

# Comparison of YOLOv5 and YOLOv6 Models for Plant Leaf Disease Detection

Ecem Iren

Izmir Kavram Vocational School, Turkiye

ecem.iren@kavram.edu.tr (corresponding author)

Received: 7 February 2024 | Revised: 7 March 2024 | Accepted: 10 March 2024

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: <https://doi.org/10.48084/etasr.7033>

## ABSTRACT

Deep learning is a concept of artificial neural networks and a subset of machine learning. It deals with algorithms that train and process datasets to make inferences for future samples, imitating the human process of learning from experiences. In this study, the YOLOv5 and YOLOv6 object detection models were compared on a plant dataset in terms of accuracy and time metrics. Each model was trained to obtain specific results in terms of mean Average Precision (mAP) and training time. There was no considerable difference in mAP between both models, as their results were close. YOLOv5, having 63.5% mAP, slightly outperformed YOLOv6, while YOLOv6, having 49.6% mAP<sub>50-95</sub>, was better in detection than YOLOv5. Furthermore, YOLOv5 trained data in a shorter time than YOLOv6, since it has fewer parameters.

**Keywords**-deep learning; YOLOv5; YOLOv6; plant leaf disease detection; convolutional neural networks

## I. INTRODUCTION

The agricultural process plays a vital role in the food supply, while the farming sector contributes to the employment [1]. Plant diseases are a significant issue since they can cause huge economic losses. Many factors lead to such infection types, such as viruses, bacteria, and fungi [2]. Farmers or experts can monitor diseases in traditional ways with some experience or training, but this method can be considered challenging, expensive, and open to mistakes. A disease can be recognized and examined at an early stage if the farmer regularly controls the crops and has sufficient information. As it is difficult to manage larger fields, a fast, healthy, and accurate disease detection system is strongly needed in cases where human evaluation is unreliable or not enough. Early detection systems can reduce large-scale crop losses by preventing the spread of diseases [2-4]. Image processing, machine learning, and deep learning methods can be used to examine infected regions of plants by predicting the class of a specific disease [5]. Machine learning algorithms try to explore hidden insights and complex patterns of a given dataset and are generally preferred for tasks related to classification, regression, and clustering. Artificial Neural Networks (ANNs) operate according to the principle of information processing in biological systems. Each processing unit is an artificial neuron that connects to others and is represented by mathematical concepts. Each connection between neurons sends signals simulating the synapses in the brain. If a specific signal ensures a certain threshold determined by an activation function, it can be processed by subsequent neurons. Typically, neurons are organized into networks with different layers. An input layer usually receives the data input, and the output layer produces the ultimate result. Deep neural networks have more than one hidden layer, usually containing complex neurons, and employ

high-level operations using multiple activation functions instead of a single neuron [6-7].

Object detection is a research area in deep learning and is more complicated than object classification. Object classification does not provide information on the location of the object in the image. Object detection can be divided into two different categories, one- and two-stage. In two-stage detection, the region proposal is generated first, and then the classification task is performed. Faster R-CNNs generally apply this mechanism to perform detection, but it cannot work in images with more than one object [8]. However, the YOLO models perform the detection process by combining two operations called classification and localization. YOLO is a one-stage detector, which means that regional proposal and classification are performed simultaneously. YOLO divides an image into a grid of cells. Each grid cell predicts the class of an object and specifies the bounding box determining the object's location. Each bounding box can be described using four descriptors: the center of a bounding box (bx, by), width (bw), height (bh), and object class (c). Non-max suppression is a common algorithm used for discarding bounding boxes when multiple boxes are predicted for the same object [9-10].

