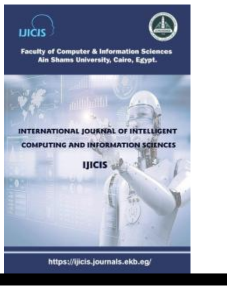
IJICIS, Vol.25, No.2, 18-31
DOI: 10.21608/ijicis.2025.374419.1387

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/


IJICIS
Faculty of Computer & Information Sciences Ain Shams University, Cairo, Egypt.
INTERNATIONAL JOURNAL OF INTELLIGENT COMPUTING AND INFORMATION SCIENCES
IJICIS
https://ijicis.journals.ekb.eg/

EVALUATING YOLOV8 VARIANTS FOR OBJECT DETECTION IN SATELLITE IMAGE

Andrew Magdy*
Department of Scientific Computing, Faculty of Computer and Information Sciences, Faculty of Computer and Information Sciences,
Cairo, Egypt
Andrew.Magdy@cis.asu.edu.eg

Marwa S. Moustafa
Department of Image Processing and its Application, National Authority for Remote Sensing and Space Sciences (NARSS)
Cairo, Egypt
Marwa@narss.sci.eg

Hala M. Ebied
Department of Scientific Computing, Faculty of Computer and Information Sciences, Faculty of Computer and Information Sciences,
Cairo, Egypt
halam@cis.asu.edu.eg

Received 2025-04-10; Revised 2025-05-26; Accepted 2025-06-10

Abstract: *Object detection in Remote sensing images enables crucial functionalities in various fields, including agriculture, environmental monitoring, urban planning, and disaster management. While traditional methods face challenges in speed and accuracy, deep learning has emerged as a powerful solution. This paper compares YOLOv8 variants (nano, small, medium, and large) for ship detection in high-resolution satellite images. The YOLOv8 model is a one-stage object detection that utilizes a Cross Stage Partial Darknet-53 (CSPDarknet53) backbone for feature extraction. A Path Aggregation Network - Feature Pyramid Network (PAN-FPN) neck for multi-scale feature fusion, and a decoupled head for final predictions. Its speed and accuracy can be adjusted by selecting from different model variants, which vary in size and complexity. The Ship dataset was chosen due to its challenging characteristics, including high-resolution imagery (30–50 cm) from Google Earth, diverse viewpoint variations, occlusions, cloud cover, shadows, varied lighting conditions, and cluttered marine backgrounds. mean average precision (mAP), recall, precision, F1-score, training time, and model size were utilized in evaluating YOLOv8 variants. The results demonstrate that YOLOv8s achieved the best balance between accuracy and efficiency, with an F1-score of 96.3% and a mAP50-95 of 70.4%. Although YOLOv8n demonstrates the highest processing speed, its detection performance is marginally inferior. In contrast, Larger models (YOLOv8m and YOLOv8l) do not show significant improvements in accuracy despite increased computational cost. The results provide insights into the effectiveness of each model for ship detection, enabling decisions for selecting the optimal model based on the balance between accuracy, speed, and resource utilization.*

Keywords: *Yolov8, Object detection, Remote sensing, Deep learning, satellite imagery.*

1. Introduction

***Corresponding Author**: Andrew Magdy

Scientific Computing Department, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt

Email address: Andrew.Magdy@cis.asu.edu.eg

Remote sensing, utilizing satellites and aircraft, gathers data without direct contact, which is crucial for different applications such as agriculture, environmental monitoring, urban planning, and disaster management [1]. Technological advancements have significantly improved the efficiency of image acquisition in remote sensing. The captured images are of high spatial, temporal, and spectral resolutions. Spatial resolution determines image detail, temporal resolution is the acquisition frequency for a specified region, and spectral resolution is the quantity of sensed electromagnetic wavelengths. These rich features indicate the growing importance of remote-sensing image analysis and classification [2]. However, the inherent complexity and dense distribution of objects within these images continue to render manual extraction processes inefficient and susceptible to errors. Hence, object detection approaches automate and improve object extraction from remote sensing imagery [3]. Traditional remote sensing image object-detection algorithms encompass several categories: (1) threshold-based methods, which rely on brightness or color differences, but are vulnerable to variations in lighting and complex backgrounds; (2) feature engineering methods utilize manually designed features like texture and shape, coupled with machine learning classifiers such as support vector machines, but require significant domain expertise; (3) template matching compares objects against predefined templates, effective for similar objects but less adaptable to variations like rotation; (4) machine learning methods employ algorithms like neural networks to automatically learn features and classifiers, requiring substantial labeled data (5) segmentation-based approaches divide images into regions for detailed analysis using methods like region growing or graph cuts, ideal for clear object-background boundaries; and (6) spectral information methods leverage spectral bands to identify objects based on spectral reflectance and angles. Each method offers distinct advantages and challenges, tailored to different remote sensing scenarios [4]. While traditional object detection methods have improved, they still struggle with speed and accuracy. Furthermore, aerial and satellite imagery introduces significant complexities, including viewpoint variations, occlusions, cloud cover, shadows, diverse lighting, and the need for noise reduction [5-6].

Nowadays, scholars predominantly employ deep learning models. Examples include the Regions with Convolutional Neural Networks (R-CNN) family (R-CNN, Fast R-CNN, Faster R-CNN), Single Shot MultiBox Detector (SSD), and You Only Look Once family (YOLO). These models employ Convolutional Neural Networks (CNNs) to directly extract features and perform object detection on feature maps [6]. Generally, deep learning approaches are broadly categorized into two main types: Two-stage detectors, such as the R-CNN variants, operate by initially generating region proposals. This involves meticulously identifying candidate regions that are likely to contain objects, often using techniques like Selective Search or a Region Proposal Network (RPN). While computationally more demanding, these methods generally achieve higher accuracy due to their precise candidate region extraction. In contrast, one-stage detectors, exemplified by YOLO and SSD, prioritize computational efficiency by directly predicting object attributes without an explicit region proposal step. This streamlined approach leads to faster processing speeds, though it may entail a slight trade-off in accuracy, particularly in scenarios involving complex object distributions or occlusions [7].

