Proceedings of the 10th International Conference on Fisheries and Aquaculture, Vol. 10, Issue 1, 2023, pp. 64-75

ISSN 2386-1282 online
DOI: https://doi.org/10.17501/23861282.2023.10106

YOLO V8 FOR ESTIMATION OF SHRIMP BODY WEIGHT FROM IMAGES

Farid I*, Lukman H, Syauqy NA and Liris M

JALA TECH Pte Ltd, Indonesia

Abstract: Penaeus vannamei, a highly cultured species, accounted for 51.7% of the total shrimp production, reaching 5.8 million tonnes globally in 2020 (FAO, 2022). Despite its substantial production, the shrimp farming industry faces various challenges, including shrimp growth monitoring, which is a critical aspect of production. Monitoring shrimp growth not only determines the rate of growth but also affects feeding efficiency. Currently, shrimp growth rate assessment relies on traditional methods that calculate average values from sampled data, introducing potential biases and necessitating time-consuming processes, such as the drying of shrimps before scaling takes place. To address these limitations, this research proposes a novel approach integrating image processing and machine learning algorithms to estimate shrimp weight. Specifically, we combine the YOLO V8 detection algorithm with logarithmic regression. YOLO V8 detects shrimp and measures their height in images, and then, utilizing the detected shrimp objects, we predict their weight through logarithmic regression. Our proposed algorithm achieves a Mean Average Error (MAE) of 0.6 grams in shrimp weight estimation, providing a more efficient and accurate alternative for shrimp growth monitoring in the industry.

Keywords: penaeus vannamei, image processing, machine learning algorithms, shrimp weight

Introduction

In shrimp farming, continual monitoring of the average weight and size distribution of shrimp in a pond is essential to optimize harvest strategies (Yu and Leung, 2005). Current best practice, both domestic and international, involves casting a net (Xi et al., 2023) that catches a sample of up to 100 shrimps called sampling activity. Captured shrimps are bulk weighed and individually counted to estimate the average weight, and that is a time-intensive task.

A limited understanding of shrimp growth can result in decreased profits (Xi et al., 2023). Sampling can provide important information for evaluating growth rates, and size distribution, which is insight for productivity, conditions of the pond and potential harvest. This information can help the farm manager predict and avoid unwanted situations. Shrimp farm technicians do sampling once a week, we aim to take advantage of this practice, with minimum disturbance of their daily operation/workflow.

We aim to provide faster insight into how the shrimps are growing over time, the project aims to develop an app and computer vision-based system to automatically measure shrimp size from images acquired when farmers are doing sampling. More precisely, we aim to develop (i) a smartphone app to collect field data using camera (image of sampled shrimp), (ii) develop a set of computer vision and machine learning-based methods to estimate shrimp weight based on those field images, and (iii) conduct an analysis of how accurately the measured shrimp sizes reveal pond status (e.g. size variation) based on field quality data.

*Corresponding Author's Email: *farid@jala.tech*

Related Works

In the rapidly evolving field of aquaculture technology, several noteworthy advancements and tools have surfaced to aid farmers and researchers. One of the pioneering companies in this domain is XpertSea, which has made significant strides in harnessing technology to address longstanding challenges in aquaculture. A standout innovation from XpertSea is XpertCount, a cutting-edge device leveraging computer vision to efficiently and accurately count and size aquatic organisms, predominantly shrimp larvae. This tool aims to replace manual counting methods that often prove time-consuming and susceptible to human error. By providing hatcheries and aquatic farms with precise data on their stock, XpertCount paves the way for more informed decisions regarding stocking density, feeding, and overall health management. Such advancements underscore the potential of technology to revolutionize traditional farming practices, optimizing both operational efficiency and productivity (Masson et al., 2013).

In early 2023, YOLOv8 was released. YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions and introduces new features and improvements to further boost performance and flexibility. YOLOv8 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks (Jocher et al., 2023). We would like to test same functionality upon this model.

Materials and Methods

Shrimp Dataset Acquisition System

This section describes how the dataset is collected and labeled.

