SETSCI Conference Proceedings
SETSCI Conference Proceedings 21, 6-11, 2024

International Trend of Tech Symposium
7-8 December 2024, İstanbul - Türkiye
https://doi.org/10.36287/setsci.21.2.006

Optimizing Accuracy and Efficiency in Ship Detection from Satellite Images through a Comparative Analysis of Object Detection Models

Nesli S. Azdavay[1*], Zofia G. Zajac[2], Hakan M. Didis[1] and Ilayda Boz[3*]

[1]*Computer Engineering Department, Mersin Univerity, Mersin, Turkey*
[2]*Department of Transport & Logistics, University of Radom, Masovian Voivodeship, Poland*
[3]*Aerospace Engineering Department, Tarsus University, Mersin, Turkey*
[*]*(nazdavay@mersin.edu.tr)*

Abstract – This study evaluates the performance of Faster R-CNN and YOLOv7 object detectors for dynamic ship detection in maritime applications. Both algorithms are tested under challenging conditions, including poor image quality due to cloud and dust obscuration, varying lighting, and high-altitude captures. The necessity for detection methods with high accuracy and efficiency in such environments is discussed. Results show that Faster R-CNN outperforms YOLOv7 in detection accuracy, achieving superior precision, recall, mAP, and F1 scores. This accuracy allows for successful identification and classification of ships even in low-quality images. However, despite its accuracy advantage, Faster R-CNN falls short in speed, with an average detection time of 53.4 seconds, making it less viable for real-time use. Conversely, YOLOv7 processes images significantly faster, with an extraction time of just 21.6 seconds, though at the cost of lower accuracy. Its rapid processing makes YOLOv7 more suitable for real-time applications requiring quick decisions. This study underscores the importance of balancing detection accuracy and speed when selecting an algorithm for ship detection, offering key insights to enhance maritime safety, traffic monitoring, and autonomous navigation systems.

Keywords – ship detection, satellite imagery, Faster R-CNN, deep learning, YOLOv7

I. Introduction

Object detection, a widely used computer vision technique, has increasingly become integrated with deep learning methods. The goal of object detection is to locate objects within an image and identify their respective classes. It plays a crucial role in numerous applications, such as detecting potholes on roads, identifying anomalies in medical images, vehicle detection, and face recognition for security purposes. Over time, the use of advanced satellite systems and unmanned aerial vehicle (UAV) technologies for object detection has become more prevalent. Computer vision methods significantly enhance the detection and recognition of objects in images captured from satellites and UAVs [1]. Each system offers distinct advantages and drawbacks depending on the use case and scenario. UAVs, for instance, are more agile and can be easily maneuvered over specific areas or targets due to their high mobility, making them ideal for dynamic tracking [2]. However, UAVs have limited range and are negatively affected by harsh weather conditions, which can hinder object detection performance. In contrast, satellites are less impacted by weather and can conduct real-time observations while covering larger areas quickly without range limitations. Additionally, satellites equipped with high-resolution cameras produce high-quality images, boosting the performance of object detection algorithms in real-time applications [3]. Satellite imagery provides valuable data for a variety of fields, including cartography, meteorology, and environmental monitoring [4].

Object detection in satellite imagery can classify an image into multiple categories and localize the position of the detected objects [5]. Satellite-based detections contribute to the monitoring and development of coastal areas and ports, playing a vital role in protecting marine ecosystems, offshore zones, and inland waterways [6]. However, factors such as waves, varying backgrounds, and inconsistent lighting on the sea surface make ship detection particularly challenging. In this context, it is crucial to account for all potential imaging variations to improve detection accuracy on the sea surface. Additionally, the small scale of target objects in satellite imagery further complicates object detection tasks [7]. Object detection systems pinpoint the locations of elements in an image that they are trained to recognize. This capability allows for the identification of coastal and offshore ships, which helps prevent collisions and enhances sea and port security. Detecting nearby ships is particularly important for collision avoidance. In civil maritime applications, ship detection helps prevent pollutants like oil and waste from contaminating water bodies and deters illegal or unauthorized activities. In military contexts, detecting ships' positions, sizes, directions, and speeds aids in monitoring cross-border movements or other unusual activities to safeguard coastal and maritime security [8].

