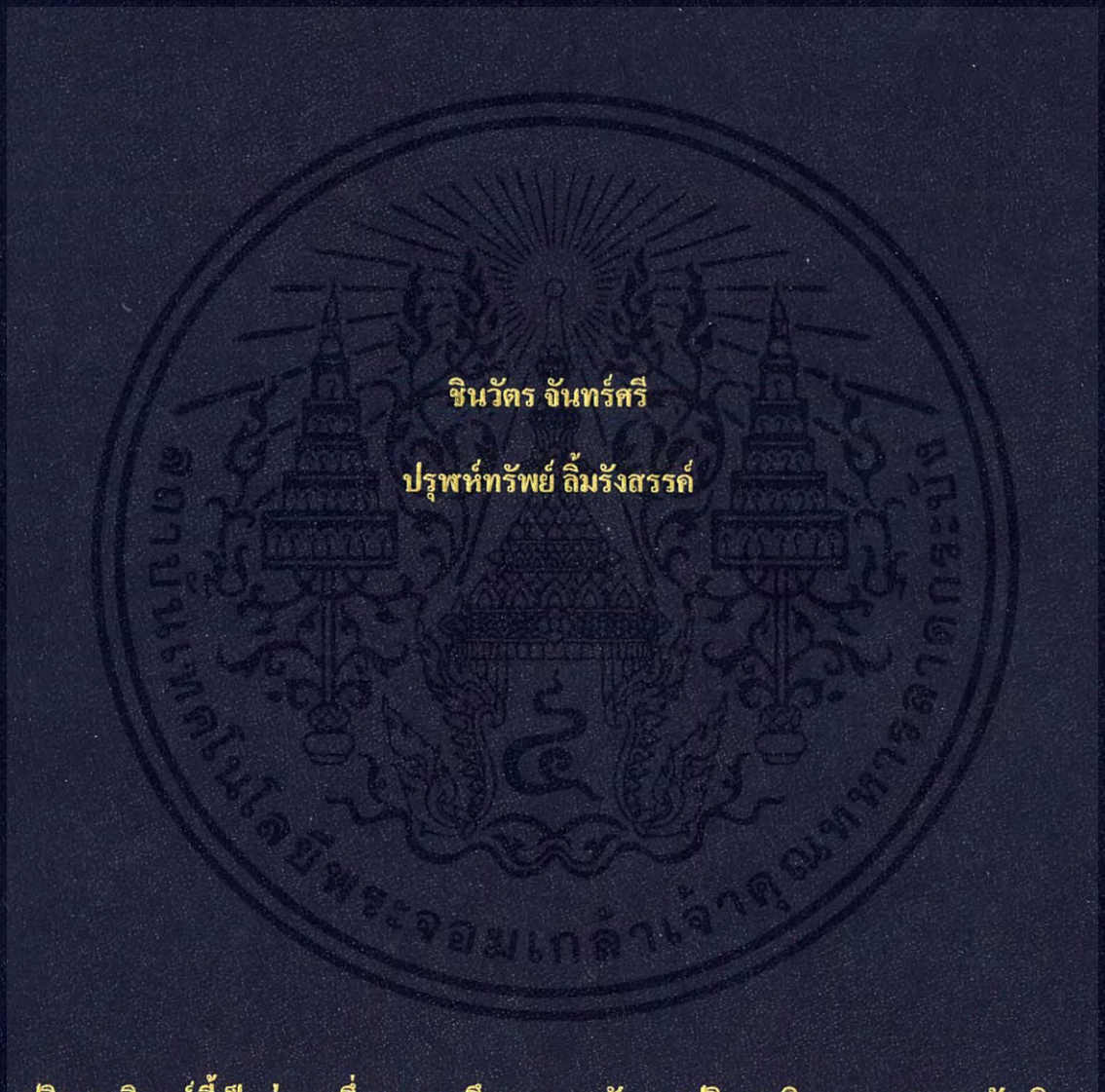
# การตรวจจับและจำแนกประเภทยานพาหนะโดยใช้ DEEP LEARNING และ IMAGE ENHANCEMENT

# VEHICLE DETECTION AND CLASSIFICATION USING DEEP LEARNING AND IMAGE ENHANCEMENT

ชินวัตร จันทร์ศรี

ปรุพห์ทรัพย์ ลิ้มรังสรรค์



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

# การตรวจจับและจำแนกประเภทยานพาหนะโดยใช้ DEEP LEARNING และ IMAGE ENHANCEMENT

# VEHICLE DETECTION AND CLASSIFICATION USING DEEP LEARNING AND IMAGE ENHANCEMENT


ชินวัตร จันทร์ศรี

ปรุพห์ทรัพย์ ลิ้มรังสรรค์


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561


เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การตรวจจับและจำแนกประเภทยานพาหนะโดยใช้ Deep Learning และ Image Enhancement

VEHICLE DETECTION AND CLASSIFICATION USING DEEP LEARNING AND IMAGE ENHANCEMENT

ผู้จัดทำ

1. นายชินวัตร จันทร์ศรี รหัสนักศึกษา 58010297

2. นายปรุฬห์ทรัพย์ ลิ้มรังสรรค์ รหัสนักศึกษา 58010753

อาจารย์ที่ปรึกษา

(อาจารย์สรยุทธ กลมกล่อม)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การตรวจจับและจำแนกประเภทยานพาหนะโดยใช้ Deep Learning และ Image Enhancement

นายชินวัตร จันทร์ศรี 58010297
นายปรุพห์ทรัพย์ ลิ้มรังสรรค์ 58010753
อาจารย์สรยุทธ กลมกล่อม อาจารย์ที่ปรึกษา
ปีการศึกษา 2561

**บทคัดย่อ**

โครงงานนี้มีจุดประสงค์เพื่อสร้างโมเดลตรวจจับและจำแนกประเภทยานพาหนะบนท้องถนน ด้วย Faster Region-Based Convolutional Neural Networks (Faster R-CNN) โดยใช้ TensorFlow Object detection API และกระบวนการ Image Pre-Processing (Contrast Transformation, Brightness Adjustment, Bilateral Filtering, Sharpening และ Grayscale) ในการเพิ่มความแม่นยำให้กับโมเดล ซึ่งทดสอบประสิทธิภาพโดยดูผลจากค่า Mean Average Precision (mAP) และ Accuracy เพื่อให้ได้โมเดลที่แม่นยำที่สุด ข้อดีของงานโครงงานนี้คือการเพิ่มความแม่นยำ ในการตรวจจับและจำแนกประเภทยานพาหนะ ทำให้สามารถวิเคราะห์สภาพการจราจรได้อย่างมีประสิทธิภาพมากขึ้นและนำไปสู่การใช้แก้ปัญหาการจราจรและพัฒนาระบบคมนาคมต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Vehicle Detection and Classification Using Deep Learning and Image Enhancement


Mr. Chinnawat Jansree 58010297

Mr. Paroonsub Limrungsan 58010753

Mr. Sorayut Glomglome Advisor

Academic Year 2561



## ABSTRACT

The purpose of this project is building vehicle detection and classification model for based on Faster Region-Based Convolutional Neural Networks (Faster R-CNN) using TensorFlow Object detection API and Image Pre-Processing (Contrast Transformation, Brightness Adjustment, Bilateral Filtering, Sharpening and Grayscale) to increase precision of the model. The metric to measure the precision of this model is Mean Average Precision (mAP) and Accuracy to make a model that most suitable for vehicle classification. The benefit of this project is to increases the performance of vehicle detection and classification, so we can use that model to analyze traffic conditions effectively and to solve traffic congestion and improve transportation system.




เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงลงไม่ได้หากปราศจากความช่วยเหลือของ อาจารย์ที่ปรึกษา อาจารย์สรยุทธ กลมกล่อม ผู้ให้คำแนะนำ คำปรึกษา และให้ความช่วยเหลือตลอดการทำปริญญานิพนธ์ ซึ่งทำให้การทำงานต่างๆ เป็นไปได้อย่างราบรื่นและสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณอาจารย์และบุคลากรต่างๆ ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำและสั่งความรู้ต่างๆ มาโดยตลอด รวมถึงห้องแล็ป HRCL (The Hybrid Computing research Laboratory) ที่ได้เอื้อเฟื้อสถานที่ในการทำวิจัยและพัฒนาปริญญานิพนธ์

ขอขอบคุณรุ่นพี่และเพื่อนหลาย ๆ คนในภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำคำปรึกษาและแบ่งปันความรู้ในทุก ๆ ด้าน

ในสุดท้ายนี้ ขอขอบคุณ บิดา มารดา และครอบครัว ที่ได้เลี้ยงดู สั่งสอน และให้การสนับสนุน พร้อมทั้งให้โอกาสในการศึกษาและให้กำลังใจเสมอมา

ชินวัตร จันทร์ศรี

ปรุพห์ทรัพย์ ลิ้มรังสรรค์



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า


บทคัดย่อภาษาไทย ..... I

บทคัดย่อภาษาอังกฤษ ..... II

กิตติกรรมประกาศ ..... III

สารบัญ ..... IV

สารบัญตาราง ..... VI

สารบัญรูป ..... VII

บทที่ 1 บทนำ ..... 1

1.1 ความเป็นมาของปัญหา ..... 1

1.2 วัตถุประสงค์โครงงาน ..... 3

1.3 ประโยชน์ที่คาดว่าจะได้รับ ..... 3

1.4 ขอบเขตของโครงงาน ..... 3

1.5 ข้อจำกัดของโปรแกรม ..... 3

1.6 แผนการดำเนินงาน ..... 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง ..... 6

2.1 Machine Learning (ML) ..... 6

2.2 TensorFlow ..... 12

2.3 Image Enhancement ..... 13

2.4 Mean Average Precision (mAP) ..... 15

2.5 งานวิจัยที่เกี่ยวข้อง ..... 17




เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

หน้า


บทที่ 3 การออกแบบและพัฒนา ..... 21

3.1 ภาพรวมการทำงานภายในระบบ ..... 21

3.2 การออกแบบขั้นตอนการทำงานของระบบ ..... 22

บทที่ 4 การทดลอง ..... 25

4.1 การทดลองเกี่ยวกับการเลือกใช้ Datasets ..... 26

4.2 การทดลองเกี่ยวกับการเลือกใช้ความละเอียดของ Input ..... 29

4.3 การทดลองเกี่ยวกับการเลือกใช้ Image Pre-Processing ..... 43

4.4 สรุปการทดลอง ..... 44

บทที่ 5 บทสรุปและข้อเสนอแนะ ..... 46

5.1 บทสรุป ..... 46

5.2 ปัญหาและอุปสรรค ..... 46

5.3 แนวทางในการพัฒนาต่อ ..... 47




เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตาราง หน้า

1.1 แสดงแผนการดำเนินงาน (1) ..... 4
1.2 แสดงแผนการดำเนินงาน (2) ..... 5
4.1 แสดงตารางเปรียบเทียบ mAP ระหว่างโมเดล 240p และ 480p ..... 28
4.2 แสดงตารางเปรียบเทียบAPและ mAP ของแต่ละความละเอียดภาพของInput ..... 31
4.3 แสดงตัวอย่าง Confusion Matrix จากโมเดลที่ Pre-Processing ด้วยวิธี Blurring (Bilateral Filtering, d=6) ..... 36
4.4 แสดงค่า TP, TN, FP, FN, Accuracy, Precision และ Recall ของแต่ละคลาสของโมเดลที่ใช้ Pre-Processing ด้วยวิธี Blurring (Bilateral Filtering, d=6) ..... 36
4.5 แสดงค่า Precision ของแต่ละโมเดล ..... 37
4.6 แสดงการเปรียบเทียบค่า Precision ของแต่ละโมเดล ..... 37
4.7 แสดงค่า Recall ของแต่ละโมเดล ..... 39
4.8 แสดงการเปรียบเทียบค่า Recall ของแต่ละโมเดล ..... 39
4.9 แสดงค่า Accuracy ของแต่ละโมเดล ..... 41
4.10 แสดงการเปรียบเทียบค่า Accuracy ของแต่ละโมเดล ..... 41
4.11 แสดงการเปรียบเทียบค่า Accuracy ของ Pre-Processing Level 1 ..... 44



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป

รูป หน้า

1.1 แสดงตัวอย่างการใช้ตรวจสอบสภาพจราจรแบบ Real Time ..... 1
2.1 การทำงานของ Neuron ..... 7
2.2 Neural Network ..... 8
2.3 Deep neural network ..... 9
2.4 แสดงองค์ประกอบของ Convolutional Neural Network layer ..... 10
2.5 แสดงโครงสร้างของFaster R-CNN ..... 11
2.6 แสดงการเปรียบเทียบ pre-trained model ..... 12
2.7 แสดงการปรับ Image Filtering ..... 13
2.8 แสดงการปรับ Contrast Adjustment ..... 13
2.9 แสดงการปรับ Morphological Operations ..... 14
2.10 แสดงการปรับ Deblurring ..... 14
2.11 แสดงการปรับ Neighborhood and Block Processing ..... 15
2.12 แสดงการคำนวณหาค่า Intersection over Union (IoU) ..... 16
2.13 เปรียบเทียบความถูกต้องในการจำแนกประเภทยานพาหนะด้วยวิธีที่ต่างกัน ..... 17
2.14 แสดงการเปรียบเทียบค่า mAP (Mean Average Precision) ของแต่ละโมเดล ..... 18
2.15 แสดงค่า GPU time ของแต่ละโมเดลที่ใช้ ..... 18
2.16 แสดงค่า Memory usage ของแต่ละโมเดลที่ใช้ ..... 19
2.17 แสดงจำนวน Datasets ในการ Training และ Testing ..... 19
3.1 System Diagram ..... 20
3.2 Flowchart แสดงการทำงานของโปรแกรมโดยรวม ..... 21
3.3 Flowchart แสดงการขั้นตอนการ Train Model ..... 22
3.4 Flowchart แสดงอัลกอริทึมของ Faster R-CNN ที่ใช้จำแนกประเภทรถ ..... 24
4.1 รูปเปรียบเทียบผลลัพธ์ที่ใช้ Datasets และ Input ที่มีความละเอียดต่างกัน ..... 27
4.2 แสดงAP และ mAP ของโมเดลความละเอียด 240p ..... 27
4.3 แสดงAP และ mAP ของโมเดลความละเอียด 480p ..... 28
4.4 แสดงAP และ mAP ของ Input144p ด้วยModel 480p ..... 29
4.5 แสดงAP และ mAP ของInput240p ด้วยModel 480p ..... 30



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป(ต่อ)

รูป หน้า

4.6 แสดงAP และ mAP ของ Input480p ด้วยModel 480p ..... 30
4.7 แสดงตัวอย่าง Datasets หลัง Pre-Processing ด้วยวิธี Blurring (Bilateral Filtering) (1)Original, (2)Blur6, (3)Blur9, (4)Blur12, (5)Blur15, (6)Blur18 ..... 33
4.8 แสดงตัวอย่าง Datasets หลัง Pre-Processing ด้วยวิธี Brightness Adjustments (1)Original, (2)Brightness30, (3) Brightness50, (4) Brightness70, (5) Brightness90, (6) Brightness110 ..... 34
4.9 แสดงตัวอย่าง Datasets หลัง Pre-Processing ด้วยวิธี Contrast Adjustments (1)Original, (2)Contrast1.1, (3) Contrast1.2, (4) Contrast1.3, (5) Contrast1.4, (6) Contrast1.5 ..... 35
4.10 แสดงตัวอย่าง Datasets หลัง Pre-Processing ด้วยวิธี Grayscale และ Sharpening (1) Original, (2) Grayscale, (3) Sharpening ..... 37
4.11 กราฟแสดงการเปรียบเทียบค่า Accuracy ของแต่ละโมเดล ..... 43
4.12 แสดงพื้นที่ที่โมเดลมีค่า Accuracy สูง ..... 45



