

การประยุกต์ใช้ YOLOv4-Tiny สำหรับตรวจจับรถยนต์
จากภาพวิดีโอที่สนใจความละเอียดต่ำ

กัณฐิกา วงษ์เกษม

TNII

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
บัณฑิตศึกษา สาขาเทคโนโลยีวิศวกรรม
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ปีการศึกษา 2566

IMPLEMENTATION OF YOLOV4-TINY FOR CAR DETECTION
WITH LOW-RESOLUTION VIDEO

Kantika Wongkasem

TNII

A Thesis Submitted in Partial Fulfillment of Requirements
for The Degree of Master of Engineering Program in Engineering Technology
Graduate Studies

Thai-Nichi Institute of Technology

Academic Year 2023

หัวข้อวิทยานิพนธ์

การประยุกต์ใช้ YOLOv4-Tiny สำหรับตรวจจับรถยนต์
จากภาพวิดีโอที่ส่งความละเอียดต่ำ

โดย

กัญญิกา วงษ์เกษม

สาขาวิชา

เทคโนโลยีวิศวกรรม

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ผู้ช่วยศาสตราจารย์ ดร.พรชัย นิเวศน์รังสรรค์

บัณฑิตศึกษา สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้บัณฑิตวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาโทบริหารบัณฑิต

.....รองอธิการบดีฝ่ายวิชาการ

(รองศาสตราจารย์ ดร.วรากร ศรีเชวงทรัพย์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ฐิตะพล หุยนันท)

.....กรรมการ

(ดร.วิชรินทร์ หนูทอง)

.....กรรมการ

(ดร.มรรณพ พักขาว)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ผู้ช่วยศาสตราจารย์ ดร.พรชัย นิเวศน์รังสรรค์)

กัญฐิกา วงษ์เกษม : การประยุกต์ใช้ YOLOv4-Tiny สำหรับตรวจจับรถยนต์จากภาพ
 วิถีทัศนความละเอียดต่ำ. อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.พรชัย นิเวศน์รังสรรค์,
 99 หน้า.

ปัจจุบันมีการนำโดรนมาใช้ประโยชน์ในด้านการบินสำรวจ การตรวจจับวัตถุและการจราจร
 งานวิจัยนี้จะศึกษาพัฒนาระบบตรวจจับรถยนต์ที่เป็นวัตถุขนาดเล็กด้วยวิถีโทางอากาศที่มีความ
 ละเอียดต่ำจากโดรน โดยประยุกต์ใช้เทคโนโลยีการเรียนรู้เชิงลึกสำหรับงานทางด้านคอมพิวเตอร์วิ
 ทัศน์ โดยที่ตัวตรวจหาวัตถุสามารถเรียนรู้คุณลักษณะของรูปภาพรถยนต์ขนาดเล็กได้ งานวิจัยได้
 เลือกใช้อัลกอริทึม YOLOv4-Tiny ซึ่งเป็นโมเดลขนาดเล็กที่มีประสิทธิภาพสูง โดยทำการเปรียบเทียบ
 องค์ประกอบของโมเดลและการปรับค่าพารามิเตอร์ต่างๆ เพื่อใช้ในการฝึกอบรมโมเดลให้ได้
 ประสิทธิภาพสูงสุดโดยเปรียบเทียบปัจจัยหลักดังนี้การเตรียมข้อมูล การทำเทคนิคการเสริมข้อมูล
 และการปรับปรุงด้วยออฟติไมเซอร์ SGDM และ ADAM เพื่อหาวิธีการปรับปรุงและเลือกค่าอินพุตที่
 เหมาะสมสำหรับการตรวจจับวัตถุขนาดเล็ก โดยพิจารณาจากผลการประเมินความสามารถของโมเดล
 ด้วยค่าความแม่นยำเฉลี่ย ซึ่งเป็นมาตรฐานชี้วัดสำหรับความแม่นยำโดยรวมของโมเดลในการตรวจจับ
 วัตถุ ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny พบว่าชุดข้อมูลที่ใช้เทคนิคการเสริมข้อมูล
 ด้วยออฟติไมเซอร์ ADAM สำหรับอินพุตภาพขนาด 603x603 พิกเซลให้ค่าความแม่นยำของการ
 ตรวจจับรถยนต์ขนาดเล็กด้วยภาพความละเอียดต่ำจากโดรน สูงที่สุด ซึ่งสามารถนำไปพัฒนาโดรนที่
 มีอัลกอริทึมตรวจจับรถยนต์ด้วยภาพความละเอียดต่ำกับวัตถุขนาดเล็กนี้ได้ รวมทั้งสามารถช่วยให้
 ผู้ใช้งานประหยัดเวลาและนำเทคโนโลยีไปใช้พัฒนาระบบโดรนได้

บัณฑิตศึกษา

สาขาวิชา เทคโนโลยีวิศวกรรม

ปีการศึกษา 2566

ลายมือชื่อนักศึกษา

ลายมือชื่ออาจารย์ที่ปรึกษา

KANTIKA WONGKASEM : IMPLEMENTATION OF YOLOV4-TINY FOR CAR DETECTION
WITH LOW-RESOLUTION VIDEO. ADVISOR : ASST. PROF. DR PORNCHAI
NIVESRANGSAN, 99 PP.

At present, drones are being utilized for various applications, including aerial surveillance, object detection, and traffic monitoring. This research focuses on developing a car detection system using low-resolution aerial images captured by drones with a specific emphasis on detecting small-sized cars. Deep learning technology in the field of computer vision was applied using the YOLOv4-Tiny algorithm. YOLOv4-Tiny is an efficient and high-performance small-sized model. This research compares model components and various input parameter adjustments to be used in the training process to achieve a maximum model's performance by comparing the following factors such as data preparation, data augmentation techniques, and the selection of the SGDM or ADAM optimizer. To enhance and select suitable input parameters for detecting small objects, the evaluation results of the model performance used average precision which is a standard measure of overall precision in object detection applications. It can be seen that the augmentation dataset with ADAM optimizer for an input size of 603x603 pixels gives the highest accuracy in detecting small-sized cars in low-resolution drone images. These findings can be applied to further the development of drone systems for car detection in similar conditions. This research has the potential to save time and resources, making drone technology more accessible for a variety of applications.

Graduate Studies

Student's Signature.....

Field of Engineering of Technology

Advisor's Signature.....

Academic Year 2023

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้ดำเนินการสำเร็จลงได้ด้วยดี ผู้วิจัยขอขอบพระคุณท่านอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.พรชัย นิเวศน์รังสรรค์ ที่กรุณาให้คำปรึกษาในงานวิจัย ให้คำแนะนำ ข้อคิดเห็นรวมถึงการให้คำวิจารณ์ต่างๆ ที่เป็นประโยชน์ต่องานวิจัยนี้ ทำให้งานวิจัยสมบูรณ์แบบ ผู้เขียนจึงขอบพระคุณไว้ ณ โอกาสนี้ รวมถึงอาจารย์ทุกท่านที่มีได้กล่าวนามที่ให้ความอนุเคราะห์ให้ความช่วยเหลือในด้านต่างๆ และคอยให้คำแนะนำที่เป็นประโยชน์กับงานวิจัย ทำให้งานวิจัยนี้สำเร็จ ลุล่วงไปได้ด้วยดี

ทั้งนี้ผู้วิจัยขอขอบพระคุณคณะกรรมการสอบทุกท่าน ที่ได้กรุณาให้ข้อเสนอแนะ คำวิจารณ์ที่เป็นประโยชน์และช่วยตรวจสอบงานวิจัย วิทยานิพนธ์ฉบับนี้จนสมบูรณ์แบบ

ขอขอบพระคุณ บริษัทแอสเซนดาส ซิสเทมส์ จำกัด ที่ให้ความอนุเคราะห์ข้อมูล สนับสนุนเครื่องมือ และให้คำปรึกษาที่ใช้ในการในการดำเนินงานวิจัยครั้งนี้

สุดท้ายนี้ผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์ และเกิดคุณค่าแก่การพัฒนาาระบบอัจฉริยะในภาคเทคโนโลยีทางด้านคอมพิวเตอร์วิทัศน์ ในการปรับปรุงและพัฒนาการตรวจจับวัตถุในรูปต่อไป

กัญฐิกา วงษ์เกษม

TNI

TAI - NICHII INSTITUTE OF TECHNOLOGY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญรูป.....	ฎ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมา และความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการศึกษา	2
1.3 ขอบเขตของการศึกษา	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 คอมพิวเตอร์วิทัศน์.....	5
2.2 การตรวจจับวัตถุ.....	6
2.3 การประมวลผลภาพ.....	9
2.4 อัลกอริทึมในงานการตรวจจับวัตถุ.....	10
2.5 อัลกอริทึม YOLO (You Only Look Once).....	12
2.6 เทคนิคการเสริมข้อมูล.....	16
2.7 การปรับปรุงคุณภาพ (Optimizer)	19
2.8 การประเมิน (Evaluation)	24
3 วิธีการดำเนินงาน	28
3.1 ขั้นตอนการทดลอง	29
3.2 การเตรียมข้อมูล (Data Preparation)	29

สารบัญ (ต่อ)

บทที่		หน้า
3	3.3 การเสริมขนาดของข้อมูล (Data Augmentation).....	34
	3.4 การกำหนดการฝึกอบรมล่วงหน้าโมเดล YOLOv4-Tiny	36
	3.5 การระบุตัวเลือกออฟติไมเซอร์ ของการฝึกอบรม	37
	3.6 การฝึกอบรมโมเดล YOLOv4-Tiny.....	40
	3.7 การฝึกอบรมโมเดล	43
4	ผลการวิจัย.....	48
	4.1 ผลจากการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยชุดข้อมูลภาพปกติที่ไม่ผ่าน การทำเทคนิคการเสริมขนาดข้อมูล.....	48
	4.2 ผลจากการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยชุดข้อมูลภาพผ่านการเสริม ข้อมูล.....	56
	4.3 ผลการประเมินความแม่นยำของโมเดลที่ฝึกอบรมทั้งหมด.....	64
5	สรุปผลและข้อเสนอแนะ	70
	5.1 สรุปผลการวิจัย.....	70
	5.2 ข้อเสนอแนะ	72
	บรรณานุกรม	74
	ภาคผนวก	81
	ภาคผนวก ก. โค้ดโปรแกรมภาษา MATLAB	82
	ภาคผนวก ข. บทความการประชุมวิชาการ	88
	ประวัติย่อผู้วิจัย	99

สารบัญตาราง

ตาราง	หน้า
2.1 เปรียบเทียบความแตกต่าง YOLOv4-Tiny และ YOLOv4.....	16
3.1 คุณสมบัติของอุปกรณ์.....	29
3.2 คุณสมบัติของวิดีโอ.....	30
3.3 การตั้งค่าพารามิเตอร์ SGDM options1.....	38
3.4 การตั้งค่าพารามิเตอร์ ADAM options2.....	39
4.1 ผลการฝึกอบรมโมเดลด้วยขนาด InputSize 416x416 พิกเซล	65



สารบัญรูป

รูป	หน้า
2.1 ตัวอย่างประเภทของอัลกอริทึมการตรวจจับวัตถุ.....	6
2.2 ตัวอย่างการตรวจจับวัตถุในรูปภาพ.....	8
2.3 การเรียนรู้ของปัญญาประดิษฐ์	11
2.4 กระบวนการทำงานของอัลกอริทึม YOLO.....	12
2.5 ระบบโครงข่ายของโมเดล YOLOv4-Tiny	15
2.6 Intersection over Union (IoU).....	25
3.1 ขั้นตอนการทดลอง.....	29
3.2 รูปภาพในการฝึกอบรมโมเดล.....	31
3.3 ขั้นตอนของกระบวนการระบุภาพ	32
3.4 ตัวอย่างไฟล์ข้อมูลที่ระบุตำแหน่งของวัตถุในภาพ.....	33
3.5 ตัวอย่างภาพการเสริมข้อมูล	36
3.6 ตัวอย่างขั้นตอนการฝึกอบรมตัวตรวจจับด้วยโมเดล YOLOv4-Tiny.....	42
3.7 ตัวอย่างกราฟผลการฝึกอบรมตัวตรวจจับด้วยโมเดล YOLOv4-Tiny	43
3.8 ตัวอย่างโปรแกรมการทดสอบโมเดลตรวจจับวัตถุด้วยรูปภาพทดสอบ	44
3.9 ผลการทดสอบการตรวจจับรถยนต์ด้วยภาพทดสอบ videoplayback1_0611.png.....	44
3.10 ตัวอย่างการทดสอบการประเมินความสามารถของโมเดล	45
3.11 กราฟการพล็อตค่า RP Curve.....	46
4.1 ผลการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM ที่ข้อมูลภาพปกติ ขนาด 416x416 พิกเซล.....	49
4.2 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล.....	49
4.3 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล.....	50
4.4 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล.....	51
4.5 ผลประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพ ทดสอบแบบปกติขนาด 416x416 พิกเซล.....	51

สารบัญรูป (ต่อ)

รูป	หน้า
4.6 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล	52
4.7 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล	53
4.8 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล	53
4.9 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล	54
4.10 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 603x603 พิกเซล	55
4.11 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 603x603 พิกเซล	55
4.12 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 603x603 พิกเซล	56
4.13 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบที่ผ่านการเสริมขนาดข้อมูลด้วยภาพขนาด 416x416 พิกเซล	57
4.14 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบที่ผ่านการเสริมขนาดข้อมูลด้วยภาพขนาด 416x416 พิกเซล	57
4.15 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลด้วยภาพขนาด 416x416 พิกเซล	58
4.16 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 416x416 พิกเซล	59
4.17 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 416x416 พิกเซล	59
4.18 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 416x416 พิกเซล	60

สารบัญรูป (ต่อ)

รูป	หน้า
4.19 ผลการฝึกสอนโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 288x288 พิกเซล.....	61
4.20 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 288x288 พิกเซล.....	61
4.21 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 288x288 พิกเซล.....	62
4.22 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 603x603 พิกเซล.....	63
4.23 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 603x603 พิกเซล.....	63
4.24 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 603x603 พิกเซล.....	64
4.25 กราฟเปรียบเทียบผลการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยออฟติไมเซอร์ ADAM กับ SGDM กับข้อมูลปกติ และ ข้อมูลที่ผ่านการเสริมขนาดข้อมูล	66
4.26 ผลการเปรียบเทียบระยะเวลาที่ใช้ฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์ ADAM	67
4.27 ผลการเปรียบเทียบค่าความแม่นยำเฉลี่ยของการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยออฟติไมเซอร์ ADAM กับข้อมูลปกติ และ ข้อมูลที่ทำการเสริมขนาดข้อมูล	68

บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญของปัญหา

การนำอากาศยานไร้คนขับหรือโดรน ไปใช้ในการตรวจจับรถยนต์ในปัจจุบัน มีจำนวนเพิ่มขึ้น การจราจรบนถนนในเมืองหลวงและเมืองหลักของประเทศไทยมีการจราจรหนาแน่นเพิ่มขึ้น ตามการเติบโตทางเศรษฐกิจ และพบอัตราการเกิดอุบัติเหตุบนถนนเพิ่มมากขึ้น ทำให้การตรวจสอบความผิดปกติต่างๆ ด้วยการนำอากาศยานไร้คนขับหรือการนำอุปกรณ์เสริมทางเทคโนโลยีอื่นๆ เข้ามาใช้นั้น มีส่วนสำคัญต่อการช่วยตรวจจับความผิดปกติต่างๆ บนถนนเพื่อ ตรวจจับวัตถุชนิดที่กำหนดที่อยู่ในรูปภาพหรือวิดีโอ เช่น มนุษย์ รถยนต์ อาคาร เป็นต้น การนำเทคนิคทางคอมพิวเตอร์เข้ามาช่วยวิเคราะห์ด้วยเทคโนโลยีทางด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) [1] การตรวจจับวัตถุในรูปภาพ (Object Detection) [2] ได้มีการพัฒนาไปอย่างก้าวหน้าในปัจจุบัน โดยได้นำเทคโนโลยีในการประมวลผลภาพด้วยคอมพิวเตอร์ที่ใช้ในการหาตำแหน่งและระบุวัตถุในรูปภาพ ซึ่งเป็นรูปแบบขั้นสูงของการจำแนกภาพ (Image Classification) [3] ด้วยโครงข่ายประสาทเทียม (Neural Network) [4] สามารถทำนายวัตถุในรูปภาพและกำหนดตำแหน่งของวัตถุด้วยกล่องครอบ (Bounding Boxes) [5] โดยภาพถ่ายจาก UAV ที่มีความละเอียดต่ำเป็นงานที่ท้าทายสำหรับการตรวจหาวัตถุขนาดเล็ก

ปัจจุบันการตรวจจับวัตถุด้วยเทคโนโลยีทางด้านคอมพิวเตอร์มีการประยุกต์ใช้อยู่หลายเทคนิค เช่น การใช้เทคนิคการเรียนรู้เชิงลึก (Deep Learning) [6] เป็นเครื่องมือที่มีประสิทธิภาพสำหรับเทคโนโลยีทางด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) โดยที่ตัวตรวจหาวัตถุสามารถเรียนรู้คุณลักษณะของรูปภาพที่จำเป็นสำหรับงานตรวจจับวัตถุได้โดยอัตโนมัติ เทคนิคสำหรับการตรวจจับวัตถุโดยใช้เทคนิคการเรียนรู้เชิงลึกที่นิยมใช้ในงานวิจัย เช่น Faster R-CNN [7] และ YOLO (You Only Look Once) [8] การประยุกต์การวิเคราะห์ด้วยเทคนิคการเรียนรู้เชิงลึกสำหรับวิดีโอจากโดรนด้วยภาพความละเอียดต่ำ โดยความสามารถของโมเดลมีความแม่นยำสูงขึ้นเนื่องจากหลายปัจจัย ซึ่งขึ้นอยู่กับองค์ประกอบ และค่าการปรับปรุงคุณภาพของภาพและตัวแปรในการปรับปรุงโมเดล

เทคนิคการเสริมข้อมูลรูปภาพ (Data Augmentation) [9] เป็นเทคนิคที่นิยมใช้ในงานการตรวจจับวัตถุด้วยภาพซึ่งเป็นกระบวนการที่นิยมใช้เพื่อปรับปรุงและขยายชุดข้อมูลสำหรับการฝึกสอน (Training)[10] โมเดล เช่น การพลิกภาพ การปรับขนาดภาพ การครอบตัดภาพ การหมุนภาพ การเพิ่มข้อมูลสี การปรับความสว่าง เป็นต้น นอกจากนี้การใช้ข้อผิดพลาด (Optimizer) [11] ด้วยการ

ปรับค่าพารามิเตอร์ที่เกี่ยวข้องจะทำให้โมเดลสามารถเรียนรู้ข้อมูลที่ฝึกสอนได้ดียิ่งขึ้น ออฟติไมเซอร์ที่นิยมใช้ในงานจำแนกวัตถุในภาพ [12] เช่น Stochastic Gradient Descent with Momentum (SGDM) [13] , Stochastic Gradient Descent (SGD), Adaptive Moment Estimate (ADAM) [14] , Root Mean Square Propagation (RMSprop) ออฟติไมเซอร์แต่ละชนิดจะมีความเหมาะสมของการใช้งานที่แตกต่างกัน สำหรับการตรวจจับวัตถุขนาดเล็กด้วยภาพนิยมใช้ SGDM และ ADAM การศึกษานี้ได้เล็งเห็นถึงความสำคัญของความสามารถของโมเดลสำหรับการตรวจจับรถยนต์ขนาดเล็กด้วยภาพความละเอียดต่ำ โดยเลือกใช้เทคนิค YOLOv4-Tiny [15] โดยเปรียบเทียบผลลัพธ์ด้วยค่าความแม่นยำเฉลี่ย (Average Precision) [16] รวมทั้งเปรียบเทียบองค์ประกอบของโมเดลและความสามารถในการตรวจจับวัตถุด้วยการปรับค่าตัวเลือกที่ต้องกำหนด ในอัลกอริทึมของเทคนิคการเสริมข้อมูลรูปภาพ เพื่อปรับปรุงคุณภาพและความหลากหลายของชุดข้อมูลภาพในการฝึกโมเดล และการเปรียบเทียบตัวแปรที่กำหนดของออฟติไมเซอร์ เพื่อลดค่าความผิดพลาดระหว่างการทำนายของโมเดลกับค่าเป้าหมายที่ต้องการให้มีความแม่นยำมากที่สุด

การฝึกสอนโมเดลจะทำการแบ่งชุดข้อมูลเป็น 2 ชุด คือชุดรูปภาพปกติ และชุดข้อมูลรูปภาพที่ใช้เทคนิคการเสริมข้อมูล เทคนิคสำหรับการเสริมข้อมูลประกอบด้วย 3 เทคนิค ได้แก่ การปรับเปลี่ยนค่าสีของภาพ การพลิกภาพในแนวนอนแบบสุ่ม และการปรับขนาดแบบสุ่ม เทคนิคที่ใช้ในงานวิจัยนี้จะช่วยให้โมเดลสามารถเรียนรู้จากข้อมูลที่มีความแตกต่างมากขึ้น รวมถึงการเลือกออฟติไมเซอร์ และค่าพารามิเตอร์ของโมเดลที่ต้องกำหนดให้เหมาะสมซึ่งมีผลต่อความสามารถของโมเดลต่อการตรวจจับวัตถุขนาดเล็กจากภาพที่มีความละเอียดต่ำ งานวิจัยจะทำการเปรียบเทียบการเลือกใช้ออฟติไมเซอร์แบบ ADAM และ SGDM ซึ่งเป็นออฟติไมเซอร์ที่นิยมใช้ในการตรวจจับวัตถุจากภาพที่มีความละเอียดต่ำ ซึ่งมีความสำคัญต่อความสามารถของโมเดล ซึ่งจะทดลองปรับค่าต่างๆ เพื่อหาตัวแปรชุดที่ดีที่สุด สำหรับอัปเดตโมเดลเพื่อให้โมเดลมีความสามารถต่อการตรวจจับวัตถุและแม่นยำมากขึ้น

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อพัฒนาวิธีการตรวจจับรถยนต์ สำหรับภาพรถยนต์ขนาดเล็กที่มีความละเอียดต่ำด้วยการใช้ YOLOv4-Tiny

1.2.2 เพื่อเปรียบเทียบวิธีการเสริมขนาดของข้อมูลรูปภาพ ขนาดอินพุตของรูปภาพ และออฟติไมเซอร์ สำหรับปรับปรุงโมเดลให้มีความแม่นยำมากขึ้น

1.3 ขอบเขตของการศึกษา

1.3.1 วิดีโอที่บันทึกไว้จะถูกนำมารวบรวมเป็นข้อมูลรูปภาพจากโดรนและเตรียมข้อมูลเพื่อนำมาฝึกสอนโมเดล ข้อมูลที่ใช้ถูกแบ่งออกเป็น ข้อมูลที่ใช้ฝึกสอน (Training set) ชุดข้อมูลตรวจสอบ (Validation set) และข้อมูลที่ใช้ทดสอบ (Test set)

1.3.2 การเสริมขนาดของข้อมูลรูปภาพ (Data Augmentation) ประกอบด้วย 3 เทคนิค ดังนี้ การปรับเปลี่ยนค่าสีของภาพ (Color jitter augmentation in HSV space) , การพลิกภาพ แนวนอนแบบสุ่ม (Random horizontal flip) , การย่อหรือขยายภาพ 10% แบบสุ่ม (Random scaling by 10 percent)

1.3.3 การกำหนดขนาดอินพุตของข้อมูลรูปภาพสำหรับตรวจจับวัตถุขนาดเล็ก (Tiny object) ที่ใช้สำหรับการสอนโมเดล YOLOv4-Tiny โดยเลือกขนาดภาพเพื่อเปรียบเทียบ 3 ขนาด ได้แก่ 288 x 288 pixel , 416 x 416 Pixel ซึ่งเป็นค่าเริ่มต้นของโมเดล YOLOv4-Tiny และ 608 x 608 Pixel

1.3.4 โมเดลที่พัฒนาใช้เพื่อเป็นต้นแบบในการนำ YOLOv4-Tiny มาใช้ในการฝึกสอนโมเดล เพื่อตรวจจับรถยนต์ภาพความละเอียดต่ำ และทำการปรับค่าการปรับปรุงโมเดลเพื่อให้ได้ผลการทดลองที่แม่นยำ โดยเลือกoptimizerที่ใช้คืออัลกอริทึมการฝึกสอนแบบ SGDM และ ADAM

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถจัดทำระบบเทคโนโลยีร่วมกันกับอากาศยานไร้คนขับเพื่อตรวจจับรถยนต์บนท้องถนนได้

1.4.2 โมเดลที่พัฒนาขึ้น สามารถนำไปเป็นแนวทางในการตรวจจับรถยนต์บนถนนที่มีภาพขนาดเล็กและความละเอียดต่ำขนาดเล็กโดยใช้เทคนิค YOLOv4-Tiny ให้มีความเหมาะสมและประสิทธิภาพสูงสุด

1.4.3 สามารถลดต้นทุนของการใช้แรงงานคนในการตรวจสอบความผิดปกติทางจราจรและเพิ่มประสิทธิภาพของการจราจรให้ดีขึ้น



บทที่ 2

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

การตรวจจบบรรยากาศบนท้องถนนด้วยเทคโนโลยี เป็นงานวิจัยที่ศึกษาการประยุกต์ใช้เทคโนโลยีทางด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) และการพัฒนาวิธีการเรียนรู้เชิงลึก (Deep Learning) ของเครื่องคอมพิวเตอร์ หรือปัญญาประดิษฐ์ (Artificial Intelligence) เพื่อนำเทคโนโลยีมาประยุกต์ใช้งานด้านจราจรหรือ สามารถตรวจสอบความผิดปกติ บนถนนได้ การศึกษานี้จะนำหลักการพื้นฐาน ทฤษฎีของงานวิจัยที่เกี่ยวข้องในการนำการเรียนรู้เชิงลึกของปัญญาประดิษฐ์ ไปประยุกต์ใช้งานทางด้านการตรวจจบบรรยากาศที่มีภาพความละเอียดต่ำจากโดรน การตรวจจบบรรยากาศเป็นสิ่งสำคัญในการปฏิบัติการจราจรบนถนนโดยใช้ยานพาหนะอากาศยานไร้คนขับ (UAV) เนื่องจาก UAV บินที่ระดับความสูง จึงได้ภาพจราจรบนถนนครอบคลุมพื้นที่ขนาดใหญ่ วัตถุที่ถูกถ่ายภาพจึงมักมีขนาดเล็ก และประเภทของวัตถุมีความไม่แน่นอนสูง เมื่อพิจารณาถึงความสามารถในการประมวลผลที่จำกัดบน UAV ที่ใช้โครงข่ายประสาทแบบคอนโวลูชันส่วนลึก (Deep Convolutional Neural Network) จึงมีปัญหาในเรื่องของความสามารถของการตรวจจบบรรยากาศแบบทันที (Real Time)

เทคนิคการเรียนรู้เชิงลึก เป็นเทคนิคที่นิยมใช้ในปัจจุบันได้ถูกนำมาใช้ในงานตรวจจบบรรยากาศ (Object Detection) ด้วยการใช้งานอัลกอริทึมด้านการตรวจจบบรรยากาศแบบทันที ได้แก่ You Only Look Once (YOLO) อัลกอริทึมนี้จะมีการฝึกสอนโมเดลสำหรับการตรวจจบบรรยากาศ จึงทำให้การตรวจจบบรรยากาศแบบทันที ในวิดีโอมีความแม่นยำสูง ซึ่งอัลกอริทึมตระกูล YOLO นี้ เหมาะสำหรับงานตรวจจบบรรยากาศแบบทันที ในวิดีโอ โดยใช้เทคนิคตัวตรวจจบบรรยากาศขั้นตอน (Single-Stage/One-Stage Detection) เช่น YOLO v1, v2, v3 [16], [17], [18] และ Single Shot MultiBox Detector (SSD) [19] และอีกวิธีการของการตรวจจบบรรยากาศ คือแบบตัวจบบรรยากาศสองขั้นตอน (Two-Stage Detection) เช่น RCNN, Fast RCNN และ Faster RCNN [20], [21], [22] ช่วงการฝึกสอนการใช้เทคนิคการเสริมข้อมูล เช่น การปรับเปลี่ยนค่าสีของภาพ การพลิกภาพแนวนอนแบบสุ่ม และการย่อหรือขยายภาพแบบสุ่ม เทคนิคเหล่านี้จะนำมาช่วยฝึกสอนให้โมเดลมีความสามารถเพิ่มขึ้น นอกจากนี้การปรับปรุงความสามารถของโมเดลต่อการเรียนรู้สามารถใช้optimizer ช่วยเพิ่มความสามารถของการตรวจจบบรรยากาศขนาดเล็กได้ดีขึ้น การประเมินความสามารถของโมเดลเป็นส่วนสำคัญที่ใช้โดยจะใช้ดัชนีต่าง ๆ เช่น ค่าความถูกต้อง ค่าความแม่นยำ ค่าความแม่นยำเฉลี่ย เป็นต้น สำหรับประเมินความสามารถของโมเดลการตรวจจบบรรยากาศ และช่วยในการปรับปรุงโมเดลให้ดีขึ้นด้วย โดยรายละเอียดที่เกี่ยวข้องของงานวิจัยจะกล่าวในหัวข้อต่อไป

2.1 คอมพิวเตอร์วิทัศน์

คอมพิวเตอร์วิทัศน์ เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่ศึกษาเกี่ยวกับการทำให้คอมพิวเตอร์สามารถเข้าใจและประมวลผลภาพได้ คอมพิวเตอร์วิทัศน์มีจุดมุ่งหมายเพื่อให้คอมพิวเตอร์สามารถรับรู้และแปลความสิ่งต่าง ๆ จากภาพได้เช่นเดียวกับมนุษย์ โดยคอมพิวเตอร์วิทัศน์จะใช้อัลกอริทึมต่าง ๆ เพื่อแยกแยะวัตถุในภาพ ตรวจจับการเคลื่อนไหว ระบุใบหน้า และอื่น ๆ

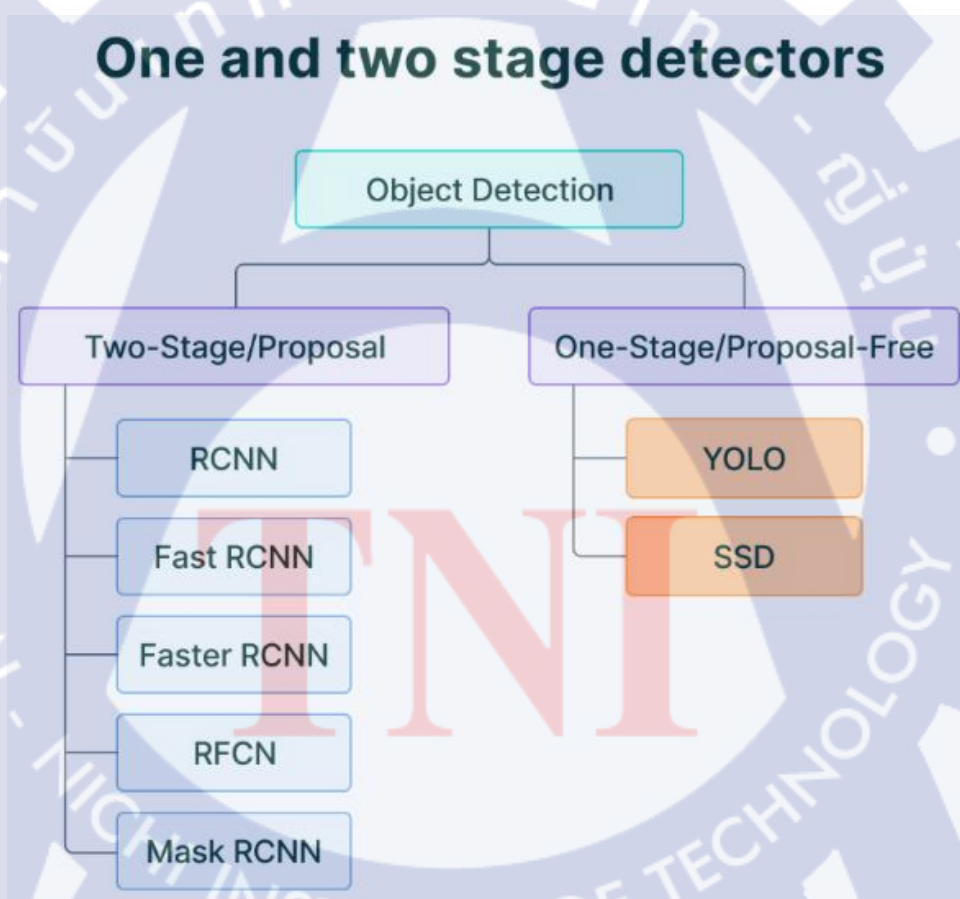
คอมพิวเตอร์วิทัศน์ เริ่มต้นขึ้นในปี ค.ศ. 1950 [23] จากการใช้ โครงข่ายประสาทเทียม (Neural Network) ในการจัดหมวดหมู่ให้แก่วัตถุอย่างง่าย ๆ เช่น รูปทรงสามเหลี่ยม สีเหลี่ยม วงกลม ด้วยการค้นหาขอบและมุม ซึ่งยังไม่แม่นยำและผิดพลาดสูง ในปี ค.ศ. 1970 [24] ได้มีการนำเอาเทคนิคประมวลผลที่เรียกว่า Optical Character Recognition [25] คือ กระบวนการที่แปลงภาพข้อความให้เป็นข้อความดิจิทัลที่สามารถแก้ไขและค้นหาได้ มาใช้ในการตีความตัวหนังสือหรือข้อความ และมีการพัฒนาอย่างรวดเร็ว เมื่อมีการพัฒนาของระบบอินเทอร์เน็ต จึงมีการอัปโหลดข้อมูลรูปภาพต่างๆ อยู่ตลอดเวลาจำนวนมาก ทำให้สามารถวิเคราะห์ข้อมูลและจดจำภาพถ่ายได้จำนวนมาก ในปัจจุบันมีการศึกษาด้านการตรวจจับวัตถุมากยิ่งขึ้น ส่งผลให้เกิดการพัฒนาการตรวจจับใบหน้าแบบทันที ตรวจจบบรรณดแบบทันที ทำให้งานทางด้านคอมพิวเตอร์วิทัศน์มีการใช้งานอย่างกว้างขวางในปัจจุบัน เช่น ในยานยนต์ไร้คนขับ เซ็นเซอร์รักษาความปลอดภัย หุ่นยนต์ และงานประยุกต์อื่น ๆ คอมพิวเตอร์วิทัศน์ช่วยให้คอมพิวเตอร์สามารถเข้าใจโลกรอบตัวได้มากขึ้น และช่วยให้คอมพิวเตอร์สามารถทำงานได้อย่างอัตโนมัติมากขึ้น จากงานประยุกต์ต่าง ๆ จึงทำให้คอมพิวเตอร์วิทัศน์เป็นสาขาที่กำลังพัฒนาอย่างรวดเร็ว และเทคโนโลยีคอมพิวเตอร์วิทัศน์กำลังถูกนำมาใช้ใหม่ ๆ อยู่เสมอ

ตัวอย่างการใช้งานคอมพิวเตอร์วิทัศน์ เช่น

- ยานยนต์ไร้คนขับใช้คอมพิวเตอร์วิทัศน์เพื่อตรวจจับวัตถุในถนน เช่น รถยนต์ จักรยาน ผู้คน และสัญญาณไฟจราจร [26]
- เซ็นเซอร์รักษาความปลอดภัยใช้คอมพิวเตอร์วิทัศน์เพื่อตรวจจับการเคลื่อนไหวของบุคคลหรือวัตถุ [27]
- หุ่นยนต์ใช้คอมพิวเตอร์วิทัศน์เพื่อนำทางและทำงาน [28]
- การตรวจจับใบหน้าใช้คอมพิวเตอร์วิทัศน์เพื่อระบุตัวบุคคล [29]

2.2 การตรวจจับวัตถุ

การตรวจจับวัตถุ เป็นสาขาหนึ่งของคอมพิวเตอร์วิทัศน์ที่ศึกษาเกี่ยวกับการระบุวัตถุในภาพหรือวิดีโอ การตรวจจับวัตถุเป็นกระบวนการที่ซับซ้อนเนื่องจากภาพหรือวิดีโออาจมีวัตถุหลายประเภทและวัตถุแต่ละประเภทอาจมีรูปร่างและขนาดที่แตกต่างกัน อัลกอริธึมการตรวจจับวัตถุที่ดีต้องสามารถระบุวัตถุได้แม่นยำและระบุชนิดของวัตถุได้อย่างถูกต้อง การระบุตำแหน่งของวัตถุในภาพหรือวิดีโอ โดยใช้อัลกอริธึมการตรวจจับวัตถุมักจะใช้ประโยชน์จากการเรียนรู้ของเครื่องหรือการเรียนรู้เชิงลึกเพื่อสร้างผลลัพธ์ที่มีความหมาย งานวิจัยประเภทการตรวจจับรถยนต์ในถนน มีด้วยกันหลากหลายวิธี แต่การตรวจจับรถยนต์ขนาดเล็กและภาพความละเอียดต่ำจากโดรนนั้น มีความสามารถในการตรวจจับวัตถุได้ไม่ดีเท่าที่ควรขึ้นอยู่กับปัจจัยหลายอย่าง ตัวอย่างเทคนิคการตรวจจับวัตถุในภาพหรือวิดีโอ มีด้วยกันหลายชนิด ดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างประเภทของอัลกอริธึมการตรวจจับวัตถุ

การตรวจจับวัตถุดังรูปที่ 2.1 มี 2 ประเภท คือ การตรวจจับวัตถุแบบขั้นตอนเดียว และการตรวจจับวัตถุแบบสองขั้นตอน [30] เป็นอัลกอริธึมการตรวจจับวัตถุทั้ง 2 ประเภทได้รับความนิยมในปัจจุบัน การตรวจจับวัตถุแบบขั้นตอนเดียว ประกอบด้วยเทคนิคต่าง ๆ เช่น YOLO, Single Shot MultiBox Detector (SSD) ซึ่งมีขั้นตอนทำงานโดยการตรวจจับวัตถุในภาพพร้อมกันโดยไม่ต้องแบ่งภาพออกเป็นบริเวณหรือพื้นที่ที่สนใจ (Region)

ขั้นตอนหลักของการตรวจจับวัตถุแบบขั้นตอนเดียว ได้แก่

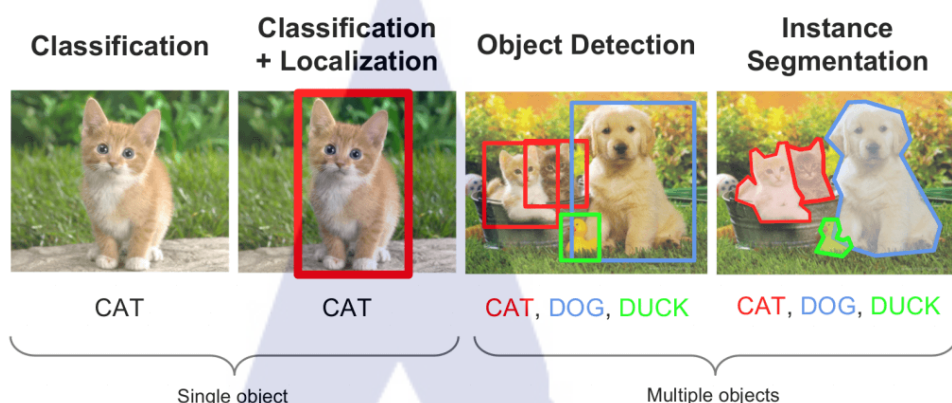
- การแยกคุณลักษณะ (Feature Extraction) ขั้นตอนนี้ใช้สำหรับวิเคราะห์คุณสมบัติที่สำคัญจากภาพ โดยใช้เทคนิคการเรียนรู้ของเครื่อง เช่น CNN
- การตรวจจับวัตถุ (Object Detection) ขั้นตอนนี้ใช้สำหรับตรวจจับวัตถุในภาพ โดยใช้เทคนิคการเรียนรู้ของเครื่อง เช่น regression

