ICIC Express Letters
Part B: Applications
Volume **15**, Number **2**, February **2024**

ICIC International ©2024 ISSN 2185-2766
pp. 139–146

INDOOR OBJECT DETECTION COMPARISON USING YOLOV8, NANODET-PLUS, AND DETECTION TRANSFORMER MODEL

Arfa Shaha Syahrulfath, Andi Dharmawan*, Ika Candradewi and Jazi Eko Istiyanto

Department of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
Bulaksumur 21, Yogyakarta 55281, Indonesia
arfa.shaha@mail.ugm.ac.id; { ika.candradewi; jazi }@ugm.ac.id
*Corresponding author: andi_dharmawan@ugm.ac.id

Received May 2023; accepted August 2023

Abstract. *The development of object detection algorithms keeps on moving at a faster speed every year. This occurence is unique because a robust object detection method will undoubtedly play a significant role in various implementations such as mobile robots and humanoid robots, especially for indoor objects, as the robot has more narrow and limited space to move. In addition to that, the poses of the camera and entities continue to change following the robot's motion so that the detection becomes less accurate. Therefore, the prerequisite for creating such robots is detecting the correct object at the right place. Thus, robust object detection is required. This research compared some of the newest object detection algorithms, specifically YOLOv8, Nanodet-Plus, and Detection Transformer (DETR). All methods are trained under the same number of epochs, which is 20. Moreover, each method also used pre-trained models, namely YOLOv8x, NanoDet-Plus-m-1.5x-416, and detr_resnet50 in the same number of datasets, namely 2213 images of objects in a room divided into eight classes. The experiment results show that YOLOv8 has a slight advantage with achieving mAP50 or a mean Average Precision calculated at Intersection over Union (IoU) 0.5 of 99.1% compared to Nanodet-Plus with 92.9% and Detection Transformer with 93.2%.*

Keywords: Object detection, Indoor object, YOLOv8, Nanodet-Plus, Detection Transformer

1. Introduction.

Due to its diverse application, object detection has continuously become a potential research topic recently [1]. Object detection is one of the techniques in computer vision to detect the semantic objects of a category in images or videos. Object detection aims to create a prediction for each object of interest and label it corresponding to its class [2]. To gain a better image understanding, we need to precisely estimate the concepts and locations of the objects within the image. Thus, object detection is required [3]. One of the object detection implementations is to detect objects inside a room which can be applied to indoor robot assistants. The detection of objects inside a room often experiences problems when implemented on the robot since robots must be able to move and interact with the real-world environment [5]. Therefore, the poses of the camera and objects will continuously change following the robot's motion, resulting in less accurate detection. Furthermore, indoor places are usually narrower and have limited space for robots to move. Hence, an accurate indoor object detection algorithm is needed.

Amongst some newest methods in the last three years, we compared three methods in this research. They are YOLOv8, Nanodet-Plus, and Detection Transformer. YOLO, or You Only Look Once, is an excellent approach to detecting objects in multiple bounding

DOI: 10.24507/icicelb.15.02.139

boxes and class probabilities for those boxes [6]. It passes the image through the CNN algorithm, which will express and extract features of the input data mathematically [7]. YOLOv8 [12], developed by Ultralytics, is the newest state-of-the-art YOLO model that can be used for object detection, image classification, and instance segmentation tasks. Next, we used Nanodet-Plus, a development from Nanodet [11], an FCOS-style [8] one-stage anchor-free object detection model that uses generalized focal loss as classification and regression loss. FCOS utilizes spatial and scale constraints to select samples [9] and solves the problem of overlaps within the ground-truth labels. The last method we compared is DETR or Detection Transformer, which predicts all objects at once and is trained end-to-end with a set loss function that performs bipartite matching between predicted and ground-truth objects [2]. By dropping multiple components that encode prior knowledge, DETR simplifies the detection pipeline [2].

