อัลกอริทึมตรวจจับช่องทางเดินรถสำหรับยานยนต์อัตโนมัติ

มหาวิทยาลัยนเรศวร

Lane Detection for NU Autonomous Vehicle Driving Control

นาย ทัศนัย อ่ำกำเนิด รหัสนิสิต 58362322

นาย ธนกร เทศนะ รหัสนิสิต 58362339

นาย ปฐวี ขอบเขต รหัสนิสิต 58362513

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2561

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน อัลกอริทึมตรวจจับช่องทางเดินรถสำหรับยานยนต์อัตโนมัติ มหาวิทยาลัยนเรศวร

ผู้ดำเนินโครงงาน

นาย ทัศนัย	อ่ำกำเนิด	รหัส 5836232 2
นาย ธนกร	เทศนะ	รหัส 5836233 9
นาย ปฐวี	ขอบเขต	รหัส 5836251 3

ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ชูพงศ์ ช่วยเพ็ญ

สาขาวิชา วิศวกรรมเครื่องกล

ภาควิชา วิศวกรรมเครื่องกล

ปีการศึกษา 2561

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมเครื่องกล

.....ที่ปรึกษาโครงงาน
(ผู้ช่วยศาสตราจารย์ ชูพงศ์ ช่วยเพ็ญ)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อนันต์ชัย อยู่แก้ว)

.....กรรมการ
(ดร.ปองพันธ์ โอทกานนท์)

ชื่อหัวข้อโครงงาน อัลกอริทึมตรวจจับช่องทางเดินรถสำหรับยานยนต์อัตโนมัติ มหาวิทยาลัยนเรศวร

ผู้ดำเนินโครงงาน นาย ทัศนัย อ่ำกำเนิด รหัส 58362322
นาย ธนกร เทศนะ รหัส 58362339
นาย ปฐวี ขอบเขต รหัส 58362513

ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ชูพงศ์ ช่วยเพ็ญ

สาขาวิชา วิศวกรรมเครื่องกล

ภาควิชา วิศวกรรมเครื่องกล

ปีการศึกษา 2561

บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนาอัลกอริทึมตรวจจับช่องทางเดินรถจากโครงการในปีการศึกษา 2560 ของนิสิตชั้นปีที่ 4 คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมเครื่องกล มหาวิทยาลัยนเรศวร จากการใช้แบบจำลองทางเดินรถให้สามารถใช้งานได้จริงกับทางเดินรถจริงรอบมหาวิทยาลัยนเรศวร โดยทำการทดลองกับรูปทางเดินรถภายในบริเวณมหาวิทยาลัยนเรศวร

การทดลองใช้การเก็บข้อมูลทางเดินรถ เป็นข้อมูลของเส้นในรูปของโคเอฟฟิเชียนท์ (Coefficient) ผลของโครงการนี้เป็นการเปรียบเทียบการใช้ค่าโคเอฟฟิเชียนท์ (Coefficient) จากข้อมูลของ Dataset และการใช้ Trained-Network ในการประมวลผลเส้นและจุดกึ่งกลางของทางเดินรถ ซึ่งใช้ข้อมูลที่ได้จาก Dataset ได้ผลที่น่าพึงพอใจ อัลกอริทึมสามารถสร้างเส้นและจุดกึ่งกลางที่ต้องการได้ แต่ค่าโคเอฟฟิเชียนท์ (Coefficient) จากการใช้ Trained-Network ยังมีความคาดเคลื่อนอยู่จึงยังไม่สามารถนำมาใช้งานได้

Project title	Lane detection for NU autonomous vehicle driving control		
Name	Mr.Thatsanai	Amkamnoed	Student ID. 58362322
	Mr.Tanakorn	Tassana	Student ID. 58362339
	Mr.Pathawee	Khobkhet	Student ID. 58362513
Project advisor	Assistant Professor Choopong Chuaypen		
Major	Mechanical Engineering		
Department	Mechanical Engineering		
Academic year	2018		

Abstract

This project aims to develop an algorithm to detect lane traffic from the project in the academic year 2017 of the 4th year students of the Faculty of Engineering. Mechanical Engineering Naresuan University From using the lane traffic model to be able to actually use with the actual route around Naresuan University By experimenting with the image of the interior of Naresuan University

The experiment using data collection Is the data of the lines in the form of Coefficient. The result of this project is a comparison of the use of coefficients from the data of Dataset and the use of Trained-Network. In the processing of lines and midpoints of the lane traffic. Which uses the data obtained from Dataset to have satisfactory results. The algorithm can create the desired line and midpoint, but the coefficient of using Trained-Network There are still discrepancy, therefore cannot be used.

กิตติกรรมประกาศ

โครงงานเรื่องอัลกอริทึมตรวจจับช่องทางเดินรถสำหรับยานยนต์ อัตโนมัติมหาวิทยาลัยนเรศวรปี 2561 จะประสบความสำเร็จไม่ได้ หากขาดบุคคลที่ช่วยเหลือ ให้คำปรึกษา แนะนำ และความอนุเคราะห์ในการดำเนินโครงงานตลอดมาจนประสบความสำเร็จ ทางคณะผู้ดำเนินโครงงานต้องขอขอบพระคุณบุคคลดังต่อไปนี้

1. พ่อและแม่ ที่คอยอบรมสั่งสอน และคอยสนับสนุนในทุก ๆ ด้าน
2. อาจารย์ชูพงศ์ ช่วยเพ็ญ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ขอขอบพระคุณสำหรับการให้คำปรึกษา คำแนะนำ และช่วยจัดหาอุปกรณ์สำคัญในการทำโครงงานนี้ ตลอดจนช่วยดูแลจนโครงงานสำเร็จลุล่วง
3. คณาจารย์ บุคลากร และเพื่อน ๆ ทุกคนสำหรับคำปรึกษา แนะนำ และให้ความอนุเคราะห์ในการดำเนินโครงงานนี้

คณะผู้ดำเนินโครงงาน

สารบัญ

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อ ..... ข
Abstract ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญรูปภาพ ..... ช
สารบัญตาราง ..... ฌ
บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญ 1
1.2 วัตถุประสงค์ของการศึกษา 2
1.3 ประโยชน์ที่คาดว่าจะได้รับ 2
1.4 ขอบเขตของโครงงาน 2
1.5 ขั้นตอนการดำเนินงาน 3
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 4
2.1 Artificial Intelligence (AI) 4
2.2 Machine Learning 6
2.3 โครงข่ายประสาทเทียม (Neural Networks) 8
2.3.1 Classification of Neural Networks 8
2.4 Deep learning 8
2.4.1 กระบวนการทำงานของ Deep Learning 9
2.4.2 การเรียนรู้ของ Deep learning 9
2.4.3 ชนิดของโครงข่าย Deep learning (Types of Deep Learning Networks) 9
2.4.4 การใช้งานโดยทั่วไปของ RNN 10
2.4.5 ความสำคัญของ Deep learning 10
2.4.6 ข้อจำกัดของ deep learning (Limitation of deep learning) 11
2.5 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Networks :CNN) 11

สารบัญ(ต่อ)

2.6 วรรณกรรมปริทรรศน์ 13
บทที่ 3 วิธีการดำเนินงาน 16
3.1 การเก็บข้อมูลและการเตรียมข้อมูลภาพ 16
3.2 การเตรียมข้อมูลสำหรับการ Train-Network การตรวจจับช่องทางเดินรถ และหาจุดกึ่งกลาง 18
3.3 อัลกอริทึมที่ใช้งาน 20
3.3.1 Code PART 1 20
3.3.2 Code PART 2 22
3.3.3 Code PART 3 22
3.4 การตรวจเช็คข้อมูล Dataset 23
3.5 สรุปการทำงานของอัลกอริทึม 25
บทที่ 4 ผลการทดลองและวิเคราะห์ผลการทดลอง 26
4.1 ผลการทดลอง 26
4.1.1 ผลเปรียบเทียบโดยใช้ code 2 26
4.1.2 ผลเปรียบเทียบโดยใช้ code 3 29
4.2 วิเคราะห์ผลการทดลอง 33
บทที่ 5 สรุปผลโครงงานและข้อเสนอแนะ 34
5.1 สรุปผลโครงงาน 34
5.2 ปัญหาที่พบ 35
5.3 ข้อเสนอแนะ 35
บรรณานุกรม 36
ประวัติผู้ดำเนินโครงงาน 39

สารบัญรูปภาพ

รูปที่ 2.1 แขนงย่อยของ AI 4
รูปที่ 2.2 องค์ประกอบของAI 5
รูปที่ 2.3 ประเภทของ Machine learning 6
รูปที่ 2.4 โครงข่ายประสาทแบบคอนโวลูชั้น CNN 12
รูปที่ 3.1 ภาพอ้างอิงระยะเบรกที่ปลอดภัย 17
รูปที่ 3.2 การวาดเส้นในแอพพลิเคชั่น Ground Truth 19
รูปที่ 3.3 แสดงการทำงานในการสร้างข้อมูล Dataset 19
รูปที่ 3.4 ส่วนที่ 1 ของ Code PATH 1 20
รูปที่ 3.5 ส่วนที่ 2 ของ Code PATH 1 21
รูปที่ 3.6 เนื้อหาบางส่วนของ Code PATH 2 22
รูปที่ 3.7 ภาพจาก Ground Truth (ด้านซ้าย) เปรียบเทียบกับภาพจากใช้ Code PART 3 (ด้านขวา) 24
รูปที่ 3.8 ผังการทำงานรวมของระบบ 25
รูปที่ 4.1 แสดงเส้น บริเวณที่มีเงาบางส่วน 26
รูปที่ 4.2 แสดงเส้น บริเวณที่มีเงาบางส่วน 26
รูปที่ 4.3 แสดงเส้น บริเวณที่มีแสงสว่างมาก 26
รูปที่ 4.4 แสดงเส้น บริเวณที่มีแสงสว่างมาก 26
รูปที่ 4.5 ตำแหน่งเส้น บริเวณที่มีเงาเป็นส่วนใหญ่ 27
รูปที่ 4.6 ตำแหน่งเส้น บริเวณที่มีเงาเป็นส่วนใหญ่ 27
รูปที่ 4.7 ตำแหน่งเส้น บริเวณที่มีแสงน้อย 27
รูปที่ 4.8 ตำแหน่งเส้น บริเวณที่มีแสงน้อย 27
รูปที่ 4.9 ตำแหน่งเส้น บริเวณทางโค้งซ้าย 27
รูปที่ 4.10 ตำแหน่งเส้น บริเวณทางโค้งซ้าย 27
รูปที่ 4.11 ตำแหน่งเส้น บริเวณที่มีแสงมากและเงาบางส่วน 28
รูปที่ 4.13 ตำแหน่งเส้น บริเวณทางโค้งขวา 28

สารบัญรูปภาพ (ต่อ)

รูปที่ 4.14 ตำแหน่งเส้น บริเวณทางโค้งขวา..... 28
รูปที่ 4.15 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงสว่างมาก 29
รูปที่ 4.16 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงสว่างมาก 29
รูปที่ 4.17 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีเงาบางส่วน 29
รูปที่ 4.18 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีเงาบางส่วน 29
รูปที่ 4.19 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงมากและเงาบางส่วน..... 30
รูปที่ 4.20 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงมากและเงาบางส่วน..... 30
รูปที่ 4.21 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีเงาเป็นส่วนใหญ่..... 30
รูปที่ 4.22 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีเงาเป็นส่วนใหญ่..... 30
รูปที่ 4.23 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงน้อย..... 31
รูปที่ 4.24 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงน้อย..... 31
รูปที่ 4.25 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณทางโค้งขซ้าย..... 31
รูปที่ 4.26 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณทางโค้งขซ้าย..... 31
รูปที่ 4.27 ตำแสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณทางโค้งขวา..... 32
รูปที่ 4.28 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณทางโค้งขวา 32

สารบัญตาราง

ตารางที่ 1 ตารางแสดงขั้นตอนการดำเนินงาน..... 3
ตารางที่ 2 แสดงระยะเบรกที่ปลอดภัย..... 16

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ทุกวันนี้มีรถยนต์จำนวนมหาศาลวิ่งอยู่บนถนน ผลที่ตามมาก็คืออุบัติเหตุซึ่งเพิ่มขึ้นเป็นอย่างมากจากความประมาทของมนุษย์ เพื่อลดอุบัติเหตุที่เกิดจากเหตุการณ์อันตรายที่เกิดขึ้นโดยไม่ได้ตั้งใจหรือไม่คาดคิดมาก่อนทำให้เกิดความเสียหายแก่ทรัพย์สินของตนเองและผู้อื่น จากสาเหตุหลักของการเกิดอุบัติเหตุมากจากพฤติกรรมของผู้ขับขี่ การหลับใน โทรศัพท์ขณะขับรถ ด้วยเหตุดังกล่าวบรรดาผู้ผลิตรถยนต์จึงเริ่มคิดค้นระบบ ช่วยเหลือในการขับขี่เพื่อให้การขับขี่นั้นมีความปลอดภัยมากขึ้นโดยระบบช่วยเหลือในการขับขี่มีด้านในหลายลักษณะ เช่น ระบบเบรกป้องกันล้อล็อค (ABS) ระบบที่กระจายแรงเบรก (EBD) ซึ่งทั้งสองระบบจะช่วยในการหยุดรถและการควบคุมในขณะเบรก ระบบช่วยเหลือการขับขี่ทั้งหลายเมื่อนำมาทำงานร่วมกันก็จะเป็นขั้นสุดท้ายคือระบบขับขี่อัตโนมัตินั่นเอง การตรวจจับช่องทางเดินรถ (Lane Detection) หรือระบบเตือนการออกนอกเลน (Lane Departure Warning System, LDWS) เป็นหนึ่งในระบบที่มีประโยชน์ในด้านความปลอดภัยของการขับขี่และถือเป็นระบบที่มีความสำคัญเป็นอย่างยิ่งในระบบขับเคลื่อนอัตโนมัติ