Many studies have been conducted by applying YOLO models. In [11], YOLOv7, YOLOv5, YOLO-X, and YOLO-R were compared in terms of Frames Per Second (FPS) and accuracy metrics, showing that YOLOR-p6 had the best results with 23 FPS, followed by YOLO-Xm and YOLOR6-p6 with 23 and 22 FPS, respectively, while YOLOv7 had the lowest speed of only 17 FPS. YOLOv7 showed the highest mean Average Precision (mAP) of approximately 60%, followed by YOLO-p6, YOLOv5, and YOLO-Xm with 56, 53, and 46%, respectively. In [12], a performance comparison was made between YOLOv3, YOLOv4, and YOLOv5 for poultry

recognition. The dataset was split into training and testing parts with a ratio of 80% and 20%, respectively. After the manual annotation process, different YOLO versions were trained. The experimental results showed that YOLOv5x achieved the highest accuracy of 99.5%, while the YOLOv4-tiny model had the lowest mAP and the least training speed.

In [13], YOLOv5 and YOLOv8 were compared on aerial image data taken from the Roboflow pedestrian dataset. Training and validation tasks were performed with 100 epochs using the Google Colab environment, showing that YOLOv8 had better precision and F1-score than YOLOv5, by 2.82% and 0.98%, respectively. In [14], a safety helmet detection system was developed based on YOLOv5. 6045 images were collected and annotated, and then some operations such as random flip, geometric and illumination distortion, random erase, cutout, and mix-up were used for data augmentation. Four YOLOv5 models were compared, showing that without pre-trained weights, YOLOv5x obtained the best mAP value of 93.6%, which was only 0.1% higher than YOLOv5l. The performance of all YOLOv5 models improved with pre-trained weights (transfer learning), and while YOLOv5s had the highest improvement rate with 1.3%, YOLOv5x was the leader in terms of accuracy.

In [15], the YOLOv5 and DETR models were compared in detecting sea cucumbers. The results showed that YOLOv5 outperformed DETR in terms of accuracy and computing resources. Authors in [13] aimed to find the most suitable hyperparameters for the detection of healthy and diseased tomato leaves, using private and public datasets, showing that YOLOv5 reached an accuracy rate of 93% during the evaluation of the test data. In [16], a YOLOv5 model with a special configuration of hyperparameters, such as learning rate and batch size, and transfer learning provided an mAP value of 60.9%. This study compares the detection performance of the YOLOv5 and YOLOv6 models on a leaf disease dataset.

## II. MATERIALS AND METHODS

This study compares the object detection performance of the YOLOv5 and YOLOv6 models on a data set that contains different types of leaf diseases. Figure 1 illustrates each step of the detection process. At first, a dataset was obtained and split into training and validation parts, and then each detection model was trained to obtain specific metric results.

### A. Dataset

The PlantDoc [17] dataset was developed at the Indian Institute of Technology and is shared on the GitHub platform. PlantDoc has 2,569 images, including 13 plant species and 30 classes as diseased and healthy for image classification and object detection purposes. The data set has 8851 annotations and was divided into training and validation parts in a 90:10 ratio. Therefore, 2328 images were used for training and 239 images were used for validation.

### B. YOLOv5 Object Detection Model

YOLOv5 has four different scale variations, named S, M, L, and X that represent Small, Medium, Large, and Extra Large, respectively. These scales contain different multipliers in terms of depth and width, while the structure of the model

remains constant. Only the complexity and size of the objects are scaled in each variation [14]. YOLOv5 is an object detection framework that has been established in the CSPDarknet53 and PyTorch frameworks. CSPDarknet53 provides a backbone architecture that consists of a Focus structure and a CSP network and aggregates image features to achieve feature extraction. The Focus layer is used to decrease the Compute Unified Device Architecture (CUDA) memory, layers, and parameters and improve the backward and forward speed. Aggregated features are directed to the Neck (PANet) structure, which includes a new Feature Pyramid Network (FPN) with many bottom-up and top-down layers. Low-level features can be obtained with the help of FPN, and lower layers improve the localization accuracy of the object. In the Head block, features from the Neck are used to detect classes in terms of class, location, confidence score, and size [18-20]. Figure 2 shows the YOLOv5 architecture.

