



## ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์

นายสุรเกียรติ ทรัพย์โภม  
นายชัชพงษ์ ทับทิม

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมไฟฟ้า  
คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
ปีการศึกษา 2561

## ระบบตรวจสอบป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์

นายสุรเกียรติ ทรัพย์โฉม  
นายชัชพงษ์ ทับทิม

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมไฟฟ้า  
คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
ปีการศึกษา 2561

คณะกรรมการสอบโครงการ

..... ประธานกรรมการและอาจารย์ที่ปรึกษา

(ดร.จักรกฤษ กันทอง)

..... กรรมการ

(ดร.ปิยะสวัสดิ์ นวรัตน์ ณ อุษณา)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.อิษฎา บุญญาอรุณเนตร)

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

หัวข้อโครงการ	ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์
หน่วยกิตของโครงการ	3 หน่วยกิต
โดย	นายสุรเกียรติ ทรัพย์โภน
	นายชัชพงษ์ ทับทิม
อาจารย์ที่ปรึกษา	ดร.จักรกฤษ กันทอง
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมไฟฟ้า
ปีการศึกษา	2561

## บทคัดย่อ

ในปัจจุบันปัญญาประดิษฐ์เข้ามาเป็นส่วนสำคัญในชีวิตประจำวันของมนุษย์ สิ่งนี้ได้เปลี่ยนแปลงทุกอย่างไป เพราะเทคโนโลยีนี้คือการที่ทำให้เครื่องจักรมีพฤติกรรม และคิดอย่างมนุษย์ เอกเช่นเดียวกับ Internet of Things (IoT) ที่ได้เปลี่ยนโลกของเรา ขณะนี้ผู้จัดทำจึงคิดค้นระบบ ตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์ที่รวมเทคโนโลยีทั้งสองอย่างที่กล่าวมาไว้ด้วยกัน เพื่อช่วยพัฒนาโลกให้ก้าวไกลยิ่งขึ้นไป

ขั้นแรก ผู้จัดทำสร้างการตรวจจับป้ายทะเบียนรถยนต์โดยใช้ Tensorflow object detection API (Application Program Interface) ต่อมาจึงสร้างการรู้จำตัวอักษร โดยใช้ K Nearest Neighbors (อัลกอริทึมชนิดหนึ่งของ Machine-Learning) เพื่ออ่านหมายเลขทะเบียนรถยนต์ จากนั้นส่งหมายเหตุ ทะเบียนรถยนต์ขึ้นคลาวด์ MQTT และใช้ ESP8266 เพื่อรับข้อมูลหมายเหตุทะเบียนรถยนต์มาใช้ ควบคุมอุปกรณ์ไฟฟ้า และสุดท้ายคือการแสดงผลสถานะของอุปกรณ์และหมายเหตุทะเบียนรถยนต์ ที่อ่านได้ผ่านทาง Line Notify และเว็บแอปพลิเคชัน ซึ่งเว็บแอปพลิเคชันได้ออกแบบเพื่อให้สามารถ ควบคุมอุปกรณ์ไฟฟ้าได้อีกด้วย

โดยสรุป ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์มีผลลัพธ์ที่ยอดเยี่ยมในการ ตรวจจับป้ายทะเบียนรถยนต์ การควบคุมอุปกรณ์ไฟฟ้า และการแสดงผลสถานะ แต่ในการอ่านป้าย ทะเบียนรถยนต์ยังคงมีความผิดพลาดในการอ่านหมายเหตุทะเบียนอยู่บ้าง ทั้งนี้เนื่องจากหลายปัจจัย ซึ่งสามารถนำไปแก้ไขและพัฒนาได้ต่อไป

## กิตติกรรมประกาศ

โครงการเรื่องระบบตรวจจับป้ายทะเบียนรถยนต์โดยบัญญาประดิษฐ์สำเร็จลุล่วงไปได้ด้วยดี ผู้จัดทำต้องขอบพระคุณ ดร.จักรกฤษ กันทอง อาจารย์ประจำภาควิชาสารวิทยา ไฟฟ้ามหा�วิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และอาจารย์ที่ปรึกษาโครงการชั้นนี้ ซึ่งท่านเคยช่วยเหลือแนะนำทางให้คำปรึกษา อธิบายองค์ความรู้ต่าง ๆ ในการทำงาน ตลอดจนช่วยแนะนำแนวทางแก้ไขปัญหาข้อบกพร่องต่าง ๆ ทุกขั้นตอนของการจัดทำโครงการ ขณะผู้จัดทำขอบพระคุณเป็นอย่างสูง

นอกจากนี้ขอรบกวนบิดามารดา ที่เป็นผู้ให้กำลังใจ และให้โอกาสการศึกษาอันมีค่ายิ่งขอบคุณเพื่อน ๆ และอาจารย์ทุกท่าน ตลอดจนผู้ที่เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวไว้ใน ณ ที่นี่ ที่มีส่วนช่วยเหลือให้โครงการฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ ขณะผู้จัดทำโครงการตรวจจับป้ายทะเบียนรถยนต์โดยบัญญาประดิษฐ์ หวังว่า โครงการชั้นนี้จะเป็นประโยชน์แก่ผู้สนใจต่อไป

## สารบัญ

### หน้า

บทคัดย่อ	๑
กิตติกรรมประกาศ	๒
สารบัญ	๓
รายการตาราง	๔
รายการรูปประกอบ	๕
<b>บทที่ 1 บทนำ</b>	
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ในการทำโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนและแผนการดำเนินงาน	1
1.5 ผลที่คาดว่าจะได้รับ	2
<b>บทที่ 2 ทฤษฎีที่เกี่ยวข้อง</b>	
2.1 ปัญญาประดิษฐ์ (Artificial Intelligence)	3
2.1.1 ประเภทของปัญญาประดิษฐ์	3
2.2 Machine Learning	3
2.2.1 ประเภทของ Machine Learning	4
2.2.2 k Nearest Neighbor (kNN)	6
2.3 Artificial Neural Network	8
2.3.1 ส่วนประกอบของ Artificial Neural Network	9
2.3.2 ส่วนประกอบของโหนด	9
2.3.3 หลักการทำงาน	10
2.3.4 Back propagation	11
2.3.5 Convolutional Neural Networks (CNN)	12
2.3.6 MobileNet	14
2.3.7 Single Shot Multibox Detector (SSD)	15
2.4 NodeMCU ESP8266	17
2.5 Computer Vision	17
2.6 Image Processing	18

## หน้า

2.6.1 Color Space	19
2.6.2 Morphological	20
2.6.3 Thresholding	23
2.6.4 Smoothing Image	23
2.6.5 Contour	23
<b>2.7 การเข้ารหัสและถอดรหัสตัวอักษร</b>	<b>24</b>
2.7.1 ASCII Code	24
2.7.2 Unicode	24
<b>2.8 ภาษาไทยตอน</b>	<b>25</b>
<b>2.9 Anaconda Distribution</b>	<b>25</b>
<b>2.10 Jupyter Notebook</b>	<b>25</b>
<b>2.11 Tensorflow</b>	<b>25</b>
<b>2.12 Keras</b>	<b>26</b>
<b>2.13 OpenCV</b>	<b>26</b>
<b>2.14 Internet of Things (IoT)</b>	<b>26</b>
2.14.1 การประยุกต์ใช้ Internet of Things (IoT)	26
<b>2.15 CloudMQTT</b>	<b>27</b>
2.15.1 MQTT (Message Queue Telemetry Transport)	27
2.15.2 CloudMQTT	28
2.15.3 โปรโตคอล	28
<b>บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน</b>	
<b>3.1 การออกแบบ</b>	<b>29</b>
3.1.1 การออกแบบการทำงานของระบบตรวจจับป้ายทะเบียนรถยนต์โดย ปัญญาประดิษฐ์	29
3.1.2 การออกแบบโปรแกรมตรวจจับป้ายทะเบียนรถยนต์	30
3.1.3 การออกแบบโปรแกรมอ่านป้ายทะเบียนรถยนต์	33
3.1.4 การออกแบบโปรแกรมส่งข้อมูลขึ้นคลาวด์	48
3.1.5 การออกแบบโปรแกรมควบคุมอุปกรณ์ไฟฟ้า และการแสดงผล	40
<b>3.2 การฝึกฝนปัญญาประดิษฐ์โปรแกรมตรวจจับป้ายทะเบียนรถยนต์</b>	<b>56</b>
3.2.1 ขั้นตอนการฝึกฝนปัญญาประดิษฐ์	57

## หน้า

3.3 การฝึกฝนปัญญาประดิษฐ์โปรแกรมอ่านป้ายทะเบียนรถยนต์	62
3.3.1 ขั้นตอนการฝึกฝน kNN	62
<b>บทที่ 4 การทดลองและการทดลอง</b>	
4.1 การทดลองตรวจจับป้ายทะเบียนรถยนต์	67
4.1.1 จุดประสงค์ของการทดลอง	67
4.1.2 ขั้นตอนการทดลอง	67
4.1.3 รูปภาพที่ใช้ในการทดสอบ	68
4.1.4 ผลการทดลอง	69
4.2 การทดลองอ่านป้ายทะเบียนรถยนต์	70
4.2.1 จุดประสงค์ของการทดลอง	70
4.2.2 ขั้นตอนการทดลอง	70
4.2.3 รูปภาพที่ใช้ในการทดสอบ	71
4.2.4 ผลการทดลอง	71
4.3 การทดลองความคุณอุปกรณ์ไฟฟ้าและการแสดงผล	72
4.3.1 จุดประสงค์ของการทดลอง	72
4.3.2 ขั้นตอนการทดลอง	72
4.3.3 ผลการทดลอง	73
<b>บทที่ 5 สรุปผลและวิจารณ์การทดลอง</b>	
5.1 สรุปผลการทดลอง	75
5.1.1 การทดลองตรวจจับป้ายทะเบียนรถยนต์	75
5.1.2 การทดลองอ่านป้ายทะเบียนรถยนต์	75
5.1.3 การทดลองความคุณอุปกรณ์และการแสดงผล	75
5.2 วิจารณ์การทดลอง	76
5.2.1 ปัญหาและอุปสรรค	76
5.2.2 แนวทางการพัฒนา	76
เอกสารอ้างอิง	77
ภาคผนวก	79
ก. โปรแกรมของระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์	80
ก. 1. โปรแกรมอ่านป้ายทะเบียนรถยนต์	81

**หน้า**

ก. 2. โปรแกรมสำหรับเรียกใช้ระบบตรวจจับป้ายทะเบียนรถยนต์โดย บัญญาประดิษฐ์	102
ก. 3. โปรแกรมอื่น ๆ	110
ก. 4. โปรแกรมควบคุมอุปกรณ์ไฟฟ้าด้วย ESP8266	123
ก. 5. โปรแกรมเว็บแอพพลิเคชั่น	127

**รายการตาราง****หน้า****ตารางที่**

1.1 แผนการดำเนินงาน	2
4.1 ผลการทดลองตรวจจับป้ายทะเบียนรถยนต์	69
4.2 ผลการทดลองการอ่านป้ายทะเบียนรถยนต์โดย $k = 1, 2$ และ $3$	71

## รายการรูปประกอบ

หน้า

### รูปที่

2.1 การเปรียบเทียบ Traditional Programming กับ Machine Learning	4
2.2 ตัวอย่าง Supervised Learning	5
2.3 ตัวอย่าง Reinforcement Learning	5
2.4 ประเภทต่าง ๆ ของ Machine Learning และ การนำไปประยุกต์ใช้งาน	6
2.5 ขั้นตอนกระบวนการของ kNN	7
2.6 โครงสร้างของเซลล์ประสาท	8
2.7 โครงสร้างของ Artificial Neural Network	9
2.8 หลักการทำงานของ Artificial Neural Network	10
2.9 Back propagation	11
2.10 โครงสร้าง Convolutional Neural Networks	12
2.11 Pooling stage	13
2.12 สรุปกระบวนการของ Convolutional neural network	14
2.13 การประมวลผลของ CNN ทั่วไป	14
2.14 Depthwise separable convolutions	15
2.15 Pointwise Convolution Filter	15
2.16 โครงสร้างของ Single Shot Multibox Detector	16
2.17 NodeMCU Version 2 (ESP8266-12E)	17
2.18 การแยกตัวในรูปภาพออกมาเป็นกลุ่มสี โดยใช้เทคนิค Image Segmentation	18
2.19 การติดตามภาพบุคคล โดยการใช้เทคนิค Tracking	18
2.20 ระบบสี Grayscale	19
2.21 การเปรียบเทียบระหว่าง RGB (ชั้น) กับ HSV (ช่วง)	20
2.22 การเปรียบเทียบรูปภาพตื้นฉบับ (ชั้น) กับรูปภาพ Erosion (ช่วง)	20
2.23 การเปรียบเทียบรูปภาพตื้นฉบับ (ชั้น) กับรูปภาพ Dilation (ช่วง)	21
2.24 การเปรียบเทียบรูปภาพตื้นฉบับ (ชั้น) กับรูปภาพ Opening (ช่วง)	21
2.25 การเปรียบเทียบรูปภาพตื้นฉบับ (ชั้น) กับรูปภาพ Closing (ช่วง)	21
2.26 การเปรียบเทียบรูปภาพตื้นฉบับ (ชั้น) กับรูปภาพ Gradient (ช่วง)	22

## รูปที่

2.27 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Top Hat (ขวา)	22
2.28 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Black Hat (ขวา)	22
2.29 Histogram	23
2.30 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Smoothing (ขวา)	23
2.31 การเปรียบเทียบการเขียนพิกัดทั้งหมด (ซ้าย) กับการเขียนพิกัดโดยประมาณ (ขวา)	24
2.32 จำนวนอุปกรณ์ IoT ที่มีการเชื่อมต่อทั่วทั้งโลก	27
2.33 แผนภาพแสดงการเชื่อมต่อของ NodeMCU ผ่าน Cloud MQTT	28
3.1 แผนผังการทำงานของระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์	29
3.2 ขั้นตอนการทำงานของโปรแกรมตรวจจับป้ายทะเบียนรถยนต์	30
3.3 การนำเข้าไลบรารี และโมดูลต่าง ๆ	31
3.4 การกำหนดตำแหน่งของโมเดลและการระบุจำนวนคลาส	31
3.5 การกำหนดฟังก์ชันแปลงรูปภาพเป็น numpy array	31
3.6 การกำหนดฟังก์ชันบันทึกรูปภาพป้ายทะเบียนรถยนต์	32
3.7 การโหลด Tensorflow model	32
3.8 การแปลง object-detection.pbtxt ให้เป็นหมวดหมู่ดังนี้เพื่อผลลัพธ์ของการทำนาย	32
3.9 กำหนดตำแหน่งไฟล์รูปภาพที่ใช้ทดสอบ และการกำหนด count	32
3.10 เริ่มให้โปรแกรมตรวจจับป้ายทะเบียนรถยนต์	33
3.11 crop ป้ายทะเบียนรถยนต์	33
3.12 ขั้นตอนการทำงานของโปรแกรมอ่านป้ายทะเบียนรถยนต์	34
3.13 การนำเข้าไลบรารีที่ต้องใช้	35
3.14 การกำหนดตัวแปร	35
3.15 โค้ดในฟังก์ชัน main	35
3.16 โค้ดในฟังก์ชัน main	36
3.17 การนำเข้าไลบรารี และโมดูลต่าง ๆ	36
3.18 การกำหนดตัวแปรต่าง ๆ	37
3.19 ฟังก์ชัน loadKNNDataAndTrainKNN	37
3.20 ฟังก์ชันย่อย findPossibleCharsInPlate	38
3.21 กำหนดฟังก์ชันย่อย checkIfPossibleChar	38

## รูปที่

3.22 พิจารณาห่างของตัวอักษร distanceBetweenChars	39
3.23 พิจารณาหักมุม angleBetweenChars	39
3.24 พิจารณาหักมุม findListOfMatchingChars	40
3.25 พิจารณาหักมุม findListOfListsOfMatchingChars	41
3.26 พิจารณาหักมุม removeInnerOverlappingChars	41
3.27 พิจารณาหักมุม recognizeCharsInPlate	42
3.28 โค้ดในพิจารณาหักมุม detectCharsInPlates	43
3.29 โค้ดในพิจารณาหักมุม detectCharsInPlates	43
3.30 โค้ดในพิจารณาหักมุม detectCharsInPlates	43
3.31 โค้ดในพิจารณาหักมุม detectCharsInPlates	44
3.32 โค้ดในพิจารณาหักมุม detectCharsInPlates	44
3.33 การนำเข้าไฟล์แลกเปลี่ยนข้อมูลตัวแปร	44
3.34 พิจารณาหักมุม preprocess	45
3.35 พิจารณาหักมุม extractValue	45
3.36 พิจารณาหักมุม extractValue	46
3.37 การนำเข้าไฟล์ที่ต้องใช้	46
3.38 Class Possible	47
3.39 แผนผัง Flow Chart การทำงานของโปรแกรมส่งข้อมูลขึ้นคลาวด์	48
3.40 กำหนดตัวแปร	49
3.41 การเชื่อมต่อคลาวด์ และส่งหมายเลขอหะเปลี่ยนขึ้นคลาวด์	49
3.42 แผนผัง Flowchart การทำงานของโปรแกรมควบคุมอุปกรณ์ไฟฟ้า	50
3.43 การตั้งค่า Wi-Fi และกำหนดตัวแปร	51
3.44 การตั้งค่า MQTT	51
3.45 กำหนดค่าเริ่มต้นของโปรแกรม	51
3.46 การทำงานวนซ้ำเพื่อเชื่อมต่อคลาวด์	52
3.47 โค้ดการอ่านข้อมูลจาก Topic	52
3.48 การสร้าง Flow เพื่อทำ Line Notification	53
3.49 การออกแบบหน้าหลักของเว็บไซต์	53
3.50 การออกแบบหน้าหลักของเว็บไซต์	53

## รูปที่

3.51 การตั้งค่า MQTT และการสร้าง Client สำหรับการเขื่อมต่อ MQTT Server	54
3.52 การรับข้อมูลจาก MQTT Server เข้ามาแสดงผลผ่านเว็บไซต์	55
3.53 โปรแกรมคำสั่งของปุ่มกด ON และปุ่มกด OFF	55
3.54 การสร้างฟังก์ชัน MQTT_Send	55
3.55 โครงสร้างของเว็บไซต์	56
3.56 โมเดลต่าง ๆ ที่ถูกฝึกฝนโดย COCO Dataset	57
3.57 หน้าเว็บ <a href="https://github.com/tensorflow/models">https://github.com/tensorflow/models</a>	58
3.58 การใช้คำสั่ง git clone	58
3.59 การ Activate Environments และการซึ่งดำเนินการไปที่ model object detection	59
3.60 การตั้ง PYTHONPATH environment variable	59
3.61 การทำ Annotation	60
3.62 การแก้ไขไฟล์ generate_tfrecord.py	61
3.63 การสร้างไฟล์ object-detection.pbtxt	61
3.64 รูปภาพสำหรับฝึกฝน	62
3.65 การนำเข้าไลบรารีต่าง ๆ	63
3.66 การกำหนดค่าวarezที่ต้องใช้	63
3.67 โค้ดในฟังก์ชัน main	63
3.68 โค้ดในฟังก์ชัน main	64
3.69 โค้ด for loop ในฟังก์ชัน main	64
3.70 โค้ด for loop ในฟังก์ชัน main	65
3.71 โค้ด for loop ในฟังก์ชัน main	65
3.72 โค้ดในฟังก์ชัน main	65
3.73 โค้ดในฟังก์ชัน main	66
4.1 การกำหนด minimum score	68
4.2 รันไฟล์ numplate_recognition_detection_Images.py โดยไม่เรียกใช้การอ่านป้ายทะเบียน	68
4.3 รูปภาพที่ใช้ในการทดลองตรวจจับป้ายทะเบียนรถยนต์	68
4.4 การกำหนดค่า k	70
4.5 รูปภาพที่ใช้ในการทดลองอ่านป้ายทะเบียนรถยนต์	71

**รูปที่**

4.6 ผลการทดสอบเมื่อป้ายทะเบียนอินพุตเป็น 4 กง 87	73
4.7 ผลการทดสอบเมื่อป้ายทะเบียนอินพุตเป็น 1 กฎ 4786	73
4.8 ผลการทดสอบควบคุมรีเลย์ ON ผ่านเว็บแอปพลิเคชัน	74
4.9 ผลการทดสอบควบคุมรีเลย์ OFF ผ่านเว็บแอปพลิเคชัน	74

## บทที่ 1

### บทนำ

#### **1.1 ที่มาและความสำคัญ**

ในปัจจุบัน ปัญญาประดิษฐ์เข้ามามีบทบาทสำคัญในชีวิตประจำวัน และเปลี่ยนแปลงทุกอย่างไป เพราะปัญญาประดิษฐ์คือเทคโนโลยีที่เครื่องจักรถูกออกแบบ และโปรแกรม ให้มีความคิดและความสามารถเหมือนมนุษย์ เคลก เช่นเดียวกับ Internet of Things (IoT) ที่ทำให้อุปกรณ์ต่าง ๆ สามารถติดต่อสื่อสารกันได้ เทคโนโลยีเหล่านี้ได้เปลี่ยนแปลงโลกให้ก้าวถัดขึ้น ผู้จัดทำจึงสร้างระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์ที่รวมเทคโนโลยีปัญญาประดิษฐ์และ IoT เข้ามาไว้ด้วยกัน เพื่อใช้ตรวจจับป้ายทะเบียนรถยนต์ อ่านป้ายทะเบียนรถยนต์ภาษาไทย และสามารถนำข้อมูลที่ได้จากการประมวลผลปัญญาประดิษฐ์ไปใช้ควบคุมอุปกรณ์ไฟฟ้าด้วยระบบ IoT

ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์สามารถนำไปประยุกต์ใช้ได้หลายด้าน เช่น การนำไปใช้เพื่อลดพนักงานจ่ายบัตรตามห้างสรรพสินค้า การนำไปใช้ตามรถที่ถูกจอดรถ การนำไปใช้กับสมาร์ทโฟน เป็นต้น

#### **1.2 วัตถุประสงค์ในการทำงาน**

1. ออกแบบและสร้างระบบตรวจจับและจดจำป้ายทะเบียนรถยนต์ภาษาไทย
2. ใช้ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์ในการควบคุมอุปกรณ์ไฟฟ้าด้วยระบบ IoT

#### **1.3 ขอบเขตของโครงการ**

ศึกษาการสร้าง Artificial Intelligence และ Machine Learning เพื่อนำมาออกแบบและสร้างระบบระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์

#### **1.4 ขั้นตอนและแผนการดำเนินงาน**

ขั้นตอนและแผนการดำเนินงานแสดงในตารางที่ 1.1 ดังนี้

## ตารางที่ 1.1 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ส.ค.-ก.ป.	ต.ค.-พ.ย.	ธ.ค.-ม.ค.	ก.พ.-มี.ค.	เม.ย.-พ.ค.
1. ประชุมและวางแผนการทำงาน					
2. ศึกษา AI และ Machine Learning					
3. จัดเตรียมอุปกรณ์ที่เกี่ยวข้อง					
4. ออกแบบ และสร้างระบบระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์					
5. ทดสอบ วิเคราะห์จากการใช้งานจริง ปรับปรุงโมเดล และประเมินผลของโครงการ					

### 1.5 ผลที่คาดว่าจะได้รับ

- สามารถตรวจจับและจดจำป้ายทะเบียนรถยนต์ที่เป็นภาษาไทยได้
- นำผลที่ได้จากการตรวจจับมาใช้ในการควบคุมอุปกรณ์ไฟฟ้าด้วยระบบ IoT ได้

## บทที่ 2

### ทฤษฎีเกี่ยวข้อง

#### **2.1 ปัญญาประดิษฐ์ (Artificial Intelligence) [2]**

AI หรือปัญญาประดิษฐ์ เป็นการสร้างและจำลองกระบวนการคิดที่คล้ายกับความคิดของมนุษย์หรือความคลาดเดียนที่สร้างขึ้นให้กับสิ่งไม่มีชีวิตหรือเครื่องจักร โดยเฉพาะกับระบบคอมพิวเตอร์ ให้สามารถเรียนรู้ การใช้และให้เหตุผล การแก้ไขหรือปรับเปลี่ยนแนวความคิดของระบบเครื่องเองได้ เพื่อใช้ในการช่วยเหลือหรือตอบสนองความต้องการของมนุษย์ในด้านต่าง ๆ

##### **2.1.1 ประเภทของปัญญาประดิษฐ์**

ประเภทของปัญญาประดิษฐ์ สามารถแบ่งออกเป็น 4 ประเภทดังนี้

###### **1. Systems that think like humans**

ระบบที่สามารถคิดแบบเดียวกับมนุษย์ ซึ่งมีความพยายามที่จะทำให้ระบบคอมพิวเตอร์ หรือเครื่องจักร ให้มีปัญญาสามารถคิดได้หรือช่วยตัดสินใจและแก้ปัญหา

###### **2. Systems that act like humans**

ระบบที่มีการกระทำเหมือนกับมนุษย์ ซึ่งจะทำให้คอมพิวเตอร์ หรือเครื่องจักรนั้นสามารถทำในสิ่งที่มนุษย์สามารถทำได้

###### **3. Systems that think rationally**

ระบบที่คิดอย่างมีเหตุผล ซึ่งปัญญาประดิษฐ์จะใช้โมเดลในการคำนวณเพื่อศึกษาวิธีการคำนวณที่สามารถรับรู้ได้ ใช้เหตุผล หลักตรรกศาสตร์ร่วมด้วย เพื่อสร้างสติปัญญาให้กับคอมพิวเตอร์ หรือเครื่องจักร

###### **4. Systems that act rationally**

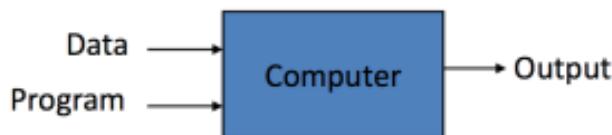
ระบบที่กระทำอย่างมีเหตุผล เป็นการศึกษาเกี่ยวกับการออกแบบระบบที่สามารถกระทำการเพื่อให้บรรลุเป้าหมายที่ตั้งไว้ได้

#### **2.2 Machine Learning**

Machine Learning คือ การเรียนรู้ของคอมพิวเตอร์ หรือจะเปรียบง่าย ๆ คือ Machine Learning คือส่วนสมองของปัญญาประดิษฐ์ โดยสมัยก่อนการที่จะทำโปรแกรมคอมพิวเตอร์ขึ้นมา ผู้พัฒนาโปรแกรมจะใช้วิธีที่เรียกว่า Traditional Programming ในการป้อนคำสั่งต่าง ๆ ให้โปรแกรมวิธีการเขียนโปรแกรมแบบนี้มีข้อจำกัดหลายอย่าง เช่น จะมีภาษาที่ซับซ้อนและจำเป็นต้องใช้ความละเอียดรอบคอบอย่างมากในการเขียนโปรแกรม ตัวอย่างเช่น ผู้พัฒนาโปรแกรมต้องการทำโปรแกรมเพื่อวินิจฉัยโรค ให้เลือดออก ผู้พัฒนาจะใช้การเขียนเชิงตรรกะ ป้อนคำสั่งเป็นขั้นตอน เช่น ผู้ป่วยมีอาการเป็นไข้หรือไม่ ถ้าเป็นไข้แล้วมีอาการเป็นไข้漫านแค่ไหน ถ้าเป็นไข้มาหลายวันแล้วมีคุณภาพตัว

หรือไม่ ถ้ามีคุณแม่ล้วมีอาการเลือดออกหรือไม่ จะใช้การเขียนเชิงตรรกะแบบตัวอย่างไปเรื่อย ๆ จนถึงผลลัพธ์ที่ได้ว่าเป็นไข้เลือดออก แต่การใช้ Machine Learning จะทำโดยการที่ผู้พัฒนาโปรแกรมใส่ข้อมูลเข้าไปให้โปรแกรมคอมพิวเตอร์เรียนรู้เอง โดยข้อมูลที่ใส่เข้าไปจะเป็นอาการของโรคไข้เลือดออก จากนั้นคอมพิวเตอร์จะทำการเรียนรู้จากข้อมูลที่เราใส่เข้าไปให้อง Machine Learning จึงมีข้อดีคือไม่ต้องใช้ภาษาที่วันซ้อนมากในการทำให้คอมพิวเตอร์เรียนรู้ แต่มีข้อเสียก็คือต้องมีปริมาณข้อมูลในการนำไปให้คอมพิวเตอร์เรียนรู้มากพอสมควรถึงจะได้ผลลัพธ์ที่มีความแม่นยำ

### Traditional Programming



### Machine Learning



รูปที่ 2.1 การเปรียบเทียบ Traditional Programming กับ Machine Learning [18]

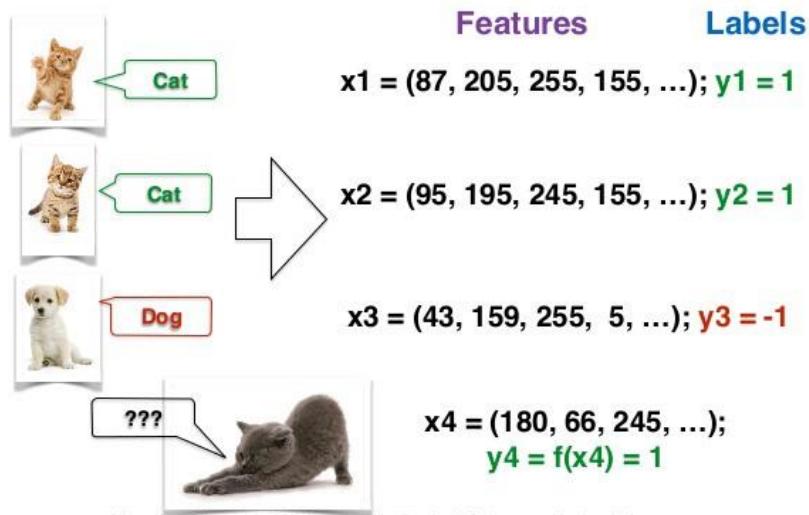
จากรูปที่ 2.1 สามารถสรุปได้ว่า Machine Learning คือ วิธีการที่ผู้พัฒนาโปรแกรมไม่ได้ใส่ Program คำตอบเข้าไป แต่ใส่ข้อมูลและผลลัพธ์เพื่อให้คอมพิวเตอร์หาความสัมพันธ์แล้วสร้างโปรแกรมเพื่อที่จะนำไปใช้ในการตอบคำถามในอนาคตว่าถ้า input เป็นแบบนี้ output จะเป็นอย่างไร

#### 2.2.1 ประเภทของ Machine Learning

##### 1. Supervised Learning

Supervised Learning คือ การเรียนรู้ที่ผู้พัฒนาโปรแกรมจะเตรียมข้อมูลที่มีทั้ง input และ output ที่ถูกต้องไว้อยู่แล้วมาใช้ในการสอนให้โปรแกรมทำความเข้าใจความสัมพันธ์ของ input และ output นั้น เช่น การสอนให้โมเดลสามารถแยกแยะได้ว่ารูปที่ป้อนเข้าไปเป็นรูปของอะไร ผู้พัฒนาโปรแกรมจะต้องเตรียมคำตอบของแต่ละรูปไว้ก่อน เพื่อนำไปใช้สอนให้โมเดลเข้าใจ และมีความแม่นยำก่อนนำไปใช้งานจริง การเรียนรู้วิธีนี้มีข้อเสียที่ต้องมีปริมาณข้อมูลจำนวนมากในการใช้สอนโมเดล

# Supervised Learning



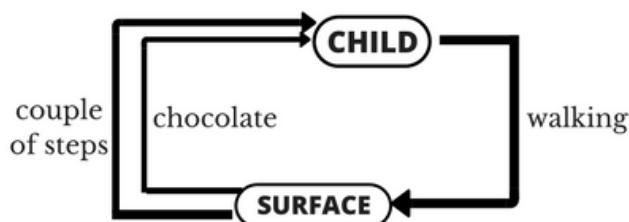
รูปที่ 2.2 ตัวอย่าง Supervised Learning [18]

## 2. Unsupervised Learning

Unsupervised Learning คือ การเรียนรู้ที่ผู้พัฒนาโปรแกรมไม่ต้องเตรียมข้อมูลที่มี input และ output ที่ถูกต้องไว้ โดยเดลจะทำการเรียนรู้จากรูปแบบความใกล้เคียงกันของข้อมูลและจัดประเภทให้อยู่ด้วยกันเอง การเรียนรู้ด้วยวิธีนี้มีข้อเสียคือ ประสิทธิภาพและบางทีจะไม่สามารถทราบหรือเข้าใจได้ว่าผลลัพธ์ที่ไม่เดลแบ่งออกมายังไงใช้เกณฑ์อะไรในการตัดสินใจ

## 3. Reinforcement Learning

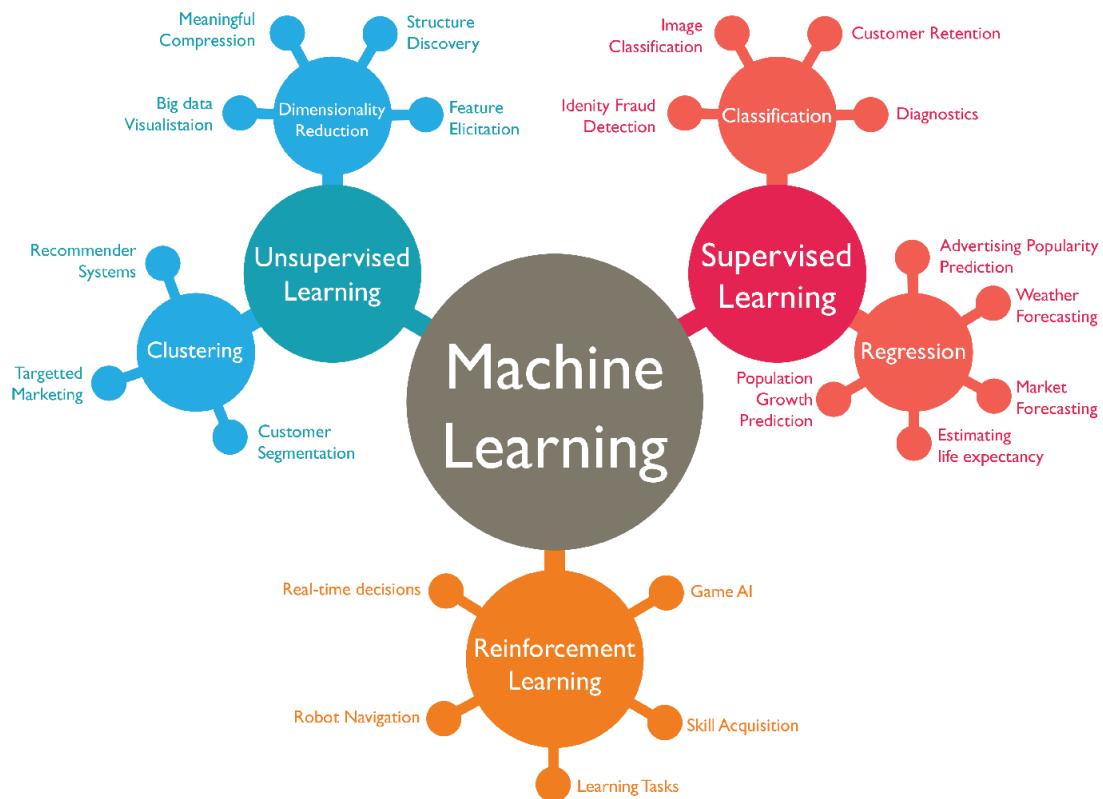
Reinforcement Learning คือ การเรียนรู้ที่จะมีการปรับเปลี่ยนไปตามสภาพแวดล้อม



รูปที่ 2.3 ตัวอย่าง Reinforcement Learning [18]

จากรูปที่ 2.3 สามารถอธิบายได้ว่าการที่เด็กจะเดินໄได้ประกอบไปด้วยองค์ประกอบหลายอย่าง เช่น การทึบนำหนักตัว ฝ่าเท้าเด็กเหยียบพื้นเต็มที่แล้ว ไหน ระดับความแตกต่างของพื้นบริเวณที่ทำการหัดเดิน และอื่น ๆ อีกมาก many การหัดเดินแบบนี้เป็นสิ่งที่ยากที่จะสอนเด็ก สิ่งที่ทำได้คือให้เด็ก

หัดเดินเอง ถ้าเด็กสามารถเดินได้ ผู้ใหญ่ก็จะให้รางวัลเด็กเพื่อเป็นสิ่งที่บอกร่วมกับสิ่งที่เด็กทำเป็นสิ่งที่ถูกต้องแล้ว ถ้าเด็กเดินแล้วล้มก็จะไม่ให้รางวัลเด็ก เพื่อเป็นการให้เด็กเรียนรู้ว่าสิ่งนี้คือผิด



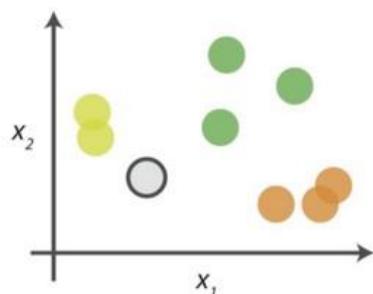
รูปที่ 2.4 ประเภทต่าง ๆ ของ Machine Learning และ การนำไปประยุกต์ใช้งาน [18]

### 2.2.2 k Nearest Neighbor (kNN)

k Nearest Neighbor (kNN) คือ อัลกอริทึมหนึ่งประเภท Supervised Learning ที่ใช้สำหรับ Classification (การแยกแยะวัตถุ) กระบวนการของ kNN คือ เมื่อทำนายผลจะนำข้อมูลที่ใกล้เคียงที่สุดมา k ตัวเพื่อแยกแยะวัตถุว่าเป็นอะไร

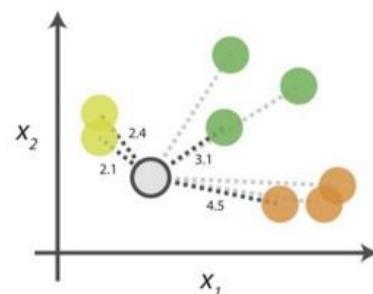
## kNN Algorithm

### 0. Look at the data



Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

### 1. Calculate distances



Start by calculating the distances between the grey point and all other points.

