International Journal on Advanced Science Engineering Information Technology

Vol.15 (2025) No. 2
ISSN: 2088-5334

The Effect of the Number of Classes on the Values Resulting from Evaluation Metrics in the YOLOv5 Model

Esi Putri Silmina [a,*], Tikaridha Hardiani [a], Silvi Lailatul Mahfida [b]

[a] *Information Technology Study Program, Universitas 'Aisyiyah Yogyakarta, Sleman, D.I. Yogyakarta, Indonesia*
[b] *Nutrition Study Program, Universitas 'Aisyiyah Yogyakarta, Sleman, D.I. Yogyakarta, Indonesia*
Corresponding author: [*]esiputrisilmina@unisayogya.ac.id

***Abstract*—Object detection is an essential field in computer vision. Building an object detection model requires specific categories during training: classes. This study aims to analyze the effect of the number of classes on the evaluation metrics obtained from detection model training. The dataset consists of ten classes, each containing one hundred images of size 416 x 416 pixels. The detection model is trained using the YOLOv5 model with one hundred epochs and a batch size of 16. Testing is done five times with the number of classes increasing gradually, namely two classes, four classes, six classes, eight classes, and ten classes. The test results in evaluation metrics, namely precision, recall, mAP@0.5, mAP@0.5:0.95, and training time, were analyzed and compared. The analysis shows that the number of classes significantly affects the accuracy of the evaluation metrics. Training a model with eight classes gives the best accuracy results with a precision of 75.5%, recall of 62.5%, mAP@0.5 of 71.2%, and mAP@0.5:0.95 of 35.5%. Meanwhile, training the model with two classes produced the lowest results, and training with 10 classes decreased compared to eight classes. A non-linear analysis relationship was also observed, as using ten classes slightly reduced the matrix value while maintaining efficiency in training time. This concludes that increasing the number of classes does not necessarily improve accuracy, and the optimal class configuration needs further consideration. These findings highlight important considerations for optimizing object detection models using YOLOv5 and contribute to future development.**

***Keywords*— YOLOv5; evaluation metrics; precision; recall.**

Manuscript received 27 Sep. 2024; revised 28 Jan. 2025; accepted 26 Mar. 2025. Date of publication 30 Apr. 2025.

I. INTRODUCTION

Object detection using computer vision has recently gained widespread attention due to its transformative applications in various fields. Object detection is a branch of computer vision used to detect an object in an image or video, which has been categorized into several predefined classes [1]. These categories represent specific labels that are trained into the detection model. One of the most popular object detection developments is YOLO. Joseph Redmon first proposed YOLO (You Only Look Once) [2], saying that YOLO is a technique that uses a fixed grid detector system with artificial neural networks to detect an object with processing that requires the shortest possible time [1]. YOLO is an accurate time detection algorithm that is very easy to develop. YOLO has several architectural models, one of which is the YOLOv5 model. YOLOv5 is a further development of the YOLO architecture model series for real-time detection and is famous for its effectiveness and accuracy [3]. YOLOv5 has surpassed previous versions of YOLO and competing detection models in terms of F1 scores and performance metrics [4]. These evaluation metrics typically include F1-score, precision, recall, mAP@0.5 and mAP@0.5:0.95 [5], which are obtained through training data using specific classes. The accuracy of the obtained values is affected by several factors, such as the diversity of the data set, the quality of the annotations, and the composition of the classes [6], [7].

The YOLOv5 model is specifically chosen for its real-time detection capability [8], adaptability [9], and comprehensive performance [10], in handling various data sets. Compared with other detection models, such as CNN [11], Faster R-CNN or SSD [12], YOLOv5 balances computational efficiency with precision [13], making it ideal for this study.

This research aims to analyze the effect of the number of classes on the performance of the resulting YOLOv5 object detection model. The focus of this research includes exploring the relationship between variations in the number of classes that are incrementally increased starting from two classes,

four classes, six classes, and ten classes, with the values of evaluation metrics such as precision, recall, and mAP@0.5, and mAP@0.5:0.95. This research also aims to identify the optimal number of classes that produce the best evaluation metrics and evaluate the impact of the number of courses on training time to understand the efficiency of data processing. This research is expected to provide insight into how variations in the number of classes affect object detection performance, which can be used as a guide in dataset design for practical applications. Furthermore, the results are expected to serve as a foundation for future studies on the effect of dataset configuration on the performance of object detection models.