YOLO stands for “You Only Look Once” and is a widely adopted one-stage object detection algorithm that employs a unified neural network to predict bounding boxes and class probabilities in a single pass. YOLOV8 optimizes feature extraction and detection through sophisticated backbone and neck designs and employs an Ultralytics anchor-free split head, resulting in improved precision and efficiency compared to traditional methods. YOLOv8 excels in object detection across various applications. Its diverse pre-trained models variants *(*Nano, Small, Medium, and Large) are designed to optimize the balance between speed, accuracy, and computational efficiency, allowing users to choose the best fit for their requirements to achieve state-of-the-art accuracy and rapid real-time inference, YOLOv8 is a reliable solution for applications such as autonomous vehicles, robotics, security and surveillance, retail and manufacturing, and medical imaging. As an open-source framework, YOLOv8 benefits from continuous development and a robust community, ensuring its adaptability and sustained relevance in achieving state-of-the-art accuracy and real-time inference capabilities across a wide range of practical scenarios [8].

This paper evaluates the performance of different YOLOv8 *variants* (YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l) using the Ship dataset. YOLOv8, a state-of-the-art single-stage object detection framework, employs a Cross Stage Partial Darknet-53 (CSPDarknet53) backbone for robust feature extraction, a Path Aggregation Network-Feature Pyramid Network (PAN-FPN) neck for effective multi-scale feature fusion, and a decoupled head for final predictions. The model variants offer flexibility in balancing speed and accuracy by varying in size and computational complexity. The Ship dataset, characterized by high-resolution imagery (30–50 cm) sourced from Google Earth, was chosen for its challenging attributes, including diverse viewpoints, occlusions, cloud cover, shadows, varying lighting conditions, and complex marine backgrounds. Performance was assessed using metrics such as mean average precision (mAP), recall, precision, F1-score, training time, and model size to comprehensively evaluate the efficacy of the YOLOv8 variants. In ship dataset, the obtained result indicate that the mAP50-95 of yolov8n, yolov8s, yolov8m, and yolov8l reached 69%, 70.4%, 68.7%, and 67.2 respectively.

The rest of the paper is organized as follows: Section 2 reviews relevant work. Section 3 introduces the object detection methodology for satellite images. Section 4 presents and discusses the results. Finally, Section 5 draws our conclusions.

2. Related work

This section provides a concise overview of scholarly work in object detection methods based on deep learning in the remote sensing domain. Technically, the main goal of object detection is to identify and locate objects in images. Yu et al. [9] proposed You Only Look Once – Shuffle Reparameterized Blocks with Dynamic Head (YOLO-SRBD), an advanced variant of the YOLOv8 architecture, specifically tailored for enhanced ship detection in synthetic aperture radar (SAR) imagery. To address challenges posed by complex inshore environments, YOLO-SRBD integrates shuffle reparameterized blocks and a dynamic head, and utilizes soft non-maximum suppression during post-processing. Experimental evaluations conducted on the SAR Ship Detection Dataset (SSDD) demonstrated a substantial improvement in average precision, increasing from 66.7% (YOLOv8) to 74.3%, thereby validating the model's enhanced detection capability and practical applicability. Similarly, Khalili et al. [10] introduced Small Object Detection YOLOv8 (SOD-YOLOv8), a modified version of YOLOv8 optimized for the accurate detection of small objects in high-altitude imagery. This architecture achieves improved performance through several key enhancements, including a Generalized Feature Pyramid Network (GFPN)-inspired multi-level feature fusion mechanism and the addition of a high-resolution detection layer. Furthermore, it incorporates an efficient multi-scale attention module and adopts Powerful-IoU (PIoU) for more effective and rapid bounding box regression. Comparative analysis against YOLOv8s revealed notable performance gains: recall increased from 40.1% to 43.9%, precision from 51.2% to 53.9%, and mAP@0.5 from 40.6% to 45.1%. These improvements, coupled with maintained computational efficiency, underscore the model's robustness and reliability in dynamic and complex real-world scenarios.

Mostafa et al. [11] introduced a new dataset consisting of occluded road scenes common in Bangladeshi driving conditions to enhance the perception capabilities of AVs by solving the major challenge of detecting occluded objects, a major contributor to reliable and safe environmental perception. Transfer learning was used to fine-tune three object detection architectures: YOLOv5, YOLOX, and Faster R-CNN, over COCO-pretrained weights. The models were then validated and compared with the tailored dataset. YOLOX was the best among them, scoring the highest mAP@0.5:0.95 (0.634), showing its improved performance in detecting occluded objects under this specific environment.
Shen et al. [12] developed Deformable Self-attention YOLO version 8 (DS-YOLOv8), a novel object detection architecture built upon YOLOv8 to address the challenges of detecting objects with varying sizes, occlusions, and small dimensions while preserving fine details by integration of a Deformable Convolution Network C2f (DCN_C2f) module, which enables adaptable receptive fields, and a Self-Calibrating Shuffle Attention (SC_SA) module, which improves focus on both spatial and channel information. These modifications. DS-YOLOv8 also utilizes Wise Intersection over Union (Wise-IoU) and a position regression loss to optimize training. The DS-YOLOv8 model achieved high accuracy across various public datasets, achieving average mAP@0.5:0.95 scores of 74.0%, 64.3%, 70.7%, and 51.1%. It also maintained real-time inference capabilities and showed significant improvements in accuracy over YOLOv8 series models and other mainstream detection systems.

Zhai et al. [13] developed an optimized YOLOv8 model to improve tiny Unmanned Aerial Vehicle (UAV) detection. By adding a high-resolution head, removing layers, using Space-to-Depth Convolution (SPD-Conv), and incorporating Global Attention Module (GAM) attention, the model achieved a significant boost in detection performance (11.9% precision, 15.2% recall, 9% mAP) while reducing parameters and size by roughly 60%. Validated on multiple datasets, this method proves highly effective for practical UAV detection. Magdy et al. [6] conducted a comparative study of four popular backbones: Residual Network with 50 layers (ResNet-50), ResNeXt50_324d, EfficientNet_B0, and Densely Connected Convolutional Network (DenseNet121)—for remote sensing object detection. Using the NWPU VHR-10 dataset, they evaluated the backbones based on precision, recall, F1-score, average precision (AP), and mean AP. Their findings revealed that ResNeXt50_324d achieved the highest performance, with a mean AP of 0.847, surpassing the other backbones tested.