Designated Box

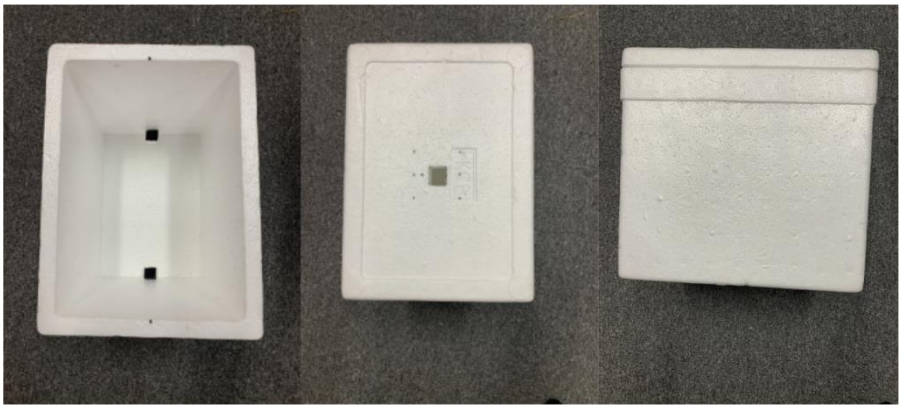

Figure 1: Styrofoam box with references points

First of all, we created a tool for dataset collection, which is a modified ice box with two reference points. These reference points are 27.5cm apart and are used to estimate shrimp width. Then, we modify the box lid by making a hole for the phone's camera to capture images. With this simple tool, farmers can easily build their own box.

Shrimp Dataset Collection

Figure 2: Shrimp sample photo

After the designated box with two reference points was created, we started to collect images to create the dataset. The image collection process involved farmers/technicians taking some shrimps from a pond, putting them on the designated box, and then we took photos of those shrimps using a phone camera. The dataset was collected from mid-February to mid-March 2023 at one of the shrimp farms in Purworejo, Jawa Tengah, Indonesia. Those field data was collected from four ponds for a period of one month, once a week. In total, 423 images of shrimp were collected. We also hand-measured the weight and length of each shrimp. From these measurements, we obtained 157 data points. These will be used to estimate weight based on the width measurements of the shrimp we obtained.

Shrimp Dataset Labeling

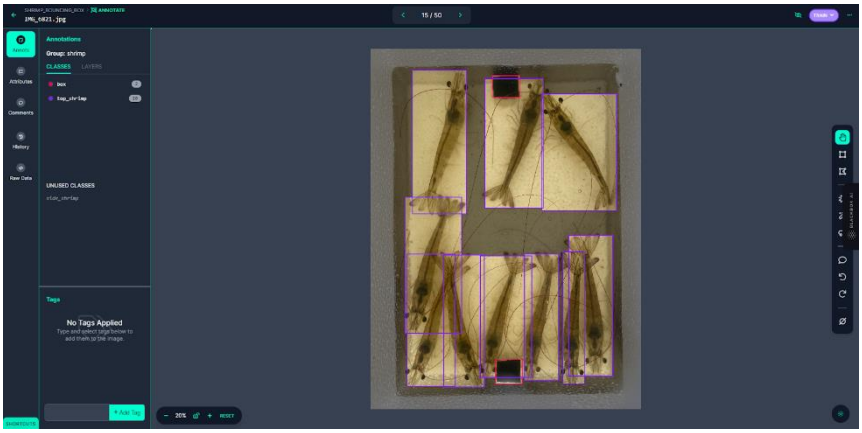

Figure 3: Image annotation using Roboflow

After we got the shrimp dataset, we labeled all those images in the dataset. Semantic segmentation, while providing a detailed analysis by labeling each pixel in the input image, is time-consuming for annotation (Hafiz and Bhat, 2020). Given the limited duration of our research, we opted for bounding box annotation, which is faster than segmentation (Dai et al., 2015). In Figure 3, we marked three labels: "box" (referring to the reference points), "side shrimp" (indicating a side view of the shrimp), and "top shrimp" (indicating a top view of the shrimp). We distinguished between side and top views of the shrimp to calculate the estimated width in different ways.

