

# INTELLIGENT ELECTRIC VEHICLE MODEL

โมเดลยานยนต์ไฟฟ้าอัจฉริยะ



นายฐชนัส ประทุมโทน  
นางสาวธัญญลักษณ์ หมั่นธูระ  
นายนนทพัทธ์ ปะละใจ

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# INTELLIGENT ELECTRIC VEHICLE MODEL



Thakanuts Pratumtone

Thanyalak Munthura

Nontapat Palajai

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADEMIC YEAR 2021


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2564  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ Intelligent Electric Vehicle Model  
โมเดลยานยนต์ไฟฟ้าอัจฉริยะ

นักศึกษาผู้จัดทำ นายฐชนัส ประทุมโทน รหัสนักศึกษา 61010264  
นางสาวธัญญลักษณ์ หมั่นธูระ รหัสนักศึกษา 61010494  
นายนนทพัทธ์ ปะละใจ รหัสนักศึกษา 61010533

ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
ปีการศึกษา 2564

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.ดร.ทวีพล ชื้อสัตย์	

หัวข้อวิทยานิพนธ์	Intelligent Electric Vehicle Model		
	โมเดลยานยนต์ไฟฟ้าอัจฉริยะ		
นักศึกษาผู้จัดทำ	นายฐชนัส	ประทุมโทน	รหัสนักศึกษา 61010264
	นางสาวธัญญลักษณ์	หมั่นธูระ	รหัสนักศึกษา 61010494
	นายนนทพัทธ์	ปะละใจ	รหัสนักศึกษา 61010533
อาจารย์ที่ปรึกษา	รศ.ดร.ทวีพล	ชื่อสัตย์	
ปีการศึกษา	2564		

### บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอโมเดลยานยนต์ไฟฟ้าอัจฉริยะ โดยวิธีการแบบ Supervised Learning ซึ่งระบบนี้จะใช้บอร์ด CorgiDude ในการประมวลผลภาพ โดยมีการเก็บและสร้าง Dataset จำนวน 2 ชุด ได้แก่ (1) ตำแหน่งของตัวรถ (2) ป้ายสัญลักษณ์และวัตถุบนถนน จากนั้นนำข้อมูลที่รวบรวมได้ อัปเดตขึ้นไปสอน (training) ด้วยหลักการ Convolutional Neural Network หรือ CNN ที่ Google Colaboratory แล้วนำโมเดล Ai ที่ได้จากการสอน โดยได้ค่าความแม่นยำตำแหน่งของรถ คือ 99.65% และ ค่าความแม่นยำของป้ายสัญลักษณ์และวัตถุบนถนน คือ 99.81% ดาวน์โหลดไปยังตัวบอร์ด โดยผ่านโปรแกรม Kflash\_gui V1.6 จากนั้นเขียนโปรแกรมคำสั่งบอร์ด CorgiDude ผ่านโปรแกรม MaixPy IDE โดยใช้กล้อง OV2640 เป็นอุปกรณ์รับภาพ เพื่อควบคุมตำแหน่งของโมเดลยานยนต์ไฟฟ้าอัจฉริยะและสามารถทำตามคำสั่งป้ายสัญลักษณ์และวัตถุบนถนนที่กำหนดได้ โดยกรณีศึกษาในวิทยานิพนธ์นี้ได้กำหนดไว้จำนวน 20 ชนิดโดยมีการแบ่งหมวดหมู่เป็น 5 หมวดหมู่ดังนี้ (1) background คือ ถนนและขอบทาง (2) Left คือ ป้ายเลี้ยวซ้ายและรูปทรง (3) Obstruction คือ กระจกบังโคลน, กระจกบังกาแฟ, กระจกบังวิตซ่า, กระจกปุกไม้และบอล (4) Right คือ ป้ายเลี้ยวขวาและรูปทรง (5) Stop คือ หุ่นมนุษย์, หุ่นแมว กล้อง, กล้องแว่นและป้ายหยุด ผลการทดลองพบว่ารถสามารถวิ่งอยู่ในเส้นทางที่กำหนดและทำตามคำสั่งป้ายสัญลักษณ์และหลบสิ่งกีดขวางได้อย่างถูกต้อง สามารถนำไปใช้กับโมเดลยานยนต์ไฟฟ้าอัจฉริยะโดยการใช้ปัญญาประดิษฐ์ (AI) ได้และพัฒนาใช้กับยานยนต์ไฟฟ้าจริงได้ในอนาคต

**Thesis Title** Intelligent Electric Vehicle Model  
**Author** Thakanuts Pratumtone Student ID 61010264  
Thanyalak Munthura Student ID 61010494  
Nontapat Palajai Student ID 61010533  
**Thesis Advisor** Assoc.Prof.Dr.Taweepol Suesut  
**Academic Year** 2021

## ABSTRACT

This thesis presents an Intelligent Electric vehicle Model. Using Supervised Learning, which uses the CorgiDude board to process images. By collecting and creating 2 sets of datasets such as the position of vehicle and the traffic symbol on the road to upload to training using Convolutional Neural Network (CNN) principles at Google Colaboratory were done. The training accuracy of AI model is 99.65% for the vehicle position and 99.81% for traffic symbol. The programKflash\_gui V1.6 is used for downloading to the CorgiDude board. The MaixPy IDE program uses the OV2640 camera as a device to receive images to control the position of the electric vehicle and be able to follow the signs and objects on the specified road by the case studies in this thesis 20 types are defined, which are categorized into 5 categories as follows: (1) background is road and curb (2) Left is left turn sign and shape (3) Obstruction is a can of coke, a coffee can, a can of visage, a wooden jar and a ball (4) Right is a right turn sign and a shape (5) Stop is a human figure, a cat puppet boxes, glasses cases and stop signs. The results show that the car is able to run in the specified route and follow the symbols and the Obstacles correctly. It can be used as a learning model for intelligent electric vehicle model using artificial intelligence (AI) and can be applied to the real vehicle in the future.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วย ความกรุณาเป็นอย่างสูงจากการให้คำปรึกษาจาก อาจารย์ที่ปรึกษา ได้แก่ รองศาสตราจารย์ ดร.ทวีพล ชี้อัสตัย และคณะอาจารย์ที่เป็นคณะกรรมการการ สอบปริญญานิพนธ์ทุกท่าน คำแนะนำในขั้นตอนที่ได้ทำการศึกษาปริญญานิพนธ์ การวางแผนการศึกษา กระบวนการออกแบบผลงาน และการนำเสนอผลงานการออกแบบที่ถูกต้องตามหลักวิชา ตลอดจนการ เขียนรายงานและรูปเล่มปริญญานิพนธ์เพื่อนำเสนอผลงาน

ขอขอบคุณคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง อันเป็นสถานศึกษาที่ประสิทธิ์ ประสาทความรู้ ขอกราบขอบพระคุณคณาจารย์ประจำคณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ได้ให้การสั่งสอนวิชา ทางวิศวกรรม ให้คำแนะนำ และข้อคิดเห็นอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์

ขอขอบคุณเจ้าหน้าที่และบุคลากรคณะวิศวกรรมศาสตร์ทุกท่านที่ให้ความช่วยเหลือและ อำนวยความสะดวกในด้านต่างๆ ขอขอบคุณเพื่อนนักศึกษาคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระ จอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ให้คำแนะนำและเป็นกำลังใจในการศึกษาตลอดมา

ผู้วิจัยมีความซาบซึ้งในความกรุณาของทุกท่านที่ได้กล่าวถึงและผู้ที่ไม่ได้เอ่ยนามในที่นี้ ได้ มีส่วนช่วยเหลือในการสนับสนุนให้กำลังใจด้วยดีตลอดมา จึงขอกราบขอบพระคุณทุกท่าน ด้วยความ จริงใจ และขอมอบคุณประโยชน์อันเกิดจากวิทยานิพนธ์ฉบับนี้เป็นกตเวทิตาคุณแด่ บิดามารดา ครู อาจารย์ และผู้เกี่ยวข้องทุกท่านที่ให้การสนับสนุนและเป็นกำลังใจด้วยดีเสมอมา ขอน้อมคารวะแด่ผู้เขียน ตำราวิชาการที่ได้ศึกษาค้นคว้าและใช้อ้างอิงทุกท่าน

คณะผู้จัดทำ

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	IX
สารบัญภาพประกอบ.....	X
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	2
1.3 ขอบเขตของปริญญาโท.....	2
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
<b>บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....</b>	<b>4</b>
2.1 เทคโนโลยีรถยนต์ไฟฟ้า.....	4
2.1.1 รถยนต์ไฮบริดพลังงานไฟฟ้าแบบชาร์จไม่ได้.....	4
2.1.2 รถยนต์ไฮบริดพลังงานไฟฟ้าแบบชาร์จได้.....	4
2.1.3 รถยนต์พลังงานไฟฟ้าจากแบตเตอรี่โดยสมบูรณ์.....	4
2.2 รถยนต์ไร้คนขับ.....	5
2.2.1 Navigation.....	5
2.2.2 Computer Vision.....	5

## สารบัญ (ต่อ)

หน้า

2.2.3 Deep Learning.....	5
2.2.4 Robotics.....	5
2.3 การประมวลผลภาพดิจิทัล.....	5
2.3.1 เทคนิคที่ใช้ในการประมวลผลภาพ.....	7
2.4 ระบบคอมพิวเตอร์วิทัศน์.....	8
2.4.1 หลักการทำงานของระบบคอมพิวเตอร์วิทัศน์.....	8
2.4.2 การเรียนรู้เชิงจักรกล.....	9
2.4.3 การทำงานของ การเรียนรู้เชิงจักรกล.....	9
2.5 การเรียนรู้ของเครื่อง.....	10
2.6 Supervised learning.....	12
2.6.1 กระบวนการทำงานของ Supervised learning.....	12
2.6.2 ขั้นตอนของกระบวนการ Supervised learning.....	13
2.7 ประเภทของกระบวนการ Supervised learning.....	14
2.7.1 Regression.....	14
2.7.1.1 Linear Regression.....	14
2.7.1.2 Logistic Regression.....	14
2.7.2 Classification.....	15
2.7.2.1 Decision Tree.....	16
2.7.2.2 Support Vector Regression.....	16
2.7.2.3 Random Forest Regressor.....	17
2.8 การเรียนรู้เชิงลึก Deep Learning.....	18
2.8.1 โครงข่ายประสาทเทียมอัจฉริยะ.....	19
2.8.1.1 ชั้นข้อมูลเข้า (Input Layer).....	20
2.8.1.2 ชั้นซ่อน (Hidden Layer).....	20
2.8.1.3 ชั้นข้อมูลออก (Output Layer).....	20

## สารบัญ (ต่อ)

หน้า

2.8.2 ฟังก์ชันกระตุ้น.....	21
2.8.2.1 ฟังก์ชันกระตุ้นค่าแบ่ง (Threshold Activation Function).....	21
2.8.2.2 ฟังก์ชันกระตุ้นซิกมอยด์ (Sigmoid Activation Function).....	21
2.8.2.3 ฟังก์ชันไฮเพอร์โบลิกแทงก์เจนท์ (Hyperbolic Tangent Function : tanh).....	21
2.8.2.4 ฟังก์ชันเรคตีไฟด์ไลน์เนี่ยยูนิต (Rectified Linear Units, ReLu).....	21
2.9 โครงข่ายประสาทเทียมคอนโวลูชัน.....	22
2.9.1 ชั้นคอนโวลูชัน (Convolutional layer).....	22
2.9.1.1 Padding.....	25
2.9.1.2 Stride.....	25
2.9.2 ชั้นพูลิ่ง (Pooling layer).....	25
2.10 การถ่ายโยงการเรียนรู้.....	26
2.11 MobileNet.....	28
<b>บทที่ 3 วิธีการดำเนินการวิจัย .....</b>	<b>29</b>
3.1 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินการ.....	29
3.1.1 คอมพิวเตอร์.....	29
3.1.2 บอร์ดCorgiDude.....	30
3.1.3 กล้อง OV2640.....	30
3.1.4 หน้าจอแสดงผลภาพ .....	31
3.1.5 บอร์ดKB32-FT.....	31
3.1.6 KB Motor Driver.....	32

## สารบัญ (ต่อ)

หน้า

3.2 Software ที่ใช้ดำเนินการ.....	33
3.2.1 โปรแกรม Colaboratory .....	33
3.2.2 โปรแกรม KB-IDE.....	34
3.2.3 Maixpy IDE.....	35
3.2.4 kflash_gui.....	36
3.3 การออกแบบระบบ.....	36
3.3.1 หลักการออกแบบการขับเคลื่อนโมเดลยานยนต์ไฟฟ้าอัจฉริยะ .....	36
3.3.2 ขั้นตอนการทำงานของ CorgiDude.....	37
3.4 โครงสร้างภายในของบอร์ด CorgiDude .....	38
3.5 โครงสร้างภายในของบอร์ด KB-32FT.....	40
3.6 การวางตำแหน่งอุปกรณ์ภายในโมเดลยานยนต์ไฟฟ้าอัจฉริยะ .....	42
3.7 ขั้นตอนในการสอน AI.....	43
3.7.1 สอน AI ในการจำแนกป้ายสัญลักษณ์และวัตถุบนถนน.....	43
3.7.2 สอน AI ให้จดจำตำแหน่งของโมเดลยานยนต์ไฟฟ้าอัจฉริยะ.....	47
<b>บทที่ 4 ผลการดำเนินงาน .....</b>	<b>53</b>
4.1 การเก็บข้อมูลรูปภาพเป็น Dataset .....	53
4.2 การทดสอบความแม่นยำของตำแหน่งรถ .....	54
4.3 การทดสอบความแม่นยำของการตรวจจับและจำแนกป้ายสัญลักษณ์และวัตถุบนถนน .....	56
4.4 ผลการทดสอบค่าความแม่นยำในการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะ.....	60
4.5 การทดสอบการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะในสนามจริง.....	65
4.6 สรุปผลการทดลอง.....	66

## สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลปัญหาและข้อเสนอแนะ .....	68
5.1 สรุปผลการดำเนินงาน.....	68
5.2 ปัญหาที่พบและแนวทางในการแก้ปัญหา.....	69
5.2.1 ปัญหาที่พบ.....	69
5.2.2 แนวทางการแก้ไขปัญหา.....	69
5.3 ข้อเสนอแนะ .....	69
บรรณานุกรม .....	70
References .....	72
ภาคผนวก ก.....	73
ภาคผนวก ข.....	75
ภาคผนวก ค.....	76

## สารบัญตาราง

ตารางที่	หน้า
3-1 คุณสมบัติของคอมพิวเตอร์ที่ใช้งานและสอน AI.....	29
3-2 รายละเอียดของกล้อง.....	30
3-3 คุณสมบัติของ KB Motor Driver.....	32
4-1 ผลการทดสอบค่าความแม่นยำของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะ.....	55
4-2 ผลการทดสอบความแม่นยำในตรวจจับป้ายสัญลักษณ์และวัตถุบนถนน.....	58
4-3 ผลการทดสอบความแม่นยำในการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะบนสนาม.....	63
4-4 ผลการทดสอบการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะในสนาม.....	66

## สารบัญภาพประกอบ

ภาพที่	หน้า
2.1 ภาพดิจิทัลอาร์เรย์ 2 มิติ.....	6
2.2 การแสดงภาพ RGB.....	6
2.3 การทำงานของ Machine Learning.....	10
2.4 หลักการประยุกต์ใช้แก้ปัญหาได้ 2 รูปแบบของ Supervised Learning.....	11
2.5 กระบวนการสอนเพื่อให้ได้โมเดลที่ต้องการ.....	11
2.6 โมเดลที่ประยุกต์ใช้กับโปรแกรมของคอมพิวเตอร์.....	11
2.7 การทำงานของ Supervised Learning.....	12
2.8 กราฟแสดงอัลกอริทึมของ Linear Regression.....	14
2.9 กราฟแสดงอัลกอริทึมของ Logistic Regression.....	15
2.10 แผนผังแสดงอัลกอริทึมของ Decision Tree.....	16
2.11 กราฟแสดงอัลกอริทึมของ Support Vector Regression.....	17
2.12 แผนภาพแสดงอัลกอริทึมของ Random Forest Regressor.....	18
2.13 ความแตกต่างระหว่างการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก.....	19
2.14 ส่วนประกอบของโครงข่ายประสาทเทียม.....	19
2.15 โครงสร้างการทำงานของโครงข่ายประสาทเทียม.....	20
2.16 ฟังก์ชันลิสต์ที่เรคตีไฟด์ลินเนียล.....	22
2.17 ภาพแสดงชั้นคอนโวลูชัน.....	23
2.18 ภาพแสดงชั้นพูลลิ่ง.....	26
2.19 การใช้เทคนิคการถ่ายโยงการเรียนรู้ (Transfer Learning).....	27
2.20 โมเดล MobileNet.....	28
3.1 คอมพิวเตอร์.....	29
3.2 บอร์ด CorgiDude.....	30

## สารบัญภาพประกอบ (ต่อ)

ภาพที่	หน้า
3.3 กล้อง OV2640.....	30
3.4 หน้าจอแสดงผลภาพ.....	31
3.5 บอร์ด KB32-FT.....	31
3.6 KB Motor Driver โมดูลขับเคลื่อนมอเตอร์.....	32
3.7 หน้าต่างของโปรแกรม Colaboratory.....	33
3.8 หน้าต่างของโปรแกรม KB-IDE.....	34
3.9 หน้าต่างของโปรแกรม Maixpy IDE.....	35
3.10 หน้าต่างของโปรแกรม Kflash_gui.....	36
3.11 แผนผังการทำงานของระบบขับเคลื่อนโมเดลยานยนต์ไฟฟ้าอัจฉริยะแบบอัตโนมัติ.....	37
3.12 การเชื่อมต่อวงจร CorgiDude กับโมเดลยานยนต์ไฟฟ้าอัจฉริยะ.....	38
3.13 รายละเอียดโครงสร้างภายในของบอร์ด CorgiDude.....	38
3.14 รายละเอียดโครงสร้างภายในของบอร์ด KB-32FT.....	40
3.15 โครงสร้างภายในของบอร์ด KB-32FT พร้อม Breakout Board.....	41
3.16 การวางตำแหน่งของอุปกรณ์ภายในโมเดลยานยนต์ไฟฟ้าอัจฉริยะ.....	42
3.17 การแยก Dataset ของป้ายสัญลักษณ์และวัตถุบนถนน.....	43
3.18 ตัวอย่าง Dataset ของป้ายสัญลักษณ์และวัตถุบนถนนที่ใช้ทำการทดลอง.....	43
3.19 การอัปโหลด Dataset ลงในโปรแกรม Calaboratory.....	44
3.20 หน้าต่างติดตั้ง Library.....	44
3.21 ไฟล์ที่ได้จากการแตกไฟล์เข้าโปรแกรม.....	45
3.22 การแบ่งไฟล์รูปภาพป้ายสัญลักษณ์และวัตถุบนถนนต่างๆเพื่อนำไปทำการสอน AI.....	45
3.23 การตั้งค่าในการสอน Dataset.....	45
3.24 คำสั่งในการสอน Dataset.....	46
3.25 การสุ่มข้อมูลตัวอย่างเพื่อตรวจสอบความแม่นยำ.....	46

## สารบัญภาพประกอบ (ต่อ)

ภาพที่	หน้า
3.26 การอัปโหลดไฟล์ .kmodel จากการสอน AI ลงบอร์ด CorgiDude ที่ตำแหน่ง 0x500000.....	47
3.27 คำสั่งควบคุมรถเพื่อทำการเก็บ Dataset ที่ตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะ.....	47
3.28 ตัวอย่าง Dataset ของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะในขอบทาง.....	48
3.29 ตัวอย่างข้อมูลในไฟล์ .csv บอกตำแหน่งที่เก็บรูปภาพและตำแหน่งของรถในแต่ละรูปภาพ.....	48
3.30 อัปโหลด Dataset ลงในโปรแกรม Colaboratory.....	49
3.31 คำสั่งติดตั้ง Library ที่ใช้ในการสอน AI.....	49
3.32 การแตกไฟล์รูปภาพเข้าไปเก็บไว้ใน KB32FT_Rover_kit.....	49
3.33 คำสั่งปรับขนาดรูปภาพก่อนนำไปสอน AI.....	50
3.34 การแบ่งรูปภาพและกำหนดจำนวนครั้งในการสอน.....	50
3.35 คำสั่งการสอน Dataset และได้ผลลัพธ์ที่ดีที่สุด.....	51
3.36 คำสั่งแปลงไฟล์ .h5 เป็นไฟล์ .kmodel.....	51
3.37 อัปโหลดไฟล์ .kmodel จากการสอน AI ลงบอร์ด CorgiDude ที่ตำแหน่ง 0x400000.....	52
4.1 ตัวอย่างของ Dataset.....	53
4.2 โปรแกรม MaixPy IDE สำหรับตรวจสอบความแม่นยำของตำแหน่งโมเดลยานยนต์ไฟฟ้า อัจฉริยะ.....	54
4.3 ค่าตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะที่แสดงผ่านโปรแกรม MaixPy IDE.....	54
4.4 กราฟแสดงผลการทดสอบค่าความแม่นยำของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะ.....	56
4.5 โปรแกรมคำสั่งตรวจจับการเคลื่อนที่และจำแนกป้ายสัญลักษณ์และวัตถุบนถนน.....	57
4.6 ตัวอย่างการจำแนกป้ายสัญลักษณ์และวัตถุบนถนนจากหน้าจอแสดงภาพ.....	57
4.7 ตัวอย่างการตรวจจับที่ผิดพลาด.....	60
4.8 โปรแกรมคำสั่งการเคลื่อนที่, ตรวจจับ, จำแนกป้ายสัญลักษณ์และวัตถุบนถนน.....	61
4.9 ตัวอย่างการเคลื่อนที่หมวดหมู่ Left.....	61

## สารบัญภาพประกอบ (ต่อ)

ภาพที่	หน้า
4.10 ตัวอย่างการเคลื่อนที่หมวดหมู่ Right.....	62
4.11 ตัวอย่างการเคลื่อนที่หมวดหมู่ Stop.....	62
4.12 ตัวอย่างการเคลื่อนที่หมวดหมู่ Obstruction.....	62
4.13 การทดสอบการเคลื่อนที่ในสนามจริง.....	65



# บทที่ 1

## บทนำ

### 1.1 ความสำคัญของปัญญาประดิษฐ์

เทคโนโลยียานยนต์ที่ขับเคลื่อนด้วยไฟฟ้า กำลังเป็นที่สนใจในประเทศไทยและทั่วโลก ซึ่งในปัจจุบันหลายองค์กรได้นำเทคโนโลยี AI หรือปัญญาประดิษฐ์ มาประยุกต์ใช้ในการทำงานทางธุรกิจ หลากหลายหน้าที่ ไม่ว่าจะเป็นการตลาด การค้นคว้าและวิจัย (R&D) และอื่นๆ เนื่องจากผู้คนส่วนใหญ่ เริ่มเล็งเห็นประโยชน์ของการนำระบบ AI ที่จะช่วยสร้างมูลค่าเพิ่มให้กับธุรกิจทั้งทางตรงและทางอ้อม และเป็นอีกหนึ่งนวัตกรรมที่สามารถนำไปต่อยอดได้ จึงได้มีการนำ AI (Artificial Intelligence) หรือ ระบบปัญญาประดิษฐ์ เข้ามาเพิ่มประสิทธิภาพของรถยนต์ที่ขับเคลื่อนด้วยไฟฟ้าเพื่อให้เกิดความสะดวกในการใช้งานรวมถึงระบบป้องกันอุบัติเหตุเพื่อความปลอดภัยของผู้ขับขี่ โดยมีหลักการ คือ คอมพิวเตอร์วิทัศน์ (Computer Vision) เปรียบเสมือนส่วนของการรับรู้ของรถ คือการใช้ กล้อง เรดาร์ หรือเลเซอร์ เพื่อให้สามารถมองเห็น รั้วระยะ ความใกล้ไกลของวัตถุ เป็นข้อมูลเพื่อใช้ในการ ขับขี่อัตโนมัติ การเรียนรู้เชิงลึก (Deep Learning) เปรียบเสมือนสมองของรถยนต์ไร้คนขับ หลังจากทีรถ มีข้อมูลที่ได้จากกล้องหรือเซนเซอร์ต่างๆ ก็จะส่งข้อมูลเข้ามาที่ส่วนของสมองส่วนนี้ ซึ่งทำหน้าที่ตั้งแต่ทำความเข้าใจว่ารูปที่ได้จากกล้องนั้น ส่วนไหนเป็นถนน หรือคนกำลังข้ามถนน จนไปถึงในการตัดสินใจว่าจะ ให้รถขับเคลื่อนไปในทิศทางไหน เลี้ยวซ้ายหรือเลี้ยวขวา หุ่นยนต์ (Robotic) เปรียบเสมือนเส้นประสาท หลังจากที่ได้ข้อมูลในการตัดสินใจจากสมองมาแล้ว ส่วนนี้จะทำหน้าที่แปลงสัญญาณที่ได้และ สัญญาณไฟฟ้าส่งไปที่ล้อและการควบคุมต่างๆ เพื่อให้รถยนต์อัตโนมัติสามารถเดินทางไปได้ การนำทาง (Navigation) เป็นระบบส่วนที่ใช้ในการติดต่อกับผู้ใช้งานสำหรับการตั้งค่าการเดินทาง ต้นทางไปยัง ปลายทาง

ปัจจุบันได้มีการใช้รถยนต์ไฟฟ้าอย่างแพร่หลาย ผู้ศึกษาได้เล็งเห็นถึงประโยชน์ที่มากมาย ของปัญญาประดิษฐ์ (Artificial Intelligence) ที่สามารถมาพัฒนาควบคุมรถยนต์อัตโนมัติได้ จึงได้จัดทำ ปริญญาณิพนธ์ฉบับนี้ขึ้นมา เพื่อเป็นแหล่งการเรียนรู้ที่ให้กับบุคคลที่สนใจได้มาศึกษาและต่อยอดได้

## 1.2 วัตถุประสงค์ของปฏิญานิพนธ์

- 1.2.1. เพื่อสร้างแบบจำลองยานยนต์ไฟฟ้าอัจฉริยะที่อาศัยกล้องและระบบสมองกลฝังตัว
- 1.2.2. ศึกษาการวิธีการประยุกต์การเรียนรู้เชิงลึกบนระบบสมองกลฝังตัว
- 1.2.3. เพื่อพัฒนาต้นแบบระบบช่วยขับซึ่งสำหรับประยุกต์ใช้กับยานยนต์ไฟฟ้าในอนาคต