Zhang et al. [14] proposed HSP-YOLOv8, an enhanced version of YOLOv8 tailored for small object detection in drone aerial images. It addresses challenges like occlusion, crowding, overlapping, and object density by adding a prediction head for higher-level features, using a Space-to-Depth Convolution (SPD-Conv) to preserve small object details, and applying Soft-NMS during post-processing to reduce false detections. On the VisDrone2019 dataset, HSP-YOLOv8 outperformed YOLOv8 and other YOLO variants (YOLOv4 to YOLOv8l), achieving up to 11% and 9.8% gains in mAP@0.5 and mAP@0.5:0.95, respectively, with a top accuracy of 49.6% mAP@0.5. Despite improved accuracy, it remains efficient with only 11.5M parameters, 50 GFLOPs, and a 6.2 ms inference time, making it well-suited for UAV small object detection. Yao et al. [15] developed High-Performance YOLO version 8 (HP-YOLOv8), an enhanced algorithm for small object detection in remote sensing. This model uses the C2f-DM module, Bi-directional Global Feature Pyramid Network (BGFPN), and Smoothed Modified Penalized Distance Intersection over Union (SMPDIoU) loss to improve detection. It achieved mAP@0.5 scores of 95.11% Remote Sensing Object Detection (RSOD), 93.05% (NWPU VHR-10), and 53.49% (VisDrone2019), and mAP@0.5:0.95 scores of 72.03%, 65.37%, and 38.91%, respectively. Nie et al. [16] improved YOLOv8n's small object detection by adding a specialized layer, using the Selective Small Feature Fusion

(SSFF) module, and replacing the Path Aggregation Network with Hierarchical Path Aggregation Network (HPANet). This resulted in substantial mAP gains (14.3%/17.9% for mAP@0.5, 17.1%/19.8% for mAP@0.5:0.95 on VisDrone and AI Tiny Object Detection (AI-TOD) datasets, while reducing parameters by 33% and model size by 31.7%, demonstrating effective small object detection.

Radionov et al. [17] improved the detection of aircraft in high-resolution satellite imagery. It used deep neural networks (YOLOv8m) with a novel sequential boundary traversal algorithm for contour detection. Preprocessing of images from the HRPlanesv2 dataset to highlight object boundaries and its combination with raw data, the method achieved higher accuracy: mAP@50 95 of 0.864. This represents a 1% and 4.8% improvement over baseline, showing high proficiency for aircraft identification. To sum up, Table 1 summarizes the recent YOLOv8-based object detection enhancements.

Table 1 Overview of recent YOLOv8-based object detection methods, their key techniques, datasets, and accurate results.

Authors	Methodology	Dataset	Accuracy
Yu et al. [9]	YOLO-SRBD: YOLOv8 with shuffle reparameterized blocks, dynamic head, soft NMS for enhanced ship detection in SAR imagery.	SAR Ship Detection Dataset (SSDD)	Average Precision: 74.3% (improved from 66.7% YOLOv8)
Khalili et al. [10]	SOD-YOLOv8: YOLOv8 enhanced with GFPN-inspired multi-level feature fusion, additional high-res detection layer, multi-scale attention module, and PIoU.	High-altitude images dataset	Recall: 43.9%, Precision: 53.9%, mAP@0.5: 45.1% (all improved over YOLOv8s)
Mostafa et al. [11]	Transfer learning fine-tuned YOLOv5, YOLOX, Faster R-CNN on new occluded road scenes dataset to improve detection under occlusion.	New occluded road scenes dataset (Bangladesh driving conditions)	YOLOX best: mAP@0.5:0.95 = 63.4%
Shen et al. [12]	DS-YOLOv8: YOLOv8 + Deformable Convolution (DCN_C2f), Self-Calibrating Shuffle Attention, Wise-IoU, position regression loss.	Various public datasets	mAP@0.5:0.95 scores: 74.0%, 64.3%, 70.7%, 51.1%
Zhai et al. [13]	Optimized YOLOv8 for tiny UAV detection: high-res head, removed layers, SPD-Conv, Global Attention Module.	Multiple datasets	Precision +11.9%, Recall +15.2%, mAP +9%, parameters reduced by ~60%
Magdy et al. [6]	Comparative study of backbones (ResNet-50, ResNeXt50_324d, EfficientNet_B0, DenseNet121) for remote sensing object detection.	NWPU VHR-10 dataset	Best backbone ResNeXt50_324d: mean AP = 84.7%
Zhang et al. [14]	HSP-YOLOv8: YOLOv8 with extra prediction head, SPD-Conv, Soft-NMS for small object detection in drone aerial images.	VisDrone2019 dataset	mAP@0.5: 49.6%, mAP@0.5:0.95 gain up to 11% and 9.8% over YOLOv8 variants
Yao et al. [15]	HP-YOLOv8: uses C2f-DM module, Bi-directional GFN, SMPDIoU loss for remote sensing small object detection.	RSOD, NWPU VHR-10, VisDrone2019 datasets	mAP@0.5: 95.11%, 93.05%, 53.49%; mAP@0.5:0.95: 72.03%, 65.37%, 38.91%
Nie et al. [16]	Improved YOLOv8n with selective small feature fusion, hierarchical path aggregation network, and specialized detection layer.	VisDrone, AI Tiny Object Detection (AI-TOD) datasets	mAP@0.5 gains: +14.3%/17.9%, mAP@0.5:0.95: +17.1%/19.8%, parameter reduction 33%

Radionov et al. [17]	YOLOv8m + sequential boundary traversal algorithm for aircraft detection in high-res satellite images with preprocessing.	HRPlanesv2 dataset	mAP@50–95 = 86.4%, +1% and +4.8% improvement over baseline

3. Methodology

YOLOv8 overall architecture, depicted in Figure 1, is composed of a Backbone, a Neck, and a Head, a common design pattern for one-stage object detectors [18]. The sub-sections below discuss the components of YOLOV8 in detail.