Shrimp Dataset Preprocessing

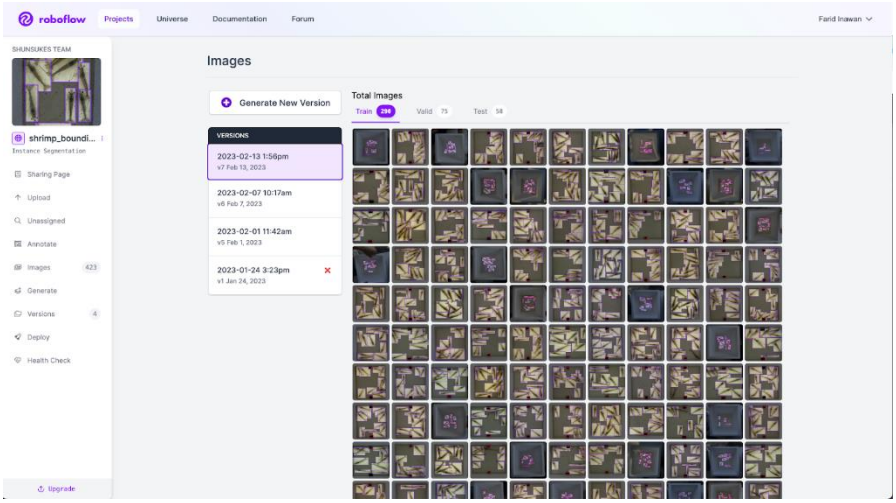

Figure 4: Interface of Roboflow for Training, Validation and Test:

After labeling the dataset, it was prepared for next processing. We split the dataset into training, validation, and testing dataset, subsequently building a model using YOLOv8, the latest version of YOLO. Currently, we are using the Roboflow platform to accomplish these tasks.

Shrimp Model

This section describes the creation of the shrimp YOLOv8 model and evaluates its performance in detecting shrimps.

YOLOv8

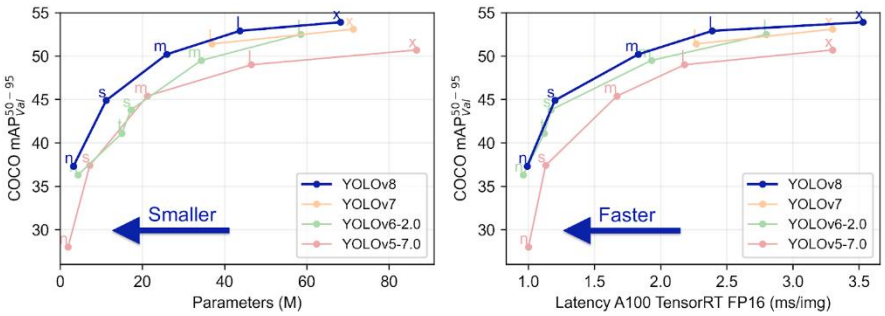

Figure 5: YOLOv8 comparison with predecessors (Jocher et al., 2023)

YOLOv8 is the latest version of the YOLO model which was released on January 1st 2023. As we can see, In Figure 5, YOLOv8 has been shown to surpass both YOLOv5 and YOLOv6 in terms of COCO mAP (mean Average Precision) when trained on 640 image resolution, and this is achieved with a similar parameter count. This points to the model's efficiency and architecture improvements made for hardware. While both YOLOv8 and YOLOv5 are products of Ultralytics, the latter has been noted for its remarkable real-time performance. Based on preliminary benchmarks from Ultralytics, it is anticipated that YOLOv8 will prioritize deployments on edge devices that demand high-speed inference (Hussain, 2023).

Model Configuration

The model we use is the YOLOv8 "s" version, which is small in size, yet fast. Its efficiency meets our criteria to deploy the detection model to the cloud service, and to make sure the cost is minimum with maximum scalability. Our model was trained over 25 epochs. An epoch represents a single iteration through the entire training dataset. The choice of 25 epochs was made based on a balance between computational efficiency and ensuring model convergence. Training a model for too few epochs might lead to underfitting, where the model may not learn the intricate patterns in the dataset. On the other hand, training for an excessive number of epochs can lead to overfitting, where the model becomes too tailored to the training data and performs poorly on unseen data. After careful observation of the training progress and validation metrics, 25 epochs were determined to be optimal for our YOLOv8 model. This allowed the model to learn patterns adequately while minimizing the risk of overfitting and ensuring efficient use of computational resources.

