Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (166-172)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.46063

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Small Object Detection and Object Counting For Primary Roe Dataset Based on Yolo

Wahyu Andi Saputra[1], Nicolaus Euclides Wahyu Nugroho[2], Dany Candra Febrianto[3], Andi Prademon Yunus[4], Muhammad Azrino Gustalika[5], and Yit Hong Choo[6]

[1,2,3,4,5]Department of Informatics Engineering, Telkom University
[6]Deakin University
[1,2,3,4,5]Jl. DI Panjaitan No.128, Karangreja, Purwokerto Kidul, Kecamatan Purwokerto Selatan, Kabupaten Banyumas, Jawa Tengah, Indonesia
[6]Building NS, Deakin University, Geelong Waurn Ponds Campus, 75 Pigdons Road, Waurn Ponds, Australia

ABSTRACT

Article:
Accepted: January 17, 2025
Revised: August 20, 2024
Issued: April 30, 2025

***Correspondence Address:**
andiwahyu@telkomuniversity.ac.id

This research offers an initial exploration into the effectiveness of three variations of the YOLOv8 model original, trimmed, and YOLOv8n.pt in combination with two distinct datasets characterized by tight and loose distributions of roe, aimed at enhancing small object detection and counting accuracy. Utilizing a primary roe dataset across 776 images, the research systematically compares these model-dataset configurations to identify the most effective combination for precise object detection. The experimental results reveal that the YOLOv8n.pt model combined with the loosely distributed dataset achieves the highest detection performance, with a mean Average Precision (mAP) of 53.86%. This outcome underscores the critical impact of both model selection and data distribution on the detection accuracy in machine learning applications. The findings highlight the importance of tailored model and dataset synergies in optimizing detection tasks, particularly in complex scenarios involving small, densely clustered objects. This research contributes valuable insights into the strategic deployment of neural network architectures for refined object detection challenges.

Keywords : *: object counting; real-time; roe dataset; small object detection; yolov8.*

1. INTRODUCTION

Indonesia possesses a wealth of fishery resources and significant potential, both in freshwater and saltwater [1]. The Gurami (Osphronemus gouramy) is a fish species with significant potential and considerable commercial significance in Indonesia. Gourami fish have a single, uninterrupted lateral line in their morphology. In addition, the fish possesses ctenoid scales and a lower jaw with teeth [2]. The calculation of fish eggs constitutes a critical aspect of supporting the cultivation and commercial exchange of Gourami fish seeds. Precision in this procedure not only facilitates the maintenance of transaction quality between farmers and buyers but also enhances production management efficiency. Accurate quantitative data enables the optimization of the spawning and seeding processes, thereby augmenting overall cultivation productivity [3], [4].

Notably, the calculation methods predominantly employed currently tend to be manual, relying purely on visual estimations Figure 1 or subjective judgment. This reliance engenders a significant source of inaccuracy, particularly given the diminutive size of the eggs and their propensity to overlap or accumulate in visual representations. Consequently, both sellers and buyers face potential losses stemming from erroneous seed quantity estimates [5].

In recent years, advancements in computer vision technology, particularly concerning small object detection, have progressed significantly and have begun to find applications across diverse fields, including precision agriculture and environmental monitoring [5], [6]. Nevertheless, challenges associated with detecting small objects remain considerable, primarily due to attributes such as their extremely small size, low resolution, background interference (noise), and overlap among objects. Numerous studies have endeavored to enhance detection accuracy by adapting well-known deep learning architectures, such as YOLO, to detect small objects in aerial imagery acquired from unmanned aerial vehicles (UAVs) [3], [5], [6].

Among the developed methodologies, RT-YOLO employs residual feature fusion and triple attention mechanisms to bolster target detection accuracy in aerial imagery [3]. MSC-YOLO, in contrast, leverages multi-scale spatial context to improve effectiveness in addressing variations in small object size [5]. Other enhancements include Small Target-YOLOv5, which optimizes the detection of exceedingly small objects in drone imagery [3], and a technique tailored specifically for the detection and classification of fish in underwater settings [4].

Moreover, the application of small object detection has extended into various sectors. For instance, a study by [7] utilized YOLOv5 to assess nutrient deficiencies in oil palm leaves, while [8] implemented a real-time detection system for small plastic waste in marine environments using UAVs. Lightweight models such as LEAF-YOLO have been developed for efficient object detection on edge devices while preserving performance [9]. LUD-YOLO has been specifically designed for resource efficiency without compromising accuracy, particularly in urban aerial image analysis [10].

