

# การเปรียบเทียบโมเดลตรวจจับวัตถุด้วยโครงข่ายประสาทเทียมแบบคอนโวลูชันในงานภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ

กิตตากร วิริยะศาสตร์<sup>1,2\*</sup> วรากร เลื่องลือวุฒิ<sup>1</sup> ปิยะรส มาลีเจริญ<sup>1</sup> สิริภาพ สันติรัตนรงค์<sup>1</sup>  
วิชัย แผ้วเกษม<sup>1</sup> พันธุ์เทพ แก้วมงคล<sup>1</sup> สัญญา มิตรเอม<sup>2</sup> และ พันศักดิ์ เทียนวิบูลย์<sup>3</sup>

วันที่รับ 19 มีนาคม 2567 วันที่แก้ไข 19 เมษายน 2567 วันที่ตอบรับ 25 เมษายน 2567

## บทคัดย่อ

ในบทความวิจัยนี้ได้ทำการศึกษาการเปรียบเทียบโมเดลตรวจจับวัตถุด้วยโครงข่ายประสาทเทียมที่ใช้ในงานภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ (Unmanned Aerial Vehicle: UAV) โดยได้ทำการตรวจจับวัตถุสองชนิด คือ สิ่งก่อสร้างและยานพาหนะ ทั้งนี้ อาศัยโมเดลการเรียนรู้ของเครื่อง (Machine Learning) ในการตรวจจับวัตถุ (Object Detection) โดยใช้โมเดลต่าง ๆ เพื่อหาว่ามีข้อดีข้อเสียแตกต่างกันอย่างไร ผ่านโมเดลที่ใช้เปรียบเทียบดังนี้ Faster R-CNN, MobileNetv1, Retinanet50, YOLOv4, YOLOv4-tiny, YOLOv7, EfficientDet ซึ่งจากการทดลองครั้งนี้ พบว่า YOLOv7 มีความแม่นยำในการตรวจจับ 58.5% ซึ่งมากกว่า MobileNetv1, YOLOv4, Faster R-CNN, YOLOv4-tiny, EfficientDet และ Retinanet50 ที่ 49.5%, 45.1%, 21.2%, 17.6%, 14.5%, 1.2% ตามลำดับ โมเดลที่มีความเร็วสูงสุด คือ MobileNetv1 มีความเร็วถึง 196.01 เฟรมต่อวินาที ซึ่งเป็นความแม่นยำและความเร็วที่เพียงพอต่องานตรวจจับวัตถุในงานภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ

**คำสำคัญ :** การเรียนรู้ของเครื่อง, โครงข่ายประสาทเทียม, การตรวจจับวัตถุ, คอมพิวเตอร์วิทัศน์, ภาพถ่ายทางอากาศ

<sup>1</sup> ฝ่ายวิจัยและพัฒนา, สถาบันเทคโนโลยีป้องกันประเทศ

<sup>2</sup> ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยธรรมศาสตร์

<sup>3</sup> ภาควิชาวิศวกรรมไฟฟ้า, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเกษตรศาสตร์

\* ผู้แต่ง, อีเมล: kittakorn.v@dti.or.th

# Comparison of Object Detection Models using Convolutional Neural Networks in Aerial Image from Unmanned Aerial Vehicles

Kittakorn Viriyasatr<sup>1,2\*</sup> Warakorn luangluewut<sup>1</sup> Piyarose Maleecharoen<sup>1</sup> Siraphob Santironnarong<sup>1</sup>  
Wichai Pawgasame<sup>1</sup> Pantape Kaewmongkol<sup>1</sup> Sanya Mitaim<sup>2</sup> and Phunsak Thiennviboon<sup>3</sup>

Received 19 March 2024, Revised 19 April 2024, Accepted 25 April 2024

## Abstract

This research article studies and compares various models used for object detection in aerial imagery captured by Unmanned Aerial Vehicle (UAV). Two types of objects are detected: buildings and vehicles. Machine learning models are used for object detection, and various models are compared to identify their advantages and disadvantages. The following models are compared: Faster R-CNN, MobileNetv1, Retinanet50, YOLOV4, YOLOV4-tiny, YOLOv7, and EfficientDet. The experiments found that YOLOV7 achieved the highest detection accuracy of 58.5%, outperforming MobileNetv1, YOLOV4, Faster R-CNN, YOLOV-tiny, EfficientDet, and Retinanet50, which achieved accuracies of 49.5%, 45.1%, 21.2%, 17.6%, 14.5%, and 1.2%, respectively. The model with the highest speed was MobileNetv1, which achieved a speed of 196.01 frames per second. This accuracy and speed are sufficient for object detection tasks in aerial image from Unmanned Aerial Vehicle.

**Keywords :** Machine learning, Neural networks, Object detection, Computer vision, Aerial image

---

<sup>1</sup> Research & Development Department, Defence Technology Institute

<sup>2</sup> Department of Electrical and Computer Engineering, Faculty of Engineering, Thammasat University

<sup>3</sup> Department of Electrical Engineering, Faculty of Engineering, Kasetsart University

\* Corresponding author, E-mail : kittakorn.v@dti.or.th

## 1. บทนำ

อากาศยานไร้คนขับ (Unmanned Aerial Vehicle: UAV) สามารถนำมาใช้งานได้หลากหลายในการเฝ้าระวังและสำรวจพื้นที่โดยค้นหาสถานที่ขนาดใหญ่และการเข้าถึงพื้นที่ที่เข้าถึงได้ยากอย่างละเอียด ตัวอย่างเช่น สนามทดสอบระยะจะต้องใช้เวลาโดยประมาณ 1-2 สัปดาห์ในการสำรวจอย่างละเอียดด้วยการเดินเท้า ความยืดหยุ่นของอากาศยานไร้คนขับช่วยให้การใช้งานรวดเร็วและคุ้มค่า นอกจากนี้ การสำรวจด้วยการเดินเท้ายังค่อนข้างอันตรายในสนามทดสอบระยะ เนื่องจากมีเศษซากวัตถุระเบิดที่ยังไม่ปลอดภัยจากจรวด ดังนั้นอากาศยานไร้คนขับจะช่วยลดความเสี่ยงต่อชีวิตมนุษย์ เนื่องจากสามารถเข้าสู่สนามได้โดยไม่ต้องเสี่ยงต่อทีมงาน ในแง่ของการรวบรวมข้อมูลอากาศยานไร้คนขับที่ติดตั้งกล้องความละเอียดสูงสามารถรวบรวมภาพจำนวนมากได้แบบตามเวลาจริง (Real-time) ช่วยในการตัดสินใจและวิเคราะห์ภาพทางอากาศที่ได้จากอากาศยานไร้คนขับที่บินเหนือสนามทดสอบระยะ ให้ภาพรวมของพื้นที่ สามารถใช้ภาพเหล่านี้เพื่อพิจารณาว่าพื้นที่นั้นปลอดภัยและพร้อมสำหรับการทดสอบระยะพื้นที่ได้รับการพิสูจน์ว่าปลอดภัยเมื่อไม่มีสัญญาณของสิ่งก่อสร้างหรือที่พักอาศัยและยานพาหนะในภาพถ่ายทางอากาศ

วิธีนี้สามารถวิเคราะห์ได้ด้วยมนุษย์แต่ต้องใช้เวลามากในการวิเคราะห์พื้นที่ทั้งหมดอย่างละเอียด คอมพิวเตอร์วิทัศน์ที่เรียกว่า การตรวจจับวัตถุ (Object Detection) ควรใช้เพื่อระบุตำแหน่งวัตถุเป้าหมายในภาพโดยอัตโนมัติ ซึ่งความก้าวหน้าของปัญญาประดิษฐ์ (AI) เฉพาะ Convolutional Neural Networks (CNN) [1] ช่วยยกระดับ

ประสิทธิภาพของงานตรวจจับวัตถุอย่างต่อเนื่องตลอดหลายปีเมื่อเปรียบเทียบกับวิธีการตรวจจับวัตถุแบบเดิมที่ใช้คุณสมบัติที่ออกแบบเองโดยใช้มือและสายตาของมนุษย์ เช่น SIFT [2], SURF [3] และ HOG [4] วิธีการตรวจจับวัตถุที่ใช้ CNN มีข้อดีหลายด้านในส่วนของประสิทธิภาพการคำนวณ ความแม่นยำ ความทนทานและความเร็ว ทำให้โมเดลที่ใช้ CNN กลายเป็นเครื่องมือที่ใช้งานได้จริงในการตรวจจับวัตถุ ซึ่งในปัจจุบันมีโมเดลตรวจจับวัตถุที่ใช้ CNN แบบใหม่หลายรุ่น ซึ่งแต่ละรุ่นมีโครงสร้างที่แตกต่างกัน

ในงานวิจัยนี้ได้ทำการศึกษาโมเดลต่าง ๆ ที่ใช้ในงานตรวจจับวัตถุในแบบต่าง ๆ เช่น R-CNN, SSD, Retinanet, YOLO, EfficientDet ว่ามีข้อดีข้อเสียแตกต่างกันอย่างไรและโมเดลใดเหมาะสมกับงานภาพถ่ายทางอากาศ (Aerial Image) มากที่สุด งานวิจัยนี้แตกต่างจากงานวิจัยอื่น ทั้งนี้ ข้อมูลภาพถ่ายทางอากาศเป็นภาพถ่ายที่ไม่จำกัดมุมมองและความสูงของภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ ซึ่งเป็นภาพถ่ายทางภูมิประเทศของไทยที่มีรูปทรงอาคารหรือยานพาหนะแตกต่างกัน