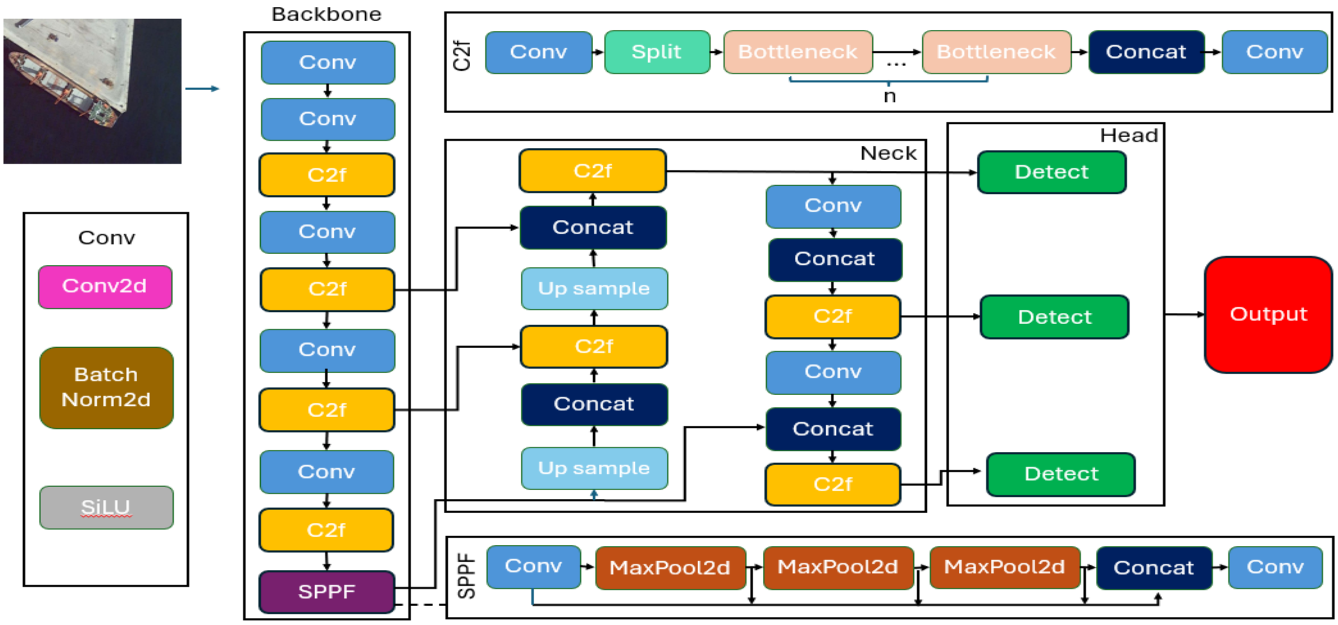


Figure. 1: The architecture of YOLOv8

3.1. Backbone Layer

The YOLOv8 architecture critically relies on its Convolutional Neural Network (CNN) backbone for feature extraction, specifically leveraging an enhanced Cross Stage Partial Darknet 53 (CSPDarknet53). This backbone integrates cross-stage partial connections to optimize inter-layer information flow and improve detection accuracy. The Key components of the backbone include: 1) The Convolution + BatchNorm + SiLU (CBS) module: This module, an evolution from the traditional Convolution + BatchNorm + Leaky ReLU (CBL), combines a 3x3 convolution, batch normalization, and the Sigmoid Linear Unit (SiLU) activation function.2) The C2f module: Replacing the C3 module from YOLOv5, C2f is a hybrid structure derived from C3 and Efficient Layer Aggregation Networks (ELAN), incorporating gradient shunt connections. This design significantly enhances information flow and contributes to a substantial improvement in detection accuracy. 3) The Spatial Pyramid Pooling – Fast (SPPF) module: Positioned at the terminal end of the backbone, the SPPF module processes features through a sequential application of three 5x5 MaxPool operations, followed by concatenation. This mechanism facilitates robust object detection across a diverse range of scales while preserving computational efficiency. [19].

3.2. Neck Layer

The YOLOv8 neck, also called a feature extractor, plays a crucial role in YOLOv8 by aggregating feature maps from the backbone's various stages to facilitate multi-scale feature integration [4]. YOLOv8's neck leverages a PAN-FPN architecture, integrating FPN and PAN to combine feature maps from various scales. This architecture, featuring upsampling, C2f modules, and a YOLOx-derived decoupled head, enhances the fusion of feature layer information, boosting detection accuracy [20].

3.3. Head Layer

YOLOv8's head takes feature maps from the backbone and generates the final output: bounding boxes and object classes. It uses multiple detectors at different scales to predict bounding box coordinates, objectness scores, and class probabilities, optimizing accuracy by specializing each scale-specific branch for different object sizes [4]. Upsample (U) layers enhance feature map resolution, while a sequence of convolutional layers followed by a linear layer predicts bounding boxes and class probabilities efficiently [8]. YOLOv8's head utilizes a decoupled structure, separating object classification and bounding box regression. It incorporates the Anchor method and employs distinct loss functions, including Binary Cross-Entropy (BCE) for classification as defined in Eq. (1) [21]. For bounding box regression, it uses Distribution Focal Loss (DFL) with Complete Intersection over Union (CIoU), as shown in Eqs. (2) [22] and (3) [21]. These components enhance detection accuracy and accelerate model convergence [19].

$$\boldsymbol{Loss_{BCE} = -W[y_n \log x_n + (1 - y_n) \log(1 - x_n)]} \tag{1}$$

where w represents the weight; $\boldsymbol{y_n}$ presents the actual value, and $\boldsymbol{x_n}$ represents the predicted value generated by the model.

$$\boldsymbol{Loss_{DFL} = -\left[(y_{n+1} - y) \log \frac{y_{n+1}-y}{y_{n+1}-y_n} + (y - y_n) \log\left(\frac{y-y_n}{y_{n+1}-y_n}\right)\right]} \tag{2}$$

$$\boldsymbol{Loss_{CIoU} = 1 - IoU + \frac{d^2}{c^2} + \frac{v^2}{(1-IoU)+v}} \tag{3}$$

where v is a parameter that measures aspect ratio consistency (defined in Eq. 4 [21]), IoU represents the overlap between predicted and ground truth boxes, 'd' is the Euclidean distance between their centers, and 'c' is the diagonal of the smallest box enclosing both.

$$\boldsymbol{v = \frac{4}{\pi^2}\left(arctan \frac{w_{gt}}{h_{gt}} - arctan \frac{w_p}{h_p}\right)^2} \tag{4}$$

where w represents the bounding box width, h represents the bounding box height, gt refers to the ground truth (actual) value, and p refers to the predicted value.

The primary distinction among different variants of YOLOv8 lies in their size and complexity. Larger and more complex models offer higher accuracy but operate at slower speeds, whereas smaller and simpler models have less accuracy for faster processing. For instance, YOLOv8n has a size of 2.4 MB with 4.7 million parameters, while YOLOv8m, with a size of 8.1 MB and 20.0 million parameters, focuses on achieving higher accuracy despite its increased computational demands [8].

