AI-Based Shrimp Detection and Size Estimation Using Smartphone Sensors

Yu-Yu Chen*[1], Eka Sari Oktarina[2], Valentinus Roby Hananto[1], and Victor Kryssanov[1]

[1] Graduate School of Information Science and Engineering, Ritsumeikan University, Osaka, Japan
is0590he@ed.ritsumei.ac.jp
[2] Department of Computer Engineering, Telkom University, Surabaya, Indonesia

Abstract. Shrimp aquaculture is a major contributor to global seafood production, where accurate monitoring of shrimp size and weight is essential for feeding management, growth evaluation, and harvest planning. In modern farming conditions, shrimp size is commonly assessed through manual sampling, which requires physical handling of individual shrimps. This process is labor-intensive and time-consuming, and repeated handling may also induce stress that negatively affects shrimp growth and survival.

This paper presents a smartphone-based system for whiteleg shrimp (*Litopenaeus vannamei*) detection, length estimation, and average weight prediction using computer vision methods. The proposed system employs a YOLOv8 object detection model to identify shrimps in images captured by standard smartphone cameras under actual aquaculture conditions. Shrimp length is estimated by converting pixel measurements obtained from detected bounding boxes into real-world dimensions through reference calibration using an object of a known size. Image-level features extracted from the detected shrimp regions are then used by a Support Vector Regression model to predict average shrimp weight.

The detection model was trained using 6,350 images and tested using a separate set of 280 images. Length estimation was evaluated using 29 separate images, while weight prediction was evaluated using another 26 images. The detection model achieved a precision of 0.988, a recall of 0.927, an mAP@0.5 of 0.982, and an mAP@0.5:0.95 of 0.877. Length estimation achieved an MAE of 1.54 cm, an RMSE of 1.63 cm, and a MAPE of 14.6%. Average weight prediction achieved an MAE of 1.05 g, an RMSE of 1.78 g, and a MAPE of 7.5%.

The results demonstrate the feasibility of combining shrimp detection, reference calibration, length estimation, and average weight prediction in one smartphone workflow under the tested outdoor farm conditions.

Keywords: Shrimp aquaculture · Object detection · YOLOv8 · Length estimation · Weight prediction · Smartphone imaging

© The Author(s) 2026
M. Murtadha et al. (eds.), *Proceedings of the 2026 11th International Conference on Sustainable Information Engineering and Technology (SIET 2026)*, Atlantis Highlights in Computer Sciences 26,
https://doi.org/10.2991/978-94-6239-828-3_28

1 Introduction

Shrimp aquaculture plays an important role in global food production, particularly in Asia, where shrimp farming supports both local consumption and international trade [1]. Accurate information about shrimp length and weight is important for growth evaluation, feeding management, and harvest planning.

In current aquaculture practice, shrimp growth is commonly evaluated through manual sampling. Farmers capture shrimps from ponds and measure their length or weight by hand. Although this method is widely used, it requires physical handling of shrimps and depends on manual labor and operator experience. Repeated handling may also cause stress and may not be suitable for frequent monitoring. In outdoor shrimp farms, image acquisition can be affected by natural lighting, reflections on the water surface, turbidity, shrimp posture, and complex backgrounds. These conditions make automatic shrimp detection and size estimation difficult under actual farming conditions.

Computer vision and deep learning methods have increasingly been applied to aquaculture monitoring [2]. However, many existing studies focus on only one task, require controlled imaging conditions, or depend on underwater cameras and specialized equipment. Practical smartphone systems that combine shrimp detection, reference calibration, length estimation, and average weight prediction under outdoor farm conditions remain limited.

Based on this research problem, the study addresses three research questions. RQ1 examines how accurately YOLOv8 can detect whiteleg shrimps (*Litopenaeus vannamei*) in smartphone images collected under outdoor farm conditions. RQ2 examines the accuracy of estimating per-image mean shrimp length using detected bounding boxes and a reference bracket of a known size. RQ3 examines whether image-level features extracted from detected shrimp regions can be used by Support Vector Regression to estimate average shrimp weight on previously unseen images.

The proposed system first detects shrimps using YOLOv8. Reference calibration then converts bounding-box measurements into estimated shrimp lengths. Six image-level features are processed by an SVR model to predict average shrimp weight. The detection, length, and weight results are finally displayed through an Android application.

The main contribution of this study is the integration of shrimp detection, reference calibration, length estimation, average weight prediction, and mobile visualization into one smartphone workflow. The detection, length, and weight tasks are evaluated using separate datasets, including an independent weight test set. The contribution focuses on practical system integration and evaluation under outdoor farm conditions rather than the introduction of a new detection or regression algorithm.