## 2. ทฤษฎีที่เกี่ยวข้อง

### 2.1 Object Detection in Aerial Images

การตรวจจับวัตถุในภาพถ่ายทางอากาศเป็นเทคนิคที่มีประโยชน์สำหรับการวิเคราะห์ภาพถ่ายทางอากาศ เนื่องจากเทคนิคนี้สามารถจำแนกและระบุตำแหน่งของวัตถุต่าง ๆ ภายในภาพได้ ข้อมูลที่ได้จากการตรวจจับวัตถุสามารถนำไปใช้จำแนกและนับจำนวนอาคาร ถนน และสิ่งก่อสร้างอื่น ๆ ภายในเขตเมือง เพื่อติดตามการขยายตัวของพื้นที่เมืองและผลกระทบต่อสิ่งแวดล้อม โดยงานวิจัย W. Pei

*et al.* [5] ได้ใช้การวิเคราะห์ภาพเชิงวัตถุเพื่อระบุการขยายตัวของเมืองและการเปลี่ยนแปลงของสภาพแวดล้อมในพื้นที่ทำเหมืองถ่านหิน Y. Liu *et al.* [6] ใช้การวิเคราะห์การเปลี่ยนแปลงของวัตถุเพื่อศึกษาผลกระทบของการขยายตัวของเหมือง

การตรวจจับวัตถุในภาพถ่ายทางอากาศยังสามารถประยุกต์ใช้ในการติดตามสัตว์ป่าในถิ่นที่อยู่อาศัยตามธรรมชาติ ช่วยให้นักวิจัยวางแผนการอนุรักษ์ได้ ดังตัวอย่างงานวิจัย [7] - [8] เทคนิคนี้ยังมีบทบาทสำคัญในการตรวจสอบป่าไม้ เนื่องจากการช่วยในการวัดลักษณะของป่าไม้ เช่น ความหนาแน่นของต้นไม้ การกระจายพันธุ์ของต้นไม้ และโครงสร้างของป่าไม้ ทั้งนี้ การระบุและนับจำนวนต้นไม้ภายในภาพถ่ายทางอากาศที่ถ่ายครอบคลุมพื้นที่ป่า [9] - [10] เทคนิคนี้ยังช่วยป้องกันไฟไหม้ [11] และการตัดไม้ทำลายป่า [12] การตรวจสอบภัยพิบัติสามารถใช้ประโยชน์จากการประเมินและระบุพื้นที่เสียหายที่ต้องการความช่วยเหลืออย่างรวดเร็ว ซึ่งได้มาจากการตรวจจับวัตถุในภาพถ่ายทางอากาศ [13] การตรวจจับวัตถุในภาพถ่ายทางอากาศต้องเผชิญกับความท้าทายที่เกี่ยวข้องกับความแปรปรวนของภาพขนาดใหญ่และมุมมองของภาพถ่ายทางอากาศ ความแม่นยำของการตรวจจับวัตถุขึ้นอยู่กับคุณภาพของภาพและประสิทธิภาพของอัลกอริทึมที่ใช้

แต่เดิมวิธีการตรวจจับวัตถุแบบดั้งเดิมอาศัยการออกแบบฟีเจอร์ด้วยมือ (Hand-crafted feature engineering) โดยฟีเจอร์ของภาพจะถูกออกแบบด้วยตนเองเพื่อแทนวัตถุในภาพ ฟีเจอร์เหล่านี้ถูกออกแบบมาเพื่อจับคุณลักษณะเฉพาะของวัตถุ เช่น รูปร่าง พื้นผิว หรือสี Scale Invariant Feature Transform (SIFT) [2] เป็นวิธีการสกัดฟีเจอร์ที่ใช้ในการอธิบายฟีเจอร์เฉพาะที่ในภาพ ฟีเจอร์ที่ SIFT อธิบายนั้นไม่แปรเปลี่ยนตามขนาดและการหมุน

รวมถึงการแปลงแบบ Affine Transformation ทำให้มีความยืดหยุ่นต่อการเปลี่ยนแปลงของมุมมองหรือแสงของวัตถุ Speeded-Up Robust Features (SURF) [3] ใช้สำหรับการตรวจจับจุดสำคัญ (Key point) ซึ่งสรุปลักษณะเฉพาะของวัตถุในภาพแบบ Local Appearance ส่วน Histogram of Oriented Gradient (HOG) [4] ใช้ฮิสโตแกรมของการไล่ระดับสี (Gradient) ในแต่ละช่องตารางของภาพ เพื่อสร้างเวกเตอร์ฟีเจอร์ที่แทนวัตถุ โดย HOG ถูกนำไปใช้กันอย่างแพร่หลายในการตรวจจับวัตถุแบบเดิมจนกระทั่งถูกแทนที่ด้วยวิธีการที่ใช้ CNN (Convolutional Neural Network) ในช่วงปีหลัง ๆ เนื่องด้วยความแม่นยำและประสิทธิภาพที่ดีกว่าแบบเดิม

วิธีการตรวจจับวัตถุแบบดั้งเดิมใช้การออกแบบฟีเจอร์ด้วยมือถูกแทนที่ด้วยวิธีการที่ใช้ CNN เนื่องจากมีข้อจำกัดด้านความแม่นยำและประสิทธิภาพในการคำนวณ วิธีการที่ใช้ CNN มีข้อดีหลายประการเหนือวิธีการแบบดั้งเดิม CNN สามารถเรียนรู้การเชื่อมโยงแบบแผนภาพ (Mapping) แบบต่อกัน (End-to-end) จากภาพที่ป้อนเข้าไปสู่ผลลัพธ์การตรวจจับวัตถุ นอกจากนี้ยังเหนือกว่าวิธีการแบบดั้งเดิมในแง่ของความแม่นยำและความเร็วในการตรวจจับวัตถุ ความแม่นยำหมายถึง ความสามารถของวิธีที่ให้การตรวจจับข้อมูลที่ถูกต้องกับสิ่งที่สนใจในภาพ ความเร็ว หมายถึง ระยะเวลาในการประมวลผลภาพยังมีค่านี้น้อยลงยิ่งดีในวิธีการนั้น ๆ

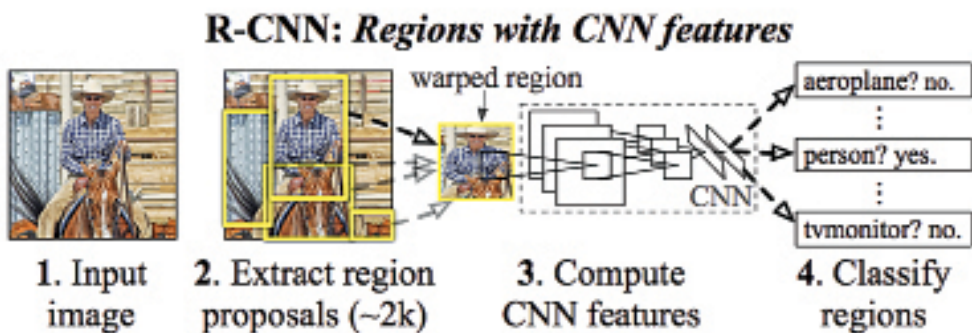
## 2.2 Regions with Convolutional Neural Network Features (R-CNN)

พัฒนาการของการตรวจจับวัตถุโดยใช้ CNN ในยุคแรกอาศัย Region Based Convolutional

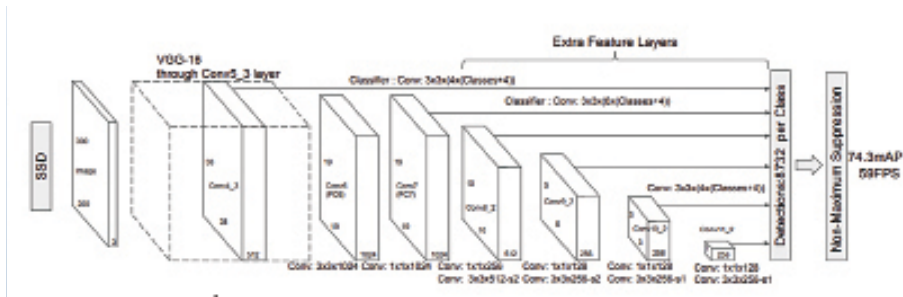
Neural Networks (R-CNN) ตามที่ R. Girshick *et al.* [14] เสนอ R-CNN เป็นวิธีการตรวจจับวัตถุแบบสองขั้นตอน ผ่านการค้นหาชุดของบริเวณที่เป็นไปได้ (Candidate regions) ก่อนที่จะจำแนกบริเวณเหล่านั้นเป็นคลาสของวัตถุที่ต่างกันและปรับแต่งกรอบ (Bounding boxes) ของวัตถุ R-CNN มีข้อเสีย คือ ใช้ทรัพยากรในการคำนวณสูง เนื่องจากต้องใช้การสกัดพีเจอร์ด้วย CNN สำหรับแต่ละบริเวณที่เป็นไปได้ มีการพัฒนาต่อยอดจาก R-CNN หลายรูปแบบเพื่อปรับปรุงประสิทธิภาพการคำนวณ จาก Fast R-CNN ซึ่งเสนอ โดย R. Girshick *et al.* [15] ช่วยปรับปรุงประสิทธิภาพการคำนวณให้ดีขึ้นกว่า R-CNN แบบเดิม ด้วยการนำ CNN เพียงตัวเดียวในการสกัดพีเจอร์จากทั้งภาพแทนที่การแยกพีเจอร์สำหรับแต่ละบริเวณที่เป็นไปได้ซึ่ง Faster R-CNN [16] พัฒนาต่อยอดจาก Fast R-CNN ด้วยการนำเสนอ Region Proposal Network (RPN) เพื่อสร้างบริเวณที่เป็นไปได้ภายในภาพ ช่วยลดความจำเป็นในการใช้ Selective Search และลดการคำนวณของโมเดล ตัวอย่างโมเดล R-CNN ดังแสดงในรูปที่ 1