### 2. Find neighbours

Point Distance	
●	2.1 → 1st NN
●	2.4 → 2nd NN
●	3.1 → 3rd NN
●	4.5 → 4th NN

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

### 3. Vote on labels

Class	# of votes	
●	2	Class ● wins the vote!
●	1	→ Point ○ is therefore predicted to be of class ●.
●	1	

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

รูปที่ 2.5 ขั้นตอนกระบวนการของ kNN [16]

จากรูปที่ 2.5 แสดงขั้นตอนกระบวนการของ kNN ได้ดังนี้

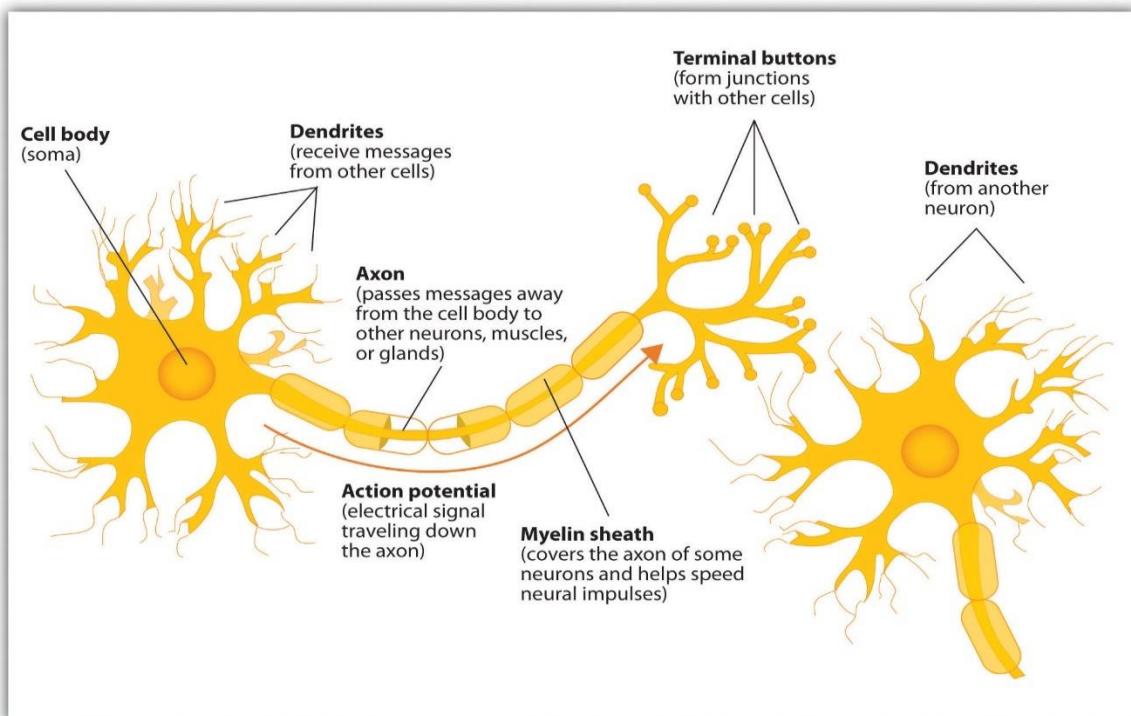
1. สังเกตข้อมูลโดยจากรูปต้องการที่จะแยกແບະหมวดหมู่ของข้อมูลสีเทา จึงนำข้อมูลทั้งหมดที่มีมาวางไว้บนกราฟ
2. คำนวณระยะห่างของข้อมูลที่มีทุกอันกับข้อมูลสีเทาที่ต้องการแยกແບະหมวดหมู่
3. จัดอันดับระยะห่างของข้อมูลทุกดัว จากใกล้สุดไปไกลสุด

4. แยกແພະນາດໝູ່ວັດຖຸຈາກຄໍາ  $k$  ສມມຕິກຳທັນຄໍາ  $k$  ເປັນ 3 ຈະຕວລງສອບສາມເຂົ້ມັດທີ່ມີ  
ຮະບະຫ່າງນ້ອຍທີ່ສຸດ ຈາກຮູປສາມເຂົ້ມັດທີ່ໄກລ໌ເຂົ້ມັດສີເຫາທີ່ສຸດຄື່ອ ສີເໜີລືອງສອງເຂົ້ມັດ ແລະ ສີເປີຍວໜິ່ງ  
ເຂົ້ມັດສ່ງຜລໃຫ້ເຂົ້ມັດສີເຫາເປັນປະເກດເດີຍກັບເຂົ້ມັດສີເໜີລືອງນັ້ນເອງ ເພຣະຈຳນວນສີເໜີລືອງມາກທີ່ສຸດ

## 2.3 Artificial Neural Network

Artificial Neural Network หรือ ภาษาไทยเรียกว่า โครงข่ายประสาทเทียม หมายถึง แบบจำลองที่จำลองการทำงานของสมองมนุษย์

การทำงานของสมองมนุษย์นั้นมีความซับซ้อนมาก สมองประกอบไปด้วยสิ่งที่เรียกว่า นิวรอนหรือ เซลล์ประสาท เซลล์ประสาทแต่ละเซลล์มีหน้าที่ประมวลผลข้อมูลที่เป็นสัญญาณไฟฟ้า ที่ได้รับเข้ามายังเด่น ไดร์ฟซึ่งเป็นเส้นสายปลายประสาทที่ต่อมาจากโซม่า หลังจากนิวรอน ประมวลผลเสร็จแล้ว นิวรอนจะส่ง output ออกไปยังนิวรอนตัวอื่น ๆ ที่เชื่อมโยงกันโดยแอ็คชัน และเทอมินอล บุทตอน ส่วนประกอบพื้นฐานของเซลล์ประสาทเป็นดังรูปที่ 2.6



รูปที่ 2.6 โครงสร้างของเซลล์ประสาท [10]

ในระบบคอมพิวเตอร์ไม่มีการทำงานประมวลผลที่มีความซับซ้อนเท่ากับสมองมนุษย์ สิ่งที่ใช้ในการทำให้โปรแกรมเข้าใจอะไรมักอย่างจะทำได้โดยวิธี Traditional Programming ซึ่งจะสูญเสียทรัพยากรมณุษย์ไปกับการพัฒนาอัลกอริทึมและโปรแกรม ยิ่งถ้าโปรแกรมมีความซับซ้อนมากการพัฒนาโปรแกรมจะมีความยากลำบากมาก จึงเกิดการทำโครงการข่ายประสาทเที่ยมขึ้นมา

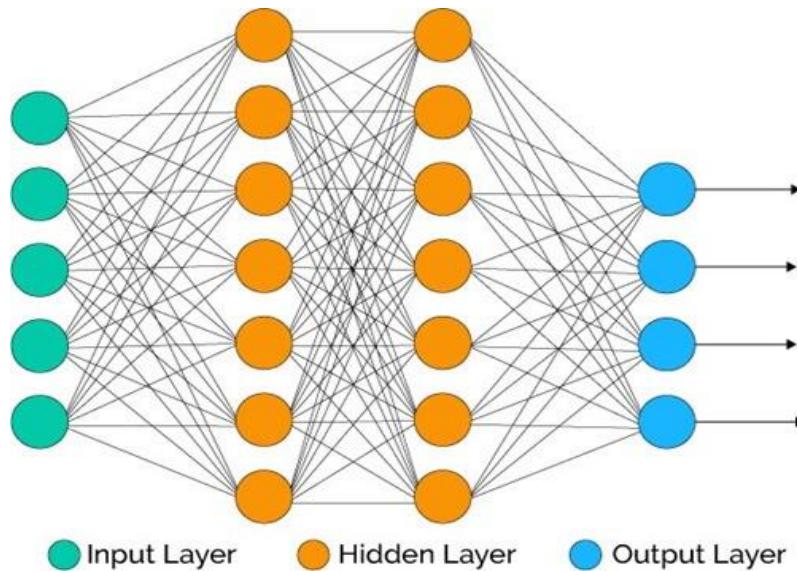
### 2.3.1 ส่วนประกอบของ Artificial Neural Network

Artificial Neural Network สามารถแบ่งออกได้เป็น 3 ส่วนดังนี้

1. Input Layer ในส่วนนี้คือ ส่วนที่ทำหน้าที่รับ input เข้ามา จำนวนโหนดจะมีจำนวนเท่ากับ input เช่น ถ้าสนใจทำโมเดลที่ใช้ในการคาดการณ์กำลังการผลิตของโซล่าเซลล์ ณ ช่วงเวลาต่าง ๆ input ในกรณีจะเป็น อุณหภูมิ ความเร็วลม ปริมาณเมฆ ปริมาณความชื้น ปริมาณฝน และคงว่าใน Input Layer จะมี 4 โหนด

2. Hidden Layer ส่วนนี้จะเป็นโหนดที่เรียกว่า Neuron ซึ่งจะมีผลต่อประสิทธิภาพของโมเดล ในแต่ละ โหนดจะมีฟังก์ชันการประมวลผลอยู่ด้วย ในHidden Layer จะมีจำนวนกี่ชั้นก็ได้ในกรณีที่ Artificial Neural Network มี Hidden Layer หลายชั้น จะถูกเรียกว่า Deep Neural Network ซึ่งเป็นกระบวนการ Deep Learning การเพิ่มชั้นและจำนวนโหนดจะส่งผลต่อการทำงานของโมเดล ไม่เสมอไปที่มีจำนวนโหนดและชั้นเยอะจะหมายความว่าสุดต่อโมเดล เนื่องจากการที่มีโหนดและชั้นเยอะจะส่งผลต่อระยะเวลาในการทำงานของโมเดลอีกมากด้วย

3. Output Layer ส่วนนี้จะต่อจาก Hidden Layer โดยจะรับข้อมูลจากการคำนวณไปใช้ต่อ จำนวนของโหนดใน Output Layer จะขึ้นอยู่กับผลลัพธ์การทำงานของโมเดล



รูปที่ 2.7 โครงสร้างของ Artificial Neural Network [14]

### 2.3.2 ส่วนประกอบของโหนด

1. Input คือข้อมูลหรือค่าที่ป้อนเข้าโหนด

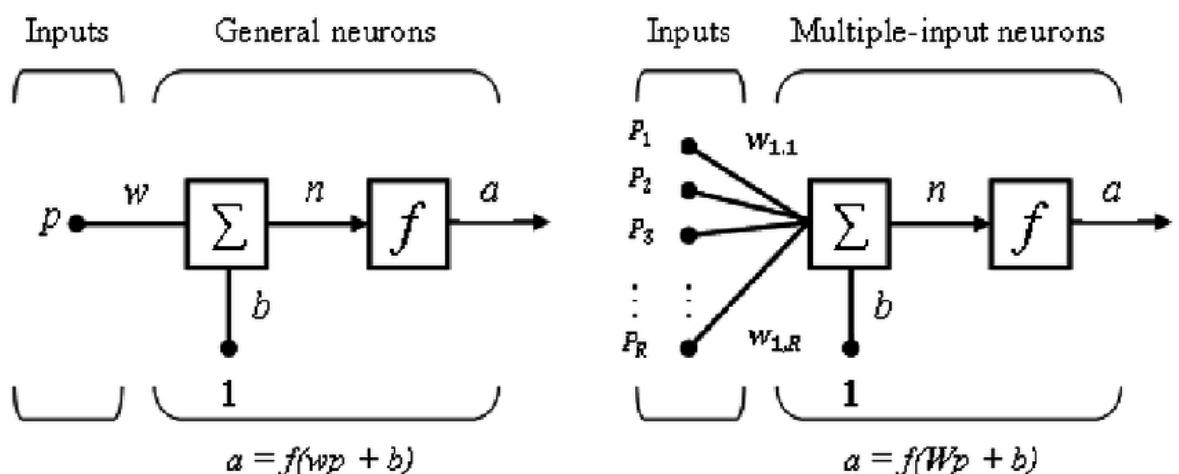
2. Weight คือ การให้น้ำหนักของข้อมูลแต่ละอันที่รับเข้ามาโดยแต่ละข้อมูลหรือแต่ละ input จะมีน้ำหนักเป็นของตัวเอง ความหมายของน้ำหนักคือ input นั้นส่งผลต่อ output มากแค่ไหน ถ้าส่งผลมากจะมีค่าน้ำหนักมาก โดยค่าน้ำหนักจะมีค่าอยู่ระหว่าง 0-1

3. Bias คือตัวเลขที่ช่วยปรับค่าที่คำนวณให้ถูกต้องมากยิ่งขึ้น

#### 4. Output คือผลลัพธ์

##### 2.3.3 หลักการทำงาน

ในโหนดของแต่ละ Hidden Layer เมื่อได้รับ input เข้ามายจะทำการรวมผลรวมของ input ที่คุณกับ Weight เรียกว่า ‘Summation’ จากนั้นจะนำไปบวก Bias แล้วเข้าสู่ Activation Function ซึ่งคือฟังก์ชันการประมวลผลขึ้นอยู่กับการนำไปใช้งาน เช่น Sigmoid, Relu, Softmax เป็นต้น ทุกโหนดจะมี Activation Functions เป็นของตัวเองซึ่งสามารถกำหนดได้ เมื่อ Activation Functions เสร็จสมบูรณ์ จะถูกส่งผลลัพธ์ไปยัง Layer ถัดไป Layer ถัดไปในที่นี้อาจจะเป็น Hidden Layer อีกชั้นหากเป็น Deep Learning หรือจะเป็น Output Layer เพื่อเป็นผลลัพธ์สุดท้ายของโมเดล หลักการทำงานนี้ถูกเรียกว่า Forward Propagation



รูปที่ 2.8 หลักการทำงานของ Artificial Neural Network [12]

จากรูปที่ 2.8 เป็นรูปสมการที่อธิบายหลักการทำงานของ Artificial Neural Network โดย

$a$  = output

$w$  = Weight

$p$  = Input

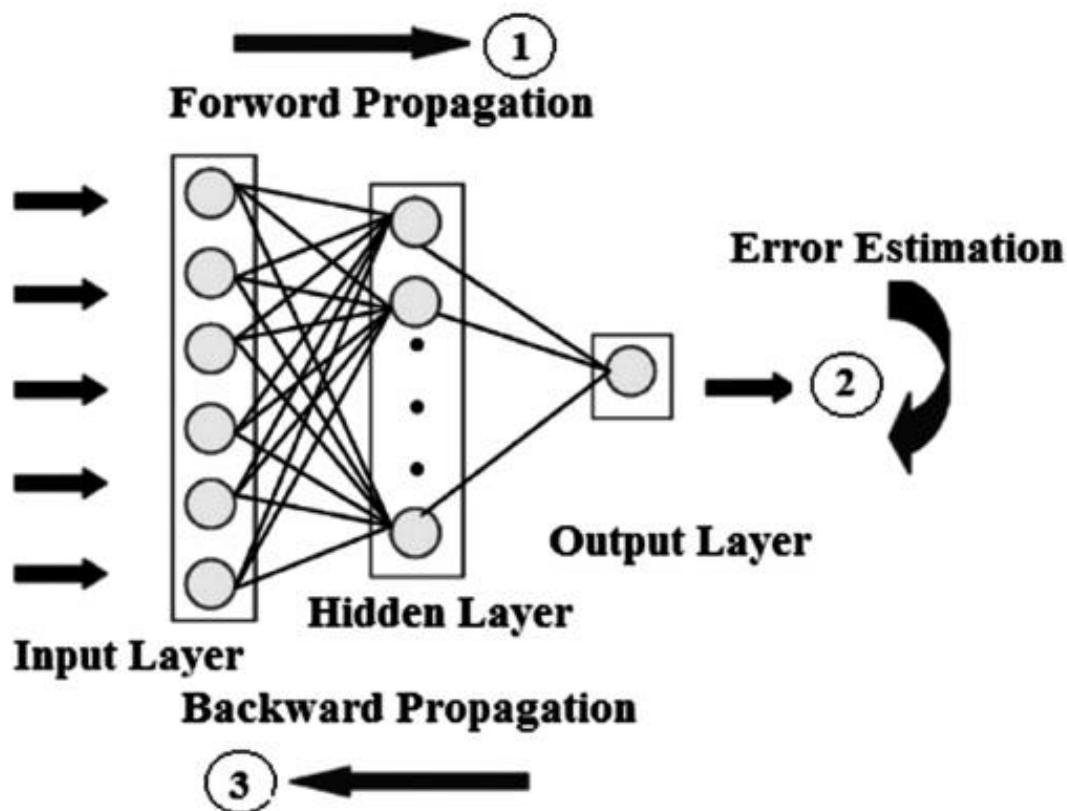
$b$  = Bias

$f$  = Activation Function

##### 2.3.4 Back propagation

Back propagation คือกระบวนการการเรียนรู้ของ Artificial Neural Network โดยเป็นการเรียนรู้จากประสบการณ์ หมายความว่า Artificial Neural Network จะคำนวณหาค่าความคลาดเคลื่อน

ของ output เพื่อนำไปใช้ในการปรับ Weight เพื่อให้ได้ค่า output ใหม่ที่มีค่าความคลาดเคลื่อนน้อยลง การคำนวณค่าความคลาดเคลื่อนจะทำการคำนวณเพื่อแก้ค่า Weight ไปเรื่อยๆ จนถึง Input Layer ก่อนหน้าโดยจะทำการคำนวณเพื่อแก้ค่า Weight ไปเรื่อยๆ จนถึง Input Layer



รูปที่ 2.9 Back propagation [5]

โดย  $x = [x_1, x_2, \dots, x_n]$  คือ รูปแบบของ input

$o = [o_1, o_2, \dots, o_n]$  คือ รูปแบบของ output ที่ต้องการ

$o' = [o'_1, o'_2, \dots, o'_3]$  คือ ผลลัพธ์ของ Neural Network

ค่าความคลาดเคลื่อนของโหนดที่  $i$  ใน Output Layer ที่  $N$  คำนวณได้จาก

$$\delta_{ij}^{(N)} = o_i - o'_i$$

การปรับ Weight ที่ Output Layer ที่  $N$  สามารถคำนวณได้จาก

$$\Delta W_{ij}^{(N-1)} = a \cdot \delta_i^{(N)} \cdot g'(h_i^{(N)}) \cdot y_j^{(N-1)}$$

โดย  $a$

คือ อัตราการเรียนรู้

$\delta_i^{(N)}$	คือ ค่าความคลาดเคลื่อนของโหนดที่ i ใน Output Layer ที่ N
$g'(h_i^{(N)})$	คือ อนุพันธ์ของ output จากโหนดที่ i ใน Output Layer
$h_i^{(N)}$	คือ input ที่โหนดที่ i ใน Output Layer สามารถคำนวณได้จาก
	$h_i = \sum_k W_{ik} y_k^{(N-1)}$
$y_k^{(N-1)}$	คือ output จากโหนดที่ j ใน Layer ที่ N-1

การปรับ Weight ใน Hidden Layer ที่ k สามารถคำนวณได้จาก

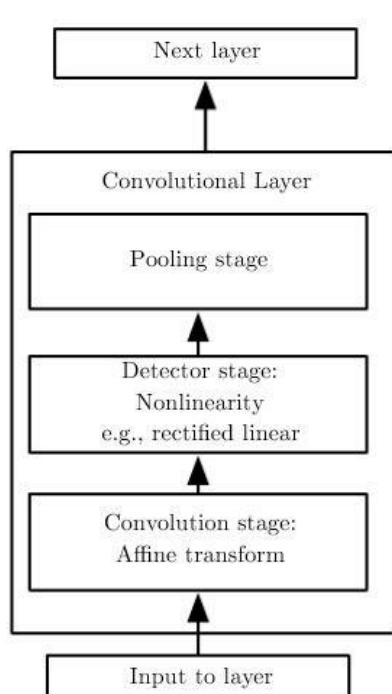
$$\Delta W_{ij}^{(k)} = a \cdot \delta_i^{(k)} \cdot y_j^{(k-1)}$$

โดย  $\delta_i^{(k)}$  คือ ค่าความคลาดเคลื่อนของโหนดที่ i ที่ Layer k สามารถคำนวณได้จาก

$$\delta_{ij}^{(k)} = \sum_i W_{ji}^{(k-1)} \delta_i^{(k+1)}$$

### 2.3.5 Convolutional Neural Networks (CNN)

Convolutional neural networks คือ รูปแบบหนึ่งของการ Artificial Neural Network มีความสามารถในการจัดการปัญหาทางด้าน Classification Recognition ทั้งภาพและเสียง โครงสร้างของ CNN จะประกอบไปด้วย 3 ส่วน ดังรูปที่ 2.10

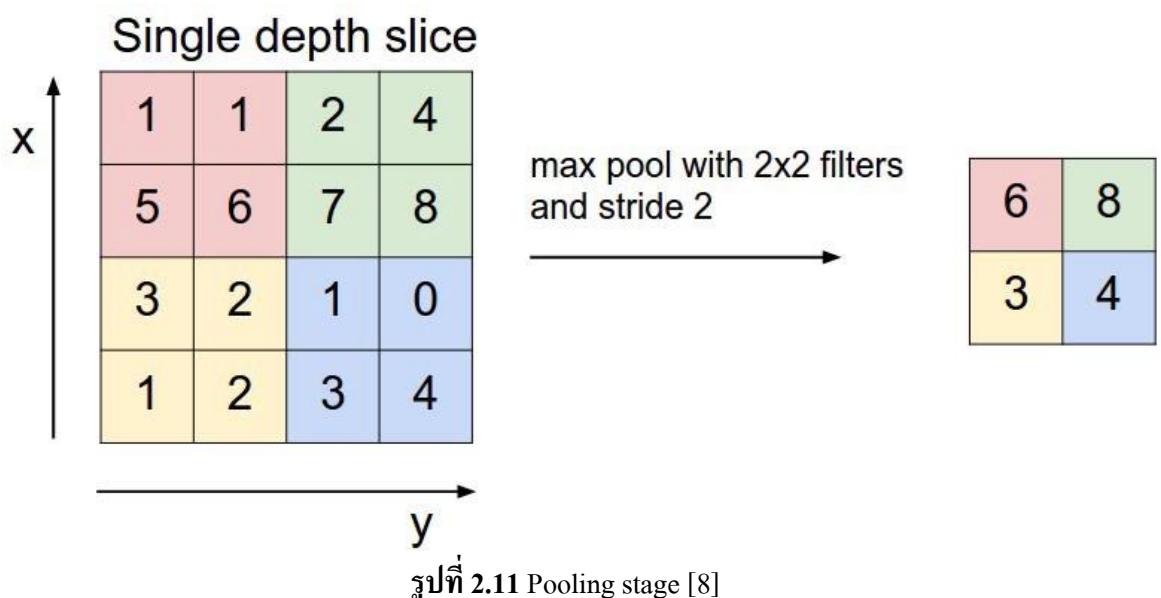


รูปที่ 2.10 โครงสร้าง Convolutional Neural Networks [6]

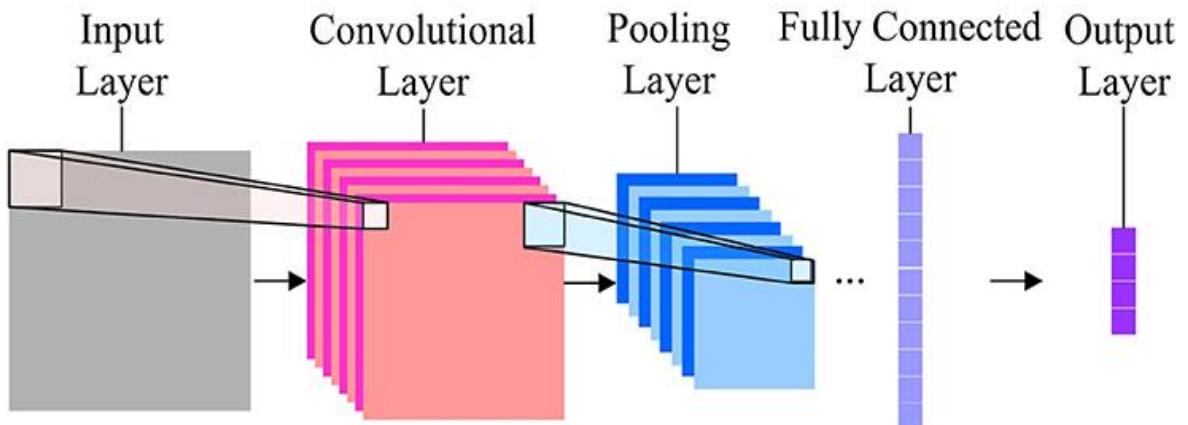
1. Convolution stage โครงสร้างนี้จะทำการแยกรูปซึ่งเป็น input ออกเป็นพิกเซลแล้วจะมีการสร้าง sliding window ขนาด  $n \times n$  พิกเซล มาตรวจสอบรูปภาพส่วนนั้น เมื่อตรวจสอบแล้วจะทำการค้นหาและแยกส่วนประกอบต่าง ๆ ของภาพว่าประกอบไปด้วยอะไรบ้าง จากนั้น Sliding window จะขยับต่อไปเรื่อย ๆ จนครอบคลุมหมดทุกพิกเซลของภาพ Sliding window แต่ละส่วนจะถูกนำมารวมกันเป็นรูป โดยแต่ละ Sliding window จะเป็นตัวแทนขององค์ประกอบของรูป เช่น สี รูปทรง เป็นต้น และในแต่ละ Sliding window จะมีค่า weight เป็นของตัวเอง

2. Detector stage โครงสร้างนี้จะรับข้อมูลต่อมากจาก Convolution stage เพื่อทำการใช้ Activation function

3. Pooling stage โครงสร้างนี้จะทำการหาค่า weight ที่มีมากสุดในแต่ละ Sliding window แล้วนำมารวมกันใหม่ซึ่งจะได้ข้อมูลใหม่มีอน input ที่ป้อนเข้ามาแต่จะมีขนาดเล็กลง Pooling มีประโยชน์ในการเพิ่มความไวในการประมวลผล



เมื่อจบกระบวนการทั้งสามขั้นตอนแล้วสิ่งสุดท้ายคือการนำ Pooling ทั้งหมดมารวมกันที่ Fully connected layer และได้ผลลัพธ์เป็น output ออกมา

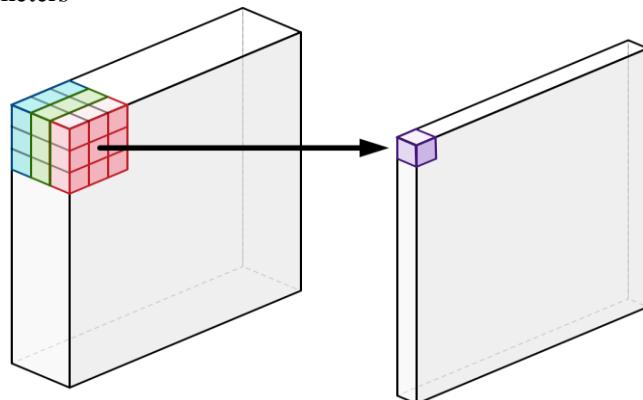


รูปที่ 2.12 สรุปกระบวนการของ Convolutional neural network [3]

### 2.3.6 MobileNet

MobileNet คือ โครงการสร้างโมเดลหนึ่งสำหรับ Convolutional Neural Networks (CNN) มีจุดเด่นที่การประมวลผลที่รวดเร็ว และกินทรัพยากรเครื่องที่น้อย จึงเหมาะสมที่จะนำไปใช้กับการใช้งานแบบ RealTime

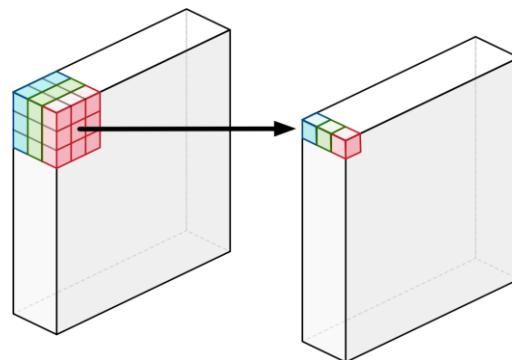
ข้อแตกต่างของ MobileNet กับ CNN ทั่วไปคือ CNN ทั่วไป convolutional filters หรือ sliding windows จะประมวลผลบนทุก channels (มีติความหนา) ของ input สมมติว่า input มี 3 channels และ sliding windows มีขนาด  $3 \times 3$  พิกเซล จำนวน 5 filter ทำให้ convolutional layer นี้มี  $5 \times 3 \times 3 \times 3 = 135$  parameters



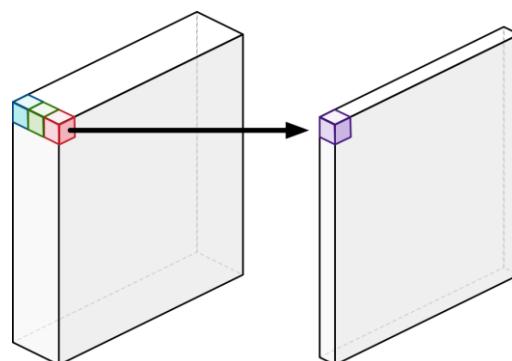
รูปที่ 2.13 การประมวลผลของ CNN ทั่วไป [7]

สำหรับ MobileNet convolutional filters หรือ sliding windows จะไม่ประมวลผลที่เดียวแต่จะแบ่งการประมวลผลเป็นสองขั้นตอนที่เรียกว่า Depthwise separable convolutions และ Pointwise Convolution Filter โดย Depthwise separable convolutions คือการประมวลผลบน channels แต่ละอัน แสดงดังรูปที่ 2.14 ส่วน Pointwise Convolution Filter คือการนำผลลัพธ์จาก Depthwise separable

convolutions ที่มี  $n$  channels มาร่วมกันเป็น channelเดียว แสดงดังรูปที่ 2.15 ทำให้ convolutional layer ของ MobileNet มี  $(5 \times 3 \times 3) + (3 \times 1 \times 1 \times 5) = 60$  parameters จำนวน parameters ที่ลดลงทำให้ใช้เวลาประมวลผลเร็วขึ้นมาก



รูปที่ 2.14 Depthwise separable convolutions [7]

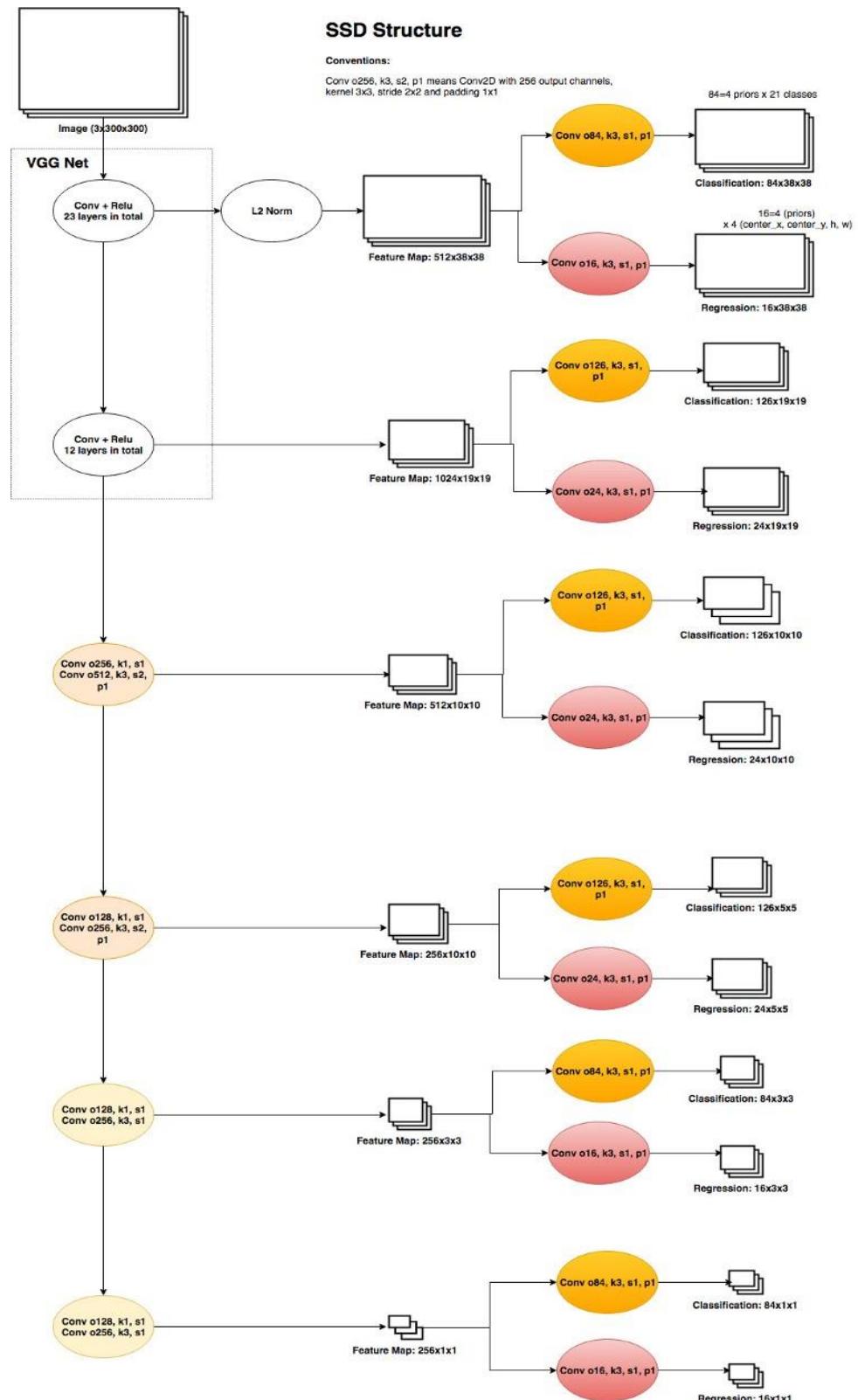


รูปที่ 2.15 Pointwise Convolution Filter [7]

### 2.3.7 Single Shot Multibox Detector (SSD)

Single Shot Multibox Detector (SSD) คือ โครงสร้างหนึ่งของ Convolutional Neural Networks (CNN) จากรูปที่ 2.16 แสดงถึงโครงสร้างของ SSD โดยใช้ โครงสร้าง VGG เป็นโครงสร้าง ตั้งต้นของ Convolutional Layer ซึ่งในจุดนี้สามารถเปลี่ยนไปใช้โครงสร้าง MobileNet ได้เพื่อความ รวดเร็วในการประมวลผล ใน colum ที่สองแสดงถึง Feature map ที่ถูกสร้างขึ้นจาก Convolutional Layer ที่แตกต่างกัน ยกตัวอย่าง เช่น Feature map อันแรกถูกสร้างมาจาก VGG Layer 23 มีขนาด  $38 \times 38$  512 channels ซึ่งทุก ๆ จุดใน  $38 \times 38$  Feature map คือส่วนของรูปภาพอินพุต และ 512 channels คือ Feature ที่ถูกดึงออกมา สามารถใช้ Feature ใน 512 channels เพื่อ classification แยกแยะวัตถุ และ regression เพื่อสร้าง bounding box สำหรับวัตถุขนาดเล็กได้ ในขณะที่ Feature map ที่สองมีขนาด  $19 \times 19$  ซึ่งหมายความว่าสำหรับใช้งานกับวัตถุที่มีขนาดใหญ่ขึ้นเดือน้อย จนถึง Feature map สุดท้ายใช้

สำหรับวัตถุขนาดใหญ่ จึงทิ้งล้ำมาทำให้ SSD เหมาะสำหรับการตรวจจับวัตถุที่มีขนาดแตกต่างกันได้



รูปที่ 2.16 โครงสร้างของ Single Shot Multibox Detector [11]

## 2.4 NodeMCU ESP8266

NodeMCU คือบอร์ดขนาดเล็กที่มีโมดูล Wi-Fi ESP8266 ที่ติดมาพร้อมกับตัวบอร์ด ซึ่งสามารถเชื่อมต่อ Wi-Fi ได้ภายในตัวเอง และนอกจากนี้ตัวบอร์ดยังมาพร้อมกับขาสำหรับเชื่อมต่อ อุปกรณ์ภายนอก ช่องเสียบสาย Micro-USB ที่เอาไว้สำหรับการส่งถ่ายข้อมูลและเป็นช่องสำหรับ จ่ายไฟให้กับตัวบอร์ด ซึ่ง NodeMCU ESP8266 สามารถเขียนโปรแกรมผ่าน Arduino IDE ได้ มี ขนาดเล็ก ราคาถูก ซึ่งเหมาะสมสำหรับในการพัฒนาทางด้าน Internet of Things



รูปที่ 2.17 NodeMCU Version 2 (ESP8266-12E) [4]

ข้อมูลทางเทคนิคของ NodeMCU ESP8266 มีดังนี้

1. Micro-USB สำหรับจ่ายแรงดันไฟเลี้ยง +5 V และสำหรับการถ่ายโอนข้อมูล
2. มีขา GPIO ที่สามารถเป็น PWM, 1-wire และ I2C ได้
3. รับส่งสัญญาณแบบไร้สาย โดย PCB antenna
4. ใช้ ESP8266-12E โดยเป็นครอคตอนโทรลเดอร์ 32 บิต มีหน่วยความจำมากถึง 4 MB
5. มีวงจรควบคุมแรงดันไฟเลี้ยง 3.3V กระแสไฟฟ้าสูงสุด 800mA
6. มีสายอากาศแบบ PCB antenna สำหรับการรับสัญญาณไร้สาย