A. Literature Review

One model of object detection that is very popular lately is YOLO. [7], with one of the development series being YOLOv5. Object detection in YOLOv5 can be done by training a specific dataset. The results of the dataset training are in the form of evaluation metric values [5]. The evaluation metric value is influenced by many factors, namely the number and diversity of datasets used, as well as the annotation tools used to label datasets that will be used to train object detection models using YOLOv5 [14]. Other research states that the factor that affects the value of evaluation metrics is the image size in the dataset used, where generally, the size used is 416 x 416. In addition to the size of the image, the study also said that the number of epochs, learning rate, and number of batches when training the dataset can affect how well the model can learn from existing data, and these results are displayed on evaluation metrics [6]. Further research states that the factor affecting evaluation metrics is class imbalance. The imbalance in question is that there is one of several classes that has a more significant number than the other classes, so that when data training is carried out, the model will work well on the majority of the data and work poorly on the minority of the other data [15].

This research will discuss the factors affecting evaluation metrics in the form of precision, recall, mAP@0.5, and mAP@0.5:0.95 by considering the influence of the number of classes trained into a detection model. In addition to using the number of classes factors, the research will apply the data model training method using a data image size of 416 x 416 and an epoch of 100 times.

B. YOLO (You Only Look Once)

The architectural model was first introduced in 2015 by Joseph Redmon [2] has undergone very significant model development in various studies conducted [16]. YOLO works to accomplish object detection through a single network, unlike F-RCNN, which produces two separate outputs: classification for probability and regression [17].

YOLO has three parts: backbone, neck, and head [18], [19]. The backbone is a Convolutional Neural Network (CNN) that is used to extract and combine features, but in other studies, the backbone uses CSPDarknet53 and integrates the RFE module at layer P5 to perform multiscale fusion [20]. Neck is a feature extraction to optimize object detection from small, medium, to large; this feature is pyramid-shaped and serves to help the model perform better generalization and scaling in object detection [21]. The head is the final part of detection that serves to apply anchor boxes to features and produce final output in the form of *class* probabilities, object-ness scores, and bounding boxes [21], [19].

The architecture of the YOLO model divides the original image into small sections of N x N grids with equal division, where N is the classes of objects.[22] to be detected and is responsible for detecting bounding boxes [23]. The architecture of the YOLO model is shown in Figure 1 [22].

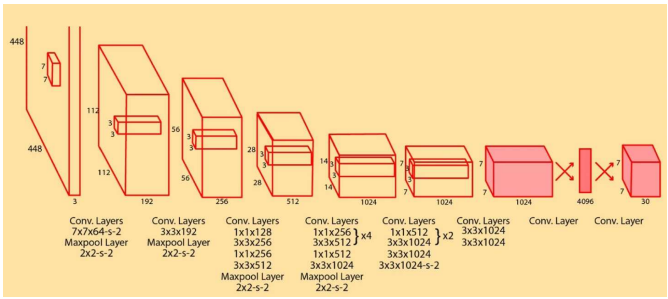

Fig. 1 YOLO Model Architecture

Figure 1 shows the final output of the YOLO network as a 7x7x30 prediction tensor, with an initial convolution of 1x1 applied to the channel output values. Next is a 3x3 convolution applied to generate the cubic output [22]. YOLO uses a ReLU (non-linear) activation architecture and linear activation for the output layers [15], [24].

C. Classes

Classes in object detection are categories or labels that represent objects that the model wants to detect [25]. Put, classes are specific categories or labels that the model will learn to be able to detect objects against these classes [17], [26]This research will use classes as a factor to determine the diversity of accuracy levels and evaluation metrics results obtained from the data model training conducted. This research will use 10 classes: chicken meat, beef, tilapia, mackerel, catfish, eggs, shrimp, tofu, and Tempe. The ten classes will gradually be trained in a data model. The evaluation metrics obtained will be analyzed, and each data training model will be compared.

D. Accuracy and Precision

The developed YOLO model series focuses on detection speed and accuracy [17], [27]. The increase in accuracy is influenced by the number of iterations [28] used so that the model can learn how to detect the data. The YOLO series has two accuracies: AP (Average Precision) and mAP (Mean Average Precision). AP is the average value of precision [29], AP is the area under the Rγ and Pγ curves by varying the value of the parameter γ [26], while mAP is a comprehensive metric used to evaluate the accuracy of object detection models [30]. mAP is the average value of AP, which is calculated separately for each class based on recall and precision [26].

Precision is calculated using Equation 1.

$$P = \frac{TP}{TP + FP} \; x \; 100\% \qquad (1)$$

True Positives (TP) in Equation 1 is the number of correctly detected targets, while False Positives (FP) is the number of backgrounds that will be detected as targets [4], [31]. The AP and mAP values are obtained using Equations 2 and 3:

$$AP = \int_0^1 p(r)d\,(r) \tag{2}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3}$$

N in Equation 3 is the number of categories or the number of classes [31], [32].