### 2.3 Single-Shot MultiBox Detection (SSD)

การพัฒนาโมเดลถัดมา คือ การตรวจจับวัตถุแบบ Single-Shot MultiBox Detection (SSD) [17] ซึ่งเป็นวิธีการแบบขั้นตอนเดียว (One-Step) ที่มุ่งเน้นการตรวจจับวัตถุในภาพด้วยขั้นตอนเดียว โดย SSD ไม่ต้องใช้ขั้นตอนการค้นหาบริเวณที่เป็นไปได้ของ R-CNN เนื่องจาก SSD ใช้ CNN เพียงขั้นตอนเดียวในการทำนายทั้งความน่าจะเป็นของคลาสวัตถุและกรอบ (Bounding boxes) สำหรับชุดของกรอบยึด (Anchor box) ที่ครอบคลุมทั้งภาพผ่านกรอบยึดเหล่านี้จะถูกกำหนดขึ้นจากอัตราส่วนภาพ (Aspect ratio) โดยอัตราส่วนภาพประกอบด้วย ความกว้าง (แนวนอน) และความสูง (แนวตั้ง) ของภาพ ทั้งนี้เลขทั้งสองตัวจะถูกคูณด้วยเครื่องหมาย (:) เช่น 3:2 คือ ภาพนั้นมีความกว้าง 3 ส่วน และความสูง 2 ส่วน เป็นต้น และสเกลของวัตถุในข้อมูลชุดฝึก (Training set) ใช้เป็นจุดยึดในการทำนายกรอบของวัตถุ โดย SSD ใช้ CNN หลายชั้นที่ทำงานกับความละเอียดของภาพที่ต่างกัน (multi-resolution CNN layers) ซึ่งแต่ละชั้นสามารถดึงพีเจอร์จากภาพที่ความละเอียดต่าง ๆ โครงสร้างแบบนี้เรียกว่าปริมาตรพีเจอร์ลำดับชั้น



รูปที่ 1 ตัวอย่างโมเดล Regions with Convolutional Neural Network Features (R-CNN) [14]



รูปที่ 2 ตัวอย่างโมเดล Single-Shot MultiBox Detection (SSD) [17]

(Pyramidal feature hierarchy) อย่างไรก็ตาม ชั้น CNN ที่มีความละเอียดต่ำไม่สามารถนำพีเจอร์จากชั้นความละเอียดสูงมาใช้ซ้ำได้ ด้วยโครงสร้างแบบนี้ทำให้ SSD รุ่นแรก [17] อาจไม่สามารถตรวจจับวัตถุขนาดเล็กได้ T. - Y. Lin *et al.* [18] ได้แก้ไขพร้อมทั้งปรับปรุงด้วย Feature Pyramid Network (FPN) เพื่อปรับปรุงประสิทธิภาพในการตรวจจับวัตถุหลายขนาดของ SSD โครงสร้างของ FPN ช่วยให้สามารถรวมพีเจอร์ความละเอียดต่ำเข้ากับพีเจอร์ความละเอียดสูงได้และแก้ไขปัญหาความแปรปรวนของขนาดวัตถุในการตรวจจับได้ โดยโมเดล SSD ดังแสดงในรูปที่ 2

## 2.4 RetinaNet

โมเดลนี้ได้พัฒนามาจากปัญหาความไม่สมดุลของคลาส (Class imbalance) ในการตรวจจับวัตถุ หมายถึง สถานการณ์ที่จำนวนตัวอย่างของบางคลาสของวัตถุที่สนใจ อาจมีมากกว่าคลาสอื่น ๆ อย่างมากหรือบริเวณส่วนใหญ่ของภาพไม่มีวัตถุอยู่เลย ปัญหานี้ส่งผลต่อการตรวจจับวัตถุ เนื่องจากโมเดลอาจมีความเอนเอียงไปทางคลาสที่มีจำนวนมากและอาจตรวจจับวัตถุในคลาสที่มีจำนวนน้อยผิดพลาดได้ ซึ่งโมเดลนี้จะมาแก้ปัญหาความไม่สมดุลของคลาสโดยบทความนี้มี 2 คลาส คือ ยานพาหนะและสิ่งก่อสร้าง

สำหรับภาพถ่ายทางอากาศที่ถ่ายครอบคลุมพื้นที่กว้างมักจะพบว่ามีเพียงวัตถุประปรายและอาจมีบางภาพที่ไม่มีวัตถุเลย ดังนั้น โมเดลตรวจจับวัตถุที่ใช้กับภาพถ่ายทางอากาศเหล่านี้อาจมีประสิทธิภาพต่ำในการตรวจจับวัตถุที่น่าสนใจ ซึ่งอยู่ในกลุ่มที่มีจำนวนน้อยกว่า ตัวอย่างเช่น T. - Y. Lin *et al.* [19] จุดเด่นสำคัญของ RetinaNet คือ ฟังก์ชันการสูญเสีย (Loss function) รูปแบบใหม่ที่เรียกว่า Focal Loss for Dense Object Detection หรือเรียกสั้น ๆ คือ Focal Loss ดังสมการที่ (1) ซึ่งช่วยแก้ปัญหาความไม่สมดุลของคลาสในการตรวจจับวัตถุ ฟังก์ชัน Focal Loss กำหนดน้ำหนักให้กับแต่ละตัวอย่างใน Training สำหรับข้อมูลตามคลาสของวัตถุ คือ คลาสที่มีจำนวนมากจะได้รับน้ำหนักน้อยกว่าคลาสที่มีจำนวนน้อยกว่า เพื่อปรับฐานให้ใกล้เคียงกันและมีความสำคัญเท่า ๆ กันในแต่ละคลาส โดยตัวอย่างโมเดล RetinaNet ดังแสดงในรูปที่ 3

สมการ Focal Loss สามารถเขียนได้ดังนี้

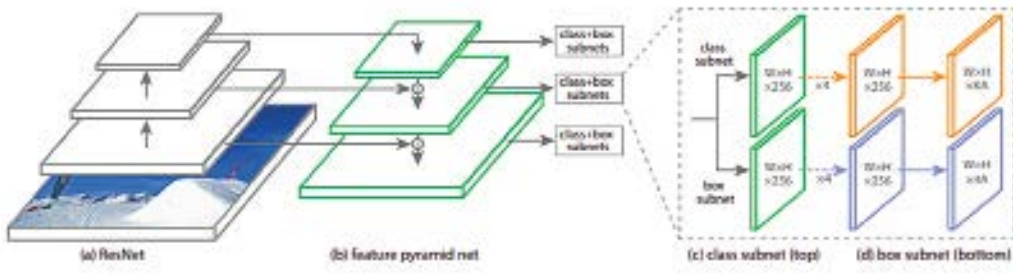
$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (1)$$

เมื่อ

$p_t$  ความน่าจะเป็นที่ทำนายของคลาสที่เป็นจริง

$\alpha_t$  ตัวปรับสมดุลเพื่อแก้ไขปัญหาความไม่สมดุลของคลาส

$\gamma$  พารามิเตอร์ในการเน้นที่ปรับการกระทำของความสูญเสีย



รูปที่ 3 ตัวอย่างโมเดล RetinaNet [19]

โดยส่วน  $-\alpha_t (1-p_t)^\gamma$  เป็นตัวควบคุม พร้อมทั้ง  $\gamma$  ปรับอัตราที่ตัวอย่างที่ง่ายหรือมีจำนวนมากในคลาสนั้น ๆ ให้ลดน้ำหนักลง เมื่อ  $\gamma$  ถูกตั้งค่าเป็น 0 Focal Loss ก็จะเทียบเท่ากับความสูญเสียของ Cross-entropy โดยมาตรฐาน Cross-entropy (Standard Cross-entropy) ดังสมการที่ (2) เป็นฟังก์ชัน Loss ที่ใช้ประเมินประสิทธิภาพของโมเดล สมการ Standard Cross-Entropy เขียนได้ดังนี้

$$CE(y, \hat{y}) = - \sum_i y_i \log(\hat{y}_i) \quad (2)$$

เมื่อ

$y$  คือ ค่าเป้าหมาย (Ground truth) ของคลาสที่ต้องการจำแนก

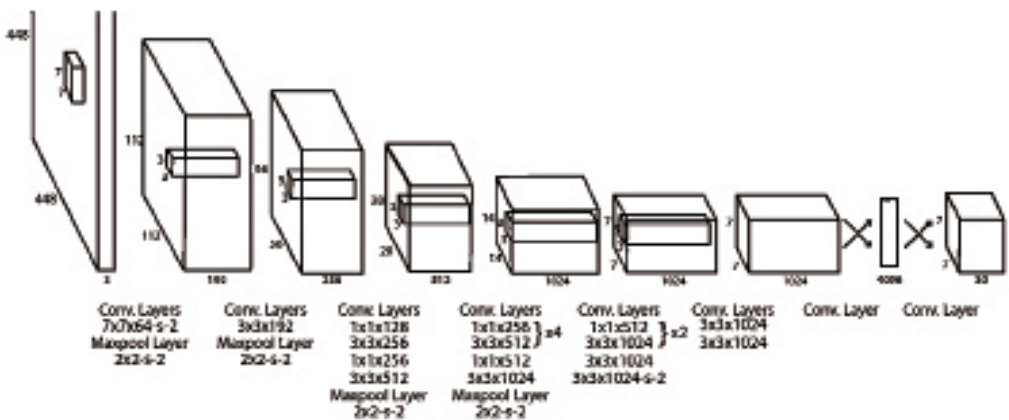
$\hat{y}$  คือ ค่าทำนาย (Prediction) ที่ได้จากโมเดล