Within the aquaculture sector, several studies have demonstrated considerable advancements, including the automatic detection of fish and their abnormal behaviors through the combination of YOLO variants and sophisticated tracking techniques such as SiamRPN++ [11], [12], [13]. Furthermore, the integration of semantic segmentation with the YOLO detector [14] has proven effective in enhancing accuracy for stacked objects or complex backgrounds. Another noteworthy innovation is the two-stage Focus-and-Detect framework [15], which utilizes a Gaussian Mixture Model to direct the system's focus on dense areas of small objects, thereby improving detection precision.

The present study employs a dataset comprising approximately 776 manually annotated Gourami images, categorized into training and validation images. The primary aim of this research is to implement the YOLOv8-based small object detection algorithm to accurately detect and count Gourami eggs, even in instances of object overlap within individual images.

A significant contribution of this study lies in the provision of a solution for small object detection in the aquaculture domain utilizing the YOLOv8 approach—an area that has thus far received limited in-depth discussion within the small object detection literature, particularly in relation to UAV imagery and precision agriculture applications.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (166-172)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.46063

Figure 1. *Display of roe inside a jar*

2. METHODS

This section outlines the experimental approach undertaken to evaluate the performance of different YOLOv8 model variants in detecting and counting small objects. The study focuses on a comparative analysis of three YOLOv8 versions—original, trimmed, and YOLOv8n.pt—used in conjunction with two distinct types of datasets characterized by tightly and loosely distributed roe instances. The methodological framework is designed to identify the most optimal model-dataset pairing to improve detection accuracy, especially in scenarios involving small and densely clustered objects. Experiments were consistently conducted using a primary roe dataset consisting of 776 images, and model performance was assessed using the mean Average Precision (mAP) metric as the principal evaluation criterion.

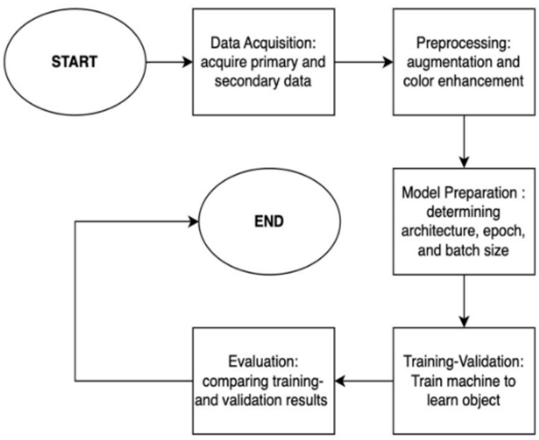


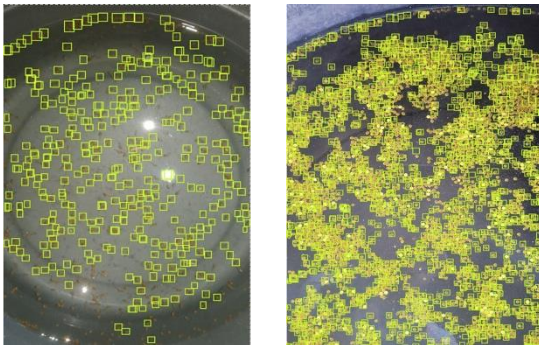
Figure 2. *Research flow of proposed methodology*

Figure 2 presented outlines a structured process for implementing a machine learning project aimed at object detection, specifically using a dataset of primary and secondary roe images. This process is divided into several key stages, each critical for ensuring the effectiveness and efficiency of the machine learning models, particularly the YOLOv8 variants used in the study.

Data Acquisition: The initial stage involves the acquisition of primary and secondary data. Primary data comprises the core dataset used for the direct training and testing of the models. In this case, it includes 776 images of roe, categorized based on their distribution - either tight or loose. Secondary data could involve additional datasets or augmented data derived from the primary data to enhance the robustness and variability of the training process[3], [4]. The acquisition of high-quality and relevant data is crucial as it forms the foundation upon which all further analysis and model training are built. Table 1 presents the dataset used in this research.