This research uses the same indoor object datasets to detect indoor objects with YOLOv8, Nanodet-Plus, and Detection Transformer. In addition, this study will also focus on the steps to determine which of the chosen methods has the most optimal performance. The bounding box loss curve and the Average Precision (AP) score, formulated from the precision and recall value, will be used as an evaluation metric in this experiment. Based on the experiment on the three methods, the result shows that the methods are proven valid for indoor object detection with more than 90% mAP50 achieved. The rest of the paper is organized as follows. Section 2 describes the architecture details of each model and training procedure. Section 3 describes and analyzes the results of the experiment. Finally, Section 4 presents the conclusion of this paper.

2. Research Methodology.

In this section, we will present the architecture of each model, the dataset used, dataset annotation, and the training procedure to produce the necessary model for every proposed system.

2.1. YOLOv8.

The YOLOv8 flowchart is shown in Figure 1. The YOLOv8 model used in this experiment is the largest one, which is YOLOv8x. Therefore, this model has the most network size compared to the other YOLOv8 model and will perform better at the cost of longer training processing times. The detailed architecture of YOLOv8x used in this experiment can be seen in Figure 2. YOLOv8 can be trained using a Command Line Interface, making training the model more intuitive. In addition to that, this CLI is an addition to the Python package that provides a more seamless coding experience than the previous model.

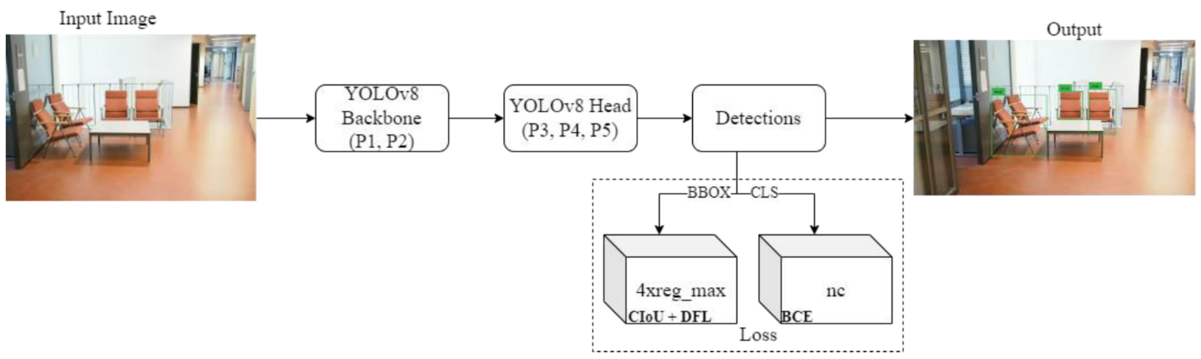


Figure 1. YOLOv8 flowchart

The YOLOv8 model which we used is the YOLOv8x model, so the depth multiple (d) is 1.00, the width multiple (w) is 1.25, and the ratio (r) is 1.00 according to the details of the architecture [12]. Since YOLOv8 is the improvement of YOLOv5, the core concept of how the method works still shares a similar mechanism. This algorithm divides the image into regions and computes bounding boxes and probabilities for each area used for bounding