$y_i$  และ  $\hat{y}_i$  คือ ค่าเป้าหมายและค่าทำนายของคลาสที่  $i$

โดย Cross-entropy ใช้กำหนดค่าความผิดพลาดระหว่างการทำนายและค่าเป้าหมายซึ่งมีลักษณะที่จะให้ค่าความผิดพลาดมากขึ้นเมื่อค่าทำนายแตกต่างจากค่าเป้าหมายมากขึ้นและให้ค่าความผิดพลาดน้อยลงเมื่อค่าทำนายใกล้เคียงค่าเป้าหมายมากขึ้น

## 2.5 You Only Look Once (YOLO)

การประยุกต์ใช้งานที่ต้องการการตัดสินใจแบบตามเวลาจริง (Real-time) หรือในช่วงเวลานั้น ๆ ความเร็วในการตรวจจับวัตถุเป็นสิ่งสำคัญ เช่น ในการติดตามทางอากาศ ซึ่งใช้ภาพถ่ายทางอากาศที่ถ่ายด้วยความเร็วสูง ยิ่งโมเดลตรวจจับวัตถุสามารถตรวจจับวัตถุได้เร็วเท่าใด การตัดสินใจที่ได้ก็จะยิ่งน่าเชื่อถือมากขึ้น ในงานวิจัย J. Redmon *et al.* [20] เสนอโมเดลตรวจจับวัตถุแบบ One-Shot ที่ทันสมัยเรียกว่า You Only Look Once (YOLO) ซึ่งมุ่งเน้นการแก้ไขปัญหาความเร็วในการตรวจจับ YOLO แบ่งภาพที่ป้อนเข้าเป็นตารางกริดที่มีขนาดเท่ากัน โดยใช้ CNN เพียงตัวเดียวในการทำนายความน่าจะเป็น คลาสของวัตถุและกรอบ (Bounding boxes) ของวัตถุในแต่ละกริด (Grid cell) ต่างจาก SSD โดย YOLO ใช้กรอบยึด (Anchor box) ที่กำหนดไว้ล่วงหน้า ซึ่งมีขนาดและอัตราส่วนภาพ (Aspect ratio) ที่ต่างกันเพื่อประมาณกรอบของวัตถุในภาพผ่านการกำหนดกรอบยึดไว้สำหรับแต่ละกริดตั้งแต่มีการนำเสนอ YOLO ใน [20] ได้มีการพัฒนาต่อยอดจากสถาปัตยกรรมของ YOLO ออกมาอีกหลายรูปแบบด้วย YOLOv4 [21] และ YOLOv7 [22] เป็นหนึ่งในสถาปัตยกรรมที่ได้รับความนิยมมากที่สุด นอกจากนี้ยังมี YOLOv4 และ YOLOv4-tiny เวอร์ชันที่เล็กและเร็วกว่า ใช้โครงสร้างที่ไม่ซับซ้อน ซึ่งรุ่นล่าสุดของ YOLO แสดงถึงการ



รูปที่ 4 ขั้นตอนการทำงานในการตรวจจับวัตถุของ YOLO [20]

ปรับปรุงที่สำคัญเหนือรุ่นก่อนหน้า ด้วยความแม่นยำที่สูงขึ้น ประสิทธิภาพที่ดีกว่า และมีฟีเจอร์ที่ทันสมัยมากขึ้น ตัวอย่างโมเดล YOLO ดังแสดงในรูปที่ 4

## 2.6 EfficientDet

โมเดลการตรวจจับวัตถุในภาพถ่ายทางอากาศมักจะทำงานภายใต้ข้อจำกัดของทรัพยากรในการคำนวณ เช่น ระบบฝังตัวบนอากาศยานไร้คนขับ ขนาดของโมเดลส่งผลโดยตรงต่อปริมาณทรัพยากรในการคำนวณที่จำเป็นสำหรับการดำเนินการของโมเดลที่ใช้จริง ทั้งนี้ โดยทั่วไปแล้วโมเดลขนาดใหญ่ต้องการการประมวลผล หน่วยความจำ และพื้นที่จัดเก็บข้อมูลในจำนวนที่มากกว่าซึ่งอาจทำให้ไม่สามารถใช้งานบนระบบที่มีทรัพยากรจำกัดได้ ตัวอย่างเช่น เมื่อทำการติดตั้งที่อากาศยานไร้คนขับ ต้องทำให้ตัวอากาศยานไร้คนขับมีน้ำหนักเบาและใช้ตัวประมวลผลที่มีขนาดเล็กและขนาดตัวความจุขนาดเล็ก รวมถึงความร้อนจากในงานวิจัย M. Tan *et al.* [23] เสนอ EfficientDet ซึ่งมุ่งเน้นการแก้ไขประสิทธิภาพในการคำนวณของโมเดลตรวจจับวัตถุ EfficientDet ใช้แนวทางการปรับขนาดแบบผสม (Compound Scaling Approach) ตามที่ [24] เสนอไว้ในการปรับขนาด

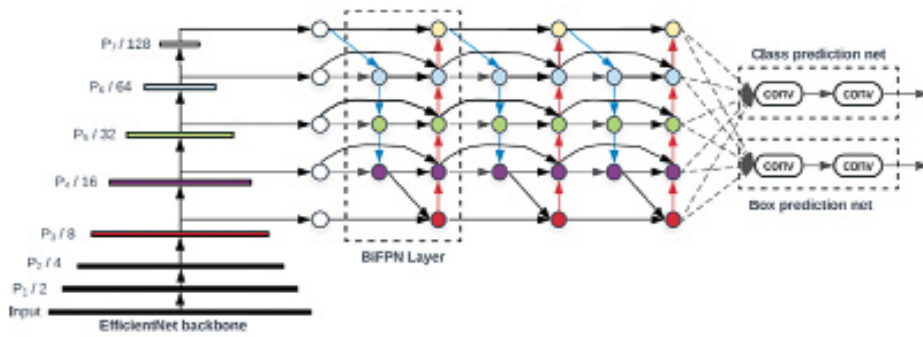
ของโครงสร้างเครือข่าย ความละเอียดของภาพที่ป้อนเข้า และขนาดของชุดข้อมูล (Batch Size) อย่างพร้อมกัน เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดที่สุทธระหว่างความแม่นยำและประสิทธิภาพ ตัวอย่างโมเดล EfficientDet ดังแสดงในรูปที่ 5

## 2.7 การตรวจจับวัตถุในภาพถ่ายทางอากาศด้วยเทคนิคโครงข่ายประสาทเทียม (Detecting Objects in Aerial photographs using Neural Network Techniques) [25]

ในงานวิจัยนี้ได้กล่าวถึงการนำโมเดลของ Objects Detection แต่ละโมเดลประกอบไปด้วย YOLO, RetinaNet และ Fast R-CNN ในการตรวจจับวัตถุ 2 อย่าง คือ สิ่งก่อสร้างและยานพาหนะ ซึ่งความน่าสนใจของงานวิจัยนี้ คือ โมเดล YOLO ที่เหมาะต่อกล้องของอากาศยานไร้คนขับที่มีความเร็วในการทำงานอยู่ที่ 25 เฟรมต่อวินาที โดย YOLO ในงานวิจัยที่กล่าวมานี้ให้ความเร็วและความแม่นยำในการตรวจจับวัตถุที่น่าสนใจของภาพขนาด 5472x3648 pixels ดังแสดงในรูปที่ 6

ทั้งนี้ เทคโนโลยีที่เกี่ยวข้องในแต่ละโมเดลใช้ฟีเจอร์ที่แตกต่างกันสรุปได้ดังตารางที่ 1





รูปที่ 5 ขั้นตอนการทำงานในการ ตรวจจับวัตถุของ EfficientDet [23]



รูปที่ 6 ภาพถ่ายทางอากาศมุมสูงโดยอากาศยานไร้คนขับขนาด 5472x3648 pixels (ภาพต้นฉบับ)