วิธีการตรวจจับวัตถุแบบขั้นตอนเดียว มีข้อดีคือมีความสามารถสูงและสามารถตรวจจับวัตถุได้อย่างรวดเร็ว เนื่องจากวิธีการนี้ไม่จำเป็นต้องแบ่งภาพออกเป็นภูมิภาคก่อน จึงช่วยลดจำนวนการคำนวณลงได้ จุดด้อยของวิธีการนี้คือ ความแม่นยำในการระบุตำแหน่งของวัตถุอาจไม่สูงเท่าวิธีการตรวจจับวัตถุแบบสองขั้นตอน

การตรวจจับวัตถุแบบสองขั้นตอน (Two-Stage detector) เช่น RCNN, Fast RCNN, Faster RCNN, Mask RCNN ทำงานโดยการแบ่งภาพออกเป็นบริเวณขนาดเล็ก และจะใช้การเรียนรู้ของเครื่องเพื่อจำแนกประเภทของวัตถุในบริเวณนั้นๆ ขั้นตอนหลักของการตรวจจับวัตถุแบบสองขั้นตอน ได้แก่

- ข้อเสนอแนะบริเวณ (Region Proposal) ขั้นตอนนี้ใช้สำหรับการสร้างบริเวณที่น่าจะเป็นไปได้ที่จะพบวัตถุในภาพ วิธีการสร้างบริเวณอาจแตกต่างกันไป เช่น Selective Search, Edge Boxes, RPN (Region Proposal Network)
- การแยกประเภท (Classification) ขั้นตอนนี้ใช้สำหรับจำแนกประเภทของวัตถุในบริเวณโดยใช้การเรียนรู้ของเครื่อง เช่น CNN
- การระบุตำแหน่ง (Localization) ขั้นตอนนี้ใช้สำหรับระบุตำแหน่งของวัตถุในบริเวณโดยใช้การเรียนรู้ของเครื่อง เช่น regression

วิธีการตรวจจับวัตถุแบบสองขั้นตอน มีข้อดี คือ วิธีนี้จะมีมีความแม่นยำในการระบุตำแหน่งของวัตถุสูง เนื่องจากสามารถประมวลผลแยกกันแต่ละบริเวณ จึงทำให้สามารถตรวจจับวัตถุที่มีขนาดเล็กและตำแหน่งที่ซับซ้อนได้ดีกว่าวิธีการตรวจจับวัตถุแบบขั้นตอนเดียว อย่างไรก็ตามวิธีการตรวจจับวัตถุแบบสองขั้นตอน อาจมีประสิทธิภาพต่ำและใช้เวลาในการประมวลผลนานกว่าวิธีการตรวจจับวัตถุแบบขั้นตอนเดียว



รูปที่ 2.2 ตัวอย่างการตรวจจำวัตถุในรูปภาพ [2]

อัลกอริธึมการตรวจจำวัตถุมีหลายประเภท และมีขั้นตอนการทำงานดังนี้ แยกแยะวัตถุในภาพหรือวิดีโอ, ระบุขอบเขตของวัตถุ, ระบุชนิดของวัตถุ

ตัวอย่างวิธีการตรวจจำวัตถุแบบเดี่ยวดังรูปที่ 2.2 ได้แก่

- วิธีการแยกประเภท (Classification)

คือกระบวนการที่คอมพิวเตอร์สามารถระบุประเภทของวัตถุในภาพหรือวิดีโอได้ ตัวอย่างเช่น โมเดลการแยกประเภท สามารถระบุได้ว่าวัตถุในภาพเป็นสุนัข แมว หรือรถยนต์ เป็นต้น

- วิธีการแยกประเภทและการระบุตำแหน่ง (Classification+Localization)

คือกระบวนการที่คอมพิวเตอร์สามารถระบุประเภทของวัตถุและตำแหน่งของวัตถุในภาพหรือวิดีโอได้ ตัวอย่างเช่น สามารถระบุได้ว่าวัตถุในภาพเป็นสุนัข และตำแหน่งของสุนัขนั้นอยู่ตรงกลางภาพ

ตัวอย่างวิธีการตรวจจำวัตถุแบบหลายวัตถุมีวิธีการตรวจจำดังนี้

- วิธีการตรวจจำวัตถุ

คือกระบวนการที่คอมพิวเตอร์สามารถระบุและตรวจจำวัตถุที่อยู่ในภาพหรือวิดีโอได้ ตัวอย่างเช่น สามารถระบุได้ว่าวัตถุในภาพเป็นสุนัข แมว และรถยนต์ เป็นต้น

- วิธีการแยกวัตถุ (Instance segmentation) [31]

คือกระบวนการที่คอมพิวเตอร์สามารถระบุและตรวจจำวัตถุต่าง ๆ ที่อยู่ในภาพหรือวิดีโอได้ โดยแยกแต่ละวัตถุออกจากกัน เช่น สามารถระบุวัตถุในภาพว่าเป็นสุนัข แมว หรือรถยนต์ โดยแยกแต่ละวัตถุออกจากกัน

การตรวจจำวัตถุมีการใช้งานอย่างกว้างขวางในปัจจุบัน การตรวจจำวัตถุช่วยให้คอมพิวเตอร์สามารถเข้าใจโลกรอบตัวได้มากขึ้น และช่วยให้คอมพิวเตอร์สามารถทำงานได้อย่าง

อัตโนมัติมากขึ้น ตัวอย่างการใช้งานการตรวจจับวัตถุ เช่น ยานยนต์ไร้คนขับจะใช้ตรวจจับวัตถุได้แก่ รถยนต์ จักรยาน ผู้คน และสัญญาณไฟจราจร เช่น เซอร์รักษาความปลอดภัยใช้การตรวจจับวัตถุเพื่อตรวจจับการเคลื่อนไหวของบุคคลหรือวัตถุ หุ่นยนต์ใช้การตรวจจับวัตถุเพื่อนำทางและทำงาน การตรวจจับใบหน้าใช้การตรวจจับวัตถุเพื่อระบุตัวบุคคล การตรวจจับวัตถุเป็นสาขาที่กำลังพัฒนาอย่างรวดเร็ว และมีการนำเทคโนโลยีการตรวจจับวัตถุ มาใช้ในงานประยุกต์ใหม่ ๆ อยู่เสมอ

2.3 การประมวลผลภาพ

การประมวลผลภาพ (Image processing) [32] เป็นกระบวนการที่ใช้คอมพิวเตอร์ในการวิเคราะห์และปรับปรุงภาพดิจิทัลหรือวิดีโอ เพื่อให้ได้ผลลัพธ์ที่ต้องการ หรือการพัฒนาคุณภาพของภาพ การประมวลผลภาพจะสามารถลดสัญญาณรบกวน, ปรับแต่งความคมชัด, หรือตรวจจับวัตถุในภาพ การประมวลผลภาพเป็นกระบวนการที่สำคัญในคณิตศาสตร์, วิทยาศาสตร์คอมพิวเตอร์, และวิศวกรรมที่ใช้ในหลากหลายงาน เช่น การประมวลผลภาพสามารถใช้ในการเพิ่มความชัดของรูปภาพ เพื่อให้ทำให้เห็นวัตถุชัดเจนมากขึ้น, ลดสัญญาณรบกวนที่ไม่ต้องการในภาพ, หรือการระบุวัตถุที่อยู่ในภาพ ภาพดิจิทัลเป็นภาพที่สร้างจากจุดเล็กๆ ที่เรียกว่า พิกเซล (pixel) พิกเซลแต่ละจุดมีค่าความสว่างหรือสีที่กำหนด ซึ่งกำหนดลักษณะของภาพ กระบวนการการประมวลผลภาพจะดำเนินการกับค่าความสว่างหรือสีของพิกเซลแต่ละจุดเพื่อปรับปรุงหรือเปลี่ยนแปลงภาพ กระบวนการประมวลผลภาพมักใช้เทคนิคและอัลกอริทึมต่าง ๆ เพื่อใช้ปรับแต่งภาพตามวัตถุประสงค์ ได้แก่ การใช้วิธีการกรองภาพ (Image Filtering) [33] เพื่อลดสัญญาณรบกวน, การแก้ไขสีและความคมชัดของภาพ, การตรวจจับวัตถุ และการแก้ไขลักษณะเฉพาะของภาพ เช่น การตรวจจับขอบ (Edge Detection) [34] และการตรวจจับสี (Color Detection)

การตรวจจับวัตถุจะเน้นการค้นหและการนำเสนอคุณสมบัติเฉพาะของวัตถุ (Features) ที่มีความสำคัญเพื่อตรวจจับและระบุวัตถุในรูปภาพ ขั้นตอนสำคัญในการแยกประเภทวัตถุและทำให้ระบบสามารถรู้จักวัตถุในภาพได้ คุณสมบัติหลายประเภทที่สามารถนำมาใช้ในการตรวจจับวัตถุ ให้เหมาะสมกับงานหรือวัตถุที่แตกต่างกัน ตัวอย่างของคุณสมบัติเฉพาะที่นิยมนำมาใช้ ได้แก่

- การหาขอบ (Edges) เทคนิคการหาขอบในรูปภาพช่วยในการระบุขอบเขตของวัตถุและการแยกแยะวัตถุจากพื้นหลัง ขอบสามารถเป็นเป็นเส้นที่แยกความเข้มหรือความสว่างในรูปภาพ
- การจำแนกลักษณะเฉพาะสี (Color) การหาคุณสมบัติสีสามารถช่วยในการระบุวัตถุ โดยการตรวจจับสีหรือรูปแบบสีที่เฉพาะของวัตถุ
- การแยกแยะรูปร่างของวัตถุ (Shape) ในบางกรณี รูปร่างของวัตถุอาจมีลักษณะเฉพาะที่ช่วยในการระบุ

- พื้นผิว (Texture) คุณสมบัติพื้นผิวมักถูกนำมาใช้ในการตรวจจับวัตถุที่มีลวดลายหรือลักษณะพื้นผิวเฉพาะ

ดังนั้นการประมวลผลภาพที่ใช้ในการตรวจจับวัตถุ จะมีขั้นตอนเริ่มต้นด้วยการแยกคุณสมบัติดังที่กล่าวข้างต้น จากรูปภาพด้วยเทคนิคการกรอง (Filter) และการแปลงรูปภาพ โดยคุณสมบัติเหล่านี้สามารถนำมาใช้ในการสร้างโมเดลตรวจจับวัตถุได้

2.4 อัลกอริทึมในงานการตรวจจับวัตถุ

การตรวจจับวัตถุด้วยการใช้อัลกอริทึม (Algorithm) ได้พัฒนาเพื่อให้คอมพิวเตอร์สามารถตรวจจับวัตถุในภาพได้มีหลายวิธี โดยบางวิธีจะใช้เทคโนโลยีการเรียนรู้เชิงลึก ที่มีประสิทธิภาพสูง และบางวิธีใช้การประมวลผลภาพ ที่ใช้อัลกอริทึมที่ซับซ้อน อัลกอริทึม คือ ชุดของขั้นตอนในการดำเนินการเพื่อแก้ปัญหาที่กำหนด อัลกอริทึมที่ใช้ในงานการตรวจจับมักจะเป็นอัลกอริทึมการประมวลผลภาพ เช่น อัลกอริทึม Edge-Based Algorithm [35] , Region-Based Algorithm [36] , Template Matching Algorithm [37]

อัลกอริทึมที่ใช้ในงานการตรวจจับวัตถุ แบ่งออกเป็น 2 ประเภท ได้แก่

2.4.1 อัลกอริทึมแบบคลาสสิก (Classical Algorithm)

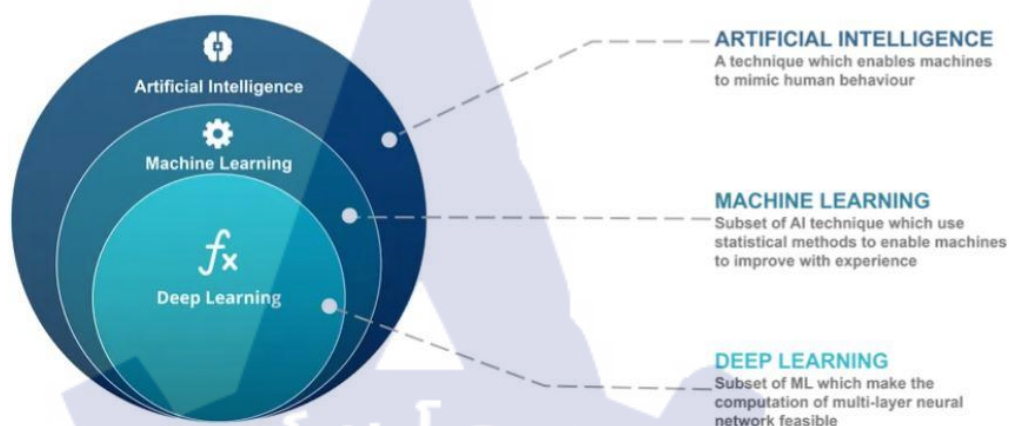
เป็นอัลกอริทึมที่พัฒนาขึ้นก่อนยุคการเรียนรู้ของเครื่อง อาศัยการหาคุณสมบัติเฉพาะของวัตถุ (Feature) มาใช้ในการตรวจจับ เช่น รูปร่าง ขอบเขต พื้นผิว สี เป็นต้น อัลกอริทึมแบบคลาสสิกมีดังนี้

- 1) Edge-Based Algorithm เป็นอัลกอริทึมที่อาศัยการหาขอบเขตของวัตถุมาใช้ในการตรวจจับ ตัวอย่างเช่น อัลกอริทึม Canny Edge Detector [38]
- 2) Region-Based Algorithm เป็นอัลกอริทึมที่อาศัยการหาพื้นที่ของวัตถุมาใช้ในการตรวจจับ ตัวอย่างเช่น อัลกอริทึม Hough Transform [39]
- 3) Template Matching Algorithm เป็นอัลกอริทึมที่อาศัยการเปรียบเทียบภาพกับเทมเพลตที่เก็บไว้ของวัตถุมาใช้ในการตรวจจับ ตัวอย่างเช่น อัลกอริทึม Brute-Force Matching [40]

2.4.2 อัลกอริทึมแบบการเรียนรู้ของเครื่อง (Machine Learning Algorithm)

เป็นอัลกอริทึมที่พัฒนาขึ้นหลังยุคการเรียนรู้ของเครื่อง ความรู้จากข้อมูลทำให้คอมพิวเตอร์มีความสามารถในการปรับปรุงประสิทธิภาพและการดำเนินงานอย่างอัตโนมัติโดยพัฒนาโมเดลและวิธีการ โดยอ้างอิงข้อมูลในขณะที่มีการเรียนรู้ ดังรูปที่ 2.3 การเลือกใช้อัลกอริทึม ควรพิจารณา

ประสิทธิภาพของอัลกอริทึม การทำงานที่ต้องการ และลักษณะของข้อมูลที่มีอยู่ในงานวิจัยเพื่อให้การเลือกใช้เป็นไปอย่างเหมาะสมและมีประสิทธิภาพสูงสุดในบริบทเฉพาะของงาน



รูปที่ 2.3 การเรียนรู้ของปัญญาประดิษฐ์ [5]

งานวิจัยนี้จะใช้อัลกอริทึมแบบการเรียนรู้ของเครื่องที่อาศัยโมเดลการเรียนรู้ของเครื่อง ในการจำแนกประเภทและระบุตำแหน่งของวัตถุ ที่เรียกว่า อัลกอริทึมการเรียนรู้เชิงลึก (Deep Learning Algorithms) โครงสร้างอัลกอริทึมสำหรับการเรียนรู้เชิงลึกที่ใช้โครงข่ายประสาทเทียม (Neural Networks) ใหญ่ ๆ เช่น Convolutional Neural Networks (CNN) สำหรับการประมวลผลภาพ

อัลกอริทึมแบบการเรียนรู้เชิงลึกของเครื่องมีดังนี้

1) Convolutional Neural Network (CNN) เป็นโมเดลการเรียนรู้ของเครื่องที่เกี่ยวข้องกับการทำงานทางรูปภาพ ที่ได้รับความนิยมมากที่สุดในงานการตรวจจับวัตถุ CNN อาศัยคุณสมบัติของวัตถุที่ได้จากฟิลเตอร์ในการจำแนกประเภทและระบุตำแหน่งของวัตถุ

2) Region-Based Convolutional Neural Network (R-CNN) เป็นโมเดล CNN ที่ถูกพัฒนาขึ้นเพื่อปรับปรุงประสิทธิภาพของ CNN ในการตรวจจับวัตถุ R-CNN แบ่งขั้นตอนการทำงานออกเป็น 2 ขั้นตอนหลัก ๆ คือ การหาขอบเขตของวัตถุ และการจำแนกประเภทและระบุตำแหน่งของวัตถุ

3) Faster R-CNN เป็นโมเดล R-CNN ที่พัฒนาขึ้นเพื่อปรับปรุงประสิทธิภาพของโมเดลในการหาขอบเขตของวัตถุ Faster R-CNN อาศัยเทคนิคที่เรียกว่า Region Proposal Network (RPN) ในการหาขอบเขตของวัตถุ

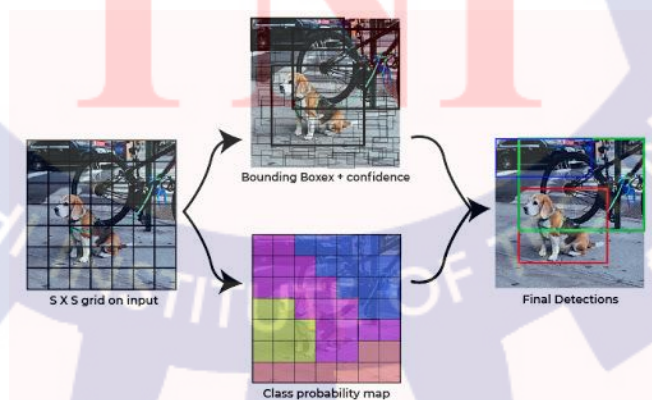
4) Single Shot MultiBox Detector (SSD) เป็นโมเดล CNN ที่พัฒนาขึ้นเพื่อปรับปรุงประสิทธิภาพในการประมวลผลภาพอย่างรวดเร็ว SSD อาศัยเทคนิคที่เรียกว่า Single Shot MultiBox Detector ในการระบุวัตถุหลายประเภทในภาพเดียว

5) YOLO เป็นอัลกอริทึมตรวจจับวัตถุแบบขั้นตอนเดียว ที่ทำทุกอย่างในครั้งเดียว โดยที่ไม่ต้องแบ่งขั้นตอนการค้นหาและการคลาดเคลื่อน (Refinement) ออกจากกัน เป็นวิธีการซึ่งจะแบ่งภาพเป็นกรอบ และทำการตรวจจับในแต่ละกรอบ โดยสามารถทำงานอย่างรวดเร็วและมีความแม่นยำสูง โมเดลของ YOLO เป็นโมเดล CNN ที่พัฒนาขึ้นเพื่อปรับปรุงประสิทธิภาพในการประมวลผลภาพอย่างรวดเร็ว YOLO แบ่งภาพออกเป็นเซลล์ขนาดเล็ก และประเมินความน่าจะเป็นของวัตถุแต่ละประเภทในเซลล์แต่ละเซลล์ YOLO สามารถทำงานด้วยความเร็วสูง ทำให้เหมาะสำหรับการตรวจจับและติดตามวัตถุในเวลาอัตราเฟรม (Frame Rate) สูง เช่น งานที่ใช้กล้องบนโดรนหรือกล้องวิดีโอในระบบการควบคุมความปลอดภัยทางจราจร จึงทำให้ผู้วิจัยได้เลือกใช้อัลกอริทึม YOLO มาใช้ในการพัฒนาการตรวจจับรถยนต์ด้วยภาพความละเอียดต่ำจากโดรน

2.5 อัลกอริทึม YOLO (You Only Look Once)

YOLO (You Only Look Once) เป็นอัลกอริทึมการตรวจจับวัตถุ ที่ถูกพัฒนาโดย Joseph Redmon และเพื่อนร่วมงานที่ UC Berkeley [41] ในปี 2015 YOLO เป็นอัลกอริทึมการตรวจจับวัตถุแบบเรียลไทม์ที่มีประสิทธิภาพสูง โดยสามารถตรวจจับวัตถุได้เร็วกว่าอัลกอริทึมการตรวจจับวัตถุแบบดั้งเดิม เช่น R-CNN และ Faster R-CNN

YOLO มีขั้นตอนการทำงานโดยแบ่งภาพออกเป็นเซลล์ (grid) จากนั้นจะประเมินความน่าจะเป็นของวัตถุแต่ละประเภทที่ปรากฏอยู่ในเซลล์แต่ละเซลล์ นอกจากนี้ YOLO ยังประเมินความแม่นยำของตำแหน่งของวัตถุในเซลล์แต่ละเซลล์ ดังรูปที่ 2.4



รูปที่ 2.4 กระบวนการทำงานของอัลกอริทึม YOLO [7]

คุณสมบัติสำคัญของ YOLO คือการทำนายตำแหน่งของวัตถุทั้งหมดในรูปภาพในขั้นตอนเดียว ด้วยการสแกนหรือการค้นหาข้ามขอบเช่นเดียวกับแบบจำลองอื่น ๆ ที่ต้องทำหลายครั้ง เช่น Faster R-CNN หรือ SSD (Single Shot MultiBox Detector) ที่ใช้การสแกนแบบละเอียดเพื่อตรวจหาวัตถุที่แตกต่างกันในขั้นตอนแรก ต่อจากนั้นจึงค้นหาขอบของวัตถุและจัดกลุ่มในขั้นตอนถัดไป ซึ่งทำให้ YOLO มีความเร็วกว่าแบบจำลองอื่นในงานตรวจหาวัตถุ YOLO สามารถทำงานในรูปแบบหลายรุ่น (versions) อัลกอริทึม YOLO นี้จะมีการฝึกสอนโมเดลสำหรับการตรวจจับวัตถุเพื่อทำให้สามารถตรวจจับวัตถุแบบทันทีในวิดีโอมีความแม่นยำสูง อัลกอริทึมตระกูล YOLO นี้เหมาะสำหรับงานตรวจจับวัตถุแบบทันทีในวิดีโอโดยใช้เทคนิคตัวตรวจจับหนึ่งขั้นตอน (Single Stage Detection) เช่น YOLO v1, v2, v3 และ SSD และวิธีการอื่นของการตรวจจับวัตถุ คือ ตัวตรวจจับแบบสองขั้นตอน (Two-Stage Detection) เช่น RCNN, Fast RCNN และ Faster RCNN [42], [43], [44]

ข้อดีของอัลกอริทึม Fast YOLO [45] คือด้านความเร็วในการตรวจจับวัตถุ แต่ความสามารถอาจลดลงสำหรับวัตถุเป้าหมายขนาดเล็กและหนาแน่น ดังนั้นผู้วิจัยจึงเล็งเห็นการพัฒนาโดยเลือกพัฒนาเทคนิคการตรวจจับวัตถุ (Object Detection) สำหรับตรวจจับรถยนต์ด้วยภาพขนาดเล็กที่มีความละเอียดต่ำขนาดเล็กในรูปแบบการทำงานแบบตัวตรวจจับขั้นตอนเดียว (One-Stage Detectors) [46] ที่สามารถถูกปรับแต่งให้เหมาะสมสำหรับการตรวจจับรถยนต์ในภาพความละเอียดต่ำ โมเดลนี้จะสามารถระบุตำแหน่งของรถยนต์ในภาพและสร้างกล่องขอบเขต (Bounding boxes) รอบรถยนต์เพื่อใช้ในกระบวนการตรวจจับได้อย่างรวดเร็ว ระบบจะนำเสนอผลลัพธ์โดยแสดงในกล่องขอบเขต (Bounding boxes) ที่ระบุตำแหน่งของรถยนต์ในภาพ และแสดงคะแนนความน่าจะเป็น (Probabilities scores) ซึ่งบ่งชี้ถึงความมั่นใจในการตรวจจับของแต่ละกล่องขอบเขต

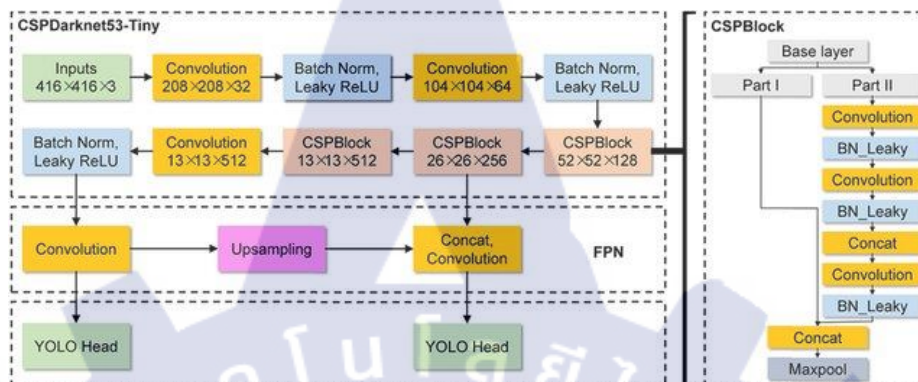
การนำอัลกอริทึมของ YOLO ไปใช้ในการตรวจจับวัตถุได้ถูกพัฒนาและถูกนำไปใช้อย่างต่อเนื่องหลายเวอร์ชัน เพื่อให้อัลกอริทึมมีประสิทธิภาพในการตรวจจับวัตถุที่แม่นยำและรวดเร็วมากยิ่งขึ้น ในปี 2022 งานวิจัยของ L. Zheng และคณะ [47] ได้นำโมเดลเวอร์ชัน YOLOv3 ในการประมวลผลภาพความละเอียดต่ำจากโดรน โมเดลนี้ได้รับการฝึกอบรมบนชุดข้อมูลภาพรถยนต์ขนาดเล็กจากโดรนจำนวน 1,000 ภาพ และทดสอบบนชุดข้อมูลทดสอบจำนวน 200 ภาพ ผลการทดสอบพบว่าโมเดลนี้สามารถตรวจจับรถยนต์ขนาดเล็กได้แม่นยำถึง 90% งานวิจัยนี้แสดงให้เห็นว่าโมเดลตรวจจับรถยนต์ขนาดเล็กแบบขั้นตอนเดียวสามารถใช้สำหรับตรวจจับรถยนต์ขนาดเล็กในภาพความละเอียดต่ำจากโดรนได้และ YOLOv3 มีความสามารถในการตรวจจับเหนือกว่า Faster R-CNN ในด้านความไวและเวลาในการประมวลผล ส่วนด้านเมตริกความแม่นยำมีค่าใกล้เคียงกัน นอกจากนี้ยังเปรียบเทียบความสามารถของ Faster R-CNN และ YOLOv3 ในแง่ของความถูกต้อง (Accuracy)

และ ระยะเวลาในการประมวลผล (Processing time) [48] สำหรับการระบุรถยนต์จากภาพถ่ายทางอากาศ ซึ่งโมเดลเหล่านี้มีข้อดีคือ สามารถทำงานได้อย่างรวดเร็วและมีประสิทธิภาพบนอุปกรณ์ที่มีทรัพยากรจำกัด เช่น โดรน ทำให้การพัฒนาโมเดลที่ใช้กรอบเทคนิคของ Convolutional Neural Network [49] นั้นมีปัจจัยที่สำคัญหลายประการต่อการพัฒนาความสามารถของโมเดล

นอกจากนี้ YOLOv4 มีรุ่นย่อยที่เร็วกว่าคือ YOLOv4-Tiny (You Only Look Once Version 4 Tiny) ที่มีความแม่นยำน้อยลงแต่มีความเร็วสูง ทำให้เหมาะสำหรับการใช้งานในแวดวงที่ต้องการการตรวจหาวัตถุแบบทันที และที่มีทรัพยากรจำกัด เช่น การตรวจหาวัตถุในวิดีโอจากกล้องวงจรปิดหรือจากโดรน โดยสามารถตรวจจับวัตถุได้เร็วกว่าอัลกอริทึมการตรวจจับวัตถุแบบดั้งเดิมมาก YOLO เป็นหนึ่งในอัลกอริทึมการตรวจจับวัตถุที่ได้รับความนิยมมากที่สุดในปัจจุบัน ข้อดีของ YOLO ได้แก่ มีประสิทธิภาพสูงมาก เร็วกว่าอัลกอริทึมการตรวจจับวัตถุแบบดั้งเดิมมาก และยังสามารถตรวจจับวัตถุได้หลายประเภทพร้อมกัน ข้อเสียของ YOLO ได้แก่ การเกิดข้อผิดพลาดในการระบุตำแหน่งของวัตถุในภาพ และการที่ไม่สามารถตรวจจับวัตถุที่ซับซ้อนหรือเล็กมากได้

YOLOv4-Tiny เป็นโมเดลการตรวจจับวัตถุที่มีประสิทธิภาพสูงและมีขนาดเล็ก โดยเป็นรุ่นย่อยของ YOLOv4 ซึ่งอยู่ในตระกูลของโมเดล YOLO ซึ่งเป็นโมเดลการตรวจจับวัตถุที่ใช้เทคโนโลยีการเรียนรู้เชิงลึก ที่มีความแม่นยำและความเร็วสูงในการตรวจจับวัตถุในภาพหรือวิดีโออัตโนมัติ โมเดล YOLOv4-Tiny ถูกออกแบบให้สามารถทำงานได้บนระบบที่มีทรัพยากรจำกัด เช่น คอมพิวเตอร์เครื่องจักรหรือโดรนที่มีข้อจำกัดในด้านความสามารถของฮาร์ดแวร์และการประมวลผลที่จำกัด YOLOv4-Tiny เป็นโมเดลขนาดเล็กกว่า YOLOv4 ดั้งเดิมมาก โดยมีขนาดเพียง 16MB เท่านั้น จึงทำให้สามารถฝึกและใช้งานบนอุปกรณ์ที่มีทรัพยากรการคำนวณจำกัดได้ ทำให้เหมาะสำหรับการใช้งานในอุปกรณ์ที่มีทรัพยากรจำกัด เช่น โดรนหรือคอมพิวเตอร์เคลื่อนที่ ซึ่งถูกพัฒนาขึ้นเพื่อให้การตรวจหาวัตถุมีความเร็วสูง และเหมาะสำหรับการใช้งานที่มีความต้องการการตรวจหาแบบทันที ที่มีความแม่นยำมากยิ่งขึ้น การกำหนดขนาดอินพุต (input size) สำหรับการฝึกโมเดล YOLOv4-Tiny หรือ โมเดลการตรวจจับวัตถุด้วยภาพความละเอียดต่ำจากโดรน ควรพิจารณาความสมดุลระหว่างขนาดอินพุตของโมเดลและความเร็วในการฝึกสอนสำหรับการนำไปใช้งานจริง การกำหนดขนาดอินพุตของโมเดลเป็นขั้นตอนที่สำคัญ เพราะมีผลต่อประสิทธิภาพในการตรวจจับและการทำงานในเวลาจริง ขนาดอินพุต (input size) สำหรับการฝึกโมเดล YOLOv4-Tiny ที่เหมาะสมขึ้นอยู่กับโครงสร้างของโมเดลและข้อมูลที่ใช้ในการฝึกสอน แต่มักจะอยู่ในช่วงขนาดประมาณ 416x416 pixels ถึง 608x608 pixels โดยทั่วไปขนาด 416x416 pixels ถือเป็นขนาดที่เริ่มต้น ที่เหมาะสมสำหรับการฝึกสอนโมเดล YOLOv4-Tiny และใช้งานได้อย่างดีในสถานการณ์สำหรับตรวจจับวัตถุทั่วไป

YOLOv4-Tiny ทำงานโดยแบ่งภาพออกเป็นเซลล์ (grid) 3×3 และจะประเมินความน่าจะเป็นของวัตถุแต่ละประเภทที่ปรากฏอยู่ในเซลล์แต่ละเซลล์ รวมทั้งประเมินความแม่นยำของตำแหน่งของวัตถุในเซลล์แต่ละเซลล์ด้วย



รูปที่ 2.5 ระบบโครงข่ายของโมเดล YOLOv4-Tiny [8]

ระบบโครงข่ายของ YOLOv4-Tiny ดังรูปที่ 2.5 มีส่วนประกอบหลัก 3 ส่วน ได้แก่

1) ส่วนหลัก (Backbone) ส่วนนี้ทำหน้าที่ในการประมวลผลภาพเบื้องต้นและทำการสร้างคุณสมบัติของภาพ โมเดล YOLOv4-Tiny มีส่วนหลักเป็นแบบ CSPDarknet [50] ซึ่งเป็นส่วนหลักที่ได้รับการปรับปรุงจาก Darknet53 โดยใช้การเชื่อมต่อแบบ Cross Stage Partial (CSP) เพื่อลดจำนวนพารามิเตอร์และปรับปรุงความสามารถ ในส่วนนี้เป็นส่วนหลักของโครงสร้างโดยที่เน้นความเร็วและความสามารถในการคำนวณ

2) ส่วนหัวของ YOLO (YOLO Heads) ส่วนนี้ทำหน้าที่ในการตรวจจับวัตถุ YOLOv4-Tiny ใช้ส่วนหัวแบบสองหัว ซึ่งแต่ละหัวจะรับผิดชอบในการตรวจจับวัตถุสองขนาดที่แตกต่างกัน หัวแต่ละหัวจะมีเซลล์การรวม (Convolutional Cells) สองเซลล์ โดยเซลล์แรกจะทำหน้าที่ในการจำแนกวัตถุ และเซลล์ที่สองจะทำหน้าที่ในการประมาณกรอบของวัตถุ

3) ส่วนคอ (Neck) ส่วนนี้ทำหน้าที่ในการเชื่อมต่อข้อมูลจากส่วนหลักไปยังส่วนหัวของ YOLO โมเดล YOLOv4-Tiny ใช้ส่วนคอแบบ FPN (Feature Pyramid Network) เพื่อนำเอาคุณสมบัติของภาพจากระดับต่างๆ มารวมกันส่วนคอของ YOLOv4-Tiny ประกอบด้วย Convolutional Layers สามชั้น ซึ่งทำหน้าที่ในการเชื่อมต่อข้อมูลจากส่วนหลักไปยังส่วนหัวของ YOLO ซึ่งจะสร้างคุณสมบัติของภาพในขนาดต่างๆ

YOLOv4-Tiny เป็นหนึ่งในโมเดลการตรวจจับวัตถุที่ได้รับความนิยมมากที่สุดในปัจจุบัน และ YOLOv4-Tiny มีการประยุกต์ใช้งานในหลายประเภท เช่น การตรวจหาวัตถุในวิดีโอจากกล้องวงจรปิด (CCTV) [51] การตรวจหาคนเดินข้ามถนน [52] ข้อดีของ YOLOv4-Tiny คือมีขนาดเล็กมาก และเหมาะสำหรับการใช้งานบนอุปกรณ์ที่มีทรัพยากรการคำนวณจำกัด ต้องการการใช้การประมวลผลที่รวดเร็ว มีประสิทธิภาพสูง และยังสามารถตรวจจับวัตถุได้หลายประเภทพร้อมกัน เป็นต้น

ความแตกต่างระหว่าง YOLOv4-Tiny และ YOLOv4 ดั้งเดิม มีรายละเอียดแสดงดังตารางที่ 2.1 จากตารางจะเห็นได้ว่าคุณสมบัติในด้าน ความแม่นยำ และ ความเร็ว วัดจากผลลัพธ์การทดสอบโมเดลบนชุดข้อมูล COCO 2017 [53] โดยเปรียบเทียบความแม่นยำในหน่วยของ Mean Average Precision (mAP) ซึ่งเป็นชุดข้อมูลขนาดใหญ่ที่มีภาพและคำอธิบายของวัตถุต่างๆ YOLOv4-Tiny มีขนาดโมเดลเล็กกว่า YOLOv4 ดั้งเดิมมาก แต่มีประสิทธิภาพใกล้เคียงกัน YOLOv4-Tiny จึงเหมาะสำหรับการใช้งานบนอุปกรณ์ที่มีทรัพยากรการคำนวณจำกัด

ตารางที่ 2.1 เปรียบเทียบความแตกต่าง YOLOv4-Tiny และ YOLOv4

Aspect	YOLOv4 (COCO)	YOLOv4-Tiny (COCO)
Model Weight (Approx.)	Around 600MB	Around 16MB
Model Size (Approx.)	Around 244MB (on disk)	Around 22MB (on disk)
Accuracy (mAP)	Around 45-50% mAP	Around 35-40% mAP
Inference Speed	Slower (e.g., 20ms/image)	Faster (e.g., 3ms/image)

2.6 เทคนิคการเสริมข้อมูล

เทคนิคการเสริมข้อมูล (Data Augmentation) เป็นเทคนิคที่ใช้ในการเสริมของชุดข้อมูล โดยการเพิ่มข้อมูลใหม่ ๆ เข้าไป ข้อมูลที่เพิ่มเข้าไปเหล่านี้สร้างขึ้นจากข้อมูลที่มีอยู่ โดยมีการปรับเปลี่ยนเล็กน้อยเพื่อให้ดูแตกต่างออกไป เทคนิคการเสริมข้อมูลมีประโยชน์ในการปรับปรุงความสามารถของอัลกอริทึมการเรียนรู้ของเครื่อง โดยช่วยให้อัลกอริทึมสามารถเรียนรู้จากข้อมูลได้มากขึ้น การเสริมข้อมูลของรูปภาพโดยใช้เทคนิค Data Augmentation เป็นกระบวนการที่นิยมใช้ในการปรับปรุงและขยายชุดข้อมูลสำหรับการฝึกโมเดลปัญญาประดิษฐ์ เช่น โมเดลการตรวจหาวัตถุหรือการจำแนกข้อมูล การเพิ่มขนาดของข้อมูลมีวัตถุประสงค์ เพื่อเพิ่มความหลากหลายในชุดข้อมูลฝึก ซึ่ง

ช่วยลดโอกาสการเกิด Overfitting [54] และเพิ่มความทนทานของโมเดลต่อข้อมูลที่มีความแปรปรวนหรือความหลากหลายสูง เนื่องจากการตรวจหาวัตถุ หรือการจำแนกต้องสามารถทำงานกับภาพที่มีความแตกต่างกันได้

งานวิจัยของ P. Adarsh และคณะ [55] ได้นำเสนอการตรวจจับวัตถุและการจดจำโดยใช้แบบจำลองที่ปรับปรุงแล้วในขั้นตอนเดียวโดยได้แนะนำ YOLO เวอร์ชันที่ได้รับการปรับปรุงที่เรียกว่า YOLO v3-Tiny อัลกอริทึมการตรวจจับวัตถุได้รับการปรับปรุงอย่างมากทั้งในด้านความเร็วและความถูกต้อง ปัจจัยหลักในการสร้างโมเดลสำหรับการตรวจจับวัตถุต้องกำจัดอิทธิพลของพื้นหลังของภาพผ่านกระบวนการก่อนการประมวลผล (Preprocessing) และการฝึกอบรมโมเดลสำหรับการตรวจจับวัตถุโดยเน้นที่รถยนต์ที่ปรากฏเป็นวัตถุขนาดเล็กตามแนวทิศทางของถนน ปัญหาการตรวจจับถูกมองว่าเป็นปัญหาการจดจำวัตถุ 3 มิติ โดยคำนึงถึงการเปลี่ยนแปลงของมุมมองและเงา คุณลักษณะที่สำคัญสำหรับการตรวจจับรถยนต์ การตรวจจับและติดตามวัตถุโดยใช้ UAV สำหรับแอปพลิเคชันต่างๆ นอกจากนี้งานวิจัยของ A. Ramachandran [31] และคณะ ได้จำแนกและอธิบายวิธีการตรวจจับวัตถุที่ใช้ในภาพ UAV และแสดงรายการชุดข้อมูล UAV เฉพาะสำหรับงานตรวจจับวัตถุ พร้อมทั้งได้สรุปงานประยุกต์อื่น ๆ ที่เกี่ยวข้อง และเสนอระบบการประมวลผลออนบอร์ดที่ปลอดภัยบนกรอบการตรวจจับวัตถุในงานทางด้านการเกษตรอย่างแม่นยำ

