มทำการเบรก

a คือ ความเร่งของรถเอทำการเบรก (หาได้จากสมการ 2.3)

สำหรับ lidar เป็นหนึ่งในอุปกรณ์สำคัญที่ต้องนำมาใช้ในการทำระบบการแจ้งเตือนการชนในรถจักรยานยนต์ เนื่องจาก lidar มีคุณสมบัติในการสแกนสิ่งกีดขวางหรือใช้ในการสำรวจ ซึ่งข้อมูลที่ได้สามารถนำไปสร้างภาพที่มีประโยชน์ และยังได้ภาพที่มีประสิทธิภาพของพื้นผิวที่มีรายละเอียดมากกว่าภาพถ่ายจากดาวเทียม โดยที่รายละเอียดการทำงานของ lidar และการเปรียบเทียบ lidar รุ่น RPLIDAR A1 กับ Smartfly EAI YDLIDAR TX20 เพื่อประกอบการตัดสินใจ โดยมีข้อมูลดังต่อไปนี้

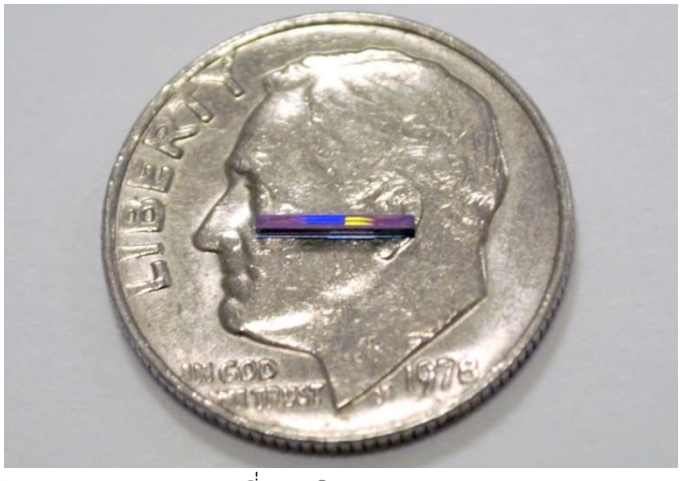
2.1.2) lidar

lidar ย่อมาจาก light detection and ranging คือเครื่องมือที่ใช้ในการสำรวจและวัดระยะทางโดยการใช้แสงเลเซอร์ในการฉายเลเซอร์ไปยังเป้าหมายที่สนใจและเซนเซอร์ที่ทำการรับแสงที่สะท้อนกลับมาและนำไปคำนวณระยะทางจากเวลาที่ใช้ในการเดินไปและกลับมายังเซนเซอร์ โดยส่วนใหญ่ lidar มักใช้ในงานที่เกี่ยวกับการสำรวจต่าง ๆ เช่น การสำรวจซากโบราณที่ถูกฝังอยู่ภายใต้พื้นดิน ดังแสดงในรูปที่ 2.4 ซึ่ง lidar จะเก็บข้อมูลอยู่ในรูปของเวกเตอร์จุดสามมิติ โดย lidar ที่ใช้ในการสำรวจนี้มักจะใช้เครื่องบินในการติดตั้ง lidar ซึ่ง lidar จะทำงานร่วมกับการใช้ตำแหน่งและระบบนำทางที่แม่นยำทำให้ในแต่ละจุดที่บันทึกจะอยู่รูปแบบของระนาบ x, y, z ที่จะมีค่าแตกต่างกันไปตามเวลาที่คำนวณได้ระหว่างการรับส่งของเลเซอร์ [6] สำหรับ lidar ที่ใช้ในงานสำรวจภูมิประเทศมีเป้าหมายเพื่อตรวจสอบพื้นผิวภูมิประเทศ โดยข้อมูลความสูงที่ได้จากการวัดระยะจากสื่อเช่น LASER ทำให้ระยะที่วัดได้มีความละเอียดสูง ก่อให้เกิดจุดข้อมูลความสูงของภูมิประเทศจำนวนมหาศาล การใช้งานต้องนำข้อมูลเหล่านี้มาประมวลผล เพื่อให้เกิดเป็นโครงสร้างภูมิประเทศในลักษณะ TIN หรือ DEM ข้อมูลดิบที่ได้ lidar จะเป็นข้อมูลที่เป็นลักษณะ DSM หรือ Digital Surface Model นั่นคือความสูงที่ได้จะเป็นที่รวมสิ่งปลูกสร้างหรือสิ่งที่ปกคลุมผิวโลก เช่น ต้นไม้ ตึกอาคาร และพืชพรรณไปด้วย ปัจจุบันข้อมูลความสูงจาก lidar จึงมีอยู่สองลักษณะคือ แบบที่เป็น Bare Ground คือได้ขจัดสิ่งที่ปกคลุมพื้นผิวไปแล้ว กับอีกแบบคือ แบบ Reflective คือข้อมูลความสูงที่รวมสิ่งปลูกสร้างไปด้วย Reflective คือ ข้อมูลที่นำมาใช้ในการดึงข้อมูลตึกและอาคารมาใช้ เพื่อให้เป็นข้อมูลสำหรับแสดงผลในลักษณะสามมิติซึ่งจะทำให้ลดกำลังภายในการใส่ข้อมูลตึกอาคารในลักษณะ Manual ไปสู่ Automatic

รูปภาพที่ 2.4 การจำลอง lidar สำหรับงานสำรวจภูมิประเทศ
(ที่มา : https://thematter.co/science-tech/meet-lidar-revealed-cambodia-lost-city/4052/,2559)

จากรูปที่ 2.4 จะแสดงให้เห็นถึงตัวอย่างของการทำงานของ lidar ที่ใช้ในการสำรวจภูมิประเทศต่างๆ โดยในภาพจะให้เห็นถึงภาพที่ได้จากการประมวลผลของ lidar ด้านขวาจะแสดงให้เห็นถึงภาพของภูมิประเทศที่ต้องการสำรวจ

ในช่วงระยะเวลาหลายปีที่ผ่านมามีการนำ lidar เข้ามาประยุกต์ใช้กับรถยนต์ซึ่ง lidar เป็นอุปกรณ์ที่สำคัญมากตัวหนึ่งในรถยนต์ระบบขับเคลื่อนอัตโนมัติ โดยเมื่อปี 2559 ห้องวิจัย Photonic Microsystems Group ของ MIT พยายามแก้ปัญหาเรื่องของขนาด น้ำหนักและราคา โดยเปลี่ยน lidar จากระบบกลไกมาเป็นชิป CMOS ซึ่งลดขนาดลงได้มากและลดต้นทุนการผลิตลง ชิป lidar ของ MIT มีขนาด 0.5 x 6 มิลลิเมตร และทำงานประมวลผลแสงได้เร็วกว่าเดิม 1,000 เท่า [7] ดังรูปที่ 2.5



รูปภาพที่ 2.5 ชิป lidar ของ MIT
(ที่มา : https://www.blognone.com/node/84463/,2559)

ในการจัดทำระบบการแจ้งเตือนการชนในรถจักรยานยนต์ได้ทำการเปรียบเทียบ lidar รุ่น RPLIDAR A1M8 กับ Smartfly EAI YDLIDAR TX20 สำหรับ RPLIDAR A1M8 ดังแสดงในรูป 2.6 สามารถสแกนได้ 360° ในรัศมี 12 เมตร มีอัลกอริธึมที่สามารถวัดระยะได้ 8000 ครั้งต่อวินาที ความถี่ที่ใช้ในกานสแกนอยู่ที่ 5.5 Hz ซึ่งสามารถเพิ่มสูงถึง 10 Hz และสำหรับ Smartfly EAI YDLIDAR TX20 ดังรูปที่ 2.7 สามารถสแกนได้ 360° ในรัศมี 20 เมตร มีอัลกอริทึมที่สามารถวัดระยะได้ 4000 ครั้งต่อวินาที ความถี่ที่ใช้ในการสแกนอยู่ที่ 5-12 Hz และยังมีสามารถใช้งานที่อยู่ภายนอกอาคารได้ เนื่องจากมีคุณสมบัติในการป้องกันแสงแดด สามารถนำคุณสมบัติของทั้งสองมาเปรียบเทียบเพื่อการตัดสินใจ ดังแสดงในตารางที่ 2.2

รูปภาพที่ 2.6 lidar รุ่น RPLIDAR A1M8
(ที่มา : https://www.slamtec.com/en/Lidar/S1)

รูปภาพที่ 2.7 Smartfly EAI YDLIDAR TX20
(ที่มา : https://th.aliexpress.com/item/4000623574990.html)

ตารางที่ 2.2 การเปรียบเทียบคุณสมบัติของ RPLIDAR A1M8 และ Smartfly EAI YDLIDAR TX20 [8],[9]

1. Model	RPLIDAR A1M8	Smartfly EAI YDLIDAR TX20
2. Dimension	2D	2D
3. ขนาด	98.5 mm x 70 mm x 60 mm	60 mm x 47.5 mm x 96 mm
4. น้ำหนัก	170 กรัม	180 กรัม
5. Angular range	360°	360°
6. ความละเอียดมุม	≤ 1°	0.63°
7. Distance Range	0.15 - 12 meter	0.1 - 20 meter
8. Scan rate	5.5 – 10 Hz	5 – 12 Hz
9. Sampling frequency	8000 time/s	4000 time/s
10. Supply voltage	5 V	5 V

จากตารางที่ 2.2 เป็นการเปรียบเทียบคุณสมบัติของ lidar สองตัวที่มี model ต่างกัน โดยที่ทั้งสอง model จะแสดงผลแบบสองมิติมี Angular range เท่ากับ 360° และ Supply voltage 5 V เหมือนกันทั้งสอง model แต่ทั้งสอง model มีคุณสมบัติที่แตกต่างกันอยู่ ได้แก่ ความละเอียดของมุมใน model RPLIDAR A1M8 มีความละเอียดมุมอยู่ที่ ≤ 1° ซึ่งมีความละเอียดน้อยกว่า model Smartfly EAI YDLIDAR TX20 ที่มีความละเอียดมุมอยู่ที่ 0.63° model RPLIDAR A1M8 มี Distance Range อยู่ที่ 0.15 - 12 meter ซึ่งน้อยกว่า model Smartfly EAI YDLIDAR TX20 ที่มี Distance Range อยู่ที่ 0.1 - 20 meter model RPLIDAR A1 มี Scan rate อยู่ที่ 5.5 – 10 Hz ซึ่งน้อยกว่า model Smartfly EAI YDLIDAR TX20 ที่มี Scan rate อยู่ที่ 5 – 20 Hz model

RPLIDAR A1 มี Sampling frequency อยู่ที่ 8000 time/s ซึ่งมีค่ามากกว่า model Smartfly EAI YDLIDAR TX20 ที่มี Sampling frequency อยู่ที่ 4000 time/s

สำหรับระบบฝังตัว หรือระบบสมองกลฝังตัว (Embedded System) คือ ระบบการประมวลผล โดยที่จะใช้ชิปหรือไมโครโพรเซสเซอร์ที่ออกแบบมาเฉพาะ ซึ่งจัดเป็นระบบคอมพิวเตอร์ขนาดจิ๋วที่ฝังไว้ในอุปกรณ์เครื่องใช้ไฟฟ้า และเครื่องเล่นอิเล็กทรอนิกส์ต่าง ๆ เพื่อเพิ่มความฉลาดความสามารถให้กับอุปกรณ์เหล่านั้นผ่านซอฟต์แวร์ซึ่งต่างจากระบบประมวลผลที่เครื่องคอมพิวเตอร์ทั่วไป และระบบฝังตัวนี้ยังถูกนำไปใช้ในยานพาหนะด้วยเช่นกัน ดังนั้นระบบการประมวลผลแบบฝังตัวนี้จึงสอดคล้องและมีความสำคัญในการดำเนินโครงงาน เนื่องจากได้ทำการเลือกบอร์ดราสเบอร์รี่พาย (Raspberry Pi) และ NVIDIA Jetson Nano Developer Kit มาเป็นหนึ่งในอุปกรณ์ที่จะเข้ามาช่วยควบคุมการตรวจจับของเซนเซอร์ ซึ่งอุปกรณ์ทั้งสองมีข้อมูลดังต่อไปนี้

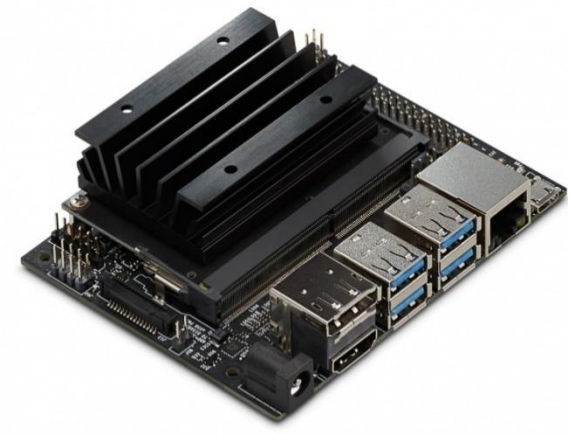
2.1.3) ราสเบอร์รี่ พาย (Raspberry Pi)

Raspberry Pi คือบอร์ดคอมพิวเตอร์ขนาดเล็กเท่าบัตรเครดิต เชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ดและเมาส์ ซึ่งถูกพัฒนาขึ้นโดยมูลนิธิ Raspberry Pi เป็นองค์กรการกุศลของสหราชอาณาจักร มีจุดประสงค์เพื่อนำพลังด้านดิจิตอลเข้าสู่ผู้ใช้งานทั่วโลก ดังนั้น จึงทำให้ผู้ใช้งานสามารถทำความเข้าใจและสร้างโลกดิจิทัลเพิ่มขึ้นได้โดยง่าย สามารถแก้ปัญหาที่สำคัญได้ และเตรียมพร้อมสำหรับงานในอนาคต Raspberry Pi นั้นถือได้ว่าเป็นคอมพิวเตอร์ที่มีประสิทธิภาพสูงและราคาประหยัด อีกทั้ง Raspberry Pi สามารถเชื่อมต่อกับระบบเครือข่ายแบบใช้สายหรือไม่ใช้สายได้ จึงทำให้กลายเป็นอุปกรณ์ Internet of Things โดยสมบูรณ์ ช่วยให้นักวิจัยและผู้ที่สนใจ สามารถนำไปประยุกต์ใช้เพื่อเชื่อมต่อกับตัวตรวจจับ (Sensor) ในการเก็บข้อมูลตามที่ต้องการ รวมถึงสามารถเชื่อมต่อกับแป้นพิมพ์และเมาส์ได้ สำหรับการเขียนโปรแกรมควบคุม Raspberry Pi สามารถใช้ภาษา Python เพื่อใช้แก้ปัญหาหรือ ใช้ในการควบคุมการทำงานของตัวตรวจจับ (Sensor) [10] โดย Raspberry Pi รุ่น Raspberry Pi 3 Model B มีลักษณะของตัวเครื่อง ดังแสดงในรูปที่ 2.8

รูปภาพที่ 2.8 บอร์ด Raspberry Pi รุ่น Raspberry Pi 3 Model B
(ที่มา : https://www.amazon.co.uk/Raspberry-Pi-Model-Quad-Motherboard/dp/B01CD5VC92?th=1)

2.1.4) NVIDIA Jetson Nano Developer Kit

NVIDIA Jetson Nano Developer Kit คือ อุปกรณ์สำหรับพัฒนาปัญญาประดิษฐ์ และ Machine Learning โดยที่เปรียบเสมือนเครื่องคอมพิวเตอร์ขนาดเล็กดังแสดงในรูปที่ 2.9 ซึ่งเป็นอุปกรณ์ที่มีความสามารถในการประมวลผลมากมายผ่าน GPU ของ NVIDIA และยังมีความสามารถในการประมวลผล neural networks ได้ในจำนวนมาก ๆ พร้อมกัน สำหรับ NVIDIA Jetson Nano Developer Kit นั้นยังมีความเหมาะสมสำหรับงานประเภท image classification, object detection และ speech processing



รูปภาพที่ 2.9 บอร์ด NVIDIA Jetson Nano Developer Kit
(ที่มา : https://www.arduitronics.com/product/2606)

Raspberry Pi และ NVIDIA Jetson Nano อุปกรณ์ทั้งสองได้นำเสนอคุณสมบัติที่น่าสนใจเพื่อผู้ใช้ที่มีทักษะระดับสูงและผู้เริ่มต้นใช้งาน โดยที่ Raspberry Pi ในอดีตนั้นถือได้ว่าเป็นผู้นำด้านประสิทธิภาพของ GPU และฟังก์ชั่นของ AI สำหรับ Raspberry Pi รุ่น 3 Bนั้นมาพร้อมกับหน่วยประมวลผล Broadcom-64-bit แบบ quad-core ARM A53 clocked at 1.2GHz, 1GB LPDDR2, ส่วนหัวของ GPIO 40 pin, HDMI และ Bluetooth 4.1 support ในทางกลับกันแล้วนั้น NVIDIA Jetson Nano ให้ความสำคัญกับการเรียนรู้และพัฒนาของ Machine Learning อย่างจริงจัง เห็นได้จาก GPU ที่มีการพัฒนาเป็น NVIDIA Maxwell 128 CUDA ที่เหมาะสำหรับการเรียนรู้ของ Machine Learning และยังมี 472 GFLOPS สำหรับการรองรับการดำเนินงานของ AI ซึ่งตรงข้ามกับ Raspberry Pi รุ่น 3 B ที่จะมีเพียง 21.4 GFLOPs .ในส่วนของการให้พลังงาน GPU เพิ่มมากขึ้นประสิทธิภาพการทำงานของ CPU เป็นแบบ 64-bit Quad-core Cortex A57 และ RAM 4GB ขนาดใหญ่ ในขณะที่ Raspberry Pi รุ่น 3 B ถูกพัฒนาขึ้นมาในรูปแบบการสร้างที่คล้ายคลึงกัน แต่กลับให้ RAM เพียง 1GB เท่านั้น สิ่งนี้จึงทำให้ Raspberry Pi รุ่น 3 B เหมาะสำหรับงานพื้นฐาน แต่อาจพบกับความยุ่งยากเมื่อใช้งานบนเดสก์ทอป แต่ RAM 4GB และ CPU ที่มีประสิทธิภาพของ NVIDIA Jetson Nano เป็นตัวเลือกที่ดีกว่าหากงานนั้นต้องดำเนินการอย่างหนักและมีการใช้งานบนเดสก์ทอป เมื่อพูดถึงการประมวลผลวิดีโอของฟังก์ชั่น AI สำหรับ NVIDIA Jetson Nano นั้นจะมีความสว่างขึ้น เนื่องจากสามารถประมวลผลวิดีโอในระดับ 4K โดยใช้ฮาร์ดแวร์ออน บอร์ด เพื่อเข้ารหัส ถอดรหัส และแสดงผลออกมา บอร์ดนี้ยังสามารถเรียกใช้เครือข่ายประสาทเทียมแบบขนานเพื่อที่จะประมวลผลวิดีโอและเซ็นเซอร์หลายตัวพร้อมกัน และสามารถประมวลผลวิดีโอได้หลายรายการ โดยที่ให้วิดีโอ 1080p สูงสุดถึง 8 ครั้ง สำหรับในแต่ละครั้ง และใช้อัลกอริทึมการเรียนรู้ของเครื่องเพื่อตรวจจับและติดตามภาพ จากข้อมูลคุณสมบัติทางเทคนิคเบื้องต้นนี้สามารถนำมาสร้างตารางเพื่อทำการเปรียบเทียบในการตัดสินใจเลือกใช้อุปกรณ์ทั้งสอง ดังแสดงในตารางที่ 2.3

ตารางที่ 2.3 การเปรียบเทียบคุณสมบัติของ Raspberry Pi 3 Model B และ NVIDIA Jetson Nano [11]