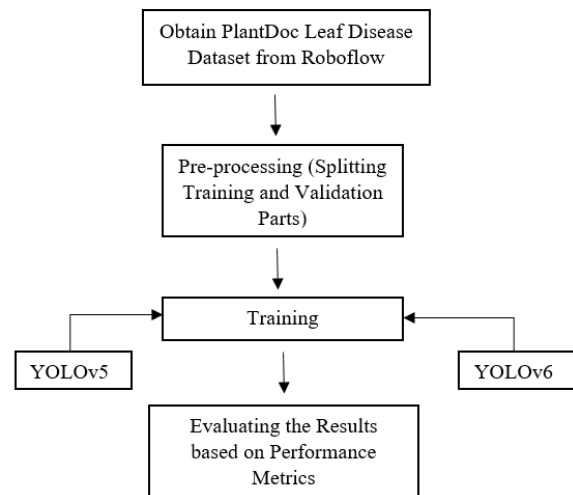


Fig. 1. Method's flow diagram.

### C. YOLOv6 Object Detection Model

YOLOv6 shows better performance than YOLOv5 in terms of detection accuracy and inference speed, making it an efficient option for industrial applications. YOLOv6 has been designed with many improvements in the Backbone, Neck, and Head blocks. The Neck and Backbone structures have been replaced with Rep-PAN and EfficientRep. Since the design of the backbone network plays an important role in detection effectiveness, a RepBlock with RepVGG is preferred since it offers a re-parameterizable structure. A multi-branch topology is used during the training process, and then RepBlock is converted to RepConv stacks of 3×3 convolutional layers with ReLU activation functions during the inference stage. EfficientRep Backbone supports hardware such as GPU and CPU. Rep-Pan Neck is considered more precise and operates faster than PANet and SPP. A convolutional layer has been added between the network and the final Head to improve performance, using processing power and memory [21-22]. Figure 3 shows the architecture of YOLOv6.

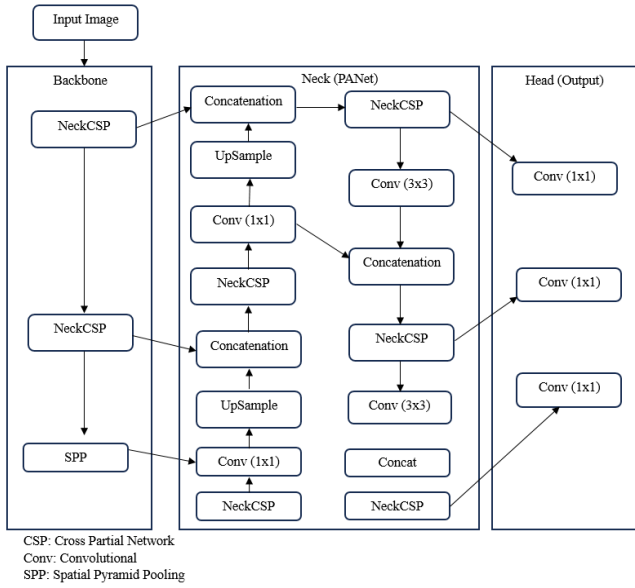


Fig. 2. YOLOv5 architecture.

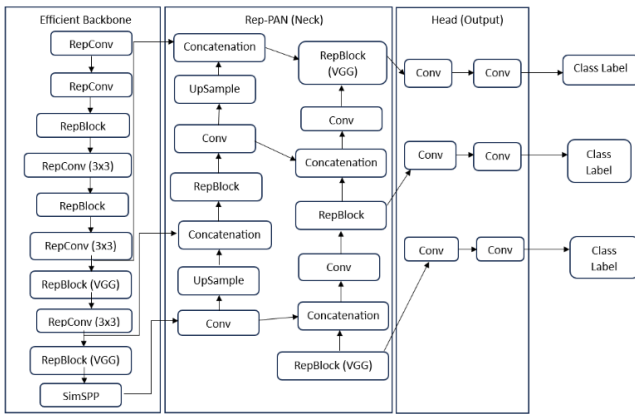


Fig. 3. YOLOv6 architecture.