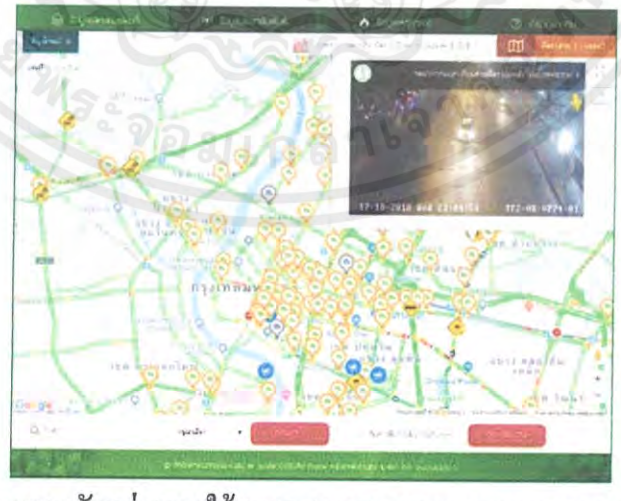
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาของปัญหา

ปัจจุบันในประเทศไทยการเดินทางและการคมนาคมขนส่งทางบกโดยรถยนต์และจักรยานยนต์เป็นหนึ่งช่องทางที่ประชาชนเลือกใช้เป็นจำนวนมาก ซึ่งเป็นช่องทางหลักที่สะดวกและรวดเร็ว จึงก่อให้เกิดการจราจรติดขัดในบางพื้นที่ โดยเฉพาะอย่างยิ่งในช่วงเวลาเร่งด่วนที่เป็นปัญหาหลักที่ต้องการได้รับการแก้ไขซึ่งเห็นได้อย่างชัดเจนในจังหวัดกรุงเทพมหานครฯ

สำนักการจราจรและขนส่ง(สจส) ได้พัฒนาแอพพลิเคชั่น BMA TRAFFIC และเว็บไซต์ www.bmatraffic.com ให้ประชาชน สามารถตรวจสอบสภาพจราจร ณ เวลาปัจจุบัน (Real-Time) จากการดึงภาพจากกล้องวงจรปิด (CCTV) กว่า 300 ตัวทั่วกรุงเทพมหานครฯ โดยรูปแบบของแอพพลิเคชั่นจะแสดงแผนที่จากGoogle Map พร้อมแสดงจุดติดตั้งกล้องวงจรปิด สามารถขยายเข้าไปดูภาพกล้องวงจรปิดแบบ Real-Time แต่ข้อมูลเหล่านี้ยังไม่เพียงพอที่ช่วยให้วิเคราะห์สภาพจราจรได้ ดังนั้นหากรู้จำนวนและประเภทของยานพาหนะบนท้องถนนได้จากกล้องวงจรปิด ก็สามารถนำข้อมูลเหล่านี้ไปใช้แก้ปัญหาการจราจรและพัฒนาระบบคมนาคมได้อย่างมีประสิทธิภาพ เช่น วางแผนการก่อสร้างถนนใหม่ วางแผนขยายถนนเพื่อรองรับปริมาณการจราจรที่มากขึ้น วิเคราะห์สาเหตุของการจราจรติดขัด การประเมินความเสี่ยงการเกิดอุบัติเหตุ ฯลฯ



**รูป 1.1 แสดงตัวอย่างการใช้ตรวจสอบสภาพจราจรแบบ Real Time** [1]

[1] https://www.bmatraffic.com

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากใช้ข้อมูลจากกล้องวงจรปิดที่มีทั้งหมด จะช่วยให้วิเคราะห์สภาพการจราจรได้อย่างมีประสิทธิภาพมากขึ้น แต่ปัญหาคือกล้องวงจรปิดแต่ละตัวจำเป็นต้องส่งภาพไปยังศูนย์ควบคุม (Server) ซึ่งมีข้อจัดเรื่องปริมาณการรับส่งของข้อมูล เพราะเป็นระบบ Real-time Server ต้องรับข้อมูลอยู่ตลอดเวลาภายใต้ อัตราการรับส่งข้อมูล (Bandwidth) ที่จำกัด หากต้องการใช้ข้อมูลจากกล้องที่มากขึ้น อัตราการรับส่งข้อมูลจะไม่เพียงพอ การลดคุณภาพของภาพจากกล้องจะสามารถช่วยแก้ปัญหาดังกล่าวได้

ระบบ AI หรือ Artificial Intelligence กำลังเป็นที่พูดถึงอย่างมาก ด้วยความฉลาดที่สามารถคิดและวิเคราะห์ รวมไปถึงการแยกแยะวิธีการจัดการในหลายๆ เรื่อง ส่งผลให้ AI ถูกจับตามองว่า จะเข้ามามีบทบาทสำคัญกับชีวิตมนุษย์ในอนาคตอันใกล้นี้ โดย AI ถูกใช้แพร่หลายในประเทศผู้นำด้านนวัตกรรม และมีการพัฒนาเพื่อเพิ่มประสิทธิภาพและความแม่นยำอย่างต่อเนื่อง โดยเฉพาะอย่างยิ่ง Machine Learning อีกแขนงหนึ่งของ AI ที่ใช้หลักการ Neural Network การประมวลผลในลักษณะคล้ายสมองของมนุษย์ กล่าวคือมีการเรียนรู้และตัดสินใจด้วยตัวเองจากการเรียนรู้ชุดข้อมูลที่มนุษย์ป้อนให้ซึ่งสามารถประยุกต์ใช้เป็นตัวช่วยในการตรวจจับและจำแนกประเภทของยานพาหนะได้เช่นกัน

ทำให้ผู้จัดทำโครงการมีความสนใจศึกษา การตรวจจับและจำแนกประเภทยานพาหนะด้วยกล้องวงจรปิด(CCTV) จากภาพวิดีโอคุณภาพต่ำ โดยนำเทคโนโลยี Machine Learning และ Image Processing มาช่วยเพิ่มประสิทธิภาพให้โมเดล (Deep learning) เพื่อเพิ่มขีดความสามารถของกล้องวงจรปิดที่ใช้เพียงบันทึกภาพ ซึ่งผู้จัดทำเลือกใช้ TensorFlow Object detection เป็น Python API ที่ช่วยให้ง่ายต่อการสร้าง ปรับปรุง และ แก้ไข โมเดล Deep Learning(CNN) จำพวก Object detection

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 วัตถุประสงค์โครงงาน

1) เพื่อศึกษาและประยุกต์ใช้งาน Deep Learning Model ในงานรูปภาพ

2) เพื่อศึกษาและประยุกต์ใช้ความรู้ด้าน Deep Learning และ Image Processing ในการจำแนกยานพาหนะบนท้องถนน

3) เพื่อศึกษา Image Processing วิธีต่างๆ ที่มีผลต่อความแม่นยำของ Convolutional Neural Network Model ในการใช้จำแนกยานพาหนะบนท้องถนน

4) เพื่อเพิ่มประสิทธิภาพความแม่นยำของ Convolutional Neural Network Model ในการใช้จำแนกยานพาหนะบนท้องถนน

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถจำแนกประเภทยานพาหนะ ในรูปภาพได้

2) ข้อมูลที่ได้สามารถนำไปต่อยอดช่วยวิเคราะห์สภาพการจราจรบนท้องถนนได้

3) เข้าใจหลักการทำงานของ Convolutional Neural Network Model

## 1.4 ขอบเขตของโครงงาน

1) ระบบที่สร้างขึ้นมาใช้เพื่อตรวจจับและจำแนกยานพาหนะจากภาพวิดีโอความละเอียดต่ำ (854x480 px) ซึ่งเป็นการประมวลผลแบบ offline

2) Input เป็นภาพสีที่ได้จากการแยกเฟรมของไฟล์วิดีโอ เฟรมเรต 29.97 เฟรมต่อวินาที

3) ไฟล์วิดีโอที่ใช้บันทึกภาพบนถนนหน้าสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังช่วงเวลากลางวัน (06.00 น. – 18.00 น.)

4) ใช้การประมวลผลและ Training model ด้วย GPU: NVIDIA GeForce GTX 1050 4GB

## 1.5 ข้อจำกัดของโปรแกรม

1) ตัวโมเดลใช้ได้กับภาพที่มาจากมุมกล้องเดียวกันกับ Datasets ที่ใช้ Train ให้กับโมเดล

2) Input ที่ใช้ไม่ได้เป็นการ Streaming จากกล้องโดยตรง แต่เป็น Input จากไฟล์วิดีโอที่ถ่ายไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.6 แผนการดำเนินงาน

| กิจกรรม | เดือน | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ส.ค. | | | | ก.ย. | | | | ต.ค. | | | | พ.ย. | | | | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |
| ค้นหาหัวข้อโครงงานที่สนใจ | | | | | | | | | | | | | | | | | | | | | | |
| เสนอหัวข้อกับอาจารย์ที่ปรึกษา | | | | | | | | | | | | | | | | | | | | | | |
| หางานวิจัยอ้างอิงที่เกี่ยวกับ Vehicle Classification และ CNN Model | | | | | | | | | | | | | | | | | | | | | | |
| ศึกษาเทคโนโลยีที่จะนำมาใช้ในโครงการ | | | | | | | | | | | | | | | | | | | | | | |
| ทดลองรัน CNN Model | | | | | | | | | | | | | | | | | | | | | | |
| การออกแบบ | | | | | | | | | | | | | | | | | | | | | | |
| เลือกและทดลอง CNN Model แต่ละแบบ | | | | | | | | | | | | | | | | | | | | | | |
| ทดลองเพื่อหาDatasetsที่เหมาะสม | | | | | | | | | | | | | | | | | | | | | | |
| ทดลองเพื่อหาคุณภาพของInputที่เหมาะสมต่อการใช้งานจริง | | | | | | | | | | | | | | | | | | | | | | |

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 1.2 แสดงแผนการดำเนินงาน(2)**

| กิจกรรม | เดือน | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ส.ค. | | | | ก.ย. | | | | ต.ค. | | | | พ.ย. | | | | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 | 1-4 |
| พัฒนา | | | | | | | | | | | | | | | | | | | | | | |
| จัดทำเอกสารตั้งแต่บทที่ 1-4 | | | | | | | | | | | | | | | | | | | | | | |
| ทดสอบสร้างและรันโมเดล Faster R-CNN บน AWS | | | | | | | | | | | | | | | | | | | | | | |
| ทำ Pre-processing กับข้อมูลที่ใช้ใน Model | | | | | | | | | | | | | | | | | | | | | | |
| เปรียบเทียบผลลัพธ์ที่ได้ก่อน-หลัง การทำ Image Pre-Processing | | | | | | | | | | | | | | | | | | | | | | |
| การปรับปรุงแก้ไข | | | | | | | | | | | | | | | | | | | | | | |
| ดูประสิทธิภาพของโมเดล จาก Image Pre-Processing แล้วเลือกมาปรับใช้กับโมเดล | | | | | | | | | | | | | | | | | | | | | | |
| จัดทำเอกสารเพื่อตีพิมพ์ รวบรวมองค์ ความรู้ทั้งหมดและสรุปผลโครงงาน | | | | | | | | | | | | | | | | | | | | | | |

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

## 2.1 Machine Learning (ML)

Machine Learning เป็นสาขาหนึ่งของปัญญาประดิษฐ์ (AI) เกี่ยวข้องกับการศึกษาและการสร้างอัลกอริทึมที่สามารถเรียนรู้ข้อมูลและทำนายข้อมูลได้ อัลกอริทึมนั้นจะทำงานโดยอาศัยโมเดลที่สร้างมาจากชุดข้อมูลตัวอย่างขาเข้าเพื่อการทำนายหรือตัดสินใจในภายหลัง แทนที่จะทำงานตามลำดับของคำสั่งโปรแกรมคอมพิวเตอร์ แบ่งออกเป็น 3ประเภท ตามลักษณะของอัลกอริทึม

1) **Supervised learning** คือ การเรียนรู้แบบได้รับคำแนะนำ เน้นสอนโดยการศึกษาจากข้อมูลตัวอย่าง มักถูกนำไปใช้แทนการทำงานแบบ rule base คือมีกฎหรือรูปแบบการทำงานที่ตายตัวหรือสามารถอธิบายเหตุผลออกมาได้อย่างชัดเจน การที่ machine learning ได้รู้ก่อนว่าผลลัพธ์ของข้อมูลแบบนี้คืออะไร แบ่งออกเป็น

- Classification เป็นการจัดหมวดหมู่ให้ข้อมูลโดยนิยมใช้กับข้อมูลที่ไม่ต่อเนื่อง (discrete)
- Regression จะเป็นการใช้วิเคราะห์ข้อมูลที่มีความต่อเนื่อง โดย supervised learning สามารถหาคำตอบได้แค่เท่าที่เราสอนไว้

2) **Unsupervised learning** เป็นการแบ่งข้อมูลเป็นกลุ่ม (clustering) แต่จะไม่สามารถระบุได้ว่าปลายทางเป็นอะไรบ้างได้แค่จัดข้อมูลแบ่งออกเป็นกลุ่มๆ เช่น การแบ่งลูกค้าออกเป็นกลุ่มย่อยๆ ตามลักษณะของลูกค้า เป็นต้น ไม่การสอนอย่างชัดเจน การเรียนรู้ของ machine learning เพื่อสร้างแบบจำลอง (model) นันมีอยู่ 3 ขั้นตอน

- Feature extraction การแปลงลักษณะของข้อมูล ที่อยู่ในรูปแบบของข้อความหรือรูปภาพให้กลายเป็นชุดตัวเลข
- Regularization พิจารณาคุณสมบัติ (feature) ของข้อมูลว่าส่งผลกระทบกับข้อมูลมากแค่ไหนเพื่อตัดหรือลดส่วนที่ไม่จำเป็นออกเพื่อทำให้ โมเดลซับซ้อนน้อยลง
- Cross-validation หลังจากได้ model มาก็นำมาทดสอบว่าโมเดลที่ได้ สามารถวิเคราะห์ข้อมูลได้อย่างแม่นยำเพียงพอหรือไม่

3) **Reinforcement learning** การเรียนรู้แบบลองผิดลองถูกเป็น machine learning ที่มีการปฏิสัมพันธ์กับระบบภายนอก โดยทุกครั้งที่เรียนรู้จะรับ input จากข้างนอกมาคำนวณด้วย inputที่นำเข้ามาคำนวณคือผลจากการลองผิดลองถูกจากครั้งก่อนหน้าซึ่งเราเรียกผลนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่า reward ซึ่งค่าreward นี้จะถูก feed กลับเข้าไปเป็น input ของการเรียนรู้รอบถัดไปและทำซ้ำไปเรื่อยๆ จนได้ผลที่ต้องการ

### 2.1.1 Artificial Neural Networks (ANN)

โครงข่ายใยประสาทเสมือน (Artificial Neuron Networks) อาศัยแนวคิดและเทคนิคจากการทำงานของระบบโครงข่ายใยประสาทในระบบประสาทของมนุษย์โดยจำลองการทำงานเหมือนกลุ่มเซลล์ประสาทที่เชื่อมโยงกันเป็นระบบประสาทที่สามารถรับรู้หลายๆสิ่งในเวลาเดียวกัน ด้วยการประมวลผลแบบขนาน (Parallel Network) ทำให้ระบบสามารถตัดสินใจได้ใกล้เคียงกับมนุษย์