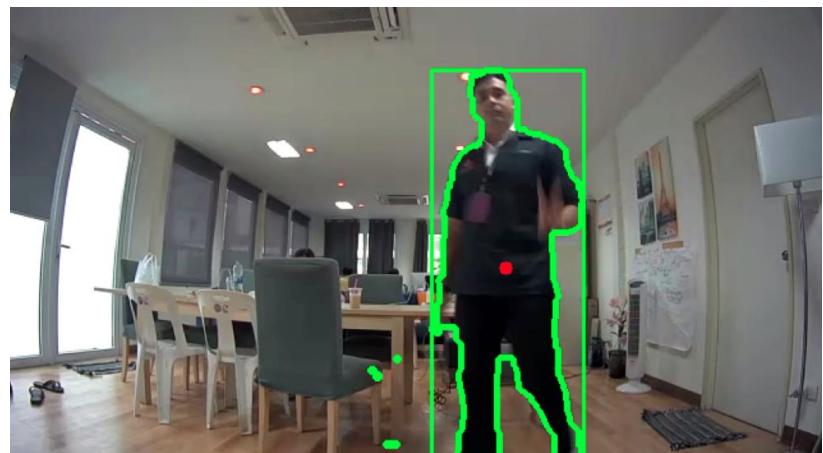
## 2.5 Computer Vision

คอมพิวเตอร์วิทัศน์ (Computer Vision) เป็นเครื่องมือที่ถูกจำลองมาจากธรรมของเห็นของ มนุษย์ใช้สำหรับการวิเคราะห์ภาพนิ่ง การประมวลผลรูปภาพและวิดีโอ โดยการใช้ความสามารถในการแยกส่วนภาพ การแบ่งสัดส่วน การจัดกลุ่มภาพ การรู้จำและการประกอบรูปภาพขึ้นมาให้เพื่อให้ คอมพิวเตอร์นั้นสามารถเข้าใจถึงองค์ประกอบของภาพที่รับเข้ามาได้ ซึ่งใช้หลักการแปลงภาพให้ เป็นแบบจำลองทางคณิตศาสตร์ที่คอมพิวเตอร์สามารถเข้าใจได้ โดยส่วนใหญ่จะนำมาใช้ในส่วนของ การคัดแยกวัตถุที่สนใจในรูปภาพอ кома ยกตัวอย่าง การใช้เทคนิค Image Segmentation เป็นการคัด

แยกบริเวณวัตถุที่สนใจออกมายเป็นกลุ่มสีอย่างชัดเจน ดังรูปที่ 2.18 การใช้เทคนิคการติดตาม (Tracking) ซึ่งหมายความว่าการติดตามวัตถุในวิดีโอ ดังรูปที่ 2.19 แสดงการติดตามภาพบุคคล



รูปที่ 2.18 การแยกวัตถุในรูปภาพอุปกรณ์เป็นกลุ่มสี โดยใช้เทคนิค Image Segmentation



รูปที่ 2.19 การติดตามภาพบุคคล โดยการใช้เทคนิค Tracking

ซึ่งนอกจากนี้ Computer Vision จะนำมาใช้ในงานของการตรวจจับและรู้จำวัตถุที่ต้องการในภาพ การติดตามทิศทางการเคลื่อนไหวของวัตถุ การค้นหาวัตถุภายในภาพ เป็นต้น

## **2.6 Image Processing**

การประมวลผลภาพ ( Image Processing ) เป็นการนำภาพมาประมวลผลด้วยคอมพิวเตอร์โดยผลที่ได้จะได้ทั้งข้อมูลเชิงคุณภาพและข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง ลักษณะ โครงสร้าง ตำแหน่งและทิศทางการเคลื่อนที่ของวัตถุ ซึ่งภาพที่กล่าวมานี้อาจมาจากภาพถ่าย ภาพวิดีโอ หรือ

รูปแบบ 2 มิติ อีน ๆ ภาพที่ได้มาจะต้องนำมาเปลี่ยนจากสัญญาณภาพอนalog ให้กลายเป็นสัญญาณภาพดิจิตอล เพื่อให้คอมพิวเตอร์สามารถประมวลผลภาพนั้นได้ โดยมีขั้นตอนหลัก ดังนี้

1. เปลี่ยนภาพเป็น Grayscale
2. การทำภาพให้มีความคมชัดมากขึ้น
3. การกำจัดสัญญาณรบกวนออกจากภาพ โดยการทำ Smoothing
4. การแยกวัตถุส่วนต่าง ๆ ที่สนใจออกจากภาพ โดยการทำ Thresholding

#### 2.6.1 Color Space

ปริภูมิสี (Color space) คือ ฐานข้อมูลสีจำนวนมาก ซึ่งสามารถแยกระบบสี (Color Model) ออกได้เป็น 3 ระบบดังนี้

##### 1. ระบบสี Grayscale

เป็นระบบสีที่อยู่ในช่วงสีเทา ซึ่งจะให้ระดับของสีที่หลากหลายมากกว่าภาพขาวดำ โดยมีการแสดงถึงความเข้มของสีในระดับที่แตกต่างกัน ซึ่งขนาดของการเก็บข้อมูลจะส่งผลต่อจำนวนระดับสี หากมีพื้นที่ในการเก็บข้อมูลที่มาก ก็จะให้ความความละเอียดของสีที่มากขึ้น



รูปที่ 2.20 รูประบบสี Grayscale

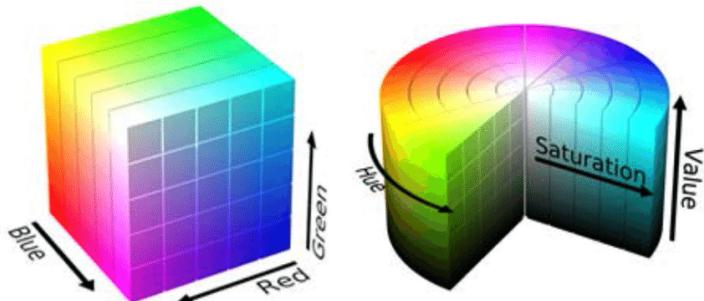
##### 2. ระบบสี RGB

เป็นระบบสีที่ประกอบไปด้วย 3 สีหลักคือ สีแดง (Red) สีเขียว (Green) และ น้ำเงิน (Blue) การผสมสีในสัดส่วนที่แตกต่างกันจะให้ค่าสีที่ต่างกัน หาร่วมครบถ้วนสีจะกล้ายเป็นสีขาว แต่ถ้าหากไม่มีสีจะกล้ายเป็นสีดำ ซึ่งข้อเสียของระบบสี RGB คือ จะไม่สามารถแยกสีของวัตถุบางสีได้อย่างชัดเจนขึ้นอยู่กับสภาพของแสง ความสว่าง ซึ่งหากนำไปใช้ในการแยกแยะวัตถุอาจจะได้ผลลัพธ์ที่ผิด

##### 3. ระบบสี HSV

เป็นระบบสีที่แบ่งออกเป็น 3 ลักษณะคือ Hue แสดงเนคสี, Saturation แสดงความอิ่มตัวของสี, Value แสดงว่าสว่างของสี ซึ่งระบบนี้สามารถแยกแยะสีได้ง่ายเพียงแค่คุ้มค่าของ H ค่าเดียวก็สามารถบอกได้ว่าเป็นสีอะไร และนอกจากนี้ระบบสี HSV จะทำการแยกความสว่างออกเป็น 1 ค่าทำ

ให้สามารถครับแสลงที่มีความหลากหลายได้ดีกว่าระบบสี RGB ซึ่งระบบสี HSV จะมีประสิทธิภาพสูงกว่าระบบสี RGB ในการแยกและวัดลักษณะการประมวลผลภาพ



รูปที่ 2.21 การเปรียบเทียบระหว่าง RGB (ซ้าย) กับ HSV (ขวา) [13]

### 2.6.2 Morphological

Morphological คือการเปลี่ยนแปลงลักษณะรูปร่างโครงสร้างของภาพ ซึ่งการเปลี่ยนแปลงภาพมีหลากหลายลักษณะดังนี้

#### 1. Erosion

Erosion คือการลบจุดบริเวณขอบของภาพออกซึ่งเป็นการกร่อนเข้าไป จะทำให้โครงสร้างภายในภาพมีขนาดบางลง ซึ่งจะช่วยในการลด noise และแยกวัตถุที่เชื่อมต่อกันให้แยกออกจากกัน



รูปที่ 2.22 การเปรียบเทียบระหว่างภาพต้นฉบับ (ซ้าย) กับรูปภาพ Erosion (ขวา) [9]

#### 2. Dilation

Dilation คือการขยายพื้นที่ของวัตถุแต่ละจุดออกไปให้มีขนาดใหญ่ขึ้น ซึ่งจะช่วยในการรวมส่วนต่างๆ ของวัตถุและช่วยเพิ่มพื้นที่ของวัตถุที่เราสนใจได้



รูปที่ 2.23 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Dilation (ขวา) [9]

### 3. Opening

Opening คือการนำรูปภาพมาผ่านกระบวนการ Erosion เพื่อลบ noise และนำมาผ่านกระบวนการ Dilatation อีกครั้งเพื่อขยายพื้นที่ของวัตถุ สามารถใช้ในการลบขอบที่ขึ้นของวัตถุได้



รูปที่ 2.24 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Opening (ขวา) [9]

### 4. Closing

Closing คือกระบวนการที่ตรงข้ามกับ Opening โดยนำรูปภาพมาผ่านกระบวนการ Dilatation ก่อน และนำไปผ่านกระบวนการ Erosion อีกครั้ง สามารถใช้ในการเชื่อมวัตถุที่แยกจากกัน เพราะ noise ได้



รูปที่ 2.25 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Closing (ขวา) [9]

### 5. Gradient

Gradient คือกระบวนการทำนำภาพที่ผ่านการ Dilation มาลับออกด้วยภาพ Erosion ผลลัพธ์ที่ได้จะเป็นขอบของวัตถุ สามารถนำไปใช้ในการ detect edge ได้



รูปที่ 2.26 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Gradient (ขวา) [9]

### 6. Top Hat

Top Hat คือกระบวนการที่ใช้แยกส่วนของรูปภาพที่สว่างกว่ารอบข้าง โดยเกิดจากการนำรูปต้นฉบับมาหักลบกับรูปที่ได้จากการกระบวนการ Opening



รูปที่ 2.27 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Top Hat (ขวา) [9]

### 7. Black Hat

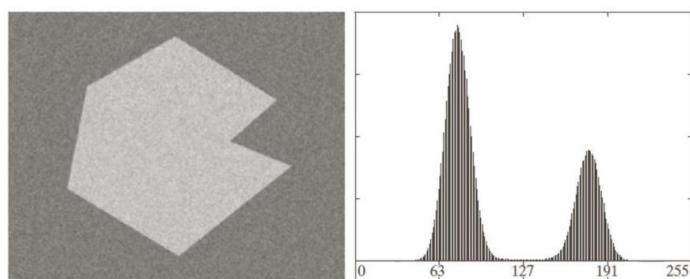
Black Hat คือกระบวนการที่ใช้แยกส่วนของรูปภาพที่มืดกว่ารอบข้าง โดยเกิดจากการนำรูปจากการ减去รูปต้นฉบับกับรูปภาพต้นฉบับ



รูปที่ 2.28 การเปรียบเทียบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Black Hat (ขวา) [9]

### 2.6.3 Thresholding

Thresholding เป็นการแบ่งระหว่างสีที่คล้ายหนึ่ง หรือที่เรียกว่า การทำปีดแบ่ง ซึ่งจะใช้เกณฑ์ Threshold หรือที่เรียกว่า ค่าปีดแบ่ง เป็นตัวกำหนดการพิจารณาว่าจุดใดควรเป็นสีดำหรือสีขาว ซึ่งค่านี้จะเป็นค่าความเข้มแสงค่าหนึ่ง เราสามารถกำหนดได้ว่าจะต้องมีค่าเท่าไหร่เพื่อใช้ในการตัดพื้นหลังออกจากวัตถุที่เราสนใจ โดยที่ค่า Threshold จะอยู่ระหว่าง 0 - 255 เท่านั้น ซึ่งการกำหนดค่า Threshold นั้นสามารถดูได้จากกราฟ Histogram ว่ามีค่าความหนาแน่นน้อยในช่วงใดของกราฟบ้าง โดยแสดงดังรูปที่ 2.29 เราสามารถกำหนดค่า Threshold ได้อยู่ที่ 127 ซึ่งระบบก็จะสามารถแยกพื้นหลังออกจากวัตถุที่เราต้องการได้



รูปที่ 2.29 Histogram [1]

### 2.6.4 Smoothing Image

การทำ Smoothing เป็นการปรับปรุงคุณภาพของภาพเพื่อให้ได้คุณภาพสำหรับการประมวลผลที่ดียิ่งขึ้น เนื่องจากภาพที่ได้มานั้นจะมีสัญญาณรบกวน หรือที่เรียกว่า noise ซึ่งจะเป็นความถี่สูง (High- Frequency) การทำ Smoothing เป็นการปรับภาพให้เรียบเนียนโดยใช้การกรองด้วยความถี่ต่ำผ่าน (Low-pass filters (LPF)) ซึ่งจะช่วยในการลดสัญญาณรบกวนในภาพลง ดังแสดงในรูปที่ 2.30 (ภาพปกติ, ภาพที่ผ่านการ Smoothing)

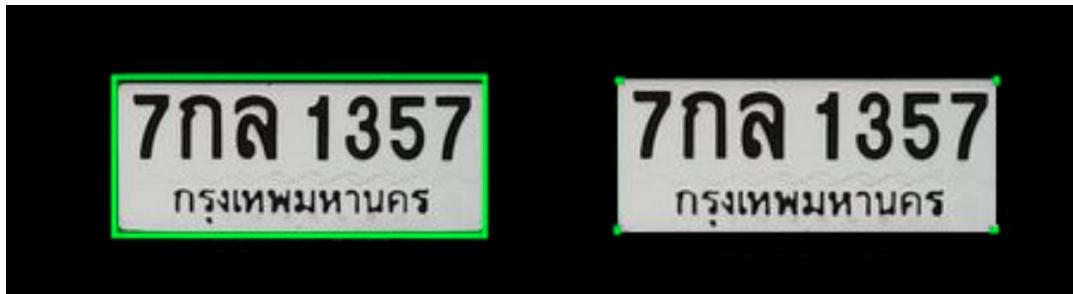


รูปที่ 2.30 การเปลี่ยนเที่ยบรูปภาพต้นฉบับ (ซ้าย) กับรูปภาพ Smoothing (ขวา)

### 2.6.5 Contour

Contour เป็นเครื่องมือสำหรับการวิเคราะห์รูป่างและตรวจจับรูปทรงของวัตถุ ซึ่ง Contour จะเป็นการหาเส้นขอบโดยใช้การรวมจุดในภาพต่อเนื่องกันเป็นรูปทรงของวัตถุต่าง ๆ โดยเส้นขอบ

จะมีสีหรือความเข้มที่เท่ากัน ซึ่งหากจะต้องเก็บข้อมูลพิกัด ( $x, y$ ) ของขอบรูปร่างในแต่ละชุดแล้ว จะต้องใช้หน่วยความจำที่เยอะขึ้น จึงจำเป็นจะต้องใช้วิธีการประมาณเส้น ลบจุดที่ซ้ำซ้อนทั้งหมด และใช้การประมาณจากจุดดันและจุดปลายเพื่อช่วยในการลดหน่วยความจำลง ดังแสดงในรูปที่ 2.31



รูปที่ 2.31 การเปรียบเทียบการเขียนพิกัดทั้งหมด (ซ้าย) กับการเขียนพิกัดโดยประมาณ (ขวา)

## 2.7 การเข้ารหัสและถอดรหัสตัวอักษร

โดยทั่วไปมนุษย์ติดต่อสื่อสารกันด้วยภาษา อักษร หรือตัวอักษรเพื่อที่จะสามารถเข้าใจถึงข้อมูลที่ต้องการติดต่อสื่อสารกันได้ แต่สำหรับคอมพิวเตอร์นั้น จะต้องใช้ภาษาเครื่องในการติดต่อซึ่งจะอยู่ในรูปแบบรหัสเลขฐานสองสำหรับการประมวลผลเพื่อรับคำสั่งและส่งข้อมูล แต่การสื่อสารระหว่างมนุษย์กับคอมพิวเตอร์ไม่สามารถใช้ภาษาเดียวกันในการสื่อสารให้เข้าใจตรงกันได้ จึงจำเป็นต้องมีกระบวนการแปลงรหัสที่แตกต่างกันให้สามารถเข้าใจกันได้ โดยที่ผู้ใช้ให้ข้อมูลแล้วทำการแปลงเป็นสัญญาณรหัสข้อมูลแล้วถูกส่งไปยังผู้รับ เรียกว่า การเข้ารหัส (Encoding) และการแปลงสัญญาณข้อมูลที่ได้รับกลับคืนมา เรียกว่า การถอดรหัส (Decoding) โดยรหัสสำคัญที่ใช้ในโภรงานนี้มี 2 ประเภท ได้แก่

### 2.7.1 ASCII Code

แอสตี ย่อมาจาก American Standard Code for Information Interchange เป็นรหัสมาตรฐานที่ใช้แทนข้อมูลตัวอักษร สัญลักษณ์เครื่องหมายต่าง ๆ ซึ่ง ASCII สามารถเก็บข้อมูลได้ 1 byte โดยที่แต่ละรหัสจะสามารถเก็บข้อมูลได้ 1 อักษร การที่รหัส ASCII สามารถเก็บข้อมูลได้น้อยทำให้เกิดข้อเสียตามมา คือ ASCII จะไม่สามารถเก็บได้หลาย ๆ ภาษา เนื่องจากมีจำนวน bit ที่น้อย 送ผลให้รหัสสูญเสียซ้ำกัน

### 2.7.2 Unicode

Unicode เป็นรหัสมาตรฐานอุตสาหกรรมที่ใช้สำหรับการเข้ารหัสข้อมูลตัวอักษร สัญลักษณ์ และตัวเลข สามารถเก็บข้อมูลได้มากกว่ารหัส ASCII ในปัจจุบัน Unicode สามารถเก็บข้อมูลตัวอักษรได้มากถึง 100,000 ตัว และสามารถทำงานได้โดยไม่ขึ้นกับภาษาและโปรแกรม จึงทำให้ Unicode สามารถเขียนข้อมูลไปมาระหว่างระบบที่หลากหลายได้โดยไม่ต้องเพียงหรือซ้ำกับภาษาอื่น ซึ่งชุดอักษรที่นิยมและเป็นที่รู้จักมากที่สุดคือ UTF-8

## **2.8 ภาษา Python**

ภาษา Python เป็นภาษาระดับสูง ที่ถูกพัฒนามาจากภาษา C มีลักษณะของตัวภาษาที่อ่านง่าย มีการย่อหน้าบรรทัดทำให้เขียนได้ง่าย ไม่ต้องเขียนวงเล็บปีกมาปิดและปีกมาปิดสำหรับการจบโปรแกรมของการเขียนโปรแกรม โครงสร้างภาษาอนุญาตสามารถเข้าใจได้ง่าย ไม่ซับซ้อนและใช้บรรทัดในการเขียนที่น้อยกว่าภาษาอื่น ๆ มาก นอกจากนี้ ภาษา Python ยังมีไลบรารีที่มากมายและครอบคลุม การทำงาน เช่น ไลบรารีสนับสนุนด้านปัญญาประดิษฐ์ ไลบรารีสนับสนุนด้านการสร้างภาพ กราฟฟิก มีไลบรารีสำหรับสร้างเอกสาร มีระบบการจัดการหน่วยความจำแบบอัตโนมัติโดยที่เราไม่ต้องกำหนดขนาดความจำของตัวแปร มีการสนับสนุนการเขียนโปรแกรมได้หลากหลายรูปแบบ ประกอบด้วย การเขียนโปรแกรมแบบขั้นตอน การเขียนโปรแกรมแบบฟังก์ชัน การเขียนโปรแกรมเชิงวัตถุ imperative สามารถรองรับการทำงานผ่านทางระบบปฏิบัติการ Ubuntu, Windows, macOS และ Raspberry Pi ทำให้เป็นภาษาที่ใช้งานง่ายและเป็นที่นิยมในปัจจุบัน

## **2.9 Anaconda Distribution**

Anaconda Distribution เป็นโปรแกรมสำหรับการจัดการ Python (ไฟฟอน) ที่รวบรวม Python library เพื่อช่วยให้ผู้ใช้นั้นสามารถใช้งานได้ง่ายและสะดวกขึ้น ซึ่งผู้จัดทำโครงงานได้ใช้โปรแกรมนี้ติดตั้ง Library ที่จำเป็น

## **2.10 Jupyter Notebook**

Jupyter Notebook เป็นเว็บแอปพลิเคชันที่สามารถเปิดให้ทุกคนได้ใช้ฟรี ซึ่งจะช่วยให้ผู้ใช้สามารถเขียนโปรแกรมภาษา Python ได้บนเว็บเบราว์เซอร์ โดยสามารถสั่งให้รันโค้ดได้ที่บรรทัดแล้วให้ผลลัพธ์ทันที ผู้ใช้ไม่จำเป็นต้องกดบันทึกหรือพิมพ์โค้ดให้ครบก่อนแล้วค่อยรันเหมือนกับโปรแกรมอื่น ๆ นอกจากนี้ Jupyter Notebook ยังเก็บข้อมูลและจำฟังก์ชันตัวแปรของบรรทัดบน ๆ ก่อนหน้าไว้ให้ด้วย ทำให้ผู้ใช้สามารถใช้ง่ายได้สะดวกและรวดเร็วยิ่งขึ้น

## **2.11 TensorFlow**

TensorFlow เป็นไลบรารีซอฟต์แวร์ Open-source ที่สามารถนำมาแก้ไข ดัดแปลงโค้ดได้โดยใช้ภาษา Python ในการพัฒนาโดย TensorFlow ถูกพัฒนาโดยทีมงาน Google สำหรับการคำนวณเชิงสมการตัวเลขที่มีประสิทธิภาพสูง มีสถาปัตยกรรมที่ยืดหยุ่นช่วยให้สามารถคำนวณและประมวลผลได้ง่ายได้หลากหลายรูปแบบทั้งใน CPU, GPUs และ TPUs ซึ่งหน้าที่หลักของ TensorFlow จะนำมาใช้ในการพัฒนาและประมวลผลตัว Machine Learning, Neural Network, Deep Learning สามารถติดตั้งได้ง่ายและรองรับการทำงานผ่านทางระบบปฏิบัติการ Ubuntu, Windows, macOS และ Raspberry Pi ทำให้ตัวไลบรารีนี้เป็นที่นิยมในการนำมาใช้บริษัทขนาดใหญ่ต่าง ๆ นำตัว TensorFlow

มาใช้ในการพัฒนาอย่างกว้างขวาง เช่น บริษัท Google Inc. บริษัท Advanced Micro Devices Inc. (AMD) บริษัท Uber Technologies Inc.

## **2.12 Keras**

Keras เป็นไลบรารีที่ถูกพัฒนาโดยใช้ภาษา Python สำหรับการทำ Neural Network API ระดับสูงให้มีความง่าย สะดวกและมีความรวดเร็วในการหาผลลัพธ์ โดย Keras มีการใช้งาน Neural Network ที่ค่อนข้างหลากหลายสามารถทำงานบน TensorFlow, CNTK หรือ Theano ได้ ช่วยในการเพิ่มประสิทธิภาพและมีเครื่องมือที่ช่วยในการทำงานกับรูปภาพและข้อมูล ได้ง่ายขึ้น นอกจากนี้ Keras ยังช่วยให้ผู้ใช้สามารถสร้างโมเดลลงบนสมาร์ทโฟน หรือบนเว็บ ได้และยังสามารถยังทำงานบน CPU และ GPU ได้อย่างราบรื่น

## **2.13 OpenCV**

Open Source Computer Vision Library หรือที่เรียกว่า OpenCV เป็นไลบรารีสำหรับการคำนวณและเน้นการใช้งานในแบบประมวลผลภาพแบบเรียลไทม์ ซึ่งสามารถนำไปใช้ในระบบด้านการอ่านภาพ การแสดงรูปภาพ ได้ด้วย และยังมีโครงสร้างที่สามารถสนับสนุนหลายโปรแกรม และหลายภาษา เช่น ภาษา C++, Java, Python เป็นต้น

## **2.14 Internet of Things (IoT)**

Internet of Things กือ เทคโนโลยีสำหรับการนำข้อมูลของอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ มาเชื่อมโยงและส่งข้อมูลผ่านระบบอินเทอร์เน็ต เช่น โทรศัพท์ คอมพิวเตอร์ โทรศัพท์มือถือ รถยนต์ กล้องวงจรปิด ศูนย์เย็น ปลั๊กไฟฟ้า และอื่น ๆ รวมเข้าไว้ด้วยกัน ซึ่งทำให้เราสามารถอ่านข้อมูล ส่งข้อมูล และรับข้อมูล ไปมาระหว่างตัวอุปกรณ์กับระบบอินเทอร์เน็ต ได้ ซึ่งจะทำให้เราสามารถตรวจสอบการทำงาน การสั่งการควบคุมตัวอุปกรณ์อิเล็กทรอนิกส์ได้ โดยการควบคุมผ่านอินเทอร์เน็ต ได้ทุกที่ เรียก อุปกรณ์ที่เชื่อมต่อกับอินเทอร์เน็ตเหล่านี้ว่า “อุปกรณ์อัจฉริยะ” หรือที่เรียกว่า SmartDevice

### **2.14.1 การประยุกต์ใช้ Internet of Things (IoT)**

Internet of Things หมายความว่า การที่สิ่งต่าง ๆ หรืออุปกรณ์ต่าง ๆ นั้นสามารถเชื่อมโยงเข้ากับอินเทอร์เน็ตเข้าไว้ด้วยกันหรือเรียกง่าย ๆ ว่า อินเทอร์เน็ตที่อยู่ในทุก ๆ สิ่ง สามารถนำมาประยุกต์ใช้ในชีวิตประจำวันหรือในเชิงอุตสาหกรรม ต่าง ๆ ได้มากmany เช่น

#### **1. Smart Home**

Smart Home เป็นแนวคิดการประยุกต์ใช้งานภายในบ้านพักอาศัย เช่นระบบสั่งการเปิด-ปิดไฟหรืออุปกรณ์เครื่องใช้ไฟฟ้าผ่านทางโทรศัพท์มือถือ

#### **2. Smart Farm**

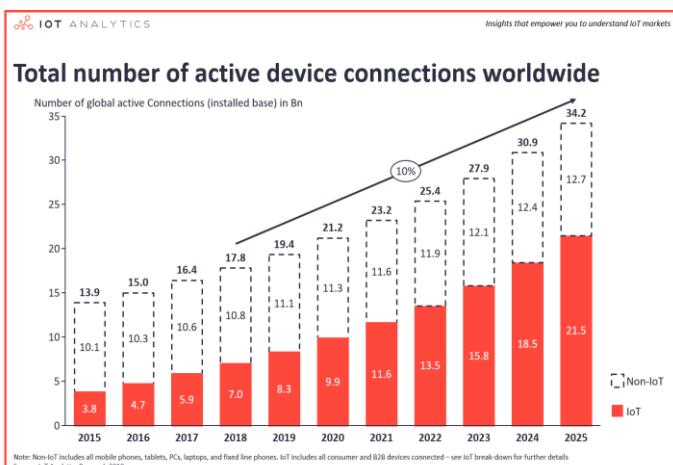
Smart Farm เป็นแนวคิดการประยุกต์ใช้งานในระบบเกษตรกรรม ซึ่งจะเป็นอุปกรณ์อัตโนมัติที่จะช่วยเพิ่มประสิทธิภาพผลผลิต ลดต้นทุนแรงงาน และช่วยในการจัดเก็บข้อมูลคิดวิเคราะห์ แก้ไขในระบบฟาร์มได้ เช่น เชนเซอร์วัสดุอุณหภูมิ ความชื้น และความอุดมสมบูรณ์ของดิน

### 3. Wearables

Wearables เป็นแนวคิดการประยุกต์ใช้งานในการส่วนใส่ตามส่วนต่าง ๆ ของร่างกาย เพื่อช่วยในการเก็บข้อมูลของร่ายกายในการนำผลมาคิดวิเคราะห์ ประเมินผลทางด้านสุขภาพ

### 4. Smart Grid

Smart Grid เป็นแนวคิดการประยุกต์ใช้งานในระบบโครงข่ายไฟฟ้าอัจฉริยะ ซึ่งเป็นระบบที่ช่วยจัดการทรัพยากรทั่วทั้งระบบไฟฟ้า ครอบคลุมด้วยแต่ระบบผลิต ระบบส่ง ระบบจำหน่าย และระบบผู้ใช้ไฟฟ้า



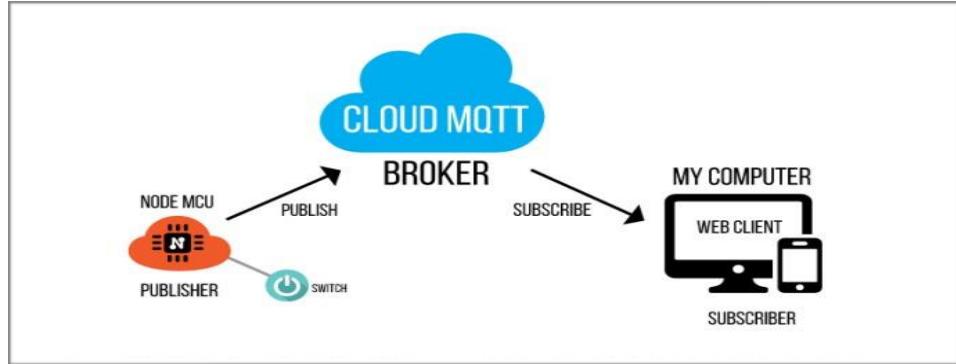
รูปที่ 2.32 จำนวนอุปกรณ์ IoT ที่มีการเชื่อมต่อทั่วโลก [15]

## 2.15 CloudMQTT

### 2.15.1 MQTT (Message Queue Telemetry Transport)

MQTT เป็นโปรโตคอล ระหว่างเครื่องจักรกับเครื่องจักรติดต่อกัน หรือการเชื่อมต่อแบบ M2M (machine-to-machine) ใช้วิธีการเหมือนกับ Message Queue ซึ่งสนับสนุนเทคโนโลยี Internet of Things นокจากจะรองรับและอ่านค่าแล้วยังสามารถสั่งงานอุปกรณ์ที่ใช้ MQTT ได้ด้วยการใช้งานโปรโตคอลนี้ต้องจะต้องมีอุปกรณ์ที่ฝั่ง MQTT Broker มาเพื่อทำหน้าที่รับส่งข้อมูลไปยังอุปกรณ์อื่นที่ใช้ MQTT เมื่อนกัน

### 2.15.2 CloudMQTT



รูปที่ 2.33 แผนภาพการเชื่อมต่อของ NodeMCU ผ่าน Cloud MQTT [17]

CloudMQTT เป็นบริการ MQTT Service บน Cloud Service ซึ่งใช้ AWS Amazon เป็นโครงสร้างพื้นฐาน ซึ่ง CloudMQTT สามารถเลือก plan การใช้งานได้ ซึ่งจะแบ่งตามคุณสมบัติต่าง ๆ ตามการใช้งาน

### 2.15.3 โปรโตคอล

โปรโตคอล (Protocol) คือ ข้อกำหนดในการสื่อสารกันของคอมพิวเตอร์ การที่คอมพิวเตอร์ต่างระบบจะสื่อสารกันได้เข้าใจจำเป็นต้องมี Protocal ตัวอย่างเช่น HTTP เป็นต้น

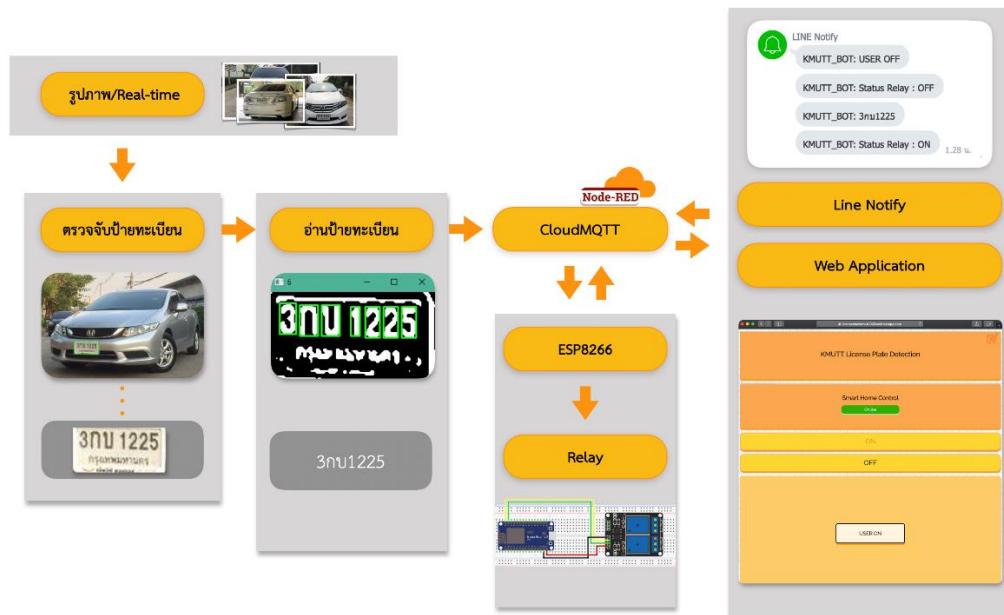
## บทที่ ๓

### 3.1 การออกแบบ

การออกแบบระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์ มีการออกแบบอยู่ ด้วยกัน 4 ขั้นตอน ได้แก่

3.1.1 การออกแบบการทำงานของระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์

หลักการทำงานของระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์เริ่มจากการ  
รถยนต์จากกล้องจะถูกนำมาตรวจจับหากป้ายทะเบียนรถยนต์โดยใช้โปรแกรมตรวจจับป้ายทะเบียน  
รถยนต์ที่ประมวลผลอยู่ในคอมพิวเตอร์ที่เชื่อมต่ออินเตอร์เน็ตไว้กับคอมพิวเตอร์จะส่งข้อมูลที่  
ประมวลผลเสร็จแล้วขึ้นคลาวด์ MQTT และ NodeMCU ที่เชื่อมต่ออินเตอร์เน็ตไว้จะรับข้อมูลจาก  
คลาวด์ MQTT มาใช้ในการสั่งการรีเลย์เพื่อควบคุมอุปกรณ์ไฟฟ้า รวมถึงมีระบบแสดงผลผ่าน Line  
Notification รวมทั้งเว็บแอปพลิเคชันที่แสดงผลหมายเลขอุปกรณ์ รวมถึงสามารถสั่งการรีเลย์ได้  
ยกตัวอย่างของการทำงาน เช่นการนำระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์ไปใช้  
กับการเปิด-ปิดไฟบ้านพักอาศัย หากกล้องตรวจพบป้ายทะเบียนรถยนต์ และสามารถอ่านป้าย  
ทะเบียนรถยนต์ได้ผลลัพธ์หมายเลขอุปกรณ์ที่ตรงกับหมายเลขอุปกรณ์ของผู้อยู่อาศัย จะสั่ง  
การให้เปิดไฟในบ้านได้ และสามารถสั่งการปิดไฟในบ้านได้ผ่านเว็บแอปพลิเคชัน แผนผังการ  
ทำงานแสดงดังรูปที่ 3.1

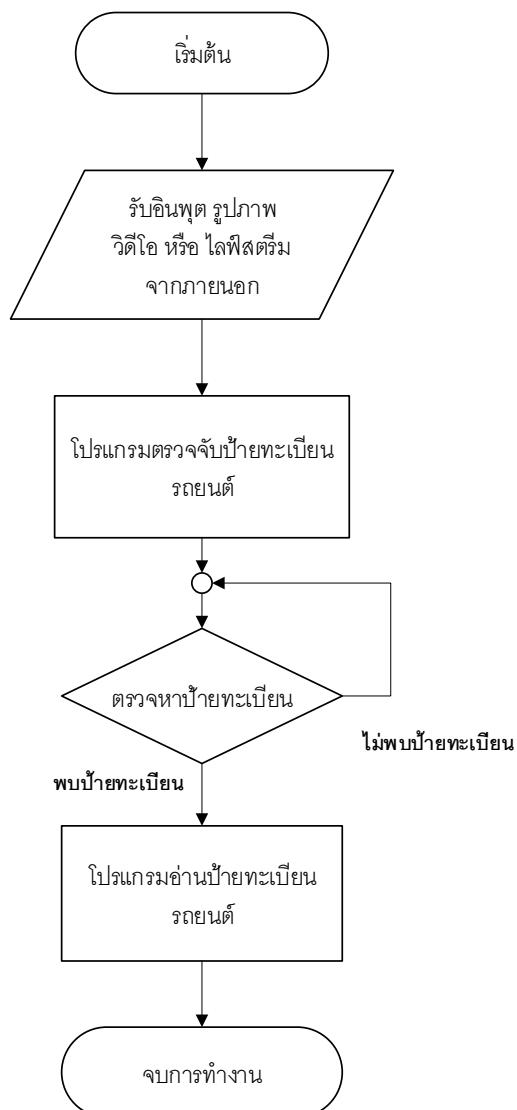


รูปที่ 3.1 แผนผังการทำงานของระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์

### 3.1.2 การออกแบบโปรแกรมตรวจจับป้ายทะเบียนรถยนต์

โปรแกรมตรวจจับป้ายทะเบียนรถยนต์ที่ได้ออกแบบจะใช้หลักการทำงานคือ การฝึกฝนปัญญาประดิษฐ์ Tensorflow Object-Detection API ซึ่งใช้กระบวนการ Convolutional Neural Network เพื่อให้โปรแกรมรู้จักว่าป้ายทะเบียนรถยนต์มีลักษณะเป็นอย่างไร โดยขั้นตอนการฝึกฝนจะกล่าวถึงในหัวข้อที่ 3.2

เมื่อโปรแกรมตรวจจับป้ายทะเบียนรถยนต์สามารถตรวจจับป้ายทะเบียนรถยนต์ได้แล้ว ขั้นตอนถัดไปคือการนำรูปภาพป้ายทะเบียนรถยนต์ที่ถูกตรวจพบไปค้นหาหมายเลขทะเบียนและแสดงผลตัวอักษร โดย Flowchart ขั้นตอนการทำงานของโปรแกรมตรวจจับป้ายทะเบียนรถยนต์แสดงดังรูปที่ 3.2