D. Environment and HyperParameters

Each model was trained in the Google Colab environment, which provides a service with powerful GPUs and TPUs for machine and deep learning problems. Python and NVIDIA Tesla T4 GPU were used for the experiments. Table I shows the hyperparameter configurations of the models, which were the same for each model for a fair comparison. Epochs are the total iterations of a training task. The learning rate determines the change level of the model based on the predicted error when updating the model weights. Batch size defines the number of samples for one epoch during training, and the optimization algorithm adjusts the weights to minimize error. Lastly, momentum deals with optimizing convergence.

TABLE I. HYPERPARAMETERS OF YOLO MODELS

Parameter	Value
Epochs	100
Learning rate	0.01
Batch size	16
Optimization algorithm	SGD (Stochastic Gradient Descent)
Momentum	0.937

III. RESULTS AND DISCUSSION

True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) were used to evaluate the results of the experiments, which are generally considered indicators of detection performance. TP refers to the number of accurate detections of diseased leaves, FP refers to the number of objects that were incorrectly identified as a diseased leaf, TN represents the number of negative samples with a negative prediction, and FN is the number of diseased leaves that were missed. Precision (P) is the rate of the total number of actual positives to all predicted positives (1), and Recall (R) refers to how many samples that had to be predicted as positive were correctly estimated as positive (2). The Average Precision (AP) balances precision and recall by taking into account both FP and FN (3). The area under the precision-recall curve is calculated to find the AP for each class, while mAP is the average of the precision-recall curve of each object class (4) [13, 23], and it is a well-known metric for measuring the accuracy of detection models.

$$\text{Precision (P)} = \frac{TP}{TP+FP} \tag{1}$$

$$\text{Recall (R)} = \frac{TP}{TP+FN} \tag{2}$$

$$\text{Average Precision (AP)} = \int_0^1 P(r) dr \tag{3}$$

$$mAP = \frac{1}{n} \sum_1^n AP_i \tag{4}$$

The mAP50 metric represents the mean average precision in an IoU at the threshold of 0.5, while mAP50-95 is a term for the mean average precision, calculated as the average of 10 AP values for 10 IoU thresholds between 0.5 and 0.95 confidence levels [23-24]. Therefore, it is a way to examine several confidence values to assess the accuracy of the detected objects. Table II shows the accuracy, precision, recall, and training duration for each YOLO model. YOLOv5 showed a slightly better result than YOLOv6, having mAP50 of 63.5% and 62%, respectively. When examining mAP50-95, YOLOv6 had a higher performance (49.6%) than YOLOv5 (48.8%). However, there is not a huge difference between the two models when analyzing the mAP metrics, as the results were close to each other. On the other hand, YOLOv6 needed much more time to train than YOLOv5. The training process for the YOLOv6 model took nearly 2.3 hours, while the YOLOv5 training process took approximately 0.8 hours. More complex models train more parameters, leading to an increased training time. The mAP graphs can be seen in Figures 4-6.

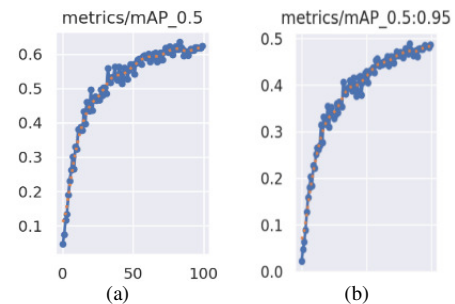


Fig. 4. YOLOv5 graphs: (a) mAP, (b) map50-95.

TABLE II. PERFORMANCE METRICS

Model	mAP50	mAP50-0.95	Precision	Recall	Duration
YOLOv5	63.5%	48.8%	64.8%	54.8%	0.772
YOLOv6	62%	49.6%	56.6%	62.4%	2.282

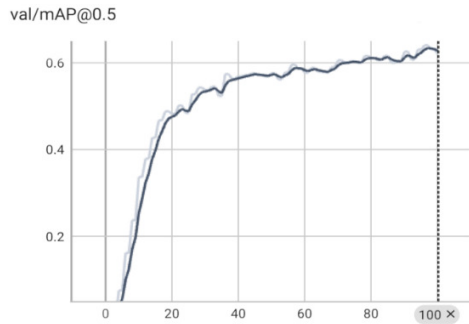


Fig. 5. mAP graph of YOLOv6.