Several models have been developed to maintain real-time object detection. Notable object detection algorithms include R-CNN [9], Faster R-CNN [10], SSD (Single Shot Detector) [11], RefineDet (Single Shot Enhancement Neural Network for Object Detection) [12], and You Only Look Once (YOLO) [13]. These algorithms have broad applications in areas such as autonomy, healthcare, security, and surveillance. Object detection methods can be classified into two types: single-stage detectors, which prioritize fast processing, and two-stage detectors, which generate region proposals before classification. Single-stage detectors use fully convolutional networks to handle classification and regression tasks in one step, whereas two-stage detectors employ deep neural networks to separate classification and regression tasks into distinct stages for more accurate results [14, 15]. Examples include the fast and efficient YOLO family of single-stage detectors and high-accuracy two-stage detectors like SSD, Faster R-CNN, and R-FCN (Region-based Fully Convolutional Network). Deep convolutional neural networks (CNNs) have played a pivotal role in object detection, making region-based CNN methods the standard while phasing out traditional techniques in favor of deep learning [16].

This study compares the performance of the YOLOv7 and Faster R-CNN algorithms for ship detection in satellite imagery. YOLOv7, as part of the YOLO family, excels in synchronously predicting bounding boxes and class probabilities in a single stage, balancing speed and accuracy for real-time applications. In contrast, Faster R-CNN is a two-stage detector known for its high accuracy, particularly in complex detection tasks, due to its region proposal process. However, this comes at the cost of slower execution. While Faster R-CNN provides superior precision and recall, YOLOv7 is more suitable for applications requiring rapid detection, making it ideal for real-time monitoring systems. This study examines the trade-offs between these models in the critical task of ship detection, offering a comprehensive analysis of their respective strengths and limitations through detailed performance measurements.

II. Methodology

A. Faster R-CNN and Convolutional Layer

Faster R-CNN is a two-stage object detection framework known for its high accuracy and efficiency. Introduced in 2015, it replaces the selective search method used in earlier R-CNN models, which was computationally expensive, with the Region Proposal Network (RPN) to improve both speed and precision. By incorporating RPN, Faster R-CNN eliminates the reliance on external region proposals, becoming one of the most effective and widely used object detection algorithms today [17]. The first step in the Faster R-CNN process involves extracting feature maps from the input image using a CNN. These feature maps are then fed into the RPN, which generates region proposals, followed by the Region of Interest (RoI) pooling and classification stages. Faster R-CNN can be understood through its four main components. The CNN consists of convolutional and pooling layers [18]. The input image is processed through the convolution layers, which apply filters to extract essential features, followed by pooling layers that downsample the feature maps. These feature maps are then passed to the RPN and subsequent fully connected layers. At this stage, the image is represented as a matrix of pixels, and the convolutional layers generate feature map outputs using activation functions like ReLU and pooling operations.

The RPN is a deep, convolutional neural network responsible for generating region proposals [19]. It does this by identifying anchors, which are potential object locations, using a sliding window mechanism over the feature maps produced by the convolutional layers. The RPN assigns a probability to each anchor through a softmax layer, and bounding box regression is applied to refine these proposals. The sliding window moves across the feature maps to generate anchors, which are boxes representing potential object locations of various shapes and sizes. To minimize redundancy, non-maximum suppression (NMS) is used to eliminate overlapping proposals. During training, the RPN generates 2000 region proposals, but this number may vary during testing. The Intersection over Union (IoU) ratio is calculated for each bounding box, which takes values between 0 and 1, measuring the overlap between predicted and ground truth boxes. Higher IoU values indicate more accurate proposals. Once the region proposals are refined, they are passed to the RoI pooling layer for further processing.