Model	Raspberry Pi 3 Model B	NVIDIA Jetson Nano
AI Performance	24 GFLOPs	472 GFLOPs
CPU	1.2 GHz 64-bit quad-core ARM Cortex-A53	Quad-Core ARM Cortex-A57 64-bit @ 1.42 GHz
GPU	Broadcom VideoCore IV	128-core NVIDIA Maxwell
RAM	1GB LPDDR2 SDRAM	4GB LPDDR4
Board dimensions	85.6 x 56 mm.	100 x 79 mm.
Networking	802.11b/g/n wireless LAN	Gigabit Ethernet / M.2 Key E (for Wifi support)
Display	Full size – HDMI, 3.5 mm analogue audio-video jack	HDMI 2.0 and eDP 1.4
Ports	4 USB 2.0, Wired Ethernet 10/100 Mbps	4xUSB 3.0, Wired Ethernet 10/100/1000 Mbps
Model	Raspberry Pi 3 Model B	NVIDIA Jetson Nano
GPIO head	40-pin	40-pin
M.2 Key E Slot	ไม่มี	มี
Storage	micro SD card	Micro-SD

จากตารางที่ 2.3 นั้นมีการเปรียบเทียบประสิทธิภาพการทำงานของ AI ที่ 472 GFLOPs ของ NVIDIA Jetson Nano จะให้ประสิทธิภาพมากกว่า 19เท่า ซึ่งมากกว่า Raspberry Pi รุ่น 3 B ที่ให้เพียง 24 GFLOPs ดังนั้นสำหรับบอร์ดคอมพิวเตอร์ based-single ที่ใช้ระบบ AI จึงเหมาะสมที่จะเลือกใช้ NVIDIA Jetson Nano

อย่างไรก็ตาม Raspberry Pi ยังมีข้อดีที่ถือได้ว่าเป็นจุดแข็ง นั่นคือ สามารถเลือกใช้ได้ในหลากหลายสถานการณ์ ตัวอย่างเช่น แม้จะมีความสามารถด้านพลังงานและ AI แต่ NVIDIA Jetson Nano ยังขาด LAN ไร้สาย ในทางตรงกันข้าม Raspberry Pi รุ่น 3 B นั้นมีคุณสมบัติ Wi-Fi ในตัว แต่ตัวควบคุมอีเธอร์เน็ตแบบใหม่ของ Raspberry Pi รุ่น 3 B นั้นให้การเชื่อมต่ออัตราความเร็วสูงสุดเพียง 100 Mb/s ซึ่งมีความช้าเมื่อเทียบกับ NVIDIA Jetson Nano ที่รองรับได้มากถึง 1000 Mb/s

กล้อง เป็นหนึ่งในอุปกรณ์สำคัญสำหรับการดำเนินโครงงาน เนื่องจากกล้องจะทำหน้าที่รับภาพที่พบเห็นจากด้านหน้า โดยที่กล้องนั้นสามารถแยกแยะว่าวัตถุที่อยู่ข้างหน้านั้นเป็นวัตถุชนิดใด หากไม่มีกล้องจะทำให้เกิดความยุ่งยากเนื่องจากอุปกรณ์ เช่น บอร์ด Raspberry Pi รุ่น 3 B และ RPLIDAR A1M8 นั้นไม่สามารถจำแนกวัตถุได้ ทางคณะผู้จัดทำได้ทำการเลือกกล้อง Logitech c922 pro โดยที่กล้องนั้นมีคุณสมบัติ ดังต่อไปนี้

2.1.5) กล้อง Logitech c922 pro

กล้อง Logitech c922 pro ดังรูปที่ 2.10 นั้นมีความสามารถในด้านสตรีมและบันทึกวิดีโอที่สดใสและสมจริง เลนส์กระจกและ Full HD 1080p จะทำให้บันทึกรายละเอียดที่ได้ดี ภาพและวิดีโอที่ได้มีความเป็นธรรมชาติและลื่นไหล ที่ 30fps ขณะที่มุมมองภาพ 78 องศาสามารถครอบคลุมคนได้ถึงสองคน และ c922 มีระบบที่สามารถโฟกัสอัตโนมัติและปรับแก้แสง HD ที่ปรับเปลี่ยนตามสภาพแสงเพื่อให้ภาพที่คุณภาพสูง สามารถสร้างตารางเพื่อศึกษาคุณสมบัติทางเทคนิคอื่น ๆ ของกล้อง Logitech c922 pro ดังตารางที่ 2.4

รูปภาพที่ 2.10 กล้อง Logitech c922 pro
(ที่มา : https://www.logitech.com/th-th/product/c922-pro-stream-webcam)

ตารางที่ 2.4 คุณสมบัติของ กล้อง Logitech c922 pro [12]

Brand	Logitech
Model	C922 PRO STREAM WEBCAM
Dimension	44 มม. x 95 มม. x 71 มม.
Weight	162 กรัม
Lens	กระจก Full HD มุมมองภาพ 78°
Resolution	1080p/30fps - 720p/60fps
Focus range	โฟกัสอัตโนมัติ
Cable Length	1.5 ม.
System Requirements	Windows® 10 หรือรุ่นใหม่กว่า, Windows 8, Windows 7

ในส่วนของ Open CV นั้นมีความสำคัญในการดำเนินโครงงานเนื่องจากเป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเป้าหมายไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์

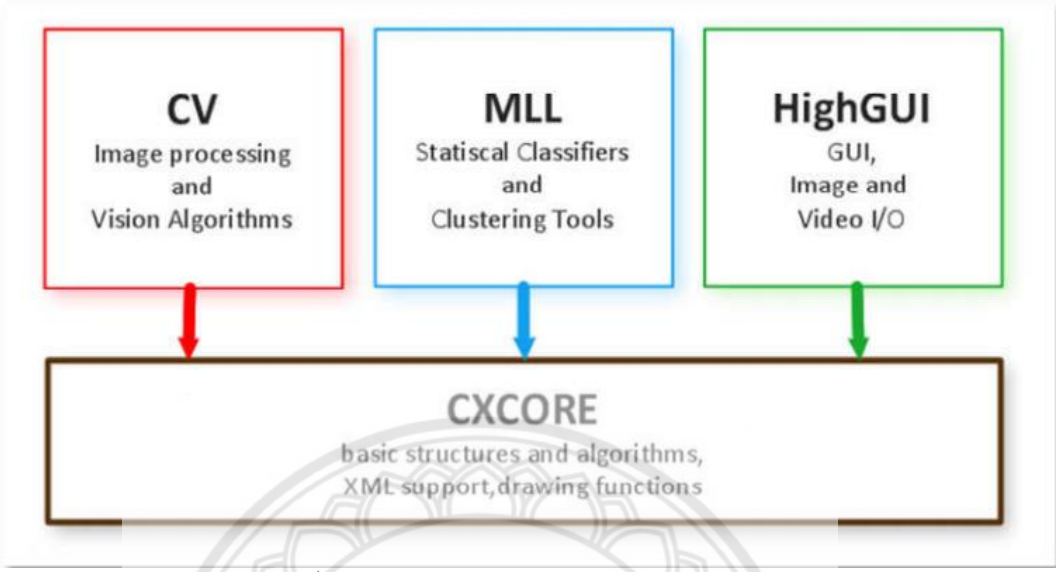
2.1.6) Open CV

Open CV มาจากคำว่า Open Source Computer Vision Library เป็นไลบรารี่ ที่ถูกพัฒนา โดยบริษัท Intel เพื่อใช้ในการประมวลภาพ (Image Processing) และงานด้านการมองของคอมพิวเตอร์(Computer Vision) เช่นความสามารถในการทำภาพเบลอ ความสามารถในด้าน เทรสซูลด (Threshold) หรือหาความแตกต่างกับจุดที่เราสนใจหรือตั้งสมมติฐานไว้ว่าจะเลือกหรือลบ เช่น ในภาพหนึ่งภาพ กำหนดค่าเทรสซูลดเป็น 100 (กระบวนการหาค่าเทรสซูลดมีความสำคัญ มีผลต่อคุณภาพของภาพ) ถาจุดพิกเซลที่สนใจมีค่าต่ำกว่าค่าเทรสซูลด คือ 100 ให้เปลี่ยน จุดพิกเซลนั้นเป็นสีขาว ในทางกลับกันถ้าค่ามากกว่า 100 ให้เปลี่ยนเป็นสีดำ

ความสามารถในด้านฮิสโตเกรม (Histogram) หรือหาค่าความแตกต่างของสีจากจุดอ้างอิงกลางความสามารถค้นหาขอบของภาพ ความสามารถด้านการตรวจสอบภาพเคลื่อนไหวหรือข้อมูลแบบวิดีโอ

Open CV ถูกพัฒนาขึ้นด้วยภาษา C และ C++ รองรับงานลักษณะมัลติ-คอร (Multi-Core) เมื่อต้องการใช้งานต้องเขียนโปรแกรมเพื่อเรียกใช้ Open CV จึงไมยึดติดกับระบบปฏิบัติการใดๆ ไมยึดติดกับโปรแกรมที่เลือกใช้ ภาษาที่นิยมเขียนโปรแกรมและเรียนใช้ไลบรารี่ Open CV เช่น C, C++, Python, Mathlab, Ruby

Open CV จัดแบ่งโครงสร้างภายในเป็นสี่ส่วนใหญ่ ๆ ดังรูปที่ 2.11



รูปภาพที่ 2.11 การแบ่งโครงสร้างภายในของ Open CV
(ที่มา : ระบบตรวจจับและพิสูจน์อัตลักษณ์ใบหน้าเพื่อคัดกรองบุคคลบริเวณพื้นที่ส่วนตัวด้วยบอร ดราสเบอรรี่ พาย/,2559)

1) CV : ประกอบไปด้วย อัลกอริทึมทางด้านการประมวลผลภาพและวิเคราะห์รูปภาพ (ข้อมูล 2 มิติ) การประมวลผลภาพเคลื่อนไหว การรู้จำ ฟังก์ชันที่สนใจ เช่น การหาขอบหรือมุมภาพ การทำHistogram
2) MLL : หรือ Machine Learning ประกอบไปด้วยฟังก์ชั่นทางสถิติ การแยกคลาส การแบงกลุ่มข้อมูล เช่น การทำ Clustering, Classification ฟังก์ชั่นการวิเคราะห์ข้อมูล การคำนวณทางสถิติ
3) HIGHGUI : เป็นฟังก์ชั่นเน้นทางสื่อสารกับลูกค้า เกี่ยวกับระบบ I/O เช่น การโหลดภาพ บันทึกภาพติดต่อกับกลองวิดีโอ สร้างหน้าต่าง แสดงภาพ ตรวจสอบเมาส แป้นพิมพ์
4) CXCORE : เป็นฟังก์ชั่นที่อธิบายถึงโครงสร้างข้อมูล (Data Structure) เช่น ขนาดอาเรย หน่วยความจำคำสั่งวาดภาพ ประกาศตัวแปร จัดการข้อผิดพลาด การแสดงข้อความ และฟังก์ชั่นทางคณิตศาสตร์ต่าง ๆ

การนำไปใช้ประโยชน์

1) ชุดเครื่องมือคุณลักษณะ 2 มิติและ 3 มิติ (2D and 3D feature toolkits)
2) การประมาณระยะในขณะเคลื่อนที่ (Egomotion Estimation)
3) ระบบรู้จำใบหน้า (Facial recognition system)

4) การจดจำท่าทาง (Gesture recognition)

5) ปฏิสัมพันธ์ระหว่างมนุษย์และคอมพิวเตอร์ (Human-Computer interaction; HCI)

การประมวลผลภาพ (Image Processing) หมายถึง การนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่ต้องการทั้งในเชิงคุณภาพและปริมาณ โดยมีขั้นตอนต่าง ๆ ที่สำคัญ คือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่สนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนของวัตถุในภาพ จากนั้นนำข้อมูลเชิงปริมาณเหล่านี้ไปวิเคราะห์ และสร้างเป็นระบบ เพื่อใช้ประโยชน์ในงานด้านต่างๆ เช่น ระบบตรวจสอบคุณภาพของผลิตภัณฑ์ในกระบวนการผลิตของโรงงานอุตสาหกรรม ระบบคัดแยกคุณภาพของพืชผลทางการเกษตร ระบบเก็บข้อมูลรถที่เข้าและออกอาคารโดยใช้ภาพถ่ายของป้ายทะเบียนรถเพื่อประโยชน์ในด้านความปลอดภัย ระบบดูแลและตรวจสอบสภาพการจราจรบนท้องถนนโดยการนับจำนวนรถบนท้องถนนในภาพถ่ายด้วยกล้องวงจรปิดในแต่ละช่วงเวลา เป็นต้น จะเห็นได้ว่าระบบเหล่านี้จำเป็นต้องมีการประมวลผลภาพจำนวนมาก และเป็นกระบวนการที่ต้องทำซ้ำ ๆ กันในรูปแบบเดิมเป็นส่วนใหญ่ ซึ่งงานในลักษณะเหล่านี้ หากให้มนุษย์วิเคราะห์เอง มักต้องใช้เวลามากและใช้แรงงานสูง จึงทำให้ผู้วิเคราะห์ภาพอาจเกิดอาการล้า ส่งผลให้เกิดความผิดพลาดขึ้นได้ ดังนั้นคอมพิวเตอร์จึงมีบทบาทสำคัญในการทำหน้าที่เหล่านี้แทนมนุษย์

2.2. วรรณกรรมปริทัศน์

จากการที่คณะผู้จัดทำได้ทำการศึกษาเอกสารงานวิจัยที่มีความเกี่ยวข้องกับ ระบบการแจ้งเตือนการชนในรถจักรยานยนต์ โดยทางคณะผู้จัดทำได้นำเสนอเอกสารงานวิจัยที่เกี่ยวข้องตามลำดับดังนี้

2.2.1) ระบบขนส่งอัจฉริยะสำหรับรถจักรยานยนต์เพื่อการแก้ปัญหาทางด้านความปลอดภัย

Kamarudin Ambak และคณะ ในปี 2009 ได้มีการศึกษาหัวข้อที่มีความเกี่ยวข้องกับระบบการแจ้งเตือนและหลีกเลี่ยงการชน โดยการศึกษาผลการรายงานของ Dagan (2004) จากการศึกษาของ Daimler-Benz ในปี 1992 ถ้าผู้ขับขี่รถยนต์มีเวลาการแจ้งเตือน 0.5 วินาที ก่อนการชนประมาณ 60% สามารถป้องกันการชนกันท้ายของรถคันข้างหน้าได้ แต่ถ้ามีเวลาเตือนเพิ่มเติมสามารถป้องกันประมาณ 90% ของการชนท้ายรถคันข้างหน้าได้ ระบบเตือนการชนด้านหน้าจะเข้าไปช่วยระบบความปลอดภัยที่ใช้งานอยู่โดยจะทำงานลักษณะแบบแฝงตัวอยู่ในระบบความปลอดภัย ระบบเตือนการชนด้านหน้าพยายามที่จะตรวจสอบความเสี่ยงการชน ระหว่างยานพาหนะสองคันโดยใช้เรดาร์และข้อมูลเซ็นเซอร์ภายในยานพาหนะ หากระบบตรวจพบความเสี่ยงต่อการชนระบบจะ

เตือนผู้ขับขี่เพื่อหลีกเลี่ยงอุบัติเหตุที่อาจเกิดขึ้น (Ararat และคณะ 2549) Ararat ระบุว่า "Tunable Avoidance Parameter" (TAP) เป็นอัลกอริทึมเพื่อให้มีความยืดหยุ่น ในอัลกอริทึมเพื่อให้มีความยืดหยุ่นมากขึ้น อัลกอริทึมที่เสนอได้รับการทดสอบโดยใช้ MEKAR-ACC Simulator พร้อมกับอัลกอริทึมอื่น ๆ สำหรับสถานการณ์ที่ระบุ อัลกอริธึมที่เสนอให้การเตือนที่สมจริงกว่าอัลกอริธึมการเตือนที่อิงความเร็วอื่น ๆ

Tsung-Ying และคณะ (2005) เสนอภาพรวมของการรับรู้ด้วยสายตาและการตัดสินใจที่คลุมเครือในการพัฒนาระบบการชนยานพาหนะอัจฉริยะ (IVCAS) ใน IVCAS มีการติดตั้งกล้อง CCD บนยานพาหนะและใช้ในการจับภาพยานพาหนะข้างหน้าและข้อมูลถนน ข้อได้เปรียบหลักของ IVCAS คือการใช้ fuzzy rules than other Systems และได้รับประสิทธิภาพที่เหนือกว่าในการหลีกเลี่ยงการชนของยานพาหนะ

Bayly และคณะ (2006) ทำการการรวบรวมข้องมูลของ collision warning and avoidance systems มีวัตถุประสงค์เพื่อตรวจสอบตำแหน่งของรถจักรยานยนต์และยานพาหนะอื่น ๆ เพื่อแจ้งเตือนหากยานพาหนะอยู่ในบริเวณใกล้เคียงที่อาจเป็นอันตราย ระบบเหล่านี้ยังไม่ได้นำมาใช้กับรถจักรยานยนต์และในความเป็นจริงก็เกิดขึ้นในยานพาหนะอื่นเท่านั้น อย่างไรก็ตาม ระบบเตือนการชนด้านได้นำถูกนำไปใช้กับ Yamaha ASV-2

2.2.2) เวลาในการตอบสนองการรับรู้ของผู้ขับขี่รถจักรยานยนต์สำหรับการหยุดรถในสถานการณ์ระยะสายตา

Seyed Rasoul Davoodi และคณะ ในปี 2012 มีการศึกษาเกี่ยวกับเวลาในตอบสนองของผู้ขับขี่รถจักรยานยนต์คือเวลาตั้งแต่ผู้ขับขี่เห็นวัตถุจนกระทั่งผู้ขับขี่ลดความเร็วของรถด้วยการเบรก โดยทำการทดลองบนถนนสองครั้งเพื่อให้ได้ค่าที่ชัดเจนของเวลาในการตอบสนองของผู้ขับขี่รถจักรยานยนต์ สำหรับเหตุการณ์ที่มีการคาดการณ์ว่าจะเกิดขึ้นจริงและเหตุการณ์ที่ไม่คาดคิดว่าจะเกิดขึ้น โดยสำหรับเหตุการณ์ที่มีการคาดการณ์ว่าจะเกิดขึ้นจริง ผู้ขับขี่รถจักรยานยนต์จำนวน 89 คน เหยียบเบรกทันทีเมื่อพบวัตถุ และในเหตุการณ์ที่ไม่คาดคิดว่าจะเกิดขึ้น ผู้ขับขี่ 16 คน ตอบสนองด้วยการเบรกแบบกะทันหันทันทีที่พบเห็นสิ่งกีดขวางบนทางของพวกเขา โดยที่ค่าเฉลี่ยของเวลาในการตอบสนองสำหรับเหตุการณ์ที่มีการคาดการณ์ว่าจะเกิดขึ้นจริง คือ 0.71 วินาที และ ในเหตุการณ์ที่ไม่คาดคิดว่าจะเกิดขึ้น คือ 1.25 วินาที สำหรับเหตุการณ์ที่ไม่คาดคิดว่าจะเกิดขึ้น ผู้ขับขี่ใช้เวลาในการตอบสนองอยู่ที่ 2.12 วินาที คิดเป็น 85 เปอร์เซ็นต์ ของผู้เข้าทดลองทั้งหมด การศึกษาครั้งนี้พบว่าผู้ขับขี่ส่วนใหญ่มีความสามารถในการตอบสนองต่อวัตถุที่ไม่คาดคิดตามถนนใน 2.5 วินาทีหรือน้อยกว่า

ดังนั้น เวลาในการตอบสนองของผู้ขับขี่จึงเป็น 2.5 วินาที ซึ่งเป็นค่าที่เหมาะสมสำหรับการออกแบบทางเรขาคณิตของเลนรถจักรยานยนต์

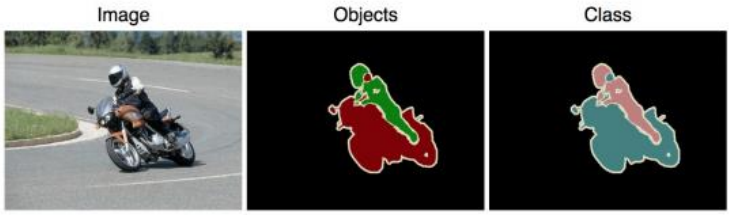
2.2.3) การแบ่งส่วนของภาพโดยการนำ Deep Learning เข้ามาช่วย