รูปที่ 7 Image With Dense Class Instances



รูปที่ 8 Image With Sparse Class Instances

ตารางที่ 1 แสดงการใช้ฟีเจอร์ที่แตกต่างกันในโมเดลต่าง ๆ

Model	Feature Extractor	Feature Detection	Feature Matching	Feature Descriptor
ResNet50	โครงข่าย CNN ที่มีความลึก (50 ชั้น)	ใช้ในการจำแนกภาพมากกว่าการตรวจจับวัตถุ	ไม่มีใน ResNet50 โดยตรง	คุณลักษณะภาพที่ได้ใช้ในการจำแนกภาพ
EfficientDet	EfficientNet	BiFPN (Bi-directional Feature Pyramid Network)	Non-maximum suppression (NMS)	ใช้คุณลักษณะจาก feature maps
YOLOv4	CSPDarknet53	แบ่งภาพเป็นกริดและตรวจจับวัตถุในแต่ละกริด	Non-maximum suppression (NMS)	ใช้คุณลักษณะจาก feature maps
YOLOv4-tiny	โครงข่ายที่เบากว่า YOLOv4	แบ่งภาพเป็นกริดและตรวจจับวัตถุในแต่ละกริด	Non-maximum suppression (NMS)	ใช้คุณลักษณะจาก feature maps
Faster R-CNN	ResNet50 หรือ VGG16	Region Proposal Network (RPN)	Non-maximum suppression (NMS)	ใช้คุณลักษณะจาก feature maps
MobileNetv1	โครงข่าย CNN ที่เบาและมีประสิทธิภาพ	ใช้ในการจำแนกภาพมากกว่าการตรวจจับวัตถุ	ไม่มีใน MobileNetv1 โดยตรง	คุณลักษณะภาพที่ได้ใช้ในการจำแนกภาพ
YOLOv7	YOLOv7-backbone	แบ่งภาพเป็นกริดและตรวจจับวัตถุในแต่ละกริด	Non-maximum suppression (NMS)	ใช้คุณลักษณะจาก feature maps

### 3. วิธีการดำเนินการ

#### 3.1 วิธีการเทรนโมเดล

ข้อมูลที่ใช้เป็นภาพถ่ายทางอากาศที่ไม่จำกัดมุมมองสูงและองศาจากอากาศยานไร้คนขับขนาด 5472x3648 pixels (ภาพต้นฉบับ) อุปกรณ์ติดตั้งกล้องถ่ายภาพกลางวัน/กลางคืน (EO/IR Payload Camera System) รัศมีปฏิบัติการ 100 - 200 กิโลเมตร เพดานบินสูงสุด 10,000 ฟุต บนอากาศยานไร้คนขับ ซึ่งเป็นเพียงการนำภาพมาใช้ในการวิจัยเท่านั้นและนำมาทำ Objects labeling ผ่านการแบ่งคลาสออกเป็น 2 คลาส คือ สิ่งก่อสร้างและยานพาหนะ ซึ่งมีภาพทั้งหมด 730 ภาพ มีรูปภาพที่มีลักษณะแตกต่างกันดังนี้

1. กระจายแบบหนาแน่น มี 382 ภาพ ในแต่ละภาพจะมีวัตถุมากกว่า 10 วัตถุ ดังรูปที่ 7 (Image with Dense Class Instances) ซึ่งเป็นรูปที่มีคลาสที่สนใจอยู่มากกว่า 10 วัตถุในภาพ 1 ภาพ ตัวอย่าง เช่น จำนวนบ้านหรืออาคาร และจำนวนรถในภาพ 1 ภาพ นั่นคือ รูปภาพที่มีการปรากฏ

ของคลาสหรือวัตถุที่เราสนใจมีจำนวนมากในภาพนั้น ๆ

2. กระจายห่างกัน มี 294 ภาพ แต่ละภาพมีวัตถุน้อยกว่า 10 วัตถุ เช่น ในรูปที่ 8 (Image with Sparse Class Instances) ซึ่งเป็นตัวอย่างของรูปที่มีคลาสที่สนใจอยู่น้อยกว่า 10 วัตถุ ในภาพ 1 ภาพ เช่น จำนวนบ้าน หรืออาคาร หรือจำนวนรถในภาพ 1 ภาพ นั่นคือ รูปภาพที่มีการปรากฏของคลาสหรือวัตถุที่เราสนใจมีจำนวนน้อยในภาพนั้น ๆ

3. ไม่มีวัตถุที่สนใจตามคลาส มี 54 ภาพที่ไม่มีรถ หรือบ้าน หรืออาคารในภาพ นั่นคือรูปภาพที่ไม่มีมีการปรากฏของคลาสหรือวัตถุที่เราสนใจในภาพนั้น ๆ

สำหรับแต่ละคลาส ข้อมูลเหล่านี้แสดงถึงความไม่สมดุลระหว่างจำนวนของวัตถุที่ป้ายชื่อว่า 'ยานพาหนะ' และจำนวนของวัตถุที่ป้ายชื่อว่า 'อาคาร' ข้อมูลภาพทางอากาศมีมุมมองที่แตกต่างกันในแต่ละภาพ แต่มีขนาดที่เปลี่ยนแปลงไม่ได้เสมอ โมเดลการตรวจจับวัตถุที่ดีควรสามารถใช้งานได้ทั้งในมุมมองที่

แตกต่างกันและขนาดที่ต่างกัน โดยเพื่อแก้ไขปัญหานี้ ภาพทางอากาศขนาดเดิม 5472x3648 pixels ถูกเปลี่ยนขนาดใหม่เป็นขนาดที่ต่างกัน เช่น 4104x2736, 2736x1824 และ 1368x912 pixels

ซึ่งเครื่องมือที่ใช้ในการติดป้ายกำกับ (Label) คือ Computer Vision Annotation Tool (CVAT) สำหรับใช้ในการติกรอบของวัตถุที่สนใจในภาพ โดยทำการติกรอบตามรูปแบบของ YOLO หรือ PASCAL VOC2007 เพื่อใช้ Training จากนั้นรูปภาพทั้งหมดที่มีขนาดแตกต่างกันถูกผสมเข้าด้วยกันเป็นชุดข้อมูลเดียวกัน ภาพที่ไม่มีวัตถุจะถูกลบทิ้ง ดังนั้น ชุดข้อมูลหลายระดับจึงเหลืออยู่ 2,704 ภาพ ในการทดลองนี้ได้ Training Model บนชุดข้อมูล 2 ชุด ที่แตกต่างกัน โดยขั้นตอนทั้งหมดเป็นการเทรนโมเดลใหม่ตั้งแต่ต้น ดังนี้

- **ชุดข้อมูลที่ 1 No scaling datasets**

ถูกสร้างขึ้นโดยการครอบรูปภาพแต่ละรูปในชุดข้อมูลภาพทางอากาศขนาดเดิม 5472x3648 pixels (ทั้งนี้ไม่รวมภาพที่ไม่มีวัตถุ) ซึ่งใช้การครอบขนาด 912x912 pixels และเลือกตำแหน่งการครอบแบบสุ่ม (Randomly Crop) และหมุนภาพ ที่สุ่มองศาระหว่าง 0-90 องศา (Rotate Cropped Images) ข้อมูลนี้เรียกว่า "No scaling" มีภาพทั้งหมด 90,364 ภาพ ในชุดข้อมูล เหตุผลที่ใช้ขนาด 912x912 pixels เพื่อแบ่งภาพให้มีขนาดที่เหมาะสมในการเข้าโมเดล เพราะโมเดลไม่สามารถรับภาพขนาด 5472x3648 pixels ได้ จึงทำการปรับสเกลในชุดข้อมูลที่ 2

- **ชุดข้อมูลที่ 2 Multi-scaling datasets**

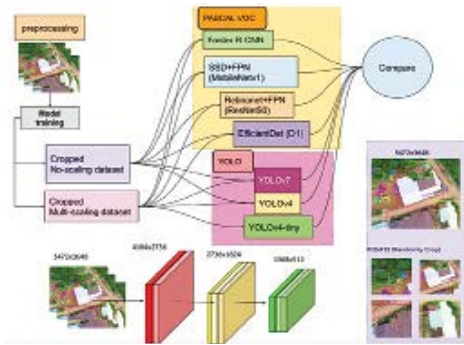
ถูกสร้างขึ้นด้วยกระบวนการเดียวกัน แต่บนชุดข้อมูลที่มีการเปลี่ยนขนาดหลายระดับ

(Rescale) ชุดข้อมูลนี้เรียกว่า "Multi-scaling" และมีภาพทั้งหมด 93,689 ภาพ

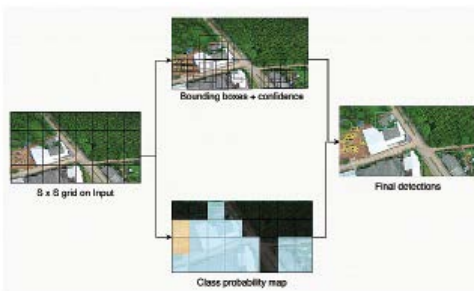
ผ่านการนำชุดข้อมูลที่ 1 และ 2 มาแบ่งเป็นส่วนสำหรับการฝึกโมเดล 80% ส่วนสำหรับการตรวจสอบความถูกต้อง 10% และส่วนสำหรับการทดสอบ 10% หลังจากนั้นในการเทรนโมเดลใช้โมเดลทั้งหมด 7 โมเดล ดังนี้ R-CNN, SSD, Retinanet, YOLOv4, YOLOv4tiny, YOLOv7 และ EfficientDet ข้อมูลภาพที่ติกรอบนั้นจะแยกเป็น 2 ประเภท คือ

