MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 2, pp. 682–692 (2025)

Development of a deep learning-based system in Python 3.9 with YOLOv5: A case study on real-time fish counting based on classification

Multajam R.[1], Ayob A. F. M.[1], Jamaludin Sh.[1], Mada Sanjaya W. S.[2], Sambas A.[3,4], Rusyn V.[5], Senyk Y. A.[6]

[1] *Universiti Malaysia Terengganu, Kuala Nerus, 21030, Malaysia*
[2] *Universitas Islam Negeri Sunan Gunung Djati, Bandung, 40614, Indonesia*
[3] *Universiti Sultan Zainal Abidin, Campus Besut, 22200, Terengganu, Malaysia*
[4] *Universitas Muhammadiyah Tasikmalaya, Tasikmalaya, 46196, Indonesia*
[5] *Yuriy Fedkovych Chernivtsi National University, 2 Kotsjubynsky Str., Chernivtsi, 58012, Ukraine*
[6] *National Forestry University of Ukraine, 103 Gen. Chuprynky Str., 79057, Lviv, Ukraine*

(Received 4 April 2025; Revised 24 June 2025; Accepted 26 June 2025)

This study developed a real-time fish classification and counting system for six types of fish using the YOLOv5 machine learning model with high accuracy. The system achieved an F1-score of 0.87 and a precision confidence curve with an all-classes value of 1.00 at a confidence level of 0.920, demonstrating the model's reliability in object detection and classification. Real-time testing showed that the system could operate quickly and accurately under various environmental conditions with an average inference speed of 30 FPS. However, several challenges remain, such as sensitivity to low-light conditions. Overall, this system has significant potential for applications in aquaculture, particularly for automated and real-time fish monitoring. With compatibility through the ONNX format, the system is also flexible for integration into IoT-based devices or cross-platform applications, providing a solid foundation for further advancements in computer vision-based fish monitoring technology.

Keywords: *deep learning; YOLOv5; fish classification methods; computer vision; real-time fish counting.*

2010 MSC: 68T10, 68T45, 92B20, 68U10 **DOI:** 10.23939/mmc2025.02.682

1. Introduction

The advancement of real-time camera technology and artificial intelligence [1, 2] has ushered in new opportunities in the fisheries industry [3,4]. The combination of real-time cameras with artificial neural networks (ANN) and deep learning techniques [5] promises an efficient and accurate solution for real-time fish detecting [6]. The Artificial Neural Network (ANN) then indicate whether the object in the camera's image is a fish [7]. The real-time use of cameras combined with deep learning [8] will result in an integrated approach to detect and classify fish in underwater videos [9] thus, in the future, this research can be applied to underwater robot systems to analyze the behavior of various fish species [10].

This research aims to implement real-time camera technology in conjunction with deep learning for direct fish counting in fisheries environments [4]. By harnessing the power of machine learning algorithms [11], this study not only seeks an efficient alternative to manual counting methods but also aims to pave the way for a more holistic and intelligent fisheries management approach.

One way to manually count the number of fish is by placing several fish on a container placed above a water tank or pond, then pushing some fish into a channel and counting them one by one [12]. Meanwhile, fish counting using artificial intelligence technology, provides the implementation of Artificial Neural Networks (ANN) [13]. Deep learning methods [14,15] are employed to classify fish species [9,16].

Therefore, the use of real-time cameras combined with deep learning [17] can result in accurate and real-time fish counting.

The results of fish classification can provide significant benefits in various aspects, including faster and more accurate identification of fish species [18], monitoring fish populations, and supporting fisheries resource management to maintain environmental sustainability and ensure an adequate supply of fish for communities [19].

2. Theoretical background

Technological advancements have significant impacts on various sectors [20], including the cultivation of flora and fauna. The use of artificial neural networks has achieved great success, particularly in handling text, images, videos, etc. [21]. Furthermore, the artificial neural networks can be employed in sorting machines and the classification of fruit ripeness [22]. Furthermore, the advanced artificial intelligence can be applied in robotics systems to recognize human faces in real-time using cameras [23], by embedding a database into the system, enabling the robot's camera to recognize these facial patterns.

Deep learning is a subset of machine learning that employs artificial neural networks with multiple layers (deep neural networks) to analyze data [24,25]. This method can capture complex patterns within data and is widely used in applications such as image recognition, natural language processing [26,27], and object counting [28].

2.1. You Look Only Once (YOLO) v5

YOLO (You Only Look Once) [29] is a renowned object detection model known for its high speed and efficient approach to address issues found in conventional object detection architectures [30]. In YOLO, object detection and class probability predictions are performed directly on the image with a single evaluation [31], avoiding the complexity of conventional detection pipelines. YOLO v5, as a recent version, continues to improve performance and accuracy by utilizing regression as the detection approach and optimizing the network architecture. With 24 convolutional layers followed by 2 fully connected layers [32], the model is capable of making global predictions about the image, resulting in fewer false positive predictions. This is illustrated in Figure 1.