Shervin Minaee และคณะ ในปี 2020 ได้ทำการศึกษา Image Segmentation ที่ใช้กันอย่างแพร่หลาย สามารถแบ่งออกได้เป็น 3 ประเภท ได้แก่ ภาพ 2D, 2.5D RGB-D และภาพ 3 มิติ ให้รายละเอียด เกี่ยวกับลักษณะของแต่ละชุดข้อมูล ชุดข้อมูลที่ระบุไว้จะมีป้ายกำกับพิกเซลที่ซึ่งสามารถใช้สำหรับการประเมินผลและประสิทธิภาพของแบบจำลอง

ข้อมูลนี้ค่อนข้างคุ้มค่าที่จะกล่าวถึง การใช้การเสริมข้อมูล (data augmentation) ของตัวอย่างที่มีข้อความ ข้อความชุดพิเศษที่จัดการกับชุดข้อมูลขนาดเล็ก เช่น ในขอบเขตทางการแพทย์ การเสริมข้อมูลทำหน้าที่เพิ่มจำนวนตัวอย่างการฝึกอบรมโดยใช้ชุดข้อมูล ไปเป็นรูปภาพ การเปลี่ยนแปลงทั่วไป ประกอบไปด้วย การแปลง, การสะท้อน, การหมุน, การแปรปรวน, การปรับขนาด , การเปลี่ยนพื้นที่ของสี, การตัดและการฉายภาพลงบนส่วนประกอบที่สำคัญ การเสริมข้อได้มีการพิสูจน์แล้วว่าช่วยเพิ่มประสิทธิภาพของโมเดล โดยเฉพาะเมื่อเรียนรู้จาดชุดข้อมูลที่จำกัด เช่น การวิเคราะห์ภาพทางการแพทย์ และยังสามารถใช้ประโยชน์ในการยอมให้การลู่เข้าเร็วขึ้นลดโอกาสที่จะเกิดข้อผิดพลาดมากเกินไปสำหรับชุดข้อมูลขนาดเล็กบางส่วนจะมีการแสดงข้อมูลเพิ่มเติมเพื่อเพิ่มประสิทธิภาพของโมเดลมากกว่า 20% ของการแปลง

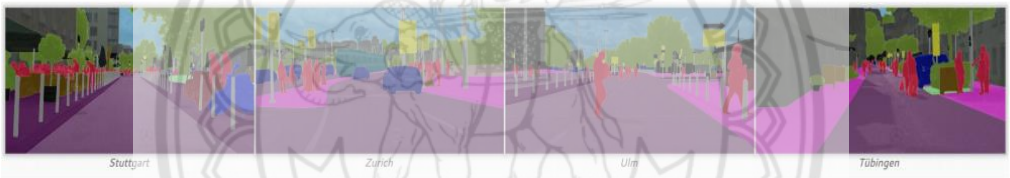
a.) 2D Datasets งานวิจัยส่วนใหญ่ของ IMAGE SEGMENTATION จะเน้นที่ภาพ 2D ดังนั้นชุดข้อมูลการ IMAGE SEGMENTATION แบบ 2D มีอยู่จำนวนมากมาย แต่ชนิดต่อไปนี้คือส่วนที่ได้รับความนิยมใช้กันมากที่สุดคือ

a.1) PASCAL Visual Object Classes (VOC) ซึ่งเป็นหนึ่งในชุดข้อมูลภาพของคอมพิวเตอร์ที่ได้รับความนิยม โดยมีลักษณะการทำงานจะมีข้อความพิกเซลกำกับคำอธิบายประกอบภาพที่มีให้ โดยมี 5 ลักษณะการทำงาน ได้แก่ การจำแนก การแบ่งส่วน การตรวจจับ การจดจำการกระทำ และการจัดวางบุคคล ซึ่งอัลกอริทึมนี้สามารถแบ่งประเภทภาพออกได้ เป็น 21 ชนิด ได้แก่ ยานพาหนะ, ครัวเรือน, สัตว์, เครื่องบิน, จักรยาน, เรือ, รถบัส, ที่จอดรถ, รถมอเตอร์ไซด์, รถไฟ, ขวด, เก้าอี้, โต๊ะ, อาหาร, กระถางต้นไม้, โซฟา, ทีวีหรือจอมอนิเตอร์, นก, แมว, วัว, สุนัข, ม้า, แกะ และบุคคล (สำหลับชนิดของวัตถุจะมีข้อความพิกเซลกำกับเป็นพื้นหลังหากพวกเขาไม่ได้อยู่ใน 21 ชนิดที่กล่าวมาข้างต้น) ชุดข้อมูลนี้แบ่งออกเป็นสองชุดการทำงานคือ training และ inspection มีชนิดของภาพ 1,464 และ 1,449 ชนิดตามลำดับ มีลักษณะการแบ่งชนิดภาพที่ชัดเจนมีการแยกชนิดของวัตถุ จะได้ได้จากรูปที่ 2.12 แสดงตัวอย่างรูปภาพและป้ายกำกับพิกเซล



รูปภาพที่ 2.12 ตัวอย่างรูปภาพจากชุดข้อมูล PASCAL VOC

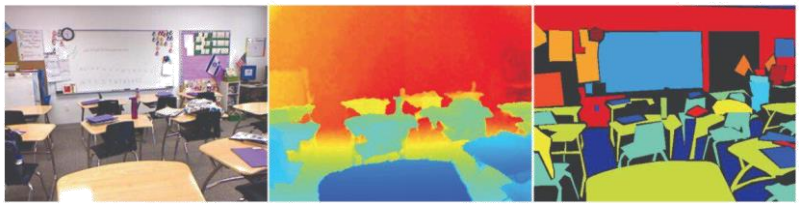
a.2) Cityscapes เป็นฐานข้อมูลขนาดใหญ่ที่ให้ความสำคัญกับความเข้าใจความหมายของภาพภายในเมือง มีชุดของวีดีโอที่บันทึกไว้ในฉากถนนในขณะขับขี่ พร้อมคำอธิบายประกอบระดับพิกเซลที่มีคุณภาพสูง แบ่งออกเป็น 8 หมวดหมู่ ได้แก่ พื้นผิวเรียบ, มนุษย์, ยานพาหนะ, สิ่งก่อสร้าง, วัตถุธรรมชาติ, ท้องฟ้า และพื้นที่ว่าง รูปที่ 2.13 แสดงแผนที่แบ่งกลุ่มตัวอย่างสี่ชุดจากชุดข้อมูลนี้



รูปภาพที่ 2.13 ภาพตัวอย่างสามภาพพร้อมแผนที่ที่เกี่ยวข้องจากชุดข้อมูล Cityscapes

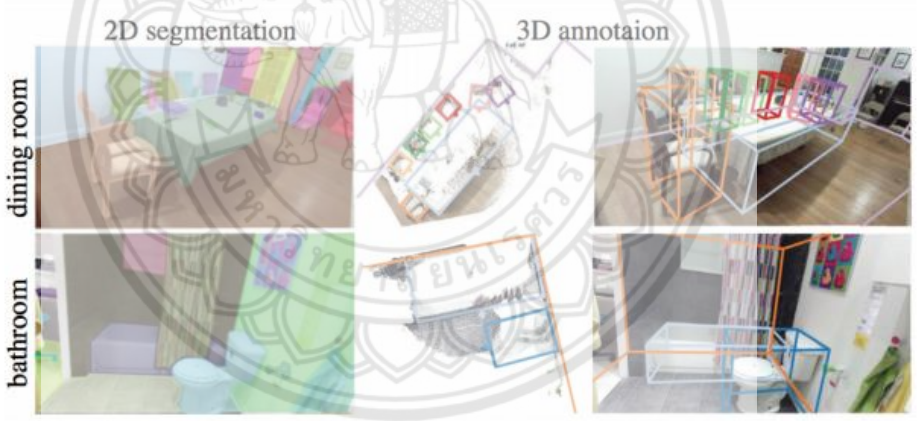
b.) 2.5D Datasets ด้วยความสามารถของสแกนเนอร์ที่มีราคาไม่แพงทำให้ภาพ RGB-D เป็นที่นิยมในงานวิจัยและอุตสาหกรรม ชุดข้อมูล RGB-D ต่อไปนี้เป็นที่นิยมใช้กันมากอุตสาหกรรม

b.1) NYU-D V2 คือลำดับวิดีโอจากฉากในอาคารโดยบันทึกด้วยกล้อง RGB และกล้องความลึกของ Microsoft Kinect แต่ละวัตถุจะมีป้ายกำกับที่มีชนิดและตัวอย่างตัวเลข (เช่น cup1, cup2, cup3 ฯลฯ) นอกจากนี้ยังมีเฟรมที่ไม่มีป้ายกำกับ 407,024 ชุดข้อมูลนี้ค่อนข้างเล็กเมื่อเทียบกับชุดข้อมูลอื่นที่มีอยู่ รูปที่ 2.14 แสดงภาพตัวอย่างและแผนที่การแบ่งส่วนภาพ (segmentation)



รูปภาพที่ 2.14 แสดงภาพตัวอย่างและแผนที่การแบ่งส่วนภาพ (segmentation)

b.2) SUN RGB-D คือการกำหนดมาตรฐาน RGB-D ของเป้าหมายเพื่อให้ภาพที่ได้จากเซ็นเซอร์มีความสมจริงใช้ในการประมวลผลเพื่อการทำความเข้าใจกับวัตถุในภาพที่สำคัญทั้งหมด วัตถุจะถูกจับโดยเซ็นเซอร์สี่ตัวที่แตกต่างกันและมี 10,000 ภาพ RGB-D ในลักษณะการทำงานคล้ายกับ PASCAL VOC โดยชุดข้อมูลทั้งหมดมีการใส่ชนิดของวัตถุประกอบภาพหรือวิดีโอที่กล้องบันทึกได้และยังรวม 146,617 2D polygons และ 58,657 3D bounding box ด้วยการจัดวางวัตถุที่แม่นยำรวมถึงประเภทห้องและการจัดวางสำหรับภาพ รูปที่ 2.15 แสดงภาพตัวอย่างสองภาพ



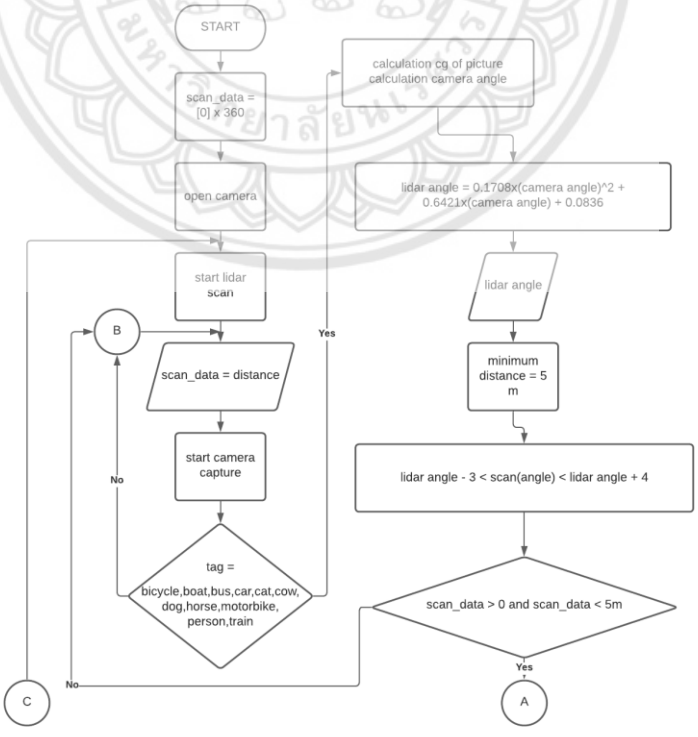
รูปภาพที่ 2.15 แสดงภาพตัวอย่างสองภาพ ของ SUN RGB-D

บทที่ 3

วิธีการดำเนินการ

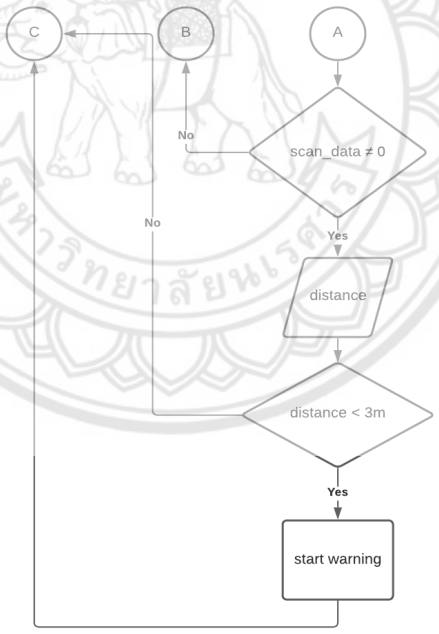
3.1. ขั้นตอนการดำเนินงาน

จากการศึกษาและค้นคว้าเกี่ยวกับอุปกรณ์และอัลกอริทึมที่ใช้ในการตรวจจับยานพาหนะ เพื่อที่จะสามารถสร้างอัลกอริทึมด้วยภาษา Python และศึกษาความสัมพันธ์ของกล้องและ lidar เพื่อนำมาใช้ในการสอบเทียบเครื่องมือวัด (Calibrate) และนำสมการจากการสอบเทียบเครื่องมือวัดมาใช้หาระยะทางจริงของภาพที่จับได้โดยกล้องจากข้อมูลระยะทางของ lidar และสร้างอัลกอริทึมสำหรับการแจ้งเตือนเมื่อถึงระยะที่กำหนด โดยการสร้างอัลกอริทึมสำหรับการแจ้งเตือนมีขั้นตอนการดำเนินงานตาม Flow chart ดังแสดงในรูปภาพที่ 3.1 และ 3.2 ตามลำดับ



รูปภาพที่ 3.1 Flow chart แสดงการสร้างอัลกอริทึมสำหรับการแจ้งเตือน

จากรูปภาพที่ 3.1 เริ่มต้นการสร้างอัลกอริทึมโดยกำหนดตัวแปรชื่อ scan_data ซึ่งจะมีค่าเท่า 0 จำนวน 360 ตัวเพื่อมารับข้อมูลระยะทางที่ได้จาก lidar ในแต่ละองศา จากนั้นทำการเปิดการทำงานของกล้อง แล้วสั่งให้ lidar เริ่มต้นการสแกนจะได้ค่าระยะทางไปเก็บไว้ในตัวแปร sacn_data แล้วสั่งให้กล้องทำการจับภาพเมื่อในภาพพบวัตถุที่มีชื่อ bicycle, boat, bus, car, cat , cow, dog, horse, motorbike, person, train จะนำข้อมูลจากภาพที่ได้ไปหาจุดกึ่งกลางภาพและคำนวณหามุมของภาพแต่ถ้าไม่ตรงกับเงื่อนไขจะย้อนกลับไปเริ่มทำงานไม่ตั้งแต่ lidar เริ่มแสกน จากนั้นนำมุมของภาพไปคำนวณหามุมของ lidar จากสมการที่ได้จากการ calibrate ดังแสดงในรูปที่ 3.1 จากนั้นกำหนดให้ระยะทางต่ำที่สุดที่จะนำค่าระยะทางมาใช้คือ 5 เมตร แล้วมุมของ lidar สแกนได้จะต้องอยู่ในช่วงของมุมของ lidar ที่คำนวณได้ลบ 3 และบวก 4 ดังแสดงในรูปที่ 3.1 จากนั้นเมื่อระยะทางที่ได้มากกว่า 0 และน้อยกว่า 5 เมตร จะนำระยะทางที่ได้ไปใช้ต่อ แต่ถ้าไม่ตรงกับเงื่อนไขจะย้อนกลับไปเริ่มทำงานไม่ตั้งแต่ lidar เริ่มแสกน

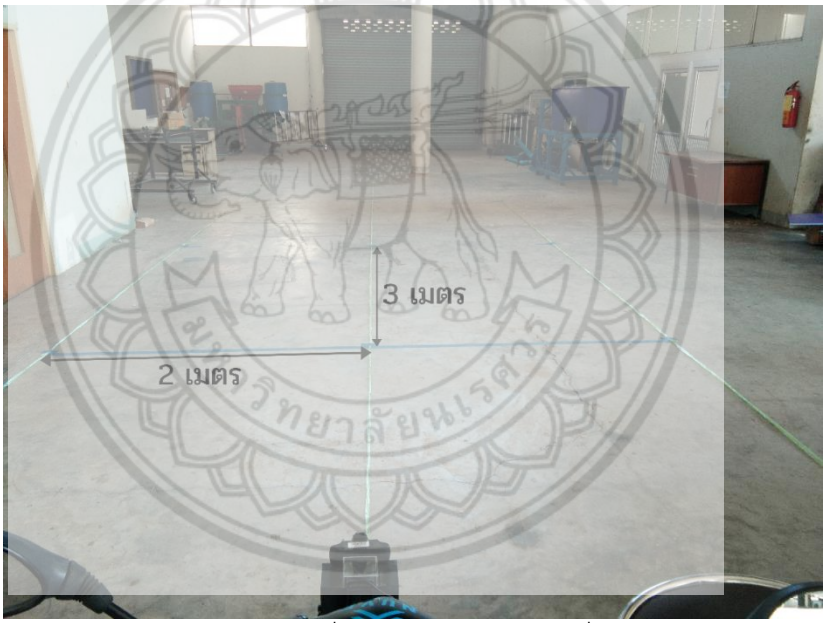


รูปภาพที่ 3.2 Flow chart แสดงการสร้างอัลกอริทึมสำหรับการแจ้งเตือน

จากรูปภาพที่ 3.2 เมื่อได้ข้อมูลระยะทางมาและมีค่าไม่เท่ากับ 0 จะได้ค่าระยะทางของวัตถุที่ตรวจจับได้แต่ถ้าไม่ตรงกับเงื่อนไขจะย้อนกลับไปเริ่มทำงานไม่ตั้งแต่ lidar เริ่มแสกน เมื่อระยะทางที่ได้มีค่ามากกว่า 3 m จะทำการเริ่มทำงานใหม่ตั้งแต่ต้นอีกครั้ง แต่ถ้าหากระยะทางที่ได้มีค่าน้อยกว่า 3 m จะทำการแจ้งเตือน และจะวนซ้ำการทำงานใหม่ตั้งแต่ต้นอีกครั้ง

3.2. การหาความสัมพันธ์ระหว่างมุมของวัตถุในภาพกับเซนเซอร์ลิดาร์

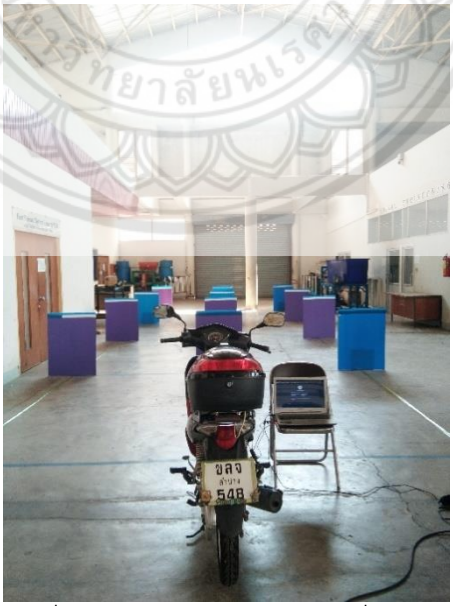
สำหรับการนำข้อมูลระยะห่างที่เซนเซอร์ลิดาร์สามารถตรวจจับได้จำเป็นต้องทำการสอบเทียบ (calibrate) ซึ่งมีความเกี่ยวข้องกับการหาความสัมพันธ์ที่เกิดขึ้นระหว่างมุมวัตถุภายในภาพที่ได้รับข้อมูลมาจากกล้องและมุมของเซนเซอร์ลิดาร์ที่ได้เก็บข้อมูลในขณะนั้น การนำค่าผลลัพธ์ที่ได้เหล่านั้นมาเปรียบเทียบเพื่อหาความสอดคล้องว่าจะมีความสัมพันธ์กันเป็นสมการใด จากนั้นสามารถบอกระยะห่างจริงของวัตถุในภาพว่ามีระยะห่างเป็นเท่าไหร่จากเซนเซอร์ลิดาร์ เริ่มทำการสอบเทียบโดยการเตรียมสถานที่ กำหนดระยะทางที่เซนเซอร์ลิดาร์รุ่น A1M8 จะสามารถตรวจจับได้อยู่ที่ 12 เมตร โดยจะแบ่งเป็นช่วงละ 3 เมตร กำหนดให้วางตำแหน่งรถจักรยานยนต์ให้อยู่บริเวณกึ่งกลางแบ่งช่วงละ 2 เมตร และนำเชือกพร้อมกับเทปกาวไปติดตั้งไว้ที่ตำแหน่งที่ได้ทำการวัดไว้ดังแสดงในรูปภาพที่ 3.3