เทคนิคการเสริมข้อมูลสามารถทำได้หลายวิธี ตัวอย่างเช่น เทคนิคที่ใช้ในการเพิ่มปริมาณและปรับปรุงคุณภาพของข้อมูลการฝึกอบรม โดยการสร้างข้อมูลใหม่จากข้อมูลที่มีอยู่ เทคนิคการเพิ่มความสามารถของข้อมูลที่ใช้ในงานการตรวจจับวัตถุมีหลายวิธี ดังนี้

2.6.1 การพลิกภาพ (Image Flipping)

การพลิกภาพเป็นเทคนิค Data Augmentation ที่ง่ายและมีประสิทธิภาพมากที่สุด เทคนิคนี้เป็นการพลิกภาพกลับแนวนอนหรือแนวตั้งแบบสุ่ม ซึ่งจะช่วยให้จำลองมุมมองหรือทิศทางที่แตกต่างกันของวัตถุ ทำให้โมเดลเรียนรู้ที่จะจัดการวัตถุจากมุมมองต่างๆ

2.6.2 การปรับขนาดแบบสุ่ม และการปรับแต่งอัตราส่วนความกว้างความยาว (Random Scaling and Aspect Ratio Modification)

เป็นการปรับขนาดภาพแบบสุ่ม การย่อ/ขยายภาพเป็นเทคนิค Data Augmentation ที่ช่วยเพิ่มความแปรปรวนในขนาดวัตถุและอัตราส่วนกว้างยาว เทคนิคนี้เป็นการย่อหรือขยายภาพแบบสุ่มภายในช่วงขนาดที่กำหนด ซึ่งจะช่วยให้เพิ่มความแปรปรวนในขนาดวัตถุและอัตราส่วนความกว้างความยาว ทำให้โมเดลปรับเปลี่ยนได้ดีขึ้นกับวัตถุที่ขนาดต่างกัน

2.6.3 การครอบตัดแบบสุ่ม (Random Cropping)

เป็นการครอบตัด ส่วนของภาพแบบสุ่ม ซึ่งจะช่วยจำลองสถานการณ์ที่วัตถุบางส่วนถูกบดบัง ทำให้โมเดลเรียนรู้ที่จะตรวจจับวัตถุในสถานการณ์ต่างๆ เมื่อวัตถุปรากฏเพียงบางส่วนหรือถูกบดบังบางส่วน

2.6.4 การหมุนและการเฉือน (Rotation and Shearing)

เป็นการหมุนหรือเฉือนภาพแบบสุ่ม ซึ่งจะช่วยจำลองสถานการณ์ที่วัตถุเอียงหรือมีทิศทางที่แตกต่างกัน ทำให้โมเดลเรียนรู้ที่จะจัดการกับวัตถุที่มีการหมุนหรือตำแหน่งที่แตกต่าง

2.6.5 การเพิ่มข้อมูลสีและการปรับความสว่าง (Color Distortion and Brightness Adjustments)

การปรับสีภาพอินพุตแบบสุ่ม เช่น การปรับความชัด เปลี่ยนความสว่าง หรือปรับความเข้มสี เพื่อเพิ่มความสามารถของโมเดลในการจัดการกับเงื่อนไขแสงที่แตกต่างกัน ซึ่งจะช่วยจำลองสถานการณ์ที่วัตถุมีสภาพแสงที่แตกต่างกัน

2.6.6 การเพิ่มสัญญาณรบกวน (Noise Injection)

การเพิ่มสัญญาณรบกวนเป็นเทคนิค Data Augmentation ที่ช่วยจำลองสถานการณ์ที่วัตถุมีสัญญาณรบกวน เทคนิคนี้เป็นการเติมสัญญาณรบกวนลงในภาพแบบสุ่มด้วยวิธีการต่างๆ เช่น Gaussian noise จะช่วยให้โมเดลทนทานต่อสัญญาณรบกวน

2.6.7 การเพิ่มข้อมูลสุ่ม (Randomized Augmentation)

การรวมเทคนิคการเสริมข้อมูลหลายอย่างแบบสุ่ม ที่สร้างความแปรปรวนในข้อมูลที่ใช้ในการฝึกสอนโมเดล เสริมความสามารถในการจัดการและความมีประสิทธิภาพของโมเดล

การตรวจจับวัตถุในภาพถ่ายโดรนจะต้องคำนึงถึงปัญหาที่เกิดจากภาพถ่ายทางอากาศ เช่น การเปลี่ยนแปลงของขนาดวัตถุ ความไม่สมดุลของความถี่ ทิศทางที่วัตถุเคลื่อนที่ และอัตราส่วนภาพขนาดใหญ่ของวัตถุ ดังนั้นการนำเทคนิคการเสริมข้อมูลภาพเพื่อเพิ่มความเป็นไปได้ของข้อมูลในการฝึกสอนโมเดลให้เรียนรู้กับภาพในลักษณะต่างๆ จากงานวิจัยที่เกี่ยวข้องทั้งหมดที่กล่าวมานั้น ผู้วิจัยจึงตระหนักถึงการนำอัลกอริทึม YOLOv4-Tiny และเทคนิคการเสริมข้อมูลมาใช้เพื่อปรับปรุงความสามารถของโมเดลตรวจจับรถยนต์ขนาดเล็กแบบขั้นตอนเดียวจากภาพความละเอียดต่ำที่บันทึกจากโดรน การใช้เทคนิคการเสริมข้อมูลจะช่วยให้โมเดลสามารถเรียนรู้จากข้อมูลที่มีความแตกต่างมากขึ้น ซึ่งจะช่วยปรับปรุงความสามารถของโมเดลได้ เช่น ช่วยให้โมเดลสามารถทำงานได้

อย่างแม่นยำมากขึ้นในสภาพแวดล้อมที่หลากหลาย ช่วยให้โมเดลสามารถตรวจจับวัตถุได้มากขึ้น ช่วยลด Overfitting ของโมเดล การเลือกใช้เทคนิคการขยายข้อมูลที่เหมาะสมขึ้นอยู่กับลักษณะงานที่ทำ ตัวอย่างเช่น หากงานที่ต้องการทำคือการจำแนกภาพรถยนต์ การพลิกภาพและการหมุนภาพอาจมีประโยชน์ในการปรับปรุงความสามารถของอัลกอริทึม เนื่องจากรถยนต์สามารถปรากฏในภาพได้หลายมุม การขยายข้อมูลเป็นเทคนิคที่สำคัญสำหรับการเรียนรู้ของเครื่องจำนวนมาก เนื่องจากช่วยให้อัลกอริทึมสามารถเรียนรู้จากข้อมูลได้มากขึ้น ซึ่งจะส่งผลให้อัลกอริทึมมีความแม่นยำสูงขึ้น

ในงานวิจัยนี้ได้เลือกใช้เทคนิคการเสริมข้อมูลด้วยกันอยู่ 3 เทคนิค

1) การปรับเปลี่ยนค่าสีของภาพในปริภูมิ HSV (Color jitter augmentation in HSV space)

วิธีนี้เป็นการปรับเปลี่ยนค่าสีของภาพ โดยเปลี่ยนค่าโทน (Hue, H), ความอิ่มตัวสี (Saturation, S), และ ความคมชัด (Value, V) ของภาพ การเปลี่ยนแปลงค่าสีเหล่านี้จะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของสีที่หลากหลายของวัตถุในภาพได้

2) การสะท้อนภาพแนวนอนแบบสุ่ม (Random Horizontal Flip)

วิธีนี้เป็นการสะท้อนภาพแนวนอน การเปลี่ยนแปลงนี้จะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของวัตถุในภาพจากมุมมองที่หลากหลายได้

3) การย่อหรือขยายภาพ 10% แบบสุ่ม (Random scaling by 10 percent)

วิธีนี้เป็นการย่อหรือขยายภาพ 10% การเปลี่ยนแปลงนี้จะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของวัตถุในภาพที่มีขนาดแตกต่างกันได้

2.7 การปรับปรุงคุณภาพ (Optimizer)

การตรวจจับวัตถุขนาดเล็กในภาพเป็นงานคอมพิวเตอร์วิทัศน์ที่ท้าทายเนื่องจากวัตถุขนาดเล็กมักจะมีอัตราส่วนพื้นผิวต่อพื้นที่ (Surface-to-Area Ratio) ต่ำ และอาจถูกบดบังโดยวัตถุอื่น ๆ ในภาพ โมเดลการระบุวัตถุขนาดเล็กจึงต้องสามารถเรียนรู้ที่จะแยกแยะวัตถุขนาดเล็กออกจากพื้นหลังและวัตถุอื่น ๆ ในภาพได้อย่างแม่นยำ ซึ่งการปรับปรุงคุณภาพของโมเดลออฟติไมเซอร์ (Optimizer) เป็นอัลกอริทึมที่ใช้ในการปรับปรุงความสามารถของโมเดลการเรียนรู้ของเครื่อง ออฟติไมเซอร์ จะทำงานโดยปรับค่าพารามิเตอร์ของโมเดลเพื่อให้โมเดลสามารถคาดการณ์ผลลัพธ์ที่ถูกต้องได้มากขึ้น ออฟติไมเซอร์ เป็นปัจจัยสำคัญอย่างหนึ่งที่ใช้กำหนดประสิทธิภาพของโมเดลการระบุวัตถุขนาดเล็ก ออฟติไมเซอร์ เป็นอัลกอริทึมที่ใช้ในการปรับพารามิเตอร์ของโมเดลให้ดีขึ้นในช่วงการฝึกอบรวม โดย ออฟติไมเซอร์ ที่ดีจะช่วยให้โมเดลสามารถเรียนรู้และปรับตัวเข้ากับข้อมูลการฝึกอบรวมได้ดียิ่งขึ้น การเลือกตัวอัลกอริทึมออฟติไมเซอร์ เพื่อปรับปรุงประสิทธิภาพของโมเดลการจำแนกภาพ

เป็นขั้นตอนสำคัญในการพัฒนาและปรับปรุงโมเดลการเรียนรู้เชิงลึก นอกจากการเลือกออฟติไมเซอร์ยังสามารถปรับแต่งพารามิเตอร์อื่น ๆ เพื่อให้โมเดลทำงานได้ดียิ่งขึ้น

การปรับปรุง Optimizer ให้มีอัตราการเรียนรู้ที่เหมาะสมเพื่อช่วยแก้ไขปัญหาการปรับปรุงประสิทธิภาพในการเลือกค่าให้เหมาะสมกับรูปภาพความละเอียดต่ำขนาดเล็กที่ได้จากการแปลงวิดีโอจากโดรนในกระบวนการเรียนรู้เชิงลึกของโมเดล CNN เพื่อแก้ปัญหาเรื่องค่าต่ำสุดเฉพาะที่ (Local Minima), พลาดเป้า (Overshoot) และ การแกว่ง (Oscillation) ที่เกิดจากอัตราการเรียนรู้คงที่ วิธีการที่นำเสนอมีประสิทธิภาพเหนือกว่าวิธีการที่มีอยู่เดิมในแง่ของ Accuracy ในการตรวจจับวัตถุขนาดเล็กที่มีความละเอียดต่ำของพิกเซล ดังนั้นการนำออฟติไมเซอร์ Adam มาใช้ในงานวิจัยเล่มนี้จะเปรียบเทียบผลลัพธ์กับการปรับปรุงออฟติไมเซอร์ SGDM เพื่อหาค่าการเลือกใช้ที่เหมาะสมสำหรับงานตรวจจับวัตถุแบบเรียลไทม์ในวิดีโอโดยใช้เทคนิคตรวจจับหนึ่งขั้นตอน YOLO ซึ่งในปัจจุบันมีการพัฒนา YOLO แบบ Lightweight เพื่อใช้ในงานบนอุปกรณ์ที่มีทรัพยากรจำกัด เช่น YOLOv3-Tiny, YOLOv4-Tiny

2.7.1 อัลกอริทึมออฟติไมเซอร์ที่ใช้ในการจำแนกภาพ

2.7.1.1 Stochastic Gradient Descent (SGD)

เป็นตัวอัลกอริทึมที่ใช้การคำนวณค่าเกรเดียนต์ (Gradient) ของฟังก์ชันค่าความสูญเสีย (Loss Function) [53] และปรับค่าพารามิเตอร์ของโมเดลในทิศทางตรงข้ามของเกรเดียนต์นั้น ซึ่งทำให้การปรับค่าพารามิเตอร์ในแต่ละรอบการฝึกโดยใช้ตัวอัตราการเรียนรู้เท่ากับค่าคงที่ ลักษณะการใช้งานที่เหมาะสม SGD เหมาะสำหรับการใช้งานที่มีข้อมูลมากและต้องการการฝึกในข้อมูลแบบมินิ-แบตช์ (Mini-Batch) ในการปรับค่าพารามิเตอร์ เช่น การจำแนกภาพในโครงการฝึกอย่างหลากหลาย

2.7.1.2 Stochastic Gradient Descent with Momentum (SGDM)

เป็นเวอร์ชันของอัลกอริทึม SGD ที่มีการเพิ่มเติมของการบริหารความเร็วเพื่อลดการแกว่งตัวในกระบวนการฝึกโมเดลการจำแนกภาพ ซึ่งจะช่วยให้การฝึกโมเดลเร็วขึ้นและมีความเสถียรมากขึ้น นอกจากนี้ค่าโมเมนตัม (Momentum) คือสิ่งที่แตกต่างจาก SGD ธรรมดา เมื่อคำนวณค่าเกรเดียนต์ใหม่ SGDM จะใช้ค่าโมเมนตัมในการปรับปรุงค่าพารามิเตอร์ ลักษณะการใช้งานที่เหมาะสม SGDM เหมาะสำหรับการฝึกโมเดลในรูปแบบทั่วไปที่ไม่ซับซ้อน โมเดลที่มีจำนวนพารามิเตอร์ไม่มากเกินไปและข้อมูลฝึกที่มีปริมาณปานกลางถึงมาก การใช้ SGDM ในกรณีนี้ช่วยให้การฝึกมีความเร็วและเสถียรมากขึ้น

2.7.1.3 Adaptive Moment Estimation (ADAM)

เป็นตัวอัลกอริทึมที่รวมรวมความเร็วของการเคลื่อนที่เฉลี่ยและข้อมูลในการปรับค่าพารามิเตอร์ อัลกอริทึมสามารถปรับค่าอัตราการเรียนรู้แต่ละรอบอัตโนมัติเพื่อช่วยในการฝึกในระหว่างข้อมูลที่แตกต่างกัน ลักษณะการใช้งานที่เหมาะสม Adam เหมาะสำหรับการใช้งานที่ต้องการความแม่นยำและความเร็วในการฝึก ซึ่งจะใช้ในงานการจำแนกภาพและตรวจหาวัตถุในภาพ

2.7.1.4 Root Mean Square Propagation (RMSprop)

เป็นตัวอัลกอริทึมที่ใช้การเรียนรู้เชิงเวียนกับอัตราการเรียนรู้สำหรับแต่ละพารามิเตอร์ [54] ซึ่งจะช่วยลดอัตราการเรียนรู้ในกรณีที่ค่าเกรเดียนต์ในตำแหน่งนั้นมีค่าใหญ่มาก

2.7.1.5 Adaptive Gradient Algorithm (AdaGrad)

เป็นตัวอัลกอริทึมที่ปรับค่าอัตราการเรียนรู้แต่ละรอบตามความถี่ของพารามิเตอร์ มันช่วยให้การฝึกโมเดลปรับไปในทิศทางที่ถูกต้องสำหรับแต่ละพารามิเตอร์ ลักษณะการใช้งานที่เหมาะสม AdaGrad เหมาะสำหรับการฝึกโมเดลที่มีข้อมูลมากและต้องการการปรับค่าพารามิเตอร์ในทิศทางที่ตรงตามค่าเกรเดียนต์ของพารามิเตอร์นั้น

2.7.1.6 Nesterov-accelerated Adaptive Moment Estimation (Nadam)

เป็นการผสมระหว่าง Nesterov Accelerated Gradient (NAG) และ Adam ซึ่งใช้การเรียนรู้เชิงเวียนและการปรับค่าอัตราการเรียนรู้อัตโนมัติ ลักษณะการใช้งานที่เหมาะสม Nadam มักถูกใช้ในการฝึกโมเดลที่ต้องการความแม่นยำและความเร็วในการฝึก ซึ่งมีแนวโน้มที่จะให้ประสิทธิภาพดีกว่าในบางกรณี

2.7.2 การเลือกใช้อัลกอริทึมออฟติไมเซอร์ในงานวิจัย

งานวิจัยนี้ได้เลือกใช้อัลกอริทึมออฟติไมเซอร์ SGDM และ ADAM มาเปรียบเทียบผลของการฝึกสอนโมเดล SGDM และ ADAM เนื่องจากการใช้ SGDM เป็นออฟติไมเซอร์ซึ่งได้รับความนิยมในการนำไปใช้กับงานคอมพิวเตอร์วิทัศน์ เนื่องจาก SGDM สามารถปรับปรุงประสิทธิภาพของโมเดลได้ดีกว่าออฟติไมเซอร์อื่น ๆ เช่น SGD เป็นต้น SGDM เป็นเวอร์ชันของอัลกอริทึม SGD ที่มีการเพิ่มเติมของการบริหารความเร็วในการลดการแกว่งตัวในกระบวนการฝึกโมเดลการจำแนกภาพ ซึ่งช่วยให้การฝึกโมเดลเร็วขึ้นและมีความเสถียรมากขึ้น การเพิ่มโมเมนตัมคือสิ่งที่แตกต่างจาก SGD ธรรมดา เมื่อคำนวณค่าเกรเดียนต์ใหม่ SGDM จะใช้ค่าโมเมนตัมในการปรับปรุงค่าพารามิเตอร์ การใช้ ADAM เป็นออฟติไมเซอร์ซึ่งเป็นตัวอัลกอริทึมที่รวมรวมความเร็วของการเคลื่อนที่เฉลี่ยและค่าเฉลี่ยเคลื่อนที่ของข้อมูลในการปรับค่าพารามิเตอร์ ซึ่งสามารถปรับค่าอัตราการเรียนรู้แต่ละรอบอัตโนมัติเพื่อช่วยในการฝึกในระหว่างข้อมูลที่แตกต่างกัน การเลือกใช้งานตัวอัลกอริทึมออฟติไมเซอร์ เพื่อปรับปรุงประสิทธิภาพของโมเดลการระบุวัตถุในภาพแบบ ADAM ซึ่งการใช้ตัวอัลกอริทึมที่ถูก

ออกแบบมาเพื่อลดความซับซ้อนในการปรับค่าอัตราการเรียนรู้และช่วยให้การฝึกโมเดลสามารถเกิดขึ้นได้อย่างมีประสิทธิภาพมากขึ้น

2.7.2.1 การทำงานของอัลกอริทึมออฟติไมเซอร์ SGD

SGD ทำงานโดยอาศัยการประมาณค่าเกรเดียนต์ของฟังก์ชันการสูญเสีย (Loss Function) ในช่วงการฝึกอบรม โดย SGD จะทำการปรับปรุงพารามิเตอร์ของโมเดลในทิศทางที่ตรงกันข้ามกับค่าเกรเดียนต์ของฟังก์ชันการสูญเสีย และ SGD จะใช้ค่าโมเมนตัมเพื่อเพิ่มการปรับปรุงพารามิเตอร์ในทิศทางเดียวกันในช่วงการฝึกอบรมหลาย ๆ รอบ การใช้ค่าโมเมนตัมช่วยให้ SGD สามารถหลีกเลี่ยงการกระโดดของพารามิเตอร์ของโมเดลมากเกินไป ซึ่งอาจทำให้โมเดลไม่สามารถเรียนรู้ได้ดีขึ้น SGD สามารถช่วยปรับปรุงประสิทธิภาพของโมเดลการตรวจจับวัตถุขนาดเล็กได้โดยการช่วยให้โมเดลสามารถเรียนรู้ที่จะแยกแยะวัตถุขนาดเล็กออกจากพื้นหลังและวัตถุอื่น ๆ ในภาพได้แม่นยำยิ่งขึ้น โดย SGD จะใช้ค่าโมเมนตัมเพื่อเพิ่มการปรับปรุงพารามิเตอร์ของโมเดลในทิศทางที่ถูกต้องและจะช่วยให้โมเดลสามารถเรียนรู้ที่จะแยกแยะวัตถุขนาดเล็กได้ดีขึ้น นอกจากนี้ SGD ยังสามารถช่วยปรับปรุงประสิทธิภาพของโมเดลโดยทำให้โมเดลเสถียรมากขึ้น โมเดลอาจมีการกระโดดของพารามิเตอร์ของโมเดลระหว่างการฝึกอบรม ซึ่งอาจทำให้ประสิทธิภาพของโมเดลลดลงได้ SGD จะใช้ค่าโมเมนตัมเพื่อลดการกระโดดของพารามิเตอร์ของโมเดล ซึ่งจะช่วยให้โมเดลมีความเสถียรมากขึ้นและสามารถเรียนรู้ได้ดีขึ้น

2.7.2.2 การคำนวณของอัลกอริทึมออฟติไมเซอร์ SGD

การคำนวณเกรเดียนต์ของ SGD จะคำนวณค่าเกรเดียนต์ของฟังก์ชันค่าความสูญเสียต่อพารามิเตอร์ของโมเดลในแต่ละรอบ โดยการเพิ่มค่าโมเมนตัมในการปรับค่าพารามิเตอร์ SGD จะเพิ่มค่าโมเมนตัม ซึ่งเป็นค่าเสริมที่ช่วยลดการแกว่งตัว โดยคำนวณค่าใหม่ของโมเมนตัมจากค่าโมเมนตัมในรอบก่อนหน้าและ ค่าเกรเดียนต์ปัจจุบันสามารถ คำนวณได้ดังสมการที่ 2.1

$$\text{ค่าโมเมนตัมใหม่} = (\text{ค่าโมเมนตัมเก่า} * \text{อัตราการเรียนรู้}) - (\text{ค่าเกรเดียนต์} * \text{อัตราการเรียนรู้}) \quad (2.1)$$

การปรับค่าพารามิเตอร์ หลังจากคำนวณค่าโมเมนตัมใหม่ SGD จะใช้ค่านี้ในการปรับค่าพารามิเตอร์ของโมเดล โดยลบค่าเกรเดียนต์ คูณกับอัตราการเรียนรู้จากค่าพารามิเตอร์ปัจจุบัน และเพิ่มค่าโมเมนตัมใหม่เข้าไปสามารถคำนวณได้ดังสมการที่ 2.2

$$\text{ค่าพารามิเตอร์ใหม่} = \text{ค่าพารามิเตอร์ปัจจุบัน} - (\text{ค่าเกรเดียนต์} * \text{อัตราการเรียนรู้}) + (\text{ค่าโมเมนตัมใหม่}) \quad (2.2)$$

SGDM เหมาะสำหรับการฝึกโมเดลในรูปแบบทั่วไปที่ไม่ซับซ้อน โมเดลที่มีจำนวนพารามิเตอร์ไม่มากจนเกินไปและข้อมูลฝึกมีปริมาณปานกลางถึงมาก การใช้ SGDM ในกรณีนี้จะช่วยให้การฝึกมีความเร็วและเสถียรมากขึ้น

2.7.2.3 การทำงานของอัลกอริทึมออฟติไมเซอร์ ADAM

หลักการทำงานของตัวอัลกอริทึมออฟติไมเซอร์ ADAM คือ การคำนวณค่าเกรเดียนต์ เช่นเดียวกับ SGD ทั่วไป ADAM จะคำนวณค่าเกรเดียนต์ของฟังก์ชันค่าความสูญเสียต่อพารามิเตอร์ของโมเดลในแต่ละรอบของการฝึก การสร้างโมเมนตัมและการปรับอัตราการเรียนรู้ ADAM ใช้การสร้างโมเมนตัมและค่าเฉลี่ยเคลื่อนที่ของค่าเกรเดียนต์เพื่อปรับอัตราการเรียนรู้ในแต่ละรอบ ค่าเหล่านี้คือค่าโมเมนตัม และ ความเร็วแต่ละค่าจะมีความสำคัญในการปรับปรุงอัตราการเรียนรู้ ADAM เหมาะสำหรับการใช้งานที่ต้องการความแม่นยำและความเร็วในการฝึก มักถูกใช้ในงานการจำแนกภาพและตรวจหาวัตถุในภาพ ADAM ช่วยให้การฝึกโมเดลมีความเร็วและประสิทธิภาพสูงขึ้นโดยปรับอัตราการเรียนรู้โดยอัตโนมัติตลอดกระบวนการฝึก นอกจากนี้ ADAM ยังช่วยลดความสั่นสะเทือนของการปรับค่าพารามิเตอร์ ช่วยให้การฝึกไม่ติดอยู่ในค่าต่ำสุดเฉพาะที่ (Local Minima) และช่วยให้การปรับค่าพารามิเตอร์เกิดอย่างมีประสิทธิภาพ

2.7.2.2 การคำนวณของอัลกอริทึมออฟติไมเซอร์ ADAM

การปรับค่าพารามิเตอร์ หลังจากคำนวณค่าโมเมนตัมและความเร็วใหม่ Adam จะใช้ค่าเหล่านี้ในการปรับค่าพารามิเตอร์ของโมเดล สูตรคำนวณค่าพารามิเตอร์ใหม่เป็นคำนวณได้ดังสมการที่ 2.3

$$\text{ค่าพารามิเตอร์ใหม่} = \text{ค่าพารามิเตอร์ปัจจุบัน} - \frac{r * m}{\sqrt{v} + \epsilon} \quad (2.3)$$

โดย r คือ อัตราการเรียนรู้ที่กำหนด,

ϵ คือ ค่าเลขคงที่เล็กน้อยที่ใช้เพื่อป้องกันการหารด้วยศูนย์,

m คือ ค่าโมเมนตัมที่สร้างตามค่าเกรเดียนต์และใช้ในการลดความ

แตกต่างระหว่างอัตราการเรียนรู้ในรอบก่อนหน้าและรอบปัจจุบัน, และ

v คือ ค่าเฉลี่ยเคลื่อนที่ของค่าเกรเดียนต์และใช้ในการปรับค่าพารามิเตอร์

2.8 การประเมิน (Evaluation)

การประเมินความสามารถของโมเดลตรวจจับรถยนต์ด้วยภาพความละเอียดต่ำจากโดรน โดยใช้ YOLOv4-Tiny นั้นเป็นกระบวนการที่สำคัญในการวัดความสามารถของโมเดลในการตรวจจับวัตถุบนภาพขนาดเล็กและความละเอียดต่ำ เช่นภาพจากโดรนหรือกล้องแบบพกพาที่มีความละเอียดต่ำกว่ากล้องที่ใช้งานปกติ จะใช้เมตริกการประเมินเพื่อวัดประสิทธิภาพของโมเดล

2.8.1 เมตริกการประเมินที่นิยมใช้ในงานตรวจจับวัตถุ

2.8.1.1 ค่าความถูกต้อง (Accuracy)

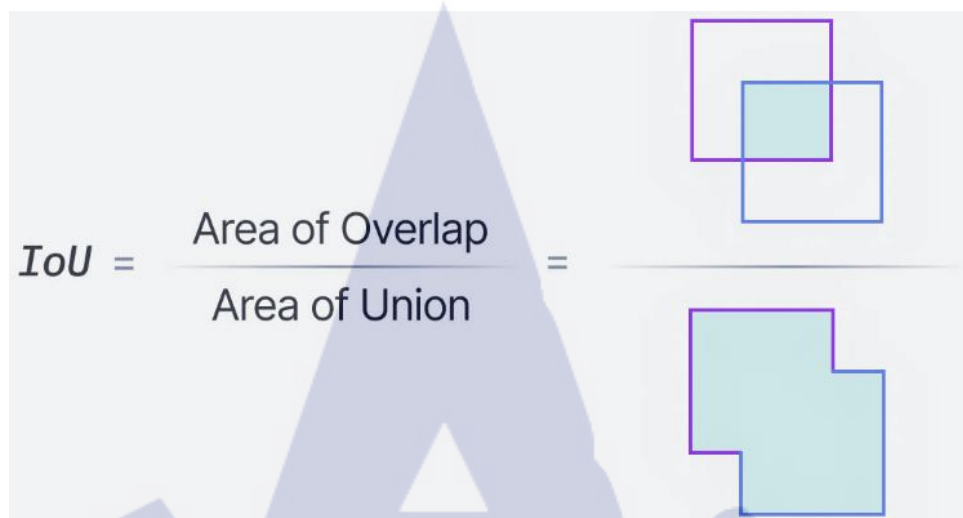
เป็นตัววัดว่าโมเดลสามารถตรวจจับรถยนต์ได้ถูกต้องกี่เปอร์เซ็นต์ โมเดลควรมีค่าความถูกต้องสูง เพื่อแสดงว่าโมเดลสามารถตรวจจับรถยนต์ได้ถูกต้องมากที่สุด คำนวณได้ดังสมการที่ 2.4

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total} \quad (2.4)$$

2.8.1.2 Intersection over Union (IoU)

เป็นการคำนวณอัตราส่วนของพื้นที่ที่ทับซ้อนระหว่างขอบเขตของวัตถุที่ตรวจพบโดยโมเดลและขอบเขตของวัตถุจริง การหา IoU ระหว่างขอบเขตที่คาดการณ์ไว้และขอบเขตที่แท้จริงของวัตถุ จะต้องหาพื้นที่ที่ตัดกันระหว่างขอบเขตทั้งสองที่สอดคล้องกับวัตถุเดียวกันก่อน แล้วจึงคำนวณพื้นที่ทั้งหมดที่ครอบคลุมขอบเขตทั้งสอง ซึ่งเรียกว่า ยูเนียน (Union) และพื้นที่ที่ทับซ้อนกันระหว่างกัน เรียกว่าอินเตอร์เซกชัน (Intersection) การแบ่งพื้นที่ที่ตัดกันด้วยยูเนียนจะให้อัตราส่วนของพื้นที่ที่ทับซ้อนกับพื้นที่ทั้งหมดซึ่งให้ค่าประมาณการที่ดีว่าขอบเขตของการคาดการณ์อยู่ใกล้กับขอบเขตดั้งเดิมมากน้อยเพียงใด โมเดลควรมีค่า IoU สูง เพื่อแสดงว่าขอบเขตของวัตถุที่ตรวจพบโดยโมเดลทับซ้อนกับขอบเขตของวัตถุจริงมากที่สุด ซึ่งสามารถคำนวณได้ดังสมการที่ 2.5

$$IoU = \frac{Intersection\ Area}{Union\ Area} \quad (2.5)$$



รูปที่ 2.6 Intersection over Union (IoU)

2.8.1.3 ความแม่นยำเฉลี่ย (Average Precision, AP)

เป็นตัววัดความแม่นยำของโมเดลที่สามารถตรวจจับวัตถุ AP คำนวณค่าเฉลี่ยของค่าความแม่นยำของทุกระดับ IoU ที่กำหนด โมเดลควรมีค่า AP สูง เพื่อแสดงว่าโมเดลสามารถตรวจจับรถยนต์ได้แม่นยำและครบถ้วนมากที่สุด AP สามารถ คำนวณได้ดังสมการที่ 2.6

$$AP = (1/N) * \text{Sum}(P(i) * IoU(i)) \quad (2.6)$$

โดย N คือ จำนวนวัตถุทั้งหมดในชุดข้อมูล

AP สามารถใช้เป็นดัชนีที่เหมาะสมสำหรับการประเมินประสิทธิภาพของโมเดลการตรวจจับวัตถุ เนื่องจากคำนึงถึงทั้งความแม่นยำและความสามารถในการตรวจพบ

2.8.1.4 ค่าความแม่นยำ (Precision)

เป็นตัววัดว่าโมเดลตรวจจับวัตถุได้อย่างแม่นยำเพียงใด ค่าความแม่นยำ คำนวณได้จากจำนวนวัตถุที่ตรวจพบอย่างถูกต้องหารด้วยจำนวนวัตถุทั้งหมดที่ตรวจพบ โมเดลควรมี Precision สูง เพื่อแสดงว่าโมเดลสามารถตรวจจับรถยนต์ได้อย่างแม่นยำมากที่สุด คำนวณได้ดังสมการที่ 2.7

$$\text{Precision} = \text{True Positive} / (\text{True Positive} + \text{False Positive}) \quad (2.7)$$

2.8.1.5 ค่าความครอบคลุม (Recall)

เป็นตัววัดว่าโมเดลสามารถตรวจจับวัตถุได้ครบถ้วนเพียงใด คำนวณได้จากจำนวนวัตถุที่ตรวจพบอย่างถูกต้องหารด้วยจำนวนวัตถุทั้งหมดที่มีอยู่ในภาพ โมเดลควรมี Recall สูงเพื่อแสดงว่าโมเดลสามารถตรวจจับวัตถุได้ครบถ้วนมากที่สุด Recall สามารถคำนวณจากสมการที่ 2.8

$$Recall = True\ Positive / (True\ Positive + False\ Negatives) \quad (2.8)$$

2.8.1.6 F1 Score

เป็นค่าที่ใช้วัดประสิทธิภาพของโมเดลการจำแนกประเภทในการเรียนรู้ของเครื่องที่ประเมินจากค่าความแม่นยำและค่าความครอบคลุม ค่า F1 [55] เป็นค่า Harmonic Mean ของ Precision และ Recall และให้ความสำคัญแก่ทั้งสองค่าเท่ากัน ซึ่งสามารถคำนวณได้ดังสมการที่ 2.9 โมเดลควรมี F1 Score สูง เพื่อแสดงว่าโมเดลสามารถตรวจจับวัตถุได้อย่างแม่นยำและครบถ้วนมากที่สุด

$$F1\ Score = 2 * (Precision * Recall) / (Precision + Recall) \quad (2.9)$$

AP เป็น Metrics ที่ใช้ประเมินความแม่นยำโดยรวมของโมเดลในการตรวจจับวัตถุ โดยพิจารณาจากทั้งความแม่นยำ Precision และ Recall ที่แตกต่างกัน โมเดลการเรียนรู้เชิงลึกเปรียบเทียบกับในแง่ของความแม่นยำในการตรวจพบวัตถุ และเวลาในการเทรนของโมเดล โดยสรุปแล้ว ดัชนีสำหรับประเมินความสามารถของโมเดลการตรวจจับวัตถุเป็นเครื่องมือสำคัญในการวัดความสามารถของโมเดลและช่วยให้เราทราบว่าโมเดลสามารถปรับปรุงได้อย่างไร

จากงานวิจัยที่เกี่ยวข้อง งานวิจัยนี้ได้นำความรู้และเทคนิคต่างๆ ในงานด้านคอมพิวเตอร์วิทัศน์ มาพัฒนาใช้กับงานด้านการตรวจจับวัตถุขนาดเล็กที่มีคุณภาพความละเอียดต่ำ ที่ถูกถ่ายวิดีโอมาจากโดรน เพื่อใช้ในการบินติดตามการจราจรหรือ เหตุการณ์ต่างๆที่เกิดขึ้นบนถนน โดยนำความรู้ด้านปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึก เลือกใช้อัลกอริทึมในงานการตรวจจับวัตถุแบบเดี่ยวของ YOLO มาพัฒนาการฝึกสอนโมเดล YOLOv4-Tiny เพราะเหมาะสำหรับการใช้งานบนอุปกรณ์ที่มีทรัพยากรการคำนวณจำกัด การฝึกสอนโมเดล YOLOv4-Tiny จะทำการเปรียบเทียบอินพุต 3 ขนาด โดยเริ่มจากขนาดอินพุตเริ่มต้น 416x416 pixels และทำการปรับค่าขนาดอินพุตลดลงและเพิ่มขึ้นในการฝึกสอน เพื่อหาขนาดที่สามารถเพิ่มความแม่นยำของระบบได้ นอกจากนี้จะเปรียบเทียบระยะเวลาในการฝึกสอนของแต่ละขนาดอินพุตและการใช้หน่วยความจำในการประมวลผลด้วย ช่วงการฝึกสอนชุดข้อมูลที่ใช้ในการ

ฝึกสอนจะถูกแบ่งเป็น 2 ชุด คือ ชุดข้อมูลปกติและชุดข้อมูลที่ทำเทคนิคการเสริมข้อมูลงานวิจัยนี้ได้เลือกใช้เทคนิคการเสริมข้อมูลด้วยการปรับเปลี่ยนค่าสีของภาพในปริภูมิ HSV, การสะท้อนภาพแนวนอนแบบสุ่ม และ การย่อหรือขยายภาพ 10% แบบสุ่ม การเตรียมข้อมูลภาพด้วยเทคนิคการทำ Data Augmentation เป็นเรื่องสำคัญเพื่อเพิ่มความแม่นยำในการตรวจจับของโมเดล ส่วนการเลือกใช้ออปติไมเซอร์ที่เหมาะสมเพื่อใช้สำหรับควบคุมกระบวนการการเรียนรู้ของโมเดล การปรับค่าพารามิเตอร์ในการเลือกใช้จะถูกเปรียบเทียบกับออปติไมเซอร์ที่ใช้คือ Adam และ SGDM ซึ่งเหมาะสมสำหรับการตรวจจับวัตถุภาพรถยนต์ความละเอียดต่ำขนาดเล็ก การใช้ออปติไมเซอร์จะทำให้ความสามารถในการตรวจจับรถยนต์ขนาดเล็กในสภาพแวดล้อมความละเอียดต่ำของรถยนต์ขนาดเล็กด้วย YOLOv4-Tiny ดีขึ้น ทั้งในด้านของความแม่นยำและความเร็วในการทำนาย