การใช้งานการตรวจจับช่องทางการเดินรถจะมีการใช้งานในระบบรักษาช่องทางการเดินรถ (Lane Keeping) ซึ่งจะทำ หน้าที่ค่อยบังคับให้รถวิ่งตามเลนที่ถูกต้อง การนำเอาเทคโนโลยีเข้ามาเป็นตัวช่วยในการควบคุมระบบ บังคับทิศทาง เช่นการควบคุมพวงมาลัยขณะที่ผู้ใช้งานต้องการพักในขณะเมื่อยล้า ระบบ ดังกล่าวจำเป็นต้องรู้ลักษณะของถนนในพื้นที่การใช้งาน การตรวจจับช่องทางเดินรถ การตรวจจับแนวขอบทาง การตรวจจับสิ่งกีดขว้าง การตรวจจับรถอื่น ๆ บนท้องถนน เทคโนโลยีอย่างหนึ่งที่สามารถนำมาประยุกต์ใช้ได้คือการประมวลผลภาพ (Image Processing) ซึ่งมีทั้งขั้นตอนและเทคนิค การประยุกต์ใช้งานที่หลากหลาย ตลอดจนการประยุกต์ใช้ระบบอัจฉริยะ (Artificially Intelligent) ช่วยในการประมวลผลภาพเพื่อวัตถุประสงค์ต่าง ๆ เมื่อระบบสามารถวิเคราะห์ข้อมูลต่าง ๆ ในช่อง ทางการเดินรถแล้วจะส่งข้อมูลให้แก่ระบบบังคับเลี้ยวของรถ ซึ่งมีความจำเป็นในการพัฒนาตัวควบคุม ให้สามารถทำงานได้อย่างสอดคล้องกับข้อมูลที่ได้จากการประมวลผลภาพดังกล่าว

จากโครงงานอัลกอริทึมตรวจจับช่องทางเดินรถสาหรับยานยนต์อัตโนมัติ มหาวิทยาลัยนเรศวร (Lane Detection For NU Autonomous Vehicle Driving Control) ซึ่งเป็นโครงงานของนิสิตวิศวกรรมเครื่องกล ปีการศึกษา 2560 โดยโครงงานใช้หลักการด้านการประมวลผลภาพเพื่อใช้ในการตรวจจับเส้นแบ่งช่องทางโดยทำการทดสอบกับแบบจำลองของถนนซึ่งผลการตรวจจับอยู่ในเกณฑ์ที่ดีดังนั้นผู้จัดทำโครงการจึงต้องการทำการพัฒนาปรับปรุงโครงงานดังกล่าวให้สามารถนำมาใช้ได้จริงในสภาพแวดล้อมจริงในเส้นทางการเดินรถในมหาวิทยาลัยนเรศวร และในโครงงานนี้จึงมุ่งเน้นการประยุกต์และพัฒนากระบวนการในการประมวลผลภาพ เพื่อใช้ในการวิเคราะห์หาขอบทางช่องทางการเดินรถ ขนาดหรือระยะห่างระหว่างช่องทางเดินรถกับขอบทาง (ฟุตบาท) ตลอดจนการควบคุม กระบวนการและกลไกการควบคุมระบบการบังคับเลี้ยวเพื่อการควบคุมให้สามารถเดินรถอยู่ในช่องทางเดินรถที่กำหนดมา

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 ศึกษาและประยุกต์ใช้การประมวลผลภาพ เพื่อวิเคราะห์และบ่งชี้ช่องทางการเดินรถในสภาพแวดล้อมจริง

1.2.2 ศึกษาและพัฒนาอัลกอริทึมการควบคุมบังคับทางการเดินรถในช่องทางที่กำหนด

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 ความรู้ความเข้าใจในการใช้เทคนิคการประมวลผลภาพเพื่อวัตถุประสงค์ต่าง ๆ ตามเทคนิคที่เหมาะสม

1.3.2 กระบวนการและอัลกอริทึมในการวิเคราะห์ช่องทางการเดินรถของถนนในบริเวณมหาวิทยาลัยนเรศวร

1.3.3 กระบวนการในการควบคุมรถให้อยู่ในช่องทางการเดินรถที่เหมาะสม

1.4 ขอบเขตของโครงงาน

1.4.1 ศึกษาหาตัวแปรที่มีผลต่อประสิทธิภาพของอัลกอริทึมที่ได้พัฒนาขึ้นมาจากเดิม

1.4.2 ออกแบบการประมวลภาพและวิเคราะห์หาขอบของถนน

1.4.3 พัฒนาอัลกอริทึมให้สามารถนำไปประยุกต์ใช้กับยานยนต์ในการควบคุมการเดินรถให้อยู่ในช่องทางเดินรถ

1.5 ขั้นตอนการดำเนินงาน

1.5.1 ศึกษาหลักการและทฤษฎีที่เกี่ยวข้องจากโครงงาน Lane Detection For NU Autonomous Vehicle Driving Control และวิเคราะห์ข้อดี ข้อด้อยของกระบวนการที่ใช้

1.5.2 ออกแบบอัลกอริทึมในการวิเคราะห์เส้นกึ่งกลางถนน ขอบฟุตบาท และองค์ประกอบที่จำเป็น รวมไปถึงการควบคุมบังคับรถให้อยู่ในช่องทางเดินรถ

1.5.3 ทำการทดสอบอัลกอริทึมที่ได้สร้างและปรับปรุง

1.5.4 ปรับปรุงและพัฒนาแบบอัลกอริทึมให้ใช้งานได้จริงกับภาพและวิดีโอ และมีความเสถียร

1.5.5 กำหนดเกณฑ์และรูปแบบการทดลอง เพื่อทดสอบและประเมินผลความสำเร็จของอัลกอริทึมที่ได้พัฒนา

1.5.6 สรุปผลการดำเนินโครงงานและจัดทำเล่มปริญญานิพนธ์

1.6 แผนการดำเนินงาน

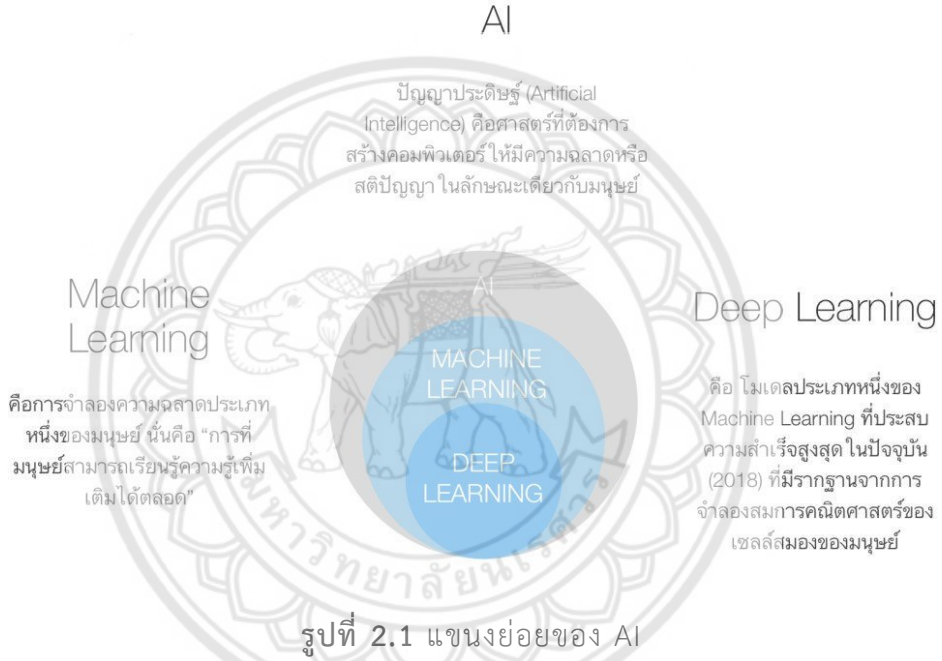
ตารางที่ 1 ตารางแสดงขั้นตอนการดำเนินงาน

กิจกรรม	พ.ศ. 2561						พ.ศ. 2562					
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.
1. ศึกษาหลักการและทฤษฎีที่เกี่ยวข้อง												
2. พัฒนาและออกแบบอัลกอริทึม												
3. ทดสอบประสิทธิภาพของระบบอัลกอริทึม												
4. การปรับปรุงและพัฒนาแบบอัลกอริทึม												
5. วิเคราะห์ และประเมินข้อมูล												
6. สรุปผลและจัดทำรูปเล่ม												

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 Artificial Intelligence (AI)

Artificial Intelligence (AI) หรือ ปัญญาประดิษฐ์ คือศาสตร์ที่ต้องการสร้างคอมพิวเตอร์ให้มีความฉลาดหรือสติปัญญาในลักษณะเดียวกับมนุษย์ ทั้งในแง่ สติปัญญา ความสามารถที่มนุษย์สามารถทำได้ ความคิดเชิงเหตุและผล รวมทั้งความคิดเชิงสร้างสรรค์ และทักษะทางสังคม เป็นต้น

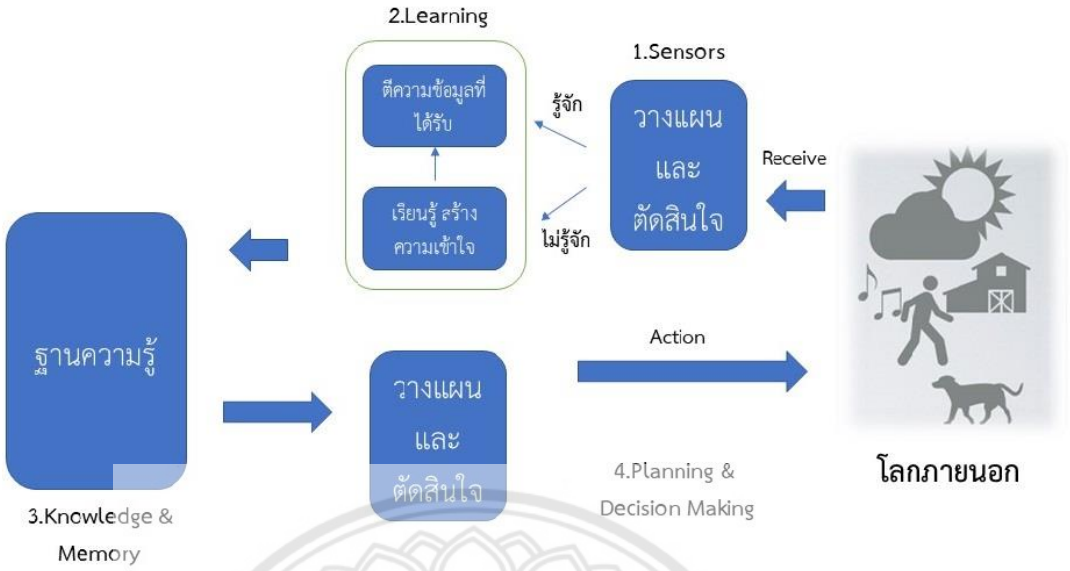


รูปที่ 2.1 แขนงย่อยของ AI

ที่มา : https://thaikeras.com/2018/11/10/ai-ml-dl-relationship/

AI นั้นเป็นชื่อที่กว้างที่สุดซึ่งหมายถึงศาสตร์แขนงใหม่ที่พยายามศึกษาและสร้างความฉลาดขึ้นมาและ Machine Learning ก็คือแขนงย่อยของ AI และ Deep Learning ก็เป็นแขนงย่อยของ Machine Learning ดังแสดงดังรูป 2.1

องค์ประกอบทั้งสี่ส่วนของ AI ดังแสดงในรูปที่ 2.2 ที่ทำให้ AI สามารถตอบสนองต่อโลกภายนอกได้ในทำนองเดียวกับมนุษย์นั้นสามารถเข้าใจได้ง่ายๆ ดังนี้



รูปที่ 2.2 องค์ประกอบของAI

ที่มา : https://thaikeras.com/2018/11/10/ai-ml-dl-relationship/

1) Sensor หรือ ส่วนรับข้อมูล การที่ AI หรือหุ่นยนต์จะตอบสนองต่อโลกภายนอกได้ ก่อนอื่นต้องสามารถรับข้อมูลจากโลกภายนอกได้ก่อน มนุษย์เรารับรู้แสง เสียง กลิ่น รสชาติ และการสั่นสะเทือนผ่านดวงตา หู จมูก ปาก และการสัมผัสทางร่างกาย ส่วนหุ่นยนต์เองก็สามารถรับข้อมูลต่าง ๆ เหล่านี้ได้เช่นเดียวกับมนุษย์โดยผ่าน Sensor แต่ละประเภทนั่นเอง เช่น ภาพจากกล้องหรือวิดิโอทำให้หุ่นยนต์รับข้อมูลภาพได้เช่นเดียวกับดวงตามนุษย์