รูปภาพที่ 3.3 การเตรียมสถานที่

ทำการออกแบบชิ้นงานสำหรับรองรับการติดตั้งกล้อง บอร์ด raspberry pi และเซนเซอร์ลิดาร์ดังแสดงในรูปภาพที่ 3.4 โดยที่ชิ้นงานนี้จะต้องยึดอุปกรณ์ทั้ง 3 ชิ้นให้มีการเคลื่อนที่น้อยที่สุดเพื่อป้องกันการคลาดเคลื่อนระหว่างการสอบเทียบหากอุปกรณ์มีการขยับ จากนั้นนำไปติดตั้งเข้ากับรถจักรยานยนต์โดยให้อุปกรณ์อยู่บริเวณด้านหน้าของรถจักรยานยนต์ดังแสดงในรูปที่ 3.4 ทำการเริ่มเก็บข้อมูลโดยนำวัตถุไปวางไว้ตามตำแหน่งที่ได้ทำการกำหนดไว้ดังแสดงในรูปที่ 3.5

รูปภาพที่ 3.4 การติดตั้งอุปกรณ์กับรถจักรยานยนต์

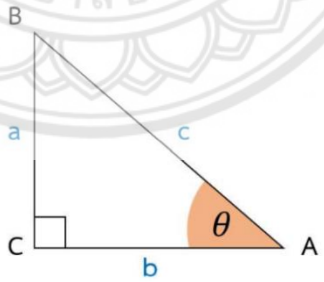


รูปภาพที่ 3.5 ตำแหน่งการวางวัตถุเพื่อเก็บข้อมูล

เริ่มทำการเก็บข้อมูลโดยการใช้โปรแกรม matlab เข้ามาช่วย โดยการจับภาพที่ได้จากกล้องส่งไปที่โปรแกรม matlab เพื่อทำการวัดหาระยะพิกเซลภายในภาพจากด้านล่างขอบของภาพจนถึงระนาบของการตรวจจับที่เซนเซอร์ลิดาร์สามารถตรวจจับได้บนวัตถุ ซึ่งจะอยู่บริเวณกึ่งกลางของภาพ ดังแสดงในรูปภาพที่ 3.6 โดยให้เริ่มทำการเก็บข้อมูลจากด้านหน้าสุดด้านซ้ายมือระยะ 3 เมตรก่อน และทำซ้ำ 5 ครั้งต่อวัตถุ 1 ชิ้น จนครบทั้งหมด 9 วัตถุ จะทำให้ได้ข้อมูลทั้งหมด 45 ข้อมูล

รูปภาพที่ 3.6 การวัดระยะพิกเซลภาพของ a และ b

นำระยะพิกเซลที่วัดได้จากภาพมาคำนวณเพื่อหาระยะของมุมที่เกิดขึ้นภายในภาพโดยใช้ความรู้ตรีโกณมิติเข้ามาช่วยคำนวณดังนี้



รูปภาพที่ 3.7 สามเหลี่ยมมุมฉากตรีโกณมิติ

รูปภาพที่ 3.7 สามารถนำไปหาค่ามุม θ ได้จากสมการอัตราส่วนตรีโกณมิติ ในสมการที่ (3.1)

$$\theta(rad) = tan^{-1}(\frac{a}{b}) \tag{3.1}$$

เมื่อ a คือ ความยาวด้านตรงข้ามมุม

b คือ ความยาวด้านประชิดมุม

จากสมการที่ (3.1) สามารถที่จะหาค่าของมุมที่เกิดขึ้นในภาพได้ดังแสดงในตารางที่ 3.1 โดยที่ในในตารางเป็นข้อมูลของวัตถุด้านซ้ายที่ระยะ 3 เมตร โดยที่ a คือระยะพิกเซลตั้งแต่ขอบล่างของภาพจนถึงจุดกึ่งกลางของวัตถุบริเวณกึ่งกลางภาพ b คือระยะทางตั้งแต่จุดกึ่งกลางของวัตถุที่วัดจนถึงจุดกึ่งกลางวัตถุบริเวณกึ่งกลางของภาพ เมื่อนำค่า a และ b ไปแทนในสมการที่ (3.1) จะได้ค่ามุมของภาพออกมา และจะทราบมุมของลิดาร์จากข้อมูลที่ได้จากการทำงานของลิดาร์

ตารางที่ 3.1 วัตถุด้านซ้ายที่ระยะ 3 เมตร

ครั้งที่	ระยะ a	ระยะ b	มุมจากภาพ (R)	มุมจาก Lidar (R)
1	250.64	394.65	1.0050	0.7817
2	250.63	387.52	0.9967	0.7694
3	250.5	387.02	0.9964	0.7694
4	250.63	386.9	0.9960	0.7653
5	250.75	386.53	0.9953	0.7604

เมื่อนำข้อมูลระยะของมุมจากภาพที่คำนวณได้มาทำการเปรียบเทียบกับมุมของลิดาร์ที่อ่านได้จากข้อมูลที่เก็บได้จากเซ็นเซอร์ลิดาร์ จะได้ความสัมพันธ์ที่เกิดขึ้นดังแสดงในสมการที่ (3.2)

$$y = 0.1708x^2 + 0.6421x - 0.0836 \quad (3.2)$$

เมื่อ x คือ ขนาดของมุมในภาพ

y คือ ขนาดของมุมเสมือนของลิดาร์

เมื่อทำการนำขนาดของมุมเสมือนที่คำนวณได้จากสมการที่ (3.2) มาเปรียบเทียบกับมุมที่ได้จากข้อมูลของเซ็นเซอร์ลิดาร์จะทำให้ทราบระยะห่างระหว่างวัตถุและเซ็นเซอร์ลิดาร์ได้

3.3. การตรวจจับและจำแนกวัตถุสำหรับ Open CV

จากความก้าวหน้าของเทคโนโลยีในปัจจุบันสามารถพัฒนาอัลกอริทึมสำหรับตรวจจับวัตถุและจำแนกชนิดของวัตถุ ทำให้นำมาใช้ประโยชน์กับระบบในรถยนต์สำหรับตรวจจับวัตถุบนท้องถนนหรือสามารถนำไปประยุกต์ใช้สำหรับการตรวจจับในด้านการจราจรในที่ชุมชน โดยการพัฒนา

อัลกอริทึม Open CV สำหรับโครงงานระบบการแจ้งเตือนการชนในรถจักรยานยนต์ที่ทางคณะผู้จัดทำได้ดำเนินการมีส่วนเกี่ยวข้องกับการนำ Open CV เข้ามาช่วยสำหรับการตรวจจับวัตถุบนท้องถนนและทำการจำแนกว่าเป็นวัตถุประเภทใด

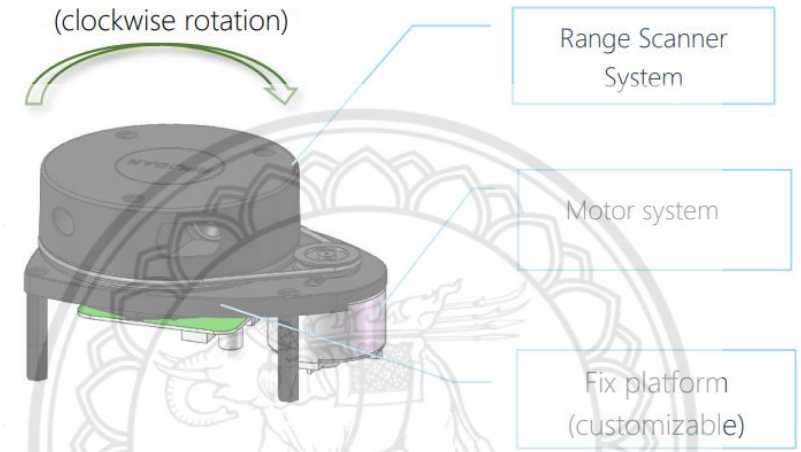
ในปัจจุบันโมเดลที่ใช้นั้นมีหลากหลาย และมีความน่าสนใจที่แตกต่างกันไป สำหรับทางคณะผู้จัดทำได้ทำการเลือกโมเดล MobileNet สำหรับใช้ในการพัฒนาอัลกอริทึมให้เหมาะสมกับระบบ MWS โดยการนำเอาโมเดลสำเร็จรูปของ MobileNet ที่มีในระบบออนไลน์มาปรับปรุงและแก้ไขเพื่อให้ได้ข้อมูลตามที่ต้องการ โดยข้อมูลที่ต้องการนั้นจะเกี่ยวข้องกับชนิดของวัตถุ ค่าความแม่นยำในการตรวจจับ ค่าแกน X, Y ค่าความกว้าง และค่าความสูงของกรอบที่เกิดขึ้นจากการตรวจจับวัตถุดังแสดงดังรูปที่ 3.8 เพื่อไปใส่ในการคำนวณในสมการที่ (3.2) ซึ่งเป็นสมการที่ได้จากการ calibrate ที่ช่วยใช้สำหรับการหามุมของภาพที่ตรวจจับได้

รูปภาพที่ 3.8 การทำงานของโมดูล MobileNet

ข้อมูลที่ได้ออกมาจะแสดงถึงมุมภาพที่แท้จริงที่เกิดขึ้นจากการเขียนโปรแกรมโดยใช้ภาษา python จากการเขียนโปรแกรมและสั่งให้เริ่มการทำงานสามารถทำให้ได้การจำแนกชนิดวัตถุ, ความแม่นยำ, ตำแหน่งของ X, Y, ความกว้างของกรอบ และความสูงของกรอบ โดยจากรูปภาพที่ 3.8 จะแสดงเพียงการตีกรอบเพื่อจำแนกชนิดวัตถุที่กล้องสามารถจับได้ รวมทั้งยังสามารถระบุได้ว่าเป็นวัตถุชนิดใด และมีความความแม่นยำออกมาด้วยเช่นกัน

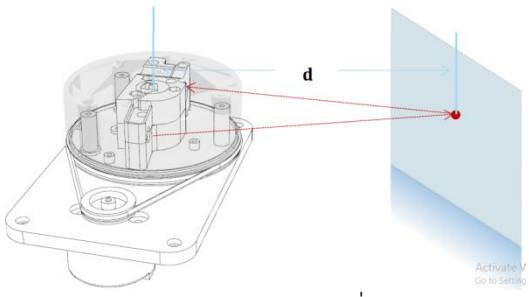
3.4. การตรวจจับวัตถุสำหรับ RPLIDAR รุ่น A1M8

จากความก้าวหน้าของเทคโนโลยีในปัจจุบันสามารถพัฒนาอุปกรณ์ที่สามารถตรวจจับและวัดระยะได้ โดยการทำงานของ RPLIDAR รุ่น A1M8 จะประกอบด้วยระบบสแกนเนอร์และระบบมอเตอร์ หลังจากเปิดเครื่องแต่ละระบบย่อยของ lidar จะเริ่มหมุนและสแกนทิศทางตามเข็มนาฬิกา ดังแสดงในรูปภาพที่ 3.9 โดยผู้ใช้งานอุปกรณ์จะได้รับช่วงสแกนข้อมูลผ่านอินเทอร์เฟซการสื่อสาร



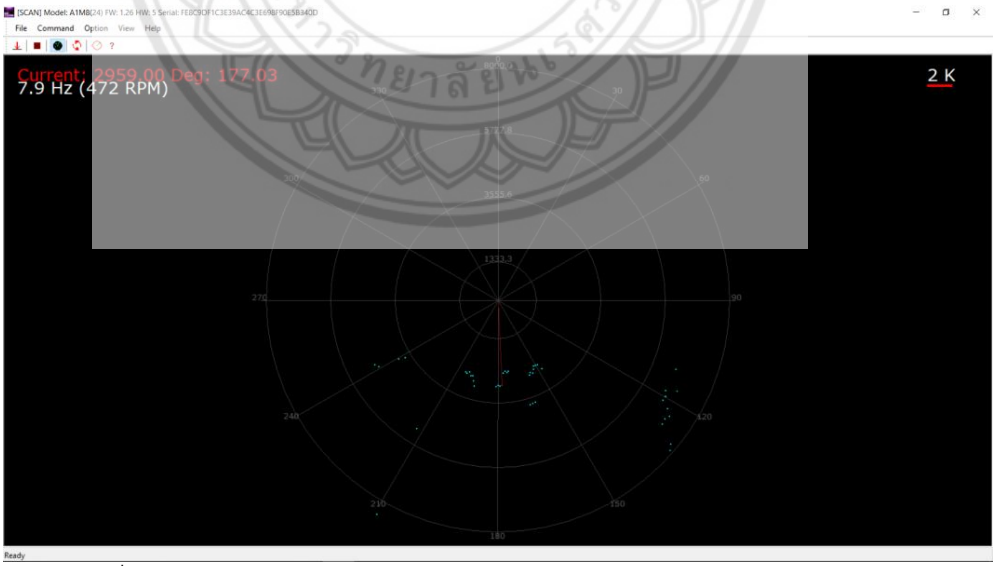
รูปภาพที่ 3.9 การสแกนทิศทางตามเข็มนาฬิกาของ RPLIDAR รุ่น A1M8

สำหรับการทำงานในด้านกลไกของ RPLIDAR รุ่น A1M8 เป็นอุปกรณ์ที่ขึ้นอยู่กับหลักการสามเหลี่ยมของเลเซอร์ และใช้ความเร็วสูงในการจับวิสัยทัศน์ และฮาร์ดแวร์ที่พัฒนาโดย SLAMTEC ข้อมูลที่ได้จากการวัดระยะมีจำนวนมากกว่า 8000 ครั้งต่อวินาที ด้วยความแม่นยำที่สูง lidar จะปล่อยสัญญาณเลเซอร์อินฟราเรดแบบมอดูเลตและสัญญาณเลเซอร์จะถูกส่งไปสะท้อนเข้ากับวัตถุแถวบริเวณที่ lidar สามารถตรวจจับได้ ดังรูปที่ 3.10 โดยวัตถุที่จะตรวจจับสัญญาณที่กลับมาจะถูกสุ่มตัวอย่างโดยระบบการมองเห็นใน lidar และ DSP ที่ฝังอยู่ภายใน lidar จะเริ่มต้นการประมวลผลข้อมูลตัวอย่างที่ตรวจจับได้ และให้ผลแสดงโชว์ออกมาเป็นค่าระยะทางและค่ามุมระหว่างวัตถุ



รูปภาพที่ 3.10 การปล่อยสัญญาณเลเซอร์อินฟราเรดเพื่อไปกระทบกับวัตถุของ RPLIDAR รุ่น A1M8

ในระหว่างกระบวนการทำงานของ RPLIDAR รุ่น A1M8 จะส่งออกข้อมูลการสุ่มตัวอย่างผ่านทางไฟล์อินเทอร์เฟซการสื่อสาร และข้อมูลจุดตัวอย่างแต่ละรายการจะประกอบไปด้วย ค่าระยะทางที่วัดได้ระหว่างการหมุนแกนกลางของ lidar และจุดสุ่มตัวอย่างมีหน่วยเป็นมิลลิเมตร มุมปัจจุบันของการวัด และคุณภาพของการวัด ที่เป็นการจำลองการทำงานผ่านฟังก์ชัน matplotlib ในภาษา python โดยใช้การเขียนโปรแกรมสำเร็จรูปจากการสืบค้นข้อมูล หรือการตรวจสอบผ่านคู่มือของทาง SLAMTEC ที่มีโปรแกรมสำเร็จรูปโดยเฉพาะให้ใช้งานมีชื่อว่า frame_grabber.exe ในที่นี้ทางคณะผู้จัดทำได้ทำการทดสอบประสิทธิภาพการตรวจจับของอุปกรณ์ RPLIDAR รุ่น A1M8 ผ่านโปรแกรม frame_grabber.exe ดังแสดงในรูปที่ 3.11



รูปภาพที่ 3.11 การทดสอบ RPLIDAR รุ่น A1M8 ผ่านโปรแกรม frame_grabber.exe

จากรูปภาพที่ 3.11 ที่ทำการทดสอบผ่านโปรแกรม frame_grabber.exe จะให้ข้อมูลของระยะที่ตรวจจับเจอวัตถุโดยรอบในหน่วยมิลลิเมตร ค่าของมุม และการจำลองการทำงานของ RPLIDAR รุ่น A1M8 ออกมาในรูปแบบของแผนภูมิวงกลม

จากการทราบข้อมูลการทำงานของ RPLIDAR รุ่น A1M8 นั้นยังไม่เพียงพอให้สามารถนำมาพัฒนาและปรับเปลี่ยนให้มีความเชื่อมโยงเข้ากับระบบการแจ้งเตือนในรถจักรยานยนต์ ทางคณะผู้จัดทำจำเป็นต้องคำนึงถึงระยะที่ใช้ในการเบรกในช่วงของความเร็วที่สนใจ โดยความเร็วที่ใช้คือ 20 - 30 กิโลเมตรต่อชั่วโมง แต่ในการเบรกแต่ละครั้งจะมีช่วงเวลาที่ใช้ในการตัดสินใจก่อนการเหยียบเบรกของผู้ขับขี่ ซึ่งจะส่งผลให้ระยะเบรกปลอดภัยนั้นมีค่าแตกต่างกัน การคำนวณหาระยะเบรกที่ปลอดภัยจำเป็นต้องใช้สมการที่(2.3) เมื่อคำนวณหาระยะเบรกปลอดภัยของความเร็ว 20, 25 และ 30 กิโลเมตรต่อชั่วโมงตามลำดับ ในช่วงระยะเวลาตัดสินใจตั้งแต่ 1 – 2.5 วินาที สามารถทำเป็นตารางสรุปผลการคำนวณได้ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 สรุปผลการคำนวณระยะเบรกปลอดภัยที่ความเร็ว 20 ถึง 30 กิโลเมตรต่อชั่วโมง ในช่วงเวลาตัดสินใจตั้งแต่ 1 ถึง 2.5 วินาที

ความเร็ว (กิโลเมตรต่อชั่วโมง)	เวลาที่ใช้ในการตัดสินใจ (วินาที)	ระยะทาง (เมตร)	ความเร่งขณะเบรก (เมตรต่อวินาที^{2})	ระยะทางที่ใช้ในการเบรก (เมตร)	ระยะทั้งหมดที่ใช้ในการเบรก (เมตร)
20	1	5.56	7.85	1.97	7.52
	1.25	6.94			8.91
	1.5	8.33			10.30
	1.75	9.72			11.69
	2	11.11			13.08
	2.25	12.50			14.47
	2.5	13.89			15.86
25	1	6.94		3.07	10.02
	1.25	8.68			11.75
	1.5	10.42			13.49
	1.75	12.15			15.23
	2	13.89			16.96
	2.25	15.63			18.70
	2.5	17.36			20.43
30	1	8.33		4.42	12.76
	1.25	10.42			14.84
	1.5	12.50			16.92
	1.75	14.58			19.01
	2	16.67			21.09
	2.25	18.75			23.17
	2.5	20.83			25.26

จากตารางที่ 3.2 จะแสดงให้เห็นถึงระยะเบรกปลอดภัยซึ่งแบ่งตามความเร็วของรถขณะกำลังเบรกกับระยะเวลาที่ใช้ในการตัดสินใจก่อนการเหยียบเบรก ซึ่งจะเห็นได้ว่าที่ความเร็วเดียวกันแต่ระยะเวลาที่ใช้ในการตัดสินใจก่อนการเหยียบเบรกต่างกันจะส่งผลให้ระยะทางที่ใช้ในการเหยียบเบรกจนกระทั่งรถหยุดนิ่งหรือระยะเบรกปลอดภัยนั้นแตกต่างกันด้วย โดยที่ความเร่งขณะเบรกจะส่งผลต่อระยะทางที่ใช้เบรก โดยคณะผู้จัดทำได้ใช้สัมประสิทธิ์แรงเสียดทานจลน์ระหว่างล้อรถกับพื้น