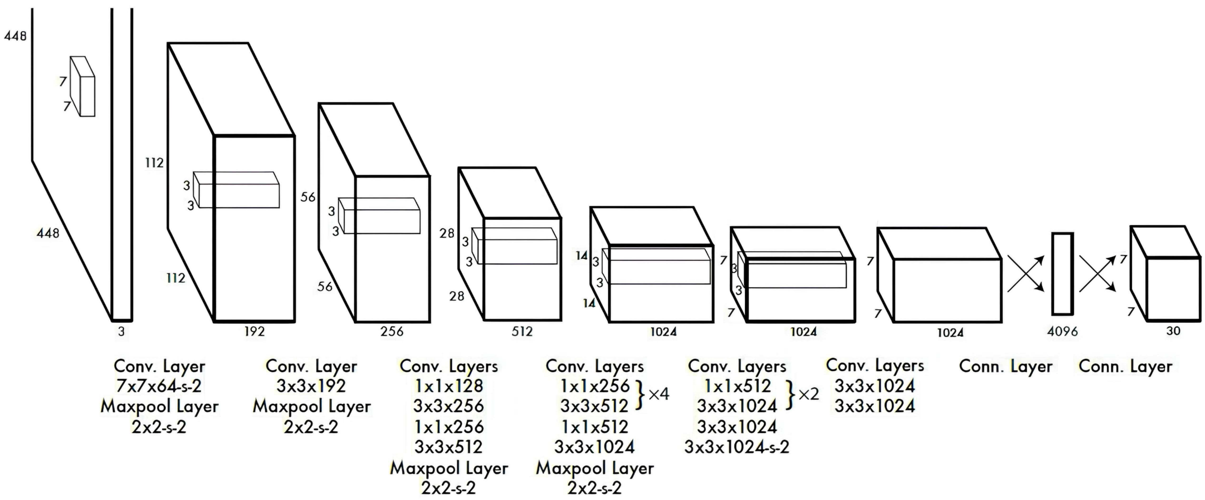

Fig. 1. The Architecture of YOLO.

In Figure 1, it can be observed that the alternating 1×1 convolutional layers are used to reduce the feature space from the previous layers [29]. Pretraining is conducted on the convolutional layers for the ImageNet classification task with half resolution (224×224 input images), and then the resolution is doubled for the detection process [29]. In addition, YOLO can also be applied to detect objects in water [33] and doing it in real-time object detection in videos on embedded devices [34].

2.2. The fish classification method using YOLOv5

With the advancement of artificial intelligence, particularly in the field of computer vision [35], the object classification process has significantly improved in efficiency and accuracy [36]. This study adopts a classification method using the YOLOv5 algorithm, a renowned object detection model known for its ability to detect and classify objects in real-time. The application of YOLOv5 in classifying fish, such as Koi fish and Goldfish [37], represents a significant step toward supporting various applications, including aquaculture, environmental monitoring, and fishery-based industries. With this method, of fish species identification can be automated, which delivers fast and accurate results while providing a more efficient solution compared to manual methods. This research outlines systematic steps in utilizing YOLOv5, ranging from data collection and annotation to model training and performance evaluation using relevant metrics [38]. The outcomes of this study are expected to contribute meaningfully to the development of computer vision-based classification technology [39], particularly in the fisheries sector, and inspire further research.

2.3. Evaluation of model

The commonly used evaluation metric is accuracy, which measures how often the model provides correct predictions in classifying images of an object. This accuracy approach is calculated by dividing the number of correct predictions by the total number of samples [40]. The commonly used metrics in cases of image classification when the number of positive and negative samples is imbalanced include precision, recall, and F1-Score.

Precision is one of the essential evaluation metrics in assessing the performance of a classification system. Precision provides an insight into how accurately the model identifies positive outcomes. The precision formula is expressed as:

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}. \tag{1}$$

Precision measures, among all the model's positive predictions, what percentage is truly relevant or correct. High precision indicates that the model tends to provide accurate positive predictions and minimizes false positive predictions.

Recall, also known as Sensitivity or True Positive Rate, is a crucial evaluation metric in assessing the performance of a classification system. Recall provides an insight into how well the model can recognize all positive outcomes that should be identified. The recall formula is expressed as:

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}. \tag{2}$$

Recall measures, among all the actual positive instances, what percentage is successfully identified by the model. High recall indicates that the model can detect most of the actual positive instances.

$$\text{Specificity} = \frac{\text{TrueNegative}}{\text{FalseNegative} + \text{TrueNegative}}. \tag{3}$$

Specificity indicates the model's ability to correctly classify all negative outcomes. Meanwhile, F1-Score provides a balanced measurement between precision and recall by combining. F1-Score provides a holistic overview of the model's performance by considering both aspects. F1-Score is calculated using the harmonic mean of precision and recall:

$$\text{F1-Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}. \tag{4}$$

F1-Score has a range of values between 0 and 1, where a value of 1 indicates perfect performance with optimal precision and recall. F1-Score is particularly valuable when there is an imbalance between positive and negative classes in the dataset, as it can provide a more accurate representation of the overall model performance.

In its application context, F1-Score is commonly used in various fields, including image processing, object detection, and evaluation of classification model performance in imbalanced class scenarios.

3. Experimental method

3.1. The method and system for fish classification

The fish classification method in this study was carried out using a computer vision-based approach with YOLOv5, an object detection algorithm known for its efficiency and high precision. The classification process includes several key steps, namely, data annotation, model training, and result evaluation. Figure 2 below shows the flow diagram of the classification process for six types of fish using YOLOv5.