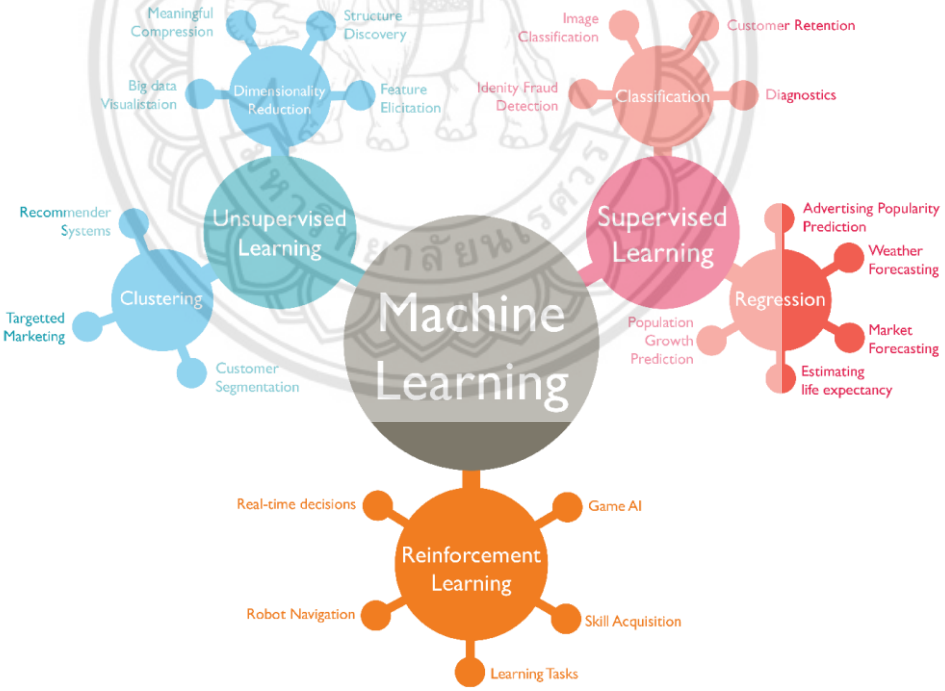
2) Learning หรือ ส่วนเรียนรู้และตีความข้อมูล เมื่อได้รับข้อมูล เช่น ภาพหรือเสียงจากโลกภายนอกแล้วก็จำเป็นต้องตีความว่า ภาพหรือเสียงนั้นคืออะไรและมีความหมายอย่างไรกันแน่ เช่น เป็นภาพและเสียงของคนที่กำลังพยายามพูดจาสื่อสารกับเรา (ซึ่งก็คือหุ่นยนต์หรือ AI) อยู่หรือไม่ จริง ๆ แล้วสาขา Machine Learning (รวมทั้ง Deep Learning) นั้นก็คือแขนงความรู้ในส่วนนี้นั่นเอง

3) Memory & Knowledge Base หรือ ส่วนฐานข้อมูลและหน่วยความจำ เช่นเดียวกับมนุษย์ที่มีสามารถจดจำข้อมูล ความรู้ต่าง ๆ ที่ผ่านมาในอดีตไว้ในสมอง คอมพิวเตอร์เองก็จำเป็นต้องมีส่วน Knowledge Base นี้ไว้เพื่อจดจำความรู้ต่าง ๆ ที่ได้เรียนรู้ในเวลาที่ผ่านมาเพื่อนำไปช่วยในการตีความในส่วน Learning หรือใช้ในการตัดสินใจแก้ปัญหาต่าง ๆ ในส่วน Planning & Decision Making

4) Planning & Decision Making หรือ ส่วนวางแผนและตอบสนองไปยังโลกภายนอก เมื่อเข้าใจความหมายของข้อมูลจากโลกภายนอกเช่นภาพและเสียงโดยสมบูรณ์แล้ว (เช่น รับรู้ได้ว่ามีคนกำลังถามเส้นทางไปโรงพยาบาล) AI ก็จำเป็นต้องคิดว่าจะตอบสนองเช่นไรดี (เช่น คำนวนและสืบค้นหาโรงพยาบาลที่ใกล้ที่สุดจากฐานความรู้ และกลั่นเส้นทางนั้นออกมาเป็นคำพูดสื่อสารกลับไปยังผู้ถาม)

2.2 Machine Learning

Machine Learning คือ การทำให้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเอง โดยใช้ “ข้อมูล” ซึ่งมันแตกต่างกับการเขียนโปรแกรมทั่วไป เพราะ Programming เราจะใส่ ข้อมูล (Data) และ Program เข้าไปเพื่อให้ได้ Output แต่ Machine Learning เราไม่ได้ Program คำตอบ เราใส่ Data และ Output (ผลลัพธ์) เข้าไป เพื่อให้หา Program ที่จะนำไปตอบในอนาคตได้ว่า Input แบบนี้ Output จะเป็นอะไร การใช้ข้อมูล ใช้ได้หลายแบบ ซึ่งมันจะแบ่งตามประเภทของ Machine Learning ประเภทของ Machine Learning แสดงดังนี้



รูปที่ 2.3 ประเภทของ Machine learning

ที่มา : https://blog.finnomena.com/machine-learning-คืออะไร-fa8bf6663c07

จากรูปที่ 2.3 Machine Learning แบ่งออกเป็น 3 แบบ ดังนี้

- Supervised Learning คือ เรียนรู้โดยมี data มาสอน
- Unsupervised Learning คือ เรียนรู้โดยไม่มี data สอน
- Reinforcement Learning คือ เรียนรู้ตามสภาพแวดล้อม

จากรูปที่ 2.1 (AI-> Machine Learning) เราจะเห็นว่า Machine Leaning นั้นก็คือองค์ประกอบในส่วนที่สองของ AI นั่นเอง ซึ่งเป็นส่วนที่สำคัญมาก ๆ เนื่องเพราะพวกเราเข้าใจกันดีอยู่แล้วตัวชี้วัดหนึ่งที่บอกว่ามนุษย์นั้นมีความฉลาดยิ่งกว่าสัตว์ประเภทอื่น ๆ นั่นก็เพราะมนุษย์สามารถ “เรียนรู้” เรื่องต่าง ๆ ได้ในระดับสูง

โดยรวมนั้นหน้าที่ของ Machine Leaning มีดังต่อไปนี้

เรียนรู้สร้างความเข้าใจในสิ่งที่ไม่รู้จัก เมื่อ AI รับข้อมูลจากโลกภายนอกและไม่สามารถตีความได้ด้วยความรู้ที่มีอยู่ ยกตัวอย่างเช่นถูกสั่งให้เล่นหมากรุก โดยไม่เคยเล่นมาก่อน ในสถานการณ์แบบนี้ AI จำเป็นต้องเรียนรู้ concepts ต่าง ๆ ใหม่ทั้งหมด ซึ่งการเรียนรู้ประเภทนี้ค่อนข้างยาก และปัญหาที่พวกเราเจอส่วนใหญ่จะสามารถเปลี่ยนเป็นปัญหาที่ง่ายกว่านี้ได้

การตีความหมายของข้อมูลที่ได้รับ ในการใช้งานจริงทั่วไปของ Machine Leaning อาจจะหมายถึง “การตีความว่าข้อมูลจากโลกภายนอก” เช่น

การตีความว่า “รูป” ที่เห็นคือรูปของอะไร โดยใน Machine Leaning จะถูกสอนมาก่อนแล้วว่าสิ่งต่าง ๆ ที่เราสนใจมีอะไรบ้าง

การตีความว่า “เสียง” ที่ได้ยินว่าเป็นประโยคภาษาอะไร และแปลว่าอะไร ผู้พูดต้องการจะสื่ออะไร ในงานนี้ก็เช่นกัน Machine Leaning จะถูกสอนเรื่องคำศัพท์ต่าง ๆ ในแต่ละภาษาและตัวอย่างการอ่านออกเสียงมาก่อนแล้ว ทำให้สามารถตีความ “เสียงอื่น ๆ” ได้

การตีความหลายขั้นตอน เช่น “การอ่านและทำความเข้าใจเอกสาร” นั้น Machine Leaning ต้องตีความว่าแต่ละรูปที่เห็นประกอบไปด้วยตัวอักษรอะไรบ้าง และตัวอักษรเหล่านั้นประกอบกันเป็นคำอะไรบ้าง และแต่ละคำมาเรียงกันเป็นประโยคนั้นมีความหมายว่าอย่างไรบ้าง และท้ายที่สุดเมื่อนำหลายๆ ประโยคในเอกสารนั้นมาเรียงต่อกัน เพื่อใจความสำคัญของเอกสารนั้น ๆ

2.3 โครงข่ายประสาทเทียม (Neural Networks)

โครงข่ายประสาทเทียม หรือ ข่ายงานประสาทเทียม (อังกฤษ: artificial neural network) คือ โมเดลทางคณิตศาสตร์หรือโมเดลทางคอมพิวเตอร์สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนคชันนิสต์ (connectionist) แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพ (bioelectric network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท (neurons) และ จุดประสานประสาท (synapses) ตามโมเดลนี้ ข่ายงานประสาทเกิดจากการเชื่อมต่อระหว่างเซลล์ประสาท จนเป็นเครือข่ายที่ทำงานร่วมกัน

กล่าวคือ โครงข่ายประสาทเทียม (Neural Networks) คือ โมเดลทางคณิตศาสตร์หรือโมเดลทางคอมพิวเตอร์สำหรับประมวลผลสารสนเทศด้วยการคำนวณร่วมกันจนเป็นเครือข่าย

2.3.1 Classification of Neural Networks

1) โครงข่ายประสาทแบบตื้น (Shallow neural network) มี Hidden layer แค่ 1 layer (แต่ยังมี input layer และ output layer ประกบเหมือนเดิม)

2) โครงข่ายประสาทแบบลึก (Deep neural network) มี Hidden layer มากกว่า 1-layer ยกตัวอย่างเช่น LeNet model ของ Google ที่ใช้สำหรับ การรับรู้ภาพ (image recognition) มี hidden layer มากถึง 22 layer

2.4 Deep learning

Deep learning คือ Software คอมพิวเตอร์ที่เลียนแบบการทำงานของระบบโครงข่ายประสาท (neurons) ในสมองมนุษย์ ถือเป็นซับเซ็ตของ Machine learning

Algorithm ของ Deep learning ถูกสร้างขึ้นจากการนำเอา Neural Network หลายๆ layer มาต่อกัน โดย layer แรกสุดจะทำหน้าที่ในการรับข้อมูล (Input layer) layer สุดท้ายจะทำหน้าที่ส่งผลลัพธ์การประมวลผลออกมา (Output layer) ส่วน layer ระว่าง layer แรกสุดและ layer สุดท้าย จะถูกเรียกว่า Hidden layer

คำว่า Deep learning มีที่มาจากการที่ใช้ layer ของ neural network หลายอันๆมาต่อกัน (โดยถ้ามี Hidden layer มากกว่า 2 layer ก็ถือว่าเป็น Deep learning) เนื่องจาก layer เหล่านี้เป็นโครงสร้างที่ถูกจัดเก็บแบบเป็นกองซ้อน (stack) จึงเปรียบได้ว่า layer ที่จะนวนเยอะๆ ก็จะทำให้มีโครงสร้างที่ลึก(deep) ยิ่งขึ้นนั่นเอง

โดย Hidden layer ของแต่ละ layer จะเปรียบเสมือนว่าประกอบด้วย เซลล์ประสาท (Neural) จำนวนมาก ซึ่งมีหน้าที่ในการประมวลผลโดยรับข้อมูลจาก layer ที่อยู่เหนือกว่า และส่งข้อมูลที่ประมวลผลเสร็จแล้วไปยัง layer ที่อยู่ล่างกว่า , ข้อดีของการส่งข้อมูลแบบนี้ก็คือ layer แต่ละ layer สามารถที่จะมี ค่าถ่วงน้ำหนัก (weight), ค่าความเอนเอียงของข้อมูล (bias) และ วิธีการประมวลผลทางคณิตศาสตร์ (activation function) ที่เป็นอิสระต่อกันได้
ซึ่งยิ่งเราป้อนข้อมูลให้กับ model มากเท่าไหร่ layer แต่ละ layer ก็จะสามารถสกัดลักษณะข้อมูลที่มีความซับซ้อนมากยิ่งขึ้น

2.4.1 กระบวนการทำงานของ Deep Learning

Model ที่ใช้ Deep Learning ให้ความแม่นยำ (Accuracy) ที่สูงในหลายๆปัญหา ตั้งแต่การตรวจจับวัตถุ (Object detection) ไปจนถึงการรู้จำเสียงพูด (Speech Recognition) โดยที่เราไม่จำเป็นต้องให้ความรู้พื้นฐานใด ๆ กับมันไว้ล่วงหน้าเลย เพียงแค่ให้ข้อมูลตัวอย่าง (input data) มันก็จะทำการเรียนรู้จากข้อมูลและสังเคราะห์เป็นองค์ความรู้ออกมาได้อย่างอัตโนมัติ อาทิเช่น การใช้ Deep Learning ในวงการเกม เราไม่จำเป็นต้องบอกมันว่าเล่นยังไง แค่ให้อัลกอริทึมเรียนรู้จากผู้เล่นที่เก่งๆเป็นจำนวนมาก มันก็เรียนรู้วิธีการเล่นเกมได้อย่างอัตโนมัติ

2.4.2 การเรียนรู้ของ Deep learning

การเรียนรู้ของ Deep learning นั้น เกิดขึ้นจาก 2 ระยะ โดยระยะแรกคือ การประยุกต์ใช้การแปลงแบบไม่เป็นเชิงเส้น (Nonlinear transformation) กับข้อมูลที่ได้รับ (Input) ได้ผลลัพธ์ (Output) ออกมาอยู่ในรูปของโมเดลทางสถิติ (statistical model) ระยะที่สอง คือ การนำโมเดลมาผ่านวิธีการทางคณิตศาสตร์ อย่าง derivative หรือ การ Differential นั้นเอง

โดยทั้ง 2 ระยะนี้จะถูกทำซ้ำแล้วซ้ำเล่าจนกว่าจะได้มาซึ่งโมเดลที่ความแม่นยำ (Accuracy) ในระดับที่น่าพึงพอใจ ซึ่งการทำซ้ำ ๆ ของ 2 ระยะนี้ มีชื่อว่า Iteration

2.4.3 ชนิดของโครงข่าย Deep learning (Types of Deep Learning Networks)