Table 1. *Dataset being used in this research*

Type	Information	Data Split
Tight Dataset – Figure 3.b.	Roe is closely positioned, increasing occlusion and stacking, challenging detection precision.	Train: 339 images Valid: 25 images
Loose Dataset - Figure 3.a.	Roe is distantly positioned, minimizing occlusion, testing broader detection capabilities.	Train: 396 images Valid: 16 images



(a) (b)

Figure 3. *(a) Sample a dataset with loosely distributed objects, (b) Sample a dataset with tightly distributed objects*

Preprocessing: Once data is acquired, preprocessing is necessary to prepare the data for effective training. This involves augmentation and color enhancement techniques. Augmentation includes methods like rotation, scaling, and flipping of images,

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (166-172)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.46063

which help in creating a more diverse set of training examples from the existing data (Table 2).

Table 2. *Data augmentation techniques used in pre-processing [5]*

No	Technique	Description	Parameter Example
1	Rotation	Random rotation of image	±15°
2	Flipping	Horizontal and vertical image flip	Horizontal = True
3	Zooming	Scale in/out for size variance	Range: 0.8–1.2
4	Brightness Adjust	Modifies brightness levels	±25%
5	Contrast Adjust	Enhances contrast	Factor: 0.8–1.2
6	Saturation Adjust	Changes color saturation	Factor: 0.9–1.1

This diversity is vital for training robust models that perform well across various scenarios and not just the conditions seen in the training data, presented in Table 3. Color enhancement can be particularly important in scenarios where the object of interest (roe in this case) might vary in visibility due to lighting conditions or background interference of roe images collection.

Table 3. *YOLOv8 model types and configuration*

Model	Information
Yolov8n.pt	The yolov8n.pt file is a pre-trained model checkpoint for YOLOv8, optimized for fast, lightweight object detection with minimal computational resources.
Yolov8ori.yaml	The yolov8.yaml file defines the model architecture, hyperparameters, and dataset configuration for training and deploying YOLOv8 object detection models.
Yolov8_tinyobj.yaml (or YOLOv8 trimmed)	The yolov8.yaml file defines the model architecture, hyperparameters, and dataset configuration for YOLOv8 specified for tiny object detection.

Model Preparation: The next phase is model preparation, which involves setting up the parameters for the machine learning model. This includes selecting the appropriate architecture (YOLOv8 original, trimmed, or YOLOv8n.pt) and determining the epochs and batch size. The choice of architecture impacts the model's ability to learn from the data, with different architectures offering various trade-offs between speed, accuracy, and computational efficiency. The number of epochs and batch size are tuned to optimize learning speed and performance, with more epochs potentially leading to better learning but also a higher risk of overfitting.

Training-Validation: During the training-validation phase, the model is developed to detect and recognize objects within images by learning from annotated examples. Validation occurs simultaneously, using a distinct dataset unseen during training, to evaluate the model's ability to generalize to new data rather than just memorizing the training set. Table 4 states the configurations used in this research.

Table 4. *CNN training and validation scenario combination of model and dataset*

1. Yolov8n.pt - Loose Dataset 2. Yolov8 Original - Loose Dataset 3. Yolov8 Trimmed - Loose Dataset	4. Yolov8n.pt - Tight Dataset 5. Yolov8 Original - Tight Dataset 6. Yolov8 Trimmed - Tight Tight
Epoch	150
Batch Size	6
Learning Rate	0.001

Evaluation: After training, the model is evaluated by comparing its performance on the training data against its performance on the validation data. This comparison is crucial for assessing the model's effectiveness and generalization capabilities. Metrics such as mean Average Precision (mAP), commonly used in object detection tasks, are calculated to quantify the model's accuracy. The primary benchmark used is mAP@0.5, following Zhang et al. [4]. Table 5 compares the performance of different model variants.

Table 5. *Model performance comparison*

Model Variant	Precision	Recall	mAP @0.5	mAP@ 0.5:0.95	Inference Time (ms)
YOLO v8n	0.88	0.85	0.87	0.63	12.4
YOLO v8s	0.91	0.88	0.90	0.67	16.7

3. RESULTS AND DISCUSSION

This section presents and discusses the experimental results obtained from evaluating the three YOLOv8 model variants—Yolov8n.pt, Yolov8ori.yaml, and Yolov8_tinyobj.yaml—across datasets with different object distributions.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (166-172)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.46063

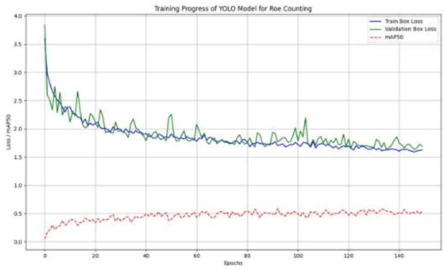


Figure 4. *Training, validation, and mAP50 result for loose dataset - YOLOv8n.pt*

Figure 4 displays the training and validation box losses demonstrated a steady decline from initial values of 3.5981 and 3.8426 to 1.6181 and 1.6903, respectively, indicating effective learning and generalization capabilities. Concurrently, the mean Average Precision (mAP50) significantly increased from 0.04853 to 0.53863, underscoring a substantial enhancement in the model's accuracy in detecting roe with at least 50% overlap with ground truths. These trends validate the model's efficacy in learning relevant features over epochs, crucial for practical deployment in fisheries management.