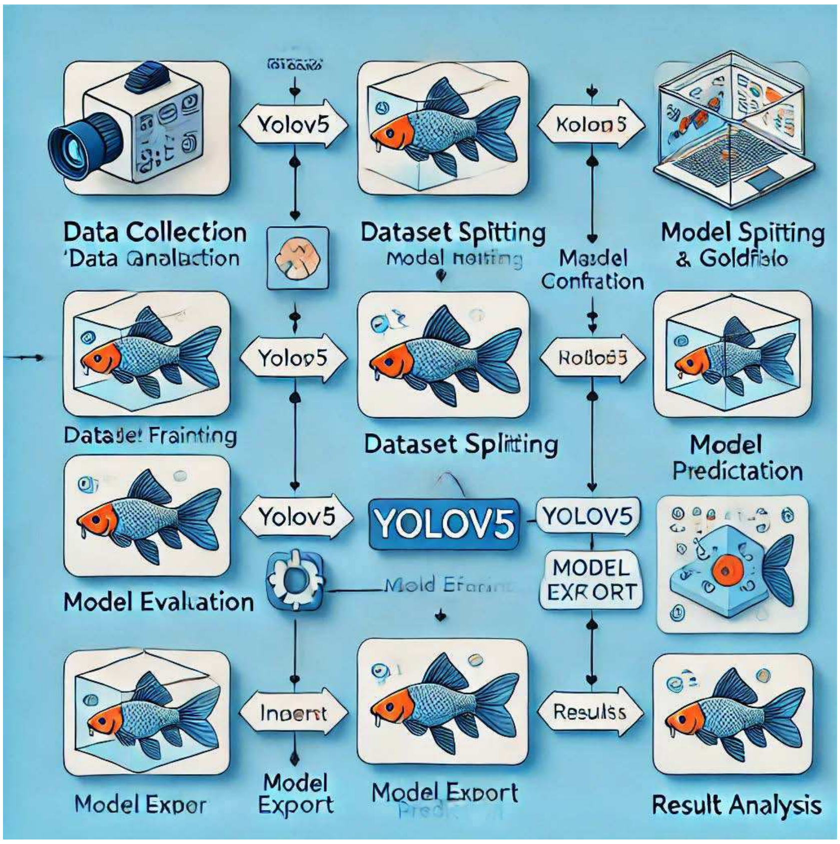

Fig. 2. Flowchart of the fish classification using YOLOv5.

The flow chart above illustrates the systematic steps for classifying fish using YOLOv5. The process begins with data collection, which involves images of fish that include variations in pose, size, background, and lighting. The images are then annotated using Roboflow to create bounding boxes around the objects and label them according to their category. The types of fish classified in this study are Angel Fish, Glow Fish, Golden Fish, Platy Fish, Kaviat Fish, and Koi Fish. The annotated data set is then divided into three subsets: training set (60%), validation set (30%) and test set (10%) to ensure the model's generalization.

YOLOv5 configuration is performed by adjusting the configuration files to match the dataset, and an appropriate model is selected based on the desired accuracy and speed. The model training process is conducted with optimal parameters, such as batch size, learning rate, and number of epochs, followed by evaluation using the validation set and test set to calculate metrics such as precision, recall, F1 score, and graphs such as the confusion matrix. Once training is complete, the best model is exported in formats such as ONNX for cross-platform compatibility and then used to detect and classify fish in new images or videos. The prediction results are analyzed to ensure that the model performs as expected and, if necessary, retraining is carried out with an updated dataset. This diagram provides a clear depiction of the entire fish classification process using YOLOv5.

1) Image Annotations: The image annotation process was carried out to produce a high-quality dataset used for model training. In this study, a total of 214 images of fish consisting of six different

species were annotated using Roboflow, a platform that facilitates the creation of bounding boxes around objects in images. Each image was accurately labeled to distinguish the six fish species, with special attention given to bounding box accuracy to ensure the model could effectively learn the visual characteristics of each class. The annotated dataset was then exported in the ONNX format, which is compatible with YOLOv5, to proceed with the training process.

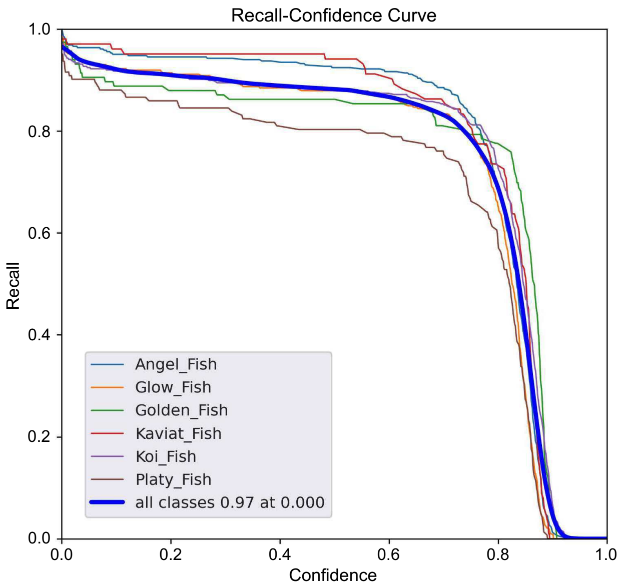

Fig. 3. Recall confidence curve.