## 1.3 ขอบเขตของปฏิญานิพนธ์

ใช้วิธีการ Supervised Learning ซึ่งระบบนี้จะใช้บอร์ด CorgiDude ในการประมวลผลภาพ โดยมีการเก็บและสร้าง Dataset จำนวน 2 ชุดได้แก่ (1) ตำแหน่งของตัวรถ (2) ป้ายสัญลักษณ์และวัตถุนถนน จากนั้นนำข้อมูลที่รวบรวมได้ อัปโหลดขึ้นไปสอน (training) ด้วยหลักการ Convolutional Neural Network หรือ CNN ที่ Google Colaboratory แล้วนำ Ai Model ที่ได้จากการเทรนหาค่าความแม่นยำที่สุดของตำแหน่งรถและป้ายสัญลักษณ์ ไปใช้งานโดยดาวน์โหลดไปยังตัวบอร์ด โดยผ่านโปรแกรม Kflash\_gui V1.6 จากนั้นเขียนโปรแกรมคำสั่งบอร์ด CorgiDude ผ่านโปรแกรม MaixPy IDE โดยใช้กล้อง OV2640 เป็นอุปกรณ์รับภาพ เพื่อควบคุมตำแหน่งของโมเดลยานยนต์ไฟฟ้าอัจฉริยะและสามารถทำตามคำสั่งป้ายสัญลักษณ์และหลบหลีกวัตถุนถนนที่กำหนดได้ เพื่อให้รถสามารถวิ่งอยู่ในเส้นทางที่กำหนด และทำตามคำสั่งป้ายสัญลักษณ์ได้อย่างถูกต้อง สามารถนำไปใช้เป็นแบบเรียนรู้เรื่องรถยนต์ขับเคลื่อนอัตโนมัติโดยการใช้ปัญญาประดิษฐ์ (Artificial Intelligence) ได้ซึ่งจะช่วยเรื่องความสะดวกสบายและความปลอดภัยบนท้องถนนได้เป็นอย่างดี

## 1.4 ขั้นตอนการศึกษา

- 1.4.1 ศึกษาหลักการการทำงานของกล้องกับการเชื่อมต่อข้อมูลคอมพิวเตอร์วิทัศน์ (Computer Vision)
- 1.4.2 ศึกษาการทำงานของบอร์ด KB-32FT
- 1.4.3 ศึกษาการเรียนรู้เชิงลึกของระบบคอมพิวเตอร์วิทัศน์ (Computer Vision) ในการประมวลผลรูปภาพ
- 1.4.4 ศึกษาหลักการ Convolutional Neural Network (CNN)
- 1.4.5 ศึกษาการใช้งานโปรแกรม KB IDE
- 1.4.6 ศึกษาการแยกแยะรูปภาพของบอร์ด CorgiDude

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ได้แบบจำลองยานยนต์ไฟฟ้าอัจฉริยะเพื่อใช้ในการเรียนรู้
- 1.5.2 ได้รับความรู้ ความเข้าใจ ขั้นตอนการทำงานและการแก้ปัญหาเกี่ยวกับเรื่องการเรียนรู้เชิงลึกบนระบบสมองกลฝังตัว
- 1.5.3 ได้รับความรู้เพื่อนำไปพัฒนาต่อยอดระบบช่วยขับขี่ยานยนต์ไฟฟ้าอัตโนมัติ ซึ่งสามารถเพิ่มความปลอดภัยและสะดวกสบายให้กับผู้ขับขี่



## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

#### 2.1 เทคโนโลยีรถยนต์ไฟฟ้า

ปัจจุบันเทคโนโลยียานยนต์ที่ขับเคลื่อนด้วยพลังงานไฟฟ้า และยานยนต์ไร้คนขับ เป็นอีกหนึ่งนวัตกรรมที่กำลังได้รับความสนใจจากทั่วโลก รถยนต์ไฟฟ้า (Electric Car) หมายถึง รถยนต์ที่ขับเคลื่อนด้วยแรงบิดของมอเตอร์ไฟฟ้าโดยใช้พลังงานไฟฟ้าที่สร้างมาจากแบตเตอรี่หรืออุปกรณ์กักเก็บพลังงานไฟฟ้ารูปแบบอื่นๆ โดยเทคโนโลยีที่ทำให้เกิดความแตกต่างระหว่างรถยนต์ไฟฟ้า และรถยนต์ใช้น้ำมันโดยทั่วไปมีอยู่ 2 ส่วน คือ (1) แบตเตอรี่ ซึ่งเป็นตัวเก็บพลังงานของรถยนต์ไฟฟ้า ที่ปัจจุบันใช้แบบ Lithium-Ion ซึ่งเป็นแบตเตอรี่ชนิดเดียวกับที่ใช้ในโทรศัพท์เคลื่อนที่ ซึ่งยังคงมีราคาสูงอยู่ และต้องได้รับการพัฒนาอย่างต่อเนื่องเพื่อรองรับการพัฒนาของรถยนต์ไฟฟ้า และ (2) ชิ้นส่วนในระบบส่งกำลัง เช่น มอเตอร์ไฟฟ้า ตัวแปลงกระแสไฟฟ้า ทั้งนี้ รถยนต์ไฟฟ้าสามารถแบ่งได้เป็น 3 ประเภท ได้แก่

##### 2.1.1 รถยนต์ไฮบริดพลังงานไฟฟ้าแบบชาร์จไม่ได้ (Hybrid-electric Vehicles หรือ HEV)

เป็นรถยนต์ประเภทที่มีเครื่องยนต์เหมือนกับรถยนต์น้ำมันทั่วไป แต่มีขนาดเล็กกว่า และใช้การสร้างพลังงานจากเครื่องยนต์สันดาป หรือเครื่องยนต์ใช้น้ำมันและจากการชาร์จกระแสไฟฟ้าในแบตเตอรี่ไฟฟ้า

##### 2.1.2 รถยนต์ไฮบริดพลังงานไฟฟ้าแบบชาร์จได้ (Plug-in Hybrid Electric Vehicles หรือ PHEV)

เป็นประเภทของรถยนต์ที่ต่างกับ HEV เพราะมีการใส่มอเตอร์ไฟฟ้าสำหรับชาร์จกระแสไฟฟ้าเพิ่มเข้ามาทำงานควบคู่ไปกับเครื่องยนต์สันดาป

##### 2.1.3 รถยนต์พลังงานไฟฟ้าจากแบตเตอรี่โดยสมบูรณ์ (Battery Electric Vehicles หรือ BEV)

เป็นรถยนต์ประเภทที่มีแต่มอเตอร์ไฟฟ้า ไม่มีเครื่องยนต์สันดาปภายใน ตัวมอเตอร์ จะชาร์จพลังงานไฟฟ้าจากแบตเตอรี่ไฟฟ้าและสร้างแรงบิดให้รถยนต์ขับเคลื่อน

## 2.2 รถยนต์ไร้คนขับ (Autonomous Car หรือ Self-driving Car)

รถยนต์ไร้คนขับ (Autonomous Car หรือ Self-driving Car) หรือรถยนต์ที่สามารถขับเคลื่อนด้วยตัวเองโดยไม่ต้องอาศัยคน รถยนต์ขับเคลื่อนด้วยตัวเองอัตโนมัติจะทำงานร่วมกันกับเทคโนโลยี ดังนี้

### 2.2.1 Navigation

Navigation หรือระบบแผนที่ ซึ่งประกอบด้วยระบบการระบุตำแหน่งของรถยนต์ขับเคลื่อนอัตโนมัติจากดาวเทียม ข้อมูลที่เก็บคือข้อมูลทุกอย่างที่เกี่ยวข้องกับการวิ่งของรถบนถนน เช่น ตำแหน่งของไฟจราจร ตำแหน่งป้ายเลี้ยว ป้ายสัญญาณห้ามไป ความกว้างของเลนถนน รวมถึงความเร็วสูงสุดที่กฎหมายอนุญาตให้รถวิ่งได้ในถนนแต่ละเส้น ซึ่งจะประมวลผลร่วมกับระบบ Sensor เพื่อเพิ่มความถูกต้องและแม่นยำในการตัดสินใจ

### 2.2.2 Computer Vision

Computer Vision หรือระบบที่ทำหน้าที่เป็นตาและหูให้กับรถยนต์ขับเคลื่อนอัตโนมัติ โดยตรวจจับการเคลื่อนที่การเปลี่ยนแปลงเมื่อรถกำลังวิ่ง

### 2.2.3 Deep Learning

Deep Learning หรือระบบประมวลผลปัญญาประดิษฐ์ ทำหน้าที่เหมือนสมองของรถยนต์ไร้คนขับ สามารถตัดสินใจประมวลผลด้วยตัวเองโดยข้อมูลที่รับมาจากระบบ Computer Vision

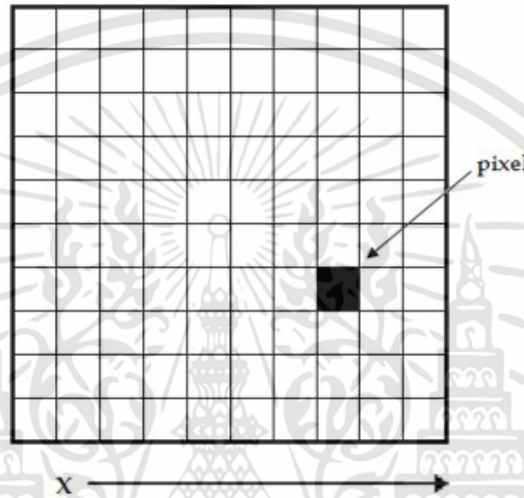
### 2.2.4 Robotics

Robotics หรือระบบที่เชื่อมต่อระบบประมวลผลส่วนกลางเข้ากับระบบเครื่องจักรต่างๆ ในตัวรถโดยทำหน้าที่เสมือนเส้นประสาทที่เชื่อมต่อสมองของมนุษย์เข้ากับแขนขาและส่วนต่างๆ ของร่างกายที่สามารถทำให้รถขับเคลื่อนไปได้

## 2.3 การประมวลผลภาพดิจิทัล

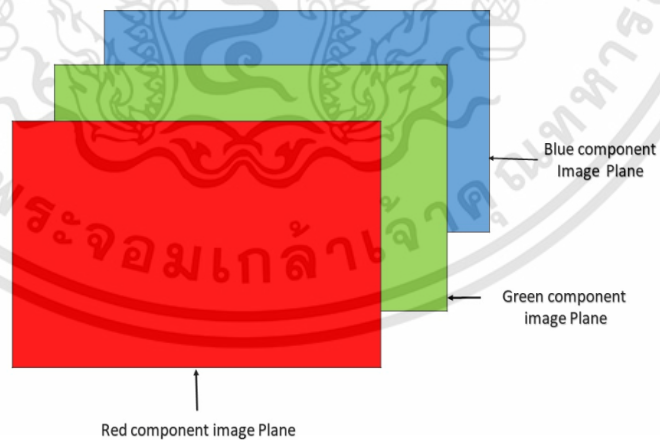
การประมวลผลภาพหมายถึงการประมวลผลภาพ และอาจรวมถึงเทคนิคต่างๆ มากมาย จนกว่าจะได้ผลลัพธ์สุดท้ายสามารถอยู่ในรูปแบบของรูปภาพที่เกี่ยวข้องของรูปภาพนั้นก็ได้ สามารถใช้สำหรับการวิเคราะห์และการตัดสินใจได้มากขึ้น

โดยรูปภาพสามารถแสดงเป็นฟังก์ชัน 2 มิติ  $F(x,y)$  โดยที่  $x$  และ  $y$  เป็นพิกัดเชิงพื้นที่ แอมพลิจูดของ  $F$  ที่ค่าเฉพาะของ  $x,y$  เรียกว่าความเข้มของภาพที่จุดนั้น ถ้า  $x,y$  และค่าแอมพลิจูดมีจำกัด เราจะเรียกว่าภาพดิจิทัล เป็นอาร์เรย์ของพิกเซลที่จัดเรียงเป็นคอลัมน์และแถว รูปภาพยังสามารถแสดงในรูปแบบ 3 มิติ โดยที่  $x,y$  และ  $z$  กลายเป็นพิกัดเชิงพื้นที่ พิกเซลถูกจัดเรียงในรูปแบบของเมทริกซ์ สิ่งนี้เรียกว่าภาพ RGB



<https://www.researchgate.net/figure/A-digital-image-is-a-2D-array-of-pixels> , [Online]

ภาพที่ 2.1 ภาพดิจิทัลอาร์เรย์ 2 มิติ



[ RGB image is a three layered image ]

<https://www.geeksforgeeks.org/matlab-rgb-image-representation> , [Online]

ภาพที่ 2.2 การแสดงภาพ RGB

### 2.3.1 เทคนิคที่ใช้ในการประมวลผลภาพ

- 1) ปรับปรุงคุณภาพของภาพ (Image Enhancement) เป็นกระบวนการในการแปลงข้อมูลภาพตัวเลข เพื่อที่จะสร้างภาพที่เน้นรายละเอียดที่ต้องการ อาจจะมีการปรับแสงของภาพเพื่อดูสิ่งที่ต้องการ
- 2) การกรองภาพหรือการกำจัดสัญญาณรบกวนออกจากภาพ (Image Filters) คือการนำภาพไปผ่านตัวกรองสัญญาณเพื่อให้ได้ภาพผลลัพธ์ออกมา โดยภาพผลลัพธ์นั้นจะแตกต่างจากภาพเดิม จุดประสงค์หลักของการกรองภาพคือการเน้น (enhance) หรือลดทอน (attenuate) คุณสมบัติบางประการของภาพ เพื่อให้ได้คุณสมบัติของภาพตามต้องการ
- 3) การซ้อนทับภาพ (Image Registration) เป็นวิธีการนำข้อมูลของภาพสองภาพขึ้นไปมารวมกัน โดยจุดประสงค์เพื่อให้ได้ภาพที่มีรายละเอียดและข้อมูลที่เพียงพอสำหรับการนำไปใช้
- 4) การคืนสภาพของภาพ (Image Restoration) การทำให้ภาพคืนสู่สภาพเดิม หรือการปรับปรุงภาพให้เหมาะสมกับการมองเห็นได้ดีขึ้น
- 5) การแบ่งส่วนภาพ (Image Segmentation) เป็นวิธีการแบ่งส่วนใดส่วนหนึ่งของภาพที่เราสนใจออกมาจากภาพที่เราต้องการ จะมีความสำคัญมากสำหรับการแพทย์ เช่น ก้อนเนื้อ กระดูก อวัยวะข้างเคียง หรือแม้กระทั่งสัญญาณรบกวน (Noise) ที่ขึ้นในขณะที่ถ่ายภาพ
- 6) การหาขอบภาพในวัตถุ (Image Segmentation and Edge detection) การหาขอบภาพ เราจะใช้หลักการหาความชันของความเข้มสี หรือ intensity เนื่องจากบริเวณขอบภาพจะมีความชันของความเข้มสีมาก
- 7) การบีบอัดภาพ (Image Compression)
  - การบีบอัดแบบไม่มีการสูญเสียรายละเอียดข้อมูล (Lossless compression) ค่าความสว่างของแต่ละจุดภาพจะยังคงอยู่เหมือนเดิมทุกประการหรือไม่มีการเปลี่ยนแปลงค่าของแต่ละจุดซึ่งวิธีนี้จะอาศัยเทคนิคการจัดเก็บข้อมูลเชิงเลขในการลดขนาดของข้อมูล
  - การบีบอัดแบบสูญเสียรายละเอียดข้อมูล (Lossy compression) วิธีการนี้ จะมีการเปลี่ยนแปลงค่าความสว่างของจุดภาพหรือมีการเปลี่ยนแปลงค่าของแต่ละจุดซึ่งวิธีการนี้ไม่เหมาะสมสำหรับข้อมูลภาพที่ต้องมีการจำแนกข้อมูล (Classification)

- 8) การสร้างภาพ 3 มิติ (3D Image Reconstruction) การวิเคราะห์ภาพโดยใช้ภาพ 3 มิติ กำลังได้รับความต้องการอย่างมากในปัจจุบัน เนื่องจากภาพ 3 มิติ สามารถแสดง ให้เห็นถึงภาพรวมหรือรายละเอียดในมุมมองต่าง ๆ

## 2.4 ระบบคอมพิวเตอร์วิชั่น

Computer Vision (CV) เป็นหมวดหมู่ย่อยของปัญญาประดิษฐ์ (AI) ที่เน้นการสร้างและการใช้ระบบดิจิทัลในการประมวลผล เป้าหมายของการมองเห็นด้วยคอมพิวเตอร์คือการทำให้อุปกรณ์คอมพิวเตอร์สามารถระบุวัตถุหรือบุคคลในภาพดิจิทัลได้อย่างถูกต้องและดำเนินการตามความเหมาะสม โดยใช้หลักการโครงข่ายประสาทเทียม (Convolutional Neural Network :CNN) เพื่อใช้ในการประมวลผลข้อมูลภาพระดับพิกเซลและ Deep Learning Recurrent Neural Network (RNN) เพื่อทำความเข้าใจว่าแต่ละพิกเซลมีความเกี่ยวข้องกันอย่างไร

รูปแบบของเทคโนโลยีคอมพิวเตอร์วิชั่นและการทำงานในสถานการณ์มีดังนี้

- 1) Object classification เป็นการแยกวิเคราะห์เนื้อหาที่เป็นภาพและจัดประเภทวัตถุดิบภาพถ่ายและวิดีโอเป็นหมวดหมู่ตัวอย่างเช่น ระบบสามารถค้นหาแมวจากวัตถุทั้งหมดที่มีในภาพ
- 2) Object identification เป็นระบบจะแยกวิเคราะห์เนื้อหาที่เป็นภาพและระบุวัตถุเฉพาะบนภาพถ่ายและวิดีโอตัวอย่างเช่น ระบบสามารถค้นหาแมวเปอร์เซีย จากในภาพที่มีแมวแต่ละสายพันธุ์
- 3) Object tracking เป็นระบบประมวลผลวิดีโอคอนเทนต์ (หรือวัตถุ) ที่ตรงกับเกณฑ์การค้นหาและติดตามการเคลื่อนไหวของวัตถุที่ต้องการ

### 2.4.1 หลักการทำงานของระบบคอมพิวเตอร์วิชั่น

คอมพิวเตอร์วิชั่นต้องการข้อมูลจำนวนมากจะวิเคราะห์ข้อมูลจนจดจำภาพได้ในที่สุด ตัวอย่างเช่น ในการฝึกคอมพิวเตอร์ให้รู้จักป้ายเลี้ยวขวา จะต้องป้อนรูปภาพป้ายเลี้ยวขวาและสิ่งของที่เกี่ยวข้องกับป้ายเลี้ยวจำนวนมากเพื่อเรียนรู้ความแตกต่างและจดจำป้ายเลี้ยวขวา เทคโนโลยีที่จำเป็นสองอย่างถูกนำมาใช้เพื่อให้บรรลุสิ่งนี้คือประเภทของการเรียนรู้ของเครื่องที่เรียกว่าการเรียนรู้เชิงลึก (deep learning) และโครงข่ายประสาทเทียม (CNN)

การเรียนรู้ของเครื่อง (Machine Learning) ใช้โมเดลอัลกอริทึมที่ช่วยให้คอมพิวเตอร์สามารถสอนตัวเองเกี่ยวกับข้อมูลภาพได้ หากมีข้อมูลเพียงพอคอมพิวเตอร์จะดู จัดจำข้อมูล และสอนตัวเองให้บอกภาพที่ต้องการจากภาพทั้งหมด โดยอัลกอริทึมช่วยให้เครื่องเรียนรู้ด้วยตัวเอง ไม่ต้องใช้มนุษย์มาใช้โปรแกรมคำสั่ง

เครือข่ายประสาทเทียม (CNN) ช่วยให้การเรียนรู้ของเครื่อง (Machine Learning) เรื่องของการดูการจดจำ โดยแบ่งรูปภาพออกเป็นพิกเซลที่ได้รับป้ายกำกับ (Label) เพื่อทำการปิด (การดำเนินการทางคณิตศาสตร์ของสองฟังก์ชันเพื่อสร้างฟังก์ชันที่สาม) และคาดการณ์เกี่ยวกับสิ่งที่เห็น เครือข่ายประสาทเทียม (CNN) จะทำการปิดเขียวและตรวจสอบความถูกต้องของการคาดคะเนเป็นชุดๆ จนกว่าการคาดการณ์จะเริ่มเป็นจริง จากนั้นจะรับรู้หรือเห็นภาพในลักษณะที่คล้ายกับมนุษย์ โครงข่ายประสาทเทียม (CNN) จะมองเห็นขอบที่แข็งและรูปร่างที่เรียบง่ายก่อน จากนั้นจึงกรอกข้อมูลในขณะที่ทำวิเคราะห์คาดการณ์ซ้ำๆ เพื่อทำความเข้าใจภาพเดี่ยว Deep Learning Recurrent Neural Network (RNN) ถูกใช้ในลักษณะเดียวกันแต่ใช้สำหรับแอปพลิเคชันวิดีโอ เพื่อช่วยให้คอมพิวเตอร์เข้าใจว่ารูปภาพในชุดเฟรมมีความสัมพันธ์กันอย่างไร

#### 2.4.2 การเรียนรู้เชิงจักรกล

การเรียนรู้เชิงจักรกลเป็นสาขาหนึ่งของปัญญาประดิษฐ์ (AI) ที่เน้นการใช้ข้อมูลและ อัลกอริทึมเพื่อเลียนแบบวิธีที่มนุษย์เรียนรู้ โดยค่อยๆ ปรับปรุงความแม่นยำซึ่งจะแตกต่างจากคอมพิวเตอร์ที่ป้อนข้อมูลเข้าไป เพื่อให้คอมพิวเตอร์ประมวลผลและแสดงผลลัพธ์ออกมา แต่การเรียนรู้เชิงกลนั้นจะเป็นการใส่ข้อมูลหรือผลลัพธ์เข้าไป เพื่อให้คอมพิวเตอร์ได้ทำการเรียนรู้ผลลัพธ์หรือข้อมูลนั้นและทำการประมวลผลโดยจะวิเคราะห์คาดการณ์ผลว่าเป็นแบบที่คาดการณ์ไว้หรือไม่

#### 2.4.3 การทำงานของ การเรียนรู้เชิงจักรกล

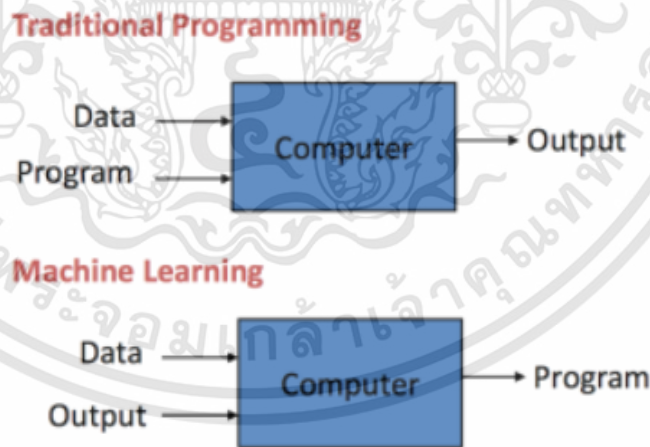
การทำงานของ การเรียนรู้เชิงจักรกล (Machine Learning) จะแบ่งการทำงานของอัลกอริทึมเป็น 3 ส่วนได้แก่

- 1) กระบวนการตัดสินใจ คือ โดยทั่วไปอัลกอริทึม Machine Learning ใช้เพื่อทำนายและจำแนกประเภทขึ้นอยู่กับข้อมูลอินพุตบางส่วนซึ่งสามารถติดป้ายกำกับ (Label) หรือไม่ติดป้ายกำกับอัลกอริทึมจะสร้างค่าประมาณเกี่ยวกับรูปแบบในข้อมูล
- 2) ฟังก์ชันข้อผิดพลาด คือ ฟังก์ชันข้อผิดพลาดใช้เพื่อประเมินการคาดการณ์ของแบบจำลอง หากฟังก์ชันเกิดข้อผิดพลาดก็สามารถเปรียบเทียบเพื่อประเมินความถูกต้องของแบบจำลองได้

- 3) กระบวนการเพิ่มประสิทธิภาพแบบจำลอง คือ หากแบบจำลองสามารถเข้ากับจุดข้อมูลในชุดการสอนได้ดีกว่าข้อมูลจะถูกปรับเพื่อลดความคลาดเคลื่อนระหว่างตัวอย่างที่ทราบ และการประมาณการแบบจำลองอัลกอริทึมจะทำการประเมินและวิเคราะห์ซ้ำจนปรับให้เหมาะสมโดยจะอัปเดตข้อมูลโดยอัตโนมัติจนกว่าจะถึงเกณฑ์ความแม่นยำ

## 2.5 การเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง (Machine Learning) คือ ส่วนการเรียนรู้ของเครื่อง ถูกใช้งานเสมือนเป็นสมองของ AI (Artificial Intelligence) ใช้ Machine Learning ในการสร้างความฉลาด ไม่ได้เกิดจากการเขียนโดยใช้มนุษย์ มนุษย์มีหน้าที่เขียนโปรแกรมให้ AI เรียนรู้จากข้อมูล ส่วน Machine Learning เรียนรู้จากสิ่งที่เราส่งเข้าไปกระตุ้น แล้วจดจำเอาไว้เป็นมันสมอง ส่งผลลัพธ์ออกมาเป็นตัวเลขหรือ code ที่ส่งต่อไปแสดงผล การนำ AI ไปแสดงการกระทำ Machine ต้องอาศัยกลไกที่เป็นโปรแกรมหรือเรียกว่า Algorithm ที่มีหลากหลายแบบ โดยมี Data Scientist เป็นผู้ออกแบบ หนึ่งในกลไกที่เป็นโปรแกรม ที่ได้รับความนิยมสูง คือ Deep Learning ซึ่งถูกออกแบบมาให้ใช้งานได้ง่าย อย่างไรก็ตามใน การทำงานจริง Data Scientist จำเป็นต้องออกแบบตัวแปรต่างๆ ทั้งในตัวของ Deep Learning เอง และต้องหา Algorithm อื่นๆ มาเป็นคู่เปรียบเทียบ เพื่อมองหา Algorithm ที่เหมาะสมที่สุดในการใช้งานจริง โดยแบ่งได้ 3 ประเภท ดังนี้