รูปที่ 3.2 ขั้นตอนการทำงานของโปรแกรมตรวจจับป้ายทะเบียนรถยนต์

อธิบายโค้ดของโปรแกรมตรวจจับป้ายทะเบียนรถยนต์ในกรณีอินพุตเป็นรูปภาพ

1. การนำเข้าไลบรารี และโมดูลต่าง ๆ ที่ต้องใช้ ดังรูปที่ 3.3 โดยในบรรทัดที่ 7 เป็นการนำเข้าโปรแกรม test\_Readplate ซึ่งเป็นโปรแกรมที่ไว้ใช้อ่านป้ายทะเบียนรถยนต์ ในบรรทัดที่ 12 และ 13 เป็นการนำเข้าโมดูล จาก Tensorflow Object-Detection API

```

1 import numpy as np
2 import os
3 import tensorflow as tf
4 from matplotlib import pyplot as plt
5 from PIL import Image
6 import cv2
7 import test_Readplate
8
9
10 from object_detection.utils import label_map_util
11 from object_detection.utils import visualization_utils as vis_util

```

รูปที่ 3.3 การนำเข้าไลบรารี และโมดูลต่าง ๆ

2. การกำหนดตำแหน่งของโมเดลที่ต้องใช้ในการทำงานของโปรแกรม โดยในบรรทัดที่ 13 เป็นชื่อโมเดลที่จะใช้ทดสอบ ไฟล์ frozen\_inference\_graph.pb ในบรรทัดที่ 14 เป็นไฟล์โมเดลที่ใช้ตรวจจับป้ายทะเบียนรถยนต์ และไฟล์ object-detection.pbtxt ซึ่งเป็นไฟล์ที่ระบุคลาสต่าง ๆ ใน frozen\_inference\_graph.pb เนื่องจากโปรแกรมนี้ใช้ตรวจจับป้ายทะเบียนรถยนต์เท่านั้น จึงมีจำนวนคลาสเป็น 1 และดังรูปที่ 3.4

```

13 MODEL_NAME = 'ssdlite_model'
14 PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
15 PATH_TO_LABELS = os.path.join('training', 'object-detection.pbtxt')
16 NUM_CLASSES = 1

```

รูปที่ 3.4 การกำหนดตำแหน่งของโมเดลและการระบุจำนวนคลาส

3. กำหนดฟังก์ชันที่ใช้ในการแปลงรูปภาพให้เป็น numpy array ดังรูปที่ 3.5

```

18 def load_image_into_numpy_array(image):
19     (im_width, im_height) = image.size
20     return np.array(image.getdata()).reshape(
21         (im_height, im_width, 3)).astype(np.uint8)

```

รูปที่ 3.5 การกำหนดฟังก์ชันแปลงรูปภาพเป็น numpy array

4. กำหนดฟังก์ชันไว้สำหรับบันทึกรูปภาพป้ายทะเบียนที่ถูกตรวจสอบ ดังรูปที่ 3.6

```
23 def save_image(img_data2):
24     img = img_data2
25     filename2 = os.path.join('images1.JPG')
26     cv2.imwrite(filename2, img)
```

รูปที่ 3.6 การกำหนดฟังก์ชั่นบันทึกรูปภาพป้ายทะเบียนรถยนต์

5. เรียกใช้ Tensorflow model ดังรูปที่ 3.7

```
28 detection_graph = tf.Graph()
29 with detection_graph.as_default():
30     od_graph_def = tf.GraphDef()
31     with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
32         serialized_graph = fid.read()
33         od_graph_def.ParseFromString(serialized_graph)
34         tf.import_graph_def(od_graph_def, name='')
```

รูปที่ 3.7 การโหลด Tensorflow model

6. โหลดไฟล์ object-detection.pbtxt ซึ่งเป็นไฟล์ที่ระบุชื่อ เมื่อ Convolutional Network ทำงานยิ่งเป็น 1 จะหมายความว่าสิ่งนั้นคือป้ายทะเบียนรถยนต์ และใช้ฟังก์ชั่นของ Tensorflow เพื่อแปลง object-detection.pbtxt ให้เป็นหมวดหมู่ดัชนีระบุผลลัพธ์ของการทำงาน ดังรูปที่ 3.8

```
37 label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
38 categories = label_map_util.convert_label_map_to_categories(
39     label_map, max_num_classes=NUM_CLASSES, use_display_name=True)
40 category_index = label_map_util.create_category_index(categories)
```

รูปที่ 3.8 การแปลง object-detection.pbtxt ให้เป็นหมวดหมู่ดัชนีซึ่งผลลัพธ์ของการทำงาน

7. กำหนดตำแหน่งไฟล์รูปภาพอินพุต และกำหนดขนาดรูปภาพที่จะนำไปใช้ในขั้นตอนแสดงผลลัพธ์ ดังรูปที่ 3.9

```
45 PATH_TO_TEST_IMAGES_DIR = 'png_tesseract/test_tesseract'
46 TEST_IMAGE_PATHS = [os.path.join(PATH_TO_TEST_IMAGES_DIR, '{}.jpg'.format(i)) for i in range(1,8)]
```

รูปที่ 3.9 กำหนดตำแหน่งไฟล์รูปภาพที่ใช้ทดสอบ และการกำหนด count

8. เรียกให้ปัญญาประดิษฐ์ Tensorflow Object-Detection API ทำงานตรวจจับป้ายทะเบียนรถยนต์จากอินพุตที่กำหนดไว้ ดังรูปที่ 3.10

```

51  with detection_graph.as_default():
52    with tf.Session(graph=detection_graph) as sess:
53      image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
54      detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
55      detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
56      detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')
57      num_detections = detection_graph.get_tensor_by_name('num_detections:0')
58    for image_path in TEST_IMAGE_PATHS:
59      image = Image.open(image_path)
60      image_np = load_image_into_numpy_array(image)
61      image_np_expanded = np.expand_dims(image_np, axis=0)
62      boxes, scores, classes, num = sess.run(
63          [detection_boxes, detection_scores, detection_classes, num_detections]
64          ,feed_dict={image_tensor: image_np_expanded})

```

รูปที่ 3.10 เริ่มให้โปรแกรมตรวจจับป้ายทะเบียนรถยนต์

9. crop ป้ายทะเบียนรถยนต์ที่ปัญญาประดิษฐ์ตรวจพบ โดยในบรรทัดที่ 75 เป็นการเรียกใช้ฟังก์ชันแปลงภาพที่ถูก crop ไปเป็นรูปภาพเพื่อนำไปเป็นอินพุตของโปรแกรมอ่านป้ายทะเบียนรถยนต์ และบรรทัดที่ 76 คือการเรียกใช้โปรแกรมอ่านป้ายทะเบียนรถยนต์ ดังรูปที่ 3.11

```

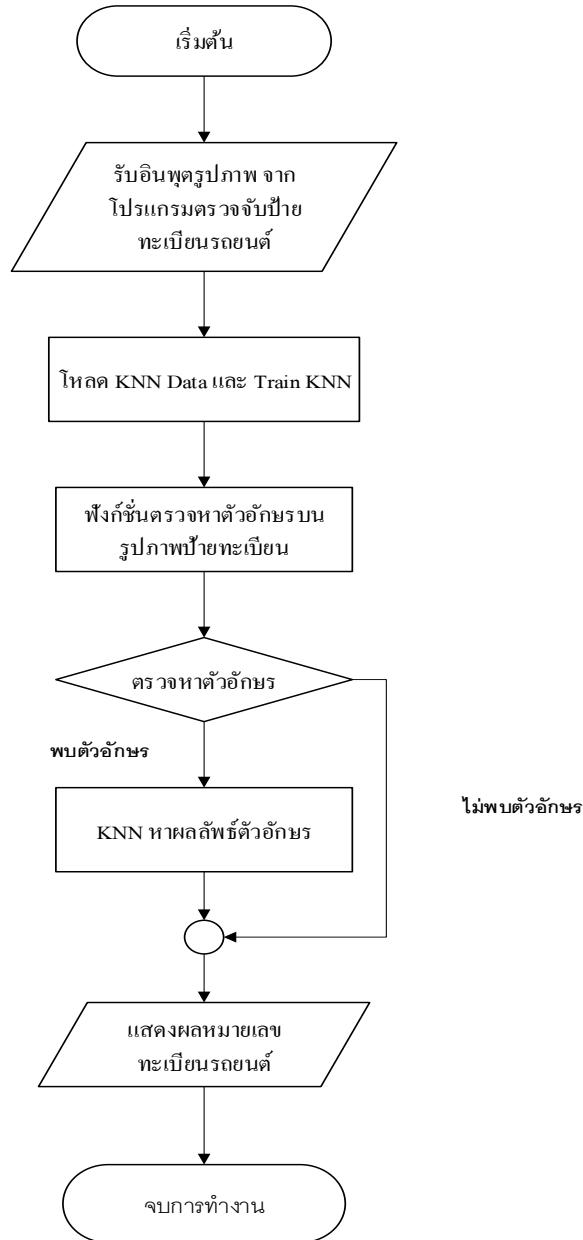
65  ymin = boxes[0,0,0]
66  xmin = boxes[0,0,1]
67  ymax = boxes[0,0,2]
68  xmax = boxes[0,0,3]
69  (im_width, im_height) = image.size
70  (xminn, xmaxx, yminn, ymaxxx) = (xmin * im_width, xmax * im_width,
71  ymin * im_height, ymax * im_height)
72  cropped_image = tf.image.crop_to_bounding_box(image_np, int(yminn),
73  int(xminn),int(ymaxx - yminn), int(xmaxx - xminn))
74  img_data2 = sess.run(cropped_image)
75  filename2 = save_image(img_data2)
76  test_Readplate.main()

```

รูปที่ 3.11 crop ป้ายทะเบียนรถยนต์

### 3.1.3 การออกแบบโปรแกรมอ่านป้ายทะเบียนรถยนต์

โปรแกรมอ่านป้ายทะเบียนรถยนต์จะรับอินพุตรูปภาพป้ายทะเบียนรถยนต์ที่ถูกตรวจพบโดยโปรแกรมตรวจจับป้ายทะเบียนรถยนต์ นำรูปภาพนั้นมาผ่านกระบวนการตรวจสอบหาตัวอักษร และใช้โมเดล Machine Learning โดยใช้หลักการที่มีชื่อว่า k Nearest Neighbor (kNN) เพื่อมาแยกและตัวอักษร และแสดงผลหมายเลขอ่านป้ายทะเบียนรถยนต์ โดย Flowchart ขั้นตอนการทำงานของโปรแกรมอ่านป้ายทะเบียนรถยนต์แสดงดังรูปที่ 3.12



รูปที่ 3.12 ขั้นตอนการทำงานของโปรแกรม อ่านป้ายทะเบียนรถยนต์

โปรแกรมอ่านป้ายทะเบียนรถยนต์ประกอบไปด้วยการทำงานร่วมกันของ 4 ไฟล์ได้แก่

1. test\_Readplate.py คือไฟล์โปรแกรมหลักที่ไว้รับอินพุตรูปภาพป้ายทะเบียนรถยนต์ที่ถูกตรวจพบโดยปัญญาประดิษฐ์, เรียกใช้ฟังก์ชัน โหลด KNN Data Train KNN, เรียกใช้ฟังก์ชันตรวจหาตัวอักษรและอ่านป้ายทะเบียนรถยนต์ จากไฟล์ test\_char.py ซึ่งคือไฟล์ที่ 2 ของโปรแกรมอ่านป้ายทะเบียนรถยนต์ และแสดงผลหมายเลขอ่านได้พร้อมส่งข้อมูลขึ้นคลาวด์

อธิบายโค้ดของ test\_Readplate.py

1. การนำเข้าไฟล์라이บรารีต่าง ๆ ที่ต้องใช้ โดยในบรรทัดที่ 4 คือการนำเข้าไฟล์ test\_char.py ซึ่งเป็นไฟล์ที่ 2 ของโปรแกรมอ่านป้ายทะเบียนรถยนต์ดังรูปที่ 3.13

```

1 import cv2
2 import numpy as np
3 import os
4 import test_char
5 import paho.mqtt.client as mqtt

```

รูปที่ 3.13 การนำเข้าไฟล์라이บรารีที่ต้องใช้

2. กำหนดตัวแปรที่ไว้ใช้ โดยในบรรทัดที่ 20 ตัวแปร showSteps หากเป็น True โปรแกรมอ่านป้ายทะเบียนรถยนต์จะแสดงผลลัพธ์การทำงานของแต่ละขั้นตอน ดังรูปที่ 3.14

```

14 SCALAR_BLACK = (0.0, 0.0, 0.0)
15 SCALAR_WHITE = (255.0, 255.0, 255.0)
16 SCALAR_YELLOW = (0.0, 255.0, 255.0)
17 SCALAR_GREEN = (0.0, 255.0, 0.0)
18 SCALAR_RED = (0.0, 0.0, 255.0)
19
20 showSteps = False

```

รูปที่ 3.14 การกำหนดตัวแปร

3. กำหนดฟังก์ชัน main โดยในบรรทัดที่ 25 คือการเรียกใช้ฟังก์ชัน loadKNNDataAndTrainKNN ของไฟล์ test\_char.py และในบรรทัดที่ 32 คือการรับอินพุตรูปป้ายทะเบียนรถยนต์ที่ปัจจุบันประดิษฐ์ตรวจพบ ดังรูปที่ 3.15

```

23 def main():
24
25     blnKNNTrainingSuccessful = test_char.loadKNNDataAndTrainKNN()
26
27     if blnKNNTrainingSuccessful == False:
28         print("\nerror: KNN training was not successful\n")
29         return
30     # end if
31
32     imgPlate = cv2.imread("images1.JPG")
33
34     if imgPlate is None:
35         print("\nerror: image not read from file \n\n")
36         os.system("pause")
37         return
38     # end if

```

รูปที่ 3.15 โค๊ดในฟังก์ชัน main

4. จากรูปที่ 3.16 บรรทัดที่ 40 คือการเรียกใช้ฟังก์ชัน detectCharsInPlates โดยใช้อินพุต เป็นรูปป้ายทะเบียนรถยนต์ และในบรรทัดที่ 44 คือการแสดงผลหมายเลขอทะเบียนรถยนต์

```

40     listOfPossiblePlates, License_Number = test_char.detectCharsInPlates(imgPlate)
41
42     cv2.imshow("imgPlate", imgPlate)
43
44     print('\nLicense Number is ', License_Number)

```

รูปที่ 3.16 โค้ดในฟังก์ชัน main

2. test\_char.py คือไฟล์ที่สองของโปรแกรมอ่านป้ายทะเบียนรถยนต์ มีหน้าที่สำคัญในการ โหลดและTrain KNN Model, ตรวจหาตัวอักษร, กรองตัวอักษร และอ่านตัวอักษร

อธิบายโค้ดของ test\_char.py เนื่องจากโค้ดซับซ้อน ผู้จัดทำจึงจะอธิบายฟังก์ชันย่อยก่อนที่ จะอธิบายฟังก์ชันหลัก (detectCharsInPlates) ฉะนั้น บรรทัดในรูปภาพที่จะแสดงอาจมีการสลับไปมา

1. การนำเข้าไลบรารี และโมดูลต่าง ๆ ที่ต้องใช้ โดยในบรรทัดที่ 8 คือการนำเข้าไฟล์ test\_Readplate บรรทัดที่ 9 คือการนำเข้าไฟล์ Preprocess.py และบรรทัดที่ 10 คือการนำเข้าไฟล์ PossibleChar.py ซึ่งทั้งสามไฟล์คือไฟล์ที่ 1, 3 และ 4 ของโปรแกรมอ่านป้ายทะเบียนรถยนต์ ตามลำดับ ดังรูปที่ 3.17

```

1 import os
2
3 import cv2
4 import numpy as np
5 import math
6 import random
7
8 import test_Readplate
9 import Preprocess
10 import PossibleChar

```

รูปที่ 3.17 การนำเข้าไลบรารี และโมดูลต่าง ๆ

2. การกำหนดตัวแปรต่าง ๆ ที่ต้องใช้ในภายหลัง โดยในบรรทัดที่ 14 คือการสร้าง โมเดล KNN ที่ว่างเปล่าไว้ ดังรูปที่ 3.18

```

14 kNearest = cv2.ml.KNearest_create()
15
16
17 MIN_PIXEL_WIDTH = 2
18 MIN_PIXEL_HEIGHT = 8
19
20 MIN_ASPECT_RATIO = 0.25
21 MAX_ASPECT_RATIO = 1.0
22
23 MIN_PIXEL_AREA = 60
24
25
26 MIN_DIAG_SIZE_MULTIPLE_AWAY = 0.3
27 MAX_DIAG_SIZE_MULTIPLE_AWAY = 5.0
28
29 MAX_CHANGE_IN_AREA = 0.5
30
31 MAX_CHANGE_IN_WIDTH = 0.8
32 MAX_CHANGE_IN_HEIGHT = 0.2
33
34 MAX_ANGLE_BETWEEN_CHARS = 12.0
35
36
37 MIN_NUMBER_OF_MATCHING_CHARS = 3
38
39 RESIZED_CHAR_IMAGE_WIDTH = 25
40 RESIZED_CHAR_IMAGE_HEIGHT = 40
41
42 MIN_CONTOUR_AREA = 100

```

รูปที่ 3.18 การกำหนดตัวแปรต่าง ๆ

3. จากรูปที่ 3.19 กำหนดฟังก์ชัน loadKNNDATAAndTrainKNN คือฟังก์ชันที่เรียกไฟล์ classifications.txt และ flattened\_images.txt ที่ได้จากการฝึกฝน KNN นำมาสร้างโมเดล KNearest เพื่อใช้อ่านตัวอักษรต่อไป

```

45 def loadKNNDATAAndTrainKNN():
46     allContoursWithData = []
47     validContoursWithData = []
48
49     try:
50         npaClassifications = np.loadtxt("classifications.txt", np.float32)
51     except:
52         print("error, unable to open classifications.txt, exiting program\n")
53         os.system("pause")
54         return False
55     # end try
56
57     try:
58         npaFlattenedImages = np.loadtxt("flattened_images.txt", np.float32)
59     except:
60         print("error, unable to open flattened_images.txt, exiting program\n")
61         os.system("pause")
62         return False
63     # end try
64
65     npaClassifications = npaClassifications.reshape((npaClassifications.size, 1))
66
67     kNearest.setDefaultK(1)
68
69     kNearest.train(npaFlattenedImages, cv2.ml.ROW_SAMPLE, npaClassifications)
70
71     return True

```

รูปที่ 3.19 ฟังก์ชัน loadKNNDATAAndTrainKNN

4. จากรูปที่ 3.20 กำหนดฟังก์ชันย่ออย findPossibleCharsInPlate คือฟังก์ชันหา contours โดยในบรรทัดที่ 235 คือการเรียกไฟล์ Possiblechar ซึ่งเป็นไฟล์ที่ 4 ของโปรแกรมอ่านป้ายทะเบียน รดยนต์เพื่อนำ contour แต่ละอันที่เจอไปเข้า Class Possiblechar คำอธิบายโค้ดจะกล่าวถึงในคำอธิบายโค้ดของไฟล์ Possiblechar.py ในบรรทัดที่ 237 คือการเรียกใช้ฟังก์ชันย่ออย checkIfPossibleChars ตรวจสอบเงื่อนไขตัวอักษร ถ้าเป็นตัวอักษร จะเพิ่ม Possiblechar นั้นเข้าไปใน listOfPossibleChars

```

226 def findPossibleCharsInPlate(imgGrayscale, imgThresh):
227     listOfPossibleChars = []
228     contours = []
229     imgThreshCopy = imgThresh.copy()
230
231     imgContours, contours, npaHierarchy = cv2.findContours(imgThreshCopy, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
232
233     for contour in contours:
234         possibleChar = PossibleChar.PossibleChar(contour)
235
236         if checkIfPossibleChar(possibleChar):
237             listOfPossibleChars.append(possibleChar)
238
239     return listOfPossibleChars
240

```

รูปที่ 3.20 ฟังก์ชันย่ออย findPossibleCharsInPlate

5. จากรูปที่ 3.21 กำหนดฟังก์ชันย่ออย checkIfPossibleChar คือ การตรวจสอบว่า contours แต่ละอันมีความเป็นไปได้ที่จะเป็นตัวอักษรหรือไม่ โดยใช้การกำหนดเงื่อนไขขนาดต่าง ๆ เทียบกับตัวแปรที่กำหนดไว้แล้วในข้อที่ 2 ตรวจสอบว่าตรงเงื่อนไขหรือไม่ ถ้าตรงเงื่อนไขทุกอย่าง ก็อตัวอักษร

```

243 def checkIfPossibleChar(possibleChar):
244
245     if (possibleChar.intBoundingRectArea > MIN_PIXEL_AREA and
246         possibleChar.intBoundingRectWidth > MIN_PIXEL_WIDTH and
247         possibleChar.intBoundingRectHeight > MIN_PIXEL_HEIGHT and
248         MIN_ASPECT_RATIO < possibleChar.fltAspectRatio and possibleChar.fltAspectRatio
249         < MAX_ASPECT_RATIO):
250         return True
251     else:
252         return False

```

รูปที่ 3.21 กำหนดฟังก์ชันย่ออย checkIfPossibleChar

6. จากรูปที่ 3.22 กำหนดฟังก์ชันย่ออย distanceBetweenChars คือฟังก์ชันที่ทำหน้าที่ คำนวณระยะห่างระหว่างตัวอักษร

```

326     def distanceBetweenChars(firstChar, secondChar):
327         intX = abs(firstChar.intCenterX - secondChar.intCenterX)
328         intY = abs(firstChar.intCenterY - secondChar.intCenterY)
329
330         return math.sqrt((intX ** 2) + (intY ** 2))

```

รูปที่ 3.22 ฟังก์ชันย่อของ distanceBetweenChars

7. จากรูปที่ 3.23 กำหนดฟังก์ชันย่อของ angleBetweenChars คือฟังก์ชันที่ทำหน้าที่คำนวณ  
มุมระหว่างตัวอักษร

```

335     def angleBetweenChars(firstChar, secondChar):
336         fltAdj = float(abs(firstChar.intCenterX - secondChar.intCenterX))
337         fltOpp = float(abs(firstChar.intCenterY - secondChar.intCenterY))
338
339         if fltAdj != 0.0:
340             fltAngleInRad = math.atan(fltOpp / fltAdj)
341         else:
342             fltAngleInRad = 1.5708
343             # end if
344
345         fltAngleInDeg = fltAngleInRad * (180.0 / math.pi)
346
347         return fltAngleInDeg

```

รูปที่ 3.23 ฟังก์ชันย่อของ angleBetweenChars

8. จากรูปที่ 3.24 กำหนดฟังก์ชันย่อของ findListOfMatchingChars คือฟังก์ชันที่ตรวจสอบ  
contour ที่อาจจะเป็นตัวอักษรทุกตัวใน listOfPossibleChars ว่ามี contour ใดที่คือตัวอักษรจริง ๆ ใน  
บรรทัดที่ 301 ถึง 303 คือการเรียกใช้ฟังก์ชันในข้อที่ 6 และ 7 ในบรรทัดที่ 305 ถึง 308 คือการสร้าง  
สมการตัวแปรใหม่ขึ้นมา เพื่อนำไปใช้สร้างเงื่อนไขพิสูจน์ตัวอักษร ตามบรรทัดที่ 311 ถึง 315 หาก  
ผ่านเงื่อนไขจะนำตัวอักษรไปเพิ่มใน listOfMatchingChars

```

291     def findListOfMatchingChars(possibleChar, listOfChars):
292
293         listOfMatchingChars = []
294
295         for possibleMatchingChar in listOfChars:
296             if possibleMatchingChar == possibleChar:
297
298                 continue
299             # end if
300
301             fltDistanceBetweenChars = distanceBetweenChars(possibleChar, possibleMatchingChar)
302
303             fltAngleBetweenChars = angleBetweenChars(possibleChar, possibleMatchingChar)
304
305             fltChangeInArea = float(abs(possibleMatchingChar.intBoundingRectArea - possibleChar.intBoundingRectArea)) / float(possibleChar.intBoundingRectArea)
306
307             fltChangeInWidth = float(abs(possibleMatchingChar.intBoundingRectWidth - possibleChar.intBoundingRectWidth)) / float(possibleChar.intBoundingRectWidth)
308             fltChangeInHeight = float(abs(possibleMatchingChar.intBoundingRectHeight - possibleChar.intBoundingRectHeight)) / float(possibleChar.intBoundingRectHeight)
309
310
311             if (fltDistanceBetweenChars < (possibleChar.fltDiagonalSize * MAX_DIAG_SIZE_MULTIPLE_AWAY) and
312                 fltAngleBetweenChars < MAX_ANGLE_BETWEEN_CHARS and
313                 fltChangeInArea < MAX_CHANGE_IN_AREA and
314                 fltChangeInWidth < MAX_CHANGE_IN_WIDTH and
315                 fltChangeInHeight < MAX_CHANGE_IN_HEIGHT):
316
317                 listOfMatchingChars.append(possibleMatchingChar)
318             # end if
319         # end for
320
321     return listOfMatchingChars           # return result

```

### รูปที่ 3.24 พังก์ชั่นย่อย findListOfMatchingChars

9. จากรูปที่ 3.25 กำหนดพังก์ชั่นย่อย `findListOfListsOfMatchingChars` คือพังก์ชั่นที่เรียกใช้พังก์ชั่นย่อย `findListOfMatchingChars` ในข้อที่ 8 ในบรรทัดที่ 265 คือการเช็คเพื่อให้ `listOfMatchingChars` มีตัวอักษรไม่ต่างกันมากกว่าตัวแปรที่กำหนดไว้ในข้อที่ 2 ซึ่งกำหนดไว้ 3 ตัวอักษรกล่าวคือหาก `listOfMatchingChars` มีตัวอักษรน้อยกว่าสามตัวจะไม่นำมาพิจารณาไว้แล้วนั่นคือ ตัวอักษรป้ายทะเบียนรถยนต์ ในบรรทัดที่ 271 หากมีตัวอักษรเกินสามตัว จะเพิ่มกลุ่มตัวอักษรนั้นเข้าไปใน `listOfListsOfMatchingChars` ในบรรทัดที่ 273 ถึง 284 คือการสร้าง `List` ใหม่เพื่อเรียกใช้ recursive function (พังก์ชั่นวน-ซ้ำ) เพื่อตรวจสอบหาตัวอักษร และกลุ่มของตัวอักษรอีกรึ้ง จนนั้นผลลัพธ์ของพังก์ชั่น `findListOfListsOfMatchingChars` คือการได้ `List` ของกลุ่มอักษรที่เป็นป้ายทะเบียนรถยนต์ `listOfListsOfMatchingChars`

```

256 def findListOfListsOfMatchingChars(listOfPossibleChars):
257     listOfListsOfMatchingChars = []
258
259     for possibleChar in listOfPossibleChars:
260         listOfMatchingChars = findListOfMatchingChars(possibleChar, listOfPossibleChars)
261
262         listOfMatchingChars.append(possibleChar)
263
264         if len(listOfMatchingChars) < MIN_NUMBER_OF_MATCHING_CHARS:
265             continue
266
267         # end if
268
269         listOfListsOfMatchingChars.append(listOfMatchingChars)
270
271     listOfPossibleCharsWithCurrentMatchesRemoved = []
272
273     listOfPossibleCharsWithCurrentMatchesRemoved = list(set(listOfPossibleChars) - set(listOfMatchingChars))
274
275     recursiveListOfListsOfMatchingChars = findListOfListsOfMatchingChars(listOfPossibleCharsWithCurrentMatchesRemoved)
276
277     for recursiveListOfMatchingChars in recursiveListOfListsOfMatchingChars:
278         listOfListsOfMatchingChars.append(recursiveListOfMatchingChars)
279     # end for
280
281     break      # exit for
282
283     # end for
284
285     return listOfListsOfMatchingChars
286
287
288

```

### รูปที่ 3.25 พังก์ชันย่อย findListOfListsOfMatchingChars

10. จากรูปที่ 3.26 กำหนดพังก์ชันย่อย removeInnerOverlappingChars คือพังก์ชันที่ใช้ป้องกันการเพิ่มตัวอักษรเดิมซ้ำจากการที่ contour ทับซ้อนกัน เช่น 0 วงแหวนด้านใน และด้านนอกอาจถูกพบเป็น contour สองอันที่ทับซ้อนกัน แต่เป็นตัวอักษรตัวเดียวจึงตัด contour ด้านในออก (contour ที่เล็กกว่า) เพื่อป้องกันการเพิ่มอักษรเดิมสองครั้ง

```

352 def removeInnerOverlappingChars(listOfMatchingChars):
353     listOfMatchingCharsWithInnerCharRemoved = list(listOfMatchingChars)
354
355     for currentChar in listOfMatchingChars:
356         for otherChar in listOfMatchingChars:
357             if currentChar != otherChar:
358
359                 if distanceBetweenChars(currentChar, otherChar) < (currentChar.fltDiagonalSize * MIN_DIAG_SIZE_MULTIPLE_AWAY):
360
361                     if currentChar.intBoundingRectArea < otherChar.intBoundingRectArea:
362                         if currentChar in listOfMatchingCharsWithInnerCharRemoved:
363                             listOfMatchingCharsWithInnerCharRemoved.remove(currentChar)
364                         # end if
365                     else:
366                         if otherChar in listOfMatchingCharsWithInnerCharRemoved:
367                             listOfMatchingCharsWithInnerCharRemoved.remove(otherChar)
368                         # end if
369                     # end if
370                 # end if
371             # end if
372         # end for
373     # end for
374
375     return listOfMatchingCharsWithInnerCharRemoved
376
377
378

```

### รูปที่ 3.26 พังก์ชันย่อย removeInnerOverlappingChars

11. จากรูปที่ 3.27 กำหนดฟังก์ชัน recognizeCharsInPlate คือฟังก์ชันที่เรียกใช้ KNN Model เพื่ออ่านตัวอักษรป้ายทะเบียนรถยนต์ โดยรับอินพุตเป็นภาพ Threshold ของป้ายทะเบียนรถยนต์ และ listOfListsOfMatchingChars ที่ผ่านฟังก์ชัน removeInnerOverlappingChars แล้วในบรรทัดที่ 386 เรียงตัวอักษรจากซ้ายไปขวา ตั้งแต่บรรทัดที่ 390 อยู่ใน for loop ทำกับทุกอักษรบนแผ่นป้ายทะเบียนรถยนต์ บรรทัดที่ 395 ตีกรอบรอบตัวอักษรบนป้ายทะเบียนรถยนต์ บรรทัดที่ 397 crop ตัวอักษรออกมา บรรทัดที่ 402 ถึง 406 คือการปรับขนาดรูปภาพตัวอักษรใหม่, แปลงรูปอักษรเป็น 1d numpy array และแปลง 1d numpy array int เป็น 1d numpy array float บรรทัดที่ 408 ถึง 412 เรียกใช้ Model KNN และอ่านตัวอักษร

```

379 def recognizeCharsInPlate(imgThresh, listOfMatchingChars):
380     strChars = ""
381
382     height, width = imgThresh.shape
383
384     imgThreshColor = np.zeros((height, width, 3), np.uint8)
385
386     listOfMatchingChars.sort(key = lambda matchingChar: matchingChar.intCenterX)
387
388     cv2.cvtColor(imgThresh, cv2.COLOR_GRAY2BGR, imgThreshColor)
389
390     for currentChar in listOfMatchingChars:
391         pt1 = (currentChar.intBoundingRectX, currentChar.intBoundingRectY)
392         pt2 = ((currentChar.intBoundingRectX + currentChar.intBoundingRectWidth),
393                (currentChar.intBoundingRectY + currentChar.intBoundingRectHeight))
394
395         cv2.rectangle(imgThreshColor, pt1, pt2, test_Readplate.SCALAR_GREEN, 2)
396
397         imgROI = imgThresh[currentChar.intBoundingRectY : currentChar.intBoundingRectY +
398                             currentChar.intBoundingRectHeight,
399                             currentChar.intBoundingRectX : currentChar.intBoundingRectX +
400                             currentChar.intBoundingRectWidth]
401
402         imgROIResized = cv2.resize(imgROI, (RESIZED_CHAR_IMAGE_WIDTH, RESIZED_CHAR_IMAGE_HEIGHT))
403
404         npaROIResized = imgROIResized.reshape((1, RESIZED_CHAR_IMAGE_WIDTH * RESIZED_CHAR_IMAGE_HEIGHT))
405
406         npaROIResized = np.float32(npaROIResized)
407
408         retval, npaResults, neigh_resp, dists = kNearest.findNearest(npaROIResized, k = 1)
409
410         strCurrentChar = str(chr(int(npaResults[0][0])))
411
412         strChars = strChars + strCurrentChar
413     # end for
414     if test_Readplate.showSteps == True: # show steps #####
415     # end if # show steps #####
416
417     return strChars
418

```

รูปที่ 3.27 ฟังก์ชัน recognizeCharsInPlate

12. กำหนดฟังก์ชันหลัก detectCharsInPlates รับอินพุตเป็นรูปภาพป้ายทะเบียนรถยนต์ที่ถูกตรวจพบ โดยปัญญาประดิษฐ์ ในบรรทัดที่ 82 เรียกใช้ฟังก์ชัน preprocess จากไฟล์ Preprocess ซึ่งเป็นไฟล์ที่ 3 ของโปรแกรมอ่านป้ายทะเบียนรถยนต์ บรรทัดที่ 91 นำรูปภาพที่ผ่านการ preprocess มาหลาย และทำ Threshold อีกครั้งเพื่อกำจัดส่วนสีเทาออกไป ดังรูปที่ 3.28

```

77 def detectCharsInPlates(imgPlate):
78     intPlateCounter = 0
79     imgContours = None
80     contours = []
81
82     imgGrayscale, imgThresh = Preprocess.preprocess(imgPlate)
83
84     if test_Readplate.showSteps == True: # show steps #####
85         # end if # show steps #####
86
87
88     imgThresh = cv2.resize(imgThresh, (0, 0), fx = 3, fy = 3)
89
90
91     thresholdValue, imgThresh = cv2.threshold(imgThresh, 0.0, 255.0, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
92
93
94     if test_Readplate.showSteps == True: # show steps #####
95         # end if # show steps #####

```

รูปที่ 3.28 โค้ดในฟังก์ชัน detectCharsInPlates

13. จากรูปที่ 3.29 บรรทัดที่ 102 เรียกใช้ฟังก์ชันย่อย findPossibleCharsInPlate ในข้อที่ 4 โดยป้อนอินพุตเป็น รูปภาพ Grayscale และ รูปภาพ Threshold ที่ได้จากการข้อที่ 12 บรรทัดที่ 119 นำผลลัพธ์จากฟังก์ชัน findPossibleCharsInPlate ไปเข้าฟังก์ชัน findListOfListsOfMatchingChars ในข้อที่ 8

```

102     listOfPossibleCharsInPlate = findPossibleCharsInPlate(imgGrayscale, imgThresh)
103
104     if test_Readplate.showSteps == True: # show steps #####
105         # end if # show steps #####
106
107
108     listOfListsOfMatchingCharsInPlate = findListOfListsOfMatchingChars(listOfPossibleCharsInPlate)
109
110     if test_Readplate.showSteps == True: # show steps #####
111         # end if # show steps #####

```

รูปที่ 3.29 โค้ดในฟังก์ชัน detectCharsInPlates

14. จากรูปที่ 3.30 บรรทัดที่ 138 ถ้าไม่พบตัวอักษรบนรูปภาพป้ายทะเบียนรถยนต์เลย ให้ส่งตัวแปร strChars ที่มีค่าคือ string ‘Not found license number’ แต่ถ้าพบตัวอักษรในรูปภาพป้าย ทะเบียนรถยนต์จะทำให้เป็นไปตามเงื่อนไข elif ในบรรทัดที่ 155 คือ for loop เพื่อใช้ฟังก์ชัน removeInnerOverlappingChars ในข้อที่ 10

```

138     if (len(listOfListsOfMatchingCharsInPlate) == 0):
139
140         if test_Readplate.showSteps == True: # show steps #####
141             # end if # show steps #####
142
143         strChars = "Not found license number"
144         return imgPlate,strChars
145
146     elif not (len(listOfListsOfMatchingCharsInPlate) == 0):
147         for i in range(0, len(listOfListsOfMatchingCharsInPlate)):
148             listOfListsOfMatchingCharsInPlate[i].sort(key = lambda matchingChar: matchingChar.intCenterX)
149             listOfListsOfMatchingCharsInPlate[i] = removeInnerOverlappingChars(listOfListsOfMatchingCharsInPlate[i])
150             # end for
151
152         if test_Readplate.showSteps == True: # show steps #####
153             # end if # show steps #####

```

รูปที่ 3.30 โค้ดในฟังก์ชัน detectCharsInPlates

15. จากรูปที่ 3.31 คือการคัดเลือกกลุ่มของตัวอักษรที่ยาวที่สุด โดยใช้การเปรียบเทียบจำนวนของตัวอักษรในแต่ละกลุ่ม

```

180     intLenOfLongestListOfChars = 0
181     intIndexOfLongestListOfChars = 0
182
183
184     for i in range(0, len(listOfListsOfMatchingCharsInPlate)):
185         if len(listOfListsOfMatchingCharsInPlate[i]) > intLenOfLongestListOfChars:
186             intLenOfLongestListOfChars = len(listOfListsOfMatchingCharsInPlate[i])
187             intIndexOfLongestListOfChars = i
188         # end if
189     # end for
190
191     # suppose that the longest list of matching chars within the plate is the actual list of chars
192     longestListOfMatchingCharsInPlate = listOfListsOfMatchingCharsInPlate[intIndexOfLongestListOfChars]
193
194     if test_Readplate.showSteps == True: # show steps #####
195         # end if # show steps #####

