POSTLARVAE SHRIMP COUNTING VIA AUTOMATIC ANNOTATION CONVERSION IN COMPUTER VISION

CHUYỂN ĐỔI NHÃN TỰ ĐỘNG TRONG THỊ GIÁC MÁY TÍNH CHO BÀI TOÁN ĐẾM TÔM GIỐNG

Can Thi Phuong[1]*, Pham Thi Kim Ngoan[1], Bui Thi Hong Minh[1], Mai Duc Thao[1], Than Van Hoan[2], Pham Quang Thuan[3]

[1]Nha Trang University, Vietnam
[2]Khanh Hoa Center for Infomation and Apllication of Science and Technology, Vietnam
[3]Nha Trang National College of Pedagogy, Vietnam

*Corresponding author: phuongct@ntu.edu.vn

(Received: April 27, 2025; Revised: May 26, 2025; Accepted: May 30, 2025)

DOI: 10.31130/ud-jst.2025.23(7).248

Abstract - In the field of computer vision, tasks such as object detection, instance segmentation, and classification typically require training data annotated in different formats. Manual annotation is often time-consuming and labor-intensive, especially when the same image dataset needs to be annotated in multiple ways to support different requirements. This challenge becomes more pronounced in the context of postlarvae shrimp counting, where the objects are small, densely packed, and frequently overlapping. This study proposes an automated method for converting polygon-based annotations into bounding box annotations, aiming to optimize the data processing workflow and improve model performance. The experimental results show that models trained on the automatically converted dataset achieved a MAPE of 3.26% in the postlarvae shrimp counting task, demonstrating the effectiveness of the proposed method in addressing the challenge of label format conversion.

Key words - Postlarvae shrimp counting; polygon annotation; bounding box annotation; segmentation; detection; YOLO

Tóm tắt - Trong lĩnh vực thị giác máy tính, các bài toán như phát hiện vật thể, phân đoạn đối tượng và phân lớp thường yêu cầu dữ liệu huấn luyện được gán nhãn theo các định dạng khác nhau. Việc gán nhãn thủ công thường tốn nhiều thời gian và công sức, đặc biệt khi cùng một tập dữ liệu ảnh cần được gán nhãn theo nhiều cách khác nhau để phục vụ cho các yêu cầu khác nhau. Thách thức này càng trở nên rõ rệt trong bài toán đếm tôm giống, nơi các đối tượng có kích thước nhỏ, mật độ cao và thường chồng lấn lên nhau. Nghiên cứu này đề xuất một phương pháp tự động chuyển đổi nhãn từ dạng đa giác sang dạng hộp giới hạn, nhằm tối ưu hóa quy trình xử lý dữ liệu và cải thiện hiệu suất mô hình. Kết quả thực nghiệm cho thấy mô hình huấn luyện trên tập dữ liệu chuyển đổi tự động đạt tỷ lệ lỗi MAPE 3,26% trong bài toán đếm tôm giống, chứng minh tính hiệu quả của phương pháp đề xuất trong việc xử lý bài toán chuyển đổi định dạng nhãn.

Từ khóa - Đếm tôm giống; nhãn đa giác; nhãn hộp giới hạn; phân vùng đối tượng; phát hiện đối tượng; YOLO

1. Introduction

The shrimp industry in Vietnam currently plays a pivotal role in seafood exports. By 2025, it is estimated that approximately 140–150 billion postlarvae shrimp will be required to meet production demands. To achieve the export target of USD 10 billion and enhance the competitiveness of the shrimp sector in the international market, automation of production processes has become essential to reduce costs. Among these processes, postlarvae shrimp counting is crucial for seed release, growth monitoring, feed optimization, assessing uniformity, and improving shrimp breeding practices. The application of artificial intelligence to automate postlarvae shrimp counting has become an urgent need.

Postlarvae shrimp are small in size and tend to cluster, leading to frequent occlusion, which makes annotation both time-consuming and labor-intensive. To address the dual tasks of counting and measuring postlarvae shrimp while reducing annotation time, our previous study [1] employed a three-point polygon annotation method. This dataset comprises images with an average density of 235.82 postlarvae shrimp per image, with an average occlusion coverage of 7.42%. In this method, annotation is performed from the end of the shrimp's tail. Polygon labels accurately match the shape and boundaries of the objects, effectively eliminating unnecessary background details. Our proposed model, which combines UNet and CBAM, achieved a MAPE of 3.83% on this dataset during testing.

For the postlarvae shrimp counting task, study [2] prioritized the use of bounding box annotation for the shrimp's head. These studies indicate that in images of postlarvae shrimp, particularly where occlusion or proximity between individuals occurs, the body and tail regions are often obscured and difficult to distinguish. In contrast, the head region is typically more visible with distinctive features such as antennae, eyes, and a pointed head shape, thus facilitating model learning and individual discrimination.