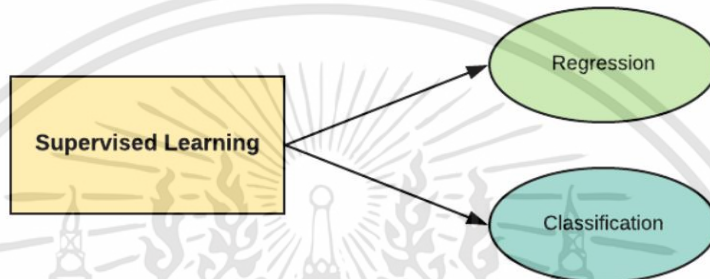


<https://www.slideshare.net/LouisaDiggs/imed-2018-an-intro-to-remote-sensing-and-machine-learning> , [Online]

ภาพที่ 2.3 การทำงานของ Machine Learning

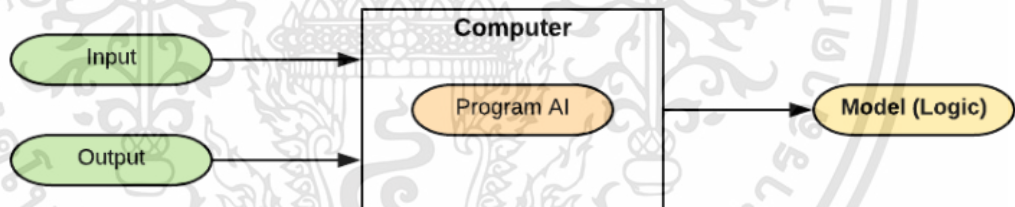
1) Supervised learning

การเรียนรู้โดยมีผู้สอน (Supervised Learning) เป็นการเรียนรู้ที่จะต้องอาศัยข้อมูลในการสอน โดยจำเป็นจะต้องอาศัยชุดข้อมูลต่าง ๆ ซึ่งประกอบไปด้วยชุดของข้อมูลและชุดผลลัพธ์ของข้อมูลที่ต้องการ โดยผลที่ได้จากการเรียนรู้คือ Machine Learning สามารถคาดคะเนผลลัพธ์ที่เกิดขึ้นจากการได้รับข้อมูล



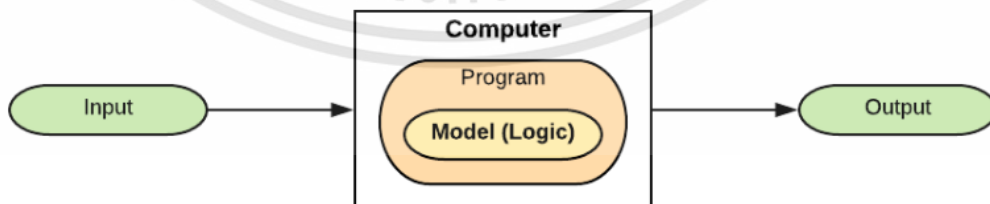
<https://www.slideshare.net/LouisaDiggs/imed-2018-an-intro-to-remote-sensing-and-machine-learning> , [Online]

ภาพที่ 2.4 หลักการประยุกต์ใช้แก้ปัญหาได้ 2 รูปแบบของ Supervised Learning



<https://www.cp.eng.chula.ac.th/> , [Online]

ภาพที่ 2.5 กระบวนการสอนเพื่อให้ได้โมเดลที่ต้องการ



<https://www.cp.eng.chula.ac.th/> , [Online]

ภาพที่ 2.6 โมเดลที่ประยุกต์ใช้กับโปรแกรมของคอมพิวเตอร์

## 2) Unsupervised Learning

การเรียนรู้โดยไม่มีผู้สอน (Unsupervised Learning) เป็นการเรียนรู้ที่ให้เครื่องจักรนั้นสามารถเรียนรู้ได้ด้วยตนเอง ซึ่งวิธีการคือมนุษย์จะเป็นผู้ใส่ข้อมูลต่าง ๆ และกำหนดสิ่งที่ต้องการจากข้อมูลเหล่านั้น โดยให้เครื่องจักรวิเคราะห์จากการจำแนกและสร้างแบบแผนจากข้อมูลที่ได้รับมา

## 3) Reinforcement Learning

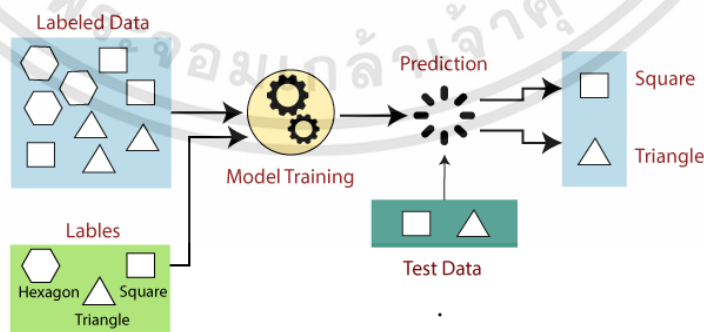
การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) เป็นการเรียนรู้สิ่งต่าง ๆ จากการลองผิดลองถูก ภายใต้แนวคิดที่ว่าจะเลือกกระทำสิ่งๆหนึ่งที่ทำให้ได้ผลลัพธ์มากที่สุด โดยทำการเรียนรู้จากการลองผิดลองถูกในสถานการณ์ที่ผ่านมาจนกระทั่งจะพัฒนาระบบการตัดสินใจของตัวเองให้ดีขึ้นเรื่อยๆ โดยที่อาจจะพัฒนาด้วยการพยายามสร้างแบบจำลองสถานการณ์ต่าง ๆ

## 2.6 Supervised learning

เป็นการเรียนรู้ข้อมูลต่างๆโดยมีผู้สอนอาศัยข้อมูลในการฝึกฝน เพื่อให้ตัวเทคโนโลยีสามารถเรียนรู้ผลและคาดคะเนผลลัพธ์ต่างๆ ได้อย่างแม่นยำมากขึ้น

### 2.6.1 กระบวนการทำงานของ Supervised learning

ในการเรียนรู้ Supervised learning โมเดลจะได้รับการสอนโดยใช้ชุดข้อมูลที่มีป้ายกำกับ(label) ซึ่งโมเดลจะเรียนรู้เกี่ยวกับข้อมูลแต่ละประเภท เมื่อกระบวนการสอนเสร็จสิ้นแบบจำลองจะได้รับการทดสอบตามข้อมูลการทดสอบ (ชุดย่อยของชุดการสอน) จากนั้นจึงคาดการณ์ผลลัพธ์



<https://medium.com/@aisika43/supervised-learning-9b001306e804> , [Online]

ภาพที่ 2.7 การทำงานของ Supervised learning

จากภาพที่ 2.7 สมมติว่ามีชุดข้อมูลของรูปทรงประเภทต่าง ๆ ขั้นตอนแรกคือ เราต้องฝึกสอนโมเดลสำหรับแต่ละรูปร่าง

หากรูปร่างที่กำหนดมีสี่ด้าน และทุกด้านเท่ากัน จะถูกระบุว่าเป็นรูปสี่เหลี่ยมจัตุรัส

หากรูปร่างที่กำหนดมีสามด้าน รูปร่างนั้นจะถูกระบุว่าเป็นรูปสามเหลี่ยม

หากรูปร่างที่กำหนดมีด้านเท่ากันหกด้าน จะถูกระบุว่าเป็นรูปหกเหลี่ยม

หลังจากการสอนจะทำการทดสอบแบบจำลองของโดยใช้ชุดทดสอบ และงานของแบบจำลองคือการระบุรูปร่าง เครื่องจักรได้รับการสอนเกี่ยวกับรูปร่างทุกประเภทแล้ว และเมื่อพบรูปร่างใหม่ จะจัดประเภทรูปร่างบนฐานของหลายด้าน และคาดการณ์ผลลัพธ์

### 2.6.2 ขั้นตอนของกระบวนการ Supervised learning

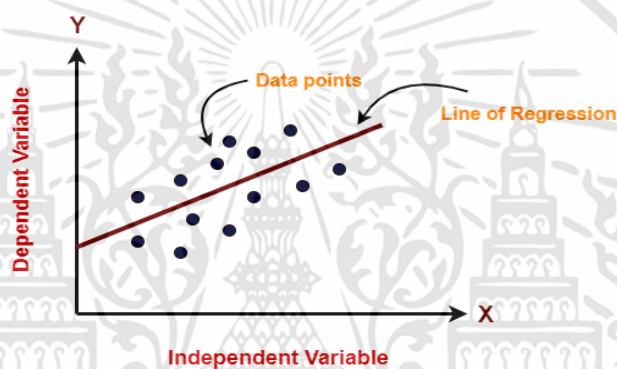
- 1) กำหนดประเภทของชุดข้อมูลการสอน (training dataset)
- 2) รวบรวมข้อมูลการฝึกสอนที่มีป้ายกำกับ (label)
- 3) แยกชุดข้อมูลออกเป็นชุดข้อมูลการสอน (dataset) ชุดข้อมูลทดสอบ (test dataset) และชุดข้อมูลการตรวจสอบ (validation dataset)
- 4) กำหนดคุณสมบัติผลลัพธ์ (Input) ของชุดข้อมูลการสอน (training dataset) ซึ่งควรมีอย่างเพียงพอเพื่อให้แบบจำลองสามารถทำนายผลลัพธ์ได้อย่างแม่นยำ
- 5) กำหนดอัลกอริทึมที่เหมาะสมสำหรับแบบจำลอง
- 6) ดำเนินการอัลกอริทึมในชุดข้อมูลการสอน (training dataset) บางครั้งก็ต้องการชุดการตรวจสอบความถูกต้องเป็นพารามิเตอร์ควบคุม ซึ่งเป็นชุดย่อยของชุดข้อมูลการสอน (training dataset)
- 7) ประเมินความถูกต้องของแบบจำลองโดยจัดเตรียมชุดทดสอบ หากแบบจำลองคาดการณ์ผลลัพธ์ที่ถูกต้อง ซึ่งหมายความว่าแบบจำลองนั้นมีความแม่นยำ

## 2.7 ประเภทของกระบวนการ Supervised learning

การเรียนรู้แบบ Supervised learning แบ่งออกเป็น 2 ประเภท คือ

**2.7.1 Regression** ในการเรียนรู้ของเครื่องจะเรียนรู้ความสัมพันธ์ภายในข้อมูลที่ให้มา และทำการคาดคะเนตามรูปแบบที่ระบุจากชุดข้อมูลนั้นๆ ดังนั้น Regression เป็นเทคนิคการเรียนรู้ของเครื่อง โดยที่แบบจำลองคาดการณ์ผลลัพธ์เป็นค่าตัวเลขต่อเนื่อง

**2.7.1.1 Linear Regression** เป็นอัลกอริธึม ML ที่ใช้สำหรับ Supervised learning และ Linear Regression โดยทำนายตัวเป้าหมายตามตัวแปรอิสระที่กำหนด ดังนั้น เทคนิคนี้ จะมีความสัมพันธ์เชิงเส้นตรงระหว่างตัวแปรตามและตัวแปรอิสระอื่นๆ



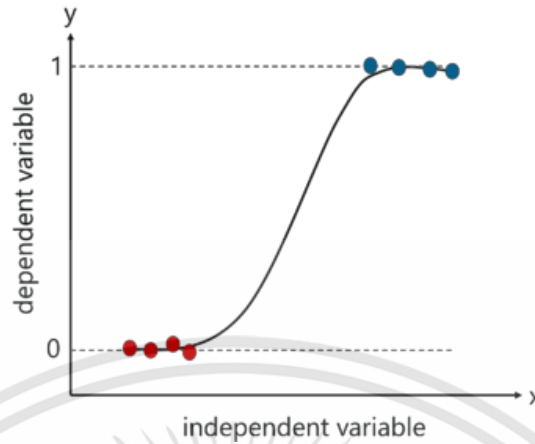
<https://medium.datadriveninvestor.com/linear-regression-model-from-scratch-59a89e96c2b3> , [Online]

ภาพที่ 2.8 กราฟแสดงอัลกอริธึมของ Linear Regression

จากภาพที่ 2.8 บนแกน X คือตัวแปรอิสระและบนแกน Y คือเอาต์พุต Linear Regression เป็นเส้นที่พอดีที่สุดสำหรับแบบจำลอง และวัตถุประสงค์หลักของเราในอัลกอริธึมนี้คือการหาเส้นที่เหมาะสมและต่อเนื่องที่สุด

### 2.7.1.2 Logistic Regression

Logistic Regression หรือเรียกอีกอย่างว่า Loss Function อัลกอริธึมนี้จะคาดการณ์ค่าที่ไม่ต่อเนื่องสำหรับชุดของตัวแปรอิสระ ที่ส่งผ่านไปยังตัวอัลกอริธึมหลังจากนั้นก็ทำการทำนายโดยการจับคู่ข้อมูล ที่ไม่อยู่ในฐานข้อมูลไปยังฟังก์ชัน logit ที่ได้รับการตั้งโปรแกรมไว้ อัลกอริธึมจะคาดการณ์ความน่าจะเป็นของข้อมูลใหม่ ซึ่งผลลัพธ์จะอยู่ระหว่างช่วง 0 ถึง 1



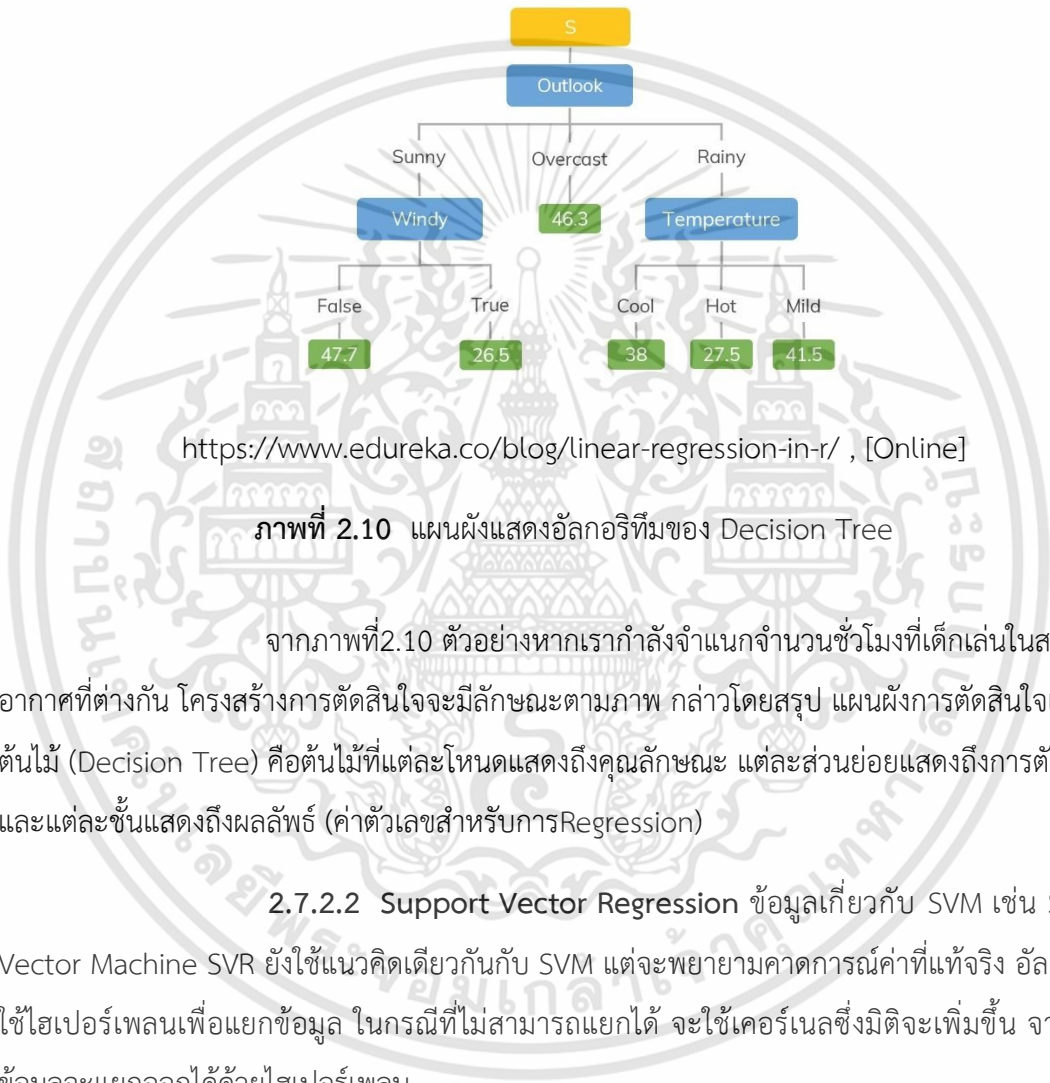
<https://www.edureka.co/blog/linear-regression-in-r/> , [Online]

ภาพที่ 2.9 กราฟแสดงอัลกอริทึมของ Logistic Regression

### 2.7.2 Classification

เป็นการเรียนรู้ประเภทหนึ่งที่อัลกอริทึมจำเป็นต้องจัดข้อมูลใหม่ที่ได้รับกับประเภท คลาสหรือประเภทนั้นจำเป็นต้องจับคู่กับ 1 หรือ 0 ผลลัพธ์จะเป็นหนึ่งในคลาสและไม่ใช่ตัวเลขเหมือนในการ regression อัลกอริทึมโดยหลักการทำงานของมันเป็นการสอนโมเดลเพื่อจัดประเภทข้อมูลของ Machine Learning โดยจะใช้ตัวแยกประเภทต่างๆ เลือกคุณสมบัติ เพื่อฝึกสอนโมเดลและประเมินผลลัพธ์ เพื่อค้นหาประเภทโมเดลการจำแนกที่ดีที่สุด โดยจะมีการจัดหาชุดข้อมูลอินพุตที่รู้จักจากตัวอย่าง และผลลัพธ์สำหรับข้อมูล เช่นป้ายกำกับ (Label) หรือคลาส ที่ใช้ข้อมูลเพื่อฝึกโมเดลที่สร้างการคาดการณ์สำหรับผลลัพธ์สำหรับข้อมูลใหม่ วิธีการจำแนกประเภทข้อมูล จะทำจากการสอนข้อมูลแล้วนำไปตรวจสอบและปรับแต่งโมเดลเพื่อแยกประเภทข้อมูลระหว่างอัลกอริทึมต่างๆ หลังจากนั้นนำมาเปรียบเทียบกับข้อผิดพลาดในการตรวจสอบและทำการเลือกโมเดลที่ดีที่สุด เพื่อนำไปใช้วิเคราะห์ในขั้นตอนต่อไป

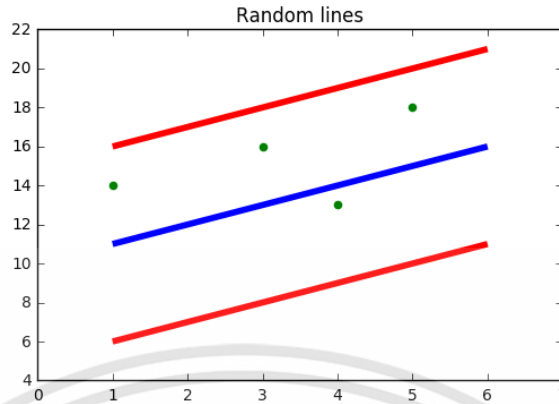
2.7.2.1 Decision Tree แบบจำลองแผนผังการตัดสินใจแบบรูปต้นไม้สามารถใช้ได้กับข้อมูลที่มีคุณลักษณะเชิงตัวเลขและลักษณะการจัดหมวดหมู่ แผนผังการตัดสินใจนั้นการจับภาพที่ไม่เป็นเชิงเส้นระหว่างจุดสนใจและตัวแปรเป้าหมายได้ดี แผนผังการตัดสินใจแบบรูปต้นไม้ (Decision Tree) ค่อนข้างจะตรงกับความคิดในระดับมนุษย์ ดังนั้นจึงมีความสามารถในการทำความเข้าใจข้อมูลได้ดี



ภาพที่ 2.10 แผนผังแสดงอัลกอริทึมของ Decision Tree

จากภาพที่ 2.10 ตัวอย่างหากเรากำหนดจำนวนชั่วโมงที่เด็กเล่นในสภาพอากาศที่ต่างกัน โครงสร้างการตัดสินใจจะมีลักษณะตามภาพ กล่าวโดยสรุป แผนผังการตัดสินใจแบบรูปต้นไม้ (Decision Tree) คือต้นไม้ที่แต่ละโหนดแสดงถึงคุณลักษณะ แต่ละส่วนย่อยแสดงถึงการตัดสินใจ และแต่ละชั้นแสดงถึงผลลัพธ์ (ค่าตัวเลขสำหรับการ Regression)

2.7.2.2 Support Vector Regression ข้อมูลเกี่ยวกับ SVM เช่น Support Vector Machine SVR ยังใช้แนวคิดเดียวกันกับ SVM แต่จะพยายามคาดการณ์ค่าที่แท้จริง อัลกอริทึมนี้ใช้ไฮเปอร์เพลนเพื่อแยกข้อมูล ในกรณีที่ไม่สามารถแยกได้ จะใช้เคอร์เนลซึ่งมิติจะเพิ่มขึ้น จากนั้นจุดข้อมูลจะแยกออกได้ด้วยไฮเปอร์เพลน



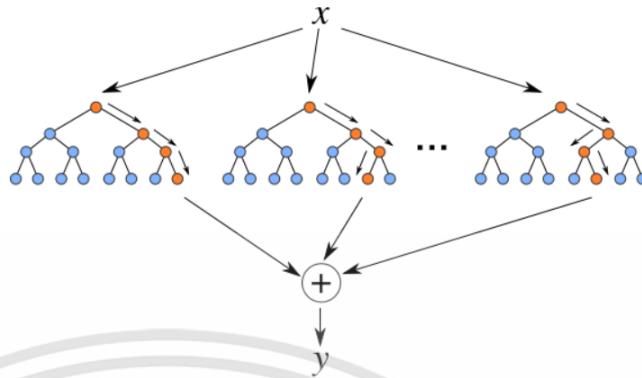
<https://www.edureka.co/blog/linear-regression-in-r/> , [Online]

ภาพที่ 2.11 กราฟแสดงอัลกอริทึมของ Support Vector Regression

จากภาพที่ 2.11 เส้นสีน้ำเงินคือ Hyper Plane; เส้นสีแดงคือเส้น Boundary จากจุดข้อมูลทั้งหมดอยู่ภายในเส้น Boundary (เส้นสีแดง) วัตถุประสงค์หลักของ SVR คือการพิจารณาจุดที่อยู่ภายในเส้น Boundary โดยพื้นฐาน

### 2.7.2.3 Random Forest Regressor

Random Forest Regressor เป็นกลุ่มเกิดจากการรวมกันของแผนผังการตัดสินใจแบบรูปต้นไม้ (Decision Tree) ใช้สำหรับการจำแนกประเภทและการถดถอย (Regression) ข้อมูลที่ป้อนเข้าจะถูกส่งผ่านแผนผังการตัดสินใจหลายรายการ จะดำเนินการโดยการสร้างแผนภูมิการตัดสินใจจำนวนที่แตกต่างกันในเวลาการสอนและแสดงคลาสที่เป็นโหมดของคลาสสำหรับการจำแนกประเภทหรือการทำนายเฉลี่ย สำหรับการถดถอย (Regression) ของแต่ละต้นไม้ (Tree)



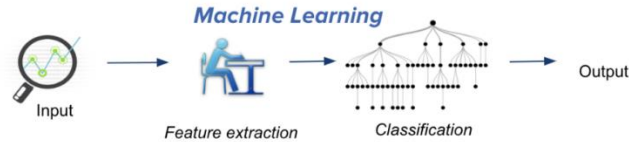
<https://medium.datadriveninvestor.com/random-forest-pros-and-cons> , [Online]

ภาพที่ 2.12 แผนภาพแสดงอัลกอริทึมของ Random Forest Regressor

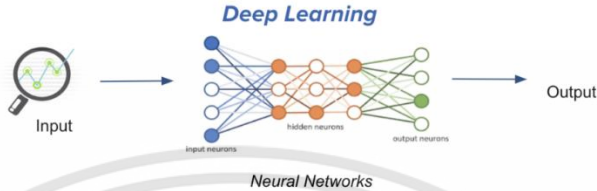
## 2.8 การเรียนรู้เชิงลึก Deep Learning

การเรียนรู้เชิงลึก (Deep Learning) คือวิธีการเรียนรู้แบบอัตโนมัติด้วยการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Neurons) โดยนำระบบโครงข่ายประสาทเทียม (Neural Network) มาซ้อนกัน หลายชั้น (Layer) และทำการเรียนรู้ข้อมูลตัวอย่าง ซึ่งข้อมูลดังกล่าวจะถูกนำไปใช้ในการตรวจจับรูปแบบ (Pattern) หรือจำแนกข้อมูล (Classify the Data) โดยทั่วไปวิธีโครงข่ายประสาทเทียมจะเรียนรู้ได้เพียงไม่กี่ชั้น เพื่อที่จะทำให้ Neural Network ของเรานั้นสามารถคิดและประมวลผลซับซ้อนได้เหมือนสมองมนุษย์ ชั้นที่เป็น Hidden Layer จึงต้องมีหลายๆชั้น เพื่อให้ส่งข้อมูลประมวลผลต่อกัน ทำให้คำนวณสิ่งที่ซับซ้อนได้มากขึ้น เนื่องจากยังไม่มีข้อมูลสอน (Training Data) ความซ้อนกันหลายๆชั้นโครงข่ายยังมีมากจะยิ่งลึกขึ้นจึงเป็นที่มาของ คำว่าการเรียนรู้เชิงลึก หรือ “Deep Learning”

(ภาพที่ 2.13) เมื่อมีข้อมูลดิบเข้ามาจะไม่มีประมวลโดยอัตโนมัติ แต่จะต้องอาศัยความรู้เฉพาะทาง (Domain Knowledge) สำหรับคุณลักษณะในการจัดหมวดหมู่ข้อมูลบางประเภท (Hand-Craft Features)



Traditional machine learning uses hand-crafted features, which is tedious and costly to develop.