The Recall-Confidence Curve above illustrates the relationship between recall values and the confidence level of the YOLOv5 model in classifying six types of fish: Angel Fish, Glow Fish, Golden Fish, Kaviat Fish, Koi Fish, and Platy Fish. From the graph, it can be observed that the highest recall reaches 0.97 at a confidence of 0.000, indicating that the model captures almost all objects when no confidence threshold is applied. As the confidence level increases, recall gradually declines until it drops sharply beyond 0.8. This suggests that higher confidence thresholds result in fewer detections by the model, leading to more unidentified objects (an increase in false negatives). Performance varies across classes, with Kaviat Fish and Angel Fish maintaining higher recall values compared to other classes, while Platy Fish experiences a more rapid decline in recall. To balance recall and precision, selecting an optimal confidence threshold - such as within the range of 0.4 to 0.7 is essential to ensure that the model can detect fish accurately without sacrificing too many correct detections.

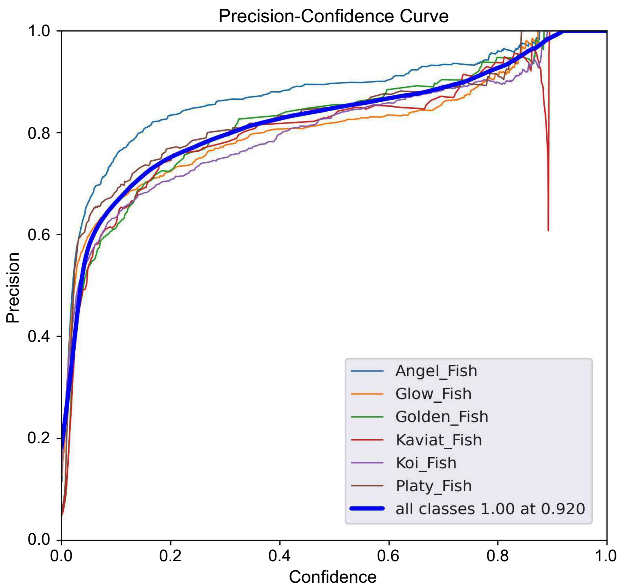

Fig. 4. Precision-confidence curve.

The Precision-Confidence curve shows that precision increases as the confidence threshold rises, reaching 1.00 at 0.920 for all classes. This indicates that at high confidence levels, the model produces fewer detection errors. Angel Fish and Golden Fish exhibit more stable precision compared to other classes, while Kaviat Fish shows slight fluctuations. An optimal confidence threshold of around $0.5-0.8$ can balance precision and recall for more accurate fish classification results.

2) The Model Training with YOLOv5: The model training was conducted using YOLOv5, known for its efficiency and accuracy in object detection tasks. The annotated dataset was divided into three subsets: training set (60%), validation set (30%), and test set (10%) to ensure the model's generalization. Training parameters, such as batch size, learning rate, and the number of epochs, were optimally configured to enable effective learning. During the training process, YOLOv5 iteratively learned patterns from the images in the training set and validated its performance using the validation set. Upon completion of training, the best model was selected based on evaluation metrics such as the F1-Score and exported in the ONNX (Open Neural Network Exchange) format to support cross-platform inference. The F1-Score from the fish classification using YOLOv5 is shown in Figure 5.

In Figure 5, the F1-Confidence Curve graph above illustrates the relationship between the F1 score and the confidence level of the YOLOv5 model in classifying six types of fish: Angel Fish, Glow Fish, Golden Fish, Kaviat Fish, Koi Fish, and Platy Fish. The highest F1 score is achieved at a confidence level of 0.595, with an F1 value of 0.87, indicating an optimal balance between precision and recall across all classes. Each fish species exhibits slightly different performance, with Angel Fish demonstrating the best performance, as its F1 curve remains more stable across various confidence levels, while Golden Fish and Platy Fish experience a more rapid decline. The sharp drop in the F1 score when confidence exceeds 0.8 indicates an increase in false negatives, leading to a decrease in recall and an overall reduction in the F1 score. Therefore, selecting a confidence threshold in the range of 0.595 to 0.7 is considered ideal for maintaining a balance between precision and recall in real-world model applications. Meanwhile, the confusion matrix results for the classification of Koi fish and Goldfish are shown in Figure 6.

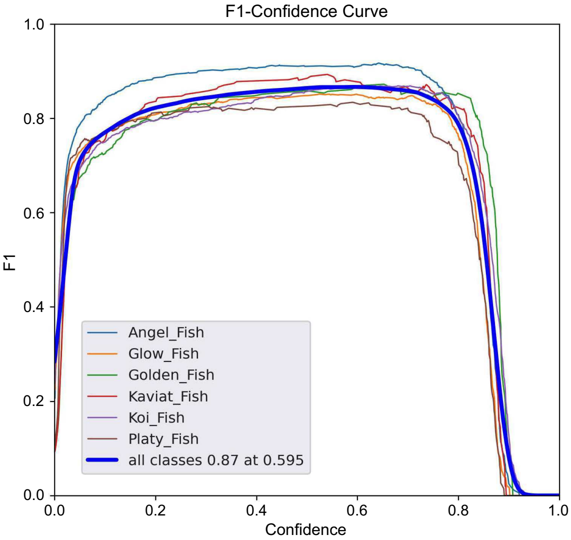

Fig. 5. The F1-Score of fish classification using YOLOv5.