The RoI pooling layer combines the feature maps with the region proposals. Its primary function is to convert regions of varying sizes into fixed-size feature maps using max pooling, which reduces the spatial dimensions of the image. A 7x7 filter with 512 convolutional layers is typically used to achieve this transformation. After this step, the image is processed through two key layers: the softmax layer, which classifies objects when multiple objects are present, and the regressor layer, which refines the bounding boxes. The RoI pooling layer outputs 7x7 proposal feature maps, which are then passed to fully connected layers. These layers predict the object category for each proposal by calculating probability scores. The object class with the highest probability is assigned to each proposal. As a result, Faster R-CNN successfully completes object detection by combining high recognition accuracy, efficient detection, and sensitivity. This deep learning framework has been widely applied in tasks such as maritime ship detection, where it significantly enhances detection performance. Fig. 1 illustrates the architecture of the Faster R-CNN algorithm.

B. YOLOv7

YOLOv7, the seventh version of the YOLO family, has gained widespread popularity due to its impressive performance since its release. Developed based on YOLOv4, YOLOv7 has garnered significant attention for its superior speed and accuracy. YOLO models, across all iterations, are single-stage object detectors that take an image as input and utilize a convolutional neural network (CNN) for feature extraction and classification of objects in the image. When compared to its predecessors, YOLOv7 demonstrates its advanced architecture with a range of speed and accuracy rates, varying from 5 FPS to 160 FPS. Fig. 2 illustrates the architecture of the YOLOv7 algorithm, highlighting its enhanced design and functionality. YOLOv7 stands out in real-time object detection tasks with its capabilities, and its architecture is composed of four main components: input, backbone, neck, and head. The image fed into the model is referred to as the input. The backbone layer consists of several convolutional layers, an updated version of the ELAN computation block called E-ELAN (Extended Efficient Layer Aggregation Networks), and a max-pooling layer. E-ELAN combines multiple convolutional blocks to enable the use of more parameters, enhancing feature extraction. The backbone, a pre-trained network, is responsible for extracting features from the input image and includes the SiLU activation function, a widely used function that offers both linear and sigmoidal properties.

The neck layer employs a PAFPN (Path Aggregation Feature Pyramid Network) structure to effectively gather features for more accurate object detection. This layer also

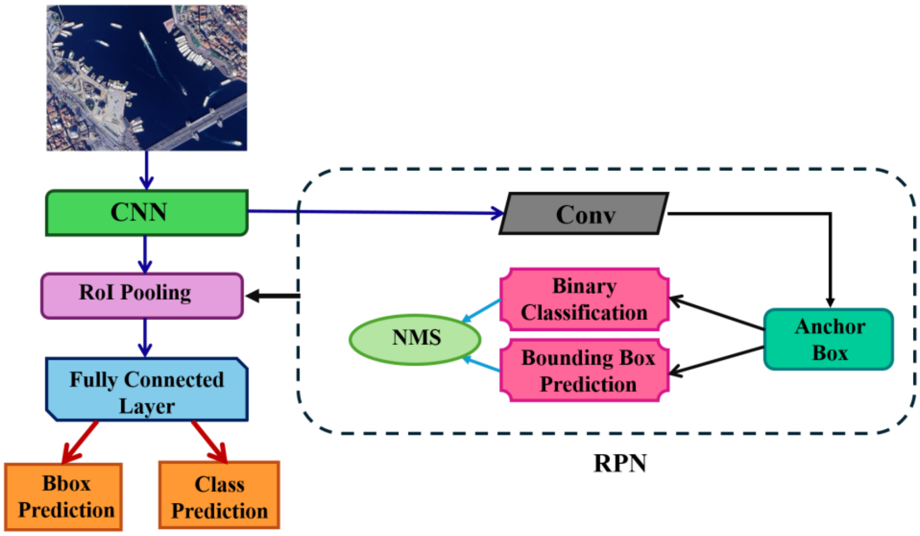

Fig. 1. Faster R-CNN network architecture.