2 Related Work

Computer vision and machine learning have increasingly been applied to aquaculture monitoring [2]. Recent studies have used deep learning methods to recognize, detect, and analyze shrimps from images and videos [3].

Vision methods have been applied to shrimp detection and counting. Hu et al. [4] developed a density estimation method for counting shrimp larvae, while Lee et al. [5] investigated shrimp counting together with size and stomach fullness measurement. Asmak et al. [6] developed a mobile application for counting shrimp larvae using YOLOv5, showing that a shrimp detection model can be connected with a mobile interface. However, the application focused on counting and did not include shrimp length or weight estimation.

Shrimp size measurement has also been investigated using CNN and underwater imaging methods. Lai et al. [7] developed an automatic body length measurement method using an underwater imaging system. Waqar et al. [8] combined instance segmentation and centerline prediction for automatic shrimp detection and size estimation. These methods can represent shrimp body shape more directly, but they require underwater image acquisition or additional segmentation and centerline-processing stages.

Studies on shrimp weight estimation remain limited. Setiawan et al. [9] used morphometric features obtained from underwater images with machine learning to estimate shrimp weight. This study shows that visual measurements can support weight prediction, although the results depend on image quality and the accuracy of the extracted body features.

Compared with previous studies, the proposed work combines shrimp detection, reference calibration, length estimation, average weight prediction, and mobile result visualization in one workflow. The system uses standard smartphone images and a reference bracket rather than requiring a fixed underwater imaging system. This integration represents the main difference from earlier studies that focused mainly on detection, counting, size measurement, or weight prediction as separate tasks.

3 Proposed Approach

3.1 System Overview

The proposed smartphone shrimp monitoring system includes image acquisition, shrimp detection, length estimation, average weight prediction, and result visualization. A smartphone is used for image acquisition, while a reference bracket is placed in the scene for geometric calibration. The overall workflow is shown in Fig. 1.

After image acquisition, YOLOv8 detects the shrimps and generates a bounding box for each detected shrimp. The bounding boxes are used for length estimation and image-level feature extraction. The estimated mean length and average weight are then displayed through the Android application.

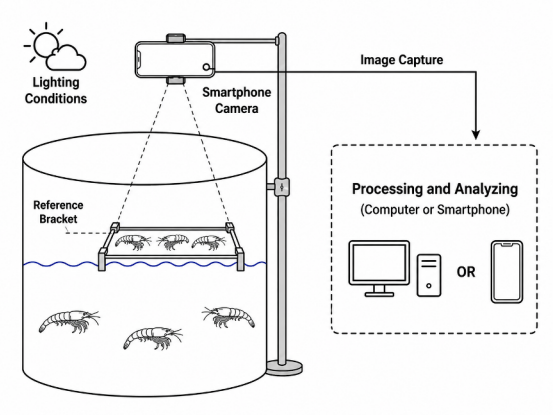


Fig. 1. Overview of the proposed smartphone shrimp monitoring system

3.2 Shrimp Detection Using YOLOv8

Shrimp detection is performed using a YOLOv8 object detection model because it provides a suitable balance between detection accuracy and computational efficiency. During image analysis, the model outputs the bounding-box coordinates, confidence score, and location of each detected shrimp. These outputs are used in the following length and weight estimation stages.

3.3 Length Estimation Using Reference Calibration

For the i-th detected shrimp, the longer side of its bounding box is used as the pixel representation of shrimp length. The pixel length and calibration ratio are calculated as

$$P_{s,i} = \max(w_i, h_i), \qquad r = \frac{L_{\text{ref}}}{P_{\text{ref}}}, \tag{1}$$

where w_i and h_i are the bounding-box width and height, L_{ref} is the known physical length of the reference bracket, and P_{ref} is its measured pixel length.

The estimated shrimp length and per-image mean length are calculated as

$$\hat{L}_{s,i} = P_{s,i} r, \qquad \bar{L} = \frac{1}{N} \sum_{i=1}^{N} \hat{L}_{s,i}, \tag{2}$$

where N is the number of detected shrimps. Because the method uses the longer bounding-box side rather than the curved body centerline, shrimp posture may affect the estimated length.