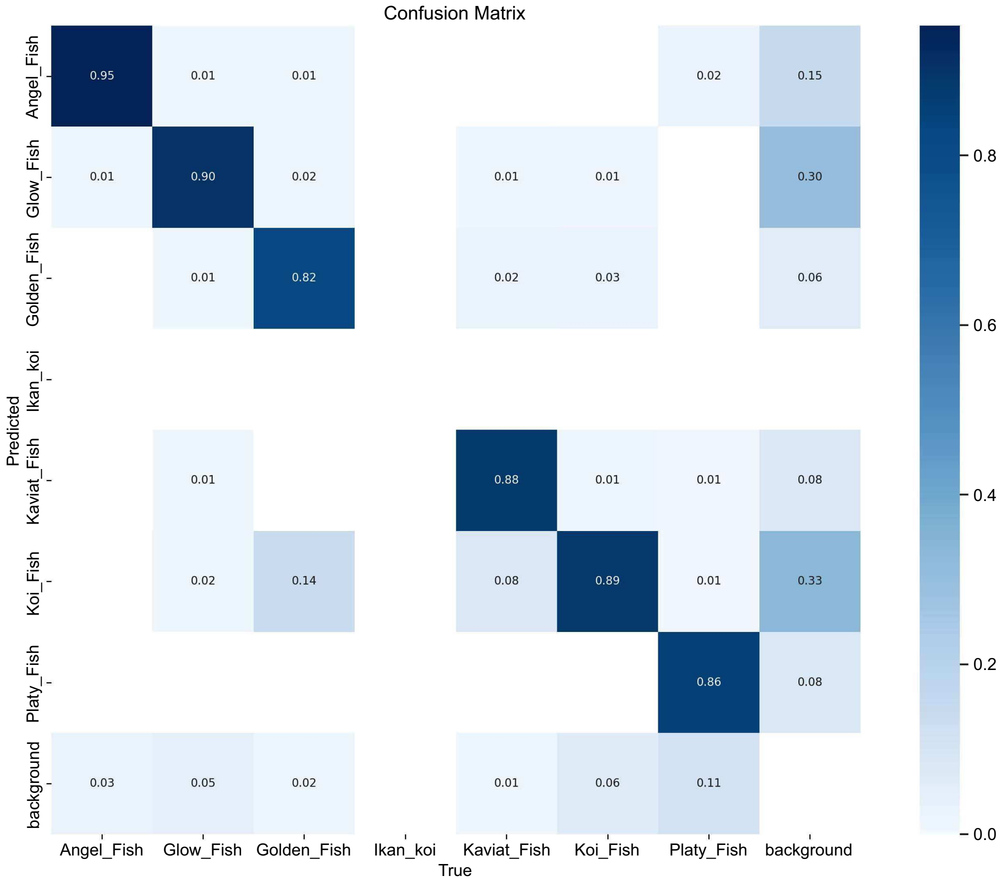

Fig. 6. Confusion matrix.

Based on the displayed confusion matrix, the model performs quite well in classifying several types of fish, such as Angel fish 95%, Glow fish 90%, Koi fish 89%, and Platy fish 86%. However, there are some significant misclassifications, such as Golden Fish, which is classified as Koi fish in 14% of cases, and Glow fish, which is misclassified into the background category at a rate of 30%. These errors are likely caused by visual similarities between fish species, an imbalanced training dataset, and the complexity of the background in the images.

3.2. Experiments and real-time system testing

The experiments and system testing were conducted to evaluate the ability of the YOLOv5 model to count the number of fish based on their classes in real-time using the ONNX file. The model was implemented through a Python-based deep learning method to detect, classify, and count fish from video or live camera feeds. The testing aimed to assess accuracy, response speed, and system stability under various conditions, such as lighting, movement, and background diversity. The flow of the fish classification and counting system testing process is illustrated in Figure 7.

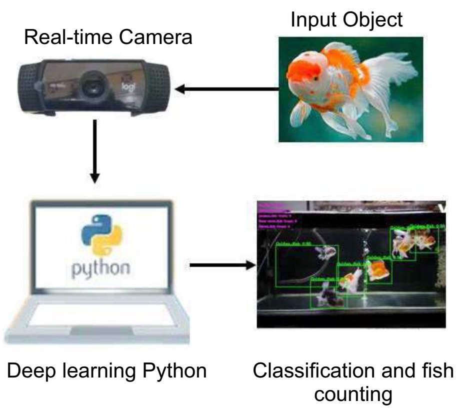

Fig. 7. Experiment for fish counting method.