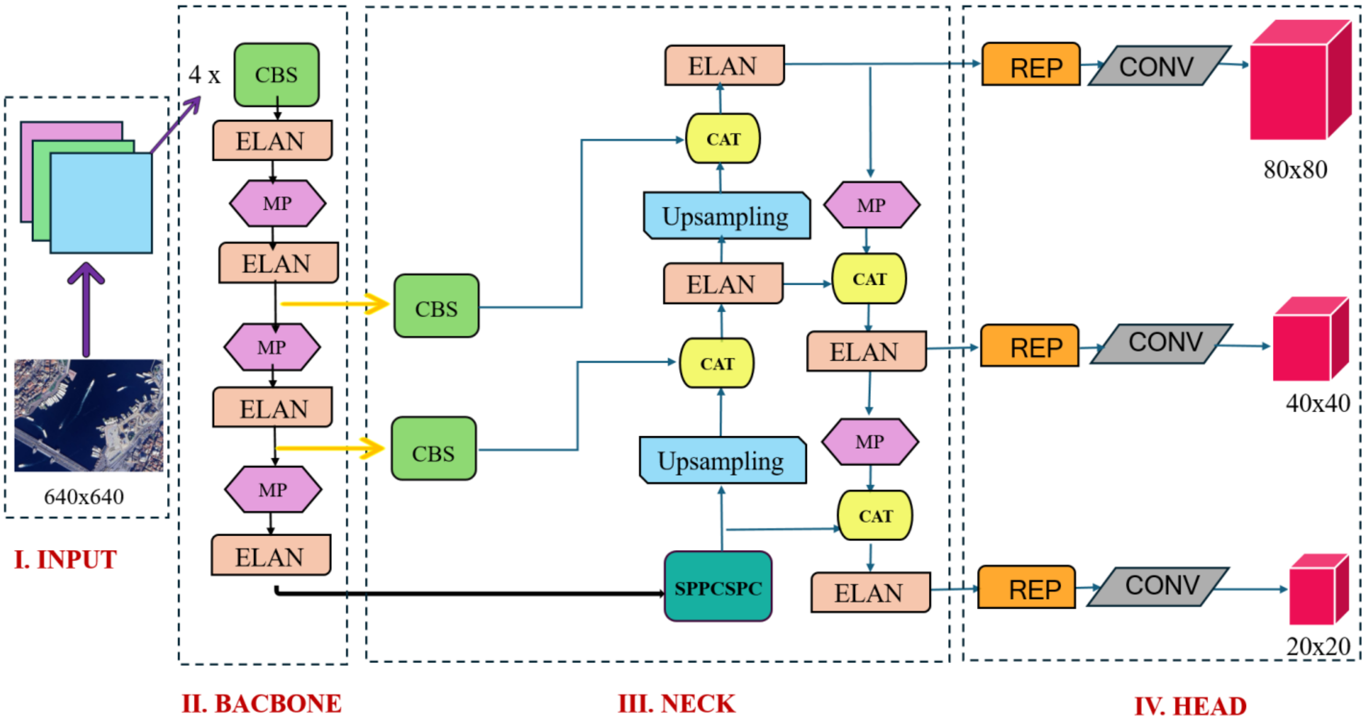

Fig. 2. YOLOv7 network architecture.

includes several convolutional blocks and the SPPCSPC (Spatial Pyramid Pooling with Cross Stage Partial Connections) structure, which integrates convolutional spatial pyramid (CSP) blocks within the spatial pyramid pooling (SPP) framework. This combination deepens the perceptual field of the network for better feature extraction [20]. Upsampling layers are used in the neck for pixel-based classifications. In the CAT (concatenate) stage, feature maps are aggregated to facilitate multi-scale detection. Following the CBS (convolutional block structure), the network performs concatenation to allow the use of additional parameters. Finally, the head component is responsible for generating the final outputs. Here, the convolutional layers take the features extracted in previous steps and create new feature maps. These maps are used to apply bounding boxes, generate object labels, and calculate object detection scores, providing the final detection results [21].