Model Performance

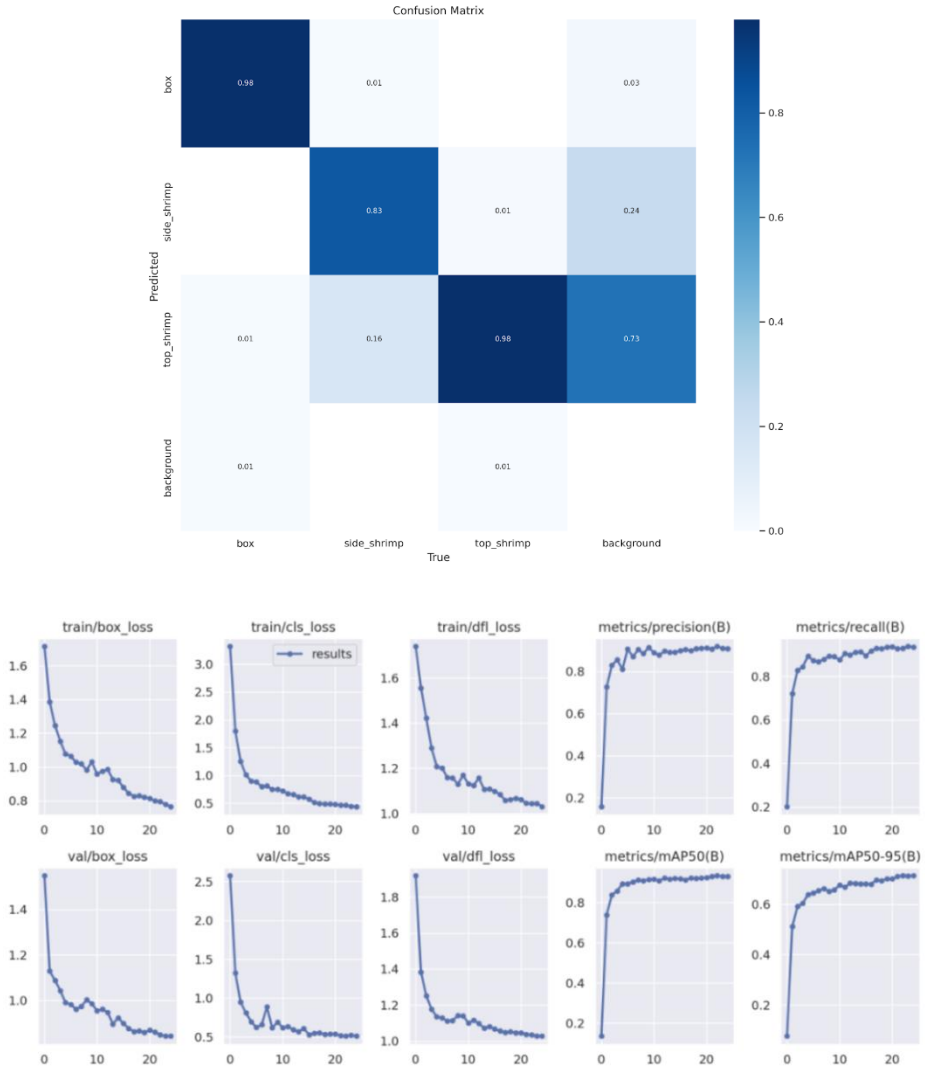

Figure 6: Confusion Matrix and Metric Scores of Trained YOLOv8 Shrimp Model

Figure 6 above are benchmarking results using YOLOv8 the results from the trained model are promising. The dark colors along the diagonal of the confusion matrix indicate high true

positive values. The darker the shade, the better the performance. In the metric visualizations, the "loss" graphs consistently trend downward while the precision, recall, and mAP metrics trend upward. This indicates that the model is progressively improving in its performance during training.

```
val: Scanning /content/datasets/shrimp_bounding_box-7/valid/labels.cache... 75 images, 0 background
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 5/5
                   all         75        866      0.927      0.924      0.935      0.748
                   box         75        106      0.942      0.981      0.967      0.803
           side_shrimp         75         90      0.901      0.822       0.86      0.647
           top_shrimp          75        670      0.938      0.968      0.977      0.795
Speed: 3.0ms pre-process, 16.6ms inference, 0.0ms loss, 2.1ms post-process per image
```

Figure 7: Accuracy of Trained YOLOv8 Shrimp Model

In Figure 7, we can see accuracies of our trained model's metrics. P stands for Precision, R stands for Recall, mAP stands for mean Average Precision, mAR stands for mean Average Recall. With those precision and recall scores, which were higher than 90, that shows that our model performance was indicated good. Generally, precision evaluates the correctness of the model's predictions, signifying the fraction of accurate predictions. On the other hand, recall quantifies the proportion of all relevant outcomes that the model successfully identified. The precision and recall are calculated using the following equations:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

TP refers to True Negative, FP refers to False Positif, and FN refers to False Negative.