4. Experimental Results

This section presents the experiments conducted on YOLOv8 assessment on Ship dataset. Section 4.1 briefly describes the Ship benchmark dataset. Section 4.2 describes the experiment setup. Section 4.

highlights the evaluation metrics. Finally, section 4.4 highlights YOLOv8's advancements and its effectiveness compared to its versions.

4.1. Dataset

The Ship dataset [23] comprises images sourced from Google Earth, with a 30 to 50 cm resolution. The images predominantly feature ships and boats against a vast ocean background, although some show vessels near the shore or in clusters. The dataset contains 556 images for training (70%), 159 for validation (20%), and 79 for testing (10%). Three augmentations suitable for aerial imagery were applied for the training images: a horizontal flip and a 2×rotation. This process tripled the number of training images to around 1,400 and increased the number of ship dataset images to 1638. Additionally, all images were resized to 640×640 pixels to match the input requirements of the YOLOv8 model, as shown in the Figure. 2. Technically, a ship detection dataset was selected over more complex multi-class datasets to facilitate focused, single-class evaluation of various YOLOv8 model variants namely: YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l. This choice allows for controlled and consistent comparisons across models. Moreover, ship detection in aerial and satellite imagery poses unique challenges, including variations in viewpoint, occlusions, cloud cover, shadows, lighting inconsistencies, and the visual complexity of maritime backgrounds (e.g., wave patterns and color gradients). These characteristics make the ship detection dataset a compelling benchmark for evaluating the robustness, adaptability, and real-world applicability of YOLOv8 variants in remote sensing tasks.

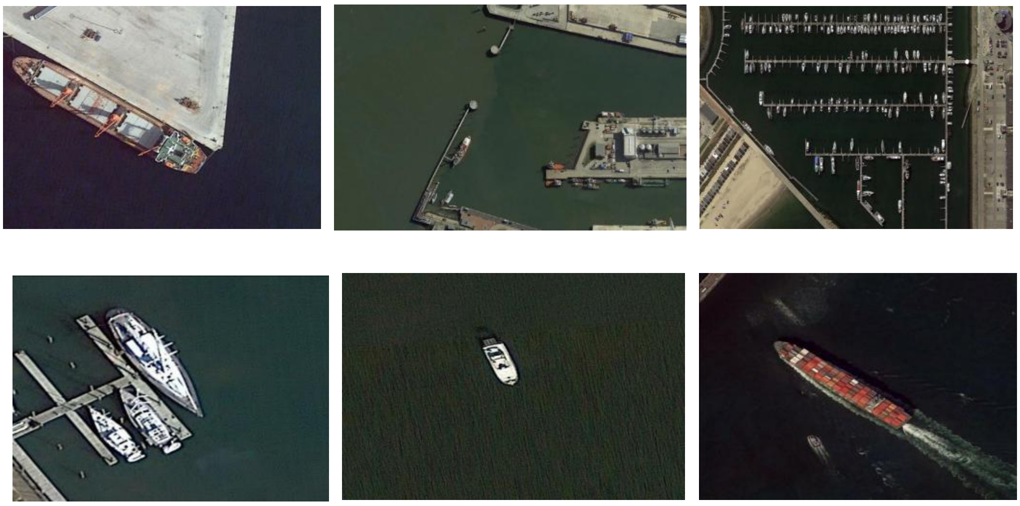


Figure. 2: Samples of the Ship dataset.

4.2. Experimental Setup

To ensure consistency and fairness in model comparison, all YOLOv8 variants (YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l) were trained under an identical experimental environment. The training process utilized the AdamW optimizer, configured with a learning rate of 0.002, a momentum factor of 0.9, a batch size of 16, and a weight decay coefficient of 0.0005.

All experiments were conducted on the Google Colab platform, leveraging an NVIDIA Tesla T4 GPU equipped with 16 GB of dedicated GPU memory. The system environment included 13 GB of system RAM and 80 GB of available disk space. This setup ensured adequate computational resources for training deep learning models while also reflecting a commonly accessible platform for practical research and experimentation. The use of a consistent hardware and software configuration across all model variants helped to isolate the effect of model architecture on performance outcomes.

4.3. Evaluation Metrics

The performance of the YOLOv8 model was assessed using a comprehensive set of evaluation metrics, including precision, recall, F1-score, Average Precision (AP), and mean Average Precision (mAP), as defined in Equations (5), (6), (7), (8), and (9) [24] respectively. In addition to these performance indicators, the evaluation also considered computational factors such as the number of model parameters, total model size, and training time.

$$\boldsymbol{P = \frac{TP}{TP+FP}} \tag{5}$$

$$\boldsymbol{R = \frac{TP}{TP+FN}} \tag{6}$$

$$\boldsymbol{F1 - score = 2\,\frac{Precision * Recall}{Precision\ and\ Recall}} \tag{7}$$

$$\boldsymbol{AP = \frac{1}{11} * Sum(11\ point\ interpolated\ precision)} \tag{8}$$