```

รูปที่ 3.31 โค้ดในฟังก์ชัน detectCharsInPlates

16. จากรูปที่ 3.32 เรียกใช้ฟังก์ชัน recognizeCharsInPlate ในข้อที่ 11 เพื่ออ่านหมายเลขทะเบียน และส่งค่า imgPlate และstrChars (หมายเลขทะเบียน) ออกໄປ

```

207     strChars = recognizeCharsInPlate(imgThresh, longestListOfMatchingCharsInPlate)
208
209     if test_Readplate.showSteps == True: # show steps #####
210         # end if # show steps #####
211
212
213     if test_Readplate.showSteps == True:
214
215         return imgPlate,strChars

```

รูปที่ 3.32 โค้ดในฟังก์ชัน detectCharsInPlates

3. Preprocess.py มีหน้าที่ทำ Image Processing รูปภาพหมายเลขทะเบียนโดยนั้นต้องตรวจสอบเพื่อนำไปใช้ในกระบวนการอ่านตัวอักษรต่อไป

อธิบายโค้ดของ Preprocess.py

1. นำเข้าไลบรารีที่ต้องงาน และกำหนดตัวแปร arguments ของ opencv ดังรูปที่ 3.33

```

3 import cv2
4 import numpy as np
5 import math
6
7 # module level variables #####
8 GAUSSIAN_SMOOTH_FILTER_SIZE = (5, 5)
9 ADAPTIVE_THRESH_BLOCK_SIZE = 19
10 ADAPTIVE_THRESH_WEIGHT = 9

```

รูปที่ 3.33 การนำเข้าไลบรารี และกำหนดตัวแปร

2. จากรูปที่ 3.34 กำหนดฟังก์ชัน preprocess ทำหน้าที่ Image Processing รูปภาพป้ายทะเบียนรถยนต์

```

13  def preprocess(imgOriginal):
14      imgGrayscale = extractValue(imgOriginal)
15
16      imgMaxContrastGrayscale = maximizeContrast(imgGrayscale)
17
18      height, width = imgGrayscale.shape
19
20      imgBlurred = np.zeros((height, width, 1), np.uint8)
21
22      imgBlurred = cv2.GaussianBlur(imgMaxContrastGrayscale, GAUSSIAN_SMOOTH_FILTER_SIZE, 0)
23
24      imgThresh = cv2.adaptiveThreshold(imgBlurred, 255.0, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
25          cv2.THRESH_BINARY_INV, ADAPTIVE_THRESH_BLOCK_SIZE, ADAPTIVE_THRESH_WEIGHT)
26
27      return imgGrayscale, imgThresh
28  # end function

```

รูปที่ 3.34 ฟังก์ชัน preprocess

3. จากรูปที่ 3.35 กำหนดฟังก์ชัน extractValue ทำหน้าที่เปลี่ยน Color space จาก BGR เป็น HSV และแยกค่า Hue, Saturation และ Value ออกมา เป็นส่วนหนึ่งของ Image Processing

```

31  def extractValue(imgOriginal):
32      height, width, numChannels = imgOriginal.shape
33
34      imgHSV = np.zeros((height, width, 3), np.uint8)
35
36      imgHSV = cv2.cvtColor(imgOriginal, cv2.COLOR_BGR2HSV)
37
38      imgHue, imgSaturation, imgValue = cv2.split(imgHSV)
39
40      return imgValue
41  # end function

```

รูปที่ 3.35 ฟังก์ชัน extractValue

4. จากรูปที่ 3.36 กำหนดฟังก์ชัน `maximizeContrast` ทำหน้าที่ใช้กระบวนการ Morphological Transformations โดยใช้ operator tophat และ blackhat จากนั้นหาผลต่างของ array tophat และ blackhat ซึ่งเป็นส่วนหนึ่งของ Image Processing

```

44 def maximizeContrast(imgGrayscale):
45     height, width = imgGrayscale.shape
46
47     imgTopHat = np.zeros((height, width, 1), np.uint8)
48     imgBlackHat = np.zeros((height, width, 1), np.uint8)
49
50     structuringElement = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
51
52     imgTopHat = cv2.morphologyEx(imgGrayscale, cv2.MORPH_TOPHAT, structuringElement)
53     imgBlackHat = cv2.morphologyEx(imgGrayscale, cv2.MORPH_BLACKHAT, structuringElement)
54
55     imgGrayscalePlusTopHat = cv2.add(imgGrayscale, imgTopHat)
56     imgGrayscalePlusTopHatMinusBlackHat = cv2.subtract(imgGrayscalePlusTopHat, imgBlackHat)
57
58     return imgGrayscalePlusTopHatMinusBlackHat
59
60 # end function

```

รูปที่ 3.36 ฟังก์ชัน `extractValue`

4. Possiblechar.py คือไฟล์ที่กำหนด Class Possiblechar

อธิบายโดย大概ของ Possiblechar.py

1. นำเข้าไลบรารีที่ต้องใช้ ดังรูปที่ 3.37

```

3 import cv2
4 import numpy as np
5 import math

```

รูปที่ 3.37 การนำเข้าไลบรารีที่ต้องใช้

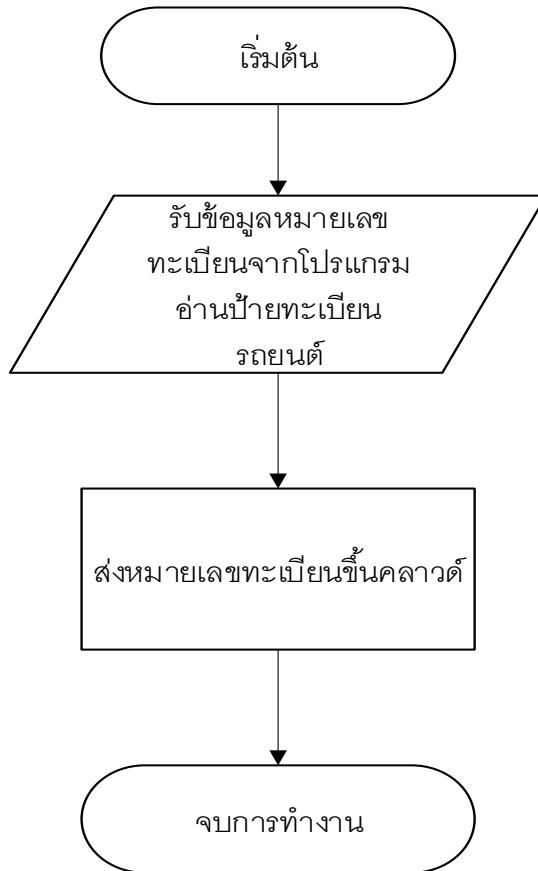
2. จากรูปที่ 3.38 กำหนด Class PossibleChar ที่ประกอบด้วยฟังก์ชัน `_init_` มีหน้าที่นำ contours ทุกอันที่อาจเป็นตัวอักษร นำมาคำนวณหาขนาด, พื้นที่, จุดศูนย์กลาง, ความยาวเส้นที่แยกนูน, อัตราส่วนของ contours เพื่อนำไปใช้พิจารณาว่าเป็นตัวอักษรหรือไม่ต่อไปในไฟล์ `test_char.py`

```
8  class PossibleChar:  
9  
10 # constructor #####  
11 def __init__(self, _contour):  
12     self.contour = _contour  
13  
14     self.boundingRect = cv2.boundingRect(self.contour)  
15  
16     [intX, intY, intWidth, intHeight] = self.boundingRect  
17  
18     self.intBoundingRectX = intX  
19     self.intBoundingRectY = intY  
20     self.intBoundingRectWidth = intWidth  
21     self.intBoundingRectHeight = intHeight  
22  
23     self.intBoundingRectArea = self.intBoundingRectWidth * self.intBoundingRectHeight  
24  
25     self.intCenterX = (self.intBoundingRectX + self.intBoundingRectX + self.intBoundingRectWidth) / 2  
26     self.intCenterY = (self.intBoundingRectY + self.intBoundingRectY + self.intBoundingRectHeight) / 2  
27  
28     self.fltDiagonalSize = math.sqrt((self.intBoundingRectWidth ** 2) + (self.intBoundingRectHeight ** 2))  
29  
30     self.fltAspectRatio = float(self.intBoundingRectWidth) / float(self.intBoundingRectHeight)  
31 # end constructor  
32  
33 # end class
```

### ຈຸດ໌ 3.38 Class Possible

### 3.1.4 การออกแบบโปรแกรมส่งข้อมูลขึ้นคลาวด์

หลักการทำงานคือ เมื่อโปรแกรมตรวจจับป้ายทะเบียนรถยนต์และอ่านป้ายทะเบียนรถยนต์ทำงานเสร็จแล้ว ข้อมูลที่ได้จะถูกส่งขึ้นคลาวด์ MQTT เพื่อเก็บข้อมูลไว้รองนำไปใช้งาน โดย Flowchart ขั้นตอนการทำงานของโปรแกรมตรวจจับป้ายทะเบียนรถยนต์แสดงดังรูปที่ 3.39



รูปที่ 3.39 แผนผัง Flow Chart การทำงานของโปรแกรมส่งข้อมูลขึ้นคลาวด์

อธิบายโค้ดโปรแกรมส่งข้อมูลขึ้นคลาวด์ โดยโค้ดทั้งหมดที่เกี่ยวกับการส่งขึ้นคลาวด์จะเขียนไว้ในไฟล์ test\_Readplate.py ซึ่งเป็นไฟล์หนึ่งของโปรแกรมอ่านป้ายทะเบียนรถยนต์ในหัวข้อที่ 3.1.3

- กำหนดตัวแปรต่าง ๆ ได้แก่ broker ที่เป็นคลาวด์ MQTT, หมายเลขพอร์ท, username และ password ดังรูปที่ 3.40

```

7 broker="m16.cloudmqtt.com"
8 port=14105
9 username="wvoywesy"
10 password="IJXMMEUFM1bj"

```

รูปที่ 3.40 กำหนดตัวแปร

2. เชื่อมต่อ กับ คลาวด์ MQTT และ ส่ง หมายเลขทะเบียน ขึ้น คลาวด์ ดัง รูปที่ 3.41

```

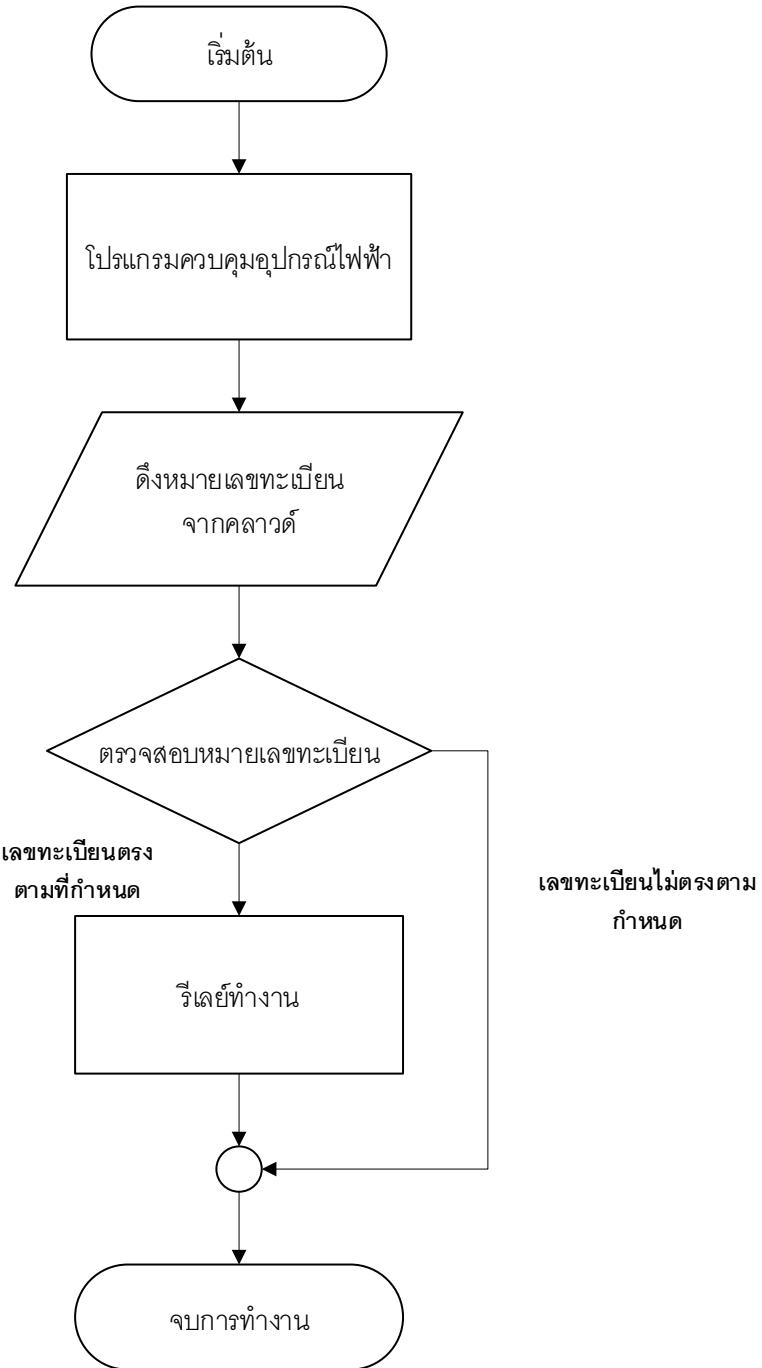
46 client = mqtt.Client()
47 client.username_pw_set(username, password)
48 client.connect(broker, port)
49 client.publish("License Number", License_Number)

```

รูปที่ 3.41 การ เชื่อมต่อ คลาวด์ และ ส่ง หมายเลขทะเบียน ขึ้น คลาวด์

### 3.1.5 การ ออ กแบบ โปรแกรม ควบคุม อุปกรณ์ ไฟฟ้า และ การ แสดงผล

หลักการทำงาน ของการ ควบคุม อุปกรณ์ ไฟฟ้า คือ NodeMCU ตรวจสอบ หมายเลขทะเบียน จาก คลาวด์ หาก หมายเลขทะเบียน ถูกต้อง ตาม ที่ กำหนด NodeMCU จะ สั่ง การ ให้ เรเลย์ ทำงาน หาก ไม่ ถูกต้อง จะ ไม่ สั่ง การ ใด ๆ โดย Flowchart การ ทำงาน ของ การ ควบคุม อุปกรณ์ ไฟฟ้า เป็น ดัง รูปที่ 3.42 อีก ทั้ง สร้าง เว็บแอพพลิเคชัน เพื่อ ควบคุม รีเลย์ ใน กรณี เหตุการณ์ สมมติ ที่ ผู้ อยู่ อาศัย จะ ออกจากบ้าน พัก อาศัย สามารถ ควบคุม รีเลย์ ให้ ปิด ได้ จาก ที่ ได้ กด ได้ ใน ส่วน ของการ แสดงผล มี หลัก การ ทำงาน คือ ใช้ Node-RED ใน การ สร้าง flow เพื่อ ดึง สถานะ จาก คลาวด์ ส่ง ไป ที่ Line Notify เพื่อ ทำ LineNotification และ เว็บ แอพพลิเคชัน ที่ สร้าง เพื่อ ควบคุม อุปกรณ์ ไฟฟ้า ยัง สามารถ แสดงผล สถานะ ข้อมูล ได้



รูปที่ 3.42 แผนผัง Flowchart การทำงานของ โปรแกรมควบคุมอุปกรณ์ไฟฟ้า

อธิบายโดยใช้โปรแกรมควบคุมอุปกรณ์ไฟฟ้า และการแสดงผล

1. ตั้งค่าซื่อและรหัส Wi-Fi และกำหนดตัวแปร ดังรูปที่ 3.43



```

MQTT_ESP8266_Control_Relay
1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>
3
4 //your network
5 const char* ssid = "Wi-Fi NAME";           //ชื่อ Wifi
6 const char* password = "Wi-Fi Password";    //รหัส Wifi
7
8 char *Relay_ON ;
9 char *Relay_OFF ;

```

รูปที่ 3.43 การตั้งค่า Wi-Fi และกำหนดตัวแปร

## 2. ตั้งค่า MQTT ดังรูปที่ 3.44

```

11 //การกำหนด MQTT Server
12 #define MQTTserver "YOUR_MQTT_SERVER"          //กำหนด MQTT Server
13 #define MQTTport YOUR_MQTT_PORT                //กำหนด MQTT Port
14 #define MQTTuser "YOUR_MQTT_USERNAME"           //กำหนด MQTT User Name
15 #define MQTTpassword "YOUR_MQTT_PASSWORD"        //กำหนด MQTT password
16
17 WiFiClient WIFI_Client;
18 PubSubClient client(WIFI_Client);
19

```

รูปที่ 3.44 การตั้งค่า MQTT

## 3. การกำหนดค่าเริ่มต้นของโปรแกรม ดังรูปที่ 3.45

```

19
20 void setup() {
21   pinMode(D0,OUTPUT);      //กำหนดสถานะขา D0 ใช้ในการส่งค่า
22   pinMode(D1,OUTPUT);      //กำหนดสถานะขา D1 ใช้ในการรับค่า
23
24   Serial.begin(115200);    //กำหนดค่าความเร็วในการรับ-ส่งข้อมูล
25   Serial.println();
26
27
28   Serial.print("Connecting to ");
29   Serial.println(ssid);
30   WiFi.begin(ssid, password); //Setup WiFi
31   while (WiFi.status() != WL_CONNECTED) {
32     delay(500);
33     Serial.print(".");
34   }
35   Serial.println("");
36   Serial.println("WiFi connected");
37   Serial.println("IP address: ");
38   Serial.println(WiFi.localIP());
39
40   client.setServer(MQTTserver, MQTTport); //Setup MQTT Server
41   client.setCallback(callback);
42 }
43

```

รูปที่ 3.45 กำหนดค่าเริ่มต้นของโปรแกรม

4. จากรูปที่ รูปที่ 3.46 การทำงานวนซ้ำต่อเนื่องของอุปกรณ์ไฟฟ้า ถ้าโปรแกรมสามารถเชื่อมต่อ MQTT ได้สำเร็จให้แสดงสถานะและติดตามรับข้อมูลจาก Topic ที่ชื่อว่า “License Number” แต่ถ้าหากเชื่อมต่อไม่ได้ จะทำการหน่วงเวลา 3 วินาที แล้วลองใหม่

```

43 void loop() {
44
45     if (!client.connected()) {
46         if (client.connect("ESP8266Client", MQTTuser, MQTTpassword)) { //ถ้าเชื่อมต่อ MQTT ได้สำเร็จให้พิมพ์หัวข้อที่ชื่อ License Number
47             Serial.println("connected");
48             client.subscribe("License Number");
49         }
50     } else {           // แต่ถ้าไม่สำเร็จ ทำการหน่วงเวลา 3 วินาที แล้วลองใหม่
51         Serial.print("failed !");
52         Serial.print(client.state());
53         Serial.println(" Try again in 3 seconds");
54         delay(3000);
55         return;
56     }
57 }
58 client.loop();
59
60 }
61

```

รูปที่ 3.46 การทำงานวนซ้ำเพื่อเชื่อมต่อคลาวด์

5. จากรูปที่ 3.47 การทำงานในส่วนของการอ่านข้อมูลจาก Topic ที่ถูกส่งมา และกำหนดเงื่อนไขว่า ถ้าได้รับข้อมูลจาก Topic แล้วข้อมูลที่ได้รับมาจาก Cloud Server ตรงกับป้ายทะเบียนที่กำหนด ให้ทำการสั่งขา D0 และ D1 ให้ Relay ON เพื่อควบคุมอุปกรณ์ต่าง ๆ หลังจากนั้นจะส่งข้อมูลสถานะเปิดกลับไปที่ Topic แต่ถ้าได้รับข้อมูลจากผู้ใช้เพื่อปิดโปรแกรมจะสั่งให้ขา D0 และ D1 ให้ Relay OFF และส่งข้อมูลสถานะปิดกลับไปที่ Topic

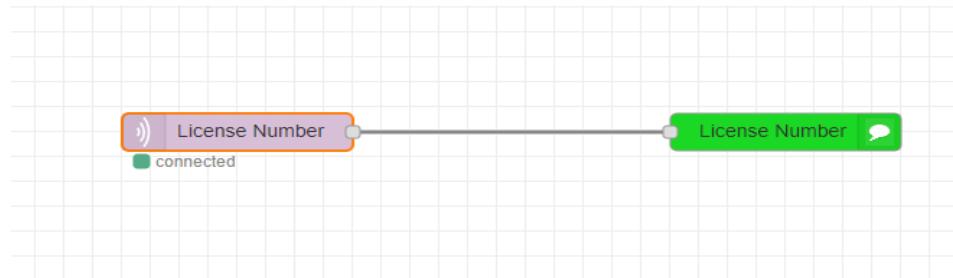
```

62 void callback(char* topic, byte* payload, unsigned int length) {
63     Serial.print("Message form topic <"); 
64     Serial.print(topic);
65     Serial.print("> : ");
66     String message = "";
67     int text=0;
68     while (text<length) {
69         message += (char)payload[text++]; //อ่านข้อมูลจาก Topic ที่ถูกส่งมา
70     }
71
72     if (message == "4n487") {           //ถ้าข้อมูลที่ได้รับจาก Topic ตรงกับป้ายทะเบียนที่กำหนด
73         digitalWrite(D0,1);           //สั่งขา D0 ให้ Relay ON
74         digitalWrite(D1,1);           //สั่งขา D1 ให้ Relay ON
75         Relay_ON = "Status Relay : ON ";
76         client.publish("License Number",Relay_ON); //ทำการส่งข้อมูลสถานะเปิด กลับไปที่ Topic
77     }
78
79     else if (message == "USER ON") {    //ถ้าข้อมูลที่ได้รับจากหัวขอเป็นผู้ใช้ปิด
80         digitalWrite(D0,1);           //สั่งขา D0 ให้ Relay ON
81         digitalWrite(D1,1);           //สั่งขา D1 ให้ Relay ON
82         Relay_ON = "Status Relay : ON ";
83         client.publish("License Number",Relay_ON); //ทำการส่งข้อมูลสถานะเปิด กลับไปที่ Topic
84     }
85
86     else if (message == "USER OFF"){   //แต่ถ้าได้รับข้อมูลจากผู้ใช้ให้ปิด
87         digitalWrite(D0,0);           //สั่งขา D0 ให้ Relay OFF
88         digitalWrite(D1,0);           //สั่งขา D1 ให้ Relay OFF
89         Relay_OFF = "Status Relay : OFF ";
90         client.publish("License Number",Relay_OFF); //ทำการส่งข้อมูลสถานะปิด กลับไปที่ Topic
91     }
92
93     Serial.println(message);          //แสดงข้อมูลที่ได้รับจาก Topic
94 }

```

รูปที่ 3.47 โค้ดการอ่านข้อมูลจาก Topic

## 6. การสร้าง Flow ใน Node-RED เพื่อทำ Line Notification และแสดงดังรูปที่ 3.48



รูปที่ 3.48 การสร้าง Flow เพื่อทำ Line Notification

อธิบายโค้ดเว็บไซต์สำหรับแสดงผลสถานะ และควบคุมอุปกรณ์ไฟฟ้า

1. การออกแบบหน้าหลักของเว็บไซต์โดยมีการกำหนดพื้นหลัง กำหนดขนาด สี การวางตำแหน่ง และลักษณะของกล่องข้อความและตัวอักษร ดังแสดงในรูปที่ 3.49 และ 3.50

```

1  <!DOCTYPE HTML>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>license plate detection</title>
6  <script src="jquery.js"></script>
7  <script src="mqtt.js"></script>
8  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway">
9  <style>
10
11 body {
12   font-family: "Raleway", sans-serif;
13   background-repeat: no-repeat;
14   background-image: url("BG.png");
15 }
16
17 #Status_relay{
18   max-width:260px;
19   min-width: 140px;
20   margin-left: auto;
21   margin-right: auto;
22   background: rgb(255, 255, 255, 0.8) ;
23   color: #000000;
24   font-size: 24px;
25   border-radius: 10px;
26   padding: 30px 30px;
27   text-align: center;
28   font-family: "Raleway", sans-serif;
29   border: 3px solid #000;
30   box-shadow: 1px 7px #ff9900 ;
31 }
32
33 #Status_relay_out{
34   background: #ffcc66 ;
35   color: #000000;
36   border-radius: 10px;
37   padding: 20px 85px;
38   text-align: center;
39   font-family: "Raleway", sans-serif;
40   box-shadow: 1px 7px #ff9900 ;
41 }
42
43
44 #Status_UI {
45   max-width:260px;
46   min-width: 140px;
47   margin-left: auto;
48   margin-right: auto;
49   background: #333;
50   color: #FFF;
51   border-radius: 15px;
52   font-weight: bold;
53   padding: 3px 6px;
54   line-height: 2;
55   box-shadow: 0px 3px 1px #fff ;
56
57   height:100%;
58   width: 30% ;
59 }
60
61 #Status_UI.error_Connect {
62   background: #F00;
63   color: #FFF;
64   box-shadow: 1px 7px #ff9900 ;
65 }

```

รูปที่ 3.49 การออกแบบหน้าหลักของเว็บไซต์

```

67 #Status_UI.connect {
68   background: #f6a00;
69   color: #FFF;
70 }
71
72 #Status_UI.Online {
73   background: #00AE04;
74   color: #FFF;
75 }
76
77
78 #Headd {
79   background: #ffa64d;
80   padding: 10px 140px;
81   border-radius: 10px;
82   text-align: center;
83   box-shadow: 1px 7px #ff9900 ;
84   height:100%;
85 }
86 #Headd3 {
87   background: #ffa64d;
88   padding: 50px 140px;
89   border-radius: 10px;
90   text-align: center;
91   box-shadow: 1px 7px #ff9900 ;
92   height:100%;
93 }
94
95
96 button {
97   margin: auto;
98   width: 100%;
99   font-size: 32px;
100  background: #ffd633;
101  text-align: center;
102  border-radius: 20px;
103  font-family: "Raleway", sans-serif;
104  border: 3px solid #ffb366;
105  padding: 20px;
106  box-shadow: 1px 7px #ff9900 ;
107 }
108
109 .BTN2:hover {
110  background-color: #FFF;
111 }
112 .img1 {
113  float: right;
114 }
115 </style>

```

รูปที่ 3.50 การออกแบบหน้าหลักของเว็บไซต์

2. การกำหนดค่าและรับการเก็บข้อมูลของการตั้งค่า MQTT และการสร้าง Client สำหรับการร้องขอบริการและการเข้าถึงข้อมูลระหว่างเครื่องผู้ให้บริการ (Server) กับเครื่องผู้ใช้บริการ (Client) หากระบบสามารถเชื่อมต่อกับ MQTT Server ได้ให้ทำการแสดงผล Online ผ่านหน้าหลักของเว็บไซต์และทำการติดตามข้อมูลจาก Topic ที่มีชื่อว่า License Number แต่ถ้าไม่สามารถเชื่อมต่อได้ให้ทำการแสดง Error ออกมานั้นหากเชื่อมต่อล้มเหลวให้ทำการหน่วงเวลา 1 วินาทีแล้วลองเชื่อมต่อใหม่อีกครั้ง ดังแสดงในรูปที่ 3.51

```

117 <script>
118   var config = {
119     Server: "YOUR_MQTT_SERVER",
120     Port: YOUR_MQTT_PORT ,
121     User: "YOUR_MQTT_USERNAME",
122     Password: "YOUR_MQTT_PASSWORD"
123   };
124
125 $(document).ready(function(e) {
126   client = new Paho.MQTT.Client(config.Server, config.Port, "web_" + parseInt(Math.random() * 100, 10));
127
128   client.connect({
129     useSSL: true,
130     userName: config.User,
131     password: config.Password,
132
133     onSuccess: function() {
134       $("#Status_UI").text("Online").removeClass().addClass("Online");
135       client.subscribe("License Number");
136       MQTT_Send("License Number", "USER OFF");
137     },
138     onFailure: function(e) {
139       $("#Status_UI").text("Connect : Error").removeClass().addClass("error_Connect");
140     }
141   });
142
143   client.onConnectionLost = function(response) {
144     if (response.errorCode !== 0) {
145       $("#Status_UI").text("onConnectionLost:" + response.errorMessage).removeClass().addClass("connect");
146       setTimeout(function() { client.connect() }, 1000);
147     }
148   }
}

```

รูปที่ 3.51 การตั้งค่า MQTT และการสร้าง Client สำหรับการเชื่อมต่อ MQTT Server

3. การรับข้อความ หากมีข้อความเข้ามาให้ทำการส่งข้อความที่ได้รับมาไปยัง Status\_relay เพื่อทำการแสดงผลผ่านเว็บไซต์ แต่ถ้าหากได้รับข้อความตามที่กำหนดไว้ เช่น ได้รับข้อความ “USER ON” เข้ามาให้ทำการปิดการใช้งานปุ่มกดของปุ่ม ON ให้ไม่สามารถทำงานได้ เช่นเดียวกัน หากได้รับข้อความ “USER OFF” ปุ่ม OFF จะถูกปิดการใช้งานให้ไม่สามารถทำงานได้ ดังแสดงในรูปที่ 3.52

```

149     client.onMessageArrived = function(message) {
150       if (message.payloadString != null ) {
151         $("#Status_relay").text(message.payloadString).addClass("Status_relay");
152       }
153     }
154   }
155
156   if (message.payloadString == "4n87" || message.payloadString == "USER ON"|| message.payloadString == "USER OFF") {
157     $("#Relay-on").attr("disabled", (message.payloadString == "USER ON"|| message.payloadString == "4n87" ? true : false));
158     $("#Relay-off").attr("disabled", (message.payloadString == "USER OFF" ? true : false));
159   }
160 }

```

รูปที่ 3.52 การรับข้อความจาก MQTT Server เข้ามาแสดงผลผ่านเว็บไซต์

4. คำสั่งของปุ่มกด ON และปุ่มกด OFF ถ้าหากปุ่ม ON ถูกกดให้ทำการส่งข้อความ “USER ON” ไปยัง Topic ที่มีชื่อว่า License Number หรือถ้าหากปุ่ม OFF ถูกกดให้ทำการส่งข้อความ “USER OFF” ไปยัง Topic ที่มีชื่อว่า License Number ดังแสดงในรูปที่ 3.53

```

162   $("#Relay-on").click(function(e) {
163     MOTT_Send("License Number", "USER ON");
164   });
165
166   $("#Relay-off").click(function(e) {
167     MOTT_Send("License Number", "USER OFF");
168   });

```

รูปที่ 3.53 โปรแกรมคำสั่งของปุ่มกด ON และปุ่มกด OFF

5. การสร้างฟังก์ชัน MQTT\_Send สำหรับนำไปใช้งานในส่งข้อความออกไปยัง MQTT Server ดังแสดงในรูปที่ 3.54

```

172   var MQTT_Send = function(MQTT_topic, MQTT_message) {
173
174     var message = new Paho.MQTT.Message(MQTT_message);
175     message.destinationName = MQTT_topic;
176
177     client.send(message);
178   }

```

รูปที่ 3.54 การสร้างฟังก์ชัน MQTT\_Send

6. โครงสร้างหลักของเว็บไซต์ ซึ่งจะมีการกำหนดข้อความของหัวเรื่องต่าง ๆ สร้างปุ่มกด กำหนด ID และ Class สำหรับการระบุตำแหน่งเพื่อใช้ในการกำหนดรูปแบบต่าง ๆ เช่น การออกแบบ คำสั่งการทำงาน ดังแสดงในรูปที่ 3.55

```

185 </img>
186 <h1 id="Headd" > KMUTT License Plate Detection </h1>
187 <h1 id="Headd2" ></h1>
188 <h3 id="Headd3"> <div style="font-size: 28px;" > Smart Home Control </div>
189 <p <span id="Status_UI" class="connect">Connect</span> </p></h3>
190
191 <button class="BTN" id="Relay-on" disabled>ON</button>
192 &nbsp;&nbsp;&nbsp;
193 <button class="BTN2" id="Relay-off" disabled>OFF</button>
194
195 <div class="text_status" >
196   <h4 id= "Status_relay_out" >
197     <p id="Status_relay"> </p>
198   </h4>
199 </div>

```

รูปที่ 3.55 โครงสร้างของเว็บไซต์

### 3.2 การฝึกฝนปัญญาประดิษฐ์โปรแกรมตรวจสอบป้ายทะเบียนรถยนต์

การฝึกฝนปัญญาประดิษฐ์จะทำโดยการใช้ Tensorflow object-detection API ซึ่งมีโครงสร้างการเรียนรู้ที่เรียกว่า Convolution Neural Networks เนื่องจากเป็นโครงสร้างที่เหมาะสมในการใช้แยกแยะวัตถุ โครงสร้าง Convolution Neural Networks ที่นำมาใช้ในโครงงานนี้คือ SSDLite Mobilenet ซึ่งเป็นโมเดลที่ได้ถูกฝึกฝนไว้แล้ว ด้วย COCO dataset ทางผู้จัดทำโครงงานนี้จะนำมาฝึกฝนใหม่ให้โมเดลรู้จักป้ายทะเบียนรถยนต์โดยใช้โครงสร้างเก่าของโมเดล สาเหตุที่เลือกใช้โมเดลนี้เนื่องจาก SSDLite Mobilenet เป็นโมเดลที่ทำงานประมวลผลได้เร็วเหมาะสมกับการใช้ในการทำงานแบบ Real time โดยโครงสร้างโมเดลต่าง ๆ ที่ถูกฝึกฝนโดย COCO dataset สามารถดูได้ดังรูปที่ 3.56 คอลัมน์ที่ 1 แสดงชื่อโมเดล, คอลัมน์ที่ 2 แสดงระยะเวลาประมวลผล, คอลัมน์ที่ 3 แสดงค่าความแม่นยำ และคอลัมน์ที่ 4 แสดงอัตราพุ่งของโมเดล

### COCO-trained models

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes
faster_rcnn_resnet50_lowproposals_coco	64		Boxes
rfcn_resnet101_coco	92	30	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes
faster_rcnn_resnet101_lowproposals_coco	82		Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	620	37	Boxes
faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco	241		Boxes
faster_rcnn_nas	1833	43	Boxes
faster_rcnn_nas_lowproposals_coco	540		Boxes
mask_rcnn_inception_resnet_v2_atrous_coco	771	36	Masks
mask_rcnn_inception_v2_coco	79	25	Masks
mask_rcnn_resnet101_atrous_coco	470	33	Masks
mask_rcnn_resnet50_atrous_coco	343	29	Masks

รูปที่ 3.56 โมเดลต่าง ๆ ที่ถูกฝึกฝนโดย COCO Dataset

#### 3.2.1 ขั้นตอนการฝึกฝนปัญญาประดิษฐ์

1. เครื่องคอมพิวเตอร์ที่ใช้ในการฝึกฝนจำเป็นต้องมีโปรแกรม Python โดยแนะนำให้ติดตั้งจาก Anaconda Distribution เนื่องจากจะติดตั้งพร้อม Library ที่สำคัญไว้หมดแล้ว

2. ติดตั้ง Library เพิ่มเติมด้วยคำสั่ง pip install

2.1 OpenCV

2.2 PIL

2.3 Jupyter Notebook

2.4 Keras

2.5 imutils

2.6 Protobuf

3. ติดตั้ง Tensorflow-gpu เช้าไปที่ Environments ที่สร้างขึ้นมา

#### 4. ติดตั้ง model object detection โดยติดตั้งจาก <https://github.com/tensorflow/models>

Models and examples built with TensorFlow

Branch: master ▾ New pull request Find file Clone or download ▾

**HelgeS and robieta** Parameter definition is in resnet\_run\_loop.py (#5770) Latest commit a649475 2 days ago

official	Parameter definition is in resnet_run_loop.py (#5770)	2 days ago
research	Update lite path + fix --config option	9 days ago
samples	Fix list format and typos.	3 months ago
tutorials	update the calculation of num_batches_per_epoch	18 days ago
.gitignore	Fixed gitignore for mac's ds_store (#4012)	3 months ago
.gitmodules	Move the research models into a research subfolder (#2430)	a year ago
AUTHORS	Spatial Transformer model	3 years ago
CODEOWNERS	Added cvt_text model	2 months ago
CONTRIBUTING.md	Fixing small typo	a year ago
ISSUE_TEMPLATE.md	Update ISSUE_TEMPLATE.md	10 months ago
LICENSE	Update LICENSE	3 years ago
README.md	Add Contribution and License in README (#4022)	7 months ago
WORKSPACE	Consolidate privacy/ and differential_privacy/.	2 years ago
README.md		

รูปที่ 3.57 หน้าเว็บ <https://github.com/tensorflow/models>

คลิก clone or download ทำการโหลด หรือใช้คำสั่ง git clone ในAnaconda Prompt ดังรูปที่ 3.58

```
(base) C:\Users\fluke>git clone https://github.com/tensorflow/models.git
```

รูปที่ 3.58 การใช้คำสั่ง git clone

5. Activate Environments ที่ติดตั้ง Library ตามข้อที่ 1, 2 และ 3 ไว้ โดยใช้คำสั่ง activate ชื่อ Environment (ผู้จัดทำติดตั้งไว้ใน Environment ที่ชื่อ tf-gpu) และชี้ตำแหน่งไปยัง model object detection ที่ติดตั้งจากข้อที่ 4 ดังรูปที่ 3.59

```
(base) C:\Users\fluke>activate tf-gpu
(tf-gpu) C:\Users\fluke>cd C:\tensorflow1\models\research\object_detection
```

รูปที่ 3.59 การ Activate Environments และการชี้ตำแหน่งไปที่ model object detection

#### 6. ตั้ง PYTHONPATH environment variable ดังรูปที่ 3.60

```
(base) C:\Users\fluke>activate tf-gpu
(tf-gpu) C:\Users\fluke>cd C:\tensorflow1\models\research\object_detection
(tf-gpu) C:\tensorflow1\models\research\object_detection>set PYTHONPATH=C:\tensorflow1\models;C:\tensorflow1\models\research;C:\tensorflow1\models\research\slim
```

รูปที่ 3.60 การตั้ง PYTHONPATH environment variable

7. ใส่โค้ดด้านล่างเพื่อ compile protobufs นี้องจาก Tensorflow ใช้ Protobufs เพื่อตั้งค่า model และ training parameter