1) โครงข่ายประสาทแบบป้อนไปหน้า (Feed-forward Neural Networks)

Feed-forward neural networks ถือเป็นโมเดลที่มีโครงสร้างที่เรียบง่ายที่สุด เพราะว่าการดำเนินการของข้อมูลจะเป็นไปในทิศทางเดียว ก็คือ รับข้อมูลจาก input layer แล้วส่งไปต่อไปยัง hidden layer เลื่อยๆ จนกระทั้งถึง output layer ก็จะหยุด (สังเกตได้ว่าจะไม่มี loop เกิดขึ้นเลย)

2) โครงข่ายแบบวนซ้ำ (Recurrent Neural Networks : RNN)

Recurrent neural networks คือ neural networks หลายเลเยอร์ที่สามารถเก็บ (store) ข้อมูล(information) ไว้ที่ node จึงทำให้มันสามารถรับข้อมูลเป็นแบบลำดับ (data sequences) และให้ผลลัพธ์ออกเป็นลำดับของข้อมูลได้ อธิบายอย่างง่ายๆ RNN ก็คือ neural network เชื่อมต่อกันหลายๆอันและยังสามารถต่อกันเป็น Loop ได้นั่นเอง เพราะฉะนั้น RNN จึงเหมาะสมในการประมวลผลข้อมูลที่เป็นลำดับอย่างมาก

2.4.4 การใช้งานโดยทั่วไปของ RNN

1) ช่วยเทรดเดอร์ในการวิเคราะห์หลักทรัพย์ต่าง ๆ (Help securities traders to generate analytic reports)

2) ตรวจสอบความผิดปกติในสัญญาการเงินต่าง ๆ (Detect abnormalities in the contract of financial statement)

3) ตรวจสอบการปลอมแปลงบัตรเครดิตร (Detect fraudulent credit-card transaction)

4) เขียนคำบรรยายให้รูปภาพ (Provide a caption for images)

5) สร้าง chatbot ที่มีประสิทธิภาพ (Power chatbots)

6) การใช้งานในงานประเภทที่ต้องอาศัยเวลาทำกับข้อมูลอย่างตัวเนื่อง(time-series data or sequences) เช่น การอัดเสียง (audio record)

2.4.5 ความสำคัญของ Deep learning

Deep learning ถือเป็นเครื่องมือที่ทรงพลังในการทำนาย (prediction) ผลลัพธ์ต่าง ๆ อีกทั้งยังสามารถหา pattern หรือ สังเคราะห์ข้อมูล ได้เหนือกว่าองค์ความรู้เดิมที่มีอยู่ (Unsupervised Learning)

Big data ก็เปรียบเสมือนเชื้อเพลิงของ deep learning การผสมผสานระหว่างทั้งสองอย่างเรียกได้ว่าสามารถทำให้มนุษยชาติก้าวไปขึ้นอีกขั้น ไม่ว่าจะเป็นด้านของ ผลผลิต การขาย การบริหาร และนวัตกรรม

อีกทั้ง deep learning ยังทำงานได้มีประสิทธิภาพได้ดีกว่าวิธีการดั่งเดิม กล่าวคือ deep learning algorithm ให้ความแม่นยำมากว่า machine learning algorithm ในด้านของจำแนกภาพ (image classification) มากถึง 41% ด้านของการรู้จำใบหน้า(face recognition)มากถึง 27% และการรู้จำเสียง (voice recognition) มากถึง 25%

2.4.6 ข้อจำกัดของ deep learning (Limitation of deep learning)

1) ต้องการระบุชื่อหรือจัดประเภทข้อมูล (Data labeling)

ปัจจุบัน AI ส่วนมากอาศัยการ train โดยใช้หลักการของ supervise learning ซึ่งนั้นก็หมายความว่า มนุษย์จะต้องตั้งชื่อและจัดประเภทของข้อมูล (Data labeling) ด้วยตัวเองก่อนจะนำไป train (Data labeling professionals : ถูกคาดว่าจะเป็นอาชีพที่ popular ในอนาคต เมื่อเข้าสู่ยุคของ AI) ซึ่งข้อมูลเหล่านี้ก็มีจำนวนมาก จึงไม่แปลกที่อาจเกิดความผิดขึ้นได้ ตัวอย่างเช่น รถยนต์ขับเคลื่อนอัตโนมัติ (self - driving car) จ้างคนหลายร้อยคนเพื่อที่จะ labeling วิดีโอที่จะใช้ในการ train ระบบ

2) ต้องการชุดข้อมูลจำนวนมาก (Obtain huge training datasets)

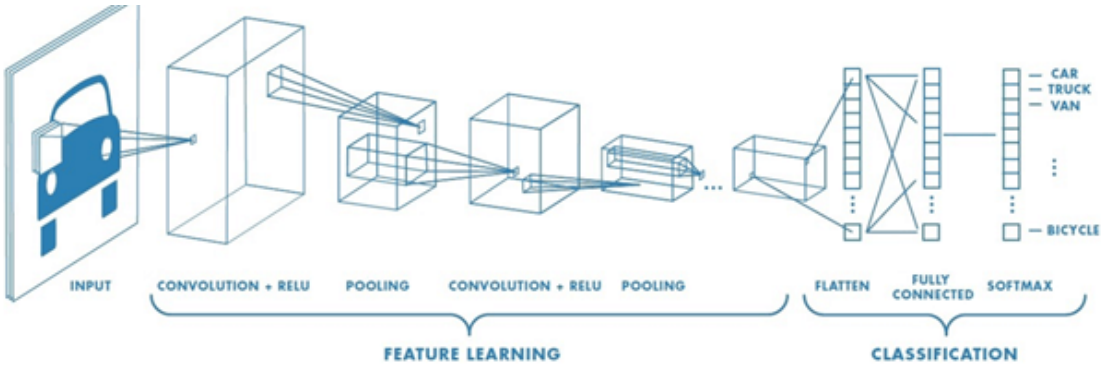
deep learning นั้นต้องข้อมูลจำนวนมากในการ train ข้อมูล ยกตัวเช่น ต้องการข้อมูลจำนวน 1000 ตัวอย่าง เพื่อจะทำให้ได้โมเดล classification ที่มีประสทธิภาพ และในบางกรณีก็ต้องการมากกว่า 1 ล้านตัวอย่างเพื่อที่จะทำให้ model ทำงานได้อย่างมีประสิทธิภาพใกล้เคียงกับมนุษย์ , แน่นอนว่า deep learning นั้นเป็นที่นิยมในบริษัทเทคโนโลยีชั้นนำทั่วโลก ซึ่งพวกเขาใช้ big data ที่สะสมข้อมูลที่มีเนื้อที่ไม่น้อยกว่า petabytes (1 ล้าน gigabytes) นั้นเป็นเหตุผลว่าทำไมบริษัทเทคโนโลยีชั้นนำจึงสามารถสร้าง deep learning model ที่มีประสิทธิภาพ และมีความแม่นยำสูง

3) ปัญหาในการอธิบาย (Explain a problem)

ไม่ใช่เรื่องง่ายที่มนุษย์เราจะอธิบาย AI model ที่ซับซ้อนได้ ยกตัวอย่างเช่น คำถามว่าที่ทำไม model ถึงเกิดการตัดสินใจแบบนั้น ? (why a particular decision was obtained?) นี่ก็เป็นอีกเหตุผลหนึ่งที่ทำไม AI ถึงได้รับการยอมรับช้านัก ก็เพราะบางอย่างมันยากที่มนุษย์เราจะสามารถอธิบายนั่นเอง

2.5 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Networks :CNN)

CNN คือ Neural Network หลายเลเยอร์ที่มีโครงสร้างเฉพาะตัวดังแสดงในรูปที่ 2.4 โดยถูกออกแบบมาเพื่อการเพิ่มความสามารถในการสกัดเอา feature ที่มีความซับซ้อนมากยิ่งขึ้นจากข้อมูล โดย CNN นั้นตอบโจทย์ปัญหาประเภทการรับรู้ (perceptual tasks) อย่างมาก



รูปที่ 2.4 โครงข่ายประสาทแบบคอนโวลูชั้น CNN

ที่มา : https://www.thaiprogrammer.org/2018/12/deep-learning-คืออะไร/

CNN นั้น มักจะถูกใช้เพื่อการสกัด feature จากข้อมูลประเภทที่ไม่ค่อยเป็นระเบียบหรือไม่ได้มีโครงสร้างเป็นรูปแบบเฉพาะตัว (unstructured data) อย่างเช่น รูปภาพ (image) เป็นต้น

ตัวอย่างการใช้งาน CNN : การคิดคำบรรยายให้รูปภาพ (predict an image caption)

สมมติ CNN รับรูปภาพ input เป็นรูปแมว , รูปภาพเหล่านี้จะถูกจัดเก็บในรูปแบบของ Pixel ซึ่งโดยทั่วไปแล้ว จะใช้ 1 layer ในการเก็บข้อมูลในรูปแบบเชดสีขาวดำ (greyscale) และจะใช้อีก 3 layer ในการเก็บข้อมูลในรูปแบบเชดสีต่าง ๆ

ระหว่างที่ model ทำการเรียนรู้ (learn หรือการสกัด feature) ที่ hidden layer, model จะทำการหา feature ที่มีความเฉพาะเจาะจงต่อข้อมูล input ที่ได้รับ อย่างเช่น ในกรณีนี้ อาจได้เป็น feature ที่ตรวจสอบหางของแมว เป็นต้น เมื่อ model ทำการเรียนรู้ (learn) เสร็จแล้ว มันจะสามารถให้คำตอบแก่แต่ละรูปภาพเป็นความน่าจะเป็น และหากรูปภาพนั้น ๆ มีความน่าจะเป็นแบบใดสูงที่สุด model จะตอบเป็นสิ่งนั้น เช่น หากโมเดลวิเคราะห์แล้วว่า รูปภาพนั้นมีความน่าจะเป็นว่าจะเป็นแมวสูง model ก็จะให้คำตอบว่ารูปภาพนั้นคือภาพแมวนั่นเอง

2.6 วรรณกรรมปริทรรศน์

ในปีพ.ศ. 2544 Frederic BONNAY, Sebastien BRIAUD, Dr. Florent COFFIN, Dr. Zied ZALILA ได้จัดทำโครงงานวิจัยเรื่อง FUZZY LANE-TRACK CONTROL FOR THE AUTOMATIC GUIDANCE OF AN AUTOMOTIVE VEHICLE โดยจะเป็นการทดลองการควบคุมเลนติดตามแบบเลื่อนที่เรียกว่า RPV ที่จะให้ความสำคัญกับการขับขี่ที่สะดวกสบาย จุดมุ่งหมายของงานวิจัยนี้คือการศึกษาการออกแบบระบบช่วยเหลือผู้ขับขี่ซึ่งจะช่วยให้มั่นใจว่ามีระบบนำทาง "เสมือนรถไฟ" ในขณะที่ผู้ขับขี่สามารถควบคุมระบบได้ตลอดเวลา ตัวควบคุมเลื่อนถูกออกแบบและติดตั้งลงบนรถยนต์ Renault Safrane เพื่อทำการควบคุมรถ, เพิ่มกล้อง, ซอฟต์แวร์ประมวลผลภาพบนตัวกล้องและไจโรสโคปเพื่อส่งข้อมูลให้แก่ผู้ขับขี่ จากข้อมูลเหล่านี้ตัวควบคุมการเลื่อนจะแสดงค่าความแปรผันของแรงบิดซึ่งใช้กับพวงมาลัยโดยใช้มอเตอร์ไฟฟ้าคู่ ผลการทดลองภาคสนามให้ผลลัพธ์เป็นที่น่าพอใจ แต่ในอีกแง่หนึ่งตัวควบคุมการติดตามเลนใช้เส้นทางที่ไม่สั่นคลอน ราบรื่นและสะดวกสบายในสถานการณ์ที่ผู้ขับขี่ไม่ควบคุมรถทดสอบ ในทางกลับกัน RPV สร้างความรู้สึกมั่นใจและความช่วยเหลือที่เห็นได้ชัดเมื่อผู้ขับขี่เก็บมือบนพวงมาลัย [22]

ในปี พ.ศ.2545 นายเกษตร์ ศิริสันติสัมฤทธิ์ และนายกมลฐิติ กุลสงค์ ได้จัดทำโครงงานวิจัยเรื่องการตรวจหาเลนด้วยแม่แบบปรับตัวเองได้ (Lane detection using adaptive template) เป็นการตรวจหาขอบเขตของเลนโดยไม่รู้รูปร่างของถนนและอยู่บนพื้นฐานของขอบภาพและคณิตศาสตร์มอร์โฟยี โดยที่ภาพแม่แบบจะถูกวางบนข้อมูลภาพระยะทางที่แปลงมาจากข้อมูลขอบภาพ ภาพแม่แบบจะถูกดำเนินการยืดและหดด้วยตัวดำเนินการไดเลคชั่นและวิธีการวนซ้ำ จนกระทั่งภาพแม่แบบใส่ลงไปได้พอดีกับเลนของถนน และภาพแม่แบบสุดท้ายจะเป็นตัวบ่งชี้ขอบเขตของเลน ซึ่งแม่แบบที่ใช้เป็นแม่แบบรูปสามเหลี่ยมซึ่งมีลักษณะคล้ายรูปร่างของเลน [23]