3.4 Average Weight Prediction Using SVR

Average shrimp weight is estimated from six image-level features extracted from the detected shrimp regions. First, the bounding-box width and height are converted into centimeters:

$$W_i = w_i r, \qquad H_i = h_i r, \qquad A_i = W_i H_i, \qquad R_i = \frac{W_i}{H_i}, \tag{3}$$

where A_i is the calibrated bounding-box area and R_i is the aspect ratio. The input feature vector is defined as

$$\mathbf{x} = [\mu_L, \sigma_L, \mu_A, \mu_R, \mu_C, N], \tag{4}$$

where μ_L and σ_L are the mean and standard deviation of estimated shrimp lengths, μ_A is the mean calibrated bounding-box area, μ_R is the mean aspect ratio, μ_C is the mean detection confidence, and N is the number of detected shrimps in the image.

The features are standardized using `StandardScaler` and processed by an SVR model with a radial basis function kernel. The prediction function is expressed as

$$\hat{W}(\mathbf{x}) = \sum_{j=1}^{m} (\alpha_j - \alpha_j^*) K(\mathbf{x}_j, \mathbf{x}) + b, \tag{5}$$

where K is the radial basis function kernel, $\mathbf{x}_j$ represents a support vector, α_j and α_j^* are the learned SVR coefficients, b is the bias term, and $\hat{W}$ is the predicted average shrimp weight. The model uses $C = 3.0$, $\epsilon = 0.05$, and $\gamma = \text{scale}$.

The method does not calculate shrimp mass through a direct volumetric formula. Instead, SVR learns the statistical relationship between the six image-level features and the manually measured average shrimp weight.

3.5 Mobile Application Integration

The trained detection and weight prediction components are integrated into an Android application. The application supports image capture or image upload, reference-bracket input, shrimp detection, estimated length calculation, average weight prediction, and result visualization. The prototype was deployed and tested on a Xiaomi 15T Pro 5G smartphone.

4 Experimental Setup

4.1 Datasets and Image Acquisition

The study used separate and non-overlapping datasets for detection training, detection testing, length evaluation, weight training, and weight testing. The YOLOv8 model was trained using 6,350 labeled shrimp images and tested using another 280 images. Length estimation was evaluated using 29 separate images.

For weight prediction, 354 images were used for model training, while another 26 previously unseen images were reserved only for testing. The weight-testing images were not used during model training or model selection.

The measurement images contained whiteleg shrimps (*Litopenaeus vannamei*) and were collected under outdoor shrimp-farm conditions in Sidoarjo and Pasuruan, East Java, Indonesia. The images included variations in lighting, water reflection, shrimp posture, turbidity, and background appearance. Examples of the images collected under these outdoor farm conditions are shown in Fig. 2.



Fig. 2. Examples of shrimp images collected under different outdoor farm conditions

4.2 Ground-Truth Measurement

Ground-truth shrimp length and weight values were obtained through manual measurement during data collection. A reference bracket with a known side length of 90 cm was positioned within the camera view to support pixel-to-centimeter conversion.

Each length image could contain several detected shrimps. Therefore, the estimated lengths were averaged at the image level and compared with the corresponding manually measured mean shrimp length. For weight prediction, the ground-truth target for each image was the manually measured average shrimp weight.

4.3 Training and Evaluation Protocol

The YOLOv8 model was trained using an input size of 640×640 pixels, a batch size of 8, and 100 epochs. Training was conducted using an RTX 2080 GPU with 8 GB memory. Detection performance was evaluated using precision, recall, mAP@0.5, and mAP@0.5:0.95 on the separate 280-image detection-testing set.

Length estimation was evaluated on the separate 29-image dataset using mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE). The estimated per-image mean length was compared with the manually measured mean length.

The `StandardScaler` and SVR model were fitted using only the 354 weight-training images. The SVR used a radial basis function (RBF) kernel with $C = 3.0$, $\epsilon = 0.05$, and $\gamma =$ scale. The fitted preprocessing and regression model were then applied to the 26 unseen weight-testing images, corresponding to an approximate training-to-testing ratio of 93:7. Weight performance was evaluated using MAE, RMSE, and MAPE.

The implementation used Python, YOLOv8, OpenCV, and scikit-learn in a Linux environment. The mobile workflow was tested using the Android prototype on a Xiaomi 15T Pro 5G smartphone.

5 Results and Discussion

5.1 Shrimp Detection Performance

The YOLOv8 model achieved a precision of 0.988, a recall of 0.927, an mAP@0.5 of 0.982, and an mAP@0.5:0.95 of 0.877 on the separate 280-image detection-testing set. These results answer RQ1 by showing that the model can accurately detect whiteleg shrimps under the tested outdoor farm conditions. Examples of the detection results are shown in Fig. 3.