Shrimp Weight Logarithmic Regression Model

This section describes how the shrimp Logarithmic Regression Model model was created, and how is its performance.

Dataset Preparation

Previously, we gathered a dataset of 157 hand measured shrimp weights and lengths. This data was cleaned and transformed into a CSV format for processing with the sklearn library. Within this dataset, three parameters were documented: weight, width, and age. We will use this data to train a regression model that estimates a shrimp's weight based on its width, as determined earlier by the YOLOv8 model.

Model Creation

After preparing the dataset, we developed a mathematical model to predict the weight of the shrimp. We used linear regression, which is commonly used to determine the relationship between a dependent variable and one or more independent variables (R et al., 2022). In our study, the dependent variable is the weight, and the independent variables are height and age. The formula of linear regression is:

$$log(y) = \beta 0 + \beta_1 x_1 + \beta_2 x_2 \quad (3)$$

Where:
β_0 is the model intercept
β_1 is the "height" feature
β_2 is the "age" feature
x_1 is the height
x_2 is the age
y is the weight

This linear regression model takes 2 inputs, there are height and age To linearize the plot, we applied the logarithm of the weight, as the relationship between the dependent and independent variables is exponential. Subsequently, to obtain the weight prediction, we used an exponential function (e) to revert the value into its original scale. The formula is shown as follows:

$$y = e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2} \quad (4)$$

By reverting using this exponential function, we can get the original value of the shrimps.

Model Performance

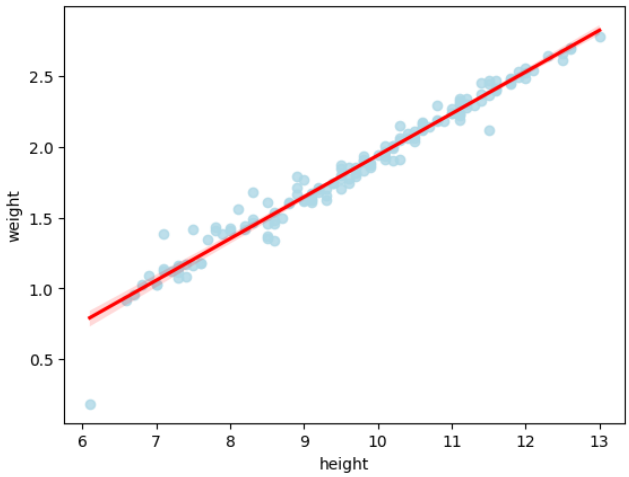

Figure 8: Linear Regression Plot with Logarithmic Scale in Weight

The performance of our linear regression model can be assessed visually and through its functional attributes as we can see in Figure 8. Upon plotting, the scatter points clustered closely around the regression line, suggesting a good fit between the predicted values and the actual data points. We used two predictor variables, height and age, to estimate the weight of the shrimps. However, it's worth noting that the weight measurements were first subjected to a logarithmic transformation, owing to their exponential nature when measured manually. This transformation ensured a more linear relationship between the predictors and the response variable. Overall, the close adherence of scatter points to the regression line and the logical inclusion of relevant

predictors indicate that our model has captured the underlying patterns of the data effectively. This means our model works well and understands the patterns in our data.

And we computed the Mean Absolute Error (MAE) for these results. MAE is commonly used to examine average model-performance error as it more naturally calculates the error (straightforwardly quantifies the discrepancy for individual values) (Willmott and Matsuura, 2005). Mean Absolute Error (MAE) offers a direct measure of prediction accuracy in our model. We calculated MAE with the following formula:

$$y = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (5)$$

Where:
n is the number of shrimp
y_i is the actual value of the shrimp

$\hat{y}_i$ is the predicted value of the shrimp

Model Implementation

This section describes how the YOLOv8 model and the Regression model works together to estimate shrimp's weight and its implementation using python.