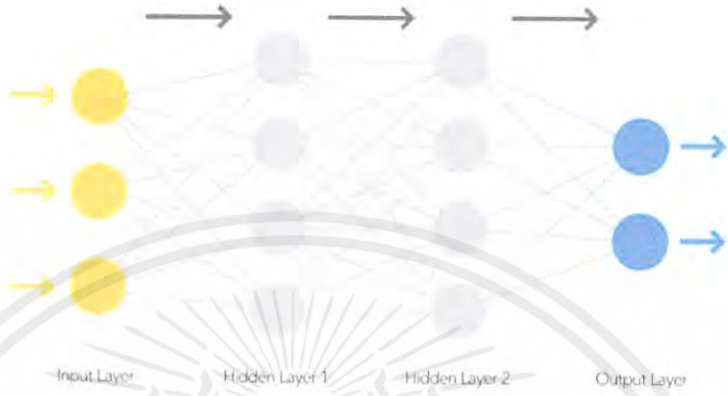
การทำงานของ Neural Networks คือเมื่อมี input เข้ามา จะนำ input มาคูณกับ weight ของแต่ละขา ผลที่ได้จาก input ทุกขาของ neuron จะนำมารวมกันแล้วมาเทียบกับ threshold ที่กำหนดไว้ ซึ่งถ้าผลรวมมีค่ามากกว่า threshold neuron จะส่ง output ออกไปยัง input ของ neuron อื่นที่เชื่อมกันใน network ถ้าค่าน้อยกว่า threshold ก็จะไม่เกิด output สิ่งสำคัญคือต้องทราบค่า weight และ threshold สำหรับสิ่งที่ต้องการเพื่อให้คอมพิวเตอร์รู้จำ ซึ่งเป็นค่าที่ไม่แน่นอน แต่สามารถกำหนดให้คอมพิวเตอร์ปรับค่าเหล่านั้นได้โดยการสอนให้คอมพิวเตอร์ pattern ของสิ่งที่ต้องการให้มันรู้จำ เรียกว่า "back propagation" ในการฝึก feed-forward Neural Networks จะมีการใช้อัลกอริทึมแบบ back-propagation เพื่อใช้ในการปรับปรุง Network Weight หลังจากใส่รูปแบบข้อมูลสำหรับฝึกให้แก่เครือข่ายในแต่ละครั้งแล้ว output จากเครือข่ายจะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งค่าความผิดพลาดก็จะถูกส่งกลับเข้าสู่เครือข่ายเพื่อใช้แก้ไขค่า weight ต่อไป

**รูป 2.1 การทำงานของ Neuron**[2]

[2] https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Feed-Forward Neural Network จะแบ่ง Neuron ออกเป็นกลุ่มๆ โดยแต่ละกลุ่มจะเรียกเป็น Layer โดยข้อมูลที่เข้ามาจะไหลไปในทิศทางเดียวไม่ไหลย้อนกลับจากLayer นึงสู่อีก Layer นึง




**รูป 2.2 Neural Network**[3]

**ส่วนประกอบของ Neural Network**

**Input Layer** ชั้นนี้จะเป็นข้อมูล input จำนวนของโหนด ขึ้นอยู่กับจำนวนของ input ว่ามีข้อมูลอะไรบ้างที่นำเข้ามาคิดในโมเดล (ปกติแล้วจะเรียกปัจจัยที่นำมาวิเคราะห์ว่า feature)

**Hidden Layer** เป็นชั้นที่อยู่ระหว่าง Input layer และ Output layer ซึ่งมีผลอย่างมากต่อประสิทธิภาพในการเรียนรู้ของโมเดล ซึ่ง hidden layer จะมีกี่ชั้นก็ได้ แล้วแต่ละชั้นจะมีจำนวนของ Neuron เท่าไหร่ก็ได้เช่นกัน ในการเพิ่มจำนวนชั้นและจำนวน neuron จะส่งผลต่อการทำงานของโมเดล

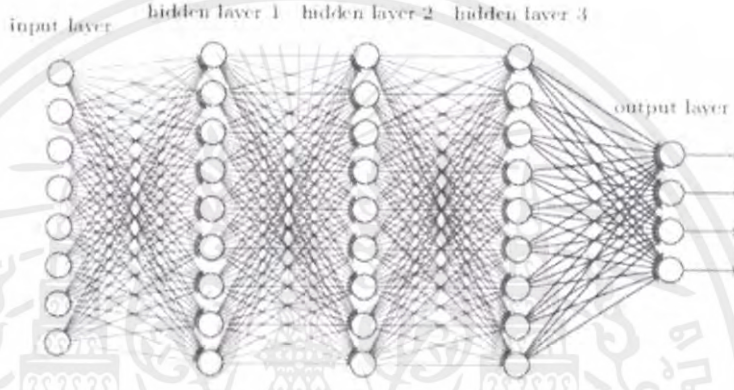
**Output Layer** ชั้นที่จะนำเอาข้อมูลจากการคำนวณไปใช้ จำนวนของโหนดในชั้นนี้นั้นจะขึ้นอยู่กับรูปแบบของ output ที่จะนำไปใช้

[3] https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2 Deep Learning

Deep Leaning Deep Leaning คือ Artificial Neuron Networks แต่ความแตกต่างระหว่าง Deep Learning กับ ANN คือระดับ hidden layer ที่ใน Deep Learning มี hidden layer มากกว่าใน ANN โดยหลักการของ Deep Learning จะเป็น ANN ที่มีโหนดหลายๆชั้น และจะใช้การประมวลผลแบบขนาน (Parallel Processing) ทำให้สามารถประมวลผลได้ครั้งละจำนวนมาก ช่วยให้การเรียนรู้ของเครื่องสามารถให้ผลลัพธ์ในการตัดสินใจและคาดการณ์ได้ดีมากยิ่งขึ้น




**รูป 2.3 Deep neural network**[4]

### 2.1.3 Convolutional Neural Network (CNN)

Convolutional Neural Network คือ Artificial Neural Network (ANN) ประเภทหนึ่งที่ใช้ Convolutional layers เพื่อกรองข้อมูลที่เป็นประโยชน์จากอินพุต การทำงานนั้นยังรวมถึงการรวมข้อมูลอินพุต (feature map) เข้ากับ Convolution kernel (filter) เพื่อสร้าง feature map ขึ้นมา โดยFilter ใน convolutional layers (conv layers) จะได้รับการปรับเปลี่ยนตามพารามิเตอร์เพื่อดึงข้อมูลที่เป็นประโยชน์ที่สุด Convolutional networks จะปรับโดยอัตโนมัติเพื่อที่จะหาคุณลักษณะที่ดีที่สุด

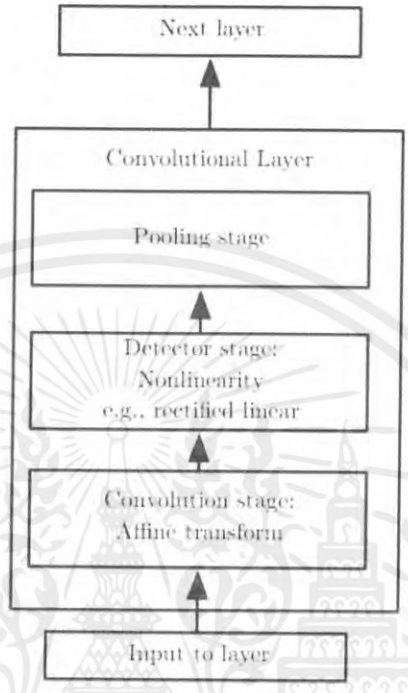
มีการประยุกต์ใช้งาน Convolutional Neural Network ในด้านต่างๆมากมาย เช่น ด้านรูปภาพ (image recognition, image classification, video labeling, text analysis) หรือทางด้านเสียงพูด (speech recognition, natural language processing, text classification) รวมไปถึงระบบ AI ที่ล้ำยุค เช่น หุ่นยนต์, ผู้ช่วยเสมือน และ รถยนต์ไร้คนขับ

---

[4] https://www.kdnuggets.com/2017/05/deep-learning-big-deal.html

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## องค์ประกอบของ Convolutional Neural Network




**รูป 2.4 แสดงองค์ประกอบของ Convolutional Neural Network layer**[5]

จากรูป จะเห็นได้ว่าใน 1 layer ประกอบไปด้วย 3 ส่วนใหญ่ๆ คือ

**1. Convolution stage**

ในขั้นตอนนี้จะสร้าง Sliding window(Filter) มาสแกนรูป input เพื่อทำ Feature map (ทำการสแกนรูปเพื่อแยกองค์ประกอบของรูป ออกมา เช่น ขอบ สี รูปทรง เป็นต้น)

**2. Detector stage**

ชั้นนี้จะทำหน้าที่รับ output จาก ข้อ1 แปลงให้อยู่ในรูปของ nonlinear โดยใช้ activation เช่น Rectified Linear Units (ReLU)

---

[5] https://medium.com/@thebear19/neural-network-101-cnn-with-tensorflow-fd5d515e979b

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

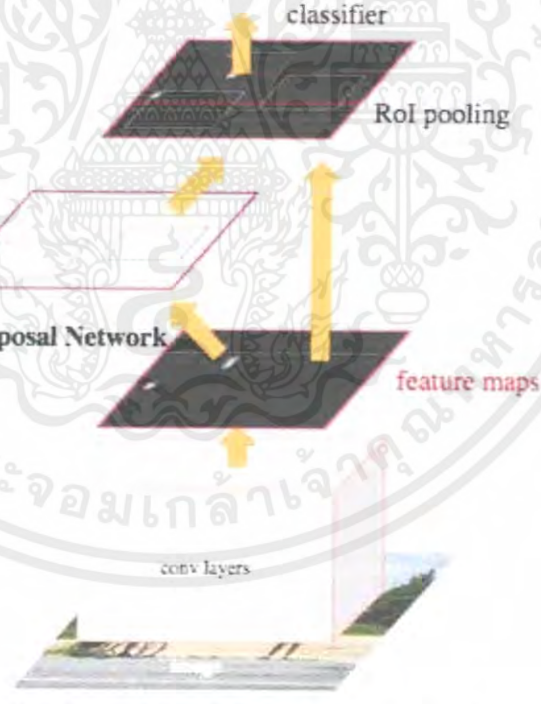
**3. Pooling stage**

Pooling ทำหน้าที่ resize ข้อมูลให้ขนาดเล็กลงโดยที่รายละเอียดของ input ยังครบถ้วนเหมือนเดิม หลักการทำงานขั้นตอนนี้คล้ายกับ 1. แต่ต่างกันที่ output ที่ได้จะมีขนาดเล็กลง โดย Pooling มีประโยชน์ในเรื่องเพิ่มความไวในการคำนวณ และแก้ปัญหา overfitting

**2.1.4 Faster R-CNN**

Faster R-CNN คือการนำ Selective Search มารวมไว้ใน Neural Network เดียวกัน ซึ่งมีส่วนประกอบหลัก 3 ส่วนคือ

1) ส่วนฐานที่ทำหน้าที่สกัด feature
2) ส่วน Region Proposal Network (RPN) ทำหน้าที่สกัดบริเวณที่น่าจะเป็นวัตถุจาก feature map
3) ส่วนจำแนกประเภท ที่นำ feature map และ region ที่ได้จาก RPN มาประมวลผลโดยทำ ROI pooling เพื่อตอบว่าบริเวณใดของภาพมีวัตถุใดอยู่




**รูป 2.5 แสดงโครงสร้างของFaster R-CNN[6]**

[6] https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 TensorFlow

TensorFlow เป็น open source library ที่พัฒนาโดย google สำหรับอำนวยความสะดวกในการใช้งาน machine learning เช่น KERAS API สำหรับการ เทรนและการสร้างโมเดล, Eager execution สำหรับวิเคราะห์และdebug model ฯลฯ โดยสามารถเลือกประมวลผลได้ในหลายๆplatform (CPUs, GPUs, TPUs) ช่วยให้การทำงานด้าน machine learning และ deep learning เป็นไปได้ง่ายและรวดเร็วยิ่งขึ้น

### TensorFlow Object detection API

TensorFlow Object detection API เป็น framework ที่สร้างบน TensorFlow อีกทีหนึ่ง ถูกพัฒนาเพื่อให้คนทั่วไปสร้างโมเดล เทรนโมเดล และใช้งานโมเดลในด้านการตรวจจับวัตถุ ได้สะดวกและง่าย โดยมีการรวบรวม pre-trained model จาก datasets (COCO, Kitti, Open Images, AVA, iNaturalist Species) พร้อมแสดงการเปรียบเทียบ Speed และ mAP (Mean Average Precision) ของแต่ละโมเดล เพื่อให้นักพัฒนาและคนทั่วไป สามารถเลือกใช้โมเดลที่เหมาะสมกับงานตัวเอง

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 |

Open Images trained models

| Model name | Speed (ms) | Open Images mAP@0.5[^2] | Outputs |
|---|---|---|---|
| [illegible] | 727 | 37 | Boxes |
| [illegible] | 347 | | Boxes |

iNaturalist Species trained models

| Model name | Speed (ms) | Pascal mAP@0.5 | Outputs |
|---|---|---|---|
| [illegible] | 366 | 58 | Boxes |
| [illegible] | 366 | 55 | Boxes |

Kitti trained models

| Model name | Speed (ms) | Pascal mAP@0.5 | Outputs |
|---|---|---|---|
| [illegible] | 79 | 87 | Boxes |

AVA v2.1 trained models

| Model name | Speed (ms) | Pascal mAP@0.5 | Outputs |
|---|---|---|---|
| [illegible] | 93 | 11 | Boxes |

**รูป 2.6 แสดงการเปรียบเทียบ pre-trained model (COCO, Kitti, Open Images, AVA, iNaturalist)**[7]

โดยผู้จัดทำได้ใช้ TensorFlow Object detection API ในการช่วยสร้างโมเดล โดยใช้การ Configure CNNs จาก Pre-trained Model ของ Faster R-CNN Resnet-101 มาปรับใช้และสร้างโมเดลสำหรับตรวจจับและจำแนกยานพาหนะขึ้นมา

---

[7] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

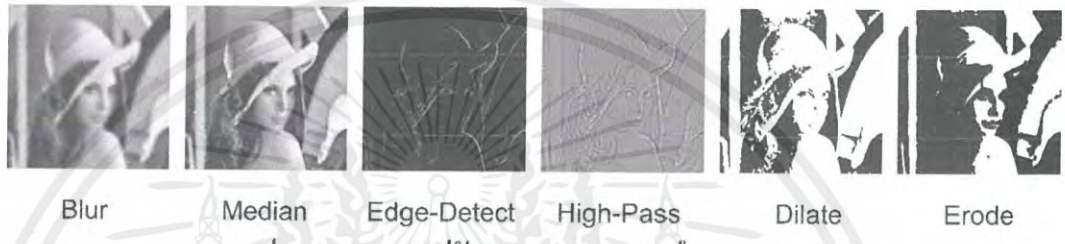
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 Image Enhancement

คือ กระบวนการปรับปรุงภาพให้มีคุณภาพที่ดีขึ้น โดยไม่มีการเปลี่ยนแปลงเนื้อภาพไปจากเดิม มีหลากหลายวิธีเช่น การลดนอยซ์ของภาพ การปรับภาพให้คมขึ้น การปรับค่าความเข้มแสง ฯลฯ

### 1) Image Filtering

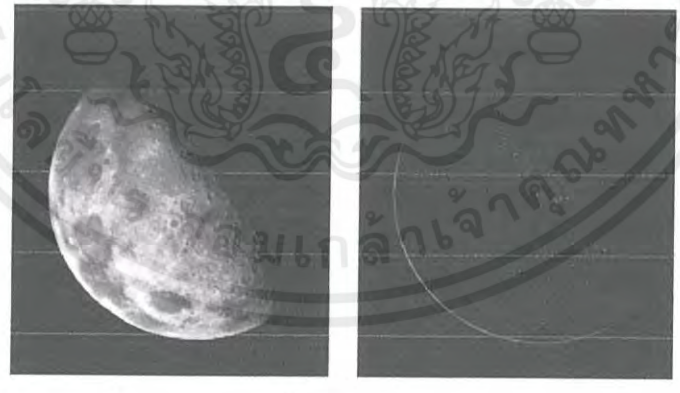
เป็นการนำภาพผ่านตัวกรองสัญญาณ เพื่อกรองข้อมูลที่สนใจหรือลดคุณสมบัติบางอย่างของภาพ เพื่อให้ภาพมีคุณสมบัติตามที่เราต้องการ