ต่อมาในปี พ.ศ.2551 ผศ.ดร.อาทิตย์ ศรีแก้ว สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุนารี และ นายเฉลิมพล หลงจาด ได้ทำโครงงานวิจัยเรื่องการตรวจจับถนนด้วยภาพสำหรับระบบนำวิถีอัตโนมัติในพาหนะแบบชาญฉลาด (Vision-based Lane Detection for Automatic Navigation in Smart Vehicle) งานวิจัยนี้เป็นวิธีการตรวจจับช่องทางเดินรถและการนำวิถีแบบอัตโนมัติโดยใช้แบบจำลองการเทียบเคียงรูปแบบ ระบบดังกล่าวจะเป็นการผสมกันระหว่าง การหาแบบจำลองช่องทางการเดินรถ การใช้แบบจำลองการเทียบเคียงรูปแบบ เทคนิคการหาจุดเริ่มต้นของการเกิดแยก การจำแนกประเภทช่องทางเดินรถและการหาค่าควรจะเป็นของช่องทางเดินรถ ซึ่งระบบนี้เป็นระบบการตรวจสอบช่องทางการเดินรถที่ไม่จำเป็นต้องมีกำหนดการกำหนดสภาวะเริ่มต้นและสามารถทำงานในสภาวะแสงสว่างที่แตกต่างกันได้ ระบบนี้ยังสามารถตรวจจับช่องทางเดินประเภทต่าง ๆ ได้ เช่น ทางตรง สามแยก และสี่แยก เป็นต้น ระบบนี้แบ่งออกเป็น 2 ระบบคือ 1.ระบบการตรวจจับช่องทางเดินรถในสภาวะยานพาหนะเคลื่อนที่ไป

ข้างหน้าขณะไม่เกิดการเลี้ยว 2.ระบบการตรวจจับช่องทางการเดินรถในสภาวะยานพาหนะเคลื่อนที่ไปข้างหน้าขณะเกิดการเลี้ยว งานนี้ได้ใช้แบบจำลองพาลาโบล่า มาเป็นตัวอธิบายลักษณะทางกายภาพของช่องทางการเดินรถ โดยมีฟังก์ชันค่าควรจะเป็นมากที่สุดเป็นตัวกำหนดความเหมาะสมสำหรับแต่ละภาพอย่างอัตโนมัติและใช้เทคนิคการจุดเริ่มต้นของการเกิดแยกในการจำแนกประเภทช่องทางเดินรถ เทคนิคนี้มีความคงทนต่อสภาพแวดล้อมต่าง ๆ เช่น สัญญาณรบกวน เงา เส้นถนนที่ไม่ต่อเนื่อง เป็นต้น ระบบนี้ให้ผลการตรวจจับช่องทางการเดินรถได้อย่างถูกต้องแม่นยำด้วยอัตราสูงสุด 96 เปอร์เซ็นต์ [20]

ในปี พ.ศ.2557 นาย กิตติศักดิ์ ทองยวน และ นาย มหศักดิ์ เกตุฉ่ำ ภาควิชาการจัดการเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ได้ทำโครงงานวิจัยอุปกรณ์ตรวจสอบช่องทางจราจรแบบเรียลไทม์สำหรับการขับรถทำงานบนระบบสมองกลฝังตัว (Real-Time Lane Detection for Driving using Embedded System) โดยโครงงานนี้ใช้วิธีตรวจสอบช่องทางจราจรในขณะที่ขับรถเพื่อช่วยแจ้งเตือนผู้ขับขี่ในขณะที่ผู้ขับขี่ประมาทหรือไม่ชำนาญ โดยใช้หลักการในการประมวลผลภาพ (Image Processing) มาประยุกต์ใช้งานกับระบบสมองกลฝังตัว โดยใช้กล้องในการบันทึกช่องทางจราจรบริเวณด้านหน้าของรถแบบทันทีทันใดและนำภาพที่บันทึกมาทำการวิเคราะห์โดยใช้หลักการในการประมวลผลภาพโดยใช้เทคนิค Edge Detection และ Hough Transform ซึ่งเป็นหลักการของการประมวลผลภาพและแสดงการแจ้งเตือนให้ผู้ขับขี่ทราบทางจอแสดงผล โดยจะใช้ระบบสมองกลฝังตัว STM32 สถาปัตยกรรม 32-bits ARM Cortex-M3 และ RapidRTM32 Blockset ซึ่งเป็นชุดทดสอบกล่องคำสั่งแบบรูปภาพที่สามารถใช้งานร่วมกับMatlab/Simulink เป็นตัวประมวลผล ซึ่งงานวิจัยนี้สามารถช่วยลดอุบัติเหตุที่จะเกิดขึ้นบนท้องถนนได้ [3]

ในปีเดียวกัน Yue Wanga, Eam Khwang Teoha, Dinggang Shenb ได้จัดทำโครงงานวิจัยเรื่อง Lane detection and tracking using B-Snake โดยการทดลองจะใช้ การตรวจจับและติดตามอัลกอริทึมการตรวจจับเลนแบบ B-Snake โดยไม่มีพารามิเตอร์ของกล้อง ตามรูปแบบช่องทางที่สามารถที่จะอธิบายช่วงกว้างของโครงสร้างถนนตั้งแต่ B-Spline สามารถแบบใด ๆ โดยชุดของจุดปัญหาของการตรวจจับทั้งสองด้านของเครื่องหมาย (หรือขอบเขต) ของเลนถูกรวมเข้าด้วยกันที่นี่เป็นปัญหาในการตรวจหาเส้นกึ่งกลางของเลนโดยใช้ความรู้เกี่ยวกับเส้นคู่ขนาน นอกจากนี้ยังมีอัลกอริทึมที่มีประสิทธิภาพซึ่งเรียกว่า CHEVP เพื่อเสนอตำแหน่งเริ่มต้นที่ดีสำหรับ B-Snake นอกจากนี้ยังมีข้อผิดพลาดน้อยที่สุดโดยวิธีค่าความคลาดเคลื่อนกำลังสองเฉลี่ยต่ำสุด (MMSE) ถูกเสนอเพื่อกำหนดจุดควบคุมของโมเดล B-Snake โดยผลกระทบของภาพโดยรวมที่ด้านข้างของเลน ผลการทดลองแสดงให้เห็นว่าวิธีการที่เสนอมีประสิทธิภาพต่อการเปลี่ยนแปลงของสัญญาณรบกวน เงา และความ

ส่องสว่างในภาพถนนที่จับได้นอกจากนี้ยังสามารถใช้ได้กับเครื่องหมาย ถนนที่เป็นเส้นประ และถนนสายสีทึบ [21]

ในปี พ.ศ.2560 นาย กฤษณ์ สอนสำแดง, นายธนบัตร ว่องวัฒนกูล และนายเรืองวิทย์ จันทะโก นิสิตคณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมเครื่องกล มหาวิทยาลัยนเรศวร ได้จัดทำโครงงานเรื่องอัลกอริทึมตรวจจับช่องทางเดินรถสำหรับยานยนต์ อัตโนมัติมหาวิทยาลัยนเรศวร มีวัตถุประสงค์เพื่อศึกษาและประยุกต์ใช้เรื่องของการประมวลผลภาพ พัฒนา อัลกอริทึมในการวิเคราะห์เส้นแบ่งแยกช่องทางเดินรถ โดยทำการทดลองกับถนนภายในมหาวิทยาลัย แต่เนื่องจากเส้นแบ่ง ช่องทางเดินรถภายในมหาวิทยาลัยนั้นมีสีที่ขาดหายและไม่ชัดเจน ทำให้ไม่สามารถนำมาใช้เป็นข้อมูล ในการประมวลผลภาพได้ ดังนั้นผู้ดำเนินโครงงานจึงได้จัดทำแบบจำลองช่องทางเดินรถขึ้นมา 2 แบบ เพื่อใช้ในการดำเนินงาน โดยใช้หลักการและทฤษฎีของระบบช่วยขับ (Assistant Driving) การตรวจจับช่องทางการเดินรถ (Lane Detection) ระบบควบคุมรถให้อยู่ในช่องทาง (Lane Keeping System) ทำการทดลองที่ค่าความเข้มแสง 15 lux และ 241 lux เพื่อหาความแตกต่าง

ผลการทดลองพบว่าที่ค่าความเข้มแสงต่างกันและแบบถนนจำลองที่ต่างกันนั้นไม่มีผลต่อการตรวจจับช่องทางเดินรถของแบบจำลอง โปรแกรมประมวลผลภาพมีการทำงานค่อนข้างเสถียร และอาจมีการคลาดเคลื่อนของจุดกึ่งกลางเล็กน้อย เนื่องจากขนาดของเส้นช่องทางเดินรถที่ไม่เท่ากันและแสงที่ตกกระทบบนถนน แต่การคำนวณของการประมวลผลภาพจะอยู่ในระยะที่ปลอดภัยสามารถใช้งานได้ปกติ [19]

ในปี พ.ศ.2561 Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, Xiaoou Tang ได้ทำการวิจัยเรื่อง Spatial CNN for Traffic Scene Understanding เป็นงานวิจัยเชิงลึกเกี่ยวกับโครงข่ายประสาทแบบคอนโวลูชัน (CNNs) ในงานวิจัยนี้จะใช้เป็น Spatial CNN (SCNN) ซึ่ง SCNN เหมาะสำหรับโครงสร้างรูปร่างที่ต่อเนื่องหรือวัตถุขนาดใหญ่ที่มีความสัมพันธ์เชิงพื้นที่ แต่มีลักษณะที่ปรากฏน้อย เช่นช่องทางจราจรเสา และกำแพง งานวิจัยจะใช้ SCNN ในการตรวจจับช่องทางจราจร ผลวิจัยพบว่า SCNN มีประสิทธิภาพสูงกว่าเครือข่าย Recurrent Neural Network (RNN) ตาท ReNet และ MRF + CNN (MRFNet) ในชุดตรวจจับช่องทาง 8.7% และ 4.6% ตามลำดับ และ SCNN ยังเป็นที่หนึ่งในการตรวจสอบเลนเกณฑ์มาตรฐานของ TuSimple ด้วยความแม่นยำ 96.53% [7]

ในการศึกษาโครงงานเกี่ยวกับการตรวจจับช่องทางเดินรถนั้น มีการใช้หลักการหลากหลายรูปแบบแตกต่างกันไป ในโครงงานเล่มนี้ได้หยิบการใช้งาน CNN การคำนวณหาจุดกึ่งกลางช่องทางเดินรถที่เหมาะสม การใช้งานฟังก์ชันต่าง ๆ ของโปรแกรมmatlab มาประยุกต์ใช้งานกับโครงงานนี้ เพื่อให้โครงงานนี้ประสบความสำเร็จไปได้ด้วยดี

บทที่ 3
วิธีการดำเนินงาน

3.1 การเก็บข้อมูลและการเตรียมข้อมูลภาพ

3.1.1 ทำการศึกษาและข้อมูลเก็บข้อมูลถนนภายใน มหาวิทยาลัยนเรศวร โดยการถ่ายภาพถนน ในช่วงเวลากลางวัน ที่จะสามารถมองเห็นเส้นถนนและขอบฟุตบาทได้ชัดเจนมากที่สุด

ตารางที่ 2 แสดงระยะเบรกที่ปลอดภัย

อัตราความเร็ว (กม/ชม.)	รถทั่วไป ระยะคิด + ระยะเบรก = ระยะหยุดรถ	รถบรรทุกขนาดใหญ่ ระยะคิด + ระยะเบรก = ระยะหยุดรถ
20	4 + 3 = 7	14 + 9 = 23
30	6 + 6 = 12	21 + 17 = 38
40	8 + 10 = 18	28 + 27 = 55
50	10 + 15 = 25	35 + 38 = 73
60	12 + 22 = 34	42 + 55 = 97
70	14 + 29 = 43	49 + 74 = 123
80	16 + 38 = 54	56 + 102 = 158
90	18 + 48 = 66	63 + 122 = 185
100	20 + 60 = 80	69 + 145 = 214

จากการศึกษามูลพบว่าระเบรกที่ปลอดภัยของรถทั่วไปมีระยะที่น้อยกว่ารถบรรทุกขนาดใหญ่ที่ขับด้วยความเร็วที่เท่ากันอยู่ประมาณ 3 เท่า เนื่องด้วยรถบรรทุกนั้นมีขนาดที่ใหญ่จึงทำให้ต้องใช้ระยะเบรกมากพอสมควรถึงจะสามารถหยุดรถได้

รูปที่ 3.1 ภาพอ้างอิงระยะเบรกที่ปลอดภัย

จากรูปที่ 3.1 เป็นภาพขนาด 1920x1080 จากกล้องที่ติดตั้งหลังกระจกมองหลัง แสดงระยะเบรกที่ได้ทำการวัดจากหน้ารถ ในรูปเส้นล่างสุดห่างจากหน้ารถ 3 เมตรตรงกับพิกัดแกน Yที่ 875 เส้นที่ 2 ห่างจากหน้ารถ 7 เมตรตรงกับพิกัดแกน Y ที่ 730 เส้นที่ 3 ห่างจากหน้ารถ 12 เมตรตรงกับพอกัดแกน Y ที่ 650 และเส้นบนสุดห่างจากหน้ารถ 18 เมตรตรงกับพิกัดแกน Y ที่ 630 ในการทดลองใช้อัตราความเร็วที่ 40 กิโลเมตรต่อชั่วโมง ดังนั้นจะใช้ระยะเบรกที่ 18 เมตร

3.1.2 นำภาพที่ได้มาคัดเลือกเฉพาะช่วงถนนที่ชัดเจน ให้พร้อมสำหรับการใช้งาน

3.1.3 เนื่องจากในการสร้างอัลกอริทึมเราใช้ตัว Pre-Train Alexnet (อัลกอริทึมในการสร้าง Train Network) ซึ่งต้องการข้อมูลภาพที่ขนาด 227 x 227 pixel เพื่อที่จะพิจารณา และกำหนดองค์ประกอบต่าง ๆ จึงต้องทำการแปลงข้อมูลภาพที่ได้ ให้เป็นได้เป็นตามที่ Pre-Train ต้องการเพื่อนำไปใช้ร่วมกับอัลกอริทึม