E. Recall

Recall is the ratio of real positive detections to total actual positives [30]. Recall is the ratio of detected objects that are relevant to be retrieved in the image [33]. Recall can be calculated using Equation 4.

$$R = \frac{TP}{TP + FN}\ x\ 100\% \tag{4}$$

False Negatives (FN) in Equation 4 serve to indicate the number of targets that will be detected as background [31], [33].

F. F1 Score

The F1 score in object detection serves as an evaluation indicator. F1 score is the average harmonic value of precision and recall [34], which provides a comprehensive evaluation of the performance of the model [35]. F1 score has a calculation equation that can be seen in Equation 5.

$$F1 - Score = \frac{2\ x\ P\ x\ R}{P + R} \tag{5}$$

The F1 score in Equation 5 will calculate the value obtained from precision and recall.

II. Materials and Methods

This research involves several significant steps, starting with collecting datasets, cleaning data, annotating data, and performing image preprocessing to train data into YOLOv5 models. It will then focus on the effect of using the number of classes on the value generated by the evaluation metrics.

A. Dataset

The dataset is a collection of data objects that represent data and their relationships with a similar structure [15]. The research will use a 100-disk dataset for each class. Figure 2 shows sample examples of the dataset used in this research.

Fig. 2 Sample Dataset

This research will use as many as ten classes to see the effect of the evaluation metrics results obtained, with the image data size used at 416x416 pixels. The analysis will be carried out by training the data model 5 times. Data model training will be carried out with two classes at the beginning. After completing the model training and obtaining evaluation metrics results, the data model will be retrained by adding two new data classes. The process will continue to be repeated until 10 classes are fulfilled. The evaluation metrics results obtained through five times of data model training will be analyzed and compared across each evaluation metric's data. The data model training in this study will use an epoch one hundred times and a batch size of 16. This study does not utilize augmentation in conducting data training models, aiming that the evaluation metrics are pure results from the datasets used in data modeling.

B. YOLOv5

YOLOv5 is a version of the YOLO model used for further object detection research and development. YOLOv5 is a single-stage detection model capable of detecting objects without an initial step, as in the case of two-stage detectors, which use an initial stage where important regions are then classified to check if objects have been detected in those areas [36]. The neck of the YOLOv5 applies the modified SPPF and CSP-PAN, while the head applies a structure that resembles the YOLOv3 [27]. YOLOv5 is the YOLO series that started to shift and switch frameworks from Darknet to Pytorch [17].

The architecture of YOLOv5 itself uses CSPDarknet53 [37] with SPP structure as backbone, PANet as neck, and YOLO as head [38]. The architecture of YOLOv5 can be seen in Figure 3 [17].

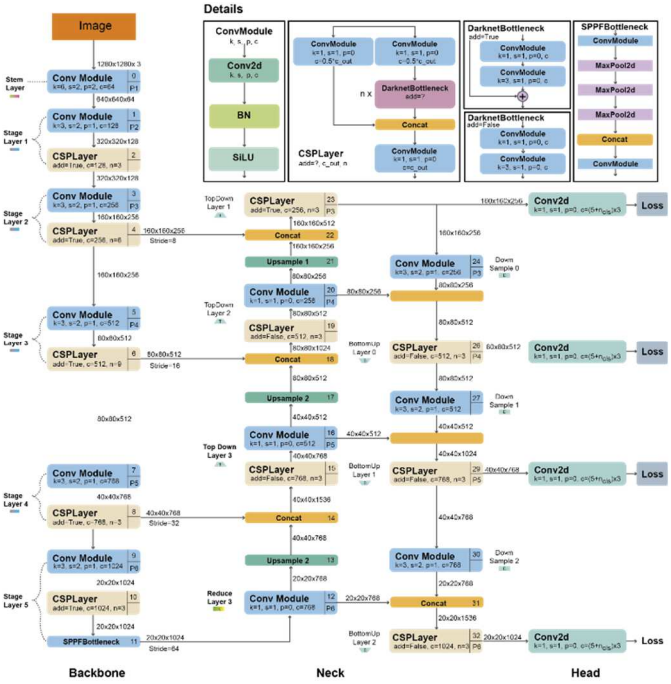

Fig. 3 YOLOv5 Architecture

The architecture of YOLOv5 in Figure 3 consists of a backbone, neck, and head [39]. As mentioned in YOLOv5, the backbone uses CSPDarknet53 [38], [37], which has been previously modified with a stem [17] to perform convolutional layers that extract features [30] in the image to be detected. Spatial Pyramid Pooling Fast (SPPF) speeds up computation and feature merging with a predetermined fixed size [40]. Convolution has a form of normalization in each batch [41] owned and activated through SiLu [40]. The SPPF is used as a neck function [17] in the YOLOv5 architecture with the addition of a modified CSP-PAN [42], while the head [17] has a similar structure to YOLOv3 [43].