**รูป 2.7 แสดงการปรับ Image Filtering**[8]

### 2) Sharpening Spatial Filter

วัตถุประสงค์หลักของ sharpening คือต้องการเพิ่มรายละเอียดหรือปรับปรุงรายละเอียดของภาพ ดำเนินการได้โดย Differentiation ใน Spatial Domain



**รูป 2.8 แสดงการปรับ Contrast Adjustment**[9]

---

[8] https://www.cs.ubc.ca/~lsigal/425_2018/Lecture4.pdf

[9] http://staff.cs.psu.ac.th/sathit/DigitalImage/Sharpening%20Spatial%20Filters.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

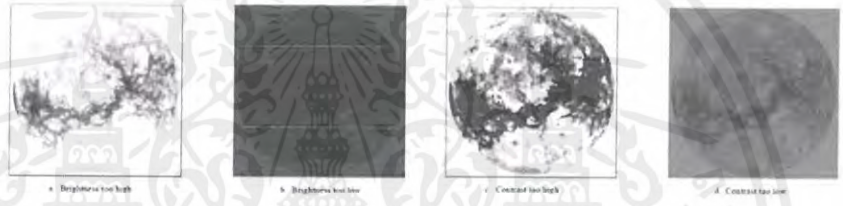
**3) Brightness and Contrast Adjustments**

Brightness ใช้ในการปรับค่าความสว่างของภาพ การปรับค่าความสว่างของภาพจะมีประสิทธิภาพในกรณีที่โทนของภาพทั้งภาพนั้นออกมาในแนวที่มืดหรือสว่างจนเกินไป และ Contrast คือระดับความแตกต่างระหว่างส่วนที่สว่างกับส่วนที่มืดของตัวภาพการเพิ่มค่า Contrast นั้นจะส่งผลให้ความแตกต่างดังกล่าวข้างต้นนั้นชัดเจนยิ่งขึ้นซึ่งทำให้ภาพนั้นมีความคมชัดมากขึ้นด้วย

สมการของค่า Brightness และค่า Contrast เป็นดังสมการข้างล่าง

$$g(x) = \alpha f(x) + \beta$$

$\alpha$ คือค่าของ Contrast และ $\beta$ คือค่าของ Brightness ใช้ในการควบคุมความคมชัดและความสว่าง



**รูป 2.9 แสดงการปรับ Brightness and Contrast**[10]

**4) Bilateral Filtering**

หลักการของ bilateral filtering นั้นจะคำนวณ pixel โดยถ่วงน้ำหนักจากทั้ง space (ระยะห่างยิ่งห่างมากยิ่งมีน้ำหนักน้อย) และ range(ความแตกต่างของสียิ่งแตกต่างมากยิ่งมีน้ำหนักน้อย) การถ่วงน้ำหนักที่ใช้ จะใช้ Guassian distribution ทั้ง space และ range

**รูป 2.10 แสดงการปรับ Deblurring**[11]

---

[10] https://hcig1.wordpress.com/2017/02/16/brightness-and-contrast-adjustments-with-opencv/

[11] https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.4 Mean Average Precision (mAP)**

mAP เป็นเครื่องมือที่ใช้ตรวจสอบความแม่นยำของNeuron Network Model นิยมใช้กับการตรวจสอบแบบMulticlass(หลายๆคลาส) โดยใช้การตรวจสอบจากผลลัพธ์ของการทาย (Predict) กับความจริง (Ground truth) โดยศัพท์ที่ควรรู้สำหรับการตรวจสอบดังกล่าวคือ

**True Positive (TP)** คือสิ่งที่ทายว่ามันมี**จริง** และ เฉลยบอกว่ามี**จริง**

**True Negative (TN)** คือสิ่งที่ทายว่ามัน**ไม่มีจริง** และเฉลยก็บอกว่ามัน**ไม่มีจริง**

**False Positive (FP)** คือสิ่งที่ทายว่ามัน**มีจริง** แต่เฉลยก็บอกว่ามัน**ไม่จริง(มันไม่ถูก)**

**False Negative (FN)** คือสิ่งที่ทายว่ามัน**ไม่มีจริง** แต่เฉลยก็บอกว่ามัน**มีจริง (มันไม่ถูก)**

**Precision** คืออัตราการทายถูกต่อ การทายทั้งหมดเขียนเป็นสมการได้ดังนี้

$$\text{precision} = \frac{tp}{tp + fp} = \frac{tp}{n} \tag{2.1}$$

**Recall** คือจำนวนที่ทายถูกต่อจำนวนของความจริงทั้งหมดเขียนเป็นสมการได้ดังนี้

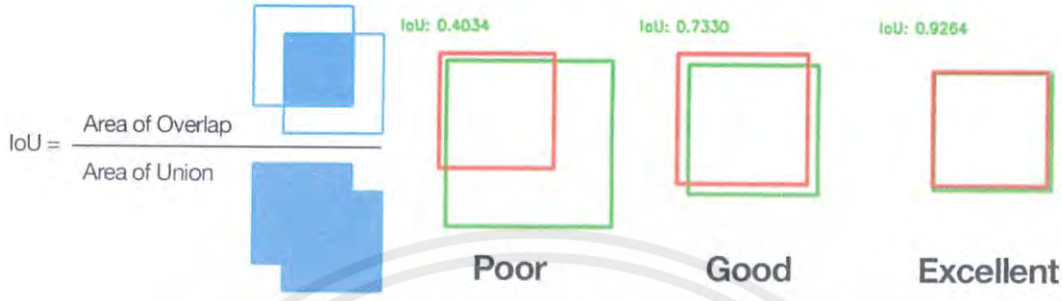
$$\text{recall} = \frac{tp}{tp + fn} \tag{2.2}$$

**Accuracy** คือความแม่นยำของระบบทั้งหมด เขียนเป็นสมการได้ดังนี้

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Intersection over Union (IoU)** คือตัวเปรียบเทียบตำแหน่งการ Box ของ ground truth กับ predict ในโมเดลที่ต้องการผลลัพธ์เป็นการแยกประเภทพร้อมระบุตำแหน่ง เพื่อเป็นตัวเช็คความถูกต้องเบื้องต้นจากการกำหนด Threshold ก่อนใช้ข้อมูลดังกล่าวคำนวณหาค่า AP




**รูป 2.12 แสดงการคำนวณหาค่า Intersection over Union (IoU)**[12]

**Average Precision (AP)** คือ ตัววัดประสิทธิภาพของคลาสจากการคำนวณพื้นที่ใต้กราฟจาก Precision และ Recall

$$\sum_{k=1}^{N} P(k)\Delta r(k) \tag{2.4}$$

โดย mAP เกิดจากการนำ AP ของทุกๆคลาสมาหาค่าเฉลี่ยได้ผลลัพธ์เป็นตัวแทนวัดความถูกต้องให้กับโมเดลทั้งหมด

[12] https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 งานวิจัยที่เกี่ยวข้อง

ในโครงงานนี้เลือกใช้ Artificial Neural Network (ANN) จำพวก Convolutional Neural Network (CNN) ในการทำ Feature extraction สำหรับการจำแนกประเภทของรถ อ้างอิงมาจากงานวิจัย Traffic video surveillance: Vehicle detection and classification โดยงานวิจัยนี้เป็นการตรวจจับแยกประเภทและนับจำนวนยานพาหนะจากกล้อง CCTV ด้วยการเปรียบเทียบ 3 วิธีคือ ANN, SVM และ k-NN ซึ่งพิสูจน์ว่า ANNให้ผลลัพธ์ความแม่นยำสูงที่สุดและใช้เวลาในการประมวลผลน้อยที่สุด

| Classification Method | ANN and HOG | SVM and PCA [20] | KNN and PCA [21] |
|---|---|---|---|
| Total No. of vehicles | 150 | 150 | 150 |
| Correct Classification | 134 | 119 | 128 |
| Classification Accuracy(%) | **89.3** | **79.8** | **85.3** |

| Classification Method | Frames per second | Computational time (second) |
|---|---|---|
| ANN | 65 | 0.015 |
| SVM | 42 | 0.023 |
| k-NN | 28 | 0.03 |

**รูป 2.13 เปรียบเทียบความถูกต้องในการจำแนกประเภทยานพาหนะด้วยวิธีที่ต่างกัน**

และในโครงงานนี้ได้เลือกใช้ Faster R-CNN Resnet-101 เป็นโมเดล อ้างอิงจากงานวิจัย Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors โดยงานวิจัยนี้ได้นำโมเดล 3 ตัว ซึ่งคือ Faster RCNN, R-FCN และ SSD มาเปรียบเทียบกันด้วยInputที่มีความละเอียด300p ในด้านความถูกต้อง ความเร็ว ทรัพยากรที่ใช้ ซึ่งใช้ COCO dataset ในการเทรนและทดสอบ และได้พิสูจน์ว่า Faster R-CNN Resnet-101 เป็นโมเดลที่เร็ว ไม่ใช้ทรัพยากรมากจนเกินไป และมีความแม่นยำสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

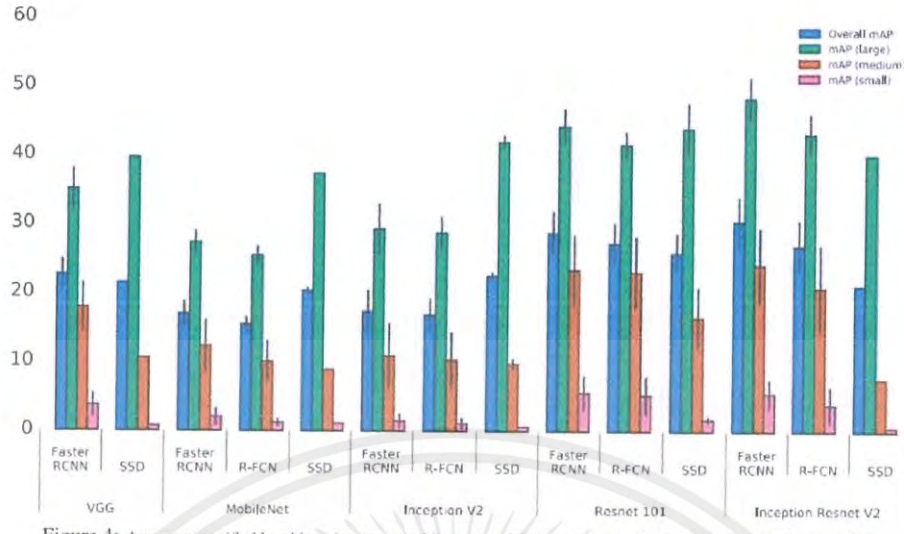



Figure 4: Accuracy stratified by object size, meta-architecture and feature extractor. We fix the image resolution to 300.

**รูป 2.14 แสดงการเปรียบเทียบค่า mAP (Mean Average Precision) ของแต่ละโมเดล**

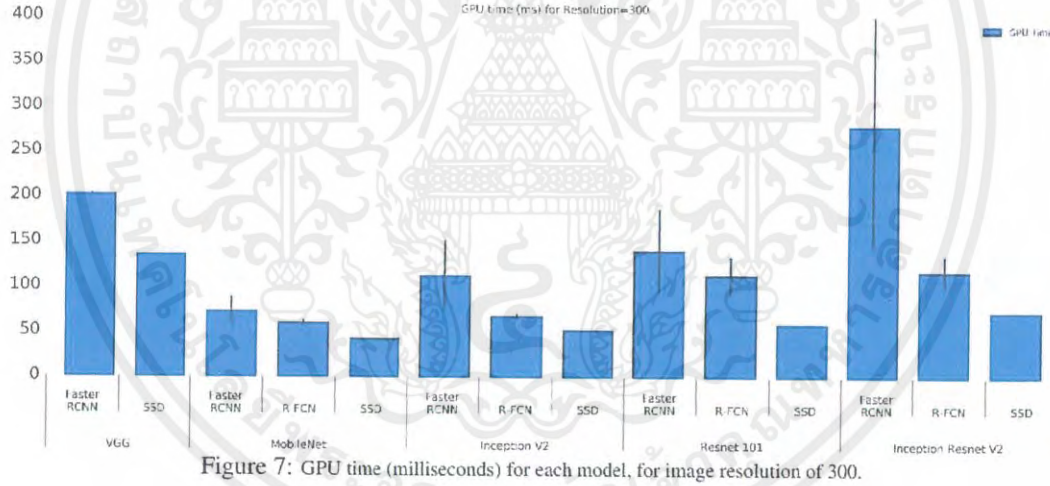



Figure 7: GPU time (milliseconds) for each model, for image resolution of 300.

**รูป 2.15 แสดงค่า GPU time ของแต่ละโมเดลที่ใช้**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

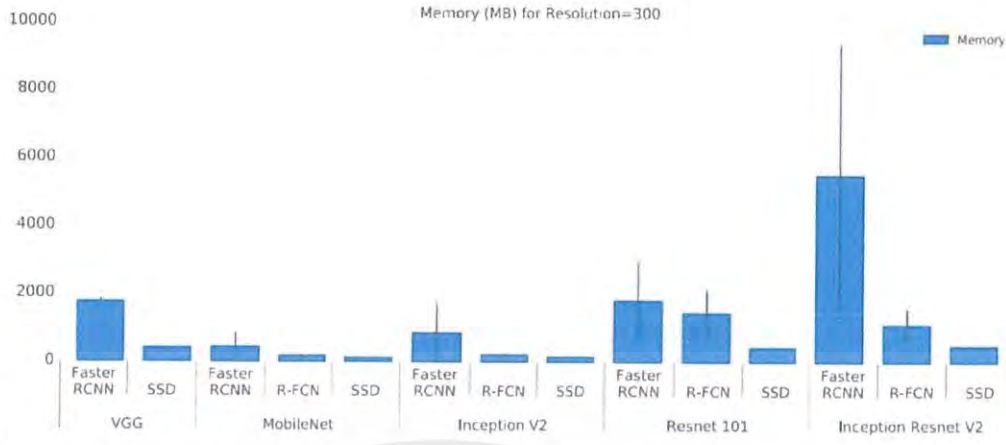



Figure 9: Memory (Mb) usage for each model. Note that we measure total memory usage rather than peak memory usage. Moreover, we include all data points corresponding to the low-resolution models here. The error bars reflect variance in memory usage by using different numbers of proposals for the Faster R-CNN and R-FCN models (which leads to the seemingly considerable variance in the Faster-RCNN with Inception Resnet bar).