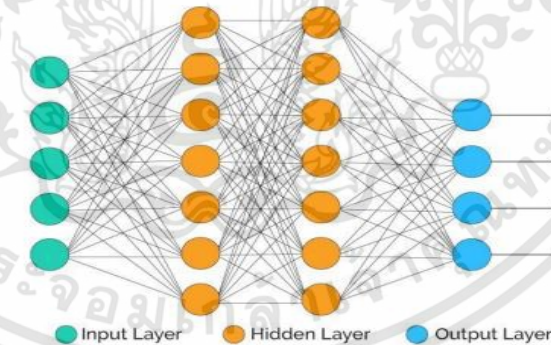
Deep learning learns hierarchical representation from the data itself, and scales with more data.

<https://www.merkleinc.com/blog/dispelling-myths-deep-learning-vs-machine-learning> ,  
[Online]

ภาพที่ 2.13 ความแตกต่างระหว่างการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก

### 2.8.1 โครงข่ายประสาทเทียมอัจฉริยะ

วิธีการสร้างแบบจำลองข้อมูลหรือโมเดลข้อมูล (Data modeling) โดยวิธีโครงข่ายประสาทเทียมอัจฉริยะ (Artificial Neuron Network: ANN) เป็นวิธีการที่มีพื้นฐานมาจากการเลียนแบบการทำงานของสมองมนุษย์ซึ่งประกอบด้วยนิวรอน (Neurons) ที่ใช้ในการเชื่อมโยงเพื่อแก้ปัญหาต่างๆ



<https://www.mindphp.com/-ai-machine-learning/5659-artificial-neural-networks.html> ,  
[Online]

ภาพที่ 2.14 ส่วนประกอบของโครงข่ายประสาทเทียม

จากภาพที่ 2.14 ส่วนประกอบของโครงข่ายประสาทเทียม ส่วนประกอบของโครงข่ายประสาทเทียม จะเห็นว่าส่วนประกอบของ โครงข่ายประสาทเทียมประกอบด้วย 3 ส่วน ดังนี้



## 2.8.2 ฟังก์ชันกระตุ้น

ฟังก์ชันกระตุ้นหรือเรียกอีกชื่อว่า “ฟังก์ชันการส่งต่อ (Transfer function)” เป็นฟังก์ชันในการคำนวณ เพื่อทำนายค่าของข้อมูลออก รูปแบบส่วนมากจะแบบที่ไม่เป็นฟังก์ชันสมการเส้นตรง (Non-linear function) เนื่องจากปัญหาความเป็นจริงมีลักษณะเป็นแบบสมการเส้นตรงน้อยมาก ฟังก์ชันกระตุ้นทำหน้าที่ในการตัดสินใจว่านิเวศจะถูกระตุ้นหรือไม่ โดยดูค่าผลรวมของข้อมูลเข้าและค่าน้ำหนัก ฟังก์ชันกระตุ้นจะถูกนำไปใช้ทั้งโหนดซ่อน (Hidden node) และโหนดข้อมูลออก (Output node) ซึ่งทั้งสองโหนดอาจใช้ฟังก์ชันกระตุ้นที่เหมือนหรือต่างกันได้ แต่ส่วนมากจะใช้ฟังก์ชันแบบไม่เป็นเชิงเส้น เนื่องจากในโหนดซ่อนจะมีการคำนวณแบบการรวมเชิงเส้น (Linear combination) ถ้าฟังก์ชันกระตุ้นของโหนดซ่อนจะมีการคำนวณแบบเชิงเส้นอีก จะเป็นการทำงานซ้ำซ้อนกับ การคำนวณแบบการรวมเชิงเส้นในชั้นข้อมูลออกและจะทำให้ผลลัพธ์เทียบเท่ากับสมการถดถอยลอสติก ฟังก์ชันกระตุ้นจะมีหลากหลายรูปแบบ ดังต่อไปนี้

### 2.8.2.1 ฟังก์ชันกระตุ้นค่าแบ่ง (Threshold Activation Function)

ฟังก์ชันนี้จะพิจารณาค่าข้อมูลเข้าว่ามากกว่าหรือน้อยกว่าค่าแบ่งที่กำหนดไว้ (threshold) หรือไม่ เพื่อส่งค่าต่อไปยังขั้นถัดไป

### 2.8.2.2 ฟังก์ชันกระตุ้นซิกมอยด์ (Sigmoid Activation Function)

เป็นฟังก์ชันทางคณิตศาสตร์ที่มี ลักษณะเป็นตัวเอส “S-curve” หรือเรียกว่า “Sigmoid curve” จะมีค่าระหว่าง 0 และ 1 ฟังก์ชันนี้จะใช้เมื่อ เพื่อทำนายความน่าจะเป็น (Probability) ของข้อมูลออก

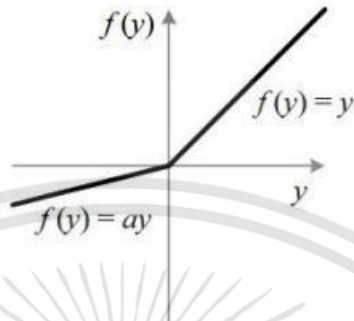
### 2.8.2.3 ฟังก์ชันไฮเปอร์โบลิกแทงก์เจนท์ (Hyperbolic Tangent Function: tanh)

มีการทำงานคล้ายฟังก์ชันกระตุ้นซิกมอยด์ แต่มีประสิทธิภาพดีกว่า จะมีค่าระหว่าง [-1, 1] ข้อดีของฟังก์ชันไฮเปอร์โบลิกแทงก์เจนท์คือ สามารถแปลงค่าข้อมูลเข้าที่มีค่าเป็นลบมาก ๆ ให้เป็นข้อมูลออกที่ติดลบได้ และแปลงให้มีค่าใกล้ศูนย์ (near-zero output)

### 2.8.2.4 ฟังก์ชันเรกติไฟด์ลินีเยยูนิต (Rectified Linear Units, ReLU)

เป็นฟังก์ชันที่นิยมใช้งาน ฟังก์ชันนี้จะมีค่าอยู่ระหว่าง  $[0, \infty)$  หมายถึงถ้าข้อมูลเข้ามีค่ามากกว่าศูนย์ข้อมูลออกไปจะเป็นค่าบวก และถ้าข้อมูลเข้ามีค่าศูนย์หรือติดลบ ข้อมูลออกจะมีค่าเป็นศูนย์ นักวิทยาศาสตร์พิสูจน์แล้วว่าฟังก์ชันเรกติไฟด์ลินีเยยูนิตมีประสิทธิภาพดีกว่าฟังก์ชันไฮเปอร์โบลิกแทงก์เจนท์สูงถึง 6 เท่า ข้อเสียของฟังก์ชันเรกติไฟด์ลินีเยยูนิตคืออาจเกิดปัญหานิวรอนตาย (dead neurons) กล่าวคือ ฟังก์ชันกระตุ้นไม่ทำงานไม่ว่าข้อมูลเข้ามีค่าเท่าไรก็ตาม

จึงมีการพัฒนา ฟังก์ชันเรกติไฟด์ลินเนี่ยยูนิตเป็นฟังก์ชัน “ลีสก์เรกติไฟด์ลินเนี่ยยูนิต (Leaky ReLu)” ซึ่งจะมีค่าจาก  $-\infty$  ถึง  $+\infty$  แสดงดังภาพที่ 2.16



<https://medium.com/mmp-li/deep-learning> , [Online]

ภาพที่ 2.16 ฟังก์ชันลีสก์เรกติไฟด์ลินเนี่ยยูนิต

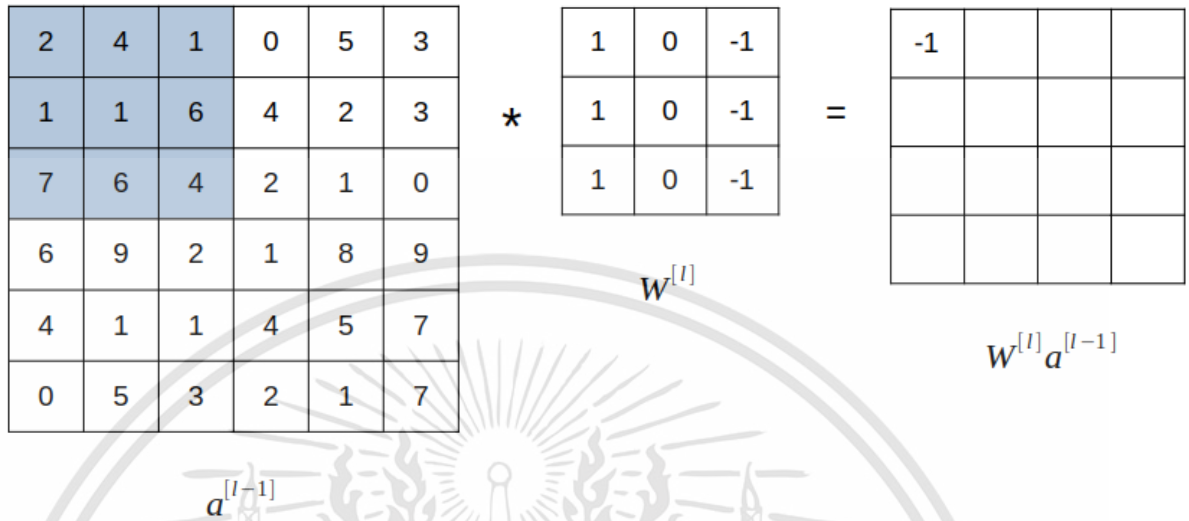
## 2.9 โครงข่ายประสาทเทียมคอนโวลูชัน

โครงข่ายประสาทเทียมคอนโวลูชัน (Convolutional Neural Network: CNN) เป็นวิธีการที่ถูกนำมาใช้อย่างกว้างขวางทั้งในประมวลผลสัญญาณ (Signal processing) รวมถึงการประมวลผลภาพ (Computer vision) คอนโวลูชัน เป็นวิธีการทางคณิตศาสตร์เพื่อศึกษาการเปลี่ยนแปลงของ ฟังก์ชัน (  $f$  ) เมื่อมีฟังก์ชัน (  $g$  ) เข้ามา CNN ถูกนำมาใช้ในรูปแบบโมเดลที่สามารถเรียนรู้และเลือกใช้ลักษณะเด่น (Feature extraction) ของรูปภาพได้ด้วยตัวเอง ข้อดีคือโมเดลจะเรียนรู้และเลือกลักษณะเด่นของภาพได้ดีกว่ามนุษย์ดังนั้นจะทำให้ได้การประมวลผลที่แม่นยำมากกว่า และโครงสร้างของ CNN ประกอบไปด้วยชั้นการทำงานจำนวน 2 ชั้น คือ

### 2.9.1 ชั้นคอนโวลูชัน (Convolutional layer)

Convolution layer จะเป็นการเก็บข้อมูลที่สำคัญ กล่าวคือไม่ได้เก็บทั้งภาพเพื่อไปทำงานในขั้นถัดไป

$$2*1 + 4*0 + 1*-1 + 1*1 + 1*0 + 6*-1 + 7*1 + 6*0 + 4*-1 = -1$$



<https://guopai.github.io/ml-blog19.html> , [Online]

ภาพที่ 2.17 ภาพแสดงชั้นคอนโวลูชัน

จากภาพที่ 2.17 สมมติเรามี Matrix ซ้ายมือ ขนาด 6x6 และมี Matrix ตรงกลาง ซึ่งเรียกว่า Filter หรือ Kernel ขนาด 3x3 เราจะนำเฉพาะ 3x3 ช่องแรกของ Matrix แรก มาคูณแบบ Element-wise กับ Filter matrix แล้วนำผลที่ได้แต่ละค่า (ซึ่งมีทั้งสิ้น 9 ค่า) มาบวกกัน แล้วนำไปใส่ในแถวแรกคอลัมน์แรกของ Matrix ที่สามซึ่งเป็นผลลัพธ์ โดยในภาพ ผลลัพธ์เท่ากับ -1

ถัดมา เราจะเลื่อนกรอบขนาด 3x3 ใน Matrix แรกไปทางขวา 1 ช่อง แล้วทำแบบเดิม ผลลัพธ์ที่ได้ นำไปใส่ในแถว 1 ช่อง 2 ของ Matrix ผลลัพธ์ ทำไปเรื่อยๆ จนสุดทาง แล้วเลื่อนกรอบ 3x3 ลงมาด้านล่าง 1 ช่อง (ขีดขอบด้านซ้ายมือ) แล้วทำแบบเดิม จนกระทั่งเต็มค่าใน Matrix ผลลัพธ์จนเต็ม

กระบวนการนี้ เรียกว่า Convolution ซึ่งแสดงสัญลักษณ์ด้วย ส่วน Neural network ที่มี Layer ที่ใช้กระบวนการ Convolution ควรมีมากกว่า 1 Layer เราก็เรียกว่า Convolutional neural network

ผลลัพธ์ในกระบวนการนี้ Matrix แรกด้านซ้ายมือที่เป็น Input ก็คือ  $a^{[l-1]}$  ส่วน Filter matrix คือ Matrix ของค่าน้ำหนัก  $W^{[l]}$  ซึ่งมีค่าที่ต้องเรียนรู้จากกระบวนการ Backward propagation ผลที่ได้จากการ Convolve 2 Matrix นี้เข้าด้วยกัน คือ Matrix ผลลัพธ์ นั่นคือ  $W^{[l]}a^{[l-1]}$  นั่นเอง จากนั้นเราก็บวก Intercept  $b^{[l]}$  เข้าไป ประกอบกันกลายเป็นผลลัพธ์ สมการเส้นตรง  $Z^{[l]}=W^{[l]}a^{[l-1]}+b^{[l]}$  แล้วก็นำ  $Z^{[l]}$  นี้ไปป้อนเข้า Activation function  $g(Z^{[l]})$  ก็จะได้ผลเป็น Activation output ของ Layer นั้น นั่นก็คือ  $A^{[l]}$

สังเกตว่า Convolution layer นี้ มีคุณสมบัติพิเศษอยู่ 2 ประการ คือ:

ทั้ง Layer ใช้ค่าน้ำหนัก  $W$  ร่วมกัน: กระบวนการ Convolution ใช้ Filter ค่าน้ำหนักตัวเดียวกัน มาคำนวณหา  $W_a$  โดยการ Convolve กับ Input ในตำแหน่ง Grid ต่างๆ หลายๆ ครั้งจนครบ ดังนั้น จำนวนของค่าน้ำหนัก  $w$  ใน Matrix ค่าน้ำหนัก  $W$  จะมีจำนวนน้อยกว่า Neural network แบบทั่วไปมาก ซึ่งนอกจากจะส่งผลให้การคำนวณทำได้เร็วขึ้นมากแล้ว การใช้ค่าน้ำหนักชุดเดิมซ้ำๆ ในหลายๆ ส่วนของภาพจะเป็นการตรวจจับคุณลักษณะบางประการในหลายๆ ส่วนของภาพ ตัวอย่างเช่น หาก Filter matrix มีแถวซ้ายเป็นค่าบวก แถวขวาเป็นค่าลบ ดังนี้:

$$f = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \quad (1)$$

ถ้าเรานำ Filter นี้ไป Convolve กับ Input matrix ที่ด้านซ้ายกับขวาสว่างไม่เท่ากัน เช่น:

$$A = \begin{pmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

จะส่งผลให้ Linear function ผลลัพธ์ มีค่ามากตรงแถวกลางๆ ดังนี้:

$$A * f = \begin{pmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{pmatrix} \quad (3)$$

ค่ามากหมายถึงมีความสว่างมาก นั่นแปลว่า Filter นี้ใช้ Detect ส่วนของภาพที่เป็นเส้นขอบแนวตั้งนั่นเอง

การเชื่อมต่อข้าม Layer แบบ Sparse: ค่าของ Output ในแต่ละ Layer เป็นผลมาจากการคำนวณค่า Input เพียงบางส่วน ซึ่งต่างกับ Neural network มีการเชื่อมต่อแบบ Dense หมายถึงค่าของ Output แต่ละ Neuron มาจากการคำนวณ Input ทุก Neuron การเชื่อมต่อแบบ Sparse ส่งผลให้การคำนวณนั้นเกิดขึ้นเร็วขึ้นมาก

### 2.9.1.1 Padding

กระบวนการ Convolution ยังมีแนวโน้มที่จะทำให้ข้อมูลที่อยู่ตามขอบของภาพนั้นไม่ถูกนำไปคำนวณอย่างเต็มที่เหมือนข้อมูลที่อยู่ส่วนกลางของภาพ เพราะ Filter มีโอกาสจับข้อมูลตามขอบภาพน้อยกว่าตรงกลางภาพ ทางออกต่อปัญหา 2 ข้อนี้ คือการทำ Padding ซึ่งก็คือการขยายขอบของข้อมูล Input ออกทุกด้านเท่าๆ กัน เช่น ถ้า Input มีขนาด 6x6 การทำ Padding  $p=1$  จะทำให้ Input มีขนาดเป็น 8x8 เราสามารถคำนวณได้ว่าเมื่อทำ Padding แล้ว มิติของ Output จะเป็นเท่าใด แต่ก่อนอื่น ลองเปรียบเทียบกับแบบธรรมดา

"Valid" convolution คือไม่มี Padding มิติของ Output คือ:

$$(n-f+1), (n-f+1) \quad (4)$$

โดย  $n$  คือมิติของ Input ส่วน  $f$  คือมิติของ Filter

"Same" convolution คือการ Pad เพื่อให้มิติของ Output เท่ากับมิติของ Input มิติของ Output คือ:

$$(n+2p-f+1), (n+2p-f+1) \quad (5)$$

$$p=f-12 \quad (6)$$

โดย  $p$  คือขนาด Pixel ของ Padding ในแต่ละด้าน

ตัวอย่างเช่น ถ้า Filter มีขนาด 5x5 เราจะต้อง Pad เป็นจำนวนดังนี้:

$$p=5-12=2 \text{ pixel} \quad (7)$$

อนึ่ง  $f$  ควรจะเป็นเลขคู่ เพื่อให้เกิดความสมมาตรและมีจุดกึ่งกลาง

### 2.9.1.2 Stride

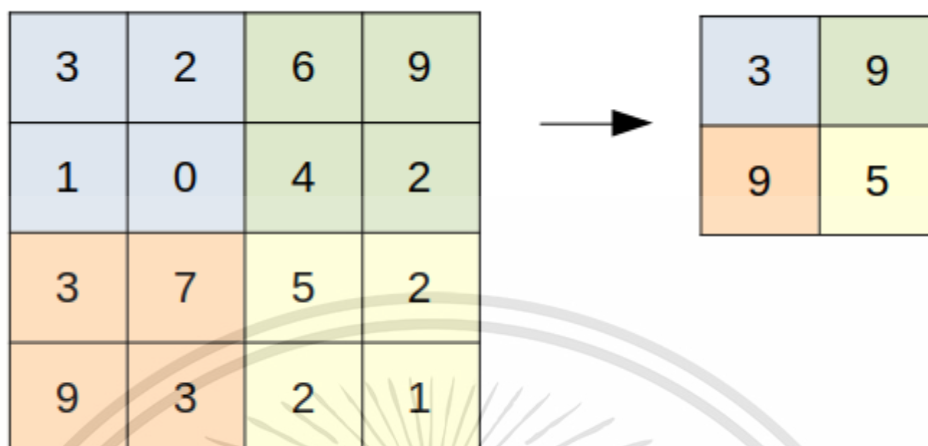
Stride หมายถึงจำนวนช่องที่จะเลื่อนไปในกระบวนการ Convolution แต่ละครั้ง เช่นถ้าเลื่อนทีละช่อง Stride  $s=1$  แต่ถ้าเลื่อนทีละ 2 ช่อง ก็คือ  $s=2$  สูตรในการคำนวณขนาด Output เมื่อใช้ Stride คือ:

$$\lfloor \frac{n+2p-fs+1}{s} \rfloor, \lfloor \frac{n+2p-fs+1}{s} \rfloor \quad (8)$$

สังเกตว่าเราจะปัดเศษลง ไม่ใช่ปัดขึ้น

### 2.9.2 ชั้นพูลิ่ง (Pooling layer)

หลังจากที่ข้อมูลผ่าน Convolution layer แล้ว จะถูกส่งเข้า Pooling layer หน้าที่ของ Pooling layer คือทำให้ผลลัพธ์มีมิติเล็กลง เพื่อเก็บ Feature สำคัญให้ AI นำมาเรียนรู้เท่านั้น เพราะโมเดลที่มีขนาดเล็ก AI จะเรียนรู้ได้อย่างรวดเร็วมากขึ้นจากภาพ แสดง Pooling layer ขนาด 2x2 โดยมีค่า Stride  $s = 2$ :



<https://guopai.github.io/ml-blog19.html> , [Online]

ภาพที่ 2.18 ภาพแสดงชั้นพูลิ่ง

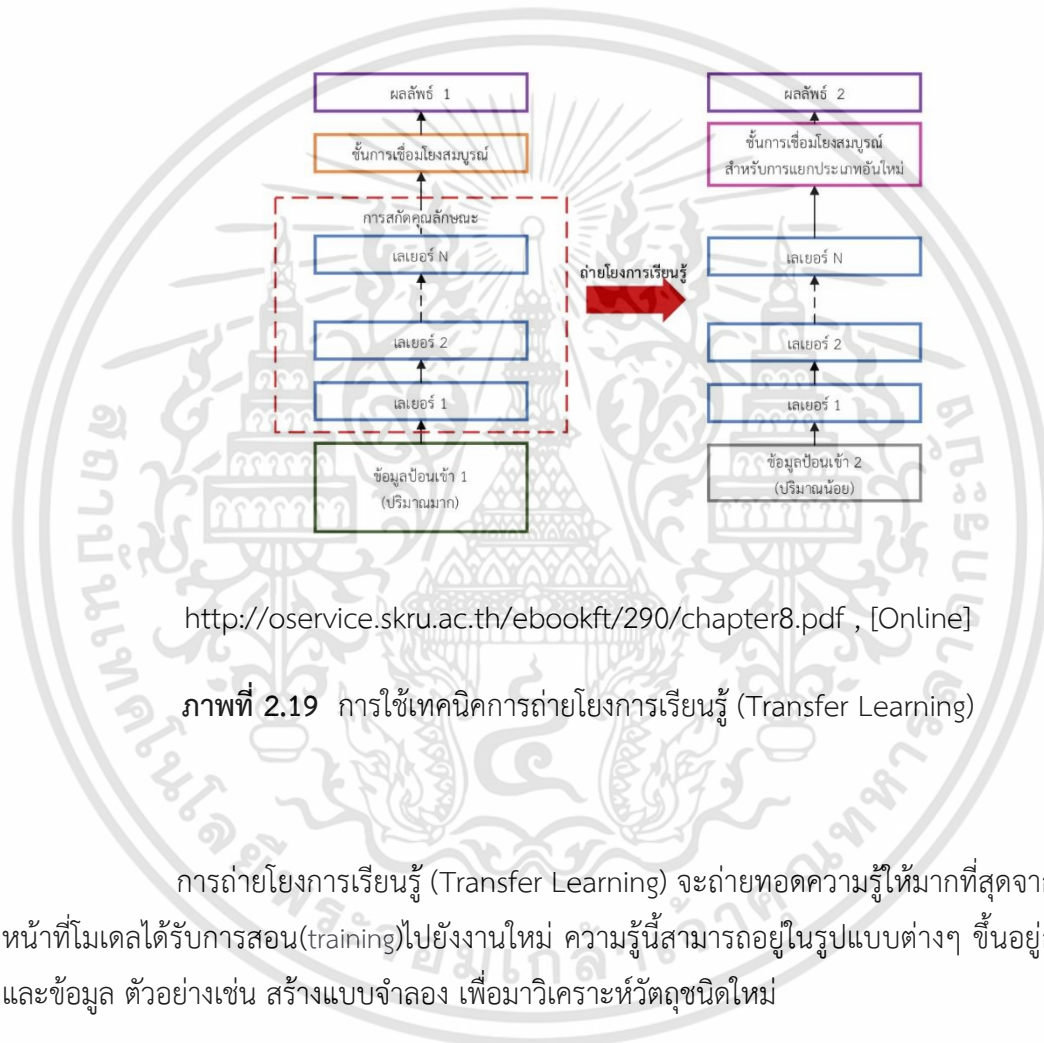
Pooling layer ที่สกัดเอาเฉพาะค่าสูงสุดของ Grid เก็บไว้ เรียกว่า Max pooling ซึ่งเป็นรูปแบบที่ใช้บ่อยที่สุด นอกจากนั้นยังมี Average pooling ซึ่งหาค่าเฉลี่ยของ Grid เก็บไว้ แต่ใช้น้อยกว่า Max pooling มาก

## 2.10 การถ่ายโยงการเรียนรู้

การถ่ายโยงการเรียนรู้ (Transfer Learning) คือการนำแบบจำลองที่ผ่านการสอน (trained) แล้วมาใช้ซ้ำในปัญหาใหม่ เพราะสามารถฝึกเครือข่ายประสาทเทียม (neural networks) ที่มีข้อมูลค่อนข้างน้อย แต่เนื่องจากปัญหาในความเป็นจริงส่วนใหญ่มักไม่มีจุดข้อมูลหลายล้านจุดในการฝึกโมเดลที่ซับซ้อนดังกล่าว

การถ่ายโยงการเรียนรู้ (Transfer Learning) เป็นความรู้เกี่ยวกับโมเดลการเรียนรู้ของเครื่องที่ได้รับการสอน (trained) แล้วจะถูกนำไปใช้กับปัญหาอื่นแต่เกี่ยวข้องกัน ด้วยการถ่ายโยงการเรียนรู้ (Transfer Learning) โดยพื้นฐานแล้ว จะพยายามใช้ประโยชน์จากสิ่งที่ได้เรียนรู้ในงานหนึ่งเพื่อปรับปรุงภาพรวมในอีกงานหนึ่ง โดยโอนน้ำหนัก (weights) ที่เครือข่ายได้เรียนรู้ที่ งาน A ไปยัง งาน B ใหม่

ตัวอย่างในการมองเห็นด้วยคอมพิวเตอร์ (computer vision) โครงข่ายประสาทเทียม (neural networks) มักจะพยายามตรวจจับขอบในชั้นก่อนหน้า รูปร่างในชั้นกลาง และคุณสมบัติเฉพาะของงานบางอย่างในชั้นต่อมา ในการถ่ายโยงการเรียนรู้ (Transfer Learning) ชั้นต้นและชั้นกลางจะถูกใช้ และเราฝึกเฉพาะชั้นหลังใหม่เท่านั้น ช่วยใช้ประโยชน์จากข้อมูลที่ติดป้ายกำกับ (label) ของงานที่ได้รับการฝึกอบรมในชั้นต้น