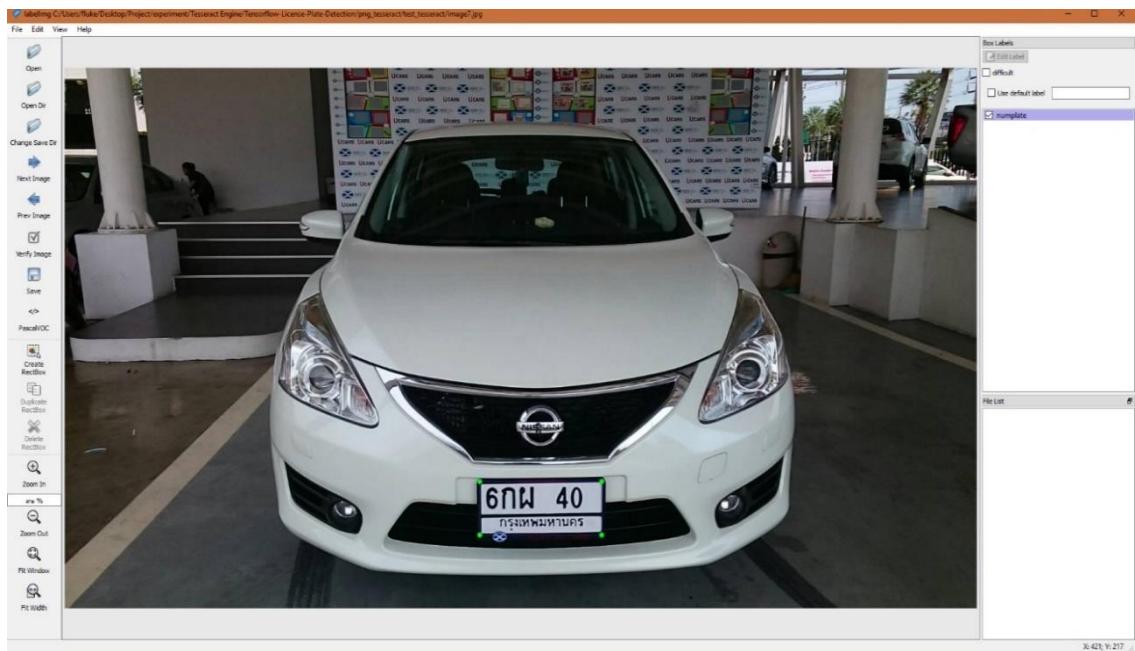
```
protoc --python_out=. \object_detection\protos\anchor_generator.proto
.\object_detection\protos\argmax_matcher.proto .\object_detection\protos\bipartite_matcher.proto
.\object_detection\protos\box_coder.proto .\object_detection\protos\box_predictor.proto
.\object_detection\protos\eval.proto .\object_detection\protos\faster_rcnn.proto
.\object_detection\protos\faster_rcnn_box_coder.proto
.\object_detection\protos\grid_anchor_generator.proto .\object_detection\protos\hyperparams.proto
.\object_detection\protos\image_resizer.proto .\object_detection\protos\input_reader.proto
.\object_detection\protos\losses.proto .\object_detection\protos\matcher.proto
.\object_detection\protos\mean_stddev_box_coder.proto .\object_detection\protos\model.proto
.\object_detection\protos\optimizer.proto .\object_detection\protos\pipeline.proto
.\object_detection\protos\post_processing.proto .\object_detection\protos\preprocessor.proto
.\object_detection\protos\region_similarity_calculator.proto
```

.\object\_detection\protos\square\_box\_coder.proto .\object\_detection\protos\ssd.proto  
 .\object\_detection\protos\ssd\_anchor\_generator.proto  
 .\object\_detection\protos\string\_int\_label\_map.proto .\object\_detection\protos\train.proto  
 .\object\_detection\protos\keypoint\_box\_coder.proto  
 .\object\_detection\protos\multiscale\_anchor\_generator.proto  
 .\object\_detection\protos\graph\_rewriter.proto

8. ใส่โค้ดคำสั่ง python setup.py build และ python setup.py install เพื่อติดตั้ง Model ให้เสร็จสมบูรณ์

9. รวมรวมรูปภาพที่ต้องการนำมาใช้เพื่อฝึกฝนปัญญาประดิษฐ์

10. ทำ Annotation ตีกรอบวัตถุที่ต้องการให้ปัญญาประดิษฐ์เรียนรู้และตั้งชื่อวัตถุนั้น จากนั้น Verify เมื่อเสร็จเรียบร้อยจะได้ไฟล์ XML โดยผู้จัดทำใช้โปรแกรม LabelImg ในการทำ Annotation ดังรูปที่ 3.61



รูปที่ 3.61 การทำ Annotation

11. สร้างไฟล์ CSV เพื่อใช้ทำ TF Records โดยใส่โค้ด python xml\_to\_csv.py ใน Anaconda prompt

12. แก้ไขไฟล์ generate\_tfrecord.py ให้ตรงตามชื่อ Class ที่ได้ทำ Annotation ไว้ ดังรูปที่ 3.62

```

31  def class_text_to_int(row_label):
32      if row_label == 'numplate':
33          return 1
34      else:
35          None

```

รูปที่ 3.62 การแก้ไขไฟล์ generate\_tfrecord.py

13. สร้าง train.record และ test.record เพื่อใช้ในการฝึกฝนปัญญาประดิษฐ์โดยพิมพ์โค้ดต่อไปนี้ใน Anaconda prompt

```

python generate_tfrecord.py --csv_input=images\train_labels.csv --image_dir=images\train --
output_path=train.record
python generate_tfrecord.py --csv_input=images\test_labels.csv --image_dir=images\test --
output_path=test.record

```

14. สร้างไฟล์ object-detection.pbtxt เพื่อใส่ Class ที่แบ่งไว้ ดังรูปที่ 3.63 แล้วบันทึกไว้ในที่ที่ต้องการ

```

1 item {
2     id: 1
3     name: 'numplate'
4 }
5
6

```

รูปที่ 3.63 การสร้างไฟล์ object-detection.pbtxt

15. นำไฟล์ ssdlite\_mobilenet\_v2\_coco.config ซึ่งเป็นโมเดลที่ต้องการนำมาใช้ (มาจาก คำแนะนำที่ git-clone จากข้อที่ 4) ในที่นี่คือ C:\tensorflow1\models\research\object\_detection\samples\configs มาไว้ที่คำแนะนำที่บันทึกไฟล์ object-detection.pbtxt

16. แก้ไขไฟล์ ssdlite\_mobilenet\_v2\_coco.config ดังนี้

- 16.1 แก้ไขบรรทัดที่ 9 ให้มี num\_classes เท่ากับจำนวน Class ที่กำหนดไว้ใน object-detection.pbtxt

16.2 แก้ไขบรรทัดที่ 106 ใส่ทีอู่ model.ckpt ของ ssdlite\_mobilenet\_v2\_coco จะ  
ตำแหน่งที่ได้ git clone ไว้

16.3 แก้ไขบรรทัดที่ 123 ใส่ที่อยู่ของไฟล์ train.record ที่สร้างมาจากข้อที่ 13

16.4 แก้ไขบรรทัดที่ 125 ใส่ที่อยู่ของไฟล์ object-detection.pbtxt ที่สร้างมาจากข้อที่ 14

16.5 แก้ไขบรรทัดที่ 130 ใส่จำนวนรูปที่ต้องการใช้ในการทดสอบ

16.6 แก้ไขบรรทัดที่ 135 ใส่ที่อยู่ของไฟล์ test.record ที่สร้างมาจากข้อที่ 13

16.7 แก้ไขบรรทัดที่ 137 ใส่ที่อยู่ของไฟล์ object-detection.pbtxt ที่สร้างมาจากข้อที่ 14

17. เริ่มฝึกฝนปัญญาประดิษฐ์โดยพิมพ์โค้ดด้านล่างใน Anaconda prompt

```
python train.py --logtostderr --train_dir=training/
```

```
pipeline config path=training/ ssdlite mobilenet v2 coco.config
```

### 3.3 การฝึกฝนปัญญาประดิษฐ์โปรแกรมอ่านป้ายทะเบียนรถยนต์

การอ่านป้ายทะเบียนรถยนต์ ผู้จัดทำใช้วิธีการหนึ่งของ Machine Learning ที่เรียกว่า k Nearest Neighbor (kNN) ซึ่งเป็นหนึ่งใน classification algorithm ใช้ kNN ในการฝึกฝนให้โปรแกรมเรียนรู้ว่าตัวอักษรแต่ละตัวอักษรมีรูปร่างลักษณะเป็นอย่างไร โดยใช้วิธีการสร้างไฟล์รูปภาพตัวอักษร และใช้การป้อนอินพุตเข้าไปโดยการกดแป้น เช่น รูปตัวอักษร ก กดแป้น ก เพื่อให้โปรแกรมเรียนรู้ว่า รูปตัวอักษร ก คือ ก

### 3.3.1 ขั้นตอนการฝึกฝน kNN

1. สร้างไฟล์รูปภาพที่ประกอบด้วยตัวอักษรภาษาไทยและตัวเลข โดยใช้ตัวอักษรสีดำเพื่อหลังสีขาว fon เดียวกันแล้วเลือกใช้fon ที่มีลักษณะใกล้เคียงกับตัวอักษรป้ายทะเบียนรถยนต์ ผู้จัดทำ จึงเลือกใช้fon Sarun's ThangLuang ดังรูปที่ 3.64

### រូបថត 3.64 រូបភាពសំខាន់បើកដំឡើង

2. เขียนโค้ดนำเข้าไลบรารีที่ต้องใช้ ดังรูปที่ 3.65

```

3   import sys
4   import numpy as np
5   import cv2
6   import os

```

รูปที่ 3.65 การนำเข้าไลบรารีต่าง ๆ

3. กำหนดตัวแปรที่จำเป็นต้องใช้ ได้แก่ ขนาด contour ที่เล็กที่สุด, ขนาดพิกเซลกว้างและยาว ที่ต้องการดังรูปที่ 3.66

```

9   MIN_CONTOUR_AREA = 100
10
11  RESIZED_IMAGE_WIDTH = 20
12  RESIZED_IMAGE_HEIGHT = 30

```

รูปที่ 3.66 การกำหนดตัวแปรที่ต้องใช้

4. สร้างฟังก์ชัน main และใส่โค้ดตามรูปที่ 3.67 ซึ่งคือการอ่านรูปคัวอักษรที่ใช้สำหรับการ ฝึกฝนที่ได้มามาจากขั้นตอนที่ 1 จากนั้นนำรูปภาพจากขั้นตอนที่ 1 มาทำการ preprocess โดยเปลี่ยนภาพ เป็น grayscale, เบลอรูปภาพ และทำ Threshold

```

15 def main():
16     imgTrainingNumbers = cv2.imread("training_chars.png")
17
18     if imgTrainingNumbers is None:
19         print ("error: image not read from file \n\n")
20         os.system("pause")
21         return
22     # end if
23
24     imgGray = cv2.cvtColor(imgTrainingNumbers, cv2.COLOR_BGR2GRAY)
25     imgBlurred = cv2.GaussianBlur(imgGray, (5,5), 0)
26
27
28     imgThresh = cv2.adaptiveThreshold(imgBlurred,
29                                         255,
30                                         cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
31                                         cv2.THRESH_BINARY_INV,
32                                         11,
33                                         2)
34
35     cv2.imshow("imgThresh", imgThresh)
36
37     imgThreshCopy = imgThresh.copy()

```

รูปที่ 3.67 โค้ดในฟังก์ชัน main

5. เจียนโค้ดดังรูปที่ 3.68 โดยในบรรทัดที่ 39 ใช้ฟังก์ชัน findContours กับรูปที่ทำการ preprocess เวิบร้อยแล้ว ในบรรทัดที่ 44 สร้าง numpy array ชื่อ npaFlattenedImages ไว้เพื่อนำไปใช้ภาษาหลัง ในบรรทัดที่ 46 สร้าง List ชื่อ intClassifications เพื่อนำไปใช้ภาษาหลัง และในบรรทัดที่ 49 สร้าง List intValidChars ที่ประกอบไปด้วยตัวเลข ASCII code ของตัวอักษรที่ต้องการให้เรียนรู้ เช่น ASCII code 48 คือเลข 0

```

39     imgContours, npaContours, npaHierarchy = cv2.findContours(imgThreshCopy,
40                                         cv2.RETR_EXTERNAL,
41                                         cv2.CHAIN_APPROX_SIMPLE)
42
43
44     npaFlattenedImages = np.empty((0, RESIZED_IMAGE_WIDTH * RESIZED_IMAGE_HEIGHT))
45
46     intClassifications = []
47
48
49     intValidChars = [48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 161, 162, 163, 164, 165, 166, 167,
```

รูปที่ 3.68 โค้ดในฟังก์ชัน main

6. เจียนโค้ดดังรูปที่ 3.69 คือการสร้าง for loop สำหรับ contour ที่ได้จากขั้นตอนที่ 5 ถ้า contour มีขนาดใหญ่กว่า MIN\_CONTOUR\_AREA ที่ได้กำหนดไว้จากขั้นตอนที่ 3 ให้สร้างกรอบสี่เหลี่ยมรอบ contour และทำการ crop รูปภาพของ contour นั้น จากนั้นทำการ Resize รูปภาพตามขนาดที่ได้กำหนดไว้จากขั้นตอนที่ 3 เพื่อรับอินพุตจากแป้นพิมพ์ต่อไป

```

51     for npaContour in npaContours:
52         if cv2.contourArea(npaContour) > MIN_CONTOUR_AREA:
53             [intX, intY, intW, intH] = cv2.boundingRect(npaContour)
54
55
56             cv2.rectangle(imgTrainingNumbers,
57                         (intX, intY),
58                         (intX+intW, intY+intH),
59                         (0, 0, 255),
60                         2)
61
62             imgROI = imgThresh[intY:intY+intH, intX:intX+intW]
63             imgROIResized = cv2.resize(imgROI, (RESIZED_IMAGE_WIDTH, RESIZED_IMAGE_HEIGHT))
64
65             cv2.imshow("imgROI", imgROI)
66             cv2.imshow("imgROIResized", imgROIResized)
67             cv2.imshow("training_numbers.png", imgTrainingNumbers)
```

รูปที่ 3.69 โค้ด for loop ในฟังก์ชัน main

7. เจียนโค้ดดังรูปที่ 3.70 คือการรออินพุตจากแป้นพิมพ์ถ้าเป็นที่กดอยู่ใน list intvalidchar ที่กำหนดไว้ในขั้นตอนที่ 5 ให้เพิ่มเลข intchar ลงไว้ใน list intClassifications ที่สร้างไว้ในขั้นตอนที่ 5

แต่ในกรณีที่เป็นที่กดเป็นตัวอักษรภาษาไทย เช่น ก มีเลข intchar (เลข ASCII) ต้องแปลงเลขเป็น Unicode ก่อนถึงจะนำໄປเพิ่มใน list intClassifications ได้ในกรณีนี้ ก มี Unicode คือ 3585

```

69         intChar = cv2.waitKey(0)

70
71     if intChar == 27:
72         sys.exit()
73     elif intChar in intValidChars:
74         if intChar == 161:
75             intClassifications.append(3585)
76         elif intChar == 162:
77             intClassifications.append(3586)
78         elif intChar == 163:
79             intClassifications.append(3587)
80         elif intChar == 164:
81             intClassifications.append(3588)

```

รูปที่ 3.70 โค้ด for loop ในฟังก์ชัน main

8. เขียนโค้ดดังรูปที่ 3.71 คือการแปลงภาพตัวอักษรแต่ละตัวให้กลายเป็น 1d numpy array และเพิ่มเข้าไปใน numpy array npaFlattenedImages ที่สร้างไว้ในขั้นตอนที่ 5 จบ for loop

```

209         npaFlattenedImage = imgROIResized.reshape((1, RESIZED_IMAGE_WIDTH * RESIZED_IMAGE_HEIGHT))
210         npaFlattenedImages = np.append(npaFlattenedImages, npaFlattenedImage, 0)
211     # end if
212     # end if
213 # end for

```

รูปที่ 3.71 โค้ด for loop ในฟังก์ชัน main

9. เขียนโค้ดดังรูปที่ 3.72 คือการแปลง list intClassification ให้เป็น numpy array float และ reshape ให้เป็น 1d numpy array

```

215     fltClassifications = np.array(intClassifications, np.float32)
216
217     npaClassifications = fltClassifications.reshape((fltClassifications.size, 1))
218
219     print ("\n\ntraining complete !!\n")

```

รูปที่ 3.72 โค้ดในฟังก์ชัน main

10. บันทึก npaClassifications และ npaFlattenedImages ที่ได้จากการฝึกฝนໄປเป็นไฟล์ txt เพื่อนำไว้สร้างโมเดล kNN ต่อไปดังรูปที่ 3.73

```
221     np.savetxt("classifications.txt", npaClassifications)
222     np.savetxt("flattened_images.txt", npaFlattenedImages)
223
224     cv2.destroyAllWindows()
225
226     return
```

ຮູບທີ 3.73 ໂກສດໃນພັກໜ້ານ main

## บทที่ 4

### การทดลองและผลการทดลอง

การทดลองระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์จะประกอบด้วย 3 การทดลอง ได้แก่ การทดลองตรวจจับป้ายทะเบียนรถยนต์ การทดลองอ่านป้ายทะเบียนรถยนต์ และการทดลองความคุณอุปกรณ์ไฟฟ้าและการแสดงผล

#### **4.1 การทดลองตรวจจับป้ายทะเบียนรถยนต์**

โปรแกรมตรวจจับป้ายทะเบียนรถยนต์ใช้ Tensorflow Object-Detection API ผู้ขัดทำนำโมเดลที่ฝึกฝนแล้วมาทดสอบประสิทธิภาพในการตรวจจับป้ายทะเบียน โมเดลที่นำมาทดลองมีทั้งสิ้น 4 โมเดลเกิดจากการฝึกฝนปัญญาประดิษฐ์ SSDlite Mobilenet ที่จำนวน Step ต่างกัน ได้แก่ 10000 steps, 20000 steps, 30000 steps และ 40000 steps

##### ประสิทธิภาพของโมเดลพิจารณาจาก

1. Accuracy Score คือคะแนนความแม่นยำในการตรวจพบป้ายทะเบียน หากโปรแกรมตรวจจับป้ายทะเบียนรถยนต์สามารถตรวจพบป้ายทะเบียนรถยนต์ได้ จะให้คะแนนเป็น 100 คะแนน แต่หากไม่สามารถตรวจพบได้จะให้คะแนน 0 คะแนน

2. Speed คือระยะเวลาที่โปรแกรมใช้ในการตรวจจับป้ายทะเบียนรถยนต์จากรูปภาพทดสอบ 15 รูป

##### 4.1.1 จุดประสงค์ของการทดลอง

จุดประสงค์ของการทดลอง เพื่อเปรียบเทียบประสิทธิภาพของโมเดลที่ใช้จำนวน Step การฝึกฝนที่ต่างกัน เมื่อได้โมเดลที่มีประสิทธิภาพที่ดีที่สุดแล้ว จะเลือกใช้โมเดลที่ได้นี้นำไปตรวจจับป้ายทะเบียนรถยนต์ และส่งรูปภาพป้ายทะเบียนรถยนต์ไปสู่การทดลองอ่านป้ายทะเบียนรถยนต์ในหัวข้อที่ 4.2

##### 4.1.2 ขั้นตอนการทดลอง

1. กำหนด minimum score เป็น 0.95 เนื่องจาก Tensorflow Object-Detection API มีการกำหนด minimum score ของการแสดงผลกรอบ Detection ผู้ขัดทำจึงกำหนด minimum score เป็น 0.95 หรือ 95 เปอร์เซ็นต์ ซึ่งหมายความว่าปัญญาประดิษฐ์ต้องมีความมั่นใจเกิน 95 เปอร์เซ็นต์ ถึงจะแสดงกรอบ Detection โดยกำหนดที่ไฟล์ visualization\_utils.py ในตำแหน่งที่ติดตั้ง Tensorflow ดังรูปที่ 4.1 ในบรรทัดที่ 696 เพื่อทำให้แน่ใจว่าสิ่งที่จะตรวจจับเป็นป้ายทะเบียนรถยนต์เท่านั้น

2. กำหนดอินพุตรูปภาพเข้าโปรแกรมในไฟล์ numplate\_recognition\_detection\_Images.py โดยใช้รูปภาพรถยนต์ที่มีป้ายทะเบียนรถยนต์อยู่ในภาพ 15 รูปโดยใน 15 รูป ประกอบด้วยรูปภาพ

รถยนต์ที่มีระยะห่าง และมุมที่แตกต่างกัน 3 กลุ่ม ได้แก่ รูปภาพด้านหน้าตรง 5 รูป, รูปภาพเอียง 5 รูป, รูปภาพด้านหลัง 5 รูป

3. รันโปรแกรม numplate\_recognition\_detection\_Images.py โดยไม่เรียกใช้งานฟังก์ชั่น mainของไฟล์ test\_Readplate.py จากบรรทัดที่ 79 ดังรูปที่ 4.2

4. บันทึกผลการทดลอง และดำเนินการทดลองจนครบ 4 โมเดล

```

688     box_to_display_str_map = collections.defaultdict(list)
689     box_to_color_map = collections.defaultdict(str)
690     box_to_instance_masks_map = {}
691     box_to_instance_boundaries_map = {}
692     box_to_keypoints_map = collections.defaultdict(list)
693     if not max_boxes_to_draw:
694         max_boxes_to_draw = boxes.shape[0]
695     for i in range(min(max_boxes_to_draw, boxes.shape[0])):
696         if scores is None or scores[i] > 0.95:
697             box = tuple(boxes[i].tolist())
698             if instance_masks is not None:
699                 box_to_instance_masks_map[box] = instance_masks[i]
700             if instance_boundaries is not None:

```

รูปที่ 4.1 การกำหนด minimum score

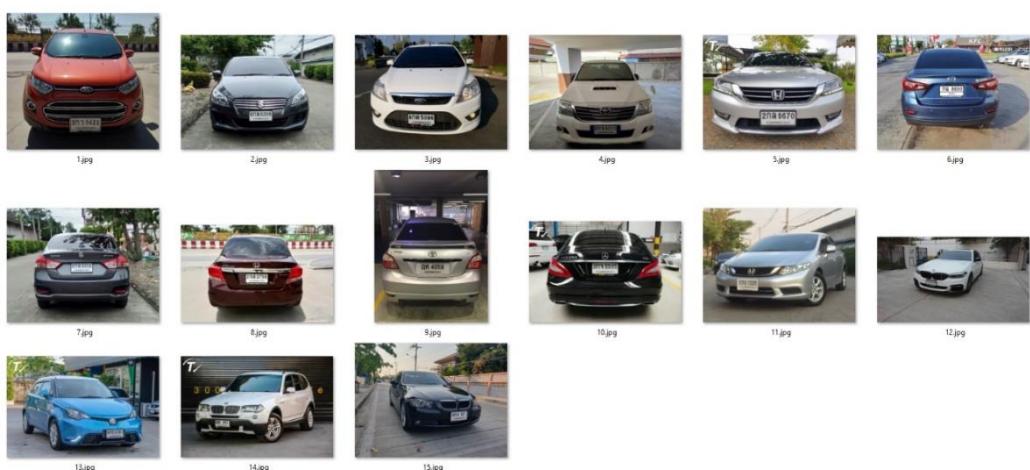
```

75     cropped_image = tf.image.crop_to_bounding_box(image_np, int(yminn),
76                                                 int(xminn),int(ymaxx - ymenn), int(xmaxx - xminn))
77     img_data2 = sess.run(cropped_image)
78     filename2 = save_image(img_data2,count)
79     #test_Readplate.main(count)
80     count +=1

```

รูปที่ 4.2 รันไฟล์ numplate\_recognition\_detection\_Images.py โดยไม่เรียกใช้การอ่านป้ายทะเบียน

#### 4.1.3 รูปภาพที่ใช้ในการทดสอบ



รูปที่ 4.3 รูปภาพที่ใช้ในการทดลองตรวจจับป้ายทะเบียนรถยนต์

#### 4.1.4 ผลการทดลอง

ตารางที่ 4.1 ผลการทดสอบตรวจจับป้ายทะเบียนรถยนต์

## 4.2 การทดลองอ่านป้ายทะเบียนรถยนต์

การอ่านป้ายทะเบียนรถยนต์จำเป็นต้องได้รับอินพุตเป็นรูปภาพเพื่อนำมาเข้ากระบวนการอ่านป้ายทะเบียนโดยใช้หลักการ k-Nearest Neighbors (kNN) ใน การอ่านตัวอักษร ฉะนั้นผู้จัดทำจึงพิจารณาจากผลการทดลองตารางที่ 4.1 และเลือกโมเดลที่มีประสิทธิภาพการตรวจจับป้ายทะเบียนรถยนต์ที่ดี ซึ่งมีทั้งสิ้น 3 โมเดล ได้แก่ โมเดลที่ 20000, 30000 และ 40000 steps โดยกำหนดค่า k เป็น 3 ค่า คือ 1, 2 และ 3 เนื่องจากจำนวนรูปแบบที่ใช้ในการฝึกฝนการอ่านป้ายทะเบียนใช้ตัวอักษร และตัวเลขที่มีขนาดแตกต่างกัน 3 ชุด จึงทดสอบกำหนดค่า k เป็น 1, 2 และ 3

### 4.2.1 จุดประสงค์ของการทดลอง

จุดประสงค์ของการทดลอง เพื่อหา โมเดลปัญญาประดิษฐ์ตรวจจับป้ายทะเบียนรถยนต์ที่มีประสิทธิภาพดีที่สุดเมื่อนำมาใช้ร่วมกับโปรแกรมอ่านป้ายทะเบียนรถยนต์ รวมถึงหาค่า k ที่ทำให้โมเดลมีประสิทธิภาพที่สุด โดยประสิทธิภาพพิจารณาจาก

1. Accuracy Score หากโปรแกรมอ่านป้ายทะเบียนรถยนต์สามารถแสดงผลหมายเหตุทะเบียนรถยนต์ได้ถูกต้องทุกตัวจะให้คะแนน 100 คะแนน แต่หากอ่านป้ายทะเบียนผิดพลาด ไม่ว่าจะขาด เกิน หรือผิดตัวอักษรจะให้คะแนน 0 คะแนน

2. Speed (วินาที) คือเวลาที่ใช้ในการประมวลผลของโปรแกรมทั้งหมด กล่าวคือระยะเวลาที่ใช้ตั้งแต่การตรวจจับป้ายทะเบียนรถยนต์ ไปจนถึงการอ่านป้ายทะเบียนรถยนต์ของรูปภาพทั้ง 15 รูป

### 4.2.2 ขั้นตอนการทดลอง

1. กำหนด minimum score เป็น 0.95 เมื่อการทดลองที่ 4.1
2. กำหนดค่า k เริ่มต้นที่ 1 ในไฟล์ test\_char.py บรรทัดที่ 408 ดังรูปที่ 4.4
3. กำหนดอินพุตรูปภาพเข้าโปรแกรมในไฟล์ numplate\_recognition\_detection\_Images.py จำนวน 15 รูปภาพ
4. เรียกใช้ไฟล์ numplate\_recognition\_detection\_Images.py โดยเรียกใช้ฟังก์ชัน main ของไฟล์ test\_Readplate.py
5. บันทึกผลการทดลอง และทดลองกับโมเดลถัดไป

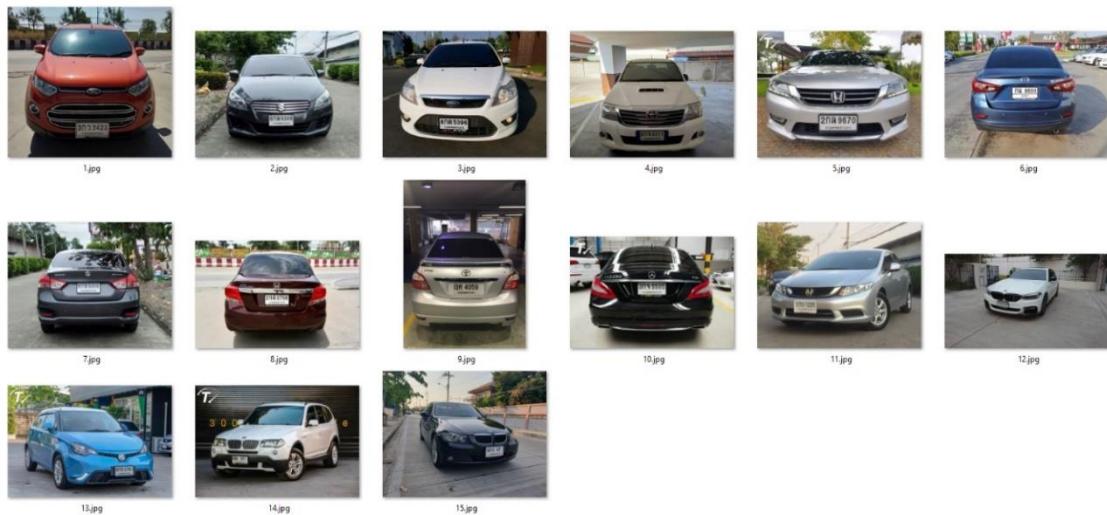
```

406     npaROIResized = np.float32(npaROIResized)
407
408     retval, npaResults, neigh_resp, dists = kNearest.findNearest(npaROIResized, k =1)
409
410     strCurrentChar = str(chr(int(npaResults[0][0])))

```

รูปที่ 4.4 การกำหนดค่า k

#### 4.2.3 รูปภาพที่ใช้ในการทดสอบ



รูปที่ 4.5 รูปภาพที่ใช้ในการทดลองอ่านป้ายทะเบียนรถยนต์

#### 4.2.4 ผลการทดลอง

ตารางที่ 4.2 ผลการทดลองการอ่านป้ายทะเบียนรถยนต์โดย  $k = 1, 2$  และ  $3$

ไม่เดล (Steps)	Accuracy score รูปด้านหน้า (%)	Accuracy score รูปด้านหลัง (%)	Accuracy score รูปเอียง (%)	Average Accuracy Score (%)	Speed (sec)
$k = 1$					
20000	80	100	20	66.66	14.2
30000	80	80	40	66.66	13.1
40000	40	80	20	46.66	13.7
$k = 2$					
20000	60	100	40	66.66	14.3
30000	100	80	80	86.66	13.5
40000	80	80	40	66.66	13.9
$k = 3$					
20000	100	100	40	80	14.1
30000	100	80	60	80	13.5
40000	80	80	60	73.33	13.9

### **4.3 การทดลองควบคุมอุปกรณ์ไฟฟ้าและการแสดงผล**

ในจุดประสงค์ของโครงการนี้ระบบตรวจสอบจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์จะมีความสามารถในการควบคุมอุปกรณ์ไฟฟ้าด้วยระบบ Internet of Things (IoT) ได้ในการทดลองนี้ผู้จัดทำได้นำระบบตรวจสอบจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์มาใช้ในการควบคุมอุปกรณ์ไฟฟ้าโดยเปรียบเทียบการณ์สมมติ เจ้าของบ้านขับรถเข้ามา เมื่อระบบตรวจสอบจับป้ายทะเบียนโดยปัญญาประดิษฐ์สามารถตรวจพบป้ายทะเบียนที่ถูกต้องได้จะส่งการให้ควบคุมอุปกรณ์ไฟฟ้าได้ โดยผู้จัดทำได้ยกตัวอย่าง การทดสอบเปิด-ปิดรีเลย์ มาทดลองควบคุมอุปกรณ์ไฟฟ้าเพื่อความสะดวกในการทดลอง โดยกำหนดเงื่อนไขหมายเลขทะเบียนที่ถูกต้องคือ 4กง87 หากระบบตรวจสอบจับป้ายทะเบียนรถยนต์สามารถตรวจจับรถยนต์ที่มีหมายเลขทะเบียน 4กง87 ได้ ระบบจะส่งการให้รีเลย์ทำงานโดยอัตโนมัติ อิกหั้งในกรณีเหตุการณ์สมมติที่เจ้าของบ้านไม่อยู่บ้าน หากมีรถยนต์คันอื่นที่เข้ามาบริเวณบ้าน ระบบจะส่งการแจ้งเตือนข้อมูลหมายเลขทะเบียนผ่านแอพพลิเคชั่นไลน์ ซึ่งถ้าหากเป็นรถยนต์ที่เจ้าของบ้านรู้จักก็สามารถควบคุมรีเลย์ เพื่อควบคุมอุปกรณ์ไฟฟ้าผ่านทางเว็บแอพพลิเคชั่นได้อีกด้วย

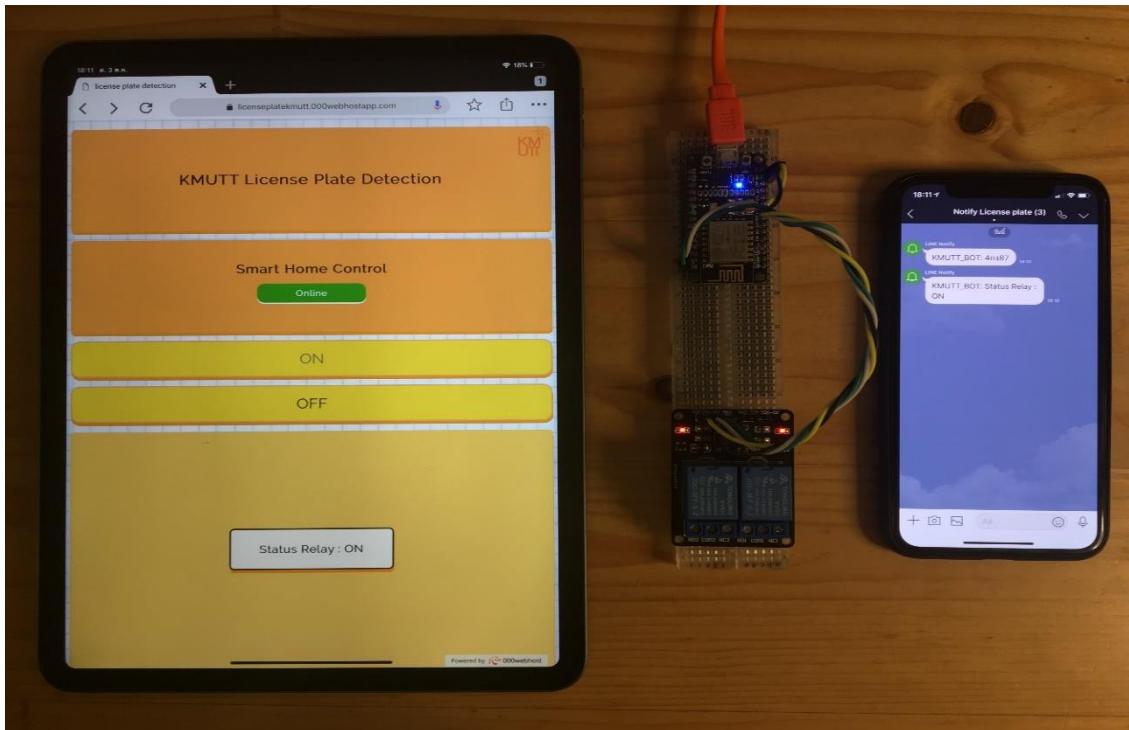
#### **4.3.1 จุดประสงค์ของการทดลอง**

จุดประสงค์ของการทดลองนี้เพื่อทดสอบระบบตรวจสอบจับป้ายทะเบียนรถยนต์ในการควบคุมอุปกรณ์ไฟฟ้าด้วยระบบ IoT, ทดสอบการสั่งการอุปกรณ์ไฟฟ้าและการแสดงผลข้อมูลผ่านเว็บแอพพลิเคชั่น และทดสอบการแสดงผลข้อมูลโดยใช้ระบบแจ้งเตือนทางไลน์

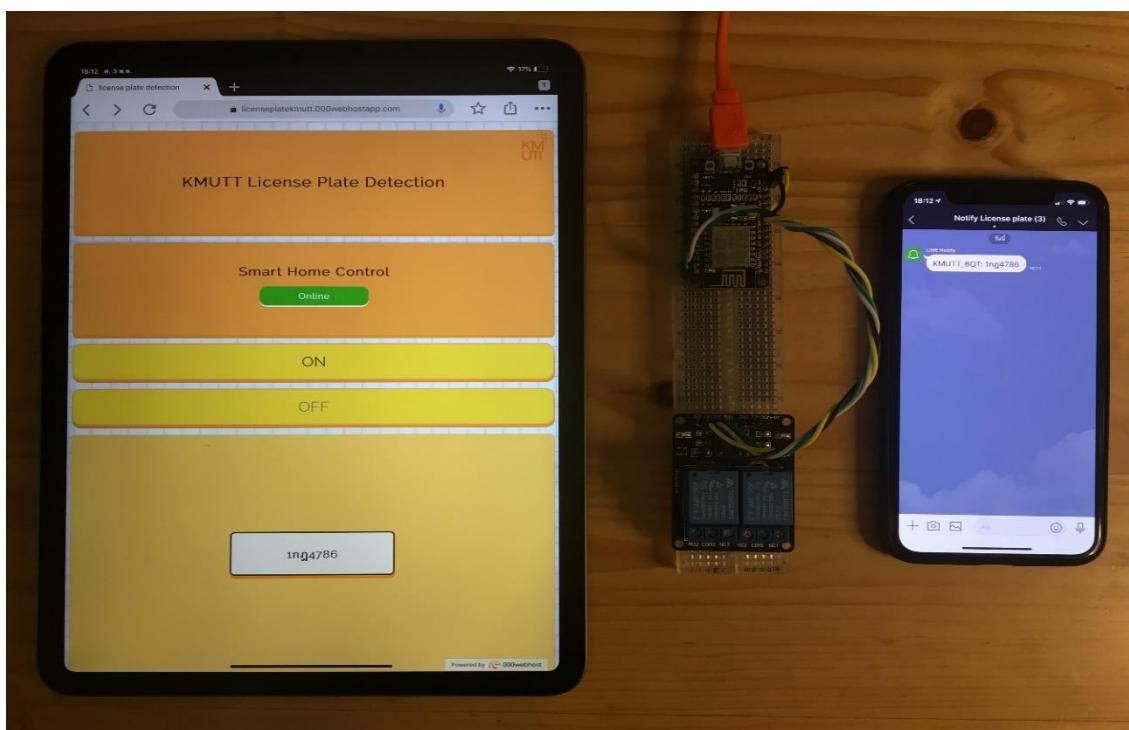
#### **4.3.2 ขั้นตอนการทดลอง**

1. อินพุตฐานข้อมูลที่มีหมายเลขทะเบียน 4กง87 ซึ่งเป็นหมายเลขทะเบียนที่กำหนดไว้ให้ระบบตรวจสอบจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์
2. สังเกตการทำงานของรีเลย์, การแจ้งเตือนผ่านแอพพลิเคชั่นไลน์, การแสดงผลผ่านเว็บแอพพลิเคชั่น และบันทึกผลการทดลอง
3. อินพุตฐานข้อมูลที่มีหมายเลขทะเบียน 1กฉ4786 ซึ่งเป็นหมายเลขทะเบียนที่ไม่ถูกกำหนดไว้ให้ระบบตรวจสอบจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์
4. สังเกตการทำงานของรีเลย์, การแจ้งเตือนผ่านแอพพลิเคชั่นไลน์, การแสดงผลผ่านเว็บแอพพลิเคชั่น และบันทึกผลการทดลอง
5. ใช้เว็บแอพพลิเคชั่นในการควบคุมรีเลย์ให้ ON และ OFF
6. สังเกตการทำงานของรีเลย์, การแจ้งเตือนผ่านแอพพลิเคชั่นไลน์ และบันทึกผลการทดลอง