Figure 7 above illustrates the workflow of a real-time fish counting system using the YOLOv5 model implemented with Python. The process begins with input from a real-time camera that captures images or videos of fish in an aquarium, serving as the system's primary input. The data is then processed using a Python-based deep learning method, where the pre-trained YOLOv5 model, exported in ONNX format, performs detection, classification, and counting of the fish. Each detected fish is marked with a bounding box labeled with its category: Angel fish, Glow fish, Goldfish, Kaviat fish, Koi fish, and Platy fish. The final output is an accurate classification and fish count displayed live on a computer screen, demonstrating an efficient real-time system implementation that can adapt to various environmental conditions.

4. Result and discussion

4.1. Experimental results and system testing

Fig. 8. Results of the fish counting test based on fish classification.

The experimental results show that the YOLOv5 model successfully detects and classifies six types of fish with relatively high accuracy. Based on testing on the test dataset, the model achieved an F1-score of 0.87, reflecting a good balance between precision and recall. The confusion matrix indicates a relatively low classification error rate, with only a few misclassified samples. Additionally, the precision confidence curve shows an all-classes value of 1.00 at a confidence level of 0.920, confirming the accuracy of the developed model in detecting objects with high confidence levels.

Real-time testing was conducted to evaluate the system's reliability in detecting and counting fish based on their classification. The system uses a real-time camera to capture video input, which is then processed through the YOLOv5 model integrated with Python Deep Learning using an ONNX file. This process produces real-time object detection with bounding boxes and labels indicating the fish categories and confidence scores. The testing workflow is illustrated in Figure 8.

In the experiments conducted, the real-time detection results demonstrated excellent model performance in identifying and counting the number of fish in an aquarium.

Figure 8 shows the analysis of fish classification and counting results using the YOLOv5 model. It indicates that the system performs quite well in detecting six types of fish with varying counts, although there are still some challenges in detection accuracy. The model successfully identified 6 Angel fish, 8 Glow fish, 2 Golden fish, 0 Platy fish, 2 Kaviat fish, and 6 Koi fish, with confidence scores ranging from 0.30 to 0.91. However, several factors affect the system's performance, such as overlapping bounding boxes, visual similarities between fish, and complex lighting and background conditions. Some misclassifications were observed, particularly with fish of similar colors, such as Golden fish occasionally being classified as Koi fish, as well as some detections with low confidence scores, indicating model uncertainty in classification. To enhance system accuracy, several improvements can be made, including improving dataset quality through data augmentation, adjusting the confidence score threshold, fine-tuning the model with a more balanced dataset, and using a higher-resolution camera to reduce noise and blur caused by fish movement. Overall, the system shows strong potential for real-time fish detection, with further refinements its accuracy and reliability can be improved to achieve more precise fish classification and counting. The results of the fish counting test can be accessed via the QR Code in Figure 9.

Fig. 9. QR code to link video result of the counting fish.

Figure 9 presents a video of the fish counting test results based on classification. From the results obtained, only platy fish were not detected. This is due to the fish's pattern, which closely resembles the aquarium background, as well as the limited number of platy fish images during the annotation process. Meanwhile, other fish species were successfully detected.

4.2. Real-time system performance evaluation

In real-time testing using a camera, the system accurately detected and counted the number of fish with an average inference speed of 30 FPS (frames per second). Variations in environmental conditions, such as lighting, background, and fish movement, did not significantly affect system performance, although minor errors were observed under dim lighting conditions. The model exported in ONNX format proved compatible with various devices, enabling implementation in IoT-based systems or cross-platform applications.

The real-time system performance evaluation is linked to the results shown in Figures 8 and 9. In Figure 8, the system successfully detected the number of fish based on their classification, with confidence scores ranging from 0.595 to 0.70, while no platy fish were detected. As shown in the video in Figure 9, this is due to the similarity of the platy fish's pattern with the aquarium background. These results indicate that the model is not yet highly accurate due to the limited number of images for platy fish.

4.3. Discussion and analysis

The combination of YOLOv5 with a learning dataset in the ONNX file format has produced a reliable fish detection model. The fact that the precision confidence curve reaches its maximum value indicates that the model has a high tolerance for prediction thresholds, making it suitable for scenarios requiring high accuracy. Each detected fish is assigned a colored bounding box according to its classification, with confidence scores ranging from 0.39 to 0.91. Some fish have low confidence scores, indicating potential detection errors, while Platy Fish were not detected at all, possibly due to their patterns resembling the aquarium background or the limited number of training data samples. Other factors affecting detection accuracy include lighting conditions, fish movement, and the quantity and quality of samples in the training dataset. To enhance system performance, additional training data should be incorporated, especially for fish species that are difficult to detect, along with improved lighting during image capture and the application of data augmentation techniques. Overall, this system is quite effective in detecting and counting fish in real-time, although some aspects still need improvement to achieve better accuracy and reliability.

5. Conclusion

This study successfully developed a real-time classification and counting system for six types of fish using the YOLOv5 machine learning model with high accuracy. The system achieved an F1-score of 0.87 and a precision confidence curve with an all-classes value of 1.00 at a confidence level of 0.920, demonstrating the model's reliability in object detection and classification. Real-time testing showed that the system could operate quickly and accurately under various environmental conditions with an average inference speed of 30 FPS. However, several challenges remain, such as sensitivity to low-light conditions. Overall, this system has significant potential for applications in aquaculture, particularly for automated and real-time fish monitoring. With compatibility via the ONNX format, the system is also flexible for integration into IoT-based devices or cross-platform applications, providing a solid foundation for further advancements in computer vision-based fish monitoring technology.