คอนกรีตในสภาพแวดล้อมปกติ จากตารางที่ 2.1 ซึ่งมีค่าเท่ากับ 0.80 ในการคำนวณหาความเร่งขณะเบรกได้ผลลัพธ์เท่ากับ 7.85 เมตรต่อวินาที^{2}

โดยจากการศึกษาหาข้อมูลระยะเวลาที่ใช้ในการเบรกของผู้ขับขี่จะขึ้นอยู่กับความสามารถในการขับขี่ของผู้ขับขี่ และสถานการณ์ในขณะนั้น ทำให้สามารถแบ่งระยะเวลาที่ใช้ในการเบรกอยู่ในช่วง 1 วินาที ถึง 2.5 วินาที เนื่องจากอุปกรณ์ RPLIDAR รุ่น A1M8 ที่ทางคณะผู้จัดทำได้เลือกใช้มีระยะการตรวจจับเพียงแค่ 12 เมตร จึงทำการเลือกใช้ระยะเวลาในการตัดสินใจที่ 1.5 วินาที สำหรับความเร็ว 25 กิโลเมตรต่อชั่วโมง มีระยะเบรกปลอดภัยที่ได้ประมาณ 13.49 เมตร ซึ่งค่ามีความใกล้เคียงกับระยะการตรวจจับของ RPLIDAR รุ่น A1M8

บทที่ 4

ผลการทดลองและการวิเคราะห์ผลการทดลอง

4.1 ผลการทดลอง

จากการสร้างระบบการแจ้งเตือนการชนในรถจักรยานยนต์โดยการสร้างอัลกอริทึมขึ้นมาด้วยภาษา python เพื่อนำไปใช้กับอุปกรณ์ที่ทางคณะผู้จัดทำได้เตรียมไว้ นั่นคือ กล้อง และ RPLIDAR รุ่น A1M8 คณะผู้จัดทำได้ทำการทดสอบอัลกอริทึมโดยแบ่งการทดสอบออกเป็น 2 รูปแบบ คือ การทดสอบอัลกอริทึมที่ไม่มีการแจ้งเตือน และการทดสอบอัลกอริทึมที่มีการสร้างระบบการแจ้งเตือน

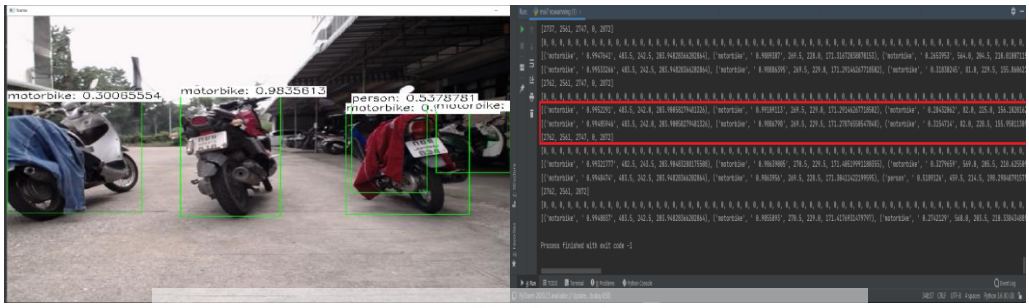
4.1.1 การทดสอบอัลกอริทึมที่ไม่มีการแจ้งเตือน

นำกล้อง และ RPLIDAR รุ่น A1M8 มาติดตั้งเข้ากับโมเดลที่ได้ทำการออกแบบไว้ จากนั้นเริ่มทำการทดสอบการทำงานของอัลกอริทึมในระยะห่างที่ 2000, 3000 และ 4000 มิลลิเมตร ตามลำดับ

ณ ระยะห่าง 2000 มิลลิเมตร

รูปภาพที่ 4.1 การทดสอบอัลกอริทึมที่ระยะห่าง 2000 มิลลิเมตร

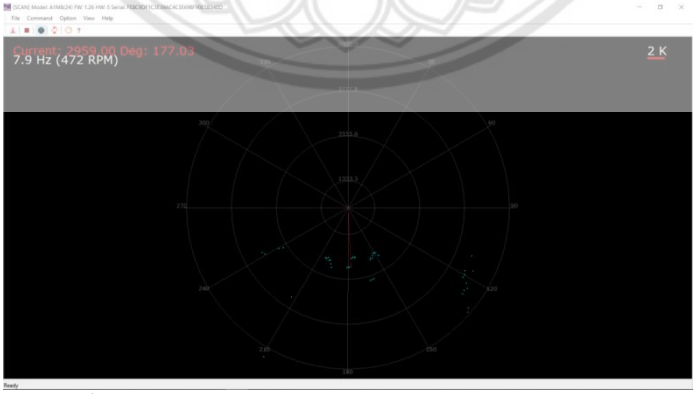
จากรูปภาพที่ 4.1 เริ่มต้นทำการทดสอบโดยจัดวางตำแหน่งให้อุปกรณ์อยู่ห่างจากรถจักรยานยนต์ทั้งสามคัน โดยให้อุปกรณ์อยู่ห่างจากรถจักรยานยนต์ที่ระยะห่าง 2000 มิลลิเมตร ทำให้ได้ผลการทำงานของอัลกอริทึมดังแสดงในรูปที่ 4.2 (ก) และ (ข)



(ก) (ข)

รูปภาพที่ 4.2 ภาพจากกล้อง (ก) และผลการทำงานของอัลกอริทึม (ข)

จากรูปที่ 4.2 ที่ระยะ 2000 มิลลิเมตร พบว่าอัลกอริทึมที่ทางคณะผู้จัดทำได้สร้างขึ้นสามารถตรวจจับเจอวัตถุด้านหน้าของกล้องนั่นคือรถจักรยานยนต์ และอัลกอริทึมสามารถจำแนกได้ว่าเป็นวัตถุชนิดใด โดยค่าที่ได้จากกล้องจะแสดงชนิดวัตถุและค่าความแม่นยำ จากรูปที่ 4.2 (ก) รถคันซ้ายมีค่าความแม่นยำเท่ากับ 0.3006554 รถคันกลางมีค่าความแม่นยำเท่ากับ 0.9835613 และจากรูปที่ 4.2 (ข) ผลการทำงานของอัลกอริทึมที่รับข้อมูลมาจากกล้องที่ทำการจำแนกชนิดวัตถุแล้วทำให้อัลกอริทึมสามารถแสดงค่าระยะห่างที่วัดได้ ซึ่งรถคันซ้ายมีระยะห่างเท่ากับ 2747 มิลลิเมตร รถคันกลางมีระยะห่างเท่ากับ 2561 มิลลิเมตร



รูปภาพที่ 4.3 ผลการทำงานของโปรแกรม frame_grabber.exe

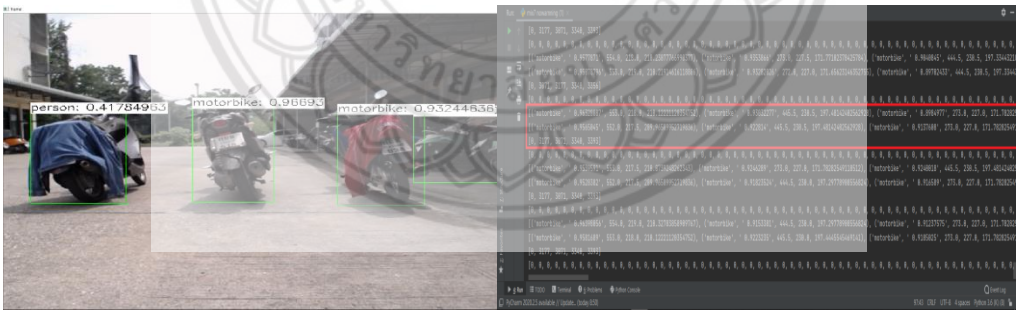
จากรูปที่ 4.3 ที่ระยะ 2000 มิลลิเมตร พบว่าเมื่อนำผลการทำงานจากอัลกอริทึมที่ทำการสร้างขึ้นมาทำการเทียบกับผลการทำงานแบบจำลองในโปรแกรม frame_grabber.exe ได้ค่า

ระยะทางที่ตรวจจับได้มีค่าเท่ากับ 2959 มิลลิเมตร สำหรับวัตถุที่อยู่บริเวณตรงกลาง ซึ่งมีความใกล้เคียงกับระยะทางที่อัลกอริทึมที่ทางคณะผู้จัดทำสร้างขึ้น

ณ ระยะห่าง 3000 มิลลิเมตร

รูปภาพที่ 4.4 การทดสอบอัลกอริทึมที่ระยะห่าง 3000 มิลลิเมตร

จากรูปภาพที่ 4.4 เริ่มต้นทำการทดสอบโดยจัดวางตำแหน่งให้อุปกรณ์อยู่ห่างจากรถจักรยานยนต์ทั้งสามคัน โดยให้อุปกรณ์อยู่ห่างจากรถจักรยานยนต์ที่ระยะห่าง 3000 มิลลิเมตร ทำให้ได้ผลการทำงานของอัลกอริทึมดังแสดงในรูปที่ 4.5 (ก) และ (ข)

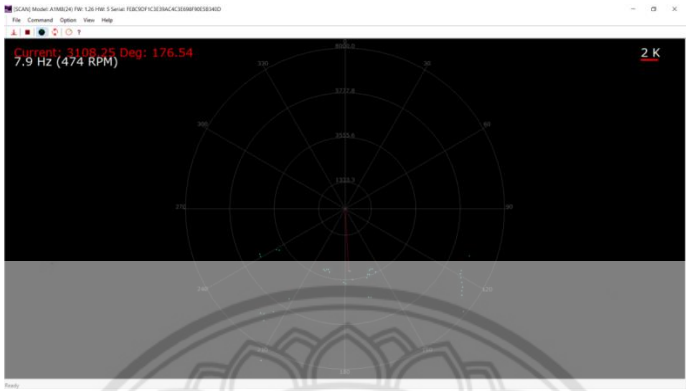


(ก) (ข)

รูปภาพที่ 4.5 ภาพจากกล้อง (ก) และผลการทำงานของอัลกอริทึม (ข)

จากรูปที่ 4.5 ที่ระยะ 3000 มิลลิเมตร พบว่าอัลกอริทึมที่ทางคณะผู้จัดทำได้สร้างขึ้นสามารถตรวจจับเจอวัตถุด้านหน้าของกล้องนั่นคือรถจักรยานยนต์ และอัลกอริทึมสามารถจำแนกได้ว่าเป็นวัตถุชนิดใด โดยค่าที่ได้จากกล้องจะแสดงชนิดวัตถุและค่าความแม่นยำ จากรูปที่ 4.5 (ก) รถคันซ้ายมีค่าความแม่นยำเท่ากับ 0.41784963 รถคันกลางมีค่าความแม่นยำเท่ากับ 0.96693 รถคันขวามีค่าความแม่นยำเท่ากับ 0.93244636 และจากรูปที่ 4.5 (ข) ผลการทำงานของอัลกอริทึมที่รับ

ข้อมูลมาจากกล้องที่ทำการจำแนกชนิดวัตถุแล้ว ทำให้อัลกอริทึมสามารถแสดงค่าระยะห่างที่วัดได้ ซึ่งรถคันขวามีระยะห่างเท่ากับ 3177 มิลลิเมตร รถคันกลางมีระยะห่างเท่ากับ 3071 มิลลิเมตร



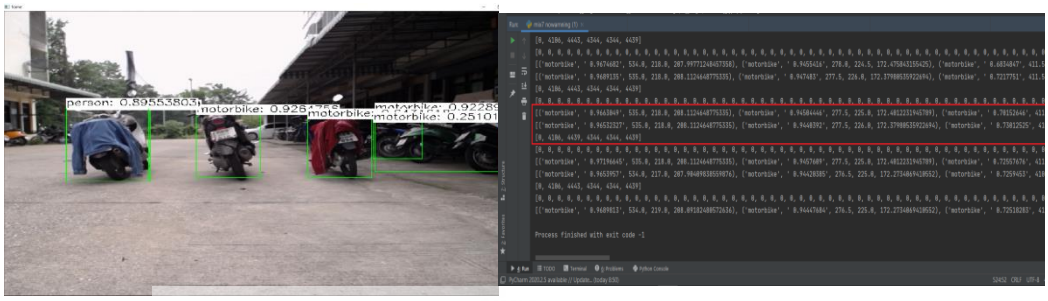
รูปภาพที่ 4.6 ผลการทำงานของโปรแกรม frame_grabber.exe

จากรูปที่ 4.6 ที่ระยะ 3000 มิลลิเมตร พบว่าเมื่อนำผลการทำงานจากอัลกอริทึมที่ทำการสร้างขึ้นมาทำการเทียบกับผลการทำงานแบบจำลองในโปรแกรม frame_grabber.exe ได้ค่าระยะทางที่ตรวจจับได้มีค่าเท่ากับ 3108.25 มิลลิเมตร สำหรับวัตถุที่อยู่บริเวณตรงกลาง ซึ่งมีความใกล้เคียงกับระยะทางที่อัลกอริทึมที่ทางคณะผู้จัดทำสร้างขึ้น

ณ ระยะห่าง 4000 มิลลิเมตร

รูปภาพที่ 4.7 การทดสอบอัลกอริทึมที่ระยะห่าง 4000 มิลลิเมตร

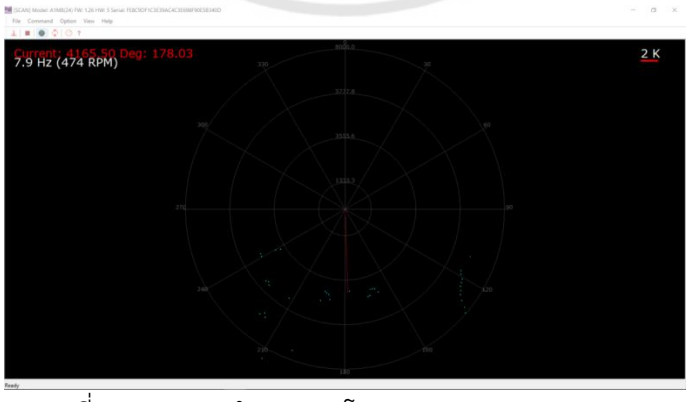
จากรูปภาพที่ 4.7 เริ่มต้นทำการทดสอบโดยจัดวางตำแหน่งให้อุปกรณ์อยู่ห่างจากรถจักรยานยนต์ทั้งสามคัน โดยให้อุปกรณ์อยู่ห่างจากรถจักรยานยนต์ที่ระยะห่าง 4000 มิลลิเมตร ทำให้ได้ผลการทำงานของอัลกอริทึมดังแสดงในรูปที่ 4.8 (ก) และ (ข)



(ก) (ข)

รูปภาพที่ 4.8 ภาพจากกล้อง (ก) และผลการทำงานของอัลกอริทึม (ข)

จากรูปที่ 4.8 ที่ระยะ 4000 มิลลิเมตร พบว่าอัลกอริทึมที่ทางคณะผู้จัดทำได้สร้างขึ้นสามารถตรวจจับเจอวัตถุด้านหน้าของกล้องนั่นคือรถจักรยานยนต์ และอัลกอริทึมสามารถจำแนกได้ว่าเป็นวัตถุชนิดใด โดยค่าที่ได้จากกล้องจะแสดงชนิดวัตถุและค่าความแม่นยำ จากรูปที่ 4.8 (ก) รถคันกลางมีค่าความแม่นยำเท่ากับ 0.926 รถคันขวามีค่าความแม่นยำเท่ากับ 0.647 และจากรูปที่ 4.8 (ข) ผลการทำงานของอัลกอริทึมที่รับข้อมูลมาจากกล้องที่ทำการจำแนกชนิดวัตถุแล้ว ทำให้อัลกอริทึมสามารถแสดงค่าระยะห่างที่วัดได้ ซึ่งรถคันขวามีระยะห่างเท่ากับ 4439 มิลลิเมตร รถคันกลางมีระยะห่างเท่ากับ 4106 มิลลิเมตร



รูปภาพที่ 4.9 ผลการทำงานของโปรแกรม frame_grabber.exe

จากรูปที่ 4.9 ที่ระยะ 4000 มิลลิเมตร พบว่าเมื่อนำผลการทำงานจากอัลกอริทึมที่ทำการสร้างขึ้นมาทำการเทียบกับผลการทำงานแบบจำลองในโปรแกรม frame_grabber.exe ได้ค่าระยะทางที่ตรวจจับได้มีค่าเท่ากับ 4165.50 มิลลิเมตร สำหรับวัตถุที่อยู่บริเวณตรงกลาง ซึ่งมีความใกล้เคียงกับระยะทางที่อัลกอริทึมที่ทางคณะผู้จัดทำสร้างขึ้น

ตารางที่ 4.1 รายละเอียดการเปรียบเทียบระยะห่างของยานพาหนะจากการทดสอบผลจากอัลกอริทึมและผลจากโปรแกรม frame_grabber.exe

ระยะห่างของยานพาหนะ (มิลลิเมตร)	ระยะห่างของยานพาหนะจากอัลกอริทึม (มิลลิเมตร)	ระยะห่างของยานพาหนะจากโปรแกรม frame_grabber.exe (มิลลิเมตร)
2000	2561	2959
3000	3071	3108.25
4000	4106	4165.50

จากตารางที่ 4.1 ในการทดสอบอัลกอริทึมก่อนมีการแจ้งเตือนเพื่อตรวจสอบความสามารถในการตรวจจับวัตถุ พบว่าสามารถตรวจจับที่ระยะ 2000, 3000 และ 4000 มิลลิเมตร อัลกอริทึมสามารถตรวจจับวัตถุและให้ค่าระยะห่างที่มีความใกล้เคียงกับความเป็นจริง โดยที่ระยะ 2000 มิลลิเมตร พบวัตถุมีระยะห่างเท่ากับ 2561 มิลลิเมตร มีค่าความคลาดเคลื่อนอยู่ที่ร้อยละ 28.05 ซึ่งเกิดความคลาดเคลื่อนเนื่องจากรถจักรยานยนต์ที่ใช้นั้นการทดสอบนั้นวางเอียงทำมุมกับอุปกรณ์ทำให้ระยะที่ตรวจพบคลาดเคลื่อน ที่ระยะ 3000 มิลลิเมตร พบวัตถุมีระยะห่างเท่ากับ 3071 มิลลิเมตร มีค่าความคลาดเคลื่อนอยู่ที่ร้อยละ 2.37 ที่ระยะ 4000 มิลลิเมตร พบวัตถุมีระยะห่างเท่ากับ 4106 มิลลิเมตร มีค่าความคลาดเคลื่อนอยู่ที่ร้อยละ 2.85

4.1.2 การทดสอบอัลกอริทึมที่มีการแจ้งเตือน

จากการสร้างระบบการแจ้งเตือนการชนในรถจักรยานยนต์ทางคณะผู้จัดทำได้เพิ่มให้อัลกอริทึมสามารถแจ้งเตือนได้ตามวัตถุประสงค์ที่ได้ตั้งไว้ โดยที่ระบบการแจ้งเตือนนี้จะสอดคล้องกับระยะเบรกปลอดภัยที่ทางคณะผู้จัดทำได้ทำการคำนวณไว้ให้มีระยะห่างน้อยกว่า 3 เมตร เริ่มทำการทดสอบอัลกอริทึมโดยนำกล้อง และ RPLIDAR รุ่น A1M8 มาติดตั้งเข้ากับโมเดลที่ได้ทำการออกแบบไว้ จากนั้นทดสอบการทำงานของอัลกอริทึมที่มีการแจ้งเตือนในระยะห่างที่ 1000, 2000 และ 3000 มิลลิเมตรตามลำดับ

(ก) (ข) (ค)

รูปภาพที่ 4.10 การการทดสอบอัลกอริทึมที่มีการแจ้งเตือน ระยะห่าง 1000 mm (ก) ระยะห่าง 2000 mm (ข) ระยะห่าง 3000 mm (ค)

จากรูปที่ 4.10 ที่ระยะ 1000 มิลลิเมตร, 2000 มิลลิเมตร และ 3000 มิลลิเมตร พบว่าอัลกอริทึมที่ทำการสร้างให้มีการแจ้งเตือนที่ระยะน้อยกว่า 3000 มิลลิเมตร นั้นสามารถส่งเสียงแจ้งเตือนเมื่อตรวจพบเจอวัตถุที่อยู่ด้านหน้าของอุปกรณ์