**รูป 2.16 แสดงค่า Memory usage ของแต่ละโมเดลที่ใช้**

และในโครงงานนี้ได้เลือกใช้ จำนวน Datasets ที่ใช้ในการเทรนโมเดลจำนวนประมาณ 4000 ตัวอย่าง อ้างอิงจากงานวิจัย Weakly-supervised Vehicle Detection and Classification by Convolutional Neural Network โดยงานวิจัยนี้ได้นำได้มีการจำแนกประเภทยานพาหนะเป็น 5 ประเภท คือ Sedan, Van, Bus, Truck และ Bike โดยใช้จำนวน Datasets ในการเทรนโมเดลจำนวน 4,097 ตัวอย่าง และได้ค่าประสิทธิภาพในการจำแนกยานพาหนะ mAP = 98.79% ค่าความแม่นยำ 98.28% และค่าประสิทธิภาพในการตรวจจับยานพาหนะ mAP = 85.26%

Table 2. Vehicle dataset from traffic surveillance

| | Total | Sedan | Van | Bus | Truck | Bike |
|---|---|---|---|---|---|---|
| Training | 4097 | 943 | 868 | 856 | 787 | 643 |
| Testing | 2051 | 426 | 412 | 462 | 421 | 330 |
| Total | 6142 | 1369 | 1280 | 1318 | 1280 | 973 |

**รูป 2.17 แสดงจำนวน Datasets ในการ Training และ Testing**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในโครงงานนี้ได้เลือกใช้ วิธีการ Image Processing 4 วิธี คือ 1.Image Blurring (Bilateral Filtering) 2.Grayscale 3.Sharpening 4.Brightness and Contrast Adjustments อ้างอิงจากงานวิจัย Road Damage Detection and Classification with Faster R-CNN เป็นงานวิจัยที่เกี่ยวกับตรวจจับและแยกประเภทรอยแตกของพื้นถนน โดยใช้ Faster R-CNN และได้มีการใช้วิธี Image Processing ต่างๆ มาใช้กับ Datasets ก่อนที่จะนำไปเทรนให้โมเดล ผลปรากฏว่า ค่า Mean F1-Score ของโมเดลมีค่ามากขึ้น

TABLE I
COMPARISON OF RESULTS FOR DIFFERENT METHODS ON THE TEST DATA.

| $N_{feat}$ | ResNet-101 | ResNet-152 | | | |
|---|---|---|---|---|---|
| Faster R-CNN | ✓ | ✓ | ✓ | ✓ | ✓ |
| parameter adjustment | | | ✓ | ✓ | ✓ |
| data augmentation | | | | ✓ | ✓ |
| fine-tuning | | | | | ✓ |
| Mean F1-Score | 0.6099 | 0.6174 | 0.6218 | 0.6242 | **0.6255** |

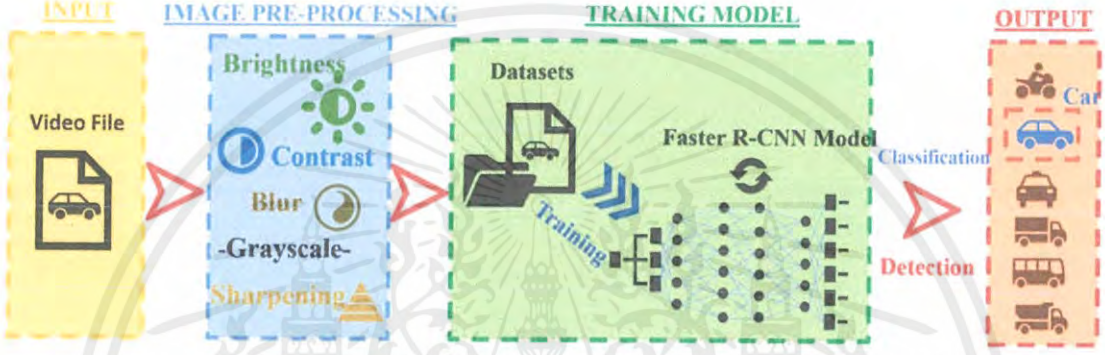
**รูป 2.18 แสดงค่า Mean F1-Score ที่ได้ในแต่ละวิธีของงานวิจัย**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3

# การออกแบบและพัฒนา

## 3.1 ภาพรวมการทำงานภายในระบบ




รูป 3.1 System Diagram

จาก Diagram สามารถแบ่งระบบเป็น 4 ส่วนหลัก ได้แก่ 1.Input 2.Image Pre-Processing 3.Training Model 4.Output

1) Input เป็นไฟล์วิดีโอ 3 ช่วงเวลา (7.30-8.00 น. ,14.00-14.30 น. และ 16.00.-17.00 น.)

2) Image Pre-Processing ใช้ 5 วิธี คือ 1. Brightness Adjustments 2. Contrast Adjustments 3. Image Blurring (Bilateral Filtering) 4. Grayscale และ 5. Sharpening

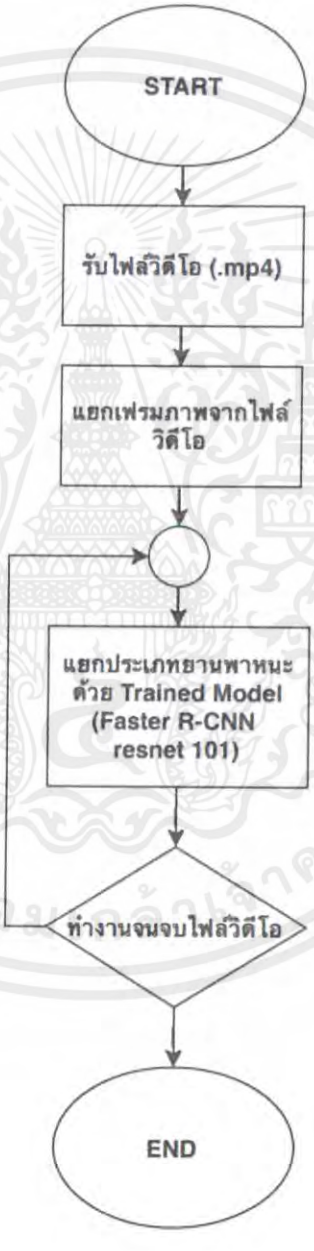
3) Training Model นำ Dataset ที่ Image Pre-Processing แล้ว มาแบ่งออกเป็น 2 ส่วนคือ Train Datasets และ Test Datasets โดย Train Datasets จะนำไปสร้าง Faster R-CNN และ Test Datasets จะนำไป Evaluate Model หลังจากนั้นนำไป Detection และ Classification เพื่อตรวจหาและแยกประเภทยานพาหนะ

4) Output เป็นผลลัพธ์ที่ได้จากการ Detection และ Classification โดยจะได้ตำแหน่งของยานพาหนะและประเภทของยานพาหนะ (รถจักรยานยนต์, รถยนต์, รถแท็กซี่, รถบรรทุก, รถสองแถว และ รถตู้)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 การออกแบบขั้นตอนการทำงานของระบบ

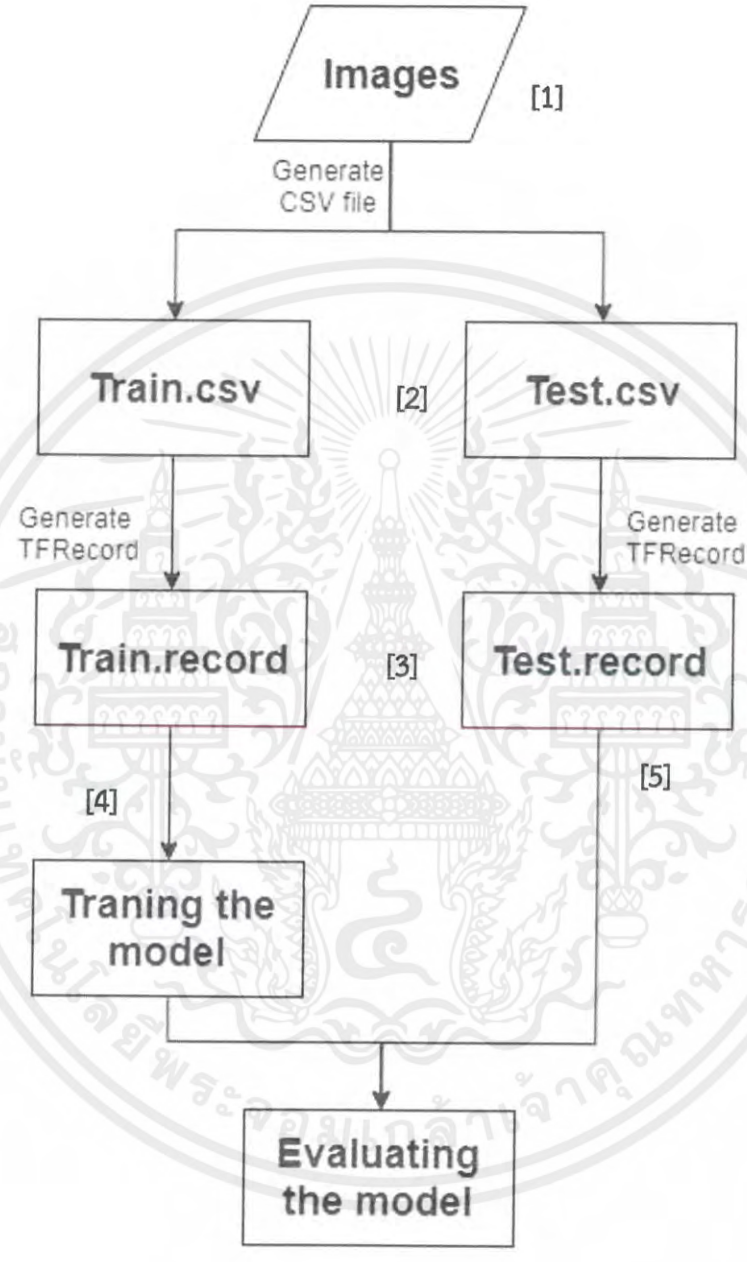
นำไฟล์วิดีโอ แยกเป็นเฟรมจากนั้นส่งไปทำ Image Pre-Processing เพื่อปรับคุณภาพของภาพให้ดีขึ้น แล้วส่งเข้าโมเดล Faster R-CNN ที่สร้างไว้ จะได้เฟรมรูปภาพที่มี Box Detection พร้อมแสดงประเภทของยานพาหนะที่ตรวจจับได้ ซึ่งมีขั้นตอนการทำงานดังนี้




**รูป 3.2 Flowchart แสดงการทำงานของโปรแกรมโดยรวม**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2.1) ขั้นตอนการ Train Model Fast R-CNN**




**รูป 3.3 Flowchart แสดงการขั้นตอนการ Train Model**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เลือกภาพที่ใช้เป็น Datasets ให้กับโมเดลโดยการกำหนด Box และ Label คลาสให้กับยานพาหนะที่ตรวจจับได้บนภาพ โดยเก็บไว้ในรูปแบบ csv ที่มีHeader คือ filename, width, height, class, xmin, ymin, xmax และ ymax

- filename: ชื่อไฟล์ของภาพจากการ Box และ Label สำหรับอ้างอิงข้อมูลภาพ
- width: ความกว้างของภาพ (pixels)
- height: ความสูงของภาพ (pixels)
- class: ประเภทของยานพาหนะ (Car, Motorcycle, Taxi, Pick-up Truck, Van และ Truck)
- xmin, ymin, xmax, ymax: พิกัดที่ Box ยานพาหนะอ้างอิงจากภาพ

2) แบ่ง Datasets เป็น2กลุ่มคือ Test และ Train โดยกลุ่มTrain เป็นข้อมูลเพื่อให้โมเดลเรียนรู้และ กลุ่มTest เป็นตัวทดสอบความถูกต้อง โดยแบ่งสัดส่วน Train:Test เป็น 80:20 ด้วยวิธีการ Split Test

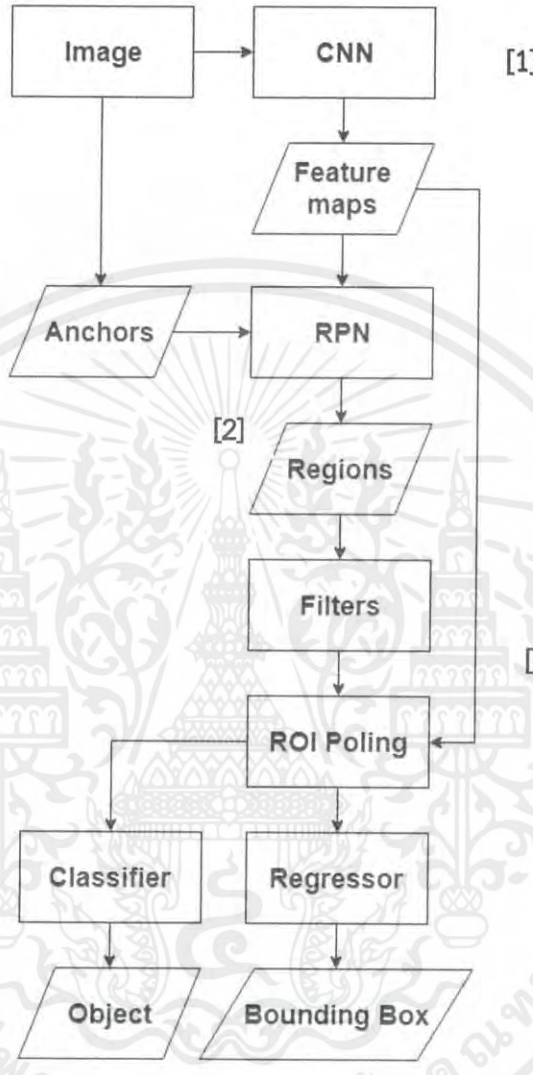
3) กลุ่มข้อมูล Train และ Test จะถูกแปลงให้อยู่ในรูปแบบ tfrecord เป็นรูปแบบสำหรับรวมไฟล์หลายๆข้อมูลจาก csv และ Images เพื่อให้ง่ายต่อการอ่านโดย TensorFlow Model

4) Train ข้อมูลให้โมเดลด้วย Train.record

5) ประเมินผลความถูกต้องของข้อมูลด้วย Test.record เรื่อยๆจนได้โมเดลที่พอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2) การทำงานของ Faster R-CNN ที่ใช้จำแนกประเภทรถ




**รูป 3.4 Flowchart แสดงอัลกอริทึมของ Faster R-CNN ที่ใช้จำแนกประเภทรถ**

1) Convolutional Neural Network (CNN) ทำหน้าที่สกัด feature สร้างเป็น feature maps ส่งไปยัง ROI Poling และ RPN
2) Region Proposal Network (RPN) ทำหน้าที่สกัดเอาบริเวณที่น่าจะเป็นวัตถุจาก feature maps
3) จำแนกประเภทโดยนำ feature maps และ region ที่ได้มาประมวลผลโดยทำ ROI pooling เพื่อตอบว่าบริเวณใดของภาพมีวัตถุใดอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

# การทดลอง

หลังจากออกแบบการทำงานภายในระบบแล้ว การทดลองในบทนี้จะเป็นการทดลองเกี่ยวกับ Datasets ที่นำไปใช้ในโมเดล Faster R-CNN Resnet-101 แบ่งออกเป็น

1) การทดลองเกี่ยวกับการเลือกใช้ Datasets

2) การทดลองเกี่ยวกับการเลือกใช้ความละเอียดของ Input

3) การทดลองเกี่ยวกับการเลือกใช้ Image Pre-Processing

โดยใช้ Configuration จาก "Faster R-CNN with Resnet-101 (v1) configured for the Oxford-IIIT Pet Dataset"

## 4.1 การทดลองเกี่ยวกับการเลือกใช้ Datasets

### 4.1.1 จุดประสงค์

เพื่อเลือกความละเอียดของภาพ Datasets ที่เหมาะสมสำหรับใช้เทรนให้โมเดล Faster R-CNN

### 4.1.2 วิธีการทดลอง

1) เลือกใช้ datasets ในการเทรนให้โมเดลที่มีความละเอียด 854x480 (480p) เปรียบเทียบกับ Datasets ที่มีความละเอียด 426x240 (240p) โดยใช้จำนวน Datasets เท่ากัน (2358 index) แบ่งเป็น 6 คลาส (Car, Motorcycle, Taxi, Pick-up Truck, Van และ Truck) เทรนด้วย Model Faster R-CNN และจำนวนรอบในการเทรนเหมือนกัน (976 steps)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก) ข)