Annotating the entire postlarvae shrimp body with bounding boxes often leads to overlap between boxes, making it challenging for the model to distinguish individual shrimp. Furthermore, full-body bounding boxes frequently include irrelevant background regions, introducing noise into the learning process. A significant advantage of head-only bounding box annotation is its simplicity and time efficiency compared to polygon annotation of the entire body. However, it should be noted that this method requires additional processing if the application is to be extended to postlarvae shrimp length measurement.

Another approach, adopted in studies [3], [4], involves annotating the entire postlarvae shrimp body, which is particularly effective for datasets with relatively low density (fewer than 200 individuals per image). This method is commonly applied when deploying YOLO (You Only Look Once) models and their variants for object detection tasks.

Full-body annotation demonstrates superior performance in object recognition from low-resolution images or in cases where postlarvae shrimp are fully visible and not occluded by other individuals. However, a notable limitation of this method is that bounding boxes often include substantial irrelevant background, which can lead to information noise and reduce overall model performance.

The data-centric AI approach, proposed by Andrew Ng at the DATA and AI Summit 2022, emphasizes that improving data quality while keeping the model unchanged yields greater effectiveness in performance enhancement.

Numerous studies have addressed annotation issues. Research has evaluated annotation quality among companies [5] to optimize annotation budgets. According to [6], authors improved the label quality of public datasets to enhance AI model performance by improving data quality rather than model architecture. Other studies have demonstrated the impact of annotation on machine learning model performance [7].

Despite extensive research on annotation methods in object recognition tasks, no study has demonstrated the effectiveness of automatic conversion from polygon to bounding box annotation in improving model performance compared to manual bounding box annotation. Moreover, there has been no comprehensive comparative study of the effectiveness of different annotation types in specific applications such as postlarvae shrimp counting.

Our study contributes the following novel aspects:

1. Proposed annotation conversion algorithm: We developed an automated method for converting polygon-annotated data into bounding boxes for both the head and full body of whiteleg postlarvae shrimp. This algorithm not only optimizes the construction of diverse datasets but also significantly reduces annotation cost and time.

2. In-depth performance analysis: We conducted a comprehensive evaluation and comparison of advanced models such as YOLOv11 and YOLOv12 on both manually annotated bounding box datasets and datasets automatically converted from polygons to bounding boxes, providing an objective perspective on the strengths of each method.

3. Analysis of YOLOv11 and YOLOv12 suitability: We analyzed the appropriateness of YOLOv11 and YOLOv12 models for postlarvae shrimp counting tasks in both object detection and instance segmentation scenarios.

The structure of this paper is as follows: Section 2 presents a review of related work; Section 3 details the research results; Section 4 discusses the advantages and limitations of the proposed method; and the final section summarizes the main contributions and outlines future research directions.

2. Related work

Among image annotation methods for deep learning tasks, two common types - polygon and bounding box - differ in annotation accuracy and cost. Polygon annotation offers the highest accuracy, as annotators must specify multiple points around the object to precisely describe its shape. However, this makes polygon annotation the most time-consuming and costly method. In contrast, bounding boxes are rectangular regions enclosing the object, requiring only two mouse clicks to create, thus saving time. Nevertheless, bounding boxes often include many pixels that do not belong to the actual object, increasing the false positive rate. Therefore, selecting an appropriate annotation method must balance the desired accuracy and implementation cost [7].

Different deep learning tasks require different annotation formats. Datasets related to object recognition may focus on classification, detection, segmentation, or instance segmentation. Well-known datasets such as MS COCO (Microsoft Common Objects in Context) [8], LVIS (Large Vocabulary Instance Segmentation) [9], and Cityscapes [10] offer various annotation types for object recognition, instance segmentation, and other computer vision tasks. Notably, these datasets primarily store polygon annotations and automatically generate bounding boxes. According to our research, this is achieved by finding the smallest bounding box, aligned with the coordinate axes, that fully covers the object.

Studies on postlarvae shrimp counting have been conducted on datasets annotated with both full-body and head-only bounding boxes. However, each study has focused on a single annotation type, and there has been no dataset annotated with polygons. The research group in [11] used a dataset annotated with full-body bounding boxes, achieving a counting error rate of 5.7% with the YOLOv5m6 model for images with fewer than 200 postlarvae shrimp. According to [3], using YOLOv3, a postlarvae shrimp density below 60 per image resulted in a counting accuracy of 76.48% with full-body bounding box annotation. Several studies have also used full-body bounding box annotation [3], [4]. Another research direction has utilized head-only bounding box annotation, arguing that this approach reduces the influence of background compared to full-body boxes, thereby improving counting performance [12]. These studies have focused solely on either full-body or head-only bounding boxes and have not compared the effectiveness of both annotation types on the same dataset.