## 2.11 MobileNet

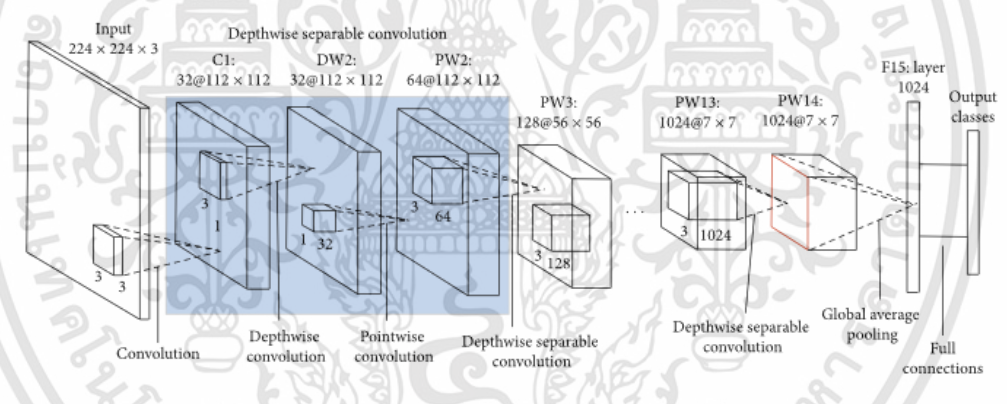
โมเดล MobileNet ได้รับการออกแบบมาเพื่อใช้ในแอปพลิเคชันมือถือ และเป็นโมเดลคอมพิวเตอร์วิชั่นซิสเต็มมือถือเครื่องแรกของ TensorFlow

MobileNet ใช้การบิดแบบแยกส่วนในเชิงลึก(Depthwise convolution) ช่วยลดจำนวนพารามิเตอร์ลงได้อย่างมากเมื่อเทียบกับเครือข่ายที่มีการบิดตัวแบบปกติที่มีความลึกเท่ากันในตาข่าย ส่งผลให้โครงข่ายประสาทเทียมลึกลงน้ำหนักเบา

การบิดแบบแยกส่วนเชิงลึกได้จากการดำเนินการ2ครั้ง คือ

- 1) การบิดตัวในเชิงลึก(Depthwise convolution)
- 2) การหมุนตามจุด(Pointwise convolution)

MobileNet เป็นคลาสของ โครงข่ายประสาทเทียมคอนโวลูชัน (Convolutional Neural Network: CNN) ที่โอเพ่นซอร์สโดย Google ดังนั้นจึงเป็นจุดเริ่มต้นที่ยอดเยี่ยมสำหรับการฝึกสอน (Traning) ตัวแยกประเภทซึ่งมีขนาดเล็กและรวดเร็ว



<https://www.hindawi.com/journals/misy/2020/7602384/> , [Online]

ภาพที่ 2.20 โมเดล MobileNet

MobileNet สามารถทำงานได้แบบเรียลไทม์ และมี Layer มากทำให้สามารถเรียนรู้วัตถุได้ผิดพลาดน้อยมาก และตรวจจับภาพทีละขั้นตอน การใช้ทรัพยากรค่อนข้างน้อยกว่า Model CNN ชนิดอื่นๆ ทำให้เกิดความแม่นยำในเกณฑ์ที่ดี

## บทที่ 3

### วิธีการดำเนินการวิจัย

#### 3.1 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินการ

##### 3.1.1 คอมพิวเตอร์

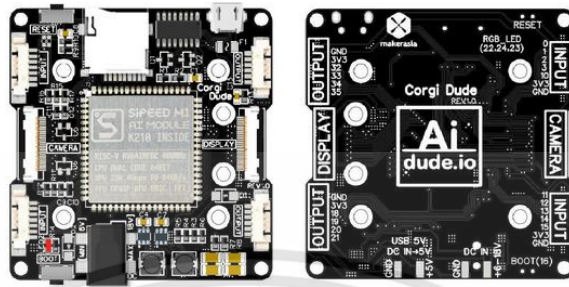


ภาพที่ 3.1 คอมพิวเตอร์

ตารางที่ 3-1 คุณสมบัติของคอมพิวเตอร์ที่ใช้งานและสอน AI

คุณสมบัติขั้นต่ำสำหรับใช้งาน	คุณสมบัติที่แนะนำสำหรับใช้งาน
1. CPU intel core I3	1. CPU intel core I5
2. RAM DDR 4 GB	2. RAM DDR 4 GB ขึ้นไป
	3. การ์ดจอ Nvidia GPU ตั้งแต่ 4 GB ขึ้นไป

### 3.1.2 บอร์ด CorgiDude



ภาพที่ 3.2 บอร์ด CorgiDude

บอร์ดสำหรับสร้างโมเดลปัญญาประดิษฐ์ Machine Learning อย่างการจำแนกภาพ ค้นหาวัตถุ ค้นหาใบหน้า มาทำการประมวลผลที่บอร์ดได้อย่างรวดเร็ว สามารถเชื่อมต่อ กล้อง จอภาพ แสดงผล หรือเซนเซอร์อื่นมาต่อและเขียนโปรแกรมเพื่อควบคุมและสั่งการได้อย่างมีประสิทธิภาพ

### 3.1.3 กล้อง OV2640



ภาพที่ 3.3 กล้อง OV2640

ตารางที่ 3-2 รายละเอียดของกล้อง

Array size	UXGA 1622X1200
Power supply	3.3V
IO voltage level	1.7V~3.3V DC
Output formats	YUV(422/420)/YCbCr422 RGB565/555 8-bit compressed data
Max image transfer rate	UXGA/SXGA 15fps, UXGA/SXGA 30fps, SVGA 30fps, CIF 60fps

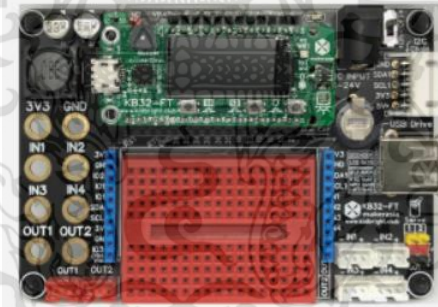
### 3.1.4 หน้าจอแสดงผลภาพ



ภาพที่ 3.4 หน้าจอแสดงผลภาพ

จอภาพขนาด 1.3 นิ้ว ความละเอียด 240x240 Pixel จอสี IPS แสดงภาพได้สว่างคมชัด มุมมองกว้าง

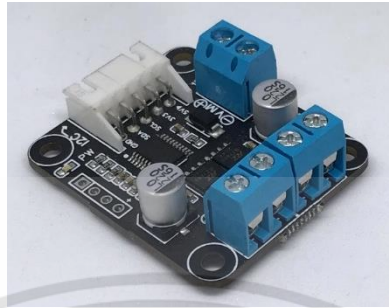
### 3.1.5 บอร์ด KB32-FT



ภาพที่ 3.5 บอร์ด KB32-FT

บอร์ด Breakout สำหรับขยายความสามารถให้เทียบเท่ากับบอร์ด Kidbright32 V1.5 มีสายรับ-ส่ง ข้อมูล Micro USB . สายไฟ I2C Chain และสาย PH2.0 สำหรับต่อ Input Output เซอร์โวมอเตอร์ SG90

### 3.1.6 KB Motor Driver



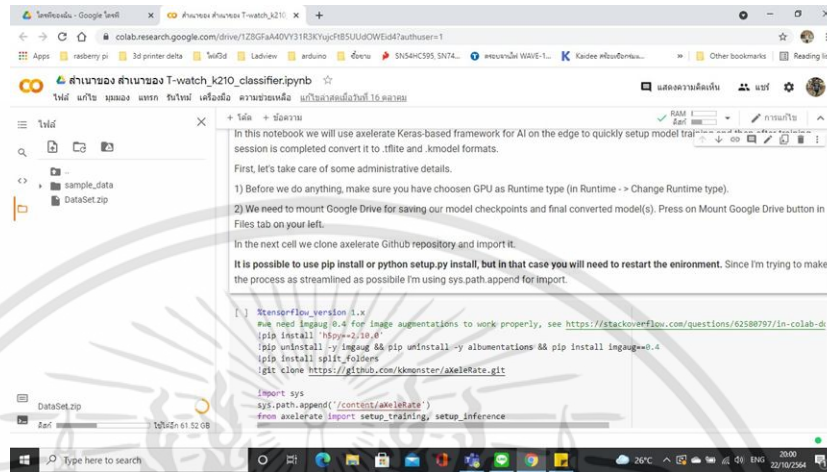
ภาพที่ 3.6 KB Motor Driver โมดูลขับมอเตอร์

ตารางที่ 3-3 คุณสมบัติของ KB Motor Driver

รับอินพุตเป็น	I2C address 0x3E
รับแรงดันไฟฟ้า	3-15V DC
กระแสที่ขับได้	3A
จำนวน Channel	2 Channel

## 3.2 Software ที่ใช้ดำเนินการ

### 3.2.1 โปรแกรม Colaboratory



ภาพที่ 3.7 หน้าต่างของโปรแกรม Colaboratory

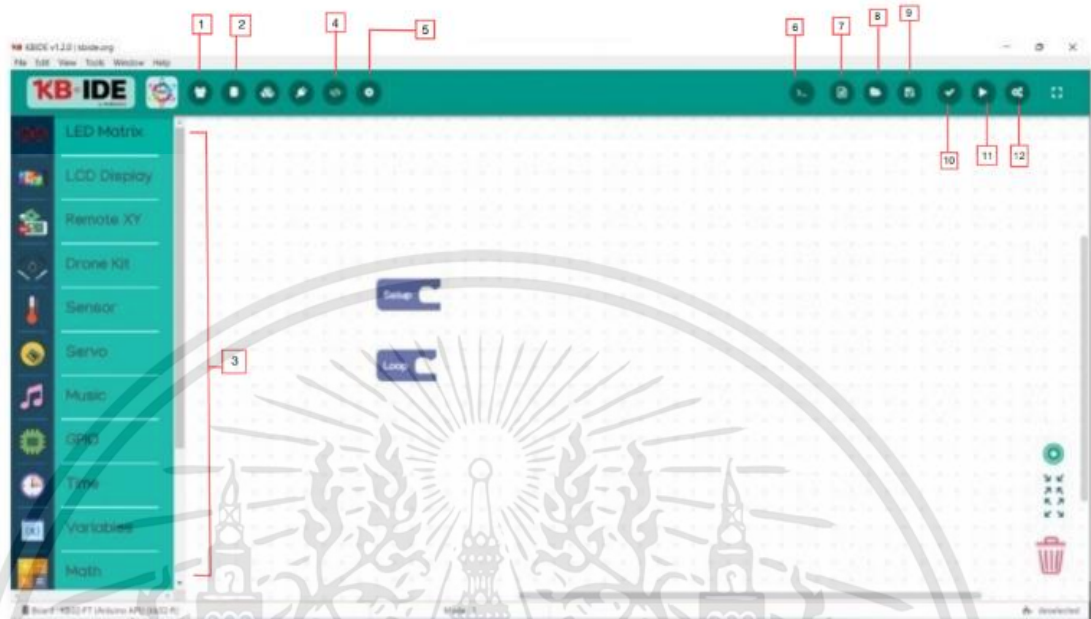
เป็นบริการคลาวด์อีกหนึ่งบริการจาก Google Research เป็น IDE ที่อนุญาตให้ผู้ใช้เขียนซอร์สโค้ดในตัวแก้ไขและเรียกใช้จากเบราว์เซอร์ โดยเฉพาะอย่างยิ่ง รองรับภาษาการเขียนโปรแกรม Python และเน้นงานแมชชีนเลิร์นนิง การวิเคราะห์ข้อมูล จากรูปที่ 3.2.1

หมายเลข 1 คือ เป็นการนำเข้าไฟล์จากคอมพิวเตอร์

หมายเลข 2 คือ พื้นที่สำหรับเขียนโปรแกรมและเงื่อนไขต่างๆ

หมายเลข 3 คือ สถานะของหน่วยความจำ

### 3.2.2 โปรแกรม KB-IDE



ภาพที่ 3.8 หน้าต่างของโปรแกรม KB-IDE

คือโปรแกรมสร้างชุดคำสั่ง เพื่อนำไปใช้ทำงานบนบอร์ด kidbright มีชุดคำสั่งแบบ block-structured programming และสามารถใช้ Code Editor ได้ หน้าต่างโปรแกรมประกอบไปด้วยส่วนต่างๆดังภาพที่3.8

หมายเลข 1 คือ คำสั่งProgramming Mode เป็นการเลือกโหมดใช้งานโปรแกรม ซึ่งหลักๆจะมี 2 ประเภทคือ Block Programming และ Text-based Programming ภาษา c/c++

หมายเลข 2 คือ คำสั่งBoard Manager ใช้สำหรับเลือกประเภทบอร์ดที่ต้องนำมาใช้งาน

หมายเลข 3 คือ ชุดคำสั่งการทำงาน

หมายเลข 4 คือ คำสั่ง Examples & Tutorials สำหรับใช้งานละเอียดอย่างเบื้องต้นของแต่ละPlugin

หมายเลข 5 คือ คำสั่ง setting เป็นการตั้งค่าโปรแกรม สำหรับเลือกสีและขนาดตัวอักษร

หมายเลข 6 คือ คำสั่ง Serial Monitor สำหรับแสดงข้อมูลที่มีความยาวมากๆแล้วแสดงไม่พอ จึงใช้คำสั่ง Serial Monitor

หมายเลข 7 คือ คำสั่ง New File สำหรับเปิดหน้าต่างไฟล์ใหม่

หมายเลข 8 คือ คำสั่ง Open File สำหรับเปิดไฟล์ที่ทำการบันทึกไว้

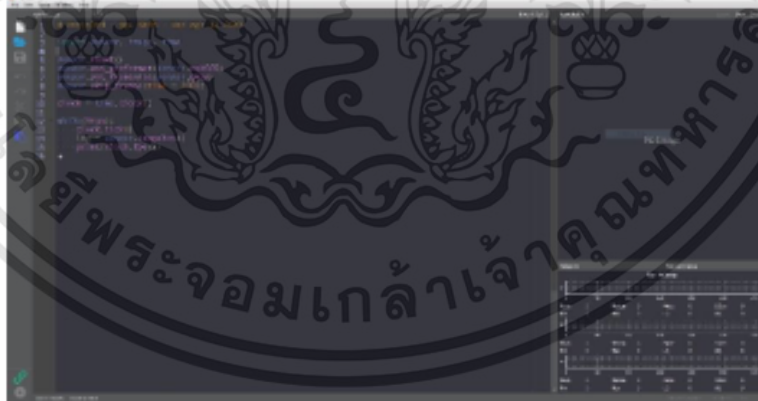
หมายเลข 9 คือ คำสั่ง Save File สำหรับการบันทึกไฟล์

หมายเลข 10 คือ คำสั่ง Just Compile สำหรับการตรวจสอบ Code ว่ามีการเขียนผิดพลาดหรือไม่

หมายเลข 11 คือ คำสั่ง Compile & Run สำหรับตรวจสอบโค้ดและอัปโหลดชุดคำสั่งลงไปบอร์ด

หมายเลข 12 คือ คำสั่ง Setup Board สำหรับการเลือกเชื่อมต่อระหว่างคอมพิวเตอร์กับตัวบอร์ด

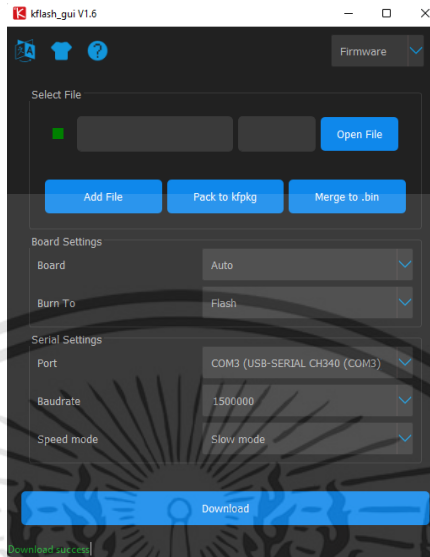
### 3.2.3 Maixpy IDE



ภาพที่ 3.9 หน้าต่างของโปรแกรม Maixpy IDE

เป็นซอฟต์แวร์ที่ประมวลผลสำหรับแปลงโค้ดคำสั่งของ microphyton ลงบอร์ด CorgiDude

### 3.2.4 kflash\_gui



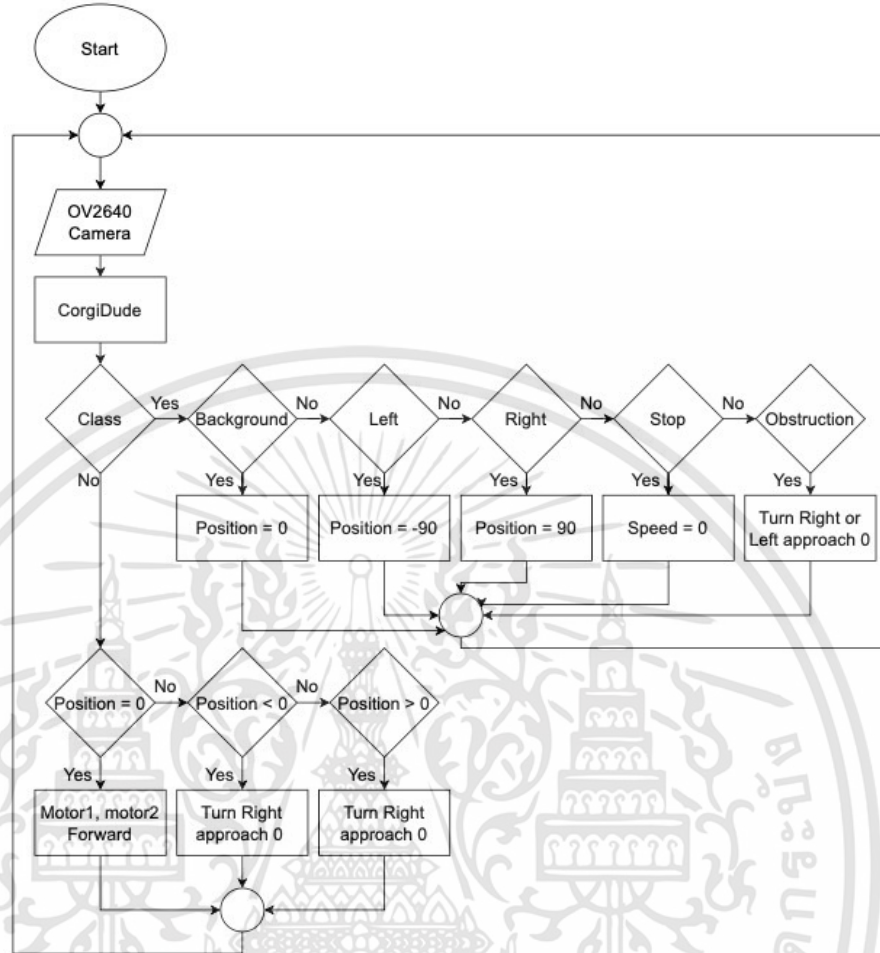
ภาพที่ 3.10 หน้าต่างของโปรแกรม Kflash\_gui

เป็นโปรแกรมสำหรับอัปเดตเฟิร์มแวร์ที่ได้จากการสอน AI ลงในตัวบอร์ด CorgiDude

## 3.3 การออกแบบระบบ

### 3.3.1 หลักการออกแบบการขับเคลื่อนโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

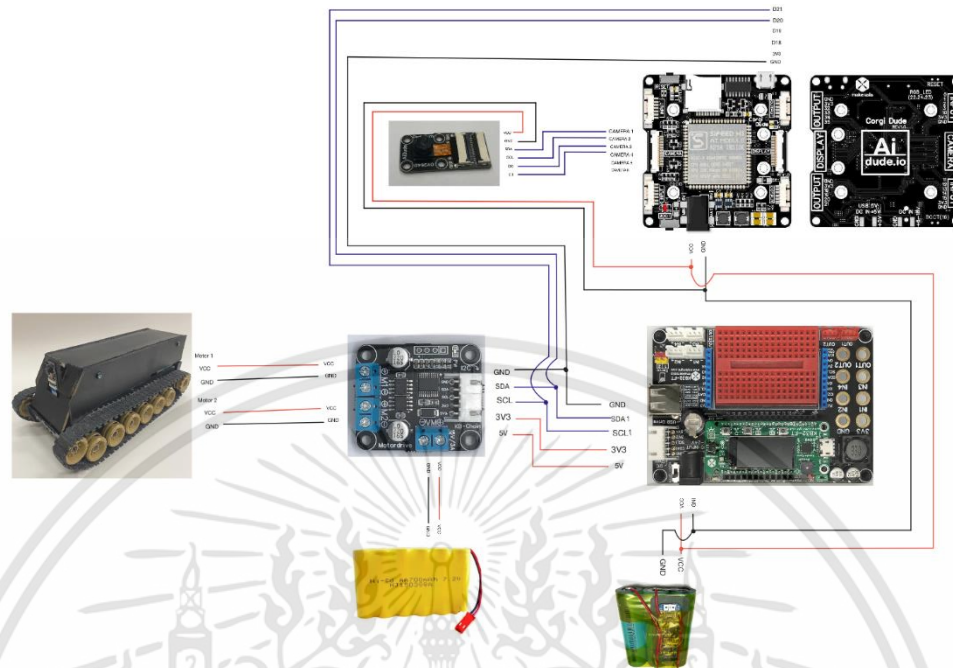
หลักการออกแบบโมเดลยานยนต์ไฟฟ้าอัจฉริยะโดยใช้การประมวลผลภาพและแสดงผลบนหน้าจอสามารถแบ่งการทำงานเป็น 2 ส่วนคือส่วนของกล้องที่ใช้ตรวจจับวัตถุ ที่ผ่านหน้าโมเดลยานยนต์ไฟฟ้าอัจฉริยะและส่วนของระบบควบคุมและขับเคลื่อนโมเดลยานยนต์ไฟฟ้าอัจฉริยะ เพื่อให้เคลื่อนที่อยู่ในช่องทางและเคลื่อนที่ตามป้ายสัญลักษณ์ การออกแบบการขับเคลื่อนโมเดลยานยนต์ไฟฟ้าอัจฉริยะแสดงในภาพที่3.11



ภาพที่ 3.11 แผนผังการทำงานของระบบการขับเคลื่อนโมเดลยานยนต์ไฟฟ้าอัจฉริยะแบบอัตโนมัติ

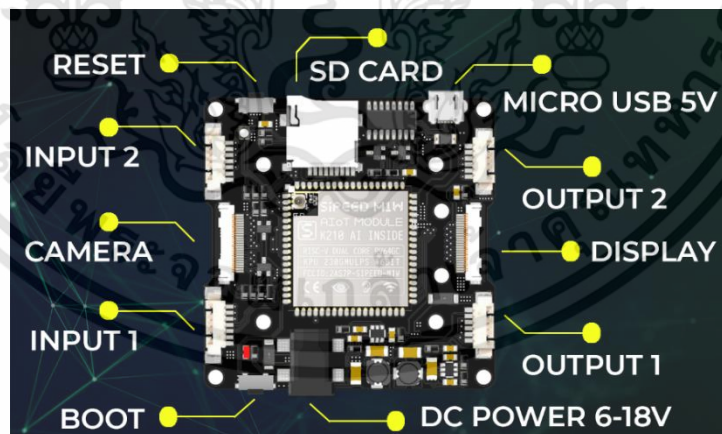
### 3.3.2 ขั้นตอนการทำงานของ CorgiDude

โดยในส่วนของ CorgiDude ออกแบบเพื่อทำการตรวจจับวัตถุบนถนนเพื่อสั่งการไปยังระบบควบคุมของโมเดลยานยนต์ไฟฟ้าอัจฉริยะ โดยการติดตั้งกล่องควบคุม CorgiDude เพิ่มเติมไว้หน้ารถโดยภายในกล่องควบคุมมีการเชื่อมต่ออุปกรณ์ แสดงในภาพที่ 3.12



ภาพที่ 3.12 การเชื่อมต่อวงจร CorgiDude กับโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

### 3.4 โครงสร้างภายในของบอร์ด CorgiDude

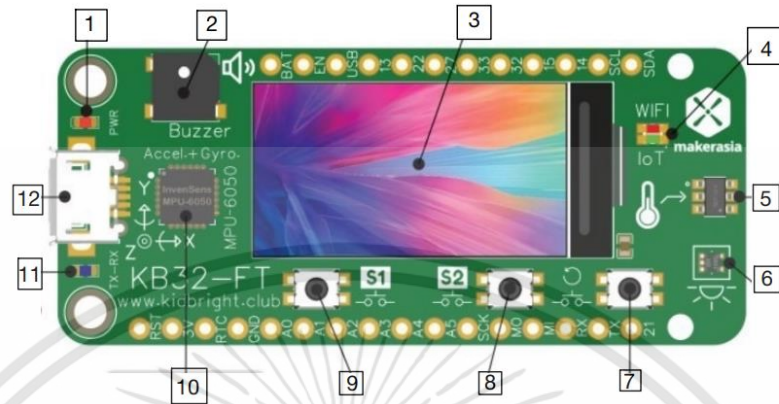


<https://www.aiiotshop.com/article/14/corgidude-> , [Online]