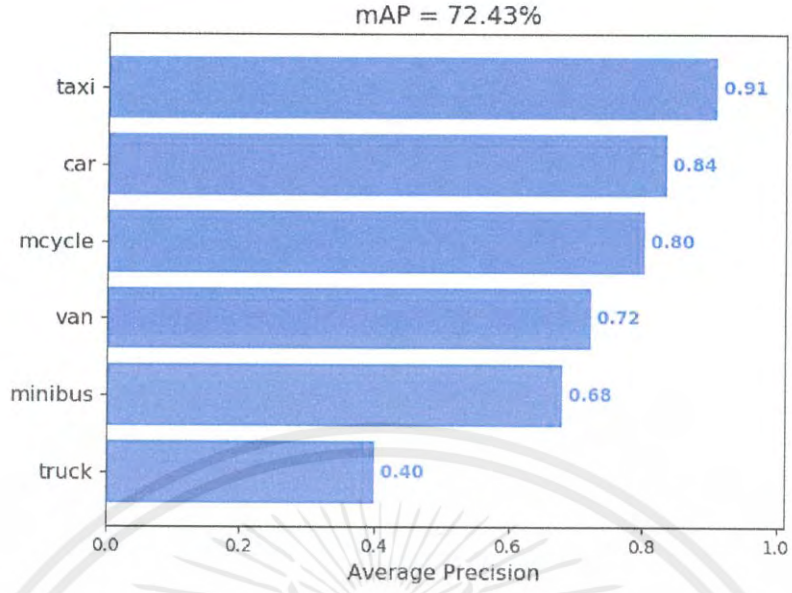
**รูป 4.1 รูปเปรียบเทียบผลลัพธ์ที่ใช้ Datasets และ Input ที่มีความละเอียดต่างกัน**

**ก) ใช้ Datasets ในการเทรนที่มีความละเอียด 240p และรับ Input ความละเอียด 240p**

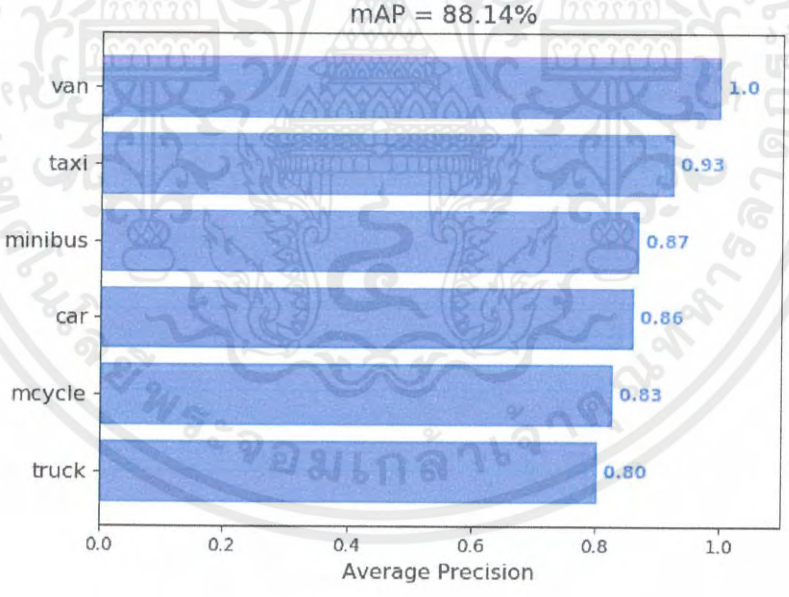
**ข) ใช้ Datasets ในการเทรนที่มีความละเอียด 480p และรับ Input ความละเอียด 480p**

2) หาค่า mAP (Mean Average Precision) ของทั้งสองโมเดลจากการใช้ชุดข้อมูลทดสอบเดียวกัน(ภาพเดียวกันแต่ความละเอียดของภาพต่างกันตามโมเดลที่ใช้ทดสอบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูป 4.2 แสดงAP และ mAP ของโมเดลความละเอียด 240p**




**รูป 4.3 แสดงAP และ mAP ของโมเดลความละเอียด 480p**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เปรียบเทียบค่า mAP ของทั้งสองโมเดล(240p,480p)

**ตาราง 4.1 แสดงตารางการเปรียบเทียบ mAP ระหว่างโมเดล 240p และ 480p**

| MODEL | AP | | | | | | mAP |
|---|---|---|---|---|---|---|---|
| | Car | Mcycle | Taxi | Pick-up Truck | Van | Truck | |
| 240p | 83.51% | 80.15% | 90.81% | 67.96% | 72.13% | 40.00% | 72.43% |
| 480p | 86.10% | 82.81% | 92.59% | 86.94% | 100.00% | 80.40% | **88.14%** |

จากภาพ4.2 โมเดล240p mAP = 72.43%

จากภาพ4.3 โมเดล480p mAP = 88.14%

จึงสรุปได้ว่าคุณภาพของ datasets มีผลต่อการจำแนกคลาสของโมเดล ยิ่งข้อมูลเทรนมีคุณภาพสูง ยิ่งทำให้Model มีประสิทธิภาพในการจำแนกคลาสได้ดี

## 4.2 การทดลองเกี่ยวกับการเลือกใช้ความละเอียดของ Input

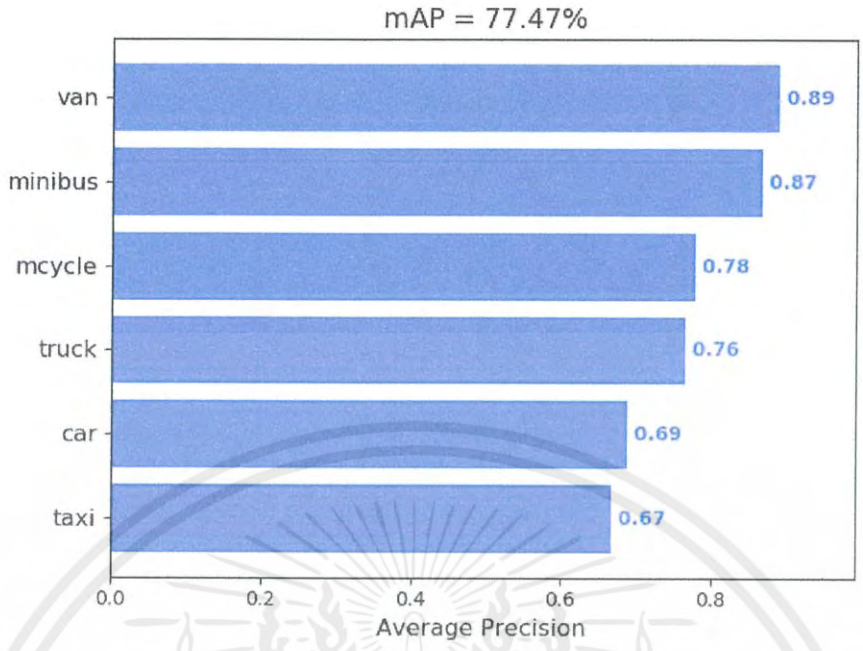
### 4.2.1 จุดประสงค์

เพื่อเปรียบเทียบและเลือกใช้ความละเอียด Input ที่เหมาะสมสำหรับนำใช้งานจริง

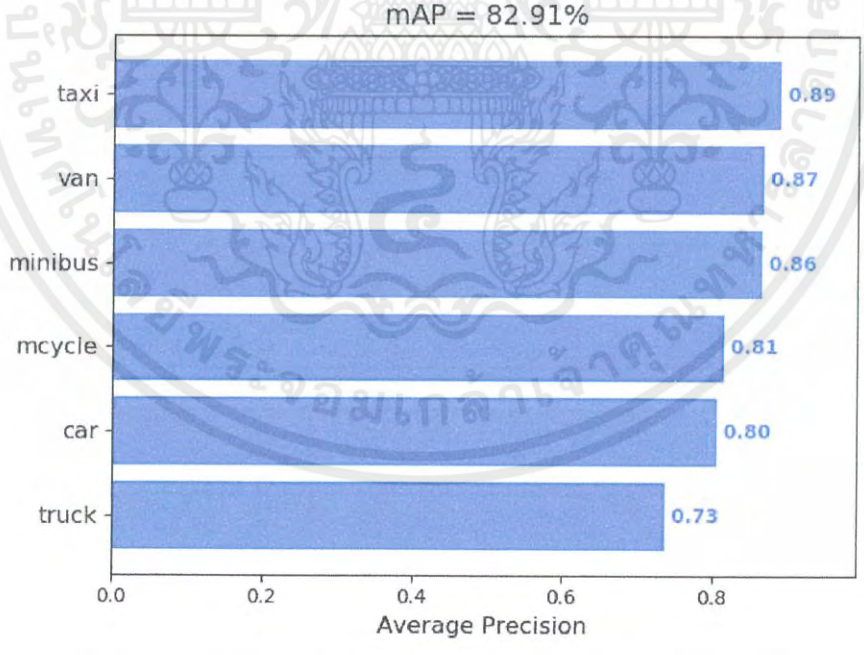
### 4.2.2 วิธีการทดลอง

1) ใช้ Input ที่มีความละเอียด480p, 240p และ 144p ทดสอบกับ Model 480 (จากการทดลองที่ 4.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

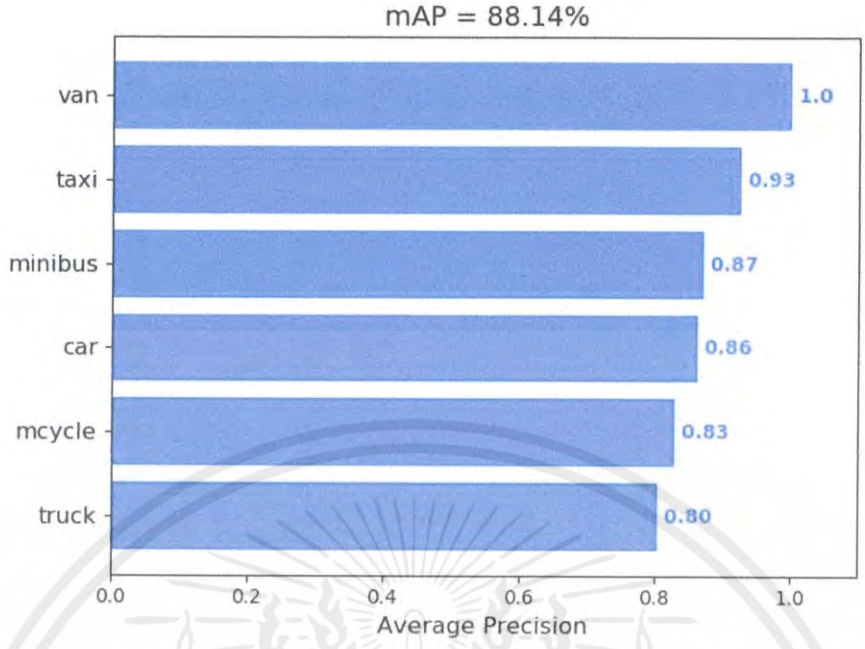



**รูป 4.4 แสดงAP และ mAP ของ Input144p ด้วยModel 480p**




**รูป 4.5 แสดงAP และ mAP ของ Input240p ด้วยModel 480p**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูป 4.6 แสดงAP และ mAP ของInput480p ด้วยModel 480p**

2) เปรียบเทียบค่า mAP ของทั้งสาม Model

**ตารางที่ 4.2 แสดงตารางเปรียบเทียบ AP(Average Precision) และ mAP(Mean Average Precision) ของแต่ละความละเอียดภาพของ Input (144p,240p,480p)**

| Input Resolution | AP | | | | | | mAP |
|---|---|---|---|---|---|---|---|
| | Car | Mcycle | Taxi | Pick-up Truck | Van | Truck | |
| 144p | 68.65% | 77.65% | 66.67% | 86.61% | 76.36% | **88.89%** | 77.47% |
| 240p | 80.49% | 81.41% | **88.89%** | 86.49% | 86.67% | 73.50% | 82.91% |
| 480p | **86.10%** | **82.81%** | 86.94% | **92.59%** | **100.00%** | 80.40% | **88.14%** |

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4.6 แสดงให้เห็นว่ายิ่ง Input มีความละเอียดของภาพสูง ยิ่งสามารถแยกประเภทของยานพาหนะได้ดียิ่งขึ้น แต่สำหรับภาพความละเอียดต่ำ (144p,240p) จะสามารถแยกประเภทของยานพาหนะได้ไม่ดีเท่ากับภาพที่มีความละเอียดสูง

จากการทดลองที่ 4.1 และ 4.2 ทำให้สรุปได้ว่าการใช้ Datasets ในการเทรนโมเดล และ Input ที่มีความละเอียด 480p จะสามารถแยกประเภทของยานพาหนะได้ดีที่สุด

## 4.3 การทดลองเกี่ยวกับการเลือกใช้ Image Pre-Processing

วิธีการ Image Pre-Processing ที่เลือกใช้มีดังต่อไปนี้ 1. Brightness Adjustments 2. Contrast Adjustments 3. Image Blurring (Bilateral Filtering) 4. Grayscale และ 5. Sharpening

### 4.3.1 จุดประสงค์

เพื่อเลือกวิธีการ Image Pre-Processing และค่าพารามิเตอร์ที่เหมาะสมสำหรับใช้เทรนให้โมเดล Faster R-CNN

### 4.3.2 วิธีการทดลอง

1) เลือกวิธีการ Image Pre-Processing แต่ละวิธีมาใช้กับ Datasets ที่จะเทรนให้กับโมเดล ด้วย Datasets ในการเทรนโมเดล และ Input ที่มีความละเอียด 854x480 (480p) โดยมีค่าพารามิเตอร์ จำนวน Datasets และ Test Input ดังต่อไปนี้

- Brightness Adjustments ปรับค่าพารามิเตอร์ 5 ค่า คือ $\beta$ = 30, 50, 70, 90, 110
- Contrast Adjustments ปรับค่าพารามิเตอร์ 5 ค่า คือ $\alpha$ = 1.1, 1.2, 1.3, 1.4, 1.5
- Bilateral Filtering ปรับค่าพารามิเตอร์ 5 ค่า คือ diameter(d) = 6,9,12,15,18
- Grayscale ปรับค่าสี RGB เป็น Grayscale
- Sharpening ใช้ kernel = $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

และใช้จำนวน Datasets เท่ากัน (2358 index) และจำนวนรอบในการเทรนเท่ากัน (20,000 steps)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูป 4.7 แสดงตัวอย่าง Datasets หลัง Pre-Processing ด้วยวิธี Blurring (Bilateral Filtering) (1)Original, (2)Blur6, (3)Blur9, (4)Blur12, (5)Blur15, (6)Blur18**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูป 4.8 แสดงตัวอย่าง Datasets หลัง Pre-Processing ด้วยวิธี Brightness Adjustments (1)Original, (2)Brightness30, (3) Brightness50, (4) Brightness70, (5) Brightness90, (6) Brightness110**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูป 4.9 แสดงตัวอย่าง Datasets หลัง Pre-Processing ด้วยวิธี Contrast Adjustments (1)Original, (2)Contrast1.1, (3) Contrast1.2, (4) Contrast1.3, (5) Contrast1.4, (6) Contrast1.5**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูป 4.10 แสดงตัวอย่าง Datasets หลัง Pre-Processing ด้วยวิธี Grayscale และ Sharpening (1) Original, (2) Grayscale, (3) Sharpening**

**ตารางที่ 4.3 แสดงตัวอย่าง Confusion Matrix จากโมเดลที่ Pre-Processing ด้วยวิธี Blurring (Bilateral Filtering, d=6)**

| | | Predicted | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | mcycle | car | taxi | pick-up truck | van | truck | N/A |
| Acture | mcycle | 50 | 0 | 0 | 0 | 0 | 0 | 24 |
| | car | 1 | 23 | 3 | 1 | 1 | 1 | 79 |
| | taxi | 0 | 0 | 17 | 0 | 0 | 0 | 2 |
| | pick-up truck | 0 | 0 | 0 | 14 | 0 | 0 | 0 |
| | van | 0 | 0 | 0 | 0 | 9 | 0 | 0 |
| | truck | 0 | 0 | 0 | 0 | 0 | 8 | 0 |
| | N/A | 1 | 0 | 0 | 0 | 0 | 1 | 0 |