III. Results and Discussion

This research compares the effect of various class sizes on the accuracy of the evaluation metrics generated by the object detection model. Evaluation metrics such as precision, recall, mAP@0.5, mAP@0.5:0.95, and training time are further analyzed and detailed. The dataset, consisting of 100 sample classes for a total of 10 classes, was trained with controlled parameters: image size (416 x 416 pixels), epoch (100), and batch size (16). The training was performed on Google Collab.

The first analysis conducted is the effect of 2 classes on the accuracy level of the evaluation metrics. The classes used are beef and chicken. The results of the data training into a detection model can be seen in Table 1.

TABLE I
RESULTS IN EVALUATION METRICS FOR TWO CLASSES

Evaluation Metrics	Results
Precision (%)	59.5
Recall (%)	52.8
mAP@0.5 (%)	54.3
mAP@0.5:0.95 (%)	19.3
Training Time (seconds)	299

Table 1, the results of training on two classes (beef and chicken) resulted in low metric values, with a precision of 59.5% and recall of 52.8%. mAP@0.5 was 54.3%, and mAP@0.5:0.95 was 19.3%. These results reflect the limitations in model generalization when the number of classes is minimal, leading to suboptimal learning and high detection error rates. The training time is short, at 299 seconds.

The second analysis adds two new classes, catfish and tilapia, so the total number of classes to be trained becomes four: beef, chicken, catfish, and tilapia. The four classes will be re-trained and will not use the results of the previous training, so the actual evaluation metrics results will be obtained. The results of data training with four classes in a detection model can be seen in Table 2.

TABLE II
RESULTS IN EVALUATION METRICS FOUR CLASSES

Evaluation Metrics	Results
Precision (%)	62.6
Recall (%)	64.2
mAP@0.5 (%)	66.6
mAP@0.5:0.95 (%)	28.2
Training Time (seconds)	531

The model training results in Table 2 show that adding catfish and tilapia increased precision to 62.6% and recall to 64.2%. mAP@0.5 increased to 66.6%, while mAP@0.5:0.95 increased to 28.2%. The improved metrics show that increasing the number of classes can increase feature diversity and lead to better generalization. However, the training time almost doubled, reaching 531 seconds.

The third analysis adds two new classes, mackerel and catfish, so the overall classification becomes six classes: beef, chicken, tilapia, catfish, mackerel, and catfish. Similar to the previous analysis, the detection model training will be redone from the beginning, and the evaluation metrics results obtained can be seen in Table 3.

TABLE III
RESULTS IN EVALUATION METRICS 6 CLASSES

Evaluation Metrics	Results
Precision (%)	65.2
Recall (%)	54.1
mAP@0.5 (%)	61.5
mAP@0.5:0.95 (%)	26.4
Training Time (seconds)	791

The model training results in Table 3, including mackerel and catfish, resulted in a precision of 65.2%, a recall of 54.1%, mAP@0.5 of 61.5%, and mAP@0.5:0.95 of 26.4%. Notably, recall decreased significantly compared to the 4-class setting, indicating that class imbalance or increased class competition may hinder model learning. The training time increased to 791 seconds.

The fourth analysis uses eight classes by adding two new classes: shrimp and chicken eggs. The total classes are beef, chicken, tilapia, catfish, mackerel, catfish, shrimp, and chicken eggs. Table 4 shows the results of training data with 8 classes into a detection model.

TABLE IV
RESULTS IN EVALUATION METRICS 8 CLASSES

Evaluation Metrics	Results
Precision (%)	75.5
Recall (%)	62.5
mAP@0.5 (%)	71.2
mAP@0.5:0.95 (%)	35.5
Training Time (seconds)	1111

The model training results in Table 4 by adding shrimp and chicken eggs, the model achieved its best performance, with precision 75.5%, recall 62.5%, mAP@0.5 71.2%, and mAP@0.5:0.95 35.5%. These results show that including diverse but manageable classes can optimize feature extraction, leading to higher accuracy. The training time was 1111 seconds.

The last analysis uses ten classes: beef, chicken meat, tilapia, catfish, mackerel, shrimp, chicken eggs, tofu, and tempeh. Two additional new classes will be used for data model training. The results of training data with ten classes in a detection model can be seen in Table 5.