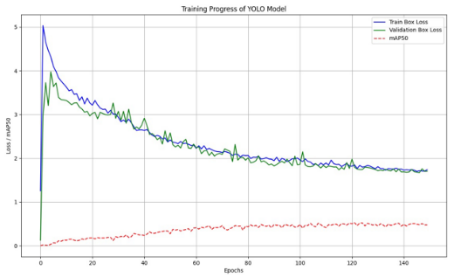


Figure 5. *Training, validation, and mAP50 result for loose dataset- YOLOv8 Original*

The analysis of the training progression for the YOLO model, as depicted in the Figure 5, illustrates a consistent decrease in both training and validation box losses over epochs, indicating effective learning and generalization. Notably, the training box loss decreased from 1.255 to 1.7373, while the validation box loss showed a reduction from 0.12321 to 1.7033. Concurrently, the model's mean Average Precision (mAP50) improved significantly, starting from 0 and reaching a peak of 0.5243, demonstrating a substantial enhancement in the model's ability to detect objects accurately. These trends suggest that the model adjustments and training regimen were well-tuned, leading to improved performance and robustness in object detection tasks.

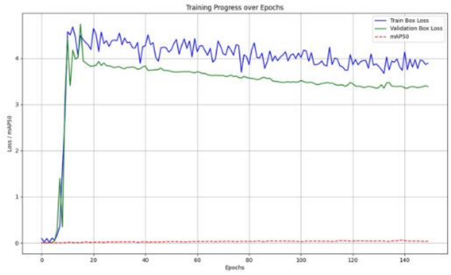


Figure 6. *Training, validation, and mAP50 result for tight dataset- YOLOv8 Tiny Model*

Figure 6 provides a detailed examination of model training dynamics, focusing on training box loss, mAP50(B), and validation box loss. Initially, the training box loss is low, suggesting overfitting, with near-zero validation loss and mAP50(B) indicating poor generalization. As training progresses, both training and validation losses increase, peaking around 4.5763 and 4.7434, respectively, before stabilizing. This pattern suggests difficulty in generalizing from training to validation data. The mAP50(B) gradually improves, peaking at 0.0621, indicating incremental gains in detection accuracy. Overall, the data highlights the need for better generalization strategies to enhance model robustness.

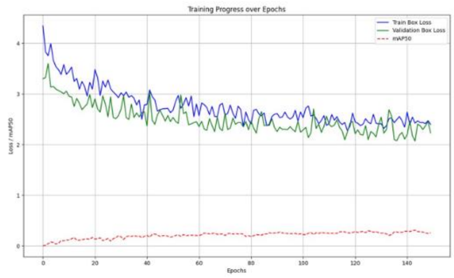


Figure 7. *Training, validation, and mAP50 result for tight dataset - YOLOv8n.pt*

Figure 7 illustrates the progression of a model's training, focusing on training box loss, metrics/mAP50(B), and validation box loss. Initially, the training box loss decreases steadily, indicating effective learning. The mAP50(B) improves significantly, peaking at 0.31074, reflecting enhanced model accuracy. Validation box loss shows a downward trend, suggesting better generalization. However, fluctuations in both training and validation losses indicate overfitting challenges. The model's performance stabilizes towards the end, with consistent mAP50(B) improvements and reduced validation losses, demonstrating successful optimization and increased robustness. Further tuning could enhance stability and accuracy.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (166-172)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.46063

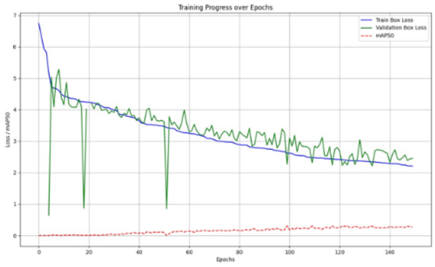


Figure 8. *Training, validation, and mAP50 result for tight dataset - YOLOv8ori.yaml*