1. ข้อมูลรูปแบบของ YOLO แต่ละภาพจะมีไฟล์ข้อความป้ายชื่อที่เชื่อมโยงอยู่ในไฟล์ข้อความป้ายชื่อแต่ละบรรทัดแสดงป้ายชื่อวัตถุในภาพ ป้ายชื่อวัตถุเป็นสตริงของค่าที่คั่นด้วยช่องว่างที่ระบุหมายเลขการระบุคลาสของวัตถุ พิกัดศูนย์กลางในแกนอนและตั้งของกล่องสำหรับโมเดล YOLOv4, YOLOv4-tiny และ YOLOv7

2. รูปแบบชื่อของ PASCAL VOC2007 แต่ละภาพจะมีเอกสาร XML ที่เชื่อมโยงโดเมน XML กำหนดแต่ละวัตถุในภาพด้วยแท็ก '<object>' แท็ก '<name>' ภายในแท็ก '<object>' ระบุชื่อของคลาสวัตถุ แท็ก '<bndbox>' ระบุพิกัดมุมบนซ้ายและมุมล่างขวาของกล่องคำ โดยไม่มีการปรับให้มีมาตรฐาน อาจใช้รูปแบบแท็กอย่างไรก็ได้ ใช้สำหรับโมเดล Faster R-CNN, MobileNetv1, ResNet50 และ EfficientDet (D1) แสดงในรูปแบบที่ 9 ขั้นตอน Training model จากนั้นนำ Weight ที่ได้จากโมเดลแต่ละโมเดลไปทำการทดสอบโมเดลว่าได้ผลลัพธ์อย่างไรและนำมาวิเคราะห์เปรียบเทียบกับว่าโมเดลทั้ง 7 โมเดล แตกต่างกันอย่างใดและเหมาะสมกับงานประเภทใด ดังแสดงในรูปแบบที่ 10



รูปที่ 9 ขั้นตอน Training model



รูปที่ 10 ตัวอย่างการทำงานที่ได้  
จากโมเดล YOLOv4 ขนาด 5472x3648 pixels

### 3.2 วิธีการทดสอบโมเดล

บทความนี้ประเมินประสิทธิภาพของโมเดลตรวจจับวัตถุ (Objects Detection) โดยอาศัยเกณฑ์การวัด 3 ประเภท ได้แก่

#### 3.2.1 ความแม่นยำ (Accuracy)

คือการวัดประสิทธิภาพของโมเดลในการระบุและกำหนดตำแหน่งของวัตถุในภาพได้อย่างถูกต้อง ซึ่งใช้กรอบสี่เหลี่ยม (Bounding Box) ประเมินผล ค่าความแม่นยำที่นิยมใช้ คือ ค่า mAP (Mean Average Precision) เป็นการรวมกันระหว่าง Precision ดังสมการที่ (3) วัดสัดส่วนของการตรวจจับวัตถุได้ถูกต้อง (True Positive) จากการตรวจจับทั้งหมด และ Recall ดังสมการที่ (4) วัดสัดส่วนของการตรวจจับวัตถุได้ถูกต้อง (True Positive) จากจำนวนวัตถุทั้งหมดที่มีจริงในภาพ การตรวจจับ

ถือว่าถูกต้อง เมื่อโมเดลระบุประเภทวัตถุได้ถูกต้อง มีคะแนนความมั่นใจ (Confidence Score) สูงกว่าเกณฑ์ที่กำหนด และมีค่า IoU (Intersection over Union) ระหว่างกรอบที่โมเดลระบุกับกรอบที่กำหนดไว้สำหรับวัตถุนั้น ๆ สูงกว่าเกณฑ์ที่กำหนด โดย Precision มีค่าระหว่าง 0 ถึง 1 ทั้งนี้ ค่ายิ่งมากยิ่งขึ้นดี เมื่อค่า Precision เข้าใกล้หรือเป็น 1 แสดงว่าระบบมีความแม่นยำมากในการระบุ Positive class หากมีน้อยมากของการทำนายผิดพลาดว่าเป็น Positive class แต่ถ้า Precision มีค่าเข้าใกล้หรือเป็น 0 แสดงว่า ระบบมีการทำนาย Positive class ผิดพลาดมาก ๆ ซึ่งมีการทำนายว่าเป็น Positive class มากเกินไปในที่จริง ๆ และมีความคลาดเคลื่อนมากในการระบุ Positive class ที่ถูกต้อง

สมการ Precision เขียนได้ดังนี้

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

เมื่อ

True Positives (TP) = จำนวนข้อมูลที่ถูกต้องที่ทำนายว่าเป็น Positive class

False Positives (FP) = ข้อมูลที่ถูกต้องที่ถูกทำนายว่าเป็น Positive class แต่ควรจะเป็น Negative class

สมการ Recall เขียนได้ดังนี้

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

เมื่อ

True Positives (TP) = จำนวนข้อมูลที่ถูกต้องที่ทำนายว่าเป็น Positive class

False Negatives (FN) = ข้อมูลที่ถูกต้องที่ถูกทำนายว่าเป็น Negative class แต่ควรจะเป็น Positive class

โดย Recall มีค่าระหว่าง 0 ถึง 1 ทั้งนี้ค่ายิ่งมากยิ่งดี เมื่อค่า Recall เข้าใกล้หรือเป็น 1 แสดงว่า ระบบมีความสามารถในการระบุ Positive class ที่ถูกต้องมาก เนื่องจากระบบสามารถหาข้อมูล Positive class ทั้งหมดที่มีในข้อมูลได้เต็มที่ แต่ถ้า Recall มีค่าเข้าใกล้หรือเป็น 0 แสดงว่า ระบบมีความสามารถในการระบุ Positive class ที่ถูกต้องน้อย เนื่องจากมีการพลาดในการระบุข้อมูล Positive class ที่มีในข้อมูลไปมากทำให้ระบบมีข้อบกพร่องในการระบุ Positive class ที่มีในข้อมูล

Intersection over Union (IoU) คือ อัตราส่วนของพื้นที่ที่ทับซ้อนระหว่างกรอบที่โมเดลระบุกับกรอบที่กำหนดไว้ โดยหลักการ IOU แสดงดังรูปที่ 11

โมเดลการตรวจจับวัตถุที่ดีจะมีค่า Precision และ Recall สูง ซึ่งหมายความว่าโมเดลสามารถตรวจจับวัตถุส่วนใหญ่ได้อย่างถูกต้องและตรวจจับวัตถุได้เกือบทั้งหมดที่มีจริงในภาพ บทความนี้จึงต้องการศึกษากราฟแสดงความสัมพันธ์ระหว่าง Precision และ Recall ที่แปรผันตามค่า Confidence Score กราฟ Precision-Recall Curve แสดงความสัมพันธ์ระหว่างอัตราการตรวจจับวัตถุที่ถูกต้อง (Precision) กับสัดส่วนของวัตถุที่ตรวจจับได้ (Recall) ค่า Average Precision (AP) คือพื้นที่ใต้กราฟ Precision-Recall Curve ใช้ประเมินประสิทธิภาพโดยรวมของโมเดล โมเดลที่ดีจะมีค่า AP ใกล้เคียงกับ 1 หมายความว่าโมเดลมีความแม่นยำสูงในทุกระดับของ Recall ซึ่งทั่วไปจะคำนวณค่า AP สำหรับแต่ละประเภทของวัตถุ ค่า mAP ได้มาจากการนำค่า AP ของทุกประเภทของวัตถุมารวมกัน โดยค่า mAP อาจเปลี่ยนแปลงไปตามเกณฑ์ของค่า IoU ที่กำหนด บทความนี้ใช้เกณฑ์ IoU ตั้งแต่ 0.5 ถึง 0.95 และคำนวณ mAP ผ่านการแบ่งวัตถุดังนี้

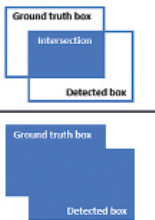
- วัตถุขนาดเล็ก หมายถึง ยานพาหนะ
- วัตถุขนาดกลาง หมายถึง บ้าน และอาคาร

ขนาดเล็ก

- วัตถุขนาดใหญ่ หมายถึง อาคารขนาดใหญ่ โรงงาน โรงพยาบาล โรงเรียน สิ่งก่อสร้างขนาดใหญ่

### 3.2.2 ความเร็ว (Speed)

วัดด้วยค่า FPS (Frames per Second)

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Intersection}}{\text{Union}}$$


รูปที่ 11 หลักการทำงานของ Intersection over Union (IOU)

หมายถึง จำนวนเฟรมภาพที่โมเดลประมวลผลได้ใน 1 วินาที ค่า FPS ยิ่งสูง ยิ่งหมายความว่าโมเดลประมวลผลภาพได้เร็ว ความเร็วของโมเดลขึ้นอยู่กับโครงสร้างของโมเดล ฮาร์ดแวร์ที่ใช้ประมวลผลและสภาพแวดล้อมของซอฟต์แวร์ บทความนี้ได้ปรับแต่งโมเดลทั้งหมดให้ทำงานบน NVIDIA Tesla V100 GPU ร่วมกับ TensorRT engine โดยแปลงค่าพารามิเตอร์ของโมเดลเป็นเลขจุดทศนิยมแบบ 16 บิต (16-bit floating-point) เพื่อให้ประมวลผลได้เร็วขึ้น แต่ส่งผลให้ความแม่นยำลดลงเล็กน้อยเมื่อเทียบกับโมเดลที่ไม่ได้ปรับแต่ง