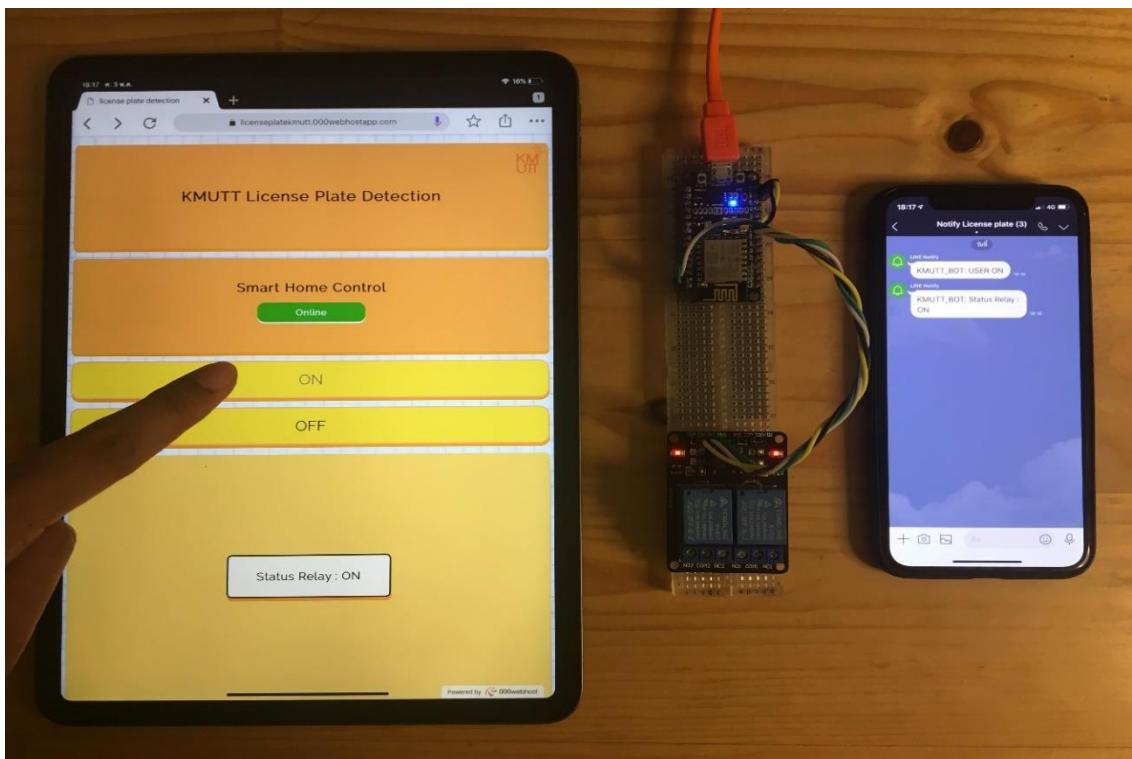
#### 4.3.3 ผลการทดลอง



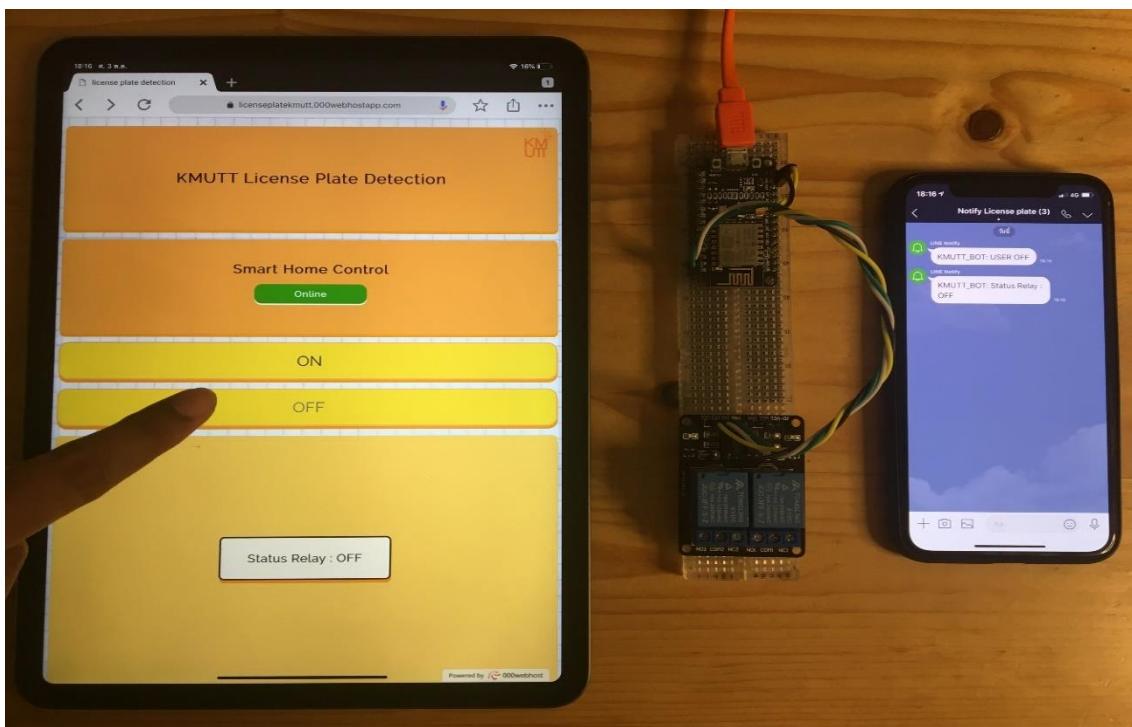
รูปที่ 4.6 ผลการทดลองเมื่อป้ายทะเบียนอินพุตเป็น 4ng4787



รูปที่ 4.7 ผลการทดลองเมื่อป้ายทะเบียนอินพุตเป็น 1ng4786



รูปที่ 4.8 ผลการทดลองความคุณรีเลย์ ON ผ่านเว็บแอพพลิเคชัน



รูปที่ 4.9 ผลการทดลองความคุณรีเลย์ OFF ผ่านเว็บแอพพลิเคชัน

## บทที่ 5

### สรุปผลและวิจารณ์การทดลอง

ในบทนี้จะกล่าวถึง สรุปผลการทดลองของโครงการระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์ ตลอดจนปัญหาและอุปสรรคของการทำโครงการนี้ รวมทั้งข้อเสนอแนะในการนำอาชีวกรรมไปพัฒนาต่อสำหรับผู้ที่สนใจ

#### **5.1 สรุปผลการทดลอง**

##### **5.1.1 การทดลองตรวจจับป้ายทะเบียนรถยนต์**

จากผลการทดลองพบว่า จำนวนขั้นการฝึกฝนตั้งแต่ 20000 steps เป็นต้นไป ไม่ส่งผลต่อการพัฒนาประสิทธิภาพของการตรวจจับป้ายทะเบียนอีกแล้ว เนื่องจากตั้งแต่ 20000 steps ขึ้นไป ไม่เกิดสามารถตรวจจับป้ายทะเบียนรถยนต์ได้แม่นยำถูกต้อง 100 เปอร์เซ็นต์ ไม่ว่าจะทดสอบกับรูปภาพด้านตรง ด้านหลัง หรือด้านเอียงก็ตาม อีกทั้งจำนวนขั้นการฝึกฝนที่มากขึ้นไม่ส่งผลกับความเร็วในการทำงานแต่อย่างใด คือใช้เวลาประมาณ 12 วินาทีในการตรวจจับป้ายทะเบียนของรูปภาพรถยนต์ทั้ง 15 รูป เคลื่อนไหวเวลา 0.8 วินาทีต่อหนึ่งรูป

##### **5.1.2 การทดลองอ่านป้ายทะเบียนรถยนต์**

จากผลการทดลองพบว่า ไม่เกิดตรวจจับป้ายทะเบียนรถยนต์และอ่านป้ายทะเบียนรถยนต์ที่มีประสิทธิภาพมากที่สุดคือ ไม่เกิดตรวจจับป้ายทะเบียนรถยนต์ที่ถูกฝึกฝน 30000 steps และกำหนดค่า k เป็น 2 โดยมีความแม่นยำในการอ่านป้ายทะเบียนรถยนต์ถูกต้อง 86.66 เปอร์เซ็นต์ และใช้เวลา 13.5 วินาทีจากรูปภาพทั้งหมด 15 รูปภาพ เคลื่อนไหวเวลา 0.9 วินาทีต่อหนึ่งรูป

##### **5.1.3 การทดลองควบคุมอุปกรณ์และการแสดงผล**

จากผลการทดลองเมื่ออินพุตรูปภาพรถยนต์ที่มีหมายเลขทะเบียนที่ถูกต้องตามเงื่อนไขสามารถควบคุมรีเลย์ให้ทำงานได้ตามที่กำหนดไว้ และสามารถแสดงผลหมายเลขอื่นๆ ได้ เมื่ออินพุตรูปภาพรถยนต์ที่มีหมายเลขทะเบียนไม่ตรงตามเงื่อนไขจะไม่ส่งการควบคุมรีเลย์แต่แสดงผลหมายเลขอื่นๆ ได้ เช่น ในส่วนของเว็บแอปพลิเคชันสามารถกดเพื่อควบคุมรีเลย์ได้ทั้ง ON และ OFF และเมื่อมีอินพุตรูปภาพรถยนต์เข้ามาที่มีการแสดงผลหมายเลขอื่นๆ ของรถยนต์ได้ถูกต้อง

## 5.2 วิจารณ์การทดลอง

### 5.2.1 ปัญหาและอุปสรรค

1. การฝึกฝนปัญญาประดิษฐ์ของโมเดลโปรแกรมตรวจจับป้ายทะเบียนรถยนต์ใช้ระยะเวลา นาน หากใช้การฝึกฝนบนคลาวด์ เช่น Google Cloud Platform จะทำให้ประหยัดเวลาได้มากพอสมควร

2. การฝึกฝนปัญญาประดิษฐ์ของโมเดลโปรแกรมอ่านป้ายทะเบียนรถยนต์ จำเป็นต้องใช้ตัวอักษรที่เหมือนกับป้ายทะเบียนจริง ๆ แต่ฟอนต์นี้ไม่แจกจ่ายต่อสาธารณะ ผู้จัดทำจึงมีความจำเป็นที่จะต้องใช้ฟอนต์ Sarun's ThangLuang ซึ่งมีลักษณะใกล้เคียงกับตัวอักษรในป้ายทะเบียนมากที่สุด

3. การตั้งค่า minimum score เป็น 0.95 เพื่อให้โปรแกรมตรวจจับป้ายทะเบียนรถยนต์มั่นใจได้ว่าวัตถุที่ตรวจจับได้ต้องมีลักษณะเหมือนป้ายทะเบียนรถยนต์ 95 เปอร์เซ็นต์ ทำให้อาจจะใช้ระยะเวลาที่มากขึ้นในการตรวจจับ การแก้ไขปัญหานี้คือ การฝึกฝนโดยใช้รูปภาพหรือจำนวนขั้นที่มากขึ้น เพื่อทำให้สามารถลด minimum score ได้โดยไม่เกิดความผิดพลาดในการตรวจจับวัตถุ

4. ความแม่นยำในการอ่านป้ายทะเบียนรถยนต์ คือ 86.66 เปอร์เซ็นต์ เนื่องจากบางครั้งโมเดลยังมีความผิดพลาดในการแยกแยะตัวอักษรที่มีความใกล้เคียงกันมาก เช่น ก กับ ก หรือ ข กับ บ ทำให้บางป้ายทะเบียนผิดพลาดแก่ตัวเดียวแต่นั่นก็ทำให้ accuracy score เป็น 0 โดยทันที การแก้ไขคือ การฝึกฝนตัวอักษรให้มากขึ้น การหาฟอนต์ตัวอักษรของป้ายทะเบียนจริง ๆ มาใช้ฝึกฝน หรือ การเปลี่ยนอัลกอริทึมในการอ่านป้ายทะเบียนรถยนต์

### 5.2.2 แนวทางการพัฒนา

1. พัฒนาให้ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์สามารถแยกแยะรูปภาพรถยนต์กับตัวรถยนต์จริง ๆ ได้ เพื่อป้องกันการนำรูปภาพมาแอบอ้างเป็นป้ายทะเบียนรถยนต์ และสามารถพัฒนาเพื่อนำไปใช้ในระบบความปลอดภัยได้

2. พัฒนาให้ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์สามารถอ่านจังหวัดบนป้ายทะเบียนได้

3. พัฒนาให้ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์มีขอบเขตบริเวณการตรวจจับได้ชัดเจน เช่น การกำหนดระยะ เมื่อเข้ามาในระยะที่กำหนดแล้วค่อยตรวจจับ หรืออ่านป้ายทะเบียน

4. พัฒนาโปรแกรมอ่านป้ายทะเบียนรถยนต์ให้มีความแม่นยำมากยิ่งขึ้น

5. พัฒนาให้ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์สามารถแยกแยะยี่ห้อ สี และรุ่นของรถยนต์ได้ เพื่อนำไปประยุกต์ใช้ในงานต่าง ๆ ได้มากยิ่งขึ้น

## เอกสารอ้างอิง

- [1] พศ.ดร.พนมขวัญ ริยะมงคล. Digital Image Processing. สืบค้นเมื่อ 21 กันยายน 2561, จาก <http://wwwecpe.nu.ac.th/panomkhawn/imagepro/pdf/ch10.pdf>
- [2] วิกิพีเดีย สารานุกรมเสรี. (2561). ปัญญาประดิษฐ์. สืบค้นเมื่อ 21 กันยายน 2561, จาก [https://th.wikipedia.org/wiki/ปัญญาประดิษฐ์#cite\\_note-1](https://th.wikipedia.org/wiki/ปัญญาประดิษฐ์#cite_note-1)
- [3] Aditya Ananthram. (2561). Deep Learning For Beginners Using Transfer Learning In Keras. สืบค้นเมื่อ 30 เมษายน 2562, จาก <https://towardsdatascience.com/keras-transfer-learning-for-beginners-6c9b8b7143e>
- [4] ambroseavery. Nodemcu, ESP8266, Wifi. สืบค้นเมื่อ 21 กันยายน 2561, จาก <https://www.thaipng.com/png-s4ift4/>
- [5] Bryant Xu. (2561). Back Propagation for Artificial Neural Network: Part One. สืบค้นเมื่อ 21 กันยายน 2561, จาก <https://www.linkedin.com/pulse/back-propagation-artificial-neural-network-part-one-bryant-xu>
- [6] Chatchawan Niyomthum. (2561). Neural Network 101 : CNN with TensorFlow. สืบค้นเมื่อ 21 กันยายน 2561, จาก <https://medium.com/@thebear19/neural-network-101-cnn-with-tensorflow-fd5d515e979b>
- [7] chullhwan-song. (2561). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. สืบค้นเมื่อ 30 เมษายน 2562, จาก <https://github.com/chullhwan-song/Reading-Paper/issues/44>
- [8] cs231n. (2558). Convolutional Neural Networks (CNNs / ConvNets). สืบค้นเมื่อ 11 กันยายน 2561, จาก <http://cs231n.github.io/convolutional-networks/#pool>
- [9] doxygen. (2560). Morphological Transformations. สืบค้นเมื่อ 30 เมษายน 2562, จาก [https://docs.opencv.org/3.3.0/d9/d61/tutorial\\_py\\_morphological\\_ops.html](https://docs.opencv.org/3.3.0/d9/d61/tutorial_py_morphological_ops.html)
- [10] Electricalcircuitleaderclub. (2561). Great Neuron Diagram Labeled The Introduction To Psychology COPY COURSES. สืบค้นเมื่อ 21 กันยายน 2561, จาก <https://electricalcircuitleaderclub/wiring-diagram-nest-thermostat/>
- [11] Hao Gao. (2561). Understand Single Shot MultiBox Detector (SSD) and Implement It in Pytorch. สืบค้นเมื่อ 30 เมษายน 2562, จาก <https://medium.com/@smallfishbigsea/understand-ssd-and-implement-your-own-caa3232cd6ad>

- [12] iBook Engineering. (2558). โครงข่ายประสาทเทียม (ARTIFICIAL NEURAL NETWORK). สืบค้นเมื่อ 21 กันยายน 2561, จาก <http://ibookengineering.blogspot.com/2015/07/artificial-neural-network.html>
- [13] Javier Abellán Abenza. (2561). HSV color model. สืบค้นเมื่อ 30 เมษายน 2562, จาก <https://medium.com/neurosapiens/segmentation-and-classification-with-hsv-8f2406c62b39>
- [14] mindphp. (2561). Artificial Neural Networks (อาทพิคเคล นิวรอล เน็ตเวริก) คืออะไร โครงข่ายประสาทเทียม. สืบค้นเมื่อ 31 สิงหาคม 2560, จาก <http://mindphp.com/บทความ/240-ai-machine-learning/5659-artificial-neural-networks.html>
- [15] Mitchell Gracie. (2562). Telecommunication and IoT Communities Partner to Push Smart Buildings. สืบค้นเมื่อ 30 เมษายน 2562, จาก <https://www.engineering.com/IOT/ArticleID/18413/Telecommunication-and-IoT-Communities-Partner-to-Push-Smart-Buildings.aspx>
- [16] Natasha Latysheva. (2559). Implementing Your Own k-Nearest Neighbor Algorithm Using Python. สืบค้นเมื่อ 30 เมษายน 2562, จาก <https://www.kdnuggets.com/2016/01/implementing-your-own-knn-using-python.html>
- [17] SPIDYHERO. (2559). NodeMCU control Web Animation through Cloud MQTT ควบคุมเว็บออนไลน์เมื่อ ด้วย NodeMCU ผ่านคลาวด์เซอร์วิซ CloudMQTT. สืบค้นเมื่อ 21 กันยายน 2561, จาก <https://spidyhero.wordpress.com/2016/03/26/nodemcu-control-web-animation-through-cloud-mqtt-%E0%B8>
- [18] Vithan Minaphinant. (2561). Machine Learning คืออะไร?. สืบค้นเมื่อ 21 กันยายน 2561, จาก <http://blog.finnomena.com/machine-learning-คืออะไร-fa8bf6663c07>

## ភាគផន្ទាត់

### ภาคผนวก หมวด ก

โปรแกรมของระบบตรวจสอบป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์

ภาคผนวก ก.

ก. 1. โปรแกรมอ่านป้ายทะเบียนรถชนต์ ประกอบด้วย 4 ไฟล์ได้แก่

### 1.1 test\_Readplate.py

```
import cv2
import numpy as np
import os
import test_char
import paho.mqtt.client as mqtt

broker="m16.cloudmqtt.com"
port=14105
username="wvoywesy"
password="IJXMMEUfM1bj"

#module level variables
#####
SCALAR_BLACK = (0.0, 0.0, 0.0)
SCALAR_WHITE = (255.0, 255.0, 255.0)
SCALAR_YELLOW = (0.0, 255.0, 255.0)
SCALAR_GREEN = (0.0, 255.0, 0.0)
SCALAR_RED = (0.0, 0.0, 255.0)

showSteps = False
#####
####
def main(count):

    blnKNNTrainingSuccessful = test_char.loadKNNDATAAndTrainKNN()

    if blnKNNTrainingSuccessful == False:
        print("\nerror: KNN traning was not successful\n")
```

```
return

# end if

imgPlate = cv2.imread("images{}{}.JPG".format(count))

if imgPlate is None:
    print("\nerror: image not read from file \n\n")
    os.system("pause")
    return
# end if

listOfPossiblePlates,License_Number = test_char.detectCharsInPlates(imgPlate)

#cv2.imshow("imgPlate", imgPlate)

print('\nLicense Number is ',License_Number)

client = mqtt.Client()
client.username_pw_set(username, password)
client.connect(broker,port)
client.publish("License Number",License_Number)
return

# end main

#####
if __name__ == "__main__":
    main()
```

การแก้ไขโค้ดเพื่อนำไปประยุกต์ใช้งานมีดังนี้

```
7 broker="m16.cloudmqtt.com"
8 port=14105
9 username="wvoywesy"
10 password="IJXMMEUFM1bj"
```

```
20 showSteps = False
```

```
49 client.publish("License Number", License_Number)
```

แก้ไขบรรทัดที่ 7 ถึง 10 เพื่อกำหนด broker, port, username และ password ของผู้ใช้งาน

แก้ไขบรรทัดที่ 20 เพื่อกำหนดตัวแปร showSteps ที่แก้ไขเป็น True จะแสดงขั้นตอนรูปผลลัพธ์แต่ละขั้นตอนของการอ่านป้ายทะเบียน

แก้ไขบรรทัดที่ 49 เพื่อกำหนด Topic ที่ต้องการส่งข้อมูลไปคลาวด์ MQTT

## 1.2 test\_char.py

```
import os
import cv2
import numpy as np
import math
import random

import test_Readplate
import Preprocess
import PossibleChar

# module level variables
#####
kNearest = cv2.ml.KNearest_create()
```

```
MIN_PIXEL_WIDTH = 10 #2
```

```
MIN_PIXEL_HEIGHT = 40#8
```

```
MIN_ASPECT_RATIO = 0.25
```

```
MAX_ASPECT_RATIO = 1.0
```

```
MIN_PIXEL_AREA = 50
```

```
MIN_DIAG_SIZE_MULTIPLE_AWAY = 0.3
```

```
MAX_DIAG_SIZE_MULTIPLE_AWAY = 5.0
```

```
MAX_CHANGE_IN_AREA = 0.5
```

```
MAX_CHANGE_IN_WIDTH = 0.8
```

```
MAX_CHANGE_IN_HEIGHT = 0.2
```

```
MAX_ANGLE_BETWEEN_CHARS = 12.0
```

```
MIN_NUMBER_OF_MATCHING_CHARS = 3
```

```
RESIZED_CHAR_IMAGE_WIDTH = 20
```

```
RESIZED_CHAR_IMAGE_HEIGHT = 30
```

```
MIN_CONTOUR_AREA = 100
```

```
#####
###
```

```
def loadKNNDATAAndTrainKNN():
```

```
allContoursWithData = []
validContoursWithData = []

try:
    npaClassifications = np.loadtxt("classifications.txt", np.float32)
except:
    print("error, unable to open classifications.txt, exiting program\n")
    os.system("pause")
    return False

# end try

try:
    npaFlattenedImages = np.loadtxt("flattened_images.txt", np.float32)
except:
    print("error, unable to open flattened_images.txt, exiting program\n")
    os.system("pause")
    return False

# end try

npaClassifications = npaClassifications.reshape((npaClassifications.size, 1))

kNearest.setDefaultK(1)

kNearest.train(npaFlattenedImages, cv2.ml.ROW_SAMPLE, npaClassifications)
return True

# end function

#####
###

def detectCharsInPlates(imgPlate):
    intPlateCounter = 0
    imgContours = None
```

```

contours = []

imgGrayscale, imgThresh = Preprocess.preprocess(imgPlate)

if test_Readplate.showSteps == True: # show steps
#####
cv2.imshow("1a", imgPlate)
cv2.imshow("1b", imgGrayscale)
cv2.imshow("1c", imgThresh)
# end if # show steps
#####

imgThresh = cv2.resize(imgThresh, (0, 0), fx = 3, fy = 3)

thresholdValue, imgThresh = cv2.threshold(imgThresh, 0.0, 255.0, cv2.THRESH_BINARY |
cv2.THRESH_OTSU)

if test_Readplate.showSteps == True: # show steps
#####
cv2.imshow("1d", imgThresh)
# end if # show steps
#####

listOfPossibleCharsInPlate = findPossibleCharsInPlate(imgGrayscale, imgThresh)

if test_Readplate.showSteps == True: # show steps
#####
height, width, numChannels = imgPlate.shape
imgContours = np.zeros((height, width, 3), np.uint8)
del contours[:] # clear the contours list
#####

```

```

for possibleChar in listOfPossibleCharsInPlate:
    contours.append(possibleChar.contour)
# end for

cv2.drawContours(imgContours, contours, -1, test_Readplate.SCALAR_WHITE)

cv2.imshow("2", imgContours)
# end if # show steps
#####
listOfListsOfMatchingCharsInPlate =
findListOfListsOfMatchingChars(listOfPossibleCharsInPlate)

if test_Readplate.showSteps == True: # show steps
#####
imgContours = np.zeros((height, width, 3), np.uint8)
del contours[:]

for listOfMatchingChars in listOfListsOfMatchingCharsInPlate:
    intRandomBlue = random.randint(0, 255)
    intRandomGreen = random.randint(0, 255)
    intRandomRed = random.randint(0, 255)

    for matchingChar in listOfMatchingChars:
        contours.append(matchingChar.contour)
# end for

cv2.drawContours(imgContours, contours, -1, (intRandomBlue, intRandomGreen,
intRandomRed))

# end for

cv2.imshow("3", imgContours)

```

```

# end if # show steps
#####
if (len(listOfListsOfMatchingCharsInPlate) == 0):
    #####
    if test_Readplate.showSteps == True: # show steps
    #####
    print("chars found in plate number " + str(
        intPlateCounter) + " = (none), click on any image and press a key to continue . . .")
    intPlateCounter = intPlateCounter + 1
    cv2.destroyWindow("4")
    cv2.destroyWindow("5")
    cv2.destroyWindow("6")
    cv2.waitKey(0)
# end if # show steps
#####

strChars = "Not found license number"
return imgPlate,strChars

elif not (len(listOfListsOfMatchingCharsInPlate) == 0):
    for i in range(0, len(listOfListsOfMatchingCharsInPlate)):
        listOfListsOfMatchingCharsInPlate[i].sort(key = lambda matchingChar:
            matchingChar.intCenterX)
        listOfListsOfMatchingCharsInPlate[i] =
            removeInnerOverlappingChars(listOfListsOfMatchingCharsInPlate[i])
    # end for

    if test_Readplate.showSteps == True: # show steps
    #####
    imgContours = np.zeros((height, width, 3), np.uint8)

```

```
for listOfMatchingChars in listOfListsOfMatchingCharsInPlate:  
    intRandomBlue = random.randint(0, 255)  
    intRandomGreen = random.randint(0, 255)  
    intRandomRed = random.randint(0, 255)  
  
    del contours[:]  
  
    for matchingChar in listOfMatchingChars:  
        contours.append(matchingChar.contour)  
    # end for  
  
    cv2.drawContours(imgContours, contours, -1, (intRandomBlue, intRandomGreen,  
intRandomRed))  
    # end for  
    cv2.imshow("4", imgContours)  
    # end if # show steps  
#####  
  
intLenOfLongestListOfChars = 0  
intIndexOfLongestListOfChars = 0  
  
for i in range(0, len(listOfListsOfMatchingCharsInPlate)):  
    if len(listOfListsOfMatchingCharsInPlate[i]) > intLenOfLongestListOfChars:  
        intLenOfLongestListOfChars = len(listOfListsOfMatchingCharsInPlate[i])  
        intIndexOfLongestListOfChars = i  
    # end if  
# end for  
  
longestListOfMatchingCharsInPlate =  
listOfListsOfMatchingCharsInPlate[intIndexOfLongestListOfChars]
```

```

if test_Readplate.showSteps == True: # show steps
#####
imgContours = np.zeros((height, width, 3), np.uint8)
del contours[:]

for matchingChar in longestListOfMatchingCharsInPlate:
    contours.append(matchingChar.contour)
# end for

cv2.drawContours(imgContours, contours, -1, test_Readplate.SCALAR_WHITE)

cv2.imshow("5", imgContours)
# end if # show steps
#####

strChars = recognizeCharsInPlate(imgThresh, longestListOfMatchingCharsInPlate)

if test_Readplate.showSteps == True: # show steps
#####
print("chars found in plate number " + str(
    intPlateCounter) + " = " + strChars + ", click on any image and press a key to continue . .
.")
intPlateCounter = intPlateCounter + 1
cv2.waitKey(0)
# end if # show steps
#####

if test_Readplate.showSteps == True:
    print("\nchar detection complete, click on any image and press a key to continue . . .\n")
    cv2.waitKey(0)

return imgPlate,strChars

```

```

# end if

# end function

def findPossibleCharsInPlate(imgGrayscale, imgThresh):
    listOfPossibleChars = []
    contours = []
    imgThreshCopy = imgThresh.copy()

    imgContours, contours, npaHierarchy = cv2.findContours(imgThreshCopy, cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)

    for contour in contours:
        possibleChar = PossibleChar.PossibleChar(contour)

        if checkIfPossibleChar(possibleChar):
            listOfPossibleChars.append(possibleChar)

    return listOfPossibleChars

# end function

def checkIfPossibleChar(possibleChar):
    if (possibleChar.intBoundingRectArea > MIN_PIXEL_AREA and
        possibleChar.intBoundingRectWidth > MIN_PIXEL_WIDTH and
        possibleChar.intBoundingRectHeight > MIN_PIXEL_HEIGHT and
        MIN_ASPECT_RATIO < possibleChar.fltAspectRatio and possibleChar.fltAspectRatio
        < MAX_ASPECT_RATIO):
        return True
    else:
        return False

# end if

```

```

# end function

def findListOfListsOfMatchingChars(listOfPossibleChars):

    listOfListsOfMatchingChars = []

    for possibleChar in listOfPossibleChars:
        listOfMatchingChars = findListOfMatchingChars(possibleChar, listOfPossibleChars)

        listOfMatchingChars.append(possibleChar)

        if len(listOfMatchingChars) < MIN_NUMBER_OF_MATCHING_CHARS:
            continue

        # end if

        listOfListsOfMatchingChars.append(listOfMatchingChars)

        listOfPossibleCharsWithCurrentMatchesRemoved = []

        listOfPossibleCharsWithCurrentMatchesRemoved = list(set(listOfPossibleChars) -
set(listOfMatchingChars))

        recursiveListOfListsOfMatchingChars =
findListOfListsOfMatchingChars(listOfPossibleCharsWithCurrentMatchesRemoved)

        for recursiveListOfMatchingChars in recursiveListOfListsOfMatchingChars:
            listOfListsOfMatchingChars.append(recursiveListOfMatchingChars)

        # end for
    
```

```

        break      # exit for

    # end for

    return listOfListsOfMatchingChars
# end function

def findListOfMatchingChars(possibleChar, listOfChars):

    listOfMatchingChars = []

    for possibleMatchingChar in listOfChars:
        if possibleMatchingChar == possibleChar:
            continue
        # end if

        fltDistanceBetweenChars = distanceBetweenChars(possibleChar, possibleMatchingChar)

        fltAngleBetweenChars = angleBetweenChars(possibleChar, possibleMatchingChar)

        fltChangeInArea = float(abs(possibleMatchingChar.intBoundingRectArea -
possibleChar.intBoundingRectArea)) / float(possibleChar.intBoundingRectArea)

        fltChangeInWidth = float(abs(possibleMatchingChar.intBoundingRectWidth -
possibleChar.intBoundingRectWidth)) / float(possibleChar.intBoundingRectWidth)

        fltChangeInHeight = float(abs(possibleMatchingChar.intBoundingRectHeight -
possibleChar.intBoundingRectHeight)) / float(possibleChar.intBoundingRectHeight)

        if (fltDistanceBetweenChars < (possibleChar.fltDiagonalSize *
MAX_DIAG_SIZE_MULTIPLE_AWAY) and

```

```

    fltAngleBetweenChars < MAX_ANGLE_BETWEEN_CHARS and
    fltChangeInArea < MAX_CHANGE_IN_AREA and
    fltChangeInWidth < MAX_CHANGE_IN_WIDTH and
    fltChangeInHeight < MAX_CHANGE_IN_HEIGHT):

        listOfMatchingChars.append(possibleMatchingChar)

    # end if

# end for

return listOfMatchingChars           # return result

# end function

#####
#####

# use Pythagorean theorem to calculate distance between two chars

def distanceBetweenChars(firstChar, secondChar):
    intX = abs(firstChar.intCenterX - secondChar.intCenterX)
    intY = abs(firstChar.intCenterY - secondChar.intCenterY)

    return math.sqrt((intX ** 2) + (intY ** 2))

# end function

#####
#####

# use basic trigonometry (SOH CAH TOA) to calculate angle between chars

def angleBetweenChars(firstChar, secondChar):
    fltAdj = float(abs(firstChar.intCenterX - secondChar.intCenterX))
    fltOpp = float(abs(firstChar.intCenterY - secondChar.intCenterY))

    if fltAdj != 0.0:
        fltAngleInRad = math.atan(fltOpp / fltAdj)

    else:

```

```

    fltAngleInRad = 1.5708

    # end if

    fltAngleInDeg = fltAngleInRad * (180.0 / math.pi)

    return fltAngleInDeg

# end function

#####
#####

def removeInnerOverlappingChars(listOfMatchingChars):
    listOfMatchingCharsWithInnerCharRemoved = list(listOfMatchingChars)

    for currentChar in listOfMatchingChars:
        for otherChar in listOfMatchingChars:
            if currentChar != otherChar:

                if distanceBetweenChars(currentChar, otherChar) < (currentChar.fltDiagonalSize *
MIN_DIAG_SIZE_MULTIPLE_AWAY):

                    if currentChar.intBoundingRectArea < otherChar.intBoundingRectArea:
                        if currentChar in listOfMatchingCharsWithInnerCharRemoved:
                            listOfMatchingCharsWithInnerCharRemoved.remove(currentChar)
                        # end if

                    else:
                        if otherChar in listOfMatchingCharsWithInnerCharRemoved:
                            listOfMatchingCharsWithInnerCharRemoved.remove(otherChar)
                        # end if

                    # end if

                # end if

            # end if

        # end if

    # end if

```



```
imgROIResized = cv2.resize(imgROI, (RESIZED_CHAR_IMAGE_WIDTH,
RESIZED_CHAR_IMAGE_HEIGHT))

npaROIResized = imgROIResized.reshape((1, RESIZED_CHAR_IMAGE_WIDTH *
RESIZED_CHAR_IMAGE_HEIGHT))

npaROIResized = np.float32(npaROIResized)

retval, npaResults, neigh_resp, dists = kNearest.findNearest(npaROIResized, k =2)

strCurrentChar = str(chr(int(npaResults[0][0])))

strChars = strChars + strCurrentChar

# end for

if test_Readplate.showSteps == True: # show steps
#####
cv2.imshow("6", imgThreshColor)

# end if # show steps
#####

return strChars

# end function
```

การแก้ไขโค้ดเพื่อนำไปประยุกต์ใช้งานมีดังนี้

```

17 MIN_PIXEL_WIDTH = 10 #2
18 MIN_PIXEL_HEIGHT = 40#8
19
20 MIN_ASPECT_RATIO = 0.25
21 MAX_ASPECT_RATIO = 1.0
22
23 MIN_PIXEL_AREA = 50
24
25
26 MIN_DIAG_SIZE_MULTIPLE_AWAY = 0.3
27 MAX_DIAG_SIZE_MULTIPLE_AWAY = 5.0
28
29 MAX_CHANGE_IN_AREA = 0.5
30
31 MAX_CHANGE_IN_WIDTH = 0.8
32 MAX_CHANGE_IN_HEIGHT = 0.2
33
34 MAX_ANGLE_BETWEEN_CHARS = 12.0
35
36
37 MIN_NUMBER_OF_MATCHING_CHARS = 3
38
39 RESIZED_CHAR_IMAGE_WIDTH = 20
40 RESIZED_CHAR_IMAGE_HEIGHT = 30
41
42 MIN_CONTOUR_AREA = 100

409
410     retval, npaResults, neigh_resp, dists = kNearest.findNearest(npaROIResized, k =2)
411

```

แก้ไขบรรทัดที่ 17 ถึง 42 เพื่อกำหนดค่าตัวแปรที่เหมาะสมกับการใช้งาน  
 แก้ไขบรรทัดที่ 410 เพื่อกำหนดค่าพารามิเตอร์ k สำหรับ k Nearest Neighbors

### 1.3 Preprocess.py

```

# Preprocess.py

import cv2
import numpy as np
import math

# module level variables
#####
GAUSSIAN_SMOOTH_FILTER_SIZE = (5, 5)
ADAPTIVE_THRESH_BLOCK_SIZE = 19
ADAPTIVE_THRESH_WEIGHT = 9