4.2 วิเคราะห์ผลการทดลอง

จากผลการทดลองที่ 4.1.1 พบว่าเมื่อนำข้อมูลที่ได้จากกล้องไปแทนค่าในสมการที่ (3.2) จะได้มุมจากข้อมูลของ lidar ที่เป็นมุมเดียวกันกับมุมที่คำนวณได้จากภาพ ส่งผลให้สามารถหาระยะห่างของวัตถุภายในภาพได้ ซึ่งจากตารางที่ 4.1 จะพบว่าค่าระยะห่างที่ได้จากอัลกอริทึมมีค่าใกล้เคียงกับระยะห่างจริงของวัตถุและใกล้เคียงกับระยะห่างที่ได้จากโปแกรม frame_grabber.exe ด้วยเช่นกัน

จากผลการทดลองที่ 4.1.2 พบว่าอัลกอริทึมที่สร้างขึ้นสามารถที่จะแจ้งเตือนผู้ขับขี่ได้เมื่อมีวัตถุอยู่ภายในระยะไม่เกิน 3 เมตร แต่มีข้อจำกัดต่างๆ ประการแรกคือ เซนเซอร์ lidar ที่ใช้การตรวจจับวัตถุด้วยแสงนั้นจะมีข้อจำกัดดังนี้ คือ

1.หากพบกับพื้นที่ที่มีการหักเหของแสงมาก จะส่งผลให้เกิดความผิดพลาดเนื่องจากเซนเซอร์ lidar ไม่สามารถตรวจจับแสงที่เกิดการหักเหออกไปได้

2.ในสภาพแวดล้อมที่มีแสงมากจะทำให้ประสิทธิภาพในการตรวจจับวัตถุได้น้อยกว่าในสภาพแวดล้อมที่มีแสงน้อย

ประการที่สองคือ ตัวโมเดล MobileNet สามารถทำงานได้รวดเร็วแต่ในการแบ่งแยกชนิดของวัตถุอาจมีการตรวจจับที่ผิดพลาดและยังไม่เสถียร เช่น ตรวจจับพบวัตถุสองชนิดในวัตถุเดียวกัน เป็นต้น และในการตรวจจับรถจักรยานยนต์ที่ใช้ในการทดสอบนั้นมีช่วงความกว้างที่น้อยทำให้ความสามารถในการตรวจจับลดลงไปด้วย

ดังนั้น การทดสอบอัลกอริทึมครั้งนี้ ยังคงเป็นเพียงการสร้างและพัฒนาระบบการแจ้งเตือนการชนในรถจักรยานยนต์ ซึ่งพบว่าผลการดำเนินงานที่ได้ คือ สามารถสร้างอัลกอริทึมสำหรับระบบการแจ้งเตือนการชนในรถจักรยานยนต์เพื่อวิเคราะห์ระยะห่างวัตถุด้วยเซนเซอร์ lidar และทำการแจ้งเตือนออกมาในระยะห่างไม่เกิน 3000 มิลลิเมตร แต่อัลกอริทึมที่สร้างขึ้นมายังคงมีข้อจำกัดในด้านของอุปกรณ์ RPLIDAR รุ่น A1M8 และการทำงานที่ยังไม่เสถียรของโมเดล MobileNet จึงทำให้ต้องมีการพัฒนาระบบการแจ้งเตือนการชนในรถจักรยานยนต์ต่อไปเพื่อลดข้อจำกัดที่ได้กล่าวมาข้างต้น

บทที่ 5

สรุปผลโครงงานและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการได้ศึกษาข้อมูลที่มีความเกี่ยวข้องกับระบบควบคุมความเร็วอัตโนมัติ (Adaptive Cruise Control) และการทำงานของหุ่นยนต์อัตโนมัติที่มีการใช้ระบบตรวจจับสิ่งกีดขวางที่มีการทำงานคล้ายคลึงกับระบบควบคุมความเร็วอัตโนมัติ การศึกษาข้อมูลที่ได้กล่าวมานี้สามารถนำไปพัฒนาและประยุกต์ให้เข้ากับการสร้างระบบการแจ้งเตือนการชนในรถจักรยานยนต์ โดยใช้อุปกรณ์ RPLIDAR รุ่น A1M8 ในการตรวจจับและหาระยะห่างของยานพาหนะที่อยู่ด้านหน้าของอุปกรณ์ และใช้กล้อง Logitech c922 pro สำหรับการตรวจจับและจำแนกชนิดของยานพาหนะหรือวัตถุบนท้องถนน ในส่วนของการสร้างนั้นทางคณะผู้จัดทำได้ทำการใช้ภาษา python เพื่อสร้างและพัฒนาอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ โดยการนำโมเดล MobileNet เข้ามาช่วยในการจำแนกประเภทของวัตถุ และมีการปรับปรุง MobileNet ให้สามารถได้ผลลัพธ์การทำงานออกมาเป็นมุมของภาพ และมีการนำโมเดลสำเร็จรูปของ RPLIDAR มาปรับปรุงเพื่อให้สามารถนำข้อมูลระยะห่างที่ต้องการมาใช้ให้สามารถทำงานร่วมกับ MobileNet ที่ถูกปรับปรุงไว้ เมื่อรวมการทำงานเข้าไว้ด้วยกันจะทำให้ได้ระยะห่างของวัตถุที่ตรวจจับได้และสามารถจำแนกชนิดของวัตถุนั้นได้ เพื่อให้การทำงานเป็นไปตามวัตถุประสงค์ทางคณะผู้จัดทำจึงพัฒนาให้ระบบอัลกอริทึมนี้สามารถแจ้งเตือนออกมาในรูปแบบของเสียงเมื่อตรวจจับเจอวัตถุในระยะห่างที่ได้ทำการกำหนดไว้ ระบบการแจ้งเตือนการชนในรถจักรยานยนต์ที่ถูกพัฒนาขึ้นมานี้ยังไม่มีความเสถียรมากนักสามารถนำไปพัฒนาต่อเพื่อทำให้ระบบมีประสิทธิภาพมากยิ่งขึ้น

จากผลการทดลอง พบว่าสมการที่ (3.2) สามารถนำมุมที่ได้จากการคำนวณหามุมภายในภาพมาแทนค่าในสมการจะได้มุมมาเปรียบเทียบกับข้อมูลของ lidar ทำให้ได้ข้อมูลระยะห่างของวัตถุที่มีค่าใกล้เคียงกับระยะห่างจริงได้ ดังแสดงในตารางที่ 4.1 และสามารถที่จะสร้างอัลกอริทึมที่จะแจ้งเตือนด้วยเสียงเมื่อตรวจจับวัตถุที่มีระยะห่างไม่เกิน 3 เมตร แต่ระบบการแจ้งเตือนการชนในรถจักรยานยนต์ยังมีข้อบกพร่อง คือ ประสิทธิภาพในการจำแนกชนิดวัตถุของ MobileNet และเซนเซอร์ lidar ยังมีข้อจำกัดในการตรวจจับวัตถุที่มีการหักเหของแสง และสภาพแวดล้อมต่าง ๆ

ที่ส่งผลให้ประสิทธิภาพในการตรวจจับระยะห่างของวัตถุน้อยลง และอัลกอริทึมของระบบการแจ้งเตือนการชนในรถจักรยานยนต์ยังไม่สมบูรณ์และการทำงานที่ไม่เสถียรเท่าที่ควร

5.2 ข้อเสนอแนะ

5.2.1 ปัญหาที่พบ

1) ประสิทธิภาพการทำงานของคอมพิวเตอร์ที่ใช้ในการประมวลผลนั้นยังมีประสิทธิภาพไม่มากพอ จึงทำให้ส่งผลต่ออุปกรณ์ที่นำมาติดตั้ง
2) โมเดล MobileNet สามารถทำงานได้รวดเร็ว แต่ยังมีความผิดพลาดในการจำแนกชนิดของวัตถุได้ไม่ถูกต้องในบางครั้ง
3) ข้อจำกัดของเซนเซอร์ lidar ที่เมื่อพบวัตถุบางชนิดที่มีความสามารถในการหักเหแสงและในสภาพแวดล้อมที่มีแสงมากทำให้ได้ข้อมูลที่ประสิทธิภาพลดลง
4) การตรวจจับระยะวัตถุจากเซนเซอร์ RPLIDAR รุ่น A1M8 เมื่อทดสอบแล้วมีประสิทธิภาพในการตรวจจับวัตถุที่ระยะสูงสุดไม่เท่ากับประสิทธิภาพที่ทางผู้ผลิตได้ระบุไว้
5) บอร์ด Raspberry Pi 3 Model B ยังไม่มีประสิทธิภาพมากพอในการใช้งานอัลกอริทึมของระบบการแจ้งเตือนการชนในรถจักรยานยนต์ ทำให้มีการทำงานที่ล่าช้า จึงจำเป็นต้องใช้คอมพิวเตอร์ในการประมวลผลของอัลกอริทึมและทำการทดสอบ
6) ความคลาดเคลื่อนจากการ calibrate ส่งผลเล็กน้อยต่อการคำนวณหาสมการที่ (3.2) และยังส่งผลต่อระยะห่างที่ออกมาจากอัลกอริทึม
7) ความเสถียรในการทำงานของระบบการแจ้งเตือนการชนในรถจักรยานยนต์นั้นยังไม่มากพอหรือไม่แน่นอน เนื่องจากในบางครั้งของการทดสอบมักพบปัญหา error ในอัลกอริทึมของระบบการแจ้งเตือนการชนในรถจักรยานยนต์

5.2.2 ข้อเสนอแนะ

1) ควรมีเครื่องมือ และอุปกรณ์ที่สามารถใช้สำหรับการทดสอบที่มีประสิทธิภาพมากกว่านี้ โดยการเปลี่ยนเซนเซอร์ลิดาร์เป็นรุ่นที่สามารถตรวจจับระยะห่างได้มากขึ้น และปรับเปลี่ยนรุ่นของบอร์ดเพื่อทำให้สามารถรองรับอัลกอริทึม
2) ควรมีสถานที่ และอุปกรณ์ที่พร้อมใช้ในการ calibrate เพื่อที่จะไม่ทำให้ส่งผลต่อความคลาดเคลื่อนในการคำนวณ
3) คอมพิวเตอร์ที่ใช้ควรมีความสามารถในการประมวลผลที่ดี เพื่อลดปัญหาในการทำงานของอัลกอริทึม
4) ควรใช้บอร์ดในการประมวลผลที่มี CPU รองรับอัลกอริทึม ที่มีประสิทธิภาพมากกว่าบอร์ด Raspberry Pi 3 Model B

บรรณานุกรม

1. สำนักงานนโยบายและแผนการขนส่งและจราจร. (สิงหาคม 2562). รายงานการวิเคราะห์สถานการณ์อุบัติเหตุทางถนนของกระทรวงคมนาคม พ.ศ. 2561. **สำนักงานนโยบายและแผนการขนส่งและจราจร.** สืบค้นเมื่อ 28 กุมภาพันธ์ 2563, จาก http://www.otp.go.th/index.php/post/view?id=3548
2. สำนักงานนโยบายและแผนการขนส่งและจราจร. (มิถุนายน 2561). รายงานการวิเคราะห์สถานการณ์อุบัติเหตุทางถนนของกระทรวงคมนาคม พ.ศ. 2560. **สำนักงานนโยบายและแผนการขนส่งและจราจร.** สืบค้นเมื่อ 28 กุมภาพันธ์ 2563, จาก http://www.otp.go.th/index.php/post/view?id=2597
3. Paul Simpson. (ไม่ทราบข้อมูลแน่ชัด). Braking – Stopping distances. **Begin Motorcycling.** สืบค้นเมื่อ 28 กุมภาพันธ์ 2563, จาก https://begin-motorcycling.co.uk/the-5-elements-of-cbt/element-c/braking/
4. Joh Burut. (ไม่ทราบข้อมูลแน่ชัด). ADAPTIVE CRUISE CONTROL คืออะไร ต่างจาก CRUISE CONTROL อย่างไร. **Joh's Autolife.** สืบค้นเมื่อ 28 กุมภาพันธ์ 2563, จาก http://www.johsautolife.com/index.php/2015-12-30-03-42-00/2015-12-30-03-43-37/146-adaptive-cruise-control
5. ภูมิพลอดุลยเดช ป.ร. (18 มกราคม 2522). **พระราชบัญญัติจราจรทางบก พ.ศ. 2522.** สืบค้นเมื่อ 1 มีนาคม 2563, จาก http://web.krisdika.go.th/data/law/law2/%A803/%A803-20-9999-update.htm
6. Egeomates. (กรกฎาคม 2560). LiDAR information - what for now. **Geo Recursos para Geo-ingeniería.** สืบค้นเมื่อ 5 มีนาคม 2563 จาก https://th.geofumadas.com/datos-lidar-ahora
7. Mk. (13 สิงหาคม 2559). MIT โชว์เทคโนโลยี LIDAR แบบใหม่ เป็นชิปขนาดเล็กกว่าเหรียญ ต้นทุนต่ำกว่า 10 ดอลลาร์. **Blognone.** สืบค้นเมื่อ 5 มีนาคม 2563 จาก https://www.blognone.com/node/84463
8. SLAMTEC. (ไม่ทราบข้อมูลแน่ชัด). **RPLIDAR A1.** สืบค้นเมื่อ 5 มีนาคม 2563 จาก https://www.slamtec.com/en/Lidar/A1
9. YDLIDAR. (ไม่ทราบข้อมูลแน่ชัด). **YDLIDAR TX20.** สืบค้นเมื่อ 5 มีนาคม 2563 จาก http://www.ydlidar.com/products/view/9.html
10. Raspberry Pi. (ไม่ทราบข้อมูลแน่ชัด). **ทำความรู้จัก Raspberry Pi.** สืบค้นเมื่อ 5 มีนาคม 2563 จาก http://www2.crma.ac.th/itd/Know/RBPI/index.asp

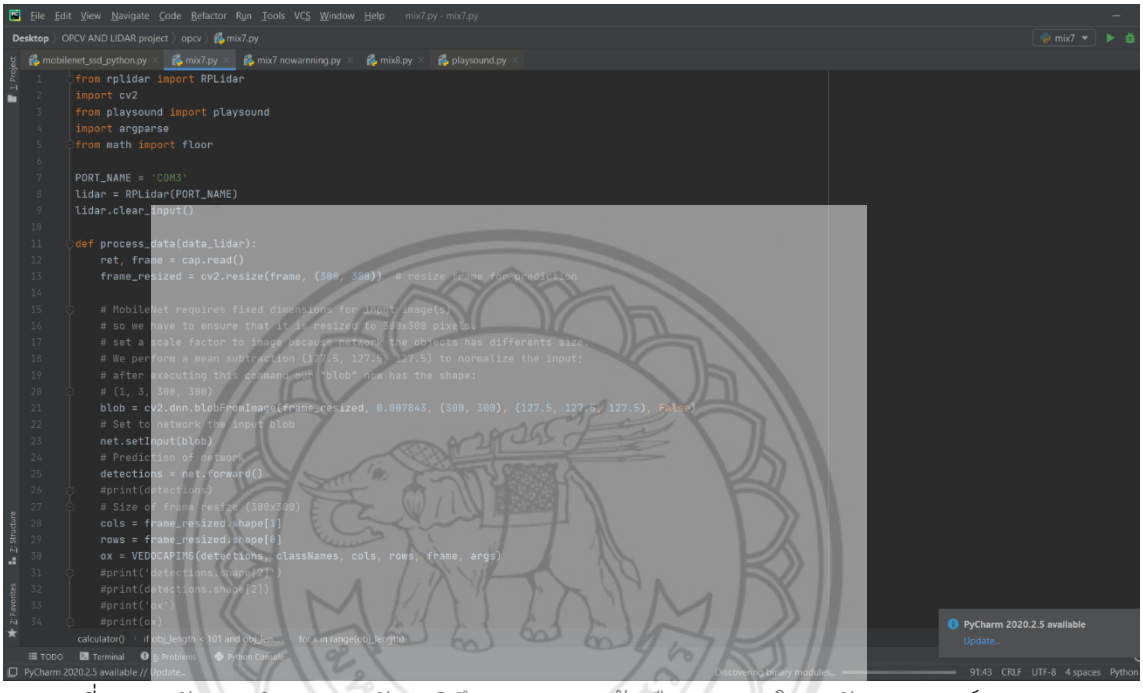
11. Kenneth Kimari. (20 มิถุนายน 2562). Nvidia Jetson Nano vs. Raspberry Pi. **Maketecheasier.** สืบค้นเมื่อ 5 มีนาคม 2563 จาก https://www.maketecheasier.com/nvidia-jetson-nano-vs-raspberry-pi/
12. Logitech. (ไม่ทราบข้อมูลแน่ชัด). **Logitech C922 pro.** สืบค้นเมื่อ 5 มีนาคม 2563 จาก https://www.logitech.com/th-th/product/c922-pro-stream-webcam
13. ธีระยุทธ สุวรรณประธีป. (2559). **หนังสือวิศวกรรมยานยนต์.** (พิมพ์ครั้งที่ 15). กรุงเทพฯ: วิทยพัฒน์.
14. Liew Wuttipat. (22 พฤศจิกายน 2562). เริ่มใช้งาน Jetson Nano DevKit. **บริษัท อันแมน เทคโนโลยี จำกัด.** สืบค้นเมื่อ 5 มีนาคม 2563 จาก https://www.anman.co.th/blog/sbc-12/post/jetson-nano-devkit-40
15. จะเด็ด ทองสองขวัญ. (2559). **ระบบตรวจจับและพิสูจนอัตลักษณใบหนาเพื่อคัดกรองบุคคลบริเวณพื้นที่สวนตัว ดวยบอรดราสเบอรรี่ พาย.** วิทยานิพนธ์ วศ.ม, มหาวิทยาลัยเทคโนโลยีมหานคร, กรุงเทพมหานคร

ภาคผนวก ก

หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

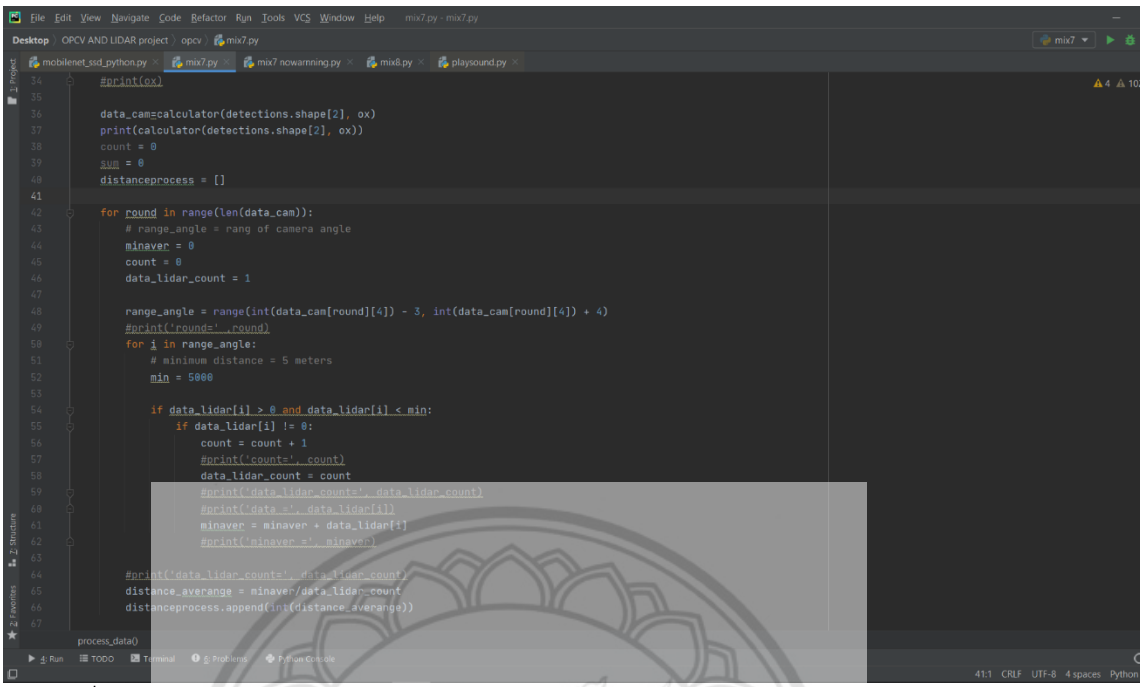
หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