Figure 8 illustrates a general trend of decreasing training box loss over time, from 6.7454 to approximately 2.2081, indicating progressive learning and adaptation by the model. Despite the initial absence of meaningful mAP50(B) values, a gradual increase is observed as training progresses, with a peak value of 0.30951, suggesting enhanced object detection accuracy. However, the validation box loss presents a more varied pattern, with values fluctuating across epochs, indicating potential overfitting or model instability in certain phases. Notably, several entries for validation box loss are recorded as 'nan', which could signify issues with data collection or model evaluation at those points. The overall data underscores the importance of continuous monitoring and adjustment in training deep learning models to optimize performance and ensure robustness.

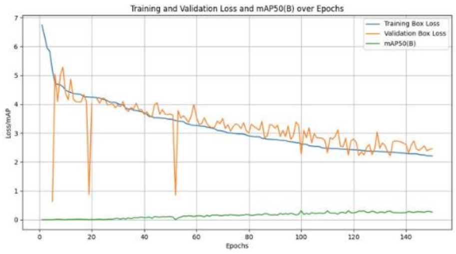


Figure 9. *Training, validation, and mAP50 result for loose dataset-YOLOv8 Tiny Model*

In Figure 9, the analysis of the training dynamics reveals a significant improvement in model performance over epochs. The training box loss consistently decreased from 0.62841 to 1.7945, indicating effective learning and adaptation by the model. Concurrently, the validation box loss similarly showed a downward trend from 5.9679 to 1.8283, suggesting that the model's generalization to unseen data improved significantly, a crucial factor for robust real-world applications. Notably, the mAP50(B) metric increased from 0.00047 to 0.44592, reflecting substantial enhancements in the model's object detection accuracy. These trends highlight the model's evolving capability to balance training and validation performance, optimizing for both accuracy and generalizability.

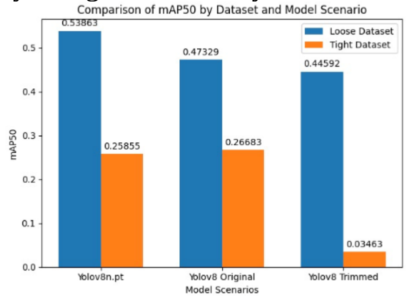


Figure 10. *Final result of model comparison in varying dataset*

The comparative analysis of mAP50 across different YOLOv8 model configurations and dataset constraints reveals distinct performance patterns. The Yolov8n.pt model excels in the loose dataset scenario with a mAP50 of 0.53863, significantly outperforming its tight dataset counterpart (mAP50 = 0.25855). The Yolov8 Original model shows more consistent results between the loose (mAP50 = 0.47329) and tight (mAP50 = 0.26683) datasets. The Yolov8 Trimmed model, however, demonstrates a substantial drop in performance on the tight dataset (mAP50 = 0.03463) compared to the loose dataset (mAP50 = 0.44592), suggesting potential limitations in handling constrained or complex scenarios. This indicates that the model's efficacy is highly context-dependent, with significant variability based on the dataset's characteristics.

CONCLUSION

The study provides a comprehensive analysis of the interplay between model variations and dataset characteristics in enhancing small object detection accuracy. The YOLOv8n.pt model, when applied to a loosely distributed dataset, achieves the highest mean Average Precision (mAP) of 53.86%, underscoring the importance of selecting appropriate model-dataset combinations. The performance disparity observed across different configurations, particularly the substantial decline in the YOLOv8 Trimmed model's accuracy on tight datasets, highlights the context-dependent efficacy of these models. These findings emphasize the necessity of strategic model and dataset alignment to optimize detection tasks, offering critical insights for deploying neural network architectures in complex detection scenarios.

REFERENCES

[1] A. H. Kristanto *et al.*, "Survey on egg and fry production of giant gourami (*Osphronemus goramy*): Current rearing practices and recommendations for future research," *J. World Aquac. Soc.*, vol. 51, no. 1, pp. 119–138, Feb. 2020, doi: 10.1111/jwas.12647.

[2] A. Bimantara, M. N. Zuhdi, A. P. P. W. Tunggali, and R. A. Manasikana, "Intensifikasi Budidaya Perikanan (Osphronemus Gouramy) Mandiri Berbasis Sumberdaya Lokal sebagai Upaya Peningkatan Produktivitas POKDAKAN Mina Sida Karya," *War. LPM*, pp. 58–70, Mar. 2024, doi: 10.23917/warta.v27i1.3388.