Shrimp Weight Prediction System

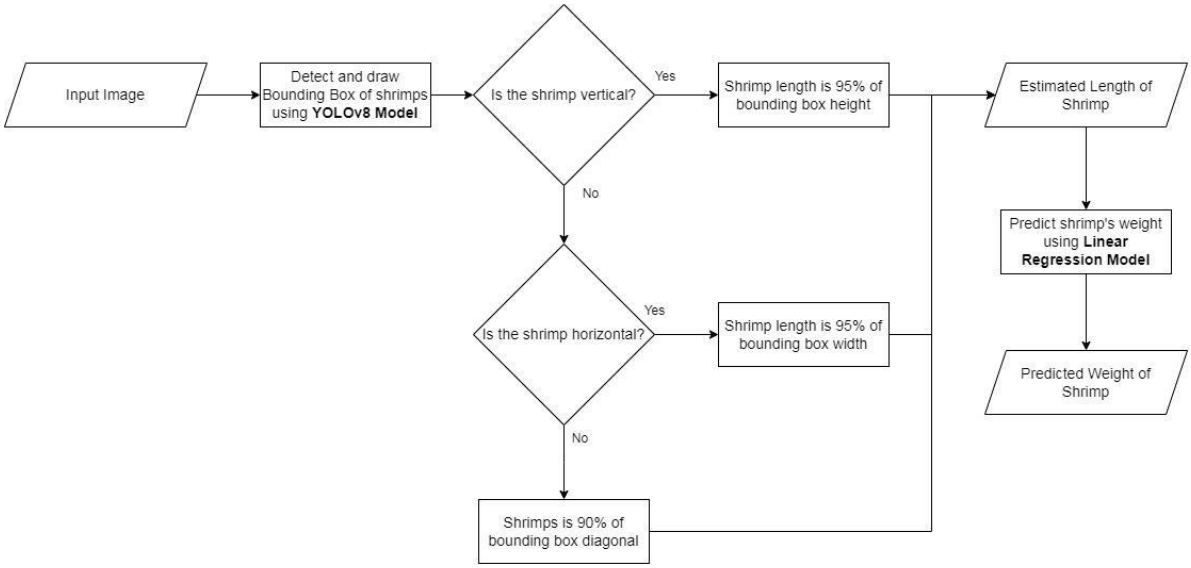

Figure 9: Shrimp weight prediction flow chart

Once the linear regression model is set up, we can proceed with predicting the weight of the shrimps using python. The flowchart in Figure 9 outlines the prediction system. Initially, the system takes an image as input, which is then processed by the YOLOv8 Model to identify shrimps and encircle them with bounding boxes. Following this, the shrimp can fall into one of three conditions, with each condition dictating a specific calculation. These three conditions are illustrated in Figure 10 below.

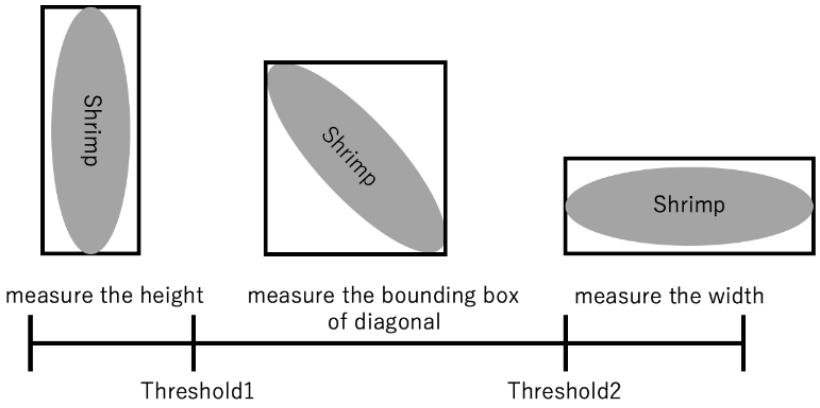

Figure 10: Three conditions of the shrimp

We set two thresholds. If the value of height/width is less than threshold1, the shrimp is categorized as vertical. If the value of height/width exceeds threshold2, it's labeled as horizontal. Values in between these thresholds classify the shrimp as diagonal. The actual length calculation varies based on the shrimp's orientation within its bounding box: for vertical or horizontal orientations, the length is set at 95% of its vertical or horizontal dimension, respectively. For diagonal orientations, the length is taken as 85% of its diagonal. These distinctions can be seen in Figure 11 below.