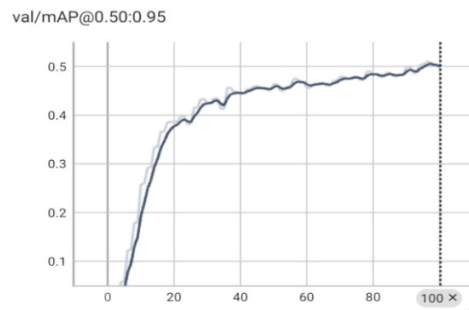


Fig. 6. mAP50-95 graph of YOLOv6.

The precision of YOLOv5 was 64.8%, which was better than YOLOv6 (56.6%). The recall of YOLOv6 (62.4%) was found to be greater than that of YOLOv5 (54.8%). Figures 7 and 8 show the precision-recall curves of each model. Figures 9-12 show the detection of some validation samples with their class categories and confidence levels for both models. In Figure 9, cherry and blueberry leaves were estimated with confidence levels of 70, 50, and 80%, respectively, using YOLOv5. Likewise, corn leaf blight disease was detected with an 80% confidence level using the same algorithm, as shown in Figure 10. Additionally, YOLOv6 detected squash powdery mildew leaf disease and apple leaf with 77% and 91% confidence levels, respectively, as shown in Figures 11 and 12.

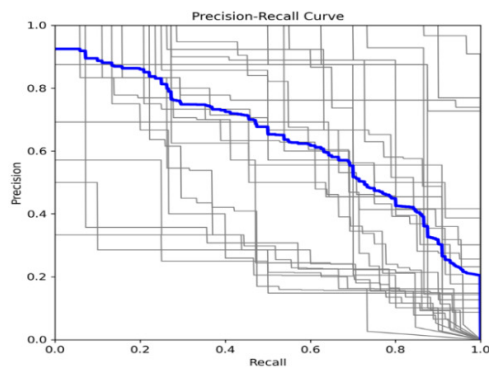


Fig. 7. Precision-recall curve of YOLOv5.

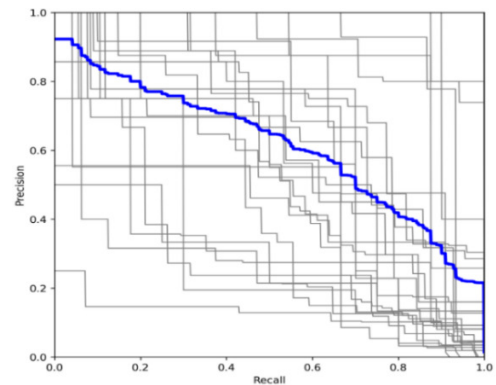


Fig. 8. Precision-recall curve of YOLOv6.

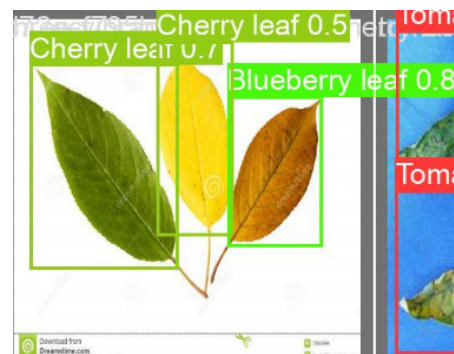


Fig. 9. YOLOv5 detection of cherry and blueberry leaves with confidence levels.



Fig. 10. YOLOv5 detection of corn leaf blight with confidence levels.