Alexnet เป็น Deep Learning Model ตัวนึง ที่สร้างขึ้นโดย Geoffrey Hinton เป็นตัวจำแนกประเภทสิ่งของต่าง ๆ ประกอบไปด้วย 25-layer และเราได้ทำการแปลง layer ที่ 20-25 เพื่อให้สอดคล้องกันการใช้งาน regression network เป็น Network ที่ใช้เวลาในการใช้งาน GPU ต่ำที่สุดในการ Pre-Train ทั้งหมดที่ MATLAB รองรับนอกจากนี้ยังมีพารามิเตอร์ ที่รองรับมากถึง 61 ล้าน พารามิเตอร์ แต่ในกรณีเดียวกันก็เป็นตัวที่มีความแม่นยำต่ำที่สุดเช่นกัน (แม่นยำที่ 52-57%)

3.1.4 นำภาพที่ได้จากการแปลงข้อมูลภาพเป็น 227 x 227 pixel มาแบ่งเป็นชุดเพื่อแยก ในการใช้งาน ดังนี้

1) สำหรับ Train จำนวน 880 รูป

2) สำหรับ Validate จำนวน 100 รูป

3) สำหรับ Test จำนวน 100 รูป

3.2 การเตรียมข้อมูลสำหรับการ Train-Network การตรวจจับช่องทางเดินรถ และหาจุดกึ่งกลาง

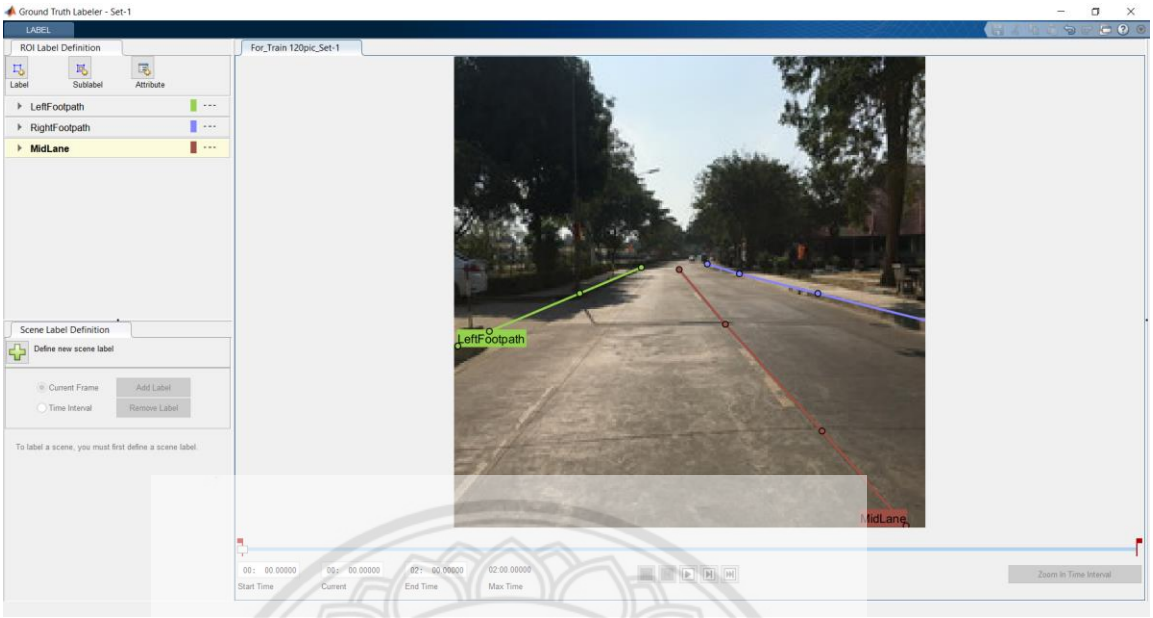
3.2.1 เริ่มศึกษาข้อมูลและเขียนอัลกอริทึมด้วย Application Ground Truth ของโปรแกรม MATLAB Version R2019a โดยการเขียนอัลกอริทึมจะแบ่ง Code เป็น 3 ชุด โดยจะกล่าวถึงในหัวข้อถัดไป

3.2.2 นำชุดข้อมูลภาพสำหรับ Train มาทำการสร้างเส้นลงบนภาพโดยใช้ Application ของ MATLAB ที่ชื่อว่า Ground Truth ในการกำหนดส่วนที่เราสนใจ ซึ่งในส่วนที่เราสนใจจะมี 3 ส่วนด้วยกัน นั่นคือ ฟุตบาทด้านซ้าย ฟุตบาทด้านขวา และเส้นกึ่งกลางถนน ซึ่งจะนำไปใช้ในการคำนวณหาจุดกึ่งกลางเลนต่อไป โดยการวาดเส้นลงบนภาพจะมีการกำหนดดังนี้ ในแต่ละภาพจะใช้เพียงแค่ 3 เส้น และแต่ละเส้นจะต้องใช้เพียงแค่ 4 จุดเท่านั้น เพื่อที่เราจะใช้ Linear Regression ในการหาข้อมูล Coefficient ของแต่ละเส้นในภาพ และในการตั้งชื่อ กำหนดดังนี้

1) LeftFootpath คือ ฟุตบาทด้านซ้าย

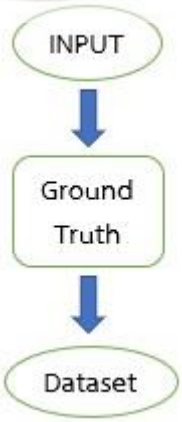
2) RightFootpath คือ ฟุตบาทด้านขวา

3) MidLane คือ เส้นกึ่งกลางถนน



รูปที่ 3.2 การวาดเส้นในแอพพลิเคชั่น Ground Truth

3.2.3 หลังจากสร้างเส้นใน Application Ground Truth ดังรูปที่ 3.1 เสร็จเรียบร้อย จากนั้นทำการ ส่งออกค่าที่ได้ (Export) ออกมาผลลัพธ์ที่ได้นี้คือ Dataset ซึ่งจะประกอบด้วยลำดับรูปภาพที่ที่เราใช้ใน Ground Truth พร้อมกับค่า Coefficient ทั้งหมด เนื่องจากแต่ละเส้นที่เราสนใจใช้จุด 4 จุดในการสร้าง จะทำให้เกิดเส้น 3 เส้นที่ต่อกัน ทำให้ได้ค่า Coefficient ของแต่ละเส้นในภาพมี 3 ค่า ดังนั้นค่าของ Coefficientของแต่ละภาพจึงมีด้วยกัน 9 ค่า



รูปที่ 3.3 แสดงการทำงานในการสร้างข้อมูล Dataset

3.3 อัลกอริทึมที่ใช้งาน

สำสำหรับการตรวจจับขอบฟุตบาทและเส้นกึ่งกลางถนน การเขียน Code โดยจะแบ่งการเขียน Code เป็น 3 ส่วน ดังนี้

3.3.1 Code PART 1

Code PART 1 คือ Code ที่ใช้สำหรับการสร้างตัว Network โดยใช้ Pre-train Alexnet ซึ่งใช้ในการสอนอัลกอริทึมหรือที่เรียกว่า Train-Network โดยจะ Input เป็นข้อมูล Dataset ซึ่งจะประกอบด้วยข้อมูลภาพที่ขนาด 227x227 pixel และข้อมูล Coefficient และ Output ที่ได้ก็คือ Train-Network ที่พร้อมใช้งาน

```
saveVarsMat = load('Export For_train 2.mat');
gTruth = saveVarsMat.gTruth;
clear saveVarsMat;
[nPic,c] = size(gTruth.LabelData);
for i=1:nPic
    xr = gTruth.LabelData.RightFootpath{i,1}{1, 1}(:,1);
    yr = gTruth.LabelData.RightFootpath{i,1}{1, 1}(:,2);
    rightCo(i,:) = polyfit(xr,yr,2);
    xl = gTruth.LabelData.LeftFootpath{i,1}{1, 1}(:,1);
    yl = gTruth.LabelData.LeftFootpath{i,1}{1, 1}(:,2);
    leftCo(i,:) = polyfit(xl,yl,2);
    xm = gTruth.LabelData.MidLane{i,1}{1, 1}(:,1);
    ym = gTruth.LabelData.MidLane{i,1}{1, 1}(:,2);
    midCo(i,:) = polyfit(xm,ym,2);
end
picLocation = gTruth.DataSource.Source;
label = [leftCo, midCo ,rightCo];
DataSet = table(picLocation, label);
clear ('c','nPic','xl','xm','xr','yl','ym','yr','i');
```

รูปที่ 3.4 ส่วนที่ 1 ของ Code PATH 1

จากรูปที่ 3.3 การทำงานของ Code คือ การรับข้อมูลจาก Ground Truth มาสร้างเป็น Dataset ที่จะใช้ในการสร้าง Trained-Network

```
20 -    rng(0);
21      % Randomly split data into a training and Validate set.
22 -    shuffledIdx = randperm(height(DataSet));
23 -    idx = floor(0.9 * height(DataSet));
24 -    imdsTrain = DataSet(shuffledIdx(1:idx),:);
25 -    imdsValidation = DataSet(shuffledIdx(idx+1:end),:);
26      % Original network
27 -    originalConvNet = alexnet;
28      % Extract layers from the original network
29 -    layers = originalConvNet.Layers;
30      % Replace the last few fully connected layers with suitable size layers
31 -    layers(20:25) = [];
32 -    outputLayers = [ ...
33      fullyConnectedLayer(24, 'Name', 'fcLane1');
34      reluLayer('Name','fcLane1Relu');
35      fullyConnectedLayer(9, 'Name', 'fcLane2');
36      regressionLayer('Name','output')];
37 -    layers = [layers; outputLayers];
38      %Train Option
39 -    options = trainingOptions('sgdm', ...
40          'MiniBatchSize',10, ...
41          'MaxEpochs',20, ...
42          'InitialLearnRate',1e-4, ...
43          'Shuffle','every-epoch', ...
44          'ValidationData',imdsValidation, ...
45          'ValidationFrequency',100, ...
46          'Verbose',false, ...
47          'Plots','training-progress');
48      %Train Network
49 -    netTransfer = trainNetwork(imdsTrain,layers,options);
```

รูปที่ 3.5 ส่วนที่ 2 ของ Code PATH 1

จากรูปที่ 3.4 บรรทัดที่ 20-49 จะเป็น Code ที่ใช้เป็นการสร้าง Trained-Network จะใช้ข้อมูล Dataset เลเยอร์ Layer และ Options (บรรทัดสุดท้าย) เพื่อที่จะสร้าง Trained-Network ออกมา

3.3.2 Code PART 2

Code PART 2 คือ Code ที่ใช้สำหรับการสร้าง Output (ค่า Coefficient) และใช้สำหรับทดสอบค่า Coefficient ที่ได้จาก Train-Network ที่ได้ออกมานั้นถูกต้องและแม่นยำ จากการใช้ Train-Network ที่ได้จาก Code PATH 1 โดยรับ Input เป็นข้อมูลภาพจากกล้องโดยตรง (Real Time)

```
% Load Network and Generate Coefficient from Network
saveVarsMat = load('nettransfer2.mat');
TrainNetwork = saveVarsMat.netTransfer;
clear saveVarsMat;
nnet = TrainNetwork;
% Load Image from camera
camera = webcam ;
pic = camera.snapshot ;
label = predict(nnet,pic);
```

รูปที่ 3.6 เนื้อหาบางส่วนของ Code PATH 2

จากรูปที่ 3.5 บรรทัดที่ 1-3 จะเป็นการเรียกข้อมูล Trained-Network และนำไปใช้กับ Input เป็นข้อมูลภาพจากกล้องโดยตรง ในบรรทัดที่ 6-8 และ บรรทัดที่ 9 ตัว Coed จะทำการสังเคราะห์ผล (Generate) ค่า Coefficient ที่เป็น Output ออกมา

3.3.3 Code PART 3

Code PART 3 คือ Code สำหรับการใช้การประเมินผล (Evaluation) ค่าของ Output ที่ได้จาก Code PATH 2 โดย นำค่าที่ได้ไปทำการคำนวณหาจุดกึ่งกลาง ซึ่งค่า Output ที่จะได้เป็นข้อมูลภาพที่แสดงจุดกึ่งกลางที่เราต้องการ และเพื่อทดสอบค่า Output ที่ได้ Code นี้จะนำข้อมูลภาพและข้อมูล Coefficient ที่ได้ไปสร้างเส้นลงบนภาพโดยใช้สมการ Polynomial อันดับสอง และจะแสดงเส้นฟุตบาทด้านซ้าย เส้นกึ่งกลาง และเส้นฟุตบาทด้านขวาลงในข้อมูลภาพนั้น ๆ

3.4 การตรวจเช็คข้อมูล Dataset

หลังจากที่ได้ Dataset มาเป็นที่เรียบร้อย ซึ่งใน Dataset จะประกอบด้วย ภาพที่เรานำเข้า และค่า Coefficient ทั้งหมด 9 ค่า และเพื่อทดสอบว่าค่าที่ได้มีความแม่นยำจริง ๆ เราจึงต้องทำการทดสอบค่า Coefficient ที่ได้ โดยการทดลองสร้างเส้น จากค่าที่ได้ลงบนภาพ โดยใช้ Code PART 3 เพื่อทำการสร้างเส้นแสดงจุดที่เราสนใจบนข้อมูลภาพขนาด 227 x 227 pixel ซึ่งข้อมูลภาพที่ใช้ในการตรวจเช็คก็จะใช้ข้อมูลภาพชุดเดียวกับที่ใช้ในการกำหนดองค์ประกอบใน Application Ground Truth