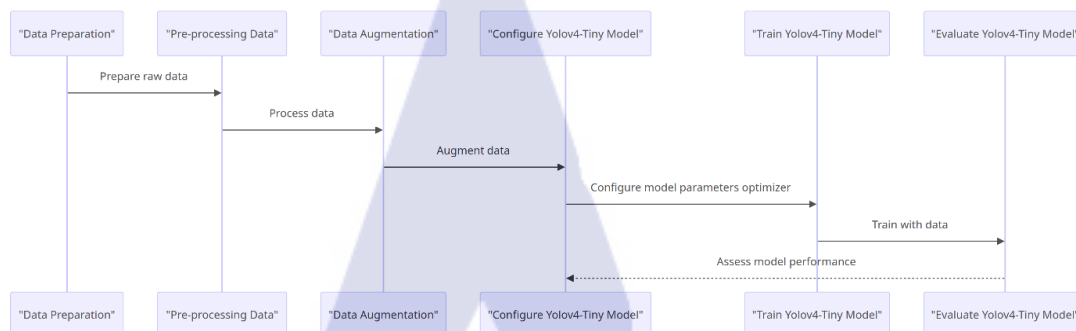


บทที่ 3

วิธีการดำเนินงาน

เนื้อหาในบทนี้กล่าวถึงระเบียบวิธีวิจัยเกี่ยวกับการตรวจจำแนกวัตถุขนาดเล็กที่มีคุณภาพ ความละเอียดต่ำจากโดรน มาพัฒนาการฝึกอบรมโมเดล YOLOv4-Tiny โดยแบ่งขั้นตอนการวิจัย เริ่มจากการเตรียมและรวบรวมข้อมูลที่ได้จากโดรน แบ่งชุดข้อมูลออกเป็น การฝึกอบรม (Training), การตรวจสอบ (Validation) และ การทดสอบ (Testing) โดยชุดข้อมูลฝึกอบรม ใช้สำหรับสอนโมเดล ให้เรียนรู้จากข้อมูลในชุดนี้เพื่อให้สามารถจำแนกหรือทำนายข้อมูลใหม่ได้อย่างถูกต้อง ชุดข้อมูล ตรวจสอบ ใช้สำหรับประเมินความสามารถของโมเดลในระหว่างการทดลอง โดยโมเดลจะถูกนำไป ทดสอบกับข้อมูลในชุดนี้ เพื่อตรวจสอบความสามารถของโมเดลที่เรียนรู้ สามารถจำแนกหรือทำนาย ข้อมูลใหม่ได้อย่างแม่นยำหรือไม่ และชุดข้อมูลทดสอบ ใช้สำหรับประเมินความสามารถของโมเดล อย่างแท้จริง โดยโมเดลจะถูกนำไปทดสอบกับข้อมูลในชุดนี้ ซึ่งจะแตกต่างจากข้อมูลที่ใช้ฝึกโมเดล และข้อมูลที่ใช้ประเมินความสามารถของโมเดล โดยจะประเมินโมเดลว่าสามารถทำงานได้อย่าง ถูกต้องเมื่อนำไปใช้กับข้อมูลใหม่ ๆ ชุดข้อมูลที่นำมาใช้ในการฝึกอบรมตัวตรวจจำ จะถูกแบ่งออกเป็น 2 ชุด คือ ชุดข้อมูลปกติคือ ชุดข้อมูลที่ไม่ผ่านเทคนิคการเสริมข้อมูล และชุดข้อมูลที่ถูกทำการเสริม ข้อมูล เพื่อนำไปเปรียบเทียบหาผลการฝึกอบรมโมเดล ที่ได้ค่าความแม่นยำของโมเดลสูงที่สุด และ เปรียบเทียบการเลือกใช้ออปติไมเซอร์ ที่ใช้ในการปรับปรุงโมเดล โดยเปรียบเทียบผลการปรับ ค่าพารามิเตอร์ออปติไมเซอร์ ระหว่าง SGDM กับ ADAM ด้วยการฝึกอบรมโมเดล YOLOv4-Tiny โดยมีการกำหนดค่าขนาดอินพุตโมเดลเริ่มต้น 416x416 pixels และทำการปรับค่าขนาดอินพุตลดลง และเพิ่มขึ้นในการฝึกอบรม เพื่อหาขนาดที่สามารถเพิ่มความแม่นยำของระบบ และทำการ เปรียบเทียบผลการฝึกอบรมโมเดลของทั้ง 3 ขนาดอินพุต เพื่อศึกษาและพัฒนาการตรวจจำแนกวัตถุ ขนาดเล็กด้วยภาพความละเอียดต่ำจากโดรน งานวิจัยนี้ได้สรุปขั้นตอนการดำเนินงานเป็นผังงาน ดัง รูปที่ 3.1

3.1 ขั้นตอนการทดลอง



รูปที่ 3.1 แผนผังลำดับขั้นตอนงานวิจัย

3.2 การเตรียมข้อมูล (Data Preparation)

3.2.1 Hardware ที่ใช้ในการประมวลผล

การทดลองการตรวจจำรถยนต์ด้วยภาพความละเอียดต่ำจากโดรน จะใช้โปรแกรม MATLAB และคอมพิวเตอร์ซึ่งมีรายละเอียดดังตารางที่ 3.1 ซึ่งเป็นส่วนของ Hardware ที่ใช้ในการประมวลผลการฝึกอบรมโมเดลในงานวิจัยนี้ ซึ่งมี GPU device NVIDIA T500 ที่ช่วยดึงประสิทธิภาพในการฝึกอบรมข้อมูลของเครื่องคอมพิวเตอร์

ตารางที่ 3.1 คุณสมบัติของอุปกรณ์

Device Specifications	
Operating System	Windows 10
Processor	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
Installed Ram	32.0 GB
System type	64-bit operating system, x64-based processor
GPU Device	NVIDIA T500
ComputeCapability	7.5

3.2.2 กระบวนการแปลงข้อมูล

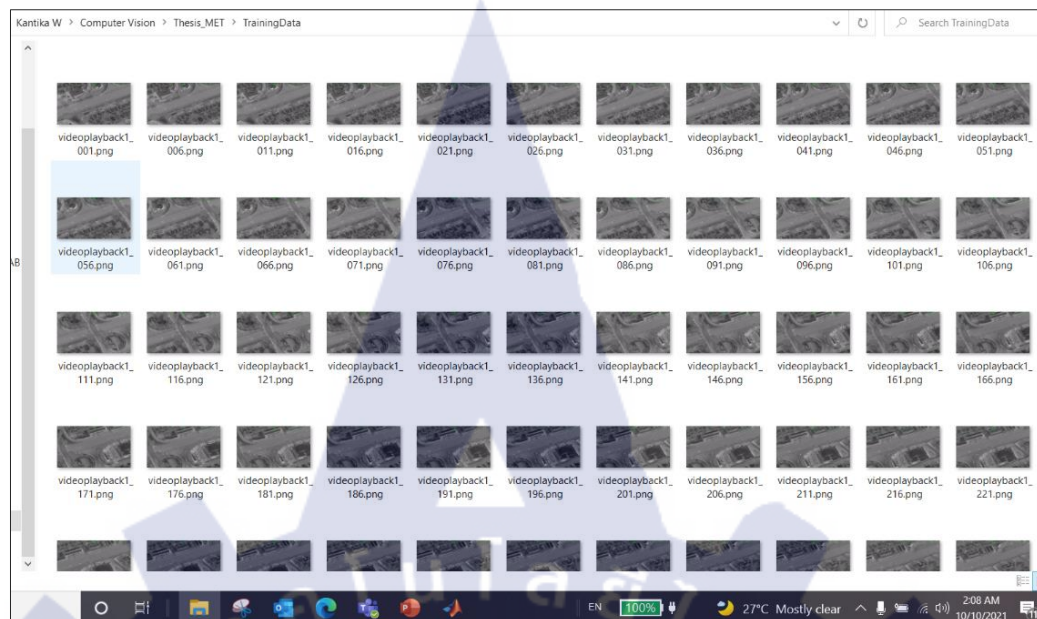
การจัดเตรียมข้อมูลรูปภาพ สำหรับการนำไปฝึกอบรมข้อมูลและสร้างโมเดลนั้น ได้นำข้อมูลอินพุตที่เป็นไฟล์วิดีโอคลิปที่ถูกบันทึกมาจากโดรนมีความยาวของวิดีโอคลิปเฉลี่ยอยู่ที่ 5 นาที 1 ไฟล์ ที่มีขนาดไฟล์เฉลี่ย 35.4 MB ซึ่งเป็นสกุลไฟล์ .mp4 เป็นไฟล์ที่มีความละเอียดต่ำ โดยมี ขนาดของรูปภาพ 854x480 pixels ที่มีเฟรมเรทอยู่ที่ 30 เฟรมต่อวินาที และมีทั้งหมด 9074 เฟรม ซึ่งมีขนาดของบิตต่อพิกเซลอยู่ที่ 24 บิตต่อพิกเซล คุณสมบัติของวิดีโอแสดงดังตารางที่ 3.2

ตารางที่ 3.2 คุณสมบัติของวิดีโอ

Video Properties		Measurement
Name	Videoplayback.mp4	.mp4
Width x Height	854 x 480	pixels
FrameRate	30	frames per second
BitsPerPixel	24	bits per pixel
VideoFormat	RGB24	AVI or MPEG-4 files with RGB24 video
Duration	302.4667	Second
NumFrames	9074	Frames

ขั้นตอนการแปลงข้อมูลมีรายละเอียดดังนี้

- นำวิดีโอความยาวเฉลี่ย 5 นาที แปลงเป็นข้อมูลรูปภาพ โดยใช้กระบวนการอ่านค่าเฟรมทั้งหมดของวิดีโอและทำการบันทึกเป็นไฟล์รูปภาพแยกกันด้วยสกุลไฟล์ .png จะได้จำนวนรูปภาพทั้งหมดที่ใช้ในการฝึกอบรมอยู่ 873 รูปภาพดังรูปที่ 3.2



รูปที่ 3.2 รูปภาพในการฝึกอบรมโมเดล

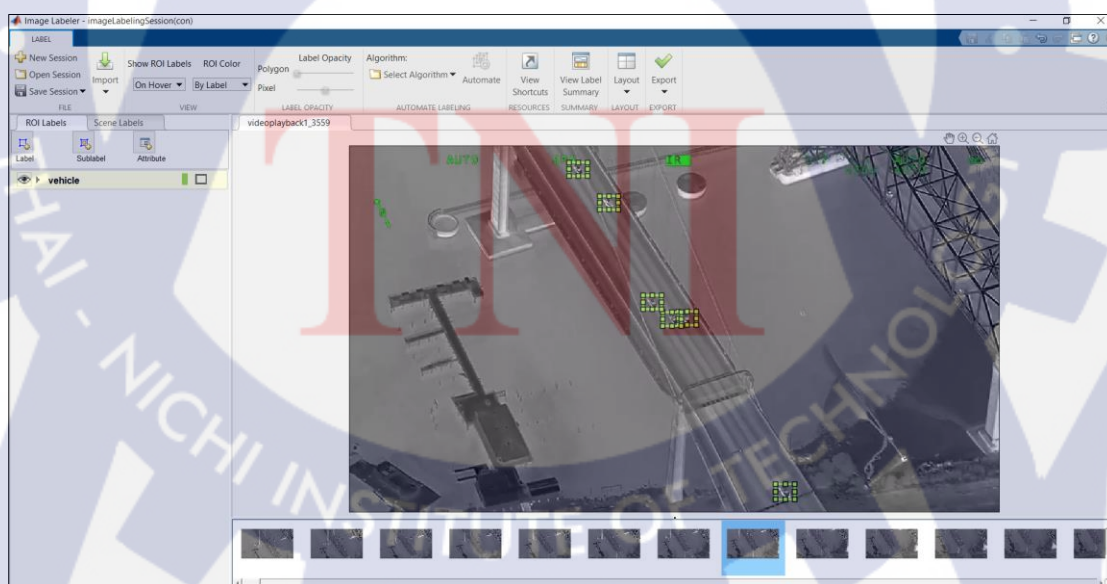
- เมื่อวีดีโอได้แปลงเป็นข้อมูลรูปภาพแล้ว ข้อมูลจะแบ่งเป็น 3 ส่วนคือ การฝึกอบรม การตรวจสอบ และ การทดสอบ ซึ่งเป็นขั้นตอนที่สำคัญในการพัฒนาและทดสอบโมเดล ข้อมูลเหล่านี้ จะทำให้สามารถประเมินประสิทธิภาพของโมเดล เพื่อทำนายข้อมูลที่ไม่เคยเห็นมาก่อน ข้อมูลทั้ง 3 ส่วนจะถูกแบ่ง แบบการสุ่มตัวอย่าง(Random Sampling) [50] จากชุดข้อมูลทั้งหมด โดยไม่คำนึงถึง ข้อมูลอื่น ๆ สัดส่วนของข้อมูลทั้ง 3 ส่วนมีรายละเอียดดังนี้ ข้อมูลฝึกอบรม 80%, ข้อมูลตรวจสอบ 10% และ ข้อมูลทดสอบ 10% ข้อมูลแต่ละส่วนมีรูปแบบการใช้งานดังนี้
 - ชุดข้อมูลฝึกอบรม เป็นชุดข้อมูลที่ใช้ในการฝึกโมเดล และเรียนรู้พฤติกรรม ของข้อมูล การฝึกอบรมโมเดลจำเป็นต้องใช้ข้อมูลฝึกอบรม เพื่อใช้ปรับพารามิเตอร์ของโมเดลให้ เหมาะสม และสามารถจำแนกหรือทำนายข้อมูลใหม่ได้อย่างถูกต้อง ข้อมูล 80% จะใช้ในการ ฝึกอบรมโมเดล การใช้ข้อมูลมากขึ้นในชุดข้อมูลฝึกอบรม ช่วยให้โมเดลมีโอกาสเรียนรู้คุณสมบัติและ หน้าที่ของข้อมูลได้ดีขึ้น ข้อมูลการฝึกอบรมเป็นชุดข้อมูลที่มีจำนวนข้อมูลมากที่สุด ของชุดข้อมูล ทั้งหมด เนื่องจากโมเดลที่ทำนายได้แม่นยำจำเป็นต้องใช้ข้อมูลที่มีมาก
 - ชุดข้อมูลตรวจสอบ เป็นชุดข้อมูลที่ใช้ในขณะฝึกเพื่อปรับปรุงความสามารถ ของโมเดล โมเดลจะถูกนำไปทดสอบกับข้อมูลในชุดนี้ เพื่อประเมินความสามารถของโมเดลในการ เรียนรู้ข้อมูล และจำแนกหรือทำนายข้อมูลใหม่ ข้อมูลชุดนี้จะมีประมาณ 10% ซึ่งจะใช้เพื่อปรับ ค่าพารามิเตอร์หรือเพื่อเลือกโมเดลที่ดีที่สุดจากหลายๆ โมเดลที่ฝึกอบรมขึ้น

- ชุดข้อมูลทดสอบ เป็นชุดข้อมูลที่ใช้ในการทดสอบความสามารถของโมเดล

หลังจากการฝึกอบรมและการปรับปรุง และยังใช้ประเมินความสามารถของโมเดลในการทำนายข้อมูลที่ไม่เคยพบมาก่อน ข้อมูลประมาณ 10% ของข้อมูลทั้งหมดจะถูกเก็บไว้เพื่อทดสอบความสามารถของโมเดลหลังจากการฝึกและการปรับปรุง ชุดข้อมูลทดสอบช่วยให้วัดความสามารถในการทำนายข้อมูล โดยไม่มีความสัมพันธ์กับการฝึกโมเดล

3.2.3 กระบวนการทำระบุนภาพ

การทำระบุน (Labeling) ภาพ มีความสำคัญอย่างยิ่งสำหรับงานด้านการเรียนรู้ของเครื่องและปัญญาประดิษฐ์ เนื่องจากเป็นกระบวนการที่จำเป็นสำหรับการเตรียมข้อมูลภาพสำหรับการฝึกอบรมโมเดล หลักการของการระบุนภาพ คือการกำหนดค่าต่างๆ ให้กับวัตถุในภาพ เช่น ประเภทของวัตถุ ตำแหน่งของวัตถุ ขนาดของวัตถุ ฯลฯ ซึ่งข้อมูลเหล่านี้จะถูกนำไปใช้ในการฝึกอบรมโมเดลเพื่อให้โมเดลสามารถจำแนกหรือทำนายข้อมูลภาพได้อย่างถูกต้อง งานวิจัยนี้ได้ทำการระบุนภาพโดยใช้ Image Labeler App ในโปรแกรม MATLAB เพื่อทำให้การระบุนตำแหน่งของรถยนต์ในภาพได้สะดวกมากยิ่งขึ้น เนื่องจากรูปภาพวัตถุมีขนาดเล็ก และขนาดของรูปภาพต้นแบบเป็นไฟล์คุณภาพต่ำ ทำให้ไม่สามารถใช้กระบวนการระบุนภาพอัตโนมัติ (Automated Labeling) [52] ได้ ผู้วิจัยจึงได้ทำการระบุนภาพด้วยวิธีระบุนภาพด้วยตนเอง (Manual Labeling) ทั้งหมด 873 รูป ตัวอย่างการระบุนภาพแสดงดังรูปที่ 3.3



รูปที่ 3.3 ขั้นตอนของกระบวนการระบุนภาพ

กระบวนการระบุภาพ ดังรูปที่ 3.3 มีขั้นตอนดังนี้

- เปิด Image Labeler App บนโปรแกรม MATLAB โดยพิมพ์คำสั่งคำว่า imageLabeler ใน Command Window แล้วกด Enter
- โหลดภาพใน Image Labeler App โดยโหลดภาพที่ต้องการระบุตำแหน่ง ด้วย ปุ่ม "Load" หรือ "Import" แล้วเลือกไฟล์ภาพที่ต้องการทำงาน
- สร้าง Region of Interest (ROI) เลือกภาพที่ต้องการทำการระบุ และทำการกำหนดพื้นที่ของวัตถุที่ต้องการระบุ ในกรณีที่มีหลายวัตถุในรูป สามารถสร้างหลายพื้นที่ของ Region of Interest (ROI) ได้
- การระบุภาพ โดยเมื่อเลือก ROI เสร็จแล้วสามารถกำหนดป้ายกำกับให้กับวัตถุ เช่น "vehicle" เป็นต้น
- การระบุวัตถุในภาพ ดังตัวอย่างในรูปที่ 3.3 กรอบสี่เหลี่ยมคือการ ระบุวัตถุที่เราสนใจ รูปรถยนต์ในภาพ มีด้วยกัน 6 คัน ซึ่งในแต่ละรูปภาพจะมีวัตถุที่เราสนใจอยู่แตกต่างกัน ในงานวิจัยนี้ค่าเฉลี่ยของวัตถุที่สนใจอยู่ที่ 8 คันต่อ 1 รูปภาพ
- หลังจากที่เราระบุ ROI และระบุภาพเรียบร้อยแล้ว ให้ทำการบันทึกค่าที่ระบุ โดยใช้ปุ่ม "Save" หรือ "Export Labels" เพื่อเก็บข้อมูล Label ที่ได้สร้างไว้
- สร้างชุดข้อมูล หลังจากที่ได้ทำการระบุภาพให้กับภาพทั้งหมด จะได้ชุดของข้อมูลสำหรับการฝึกอบรมโมเดล และทำการ "Export Labels" นำออกไปเป็นตารางเพื่อใช้งาน ตารางนี้จะมีการระบุตำแหน่งของวัตถุในภาพ ดังรูปภาพที่ 3.4 ทำให้ได้ข้อมูลที่ผ่านมากระบวนการระบุภาพ ด้วยตัววัตถุที่สนใจออกมาเป็นเมตริกซ์

vehicleTrainingData		
873x2 table		
	1 imageFilename	2 vehicle
31	'D:\DriveD\Kantika W\Computer Vision\Thesis_MET\videoplayback1_156.png'	[268,252,26,15;553,274,26,15]
32	'D:\DriveD\Kantika W\Computer Vision\Thesis_MET\videoplayback1_161.png'	[264,247,26,15;523,272,26,15]
33	'D:\DriveD\Kantika W\Computer Vision\Thesis_MET\videoplayback1_166.png'	[264,246,26,15;495,276,26,15]
34	'D:\DriveD\Kantika W\Computer Vision\Thesis_MET\videoplayback1_171.png'	[260,247,26,15;470,283,26,15]

รูปที่ 3.4 ตัวอย่างไฟล์ข้อมูลที่ระบุตำแหน่งของวัตถุในภาพ

3.3 การเสริมขนาดของข้อมูล (Data Augmentation)

การเสริมขนาดของข้อมูลรูปภาพ เป็นกระบวนการที่นิยมใช้ในการปรับปรุงและขยายชุดข้อมูลสำหรับการฝึกโมเดลปัญญาประดิษฐ์ โดยการสร้างข้อมูลจำลองที่หลากหลายจากข้อมูลที่มีอยู่ ซึ่งจะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของข้อมูลได้ดีขึ้น และสามารถทำงานได้อย่างแม่นยำยิ่งขึ้น

วัตถุประสงค์หลักของการเสริมขนาดของข้อมูล คือ การเพิ่มความหลากหลายในชุดข้อมูลฝึก ซึ่งการเสริมขนาดของข้อมูล จะช่วยให้โมเดลสามารถเรียนรู้จากข้อมูลที่มีความแตกต่างมากขึ้น และช่วยปรับปรุงความสามารถของโมเดลได้ ช่วยให้โมเดลสามารถทำงานได้อย่างแม่นยำมากขึ้นในสภาพแวดล้อมที่หลากหลาย และยังช่วยลด Overfitting ของโมเดลได้ ดังที่กล่าวในหัวข้อที่ 2.6 การเสริมขนาดของข้อมูลที่ใช้ในงานวิจัย ได้แก่

3.3.1 การปรับเปลี่ยนค่าสีของภาพในปริภูมิ HSV (Color jitter augmentation in HSV space)

วิธีนี้เป็นการปรับเปลี่ยนค่าสีของภาพ โดยเปลี่ยนค่าสีในพื้นที่สี HSV เป็นวิธีการเพิ่มความแปรปรวนในสีของรูปภาพ โดยการเปลี่ยนแปลงค่าของสีหลัก ๆ ในรูปภาพ เช่น โทน (Hue), ความอิ่มตัวสี (Saturation), และความคมชัด (Value) ใน HSV space โดยสุ่ม การปรับค่าพารามิเตอร์เหล่านี้ช่วยให้โมเดลเรียนรู้ว่าวัตถุมีลักษณะสีแตกต่างกันในเงามุมและสภาพแวดล้อมต่าง ๆ การเปลี่ยนแปลงค่าสีเหล่านี้จะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของสีที่หลากหลายของวัตถุในภาพได้

การปรับเปลี่ยนค่าสีของภาพแบบ HSV จะไม่มีผลต่อความสามารถในการตรวจจับวัตถุ เช่น การเปลี่ยนค่าโทน อาจทำให้สีแดงเป็นสีน้ำเงินแต่ยังคงรู้ว่าเป็นสี "หลัก" การลดความซับซ้อนของการเปลี่ยนแปลงการทำงานในปริภูมิ HSV จะช่วยลดความซับซ้อนของการเปลี่ยนแปลงสีเมื่อเทียบกับปริภูมิ RGB ที่มีการผสมสีและการเปลี่ยนแปลงสีที่ซับซ้อนกว่า การใช้ปริภูมิ HSV ช่วยให้การแปลงสีมีผลกับรูปภาพน้อยลง และรักษาความเหมาะสมของสีในบางกรณี และความสว่างของภาพคงที่ การแปลงภาพในปริภูมิ HSV ไม่มีผลต่อความสว่างของภาพ ซึ่งทำให้ความสว่างของวัตถุในภาพคงที่ ซึ่งเป็นสิ่งสำคัญในการตรวจจับวัตถุในเงื่อนไขแวดล้อมที่มีการเปลี่ยนแปลงของแสง ตัวอย่างการปรับเปลี่ยนค่าสีของภาพในปริภูมิ HSV แสดงดังรูปที่ 3.5(d) โดยมีขั้นตอนดังนี้

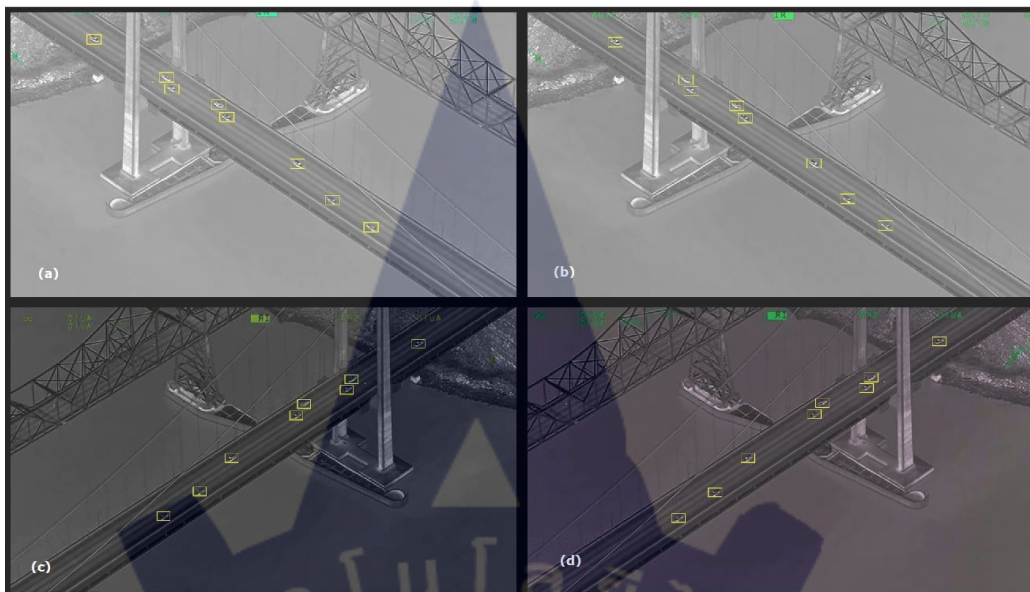
- แปลงภาพจากปริภูมิ RGB เป็นปริภูมิ HSV
- เปลี่ยนค่า H, S, และ V ของภาพตามค่าที่กำหนด
- แปลงภาพกลับจากปริภูมิ HSV เป็นปริภูมิ RGB

3.3.2 การสะท้อนภาพแนวนอนแบบสุม

วิธีนี้เป็นการสะท้อนภาพในแนวนอน เป็นการสับเปลี่ยนภาพจากซ้ายไปขวาหรือจากขวาไปซ้าย ซึ่งเป็นวิธีการเพิ่มข้อมูลที่ถูกทำเป็นประจำในงานตรวจจับวัตถุ การเปลี่ยนแปลงนี้จะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของวัตถุในภาพจากมุมมองที่หลากหลายได้ การทำแบบสุม (50% ขวา/50% ซ้าย) ทำให้โมเดลเรียนรู้ว่าวัตถุอาจปรากฏในทิศทางต่าง ๆ ในรูปภาพ ดังรูปที่ 3.5 (c)

3.3.3 การย่อหรือขยายภาพ 10% แบบสุม

วิธีนี้เป็นการปรับย่อหรือขยายภาพ 10% การเปลี่ยนแปลงนี้จะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของวัตถุในภาพที่มีขนาดแตกต่างกันได้ การปรับสัดส่วนแบบสุมจะช่วยให้โมเดลเรียนรู้การตรวจจับวัตถุที่มีขนาดต่าง ๆ ในภาพ การสุมปรับขนาดรูปภาพในช่วง 90% ถึง 110% ของขนาดเดิม จะทำให้โมเดลสามารถตรวจจับวัตถุที่มีขนาดเล็กหรือใหญ่กว่าของข้อมูลต้นฉบับ และทำให้โมเดลมีความสามารถในการทำงานกับข้อมูลที่มีขนาดต่าง ๆ การใช้เทคนิคการเสริมข้อมูลจะช่วยให้โมเดลมีความสามารถในการทำนายและตรวจจับวัตถุที่ดีขึ้นในสภาพแวดล้อมที่หลากหลาย และช่วยป้องกันปัญหาการเรียนรู้แบบ Overfitting โดยทำให้โมเดลไม่มากเกินไปกินข้อมูลฝึก ในรูปแบบที่เหมือนกับข้อมูลฝึกที่ใช้สำหรับการทดสอบ ตัวอย่างการโปรแกรมเทคนิคการเพิ่มขนาดของข้อมูล อยู่ในภาคผนวก ก. หัวข้อโปรแกรมฟังก์ชันการทำเทคนิคการเพิ่มขนาดของข้อมูล โดยผลของการทำเทคนิคการเสริมข้อมูล แสดงดังรูปที่ 3.5 (a) เป็นรูปที่ผ่านการปรับความสว่าง (b) เป็นรูปภาพต้นแบบของไฟล์รูปภาพ (c) และ (d) เป็นรูปภาพที่ทำการเปลี่ยนทิศทางตามแนวนอน โดยทำให้วัตถุเหมือนถูกถ่ายมุมมองจากทิศทางที่แตกต่าง และทำการย่อขนาดภาพ แปลงค่าสี เป็นต้น



รูปที่ 3.5 ตัวอย่างภาพการเสริมข้อมูล

3.4 การกำหนดการฝึกอบรมล่วงหน้าโมเดล YOLOv4-Tiny

การสร้างแบบจำลองโมเดล ในงานวิจัยนี้เลือกใช้การฝึกอบรมล่วงหน้าโมเดล YOLOv4-Tiny (Pretrained YOLOv4-Tiny) ซึ่งเป็นโมเดลการตรวจจับวัตถุแบบ YOLOv4-Tiny ที่ถูกฝึกก่อนแล้วด้วยชุดข้อมูลใหญ่ บนชุดข้อมูล COCO ดังที่กล่าวถึงในหัวข้อ 2.5 ซึ่งประกอบด้วยภาพและข้อมูลคำอธิบายของวัตถุต่างๆ กว่า 80 ประเภท และมีความสามารถในการตรวจจับวัตถุในภาพ YOLOv4-Tiny เป็นเวอร์ชันย่อยขนาดเล็กของ YOLOv4 ซึ่งถูกออกแบบมาเพื่อให้มีความสามารถสูงและทำงานได้อย่างรวดเร็วบนอุปกรณ์ที่มีทรัพยากรจำกัด การตั้งค่าการเรียนรู้ของโมเดลตรวจจับวัตถุ ได้ทำการกำหนดคลาสในการตรวจสอบเพียงแค่ 1 คลาส คือ รถยนต์ รายละเอียดของขั้นตอนต่างๆ มีดังนี้

3.4.1 การติดตั้งแพ็คเกจ YOLOv4-Tiny

แพ็คเกจ YOLOv4-Tiny สามารถใช้ได้ฟรีซึ่งสามารถดาวน์โหลดได้จากเว็บไซต์ โดยสามารถติดตั้งได้จากเมนูของโปรแกรม MATLAB โดยไปที่เมนู Home > Add-Ons > Get Add-Ons

3.4.2 การกำหนดขนาดอินพุตของข้อมูลรูปภาพ

รูปภาพมีขนาดจริงเป็น 854 x 480 พิกเซล แต่ขนาดอินพุตของข้อมูลรูปภาพที่ใช้สำหรับการฝึกโมเดล YOLOv4-Tiny มีค่าเริ่มต้นกำหนดเป็น 416x416 พิกเซล ซึ่งเป็นขนาดอินพุตเริ่มต้นของโมเดลนี้ หากวัตถุที่ต้องการตรวจจับมีขนาดเล็กหรืออยู่ไกล โมเดลอาจจำเป็นต้องใช้ขนาด

อินพุตที่ใหญ่ขึ้น เพื่อให้สามารถตรวจจับวัตถุเหล่านี้ได้อย่างแม่นยำมากขึ้น เช่น กรณีที่ต้องการตรวจจับใบหน้าที่กำลังเดินอยู่ในระยะไกล โมเดลอาจจำเป็นต้องใช้ขนาดอินพุตที่ 608×608 พิกเซล หรือมากกว่านั้น ในทางกลับกัน หากวัตถุที่ต้องการตรวจจับมีขนาดใหญ่หรืออยู่ใกล้ โมเดลอาจไม่จำเป็นต้องใช้ขนาดอินพุตที่ใหญ่มากนัก เช่น กรณีที่ต้องการตรวจจับรถยนต์บนท้องถนน โมเดลอาจสามารถใช้ขนาดอินพุตที่ 288×288 พิกเซล หรือน้อยกว่านั้น

จากจุดประสงค์ของงานวิจัย เพื่อให้ได้โมเดลที่มีความสามารถสูงสุดและสอดคล้องกับความต้องการ ดังนั้นงานวิจัยนี้ได้กำหนดขนาดรูปภาพอินพุตให้กับโมเดล YOLOv4-Tiny เป็น 3 ขนาดคือ 288×288 พิกเซล, 416×416 พิกเซล และ 608×608 พิกเซล เพื่อเปรียบเทียบ ความสามารถของโมเดล ที่อินพุตขนาดของภาพต่างกัน

3.5 การระบุตัวเลือกออฟติไมเซอร์ ของการฝึกอบรม

การระบุตัวเลือกการฝึกอบรม คือกระบวนการกำหนดพารามิเตอร์ต่างๆ ที่ใช้ตัวเลือกการฝึกอบรมเหล่านี้สามารถส่งผลต่อความสามารถของโมเดล ดังนั้นการกำหนดตัวเลือกที่เหมาะสมกับชุดข้อมูลและเป้าหมายของการฝึกอบรมจึงมีความสำคัญ งานวิจัยนี้ได้แบ่งการระบุตัวเลือกการฝึกอบรมออกเป็น 2 ชุด และมีการกำหนดค่าพารามิเตอร์สำหรับการฝึกอบรมตัวตรวจจับ YOLO v4 โดยใช้ อัลกอริทึมการฝึกอบรมแบบ SGDM เป็นขั้นตอนสำคัญในการให้โมเดลสามารถเรียนรู้ได้อย่างมีประสิทธิภาพ การกำหนดค่าพารามิเตอร์ที่เหมาะสมควรคำนึงถึงค่าต่าง ๆ ดังนี้

1) อัตราการเรียนรู้ (Learning Rate)

เป็นตัวกำหนดขนาดของ Step ในการ Update น้ำหนักของโมเดล ซึ่งมีความสำคัญในการตรวจจับวัตถุ ควรกำหนดให้มีค่าไม่เยอะเกินไปหรือน้อยเกินไป คำสั่งใน MATLAB จะใช้ InitialLearnRate ซึ่งมีค่าเริ่มต้นประมาณ 0.001 ถือเป็นค่าเริ่มต้นที่ดี และควรทดลองปรับแต่งให้เหมาะกับงานและชุดข้อมูล

2) การลดขนาดรุ่นการผลิต

เป็นขนาดของ Batch ที่ใช้สำหรับการคำนวณค่าเกรเดียนต์ ในแต่ละรอบ ค่าที่เหมาะสมขึ้นอยู่กับขนาดของชุดข้อมูล และทรัพยากรฮาร์ดแวร์ที่ใช้ คำสั่งใน MATLAB จะใช้ MiniBatchSize ซึ่ง ค่าที่ใช้สำหรับงานตรวจจับวัตถุ จะอยู่ที่ประมาณ 16 ถึง 32 ขึ้นอยู่กับขนาดของชุดข้อมูลและความสามารถของฮาร์ดแวร์

3) L2 Regularization

เป็นเทคนิคที่ใช้เพื่อลด Overfitting โดยการเพิ่มค่าให้กับฟังก์ชันสูญเสีย เพื่อลดขนาดของน้ำหนักของโมเดล คำสั่ง MATLAB จะใช้ L2Regularization ซึ่งโดยทั่วไป จะมีค่าประมาณ 0.0001 ถึง 0.001 สำหรับงานด้านการตรวจจับวัตถุ

4) จำนวนรอบการฝึกสูงสุด (Maximum Epochs)

เป็นจำนวนรอบการฝึกสูงสุด ที่ต้องการให้โมเดลฝึก คำสั่งใน MATLAB จะใช้ MaxEpochs ซึ่งเป็นตัวกำหนดจำนวนรอบของการฝึกอบรม โมเดลจะเรียนรู้ในชุดข้อมูล โดยการปรับค่าน้ำหนักของโมเดลในแต่ละรอบ ค่า MaxEpochs ที่สูงขึ้น จะทำให้โมเดลมีเวลาในการเรียนรู้ข้อมูลใน Dataset ได้อย่างเพียงพอ ซึ่งจะช่วยในการปรับปรุงความสามารถของโมเดล

5) การสลับข้อมูล (Shuffling Data)

การสลับข้อมูลในแต่ละรอบการฝึกอบรม ช่วยลดความเกี่ยวข้องของข้อมูล และเพิ่มการสุ่มในการเรียนรู้ คำสั่งใน MATLAB จะใช้ Shuffle ควรกำหนดให้ค่าเป็น "every-epoch" เพื่อที่ข้อมูลจะถูกสลับในแต่ละรอบการฝึก

6) การปรับค่าข้อมูลให้อยู่ในขอบเขตที่กำหนด (Batch Normalization Statistics)

เป็นคำสั่งใน MATLAB จะใช้ BatchNormalizationStatistics ซึ่งการใช้ "moving" เป็นค่าเริ่มต้นจะช่วยให้โมเดลเรียนรู้ได้อย่างมีประสิทธิภาพ

3.5.1 การเลือกใช้อัลกอริทึมการฝึกอบรมแบบ SGDM

ทำการฝึกอบรมตัวตรวจจับ YOLO v4 โดยกำหนดค่าพารามิเตอร์ต่างๆสำหรับการฝึกอบรม ในส่วนของโปรแกรมจะกำหนดตัวแปรเป็น “options1” โดยกำหนดค่าการเลือกใช้ออปติไมเซอร์ เป็นแบบ SGDM และกำหนดค่าอัตราการเรียนรู้ เป็น 0.0001 โดยกำหนดค่าขนาดของ Batch ที่ใช้สำหรับการคำนวณค่าเกรเดียนต์ ในแต่ละรอบ โดยค่า MiniBatchSize ที่เหมาะสมขึ้นอยู่กับขนาดของโมเดลและขนาดของชุดข้อมูล ในงานวิจัยนี้ขนาดข้อมูลมีจำนวนไม่เกิน 1000 รูปภาพ จึงเหมาะสมกับการใช้ค่าเริ่มต้นด้วยค่า MiniBatchSize ขนาดเล็ก เช่น 16 สำหรับการกำหนดค่า L2Regularization เป็น 0.0005 เพื่อให้มีการลดน้ำหนักของพารามิเตอร์ที่เหมาะสม เนื่องจากจะทำให้โมเดลสามารถเรียนรู้ข้อมูลในชุดข้อมูล ได้อย่างมีประสิทธิภาพ โดยไม่ทำให้โมเดลเกิดการ Overfitting และกำหนดจำนวนรอบของการฝึกอบรม (MaxEpochs) เท่ากับ 100 รอบ การกำหนดค่าพารามิเตอร์ แสดงดังตารางที่ 3.3

ตารางที่ 3.3 การตั้งค่าพารามิเตอร์ SGDM options1

Configuration	Description
trainingOptions	“SGDM”
InitialLearnRate	0.0001
MiniBatchSize	16
L2Regularization	0.0005

ตารางที่ 3.3 การตั้งค่าพารามิเตอร์ SGDM options1 (ต่อ)

Configuration	Description
MaxEpochs	100

การเลือกใช้อัลกอริทึมการฝึกอบรมแบบ SGDM เหมาะสมสำหรับการฝึกโมเดลในรูปแบบทั่วไปที่ไม่มีความซับซ้อนมาก โมเดลที่มีจำนวนพารามิเตอร์ไม่มากจนเกินไปและข้อมูลฝึกอบรมมีปริมาณค่าข้อมูลแบบปานกลางถึงค่าข้อมูลมาก การใช้ SGDM ในกรณีนี้ช่วยให้การฝึกมีความเร็วและเสถียรมากขึ้น SGDM ทำงานโดยการคำนวณค่าเกรเดียนต์ ของฟังก์ชันสูญเสียในแต่ละกลุ่ม แล้วทำการปรับน้ำหนักของโมเดลตามทิศทางของเกรเดียนต์ โดยใช้โมเมนตัม เพื่อเพิ่มความเร็วในการลู่เข้า (Converge) ดังสมการในหัวข้อที่ 2.7