$$\boldsymbol{mAP = \frac{1}{n} * sum(AP)} \tag{9}$$

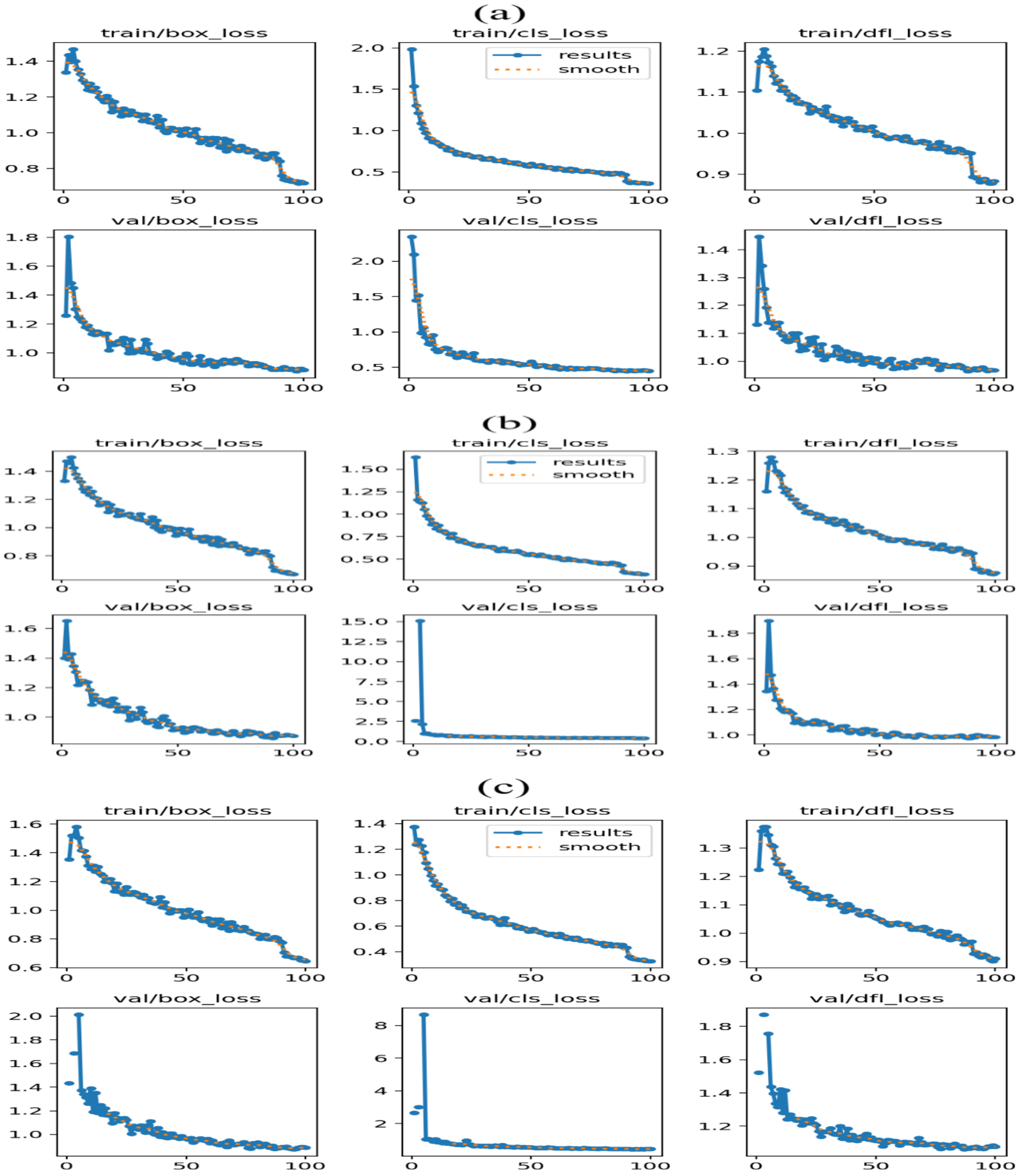
where TP is the (True Positives) representing correctly identified positive cases, FP is the (False Positives) representing negative cases incorrectly identified as positive, and FN is the (False Negatives) representing positive cases incorrectly identified as negative.

4.4. Results

This section investigates the performance of different YOLOv8 model variants on a single-class ship detection dataset. The evaluation focuses on both accuracy and computational efficiency, using metrics such as precision, recall, F1-score, average precision (AP), number of model parameters, model size, and training time.

First, Figure 3 depicts the training and validation loss curves for YOLOv8 variants: (a) YOLOv8n, (b) YOLOv8s, (c) YOLOv8m, and (d) YOLOv8l, consistently decrease, indicating effective learning across all models. Larger models (YOLOv8m, YOLOv8l) start with slightly higher training losses but ultimately achieve lower final values. Validation losses drop sharply initially before stabilizing, with YOLOv8m and YOLOv8l reaching lower final values. In comparison, YOLOv8n and YOLOv8s exhibit smoother but slightly higher validation losses. The validation/classification loss (the box loss, cls loss, and dfl loss curves) decreases significantly for YOLOv8m and YOLOv8l, highlighting their enhanced learning capability.

Next, Table 2 presents a comparison of different YOLOv8 variants: Nano (YOLOv8n), Small (YOLOv8s), Medium (YOLOv8m), and Large (YOLOv8l) based on Recall, Precision, F1-Score, mAP50-95, Training Time, and Number of Parameters. Among them, YOLOv8s demonstrates the best balance of accuracy and efficiency, achieving the highest Recall (90.1), Precision (92.3), F1-score (96.3), and mAP50-95 (70.4). YOLOv8n is the most lightweight model, with the smallest parameter count (3.2M) and shortest training time (0.736 hours), making it ideal for resource-constrained environments, though with slightly lower accuracy. In contrast, YOLOv8m and YOLOv8l have significantly larger parameter sizes (25.9M and 43.7M, respectively) and require more training time but do not show substantial improvements over YOLOv8s in key performance metrics like mAP50-95. Notably, YOLOv8l achieves the highest precision (93.0) but has a lower mAP50-95 (67.2), indicating a trade-off between precision and overall detection performance. The trade-off between precision and recall for each version of YOLOv8 is depicted in Figure 4.



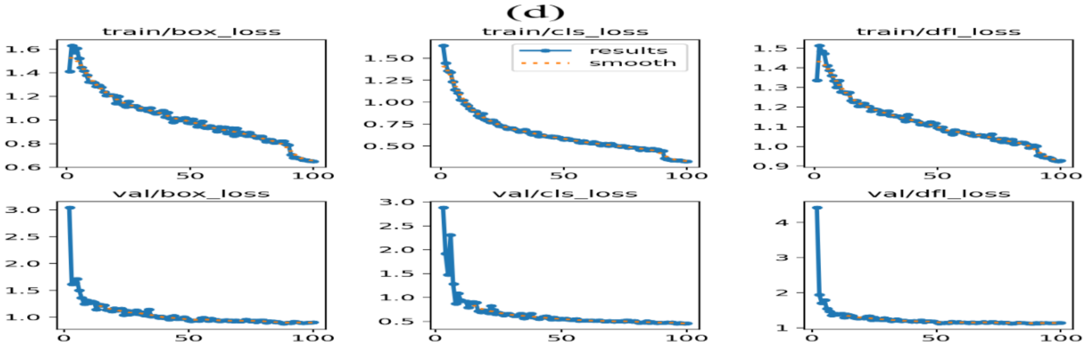


Figure. 3: The training and validation loss (a) yolov8n, (b) yolov8s, (c) yolov8m, and (d) yolov8l.

Table 2 Comparison of Efficiency Gains in YOLOv8 versions.