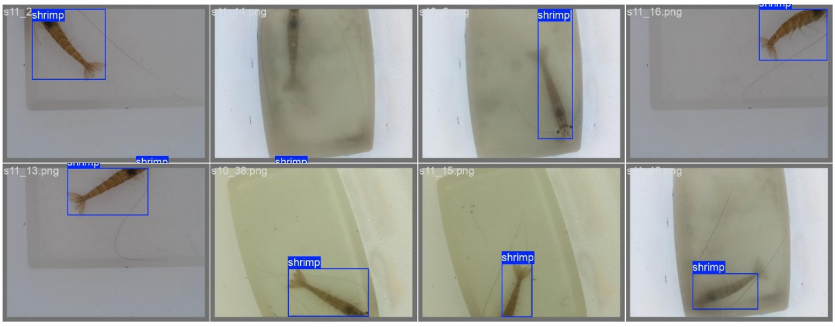


Fig. 3. Examples of whiteleg shrimp detection results produced by YOLOv8

Observed failure cases included overlapping shrimp bodies, partially visible shrimps, and shrimps with low visual contrast against the background. Water reflections and turbidity could also reduce the visibility of the shrimp body.

5.2 Length Estimation

Length estimation was evaluated using 29 separate images. Figure 4 shows the estimated mean shrimp length for each image in relation to the practical reference range of 8–10 cm.

The method achieved an MAE of 1.54 cm, an RMSE of 1.63 cm, and a MAPE of 14.6%. These results answer RQ2 by showing that reference calibration can provide practical mean-length estimates under the tested farm conditions.

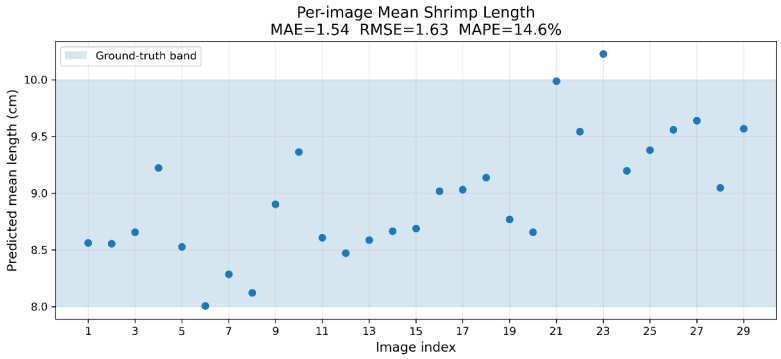


Fig. 4. Estimated mean shrimp length for each image and the practical reference range of 8–10 cm

A likely source of length error is that the longer side of the bounding box represents the straight image extent of the shrimp rather than its curved body centerline. Strong body curvature may therefore underestimate the actual length. Errors may also occur when the smartphone is held at an oblique angle or when the reference bracket and shrimps are positioned at different distances from the camera.

5.3 Average Weight Prediction

The SVR model achieved an MAE of 1.05 g, an RMSE of 1.78 g, and a MAPE of 7.5% on the separate 26-image weight-testing set. All 26 test images were previously unseen during model training. Figure 5 compares the predicted and manually measured average shrimp weights.

These results answer RQ3 by showing that the six image-level features can be used by SVR to estimate average shrimp weight on previously unseen images. However, the limited number of test images means that broader generalization still requires additional evaluation.

5.4 Discussion

Compared with the method of Waqar et al. [8], which reported an MAE of 1.02 cm and an RMSE of 1.27 cm, the proposed method produced higher length errors. Their segmentation and centerline approach can represent curved shrimp bodies more directly, while the proposed method uses a simpler bounding-box and reference-calibration procedure. Direct comparison should be interpreted carefully because the datasets, imaging conditions, and measurement procedures are different.

Camera-distance variation is partly handled by the reference-calibration ratio. However, measurement remains sensitive when the reference bracket and shrimps are not positioned on approximately the same plane. To reduce errors,

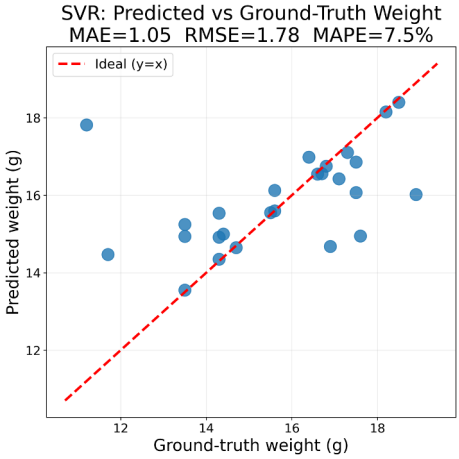


Fig. 5. Predicted and manually measured average shrimp weights on the 26-image test set

the smartphone should be held as parallel as possible to the measurement area, the reference bracket should remain fully visible, and strongly curved or heavily occluded shrimps should be avoided.