3.5.2 การเลือกใช้อัลกอริทึมการฝึกอบรมแบบ ADAM

เพื่อใช้ฝึกอบรมตัวตรวจจับ YOLO v4 มีการกำหนดค่าพารามิเตอร์ต่างๆสำหรับการฝึกอบรม โดยกำหนดเลือกoptimizerเป็น ADAM ในส่วนของโปรแกรมจะกำหนดตัวแปรเป็น “options2” และกำหนดค่าอัตราการเรียนรู้ เป็น 0.0001 โดยค่า MiniBatchSize ขนาดเล็กกำหนดเป็นค่าเริ่มต้นที่ 16 และ L2Regularization เป็น 0.0005 โดยการกำหนดพารามิเตอร์ของ ADAM จะมีการกำหนดอีก 2 ค่าเพิ่มเข้ามา คือค่า Gradient Decay Factor เป็น 0.9 และ Squared Gradient Decay Factor เป็น 0.999 ค่าเกรเดียนต์ในแต่ละรอบจะถูกลดลง 10% เมื่อใช้ในการอัปเดตน้ำหนักของโมเดล การลดค่าเกรเดียนต์ช่วยในการควบคุมอัตราการเรียนรู้ของโมเดล การใช้ค่านี้ช่วยให้กระบวนการฝึกเสถียรและไม่แย่งกับความเร็วในการลดค่าของ เกรเดียนต์ที่มีค่ามากเกินไป โดยที่ยังคงเรียนรู้ได้อย่างมีประสิทธิภาพ และกำหนดจำนวนรอบของการฝึกอบรม (MaxEpochs) เท่ากับ 100 รอบ ค่าพารามิเตอร์ ADAM ที่กำหนดแสดงดังตารางที่ 3.4

ตารางที่ 3.4 การตั้งค่าพารามิเตอร์ ADAM options2

Configuration	Description
trainingOptions	"ADAM"
InitialLearnRate	0.0001
MiniBatchSize	16
L2Regularization	0.0005
GradientDecayFactor	0.9

ตารางที่ 3.4 การตั้งค่าพารามิเตอร์ ADAM options2 (ต่อ)

Configuration	Description
MaxEpochs	100

ค่าพารามิเตอร์เหล่านี้เกี่ยวข้องกับการควบคุมการปรับปรุงของค่าเกรเดียนต์ในออฟติไมเซอร์ ADAM ซึ่งมีบทบาทในการควบคุมการลดค่าเรียนรู้ ช่วยให้สามารถควบคุมการลดค่าเกรเดียนต์ของโมเดลในแต่ละรอบของการฝึกเพื่อปรับน้ำหนักให้มีประสิทธิภาพและเสถียร อัตราการเรียนรู้ของ ADAM จะปรับตามค่าของโมเมนตัม เพื่อให้การเรียนรู้เป็นไปอย่างรวดเร็วหรือช้าตามลักษณะของข้อมูล การปรับค่าอัตราการเรียนรู้จะช่วยให้ ADAM ทำงานได้อย่างมีประสิทธิภาพในงานตรวจจับวัตถุที่มีข้อมูลที่ซับซ้อน

3.6 การฝึกอบรมโมเดล YOLOv4-Tiny

การฝึกอบรมโมเดล YOLOv4-Tiny ที่ได้กำหนดค่าพารามิเตอร์เรียบร้อยแล้วจากขั้นตอนข้างต้น ในขั้นตอนนี้จะทำการฝึกอบรมโมเดลตัวตรวจจับ โมเดล YOLOv4-Tiny ทั้งหมด 8 ครั้ง โดยชุดข้อมูลที่ใช้ในการฝึกอบรมจะแบ่งเป็น 2 ชุด ดังนี้

3.6.1 การฝึกอบรมด้วยชุดข้อมูลปกติ

งานวิจัยนี้จะใช้แทนคำว่า Data Normal คือ ชุดข้อมูลภาพที่ไม่ผ่านการทำเทคนิคการเสริมข้อมูล ในส่วนของโปรแกรมจะแทนด้วยตัวแปรชื่อ “TrainingData” โดยมีขั้นตอนการฝึกอบรมตัวตรวจจับวัตถุด้วยโมเดล YOLOv4-Tiny การฝึกอบรมจะแบ่งตามการกำหนดค่าอินพุตของโมเดล ดังนี้

3.6.1.1 ขนาดภาพอินพุตของโมเดลที่ [416 416 3]

ค่าเริ่มต้นของโมเดล YOLOv4-Tiny จะใช้ขนาดที่ [416 416 3] โดยขั้นตอนการกำหนดค่าพารามิเตอร์ Option ที่จะทำการฝึกอบรมด้วยออฟติไมเซอร์ จะปรับปรุงตัวเลือกของโมเดล ดังนี้

- ตัวเลือกที่ 1 จะกำหนดเป็นออฟติไมเซอร์ การปรับปรุงแบบ SGDM
- ตัวเลือกที่ 2 จะกำหนดเป็นออฟติไมเซอร์ การปรับปรุงแบบ ADAM

3.6.1.2 ขนาดภาพอินพุตของโมเดลที่ [288 288 3]

ขนาดภาพอินพุตโมเดลจะมีขนาดลดลง กรณีที่ต้องการตรวจจับรถยนต์บนถนน การกำหนดค่าพารามิเตอร์ Option ที่จะทำการฝึกอบรมด้วยออฟติไมเซอร์ จะปรับปรุงโมเดลแบบ 1 ตัวเลือก ดังนี้

- ตัวเลือกที่ 2 จะกำหนดเป็นออฟติไมเซอร์ การปรับปรุงแบบ ADAM

3.6.1.3 ขนาดภาพอินพุตของโมเดลที่ [603 603 3]

ขนาดภาพอินพุตโมเดลจะเพิ่มขึ้น กรณีที่ต้องการตรวจจับวัตถุอยู่ในระยะไกล การกำหนดค่าพารามิเตอร์ Option ที่จะทำให้การฝึกอบรมด้วยออฟติไมเซอร์จะปรับปรุงตัวเลือก ดังนี้

- ตัวเลือกที่ options2 จะกำหนดเป็น Optimizer การปรับปรุงแบบ ADAM

3.6.2 การฝึกอบรมด้วยชุดข้อมูลที่ทำเทคนิคการเสริมข้อมูล

ส่วนของโปรแกรมจะแทนด้วยตัวแปรชื่อ “augmentedTrainingData” โดยขั้นตอนการฝึกอบรมตัวตรวจจับวัตถุด้วยโมเดล YOLOv4-Tiny จะแบ่งตามการกำหนดค่าอินพุตของโมเดล ดังนี้

3.6.2.1 ขนาดภาพอินพุตของโมเดลที่ [416 416 3]

ขนาดภาพอินพุตนี้เป็นค่าเริ่มต้นของโมเดล YOLOv4-Tiny การกำหนดค่าพารามิเตอร์ Option ที่จะทำให้การฝึกอบรมด้วยออฟติไมเซอร์ มีรายละเอียดดังนี้

- ตัวเลือกที่ จะกำหนดเป็นออฟติไมเซอร์ การปรับปรุงแบบ SGDM
- ตัวเลือกที่ จะกำหนดเป็นออฟติไมเซอร์ การปรับปรุงแบบ ADAM

3.6.2.2 ขนาดภาพอินพุตของโมเดลที่ [288 288 3]

ภาพอินพุตโมเดลจะมีขนาดลดลง กรณีที่ต้องการตรวจจับรถยนต์บนถนน การกำหนดค่าพารามิเตอร์ Option ที่จะทำให้การฝึกอบรมด้วยออฟติไมเซอร์ มีรายละเอียดดังนี้

- ตัวเลือกที่ 2 จะกำหนดเป็นออฟติไมเซอร์ การปรับปรุงแบบ ADAM

3.6.2.3 ขนาดภาพอินพุตของโมเดลที่ [603 603 3]

ภาพอินพุตโมเดลจะมีขนาดเพิ่มขึ้น กรณีที่ต้องการตรวจจับวัตถุอยู่ในระยะไกล ขั้นตอนการกำหนดค่าพารามิเตอร์ Option ที่จะทำให้การฝึกอบรมด้วยออฟติไมเซอร์ มีรายละเอียดดังนี้

- ตัวเลือกที่ 2 จะกำหนดเป็นออฟติไมเซอร์ การปรับปรุงแบบ ADAM

3.6.3 ขั้นตอนการฝึกอบรมตัวตรวจจับวัตถุด้วยโมเดล YOLOv4-Tiny

ใช้ฟังก์ชัน trainYOLOv4ObjectDetector ใน MATLAB เพื่อฝึกอบรมตัวตรวจจับวัตถุแบบ YOLOv4-Tiny ผลลัพธ์ของการฝึกอบรมจะแสดงออกมาในรูปแบบตารางดังรูปที่ 3.6 ซึ่งแสดงรายละเอียดของรายงานที่เกี่ยวกับการฝึกอบรม เช่น จำนวนรอบการฝึกอบรม, เวลาที่ใช้, ค่าสูญเสีย (Loss) ของการฝึกอบรม และค่าสูญเสียของการทดสอบ ดังรูปที่ 3.6

```

111 The training time will vary depending on the hardware you use.

[detector,info] = trainYOLOv4ObjectDetector(augmentedTrainingData,detector,options2);

*****
Training a YOLO v4 Object Detector for the following object classes:

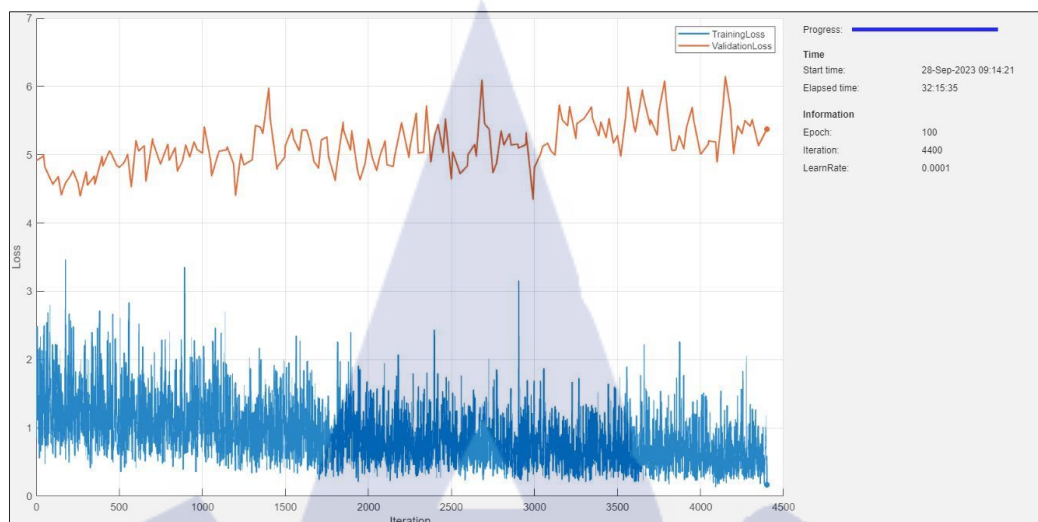
* vehicle

Epoch      Iteration    TimeElapsed    LearnRate    TrainingLoss    ValidationLoss
-----
0           0           0             0            0              0
Starting parallel pool (parpool) using the 'Processes' profile ...
Connected to the parallel pool (number of workers: 4).
1           20         00:12:45      0.0001       1.3234
1           40         00:17:15      0.0001       1.0008
2           60         00:31:23      0.0001       0.7481
2           80         00:36:16      0.0001       1.9025
3          100         00:49:41      0.0001       0.6252          4.5703
3          120         00:56:09      0.0001       0.75486

```

รูปที่ 3.6 ตัวอย่างขั้นตอนการฝึกอบรมตัวตรวจจับวัตถุด้วยโมเดล YOLOv4-Tiny

โดยที่ ‘augmentedTrainingData’ เป็นข้อมูลการฝึกอบรมที่ถูกเสริมค่าความหลากหลายด้วยเทคนิค Data Augmentation เพื่อเพิ่มความหลากหลายของข้อมูลการฝึกอบรม ‘detector’ เป็นโมเดลตัวตรวจจับวัตถุเริ่มต้นหรือโมเดลที่ได้รับการกำหนดค่าล่วงหน้าข้างต้น ‘options2’ เป็นตัวเลือกการฝึกอบรมเช่นจำนวนรอบ, การลดความเร็วในระหว่างการฝึกอบรม, และอื่น ๆ ตามที่ได้กำหนดไว้ในหัวข้อที่ 3.5 ผลลัพธ์หลังจากฝึกอบรมโมเดลเสร็จแสดงค่าดังรูปที่ 3.7 ซึ่งมีรายละเอียดระยะเวลาที่ใช้ในการฝึกอบรมโมเดล และจำนวนรอบในการฝึกอบรมทั้งหมด 100 รอบ การส่งชุดข้อมูลฝึกทั้งหมดผ่านโมเดลหนึ่งครั้งในทุกกรอบ ยังมี epoch มากเท่าไร โมเดลจะได้เรียนรู้มากขึ้น โดยที่การวนซ้ำ (Iteration) จะระบุการอัปเดตน้ำหนักของโมเดล แต่ครั้งที่ผ่านไป ทั้งหมดของชุดข้อมูลฝึกอบรม โดยมีรูปแบบของเวลาเป็น ชั่วโมง:นาที:วินาที ตั้งแต่เริ่มต้นการฝึกอบรมจนถึงครบ 100 รอบของการฝึกอบรม ดังรูปที่ 3.7



รูปที่ 3.7 ตัวอย่างกราฟผลการฝึกอบรมตัวตรวจจับวัตถุด้วยโมเดล YOLOv4-Tiny

3.7 การฝึกอบรมโมเดล

การฝึกอบรมโมเดลทั้งหมด แบ่งออกเป็นการกำหนดออฟติไมเซอร์ แบบ SGDM และ ADAM เพื่อใช้ฝึกอบรมกับชุดข้อมูลปกติที่ไม่ผ่านการทำเทคนิคการเสริมข้อมูลและ ชุดข้อมูลที่ทำเทคนิคการเสริมข้อมูล ทั้ง 3 วิธี และขนาดอินพุตของโมเดล YOLOv4-Tiny แบ่งเป็น 3 ขนาด เพื่อทำการฝึกอบรมโมเดล เนื่องจากขนาดของรูปภาพต้นแบบมีขนาดเป็น 854×480 พิกเซล แต่ขนาดอินพุตของข้อมูลรูปภาพที่ใช้สำหรับการฝึกโมเดล YOLOv4-Tiny มีค่าเริ่มต้นกำหนดเป็น 416×416 พิกเซล จึงทำการกำหนดขนาดอินพุตโมเดล YOLOv4-Tiny ที่ใช้ในการฝึกอบรมออกเป็น 288×288 Pixel, 416×416 Pixel และ 608×608 Pixel เมื่อทำการฝึกอบรมทั้งหมดเรียบร้อยแล้ว ชุดข้อมูลของรูปภาพที่ใช้เป็นชุดทดสอบ ซึ่งแบ่งไว้จำนวน 84 รูปภาพ และเป็นรูปภาพที่ไม่เคยถูกนำไปฝึกอบรมโมเดลมาก่อน เมื่อทำการทดสอบด้วยตัวตรวจจับ YOLOv4-Tiny ที่ถูกฝึกอบรมโมเดลเสร็จเรียบร้อยแล้ว ผลที่ได้จะถูกวิเคราะห์ผลของการฝึกอบรมโมเดลในแต่ละแบบเปรียบเทียบเป็นตารางในผลการทดลอง ดังตารางเปรียบเทียบผลการทดลองในบทที่ 4

3.7.1 การทดสอบโมเดลตรวจจับวัตถุด้วยรูปภาพทดสอบ

การทดสอบโมเดลตัวตรวจจับวัตถุ YOLOv4-Tiny ที่ผ่านการฝึกอบรม บนรูปภาพทดสอบ เริ่มจากขั้นตอนการทดสอบโมเดลกับรูปภาพ Test Data ที่ได้ทำการแบ่งไว้ มีจำนวนทั้งหมด 87 รูปภาพ ตัวอย่างรูปภาพ และชื่อไฟล์รูปภาพ ที่ใช้ในการทดสอบ คือ videoplayback1_0611.png การทดสอบตัวตรวจจับวัตถุบนรูปภาพทดสอบโดยมีตัวอย่างโปรแกรมแสดงดังรูปที่ 3.8

	Run the detector YOLOv4-Tiny ที่ถูกเทรนเรียบร้อยแล้ว ด้วยรูปภาพทดสอบ.
120	<code>I = imread("videoplayback1_0611.png");</code>
121	<code>[bboxes,scores,labels] = detect(detector,I);</code>

รูปที่ 3.8 ตัวอย่างโปรแกรมการทดสอบโมเดลตรวจจับวัตถุด้วยรูปภาพทดสอบ

ผลลัพธ์บนรูปภาพทดสอบด้วยโปรแกรมตัวอย่าง แสดงดังรูปที่ 3.9 โดยที่ตัวพารามิเตอร์ bboxes ในคำสั่ง เป็นตำแหน่งของกล่องขอบเขตที่พบวัตถุและ scores เป็นคะแนนความถูกต้องในการตรวจจับวัตถุ จะมีค่าสูงสุดที่ 1 เมื่อโมเดลสามารถจำแนกประเภทได้อย่างถูกต้องทั้งหมด และป้ายกำกับวัตถุที่ตรวจจับได้จะแสดงดังรูปที่ 3.9



รูปที่ 3.9 ผลการทดสอบการตรวจจับรถยนต์ด้วยภาพทดสอบ videoplayback1_0611.png

3.7.2 การประเมินผลของโมเดล

การประเมินผลของโมเดล มีดังนี้ คำนวณความแม่นยำเฉลี่ยกับรูปภาพทดสอบทั้งหมด และวิเคราะห์ผลการประเมินผล เปรียบเทียบกับพารามิเตอร์ต่างๆที่กำหนดไว้ โดยใช้ค่าความแม่นยำเฉลี่ย หรือ AP เป็นตัววัดผลการประเมินความสามารถของโมเดล ซึ่งเป็นค่าเฉลี่ยของอัตราส่วนของวัตถุที่ตรวจจับได้ถูกต้องต่อจำนวนวัตถุทั้งหมดที่ตรวจจับได้ (Precision) และอัตราส่วนของวัตถุที่ตรวจจับได้ถูกต้องต่อจำนวนวัตถุทั้งหมดที่มีอยู่ในภาพ (Recall) ซึ่งเป็น Metrics ที่ใช้ใน

การประเมินผล วัดประสิทธิภาพของโมเดลตรวจจับรถยนต์ขนาดเล็กแบบขั้นตอนเดียว ที่ใช้ YOLOv4-Tiny สำหรับภาพความละเอียดต่ำจากโดรน ในงานวิจัยนี้ได้ทำการเปรียบเทียบค่าผลการฝึกอบรมโมเดลที่ค่าพารามิเตอร์ต่างๆ กับชุดข้อมูลจำนวน 2 ชุด

การคำนวณความแม่นยำเฉลี่ย ซึ่งเป็นตัวชี้วัดเพื่อประเมินความสามารถของโมเดล YOLOv4-Tiny ที่ได้ถูกฝึกอบรม ซึ่งเป็นเมตริกในการประเมินประสิทธิภาพของตัวตรวจจับวัตถุ คือ การให้คะแนนแบบเดียวที่รวมความสามารถของตัวตรวจจับวัตถุในการทำการจำแนกอย่างถูกต้อง และความสามารถในการค้นหาวัตถุทั้งหมด ขั้นตอนในการประเมินประสิทธิภาพของโมเดลการตรวจจับวัตถุโดยใช้เมตริกความแม่นยำเฉลี่ย สามารถสรุปได้ดังนี้

3.7.2.1. ทดสอบโมเดลการตรวจจับวัตถุ

การทดสอบโมเดลการตรวจจับวัตถุที่ได้ทำการฝึกอบรมโมเดลตามค่าพารามิเตอร์ต่างๆที่ปรับ ตามเงื่อนไขที่กำหนดในหัวข้อที่ 3.6 กับรูปภาพทดสอบทั้งหมด คำสั่งใน MATLAB สามารถใช้ฟังก์ชัน `detect()` เพื่อทดสอบโมเดลการตรวจจับวัตถุกับรูปภาพทดสอบทั้งหมด ตัวอย่างโปรแกรมแสดงดังรูปที่ 3.10

3.7.2.2. ประเมินความสามารถของโมเดลตรวจจับวัตถุ

โดยใช้เมตริกความแม่นยำเฉลี่ยคำสั่งใน MATLAB สามารถใช้ฟังก์ชัน `evaluateDetectionPrecision()` เพื่อประเมินความสามารถของโมเดลการตรวจจับวัตถุโดยใช้เมตริกความแม่นยำเฉลี่ย ตัวอย่างโปรแกรมแสดงดังรูปที่ 3.10

	Run the detector on all the test images.
125	<code>detectionResults = detect(detector,testData);</code>
	Evaluate the object detector using average precision metric.
126	<code>[ap,recall,precision] = evaluateDetectionPrecision(detectionResults,testData);</code>

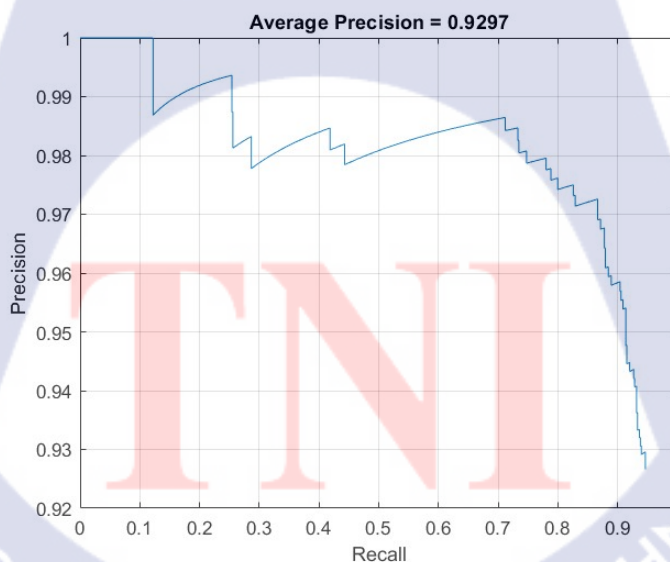
รูปที่ 3.10 ตัวอย่างการทดสอบการประเมินความสามารถของโมเดล

ตัวแปร `ap` ในฟังก์ชันที่ใช้จะเป็นคะแนนความแม่นยำเฉลี่ย ซึ่งเป็นตัวเลขเดียวที่รวมเอาความแม่นยำและความสามารถในการค้นหาวัตถุทั้งหมดของโมเดลการตรวจจับวัตถุ ส่วนตัวแปร `recall` และ `precision` จะเป็นคะแนนความสามารถในการค้นหาวัตถุทั้งหมดและความแม่นยำ ตามลำดับ ในขั้นตอนนี้ การประเมินผลลัพธ์จากตัวตรวจจับวัตถุด้วยการใช้ค่าความแม่นยำเฉลี่ย และกราฟความแม่นยำเฉลี่ย (Precision/Recall) ที่สามารถ plot ได้ตั้งขั้นตอนถัดไป

3.7.2.3 การประเมินความสามารถของเครื่องตรวจจับวัตถุ

กราฟเส้นโค้งความแม่นยำต่อความสามารถในการค้นหาวัตถุทั้งหมด (Precision/Recall curve) เป็นกราฟที่แสดงความสัมพันธ์ระหว่างความแม่นยำ (Precision) และ อัตราการพบวัตถุ (Recall) ที่ระดับต่างๆ ค่าความแม่นยำและ Recall จะมีค่าสูงสุดที่ 1 เมื่อโมเดลสามารถจำแนกประเภทได้อย่างถูกต้องทั้งหมด การใช้ข้อมูลจำนวนมากขึ้นสามารถช่วยปรับปรุงความแม่นยำเฉลี่ยได้ แต่ระยะเวลาการฝึกอบรมจะมากขึ้นเช่นกัน ตัวอย่างเช่น หากโมเดลสามารถตรวจจับรถยนต์ได้อย่างถูกต้องทั้งหมด 100 คัน จากระดยนต์ทั้งหมด 100 คัน ค่าความแม่นยำและ Recall ของโมเดลจะเท่ากับ 1 ในทางกลับกัน หากโมเดลตรวจจับรถยนต์ได้เพียง 50 คัน จากระดยนต์ทั้งหมด 100 คัน ค่าความแม่นยำและ Recall ของโมเดลจะเท่ากับ 0.5 จึงทำให้ค่าความแม่นยำและ Recall เป็นค่าที่มีประโยชน์ในการเปรียบเทียบความสามารถของโมเดลต่างๆ ร่วมกัน ขั้นตอนการพล็อตเส้นโค้ง PR Curve มีรายละเอียดดังนี้

- 1) คำนวณความแม่นยำและอัตราการพบวัตถุที่ระดับต่างๆ
- 2) นำค่าความแม่นยำและอัตราการพบวัตถุที่ได้มาพล็อตลงบนกราฟ
- 3) เชื่อมจุดที่พล็อตทั้งหมดด้วยเส้น



รูปที่ 3.11 กราฟการพล็อตค่า RP Curve

ตัวอย่างกราฟเส้นโค้ง PR ของเครื่องตรวจจับวัตถุที่ถูกฝึกอบรมด้วยรูปภาพที่ทำเทคนิค Data Augmentation และใช้ออปติไมเซอร์แบบ ADAM โดยมีขนาด Input รูปภาพเป็น

603 x 603 พิกเซล แสดงดังรูปที่ 3.11 พบว่าเส้นโค้ง PR อยู่ใกล้เคียงกับเส้นทแยงมุมมาก ซึ่งหมายความว่าเครื่องตรวจจับวัตถุนี้มีประสิทธิภาพสูง ทั้งในด้านความแม่นยำและอัตราการพบวัตถุ



บทที่ 4

ผลการวิจัย

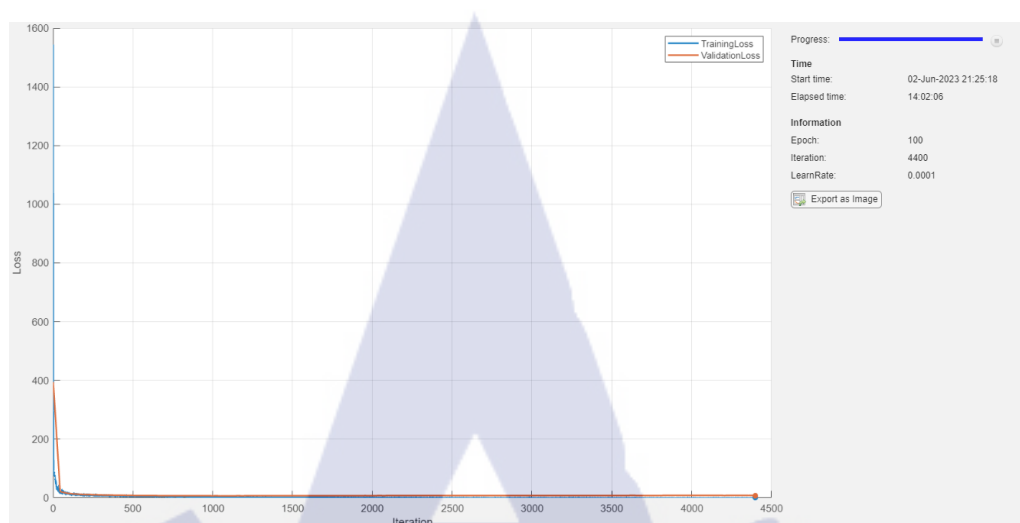
จากการฝึกอบรมโมเดลการเรียนรู้เชิงลึก โดยนำอัลกอริทึม YOLOv4-Tiny ที่ผู้วิจัยได้ทำการปรับแต่งค่าพารามิเตอร์ต่างๆ มาฝึกอบรมด้วยภาพความละเอียดต่ำจากโดรนที่ถูกแบ่งออกเป็น 2 ชุดข้อมูล คือข้อมูลภาพปกติที่ไม่ผ่านและผ่านการทำเทคนิคการเสริมขนาดข้อมูลทั้ง 3 แบบ (การปรับเปลี่ยนค่าสีของภาพในปริภูมิ HSV การสะท้อนภาพแนวนอนแบบสุ่ม และการย่อหรือขยายภาพ 10% แบบสุ่ม) โดยแบ่งผลจากการทดลองออกเป็น 2 หัวข้อ ผลการวิจัยของการฝึกอบรมโมเดลตัวตรวจจับวัตถุ YOLOv4-Tiny ทั้งหมด 8 ครั้ง สามารถแบ่งการวิเคราะห์ผลการทดลอง ดังต่อไปนี้

4.1 ผลจากการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยชุดข้อมูลภาพปกติที่ไม่ผ่านการทำเทคนิคการเสริมขนาดข้อมูล

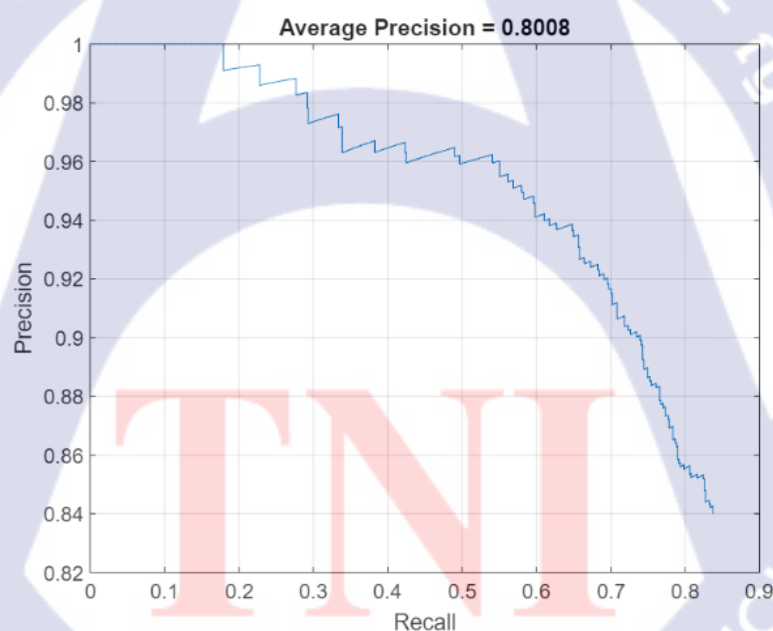
ผลจากการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยชุดข้อมูลภาพปกติ ที่ไม่ผ่านการทำเทคนิคการเสริมขนาดข้อมูล จะแบ่งการวิเคราะห์ผลการอบรม โดยแยกตามการกำหนดค่าอินพุตของโมเดล

4.1.1) ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล

ผลการฝึกสอนโมเดล YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบ ขนาด 416x416 พิกเซล แสดงรูปที่ 4.1 เส้นกราฟของ Training Loss จะลดลงเมื่อเวลาเพิ่มขึ้น แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่ ส่วน Validation Loss จะลดลงเรื่อยๆ เช่นกัน แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่เช่นกัน โดยใช้ระยะเวลาในการฝึกสอน 14 ชั่วโมง 2 นาที 6 วินาที กับรอบการฝึก 100 รอบ และมีค่าอัตราการเรียนรู้เป็น 0.0001 และ Iteration เท่ากับ 4000 ส่วนผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยรูปภาพทดสอบแสดงดังรูปที่ 4.2 พบว่า ความสามารถของโมเดลที่ผ่านการฝึกอบรมโดยพิจารณาจากทั้งค่าความแม่นยำและ ค่าความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมดมีค่าเฉลี่ยที่ 0.8008



รูปที่ 4.1 ผลการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM ที่ข้อมูลภาพปกติ ขนาด 416x416 พิกเซล



รูปที่ 4.2 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล

ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล ด้วยไฟล์ภาพที่ videoplayback1_0611.png ดังขั้นตอนในหัวข้อ 3.7 แสดงดังรูปที่ 4.3 พบว่า ตำแหน่งที่รถอยู่ใกล้ทางด่วน 1 จุดไม่สามารถ

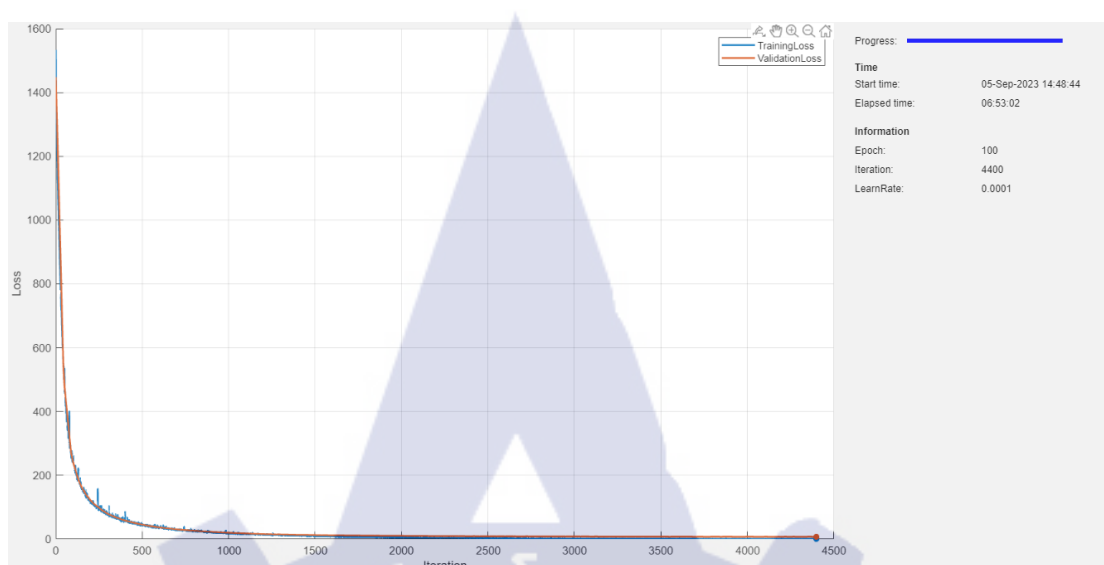
ตรวจจับได้ และตำแหน่งที่ตรวจพบอีก 1 จุดที่มี Scores เท่ากับ 0.649 ซึ่งอยู่ต่ำกว่าค่าเฉลี่ยและโมเดลสามารถตรวจจับรถยนต์ได้ทั้งหมด 7 คัน จากทั้งหมด 8 คันในภาพ



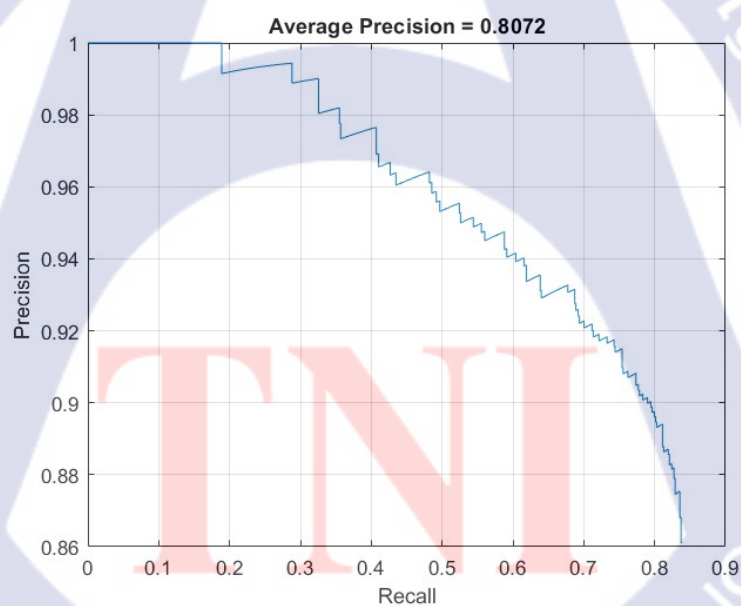
รูปที่ 4.3 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล

4.1.2) ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล

ผลการฝึกสอนโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบขนาด 416x416 พิกเซล แสดงดังรูปที่ 4.4 เส้นกราฟของ Training Loss จะลดลงเมื่อเวลาเพิ่มขึ้น แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่ ส่วน Validation Loss จะลดลงเรื่อยๆ เช่นกัน แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่เช่นกัน โดยใช้ระยะเวลาในการฝึกสอน 6 ชั่วโมง 53 นาที 2 วินาที กับรอบการฝึก 100 รอบ และมีค่าอัตราการเรียนรู้เป็น 0.0001 และ Iteration เท่ากับ 4000 รอบ ส่วนผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยรูปภาพทดสอบแสดงดังรูปที่ 4.5 พบว่า ความสามารถของโมเดลที่ผ่านการฝึกอบรม โดยพิจารณาจากทั้งค่าความแม่นยำและ วัดความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมดมีค่าเฉลี่ยอยู่ที่ 0.8072



รูปที่ 4.4 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล



รูปที่ 4.5 ผลประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล

ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล ด้วยไฟล์ภาพที่ videoplayback1_0611.png ดังขั้นตอนในหัวข้อ 3.7 แสดงดังรูปที่ 4.6 พบว่า ตำแหน่งที่รถอยู่ริมขอบซ้าย 1 จุดไม่สามารถตรวจจับได้ และโมเดลสามารถตรวจจับรถยนต์ได้ทั้งหมด 7 คันจากทั้งหมด 8 คันในภาพ

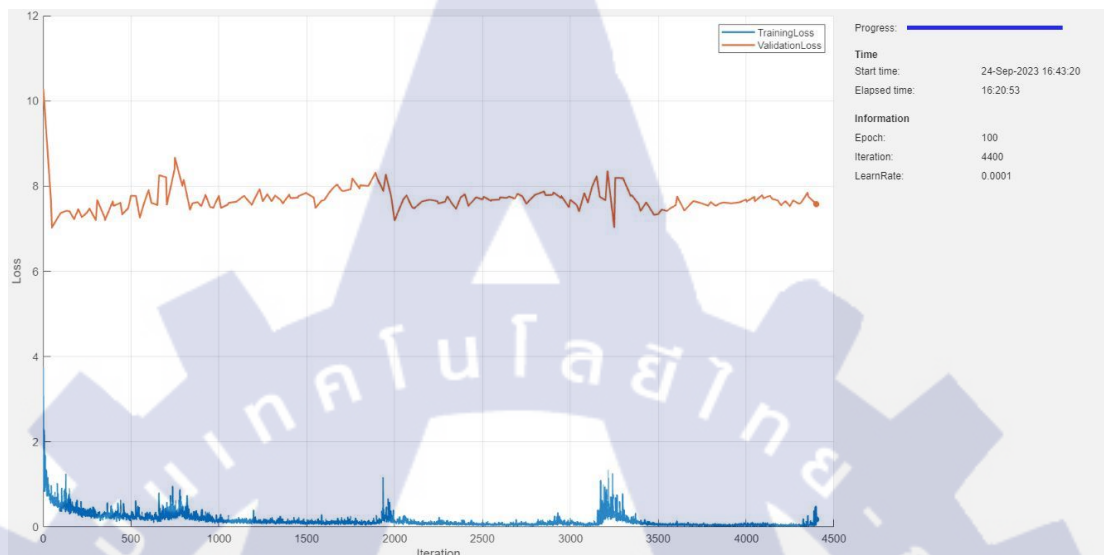


รูปที่ 4.6 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 416x416 พิกเซล

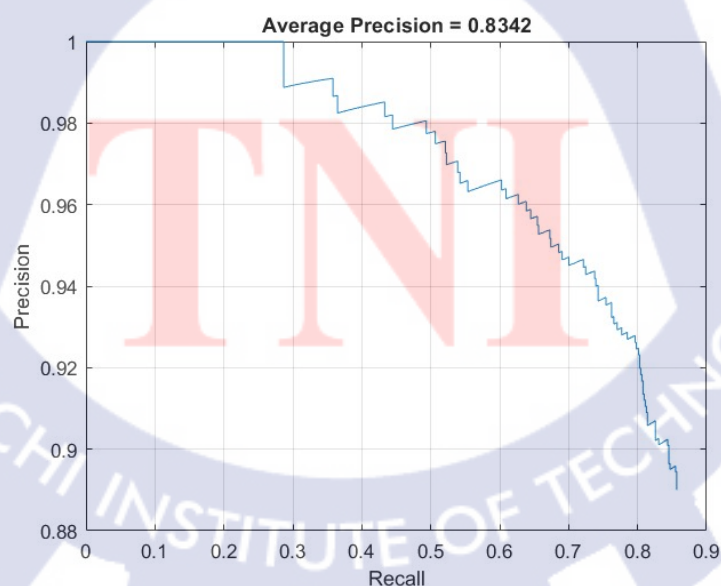
4.1.3) ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล

ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล แสดงดังรูปที่ 4.7 พบว่าเส้นกราฟของ Training Loss ลดลงเรื่อยๆ เมื่อเวลาผ่านไป แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่ ส่วน Validation Loss ของประสิทธิภาพของโมเดลบนข้อมูลชุด Validation แสดงว่าโมเดลกำลัง Overfitting หมายความว่าโมเดลเรียนรู้ข้อมูลชุด Training มากเกินไปจนไม่สามารถนำไปใช้กับข้อมูลชุดใหม่ได้ ส่งผลให้ความสามารถของโมเดลบนข้อมูลชุด Validation ต่ำ โดยใช้ระยะเวลาในการฝึกสอน 16 ชั่วโมง 20 นาที 53 วินาที กับรอบการฝึก 100 รอบ และมีค่าอัตราการเรียนรู้เป็น 0.0001 และ Iteration เท่ากับ 4000 รอบ ส่วนของผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-

Tiny ด้วยรูปภาพทดสอบทำให้เห็นว่าแสดงดังรูปที่ 4.8 พบว่า ความสามารถของโมเดลที่ผ่านการฝึกรอบโดยพิจารณาจากทั้งค่าความแม่นยำและวัดความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมดมีค่าเฉลี่ยอยู่ที่ 0.8342



รูปที่ 4.7 ผลการฝึกรอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล



รูปที่ 4.8 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล

ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล ด้วยไฟล์ภาพที่ videoplayback1_0611.png ดังขั้นตอนในหัวข้อ 3.7 แสดงดังรูปที่ 4.9 พบว่า ตำแหน่งที่รถอยู่ริมขอบซ้าย 1 จุดไม่สามารถตรวจจับได้ และมี 3 จุดที่ Scores การตรวจจับรถยนต์มีค่าเป็น 1 ซึ่งหมายความว่าค่าความแม่นยำจะมีค่าเท่ากับความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมดซึ่งในภาพนี้โมเดลสามารถตรวจจับรถยนต์ได้ทั้งหมด 7 คัน

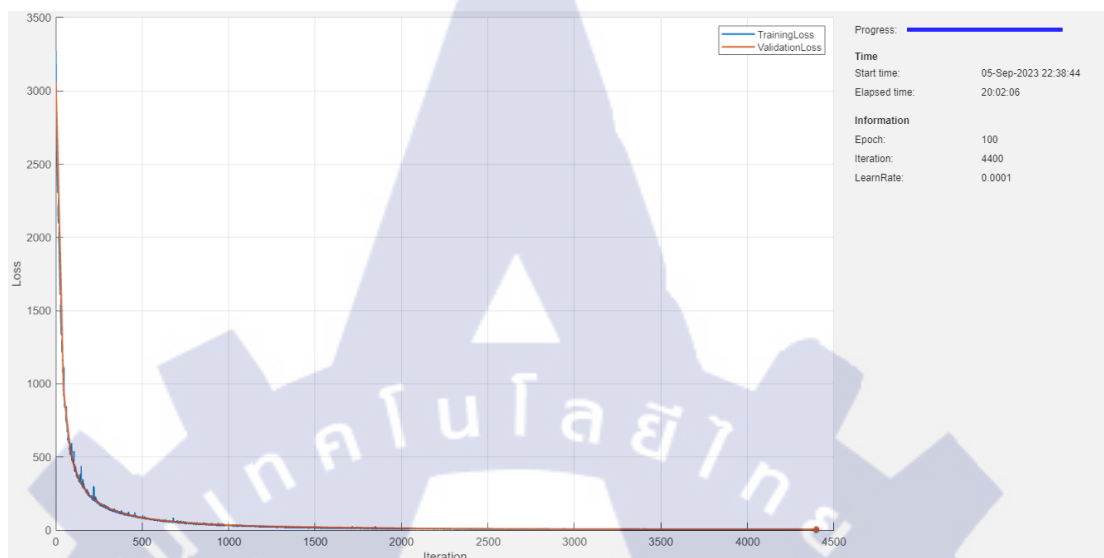


รูปที่ 4.9 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 288x288 พิกเซล

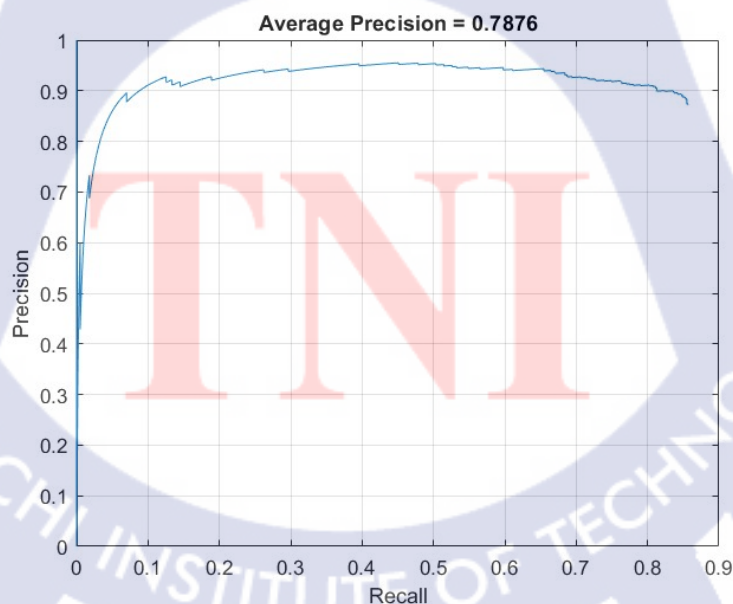
4.1.4) ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 603x603 พิกเซล

ผลการฝึกสอนโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบขนาด 603x603 พิกเซล แสดงดังรูปที่ 4.10 พบว่า กราฟเส้นโค้ง Training Loss จะลดลงเมื่อเวลาเพิ่มขึ้น แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่ ส่วน Validation Loss จะลดลงเรื่อยๆ เช่นกัน แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่เช่นกัน โดยใช้ระยะเวลาในการฝึกสอน 20 ชั่วโมง 02 นาที 06 วินาที กับรอบการฝึก 100 รอบ และ มีค่าอัตราการเรียนรู้เป็น 0.0001 และ Iteration เท่ากับ 4000 รอบ ส่วนของผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยรูปภาพทดสอบแสดงดังรูปที่ 4.11 พบว่า ความสามารถของโมเดล

ที่ผ่านการฝึกอบรม โดยพิจารณาจากทั้งค่าความแม่นยำและ วัดความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมดมีค่าเฉลี่ยอยู่ที่ 0.7876



รูปที่ 4.10 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 603x603 พิกเซล



รูปที่ 4.11 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 603x603 พิกเซล

ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 603x603 พิกเซล ด้วยไฟล์ภาพที่ videoplayback1_0611.png ดังขั้นตอนในหัวข้อ 3.7 แสดงดังรูปที่ 4.12 พบว่า ตำแหน่งที่รถอยู่ริมขอบซ้าย 1 จุดไม่สามารถตรวจจับได้ และโมเดลสามารถตรวจจับรถยนต์ได้ทั้งหมด 7 คัน



รูปที่ 4.12 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบแบบปกติขนาด 603x603 พิกเซล

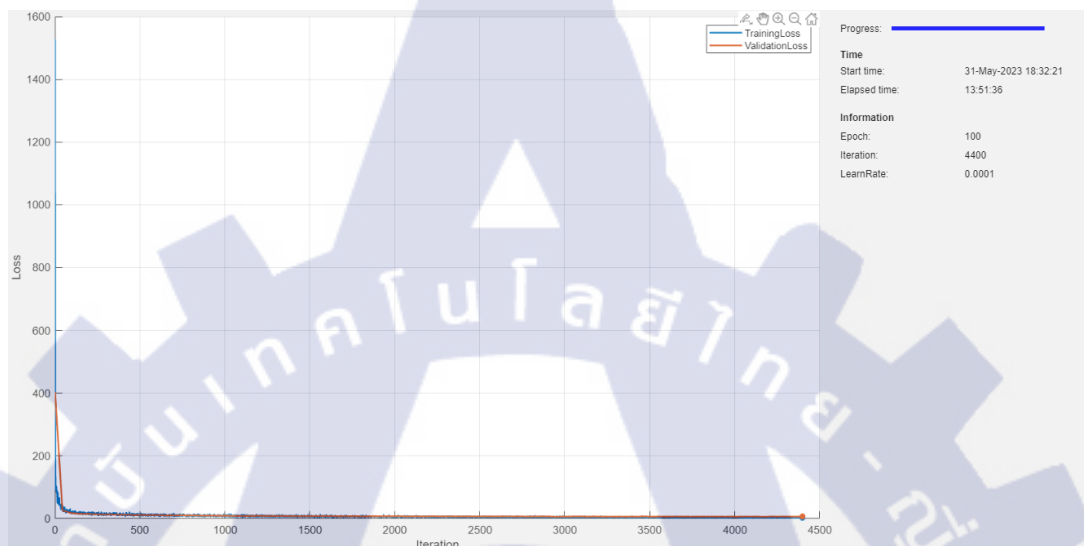
4.2 ผลจากการฝึกรวมโมเดล YOLOv4-Tiny ด้วยชุดข้อมูลภาพผ่านการเสริมข้อมูล

ผลจากการฝึกรวมโมเดล YOLOv4-Tiny ด้วยชุดข้อมูลภาพที่ผ่านการทำเทคนิคการเสริมข้อมูลทั้ง 3 แบบซึ่งประกอบด้วย การปรับเปลี่ยนค่าสีของภาพในปริภูมิ HSV การสะท้อนภาพ แนวนอนแบบสุ่ม และการย่อหรือขยายภาพ 10% แบบสุ่มจะแบ่งการวิเคราะห์ผลการฝึกรวมโดยแยกตามการกำหนดค่าอินพุตของโมเดล

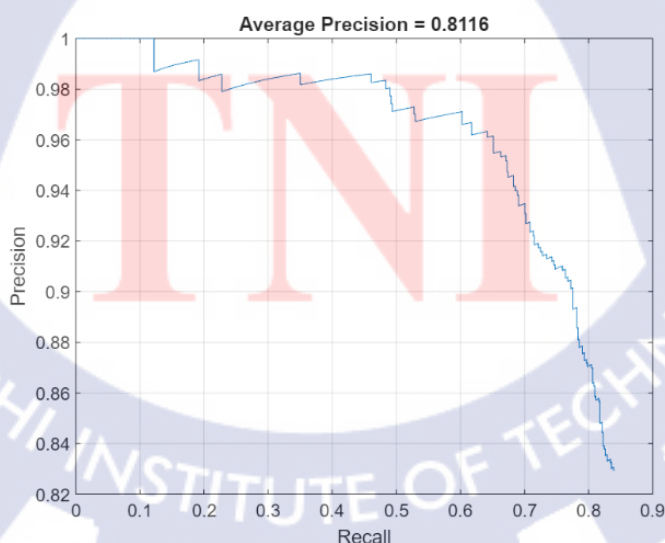
4.2.1) ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบที่ผ่านการเสริมขนาดข้อมูลด้วยภาพขนาด 416x416 พิกเซล

ผลการฝึกสอนโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบขนาด 416x416 พิกเซล ที่ผ่านการเสริมข้อมูลทั้ง 3 แบบแสดงดังรูปที่ 4.13 พบว่ากราฟเส้นโค้งของ Training Loss ลดลงเมื่อเวลาเพิ่มขึ้น แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่ ส่วน Validation Loss ก็ลดลงเรื่อยๆ เช่นกัน แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่เช่นกัน โดยใช้ระยะเวลาในการฝึกสอน 13 ชั่วโมง 51 นาที 36 วินาที

กับรอบการฝึก 100 รอบ และมีค่าอัตราการเรียนรู้เป็น 0.0001 และ Iteration เท่ากับ 4000 รอบ ส่วนของผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยรูปภาพทดสอบแสดงดังรูป 4.14 พบว่าความสามารถของโมเดลที่ผ่านการฝึกอบรม โดยพิจารณาทั้งค่าความแม่นยำ และ วัดความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมด มีค่าเฉลี่ยอยู่ที่ 0.8116



รูปที่ 4.13 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบที่ผ่านการเสริมขนาดข้อมูลด้วยภาพขนาด 416x416 พิกเซล



รูปที่ 4.14 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบที่ผ่านการเสริมขนาดข้อมูลด้วยภาพขนาด 416x416 พิกเซล

ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบที่ทำเทคนิคการเสริมข้อมูลทั้ง 3 แบบที่มีขนาด 416x416 พิกเซล ด้วยไฟล์ภาพที่ videoplayback1_0611.png ดั้งชั้นตอนในหัวข้อ 3.7 แสดงดังรูปที่ 4.15 พบว่า ตำแหน่งรถอยู่ริมขอบซ้าย 1 จุดที่ไม่สามารถตรวจจับได้ และมี 1 จุดที่ Scores อยู่ต่ำกว่าค่าเฉลี่ย เป็น 0.6787 ภาพรวมโมเดลสามารถตรวจจับรถยนต์ได้ทั้งหมด 6 คัน

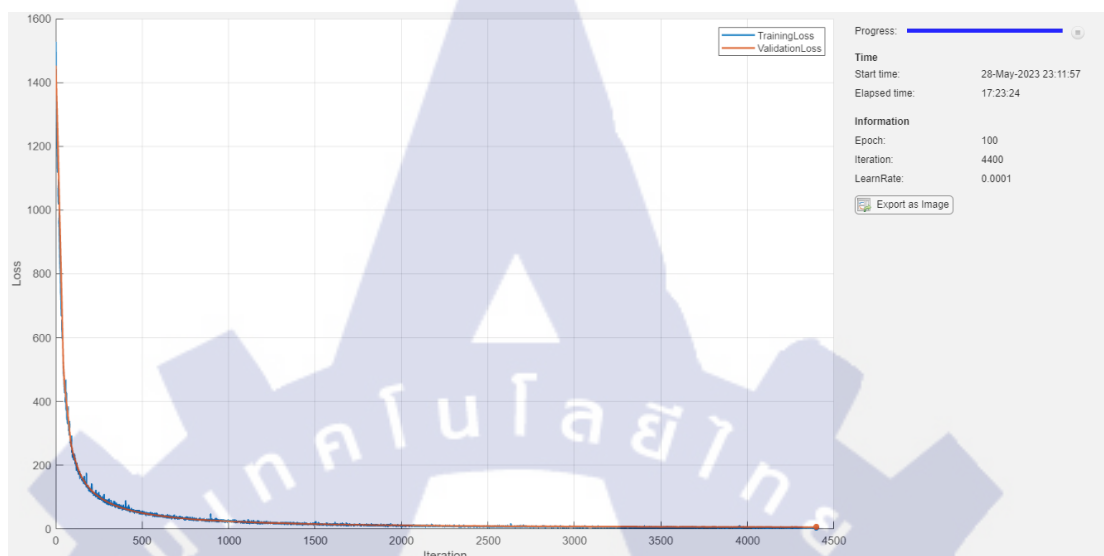


รูปที่ 4.15 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ SGDM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลด้วยภาพขนาด 416x416 พิกเซล

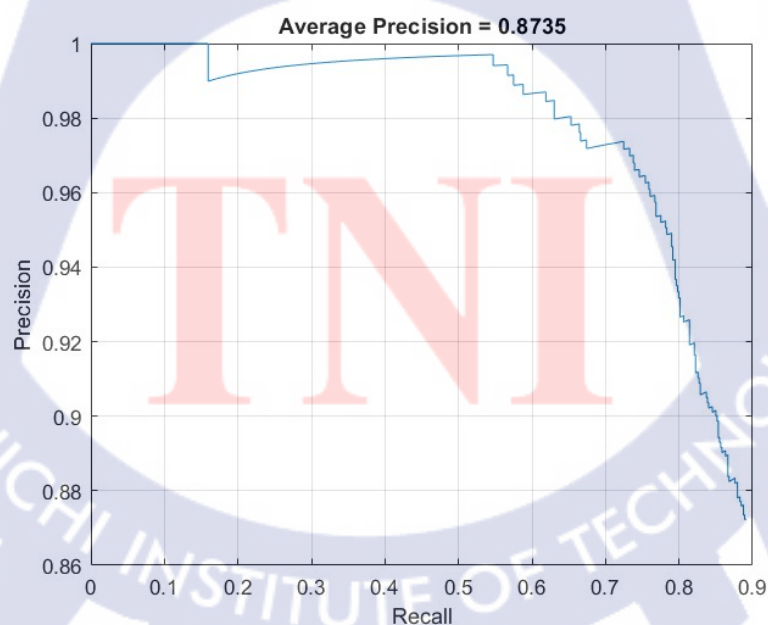
4.2.2) ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 416x416 พิกเซล

ผลการฝึกสอนโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบขนาด 416x416 พิกเซล แสดงดังรูปที่ 4.16 เส้นกราฟของ Training Loss จะลดลงเมื่อเวลาเพิ่มขึ้น แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่ ส่วน Validation Loss ก็ลดลงเรื่อยๆ เช่นกัน แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่เช่นกัน โดยใช้ระยะเวลาในการฝึกสอน 17 ชั่วโมง 23 นาที 24 วินาที กับรอบการฝึก 100 รอบ และมีค่าอัตราการเรียนรู้เป็น 0.0001 และ Iteration เท่ากับ 4000 ส่วนผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยรูปภาพทดสอบแสดงดังรูปที่ 4.17 พบว่า ความสามารถของโมเดลที่ผ่านการฝึกอบรม โดย

พิจารณาจากทั้งค่าความแม่นยำและ วัดความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมดมีค่าเฉลี่ยอยู่ที่ 0.8735



รูปที่ 4.16 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 416x416 พิกเซล



รูปที่ 4.17 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 416x416 พิกเซล

ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 416x416 พิกเซล ด้วยไฟล์ภาพที่ videoplayback1_0611.png ดั้งชั้นตอนในหัวข้อ 3.7 แสดงดังรูปที่ 4.18 พบว่า ตำแหน่งที่รถอยู่ริมขอบซ้าย 1 จุดไม่สามารถตรวจจับได้ และมี 1 จุดที่ Scores อยู่ต่ำกว่าค่าเฉลี่ยเป็น 0.601 โมเดลสามารถตรวจจับรถยนต์ได้ทั้งหมด 7 คัน

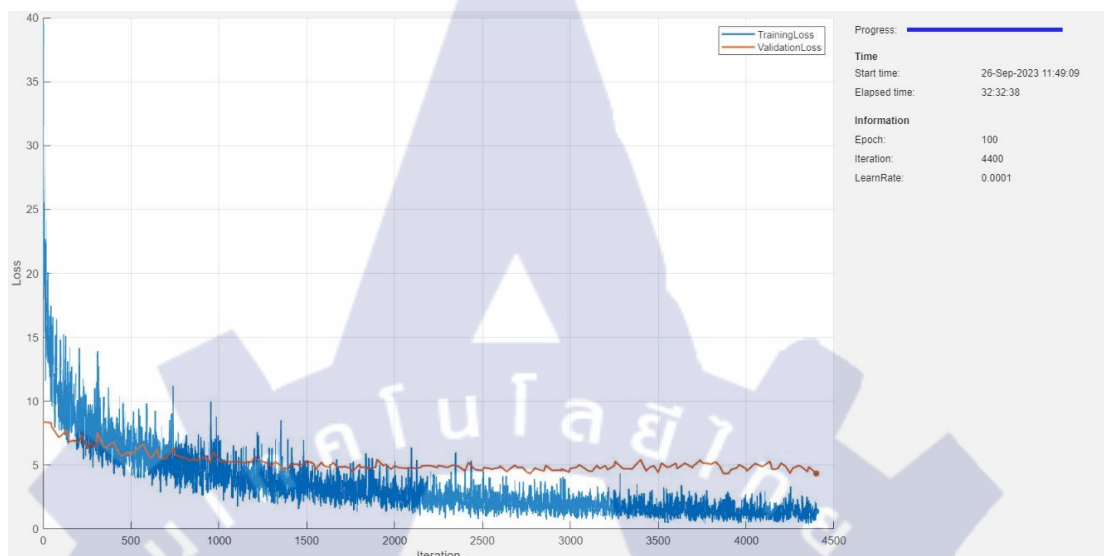


รูปที่ 4.18 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 416x416 พิกเซล

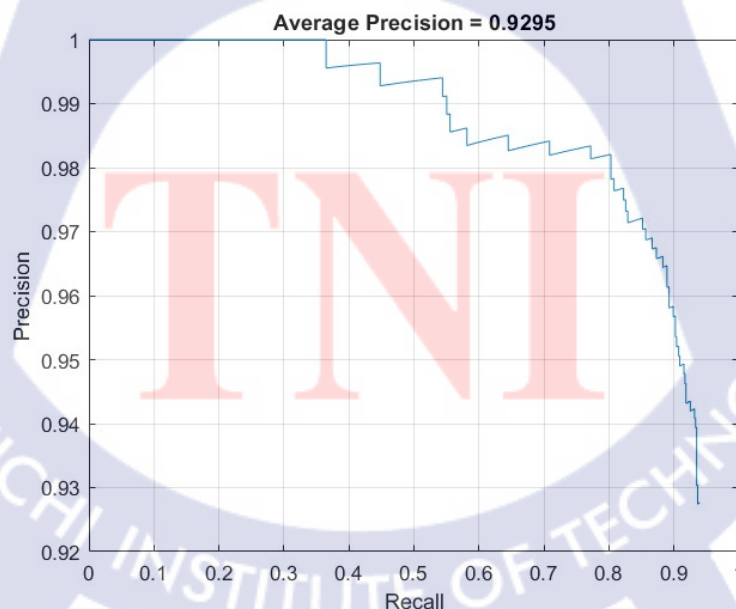
4.2.3) ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 288x288 พิกเซล

ผลการฝึกสอนโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบขนาด 288x288 พิกเซล ที่ผ่านการเสริมข้อมูลทั้ง 3 แบบแสดงดังรูปที่ 4.19 เส้นกราฟของ Training Loss จะลดลงเมื่อเวลาเพิ่มขึ้น แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่ แต่อาจไม่มีประสิทธิภาพเท่า Training Loss มากนัก เนื่องจาก Validation Loss เป็นการวัดประสิทธิภาพของโมเดลบนข้อมูลชุดใหม่ โดยใช้ระยะเวลาในการฝึกสอน 32 ชั่วโมง 32 นาที 38 วินาที กับรอบการฝึก 100 รอบและมีค่าอัตราการเรียนรู้เป็น 0.0001 และ Iteration เท่ากับ 4000 ส่วนผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยรูปภาพทดสอบแสดงดังรูปที่ 4.20

พบว่า ความสามารถของโมเดลที่ผ่านการฝึกอบรม โดยพิจารณาจากทั้งค่าความแม่นยำและ วัดความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมดมีค่าเฉลี่ยอยู่ที่ 0.9295



รูปที่ 4.19 ผลการฝึกสอนโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 288x288 พิกเซล



รูปที่ 4.20 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 288x288 พิกเซล

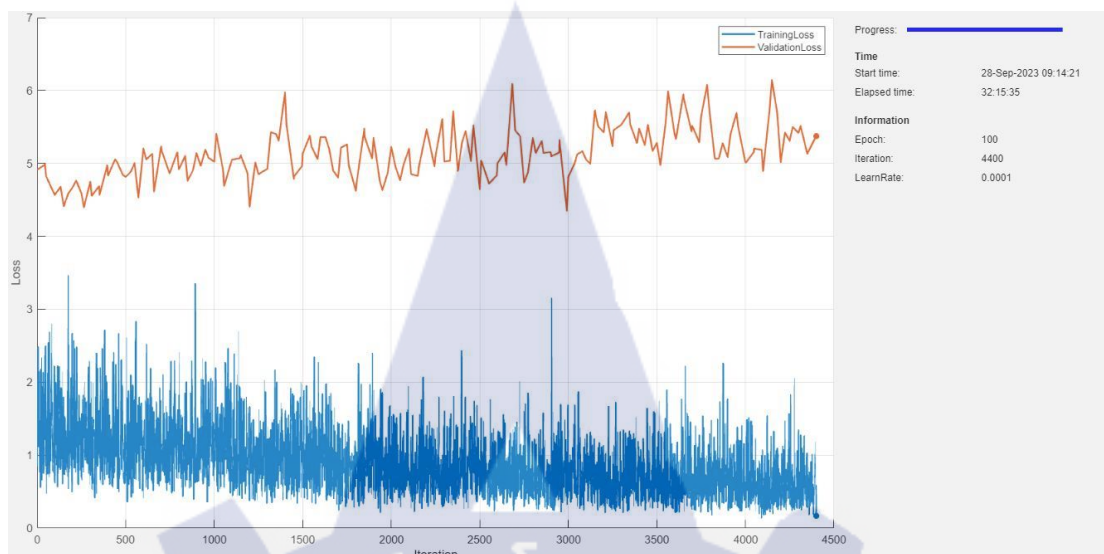
ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบขนาด 288x288 พิกเซล ที่ผ่านการเสริมข้อมูลทั้ง 3 แบบด้วยไฟล์ภาพที่ videoplayback1_0611.png ดังขั้นตอนในหัวข้อ 3.7 แสดงดังรูปที่ 4.21 พบว่า ตำแหน่งที่รถอยู่ใกล้ทางด่วน 1 จุดไม่สามารถตรวจจับได้ และสามารถตรวจจับรถยนต์ได้ทั้งหมด 7 คัน



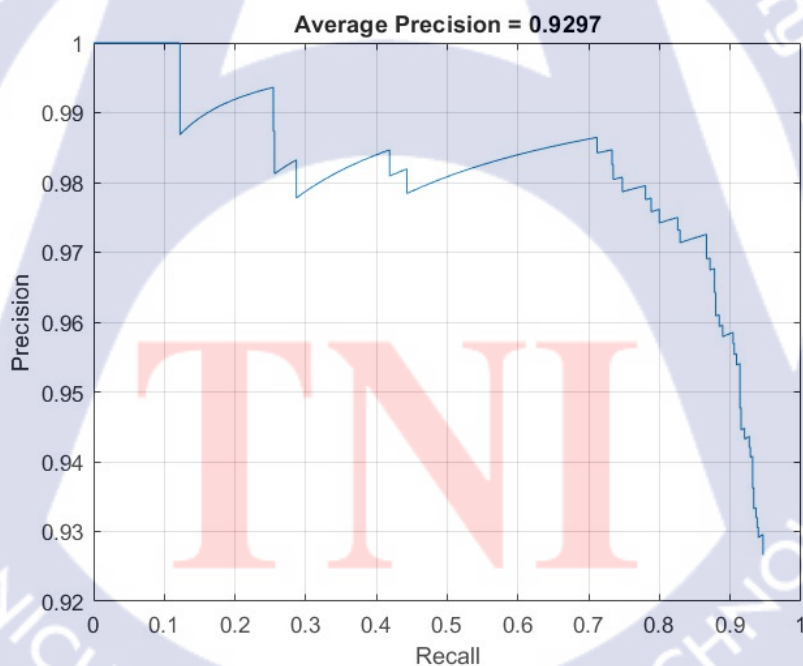
รูปที่ 4.21 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 288x288 พิกเซล

4.2.4) ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 603x603 พิกเซล

ผลการฝึกสอนโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบขนาด 603x603 พิกเซล ที่ผ่านการเสริมข้อมูลทั้ง 3 แบบแสดงดังรูปที่ 4.22 เส้นกราฟของ Training Loss จะลดลงเมื่อเวลาเพิ่มขึ้น แสดงว่าโมเดลกำลังเรียนรู้และปรับปรุงประสิทธิภาพอยู่ แต่อาจไม่มีประสิทธิภาพเท่า Training Loss มากนัก เนื่องจาก Validation Loss เป็นการวัดประสิทธิภาพของโมเดลบนข้อมูลชุดใหม่ โดยใช้ระยะเวลาในการฝึกสอน 32 ชั่วโมง 15 นาที 35 วินาที กับรอบการฝึก 100 รอบและมีค่าอัตราการเรียนรู้เป็น 0.0001 และ Iteration เท่ากับ 4000 ส่วนผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยรูปภาพทดสอบแสดงดังรูปที่ 4.23 พบว่า ความสามารถของโมเดลที่ผ่านการฝึกอบรม โดยพิจารณาจากทั้งค่าความแม่นยำและ วัดความสามารถของโมเดลในการค้นหาวัตถุที่เกี่ยวข้องทั้งหมดมีค่าเฉลี่ยอยู่ที่ 0.9297



รูปที่ 4.22 ผลการฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 603x603 พิกเซล



รูปที่ 4.23 ผลการประเมินโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออปติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 603x603 พิกเซล

ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบขนาด 603x603 พิกเซล ที่ผ่านการเสริมข้อมูลทั้ง 3 แบบด้วยไฟล์ภาพที่ videoplayback1_0611.png ดังขั้นตอนในหัวข้อ 3.7 แสดงดังรูปที่ 4.24 พบว่า โมเดลที่เลือกออฟติไมเซอร์แบบ ADAM ร่วมกับการเสริมขนาดข้อมูลทั้ง 3 แบบมีความแม่นยำในการตรวจจับรถยนต์ในภาพมากกว่าโมเดลทั้งหมดที่ถูกฝึกอบรม และโมเดลสามารถตรวจจับรถยนต์ได้ทั้งหมด 8 คัน และค่าคะแนนความแม่นยำของทุกคนมีค่ามากกว่า 0.9



รูปที่ 4.24 ผลการทดสอบโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM โดยใช้ภาพทดสอบที่ผ่านการเสริมข้อมูลขนาด 603x603 พิกเซล

4.3 ผลการประเมินความแม่นยำของโมเดลที่ฝึกอบรมทั้งหมด

4.3.1 ผลการเปรียบเทียบการปรับปรุงคุณภาพของภาพปกติและภาพที่ผ่านการเสริมขนาดข้อมูล

การเปรียบเทียบนี้จะทำการฝึกอบรมโมเดล YOLOV4-Tiny ที่กำหนดขนาดภาพเป็น 416x416 พิกเซล จากวัตถุประสงค์ของการศึกษาเปรียบเทียบการปรับปรุงคุณภาพของโมเดลด้วยเทคนิคการเสริมข้อมูลชุดฝึกอบรมและ เลือกใช้ออฟติไมเซอร์สำหรับปรับปรุงคุณภาพของโมเดลที่เหมาะสมกับโมเดลการตรวจจับรถยนต์ภาพความละเอียดต่ำ ผลการเปรียบเทียบแสดงดังตารางที่ 4.1 โดยคำนวณ การเปรียบเทียบเวลาที่ใช้ในการฝึกอบรมระหว่างสองวิธีนี้ โดยใช้สูตรการคำนวณเปอร์เซ็นต์การเปลี่ยนแปลงซึ่งมีดังสมการที่ 4.1

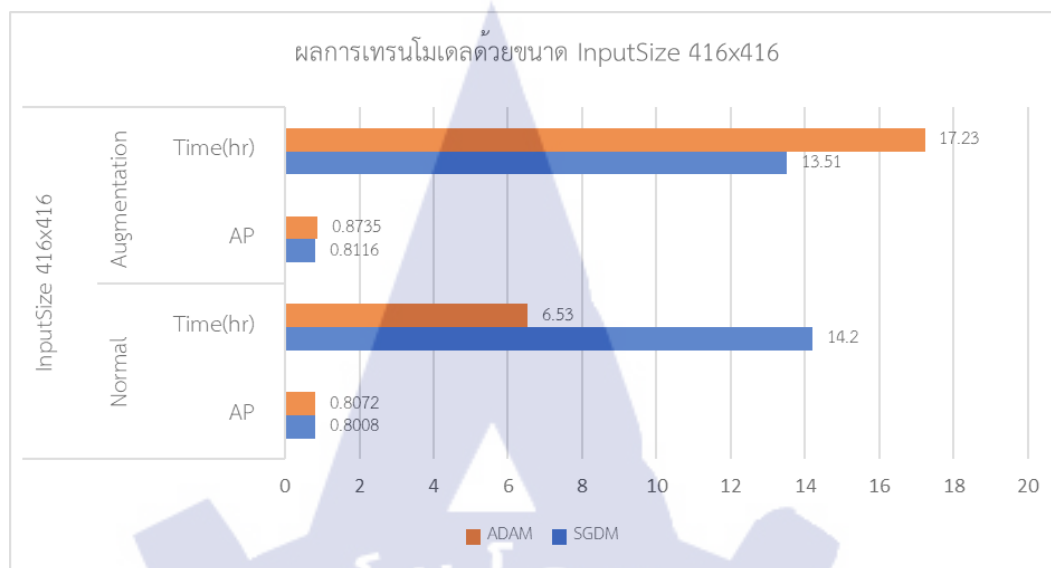
$$\text{เปอร์เซ็นต์การเปลี่ยนแปลง} = \left(\frac{\text{ค่าเริ่มต้น} - \text{ค่าใหม่}}{\text{ค่าเริ่มต้น}} \right) \times 100 \quad (4.1)$$

โดยที่ ค่าเริ่มต้น คือ เวลาที่ใช้ในการฝึกอบรมโดยใช้เทคนิคการเสริมข้อมูล และ
ค่าใหม่ คือ เวลาที่ใช้ในการฝึกอบรมโดยไม่ผ่านเทคนิคการเสริมข้อมูล

ตารางที่ 4.1 ผลการฝึกอบรมโมเดลด้วยขนาด InputSize 416x416 พิกเซล

Optimizer Model	InputSize 416x416			
	Normal		Augmentation	
	AP	Time(hr)	AP	Time(hr)
SGDM	0.8008	14.2	0.8116	13.51
ADAM	0.8072	6.53	0.8735	17.23

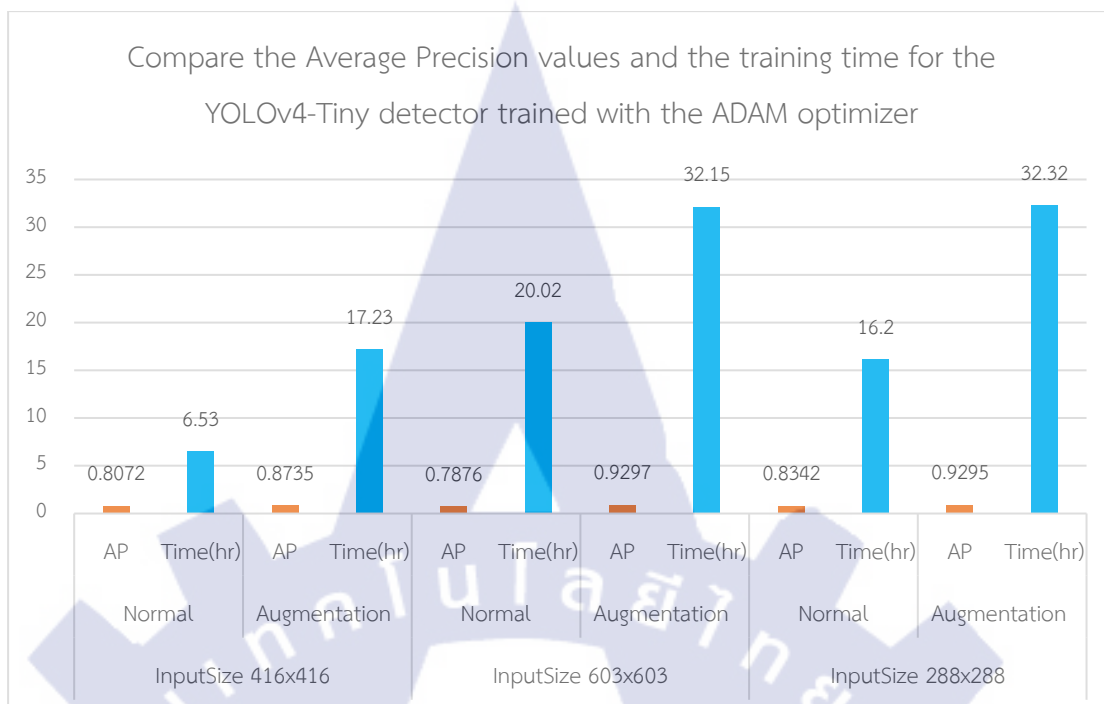
ผลการประเมินการโมเดลตัวตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์แบบ ADAM สำหรับโมเดลที่ผ่านเทคนิคการเสริมขนาดข้อมูลทั้ง 3 แบบแสดงดังรูปที่ 4.25 ให้ค่าความแม่นยำเฉลี่ยสูงที่สุดเป็น 0.8735 แต่ใช้ระยะเวลาในการฝึกอบรมนานที่สุดด้วยเวลา 17.23 ชั่วโมง แต่หากโมเดลที่ถูกฝึกอบรมด้วยชุดข้อมูลปกติ ที่ไม่ผ่านการทำเทคนิคการเสริมข้อมูลทั้ง 3 แบบ ใช้เวลาในการฝึกอบรม 6.53 ชั่วโมง เมื่อเปรียบเทียบค่าเวลาที่แตกต่างกัน จากการคำนวณหาเปอร์เซ็นต์การเปลี่ยนแปลงดังสมการที่ 4.1 ด้วยการนำค่าของเวลาการฝึกสอนเมื่อใช้เทคนิคเสริมภาพเทียบกับการไม่ใช้เทคนิคเสริมภาพในการฝึกสอนด้วย ADAM ระยะเวลาในการฝึกอบรมของโมเดลที่ถูกฝึกอบรมด้วยชุดข้อมูลปกติลดลง 62.10% เมื่อไม่ใช้เทคนิคการเสริมข้อมูล และเมื่อเทียบกับโมเดลที่ผ่านเทคนิคการเสริมข้อมูลทั้ง 3 แบบ พบว่า โมเดลที่ผ่านการเสริมข้อมูลมีความสามารถสูงกว่าโมเดลที่ไม่ได้ผ่านการเสริมขนาดของข้อมูลในด้านความแม่นยำในการจำแนกประเภทวัตถุ แต่ระยะเวลาในการฝึกอบรมจะนานกว่ามาก



รูปที่ 4.25 กราฟเปรียบเทียบผลการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยออฟติไมเซอร์ ADAM กับ SGDM กับข้อมูลปกติ และ ข้อมูลที่ผ่านการเสริมขนาดข้อมูล

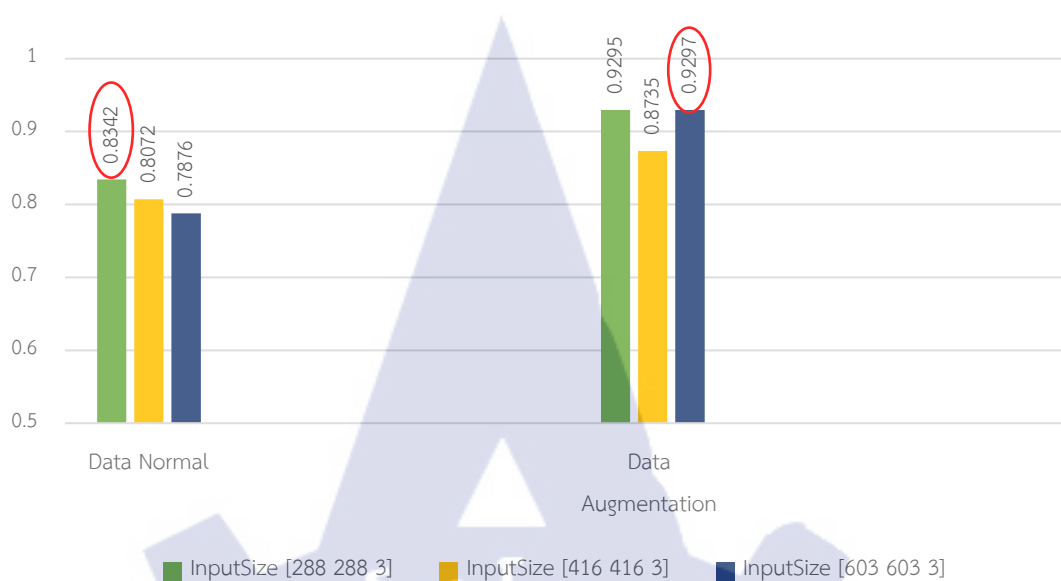
4.3.2 ผลการเปรียบเทียบ ขนาดอินพุตภาพของการฝึกอบรมโมเดล YOLOV4-Tiny