### 3.2.3 ความซับซ้อน (Complexity)

วัดด้วยจำนวนพารามิเตอร์ของโมเดลแต่ละโมเดล การวัดความซับซ้อนเป็นสิ่งสำคัญอย่างยิ่ง โดยเฉพาะเมื่อทรัพยากรบนอุปกรณ์เป้าหมายมีจำกัด เช่น ในการตรวจจับวัตถุในภาพถ่ายทางอากาศ เราต้องคำนึงถึงการนำระบบตรวจจับวัตถุไปใช้งานบนอุปกรณ์คอมพิวเตอร์บนอากาศยานไร้คนขับ ซึ่งมักจะมีทรัพยากรการคำนวณและหน่วยความจำที่

จำกัด (Model Size) ดังนั้น โมเดลการตรวจจับวัตถุที่นำไปใช้งานต้องเป็นไปตามข้อกำหนดของอุปกรณ์เป้าหมาย เพื่อให้มีประสิทธิภาพในการทำงานได้อย่างเหมาะสม น้ำหนักเบา และการประมวลผลที่อาจมีขนาดเล็กกว่าคอมพิวเตอร์ทั่วไป

#### 4. ผลการศึกษา/ผลการดำเนินการ

ผลจากการนำ Weight ของโมเดลที่ได้จากการเทรนโมเดลไปทดสอบโมเดลทั้งหมด 7 โมเดล คือ Faster R-CNN, MobileNetv1, ResNet50, YOLOv4, YOLOv4-tiny, YOLOv7 และ EfficientDet

**ตารางที่ 2** แสดงจำนวนรวมของวัตถุสำหรับแต่ละคลาสและจำนวนเฉลี่ยของวัตถุในภาพ

Objects	Total number of objects	Avg. number of objects per image
Vehicle	3937	5.39
Building	14263	19.54

จากตารางที่ 2 แสดงจำนวนรวมของวัตถุสำหรับแต่ละคลาสและจำนวนเฉลี่ยของวัตถุในภาพสำหรับแต่ละคลาส ข้อมูลเหล่านี้แสดงถึงความไม่สมดุลระหว่างจำนวนของวัตถุที่ป้ายชื่อว่า 'ยานพาหนะ' และจำนวนของวัตถุที่ป้ายชื่อว่า 'อาคาร' ข้อมูลภาพทางอากาศมีมุมมองที่แตกต่างกันในแต่ละภาพส่งผลให้โมเดลที่ไม่มีการปรับน้ำหนักหรือให้ความสำคัญจำนวนคลาสที่มีภาพจำนวนน้อย มีประสิทธิภาพที่ลดลง เช่น โมเดล ResNet50 ในตารางที่ 3

จากตารางที่ 3 แสดงขนาดความจุของโมเดลและความเร็วของโมเดลในแต่ละตัว จะเห็นได้ว่า SSD+FPN (MobileNetv1) มีความน่าสนใจคือ มีความเร็วกว่าโมเดลอื่น ๆ ที่ได้ทำการทดลองมา ซึ่งมีความเร็วถึง 196.01 เฟรมต่อวินาที แต่ในเชิง

**ตารางที่ 3** ความเร็ว (fps) และขนาดความจุของโมเดล

Model	Evaluation Metrics	
	speed in FPS (ms)	Model Size (MB)
YOLOv4	74.89	64.00
Faster R-CNN	101.25	28.30
EfficientDet (D1)	115.06	5.29
RetinaNet+FPN (ResNet50)	152.56	50.70
YOLOv7	158.13	36.50
YOLOv4-tiny	158.30	5.88
SSD+FPN (MobileNetv1)	196.01	29.90

ขนาดความจุของโมเดลนั้น พบว่า YOLOv4-tiny มีความจุโมเดลที่ต่ำกว่า 5.09 เทา และมีความเร็วเป็นรอง MobileNetv1 อยู่ที่ 158.3 เฟรมต่อวินาทีเท่านั้น ในเรื่องของ Model Size จะได้เปรียบเนื่องจากในการประมวลผลภาพถ้าต้องการให้ประมวลผลภาพแบบ Real-time ไม่จำเป็นต้องใช้ทรัพยากรในการประมวลผลที่มากและมีขนาดใหญ่ Memory ที่มากอาจส่งผลให้ชิ้นส่วนมีขนาดใหญ่ขึ้น และการประมวลผลที่มากขึ้น ดังนั้น ในการเก็บข้อมูลและประมวลผลภาพต้องคำนึงถึงน้ำหนักในอากาศยานไร้คนขับด้วย ซึ่งความจุในที่นี้หมายถึง Capacity ที่ใช้ในหน่วยความจำ

จากตารางที่ 4 จะเห็นได้ว่า YOLOv7 มีความแม่นยำที่สูงที่สุดจากทุกโมเดลที่ได้ทำการทดลองมาอยู่ที่ 58.5% และรองลงมา คือ SSD+FPN (MobileNetv1) มีความแม่นยำที่รองลงมาจาก YOLOv7 ทั้งนี้ จากตารางที่ 3 และตารางที่ 4 จะเห็นได้ว่า SSD+FPN (MobileNetv1) เป็นโมเดลที่เกือบดีที่สุดในงานตรวจจับวัตถุจากภาพถ่ายทางอากาศทั้งเรื่องความเร็วและความแม่นยำ ถึงแม้ YOLOv7 ที่มี

**ตารางที่ 4** ความแม่นยำและขนาดความจุของโมเดล

Model	Evaluation Metrics	
	mAP (%)	Model Size (MB)
RetinaNet+FPN (ResNet50)	1.2	50.70
EfficientDet (D1)	14.5	5.29
YOLOv4-tiny	17.6	5.88
Fasster R-CNN	21.2	28.30
YOLOv4	45.1	64.00
SSD+FPN (MobileNetv1)	49.5	29.90
YOLOv7	58.5	36.50

แม่นยำสูงกว่ายังมีความเร็วที่ช้ากว่า เมื่อเทียบกับ SSD+FPN (MobileNetv1) อยู่ระดับหนึ่ง ส่วนในโมเดลตัวอื่น ๆ ความเร็วอาจจะใกล้เคียงกัน ยกเว้น YOLOv4 ที่มีความเร็วต่ำสุดอยู่เพียงแค่ 74.89 เฟรมต่อวินาที ส่วนในด้านความจุ พบว่า YOLOv4-tiny มีความจุโมเดลที่ต่ำอยู่ที่ 5.88 MB เท่านั้น แต่ในความแม่นยำยังไม่สูงเมื่อเทียบกับ SSD+FPN (MobileNetv1) และ YOLOv7 แต่ยังคงมีความแม่นยำสูงกว่า RetinaNet+FPN (ResNet50) ที่มีเพียง 1.2 % เท่านั้น ดังแสดงในรูปที่ 12 ตัวอย่างภาพถ่ายทางอากาศที่ทดสอบเสร็จสิ้น

## 5. สรุปและอภิปรายผลการทดลอง

จากการทดลองทั้งหมด 7 โมเดล พบว่าความแม่นยำสูงสุดคือ YOLOv7 อยู่ที่ 58.5% ซึ่งยังไม่ถือว่าสูงมากถึงแม้ว่าในงานวิจัยที่ผ่านมา [21] - [27] โมเดลตรวจจับวัตถุจะมีความแม่นยำถึง 70 - 80% เนื่องจากจำนวนข้อมูลของงานวิจัยนั้นมีจำนวนมากกว่าและเป็นข้อมูลที่แตกต่างจากข้อมูลในบทความนี้ โดยปัจจัยหนึ่งมาจากข้อมูลที่ใช้ในการทำการทดลองมีจำนวนน้อยกว่า หากมีชุดข้อมูลที่