C. Training the Algorithms

The satellite images used to train the Faster R-CNN and YOLOv7 object detectors in this study were sourced from the Google Earth platform. Detecting complex and small-scale objects presents significant challenges, especially in real-time applications. Small objects are particularly difficult for algorithms to detect, often resulting in lower detection success rates. Therefore, it is essential that the dataset includes diverse variations, as the quality of training directly impacts model performance. To address this, ship images of various models and colors were collected under different weather conditions and backgrounds, ensuring a wide color spectrum was represented. Special care was taken to include images with diverse characteristics to enhance the robustness of the models. A comprehensive dataset of 3,694 satellite images, containing a total of 8,592 distinct ship instances, was compiled. The dataset was split into training, validation, and test sets, with 70% allocated for training, 10% for validation, and 20% for testing.

Fig. 3. Examples of ship detection using the Faster R-CNN and YOLOv7 algorithms.

III. Results

In the first image of the detections made using Faster R-CNN in Fig. 3, the detection was challenging due to the cloud partially obscuring the ship on the sea surface. Despite this, the model successfully detected the ship. However, the smaller ship on the right side of the image could not be identified due to the foggy weather and reduced image quality. In Visual 2, the model accurately detected nine ships on the sea surface, framed in bounding boxes, despite the rich mix of blue and green tones in the background. Visual 3 shows three ships near the shore, two of which were correctly detected despite the challenge of adjacent ship placement. However, the ship positioned sideways at the bottom was missed by the model. In Image 4, four ships were successfully framed, and the detection of a small ship between two larger vessels highlights the strengths of Faster R-CNN. However, an error occurred where adjacent ships were incorrectly detected as a single object due to overlapping bounding boxes. Nevertheless, two ships in the upper-left corner were detected effectively.

In the first image tested with YOLOv7, the ship was partially obscured by cloud-induced haze, and the algorithm failed to draw a bounding box covering the entire ship, resulting in poorer performance compared to Faster R-CNN. Additionally, the reduced image quality in the left part of the image prevented the detection of a small ship. In Visual 2, with a sea background in shades of green, the detection of eight ships of varying sizes was successful, though the smallest ship leaving a water trail in the top left corner was missed. Visual 3 shows a large ship and a smaller one next to it, but YOLOv7 mistakenly detected them as a single ship, lagging behind Faster R-CNN's performance in handling adjacent ships. Furthermore, two closely positioned ships at the bottom were not separately detected, highlighting YOLOv7's difficulty in identifying clustered ships. A coastal extension resembling a ship was also incorrectly detected due to remote sensing limitations. In the lower-right corner of Visual 3, YOLOv7 failed to detect a ship that was partially visible, and while the ship oriented differently on the coastline was clear, it was missed, similarly to Faster R-CNN. In Visual 5, YOLOv7 correctly detected four ships aligned along the shore but failed to identify a small ship next to the third one, resulting in an incomplete detection. Moreover, two ships with similar tones on the left were detected as one, and the anchored ship detected by Faster R-CNN on the upper left went unnoticed by YOLOv7 due to its color blending with the shore. These tests demonstrate YOLOv7's weaknesses, especially under challenging conditions, compared to the performance of Faster R-CNN.

Table 1 outlines the performance metrics, showing that the Faster R-CNN algorithm achieves an accuracy of 80.5% in the coastal region, while YOLOv7 follows with 77.9%. In open sea conditions, Faster R-CNN scores an accuracy of 0.713, compared to 0.690 for YOLOv7. Overall, comparing the weighted average precision values, Faster R-CNN records 0.759 and YOLOv7 0.734, demonstrating that Faster R-CNN outperforms YOLOv7 in both regions. In terms of recall, the two algorithms show close results in the coastal region, with Faster R-CNN at 0.758 and YOLOv7 at 0.752, though YOLOv7 slightly underperforms. In the open sea, the recall rates are 0.684 for Faster R-CNN and 0.644 for YOLOv7, indicating neither algorithm has a clear advantage in this scenario. Faster R-CNN also surpasses YOLOv7 in mAP50, with values of 0.623 in the coastal region and 0.558 in the open sea, compared to YOLOv7's 0.586 and 0.515, respectively. Similarly, for mAP75, Faster R-CNN achieves 0.479 in the coastal region and 0.444 in the open sea, whereas YOLOv7 lags behind with 0.458 and 0.393, respectively. These mAP50 and mAP75 scores show that Faster R-CNN provides more accurate predictions than YOLOv7. For the F1 score, YOLOv7 records 0.765 in the coastal region and 0.664 in the open sea, while Faster R-CNN performs better with an F1 score of 78.1% in the coastal region and 0.698 in open sea conditions. These results indicate that Faster R-CNN offers a more robust performance across various metrics, especially in the coastal region.