การฝึกอบรมเพื่อให้ได้โมเดลที่มีความสามารถสูงสุด จากการเลือกใช้การปรับปรุงคุณภาพของโมเดลด้วยออฟติไมเซอร์ ADAM ที่เหมาะสมกับการตรวจจับรถยนต์ภาพความละเอียดต่ำ ภาพต้นแบบของการฝึกสอนมีขนาด 854x480 พิกเซล ที่ผ่านการทำเทคนิคการเสริมขนาดข้อมูลซึ่งไม่เหมาะสมกับการนำไปใช้งานเป็นอินพุตของโมเดล YOLOv4-Tiny การลดขนาดอินพุตภาพลงเป็น 288x288 พิกเซล เป็นการทำให้ขนาดอินพุตของโมเดลที่เล็กลง ส่งผลให้โมเดลมีความเร็วในการประมวลผลที่เร็วขึ้น แต่อาจส่งผลให้ความสามารถในการตรวจจับวัตถุขนาดเล็กลดลง และการเพิ่มขนาดภาพอินพุตเป็น 603x603 พิกเซล ซึ่งเป็นขนาดอินพุตของโมเดลที่ใหญ่ขึ้น และจะส่งผลให้โมเดลสามารถตรวจจับวัตถุขนาดเล็กได้ดีขึ้น แต่ใช้ระยะเวลาในการฝึกสอนนานถึง 1 วัน 8 ชั่วโมง ดังรูปที่ 4.26 โดยวัตถุที่มีขนาดเล็กเมื่อเทียบกับขนาดของภาพ เช่น รถยนต์ขนาดเล็กที่มีภาพความละเอียดต่ำ เมื่อโมเดลมีข้อมูลภาพมากขึ้นในการมองเห็นและตรวจจับวัตถุ จึงทำให้มีค่าความแม่นยำในการตรวจจับสูงที่สุดมีค่าเท่ากับ 0.9297



รูปที่ 4.26 ผลการเปรียบเทียบระยะเวลาที่ใช้ฝึกอบรมโมเดลตรวจจับวัตถุ YOLOv4-Tiny ด้วยออฟติไมเซอร์ ADAM

ผลการเปรียบเทียบค่าความแม่นยำเฉลี่ยของภาพอินพุตขนาดต่าง ๆ ที่เป็นภาพปกติ และภาพที่ผ่านการทำเทคนิคการเสริมขนาดข้อมูล เมื่อนำมาฝึกอบรมด้วยออฟติไมเซอร์ ADAM แสดงดังรูปที่ 4.27 โดยกราฟแท่งด้านซ้ายเป็นผลการวิเคราะห์ของภาพปกติ และด้านขวาเป็นผลการวิเคราะห์ของภาพที่ผ่านการเสริมขนาดข้อมูล จะเห็นว่าภาพที่ผ่านการเสริมขนาดข้อมูลจะมีความสามารถในการตรวจจับวัตถุขนาดเล็กได้ดีที่สุด โดยมีค่าความแม่นยำเฉลี่ยสูงสุดที่ 0.9297 สำหรับขนาดภาพที่ 603x603 พิกเซล และค่าความแม่นยำเฉลี่ยที่ 0.9295 และ 0.8735 สำหรับภาพอินพุตที่ 288x288 และ 416x416 พิกเซล ตามลำดับ ส่วนชุดข้อมูลปกติที่ไม่ผ่านการเสริมขนาดข้อมูล เมื่อนำมาฝึกอบรมด้วยออฟติไมเซอร์แบบ ADAM พบว่าขนาดอินพุตของโมเดลที่ 288x288 พิกเซล จะมีความสามารถในการตรวจจับวัตถุขนาดเล็กได้ดีที่สุด โดยมีค่าความแม่นยำเฉลี่ยที่ 0.8342 และค่าความแม่นยำเฉลี่ยสำหรับขนาดภาพอินพุตที่ 416x416 และ 603x603 พิกเซล มีค่าที่ 0.8072 และ 0.7876 ตามลำดับ



รูปที่ 4.27 ผลการเปรียบเทียบค่าความแม่นยำเฉลี่ยของการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยออปติไมเซอร์ ADAM กับข้อมูลปกติ และ ข้อมูลที่ทำการเสริมขนาดข้อมูล

การคำนวณเพื่อหาค่าเฉลี่ยของเปอร์เซ็นต์การเพิ่มขึ้นในความแม่นยำเฉลี่ย สำหรับโมเดลที่ฝึกอบรมด้วยเทคนิคการเสริมข้อมูลเทียบกับโมเดลที่ไม่ใช้เทคนิคการเสริมข้อมูล สามารถคำนวณได้ด้วยสูตรการหาค่าเฉลี่ยของความแตกต่างเป็นเปอร์เซ็นต์ ดังนี้

- คำนวณความแตกต่างในค่าความแม่นยำเฉลี่ย สำหรับแต่ละขนาดภาพอินพุต

$$\text{ค่าความแตกต่าง} = \text{ค่าความแม่นยำเฉลี่ยของโมเดลเทคนิคการเสริมข้อมูล} - \text{ค่าความแม่นยำเฉลี่ยของโมเดลไม่ใช้เทคนิคการเสริมข้อมูล} \quad (4.2)$$

- หาค่าเฉลี่ยของความแตกต่าง

$$\text{ค่าเฉลี่ยที่เพิ่มขึ้น} = \frac{\text{ค่าความแตกต่างอินพุต603} + \text{ค่าความแตกต่างอินพุต288} + \text{ค่าความแตกต่างอินพุต416}}{3} \quad (4.3)$$

- แปลงค่าเฉลี่ยของความแตกต่างที่ได้เป็นเปอร์เซ็นต์เทียบกับค่าเฉลี่ยของ AP ของโมเดลที่ไม่ใช้เทคนิคการเสริมข้อมูล

$$\% \text{ค่าเฉลี่ยที่เพิ่มขึ้น} = \left(\frac{\text{ค่าเฉลี่ยที่เพิ่มขึ้น}}{\text{ค่าเฉลี่ยของค่าความแม่นยำเฉลี่ยของโมเดลที่ไม่ใช้เทคนิคการเสริมข้อมูล}} \right) \times 100 \quad (4.4)$$

โดยที่

$$\begin{aligned} & \text{ค่าเฉลี่ยของ ค่าความแม่นยำเฉลี่ยของโมเดลที่ไม่ใช้เทคนิคการเสริมข้อมูล} = \\ & \frac{AP_{\text{ข้อมูลปกติ ขนาดรูปภาพ 603}} + AP_{\text{ข้อมูลปกติ ขนาดรูปภาพ 288}} + AP_{\text{ข้อมูลปกติ ขนาดรูปภาพ 416}}}{3} \end{aligned} \quad (4.5)$$

การคำนวณเพื่อหาค่าเฉลี่ยของเปอร์เซ็นต์การเพิ่มขึ้นในความแม่นยำเฉลี่ย สำหรับโมเดลที่ฝึกอบรมด้วยเทคนิคการเสริมข้อมูลเทียบกับโมเดลที่ไม่ใช้เทคนิคการเสริมข้อมูลได้ผลเป็น 12.50%

ปัญหาและข้อจำกัดในการฝึกอบรมโมเดลตรวจจับรถยนต์จากรูปภาพความละเอียดต่ำจากโดรนพบว่า การฝึกอบรมตัวตรวจจับวัตถุจำเป็นต้องมีการฝึกอบรมด้วยข้อมูลขนาดใหญ่ ซึ่งเป็นปัจจัยหลักและข้อจำกัดในด้านทรัพยากรฮาร์ดแวร์ที่ใช้สำหรับฝึกอบรม ในเรื่องของการใช้งานคอมพิวเตอร์ที่มีความสามารถไม่เพียงพอต่อชุดข้อมูลขนาดใหญ่ จึงทำให้มีการตัดแบ่งส่วนของข้อมูลในการฝึกอบรมออกไปบางส่วน โดยแบ่งหัวข้อปัญหาและข้อจำกัดในแต่ละด้าน ดังนี้

- 1) ข้อมูลการฝึกอบรมจะต้องมีข้อมูลที่เหมาะสมที่ถูกตรวจจับและเตรียมข้อมูลอย่างถูกต้องสำหรับการเรียนรู้ของโมเดล YOLOv4-Tiny ที่ต้องใช้ระยะเวลาในการเตรียมข้อมูล
- 2) ระยะเวลาที่ต้องใช้สำหรับการฝึกอบรมจะนานขึ้น ซึ่งขึ้นอยู่กับความซับซ้อนในขั้นตอนการสอนด้วยข้อมูลการทดสอบ การเรียนรู้เชิงลึกและการตรวจจับวัตถุ โดยจะต้องใช้เวลาในการสอนและปรับแต่งโมเดล เพื่อให้มีความแม่นยำที่สูง
- 3) ข้อจำกัดของโดรน ระบบการถ่ายภาพที่ใช้ในการเก็บข้อมูลอาจมีข้อจำกัดในการเคลื่อนที่และเวลาถ่ายภาพ ซึ่งทำให้เป็นที่ยากที่จะได้รับข้อมูลภาพที่เหมาะสมสำหรับการเรียนรู้
- 4) ความสามารถของโมเดล YOLOv4-Tiny ที่ขนาดเล็กและรวดเร็ว แต่อาจมีความจำกัดในเรื่องความแม่นยำ โดยเฉพาะเมื่อต้องตรวจจับวัตถุขนาดเล็กในสภาพแวดล้อมที่ซับซ้อน

บทที่ 5

สรุปผลและข้อเสนอแนะ

การฝึกอบรมโมเดลตรวจจับวัตถุขนาดเล็กโดยใช้การเรียนรู้เชิงลึก YOLOv4-Tiny ด้วยรูปภาพความละเอียดต่ำจากโดรน โดยแบ่งปัจจัยในการฝึกอบรมออกเป็นผลการปรับค่าพารามิเตอร์ต่าง ๆ ในเรื่องของขนาดภาพอินพุตที่เหมาะสมกับชุดข้อมูลภาพและตัววัตถุที่จะทำการตรวจจับ การเลือกใช้ออปติไมเซอร์ที่เหมาะสมกับการตรวจจับวัตถุภาพรยนต์ความละเอียดต่ำขนาดเล็ก และเทคนิคการเตรียมข้อมูลภาพการทำการเสริมขนาดข้อมูลทั้ง 3 แบบได้แก่ การปรับเปลี่ยนค่าสีของภาพในปริภูมิ HSV การสะท้อนภาพแนวนอนแบบสุ่ม และการย่อหรือขยายภาพ 10% แบบสุ่มโดยมีผลสรุปของงานวิจัยและข้อเสนอแนะดังนี้

5.1 สรุปผลการวิจัย

ออปติไมเซอร์ที่เหมาะสมที่ใช้ในการฝึกอบรมโมเดล YOLOv4-Tiny สำหรับงานตรวจจับรถยนต์ด้วยภาพความละเอียดต่ำขนาดเล็ก สามารถสรุปได้ดังนี้

- 1) การใช้ออปติไมเซอร์ ยังมีข้อจำกัดบางประการ เช่น SGDM อาจใช้ระยะเวลาในการลู่เข้า สำหรับกระบวนการฝึกอบรม
จุดที่มีค่าการสูญเสียไม่เปลี่ยนแปลงนานกว่า ADAM
- 2) การฝึกอบรมโดยใช้ออปติไมเซอร์แบบ ADAM กับโมเดลการตรวจจับวัตถุที่ใช้ในงานวิจัย YOLOv4-Tiny ได้ผลการทดสอบที่มีความแม่นยำเฉลี่ยสูงกว่าการใช้ออปติไมเซอร์แบบ SGDM ทุกกรณี
- 3) ออปติไมเซอร์แบบ ADAM จะมีความสามารถมากกว่า SGDM เนื่องจาก ADAM ใช้อัตราการเรียนรู้ที่สามารถปรับให้เหมาะสม (Adaptive Learning Rate) และโมเมนตัม ซึ่งช่วยให้โมเดลสามารถเรียนรู้ได้อย่างมีประสิทธิภาพมากขึ้น
- 4) การเลือกใช้ออปติไมเซอร์แบบ ADAM มีความเหมาะสมที่ใช้สำหรับการปรับปรุงโมเดลในงานทางด้านการตรวจจับรถยนต์ที่มีภาพความละเอียดต่ำขนาดเล็ก

การปรับขนาดภาพอินพุตที่ใช้ในการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยภาพ 3 ขนาดคือ 288x288, 416x416, และ 603x603 พิกเซลสรุปได้ดังนี้

- 1) ขนาดภาพอินพุตที่ใหญ่ขึ้นจะช่วยให้โมเดลสามารถมองเห็นรายละเอียดของวัตถุได้มากขึ้นและอาจทำให้โมเดลสามารถตรวจจับวัตถุได้แม่นยำยิ่งขึ้น

2) การเลือกขนาดอินพุตที่เหมาะสมเป็นปัจจัยที่สำคัญที่ส่งผลต่อการตรวจจับวัตถุด้วยโมเดล YOLOv4-Tiny เมื่อใช้ภาพความละเอียดต่ำจากโดรน การเลือกขนาดที่เหมาะสมจะส่งผลต่อความสามารถของการตรวจจับและความเร็วในการทำนายผลลัพธ์

3) จากการวิเคราะห์ค่าความแม่นยำเฉลี่ยของโมเดล YOLOv4-Tiny ที่ฝึกอบรมด้วยออฟติไมเซอร์ ADAM พบว่าขนาดภาพอินพุตที่ 603x603 พิกเซล มีผลการทดสอบที่ดีที่สุด เมื่อเทียบกับขนาดภาพอินพุตที่เล็กกว่า โดยให้ค่า AP ที่สูงสุดเท่ากับ 0.9297 สำหรับข้อมูลที่ผ่านมาเทคนิคการเสริมข้อมูล และ 0.7876 สำหรับข้อมูลปกติที่ไม่ผ่านเทคนิคการเสริมข้อมูล การใช้ขนาดภาพใหญ่กว่าในการฝึกอบรมอาจช่วยให้โมเดลจับภาพลักษณะเฉพาะของวัตถุได้ดีขึ้น ซึ่งส่งผลให้ความแม่นยำในการตรวจจับเพิ่มขึ้น ดังนั้น ขนาดภาพอินพุต 603x603 พิกเซล จึงเป็นขนาดที่เหมาะสมที่สุดสำหรับการฝึกอบรมโมเดลนี้ ในการจำแนกวัตถุในทั้งสภาวะปกติและเทคนิคการเสริมข้อมูล

เทคนิคการเสริมข้อมูลจะช่วยให้โมเดลสามารถเรียนรู้ลักษณะของวัตถุในสถานการณ์ที่แตกต่างกัน ซึ่งอาจทำให้โมเดลสามารถทำงานได้อย่างแม่นยำยิ่งขึ้น การเสริมข้อมูลด้วยเทคนิค 3 แบบคือ การปรับเปลี่ยนค่าสีของภาพในปริภูมิ HSV การสะท้อนภาพแนวนอนแบบสุ่ม และการย่อหรือขยายภาพ 10% แบบสุ่ม สรุปผลได้ดังนี้

1) โมเดลที่ฝึกอบรมด้วยเทคนิคการเสริมข้อมูลทั้ง 3 แบบ จะให้ค่าความแม่นยำเฉลี่ยที่สูงกว่าโมเดลที่ฝึกอบรมโดยไม่ใช้เทคนิคการเสริมข้อมูล 12.50% เมื่อคิดเป็นเปอร์เซ็นต์เปรียบเทียบ ค่าความแม่นยำที่แตกต่างกันระหว่างโมเดลที่ถูกฝึกอบรมด้วยข้อมูลที่ผ่านมาและไม่ผ่านการทำเทคนิคการเสริมข้อมูล

2) ระยะเวลาที่ใช้ในการฝึกอบรมของโมเดลที่ใช้เทคนิคการเสริมข้อมูลด้วยภาพอินพุตทั้ง 3 ขนาดจะนานกว่าการฝึกอบรมของโมเดลที่ไม่ใช้เทคนิคการเสริมข้อมูลโดยภาพอินพุตขนาด 416x416 พิกเซลจะใช้ระยะเวลาในการฝึกอบรมน้อยกว่าภาพอินพุตขนาดอื่น

จากผลการฝึกอบรมโมเดลสรุปได้ว่าการใช้เทคนิคการเสริมข้อมูล มีผลทำให้ระยะเวลาที่ใช้ในการฝึกอบรมโมเดล YOLOv4-Tiny ด้วยออฟติไมเซอร์ ADAM เพิ่มขึ้นอย่างชัดเจนเมื่อเทียบกับการไม่ใช้เทคนิคการเสริมข้อมูลสำหรับทุกขนาดภาพอินพุตที่ทดสอบ ดังนี้

1) สำหรับขนาดภาพอินพุต 416x416 พิกเซล เวลาในการฝึกอบรมเพิ่มขึ้นจาก 6.53 ชั่วโมง เป็น 17.23 ชั่วโมง เมื่อใช้เทคนิคการเสริมข้อมูล

2) ขนาดภาพอินพุต 603x603 พิกเซล, เวลาในการฝึกอบรมเพิ่มขึ้นจาก 20.02 ชั่วโมง เป็น 32.15 ชั่วโมง

3) สำหรับขนาดภาพอินพุต 288x288 พิกเซล, เวลาในการฝึกอบรมเพิ่มขึ้นจาก 16.2 ชั่วโมงเป็น 32.32 ชั่วโมง

การใช้เทคนิคการเสริมข้อมูลยังส่งผลให้ค่าความแม่นยำของโมเดลเพิ่มขึ้นสำหรับทุกขนาดภาพอินพุตที่ทดสอบ ซึ่งแสดงให้เห็นว่าการเสริมข้อมูลช่วยปรับปรุงคุณภาพของโมเดลแม้ว่าจะต้องใช้เวลาในการฝึกอบรมที่นานขึ้น

ค่า Training Loss และ Validation Loss ที่แสดงในกราฟของการฝึกอบรม สรุปได้ดังนี้

- กราฟของ Training Loss และ Validation Loss ควรลดลงเมื่อเวลาผ่านไป
- Training Loss ควรลดลงเร็วกว่า Validation Loss เล็กน้อย
- กราฟของ Training Loss และ Validation Loss ควรแยกออกจากกันเล็กน้อย
- หากกราฟมีลักษณะแตกต่างจากที่กล่าวข้างต้นอาจบ่งชี้ว่าโมเดลอาจถูกกระทบด้วยปัจจัยบางอย่าง เช่น Underfitting หรือ Overfitting การเลือกใช้ออปติไมเซอร์ที่เหมาะสมนั้นมีความสำคัญต่อการควบคุมกระบวนการ

เรียนรู้ของโมเดล ซึ่งสรุปได้ดังนี้

- การเลือกใช้ออปติไมเซอร์แบบ ADAM มีความเหมาะสมที่จะใช้เป็นตัวเลือกสำหรับการฝึกอบรมโมเดลมากกว่าออปติไมเซอร์แบบ SGDM เพื่อใช้ตรวจจับวัตถุหรือภาพรถยนต์ที่มีความละเอียดต่ำขนาดเล็ก
- เทคนิคการเสริมข้อมูลภาพอินพุต เป็นปัจจัยสำคัญเพื่อใช้เพิ่มความแม่นยำในการตรวจจับวัตถุของโมเดล

5.2 ข้อเสนอแนะ

การปรับแต่งค่าพารามิเตอร์ต่าง ๆ ของโมเดลเพื่อให้โมเดลทำงานได้ดีเป็นสิ่งที่อาจจะต้องทดลองกำหนดและปรับปรุงค่าอยู่ตลอดเวลา หากต้องการปรับปรุงความสามารถของโมเดลเพื่อแก้ไขปัญหาละเอียดต่ำขนาดเล็กเหล่านี้ สามารถทำได้ดังนี้

- 1.) เพิ่มขนาดของข้อมูลชุดข้อมูลฝึกอบรม เพื่อให้โมเดลเรียนรู้ข้อมูลได้มากขึ้น
- 2.) ปรับโครงสร้างของโมเดลให้เหมาะสมกับข้อมูลชุดข้อมูลฝึกอบรม เพื่อให้โมเดลสามารถเรียนรู้ข้อมูลที่ซับซ้อนได้มากขึ้น
- 3.) เลือกใช้อัลกอริทึมที่มีประสิทธิภาพในการเรียนรู้ข้อมูลที่ซับซ้อน เช่น อัลกอริทึมการเรียนรู้เชิงลึกที่มีการพัฒนาและปรับปรุงเวอร์ชันปัจจุบัน

นอกจากนี้ การพิจารณาใช้เทคนิคอื่นๆ เพื่อปรับปรุงความสามารถของโมเดล เช่น การเพิ่มจำนวน Epoch ในการฝึกอบรมโมเดล การปรับค่า Hyperparameters ของโมเดล การใช้เทคนิค Regularization เพื่อลดปัญหา Overfitting ค่าพารามิเตอร์ต่างๆ ของโมเดลตรวจจับวัตถุสามารถทดลองปรับค่าและทดสอบการตรวจจับวัตถุเพื่อให้ได้ค่าที่เหมาะสมซึ่งค่าพารามิเตอร์เหล่านี้จะพิจารณาแนวทางในการปรับปรุงโมเดลและการเตรียมข้อมูล เพื่อให้สามารถตรวจจับรถยนต์ขนาดเล็ก

เล็กด้วยภาพในสภาพแวดล้อมความละเอียดต่ำ ได้อย่างมีประสิทธิภาพสูงสุด ทั้งนี้การใช้งานตรวจจับวัตถุด้วย YOLOv4-Tiny ขึ้นอยู่กับข้อมูลและความต้องการประยุกต์ใช้งาน ความแม่นยำและความเร็วในการทำนายเป็นสิ่งสำคัญ เพื่อนำไปต่อยอด พัฒนา งานที่เกี่ยวข้องกับการใช้โดรนในการตรวจจับวัตถุแบบทันที



บรรณานุกรม

TNI

บรรณานุกรม

- [1] M. El-Sharkawy, "Real-time embedded implementation of improved object detector for resource-constrained devices," *Journal of Low Power Electronics and Applications*, vol. 12, no. 2, pp. 21, April 2022.
- [2] S. Ren et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems Journal*, vol. 28, pp.1137-1149, June 2015.
- [3] A. Yadav et al., "Image classification using deep learning and tensorflow," *International Journal of Research in Advanced Science and Engineering Technology*, vol. 4, no. 3, pp. 168-172, May 2022.
- [4] S. Zhang et al., "Object tracking in unmanned aerial vehicle (UAV) videos using a combined approach," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, USA, March 23, 2005, vol. 682, pp. 681-684.
- [5] Y. Wang et al., "A convolutional neural network method based on Adam optimizer with power-exponential learning rate for bearing fault diagnosis," *Journal of Vibroengineering*, vol. 24, no. 4, pp. 22271, June 2022.
- [6] M. Atik et al., "Comparison of YOLO versions for object detection from aerial images," *International Journal of Environment and Geoinformatics*, vol. 9, no. 2, pp. 87-93, June 2022.
- [7] M. Carranza-García et al., "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sensing*, vol. 13, no. 1, pp. 1-23, January 2021.
- [8] P. Dollár et al., "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743-761, April 2012.
- [9] Q. Liu et al., "Object detection based on Yolov4-Tiny and improved bidirectional feature pyramid network," *Journal of Physics: Conference Series*, vol. 2209, no. 1, pp. 8, February 2022.

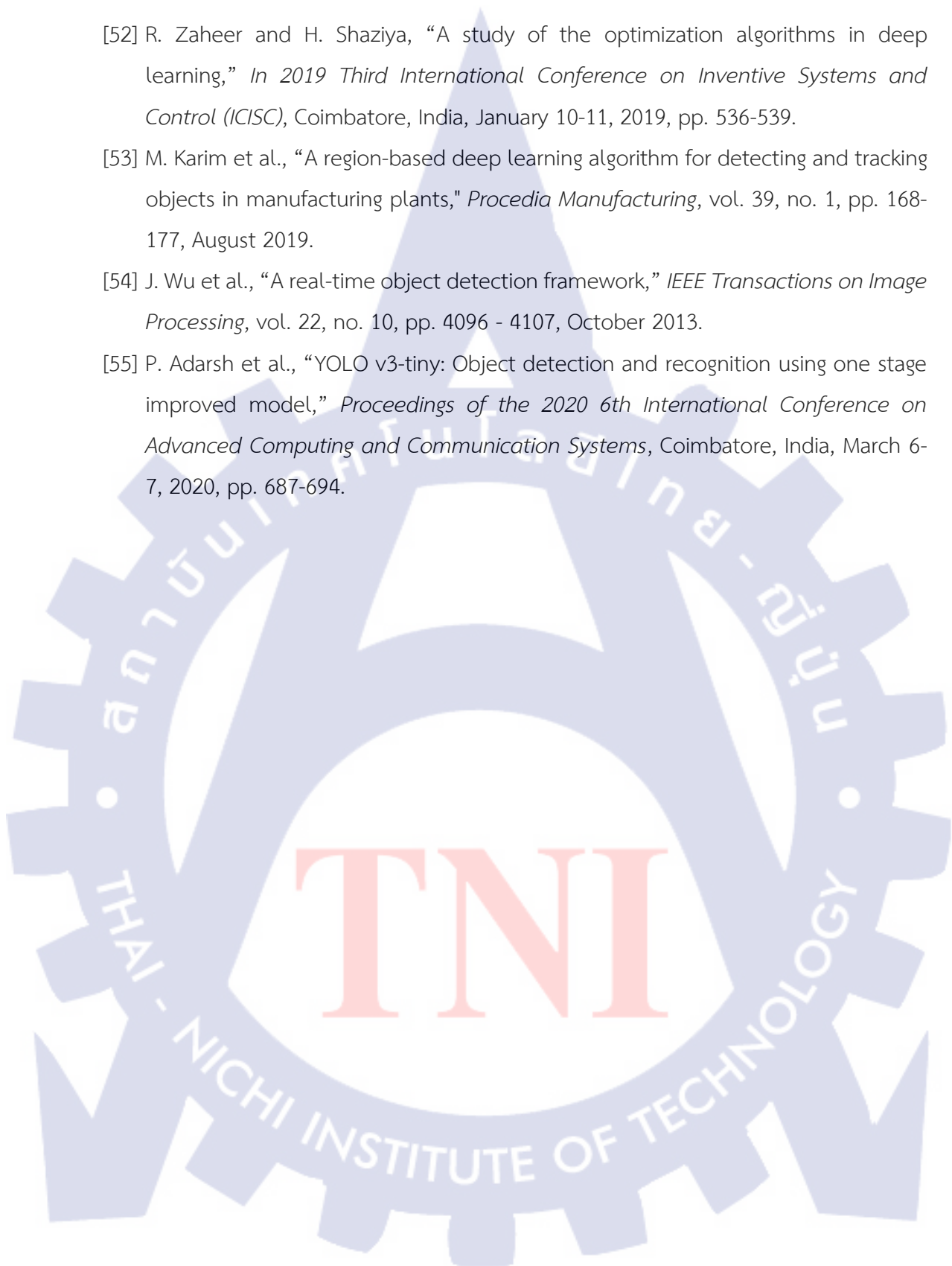
- [10] M. Ju et al., "SuperDet: An efficient single-shot network for vehicle detection in remote sensing images," *Electronics*, vol. 12, no. 6, pp. 1312, March 2023.
- [11] S. Porcu et al., "Evaluation of data augmentation techniques for facial expression recognition systems," *Electronics*, vol. 9, no. 11, pp. 1892, November 2020.
- [12] S. Seo et al., "Automatic detection of construction workers' helmet wear based on lightweight deep learning," *Applied Sciences (Switzerland)*, vol. 12, no. 20, pp. 10369, October 2022.
- [13] J. Yang et al., "A comparative evaluation of convolutional neural networks, training image sizes, and deep learning optimizers for weed detection in alfalfa," *Weed Technology*, vol. 36, no. 4, pp. 512-522, August 2022.
- [14] Y. Ren et al., "Small object detection in optical remote sensing images via modified Faster R-CNN," *Applied Sciences (Switzerland)*, vol. 8, no. 5, pp. 813, May 2018.
- [15] J. Salazar González et al., "Real-time gun detection in CCTV: An open problem," *Neural Networks*, vol. 132, no.1, pp. 297-308, September 2020.
- [16] A. Ramachandran et al., "A review on object detection in unmanned aerial vehicle surveillance," *International Journal of Cognitive Computing in Engineering*, vol. 2, no.1, pp. 215-228, June 2021.
- [17] S. Lu et al., "A real-time object detection algorithm for video," *Computers and Electrical Engineering*, vol. 77, no.2, pp. 398-408, July 2019.
- [18] S. Sergio et al., "Developing a real-time social distancing detection system based on YOLOv4-tiny and bird-eye view for COVID-19," *Journal of Real-Time Image Processing*, vol. 19, pp. 551-563, February 2022.
- [19] L. Tan et al., "Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, pp. 324, November 2021.
- [20] W. Wang et al., "GAN-knowledge distillation for one-stage object detection," *IEEE Access*, vol. 8, pp. 60719-60727, March 2020.
- [21] Y. Xu et al., "Training robust object detectors from noisy category labels and imprecise bounding boxes," *IEEE Transactions on Image Processing*, vol. 30, pp. 5782-5792, June 2021.

- [22] M. Zhang et al., "Multi-scale aerial target detection based on densely connected inception resnet," *IEEE Access*, vol. 8, pp. 84867-84878, May 2020.
- [23] T. Zhao et al., "Car detection in low resolution aerial images," *Image and Vision Computing*, vol. 21, no. 8, pp. 693-703, August 2003.
- [24] Z. Zou et al., "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 30, pp. 257-276, May 2019.
- [25] I. Ward et al., "RGB-D image-based object detection: from traditional methods to deep learning techniques," *Advances in Computer Vision and Pattern Recognition*, vol. 30, pp. 169-201, October 2019.
- [26] S. Abbas et al., "Region-based object detection and classification using faster R-CNN," *Proceedings of the 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT)*, Ghaziabad, India, February 9-10, 2018, pp. 1-6.
- [27] M. Zhang et al., "Using computer vision to automate hand detection and tracking of surgeon movements in videos of open surgery," *Proceedings of the American Medical Informatics Association (AMIA)*, vol. 2020, no. 1, pp. 1373-1382, January 2021.
- [28] J. Redmon et al., "YOLO9000: better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 21-26, 2017, pp. 106-115.
- [29] J. Redmon et al., "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 27-30, 2016, pp. 779-788.
- [30] Z. Wang et al., "A review of object detection based on convolutional neural network," in *36th Chinese Control Conference*, Dalian, China, July 26-28, 2017, pp. 11104-11109.
- [31] A. Ramachandran et al., "A review on object detection in unmanned aerial vehicle surveillance," *Proceedings of the International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 215-228, June 2021.
- [32] K. Pal et al., "Deep learning in multi-object detection and tracking: State of the art," *Applied Intelligence*, vol. 51, no. 9, pp. 4845-4873, September 2021.

- [33] A. Bochkovskiy et al., "YOLOv4: optimal speed and accuracy of object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, Washington, USA, June 14-19, 2020, pp. 10934.
- [34] N. Zarei et al., "Fast-yolo-rec: Incorporating yolo-base detection and recurrent-base prediction networks for fast vehicle detection," *Journal of IEEE Xplore*, vol. 10, pp. 120592 – 120605, November 2022.
- [35] S. Srivastava et al., "Optical character recognition techniques: A review," *IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, BHOPAL, India, February 19-20, 2022, pp. 1-6.
- [36] M. Babanezhad et al., "An optimization method for convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 5789-5803, December 2018.
- [37] P. Ganesan et al., "A comprehensive study of edge detection for image processing applications," in *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, BHOPAL, India, March 17-18, 2017, pp. 1-6.
- [38] R. Girshick et al., "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 2, pp. 202-214, May 2015.
- [39] R. Abushahma et al., "Region-based convolutional neural network as object detection in Images," *IEEE 7th Conference on Systems, Process and Control (ICSPC)*, Melaka, Malaysia, December 13-14, 2019, pp. 1-6.
- [40] L. Huang et al., "Data augmentation for deep learning-based radio modulation classification," *Journal of IEEE Access*, vol. 8, pp. 1498-1506, December 2019.
- [41] J. Wei et al., "The status and challenges of image data augmentation algorithms," *Journal of Physics: Conference Series*, vol. 2456, no. 1, pp. 12041, March 2023.
- [42] E. Hassan et al., "The effect of choosing optimizer algorithms to improve computer vision tasks: A comparative study," *Multimedia Tools and Applications*, vol. 82, no. 11, pp. 16591-16633, September 2022.

- [43] B. Tian and W. Wei, "Research overview on edge detection algorithms based on deep learning and image fusion," *In Security and Communication Networks Hindawi Journal*, vol. 2022, pp. 13809, January 2022.
- [44] S. Pillai et al., "Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning," *Soft Computing*, vol. 25, no. 1, pp. 11929–11940, September 2021.
- [45] I. Park and S. Kim, "Performance indicator survey for object detection," *In 2020 20th International Conference on Control, Automation and Systems (ICCAS)*, Busan, Korea (South), October 13-16, 2020, pp. 171-176.
- [46] S. Postalcioglu, "Performance analysis of different optimizers for deep learning-based image recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 2, pp. 1793-6381, February 2020.
- [47] L. Zheng et al., "YOLOv3: Face detection in complex environments," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 1153-1160, July 2020.
- [48] T. Hafsia et al., "Implementing canny edge detection algorithm for different blurred and noisy images," *In 2022 IEEE 21st International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Sousse, Tunisia, December 19-21, 2022, pp. 342-349.
- [49] P. Ganesan et al., "HSV color space based segmentation of region of interest in satellite images," *In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kanyakumari, India, July 10-11, 2014, pp. 118-123.
- [50] Y.-y. Zheng et al., "Edge detection methods in digital image processing," *In 2010 5th International Conference on Computer Science & Education (ICCSE)*, Hefei, China, August 24-27, 2010, pp. 471-473.
- [51] C. F. Santos and J. P. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Computing Surveys*, vol. 54, no. 10, pp. 1-25, September 2022.

- [52] R. Zaheer and H. Shaziya, "A study of the optimization algorithms in deep learning," In *2019 Third International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, January 10-11, 2019, pp. 536-539.
- [53] M. Karim et al., "A region-based deep learning algorithm for detecting and tracking objects in manufacturing plants," *Procedia Manufacturing*, vol. 39, no. 1, pp. 168-177, August 2019.
- [54] J. Wu et al., "A real-time object detection framework," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 4096 - 4107, October 2013.
- [55] P. Adarsh et al., "YOLO v3-tiny: Object detection and recognition using one stage improved model," *Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems*, Coimbatore, India, March 6-7, 2020, pp. 687-694.





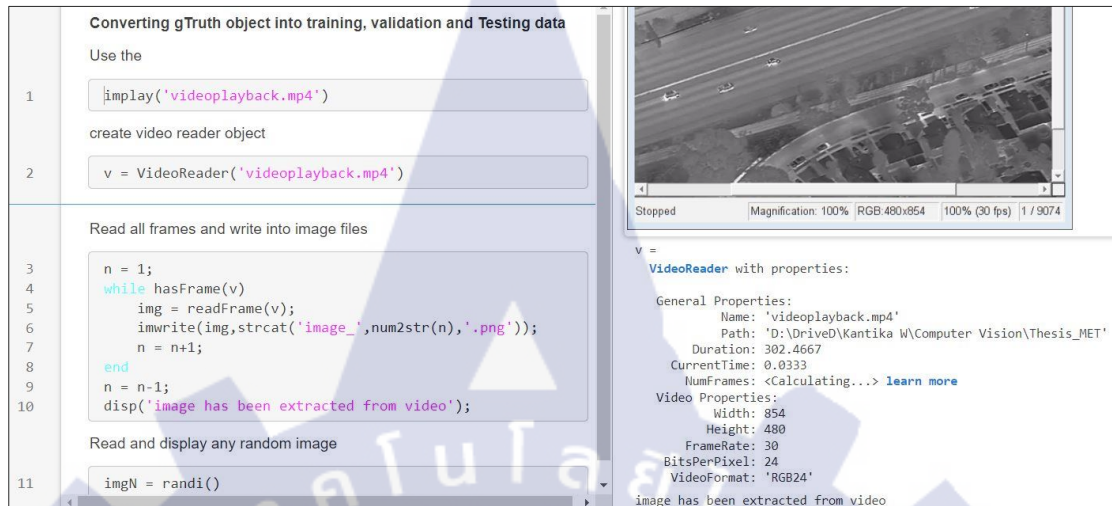
ภาคผนวก

ภาคผนวก ก.

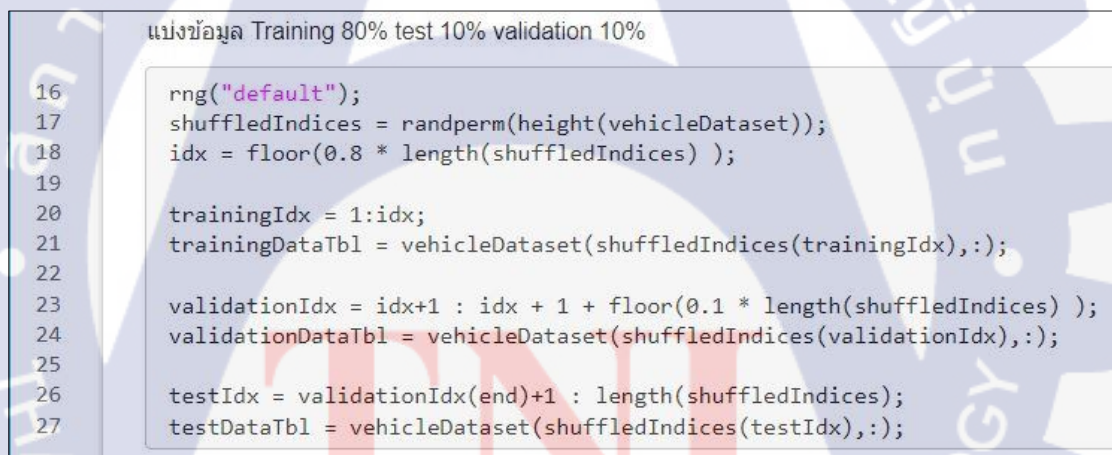
โค้ดโปรแกรมภาษา MATLAB



เปลี่ยนข้อมูลวิดีโอเป็นรูปภาพ



รูปที่ ก.1 แบ่งชุดข้อมูลออกเป็น Training , Test , Validation



รูปที่ ก.2 แบ่งชุดข้อมูลออกเป็น Training , Test , Validation

```

1  function data = augmentData(A)
2  %สร้างฟังก์ชันสำหรับการทำ Augmentation
3  %ใช้ฟังก์ชัน jitterColorHSV
4  data = cell(size(A));
5  for ii = 1:size(A,1)
6      I = A{ii,1};
7      bboxes = A{ii,2};
8      labels = A{ii,3};
9      size = size(I);
10
11      if numel(sz) == 3 && sz(3) == 3
12          I = jitterColorHSV(I,...
13              contrast=0.5,...
14              Hue=0.5,...
15              Saturation=0.2,...
16              Brightness=0.8);
17      end
18      % Randomly flip image.
19      transform = randomAffine2d(XReflection=true,Scale=[1 1.1]);
20      rout = affineOutputView(sz,transform,BoundsStyle="centerOutput");
21      I = imwarp(I,transform,OutputView=rout);
22

```

รูปที่ ก.3 ฟังก์ชันการทำ Data Augmentation

Create a YOLO v4 Object Detector Network

Specify the network input size to be used for training.