ภาพที่ 3.13 รายละเอียดโครงสร้างภายในของบอร์ด CorgiDude

- 1) ช่องต่อ Input พอร์ตสำหรับต่อเซนเซอร์ที่อ่านค่าได้ เช่นอุณหภูมิ ความชื้น ความเร็ว ความเร่ง แรงดันไฟฟ้า โดยจะมีทั้งหมด 2 ช่อง (INPUT1 และ INPUT2) ช่องละ 5 GPIO นับรวมไฟเลี้ยง 3.3v รวมเป็น 7 Pin รวมกันแล้วสามารถใช้ GPIO ได้ทั้งหมด 10 Pin
- 2) ช่องต่อ Output มีลักษณะคล้ายกันกับ Input แต่ใช้สำหรับต่ออุปกรณ์ Output เช่น ควบคุมมอเตอร์ เซอร์โวมอเตอร์ ปิดเปิด Relay โดยช่อง Output จะมี 2 ช่อง คือ OUTPUT1 และ OUTPUT2 แต่ละช่องจะมี GPIO ทั้งหมด 4 ตัว และไฟเลี้ยง 2 ขา รวมเป็น 6 ขา รวมแล้วใช้ GPIO ได้ทั้งหมด 8 Pin ด้วยกัน
- 3) ช่องต่อกล้อง จะมีคำว่า “CAMERA” เขียนกำกับไว้ โดยกล้องที่รองรับจะเป็นรุ่น OV2640 24Pin
- 4) ช่องต่อจอภาพ จอภาพที่ต่อได้จะเป็นรุ่น ST7789 การต่อคล้ายกันกับการต่อกล้อง คือ จะมีคอนเน็คเตอร์เขียนว่า “DISPLAY” กำกับเอาไว้
- 5) ช่องต่อไฟเลี้ยง ไฟเลี้ยงบอร์ดสามารถใช้ไฟจาก USB 5V หรือเลือกเสียบ Adaptor ที่จ่ายไฟเลี้ยงได้ตั้งแต่ 6-18V DC โดยใช้แจ๊คขนาดประมาณ 2x5.5mm ปุ่มกดในบอร์ด จะมีทั้งหมด 2 ปุ่มคือ
  - RESET ใช้สำหรับกด reset บอร์ดให้เริ่มทำงานใหม่
  - BOOT ใช้สำหรับกดเพื่อควบคุมการเข้า Boot โหมด หรือ สั่งทำงานต่าง ๆ โดยต่อไว้กับ GPIO 16
- 6) ช่องเสียบ Micro-SD ใช้สำหรับการโหลดไฟล์หรือบันทึกไฟล์ต่างๆ
- 7) หลอดไฟ RGB สำหรับแสดงสถานะต่างๆ

### 3.5 โครงสร้างภายในของบอร์ด KB-32FT



ภาพที่ 3.14 รายละเอียดและโครงสร้างภายในของบอร์ด KB-32FT

ในส่วนของการเขียนโปรแกรมเพื่อให้ตัวบอร์ด KB-32FT จำค่าที่ได้กำหนดไว้ในเงื่อนไขของโปรแกรมหน้าต่างของโปรแกรมประกอบด้วยส่วนต่างๆ แสดงในภาพที่ 3.14

หมายเลข 1 คือ Power LED

หมายเลข 2 คือ electromagnetic buzzer รุ่น MLT-5020

หมายเลข 3 คือ จอแสดงภาพ TFT LCD 0.96” Simulation LED Matrix 16x8,LCD Drive Mode

หมายเลข 4 คือ LED wifi & IoT 2 LED

หมายเลข 5 คือ Temperature Sensor รุ่น LM73

หมายเลข 6 คือ Ambient light sensor รุ่น BH1680FVC

หมายเลข 7 คือ ปุ่ม Reset

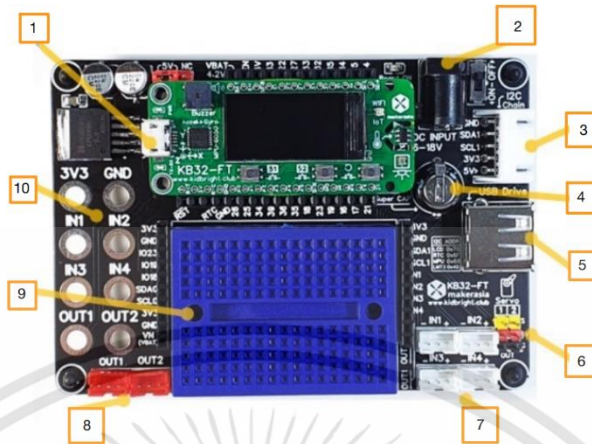
หมายเลข 8 คือ สวิตช์ที่ 2

หมายเลข 9 คือ สวิตช์ที่ 1

หมายเลข 10 คือ Gyro & Accelerometer รุ่น MPU-6050

หมายเลข 11 คือ Communication Indicator LED TX & RX

หมายเลข 12 คือ Micro USB



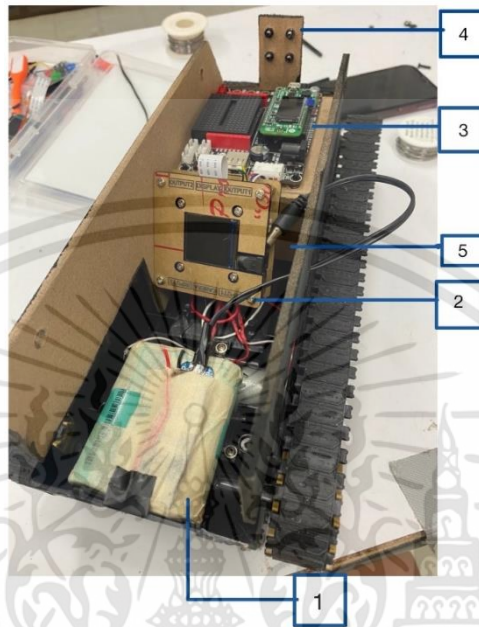
ภาพที่ 3.15 โครงสร้างภายในของบอร์ด KB-32FT พร้อม Breakout Board

รายละเอียดและโครงสร้างภายในบอร์ด KB-32FT พร้อม Breakout Board แสดงในภาพ

ที่ 3.15

- หมายเลข 1 คือ KB32-FT
- หมายเลข 2 คือ INPUT 6-24v DC
- หมายเลข 3 คือ KB-CHAIN Connector
- หมายเลข 4 คือ RTC battery
- หมายเลข 5 คือ USB Power Connector
- หมายเลข 6 คือ Servo Connector
- หมายเลข 7 คือ JST Connector (Input)
- หมายเลข 8 คือ JST Connector (Output)
- หมายเลข 9 คือ Breadboard
- หมายเลข 10 คือ Banana Connector 4 mm.

### 3.6 การวางตำแหน่งอุปกรณ์ภายในโมเดลยานยนต์ไฟฟ้าอัจฉริยะ



ภาพที่ 3.16 การวางตำแหน่งของอุปกรณ์ภายในโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

ในการออกแบบระบบกล้องควบคุมสำหรับใช้งานในโมเดลยานยนต์ไฟฟ้าอัจฉริยะเป็นการออกแบบอุปกรณ์ที่ติดตั้งเสริมไปทำให้ระบบที่ใช้ในการนำทางมีประสิทธิภาพเพิ่มมากขึ้นโดยมีการวางตำแหน่งอุปกรณ์ภายในโมเดลยานยนต์ไฟฟ้าอัจฉริยะ แสดงในภาพที่ 3.16

หมายเลข 1 คือ แบตเตอรี่สำหรับชาร์จ 12.6V

หมายเลข 2 คือ ตัวบอร์ด CorgiDude ประกอบไปด้วยกล้องและหน้าจอแสดงผล

หมายเลข 3 คือ บอร์ด KB-32FT พร้อม Breakout Board

หมายเลข 4 คือ ตำแหน่งกล้อง OV2640

หมายเลข 5 คือ KB Motor Driver

### 3.7 ขั้นตอนในการสอน AI

มีขั้นตอนในการสอน 2 ส่วน

- 1) สอน AI ในการจำแนกป้ายสัญลักษณ์และวัตถุบนถนน
- 2) สอน AI ให้จดจำตำแหน่งของโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

#### 3.7.1 สอน AI ในการจำแนกป้ายสัญลักษณ์และวัตถุบนถนน

ขั้นตอนที่ 1 : ทำการเก็บ Dataset ของป้ายสัญลักษณ์และวัตถุบนถนนมาแยกเป็นหมวดหมู่ที่นำมาใช้ในการทดลอง แยกเป็น Class ต่างๆไว้ในโฟลเดอร์

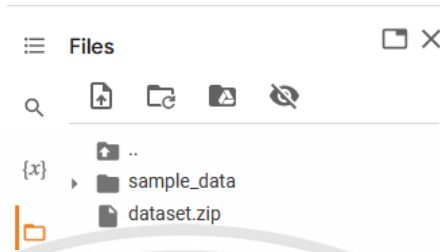


ภาพที่ 3.17 การแยก Dataset ของป้ายสัญลักษณ์และวัตถุบนถนน



ภาพที่ 3.18 ตัวอย่าง Dataset ของป้ายสัญลักษณ์และวัตถุบนถนนที่ใช้ทำการทดลอง

ขั้นตอนที่ 2 : ทำการบีบอัดไฟล์ที่ได้จากการเก็บ Dataset และอัปโหลดลงในโปรแกรม Colaboratory



ภาพที่ 3.19 อัปโหลด Dataset ลงในโปรแกรม Colaboratory

ขั้นตอนที่ 3 : ทำการติดตั้งไลบรารี tensorflow Version.1.x0 และ split\_folders และทำการ Clone-Module : <https://github.com/AIWintermuteAI/aXeLeRate.git>

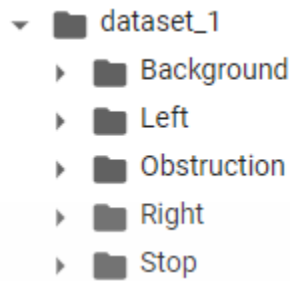
```
# tensorflow version 1.x
# we need imgaug 0.4 for image augmentations to work properly, see https://stackoverflow.com/questions/62580797/in-colab-doing-image-data-augmentation-with-imgaug-is-not-working-as-intended
!pip uninstall -y imgaug && pip uninstall -y albumentations && pip install imgaug==0.4
!pip install split_folders
#!git clone https://github.com/kkmonster/aXeLeRate.git
!git clone -b legacy-yolov2 https://github.com/AIWintermuteAI/aXeLeRate.git

import sys
sys.path.append('/content/aXeLeRate')
from axelerate import setup_training, setup_inference
```

ภาพที่ 3.20 หน้าต่างติดตั้ง Library

ขั้นตอนที่ 4 : ทำการแตกไฟล์ Dataset ที่ได้ทำการอัปโหลด ได้ไฟล์ที่มี 5 Class ได้แก่

- Class 1 หรือ background = ถนนและขอบทาง
- Class 2 หรือ Left = ป้ายเลี้ยวซ้าย 5 รูปทรง
- Class 3 หรือ Obstruction = กระจบองโค้ง กระจบองกาแพ กระจบองวิตซ่า กระจบองไม้และบอล
- Class 4 หรือ Right = ป้ายเลี้ยวขวา 5 รูปทรง
- Class 5 หรือ Stop = หุ่นมนุษย์ หุ่นแมว กล่อง กล่องแว่น และ ป้ายหยุด



ภาพที่ 3.21 ไฟล์ที่ได้จากการแตกไฟล์เข้าโปรแกรม

ขั้นตอนที่ 5 : ทำการแบ่งไฟล์รูปภาพป้ายสัญลักษณ์และวัตถุบนถนนต่างๆ เพื่อนำไปทำการสอน AI

```

import splitfolders

# The path to the directory where the original dataset was uncompressed
input_folder = '/content/dataset'
# The directory where we will store our smaller dataset
output_folder = '/content/data_for_training'

splitfolders.ratio(input_folder, output_folder, seed=1337, ratio=(.8, .2), group_prefix=None)
  
```

ภาพที่ 3.22 การแบ่งไฟล์รูปภาพป้ายสัญลักษณ์และวัตถุบนถนนต่างๆเพื่อนำไปทำการสอน AI

ขั้นตอนที่ 6 : ทำการตั้งค่าต่างๆที่ใช้สำหรับการสอน AI เช่น ประเภทของAI backend ขนาดรูปภาพ จำนวนรอบในการสอน และตำแหน่งที่เก็บข้อมูลที่ได้จากการสอน

```

config = {
    "model": {
        "type": "Classifier",
        "architecture": "MobileNet7_5",
        "input_size": 224,
        "fully-connected": [],
        "labels": [],
        "dropout": 0.2
    },
    "weights": {
        "full": "",
        "backend": "imagenet",
        "save_bottleneck": False
    },
    "train": {
        "actual_epoch": 50,
        "train_image_folder": "/content/data_for_training/train",
        "train_times": 2,
        "valid_image_folder": "/content/data_for_training/val",
        "valid_times": 2,
        "valid_metric": "val_accuracy",
        "batch_size": 128,
        "learning_rate": 1e-4,
        "saved_folder": "classifier",
        "first_trainable_layer": "",
        "augmentation": False
    },
    "converter": {
        "type": ["k210", "tflite"]
    }
}
  
```

ภาพที่ 3.23 การตั้งค่าต่างในการสอน Dataset

ขั้นตอนที่ 7 : ทำการสอน Dataset ป้ายสัญลักษณ์และวัตถุบนถนนได้ผลลัพธ์ที่ดีที่สุดมาใช้งานในการสอน AI โดยได้เป็นไฟล์.kmodel มีค่า val\_accuracy = 0.9981

```
Epoch 18/20  
206/206 [=====] - ETA: 0s - loss: 0.0208 - accuracy: 0.9935  
Epoch 18: val_accuracy improved from 0.99778 to 0.99813, saving model to projects/classifier/2022-05-30-18-37-42/Classifier_best_val_accuracy.h5  
Epoch 00017: Learning rate is 3.392610517141681e-06.  
  
206/206 [=====] - 200s 959ms/step - loss: 0.0208 - accuracy: 0.9935 - val_loss: 0.0060 - val_accuracy: 0.9981
```

ภาพที่ 3.24 คำสั่งในการสอน Dataset

ขั้นตอนที่ 8 : สุ่มข้อมูลตัวอย่างเพื่อมาทำการตรวจสอบความแม่นยำ

```
%matplotlib inline  
from keras import backend as K  
K.clear_session()  
setup_inference(config, model_path)
```



projects/classifier/2022-05-25\_09-47-30/Inference\_results/1850.jpg  
Left:0.9970856414604187

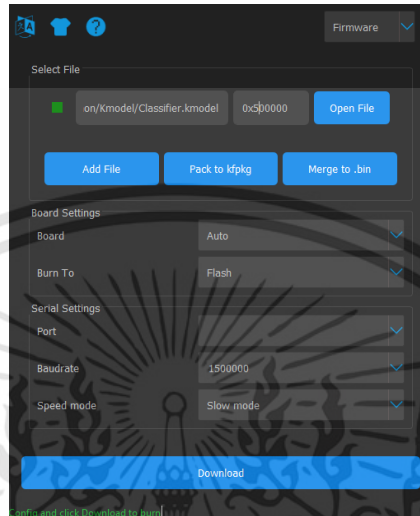


projects/classifier/2022-05-25\_09-47-30/Inference\_results/1892.jpg  
Left:0.9986874461174011

projects/classifier/2022-05-25\_09-47-30/Inference\_results/1308.jpg

ภาพที่ 3.25 การสุ่มข้อมูลตัวอย่างเพื่อตรวจสอบความแม่นยำ

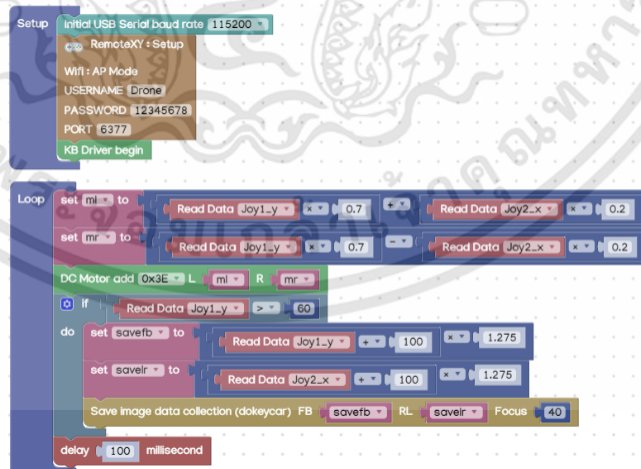
ขั้นตอนสุดท้าย : เมื่อทำการตรวจสอบความแม่นยำแล้วสามารถนำไปใช้งานได้จริง ทำการอัปโหลดไฟล์.kmodel ลงในบอร์ด CorgiDude ผ่านโปรแกรม Kflash\_gui



ภาพที่ 3.26 การอัปโหลดไฟล์.kmodel จากการสอน AI ลงบอร์ด CorgiDude ที่ตำแหน่ง 0x500000

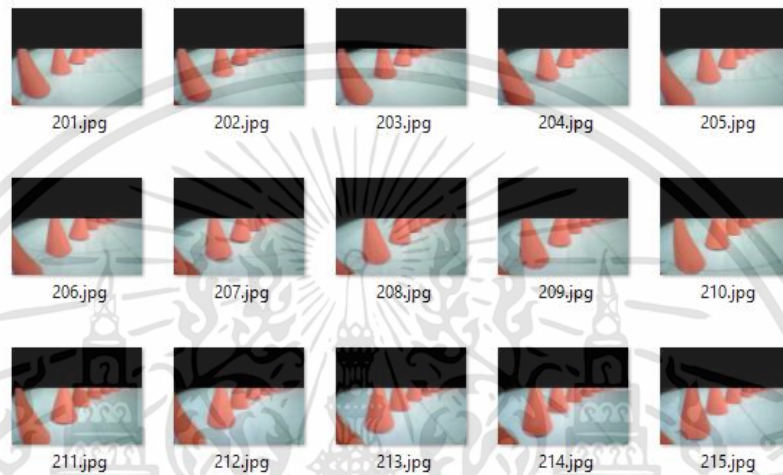
### 3.7.2 สอน AI ให้จดจำตำแหน่งของโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

ขั้นตอนที่ 1 : ใช้บอร์ด KB32-FT ในการควบคุมรถช่วยเก็บ Dataset ของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะโดยวิ่งไปตามขอบทาง เช่น ซิดซ้ายสุด ซิดขวาสุด และตรงกลาง โดยรับคำสั่งจากโปรแกรม KB IDE



ภาพที่ 3.27 คำสั่งควบคุมรถเพื่อทำการเก็บ Dataset ตำแหน่งของโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

ขั้นตอนที่ 2 : ทำการเก็บ Dataset ของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะที่อยู่ในขอบทาง เช่น ซิดซ้ายสุด ซิดขวาสุด และตรงกลางโดยแต่ละรูปมีการกำหนดตำแหน่งอยู่ในไฟล์.csv

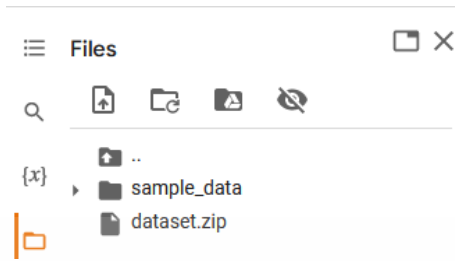


ภาพที่ 3.28 ตัวอย่าง Dataset ของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะในขอบทาง

10/1021	0	0
10/1022	0	0
10/1023	0	0
10/1024	0	31
10/1025	0	31
10/1026	0	31

ภาพที่ 3.29 ตัวอย่างข้อมูลในไฟล์.csv บอกตำแหน่งที่เก็บรูปภาพและตำแหน่งของรถในแต่ละรูปภาพ

ขั้นตอนที่ 3 : ทำการบีบอัดไฟล์ที่ได้จากการเก็บ Dataset และอัปโหลดลงในโปรแกรม Colaboratory



ภาพที่ 3.30 อัปโหลด Dataset ลงในโปรแกรม Colaboratory

ขั้นตอนที่ 4 : ทำการติดตั้งไลบรารี tensorflow Version.1.x0, ติดตั้ง h5py Version

2.10.0 และ split\_folders และทำการ Clone-Module :

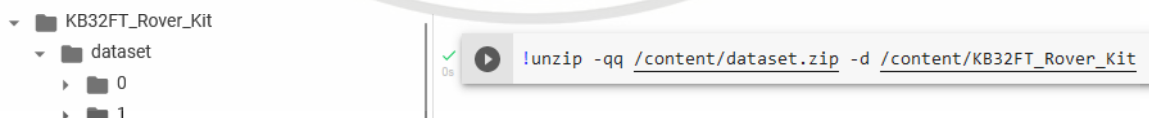
[https://github.com/kkmonster/KB32FT\\_Rover\\_Kit.git](https://github.com/kkmonster/KB32FT_Rover_Kit.git)

```
%tensorflow_version 1.x
!git clone https://github.com/kkmonster/KB32FT_Rover_Kit.git
!mkdir -p /content/KB32FT_Rover_Kit/ncc
!wget -P /content/KB32FT_Rover_Kit/ https://github.com/kendryte/nncase/releases/download/v0.1.0-rc5/ncc-linux-x86_64.tar.xz
!wget -P /content/KB32FT_Rover_Kit/ https://github.com/kendryte/nncase/releases/download/v0.2.0-beta2/ncc_linux_x86_64.tar.xz
!tar xf /content/KB32FT_Rover_Kit/ncc-linux-x86_64.tar.xz --directory=/content/KB32FT_Rover_Kit/ncc/
!tar xf /content/KB32FT_Rover_Kit/ncc_linux_x86_64.tar.xz --directory=/content/KB32FT_Rover_Kit/ncc/
!pip uninstall h5py -y
!pip install h5py==2.10.0
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # showing and rendering figures
%matplotlib inline
```

ภาพที่ 3.31 คำสั่งติดตั้ง Library ที่ใช้ในการสอน AI

ขั้นตอนที่ 5 : ทำการแตกไฟล์ Dataset ที่ได้ทำการอัปโหลดและนำไปเก็บในไฟล์

KB32FT\_Rover\_kit



ภาพที่ 3.32 การแตกไฟล์รูปภาพเข้าไปเก็บไว้ใน KB32FT\_Rover\_kit

## ขั้นตอนที่ 6 : ทำการปรับขนาดของรูปภาพก่อนนำไปสอน AI

```
import math
import csv
import cv2
import keras
import numpy as np
from keras import backend as K
from tensorflow.keras.optimizers import Adam
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
from keras.preprocessing import image
from keras.models import Model
from keras.applications import ImageNet_utils
from keras.layers import Dense, GlobalAveragePooling2D, Dropout, Conv2D
from keras.applications.mobilenet import preprocess_input
from keras.callbacks import ModelCheckpoint
from keras.applications.mobilenet import MobileNet

def dev_image_generator(input_feed,data_folder, bs, mode="train", aug=None):
    index = 0
    NUM_IMAGES = len(input_feed)
    while True:
        # initialize our batches of images and labels
        images = []
        poly_coef = []
        # keep looping until we reach our batch size
        while len(images) < bs:
            line = input_feed[index]
            index = index + 1
            if index >= len(input_feed):
                index = 0;
            center_img_path = line[1]
            file_name = center_img_path
            # print("./{}{}".format(data_folder, file_name))
            if len(file_name) > 0:
                image = cv2.imread("./{}{}".format(data_folder, file_name))
                # image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                # image = cv2.resize(image,(100,100))
                if image is not None:
                    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                    image = cv2.resize(image,(100,100))
                    # pixels = np.asarray(image)
                    # pixels = pixels.astype('float32')
                    # pixels = pixels/255
                    pixels = preprocess_input(image)
                    images.append(pixels)
                # images.append(np.flipr(image))
            xi = ((float(line[1])/137.5) -1)*1.3
            poly_coef.append(xi)
        # if the data augmentation object is not None, apply it
        if aug is not None:
            (images, labels) = next(aug.flow(np.array(images),
            labels, batch_size=bs))
        # yield the batch to the calling function
        yield (np.array(images), np.array(poly_coef))
    image_load_data = []
    with open("dataset_prepare.csv") as f:
        next(f, None)
        reader = csv.reader(f)
        for csv_line in reader:
            image_load_data.append(csv_line)
            # print(csv_line)
    image_load_data = shuffle(image_load_data)
```

ภาพที่ 3.33 คำสั่งปรับขนาดรูปภาพเตรียมก่อนนำไปสอน AI

## ขั้นตอนที่ 7 : ทำการแบ่งรูปเป็นสอน (Train) และทดสอบ (Test) เพื่อทำการสอน AI

และกำหนดจำนวนรอบในการสอน รวมถึงความเร็วของรูปในการสอนต่อ1ครั้ง

```
NUM_EPOCHS = 100
```

```
BATCH_SIZE = 64
```

```
data_folder = 'dataset'
```

```
# initialize the total number of training and testing image
```

```
NUM_TRAIN_IMAGES = len(data_train)
```

```
NUM_TEST_IMAGES = len(data_test)
```

```
trainGen = dev_image_generator(data_train,data_folder,BATCH_SIZE, mode = "train", aug = None)
```

```
testGen = dev_image_generator(data_test,data_folder,BATCH_SIZE, mode = "train", aug = None)
```

```
print("NUM_TRAIN_IMAGES", NUM_TRAIN_IMAGES)
```

```
print("NUM_TEST_IMAGES", NUM_TEST_IMAGES)
```

```
NUM_TRAIN_IMAGES 6785
```

```
NUM_TEST_IMAGES 1697
```

ภาพที่ 3.34 การแบ่งรูปภาพและกำหนดจำนวนครั้งในการสอน

ขั้นตอนที่ 8 : คำสั่งทำการสอน Dataset และได้ผลลัพธ์ที่ดีที่สุด โดยได้เป็นไฟล์.h5 มีค่า val\_loss = 0.00355 สามารถนำไปใช้งานได้จริง

```
base_model=MobileNet(input_shape=(160, 160, 3), alpha = 0.5,depth_multiplier = 1, dropout = 0.001,include_top = False)

x=base_model.output
x= Conv2D(1, (1, 1), padding='same',use_bias=False,strides=(1, 1))(x)
preds=GlobalAveragePooling2D()(x)
# x=Dropout(0.3)(x)
# x=Flatten()(x)
# x=Dropout(0.2)(x)
# x=Dense(10,activation='relu')(x)
# preds=Dense(1,activation='linear')(x)
model=Model(inputs=base_model.input,outputs=preds)

# model.summary()

filepath="w-imp-{epoch:02d}-{val_loss:.5f}.h5"
checkpoint = ModelCheckpoint(filepath, monitor='val_loss', verbose=1, save_best_only=True, mode='min')
callbacks_list = [checkpoint]

model.compile(optimizer='adam', loss='mse')

print("[INFO] training w/ generator...")
H = model.fit_generator(trainGen, steps_per_epoch=NUM_TRAIN_IMAGES // BATCH_SIZE,
                        validation_data=testGen, validation_steps=NUM_TEST_IMAGES // BATCH_SIZE,
                        epochs=NUM_EPOCHS, callbacks=callbacks_list)

Epoch 50/100
106/106 [=====] - 16s 149ms/step - loss: 0.0040 - val_loss: 0.0035
```