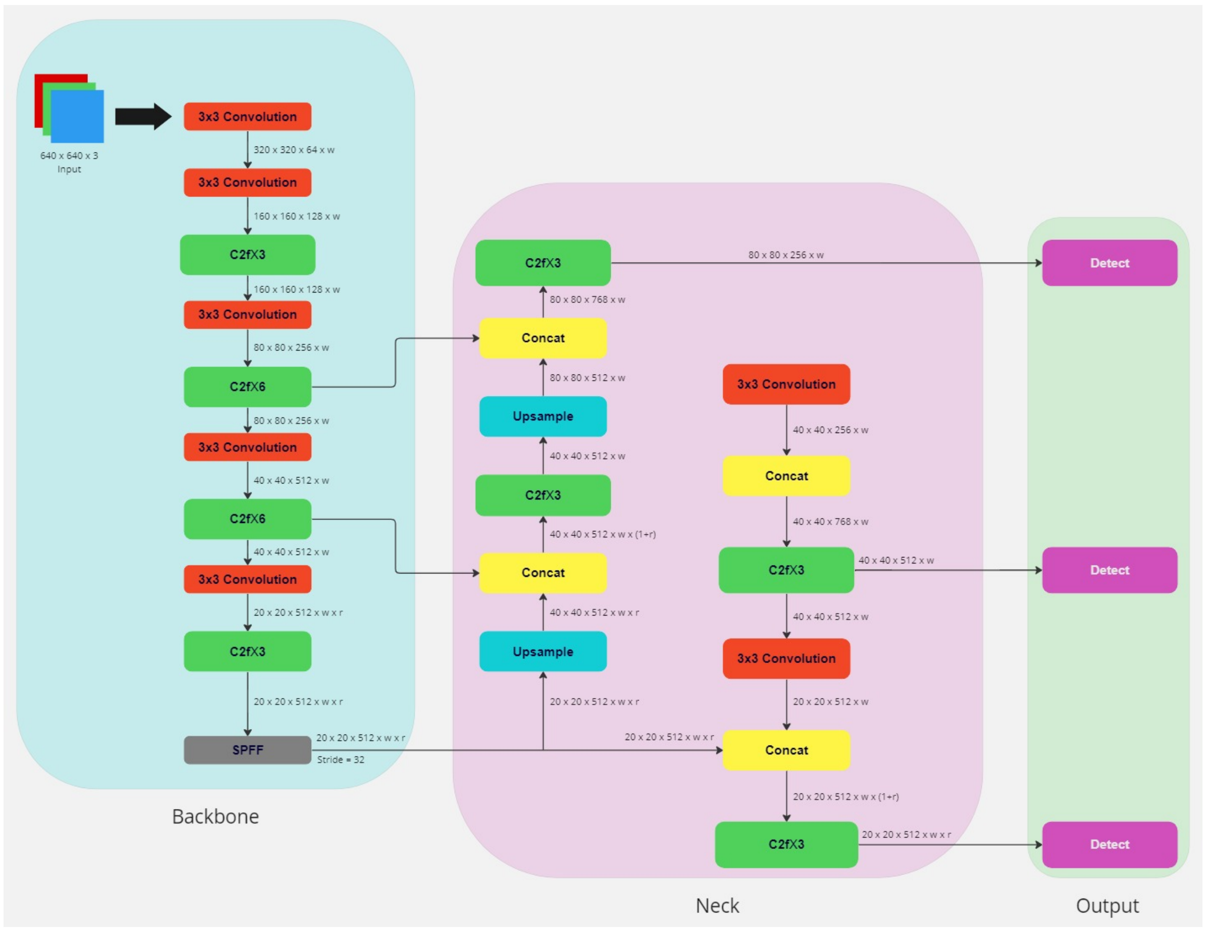


FIGURE 2. YOLOv8 architecture

box weighting. The prediction module used a combination of Generalized Intersection over Union (GIoU) as the loss function of the bounding box frame [4] and weighted Non-Maximum Suppression (NMS) to make the network achieve better convergence [13].

The architecture changes from YOLOv8 compared to YOLOv5 start from the C2f module. Previously on YOLOv5, it was the C3 module. In C3, only the output of the last Bottleneck was used, but in C2f, all the outputs from the bottleneck are concatenated. Then, in the backbone, YOLOv5 used a 6×6 conv, while YOLOv8 changed the first 6×6 conv to a 3×3 conv. Furthermore, two convs in YOLOv5 Config get deleted in YOLOv8. In bottleneck, YOLOv8 also replaces YOLOv5's first 1×1 conv with a 3×3 conv. Lastly, YOLOv8 deletes the objectness branch in YOLOv5 and uses decoupled head. There are two other new updates in YOLOv8. It removes the usage of mosaic augmentation and directly predicts an object's center instead of using the offset from anchor boxes.