TABLE V
RESULT IN EVALUATION METRICS FOR 10 CLASSES

Evaluation Metrics	Results
Precision (%)	73.1
Recall (%)	62.3
mAP@0.5 (%)	68.5
mAP@0.5:0.95 (%)	34.9
Training Time (seconds)	1286

The model training results in Table 5 add tofu and tempeh, slightly reducing precision to 73.1%, recall to 62.3%, mAP@0.5 to 68.5%, and mAP@0.5:0.95 to 34.9%. This nonlinear behavior indicates a saturation point where increasing classes introduce additional complexity, reducing accuracy gains while slightly affecting training efficiency (1286 seconds). Figure 4 shows the comparison graph of the evaluation metrics' accuracy results from the five tests.

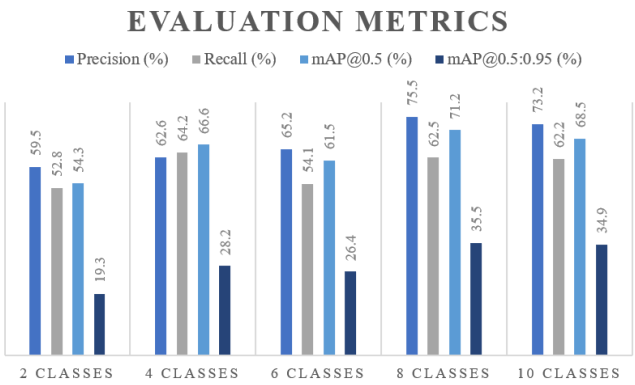

Fig. 4 Evaluation Metrics

Figure 4 shows that testing with eight classes produces the highest evaluation metric accuracy compared to other tests. However, the results of testing with eight classes are slightly different from testing using ten classes. This shows that the detection performance in the 8eightand 10-class configurations is relatively similar, with precision values above 70%. The recall accuracy for these two configurations shows the same result, which is 62.5%. The difference in accuracy is seen in the value of mAP@0.5, with a difference of 2.7%, and mAP@0.5:0.95, with a difference of 0.6%. The increase in accuracy in the eight-class test compared to the previous test shows that a more balanced number of classes can improve the model's ability to detect objects accurately. However, the results from testing 10 classes show that increasing the number of classes further does not always provide a significant improvement, possibly due to increased data complexity or overfitting.

The number of classes significantly influences the value of evaluation metrics such as precision, recall, mAP@0.5, and mAP@0.5:0.95. Five configurations of the number of classes were analyzed: 2, 4, 6, 8, and 10 classes. Precision improved until the configuration of 8 classes, which indicates the ability of the model to detect the correct objects accurately. However, at the 10-class configuration, precision experienced a slight decrease, which could be due to the added complexity of detecting more diverse objects. Recall also shows fluctuations, with the highest value at eight classes. This indicates that adding more classes can improve detection sensitivity, but at some point, the model faces challenges in correctly recognizing all objects.

The mAP@0.5 value increases consistently up to the eight classes configuration but experiences a small decrease in the 10 classes configuration. This decrease can be attributed to overfitting or a lack of model generalization on datasets with more classes. In contrast, mAP@0.5:0.95 tends to show more stable results but still experiences a slight decrease at the 10 classes configuration. As the number of classes increases, the complexity of the dataset increases, which affects the ability of the model to learn from the data effectively. The model has more specific and limited data at smaller class configurations (2 and 4 classes), resulting in low accuracy. However, when the number of classes increases to 8, the model achieves an optimal balance between data diversity and the ability to learn effectively.

In contrast, testing with two classes showed the lowest results. The evaluation metrics accuracy value did not reach 60%, with precision of 59.5%, recall of 52.8%, mAP@0.5 of 54.3%, and mAP@0.5:0.95 of 28.2%. This low accuracy could be due to the lack of data diversity in a very small configuration of classes, which limits the model's ability to recognize patterns effectively.

When the number of classes increased to 4 and 6, the accuracy results showed a significant increase. In the four-class configuration, the precision accuracy reached 62.6%, recall 64.2%, mAP@0.5 of 66.6%, and mAP@0.5:0.95 of 28.2%. However, there was a slight decrease in the six-class configuration in the recall and mAP values, with a recall of 54.1%, mAP@0.5 of 61.5%, and mAP@0.5:0.95 of 26.4%. This decrease indicates that as the number of classes increases, the model may face challenges in generalizing more complex patterns.

Data model training and testing also require model training time; the results of this time are shown in Figure 5.