YOLOv11 [13], introduced by Ultralytics in 2024, possesses several features suitable for evaluating datasets automatically generated from polygons. With the C3k2 block, the model significantly reduces parameter count and optimizes computational resources, making large-scale automatic data processing more efficient. C3k2 is based on the Cross Stage Partial Network (CSP); according to Wang et al. [14], this architecture preserves gradient diversity by combining feature maps from the beginning and end of each network stage, a particularly important approach when evaluating synthetic data that may lack natural diversity. The C2PSA block enhances spatial attention, enabling the model to focus on important regions in the image, especially when identifying small objects such as postlarvae shrimp. This attention mechanism can help detect subtle discrepancies between real and synthetic data, as demonstrated in [1] when combining CBAM with U-Net for shrimp segmentation and counting.

YOLOv12 [15] emphasizes attention mechanisms. This model integrates Area Attention and FlashAttention, leveraging the benefits of attention while maintaining the fast processing speed of CNNs. Area Attention partitions the feature map into evenly sized, non-overlapping segments along horizontal or vertical axes, avoiding complex window partitioning methods such as Shifted Window [16], Criss-Cross Attention [17], or Axial Attention [18]. This feature is particularly valuable when evaluating data generated from polygons, as it allows the model to effectively process object boundaries and shapes - factors that are often distorted during data synthesis.

3. Research results

3.1. Experimental environment, dataset, and performance metrics

The original input image size of 2592×3872 pixels was standardized to 640×640 pixels. Training was conducted for 500 epochs with a learning rate of 10^{-2}, a batch size of 64, and optimization using the SGD algorithm. The IoU threshold was set at 0.7. The experiments were performed on a server equipped with an NVIDIA® GeForce RTX™ 4090 GPU and 64GB RAM.

Image data were collected from the hatchery of Nha Trang Marine Research Station Co., Ltd., featuring postlarvae shrimp at stages PL28 (Postlarvae 28 – 28 days old) and PL25 (Postlarvae 25 days old). We captured images of shrimp of varying sizes using a specialized camera. Shrimp were randomly selected and placed in a white styrofoam box with variable water levels. Images were taken under different lighting conditions beneath the hatchery's netting, from a top-down angle, and at distances of 100–150 cm. These factors were considered to diversify the sample set, thereby enhancing the accuracy and generalizability of the model.

We utilized 103 images from the NTUShrimp dataset, previously polygon-annotated in [1]. This dataset contains over 20,000 postlarvae shrimp labels, averaging 235.82 shrimp per image. The dataset size meets quality standards as it exceeds the 10,000 labels per object benchmark published by Ultralytics [19]. The data were split into 80% for training (82 images), 10% for validation (11 images), and 10% for testing (10 images). The test images were captured on a different day, featuring PL25 shrimp samples to ensure differentiation from the training images (PL28).

Based on this dataset, we propose methods to generate datasets with full-body bounding boxes (BodyBboxShrimp) and head bounding boxes (HeadBboxShrimp) for postlarvae shrimp. These datasets were then used to train and test YOLOv11 and YOLOv12 models for both object detection and segmentation tasks.

To evaluate the effectiveness of automated label generation from polygon annotations, we also constructed two manually annotated datasets for benchmarking purposes: the manually labeled full-body bounding box dataset (BodyBboxShrimpManual) and the manually labeled head bounding box dataset (HeadBboxShrimpManual).

After training, validation, and testing, model performance was assessed using mAP50 and MAPE metrics:

- **MAPE** is the mean absolute percentage error between the predicted value $\hat{y}$ and the actual value (y), calculated as follows:

$$MAPE = 100\% * \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right|, \quad (1)$$

where, n is the number of samples.

- mAP is the mean of average precision (AP) across all classes. AP evaluates overall model performance by considering the balance between precision and recall, calculated as:

$$AP = \int_0^1 Precision(Recall)d(Recall). \quad (2)$$

- IoU (Intersection over Union) is the ratio of the intersection to the union of the predicted region (P) and the ground truth region (G), computed as:

$$IoU(P,G) = \frac{P \cap G}{P \cup G} \quad (3)$$

3.2. Proposed Automatic Label Generation Method

In the case of postlarvae shrimp, the objects are small and transparent, making them susceptible to background interference. In [1], we proposed an improved polygon annotation method (Figure 1).