2.2. **Nanodet-Plus.** The flowchart and architecture visualization of Nanodet-Plus, which we used in this experiment, is shown in Figure 3. Unlike the previous Nanodet version, Nanodet-Plus designed a lighter and simpler training auxiliary module named Assign Guidance Module (AGM). It also worked with the dynamic soft label assignment strategy Dynamic Smooth Label Assignment (DSLA) to solve the optimal label matching problem in lightweight models.

The AGM or Assistant Guidance Module consists of only four 3×3 convolutions that use Group Normalization (GN) as the Normalize Layer and share parameters among feature maps of different scales. The matching cost will be calculated from the classification probability and detection frame predicted by AGM and sent to the Dynamic Smooth Label Assignment (DSLA) module.

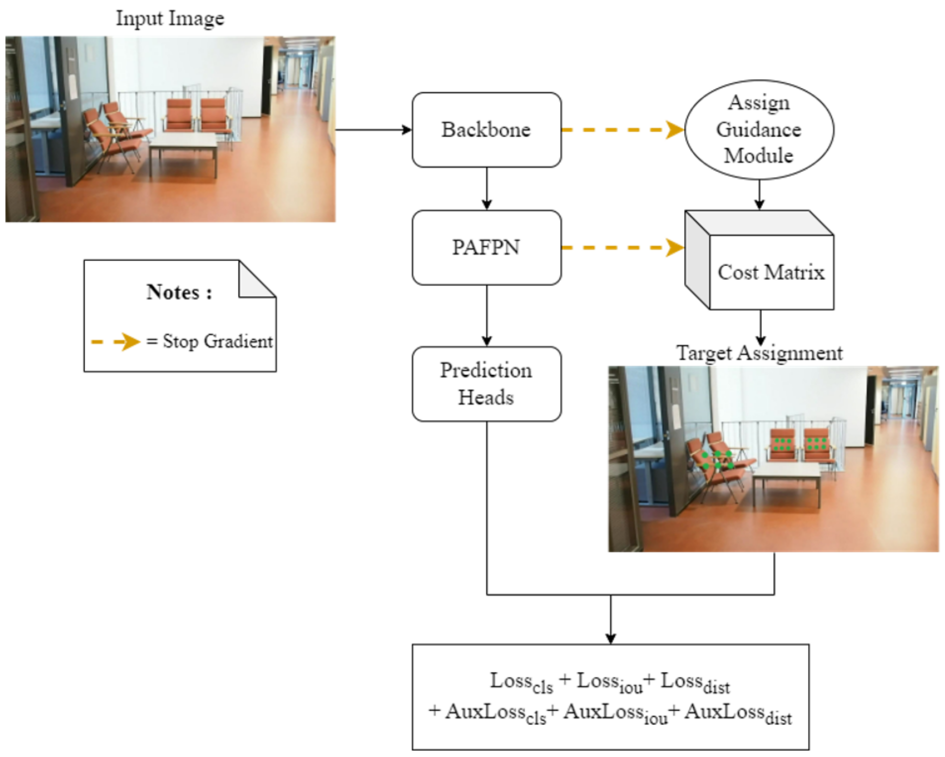

Input Image
Backbone
Assign Guidance Module
PAFPN
Cost Matrix
Notes :
= Stop Gradient
Prediction Heads
Target Assignment
$\text{Loss}_{\text{cls}} + \text{Loss}_{\text{iou}} + \text{Loss}_{\text{dist}} + \text{AuxLoss}_{\text{cls}} + \text{AuxLoss}_{\text{iou}} + \text{AuxLoss}_{\text{dist}}$

FIGURE 3. Nanodet-Plus architecture and flowchart

In terms of the architecture improvement from the previous Nanodet version, Nanodet-Plus has several new inventions. Before Nanodet-Plus, non-convolutional PAFPN [15] was used as the neck, and the classification and regression branches were merged on the detection head. In addition to that, only two sets of depth-separable convolutions were used. In Nanodet-plus, we used Ghost-Pan, a light feature pyramid for processing feature fusion between multiple layers. The structure of this pyramid consists of a set of 1×1 convolutions and 3×3 depthwise convolutions. The entire Ghost-Pan has a total of 190k parameters. In the detection head, Nanodet-Plus changed the convolution kernel size of the depthwise convolution of the detection from 3×3 to 5×5 conv.