Model	Recall	Precision	F1-Score	mAP50-95	Time (Hr)	# Parameter (M)
YOLOv8n	88.6	90.3	95.0	69.0	0.736	3.2
YOLOv8s	90.1	92.3	96.3	70.4	0.836	11.2
YOLOv8m	89.1	91.1	77.9	68.7	1.423	25.9
YOLOv8l	88.8	93.0	81.4	67.2	2.317	43.7

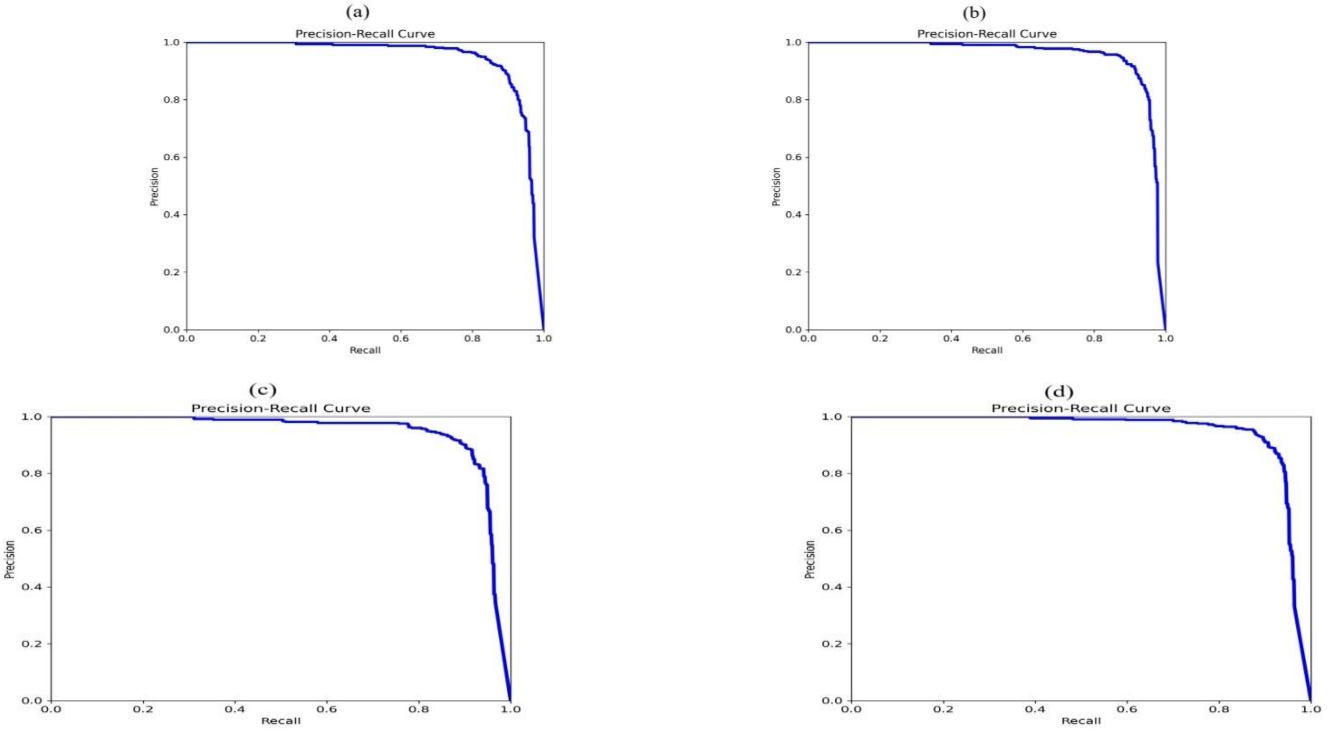


Figure. 4: Precision-recall curve of the (a) yolov8n, (b) yolov8s, (c) yolov8m, and (d) yolov8l

Finally, Four Precision-Recall curves (a, b, c, and d) were examined to understand the trade-off between precision (accuracy of positive predictions) and recall (ability to find all actual positives) in classification models. Curves (a) and (c) show strong, consistent performance, maintaining high precision (above 0.8) across a wide range of recall values. Curves (b) and (d), however, reveal more variability. Curve (b) demonstrates a sharp drop in precision after a recall of 0.8, suggesting decreasing accuracy as more true positives are targeted. Curve (d) shows a more gradual decline in precision, starting at a recall of 0.6, followed by a steeper drop at higher recall. These variations, even potentially within similar experimental conditions, highlight the importance of comprehensive evaluation, as model performance is sensitive to subtle changes. Samples of the obtained results of the YOLOv8s are shown in Figure 5.

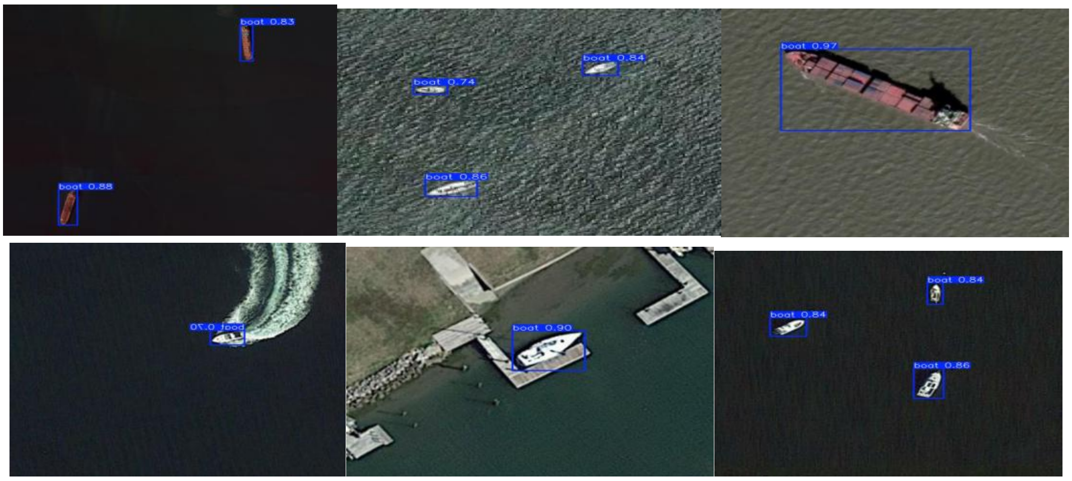


Figure. 5: Samples of the obtained results of the YOLOv8s for the ship dataset

5. Conclusion

Object detection plays a critical role in a variety of remote sensing applications, including agriculture, environmental monitoring, and urban planning. However, manual image analysis is often time-consuming and inefficient. Automated object detection, therefore, is essential for extracting accurate and timely information from complex imagery. YOLOv8, a one-stage deep learning model, addresses this need by employing a single neural network to concurrently predict object locations (bounding boxes) and classifications (class probabilities). The model is available in several pre-trained *variants* (YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l) offering a different balance between speed, accuracy, and computational efficiency.