กำหนดขนาดอินพุตของโมเดล YOLOv4-Tiny

50

Specify the name of the object class to detect.

กำหนด Class มี 1 Class คือ vehicle

51

Use the `estimateAnchorBoxes` function to estimate anchor boxes based on the size of objects in the training data. |

52
53
54
55
56
57
58
59
60
61
62
63

```

rng("default")
trainingDataForEstimation = transform(trainingData,@(data)preprocessData(data,inputSize));
numAnchors = 9;
[anchors,meanIoU] = estimateAnchorBoxes(trainingDataForEstimation,numAnchors);

area = anchors(:, 1).*anchors(:,2);
[~,idx] = sort(area,"descend");

anchors = anchors(idx,:);
anchorBoxes = {anchors(1:3,:),
               anchors(4:6,:)};

```

รูปที่ ก.4 กำหนดขนาดอินพุตของโมเดล YOLOv4

```

64 detector = yolov4ObjectDetector("tiny-yolov4-coco",className,anchorBoxes,InputSize=inputSize);
65 detector

detector =
  yolov4ObjectDetector with properties:
    Network: [1x1 dlnetwork]
    AnchorBoxes: {2x1 cell}
    ClassNames: {'vehicle'}
    InputSize: [608 608 3]
    ModelName: 'tiny-yolov4-coco'

```

รูปที่ ก.5 กำหนดโมเดลตัวตรวจจับวัตถุ YOLOv4-Tiny

Specify Training Options

```

77 options1 = trainingOptions("sgdm",...
78   InitialLearnRate=0.0001,...
79   LearnRateSchedule="none",...
80   MiniBatchSize=16,...
81   L2Regularization=0.0005,...
82   MaxEpochs=100,...
83   BatchNormalizationStatistics="moving",...
84   DispatchInBackground=true,...
85   ResetInputNormalization=false,...
86   VerboseFrequency=20,...
87   Shuffle="every-epoch",...
88   CheckpointPath=tmpdir,...
89   ValidationData=validationData,...
90   ExecutionEnvironment="auto",...
91   Plots="training-progress");
92 %teainedDetector = trainYOLOv4ObjectDetector(trainingData,detector,options);
93

```

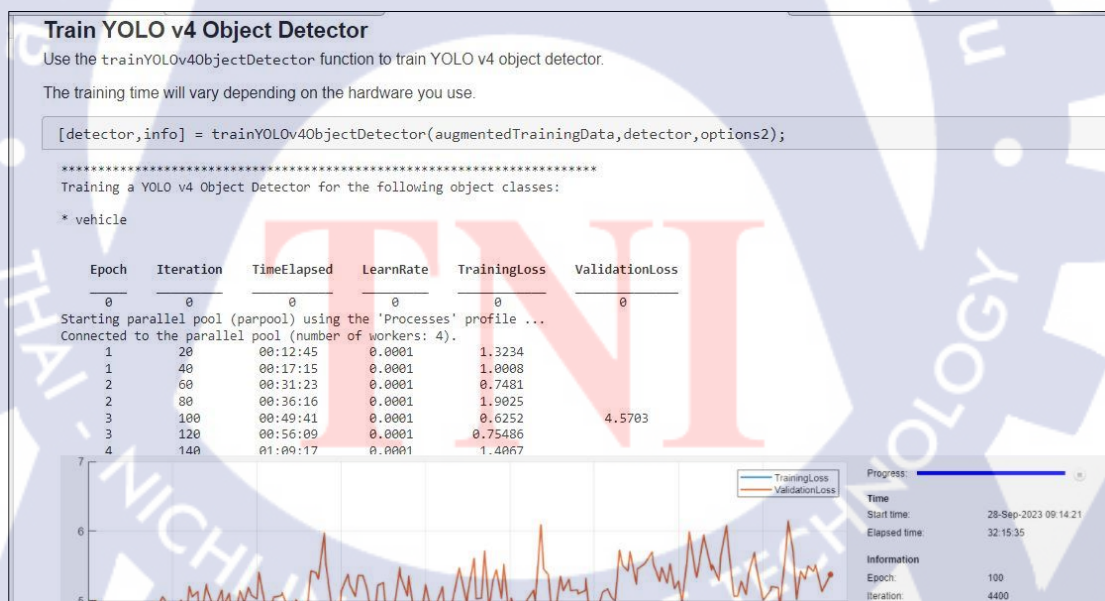
รูปที่ ก.6 กำหนด Option Optimizer SGDM ใช้ในการเทรน Detector


```

94 options2 = trainingOptions("adam",...
95     GradientDecayFactor=0.9,...
96     SquaredGradientDecayFactor=0.999,...
97     InitialLearnRate=0.0001,...
98     LearnRateSchedule="none",...
99     MiniBatchSize=16,...
100    L2Regularization=0.0005,...
101    MaxEpochs=100,...
102    BatchNormalizationStatistics="moving",...
103    DispatchInBackground=true,...
104    ResetInputNormalization=false,...
105    Shuffle="every-epoch",...
106    VerboseFrequency=20,...
107    CheckpointPath=tmpdir,...
108    ValidationData=validationData,...
109    ExecutionEnvironment="auto",...
110    Plots="training-progress");

```

รูปที่ ก.7 กำหนด Option Optimizer ADAM ใช้ในการเทรน Detector



รูปที่ ก.8 เทรนตัวตรวจจับวัตถุ YOLOv4-Tiny กับ Optimizer ที่กำหนด

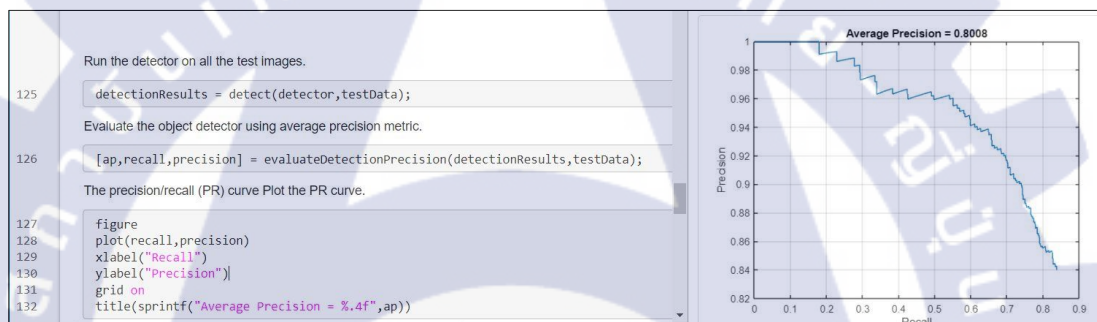
Run the detector on a test image.

```
I = imread("videoplayback1_286.png");
[bboxes,scores,labels] = detect(detector,I);
```

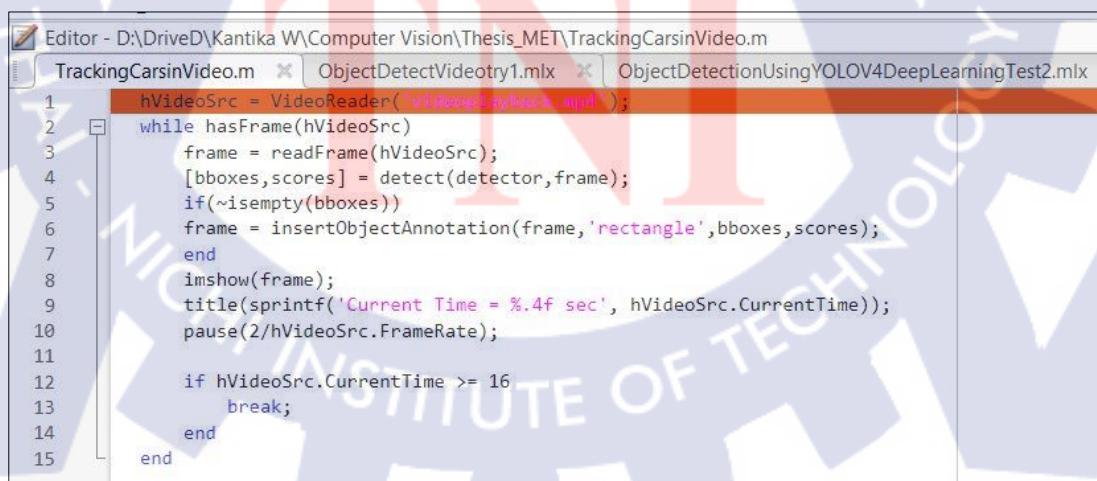
Display the results.

```
I = insertObjectAnnotation(I,"rectangle",bboxes,scores);
figure
imshow(I)
```

รูปที่ ก.9 รันตัวตรวจจับวัตถุกับภาพทดสอบ



รูปที่ ก.10 ประเมินประสิทธิภาพของโมเดลตัวตรวจจับวัตถุที่ถูกเทรนแล้ว



รูปที่ ก.11 รันวิดีโอทั้งหมดกับโมเดลตัวตรวจจับวัตถุที่ถูกเทรนแล้ว

ภาคผนวก ข.
บทความการประชุมวิชาการ





TSAE 2023

**THE 16th TSAE INTERNATIONAL CONFERENCE
& 24th TSAE NATIONAL CONFERENCE**

26-27 July 2023

Available online at tsae2023.rmutto.ac.th

1 Car Detection based on YOLOv4-Tiny Using Low-resolution Video from Drone

2 Kantika Wongkasem*, Pornchai Nivesrangsan

3 Thai-Nichi Institute of Technology, Faculty of Engineering, Bangkok, Thailand

4 *Corresponding author: Kantika Wongkasem Tel: +66-9-8395-5444, E-mail: wo.kantika_st@tni.ac.th

5 Abstract

Object detection is a computer vision technology that localizes and identifies objects in an image. An advanced form of image classification is where a neural network predicts objects in an image and points them out in the form of bounding boxes. Low-resolution of vehicle object detection from UAV images is a challenging task for tiny object detection. One of the major challenges is to use low-resolution video from unmanned aerial vehicles or aerial photography by drone. Deep learning technique is powerful for computer vision technology in which the object detector automatically learns image features required for detection tasks. Several techniques for object detection using deep learning are available such as Faster R-CNN, and YOLO family. In this paper, we compare the performance of tiny car detection state-of-the-art model based on the YOLOv4-Tiny deep learning algorithm for drone videos, in the context of low-resolution, data augmentation, and optimizers comparison. We trained an algorithm on a car image dataset in a video and we compared the results with the average precision. Experimental results in the study compares the performance of different optimizers and explores the impact of data augmentation techniques on detection results. Detection performance shows that the average precision of applying data augmentation techniques during the training process of YOLOv4-Tiny with Adam optimizer outperforms Sgdm optimizer reaching 0.8735 for the correct detection. This paper tries to compare the selection of the best advantage parameter, options, and training time in- tiny vehicle data detection framework. The findings demonstrate the potential of YOLOv4-Tiny for car detection and highlight the significance of optimizer selection and data augmentation in improving detection performance.

6 Keywords: Vehicles detection, Data Augmentation, Optimizers, Object detection, YOLOv4-Tiny

7 1 Introduction

Object Detection is a branch of computer vision technology, a field that is widely used in self-driving cars to detect vehicles or detect pedestrians and so on that localizes and identifies objects in digital images or videos. Most of the techniques were developed from deep learning techniques. Deep Learning for object detection has become so ubiquitous and ambient that many people do not recognize what they are using on a daily basis. such as face detection for cameras on phones(Zou et al. (2019)), pedestrian detection(Dollar et al. (2012)), worker's helmet detection(Liang et al. (2022)), and pill detection(Tan et al. (2021)). Vehicle detection with low-resolution(Zhao et al. (2003)) video from drones is a significant application.

Drones equipped with cameras are widely used for aerial surveillance(Ramachandran et al. (2021)), traffic monitoring, security purposes, and various other applications. The ability to detect and track cars in low-resolution drone videos(zhang (2020)) can provide valuable insights into traffic patterns, vehicle behavior, and event monitoring in real time. Overall,

car detection based on using low-resolution video from drones has numerous practical applications across various domains, including transportation, security, urban planning, and environmental impact assessment. It allows for real-time analysis(Lu et al. (2019)), efficient resource utilization, and informed decision-making based on the detected car information. The increasing use of UAVs in various fields is due to their higher mobility and flexibility compared to CCTV(Gonzalez et al. (2020)). In recent years many object detection focuses on addressing a safety concern related to using UAVs in urban environments and how real-time deep learning algorithms(Abbas et al. (2017)) with improved speed and accuracy are changing how UAVs are being utilized in modern society.

For object detection that there are two key areas of target detectors based on deep convolutional neural networks two-stage object detectors(Carranza et al. (2021)) and one-stage object detectors(Wang et al. (2020)). Two-stage object detectors follow a two-step process of region proposal and object classification. In the region proposal stage, these detectors generate a set of region proposals that are likely to contain

objects. This is typically achieved using a region proposal network (RPN) or a similar mechanism (Ren et al. (2018)). The RPN proposes a set of candidate regions or bounding boxes in an image based on predefined anchor boxes and their spatial features. In the object classification stage, the proposed regions are refined and classified to determine the presence and class of objects. This step involves extracting region features (Zhiqiang et al. (2017)), performing bounding box regression to refine the proposals (Xu et al. (2021)), and classifying the refined regions using a classification network. Two popular examples of two-stage object detectors are Faster R-CNN (Region-based Convolutional Neural Networks) (Abbas et al. (2018)) and Mask R-CNN. These detectors generally provide accurate object detection results, but they can be computationally expensive due to the two-stage process. One-stage object detectors directly predict the presence, class, and bounding box coordinates of objects in a single pass through the network.

These detectors eliminate the region proposal step, resulting in faster inference times compared to two-stage detectors (Li et al. (2022)). One-stage detectors divide the input image into a grid of predefined anchor boxes and predict the objectness score, class probabilities, and bounding box offsets for each anchor box. They use convolutional layers to simultaneously predict multiple objects at different scales (Pal et al. (2021)) and aspect ratios. Examples of one-stage detectors include YOLO (Atik et al. (2022)), YOLO v2 (Redmon et al. (2017)), YOLO v3 (Redmon et al. (2016)), YOLOv4-Tiny (Liu et al. (2022)) and so on. This study focuses on the use of deep learning for object detection and emphasizes the importance of datasets and more powerful models to improve the performance of deep learning approaches to object detection with low-resolution images. The choice of optimizer does have an impact on training the Car Detector model (Ju et al. (2023)), and different optimizers have their own strengths and weaknesses.

In this paper, we consider optimizers technique, Adam optimizer with Sgdm optimizer and modified hyperparameters for comparison based on yolov4-tiny model in low-resolution image datasets and enhance the network's performance by conducting data augmentation with techniques color jitter augmentation in HSV space, random horizontal flip on the training dataset and random scaling by 10 percent. There are many factors that explain different reported results and use the average precision metric to evaluate the performance detector to compare the results.

2 Materials and Methods

YOLO (You Only Look Once) is an object detection algorithm known for its real-time

performance and accuracy. YOLOv4 and YOLOv4-Tiny are two different versions of the YOLO architecture, each with its own characteristics and trade-offs. Here are the main for YOLOv4-Tiny.

This table provides a concise overview of the architectural differences, layer counts, input sizes, parameters, model sizes, detection speed, and performance between YOLOv4 and YOLOv4-Tiny.

Table 1 Differences between YOLOv4 and YOLOv4-Tiny.

Aspect	YOLOv4	YOLOv4-Tiny
Architecture	Darknet-53	Darknet-53
Backbone	53 convolutional layers	21 convolutional layers
Output Layers	3	2
Input Size	Larger	Smaller
Parameters	More	Less
Model Size	Larger	Smaller
Detection Speed	Slower	Faster
Performance	Higher accuracy and precision	Slightly lower accuracy and precision

2.1 Data Used

This paper uses a vehicle dataset extracted from Video that contains 873 images. Each image contain three to ten labeled instances of a vehicle. The property of video have Bits per pixel is 24, Framerate 30, Height and Width is 480 x 854 and split the dataset into training 80%, validation 10%, test sets 10%. The experimental run on an NVIDIA T500 with processor 11th Gen Intel® Core™ i7-1165G7 @2.80 GHz.

2.2 YOLOv4-Tiny

YOLOv4-Tiny is a variant of the YOLO (You Only Look Once) object detection model, designed to be smaller and faster compared to the original YOLOv4 model. YOLOv4-Tiny follows the same principle as YOLO in that it performs object detection in a single pass through the network, enabling real-time inference on resource-constrained devices. Architecture of YOLOv4-Tiny (Sergio et al. (2022)) consists of a convolutional neural network architecture composed of multiple layers. It typically has fewer layers compared to the full YOLOv4 model, resulting in a more lightweight and compact model.



Figure 1 YOLOv4-Tiny Network architecture

Feature Extraction of YOLOv4-Tiny employs a series of convolutional layers to extract features from the input image. These layers utilize filters and activation functions to capture different levels of visual information, allowing the model to understand various object features and their spatial relationships. Detection Layers of YOLOv4-Tiny has two detection layers that are responsible for predicting bounding boxes and associated class probabilities. These detection layers capture objects at different scales and resolutions, enabling the detection of objects of various sizes. Anchor Boxes of YOLOv4-Tiny utilizes anchor boxes, which are pre-defined bounding box priors that represent different object aspect ratios and scales. These anchor boxes are used during the prediction process to match them with the detected objects. Prediction Process of YOLOv4-Tiny divides the input image into a grid of cells. Each cell is responsible for predicting the bounding boxes (Gavrilova et al., (2019)) and class probabilities for objects within its spatial region. The model outputs the predicted bounding boxes, associated class labels, and confidence scores for each predicted object. Speed and Efficiency of YOLOv4-Tiny is optimized for fast inference on resource-constrained devices, such as embedded systems or mobile devices. Its reduced model size and simplified architecture allow for faster computation, making it suitable for real-time applications where speed is crucial. While YOLOv4-Tiny sacrifices some accuracy compared to the full YOLOv4 model, it offers a good balance between speed and detection performance. YOLOv4-Tiny is commonly used in scenarios where real-time object detection is required on devices with limited computational resources, or when processing high-resolution images or video streams.

The important to note that the specific architecture and implementation details of YOLOv4-Tiny may vary slightly depending on the specific implementation or framework being used.

2.3 Optimizer Technique

The choice of optimizer does have an impact on training the Car Detector model, and different

optimizers have their own strengths and weaknesses. While both Adam (Adaptive Moment Estimation) and Sgdm (Stochastic Gradient Descent with Momentum) are widely used optimizers, their performance can vary depending on the specific characteristics of the dataset and task. In this work have some considerations. For generalization Sgdm, with its simplicity and stability, may sometimes lead to better generalization compared to Adam.

Adam optimizer combines the benefits of adaptive learning rates and momentum to efficiently update model parameters during training. The key idea behind Adam is to compute adaptive learning rates for each parameter by estimating the first and second moments of the gradients. It maintains two moving average accumulators: the first moment (mean) of the gradients and the second moment (variance) of the gradients. By considering these moments, Adam adapts the learning rates for each parameter, allowing it to converge faster and handle different gradients efficiently. The algorithm has been designed to include a bias correction method that effectively handles the initialization bias of the moving average estimators. Therefore Adam has been shown to work well in practice for a wide range of tasks and is known for its stability and fast convergence.

Sgdm optimizer is an extension of the classic Stochastic Gradient Descent (SGD) that introduces a momentum term. Momentum helps accelerate convergence and overcome local optima by adding a fraction of the previous update to the current update. The momentum term accumulates the past gradients direction and provides a smoother and more consistent update direction, especially when the gradients have high variance. And Sgdm performs parameter updates by multiplying the current gradient by the learning rate and adding the momentum term. The momentum hyperparameter controls the influence of previous updates on the current update. Higher momentum values allow the optimizer to move faster in relevant directions and dampen oscillations in flat or noisy regions. Therefore Sgdm is widely used and provides good convergence behavior in many optimization scenarios, including deep learning tasks. In the context of training the car detector, both Adam and Sgdm can yield satisfactory results. However, the choice between the two depends on factors such as dataset characteristics, model complexity, and convergence speed. It is recommended to experiment with different optimizers and evaluate their performance on your specific task to determine the most suitable option. While Adam is often preferred due to its adaptive learning rate and robustness to different datasets, it may not always outperform Sgdm in every scenario. Therefore, it is essential to conduct experiments and compare the performance of both optimizers in terms

of metrics such as accuracy, convergence speed, and generalization capabilities on your car detection task. This empirical evaluation will provide valuable insights into which optimizer is better suited for data specific application. For convergence speed Adam optimizer, which combines the benefits of adaptive learning rates and momentum, often converges faster than traditional Sgdm. It tends to handle sparse gradients and noisy data better, making it a popular choice for deep learning tasks. This can be advantageous when training the car detector model, as faster convergence means fewer training iterations to reach a satisfactory performance level. For hyperparameter sensitivity Adam(Wang et al. (2022)) has additional hyperparameters compared to Sgdm, such as the learning rate and adaptive moment estimation parameters. While these additional parameters can ease the tuning process, they also introduce sensitivity to hyperparameter choices. Fine-tuning Adam's hyperparameters may be required to achieve optimal performance. On the other hand, Sgdm has fewer hyperparameters, making it somewhat less sensitive to their values. The last is batch size Adam can be more suitable for larger batch sizes due to its adaptive learning rate properties, while Sgdm may require careful tuning of learning rate schedules to ensure stable convergence. In the case of training the car detector model, you may need to consider the available computational resources and determine an appropriate batch size accordingly. It's important to note that the performance of optimizers can depend on the specific dataset, network architecture, hyperparameter settings, and other factors.

In this study, the performance of the Car Detector model was evaluated using both Sgdm and Adam optimizers. The decision to include both optimizers was driven by the desire to explore and compare their effects on the training process and detection performance. Therefore, the optimizer used in the final implementation of the Car Detector model was Adam, as it outperformed the alternative optimizer in terms of the evaluation metrics mentioned earlier. This choice was deemed more suitable for the specific car detection task and dataset."

2.4 Data Augmentation

In object detection, data augmentation can be used to increase the number of images with objects in different positions and orientations. This can help models learn to detect objects in a wider variety of contexts. Data augmentation is a crucial technique in training deep learning models for car detection, including YOLOv4-Tiny. It involves applying various transformations to the training dataset to increase its diversity and improve the model's generalization capabilities. The following are the

theoretical principles and reasons for utilizing data augmentation in car detection. For increased training data with data augmentation allows us to generate additional training samples by applying transformations to the existing dataset. By augmenting the data, we effectively increase the size of the training set, which helps prevent overfitting and improves the model's ability to generalize to unseen data. One thing for augmentation of the training data can increase variability. A wide range of variations can occur in real-world car detection scenarios. This includes variations in object size, aspect ratio, position, and appearance. By exposing the model to these variations during training, it learns to recognize and handle them effectively, improving its detection performance in diverse situations.

Data augmentation helps improve the model's generalization, robustness, and ability to handle different variations in real-world scenarios. For car detection models need to improve robustness including changes in lighting conditions, object orientations, and scales. By applying data augmentation techniques, such as color jitter, augmentation in HSV space, random horizontal flips, and random scaling. Therefore, variability in the training data. This helps the model learn to recognize cars under different conditions, making it more robust and adaptable to car detection scenarios.

Data augmentation with the techniques mentioned (color jitter augmentation in HSV space, random horizontal flip, and random scaling by 10 percent) is commonly used in computer vision tasks, including object detection. Each technique serves a specific purpose in improving the model's performance and generalization ability. The First one in this experiment we used Color Jitter Augmentation in HSV Space. Color jitter augmentation aims to introduce variations in color and improve the model's robustness to changes in lighting conditions. By applying the augmentation in the HSV color space, in this experiment that modify the hue, saturation, and value components individually. The second one is Random Horizontal Flip. Random horizontal flipping is a common augmentation technique that helps the model learn to detect objects from different viewpoints and orientations. By randomly flipping the input images, can increase the variability of the training data without changing the object's class label. This augmentation helps the model become invariant to horizontal mirroring or changes in the object's position within the image. The last technique used random scaling by 10 percent. The random scaling introduces variations in the scale or size of objects within the input images. By randomly scaling the images by a certain percentage, such as 10 percent in this case. In this paper simulate different distances from the camera or changes in object sizes. This

augmentation helps the model handle objects at different scales and improves its ability to detect objects of various sizes in real-world scenarios.

2.5 Deep Convolutional Neural Networks(CNNs)

Deep Convolutional Neural Networks (CNNs) are a type of artificial neural network designed specifically for analyzing visual data such as images. They are composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. CNNs excel at extracting features from input data through convolutional operations, enabling them to capture spatial relationships and hierarchical representations within images. Deep CNNs have revolutionized computer vision tasks by achieving state-of-the-art performance in various applications, including image classification, object detection, and image segmentation.

2.6 Anchor Boxes

Anchor boxes, also known as prior boxes, are a concept used in object detection algorithms, particularly in two-stage detectors like Faster R-CNN and SSD (Single Shot MultiBox Detector). Object detection involves identifying and localizing multiple objects within an image. Anchor boxes are predefined bounding boxes of different scales and aspect ratios that are placed at various positions across the image. During the training process, these anchor boxes act as reference templates, and the model predicts the offsets from these anchor boxes to accurately localize objects. By using anchor boxes, the model can handle objects of different sizes and aspect ratios efficiently.

2.7 Bounding Box

A bounding box is a rectangular box that is used to define the spatial extent of an object within an image or video frame. In the context of "bounding box for tiny car detection," it refers to the process of detecting and precisely localizing small or tiny car objects within an image or video. A bounding box is typically represented by its coordinates, such as the x and y coordinates of the top-left corner of the box and its width and height. By enclosing the detected tiny car within a bounding box, the system can accurately locate and identify these objects for further analysis or processing.

3 Results and Discussion

This paper proposed a deep learning base on YOLOv4-Tiny model for car detection video from drone. The comparison results of average precision shown different evaluation metrics were used in different conditions of optimizer technique separate into Adam and Sgdm with same condition followed by this table

Table 2 condition hyperparameters used in this experiment

Inputsize	[416 416 3]
ClassNames	{'vehicle'}
ModelName	'yolov4-tiny'
MiniBatchSize	16
MaxEpochs	100
InitialLearnRate	0.0001

The YOLOv4-Tiny model was trained using MATLAB deep learning framework. The training process consisted of several steps. First, the dataset of low-resolution videos from drones, containing car images, was prepared. The dataset was annotated with bounding box labels indicating the location of cars in each frame. The YOLOv4-Tiny model was then initialized with pre-trained weights, either from ImageNet or pre-trained on a large-scale dataset. This initialization helps the model learn generic features and speeds up convergence.

During the training process, the model was optimized using either SGDM or ADAM optimizer, as mentioned earlier. The hyperparameters such as learning rate, mini-batch size, and number of epochs were set to specific values. It is worth noting that these hyperparameters were carefully selected through experimentation and fine-tuning to achieve optimal performance.

And separate the training data into dataset normal and data sets with augmentation techniques. For data augmentation used the following techniques

1. Color jitter augmentation in HSV space
2. Random horizontal flip
3. Random scaling by 10 percent

The results can compare in the table below

Table 3 Comparison results of YOLOv4-Tiny algorithm. The values are given in terms of average precision

Optimizer	Average precision	
	Dataset Normal	Dataset with Augmentation
SGDM	0.7855	0.8116
ADAM	0.7920	0.8735

In this table can be described the following optimizer techniques and data for used in training. The important point is the performance of optimizers can depend on the specific dataset, network architecture, hyperparameter settings, and other factors. While Adam has shown promising performance in many scenarios.

In terms of dataset comparison with augmentation techniques discuss the theoretical reasons behind selecting the first one Color jitter augmentation in HSV (Hue, Saturation, Value) space is chosen because it provides a way to introduce controlled variations in color attributes while maintaining the semantic content of the images. By applying adjustments to the hue, saturation, and value components, the augmentation technique helps enhance the model's ability to generalize to different lighting conditions, color variations, and contrast levels. The second is the random horizontal flip technique helps to increase the diversity of training samples and reduces the model's bias toward specific object orientations. The probability of flipping an image horizontally can be mentioned to indicate the likelihood of this augmentation being applied during training. The last technique is random scaling by 10 percent which introduces variability in object sizes. By randomly scaling the images up or down by a factor of 10 percent, the augmentation helps the model learn to detect objects at different scales. The scaling factor can be determined based on the original size of the objects in the dataset.

For the Sgdm Optimizer Technique without data augmentation, the average precision was found to be 0.7855. On the other hand, for the same conditions and with the application of data augmentation techniques, including color jitter augmentation in HSV space, random horizontal flip, and random scaling by 10 percent, the results for the Adam optimizer technique yielded the highest average precision, with a value of 0.8735 but take a long time for training detector to 17 hours with 4400 Iteration



Figure 2 Training YOLOv4-Tiny with Adam optimizer with data augmentation

And test the model with the test image. the results of car detection using YOLOv4-Tiny with different optimizers and data augmentation techniques are evaluated in terms of average precision



Figure 3 Run the detector YOLOv4-Tiny with Adam optimizer with data augmentation on a test image

Average Precision measures the accuracy of object detection by calculating the area under the precision-recall curve. It considers both precision (the percentage of correctly detected objects among the predicted ones) and recall (the percentage of correctly detected objects among all the ground truth objects). AP is often reported at different intersections over union (IoU) thresholds. Average precision provides a comprehensive assessment of the car detector's performance. A higher average precision indicates better detection performance, with values typically ranging from 0 to 1.

as shown in the figure below

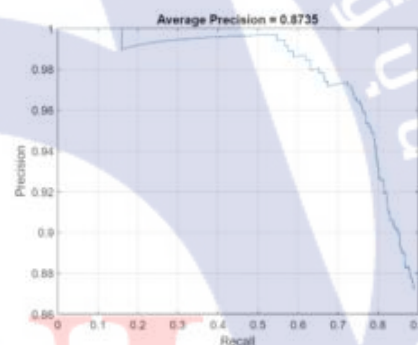


Figure 4 Evaluate Object Detector using Average Precision Metric

This picture shows the average precision of training data with augmentation techniques with Adam optimizer. The calculated average precision for the car detector in my research is 0.8735, which is a comprehensive measure that considers both precisions that the ability to make correct classifications and recall that the ability to detect all relevant objects. Our study makes several significant contributions to the field of car detection from low-resolution drone videos using the YOLOv4-Tiny

model. Firstly, we compared the performance of two optimizer techniques, Adam and SGDM, in training the model. While previous studies have explored these optimizers in various domains, their application specifically in the context of tiny car detection has been limited. By conducting a comprehensive comparison, we provide insights into the impact of optimizer choice on the model's performance, helping researchers and practitioners make informed decisions.

4 Conclusions

In this study, the comparison of optimizer techniques revealed that Adam outperformed Sgdm with a slightly higher average precision for car detection under the specified conditions. In summary, data augmentation techniques in car detection using YOLOv4-Tiny are based on the principle used, By leveraging data augmentation, we enhance the model's ability to accurately detect cars in various conditions and improve its overall performance. For data augmentation techniques, when combined with Adam optimizer, further improved the performance, resulting in the highest average precision.

The results suggest that incorporating data augmentation techniques, such as color jitter augmentation, horizontal flip, and scaling, can enhance the YOLOv4-Tiny model's ability to detect cars in drone videos. The research demonstrates the importance of optimizer selection and the beneficial impact of data augmentation in improving the model's performance and robustness. Finally the proposed YOLOv4-Tiny model, coupled with Adam optimizer and data augmentation, has promising potential for real-world applications in car detection from drone videos. Therefore the choice of optimizer does matter in training the Car Detector, and both Adam and Sgdm have their strengths and weaknesses. The selection should be based on empirical evaluation, considering factors such as dataset characteristics and optimization requirements, to determine which optimizer performs better for your specific car detection task. The YOLOv4-Tiny model was trained using a deep learning framework. The training process involved dataset preparation, model initialization, optimization with either SGDM or ADAM optimizer, and evaluation using various metrics such as average precision, precision-recall curves, and IoU scores. Concerning the processing for data augmentation techniques contribute to better performance by increasing the amount and diversity of training data. This helps the model capture a wider range of car variations and better generalize to unseen scenarios.

In the future optimizer technique, the choice of the other optimizer may have yielded better results due to its adaptive learning rate, which allowed the model to

effectively navigate the optimization landscape and converge faster. Data augmentation techniques used in object detection have many techniques some commonly used data augmentation techniques for object detection such as image flipping, random cropping, color distortion and brightness adjustments, and noise injection can do in the future to improve results in research. Overall, our work contributes to the existing literature by offering a comparative analysis of optimizer techniques, exploring the effectiveness of data augmentation for tiny car detection, and evaluating the performance of the YOLOv4-Tiny model in the context of low-resolution drone videos. We believe that these findings will pave the way for future advancements in car detection algorithms and provide valuable guidance for researchers and practitioners working in the field.

5 References

- 1) Journal article
- Abbas, Q., Ibrahim, M., Jaffar, M., 2018. Video scene analysis: an overview and challenges on deep learning algorithms. *Multimedia Tools and Applications*. 77(16), 20415-20453.
- ATİK, Muhammed E., DURAN, Z., ÖZGÜNLÜK, R., 2022. Comparison of YOLO Versions for Object Detection from Aerial Images. *International Journal of Environment and Geoinformatics*. 9(2), 87-93.
- Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., Gara-Benítez, P., 2021. On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*. 13(1), 1-23.
- Dollár, P., Wojek, C., Schiele, B., Perona, P., 2012. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 34(4), 743-761.
- Dollár, P., Wojek, C., Schiele, B., Perona, P., 2012. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 34(4), 743-761.
- Ju, M., Niu, B., Niu, B., Jin, S., Liu, Z., 2023. SuperDet: An Efficient Single-Shot Network for Vehicle Detection in Remote Sensing Images. *Electronics*. 12(6), 1312.
- Li, Z., Wang, Y., Zhang, N., Zhang, Y., Zhao, Z., Zhao, Z., Xu, D., Ben, G., Gao, Y., 2022. Deep Learning-Based Object Detection Techniques for Remote Sensing Images: A Survey. *Remote Sensing*. 14(10).
- Liang, H., Seo, S., 2022. Automatic Detection of Construction Workers' Helmet Wear Based on Lightweight Deep Learning. *Applied Sciences (Switzerland)*. 12(20).
- Liu, Q., Fan, X., Xi, Z., Yin, Z., Yang, Z., 2022. Object detection based on Yolov4-Tiny and

- 1 Improved Bidirectional feature pyramid network.
- 2 *Journal of Physics: Conference Series*.
- 3 Lu, S., Wang, B., Wang, H., Chen, L., Linjian, M.,
- 4 Zhang, X., 2019. A real-time object detection
- 5 algorithm for video. *Computers and Electrical*
- 6 *Engineering*. 77, 398-408.
- 7 Pal, K., Pramanik, A., Maiti, J., Mitra, P., 2021. Deep
- 8 learning in multi-object detection and tracking:
- 9 state of the art. *Applied Intelligence*. 51(9), 6400-
- 10 6429.
- 11 Ramachandran, A., Sangaiah, A., 2021. A review on
- 12 object detection in unmanned aerial vehicle
- 13 surveillance. *International Journal of Cognitive*
- 14 *Computing in Engineering*. 2, 215-228.
- 15 Ren, Y., Zhu, C., Xiao, S., 2018. Small object
- 16 detection in optical remote sensing images via
- 17 modified Faster R-CNN. *Applied Sciences*
- 18 *(Switzerland)*. 8(5).
- 19 Salazar González, J., Zaccaro, C., Álvarez-García, J.,
- 20 Soria Morillo, L., Soria Morillo, L., 2020. Real-
- 21 time gun detection in CCTV: An open problem.
- 22 *Neural Networks*. 132, 297-308.
- 23 Sergio, S., Abdussalam, E., Qinghe, Z., 2022.
- 24 Developing a real-time social distancing detection
- 25 system based on YOLOv4-tiny and bird-eye view
- 26 for COVID-19. *Journal of Real-Time Image*
- 27 *Processing*. 19, 551-563.
- 28 Tan, L., Huangfu, T., Wu, L., Chen, W., 2021
- 29 Comparison of RetinaNet, SSD, and YOLO v3 for
- 30 real-time pill identification. *BMC Medical*
- 31 *Informatics and Decision Making*. 21(1).
- 32 Wang, Y., Xiao, Z., Cao, G., 2022. A convolutional
- 33 neural network method based on Adam optimizer
- 34 with power-exponential learning rate for bearing
- 35 fault diagnosis. *Journal of Vibroengineering*.
- 36 24(4), 666-678.
- 37 Wang, W., Hong, W., Wang, F., Yu, J., 2020. GAN-
- 38 Knowledge Distillation for One-Stage Object
- 39 Detection. *IEEE Access*. 8, 60719-60727.
- 40 Xu, Y., Xu, Y., Yang, Y., Wu, F., 2021. Training
- 41 Robust Object Detectors from Noisy Category
- 42 Labels and Imprecise Bounding Boxes. *IEEE*
- 43 *Transactions on Image Processing*. 30, 5782-
- 44 5792.
- 45 Zhang, M., Pang, K., Gao, C., Xin, M., 2020. Multi-
- 46 Scale Aerial Target Detection Based on Densely
- 47 Connected Inception ResNet. *IEEE Access*. 8,
- 48 84867-84878.
- 49 Zhao, T., Zhao, T., 2003. Car detection in low
- 50 resolution aerial images. *Image and Vision*
- 51 *Computing*. 21(8), 693-703.
- 52 Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J., 2019.
- 53 Object Detection in 20 Years: A Survey.
- 54 *Proceedings of the IEEE*. 30, 257-276.
- 55
- 56 2) Conference proceedings
- 57 Gavrilova, M., Chang, J., Thalmann N, M., Hitzer, E.,
- 58 Ishikawa, H., 2019. Advances in Computer
- 59 Graphics. Object Perception in the RGB Image.
- 60 36th Computer Graphics International Conference,
- 61 Canada.
- 62 Mazhar Abbas, S., Mazhar Abbas, S., 2018. Region-
- 63 based Object Detection and Classification using
- 64 Faster R-CNN. International Conference on
- 65 "Computational Intelligence and Communication
- 66 Technology".
- 67 Pranav, A., Pratibha, R., 2020. YOLO v3-Tiny:
- 68 Object Detection and Recognition using one stage
- 69 improved model. 6th International Conference on
- 70 Advanced Computing and Communication
- 71 Systems. 6-7 March 2020, Coimbatore, India.
- 72 Redmon, J., Farhadi, A., 2017. YOLO9000: better,
- 73 faster, stronger. IEEE conference on computer
- 74 vision and pattern recognition. 21-26 July 2017,
- 75 Honolulu, HI, USA.
- 76 Redmon, J., Divvala, S., Girshick, R., Farhadi, A.,
- 77 2016, You only look once: Unified, real-time
- 78 object detection. IEEE conference on computer
- 79 vision and pattern recognition. 27-30 June 2016,
- 80 Las Vegas, NV, USA.
- 81 Zhiqiang, W., Jun, L., 2017. A Review of Object
- 82 Detection Based on Convolutional Neural
- 83 Network. 36th Chinese Control Conference. 26-28
- 84 July 2017, Dalian, China.