The small length evaluation and weight test sets are the primary limitation of this study. The 29 length images and 26 weight test images do not support generalization across shrimp sizes, farms, seasons, devices, and environmental conditions. Future studies should include larger independent test datasets and additional outdoor farms. Segmentation, keypoint detection, or centerline estimation may also improve the measurement of curved shrimp bodies.

The Android prototype confirmed the functional integration of image input, shrimp detection, length estimation, weight prediction, and result display on a smartphone. The current study focuses on technical performance, while formal usability evaluation with farm users remains future work.

6 Conclusion

This paper presented a smartphone system for whiteleg shrimp detection, length estimation, and average weight prediction under outdoor aquaculture conditions. YOLOv8 was used for shrimp detection, reference calibration was used for length estimation, and six image-level features were processed by SVR to estimate average shrimp weight.

For RQ1, the detection model achieved a precision of 0.988, a recall of 0.927, an mAP@0.5 of 0.982, and an mAP@0.5:0.95 of 0.877 on 280 separate detection-testing images. For RQ2, the reference-calibration method achieved an MAE of 1.54 cm, an RMSE of 1.63 cm, and a MAPE of 14.6% on 29 separate length-evaluation images. For RQ3, the SVR model trained using 354 weight images

achieved an MAE of 1.05 g, an RMSE of 1.78 g, and a MAPE of 7.5% on 26 separate and previously unseen weight-testing images.

The results demonstrate the feasibility of integrating shrimp detection, reference calibration, length estimation, average weight prediction, and mobile visualization into one smartphone workflow. However, the small measurement datasets, shrimp posture variation, occlusion, camera angle, and differences in the calibration plane limit broader generalization. Future work will expand the length and weight datasets, evaluate additional farms and smartphone devices, improve curved-body measurement, and conduct formal usability evaluation with farm users.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Flegel, T.W.: A future vision for disease control in shrimp aquaculture. J. World Aquacult. Soc. 50, 249–266 (2019)
2. Gladju, J., Kamalam, B.S., Kanagaraj, A.: Applications of data mining and machine learning framework in aquaculture and fisheries: A review. Smart Agricultural Technology 2, Article 100061 (2022)
3. Hu, W.-C., Wu, H.-T., Zhang, Y.-F., Zhang, S.-H., Lo, C.-H.: Shrimp recognition using ShrimpNet based on convolutional neural network. Journal of Ambient Intelligence and Humanized Computing (2020). https://doi.org/10.1007/s12652-020-01727-3
4. Hu, W.-C., Chen, L.-B., Hsieh, M.-H., Ting, Y.-K.: A deep-learning-based fast counting methodology using density estimation for counting shrimp larvae. IEEE Sensors Journal 23(1), 527–535 (2023)
5. Lee, Y.-K., Lin, B.-Y., Weng, T.-H., Huang, C.-K., Liu, C., Liu, C.-C., et al.: Counting and measuring the size and stomach fullness levels for an intelligent shrimp farming system. Connection Science 35(1), Article 2268878 (2023)
6. Asmak, S.Y., Rizaldi, D.D., Saputra, R.A.F., Abseno, A.P., Hananto, V.R., Oktarina, E.S.: A mobile app for counting shrimp larvae based on the YOLO V5 method. J. Comput. Electron. Telecommun. 5(2) (2024)
7. Lai, P.-C., Lin, H.-Y., Lin, J.-Y., Hsu, H.-C., Chu, Y.-N., Liou, C.-H., et al.: Automatic measuring shrimp body length using CNN and an underwater imaging system. Biosystems Engineering 221, 224–235 (2022)
8. Waqar, M.M., Ali, H., Zhou, H., Mohamed, H.G., Kim, S.C., Strzelecki, M.: A dual-segmentation framework for the automatic detection and size estimation of shrimp. Sensors 25(18), Article 5830 (2025). https://doi.org/10.3390/s25185830
9. Setiawan, A., Hadiyanto, H., Widodo, C.E.: Shrimp body weight estimation in aquaculture ponds using morphometric features based on underwater image analysis and machine learning approach. International Journal of Robotics and Automation 36(6), Article 360611 (2022)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