Fig. 5 Training Time

By previous research that discusses the training time of the model [44], the training time is influenced by the number of datasets used [45]. This study aims to analyze the effect of the number of classes on the accuracy of the evaluation metrics obtained. Each class consists of one hundred images with a resolution of 416 x 416 pixels. Figure 5 shows that the training time for a configuration with two classes takes 299 seconds. This training time is relatively fast, considering that the total dataset used consists of only two hundred images. In the second test with four classes, the training time is almost double that of the first configuration, which is 531 seconds with a total dataset of four hundred images. The third training, which used six classes, took 791 seconds. This shows that the training time increases as the number of classes increases, but the increase is not always linear. The total dataset in this configuration was 600 images. The training time was recorded at 1111 seconds in the fourth training with eight classes. However, the number of classes increased from 2 to 8 with a total dataset of 800 images. The fifth training, with a configuration of 10 classes, took 1286 seconds to process 1000 images. This training time was the highest compared to the other configurations. However, the increase in training time from the fourth configuration (8 classes) to the fifth configuration (10 classes) was only 175 seconds, indicating that the training efficiency was maintained despite the increased number of classes.

The results show that the training time increases as the number of classes increases because the model must process more information. However, the increase in training time is

not always proportional to the increase in accuracy. Although the training time increased in the configuration with 10 classes, the accuracy decreased. This indicates that increasing the number of classes does not always positively impact model performance.

IV. Conclusion

This study aims to analyze the effect of the number of classes on the accuracy of evaluation metrics in the YOLOv5 model. The results show that the configuration with eight classes produces the best evaluation metrics, including precision of 75.5%, recall of 62.5%, mAP@0.5 of 71.2%, and mAP@0.5:0.95 of 35.5%. This configuration reflects the optimal balance between the number of classes and the model's ability to recognize patterns effectively without adding excessive complexity. In contrast, the configuration with ten classes limits the model's generalization. Although the training time increased, the accuracy did not significantly improve, with a difference in precision of 2.3% lower than the configuration with eight classes. This suggests that increasing the number of classes may increase the complexity of the model without providing a proportional accuracy gain. In addition, the training time increases consistently as the number of classes increases. For example, training with 10 classes takes 1286 seconds, while the configuration with eight classes takes 1111 seconds.

However, increased training time is not always followed by increased accuracy, underscoring the importance of time efficiency in experiment design. This research shows that the optimal number of classes is essential for maximizing object detection models' accuracy and training efficiency. These results guide dataset design and model setup in object detection applications, emphasizing the balance between the number of classes, training time, and optimal accuracy results. This research provides important insights that determine the correct number of classes, which is critical to maximizing the performance of object detection models. These results can be used as a guide in dataset design and model training in future object detection applications.

Acknowledgment

We gratefully acknowledge the support the Ministry of Education, Culture, Research and Technology provided for the grant to conduct this research.

References

[1] J. Kaur and W. Singh, *A systematic review of object detection from images using deep learning*, vol. 83, no. 4. Springer US, 2024. doi:10.1007/s11042-023-15981-y.

[2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified real-time object detection", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[3] K. Zhao, "Enhancing the Performance and Accuracy in Real-Time Football and Player Detection Using Upgraded YOLOv5 Architecture," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, 2024, doi:10.1007/s44196-024-00565-x.

[4] A. J. Azmawi, W. N. Mohd-Isa, and A. A. A. Rahman, "Telecommunication Fiber Box Detection Using YOLO in Urban Environment," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 13, no. 6, pp. 2136-2144, 2023, doi: 10.18517/ijaseit.13.6.19027.

[5] A. Balasundaram, A. Mohanty, A. Shaik, K. Pradeep, K. P. Vijayakumar, and M. S. Kavitha, "Zero-DCE++ Inspired Object Detection in Less Illuminated Environment Using Improved YOLOv5," *Comput. Mater. Contin.*, vol. 77, no. 3, pp. 2751-2769, 2023, doi: 10.32604/cmc.2023.044374.

[6] J. Kaur and W. Singh, "Tools, techniques, datasets and application areas for object detection in an image: a review," *Multimed. Tools Appl.*, vol. 81, no. 27, pp. 38297-38351, 2022, doi: 10.1007/s11042-022-13153-y.

[7] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, pp. 1-29, 2023, doi: 10.1007/s44196-023-00302-w.

[8] S. S. Debnath, R. K. Verma, and P. Mahudapathi, *Real-Time Object Detection Using YOLOv5*, vol. 405 SIST. Springer Nature Singapore, 2024. doi: 10.1007/978-981-97-6222-4_38.