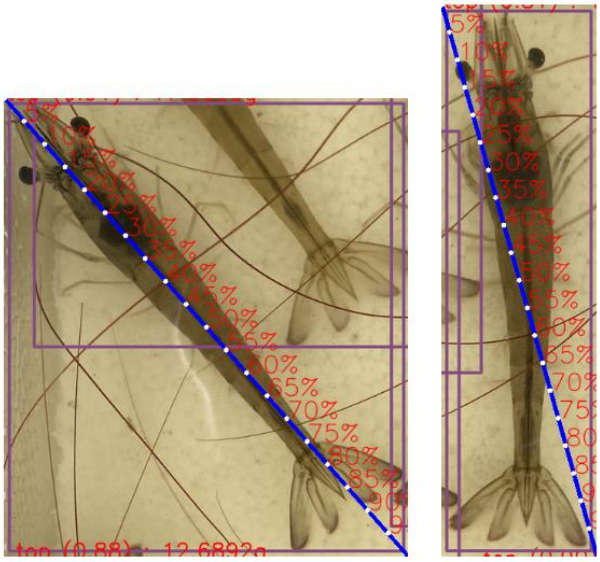

Figure 11: Diagonal and Vertical Shrimp

Once we obtain the estimated length of the shrimp, we can predict its weight using the previously created linear regression model. This model requires two inputs: the estimated length and the shrimp's age.

Serverless Implementation

After we finished testing all the systems, we implemented this system using the Serverless framework. We developed an application in serverless Python, hosting all models and resources in the cloud. This ensures that for future developments, we can simply invoke the API endpoint from this application to generate the average predicted weight of the shrimp.

Results and Discussion

System output

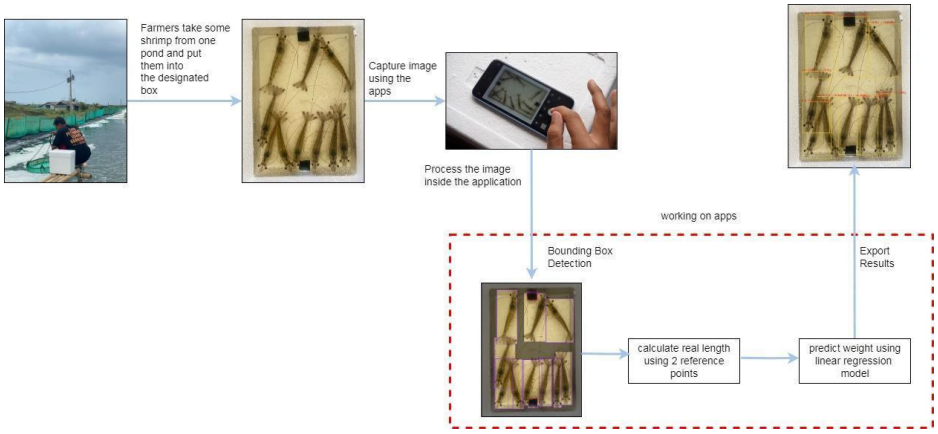

Figure 12: Process for measuring shrimp weight using Image Processing and Machine Learning pipeline.

The process of estimating weight and length of shrimps is presented in Figure 12. The process started by farmers/technicians taking some shrimps and putting them on the designated box and capturing images of shrimps within the box using an app on smartphones. The images are then captured and processed on smartphones, where shrimps are identified using models based on YOLOv8. This produces bounding boxes around each shrimp, which are then used to estimate their length and weight.

The output of our application comprises two elements: the average of all the predicted weights and the input image annotated with bounding boxes alongside their corresponding predicted weights. These outputs can be viewed in Figure 13 and Figure 14 below.


```
{
    "statusCode": 200,
    "body": "{\"data\": 14.60946831929285}"
}
```


Figure 13: Output 1 - Average predicted weight

Figure 14: Output 2 - Annotated image

We display the average of the weights in our output because shrimp farmers usually sample 10 shrimps each pond. They then derive insights from this sample based on the average weight. For farmers, it's more convenient to know a single averaged weight than the individual weight of each shrimp.