[1] Chen P.-H. C., Gadepalli K., MacDonald R., Liu Y., Kadowaki S., Nagpal K., Kohlberger T., Dean J., Corrado G. S., Hipp J. D., Mermel C. H., Stumpe M. C. An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis. Nature Medicine. **25**, 1453–1457 (2019).

[2] Multajam R., Ayob A. F. M., Mada Sanjaya W. S., Sambas A., Rusyn V. Color-based image processing techniques for laser range finder: a comparative study on air and water distance detection. Sixteenth International Conference on Correlation Optics. 129381A (2024).

[3] Isabelle D. A., Westerlund M. A review and categorization of artificial intelligence-based opportunities in wildlife, ocean and land conservation. Sustainability. **14** (4), 1979 (2022).

[4] Khokher M. R., Little L. R., Tuck G. N., Smith D. V., Qiao M., Devine C., O'Neill H., Pogonoski J. J., Arangio R., Wang D. Early lessons in deploying cameras and artificial intelligence technology for fisheries catch monitoring: where machine learning meets commercial fishing. Canadian Journal of Fisheries and Aquatic Sciences. **79** (2), 257–266 (2022).

[5] Ismail N., Malik O. A. Real-time visual inspection system for grading fruits using computer vision and deep learning techniques. Information Processing in Agriculture. **9** (1), 24–37 (2022).

[6] Hu J., Zhao D., Zhang Y., Zhou C., Chen W. Real-time nondestructive fish behavior detecting in mixed polyculture system using deep-learning and low-cost devices. Expert Systems with Applications. **178**, 115051 (2021).

[7] Rico-Diaz A. J., Rabunal J. R., Gestal M., Mures O. A., Puertas J. An application of fish detection based on eye search with artificial vision and artificial neural networks. Water. **12** (11), 3013 (2020).

[8] Unlu E., Zenou E., Riviere N., Dupouy P.-E. Deep learning-based strategies for the detection and tracking of drones using several cameras. IPSJ Transactions on Computer Vision and Applications. **11**, 7 (2019).

[9] Jalal A., Salman A., Mian A., Shortis M., Shafait F. Fish detection and species classification in underwater environments using deep learning with temporal information. Ecological Informatics. **57**, 101088 (2020).

[10] Boudhane M., Nsiri B. Underwater image processing method for fish localization and detection in submarine environment. Journal of Visual Communication and Image Representation. **39**, 226–238 (2016).

[11] Reynard D., Shirgaokar M. Harnessing the power of machine learning: Can twitter data be useful in guiding resource allocation decisions during a natural disaster? Transportation Research Part D: Transport and Environment. **77**, 449–463 (2019).

[12] Klapp I., Arad O., Rosenfeld L., Barki A., Shaked B., Zion B. Ornamental fish counting by non-imaging optical system for real-time applications. Computers and Electronics in Agriculture. **153**, 126–133 (2018).

[13] Zhang L., Li W., Liu C., Zhou X., Duan Q. Automatic fish counting method using image density grading and local regression. Computers and Electronics in Agriculture. **179**, 105844 (2020).

[14] Liu H., Lang B. Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences. **9** (20), 4396 (2019).

[15] Hong S., Zhou Y., Shang J., Xiao C., Sun J. Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review. Computers in Biology and Medicine. **122**, 103801 (2020).

[16] Iqbal M. A., Wang Z., Ali Z. A., Riaz S. Automatic fish species classification using deep convolutional neural networks. Wireless Personal Communications. **116**, 1043–1053 (2021).

[17] Wang D., Li W., Liu X., Li N., Zhang C. UAV environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution. Computers and Electronics in Agriculture. **175**, 105523 (2020).

[18] Li X., Shang M., Qin H., Chen L. Fast accurate fish detection and recognition of underwater images with fast R-CNN. OCEANS 2015 - MTS/IEEE Washington. 1–5 (2015).

[19] Brownscombe J. W., Hyder K., Potts W., Wilson K. L., Pope K. L., Danylchuk A. J., Cooke S. J., Clarke A., Arlinghaus R., Post J. R. The future of recreational fisheries: Advances in science, monitoring, management, and practice. Fisheries Research. **211**, 247–255 (2019).

[20] Khan A. N., En X., Raza M. Y., Khan N. A., Ali A. Sectorial study of technological progress and CO_2 emission: Insights from a developing economy. Technological Forecasting and Social Change. **151**, 19862 (2020).

[21] Fan F.-L., Xiong J., Li M., Wang G. On interpretability of artificial neural networks: A survey. IEEE Transactions on Radiation and Plasma Medical Sciences. **5** (6), 741–760 (2021).

[22] Abdul Aziz M., Bukhari W., Sukhaimie M., Izzuddin T., Norasikin M., Rasid A., Bazilah N. Development of smart sorting machine using artificial intelligence for chili fertigation industries. Journal of Automation Mobile Robotics and Intelligent Systems. **15**, 44–52 (2021).

[23] Sanjaya W. M., Anggraeni D., Zakaria K., Juwardi A., Munawwaroh M. The design of face recognition and tracking for human-robot interaction. 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE). 315–320 (2017).