[9] V. A. Kich *et al.*, "Precision and Adaptability of YOLOv5 and YOLOv8 in Dynamic Robotic Environments," in *2024 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, 2024, pp. 514-519. doi: 10.1109/CIS-RAM61939.2024.10673292.

[10] J. Wang *et al.*, "Research on Tea Trees Germination Density Detection Based on Improved YOLOv5," *Forests*, vol. 13, no. 12, 2022, doi:10.3390/f13122091.

[11] J. H. Lee and K. S. Song, "Comparison and Analysis of CNN Models to Improve a Facial Emotion Classification Accuracy for Koreans and East Asians," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 14, no. 3, pp. 811-817, 2024, doi: 10.18517/ijaseit.14.3.18078.

[12] T. Mahendrakar *et al.*, "Performance Study of YOLOv5 and Faster R-CNN for Autonomous Navigation around Non-Cooperative Targets," *IEEE Aerosp. Conf. Proc.*, vol. 2022-March, pp. 1-12, 2022, doi:10.1109/aero53065.2022.9843537.

[13] R. Han, Y. Zheng, R. Tian, and L. Shu, "An image dataset for analyzing tea picking behavior in tea plantations," no. January, pp. 1-8, 2025, doi: 10.3389/fpls.2024.1473558.

[14] A. Vijayakumar and S. Vairavasundaram, *YOLO-based Object Detection Models: A Review and its Applications*, no. 0123456789. Springer US, 2024. doi: 10.1007/s11042-024-18872-y.

[15] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243-9275, 2023, doi: 10.1007/s11042-022-13644-y.

[16] J. Lee and K. il Hwang, "YOLO with adaptive frame control for real-time object detection applications," *Multimed. Tools Appl.*, vol. 81, no. 25, pp. 36375-36396, 2022, doi: 10.1007/s11042-021-11480-0.

[17] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680-1716, 2023, doi: 10.3390/make5040083.

[18] F. Prinzi, M. Insalaco, A. Orlando, S. Gaglio, and S. Vitabile, "A Yolo-Based Model for Breast Cancer Detection in Mammograms," *Cognit. Comput.*, vol. 16, no. 1, pp. 107-120, 2024, doi: 10.1007/s12559-023-10189-6.

[19] J. Zhong, Q. Cheng, X. Hu, and Z. Liu, "YOLO Adaptive Developments in Complex Natural Environments for Tiny Object Detection," *Electron.*, vol. 13, no. 13, 2024, doi:10.3390/electronics13132525.

[20] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, and X. Wang, "YOLO-FaceV2: A scale and occlusion aware face detector," *Pattern Recognit.*, vol. 155, p. 110714, 2024, doi:10.1016/j.patcog.2024.110714.

[21] S. P. Yadav, M. Jindal, P. Rani, V. H. C. de Albuquerque, C. dos Santos Nascimento, and M. Kumar, "An improved deep learning-based optimal object detection system from images," *Multimed. Tools Appl.*, vol. 83, no. 10, pp. 30045-30072, 2024, doi: 10.1007/s11042-023-16736-5.

[22] F. Rashidi Ranjbar and A. Zamanifar, "Autonomous dental treatment planning on panoramic x-ray using deep learning based object detection algorithm," *Multimed. Tools Appl.*, vol. 83, no. 14, pp. 42999-43033, 2024, doi: 10.1007/s11042-023-17048-4.

[23] A. Bal, M. Das, and S. M. Satapathy, "YOLO as a Region Proposal Network for Diagnosing Breast Cancer," *2021 Grace Hopper Celebr. India, GHCI 2021*, pp. 1-6, 2021, doi:10.1109/ghci50508.2021.9513988.

[24] A. Karaman *et al.*, "Robust real-time polyp detection system design based on YOLO algorithms by optimizing activation functions and hyper-parameters with artificial bee colony (ABC)," *Expert Syst. Appl.*, vol. 221, p. 119741, 2023, doi: 10.1016/j.eswa.2023.119741.

[25] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," *Digit. Signal Process.*, vol. 132, p. 103812, 2023,