ซึ่งค่า Coefficient ที่ได้ออกมาจากเส้นที่เราวาดทั้ง 3 เส้น นั้นจะได้เป็นจำนวนจริง 9 ค่า เพื่อที่จะนำไปใช้ร่วมกับสมการสมการ Polynomial อันดับที่สอง เพื่อให้ได้พิกัดจุดที่จะนำไปสร้างเส้นแสดงจุดที่เราสนใจนั่นเอง

โดยสมการ Polynomial อันดับสอง ที่ใช้จะอยู่ในรูป ax^2+bx+c ซึ่ง a b และ c คือค่า Coefficient ที่เราจะนำมาใส่เพื่อนำไปหาพิกัด (X,Y) ของแต่ละช่วงที่เราสนใจก่อนที่จะนำจุดที่ได้ไปสร้างเป็นเส้นของฟุตบาทหรือ เส้นแบ่งช่องทางเดินรถที่เราสนใจ โดยการสร้างนี้จะใช้ Code PART 2 ในการสร้างเส้นโดยใช้ค่า Coefficient และแสดงเป็นบนข้อมูลภาพ Output

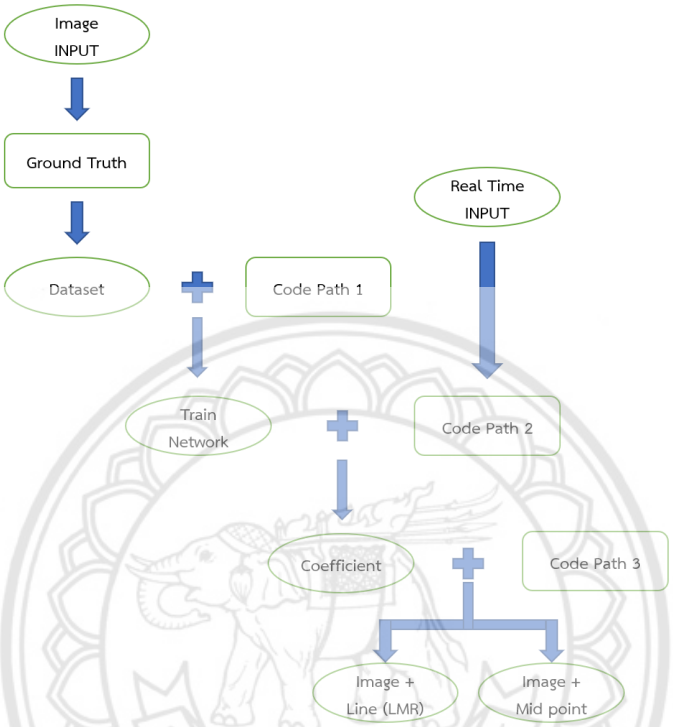
แต่เนื่องจากในกรณีนี้ Train-Network ยังไม่ประสบความสำเร็จ ซึ่งถ้า Train-Network ใช้ได้จริง ค่า Coefficient ที่ได้ ก็จะเป็นค่าที่แม่นยำ เช่นเดียวค่าที่ได้ข้อมูล Dataset ดังนั้นค่า Coefficient ที่ได้จะใช้ข้อมูล Dataset แทนข้อมูลที่จะได้จาก Train-Network และเพื่อทดสอบค่า Output ที่ได้ Code นี้จะนำข้อมูลภาพและข้อมูล Coefficient ที่ได้ไปสร้างเส้นลงบนภาพโดยใช้สมการ Polynomial อันดับสอง และจะแสดงเส้นฟุตบาทด้านซ้าย เส้นกึ่งกลาง และเส้นฟุตบาทด้านขวาลงในข้อมูลภาพนั้น ๆ

รูปที่ 3.7 ภาพจาก Ground Truth (ด้านซ้าย) เปรียบเทียบกับภาพจากใช้ Code PART 3 (ด้านขวา)

จากรูปที่ 3.3 จะแสดงให้เห็นว่า เส้นที่ได้จากใช้ Code PART 3 โดยใช้ข้อมูลจาก Dataset มีความแม่นยำสูง แสดงให้เห็นว่าค่า Coefficient จาก Dataset มีความน่าเชื่อถือ

3.5 สรุปการทำงานของอัลกอริทึม

เพื่อที่จะให้เห็นลำดับการทำงานที่ชัดเจน ผังการทำงานของระบบโดยรวมทั้งหมดจะแสดงดังรูปต่อไปนี้



รูปที่ 3.8 ผังการทำงานรวมของระบบ

ดังรูปที่ 3.4 เป็นการทำงานรวมของระบบการตรวจจับขอบฟุตบาท จะให้เห็นได้ว่าเมื่อได้ค่า Dataset ออกมาแล้วนั้น นำข้อมูล Dataset ไปใช้กับ Code Path 1 เพื่อทำการ training network เมื่อทำการเทรนเสร็จเอาต์พุตที่จะได้จะเป็นตัว trained network จากนั้นนำเอาต์พุตไปใช้งานกับ Code Path 2 ที่รับอินพุตเป็นภาพเรียลไทม์จากกล่องเพื่อนให้ได้ค่าโคเอฟฟิเชียนท์ จากค่าโคเอฟฟิเชียนท์ที่ได้จะใช้ในขั้นตอนสุดท้ายคือการทดสอบความแม่นยำโดยแสดงผลออกมาเป็นเส้นคล้ายกับเส้นที่วาดใน Application Ground Truth และทำการคำนวณหาจุดกึ่งกลางเลนด้านซ้ายและแสดงผลลัพธ์สุดท้ายออกมาเป็นภาพที่คำนวณหาจุดกึ่งกลางเลนด้านซ้ายและสร้างมาร์กเรียบร้อยแล้วโดยใช้ Code Path 3

ในการทดลองนี้ เป็นการเปรียบเทียบการใช้งานค่าโคเอฟฟิเชียนท์จาก Dataset และ ค่าโคเอฟฟิเชียนท์จาก trained network เพื่อพิสูจน์ว่า ถ้าหากผลการทดลองการเปรียบเทียบนี้ออกมาไม่ต่างกัน ค่าโคเอฟฟิเชียนท์จาก Trained-Network มารถนำไปใช้งานได้จริง แต่ถ้าหากผลการทดสอบเปรียบเทียบออกมาแตกต่างกัน แสดงว่าการ training network ด้วยตัว Pre-train Alexnet นั้นไม่สามารถนำมาใช้งานกับงานวิจัยนี้ได้

บทที่ 4
ผลการทดลองและวิเคราะห์ผลการทดลอง

4.1 ผลการทดลอง

การทดลองเปรียบเทียบผล การใช้อัลกอริทึม code 2 และ code 3 ตรวจจับช่องทางการเดินรถโดยใช้ ข้อมูล Dataset และ ข้อมูล Network

4.1.1 ผลเปรียบเทียบโดยใช้ code 2

ผลการใช้อัลกอริทึมตรวจจับช่องทางการเดินรถโดยใช้ code 2 และข้อมูล Dataset ข้อมูล Network ตามลำดับ ซ้าย ขวา

รูปที่ 4.1 แสดงเส้น บริเวณที่มีเงาบางส่วน

รูปที่ 4.2 แสดงเส้น บริเวณที่มีเงาบางส่วน

รูปที่ 4.3 แสดงเส้น บริเวณที่มีแสงสว่างมาก

รูปที่ 4.4 แสดงเส้น บริเวณที่มีแสงสว่างมาก

รูปที่ 4.5 ตำแหน่งเส้น บริเวณที่มีเงาเป็นส่วนใหญ่

รูปที่ 4.6 ตำแหน่งเส้น บริเวณที่มีเงาเป็นส่วนใหญ่

รูปที่ 4.7 ตำแหน่งเส้น บริเวณที่มีแสงน้อย

รูปที่ 4.8 ตำแหน่งเส้น บริเวณที่มีแสงน้อย

รูปที่ 4.9 ตำแหน่งเส้น บริเวณทางโค้งซ้าย

รูปที่ 4.10 ตำแหน่งเส้น บริเวณทางโค้งซ้าย

รูปที่ 4.11 ตำแหน่งเส้น บริเวณที่มีแสงมากและเงาบางส่วน

รูปที่ 4.12 ตำแหน่งเส้น บริเวณที่มีแสงมากและเงาบางส่วน

รูปที่ 4.13 ตำแหน่งเส้น บริเวณทางโค้งขวา

รูปที่ 4.14 ตำแหน่งเส้น บริเวณทางโค้งขวา

จากการทดลอง เปรียบเทียบโดยใช้ code 2 สร้างเส้นที่สร้างโดยใช้ค่า coefficient จากข้อมูลของ dataset จากรูปด้านซ้ายพบว่าเส้นทั้ง 3 เส้นนั้นมีความแม่นยำสูงสามารถตรวจพบขอบฟุตบาทได้ แต่ก็ยังมีบางค่าที่ผิดผลาดไป อย่างเช่นรูปที่ 4.9 เส้นทางด้านซ้ายที่ไม่สามารถสร้างตามแนวขอบฟุตบาทได้ และเส้นกึ่งกลางเลนของรูปที่ 4.13 ที่มีค่าช่วงต้นของเส้นนั้นแตกต่างออกไป และเส้นจากรูปด้านขวานั้นสร้างโดยใช้ค่า coefficient จากข้อมูลของ train network จะพบว่าเส้นที่สร้างได้นั้นมีลักลักษณะเป็นเส้นตรงไม่โค้งตามแนวขอบฟุตบาทและเส้นที่ได้ออกมานั้นไม่ตรงกับขอบฟุตบาท อย่างเช่นรูปที่ 4.2 จะเห็นได้ว่าเส้นที่สร้างออกมานั้นอยู่สูงกว่าขอบฟุตบาทดังรูป

4.1.2 ผลเปรียบเทียบโดยใช้ code 3

ผลการใช้อัลกอริทึมตรวจจับช่องทางการเดินรถโดยใช้ code 3 และข้อมูล Dataset ข้อมูล Network ตามลำดับ ซ้าย ขวา

รูปที่ 4.15 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงสว่างมาก

รูปที่ 4.16 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงสว่างมาก

รูปที่ 4.17 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีเงาบางส่วน

รูปที่ 4.18 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีเงาบางส่วน

รูปที่ 4.19 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงมากและเงาบางส่วน

รูปที่ 4.20 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงมากและเงาบางส่วน

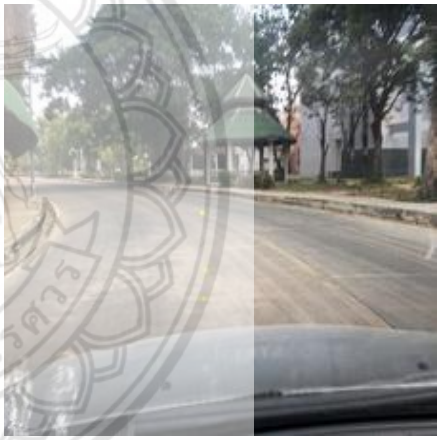
รูปที่ 4.21 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีเงาเป็นส่วนใหญ่

รูปที่ 4.22 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีเงาเป็นส่วนใหญ่

รูปที่ 4.23 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงน้อย

รูปที่ 4.24 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณที่มีแสงน้อย

รูปที่ 4.25 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณทางโค้งขซ้าย



รูปที่ 4.26 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณทางโค้งขซ้าย

รูปที่ 4.27 ตำแสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณทางโค้งขวา

รูปที่ 4.28 แสดงตำแหน่งจุดกึ่งกลางของช่องทางเดินรถ บริเวณทางโค้งขวา

จากทดลอง เปรียบเทียบโดยใช้ Code PATH 3 แนวโน้มจุดกึ่งกลางเลนด้านซ้าย ที่สร้างจากค่า Coefficient จากข้อมูลของ Dataset แสดงตำแหน่งจุดกึ่งกลางของเลนด้านซ้ายได้ถูกต้องและมีความแม่นยำดังรูปด้านซ้าย แต่ยังมีการผิดพลาดในการคำนวณหาจุดกึ่งกลางอยู่ อย่างเช่นรูปที่ 4.25 พบว่ามีการผิดพลาดในการคำนวณทำให้รูปไม่สามารถแสดงจุดกึ่งกลางของเลนด้านซ้ายออกมาได้ แต่รูปที่ 4.23 ที่จุดบนสุดจะเห็นได้ว่าจุดไม่อยู่ตรงถนน ซึ่งไม่ได้เกิดจากการผิดพลาดทางการคำนวณแต่เป็นการผิดพลาดทางเทคนิคการถ่ายภาพของถนนรูปนี้ ส่วนรูปด้านขวาจะแสดงแนวโน้มจุดกึ่งกลางเลนด้านซ้าย ที่สร้างจากค่า Coefficient จากข้อมูลของ Train-Network พบว่าจุดกึ่งกลางของเลนด้านซ้ายมีค่าคงที่ จากรูปด้านขวาจะเห็นได้ว่าจุดกึ่งกลางของเลนด้านซ้ายทั้ง 3 จุดจะอยู่ในตำแหน่งเดียวกันทั้งหมด