Result Comparison

We compared the weight predictions from our application to the actual hand-measured weights of the shrimps taken during sampling in week 4 (batch 4) from ponds B1 and B3.

Table 1: Actual and Predicted Weight comparison

Batch/Pond	Actual Weight	Predicted Weight
Batch 4 / B1	13.34	14.51
Batch 4 / B3	11.75	11.45

Using the specified formula, we obtained a MAE of 0.68 for Batch 4/B3 and 1.01 for Batch 4/B1. From those results, the MAE for Batch 4/B3 is 0.68, which suggests a relatively close prediction to the actual shrimp weights for this batch and pond. On the other hand, the MAE of 1.01 for Batch 4/B1 indicates slightly less accuracy for this particular batch and pond. Although both values show our model's predictions are generally in the ballpark of the actual measurements.

Conclusion

Using this approach, which involves utilizing bounding box dataset labeling, training with YOLOv8s, and calculating distances using two reference points to determine the real scale of the bounding box and predict the weight using linear regression model, there appears to be potential for refinement, especially from the prediction for Batch 4/B1 above which still shows a noticeable difference. It is important to optimize our models by collecting more data so that our dataset can improve accuracy. Notably, despite these areas for improvement, the model's prediction errors remain low, highlighting its effectiveness and dependability. Looking ahead, future research will benefit from an expanded dataset. Exploring alternative labeling methods, such as shrimp segmentation or oriented bounding box rather than usual bounding boxes (Axis-Aligned Bounding Box), could offer more detailed insights. Furthermore, acquiring more hand-measured data on shrimp length and weight would contribute to refining our linear regression model's precision. Overall, our current research provides a valuable asset for shrimp farmers, seamlessly merging accuracy with utility, and setting the stage for more data-centric methodologies in shrimp agriculture.

Acknowledgements

The authors would like to thank Naufal Halim, Abdillah Akmal Firdaus, Nina Grizka Deslanya, Reno Marcello Mohammad, Andriansyah Rafi Rahmadhian, Tomine Taiki, and Shunsuke Tamiya for contributing in this project. We also appreciate the Department of Electrical

Engineering and Information of Gadjah Mada University, as well as the Department of Computer Science and Engineering of Toyohashi University, for their support in this project.

References

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research, 30, 79-82. doi:10.3354/cr030079

Dai, J., He, K., & Sun, J. (2015). BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation. Proceedings of the IEEE International Conference on Computer Vision (ICCV). doi:10.1109/ICCV.2015.191

Yu, R., & Leung, P. (2005). Optimal harvesting strategies for a multi-cycle and multi-pond shrimp operation: A practical network model. Mathematics and Computers in Simulation, 68(4), 339-354. doi:https://doi.org/10.1016/j.matcom.2005.01.018

Xi, M., Rahman, A., Nguyen, C., Arnold, S., & McCulloch, A. (2023). Smart headset, computer vision and machine learning for efficient prawn farm management. Aquacultural Engineering, 102, 102339. doi:https://doi.org/10.1016/j.aquaeng.2023.102339

Arumuganathan, T., Shruthi, R., & Raghavendar, S. (2022). Soil NPK Prediction Using Multiple Linear Regression. 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), (pp. 542-546). doi:10.1109/ICACCS54159.2022.9785338

FAO. (2022). The State of World Fisheries and Aquaculture. FAO, Rome.

Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: state of the art. International Journal of Multimedia Information Retrieval, 9. doi:https://doi.org/10.1007/s13735-020-00195-x

Masson, S., Lavigne, S., Robitaille, V., & Andrews, C. (2013). The XperCount, a fast and cost-effective method for the enumeration of organisms in environmental media. Journal of Xenobiotics, 3, 26-28. doi:doi:10.4081/xeno.2013.s1.e10

Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. Machines, 11(7). doi:10.3390/machines11070677

Jocher, G., Chaurasia, A., & Qiu, J. (2023, January 1). YOLO by Ultralytics. Retrieved 09 22, 2023, from Github: https://github.com/ultralytics/ultralytics