doi: 10.1016/j.dsp.2022.103812.
[26] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39-64, 2020, doi:10.1016/j.neucom.2020.01.085.
[27] M. Hussain, "YOLOv1 to v8: Unveiling Each Variant-A Comprehensive Review of YOLO," *IEEE Access*, vol. 12, no. February, pp. 42816-42833, 2024, doi:10.1109/access.2024.3378568.
[28] R. Al Amin, M. Hasan, V. Wiese, and R. Obermaisser, "FPGA-Based Real-Time Object Detection and Classification System Using YOLO for Edge Computing," *IEEE Access*, vol. 12, no. May, pp. 73268-73278, 2024, doi: 10.1109/ACCESS.2024.3404623.
[29] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257-276, 2023, doi:10.1109/JPROC.2023.3238524.
[30] R. Sapkota *et al.*, "YOLOv10 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once Series," 2024, arXiv: 2406.19407, doi: 10.2139/ssrn.4874098
[31] X. Zhai, Z. Huang, T. Li, H. Liu, and S. Wang, "YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection," *Electron.*, vol. 12, no. 17, 2023, doi: 10.3390/electronics12173664.
[32] J. Miao *et al.*, "YOLO-VSF: An Improved YOLO Model by Incorporating Attention Mechanism for Object Detection in Traffic Scenes," *J. Shanghai Jiaotong Univ.*, 2024, doi: 10.1007/s12204-024-2751-y.
[33] X. Han, J. Chang, and K. Wang, "Real-time object detection based on YOLO-v2 for tiny vehicle objects," *Procedia Comput. Sci.*, vol. 183, pp. 61-72, 2021, doi: 10.1016/j.procs.2021.02.031.
[34] C. Zhao *et al.*, "AC-YOLO: Multi-category and high-precision detection model for stored grain pests based on integrated multiple attention mechanisms," *Expert Syst. Appl.*, vol. 255, p. 124659, 2024, doi: 10.1016/j.eswa.2024.124659.
[35] X. Cao, J. Wu, J. Chen, and Z. Li, "Complex Scenes Fire Object Detection Based on Feature Fusion and Channel Attention," *Arab. J. Sci. Eng.*, 2024, doi: 10.1007/s13369-024-09471-y.
[36] C. Santos, M. Aguiar, D. Welfer, and B. Belloni, "A New Approach for Detecting Fundus Lesions Using Image Processing and Deep Neural Network Architecture Based on YOLO Model," *Sensors*, vol. 22, no. 17, 2022, doi: 10.3390/s22176441.
[37] J. Ye, Z. Yuan, C. Qian, and X. Li, "CAA-YOLO: Combined-Attention-Augmented YOLO for Infrared Ocean Ships Detection," *Sensors*, vol. 22, no. 10, pp. 1-23, 2022, doi: 10.3390/s22103782.
[38] L. Zhao, T. Tohti, and A. Hamdulla, "BDC-YOLOv5: a helmet detection model employs improved YOLOv5," *Signal, Image Video Process.*, vol. 17, no. 8, pp. 4435-4445, 2023, doi: 10.1007/s11760-023-02677-x.
[39] L. Zhang, J. Li, and F. Zhang, "An Efficient Forest Fire Target Detection Model Based on Improved YOLOv5," *Fire*, vol. 6, no. 8, 2023, doi: 10.3390/fire6080291.
[40] H. Liu, X. Duan, H. Lou, J. Gu, H. Chen, and L. Bi, "Improved GBS-YOLOv5 algorithm based on YOLOv5 applied to UAV intelligent traffic," *Sci. Rep.*, vol. 13, no. 1, pp. 1-12, 2023, doi: 10.1038/s41598-023-36781-2.
[41] N. T. Nguyen, Q. Tran, C. H. Dao, D. A. Nguyen, and D. H. Tran, "Automatic Detection of Personal Protective Equipment in Construction Sites Using Metaheuristic Optimized YOLOv5," *Arab. J. Sci. Eng.*, vol. 49, no. 10, pp. 13519-13537, 2024, doi:10.1007/s13369-023-08700-0.
[42] S. Li, S. Liu, Z. Cai, Y. Liu, G. Chen, and G. Tu, "TC-YOLOv5: rapid detection of floating debris on raspberry Pi 4B," *J. Real-Time Image Process.*, vol. 20, no. 2, pp. 1-13, 2023, doi: 10.1007/s11554-023-01265-z.
[43] H. Wang, Y. Jin, H. Ke, and X. Zhang, "DDH-YOLOv5: improved YOLOv5 based on Double IoU-aware Decoupled Head for object detection," *J. Real-Time Image Process.*, vol. 19, no. 6, pp. 1023-1033, 2022, doi: 10.1007/s11554-022-01241-z.
[44] W. Sun, H. Li, Q. Liang, X. Zou, M. Chen, and Y. Wang, *On data efficiency of univariate time series anomaly detection models*, vol. 11, no. 1. Springer International Publishing, 2024. doi: 10.1186/s40537-024-00940-7.
[45] Z. Hammoudeh and D. Lowd, *Training data influence analysis and estimation: a survey*, vol. 113, no. 5. Springer US, 2024. doi:10.1007/s10994-023-06495-7.