ใช้ทดลองมากขึ้น ความแม่นยำจะมีค่าสูงมากขึ้น และอีกปัญหาหนึ่งในภาพถ่ายทางอากาศคือ มีการถ่ายในความสูงที่ไม่เท่ากันและมุมกล้องแตกต่างกัน ทำให้ผลความแม่นยำต่ำลง หากต้องการความแม่นยำที่สูงกว่าจำเป็นต้องมีข้อมูลภาพที่มากขึ้น และมีมุมกล้องและความสูงในระยะที่กำหนดที่ทราบผลจากการบินในระยะนั้น ๆ ถ้าหากต้องการโมเดลที่ใช้ความจุต่ำแต่ความแม่นยำสูง แต่ไม่สูงที่สุด ควรเลือก YOLOv4-tiny ที่ใช้ความจุเพียงแค่ 5.88 MB เท่านั้นและโมเดลที่ดีที่สุดที่ได้ทำการทดลองมาคือ SSD+FPN (MobileNetv1) มีความเร็วในการทำงานสูงที่สุดถึงแม้ความแม่นยำจะเป็นรอง YOLOv7 อยู่ แต่ก็ดีกว่าโมเดลอื่น ๆ ที่ทำการทดลองมาในทุก ๆ โมเดล ซึ่งจากความเร็วจะเห็นได้ว่าโมเดลทุกโมเดลมีความเร็วเกิน 25 เฟรมต่อวินาทีที่เป็นความเร็วของความเร็วกล้องของอากาศยานไร้คนขับทั้งหมด โดยโมเดลที่ได้ทำการทดลองเหมาะสมในงานภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ ซึ่งจำนวนรวมของวัตถุสำหรับแต่ละคลาสและจำนวนเฉลี่ยของวัตถุในภาพสำหรับแต่ละคลาสข้อมูลเหล่านี้แสดงถึงความไม่สมดุลระหว่างจำนวนของวัตถุที่ป้ายชื่อว่า 'ยานพาหนะ' และจำนวนของวัตถุที่ป้ายชื่อว่า 'อาคาร' ข้อมูลภาพทางอากาศมีมุมมองที่แตกต่างกันในแต่ละภาพส่งผลให้โมเดลที่ไม่มีการปรับน้ำหนักหรือให้ความสำคัญจำนวนคลาสที่มีภาพจำนวนน้อย มีประสิทธิภาพที่ลดลง ตัวอย่างเช่น โมเดล ResNet50 ที่มีประสิทธิภาพลดลง ซึ่งขนาดความจุของโมเดลและความเร็วของโมเดลในแต่ละตัวมีลักษณะแตกต่างกัน จะเห็นได้ว่า SSD+FPN (MobileNetv1) มีความน่าสนใจ คือ มีความเร็วกว่าโมเดลอื่น ๆ ที่ได้ทำการทดลองมา ซึ่งมีความเร็วถึง 196.01 เฟรมต่อวินาที แต่ในเชิงขนาดความจุของโมเดลนั้น พบว่า YOLOv4-tiny มีความ



รูปที่ 12 ตัวอย่างภาพถ่ายทางอากาศที่ทดสอบเสร็จสิ้น

จุ่มเดลที่ต่ำกว่า 5.09 เท่า และมีความเร็วเป็นรอง MobileNetv1 อยู่ที่ 158.3 เฟรมต่อวินาที เท่านั้น ในเรื่องของ Model Size จะได้เปรียบเนื่องจากการประมวลผลภาพถ้าต้องการให้ประมวลผลภาพแบบ Real-time ไม่จำเป็นต้องใช้ทรัพยากรในการประมวลผลที่มากและมีขนาดใหญ่ Memory ที่มากอาจส่งผลให้ชิ้นส่วนมีขนาดใหญ่ขึ้นและการประมวลผลที่มากขึ้น ดังนั้น ในการเก็บข้อมูลและประมวลผลภาพต้องคำนึงถึงน้ำหนักในอากาศยานไร้คนขับด้วย ซึ่งความจุในที่นี้หมายถึง Capacity ที่ใช้ในหน่วยความจำ

ทั้งนี้ การใช้ GPU, CPU ที่แตกต่างกันอาจมีผลในเรื่องการประมวลผลที่แตกต่างกัน งานวิจัยนี้ได้นำข้อมูลทั้งหมด Training บน GPU ของ NVIDIA V100 TENSOR CORE GPU 32GB ใช้ Ubuntu 18.04 Desktop ในการพัฒนาต่อจากงานวิจัยนี้สามารถนำเทคโนโลยี Object Detection ไปใช้ในงานภาพถ่ายทางอากาศในด้านทางการทหารที่มีประโยชน์มากมาย ตัวอย่างเช่น

1. การตรวจจับและระบุวัตถุบนภาพถ่ายทางอากาศได้อย่างรวดเร็วและแม่นยำ เช่น รถถังที่กำลังเคลื่อนที่ เรือ หรือเครื่องบินศัตรู เป็นต้น ซึ่งสามารถช่วยให้ทหารมีข้อมูลที่สำคัญในการวางแผนการทำงานและการตอบสนองต่อสถานการณ์ได้อย่างมีประสิทธิภาพมากขึ้น

2. การตรวจสอบพื้นที่ในบริเวณที่ภาพถ่ายถูกถ่ายมา เช่น การตรวจสอบพื้นที่ที่เป็นเขตอันตราย การตรวจสอบสภาพอากาศหรือการตรวจสอบพื้นที่สำคัญสำหรับการวางกองทัพที่สามารถช่วยให้ทหารได้ข้อมูลที่สำคัญ

3. การแสดงข้อมูลในเวลาจริง การนำเทคโนโลยี Object Detection มาใช้ในภาพถ่ายทางอากาศช่วยในการแสดงข้อมูลที่รวดเร็วในเวลาจริงที่สามารถช่วยให้ทีมงานทหารได้รับข้อมูลสถานการณ์ที่อัปเดตและแม่นยำได้

## 6.เอกสารอ้างอิง

- [1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278 – 2324, 1998.
- [2] D. G. Lowe, "Object Recognition from Local Scale-invariant Features," in *Proc. 7th IEEE Int. Conf. Comput. Vision (ICCV'99)*, Kerkyra,



- Greece, 1999, pp. 1150-1157.
- [3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346 - 359, 2008.
- [4] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit. (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893.
- [5] W. Pei *et al.*, "Mapping and Detection of Land Use Change in a Coal Mining Area Using Object-based Image Analysis," *Environ. Earth Sci.*, vol. 76, pp. 1 – 16, 2017.
- [6] Y. Liu, S. U. Din, and Y. Jiang, "Urban Growth Sustainability of Islamabad, Pakistan, Over the Last 3 Decades: A Perspective based on Object-based Backdating Change Detection," *GeoJournal*, vol. 86, pp. 2035 – 2055, 2020.
- [7] M. Choinski, M. Rogowski, P. Tynecki, D. P. J. Kuijper, M. Churski, and J. W. Bubnicki, "A First Step Towards Automated Species Recognition from Camera Trap Images of Mammals Using AI in a European Temperate Forest," in *Int. Conf. Comput. Inf. Syst. Ind. Manage. (CISIM 2021)*, Etk, Poland, 2021, pp. 299 – 310.
- [8] W. Dai, H. Wang, Y. Song, and Y. Xin, "Wildlife Small Object Detection based on Enhanced Network in Ecological Surveillance," in *2021 33rd Chin. Control Decis. Conf. (CCDC)*, Kunming, China, 2021, pp. 1164-1169.
- [9] L. Dutrieux *et al.*, "Tree Species Detection and Identification from UAV Imagery to Support Tropical Forest Monitoring," in *EGU General Assem. Conf. (EGU 2020)*, 2020, p. 17759.
- [10] W. Lim, K. Choi, W. Cho, B. Chang, and D. W. Ko, "Efficient Dead Pine Tree Detecting Method in the Forest Damaged by Pine Wood Nematode (*Bursaphelench usxylophilus*) Through Utilizing Unmanned Aerial Vehicles and Deep Learning-based Object Detection Techniques," *Forest Sci. Technol.*, vol. 18, no. 1, pp. 36 – 43, 2022.
- [11] G. D. Georgiev, G. Hristov, P. Zahariev, and D. Kinaneva, "Forest Monitoring System for Early Fire Detection Based on Convolutional Neural Network and UAV Imagery," in *2020 28th Nat. Conf. Int. Participation (TELECOM 2020)*, Sofia, Bulgaria, 2020, pp. 57 - 60.
- [12] L. Shumilo, M. Lavreniuk, N. Kussul, and B. Shevchuk, "Automatic Deforestation Detection based on the Deep Learning in Ukraine," in *2021 11th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst.: Technol. Appl. (IDAACS 2021)*, Cracow, Poland, 2021, pp. 337 - 342.
- [13] K. - C. Chang, S. - H. Lin, J. - W. Huang, and Y. - F. Wu, "Automatic Incremental Training of Object Detection by Using GAN for River Level Monitoring," in *2021 IEEE Int. Conf. Consum. Electron. - Taiwan (ICCE-TW)*, Penghu, Taiwan, China, 2021, pp. 1 – 2.
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic

- Segmentation,” in *2014 IEEE Conf. Comput. Vision Pattern Recognit.*, Columbus, OH, USA, 2013, pp. 580 - 587.
- [15] R. Girshick, “Fast R-CNN,” in *2015 IEEE Int. Conf. Comput. Vision (ICCV 2015)*, Santiago, Chile, 2015, pp. 1440 - 1448.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137 - 1149, 2017.
- [17] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” in *14th Eur. Conf. Comput. Vision (ECCV 2016)*, Amsterdam, Netherlands, 2016, pp. 21 - 37.
- [18] T. - Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *2017 IEEE Conf. Comput. Vision Pattern Recognit. (CVPR 2017)*, Honolulu, HI, USA, 2017, pp. 936 - 944.
- [19] T. - Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318 - 327, 2017.
- [20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779 - 788.
- [21] A. Bochkovskiy, C. - Y. Wang, and H. - Y. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 2020, arXiv: 2004.10934.
- [22] C. - Y. Wang, A. Bochkovskiy, and H. - Y. M. Liao, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” in *2023 IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, 2023, pp. 7464 - 7475.
- [23] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and Efficient Object Detection,” in *2020 IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 10778 - 10787.
- [24] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” 2019, ArXiv: 1905.11946.
- [25] W. Luangluewut, K. Viriyasatr, W. Pawgasame, P. Kaewmongkol, and S. Mitaim, “Detecting Objects in Aerial Photographs Using Neural Network Techniques”, *Def. Technol. Acad. J.*, vol. 5, no. 12, pp. 4-11, Nov. 2023.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770 - 778.
- [27] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017, ArXiv: 1704.04861.