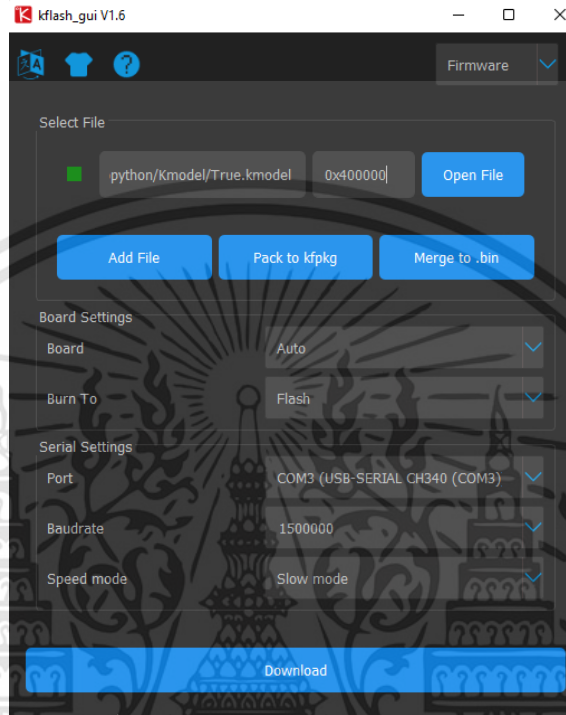
ภาพที่ 3.35 คำสั่งการสอน Dataset และได้ผลลัพธ์ที่ดีที่สุด

ขั้นตอนที่ 9 : ทำการแปลงไฟล์.h5 เป็นไฟล์.kmodel เพื่อนำไปใช้งานในบอร์ด CorgiDude

```
!tfllite_convert --output_file=k210_160.tflite --keras_model_file=w-imp-50-0.00355.h5
!./ncc compile k210_160.tflite "/content/ai_model.kmodel" -i tflite -o kmodel --dataset /content/KB32FT_Rover_Kit/dataset/0 --input-mean 0.5 --input-std 0.5
```

ภาพที่ 3.36 คำสั่งแปลงไฟล์.h5 เป็นไฟล์.kmodel

ขั้นตอนสุดท้าย : ทำการอัปโหลดไฟล์.kmodel ลงในบอร์ด CorgiDude ผ่านโปรแกรม Kflash\_gui ลงคนละตำแหน่งกับ ไฟล์.kmodel ที่ได้จากการเรนป้ายสัญลักษณ์และวัตถุดิบถนน



ภาพที่ 3.37 อัปโหลดไฟล์.kmodel จากการสอน AI ลงบอร์ด CorgiDude ที่ตำแหน่ง 0x400000

## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 การเก็บข้อมูลรูปภาพเป็น Dataset

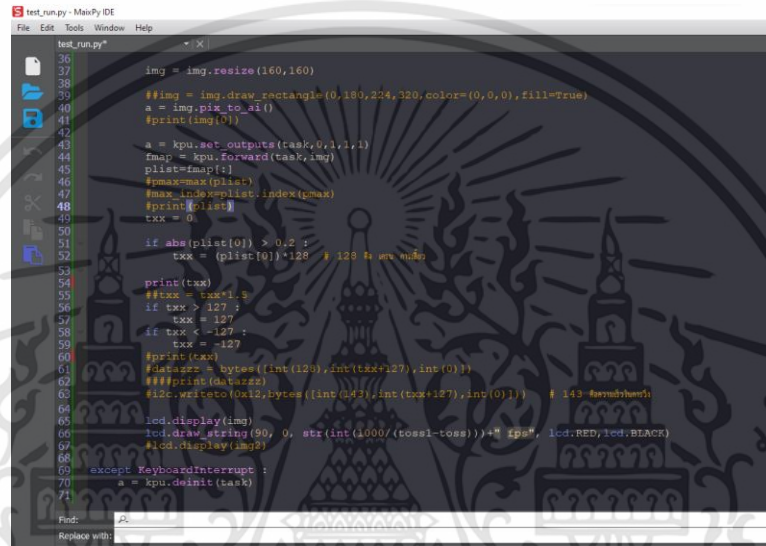
ในการสอน AI จะมีจำนวนรูปภาพที่เป็น Dataset 2 ชุดด้วยกันได้แก่ Dataset ของตำแหน่งรถ และ Dataset ของป้ายสัญลักษณ์และวัตถุบนถนน เพื่อให้มีจำนวนรูปภาพที่มากพอสำหรับการสอน AI ซึ่งจะมีการสอน Dataset 2 ครั้ง และแบ่งเป็นชุดข้อมูลที่ได้ 2 ชุดตามที่อธิบายในบทที่ 3



ภาพที่ 4.1 ตัวอย่างของ Dataset

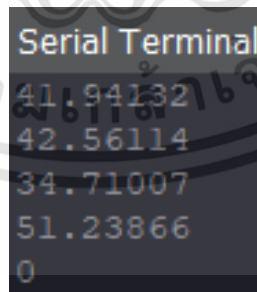
## 4.2 การทดสอบความแม่นยำของตำแหน่งรถ

การทดสอบความแม่นยำของตำแหน่งรถโดยเขียนโปรแกรมคำสั่งผ่านโปรแกรม MaixPy IDE ตรวจสอบความถูกต้องของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะ เมื่อทำการวางโมเดลยานยนต์ไฟฟ้าอัจฉริยะอยู่ชิดขอบถนนฝั่งซ้ายสุดจะได้แสดงค่า -150 ชิดขอบถนนฝั่งขวาสุดจะแสดงค่า 150 และเมื่ออยู่ตรงกลางระหว่างขอบถนนทั้งสองข้างจะแสดงค่าเป็น 0



```
test_run.py* - MaixPy IDE
File Edit Tools Window Help
test_run.py*
36 img = img.resize(160,160)
37
38 #img = img.draw_rectangle(0,180,224,320,color=(0,0,0),fill=True)
39
40 a = img.pix_to_ai()
41 #print(img(0))
42
43 a = kpu.set_outputs(task,0,1,1,1)
44 fmap = kpu.forward(task,img)
45 plist=fmap[1]
46 #pmax=max(plist)
47 #max_index=plist.index(pmax)
48 #print(plist)
49 txx = 0
50
51 if abs(plist[0]) > 0.2 :
52     txx = (plist[0])*128 # 128 # ระยะ ขอบ
53
54 print(txx)
55 ##txx = txx*1.5
56 if txx > 127 :
57     txx = 127
58 if txx < -127 :
59     txx = -127
60 #print(txx)
61 #datazzz = bytes([int(128),int(txx+127),int(0)])
62 ###print(datazzz)
63 #lcd.writeto(0x12,bytes([int(143),int(txx+127),int(0)])) # 143 #ตำแหน่งรถ
64
65 lcd.display(img)
66 lcd.draw_string(50,0, str(int(1000/(toss1-toss)))+ "%", lcd.RED, lcd.BLACK)
67 #lcd.display(img)
68
69 except KeyboardInterrupt :
70     a = kpu.deinit(task)
71
Find:
Replace with:
```

ภาพที่ 4.2 โปรแกรม MaixPy IDE สำหรับตรวจสอบความแม่นยำของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

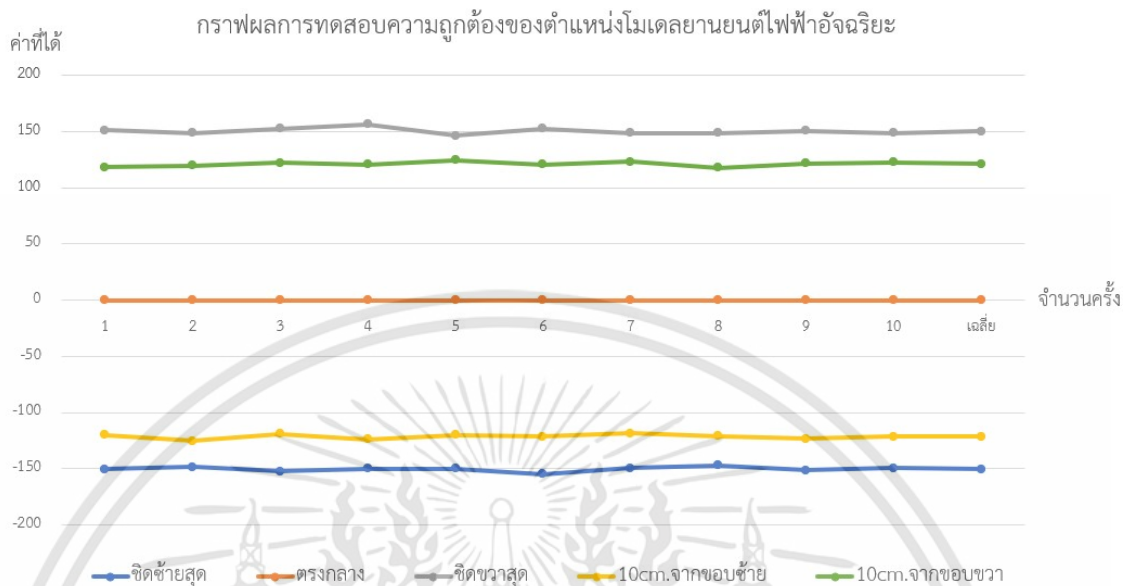


```
Serial Terminal
41.94132
42.56114
34.71007
51.23866
0
```

ภาพที่ 4.3 ค่าตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะที่แสดงผ่านโปรแกรม MaixPy IDE

ตารางที่ 4-1 ผลการทดสอบค่าความแม่นยำของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

การทดสอบครั้งที่	ค่าที่ได้ของแต่ละตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะเมื่ออยู่ในขอบทาง				
	ขีดซ้ายสุด	10 เซนจาก ขอบซ้าย	ตรงกลาง	10 เซนจาก ขอบขวา	ขีดขวาสุด
1	-150.4101	-120.2454	0	118.2654	151.0300
2	-148.7573	-125.2040	0	119.5465	148.5508
3	-152.2696	-199.2124	0	122.3759	152.4763
4	-150.2035	-123.7578	0	120.8623	156.4018
5	-149.9969	-120.0388	0	124.7811	146.2781
6	-154.9555	-121.6917	0	120.6594	152.6829
7	-149.7903	-118.3776	0	123.1154	148.9640
8	-146.8978	-120.8654	0	117.7463	148.6499
9	-151.6498	-123.4106	0	121.7610	150.7574
10	-149.4515	-121.6695	0	122.4398	148.5713
เฉลี่ย	-150.4382	-121.4473	0	121.1553	150.4362
%ค่าความแม่นยำ เฉลี่ย	99.7087 %	98.7940 %	100 %	99.0372 %	99.7100 %



ภาพที่ 4.4 กราฟผลการทดสอบค่าความแม่นยำของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

จากผลการทดสอบค่าความแม่นยำของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะเมื่อรถอยู่ตำแหน่งตรงกลางระหว่างขอบทางทั้งสอง จะมีค่าคงที่มากกว่าตำแหน่งซ้ายหรือขวา

### 4.3 การทดสอบความแม่นยำของการตรวจจับและจำแนกป้ายสัญลักษณ์และวัตถุบนถนน

การทดสอบความแม่นยำของการตรวจจับและจำแนกป้ายสัญลักษณ์และวัตถุบนถนน โดยเขียนโปรแกรมคำสั่งผ่านโปรแกรม MaixPy IDE โดยมีการตรวจสอบทั้งหมด 20 ชนิดโดยมีการแบ่งหมวดหมู่เป็น 5 หมวดหมู่ดังนี้

- Class 1 หรือ background = ถนนและขอบทาง ( มีผลต่อการเคลื่อนที่ )
- Class 2 หรือ Left = ป้ายเลี้ยวซ้าย 5 รูปทรง

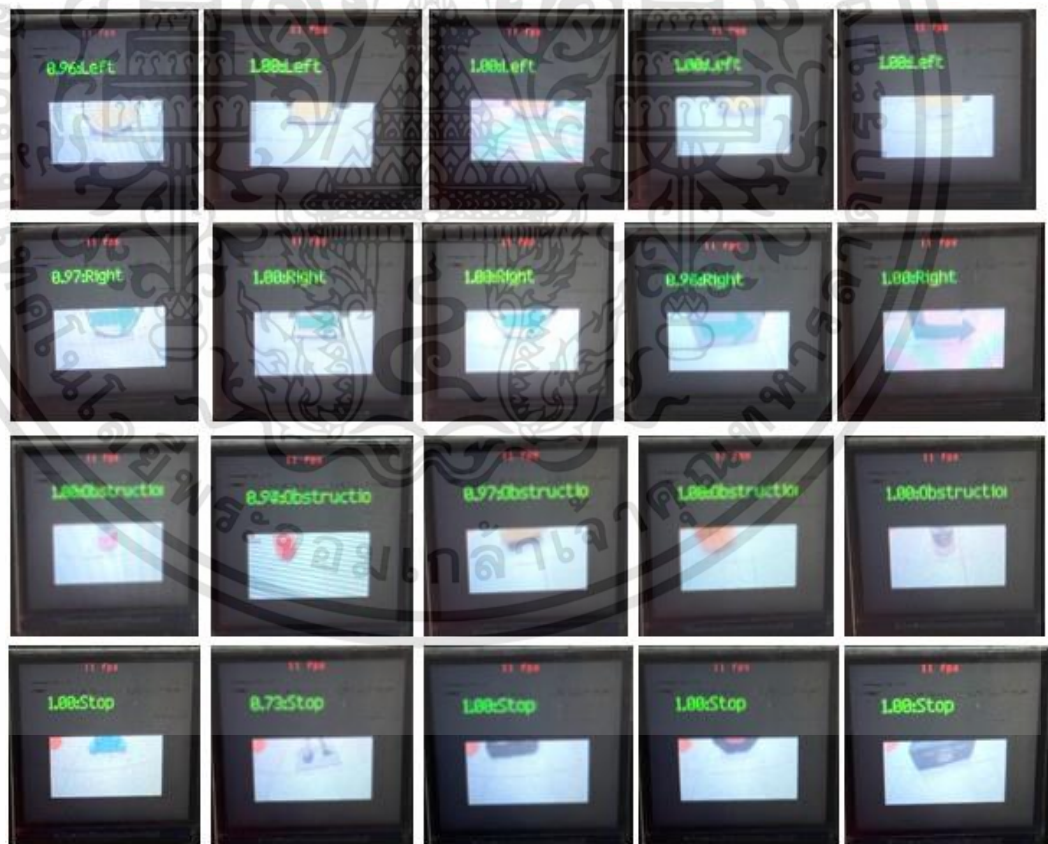
- Class 3 หรือ Obstruction = ระวังโค้ง ระวังกำแพง ระวังวิตซ่า ระวังปลูกไม้และบอล
- Class 4 หรือ Right = ป้ายเลี้ยวขวา 5 รูปทรง
- Class 5 หรือ Stop = หุ่นมนุษย์ หุ่นแมว กล้อง กล้องแว่น และป้ายหยุด

```

14 #sensor.set_flip_frames(time = 2000)
15 sensor.set_vflip(0)
16 sensor.set_hmirror(0)
17 sensor.run(1)
18
19 clock = time.clock()
20 toss1 = 0
21
22 lcd.init(type=2, freq=1500000, color=led.BLACK) #hdmi
23
24 task = kpu.load(0x300000) #hdmi+flash
25 a = kpu.set_outputs(task, 0, 1, 1)
26 labels=["Background", "Left", "Obstruction", "Right", "Stop"]
27 task2 = kpu.load(0x600000)
28 B = kpu.set_outputs(task2, 0, 1, 1, 5)
29 #try:
30
31 p_error_prev = 0.0
32 p_error = 0.0
33
34 x_ref = 0
35 speed = 100
36
37 def output_limit( input_val ):
38     if input_val > 100:
39         input_val = 100
40     if input_val < -100:
41         input_val = -100
42     return input_val
43
44 print("start")
45 devices = lcd.scan()
46 for device in devices:
47     print(device)
48 stop_ct = 0

```

ภาพที่ 4.5 โปรแกรมคำสั่งตรวจจับการเคลื่อนที่และจำแนกป้ายสัญลักษณ์และวัตถุบนถนน



ภาพที่ 4.6 ตัวอย่างการจำแนกป้ายสัญลักษณ์และวัตถุบนถนนจากหน้าจอแสดงภาพ

ตารางที่ 4-2 ผลการทดลองค่าความแม่นยำในตรวจจับป้ายสัญลักษณ์และวัตถุบนถนน

Class	ชนิด	การตรวจจับครั้งที่											
		1	2	3	4	5	6	7	8	9	10	% เฉลี่ย	
Left	ป้ายเลียวซ้าย ทรงกลม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลียวซ้าย พีวเจอร์บอร์ด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลียวซ้าย ทรงแปดเหลี่ยม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลียวซ้าย ทรงสี่เหลี่ยม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลียวซ้าย ทรงสามเหลี่ยม	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	90 %
Obstruction	กระป๋องโค้ก	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	กระป๋องวิตซ่า	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	กระป๋องกาแฟ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	กระปุกไม้	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	80 %
	กระปุกบอล	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %

ตารางที่ 4-2 ผลการทดสอบค่าความแม่นยำในการตรวจจับป้ายสัญลักษณ์และวัตถุบนถนน (ต่อ)

Class	ชนิด	การตรวจจับครั้งที่											
		1	2	3	4	5	6	7	8	9	10	% เฉลี่ย	
Right	ป้ายเลี้ยวขวา ทรงกลม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลี้ยวขวา ฟิวเจอร์บอร์ด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลี้ยวขวา ทรงแปดเหลี่ยม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลี้ยวขวา ทรงสี่เหลี่ยม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลี้ยวขวา ทรงสามเหลี่ยม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
Stop	หุ่นมนุษย์	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	กล่อง	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	กล่องแวน	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	หุ่นแมว	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายหยุด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %

จากผลการทดสอบความแม่นยำในการตรวจจับป้ายสัญลักษณ์และวัตถุบนท้องถนนตามตารางที่ 4-2 พบว่า AI มีความแม่นยำในการตรวจจับ หมวดหมู่เลียขวา(Right) หมวดหมู่หยุด(Stop) หมวดหมู่ภาพพื้นหลัง(Background) คือ 100 % หมวดหมู่เลียซ้าย(Left) คือ 98 % หมวดหมู่หลบหลีก (Obstruction)ที่มีความแม่นยำในการตรวจจับ คือ 96 % ความผิดพลาดอาจเกิดจากข้อมูลที่ทำให้การสอน AI ชนิดนั้นยังไม่ดีพอ



ภาพที่ 4.7 ตัวอย่างการตรวจจับที่ผิดพลาด

#### 4.4 ผลการทดสอบค่าความแม่นยำในการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะ

การทดสอบค่าความแม่นยำในการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะว่ามีการเคลื่อนที่ได้ตามที่ตั้งโปรแกรมคำสั่งไว้ โดยเขียนโปรแกรมคำสั่งผ่านโปรแกรม MaixPy IDE โดยมีการตรวจสอบทั้งหมด 20 ชนิดโดยมีการแบ่งหมวดหมู่เป็น 5 หมวดหมู่ดังนี้

- Class 1 หรือ background = ถนนและขอบทาง ( มีผลต่อการเคลื่อนที่ )
- Class 2 หรือ Left = ป้ายเลียซ้าย 5 รูปทรง
- Class 3 หรือ Obstruction = กระจบองค้ก กระจบองกาแพ กระจบองวิตซ่า กระจบูกไม้และบอล
- Class 4 หรือ Right = ป้ายเลียขวา 5 รูปทรง
- Class 5 หรือ Stop = หุ่นมนุษย์ หุ่นแมว กล่อง กล่องแวน และ ป้ายหยุด

```
14 #sensor.skip_frames(time = 2000)
15 sensor.set_vflip(0)
16 sensor.set_hmirror(0)
17 sensor.run(1)
18
19 clock = time.clock()
20 toss1 = 0
21
22 lcd.init(type=2, freq=15000000, color=lcd.BLACK) #ตั้งค่าจอ
23
24 task = kpu.load(0x300000) #สำหรับการทำงานของ Flash
25 a = kpu.set_outputs(task,0,1,1,1)
26 labels=['Background','Left','Obstruction','Right','Stop']
27 task2 = kpu.load(0x600000)
28 b = kpu.set_outputs(task2, 0, 1, 1, 5)
29 #try:
30
31 p_error_prev = 0.0
32 p_error = 0.0
33
34 x_ref = 0
35 speed = 100
36
37 def output_limit( input_val):
38     if input_val > 100:
39         input_val = 100
40
41     if input_val < -100:
42         input_val = -100
43
44     return input_val
45
46 print("start")
47 devices = i2c.scan()
48 for device in devices:
49     print(device)
50
51
52 stop_ct = 0
```

ภาพที่ 4.8 โปรแกรมคำสั่งการเคลื่อนที่, ตรวจสอบ, จำแนกป้ายสัญลักษณ์และวัตถุบนถนน



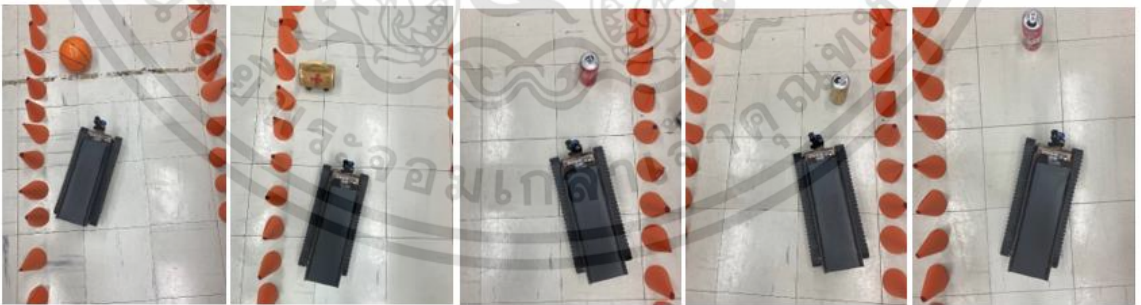
ภาพที่ 4.9 ตัวอย่างการเคลื่อนที่หมวดหมู่ Left



ภาพที่ 4.10 ตัวอย่างการเคลื่อนที่หมวดหมู่ Right



ภาพที่ 4.11 ตัวอย่างการเคลื่อนที่หมวดหมู่ Stop



ภาพที่ 4.12 ตัวอย่างการเคลื่อนที่หมวดหมู่ Obstruction

ตารางที่ 4-3 ผลการทดสอบค่าความแม่นยำในการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะบน  
สนาม

Class	ชนิด	การทดสอบการเคลื่อนที่ของรถจำลองเคลื่อนที่ได้ถูกต้องตามที่โปรแกรมครั้งที่										
		1	2	3	4	5	6	7	8	9	10	% เฉลี่ย
Left	ป้ายเลียซ้าย ทรงกลม	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	80 %
	ป้ายเลียซ้าย ฟิวเจอร์บอร์ด	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	90 %
	ป้ายเลียซ้าย ทรงแปด เหลี่ยม	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	90 %
	ป้ายเลียซ้าย ทรงสี่เหลี่ยม	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	80 %
	ป้ายเลียซ้าย ทรงสามเหลี่ยม	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
Obstruction	กระป๋องโค้ก	✓	✗	✗	✓	✗	✓	✓	✓	✓	✓	70 %
	กระป๋องวิเศษ	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	80 %
	กระป๋องกาแฟ	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	90 %
	กระปุกไม้	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	90 %
	กระปุกบอล	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	70 %

ตารางที่ 4-3 ผลการทดสอบค่าความแม่นยำในการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะบน  
สนาม (ต่อ)

Class	ชนิด	การทดสอบการเคลื่อนที่ของรถจำลองเคลื่อนที่ได้ถูกต้องตามที่โปรแกรมครั้งที่										
		1	2	3	4	5	6	7	8	9	10	% เฉลี่ย
Right	ป้ายเลี้ยวขวา ทรงกลม	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	80 %
	ป้ายเลี้ยวขวา พิวเจอร์บอร์ด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายเลี้ยวขวา ทรงแปดเหลี่ยม	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	90 %
	ป้ายเลี้ยวขวา ทรงสี่เหลี่ยม	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	90 %
	ป้ายเลี้ยวขวา ทรงสามเหลี่ยม	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	70 %
Stop	หุ่นมนุษย์	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	90 %
	กล่อง	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	กล่องแฉ่น	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	หุ่นแมว	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ป้ายหยุด	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
Background	ถนน	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %
	ขอบทาง	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	100 %

จากผลการทดสอบความแม่นยำในการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะตาม ตารางที่ 4-3 พบว่าค่าการเคลื่อนที่ถูกต้องเฉลี่ยของหมวดหมู่ซ้ายซ้าย(Left) คือ 88 % ค่าการเคลื่อนที่ ถูกต้องเฉลี่ยของหมวดหมู่หลบหลีก(Obstruction) คือ 80 % ค่าการเคลื่อนที่ถูกต้องเฉลี่ยของหมวดหมู่ เลี้ยวขวา(Right) คือ 86 % ค่าการเคลื่อนที่ถูกต้องเฉลี่ยของหมวดหมู่หยุด(Stop) คือ 98 % ค่าการ เคลื่อนที่ถูกต้องเฉลี่ยของหมวดหมู่ภาพพื้นหลัง(Background) คือ 100 %

จะเห็นได้ว่า AI มีความแม่นยำในการเคลื่อนที่ในหมวดหมู่หลบหลีก(Obstruction) น้อย ที่สุดอาจจะเป็นเพราะวัตถุมีขนาดใหญ่รถจำลองต้องทำการเคลื่อนที่หลบไปมากกว่าปกติทำให้ไม่สามารถ เคลื่อนที่กลับได้ทันจนเกิดการชนขอบทาง

#### 4.5 การทดสอบการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะในสนามจริง

โดยการทดสอบใน 1 ครั้งจะแบ่งเป็น 2 รอบ คือ รอบที่ 1 จะเริ่มจากการสุมหมวดหมู่ เคลื่อนที่ตามเส้นทางและทำการตรวจจับป้ายเลี้ยวซ้ายจนครบหนึ่งรอบและทำการตรวจจับสิ่งกีดขวาง จากการสุมที่ต้องหลบหลีก 2 ครั้ง รอบที่ 2 จะเริ่มจากการสุมหมวดหมู่เคลื่อนที่ตามเส้นทางและทำการ ตรวจจับป้ายเลี้ยวขวาจนครบหนึ่งรอบและทำการตรวจจับสิ่งกีดขวางจากการสุมที่ต้องหลบหลีก 1 ครั้ง และสิ่งทำการตรวจจับสิ่งกีดขวางที่ต้องหยุด 1 ครั้ง