2.3. **Detection Transformer.** The overall DETR flowchart and architecture visualization is shown in Figure 4. It consists of three main components, a CNN backbone, an Transformer encoder-decoder, and a simple Feed-Forward Network or FFN [2]. DETR consists of a set-based global loss, which forces unique predictions via bipartite matching, and a Transformer encoder-decoder architecture.

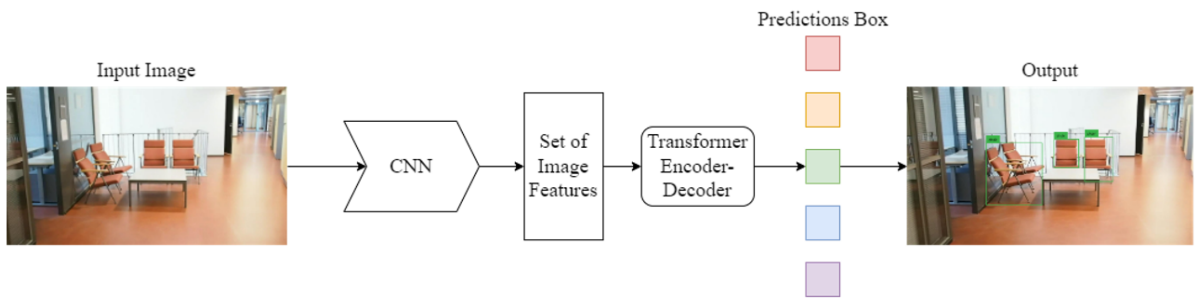


FIGURE 4. (color online) DETR architecture and flowchart

Firstly, a CNN [14] backbone will generate a lower-resolution activation map and a set of image features from the input image we provide. Then in the Transformer encoder-decoder, 1×1 convolution will reduce the channel dimension from the activation map to a smaller dimension, thus creating a new feature map. A standard architecture for every encoder layer consists of a multi-head self-attention module and a feed-forward network [2]. The decoder continues the process by taking a small fixed number of object queries as input and attending the encoder output. The decoder output will be passed to the feed-forward network to predict a detection. Lastly, the final prediction is produced by a 3-layer perceptron with a ReLU activation function, hidden dimension, and a linear projection layer [2].

2.4. **Dataset and annotation.** The datasets used in this study are from an indoor office room which we obtained from an open-source datasets repository named Roboflow [10]. The dataset consists of 2213 images, including eight classes: chair, clock, empty, exit, fire extinguisher, printer, screen, and trashbin. The annotation is already done within the Roboflow repository using a different color of bounding boxes for each class. Each class has a different color of the bounding box, denoted in Table 1. An empty class is not annotated as it is just for indicating if there is no detected class in the predicted image, however, the number of empty class annotation in the generated dataset is zero. The annotation format we used depends on each method's requirements. In this research, all three methods we examined use the same annotation in COCO format; hence the annotation is written in a JSON file. The 2213 total images are divided into 1771 training images, 221 validation images, and 221 testing images.

Table 1. Detailed distribution of the dataset

Class	**Number of instances in training images**	**Number of instances in validation images**	**Number of instances in testing images**	**Bounding box color**
Chair	682	93	76	Yellow
Clock	231	22	24	Pink
Exit	403	50	51	Purple
Fire extinguisher	647	82	89	Red
Printer	67	6	8	Green
Screen	73	6	16	Orange
Trashbin	137	19	15	Blue

3. **Discussion.** Determining the best object detection method is a challenging problem as it is necessary to accurately predict and draw a bounding box for each detected object in the frame. Therefore, using the appropriate metrics is important to evaluate an object detector's performance accurately. We need to ensure the parameters used as the evaluation metrics exist in the results of all the methods since different methods will also generate different training evaluation metrics. In the case of YOLOv8, Nanodet-Plus, and Detection Transformer, the evaluation metrics in all of the method's results are Average Precision and the bounding box loss. Hence, those two parameters are used and compared in Table 2.