Other studies that compared YOLOv5 and YOLOv6 in different datasets showed that mAP values can vary according to different parameters such as class category distribution, total number of class categories, hyperparameters, and augmentation methods. In [25], YOLO models were trained on a dataset containing road cracks and potholes, reaching mAP50 values of 77 and 68.11% for YOLOv5-s and YOLOv6-s, respectively. In [26], YOLOv5 and YOLOv6 were compared on the COCO dataset with 30 class categories, and YOLOv5-m had a mAP of 9.28%, followed by YOLOv6-m with 8.51%. In [24], two models were compared in an oil tank dataset, showing mAP50-95 of 69.69 and 65.60% for YOLOv5-x and YOLOv6-l, respectively. In [21], the same models were compared on a lunar crater dataset, where OLOv5 achieved 72% mAP and YOLOv6 had 62% with SGD optimization.



Fig. 11. Detection of squash powdery mildew leaf with confidence level by YOLOv6.



Fig. 12. Detection of apple leaf with confidence level by YOLOv6.

#### IV. CONCLUSION

This study carried out a comparative analysis of YOLOv5 and YOLOv6 object detection models on a plant leaf disease dataset to evaluate their detection and time performance. The results showed that there were no great differences between the detection performance of each model according to the mAP metric. If saving time is significant, the YOLOv5 model can be chosen since it is trained in a shorter time compared to YOLOv6. In the future, the hyperparameters of these models can be tuned by changing the learning rate, batch size, optimization method, and momentum. Furthermore, advanced YOLO versions, such as YOLOv7 and YOLOv8, can be studied with the same dataset to discover and compare their performance results.

#### REFERENCES

- [1] L. Christiaensen, Z. J. Rutledge, and J. E. Taylor, "The Future of Work in Agriculture: Some Reflections," The World Bank, Policy Research Working Paper 9193, Mar. 2020.
- [2] S. Nigam and R. Jain, "Plant disease identification using Deep Learning: A review," *Indian Journal of Agricultural Sciences*, vol. 90, no. 2, pp. 249–257, Mar. 2020, <https://doi.org/10.56093/ijas.v90i2.98996>.
- [3] R. Rajamohan and B. C. Latha, "An Optimized YOLO v5 Model for Tomato Leaf Disease Classification with Field Dataset," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12033–12038, Dec. 2023, <https://doi.org/10.48084/etasr.6377>.
- [4] N. Shelar, S. Shinde, S. Sawant, S. Dhumal, and K. Fakir, "Plant Disease Detection Using CNN," *ITM Web of Conferences*, vol. 44, 2022, Art. no. 03049, <https://doi.org/10.1051/itmconf/20224403049>.
- [5] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Information Processing in Agriculture*, vol. 4, no. 1, pp. 41–49, Mar. 2017, <https://doi.org/10.1016/j.inpa.2016.10.005>.
- [6] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, <https://doi.org/10.1007/s12525-021-00475-2>.
- [7] J. F. Mas and J. J. Flores, "The application of artificial neural networks to the analysis of remotely sensed data," *International Journal of Remote Sensing*, vol. 29, no. 3, pp. 617–663, Feb. 2008, <https://doi.org/10.1080/01431160701352154>.
- [8] J. Kaur and W. Singh, "Tools, techniques, datasets and application areas for object detection in an image: a review," *Multimedia Tools and Applications*, vol. 81, no. 27, pp. 38297–38351, Nov. 2022, <https://doi.org/10.1007/s11042-022-13153-y>.
- [9] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data," *Remote Sensing*, vol. 13, no. 1, Jan. 2021, Art. no. 89, <https://doi.org/10.3390/rs13010089>.
- [10] S. Norkobil Saydirasulovich, A. Abdusalomov, M. K. Jamil, R. Nasimov, D. Kozhamzharova, and Y.-I. Cho, "A YOLOv6-Based Improved Fire Detection Approach for Smart City Environments," *Sensors*, vol. 23, no. 6, Jan. 2023, Art. no. 3161, <https://doi.org/10.3390/s23063161>.
- [11] M. R. M. Ismat Saira Gillani1, "Yolov5, Yolo-x, Yolo-r, Yolov7 Performance Comparison: A Survey," *CS & IT Conference Proceedings*, vol. 12, no. 16, Sep. 2022.
- [12] O. Kivrak and M. Z. Gürbüz, "Performance Comparison of YOLOv3, YOLOv4 and YOLOv5 algorithms: A Case Study for Poultry Recognition," *Avrupa Bilim ve Teknoloji Dergisi*, no. 38, pp. 392–397, Aug. 2022, <https://doi.org/10.31590/ejosat.1111288>.
- [13] I. P. Sary, S. Andromeda, and E. U. Armin, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images," *Ultima Computing: Jurnal Sistem Komputer*, vol. 15, no. 1, pp. 8–13, Jun. 2023, <https://doi.org/10.31937/sk.v15i1.3204>.
- [14] F. Zhou, H. Zhao, and Z. Nie, "Safety Helmet Detection Based on YOLOv5," in *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, Shenyang, China, Jan. 2021, pp. 6–11, <https://doi.org/10.1109/ICPECA51329.2021.9362711>.
- [15] X. Yuan *et al.*, "Performance Comparison of Sea Cucumber Detection by the Yolov5 and DETR Approach," *Journal of Marine Science and Engineering*, vol. 11, no. 11, Nov. 2023, Art. no. 2043, <https://doi.org/10.3390/jmse11112043>.
- [16] T. Saidani, "Deep Learning Approach: YOLOv5-based Custom Object Detection," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12158–12163, Dec. 2023, <https://doi.org/10.48084/etasr.6397>.
- [17] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A Dataset for Visual Plant Disease Detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, Hyderabad, India, Jan. 2020, pp. 249–253, <https://doi.org/10.1145/3371158.3371196>.
- [18] O. E. Olorunshola, M. E. Irhebhude, and A. E. Ewwiekpaefe, "A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms," *Journal of Computing and Social Informatics*, vol. 2, no. 1, pp. 1–12, Feb. 2023, <https://doi.org/10.33736/jcsi.5070.2023>.
- [19] H. K. Jung and G. S. Choi, "Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions," *Applied Sciences*, vol. 12, no. 14, Jan. 2022, Art. no. 7255, <https://doi.org/10.3390/app12147255>.
- [20] T. Saidani, R. Ghodhbani, A. Alhomoud, A. Alshammari, H. Zayani, and M. B. Ammar, "Hardware Acceleration for Object Detection using YOLOv5 Deep Learning Algorithm on Xilinx Zynq FPGA Platform," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 13066–13071, Feb. 2024, <https://doi.org/10.48084/etasr.6761>.
- [21] N. Aburaed, M. Alsaad, S. A. Mansoori, and H. Al-Ahmad, "A Study on the Autonomous Detection of Impact Craters," in *Artificial Neural*