Table 1. Comparison of performance metrics for ship detection between the Faster R-CNN and YOLOv7 algorithms.

Detector	Location Context	Evaluation Metric					
		Precision	Recall	mAP^{50}	mAP^{75}	F1 Score	Mean FPS
Faster R-CNN	Coastal Region	0.805	0.758	0.623	0.479	0.781	24.48
	Offshore Region	0.713	0.684	0.558	0.444	0.698	21.16
	Weightd Average	0.759	0.721	0.591	0.462	0.739	22.81
YOLOv7	Coastal Region	0.779	0.752	0.586	0.458	0.765	37.39
	Offshore Region	0.690	0.644	0.515	0.393	0.664	31.59
	Weighted Average	0.734	0.698	0.551	0.425	0.716	34.49

The values presented in the table emphasize Faster R-CNN's superior true positive detection rates and classification success in maritime ship detection scenarios, contributing to its overall performance enhancement. Analyzing the FPS values, which indicate the performance of these algorithms in real-time applications, reveals that both YOLOv7 and Faster R-CNN perform better in coastal areas compared to open sea conditions. Specifically, the average FPS values for YOLOv7 and Faster R-CNN are 34.49 and 22.81, respectively. The FPS metric, which reflects the speed of real-time applications and the number of images processed per second, highlights YOLOv7's advantage in processing speed. Although YOLOv7 exhibits lower accuracy rates, it excels in critical situations requiring rapid inference, efficient data processing, and swift decision-making. By evaluating the speed and accuracy parameters in the context of the application scenarios outlined in Table 1, users can choose between YOLOv7, which excels in speed, and Faster R-CNN, which offers greater accuracy in detection.

IV. CONCLUSION

This study provides a comprehensive evaluation of the detection capabilities of Faster R-CNN and YOLOv7 for ship detection from satellite images, highlighting their strengths in handling varying levels of complexity. The detection tests show that Faster R-CNN excels in terms of sensitivity, recall, mAP, and F1 score, making it a suitable choice for scenarios where ships are densely clustered or scattered near the shore. The model demonstrated strong performance even in challenging conditions, such as when background colors closely match the ships or when ships are positioned adjacent to each other. However, its effectiveness diminishes when detecting smaller or partially obscured ships, especially under poor atmospheric conditions that reduce image quality. Conversely, YOLOv7 offers a clear advantage in detection speed, making it more appropriate for real-time or near-real-time applications where rapid processing is essential. While its detection accuracy is slightly lower than that of Faster R-CNN, YOLOv7 remains significantly faster and delivers adequate performance in most scenarios. This trade-off between detection accuracy and inference speed suggests that the choice of algorithm should be based on the specific requirements of the application. Faster R-CNN is ideal for tasks prioritizing high precision and recall, whereas YOLOv7 is more suited for speed-critical applications. Overall, this study underscores the importance of balancing accuracy and efficiency in the development of effective ship detection systems using satellite images. It highlights the need to optimize both elements to ensure robust performance across different detection scenarios.