The mean Average Precision (mAP) is calculated mainly from two metrics, Precision and Recall. The formulas for calculating Precision, Recall, and mAP are shown by the equations below:

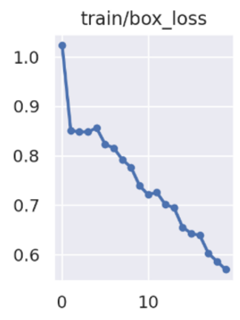
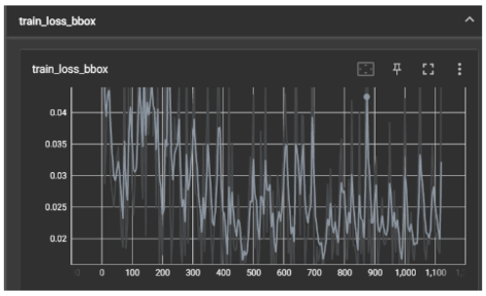
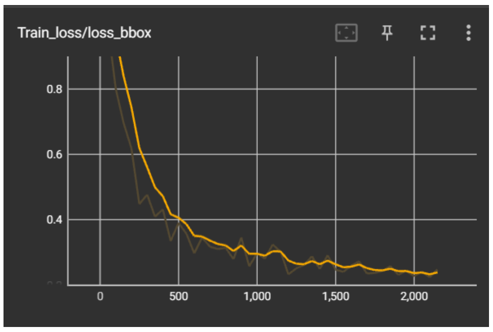
$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$

- Precision (P) measures the proportion of true positives (i.e., correct positive predictions) among all positive predictions made by the model.
- Recall (R) measures the proportion of true positives among all actual positive cases in the dataset.
- True Positive (TP) is when the model's prediction exists in the right position and is a correct prediction.
- False Positive (FP) is when the model successfully predicts an object but is labeled as an incorrect class.
- False Negative (FN) is when the model did not predict an object even though the object exists in the image.

Table 2. Performance of the three different methods

Method	**mAP@0.5**	**Box loss curve**	**Prediction on video**
YOLOv8	0.991		
Detection Transformer	0.932		
Nanodet-Plus	0.929		

Besides those four metrics, Intersection over Union (IoU) is usually used to compare with a given threshold to produce a correct or incorrect classification. Intersection over Union (IoU) itself quantifies the closeness of two bounding boxes, the ground truth box and the prediction box. Since a different Average Precision (AP) metric can be produced from each value of the IoU threshold, it is necessary to specify this value. In this experiment, we used 0.5 as the IoU threshold. After calculating the Average Precision on a certain IoU threshold, we can measure the mean Average Precision (mAP) value by taking the average Average Precision (AP) across all classes (n) under a certain IoU threshold. In the equation above, the n symbol represents the number of classes. From Table 2, we can infer that YOLOv8 has the highest mAP@0.5 number with 0.991, followed by Detection Transformer (DETR) with 0.932, and lastly Nanodet-Plus with 0.929.

Since object detection involves localization and classification, the next metric we use is bounding box loss, as localizing multiple objects in an image is mainly done by bounding boxes. The bounding box is predicted using a loss function which will give the error between the ground truth and the predicted bounding box. Lastly, we also tested each of the trained model's inferences on video, and consistent with the mAP, YOLOv8 got the highest confidence level percentage on the trash bin prediction with 97% accuracy. One of the conceptual reasons why YOLOv8 has a slightly better outcome is that in YOLOv8, the method stops the usage of mosaic augmentation. This type of augmentation has been experimentally found to degrade the performance when performed throughout the whole training routine.