ภาพที่ 4.13 การทดสอบการเคลื่อนที่ในสนามจริง

ตารางที่ 4-4 การทดสอบการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะในสนามจริง

ทดสอบครั้งที่	การชนขอบทาง	ป้ายซ้าย		ป้ายขวา		ป้ายหยุด		ป้ายหลบหลีก	
		การชน	ตรวจจับ	การชน	ตรวจจับ	การชน	ตรวจจับ	การชน	ตรวจจับ
1	ไม่ชน	ไม่ชน	✓	ไม่ชน	✓	ไม่ชน	✓	ไม่ชน	✓
2	ชน	ไม่ชน	✓	ไม่ชน	✓	ไม่ชน	✓	ชน	✓
3	ไม่ชน	ไม่ชน	✓	ไม่ชน	✓	ไม่ชน	✓	ไม่ชน	✓
4	ไม่ชน	ชน	✗	ไม่ชน	✓	ไม่ชน	✓	ชน	✓
5	ไม่ชน	ไม่ชน	✓	ไม่ชน	✓	ไม่ชน	✓	ไม่ชน	✓
%ค่าแม่นยำเฉลี่ย	80%	80%	80%	100%	100%	100%	100%	60%	100%

จากตารางที่ 4-4 จะสรุปได้ว่าโมเดลยานยนต์ไฟฟ้าอัจฉริยะเคลื่อนที่ในสนามจริงพบว่ามีค่าความแม่นยำเฉลี่ยในการชนขอบทาง คือ 80 % ค่าความแม่นยำเฉลี่ยในการตรวจจับหมวดหมู่ป้ายซ้าย (Left) และเกิดการชน คือ 80% ค่าความแม่นยำเฉลี่ยในการตรวจจับหมวดหมู่ป้ายขวา (Right) คือ 100 % และไม่เกิดการชน ค่าความแม่นยำเฉลี่ยในการตรวจจับหมวดหมู่ป้ายหยุด (Stop) คือ 100 % และไม่เกิดการชน ค่าความแม่นยำเฉลี่ยในการตรวจจับหมวดหมู่ป้ายหลบหลีก (Obstruction) คือ 100 % และเกิดการชน คือ 60 %

#### 4.6 สรุปผลการทดลอง

จากการทดลองการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะในสนามและตรวจจับป้ายสัญลักษณ์ โดยการสอน AI ผ่านโปรแกรม Colaboratory และเขียนโปรแกรมคำสั่งลงบอร์ด CorgiDude ผ่านโปรแกรม MaixPy IDE และใช้กล้อง OV2640 ในการตรวจจับภาพหน้ารถ ซึ่งจะมีการคำนวณตำแหน่งของตัวรถให้เคลื่อนที่อยู่ภายในเส้นทางและทำการตรวจจับและจำแนกป้ายสัญลักษณ์ ได้แก่ ป้ายป้ายซ้าย ป้ายป้ายขวา ป้ายหยุด และป้ายหลบหลีก

จากตารางผลการทดลองทั้งหมดจะเห็นว่า ตำแหน่งของโมเดลยานยนต์ไฟฟ้าอัจฉริยะมีความคลาดเคลื่อนเล็กน้อยแต่อยู่ในเกณฑ์ที่ยอมรับสามารถใช้งานได้จริงและการตรวจจับป้ายสัญลักษณ์เพื่อทำตามคำสั่งที่ตั้งไว้มีความถูกต้องซึ่งอาจเกิดจากที่ป้ายสัญลักษณ์แต่ละชนิดมีรูปแบบที่ไม่ตายตัว ทำให้การตรวจจับป้ายสัญลักษณ์อาจเกิดข้อผิดพลาดในการทดลอง



## บทที่ 5

### สรุปผลปัญหาและข้อเสนอแนะ

#### 5.1 สรุปผลการดำเนินงาน

จากการทดลองการเคลื่อนที่ของโมเดลยานยนต์ไฟฟ้าอัจฉริยะในสนามและตรวจจับป้ายสัญลักษณ์ โดยการสอน AI ผ่านโปรแกรม Colaboratory และเขียนโปรแกรมคำสั่งลงบอร์ด CorgiDude ผ่านโปรแกรม MaixPy IDE และใช้กล้อง OV2640 ในการตรวจจับภาพหน้ารถพบว่าเมื่อนำเข้าข้อมูลรูปภาพที่มีปริมาณมากไปสอน AI ในโปรแกรมจะใช้ระยะเวลาเพิ่มขึ้นอยู่กับจำนวนรูปที่นำมาสอนยิ่งมากก็จะยิ่งใช้เวลาในการสอนที่นานขึ้น จากผลการทดลองจะความแม่นยำของตำแหน่งโมเดลยานยนต์ไฟฟ้าอัจฉริยะ พบว่าเมื่อโมเดลยานยนต์ไฟฟ้าอัจฉริยะอยู่ชิดขอบถนนฝั่งซ้ายสุดจะได้ค่าความแม่นยำเฉลี่ย คือ 99.7087 % ชิด10เซนติเมตรจากขอบถนนฝั่งซ้ายจะได้แสดงค่าความแม่นยำเฉลี่ย คือ 98.7940 % ชิดขอบถนนฝั่งขวาสุดจะได้ค่าความแม่นยำเฉลี่ย คือ 99.7100 % ชิด10เซนติเมตรจากขอบถนนฝั่งขวาจะได้แสดงค่าความแม่นยำเฉลี่ย คือ 99.0372 % และเมื่ออยู่ตรงกลางระหว่างขอบถนนทั้งสองข้างจะแสดงค่าเป็น 0 ค่าความแม่นยำเฉลี่ย คือ 100% จากผลการทดสอบโมเดลยานยนต์ไฟฟ้าอัจฉริยะเคลื่อนที่ในสนามจริงพบว่ามีค่าความแม่นยำเฉลี่ยในการชนขอบทาง คือ 80 % ค่าความแม่นยำเฉลี่ยในการตรวจจับหมวดหมู่ซ้าย(Left) และเกิดการชน คือ 80% ค่าความแม่นยำเฉลี่ยในการตรวจจับหมวดหมู่ซ้ายขวา(Right) คือ 100 % และไม่เกิดการชน ค่าความแม่นยำเฉลี่ยในการตรวจจับหมวดหมู่หยุด(Stop) คือ 100 % และไม่เกิดการชน ค่าความแม่นยำเฉลี่ยในการตรวจจับหมวดหมู่หลบหลีก(Obstruction) คือ 100 % และเกิดการชน คือ 60 % สามารถใช้ในการเคลื่อนที่และตรวจจับป้ายสัญลักษณ์ วัตถุบนถนน ได้ค่อนข้างแม่นยำใช้งานได้จริง

## 5.2 ปัญหาที่พบและแนวทางในการแก้ปัญหา

### 5.2.1 ปัญหาที่พบ

5.2.1.1 เนื่องจากตัวกล้องมีมุมมองความกว้างน้อยทำให้การตรวจจับวัตถุอาจจะคลาดเคลื่อน

5.2.1.2 การที่ตัว model robot เป็นแบบล้อตะขาบ 2 มอเตอร์ทำให้การเลี้ยวเกิดความยากมากกว่า model robot แบบ 4 ล้อ

5.2.1.3 กล้อง OV2640 มีความละเอียดที่ 2 ล้านพิกเซล ทำให้การตรวจจับวัตถุในสถานะแสงน้อยไม่ดีเท่าที่ควร

### 5.2.2 แนวทางการแก้ไขปัญหา

5.2.2.1. ติดตั้งเซนเซอร์เพิ่มเติมรอบตัวรถทำให้ช่วยเรื่องการตรวจจับวัตถุขึ้น

5.2.2.2. ทำการติดตั้งเบรกมือเสริมเพื่อป้องกันกรณีที่รถมีแรงเฉื่อยมากจนการเบรกด้วยมอเตอร์ไม่หยุดในทันที

5.2.2.3. ติดตั้งไฟหน้ารถเพื่อเสริมประสิทธิภาพของกล้องในสถานะที่แสงน้อย

## 5.3 ข้อเสนอแนะ

5.3.1 ควรใช้กล้องที่มีความละเอียดที่สูงขึ้นเพื่อประสิทธิภาพการตรวจจับในสถานะแสงน้อย

5.3.2.. ควรใช้โมเดลที่สามารถควบคุมทิศทางการหมุนของล้อได้

5.3.3 ควรมีห้องทดลองที่มีแสงส่องสว่างที่เพียงพอ

## บรรณานุกรม

- [1] บริษัท ไอเจ็น จำกัด. (9 ธันวาคม 2564). 7 ประโยชน์ของการนำเทคโนโลยี AI มาใช้ในภาคธุรกิจ. สืบค้นเมื่อ 1 พฤษภาคม 2565, จาก <https://aigencorp.com/7-ai-benefits-for-business/>
- [2] Business Insider. (พฤษภาคม 2565). เทคโนโลยีที่สำคัญในยุคดิจิทัล: เทคโนโลยีรถยนต์ไฟฟ้าและไร้คนขับ. สืบค้นเมื่อ 1 พฤษภาคม 2565, จาก <https://www.depa.or.th/th/article-view/tech-series-electric-and-autonomous-cars>
- [3] nextsoftwarehouse. (22 พฤษภาคม 2557). ความรู้เบื้องต้นเกี่ยวกับการประมวลผลภาพดิจิทัล. สืบค้นเมื่อ 1 พฤษภาคม 2565, จาก <https://nextsoftwares.wordpress.com/2014/05/22/>
- [4] Suan Dusit University. (ม.ป.ป.). Chapter 1 Introduction to Digital Image Processing. **Introduction to Digital Image Processing**. สืบค้นเมื่อ 1 พฤษภาคม 2565, จาก [http://dusithost.dusit.ac.th/~juthawut\\_cha/download/Chapter-01-Introduction-to-Digital-Image-Processing%20\(JC\).pdf](http://dusithost.dusit.ac.th/~juthawut_cha/download/Chapter-01-Introduction-to-Digital-Image-Processing%20(JC).pdf)
- [5] GPBS Team. (21 กรกฎาคม 2563). รถยนต์อัตโนมัติแห่งอนาคต (Self-Driving Cars). สืบค้นเมื่อ 1 พฤษภาคม 2565, จาก <https://www.gpbs.co.th/self-driving-cars/>
- [6] Wikipedia. (23 มีนาคม 2565). ปัญหาประดิษฐ์. การเรียนรู้เชิงลึก. สืบค้นเมื่อ 1 พฤษภาคม 2565, จาก <https://th.wikipedia.org/wiki/การเรียนรู้เชิงลึก>
- [7] Surapong Kanoktipsatharporn. (14 มกราคม 2563). AI, Machine Learning & Deep Learning Technologies. **mobilenet-model-transfer-learning-machine-learning**. สืบค้นเมื่อ 1 พฤษภาคม 2565, จาก <https://www.bualabs.com/archives/3493/what-is-transfer-learning->

build-headless-mobilenet-model-transfer-learning-machine-learning-mobilenet-json-retrain-webcam-tensorflow-js-tfjs-ep-10/

[8] ชิตพงษ์ กิตตินราดร. (มกราคม 2563). **Convolutional Neural Network**. สืบค้นเมื่อ 1 พฤษภาคม 2565, จาก <https://guopai.github.io/ml-blog19.html>



## References

- [1] Neetika Khandelwal. (January 17, 2022). Image Processing in Python. **What is image processing?**. Retrieved May 1, 2022, from <https://neptune.ai/blog/image-processing-python>
- [2] JavaTpoint. (2022). Supervised Machine Learning. **Supervised Machine Learning**. Retrieved May 1, 2022, from <https://www.javatpoint.com/supervised-machine-learning>
- [3] Gaurav Sharma. (May 26, 2021). 5 Regression Algorithms you should know. **5 Regression Algorithms**. Retrieved May 1, 2022, from <https://www.analyticsvidhya.com/blog/2021/05/5-regression-algorithms-you-should-know-introductory-guide/>
- [4] Niklas Donges. (June 16, 2019). What Is Transfer Learning? Exploring the Popular Deep Learning Approach. **What Is Transfer Learning?**. Retrieved May 1, 2022, from <https://builtin.com/data-science/transfer-learning>
- [5] Nick Babich. (Jul 28, 2020). What Is Computer Vision & How Does it Work? An Introduction. **What is computer vision?**. Retrieved May 1, 2022, from <https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/>

## ภาคผนวก ก

### CorgiDude

#### ก.1 คู่มือการใช้งานบอร์ดคอร์กี้ดู้ด CorgiDude

CorgiDude คือบอร์ดพัฒนาไมโครคอนโทรลเลอร์ประสิทธิภาพสูงที่มีความสามารถในการนำโมเดลทางคณิตศาสตร์ วิทยาการคำนวณ หรือแม้กระทั่งโมเดลปัญญาประดิษฐ์ Machine Learning อย่างการจำแนกภาพ คั่นหาวัตถุ คั่นหาใบหน้า มาทำการประมวลผลที่บอร์ดได้อย่างรวดเร็ว สามารถเชื่อมต่อกับกล้อง จอภาพแสดงผล หรือเซนเซอร์อื่นมาต่อและเขียนโปรแกรมเพื่อควบคุมและสั่งการได้อย่างมีประสิทธิภาพ สามารถเขียนโปรแกรมหลักด้วยภาษา Python (micropython) ที่ง่ายต่อการเรียนรู้ หรือเลือกพัฒนาด้วยภาษา C/C++ เพื่อเข้าถึงประสิทธิภาพสูงสุดก็ได้เช่นกัน โดยสามารถเลือกพัฒนาได้ผลงานได้อย่างหลากหลาย เช่น

คุณสมบัติของ CorgiDude

- โมเดลจำแนกรูปภาพรูปถ่ายสิ่งของต่าง ๆ ได้ถึง 1000 ชนิด
- ทำเครื่องคัดแยกขยะ แยกกระป๋อง แก้วน้ำ
- ตรวจสอบวัตถุแปลกปลอมในสายพานลำเลียง
- ระบบคัดแยกพัสดุอัตโนมัติ
- โมเดลคั่นหาวัตถุในภาพ
- ตรวจสอบมอเตอร์ไซค์บนทางเท้า
- เครื่องไล่สุนัขหน้าบ้าน
- เครื่องนับรถการจราจรบนท้องถนน
- โมเดลคั่นหา / รู้จำใบหน้า
- ระบบเช็คชื่อ เข้างาน เข้าเรียนด้วยใบหน้า
- ระบบตรวจสอบใบหน้าที่ก่อนเปิดประตู
- ระบบรู้จำลูกค้าโดยใช้ใบหน้า
- พัฒนาโมเดล AI เอง
- รถวิ่งอัตโนมัติ หลบสิ่งกีดขวาง
- ระบบพยากรณ์ล่วงหน้าจากข้อมูลเซนเซอร์

- ระบบรู้จำเสียง สั่งงานด้วยเสียง
- ค้นหาบาร์โค้ด QR Code
- ตรวจจับสีวัตถุ ระบบประมวลผลภาพ
- เชื่อมต่อ WiFi ส่งการแจ้งเตือน ทำระบบ IoT ต่าง ๆ (เฉพาะในรุ่น WiFi)



## ภาคผนวก ข

### KB32FT และ Breakout board

KB32-FT เป็นบอร์ด Microcontroller ขนาดเล็ก ESP32 ที่มี WiFi , Bluetooth, จอ TFT LCD และอุปกรณ์อื่นๆ เช่น Acc&Gyro meter, Buzzer, Temperature sensor, Light sensor, Realtime Clock โดยตัวบอร์ด KB32-FT นั้นมีขา compatible กับบอร์ด Adafruit Feather และยังมี Breakout Board ที่ทำให้ บอร์ด KB32-FT นั้น compatible กับบอร์ด KIDBRIGHT32 V1.5 KB32-FT มีจอ TFT LCD Color 0.96 นิ้ว ใช้ Controller ST7735 ขนาด 0.96 นิ้ว ความละเอียด 80x160 Pixel ที่มีโหมดการทำงานอยู่ 2 แบบคือ

1. โหมด LED จำลองการทำงานเหมือน LED matrix ของ KIDBRIGHT
2. โหมด LCD ทำหน้าที่เป็นจอสี แสดงรูปภาพ, Animation, รวมถึงการแสดงผลกราฟฟิคต่าง ๆ ได้อย่างสวยงาม

บอร์ด KB32-FT นั้นรองรับการเขียนโปรแกรมได้อย่างหลากหลาย ทั้ง Kidbright IDE, KB-IDE, Arduino IDE, microBlock IDE, ESP-IDF, และยังสามารถรันด้วย Micropython ได้ สำหรับ Kidbright IDE และ KB-IDE มีบล็อก Plugin ให้ใช้ Download พร้อมงาน สำหรับ Arduino IDE มี Library สำหรับ จอ TFT LCD แยกได้ และสามารถใช้ KidBright IDE เขียนโปรแกรมสั่งจอ เล่นเสียง อ่านอุณหภูมิได้หมด บอร์ดมี Hardware Compatibles Layer จำลองการทำงานอุปกรณ์ต่าง ๆ โดยไม่ต้องปรับอะไรเพิ่มเติม เสมือนได้ NodeMCU ESP32 ตัวบอร์ดออกแบบมาคล้ายกับ NodeMCU ชาวทุกขาสามารถเขียนโปรแกรมสั่งการได้ ถ้าหากจะเปลี่ยนจาก KidBright IDE มาทำ Blynk กับ Arduino IDE ก็ทำได้ทันที ตัวบอร์ดออกมาพร้อมกับจอสี นอกจากจะจำลอง led dot matrix แล้ว สามารถใช้ความสามารถของจอได้อย่างเต็มที่ วาดรูปภาพ แสดงภาพเคลื่อนไหวได้ เซนเซอร์ นอกจากที่เป็นชุดเซนเซอร์ของ KidBright Compatible แล้ว ยังมีเซนเซอร์ความเร่ง ไจโร และ ตัววัดแสงเปลี่ยนเป็นตัวที่ติดตั้ง ให้การวัดแสงในย่านแสงขาวที่แม่นยำถึงแม้สภาวะแวดล้อมเปลี่ยนแปลง รองรับอุปกรณ์เสริม ด้วยการออกแบบขาเหมือนกับ Adafruit Feature

## ภาคผนวก ค

### โปรแกรม Colab

#### 1. การเริ่มต้นใช้งานโปรแกรม

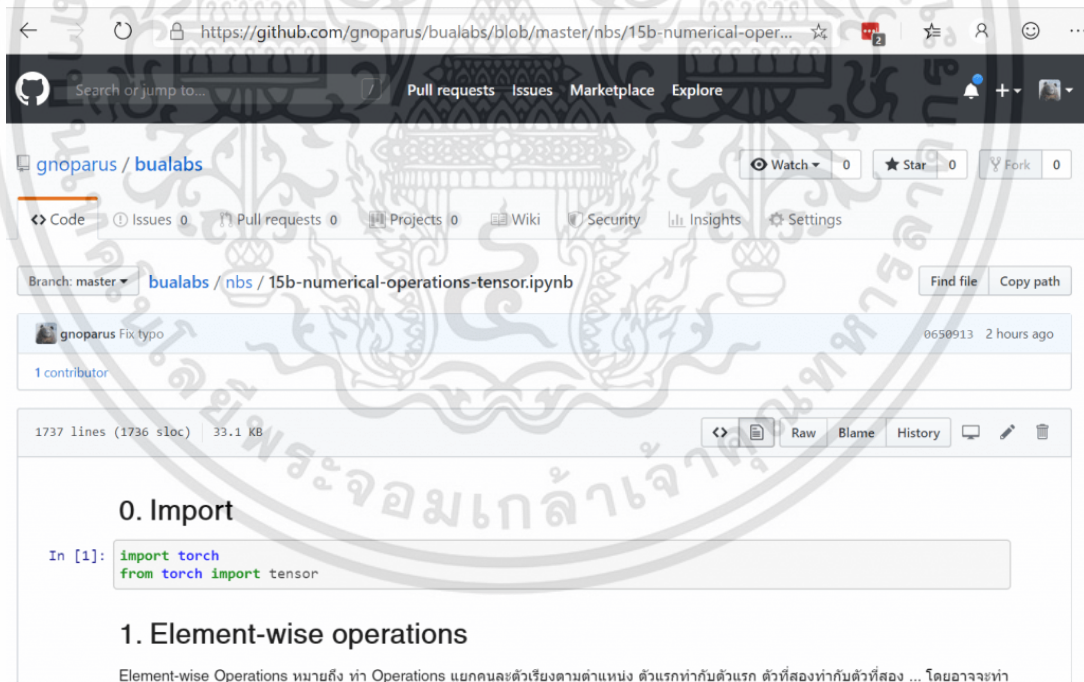
Colab คืออะไร

Google Colab ชื่อเต็มคือ Google Colaboratory เป็นบริการ Software as a Service (SaaS) โฮสต์โปรแกรม Jupyter Notebook บน Cloud จาก Google Google Colab ใช้อย่างไร เราสามารถใช้ Google Colab สร้าง Notebook เขียนโปรแกรมภาษา Python

##### 1. เข้าไปดู URL ของ Notebook ที่ GitHub

เข้าไปใน Repository ของเรา Browse ไปยัง Notebook ที่เราต้องการ แล้ว Copy URL มาเก็บไว้ก่อน สังเกตที่ตัวหนา หลัง github.com/ ตัวอย่างเช่น

<https://github.com/gnoparus/bualabs/blob/master/nbs/15b-numerical-operations->



tensor.ipynb

## 2. นำ URL มาใส่ใน โค้ด HTML

เราจะนำ URL ด้านบน ตั้งแต่ <https://github.com/> เป็นต้นไป สังเกตตัวหนา ไปใส่ไว้ในลิงค์ หลัง <https://colab.research.google.com/github/> แล้วใส่ในช่อง Web Address ได้เลย เช่น โค้ด ตัวอย่าง ดังนี้

ลิงค์ Link

```
https://colab.research.google.com/github/gnoparus/bualabs/blob/master/nbs/15b-numerical-operations-tensor.ipynb
```

ปุ่ม HTML Button

```
<a href="https://colab.research.google.com/github/gnoparus/bualabs/blob/master/nbs/15b-numerical-operations-tensor.ipynb">  
    
</a>
```

ปุ่ม Markdown

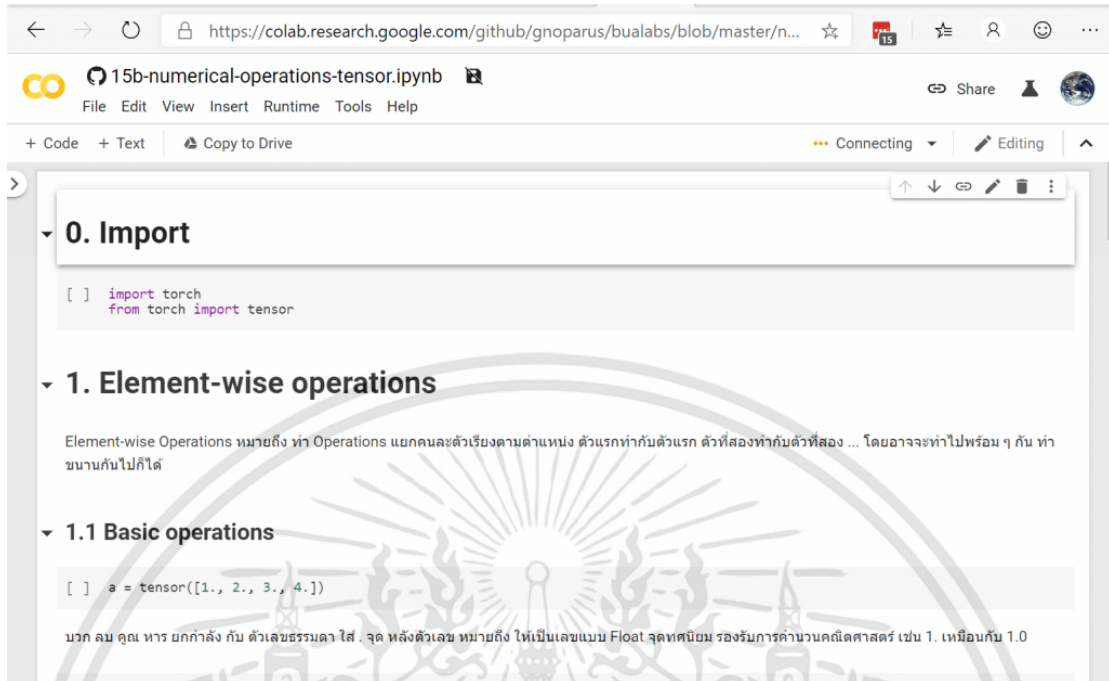
```
[[!OpenInColab](https://colab.research.google.com/assets/colabbadge.svg)](https://colab.research.google.com/github/gnoparus/bualabs/blob/master/nbs/15b-numerical-operations-tensor.ipynb)
```

## 3. Save และ Publish

Save และ Publish ขึ้น CMS ที่เราใช้จัดการเว็บไซต์อยู่ จะได้ปุ่มตั้งด้านล่าง

## 4. กดปุ่ม Open In Colab

จะเปิดหน้า Google Colab ขึ้นมา ถ้าเป็น Private Repository ก็จะให้ Login และทำการ Link Google Account และ GitHub Account ถ้าเป็น Public Repository ก็ไม่ต้องทำอะไรเลย



Google Collaps gnoparus/bualabs/nbs/15b จูปีเตอร์เน็ตบุ๊กภาพหน้าจอ อาจจะมีข้อความเตือนว่า Notebook นี้ไม่ได้พัฒนาโดย Google เป็น Notebook ของผู้อื่น โปรดใช้ ความระมัดระวังก่อน Run โปรแกรมใด ๆ

## 5. Save Notebook ไปยัง Google Drive หรือ GitHub

เมื่อเราแก้ไข ใช้งาน Jupyter Notebook เรียบร้อยแล้ว ถ้าเราต้องการจะ Save กลับไปยัง GitHub หรือ Google Drive ให้ไปที่เมนู File → Save a copy in Drive หรือ File → Save a copy to GitHub