This study systematically evaluated the performance of various YOLOv8 models in the context of remote sensing, specifically for ship detection tasks. The architecture of YOLOv8 comprises three main components: the Backbone for feature extraction, the Neck for multi-scale feature fusion, and the Head for final object prediction. Experimental results using the Ship dataset highlight that YOLOv8s achieves the most favourable balance, attaining a high F1-score of 96.3% and a mAP50-95 of 70.4%, making it well-suited for deployment on resource-constrained systems. While YOLOv8n demonstrated the fastest training time and smallest model size, its detection performance was marginally lower. Conversely, YOLOv8m and YOLOv8l, despite requiring more computational resources and training time, did not yield significant improvements over YOLOv8s. Notably, YOLOv8l achieved the highest precision but recorded a lower mAP50-95, illustrating the nuanced trade-offs between performance metrics. In Future, we will enhance YOLOv8's performance in complex situations through domain adaptation and multispectral data integration. Pruning and quantization methods could improve edge device applicability. Broader tests with other datasets and real-time applications can further validate and extend its applications to remote sensing.

References

1. S. Ahmed, Comparison of satellite images classification techniques using Landsat-8 data for land cover extraction, Int. J. Intell. Comput. Inf. Sci. 21(3) (2021) 29–43.
2. M.A. Shafaey, M. ElBery, M.A.-M. Salem, H. Moushier, E.-S.A. El-Dahshan, M. Tolba, Hyperspectral image analysis using a custom spectral convolutional neural network, Int. J. Intell. Comput. Inf. Sci. 22(4) (2022) 146–158.
3. Z. Li, J. Yuan, G. Li, H. Wang, X. Li, D. Li, RSI-YOLO: Object Detection Method for Remote Sensing Images Based on Improved YOLO, Sensors. 23(14) (2023) 6414.
4. T. Wu, Y. Dong, YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition, Applied Sciences (Switzerland). 13(24) (2023) 12977.
5. N. Laban, B. Abdellatif, H.M. Ebied, H.A. Shedeed, M.F. Tolba, Enhanced pixel-based urban area classification of satellite images using convolutional neural network, Int. J. Intell. Comput. Inf. Sci. 21(3) (2021) 13–28.
6. A. Magdy, MS. Moustafa, HM. Ebied, MF. Tolba, Backbones-Review: Satellite Object Detection Using Faster-RCNN, In: Springer Proceedings in Earth and Environmental Sciences, 2023, p.241-248.
6. M. Ma, H. Pang, SP-YOLOv8s: An Improved YOLOv8s Model for Remote Sensing Image Tiny Object Detection, Applied Sciences (Switzerland). 13(14) (2023) 8161.
8. M. Sohan, T. Sai Ram, C. V. Rami Reddy, A review on YOLOv8 and its advancements, In: International Conference on Data Intelligence and Cognitive Informatics, Springer, Singapore, 2024, p.529–545.
9. C. Yu, Y. Shin, An efficient YOLO for ship detection in SAR images via channel shuffled reparameterized convolution blocks and dynamic head, ICT Express. 10(3) (2024) 673–679.
10. B. Khalili, A.W. Smyth, SOD-YOLOv8—Enhancing YOLOv8 for small object detection in aerial imagery and traffic scenes, Sensors. 24(19) (2024) 6209.
11. T. Mostafa, S. J. Chowdhury, M. K. Rhaman, M. G. R. Alam, Occluded object detection for autonomous vehicles employing YOLOv5, YOLOX and Faster R-CNN, In: IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2022, p.0405–0410.
12. L. Shen, B. Lang, Z. Song, DS-YOLOv8-Based Object Detection Method for Remote Sensing Images, IEEE Access. 11 (2023) 125122-125137.
13. X. Zhai, Z. Huang, T. Li, H. Liu, S. Wang, YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection, Electronics (Switzerland). 12(17) (2023) 3664.
14. H. Zhang, W. Sun, C. Sun, R. He, Y. Zhang, HSP-YOLOv8: UAV aerial photography small target detection algorithm, Drones. 8(9) (2024) 453.
15. G. Yao, S. Zhu, L. Zhang, M. Qi, HP-YOLOv8: High-Precision Small Object Detection Algorithm for Remote Sensing Images, Sensors. 24(15) (2024) 4858.
16. H. Nie, H. Pang, M. Ma, R. Zheng, A Lightweight Remote Sensing Small Target Image Detection Algorithm Based on Improved YOLOv8, Sensors. 24(9) (2024) 2952.
17. Y.D. Radionov, V.Y. Kashtan, V.V. Hnatushenko, O.V. Kazymyrenko, Aircraft detection with deep neural networks and contour-based methods, Radio Electronics, Computer Science, Control. (4) (2024) 121–129.

18. M. Liu, M. Zhang, X. Chen, C. Zheng, H. Wang, YOLOv8-LMG: An Improved Bearing Defect Detection Algorithm Based on YOLOv8, Processes. 12(5) (2024) 930.

19. C. Liu, F. Meng, Z. Zhu, L. Zhou, Object Detection of UAV Aerial Image based on YOLOv8, Frontiers in Computing and Intelligent Systems. 5(3) (2023) 46-50.

20. H.Lou, X. Duan, J. Guo, H. Liu, J. Gu, L. Bi, DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor, Electronics (Switzerland). 12(10) (2023) 2323.

21. C.T. Chien, R.Y. Ju, K.Y. Chou, E. Xieerke, J.S. Chiang, YOLOv8-AM: YOLOv8 based on effective attention mechanisms for pediatric wrist fracture detection, IEEE Access (2025).

22. A. Sahafi, A. Koulaouzidis, M. Lalinia, Polypoid Lesion Segmentation Using YOLO-V8 Network in Wireless Video Capsule Endoscopy Images, Diagnostics. 14(5) (2024) 474.

23. M. Dorrich, M. Fan, AM. Kist, Impact of Mixed Precision Techniques on Training and Inference Efficiency of Deep Neural Networks, IEEE Access. 11 (2023) 57627-57634.

24. K. Patel, C. Bhatt, PL. Mazzeo, Deep Learning-Based Automatic Detection of Ships: An Experimental Study Using Satellite Images, J Imaging. 8(7) (2022) 182.