- Networks in Pattern Recognition*, Dubai, United Arab Emirates, 2023, pp. 181–194, [https://doi.org/10.1007/978-3-031-20650-4\\_15](https://doi.org/10.1007/978-3-031-20650-4_15).
- [22] C. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications." arXiv, Sep. 07, 2022, <https://doi.org/10.48550/arXiv.2209.02976>.
- [23] T. Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, Zurich, Switzerland, 2014, pp. 740–755, [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48).
- [24] T. Bakirman, "An Assessment of YOLO Architectures for Oil Tank Detection from SPOT Imagery," *International Journal of Environment and Geoinformatics*, vol. 10, no. 1, pp. 9–15, Mar. 2023, <https://doi.org/10.30897/ijegeo.1196817>.
- [25] N. I. M. Yusof, A. Sophian, H. F. M. Zaki, A. A. Bawono, A. H. Embong, and A. Ashraf, "Assessing the performance of YOLOv5, YOLOv6, and YOLOv7 in road defect detection and classification: a comparative study," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 350–360, Feb. 2024, <https://doi.org/10.11591/eei.v13i1.6317>.
- [26] M. Horvat, L. Jelečević, and G. Gledec, "Comparative Analysis of YOLOv5 and YOLOv6 Models Performance for Object Classification on Open Infrastructure: Insights and Recommendations," presented at the 34th Central European Conference on Information and Intelligent Systems, Zagreb, Croatia, 2023, pp. 317–324.