```

```
#####
#####
def preprocess(imgOriginal):
    imgGrayscale = extractValue(imgOriginal)

    imgMaxContrastGrayscale = maximizeContrast(imgGrayscale)

    height, width = imgGrayscale.shape

    imgBlurred = np.zeros((height, width, 1), np.uint8)

    imgBlurred = cv2.GaussianBlur(imgMaxContrastGrayscale,
        GAUSSIAN_SMOOTH_FILTER_SIZE, 0)

    imgThresh = cv2.adaptiveThreshold(imgBlurred, 255.0,
        cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
        cv2.THRESH_BINARY_INV, ADAPTIVE_THRESH_BLOCK_SIZE,
        ADAPTIVE_THRESH_WEIGHT)

    return imgGrayscale, imgThresh
# end function

#####
#####
def extractValue(imgOriginal):
    height, width, numChannels = imgOriginal.shape

    imgHSV = np.zeros((height, width, 3), np.uint8)

    imgHSV = cv2.cvtColor(imgOriginal, cv2.COLOR_BGR2HSV)
```

```

imgHue, imgSaturation, imgValue = cv2.split(imgHSV)

return imgValue
# end function

#####
#####

def maximizeContrast(imgGrayscale):

    height, width = imgGrayscale.shape

    imgTopHat = np.zeros((height, width, 1), np.uint8)
    imgBlackHat = np.zeros((height, width, 1), np.uint8)

    structuringElement = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))

    imgTopHat = cv2.morphologyEx(imgGrayscale, cv2.MORPH_TOPHAT, structuringElement)
    imgBlackHat = cv2.morphologyEx(imgGrayscale, cv2.MORPH_BLACKHAT,
        structuringElement)

    imgGrayscalePlusTopHat = cv2.add(imgGrayscale, imgTopHat)
    imgGrayscalePlusTopHatMinusBlackHat = cv2.subtract(imgGrayscalePlusTopHat,
        imgBlackHat)

    return imgGrayscalePlusTopHatMinusBlackHat
# end function

```

#### 1.4 PossibleChar.py

```
# PossibleChar.py
```



```

    self.fltAspectRatio = float(self.intBoundingRectWidth) / float(self.intBoundingRectHeight)

# end constructor

# end class

```

ก. 2. โปรแกรมสำหรับเรียกใช้ระบบตรวจจับป้ายทะเบียนรถยนต์โดยปัญญาประดิษฐ์ มีทั้งหมด 3 ไฟล์ตามประเภทอินพุต ได้แก่

### 2.1 numplate\_recognition\_detection\_Images.py สำหรับอินพุตเป็นรูปภาพ

```

import numpy as np
import os
import tensorflow as tf
from PIL import Image
import cv2
import test_Readplate

from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as vis_util

MODEL_NAME = 'ssdlite_20000'
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
PATH_TO_LABELS = os.path.join('training', 'ssdlite','object-detection.pbtxt')
NUM_CLASSES = 1

def load_image_into_numpy_array(image):
    (im_width, im_height) = image.size
    return np.array(image.getdata()).reshape(
        (im_height, im_width, 3)).astype(np.uint8)

def save_image(img_data2,count):
    img = img_data2

```

```

filename2 = os.path.join('images{}.JPG'.format(count))

cv2.imwrite(filename2,img)

detection_graph = tf.Graph()

with detection_graph.as_default():

    od_graph_def = tf.GraphDef()

    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:

        serialized_graph = fid.read()

        od_graph_def.ParseFromString(serialized_graph)

        tf.import_graph_def(od_graph_def, name="")

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)

categories = label_map_util.convert_label_map_to_categories(
    label_map, max_num_classes=NUM_CLASSES, use_display_name=True)

category_index = label_map_util.create_category_index(categories)

PATH_TO_TEST_IMAGES_DIR = 'png_tesseract/test_tesseract'

TEST_IMAGE_PATHS = [ os.path.join(PATH_TO_TEST_IMAGES_DIR, '{}.jpg'.format(i)) for i
in range(1,9) ]

with detection_graph.as_default():

    with tf.Session(graph=detection_graph) as sess:

        image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

        detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')

        detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')

        detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

        num_detections = detection_graph.get_tensor_by_name('num_detections:0')

        count=1

        for image_path in TEST_IMAGE_PATHS:

            image = Image.open(image_path)

            image_np = load_image_into_numpy_array(image)

```

```

image_np_expanded = np.expand_dims(image_np, axis=0)

(boxes, scores, classes, num) = sess.run(
    [detection_boxes, detection_scores, detection_classes, num_detections]
    ,feed_dict={image_tensor: image_np_expanded})

ymin = boxes[0,0,0]
xmin = boxes[0,0,1]
ymax = boxes[0,0,2]
xmax = boxes[0,0,3]

(im_width, im_height) = image.size

(xminn, xmaxx, yminn, ymaxx) = (xmin * im_width, xmax * im_width,
                                 ymin * im_height, ymax * im_height)

cropped_image = tf.image.crop_to_bounding_box(image_np, int(yminn),
                                              int(xminn),int(ymaxx - yminn), int(xmaxx - xminn))

img_data2 = sess.run(cropped_image)

filename2 = save_image(img_data2,count)

test_Readplate.main(count)

count +=1

12 MODEL_NAME = 'ssdlite_30000'
13 PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
14 PATH_TO_LABELS = os.path.join('training', 'ssdlite','object-detection.pbtxt')
15 NUM_CLASSES = 1

46 PATH_TO_TEST_IMAGES_DIR = 'png_tesseract/test_tesseract'
47 TEST_IMAGE_PATHS = [ os.path.join(PATH_TO_TEST_IMAGES_DIR, '{}.jpg'.format(i)) for i in range(1,9) ]

```

แก้ไขบรรทัดที่ 12 ถึง 15 ซึ่งคือการกำหนดตำแหน่งของโมเดลที่ฝึกฝนเรียบร้อยแล้ว

แก้ไขบรรทัดที่ 46 และ 47 เพื่อกำหนดตำแหน่งของไฟล์รูปภาพที่จะนำมาเป็นอินพุต

## 2.2 numplate\_recognition\_detection\_webcam.py สำหรับอินพุตจากเว็บแคม

```

import numpy as np
import os
import tensorflow as tf

```

```

from PIL import Image
import cv2
import test_Readplate
import time

from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as vis_util


def load_image_into_numpy_array(image):
    (im_width, im_height) = image.size
    return np.array(image.getdata()).reshape(
        (im_height, im_width, 3)).astype(np.uint8)

def save_image(img_data2,count):
    img = img_data2
    filename2 = os.path.join(f'images\{count}.JPG')
    cv2.imwrite(filename2,img)

cap = cv2.VideoCapture(0)

MODEL_NAME = 'ssdlite_30000'
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
PATH_TO_LABELS = os.path.join('training','ssdlite', 'object-detection.pbtxt')
NUM_CLASSES = 1

detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()

```

```

od_graph_def.ParseFromString(serialized_graph)
tf.import_graph_def(od_graph_def, name="")

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(
    label_map, max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

with detection_graph.as_default():

    with tf.Session(graph=detection_graph) as sess:

        image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
        detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
        detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
        detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')
        num_detections = detection_graph.get_tensor_by_name('num_detections:0')

        while True:

            ret, image_np = cap.read()
            image_np_expanded = np.expand_dims(image_np, axis=0)
            (boxes, scores, classes, num) = sess.run(
                [detection_boxes, detection_scores, detection_classes, num_detections],
                feed_dict={image_tensor: image_np_expanded})

            vis_util.visualize_boxes_and_labels_on_image_array(
                image_np,
                np.squeeze(boxes),
                np.squeeze(classes).astype(np.int32),
                np.squeeze(scores),
                category_index,
                use_normalized_coordinates=True,
                line_thickness=5)
            cv2.imshow('numplate detection',image_np)
            if cv2.waitKey(1) == ord('q'):
                cv2.destroyAllWindows()

```

```

break

if os.path.isfile('frame1.jpg'):

    count=1

    image = Image.open('frame1.jpg')

    image_detect = load_image_into_numpy_array(image)

    ymin = boxes[0,0,0]

    xmin = boxes[0,0,1]

    ymax = boxes[0,0,2]

    xmax = boxes[0,0,3]

    (im_width, im_height) = image.size

    (xminn, xmaxx, yminn, ymaxxx) = (xmin * im_width, xmax * im_width, ymin * im_height,

    ymax * im_height)

    cropped_image = tf.image.crop_to_bounding_box(image_detect, int(yminn),

    int(xminn),int(ymaxxx - yminn), int(xmaxx - xminn))

    img_data = sess.run(cropped_image)

    filename = save_image(img_data,count)

    test_Readplate.main(count)

    os.remove('frame0.jpg')

    os.remove('frame1.jpg')

    time.sleep(1.5)

```

### การแก้ไขโค้ดเพื่อนำไปประยุกต์ใช้งานมีดังนี้

```

27 MODEL_NAME = 'ssdlite_30000'
28 PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
29 PATH_TO_LABELS = os.path.join('training','ssdlite', 'object-detection.pbtxt')
30 NUM_CLASSES = 1

```

แก้ไขบรรทัดที่ 27 ถึง 30 ซึ่งคือการกำหนดตำแหน่งของโมเดลที่ฝึกฝนเรียบร้อยแล้ว

### 2.3 numplate\_recognition\_detection\_IP\_Cam.py สำหรับอินพุตจากกล้อง IP Camera

```

import numpy as np

import os

import tensorflow as tf

```

```

from PIL import Image
import cv2
import test_Readplate
import time

from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as vis_util

def load_image_into_numpy_array(image):
    (im_width, im_height) = image.size
    return np.array(image.getdata()).reshape(
        (im_height, im_width, 3)).astype(np.uint8)

def save_image(img_data2,count):
    img = img_data2
    filename2 = os.path.join('images\{count}.JPG')
    cv2.imwrite(filename2,img)

cap = cv2.VideoCapture('http://192.168.43.1:8080/video')

MODEL_NAME = 'ssdlite_30000'
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
PATH_TO_LABELS = os.path.join('training', 'ssdlite','object-detection.pbtxt')
NUM_CLASSES = 1

detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name="")

```

```

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(
    label_map, max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

with detection_graph.as_default():

    with tf.Session(graph=detection_graph) as sess:

        image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
        detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
        detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
        detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')
        num_detections = detection_graph.get_tensor_by_name('num_detections:0')

    while True:

        for i in range(10):

            cap.grab()

            ret, image_np = cap.read()
            image_np_expanded = np.expand_dims(image_np, axis=0)

            (boxes, scores, classes, num) = sess.run(
                [detection_boxes, detection_scores, detection_classes, num_detections],
                feed_dict={image_tensor: image_np_expanded})

            vis_util.visualize_boxes_and_labels_on_image_array(
                image_np,
                np.squeeze(boxes),
                np.squeeze(classes).astype(np.int32),
                np.squeeze(scores),
                category_index,
                use_normalized_coordinates=True,
                line_thickness=5)

            cv2.imshow('numplate detection', image_np)

            if cv2.waitKey(1) == ord('q'):

```

```

cv2.destroyAllWindows()

break

if os.path.isfile('frame1.jpg'):

    count=1

    image = Image.open('frame1.jpg')

    image_detect = load_image_into_numpy_array(image)

    ymin = boxes[0,0,0]

    xmin = boxes[0,0,1]

    ymax = boxes[0,0,2]

    xmax = boxes[0,0,3]

    (im_width, im_height) = image.size

    (xminn, xmaxx, yminn, ymaxxx) = (xmin * im_width, xmax * im_width, ymin * im_height,

    ymax * im_height)

    cropped_image = tf.image.crop_to_bounding_box(image_detect, int(yminn),

    int(xminn),int(ymaxxx - yminn), int(xmaxx - xminn))

    img_data = sess.run(cropped_image)

    filename = save_image(img_data,count)

    test_Readplate.main(count)

    os.remove('frame0.jpg')

    os.remove('frame1.jpg')

    time.sleep(1.5)

```

การแก้ไขโค้ดเพื่อนำไปประยุกต์ใช้งานมีดังนี้

```

24 cap = cv2.VideoCapture('http://192.168.43.1:8080/video')
25
26 MODEL_NAME = 'ssdlite_30000'
27 PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
28 PATH_TO_LABELS = os.path.join('training', 'ssdlite','object-detection.pbtxt')
29 NUM_CLASSES = 1

```

แก้ไขบรรทัดที่ 24 เพื่อระบุหมายเลข IP จากกล้องของผู้ใช้งาน  
 แก้ไขบรรทัดที่ 26 ถึง 29 ซึ่งคือการกำหนดตำแหน่งของโมเดลที่ศึกษาเรียบร้อย  
 แล้ว

ก. 3. โปรแกรมอื่น ๆ ที่จำเป็นต้องแก้ไขเพื่อการใช้งานระบบตรวจจับป้ายทะเบียนรถยนต์โดย  
 ปัญญาประดิษฐ์กรณีที่อินพุตเป็น เว็บแคม และ IP Camera

3.1 ไฟล์ visualization\_utils ของ Tensorflow อูฟ์ที่ดำเนินการที่ติดตั้ง Tensorflow จาก github ดาวน์โหลดมาแล้วจะมีไฟล์ชื่อ C:\tensorflow\models\research\object\_detection\utils เป็นต้น โดยการแก้ไขคือการเพิ่มเติมโค้ดบางส่วนในฟังก์ชัน visualize\_boxes\_and\_labels\_on\_image\_array โค้ดส่วนนี้จะเนื่องกับไฟล์ visualization\_utils ของ Tensorflow ดังเดิม โค้ดด้านล่างนี้จึงเป็นเพียงโค้ดในฟังก์ชัน visualize\_boxes\_and\_labels\_on\_image\_array เพียงเท่านั้น

```
def visualize_boxes_and_labels_on_image_array(
    image,
    boxes,
    classes,
    scores,
    category_index,
    instance_masks=None,
    instance_boundaries=None,
    keypoints=None,
    use_normalized_coordinates=False,
    max_boxes_to_draw=20,
    min_score_thresh=.5,
    agnostic_mode=False,
    line_thickness=4,
    groundtruth_box_visualization_color='black',
    skip_scores=False,
    skip_labels=False):
    """Overlay labeled boxes on an image with formatted scores and label names.
```

This function groups boxes that correspond to the same location  
and creates a display string for each detection and overlays these  
on the image. Note that this function modifies the image in place, and returns  
that same image.

Args:

image: uint8 numpy array with shape (img\_height, img\_width, 3)

boxes: a numpy array of shape [N, 4]

classes: a numpy array of shape [N]. Note that class indices are 1-based, and match the keys in the label map.

scores: a numpy array of shape [N] or None. If scores=None, then this function assumes that the boxes to be plotted are groundtruth boxes and plot all boxes as black with no classes or scores.

category\_index: a dict containing category dictionaries (each holding category index 'id' and category name 'name') keyed by category indices.

instance\_masks: a numpy array of shape [N, image\_height, image\_width] with values ranging between 0 and 1, can be None.

instance\_boundaries: a numpy array of shape [N, image\_height, image\_width] with values ranging between 0 and 1, can be None.

keypoints: a numpy array of shape [N, num\_keypoints, 2], can be None

use\_normalized\_coordinates: whether boxes is to be interpreted as normalized coordinates or not.

max\_boxes\_to\_draw: maximum number of boxes to visualize. If None, draw all boxes.

min\_score\_thresh: minimum score threshold for a box to be visualized

agnostic\_mode: boolean (default: False) controlling whether to evaluate in class-agnostic mode or not. This mode will display scores but ignore classes.

line\_thickness: integer (default: 4) controlling line width of the boxes.

groundtruth\_box\_visualization\_color: box color for visualizing groundtruth boxes

skip\_scores: whether to skip score when drawing a single detection

skip\_labels: whether to skip label when drawing a single detection

Returns:

uint8 numpy array with shape (img\_height, img\_width, 3) with overlaid boxes.

"""

```
# Create a display string (and color) for every box location, group any boxes
```

```

# that correspond to the same location.

box_to_display_str_map = collections.defaultdict(list)
box_to_color_map = collections.defaultdict(str)
box_to_instance_masks_map = {}
box_to_instance_boundaries_map = {}
box_to_keypoints_map = collections.defaultdict(list)

if not max_boxes_to_draw:
    max_boxes_to_draw = boxes.shape[0]

for i in range(min(max_boxes_to_draw, boxes.shape[0])):
    if scores is None or scores[i] > 0.95:
        box = tuple(boxes[i].tolist())
        if instance_masks is not None:
            box_to_instance_masks_map[box] = instance_masks[i]
        if instance_boundaries is not None:
            box_to_instance_boundaries_map[box] = instance_boundaries[i]
        if keypoints is not None:
            box_to_keypoints_map[box].extend(keypoints[i])
        if scores is None:
            box_to_color_map[box] = groundtruth_box_visualization_color
        else:
            display_str = ""
            if not skip_labels:
                if not agnostic_mode:
                    if classes[i] in category_index.keys():
                        class_name = category_index[classes[i]]['name']
                        if class_name == 'numplate':
                            if not os.path.exists('frame0.jpg'):
                                time.sleep(1.5)
                                cv2.imwrite("frame0.jpg" , image)
                            elif os.path.exists('frame0.jpg'):
                                cv2.imwrite("frame1.jpg" , image)
                        else:

```

```

class_name = 'N/A'

display_str = str(class_name)

if not skip_scores:

    if not display_str:

        display_str = '{}%'.format(int(100*scores[i]))

    else:

        display_str = '{}: {}'.format(display_str, int(100*scores[i]))

box_to_display_str_map[box].append(display_str)

if agnostic_mode:

    box_to_color_map[box] = 'DarkOrange'

else:

    box_to_color_map[box] = STANDARD_COLORS[
        classes[i] % len(STANDARD_COLORS)]


# Draw all boxes onto image.

for box, color in box_to_color_map.items():

    ymin, xmin, ymax, xmax = box

    if instance_masks is not None:

        draw_mask_on_image_array(
            image,
            box_to_instance_masks_map[box],
            color=color
        )

    if instance_boundaries is not None:

        draw_mask_on_image_array(
            image,
            box_to_instance_boundaries_map[box],
            color='red',
            alpha=1.0
        )

    draw_bounding_box_on_image_array(
        image,

```

```

ymin,
xmin,
ymax,
xmax,
color=color,
thickness=line_thickness,
display_str_list=box_to_display_str_map[box],
use_normalized_coordinates=use_normalized_coordinates)

if keypoints is not None:
    draw_keypoints_on_image_array(
        image,
        box_to_keypoints_map[box],
        color=color,
        radius=line_thickness / 2,
        use_normalized_coordinates=use_normalized_coordinates)

return image

```

โดยลิ้งที่เพิ่มเติมเข้ามาจากไฟล์ต้นฉบับได้แก่

```

693 if not max_boxes_to_draw:
694     max_boxes_to_draw = boxes.shape[0]
695 for i in range(min(max_boxes_to_draw, boxes.shape[0])):
696     if scores is None or scores[i] > 0.95:
697
712         if class_name == 'numplate':
713             if not os.path.exists('frame0.jpg'):
714                 time.sleep(1.5)
715                 cv2.imwrite("frame0.jpg" , image)
716             elif os.path.exists('frame0.jpg'):
717                 cv2.imwrite("frame1.jpg" , image)

```

ในบรรทัดที่ 696 กำหนด minimum score โดยกำหนดให้โปรแกรมมีความแม่นยำมากกว่า 95 เปอร์เซ็นต์ถึงจะตรวจจับวัตถุ

ในบรรทัดที่ 712 ถึง 717 ถ้า class ของวัตถุที่ตรวจจับได้มีชื่อว่า numplate (ป้ายทะเบียนรถยนต์) ให้ตรวจสอบเงื่อนไขเพื่อสร้างไฟล์ frame0.jpg และ frame1.jpg สำหรับนำไปใช้ในโปรแกรม numplate\_recognition\_detection\_webcam.py และ numplate\_recognition\_detection\_IP\_

Cam.py

3.2 ไฟล์ GenData.py มีหน้าที่สร้างไฟล์สำหรับฝึกฝนโมเดล kNN

```
import sys
```

```
import numpy as np
```

```
import cv2
```

```
import os
```

## #modulelevelvariables

MIN CONTOUR AREA = 100

RESIZED IMAGE WIDTH = 20

RESIZED IMAGE HEIGHT = 30

#include <sys/types.h>  
#include <sys/conf.h>

#/#/#

```
def main():
```

```
imgTrainingNumbers = cv2.imread("training_chars.png")
```

if imgTrainingNumbers is None:

```
print ("error: image not read from file \n\n")
```

```
os.system("pause")
```

return

# end if

```
imgGray = cv2.cvtColor(imgTrainingNumbers, cv2.COLOR_BGR2GRAY)
```

```
imgBlurred = cv2.GaussianBlur(imgGray, (5,5), 0)
```

```
imgThresh = cv2.adaptiveThreshold(imgBlurred,
```

255,

## cv2.ADAPTIVE THRESH GAUSSIAN C.

```

cv2.THRESH_BINARY_INV,
11,
2)

cv2.imshow("imgThresh", imgThresh)

imgThreshCopy = imgThresh.copy()

imgContours, npaContours, npaHierarchy = cv2.findContours(imgThreshCopy,
cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

npaFlattenedImages = np.empty((0, RESIZED_IMAGE_WIDTH *
RESIZED_IMAGE_HEIGHT))

intClassifications = []

intValidChars = [48, 49, 50, 51, 52, 53, 54, 55, 56,
57, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,
183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 197, 199, 200, 201, 202, 203, 204, 205, 206, 208,
209, 210, 212, 213, 214, 215, 216, 217, 224, 225, 226, 228, 227, 231, 232, 233, 234, 235, 236, 211]

for npaContour in npaContours:
    if cv2.contourArea(npaContour) > MIN_CONTOUR_AREA:
        [intX, intY, intW, intH] = cv2.boundingRect(npaContour)

        cv2.rectangle(imgTrainingNumbers,
                     (intX, intY),
                     (intX+intW, intY+intH),

```

```
(0, 0, 255),  
2)  
  
imgROI = imgThresh[intY:intY+intH, intX:intX+intW]  
imgROIResized = cv2.resize(imgROI, (RESIZED_IMAGE_WIDTH,  
RESIZED_IMAGE_HEIGHT))  
  
cv2.imshow("imgROI", imgROI)  
cv2.imshow("imgROIResized", imgROIResized)  
cv2.imshow("training_numbers.png", imgTrainingNumbers)  
  
intChar = cv2.waitKey(0)  
  
if intChar == 27:  
    sys.exit()  
elif intChar in intValidChars:  
    if intChar == 161:  
        intClassifications.append(3585)  
    elif intChar == 162:  
        intClassifications.append(3586)  
    elif intChar == 163:  
        intClassifications.append(3587)  
    elif intChar == 164:  
        intClassifications.append(3588)  
    elif intChar == 165:  
        intClassifications.append(3589)  
    elif intChar == 166:  
        intClassifications.append(3590)  
    elif intChar == 167:  
        intClassifications.append(3591)  
    elif intChar == 168:  
        intClassifications.append(3592)
```

```
elif intChar == 169:  
    intClassifications.append(3593)  
elif intChar == 170:  
    intClassifications.append(3594)  
elif intChar == 171:  
    intClassifications.append(3595)  
elif intChar == 172:  
    intClassifications.append(3596)  
elif intChar == 173:  
    intClassifications.append(3597)  
elif intChar == 174:  
    intClassifications.append(3598)  
elif intChar == 175:  
    intClassifications.append(3599)  
elif intChar == 176:  
    intClassifications.append(3600)  
elif intChar == 177:  
    intClassifications.append(3601)  
elif intChar == 178:  
    intClassifications.append(3602)  
elif intChar == 179:  
    intClassifications.append(3603)  
elif intChar == 180:  
    intClassifications.append(3604)  
elif intChar == 181:  
    intClassifications.append(3605)  
elif intChar == 182:  
    intClassifications.append(3606)  
elif intChar == 183:  
    intClassifications.append(3607)  
elif intChar == 184:  
    intClassifications.append(3608)
```

```
elif intChar == 185:  
    intClassifications.append(3609)  
elif intChar == 186:  
    intClassifications.append(3610)  
elif intChar == 187:  
    intClassifications.append(3611)  
elif intChar == 188:  
    intClassifications.append(3612)  
elif intChar == 189:  
    intClassifications.append(3613)  
elif intChar == 190:  
    intClassifications.append(3614)  
elif intChar == 191:  
    intClassifications.append(3615)  
elif intChar == 192:  
    intClassifications.append(3616)  
elif intChar == 193:  
    intClassifications.append(3617)  
elif intChar == 194:  
    intClassifications.append(3618)  
elif intChar == 195:  
    intClassifications.append(3619)  
elif intChar == 196:  
    intClassifications.append(3620)  
elif intChar == 197:  
    intClassifications.append(3621)  
elif intChar == 199:  
    intClassifications.append(3623)  
elif intChar == 200:  
    intClassifications.append(3624)  
elif intChar == 201:  
    intClassifications.append(3625)
```

```
elif intChar == 202:  
    intClassifications.append(3626)  
elif intChar == 203:  
    intClassifications.append(3627)  
elif intChar == 204:  
    intClassifications.append(3628)  
elif intChar == 205:  
    intClassifications.append(3629)  
elif intChar == 206:  
    intClassifications.append(3630)  
elif intChar == 208:  
    intClassifications.append(3632)  
elif intChar == 209:  
    intClassifications.append(3633)  
elif intChar == 210:  
    intClassifications.append(3634)  
elif intChar == 211:  
    intClassifications.append(3635)  
elif intChar == 212:  
    intClassifications.append(3636)  
elif intChar == 213:  
    intClassifications.append(3637)  
elif intChar == 214:  
    intClassifications.append(3638)  
elif intChar == 215:  
    intClassifications.append(3639)  
elif intChar == 216:  
    intClassifications.append(3640)  
elif intChar == 217:  
    intClassifications.append(3641)  
elif intChar == 224:  
    intClassifications.append(3648)
```

```

elif intChar == 225:
    intClassifications.append(3649)

elif intChar == 226:
    intClassifications.append(3650)

elif intChar == 227:
    intClassifications.append(3651)

elif intChar == 228:
    intClassifications.append(3652)

elif intChar == 231:
    intClassifications.append(3655)

elif intChar == 232:
    intClassifications.append(3656)

elif intChar == 233:
    intClassifications.append(3657)

elif intChar == 234:
    intClassifications.append(3658)

elif intChar == 235:
    intClassifications.append(3659)

elif intChar == 236:
    intClassifications.append(3660)

else :
    intClassifications.append(intChar)      # append classification char to integer list of
chars (we will convert to float later before writing to file)

npaFlattenedImage = imgROIResized.reshape((1, RESIZED_IMAGE_WIDTH *
RESIZED_IMAGE_HEIGHT))

npaFlattenedImages = np.append(npaFlattenedImages, npaFlattenedImage, 0)

# end if

# end if

# end for

fltClassifications = np.array(intClassifications, np.float32)

```

```

    npaClassifications = fltClassifications.reshape((fltClassifications.size, 1))

    print ("\n\ntraining complete !!\n")

    np.savetxt("classifications.txt", npaClassifications)
    np.savetxt("flattened_images.txt", npaFlattenedImages)

    cv2.destroyAllWindows()

    return

#####
#####
if __name__ == "__main__":
    main()
# end if

```

#### ก. 4. โปรแกรมควบคุมอุปกรณ์ไฟฟ้าด้วย ESP8266

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

//your network
const char* ssid = "Wi-Fi NAME";           //ชื่อ Wifi
const char* password = "Wi-Fi Password";    //รหัส Wifi

char *Relay_ON ;
char *Relay_OFF ;

//การกำหนด MQTT Server

```

```

#define MQTTserver "YOUR_MQTT_SERVER"      //กำหนด MQTT Server
#define MQTTport YOUR_MQTT_PORT           //กำหนด MQTT Port
#define MQTTuser "YOUR_MQTT_USERNAME"     //กำหนด MQTT User Name
#define MQTTpassword "YOUR_MQTT_PASSWORD"  //กำหนด MQTT password

WiFiClient WIFI_Client;
PubSubClient client(WIFI_Client);

void setup() {
    pinMode(D0,OUTPUT);   //กำหนดสถานะขา D0 ใช้ในการส่งค่า
    pinMode(D1,OUTPUT);   //กำหนดสถานะขา D1 ใช้ในการส่งค่า

    Serial.begin(115200); //กำหนดค่าความเร็วในการรับ-ส่งข้อมูล
    Serial.println();

    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password); //Setup WiFi
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    client.setServer(MQTTserver, MQTTport); //Setup MQTT Server
    client.setCallback(callback);
}

```

```

void loop() {

    if (!client.connected()) {
        if (client.connect("ESP8266Client", MQTTuser, MQTTpassword)) { //ถ้าเชื่อมต่อ MQTT ได้
            Serial.println("connected");
            client.subscribe("License Number");
        }
    } else {           // แต่ถ้าไม่สำเร็จ ทำการหน่วงเวลา 3 วินาที แล้วลองใหม่
        Serial.print("failed !");
        Serial.print(client.state());
        Serial.println(" Try again in 3 seconds");
        delay(3000);
    }
    return;
}

client.loop();
}

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message form topic <");
    Serial.print(topic);
    Serial.print("> : ");
    String message = "";
    int text=0;
    while (text<length) {
        message += (char)payload[text++]; //อ่านข้อความจาก Topic ที่สูกส่งมา
    }

    if (message == "40487") {      //ถ้าข้อความที่ได้รับจาก Topic ตรงกับป้ายทะเบียนที่กำหนด
        digitalWrite(D0,0);       //สั่งขา D0 ให้ Relay ON
        digitalWrite(D1,0);       //สั่งขา D1 ให้ Relay ON
    }
}

```

```

Relay_ON = "Status Relay : ON " ;
client.publish("License Number",Relay_ON); //ทำการส่งข้อความสถานะเปิด กลับไปที่ Topic
}

else if (message == "USER ON") { //ถ้าข้อความที่ได้รับจากผู้ใช้เปิด
digitalWrite(D0,0); //สั่งขา D0 ให้ Relay ON
digitalWrite(D1,0); //สั่งขา D1 ให้ Relay ON
Relay_ON = "Status Relay : ON " ;
client.publish("License Number",Relay_ON); //ทำการส่งข้อความสถานะเปิด กลับไปที่ Topic
}

else if (message == "USER OFF"){ //แต่ถ้าได้รับข้อความจากผู้ใช้ให้ปิด
digitalWrite(D0,1); //สั่งขา D0 ให้ Relay OFF
digitalWrite(D1,1); //สั่งขา D0 ให้ Relay OFF
Relay_OFF = "Status Relay : OFF" ;
client.publish("License Number",Relay_OFF); //ทำการส่งข้อความสถานะปิด กลับไปที่ Topic
}

Serial.println(message); //แสดงข้อความที่ได้รับจาก Topic
}

การแก้ไขโค้ดเพื่อนำไปประยุกต์ใช้งานมีดังนี้
4 //your network
5 const char* ssid = "Wi-Fi NAME"; //ชื่อ Wifi
6 const char* password = "Wi-Fi Password"; //รหัส Wifi

11 //การกำหนด MQTT Server
12 #define MQTTserver "YOUR_MQTT_SERVER" //กำหนด MQTT Server
13 #define MQTTport YOUR_MQTT_PORT //กำหนด MQTT Port
14 #define MQTTuser "YOUR_MQTT_USERNAME" //กำหนด MQTT User Name
15 #define MQTTpassword "YOUR_MQTT_PASSWORD" //กำหนด MQTT password

72 if (message == "4ง87") { //ถ้าข้อความที่ได้รับจาก Topic ตรงกับป้ายทะเบียนที่กำหนด
73   digitalWrite(D0,0); //สั่งขา D0 ให้ Relay ON
74   digitalWrite(D1,0); //สั่งขา D1 ให้ Relay ON
75   Relay_ON = "Status Relay : ON " ;
76   client.publish("License Number",Relay_ON); //ทำการส่งข้อความสถานะเปิด กลับไปที่ Topic
}

```

แก้ไขบรรทัดที่ 5 และ 6 เพื่อกำหนดชื่อ ไวไฟ และพาสเวิร์ด  
 แก้ไขบรรทัดที่ 12 ถึง 16 เพื่อกำหนดค่า MQTT Server ซึ่งใช้ในการรับข้อมูล และ<sup>1</sup>  
 ส่งข้อมูลไปคลาวด์MQTT  
 แก้ไขบรรทัดที่ 72 เพื่อกำหนดหมายเลขเบียนที่ต้องการนำไปควบคุมอุปกรณ์  
 ไฟฟ้า

### ก. 5. โปรแกรมเว็บแอพพลิเคชัน

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>license plate detection</title>
<script src="jquery.js"></script>
<script src="mqtt.js"></script>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway">
<style>
body {
  font-family: "Raleway", sans-serif;
  background-repeat: no-repeat;
  background-image: url("BG.png");
}

#Status_relay{
  max-width:260px;
  min-width: 140px;
  margin-left: auto;
  margin-right: auto;
  background: rgb(255, 255, 255,0.8) ;
  color: #000000;
  font-size: 24px;
  border-radius: 10px;
```

```
padding: 30px 30px;  
text-align: center;  
font-family: "Raleway", sans-serif;  
border: 3px solid #000;  
box-shadow: 1px 7px #ff9900 ;  
}
```

```
#Status_relay_out{  
background: #ffcc66 ;  
color: #000000;  
border-radius: 10px;  
padding: 200px 85px;  
text-align: center;  
font-family: "Raleway", sans-serif;  
box-shadow: 1px 7px #ff9900 ;  
}
```

```
#Status_UI {  
max-width:260px;  
min-width: 140px;  
margin-left: auto;  
margin-right: auto;  
background: #333;  
color: #FFF;  
border-radius: 15px;  
font-weight: bold;  
padding: 3px 6px;  
line-height: 2;  
box-shadow: 0px 3px 1px #fff ;  
  
height:100%;
```

```
width: 30% ;  
  
}  
  
#Status_UI.error_Connect {  
background: #F00;  
color: #FFF;  
box-shadow: 1px 7px #ff9900 ;  
}  
  
#Status_UI.connect {  
background: #ff6a00;  
color: #FFF;  
}  
  
#Status_UI.Online {  
background: #00AE04;  
color: #FFF;  
}  
  
  
  
#Headd {  
background: #ffa64d;  
padding: 100px 140px;  
border-radius: 10px;  
text-align: center;  
box-shadow: 1px 7px #ff9900 ;  
height:100%;  
}  
  
#Headd3 {  
background: #ffa64d;  
padding: 50px 140px;  
border-radius: 10px;
```

```

    text-align: center;
    box-shadow: 1px 7px #ff9900 ;
    height:100%;

}

button {
    margin: auto;
    width: 100%;
    font-size: 32px;
    background: #ffd633;
    text-align: center;
    border-radius: 20px;
    font-family: "Raleway", sans-serif;
    border: 3px solid #ffb366;
    padding: 20px;
    box-shadow: 1px 7px #ff9900 ;
}

}

.BTN2:hover {
    background-color: #FFF;
}

.img1 {
    float: right;
}

</style>
```

```

<script>
var config = {
    Server: "m16.cloudmqtt.com",
    Port: 34105,
    User: "wwoywesy",
```

```

Password: "IJXMMEUfM1bj"
};

$(document).ready(function(e) {
    client = new Paho.MQTT.Client(config.Server, config.Port, "web_" + parseInt(Math.random() *
100, 10));

    client.connect({
        useSSL: true,
        userName: config.User,
        password: config.Password,

        onSuccess: function() {
            $("#Status_UI").text("Online").removeClass().addClass("Online");
            client.subscribe("License Number");
            MQTT_Send("License Number", "USER OFF");
        },
        onFailure: function(e) {
            $("#Status_UI").text("Connect : Error").removeClass().addClass("error_Connect");
        }
    });

    client.onConnectionLost = function(response) {
        if (response.errorCode !== 0) {
            $("#Status_UI").text("onConnectionLost:" +
            response.errorMessage).removeClass().addClass("connect");
            setTimeout(function() { client.connect() }, 1000);
        }
    }

    client.onMessageArrived = function(message) {
        if (message.payloadString != null ) {

```

```

        $("#" + Status_relay).text(message.payloadString).addClass("Status_relay");

    }

    if (message.payloadString == "4f487" || message.payloadString == "USER ON" ||
message.payloadString == "USER OFF") {
        $("#Relay-on").attr("disabled", (message.payloadString == "USER ON" ||
message.payloadString == "4f487" ? true : false));
        $("#Relay-off").attr("disabled", (message.payloadString == "USER OFF" ? true :
false));
    }

    $("#Relay-on").click(function(e) {
        MQTT_Send("License Number", "USER ON");
    });

    $("#Relay-off").click(function(e) {
        MQTT_Send("License Number", "USER OFF");
    });
});

var MQTT_Send = function(MQTT_topic, MQTT_message) {
    var message = new Paho.MQTT.Message(MQTT_message);
    message.destinationName = MQTT_topic;

    client.send(message);
}

```

```
</script>
</head>

</img>
<h1 id="Headd" > KMUTT License Plate Detection </h1>
<h1 id="Headd2" ></h1>
<h3 id="Headd3"> <div style="font-size: 28px;" > Smart Home Control </div>
<p <span id="Status_UI" class="connect">Connect</span> </p></h3>

<button class="BTN" id="Relay-on" disabled>ON</button>
&nbsp;&nbsp;&nbsp;&nbsp;
<button class="BTN2" id="Relay-off" disabled>OFF</button>

<div class="text_status" >
  <h4 id= "Status_relay_out" >
    <p id="Status_relay"> </p>
  </h4>
</div>

</body>
</html>
```

### การแก้ไข โค้ดเพื่อ ภาษา Python ให้ งาน มี ดังนี้

```
118 var config = {  
119     Server: "m16.cloudmqtt.com",  
120     Port: 34105,  
121     User: "wvoywesy",  
122     Password: "IJXMMEUFM1bj"
```

```
156     if (message.payloadString == "4&#x87" || message.payloadString == "USER ON" || message.payloadString == "USER OFF") {  
157         $("#Relay-on").attr("disabled", (message.payloadString == "USER ON" || message.payloadString == "4&#x87" ? true : false));  
158         $("#Relay-off").attr("disabled", (message.payloadString == "USER OFF" ? true : false));
```

แก้ไขบรรทัดที่ 119 ถึง 122 เพื่อตั้งค่า MQTT Server  
แก้ไขบรรทัดที่ 156 กำหนดหมายเลขทะเบียนที่ต้องการให้ตรงกับหมายเลขทะเบียน  
ที่กำหนดเพื่อควบคุมอุปกรณ์ไฟฟ้า