การสร้างระบบการแจ้งเตือนการชนในรถจักรยานยนต์ในด้านของอุปกรณ์นั้นต้องประกอบไปด้วย RPLIDAR และกล้อง แต่สำหรับอัลกอริทึมนั้นสามารถสร้างและอธิบายการทำงานได้ดังแสดงในรูปที่ ก.1, ก.2, ก.3, ก.4, ก.5, ก.6 และ ก.7



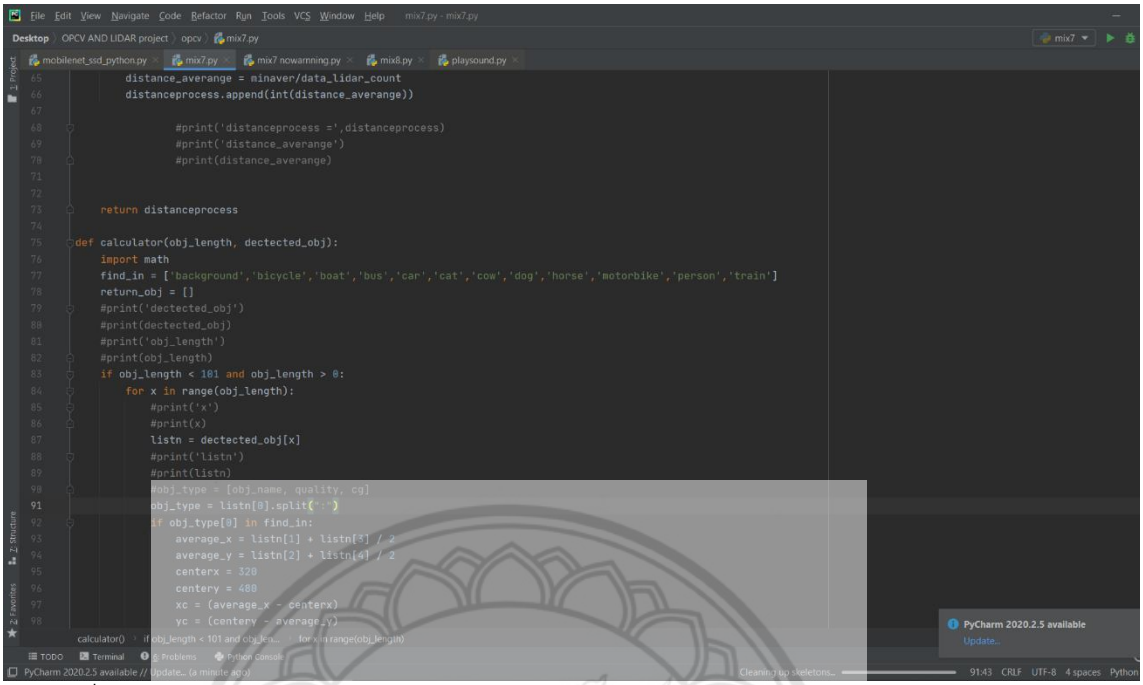
รูปที่ ก.1 หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

จากรูปที่ ก.1 เริ่มต้นจากการ import ไลบรารี่ที่จำเป็น คือ RPLidar, cv2, playsound, argparse และ floor จากนั้นกำหนดให้ PORT_NAME = port COM3 และกำหนดให้ตัวแปร lidar คือ RPLidar จาก port COM3 จากนั้นทำการสร้างฟังก์ชั่นที่มีชื่อว่า process_data โดยให้นำข้อมูลที่มีชื่อว่า data_lidar มาใช้ ทำการจับภาพด้วยกล้องแล้วเก็บข้อมูลไว้ใน ret และ frame แล้วทำการปรับขนาดของ frame สำหรับใช้กับโมเดล MobileNet จากนั้นให้เก็บข้อมูลจากภาพไว้ในตัวแปรชื่อ cols และ rows สร้างตัวแปรเก็บค่าชื่อ ox โดยเก็บข้อมูลจากฟังก์ชั่น VEDOCAPIMG (detections, classNames, cols, rows, frame, args)



รูปที่ ก.2 หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

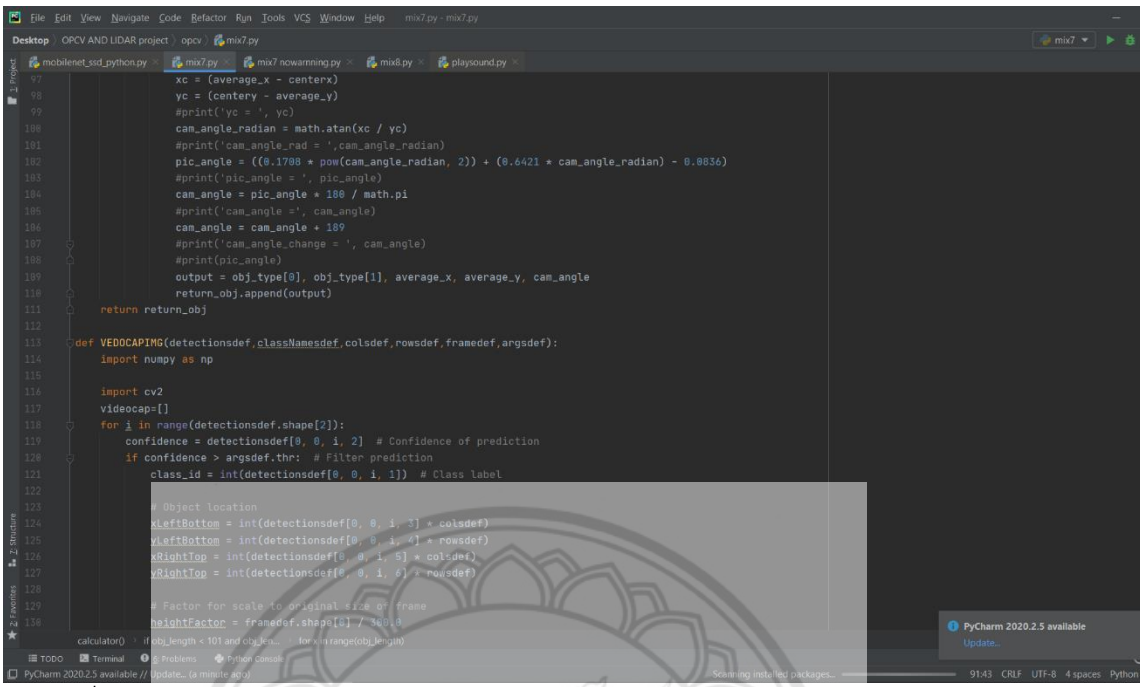
จากรูปที่ ก.2 ทำการสร้างตัวแปรชื่อ data_cam เพื่อมาเก็บข้อมูลที่ได้จากกล้อง และสั่งให้แสดงผลของค่านั้น จากนั้นสร้างตัวแปร count ให้มีค่าเท่ากับ 0 และตัวแปร sum ให้มีค่าเท่ากับ 0 และสร้างตัวแปร distanceprocess เพื่อมารับค่าจากระยะทางที่เป็นผลลัพธ์จากฟังก์ชั่น process_data ทำการสร้าง loop for โดยกำหนดให้ตัวแปร round มีค่าเท่ากับ ช่วงของขนาดของตัวแปร data_cam สร้างตัวแปร minaver ให้มีค่าเท่า 0 ทำการสร้างตัวแปร count ให้มีค่าเท่ากับ 0 และสร้างตัวแปร data_lidar_count ให้มีค่าเท่ากับ 1 จากนั้นสร้างตัวแปร range_angle ให้มีค่าเป็นช่วงของข้อมูลของมุมที่พบวัตถุ สร้าง loop for โดยกำหนดให้ตัวแปร i มีค่าเท่ากับ range_angle กำหนดให้ min มีค่าเท่ากับ 5000 มิลลิเมตร สร้างเงื่อนไขถ้าข้อมูลระยะทางที่ได้มีค่ามากกว่า 0 และน้อยกว่าค่า min หากข้อมูลที่ได้ตรงตามเงื่อนไขแสดงว่าผ่านเงื่อนไขที่ได้ตั้งไว้ และถ้าข้อมูลที่ได้ไม่เท่ากับ 0 จะให้ตัวแปร count มีค่าเพิ่มขึ้น 1 และให้ data_lidar_count มีค่าเท่ากับ count ให้ตัวแปร minaver มีค่าเท่ากับระยะทางผลรวมของระยะทางทั้งหมดที่สามารถตรวจจับได้ของวัตถุแต่ละชนิด สร้างตัวแปร distance_averange .ให้มีค่าเท่ากับค่าเฉลี่ยของระยะทางที่ตรวจพบ ทำการเก็บข้อมูลจากตัวแปร distance_averange ไว้ในตัวแปร distanceprocess ในลักษณะของ list



```
        distance_averange = minaver/data_lidar_count
        distanceprocess.append(int(distance_averange))

                #print('distanceprocess =',distanceprocess)
                #print('distance_averange')
                #print(distance_averange)


    return distanceprocess

def calculator(obj_length, dectected_obj):
    import math
    find_in = ['background','bicycle','boat','bus','car','cat','cow','dog','horse','motorbike','person','train']
    return_obj = []
    #print('dectected_obj')
    #print(dectected_obj)
    #print('obj_length')
    #print(obj_length)
    if obj_length < 101 and obj_length > 0:
        for x in range(obj_length):
            #print('x')
            #print(x)
            listn = dectected_obj[x]
            #print('listn')
            #print(listn)
            #obj_type = [obj_name, quality, cg]
            obj_type = listn[0].split(":")
            if obj_type[0] in find_in:
                average_x = listn[1] + listn[3] / 2
                average_y = listn[2] + listn[4] / 2
                centerx = 320
                centery = 480
                xc = (average_x - centerx)
                yc = (centery - average_y)
```

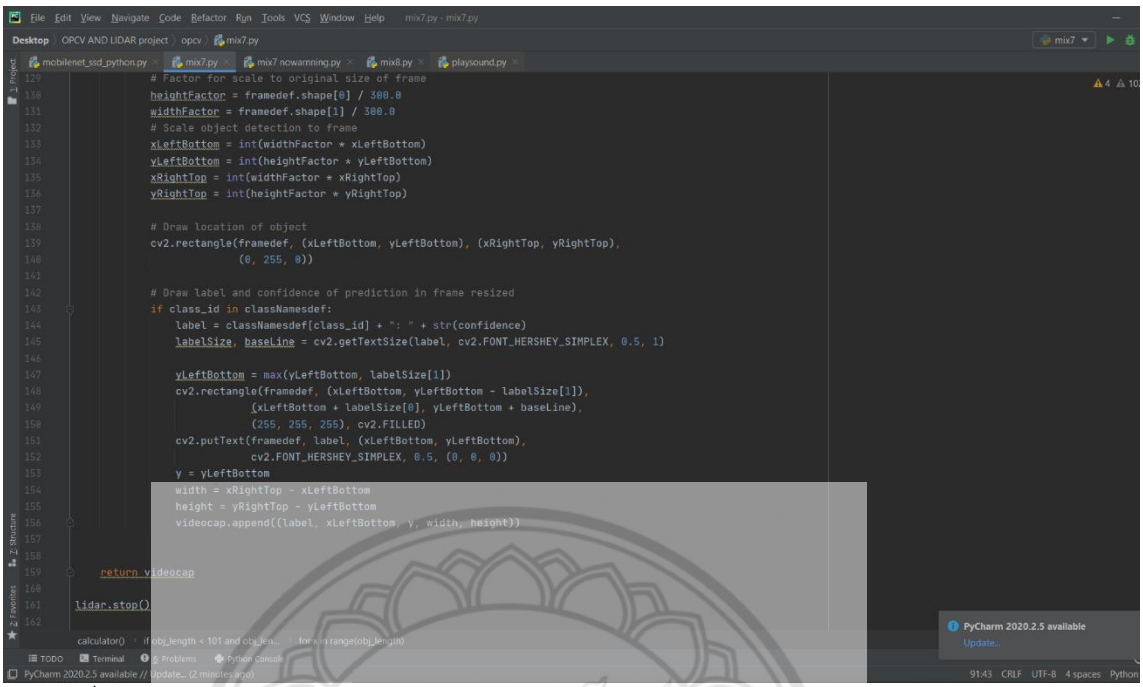
รูปที่ ก.3 หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

จากรูปที่ ก.3 ส่งผลลัพธ์ของฟังก์ชั่น process_data ออกมาเป็นข้อมูลในตัวแปร distanceprocess จากนั้นสร้างฟังก์ชันชื่อ calculator โดยรับข้อมูล obj_length และ dectected_obj มาใช้ ทำการ import ไลบรารี่ math กำหนดตัวแปรชื่อ find_in ให้มีข้อมูลคือ background, bicycle, boat, bus, car, cat, cow, dog, horse, motorbike, person และ train ลำดับต่อไปสร้างตัวแปร return_obj สำหรับเก็บผลลัพธ์ของฟังก์ชั่น calculator จากนั้นสร้างเงื่อนไขเมื่อความกว้างของช่วงข้อมูลน้อยกว่า 101 ข้อมูล และความกว้างของช่วงข้อมูลมีค่ามากกว่า 0 ให้ทำการสร้างตัวแปรชื่อว่า listn เพื่อให้เก็บค่าของข้อมูลที่ตรวจจับได้ และสร้างตัวแปร obj_type สำหรับเก็บข้อมูลที่ทำการแยกชุดข้อมูลแล้วจาก listn สร้างเงื่อนไขเมื่อ obj_type มีข้อมูลตรงกับข้อมูลใน find_in เมื่อเป็นไปตามเงื่อนไขให้ทำการคำนวณหาจุดกึ่งกลางของวัตถุเก็บไว้ในตัวแปรชื่อ average_x และ average_y กำหนดให้ตัวแปร centerx เท่ากับ 320 และ centery มีค่าเท่ากับ 480 เพื่อใช้ในการปรับแกนของข้อมูล ทำการหาจุดกึ่งกลางของวัตถุเมื่อปรับแกนแล้วจะได้ค่า xc และ yc ออกมา



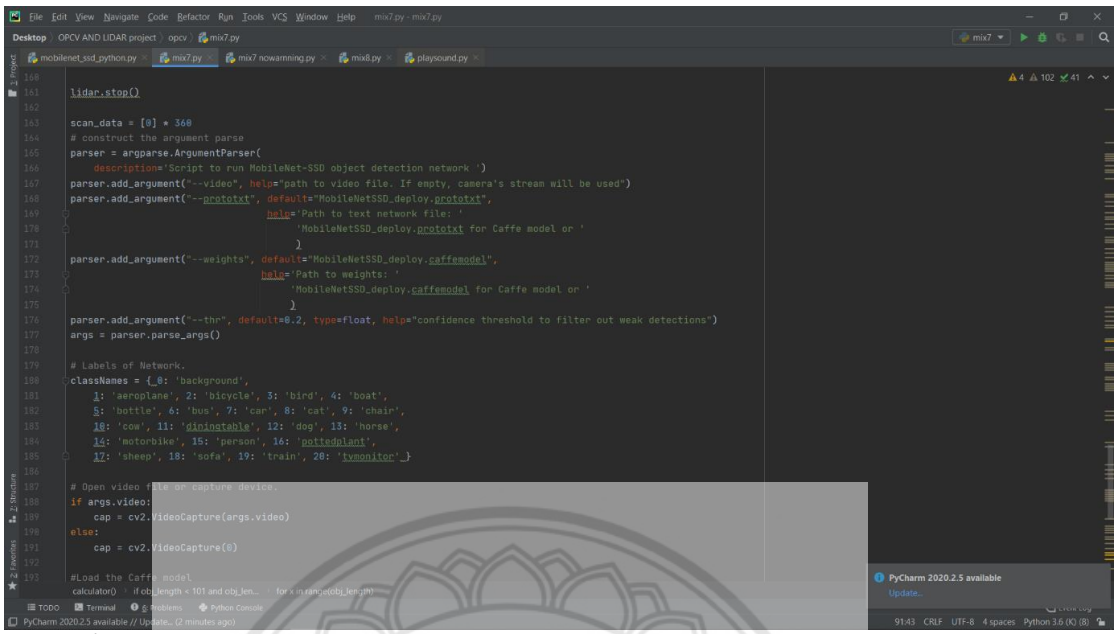
รูปที่ ก.4 หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

จากรูปที่ ก.4 สร้างตัวแปร cam_angle_radian เก็บค่าของมุมของวัตถุภายในภาพในหน่วยเรเดียน ทำการสร้างตัวแปร pic_angle จากสมการที่ (3.2) เพื่อหามุมที่นำไปใช้เปรียบเทียบกับมุมจากข้อมูลของ lidar เปลี่ยนหน่วยให้เป็นองศา จะได้ตัวแปรชื่อว่า cam_angle และทำการเปลี่ยนแกนเพื่อนำไปใช้ในการเปรียบเทียบกับมุมจากข้อมูลของ lidar ได้ จากนั้นสร้างตัวแปรชื่อ output ให้เก็บค่าของ obj_type[0], obj_type[1], average_x, average_y และ cam_angle ทำการเก็บของมูลของ output ไว้ในตัวแปร return_obj ในลักษณะของ list และส่งผลลัพธ์ของฟังก์ชั่น calculator ออกมาเป็นข้อมูลในตัวแปร return_obj ทำการสร้างฟังก์ชั่นชื่อ VEDOCAPIMG โดยรับข้อมูล detectionsdef, classNamesdef, colsdef, rowsdef, framedef และargsdef มาใช้ ทำการ import ไลบรารี่ numpy และ cv2 แล้วสร้างตัวแปรชื่อ videocap สำหรับเก็บผลลัพธ์ของฟังก์ชั่น VEDOCAPIMG ขั้นตอนต่อไปจะเป็นกระบวนการสร้างกรอบในการจำแนกวัตถุของโมเดล MobileNet