[3] J. Zhou, T. Su, K. Li, and J. Dai, "Small Target-YOLOv5: Enhancing the Algorithm for Small Object Detection in Drone Aerial Imagery Based on YOLOv5," *Sensors*, vol. 24, no. 1, p. 134, Dec. 2023, doi: 10.3390/s24010134.

[4] A. Kuswantori, T. Suesut, W. Tangsrirat, G. Schleining, and N. Nunak, "Fish Detection and Classification for Automatic Sorting System with an Optimized YOLO Algorithm," *Appl. Sci.*, vol. 13, no. 6, p. 3812, Mar. 2023, doi: 10.3390/app13063812.

[5] X. Tang, C. Ruan, X. Li, B. Li, and C. Fu, "MSC-YOLO: Improved YOLOv7 Based on Multi-Scale Spatial Context for Small Object Detection in UAV-View," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 983–1003, 2024, doi: 10.32604/cmc.2024.047541.

[6] P. Zhang, H. Deng, and Z. Chen, "RT-YOLO: A Residual Feature Fusion Triple Attention Network for Aerial Image Target Detection," *Comput. Mater. Contin.*, vol. 75, no. 1, pp. 1411–1430, 2023, doi: 10.32604/cmc.2023.034876.

[7] Visen and N. Charibaldi, "Penerapan Object Detection Menggunakan Deep Learning Yolov8 Untuk Mengidentifikasi Sampah Anorganik (Maksimal Sepuluh Objek) Dalam Satu Citra," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 12, no. 1, pp. 195–202, Feb. 2025, doi: 10.25126/jtiik.20251219012.

[8] D. Triyanto, M. Zidan, M. Wahyudi, L. Pujiastuti, and S. Sumanto, "Pengembangan Sistem Deteksi Objek Botol Real-Time dengan YOLOv8 untuk Aplikasi Vision," *Indones. J. Comput. Sci.*, vol. 3, no. 1, pp. 44–50, Apr. 2024, doi: 10.31294/ijcs.v3i1.6070.

[9] V. Q. Nghiem, H. H. Nguyen, and M. S. Hoang, "LEAF-YOLO: Lightweight Edge-Real-Time Small Object Detection on Aerial Imagery," *Intell. Syst. Appl.*, vol. 25, p. 200484, Mar. 2025, doi: 10.1016/j.iswa.2025.200484.

[10] Q. Fan, Y. Li, M. Deveci, K. Zhong, and S. Kadry, "LUD-YOLO: A novel lightweight object detection network for unmanned aerial vehicle," *Inf. Sci.*, vol. 686, p. 121366, Jan. 2025, doi: 10.1016/j.ins.2024.121366.

[11] H. Li *et al.*, "Robust detection of farmed fish by fusing YOLOv5 with DCM and ATM," *Aquac. Eng.*, vol. 99, p. 102301, Nov. 2022, doi: 10.1016/j.aquaeng.2022.102301.

[12] H. Wang, S. Zhang, S. Zhao, Q. Wang, D. Li, and R. Zhao, "Real-time detection and tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN++," *Comput. Electron. Agric.*, vol. 192, p. 106512, Jan. 2022, doi: 10.1016/j.compag.2021.106512.

[13] M. Wei, K. Chen, F. Yan, J. Ma, K. Liu, and E. Cheng, "YOLO-ESFM: A multi-scale YOLO algorithm for sea surface object detection," *Int. J. Nav. Archit. Ocean Eng.*, vol. 17, p. 100651, 2025, doi: 10.1016/j.ijnaoe.2025.100651.

[14] R. Efendi, N. L. Tusya'diah, and R. Faurina, "Detecting Palm Oil Deficiencies: A Study of Boron, Nitrogen, Potassium, And Magnesium Deficiencies Using Yolov5 Model," *J. Tek. Inform.*, vol. 16, no. 2, pp. 141–150, Dec. 2023, doi: 10.15408/jti.v16i2.33523.

[15] O. C. Koyun, R. K. Keser, İ. B. Akkaya, and B. U. Töreyin, "Focus-and-Detect: A Small Object Detection Framework for Aerial Images," *Signal Process. Image Commun.*, vol. 104, p. 116675, May 2022, doi: 10.1016/j.image.2022.116675.