4.2 วิเคราะห์ผลการทดลอง

ผลเปรียบเทียบจาก Code PATH 2 จะเห็นได้ว่าเส้นที่แสดงโดยใช้ค่า Coefficient จากข้อมูลdataset ทำให้เส้นที่ได้มีความแม่นยำ แต่บริเวณทางโค้งแคบ หรือ บริเวณของรูปที่มีฝากระโปงอยู่มาก อาจทำให้เกิดค่า error ของค่า Coefficient ได้ ส่วนเส้นที่ใช้ Coefficient จากข้อมูลของ Train-Network มีค่าความแม่นยำที่ต่ำและไม่ตรงจุดตามที่ต้องการ เนื่องจากการสร้าง Train-Network ที่ไม่สมบูรณ์จึงทำให้ Train-network ไม่สามารถสังเคราะห์ค่า Coefficient จากรูปภาพออกมาได้แบบต่อเนื่อง จึงทำให้ค่า Coefficient กลายเป็นค่าคงที่ค่าเดียวกันทั้งหมดทำให้เกิดผลกับการแสดงตำแหน่งของเส้น ในกรณีเดียวกันเมื่อเปรียบเทียบโดยใช้ Code PATH 3 ให้ผลของแนวโน้มจุดกึ่งกลางถนน มีผลการทดลองที่คล้ายคลึงกัน เนื่องมาจากปัจจัยเดียวกัน ซึ่งจากการทดลองนี้สามารถบอกได้ว่า ข้อมูลในการเทรนไม่เพียงพอและอาจมีปัจจัยร่วมทำให้ ตัว Train-Network ไม่สามารถใช้ได้งานจริง ทำให้ผลการทดลองนี้แสดงออกมาได้ดังภาพข้างต้นที่แสดงผลที่ได้ออกมา

บทที่ 5
สรุปผลโครงงานและข้อเสนอแนะ

5.1 สรุปผลโครงงาน

จากการศึกษาข้อมูลเกี่ยวกับ ระบบช่วยขับ (Assistant Driving) การตรวจจับช่องทางการเดินรถ (Lane Detection) ระบบควบคุมรถให้อยู่ในช่องทาง (Lane Keeping System) และ Deep Learning เพื่อนำมาจัดทำอัลกอริทึมในโปรแกรมแมทแลป (MATLAB) เพื่อตรวจจับช่องทางการเดินรถบริเวณรอบมหาวิทยาลัยนเรศวร .ในขณะที่ทำการพัฒนาอัลกอริทึมได้มีข้อผิดพลาดเกิดขึ้นทำให้ไม่สามารถนำข้อมูลเน็ตเวิร์ค (Network) มาใช้งานได้ เนื่องจากค่าของตัวเน็ตเวิร์คไม่เปลี่ยนไปตามภาพที่ได้ทำการนำเข้า (Input) ซึ่งปกติแล้วเมื่อนำเข้าภาพที่แตกต่างกันค่าของเน็ตเวิร์คจะเปลี่ยนไปตามภาพที่นำเข้า แต่ในระหว่างที่ทำการทดลองการนำเข้าภาพที่แตกต่างค่าเน็ตเวิร์คมีค่าคงที่ไม่เปลี่ยนไปตามภาพที่ได้นำเข้า ส่งผลให้ตัวเน็ตเวิร์คไม่สามารถใช้งานได้

ในกรณีนี้ผลการทดลองกับภาพที่ได้นั้นใช้ค่าโคเอฟฟิเชียนท์ (Coefficient) ที่ได้จากชุดข้อมูล (Dataset) ที่ได้จากข้อมูลที่ผ่านการใช้งานฟังก์ชันกราวทรูต (Ground Truth Labeler) และนำมาทำการสอนอัลกอริทึม ทำให้ได้ตัวชุดข้อมูลหรือDataset ซึ่งเป็นค่าโคเอฟฟิเชียนท์ที่นำมาการทดลองกับภาพ ซึ่งผลได้จากตรวจจับเลนมีความแม่นยำสูงเพราะภาพที่ทำการนำเข้านั้นเป็นภาพชุดเดียวกันกับการใช้งานฟังก์ชันกราวทรูต แต่ผลการทดลองการคำนวณตำแหน่งที่รถควรอยู่ในช่องทางการเดินรถนั้นผลออกมาไม่ดีเท่าที่ควรจะเป็นเพราะข้อมูลที่ใช้เป็นข้อมูลจาก Dataset ไม่ใช่ข้อมูลจากเน็ตเวิร์คส่งผลให้เกิดความคลาดเคลื่อนพอสมควร

5.2 ปัญหาที่พบ

1) โปรแกรม MATLAB ไม่พร้อมใช้งานเนื่องจากบาง Function และ Application ที่จำเป็นต่อการใช้งาน จะต้องใช้ License แท้เพื่อเปิดใช้งาน

2) คอมพิวเตอร์ที่ใช้ในการประมวลยังมีประสิทธิภาพไม่เพียงพอต่อการใช้งาน

3) เนื่องจากการจราจรมีความหนาแน่น ทำให้ในการเก็บข้อมูลภาพของถนนภายในมหาวิทยาลัยนเรศวร มีความยากพอสมควร ที่จะทำให้ได้ภาพของถนนที่มีรถยนต์หรือรถจักรยานยนต์สัญจรที่น้อย

4) การเก็บข้อมูลภาพใช้ภาพถ่ายจากโทรศัพท์มือทำให้ภาพที่ได้ไม่มีความเสถียร ทำให้ข้อมูลที่ไม่ได้ความแม่นยำที่ต้องการ

5) ปัญหาในการเขียนอัลกอริทึมหรือโค้ด (Code)

6) ไม่สามารถใช้ Network ที่ผ่านการ Train ได้

5.3 ข้อเสนอแนะ

1) ควรจะมี โปรแกรม MATLAB ที่สามารถใช้ Deep learning Toolbox, Deep learning Toolbox, GPU Coder, Computer Vision Toolbox, Automated Driving Toolbox และ Parallel Computing Toolbox ได้

2) คอมพิวเตอร์ที่ใช้ควรเป็นคอมพิวเตอร์ประสิทธิภาพสูง เพื่อใช้ในการประมวลผล

3) ควรเลือกวันและเวลาให้เหมาะสมในการเก็บข้อมูลภาพของถนนภายในมหาวิทยาลัยนเรศวร

4) ควรใช้กล้องติดหน้ารถยนต์สำหรับการเก็บข้อมูลภาพเพื่อที่จะได้มุมภาพที่เท่ากันและราบรื่นตลอดการเก็บข้อมูล

5) ควรมีพื้นฐานในการเขียนอัลกอริทึมหรือโค้ด (Code) หรือควรให้เวลาในการศึกษาและค้นคว้าเกี่ยวกับ โปรแกรม MATLAB และ Code ใช้งานต่าง ๆ

6) ควรศึกษาเชิงลึก Function Train Network และควร เตรียมข้อมูลภาพสำหรับ Train ให้มากขึ้น

7) ข้อมูลสำหรับ Train สามารถแบ่งเป็นหลายๆส่วนได้ เพื่อทำการ train network และสามารถ ทำการ train network ซ้ำได้ด้วยข้อมูลชุดเดิม และข้อมูลชุดใหม่

บรรณานุกรม

1. นายจีระพล คุ่มเคี่ยม.//(2561).//Lane detection.//สืบค้นเมื่อ 22 พฤศจิกายน 2561,/ จาก/http://tdc.thailis.or.th/tdc/search_result.php
2. Mehdi Sqalli.//(2559).//Lane detection.//สืบค้นเมื่อ 23 ตุลาคม 2561,/ จาก/https://medium.com/@MSqalli/lane-detection-446986c44021?fbclid=IwAR1vL3qTJ5l60v5QGhupem2reHXf6slVTr3laOLRBQJqciZ5ssDXDHVqzWpY
3. ผศ.ดร. มหศักดิ์ เกตุฉ่ำ.//(ไม่ทราบ).//edge detection.//สืบค้นเมื่อ 21 พฤศจิกายน 2561,/ จาก/http://www2.it.kmutnb.ac.th/teacher/FileDL/DrMahasak142256116161.pptx
4. Nessessence.//(2561).//Deep learning คืออะไร.//สืบค้นเมื่อ 26 เมษายน 2562,/ จาก/https://www.thaiprogrammer.org/2018/12/deep-learning-คืออะไร/
5. Vithan Minaphinant.//(2561).//Machine Learning คืออะไร.// สืบค้นเมื่อ 26 เมษายน 2562,/จาก/https://blog.finnomena.com/machine-learning-คืออะไร-fa8bf6663c07
6. Matana Wiboonyasake.//(ไม่ทราบ).//Machine Learning คืออะไร.// สืบค้นเมื่อ 26 เมษายน 2562,/จาก/https://www.aware.co.th/machine-learning
7. Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, Xiaoou Tang.//(2561).// Spatial as Deep: Spatial CNN for Traffic Scene Understanding.// สืบค้นเมื่อ 25 เมษายน 2562,/ จาก/https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPaper/16802
8. Anonymous.//2560).//สรุปแนวคิด Neural Network แบบไม่มี Math.// สืบค้นเมื่อ 27 เมษายน 2562,/จาก/https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic
9. Authors list.//(2560).//โครงข่ายประสาทเทียม.//สืบค้นเมื่อ 28 เมษายน 2562,/ จาก/https://th.wikipedia.org/wiki/โครงข่ายประสาทเทียม
10. Avi Nehemiah.//(2560).//Deep Learning for Automated Driving.// สืบค้นเมื่อ 26 เมษายน 2562,/จาก/https://blogs.mathworks.com/deep-learning/2017/11/03/deep-learning-for-automated-driving-part-1-vehicle-

detection/, https://blogs.mathworks.com/deep-learning/2017/11/17/deep-learning-for-automated-driving-part-2-lane-detection/

11. Avi Nehemiah.//(2560).// Deep Learning: Transfer Learning in 10 lines of MATLAB Code.//สืบค้นเมื่อ 26 เมษายน 2562,/ จาก/https://blogs.mathworks.com/pick/2017/02/24/deep-learning-transfer-learning-in-10-lines-of-matlab-code/
12. MathWorks.//(2561).//pixelLabelTrainingData.//สืบค้นเมื่อ 1 พฤษภาคม 2562,/ จาก/https://ch.mathworks.com/help/vision/ref/pixellabeltrainingdata.html
13. MathWorks.//(2560).//fasterRCNNObjectDetector.//สืบค้นเมื่อ 1 พฤษภาคม 2562,/ จาก/https://ww2.mathworks.cn/help/vision/ref/fasterrcnnobjectdetector.html
14. MathWorks.//(2559).//trainNetwork.//สืบค้นเมื่อ 3 พฤษภาคม 2562,/ จาก/https://ww2.mathworks.cn/help/deeplearning/ref/trainnetwork.html
15. MathWorks.//(2559).//trainingOptions.//สืบค้นเมื่อ 3 พฤษภาคม 2562,/ จาก/https://ww2.mathworks.cn/help/deeplearning/ref/trainingoptions.html
16. MathWorks.//(2559).// Pretrained Deep Neural Networks.// สืบค้นเมื่อ 3 พฤษภาคม 2562,/ จาก/https://ww2.mathworks.cn/help/deeplearning/ug/pretrained-convolutional-neural-networks.html
17. MathWorks.//(2549).//polyfit.//สืบค้นเมื่อ 10 พฤษภาคม 2562,/ จาก/https://ww2.mathworks.cn/help/matlab/ref/polyfit.html
18. MathWorks.//(2560).//alexnet.//สืบค้นเมื่อ 10 พฤษภาคม 2562,/ จาก/https://ww2.mathworks.cn/help/deeplearning/ref/alexnet.html
19. นาย กฤษณ์ สอนสำแดง, นายธนบัตร ว่องวัฒนกูล, นายเรืองวิทย์ จันทะโก. อัลกรอลิทึมตรวจจับช่องทางเดินรถสำหรับยานยนต์อัตโนมัติ มหาวิทยาลัยนเรศวร. ปริญญานิพนธ์ กศ.ม.นเรศวร : มหาวิทยาลัยนเรศวร, 2560
20. ผศ.ดร.อาทิตย์ ศรีแก้ว.//(2551).// การตรวจจับถนนด้วยภาพสำหรับระบบนำวิถีอัตโนมัติในพาหนะแบบชาญฉลาด (Vision-based Lane Detection for Automatic Navigation in Smart Vehicle).//สืบค้นเมื่อ 23 พฤศจิกายน 2561,/จาก/ http://sutir.sut.ac.th:8080/sutir/bitstream/123456789/4145/2/Fulltext.pdf

21. Yue Wanga, Eam Khwang Teoha, Dinggang Shenb.//(2557).//
Lane detection and tracking using B-Snake.//สืบค้นเมื่อ 23 พฤศจิกายน 2561,/
จาก/https://www.sciencedirect.com/science/article/pii/S0262885603002105?fbclid=IwAR0M49y9zrWUD8wrxygTW2JruSJ6JeVscPXziJxztOwKu1JnrOOK-Fspu0

22. Frederic BONNAY, Sebastien BRIAUD, Dr. Florent COFFIN, Dr. Zied ZALILA.//
(2544).//Fuzzy Lane-Track Control for the Automatic Guidance of an
Automotive Vehicle.//สืบค้นเมื่อ 23 พฤศจิกายน 2561,/จาก/
https://www.sciencedirect.com/science/article/pii/S1474667017343744?fbclid=IwAR2OfGIsBMpWywjfp7SJhi_E1hGRpf6WkZ3i0q7_HaIVSa95mWT-9CS2zcw

23. นายเกษตร์ ศิริสันติสัมฤทธิ์, นายกมลฐิติ กุลสงค์.//(2545).//การตรวจหาเลนด้วยแม่แบบปรับตัวเองได้
(Lane detection using adaptive template).//สืบค้นเมื่อ 23 พฤศจิกายน 2561,/จาก/
http://kulc.lib.ku.ac.th/kucon/index.php/component/docman/doc_download/99-i11?lang=

24. The Neural Engineer.//(2561).//AI, Machine Learning และ Deep Learning เกี่ยวข้องกันอย่างไร.//สืบค้นเมื่อ 27 เมษายน 2562,/จาก/
https://thaikeras.com/2018/11/10/ai-ml-dl-relationship/

25. MotorTravel.//2558.//ระยะเบรกที่ปลอดภัย.//15 มิถุนายน 2562,/จาก/
https://motortravel.in.th/?p=2323