รูปที่ ก.5 หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

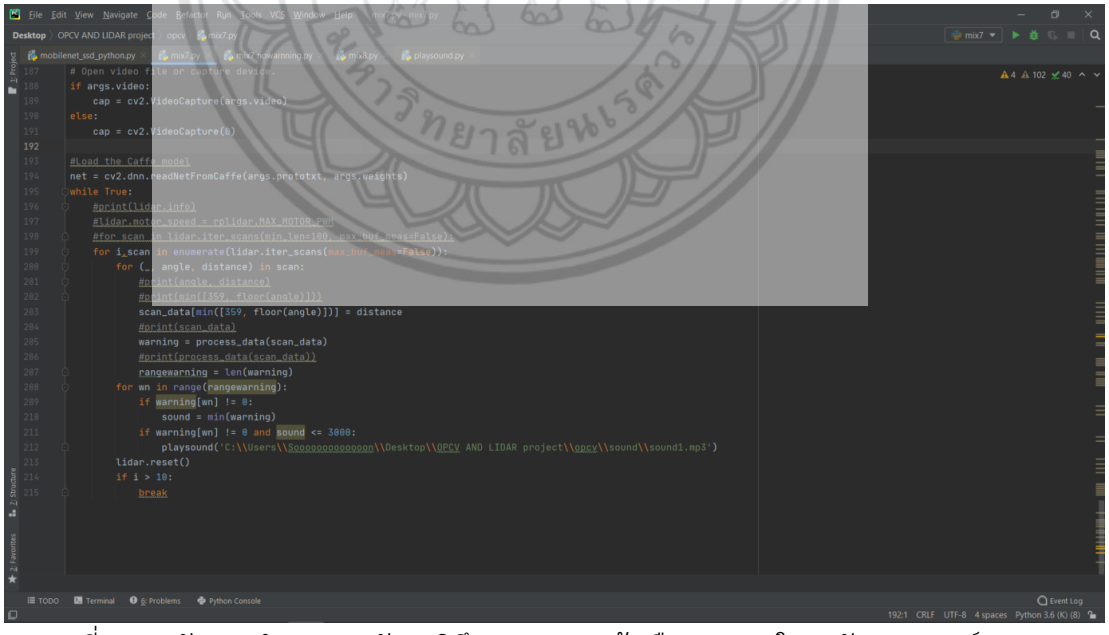
จากรูปที่ ก.5 เมื่อสร้างกรอบจำแนกชนิดเสร็จจึงทำการเก็บข้อมูล โดยสร้างตัวแปร y ให้เก็บข้อมูลของ yLeftBottom และคำนวณหาความกว้างและความสูงของกรอบภาพจำแนกวัตถุไว้ในตัวแปร width และ height และทำการเก็บของมูล label, xLeftBottom, y, width และ height ไว้ในตัวแปร videocap ในลักษณะของ list ส่งผลลัพธ์ของฟังก์ชั่น VEDOCAPIMG ออกมาเป็นข้อมูลในตัวแปร videocap



```
lidar.stop()

scan_data = [0] * 360
# construct the argument parse
parser = argparse.ArgumentParser(
    description='Script to run MobileNet-SSD object detection network ')
parser.add_argument("--video", help="path to video file. If empty, camera's stream will be used")
parser.add_argument("--prototxt", default="MobileNetSSD_deploy.prototxt",
                                  help='Path to text network file: '
                                       'MobileNetSSD_deploy.prototxt for Caffe model or '
                                       )
parser.add_argument("--weights", default="MobileNetSSD_deploy.caffemodel",
                                 help='Path to weights: '
                                      'MobileNetSSD_deploy.caffemodel for Caffe model or '
                                      )
parser.add_argument("--thr", default=0.2, type=float, help="confidence threshold to filter out weak detections")
args = parser.parse_args()

# Labels of Network.
classNames = { 0: 'background',
    1: 'aeroplane', 2: 'bicycle', 3: 'bird', 4: 'boat',
    5: 'bottle', 6: 'bus', 7: 'car', 8: 'cat', 9: 'chair',
    10: 'cow', 11: 'diningtable', 12: 'dog', 13: 'horse',
    14: 'motorbike', 15: 'person', 16: 'pottedplant',
    17: 'sheep', 18: 'sofa', 19: 'train', 20: 'tvmonitor' }

# Open video file or capture device.
if args.video:
    cap = cv2.VideoCapture(args.video)
else:
    cap = cv2.VideoCapture(0)

#Load the Caffe model
```

รูปที่ ก.6 หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

จากรูปที่ ก.6 กำหนดตัวแปร scan_data ให้มีข้อมูล 0 จำนวน 360 ตัว จากนั้นทำการสร้างโมเดล MobileNet เมื่อสร้างเสร็จจึงทำการเปิดการทำงานของกล้อง



```
# Open video file or capture device.
if args.video:
    cap = cv2.VideoCapture(args.video)
else:
    cap = cv2.VideoCapture(0)

#Load the Caffe model
net = cv2.dnn.readNetFromCaffe(args.prototxt, args.weights)
while True:
    #print(lidar.info)
    #lidar.motor_speed = rplidar.MAX_MOTOR_PWM
    #for scan in lidar.iter_scans(min_len=180, max_buf_meas=False):
    for i,scan in enumerate(lidar.iter_scans(max_buf_meas=False)):
        for (_, angle, distance) in scan:
            #print(angle, distance)
            #print(min([359, floor(angle)]))
            scan_data[min([359, floor(angle)])] = distance
            #print(scan_data)
            warning = process_data(scan_data)
            #print(process_data(scan_data))
            rangewarning = len(warning)
        for wn in range(rangewarning):
            if warning[wn] != 0:
                sound = min(warning)
            if warning[wn] != 0 and sound <= 3000:
                playsound('C:\\Users\\Soooooooooooon\\Desktop\\OPCV AND LIDAR project\\opcv\\sound\\sound1.mp3')
        lidar.reset()
        if i > 10:
            break
```

รูปที่ ก.7 หลักการทำงานของอัลกอริทึมระบบการแจ้งเตือนการชนในรถจักรยานยนต์ (MWS)

จากรูปที่ ก.7 เริ่มทำการดาวน์โหลด Caffe model สำหรับ MobileNet แล้วสร้าง loop while true สำหรับการทำงานหลักของอัลกอริทึมนี้ สร้าง loop for สำหรับเก็บข้อมูลของ lidar ไว้

ในตัวแปร i และ scan สร้าง loop for สำหรับข้อมูล angle, distance ในข้อมูล scan เก็บข้อมูลระยะทางของ lidar ไว้ในตัวแปร scan_data โดยให้เก็บในรูปของจำนวนเต็ม ให้เก็บข้อมูลจากฟังก์ชั่น process_data ไว้ในตัวแปร warning และสร้างตัวแปร rangewarning ให้มีค่าเท่าจำนวนข้อมูลใน warning สร้าง loop for ให้ตัวแปร wn มีค่าเท่ากับ ช่วงของข้อมูล rangewarning ทำการกำหนดเงื่อนไขเมื่อระยะทางที่ได้ไม่เท่ากับ 0 จะกำหนดให้ตัวแปร sound มีค่าเท่ากับระยะทางที่ต่ำที่สุดในตัวแปร warning ทำการกำหนดเงื่อนไขถ้าระยะทางที่ได้ไม่เท่ากับ 0 และ sound มีค่าน้อยกว่า 3000 จะทำการแจ้งเตือนด้วยเสียง และกำหนดเงื่อนไขถ้าหาก i มีค่ามากกว่า 10 จะทำการหยุด loop

ภาคผนวก ข

หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

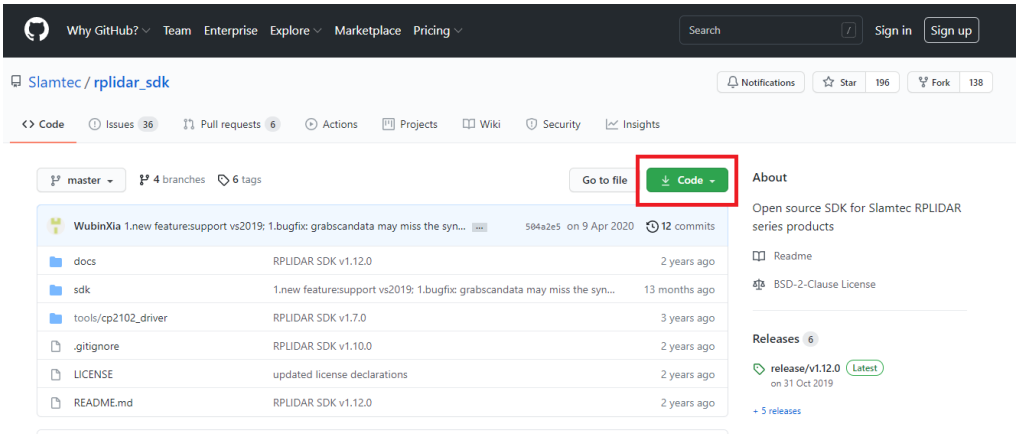
หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

โปรแกรม frame_grabber.exe เป็นโปรแกรมสำหรับทดสอบและเป็นแบบจำลองการทำงานเมื่อลิดาร์เริ่มทำการตรวจจับวัตถุบริเวณรอบๆอุปกรณ์ เป็นโปรแกรมสำหรับ RPLIDAR รุ่น A1M8 การติดตั้งนี้เป็นวิธีเฉพาะสำหรับผู้ใช้งาน Windows โดยมีขั้นตอนการติดตั้งโปรแกรมดังต่อไปนี้

1. ติดตั้ง driver โดยการเข้าไปดาวน์โหลด driver จากเว็บไซต์ของทางบริษัท SLAMTEC ซึ่งเป็นผู้ผลิต RPLIDAR จากนั้น Click ไปที่ GitHub Repository ดังรูปที่ ข.1

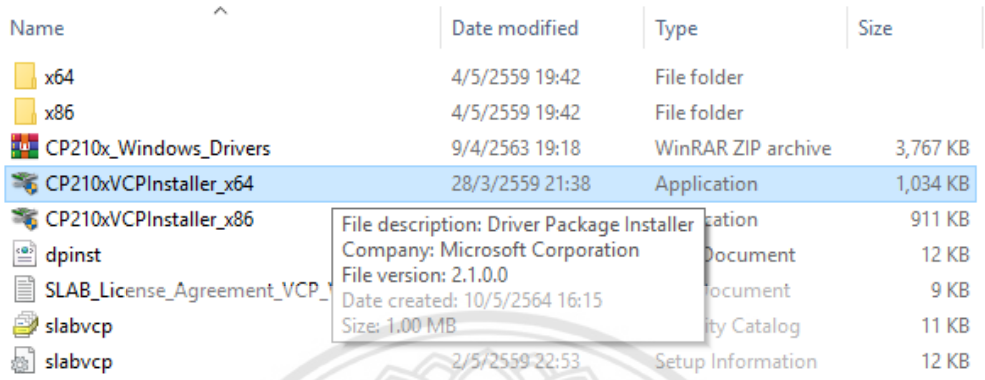
รูปที่ ข.1 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

2. ลิงก์ที่ทำการ Click จะนำไปสู่หน้าของ GitHub ให้ทำการกดเลือกที่ Code เพื่อดาวน์โหลด Code ดังรูปที่ ข.2



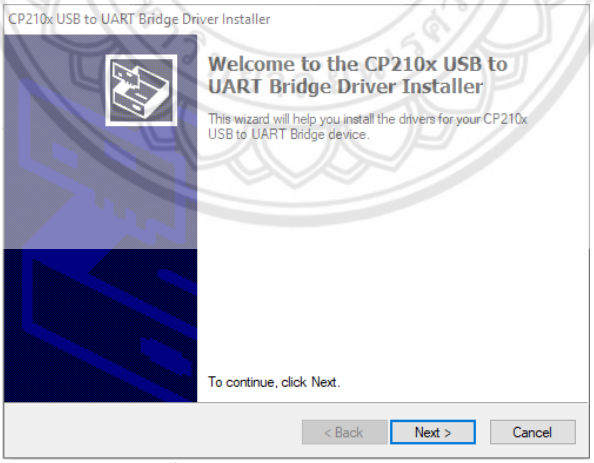
รูปที่ ข.2 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

3. ไฟล์ที่ได้จะมีลักษณะเป็นไฟล์ zip และมีชื่อว่า rplidar_sdk-master ให้ทำการ extract file และทำการ Click เข้าไปในไฟล์ rplidar_sdk-master -> tools -> cp2102_driver -> ทำการ extract file ที่มีชื่อว่า CP210x_Windows_Drivers ดังรูปที่ ข.3



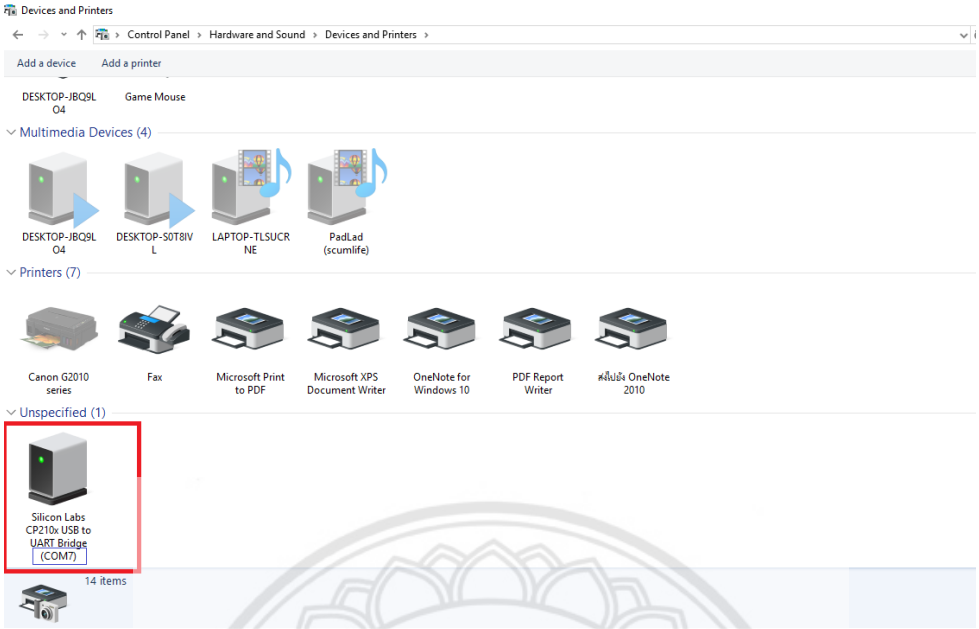
รูปที่ ข.3 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

4. ทำการ Click ที่ CP210xVCPInstaller_x64 เพื่อเริ่มทำการติดตั้ง driver เมื่อ Click แล้วจะพบกับหน้าที่แสดงการติดตั้ง driver ดังรูปที่ ข.4 ให้ทำการ Click ที่ Next -> เลือกที่ I accept this agreement แล้วกดที่ Next -> Finish



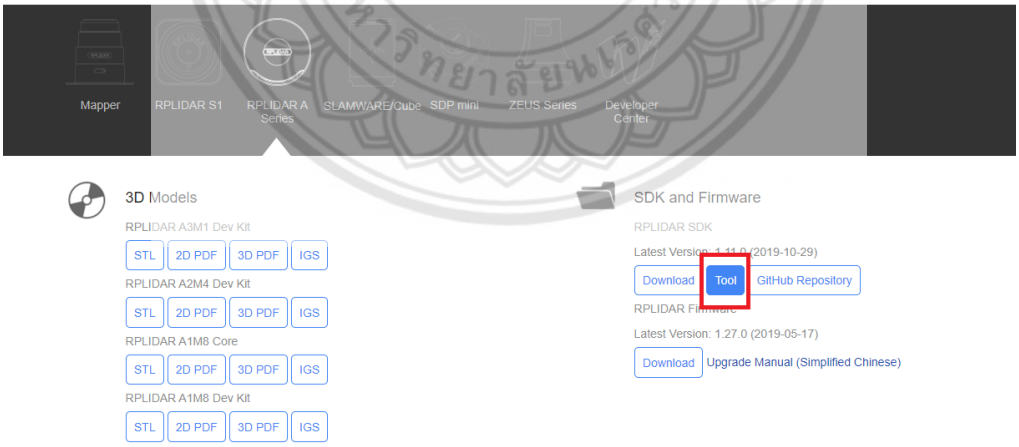
รูปที่ ข.4 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

5. ตรวจสอบ driver ที่ทำการติดตั้งโดยการเข้าไปตรวจสอบที่ Control Panel -> Hardware and Sound -> Devices and Printers จะพบกับ driver ที่มีชื่อว่า Silicon Labs CP210x USB to UART Bridge ดังรูปที่ ข.5 แสดงว่าได้ทำการติดตั้ง driver สำเร็จแล้วและสามารถบอกได้ว่า RPLIDAR นั้นเชื่อมอยู่กับ Ports ช่องใด

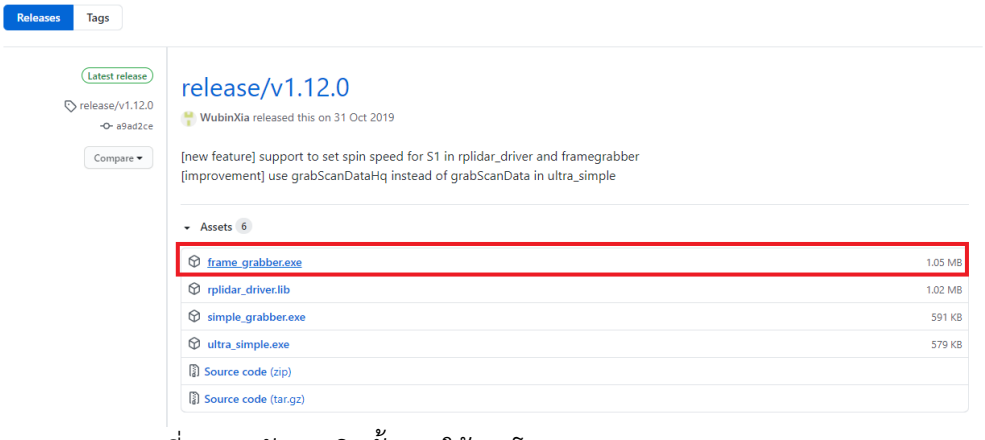


รูปที่ ข.5 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

6. ติดตั้งโปรแกรม frame_grabber.exe โดยการเข้าไปที่เว็บไซต์ของ SLAMTEC อีกครั้ง จากนั้น Click ที่ Tool ดังรูปที่ ข.6 เมื่อ Click เข้าไปแล้วจะพบกับหน้าของเว็บไซต์ GitHub ให้ทำการดาวน์โหลด frame_grabber.exe ดังรูปที่ ข.7



รูปที่ ข.6 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

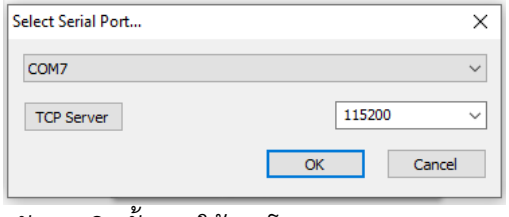


รูปที่ ข.7 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

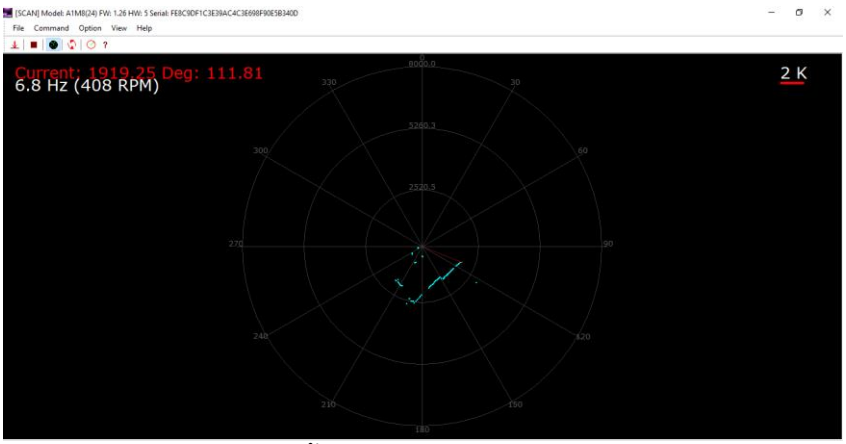
7. จากการดาวน์โหลดจะได้โปรแกรมที่มีลักษณะดังรูปที่ ข.8

รูปที่ ข.8 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

8. เริ่มการใช้งานโดยการ Click ไปที่โปรแกรม frame_grabber.exe จะพบกับหน้าแสดงการให้เลือก Serial Ports และ Band rate เลือก Ports ที่ตรงกับเครื่องของผู้ใช้งานดังรูปที่ ข.9 และเมื่อกดตกลงเพื่อเริ่มการทำงานโปรแกรมจะแสดงหน้าตาการทำงานของโปรแกรม frame_grabber.exe ดังรูปที่ ข.10 ผู้ใช้งานจำเป็นต้องทราบ Ports ที่เชื่อมกับการทำงานของ RPLIDAR โดยสามารถตรวจสอบได้จาก Control Panel



รูปที่ ข.9 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe



รูปที่ ข.10 หลักการติดตั้งและใช้งานโปรแกรม frame_grabber.exe

สำหรับฟังก์ชั่นการทำงานพื้นฐานของโปแกรม frame_grabber.exe สามารถแสดงให้เห็นดังตารางที่ ข.1

ตารางที่ ข.1 การทำงนขั้นพื้นฐานของโปรแกรม frame_grabber.exe

ปุ่มการทำงาน	ฟังก์ชั่น	หน้าที่
	Dump Data	บันทึกข้อมูลแสกน ณ ปัจจุบันที่ RPLIDAR สามารถแสกนได้
	Stop	หยุดการทำงานแสกนข้อมูลของ RPLIDAR
	Start Scan	เริ่มต้นให้ RPLIDAR ทำการแสกนข้อมูล
	Reset	Reset การทำงานและแสกนข้อมูลของ RPLIDAR
	Set motor pwm	ตั้งค่าการทำงานสำหรับมอเตอร์ของ RPLIDAR