**ตารางที่ 4.4 แสดงค่า TP, TN, FP, FN, Accuracy, Precision และ Recall ของแต่ละคลาสของโมเดลที่ใช้ Pre-Processing ด้วยวิธี Blurring (Bilateral Filtering, d=6)**

| | TP | TN | FP | FN | Recall(%) | Precision(%) | Accuracy(%) |
|---|---|---|---|---|---|---|---|
| mcycle | 50 | 71 | 2 | 24 | 67.60 | 96.62 | 83.30 |
| car | 23 | 98 | 0 | 86 | 21.11 | 100.00 | 58.50 |
| taxi | 17 | 104 | 3 | 2 | 89.50 | 85.00 | 96.00 |
| pick-up truck | 14 | 107 | 1 | 0 | 100.00 | 93.33 | 99.20 |
| van | 9 | 112 | 1 | 0 | 100.00 | 90.00 | 99.20 |
| truck | 8 | 113 | 2 | 0 | 100.00 | 80.00 | 98.40 |
| allclass | 50 | 71 | 3 | 111 | 79.70 | 90.80 | 51.50 |

2) เปรียบเทียบค่า Precision ของแต่ละโมเดล ได้ผลลัพธ์ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.5 แสดงค่า Precision ของแต่ละโมเดล**

| Precision | | | | | | | |
|---|---|---|---|---|---|---|---|
| Method | mcycle | car | taxi | pick-up truck | van | truck | mAP |
| original | 96.67 | 98.44 | 94.12 | 93.33 | 88.89 | 89.89 | **93.39** |
| gray | 92.31 | 100.00 | 91.67 | 93.33 | 90.00 | 77.78 | 90.85 |
| sharpening | 86.21 | 93.85 | 100.00 | 93.33 | 56.25 | 85.71 | 85.89 |
| contrast1.1 | 96.67 | 100.00 | 100.00 | 87.50 | 72.73 | 88.89 | 90.96 |
| contrast1.2 | 91.67 | 95.65 | 100.00 | 93.33 | 88.89 | 80.00 | 91.59 |
| contrast1.3 | 96.15 | 97.30 | 100.00 | 93.33 | 88.89 | 88.89 | 94.09 |
| contrast1.4 | 96.77 | 88.57 | 88.24 | 93.33 | 88.89 | 80.00 | 89.30 |
| contrast1.5 | 97.30 | 94.59 | 100.00 | 93.33 | 88.89 | 87.50 | 93.60 |
| brightness30 | 97.56 | 98.11 | 100.00 | 93.33 | 81.82 | 88.89 | 93.29 |
| brightness50 | 94.74 | 97.18 | 100.00 | 93.33 | 88.89 | 88.89 | 93.84 |
| brightness70 | 92.31 | 91.84 | 100.00 | 87.50 | 88.89 | 80.00 | 90.11 |
| brightness90 | 92.00 | 97.73 | 94.44 | 93.33 | 88.89 | 97.50 | 92.32 |
| brightness110 | 64.95 | 95.52 | 94.44 | 93.33 | 88.89 | 85.71 | 87.14 |
| blur6 | 96.15 | 100.00 | 85.00 | 93.33 | 90.00 | 80.00 | 90.75 |
| blur9 | 95.00 | 100.00 | 94.44 | 93.33 | 88.89 | 80.00 | 91.94 |
| blur12 | 95.00 | 96.23 | 90.00 | 93.33 | 88.89 | 88.89 | 92.06 |
| blur15 | 91.30 | 100.00 | 94.74 | 93.33 | 90.00 | 80.00 | 91.56 |
| blur18 | 87.76 | 96.15 | 100.00 | 93.33 | 69.23 | 70.00 | 86.08 |

**ตารางที่ 4.6 แสดงการเปรียบเทียบค่า Precision ของแต่ละโมเดล**

| Compared Precision | | | | | | | |
|---|---|---|---|---|---|---|---|
| Method | mcycle | car | taxi | pick-up truck | van | truck | mAP |
| gray | -4.36 | 1.56 | -2.45 | 0.00 | 1.11 | -12.11 | -2.54 |
| sharpening | -10.46 | -4.59 | 5.88 | 0.00 | **-32.64** | -4.18 | **-7.50** |
| contrast1.1 | 0.00 | 1.56 | 5.88 | -5.83 | -16.16 | -1.00 | -2.43 |
| contrast1.2 | -5.00 | -2.79 | 5.88 | 0.00 | 0.00 | -9.89 | -1.80 |
| contrast1.3 | -0.52 | -1.14 | 5.88 | 0.00 | 0.00 | -1.00 | **0.70** |
| contrast1.4 | 0.10 | -9.87 | -5.88 | 0.00 | 0.00 | -9.89 | -4.09 |
| contrast1.5 | 0.63 | -3.85 | 5.88 | 0.00 | 0.00 | -2.39 | **0.21** |
| brightness30 | 0.89 | -0.33 | 5.88 | 0.00 | -7.07 | -1.00 | -0.10 |
| brightness50 | -1.93 | -1.26 | 5.88 | 0.00 | 0.00 | -1.00 | **0.45** |
| brightness70 | -4.36 | -6.60 | 5.88 | -5.83 | 0.00 | -9.89 | -3.28 |
| brightness90 | -4.67 | -0.71 | 0.32 | 0.00 | 0.00 | **7.61** | -1.07 |
| brightness110 | **-31.72** | -2.92 | 0.32 | 0.00 | 0.00 | -4.18 | **-6.25** |
| blur6 | -0.52 | 1.56 | -9.12 | 0.00 | 1.11 | -9.89 | -2.64 |
| blur9 | -1.67 | 1.56 | 0.32 | 0.00 | 0.00 | -9.89 | -1.45 |
| blur12 | -1.67 | -2.21 | -4.12 | 0.00 | 0.00 | -1.00 | -1.33 |
| blur15 | -5.37 | 1.56 | 0.62 | 0.00 | 1.11 | -9.89 | -1.83 |
| blur18 | -8.91 | -2.29 | 5.88 | 0.00 | -19.66 | -19.89 | **-7.31** |

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**จากตารางที่ 4.5 และ 4.6**

วิธี Image Pre-Processing ที่ให้ค่า mAP สูงสุด 3 อันดับแรกได้แก่

1) Contrast Adjustments ($\alpha$=1.3) ให้ค่า mAP = 94.09% เพิ่มมากจากโมเดลเดิม **0.70%**
2) Brightness Adjustments ($\beta$=50) ให้ค่า mAP = 93.84% เพิ่มมากจากโมเดลเดิม **0.45%**
3) Contrast Adjustments ($\alpha$=1.5) ให้ค่า mAP = 93.60% เพิ่มมากจากโมเดลเดิม **0.21%**

วิธี Image Pre-Processing ที่ให้ค่า mAP แย่ลง 3 อันดับแรกได้แก่

1) Sharpening ให้ค่า mAP = 90.85% ลดลงจากโมเดลเดิม **7.50%**
2) Brightness Adjustments ($\beta$=130) ให้ค่า mAP = 90.85% ลดลงจากโมเดลเดิม **7.31%**
3) Bilateral Filtering (d=18) ให้ค่า mAP = 90.85% ลดลงจากโมเดลเดิม **6.25%**

วิธี Image Pre-Processing ที่ส่งผลต่อการเปลี่ยนแปลงค่าAPของคลาสที่โดดเด่นได้แก่

1) Sharpening ให้ค่า AP ของรถตู้ (truck) **ลดลง 32.64%**
2) Brightness Adjustments ($\beta$=110) ให้ค่า AP ของรถจักรยานยนต์ (mcycle) **ลดลง 31.72%**
3) Brightness Adjustments ($\beta$=90) ให้ค่า AP ของรถบรรทุก (truck) **เพิ่มขึ้น 7.61%**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เปรียบเทียบค่า Recall ของแต่ละโมเดล ได้ผลลัพธ์ดังต่อไปนี้

**ตารางที่ 4.7 แสดงค่า Recall ของแต่ละโมเดล**

| Recall | | | | | | | |
|---|---|---|---|---|---|---|---|
| Method | mcycle | car | taxi | pick-up truck | van | truck | mAR |
| original | 39.19 | 57.80 | 84.21 | 100.00 | 88.89 | 100.00 | 78.35 |
| gray | 64.86 | 61.47 | 57.89 | 100.00 | 100.00 | 87.50 | 78.62 |
| sharpening | 33.78 | 55.96 | 78.95 | 100.00 | 100.00 | 75.00 | 73.95 |
| contrast1.1 | 39.19 | **19.27** | 84.21 | 100.00 | 88.89 | 100.00 | **71.93** |
| contrast1.2 | 74.32 | 60.55 | 78.95 | 100.00 | 88.89 | 100.00 | **83.79** |
| contrast1.3 | 33.78 | 66.06 | 89.47 | 100.00 | 88.89 | 100.00 | 79.70 |
| contrast1.4 | 40.54 | 28.44 | 78.95 | 100.00 | 88.89 | 100.00 | **72.80** |
| contrast1.5 | 48.65 | 32.11 | 78.95 | 100.00 | 88.89 | 87.50 | **72.68** |
| brightness30 | 54.05 | 47.71 | 89.47 | 100.00 | 100.00 | 100.00 | 81.87 |
| brightness50 | 48.65 | 63.30 | 84.21 | 100.00 | 88.89 | 100.00 | 80.84 |
| brightness70 | 48.65 | 52.29 | 73.68 | 100.00 | 88.89 | 100.00 | 77.25 |
| brightness90 | 62.16 | 39.46 | 89.47 | 100.00 | 88.89 | 87.50 | 77.91 |
| brightness110 | **85.14** | 58.72 | 89.47 | 100.00 | 88.89 | 75.00 | **82.87** |
| blur6 | 67.57 | **21.10** | 89.47 | 100.00 | 100.00 | 100.00 | 79.69 |
| blur9 | 51.35 | 46.79 | 94.74 | 100.00 | 88.89 | 100.00 | 80.29 |
| blur12 | 51.35 | 46.79 | 94.74 | 100.00 | 88.89 | 100.00 | 80.29 |
| blur15 | 56.76 | 38.53 | 94.74 | 100.00 | 100.00 | 100.00 | 81.67 |
| blur18 | 58.11 | 68.81 | 94.74 | 100.00 | 100.00 | 87.50 | **84.86** |

**ตารางที่ 4.8 แสดงการเปรียบเทียบค่า Recall ของแต่ละโมเดล**

| Compared Recall | | | | | | | |
|---|---|---|---|---|---|---|---|
| Method | mcycle | car | taxi | pick-up truck | van | truck | mAR |
| gray | 25.67 | 3.67 | -26.32 | 0.00 | 11.11 | -12.50 | 0.27 |
| sharpening | -5.41 | -1.84 | -5.26 | 0.00 | 11.11 | -25.00 | -4.40 |
| contrast1.1 | 0.00 | **-38.53** | 0.00 | 0.00 | 0.00 | 0.00 | **-6.42** |
| contrast1.2 | 35.13 | 2.75 | -5.26 | 0.00 | 0.00 | 0.00 | **5.44** |
| contrast1.3 | -5.41 | 8.26 | 5.26 | 0.00 | 0.00 | 0.00 | 1.35 |
| contrast1.4 | 1.35 | -29.36 | -5.26 | 0.00 | 0.00 | 0.00 | **-5.55** |
| contrast1.5 | 9.46 | -25.69 | -5.26 | 0.00 | 0.00 | -12.50 | **-5.67** |
| brightness30 | 14.86 | -10.09 | 5.26 | 0.00 | 11.11 | 0.00 | 3.52 |
| brightness50 | 9.46 | 5.50 | 0.00 | 0.00 | 0.00 | 0.00 | 2.49 |
| brightness70 | 9.46 | -5.51 | -10.53 | 0.00 | 0.00 | 0.00 | -1.10 |
| brightness90 | 22.97 | -18.34 | 5.26 | 0.00 | 0.00 | -12.50 | -0.44 |
| brightness110 | **45.95** | 0.92 | 5.26 | 0.00 | 0.00 | -25.00 | **4.52** |
| blur6 | 28.38 | **-36.70** | 5.26 | 0.00 | 11.11 | 0.00 | 1.34 |
| blur9 | 12.16 | -0.92 | 5.26 | 0.00 | 0.00 | 0.00 | 2.75 |
| blur12 | 12.16 | -11.01 | 10.53 | 0.00 | 0.00 | 0.00 | 1.94 |
| blur15 | 17.57 | -19.27 | 10.53 | 0.00 | 11.11 | 0.00 | 3.32 |
| blur18 | 18.92 | 11.01 | 10.53 | 0.00 | 11.11 | -12.50 | **6.51** |

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**จากตารางที่ 4.7 และ 4.8**

วิธี Image Pre-Processing ที่ให้ค่า Recall สูงสุด 3 อันดับแรกได้แก่

1) Bilateral Filtering (d=18) ให้ค่า Recall = 84.86% เพิ่มมากจากโมเดลเดิม **6.51%**
2) Contrast Adjustments ($\alpha$=1.2) ให้ค่า Recall = 83.79% เพิ่มมากจากโมเดลเดิม **5.44%**
3) Brightness Adjustments ($\beta$=110) ให้ค่า Recall = 82.87% เพิ่มมากจากโมเดลเดิม **4.52%**

วิธี Image Pre-Processing ที่ให้ค่า Recall แย่ลง 3 อันดับแรกได้แก่

1) Contrast Adjustments ($\alpha$=1.1) ให้ค่า Recall = 71.93% ลดลงจากโมเดลเดิม 6.42%
2) Contrast Adjustments ($\alpha$=1.5) ให้ค่า Recall = 72.68% ลดลงจากโมเดลเดิม 5.67%
3) Contrast Adjustments ($\alpha$=1.4) ให้ค่า Recall = 72.80% ลดลงจากโมเดลเดิม 5.55%

วิธี Image Pre-Processing ที่ส่งผลต่อการเปลี่ยนแปลงค่า Recall ของคลาสที่โดดเด่นได้แก่

1) Brightness Adjustments ($\beta$=110) ให้ค่า Recall ของรถจักรยานยนต์ (mcycle) **เพิ่มขึ้น 45.95%**
2) Contrast Adjustments ($\alpha$=1.1) ให้ค่า Recall ของรถยนต์ (car) **ลดลง 38.53%**
3) Bilateral Filtering (d=6) ให้ค่า Recall ของรถยนต์ (car) **ลดลง 36.70%**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เปรียบเทียบค่า Accuracy ของแต่ละโมเดล ได้ผลลัพธ์ดังต่อไปนี้

**ตารางที่ 4.9 แสดงค่า Accuracy ของแต่ละโมเดล**

| Accuracy | |
|---|---|
| Method | Accuracy |
| original | 59.47 |
| gray | 65.55 |
| sharpening | 53.94 |
| contrast1.1 | 40.68 |
| contrast1.2 | **69.17** |
| contrast1.3 | 61.02 |
| contrast1.4 | 44.54 |
| contrast1.5 | 48.94 |
| brightness30 | 59.32 |
| brightness50 | 63.71 |
| brightness70 | 57.08 |
| brightness90 | 56.49 |
| brightness110 | 65.90 |
| blur6 | 51.45 |
| blur9 | 62.29 |
| blur12 | 57.32 |
| blur15 | 55.58 |
| blur18 | 67.76 |