REFERENCES

[1] S. Wu and L. Zhang, "Using Popular Object Detection Methods for Real Time Forest Fire Detection," 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2018, pp. 280-284. https://doi.org/10.1109/ISCID.2018.00070

[2] M. Bakirci, "Evaluating the impact of unmanned aerial vehicles (UAVs) on air quality management in smart cities: A comprehensive analysis of transportation-related pollution," Computers and Electrical Engineering, vol. 119, no. 109556, 2024. https://doi.org/10.1016/j.compeleceng.2024.109556

[3] Yang, J. Zhang, G. Shi, J. Hu, Y. Wu, "Maneuver decision of UAV in short-range air combat based on deep reinforcement learning," IEEE Access, 8, 363-378, 2019. https://doi.org/10.1109/ACCESS.2019.2961426

[4] Ma Yuehao, Lu Xiao, Dong Pei et al., "UAV fire detection based on image features", Fire Science and Technology, vol. 38, no. 5, pp. 658-660, 2019.

[5] Zheng, X., Chen, F., Lou, L., Cheng, P., & Huang, Y., "Real-Time Detection of Full-Scale Forest Fire Smoke Based on Deep Convolution Neural Network," Remote Sensing, 14(3), 536, 2021. https://doi.org/10.3390/rs14030536

[6] M. Bakirci, I. Bayraktar, I., "Improving coastal and port management in smart cities with UAVs and deep learning," 2024 Mediterranean Smart Cities Conference (MSCC), pp. 1-6, Martil - Tetuan, Morocco, 2024. https://doi.org/10.1109/MSCC62288.2024.10697069

[7] Tan, M., Pang, R., & Le, Q. V., "Efficientdet: scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition," pp. 10781–10790, 2020.

[8] Girshick, R. B., Donahue, J., Darrell, T., & Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587, 2014.

[9] Girshick, R. B., "Fast R-CNN," In Proceedings of the IEEE international conference on computer vision, pp. 1440–1448, 2015.

[10] Ren, S., He, K., Girshick, R. B., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, et al. (Eds.), Proceedings of the 29th international conference on neural information processing systems (pp. 91–99). Red Hook: Curran Associates. https://doi.org/10.1109/TPAMI.2016.2577031

[11] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C.-Y., et al. (2016). SSD: single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, et al. (Eds.), Proceedings of the 14th European conference on computer vision (pp. 21–37). Cham: Springer. https://doi.org/10.1007/978-3-319-46448-0_2

[12] Jeong, J., Park, H., & Kwak, N. (2017). Enhancement of SSD by concatenating feature maps for object detection. In Proceedings of the British machine vision conference (pp. 1–12). https://doi.org/10.48550/arXiv.1705.09587

[13] M. Bakirci, P. Dmytrovych, I. Bayraktar, O. Anatoliyovych, "Multi-Class Vehicle Detection and Classification with YOLO11 on UAV-

Captured Aerial Imagery," 2024 IEEE 7th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD), Kyiv, Ukraine, 2024, pp. 191-196. https://doi.org/10.1109/APUAVD64488.2024.10765862

[14] Redmon, J., Divvala, S. K., Girshick, R. B., & Farhadi, A (2016). You only look once: unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779–788).

[15] Song, Y., & Fu, Z. (2018). Uncertain multivariable regression model. Soft Computing, 22(17), 5861–5866. https://doi.org/10.1007/s00500-018-3324-5

[16] M. Bakirci, "A novel swarm unmanned aerial vehicle system: Incorporating autonomous flight, real-time object detection, and coordinated intelligence for enhanced performance," Traitement du Signal, vol. 40, no. 5, pp. 2063-2078, 2023. https://doi.org/10.18280/ts.400524

[17] Bochkovskiy, A., Wang, C., & Liao, H. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. ArXiv. /abs/2004.10934

[18] Wu, D., Lv, S., Jiang, M., & Song, H. (2020). Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers and Electronics in Agriculture, 178, 105742. https://doi.org/10.1016/j.compag.2020.105742

[19] M. Bakirci, "Real-time vehicle detection using YOLOv8-nano for intelligent transportation systems," Traitement du Signal, vol. 41, no. 4, pp. 1727-1740, 2024. https://doi.org/10.18280/ts.410407

[20] Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2024). Yolov9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616.