[24] Chahal A., Gulia P. Machine Learning and Deep Learning. International Journal of Innovative Technology and Exploring Engineering. **8** (12), 4910–4914 (2019).

[25] Choi R. Y., Coyner A. S., Kalpathy-Cramer J., Chiang M. F., Campbell J. P. Introduction to machine learning, neural networks, and deep learning. Translational Vision Science & Technology. **9** (2), 14 (2020).

[26] Torfi A., Shirvani R. A., Keneshloo Y., Tavaf N., Fox E. A. Natural language processing advancements by deep learning: A survey. Preprint arXiv:2003.01200 (2020).

[27] Dong S., Wang P., Abbas K. A survey on deep learning and its applications. Computer Science Review. **40**, 100379 (2021).

[28] Akcay H. G., Kabasakal B., Aksu D., Demir N., Oz M., Erdogan A. Automated bird counting with deep learning for regional bird distribution mapping. Animals. **10** (7), 1207 (2020).

[29] Redmon J., Divvala S., Girshick R., Farhadi A. You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 779–788 (2016).

[30] Zhao Z.-Q., Zheng P., Xu S.-T., Wu X. Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems. **30** (11), 3212–3232 (2019).

[31] Jing L., Yang X., Tian Y. Video you only look once: Overall temporal convolutions for action recognition. Journal of Visual Communication and Image Representation. **52**, 58–65 (2018).

[32] Du J. Understanding of object detection based on CNN family and YOLO. Journal of Physics: Conference Series. **1004**, 012029 (2018).

[33] Ayob A., Khairuddin K., Mustafah Y., Salisa A., Kadir K. Analysis of pruned neural networks (mobilenetv2-yolo v2) for underwater object detection. 11th National Technical Seminar on Unmanned System Technology 2019. 87–98 (2021).

[34] Shafiee M. J., Chywl B., Li F., Wong A. Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video. Journal of Computational Vision and Imaging Systems. **3** (1), (2017).

[35] Patricio D. I., Rieder R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Computers and Electronics in Agriculture. **153**, 69–81 (2018).

[36] Diwan T., Anirudh G., Tembhurne J. V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. Multimedia Tools and Applications. **82**, 9243–9275 (2023).

[37] Multajam R., Ayob A. F. M., Mada Sanjaya W. S., Sambas A., Rusyn V., Samila A. Real-time detection and classification of fish in underwater environment using YOLOV5: A comparative study of deep learning architectures. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska. **14**, 91–95 (2024).

[38] Ajayi O. G., Ashi J., Guda B. Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images. Smart Agricultural Technology. **5**, 100231 (2023).

[39] Abhijit Akhil S., Kumar V. A., Jose B. K., Abubeker K. Computer vision assisted bird–eye chilli classification framework using yolo v5 object detection model. Power Engineering and Intelligent Systems. 217–226 (2023).

[40] Mada Sanjaya W. S. Deep Learning Citra Medis Berbasis Pemrograman Python. BOLABOT (2023).

Розробка глибинного навчання із використанням Python 3.9 та YOLOv5: приклад підрахунку риби в реальному часі на основі класифікації

Мультаджам Р.[1], Айоб А. Ф. М.[1], Джамалудін Ш.[1], Мада Санджая В. С.[2], Самбас А.[3,4], Русин В.[5], Сеник Ю. А.[6]

[1] *Університет Малайзія Теренггану, Куала Нерус, 21030, Малайзія*
[2] *Університет Іслам Негері Сунан Гунунг Джаті, Бандунг, 40614, Індонезія*
[3] *Університет Султан Зайнал Абідін, Кампус Бесут, 22200, Теренггану, Малайзія*
[4] *Університет Мухаммадія Тасікмалая, Тасікмалая, 46196, Індонезія*
[5] *Чернівецький національний університет імені Юрія Федьковича, вул. Коцюбинського, 2, Чернівці, 58012, Україна*
[6] *Національний лісотехнічний університет України, вул. Ген. Чупринки, 103, 79057, Львів, Україна*

У цьому дослідженні розроблено систему класифікації та підрахунку риб у реальному часі для шести типів риб з використанням моделі машинного навчання YOLOv5 з високою точністю. Система досягла значення F1-міри 0.87 та показала точність класифікації на рівні 1.00 за усіма класами при порозі впевненості 0.920, що підтверджує надійність моделі у виявленні та класифікації об'єктів. Під час тестування в реальному часі встановлено, що система здатна швидко та точно працювати за різних умов середовища із середньою швидкістю обробки 30 кадрів за секунду. Проте залишаються певні виклики, зокрема чутливість до умов недостатнього освітлення. Загалом система має значний потенціал для використання в аквакультурі, особливо для автоматизованого моніторингу риб у реальному часі. Завдяки сумісності з форматом ONNX, систему також можна гнучко інтегрувати в пристрої на базі Інтернету речей (IoT) або міжплатформні застосунки, що створює надійне підґрунтя для подальшого розвитку технологій комп'ютерного зору у сфері моніторингу риб.

Ключові слова: *глибинне навчання; YOLOv5; методи класифікації риби; комп'ютерний зір; підрахунок риби в реальному часі.*