**ตารางที่ 4.10 แสดงการเปรียบเทียบค่า Accuracy ของแต่ละโมเดล**

| Compared Accuracy | |
|---|---|
| Method | Accuracy |
| gray | 6.08 |
| sharpening | -5.53 |
| contrast1.1 | -18.79 |
| contrast1.2 | 9.70 |
| contrast1.3 | 1.55 |
| contrast1.4 | -14.93 |
| contrast1.5 | -10.53 |
| brightness30 | -0.15 |
| brightness50 | 4.24 |
| brightness70 | -2.39 |
| brightness90 | -2.98 |
| brightness110 | 6.43 |
| blur6 | -8.02 |
| blur9 | 2.82 |
| blur12 | -2.15 |
| blur15 | -3.89 |
| blur18 | 8.29 |

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**จากตารางที่ 4.9 และ 4.10**

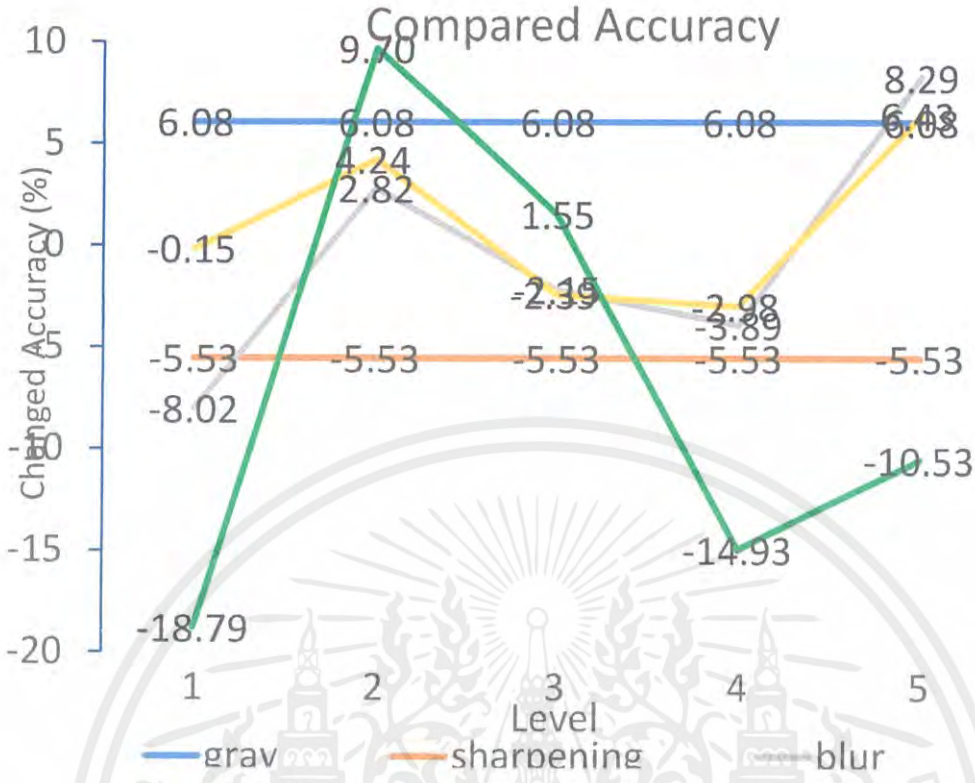
วิธี Image Pre-Processing ที่ให้ค่า Accuracy สูงสุด 3 อันดับแรกได้แก่

1. Contrast Adjustments ($\alpha$=1.2) ให้ค่า Accuracy = 69.17% **เพิ่ม**มากจากโมเดลเดิม **9.70%**
2. Bilateral Filtering ($d$=18) ให้ค่า Accuracy = 67.76% **เพิ่ม**มากจากโมเดลเดิม **8.29%**
3. Brightness Adjustments ($\beta$=110) ให้ค่า Accuracy = 65.90% **เพิ่ม**มากจากโมเดลเดิม **6.43%**

วิธี Image Pre-Processing ที่ให้ค่า Accuracy แย่ลง 3 อันดับแรกได้แก่

1. Contrast Adjustments ($\alpha$=1.1) ให้ค่า Accuracy = 40.68% ลดลงจากโมเดลเดิม **18.79%**
2. Contrast Adjustments ($\alpha$=1.4) ให้ค่า Accuracy = 44.54% ลดลงจากโมเดลเดิม **14.93%**
3. Contrast Adjustments ($\alpha$=1.5) ให้ค่า Accuracy = 48.94% ลดลงจากโมเดลเดิม **10.53%**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูป 4.11 กราฟแสดงการเปรียบเทียบค่า Accuracy ของแต่ละโมเดล**

แม้ใช้วิธีทาง Pre-Processing เดียวกัน แต่การปรับสัมประสิทธิ์ มาก-น้อย ส่งผลต่อโมเดลต่างกัน ไม่ได้เป็นไปในทิศทางเดียวกันเสมอไป สังเกตุได้จากตารางที่ 4.10

1) จากกราฟเส้นสีเขียว (Contrast) เมื่อปรับเพิ่มค่า $\alpha$ ค่า Accuracy เพิ่ม/ลด ในลักษณะฟันปลา (กราฟSin) ที่มีจุด Peak สูงสุด และต่ำสุดหากเทียบกับกราฟเส้นอื่น (Max = 9.70, Min = -18.79) จากตารางที่ 4.6 และ 4.8 จะสังเกตุได้ว่า Contrast ที่สูงมากไป ($\alpha$ >1.3)จะมีผลกระทบมากในการตรวจจับและแยกประเภทใน Class Car ค่าสัมประสิทธิ์ที่เหมาะสมสำหรับปรับใช้ใช้คือ $\alpha$ **= 1.2**

2) จากกราฟเส้นสีส้ม(Brightness) มีลักษณะกราฟ ในลักษณะฟันปลาเล็กๆที่มีความเป็นเปลี่ยนแปลงในทางที่ดีของแต่ละLevelไม่มากนัก จากตารางที่4.6 และ 4.8 จะสังเกตุได้ว่าค่า Precision และ Recall แปรผกผันกัน เห็นได้ชัดจากโมเดลสามารถแยกประเภท Motorcycle ได้ดีขึ้น แต่ตรวจจับเจอMotorcycle น้อยลง ตามค่า Brightness ที่สูงขึ้น ค่าสัมประสิทธิ์ที่เหมาะสมสำหรับปรับใช้ใช้คือ $\beta$ **= 50 , 110**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) จากกราฟเส้นสีเทา (Blur) มีลักษณะการแปรผกผันระหว่าง Precision และ Recall เช่นเดียวกันกับกราฟสีส้ม (Brightness) ค่าสัมประสิทธิ์ที่เหมาะสมสำหรับปรับใช้ใช้คือ **d= 9,18**

4) จากกราฟเส้นสีน้ำเงิน (Gray scale) และสีแดง (Sharpening) ทั้งสองเป็นการปรับ filter ในลักษณะที่ไม่มีการปรับเปลี่ยนค่าสัมประสิทธิ์ ทำให้ทุกระดับมีค่าเท่ากันโดย gray scale ทำให้ accuracy **เพิ่มขึ้น 6.08 %** ส่วน Sharpening ทำให้ค่า Accuracy **ลดลงถึง 8.02%**

5) ในแต่ละวิธีการPre-Processing จากตารางที่4.11 สังเกตุได้ว่าในทุกการปรับค่าสัมประสิทธิ์ Level 1 จะทำให้โมเดลมีค่า Accuracy ลดลงเสมอ และจะดีขึ้นเรื่อยๆจนถึงค่าๆนึง (ในช่วง Gap ระหว่างLevel 1 ถึง Level 3)

**ตารางที่ 4.11 แสดงการเปรียบเทียบค่า Accuracy ของ Pre-Processing Level 1**

| Method | acc |
|---|---|
| contrast1.1 | -18.79 |
| brightness30 | -0.15 |
| blur6 | -8.02 |

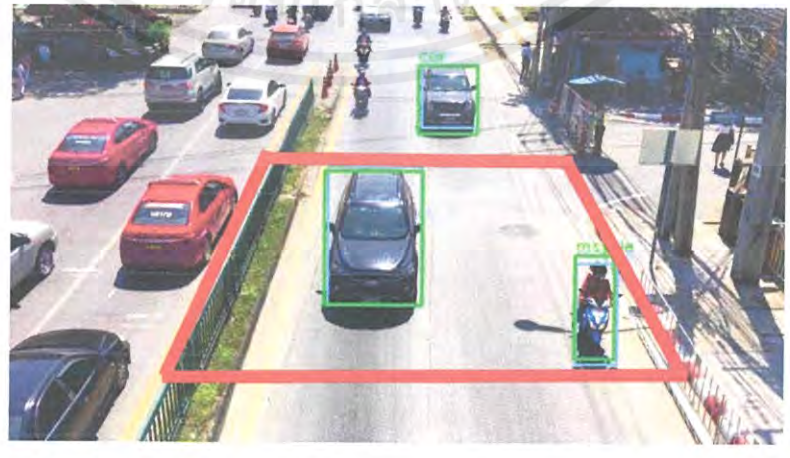
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.4 สรุปการทดลอง

จากการทดลองด้วย วิธี Pre-Processing จากการ Image Enhancement สามารถสรุปผลการทดลองได้ดังนี้

1) การใช้วิธี Grayscale เหมาะสมกับโมเดลที่ต้องการตรวจจับรถจักรยานยนต์เนื่องจากให้ค่า Recall เพิ่มขึ้นถึง 25.67%
2) การใช้วิธี Image Blurring (Bilateral Filtering) และ วิธี Brightness Adjustments เหมาะสมกับโมเดลที่ต้องการตรวจจับรถจักรยานยนต์ เนื่องจากในการปรับค่าพารามิเตอร์ทุกๆระดับ ทำให้ค่า Recall มีเปอร์เซ็นเพิ่มขึ้นเสมอ
3) การใช้วิธี Image Blurring (Bilateral Filtering) เหมาะสมกับโมเดลที่ต้องการตรวจจับรถแท็กซี่ เนื่องจากในการปรับค่าพารามิเตอร์ทุกๆระดับ ทำให้ค่า Recall มีเปอร์เซ็นเพิ่มขึ้นเสมอ
4) การใช้วิธี Contrast Adjustments ด้วยการปรับค่าพารามิเตอร์ $\alpha$ = 1.2 ทำให้ค่า Accuracy โดยรวมของโมเดลเพิ่มขึ้นมากที่สุด คือ 9.70%

เนื่องด้วยการวัดผลโมเดลจาก Test Datasets ที่สุ่มมา ทำให้พบว่ายานพาหนะที่อยู่ระยะไกลจากภาพ สามารถตรวจจับได้น้อย ทำให้ค่า Accuracy ของโมเดลน้อยลงมา จากรูป 4.12 หากเน้นวัดผลเฉพาะยานพาหนะที่อยู่ระยะสายตา ประมาณ 15-20 เมตรจากกล้อง (บริเวณเส้นสีแดง) จะทำให้ค่า Accuracy เพิ่มสูงขึ้นถึง 80% ประกอบกับการใช้ปรับ Contrast Adjustments ด้วยค่าพารามิเตอร์ $\alpha$ = 1.2 ทำให้ Accuracy เพิ่มสูงขึ้นถึง 86%



**รูป 4.12 แสดงพื้นที่ที่โมเดลมีค่า Accuracy สูง**

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# บทสรุปและข้อเสนอแนะ

## 5.1 บทสรุป

จากการที่ได้ทดลองใช้ Model Faster R-CNN Resnet-101 มาตรวจจับและจำแนกยานพาหนะ ได้ผลดังนี้

ความละเอียดของภาพ Datasets ที่ใช้ในการเทรนให้โมเดล และ ความละเอียดของภาพ Input ส่งผลต่อความแม่นยำในการตรวจจับและจำแนกยานพาหนะของโมเดลเป็นอย่างมาก โดยหากใช้ภาพ Datasets ที่มีความละเอียดสูงในการเทรนให้โมเดล ความแม่นยำของโมเดลในการตรวจจับและแยกประเภทยานพาหนะจะมากขึ้นตามไปด้วย เช่นเดียวกันกับความละเอียดของภาพ Input รวมไปถึงวิธีการ Image Pre-Processing ด้วย Image Enhancement เป็นอีกวิธีการที่เพิ่มประสิทธิภาพให้กับโมเดลได้

โดยโมเดลจาก Image Pre-Processing ด้วย Image Enhancement แต่ละวิธีมีข้อดีข้อเสียต่างกัน เหมาะกับความต้องการใช้ที่ต่างกัน เช่น หากต้องการโมเดลที่เน้นไปที่การตรวจจับและแยกประเภทจักรยานยนต์ Grayscale, Brightness และ Blur (Bilateral Filtering) เป็นตัวเลือกที่ดี หากต้องการโมเดลที่เน้นไปที่การตรวจจับแท็กซี่ Blur เป็นตัวเลือกที่ดี

## 5.2 ปัญหาและอุปสรรค

1) การประมวลผลและTraining Model ใช้เวลานาน (GPU: NVIDIA GeForce GTX 1050 4GB) ทำให้กินระยะเวลาในการทดสอบโมเดลแต่ละตัว ไม่เหมาะกับการทำงาน Real Time
2) Trained Model ในตอนนี้ให้ค่าความแม่นยำยังน้อย (Accuracy = 69.17%)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.3 แนวทางในการพัฒนาต่อ

**ในด้านการเพิ่มประสิทธิภาพของโมเดล**

1) ใช้ Image Pre-processing method หลายวิธีพร้อมกัน(ผสมกัน)

2) เพิ่ม Attributes ในการส่งข้อมูลเข้าโมเดล (ปัจจุบันใช้เพียงค่า Pixel ของภาพ) เช่น เวลา, ค่าความเข้มแสง เป็นต้น

3) แบ่งประเภทยานพาหานะให้ละเอียดกว่าเดิม

**ในด้านการประยุกต์ใช้โมเดล**

1) ระบบนับรถ/ตรวจจับความเร็วรถ/แยกประเภทรถ/จากกล้องวงจรปิด

2) ระบบค้นหาและติดตามรถต้องสงสัย

3) ระบบตรวจจับรถจักรยานยนต์ที่ทำผิดกฎจราจร

4) ระบบตรวจจับจำนวนรถแท็กซี่ตามห้างสรรพสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors", *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, july 2017.

KB Saran, G Sreelekha, "Traffic Video Surveillance: Vehicle Detection and Classification", International Conference on Control Communication and Computing India, 2015.

C. Jiang, B. Zhang, "Weakly-supervised vehicle detection and classification by convolutional neural network", *Int. Congr. Image and Sig. Proc. BioMedical Eng. and Informatics.* IEEE, pp. 570-575, 2016.

S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." In IEEE Trans. Pattern Anal. Mach. Intell. 39(6): 1137–1149 (2017)

Wenzhe Wang, Bin Wu, Sixiong Yang, Zhixiang Wang, "Road Damage Detection and Classification with Faster R-CNN", IEEE International Conference on Big Data, 2018.

ระบบกล้องโทรทัศน์วงจรปิด (cctv). [Online].
Available: https://www.bmatraffic.com

Deep Learning คืออะไร. [Online].
Available: https://medium.com/@athivvat

สรุปแนวคิด Neural Network แบบไม่มี Math. [Online].
Available: https://coladev.com/



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Neural Network 101: CNN with TensorFlow(cctv). [Online].
Available: https://medium.com/@thebear19

R-CNN, Fast R-CNN, Faster R-CNN, YOLO—Object Detection Algorithms. [Online].
Available: https://towardsdatascience.com

What is mAP Evaluation?. [Online].
Available: https://medium.com/boobeejung/

Tensorflow Object Detection API. [Online].
Available: https://github.com/tensorflow/

Get Started with TensorFlow. [Online].
Available: https://www.tensorflow.org/tutorials/

Convolutional Neural Network (CNN). [Online].
Available: https://developer.nvidia.com/

Convolutional Networks. [Online].
Available: http://www.deeplearningbook.org/contents/convnets.html

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้