4. **Conclusions and Future Work.**

In this research, we have compared and evaluated the performance of YOLOv8, Nanodet-Plus, and Detection Transformer in detecting indoor objects. With the same treatment, overall, YOLOv8, which was newly developed in 2023, slightly outperforms the other methods regarding the bounding box loss and the mean Average Precision (mAP) on 0.5 IoU threshold. This work only achieves object detection in a local computer. Hence in future work, the developer can be more focused on training and implementing the most effective method directly to the robot. Moreover, future work can also evaluate whether implementing YOLOv8 in the robot directly can perform as optimally as running it on a local computer.

Acknowledgment. This work is supported by the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Republic of Indonesia. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Z. Shi, Object detection models and research directions, *IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp.546-550, DOI: 10.1109/ICCECE51280.2021.9342049, 2021.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, End-to-end object detection with transformers, in *Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science*, A. Vedaldi, H. Bischof, T. Brox and J. M. Frahm (eds.), Cham, Springer, 2020.

[3] Z.-Q. Zhao, P. Zheng, S.-T. Xu and X. Wu, Object detection with deep learning: A review, *IEEE Transactions on Neural Networks and Learning Systems*, vol.30, no.11, pp.3212-3232, DOI: 10.1109/TNNLS.2018.2876865, 2019.

[4] W. H. E. Mg and T. T. Zin, Cattle face detection with ear tags using YOLOv5 model, *ICIC Express Letters, Part B: Applications*, vol.14, no.1, pp.65-72, DOI: 10.24507/icicelb.14.01.65, 2023.

[5] A. Hernández, C. Gomez, J. Crespo and R. Barber, Object detection applied to indoor environments for mobile robot navigation, *Sensors*, vol.16, 1180, DOI: 10.3390/s16081180, 2016.

[6] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, You only look once: Unified, real-time object detection, *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, DOI: 10.1109/CVPR.2016.91, 2016.

[7] P. Lu, Y. Ding and C. Wang, Multi-small target detection and tracking based on improved YOLO and SIFT for drones, *International Journal of Innovative Computing, Information and Control*, vol.17, no.1, pp.205-224, DOI: 10.24507/ijicic.17.01.205, 2021.

[8] Z. Tian, C. Shen, H. Chen and T. He, FCOS: Fully convolutional one-stage object detection, *The IEEE/CVF International Conference on Computer Vision*, pp.9627-9636, DOI: 10.48550/arXiv. 1904.01355, 2019.

[9] S. Zhang, C. Chi, Y. Yao, Z. Lei and S. Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, DOI: 10.1109/CVPR42600.2020.00978, 2020.

[10] Indoor, indoor-v2 Dataset, *Roboflow Universe*, https://universe.roboflow.com/indoor-euxjy/indoor-v2, 2022.

[11] C. Liu, D. Yang, L. Tang, X. Zhou and Y. Deng, A lightweight object detector based on spatial-coordinate self-attention for UAV aerial images, *Remote Sensing*, vol.15, no.1, DOI: 10.3390/rs1501 0083, 2023.

[12] H. Lou, X. Duan, J. Guo, H. Liu, J. Gu, L. Bi and H. Chen, DC-YOLOv8: Small-size object detection algorithm based on camera sensor, *Electronics (Switzerland)*, vol.12, no.10, DOI: 10.3390/electronics 12102323, 2023.

[13] J. Mo, R. Zhu, H. Yuan and Z. Shou, Detection of students' classroom concentration based on component attention, *International Journal of Innovative Computing, Information and Control*, vol.19, no.3, pp.877-891, DOI: 10.24507/ijicic.19.03.877, 2023.

[14] N. Milosevic, *Introduction to Convolutional Neural Networks*, DOI: 10.1007/978-1-4842-5648-0, 2020.

[15] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, Path aggregation network for instance segmentation, *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, DOI: 10.1109/CVPR.2018.00913, 2018.