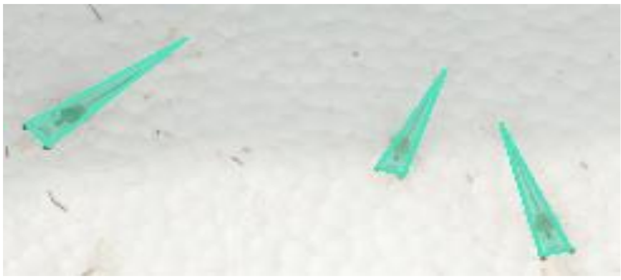

Figure 1. *Polygon Annotation Style*

Each shrimp label starts at two points (the eyes) and extends to the tail tip. While this method is time-consuming, it is more effective than bounding box annotation when shrimp are densely clustered or overlapping. This approach minimizes background influence and reduces annotation time compared to multi-point polygon labeling.

The methodology (Figure 2) consists of several stages designed to compare the automatically converted dataset with manually labeled datasets for both detection and segmentation tasks. First, the NTUShrimp dataset was manually labeled with full-body bounding boxes (BodyBboxShrimpManual), head bounding boxes

(HeadBboxShrimpManual), and three-point polygons. Next, the polygon-labeled dataset was automatically converted to head bounding box (HeadBboxShrimp) and full-body bounding box (BodyBboxShrimp) formats. YOLOv11 and YOLOv12 were then used for training and validation in both segmentation and detection tasks. Finally, the models were exported, tested, and compared.

During data preprocessing for the segmentation task, we converted YOLO bounding box format (class_id, x_center, y_center, width, height) to four-point coordinates (class_id, x1, y1, x2, y2, x3, y3, x4, y4). The coordinates of the top-left, top-right, bottom-left, and bottom-right corners were calculated based on the center coordinates and box dimensions, and normalized to the range (0,1). The BodyBboxShrimp and HeadBboxShrimp datasets were then converted to BodyBboxShrimpSeg and HeadBboxShrimpSeg, respectively.

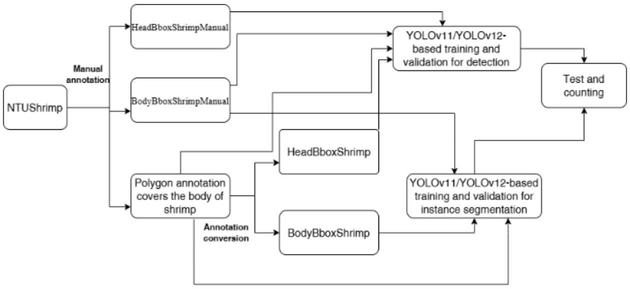
NTUShrimp
Manual annotation
HeadBboxShrimpManual
BodyBboxShrimpManual
Polygon annotation covers the body of shrimp
Annotation conversion
HeadBboxShrimp
BodyBboxShrimp
YOLOv11/YOLOv12-based training and validation for detection
YOLOv11/YOLOv12-based training and validation for instance segmentation
Test and counting

Figure 2. *Workflow of the Proposed Method*

In our previous study [1], polygon-based annotation was used to minimize background noise. However, most international studies on shrimp counting utilize bounding box annotation [2], [3], [4]. Converting labels from manually annotated polygons to bounding box format reduces cost and mitigates subjective errors during manual annotation [8], [9], [10]. To meet the requirements for diverse datasets in computer vision tasks related to shrimp counting, we propose a conversion algorithm as described below.

The algorithm aims to convert the original polygon annotation into two bounding boxes: one covering the entire shrimp body, and one covering the head region, which contains key biological features for object recognition.

Algorithm: Automatic conversion from polygon annotation to head and full-body bounding boxes.

Input:

- P: P: Set of polygon points (xi, yi) representing the shrimp contour

- P: Set of polygon points (xi, yi) representing the shrimp contour

P: Set of polygon points (xi, yi) representing the shrimp contour

Output:

- P: Set of polygon points (xi, yi) representing the shrimp contour

- P: Set of polygon points (xi, yi) representing the shrimp contour

Step 1: Compute the bounding box for the entire shrimp body

a. Find the minimum and maximum x and y values from polygon points P: x_min, x_max, y_min, y_max

b. Compute the center coordinates:

center_x = (x_min + x_max)/2,

center_y = (y_min + y_max)/2

c. Compute width and height:

width = x_max - x_min, height = y_max - y_min

d. Normalize these values:

center_x_norm = center_x / img_width

center_y_norm = center_y / img_height

width_norm = width / img_width

height_norm = height / img_height

e. Create box_full = (class_id, center_x_norm, center_y_norm, width_norm, height_norm)

Step 2: Identify the shrimp head region based on the shortest polygon edge

a. Initialize min_distance = ∞, edge = ∅

b. Iterate through all point pairs in P, compute Euclidean distance:

$$d = \sqrt{(xi - xj)^2 + (yi - yj)^2}$$

If d < min_distance, update edge ← $\{(x_i, y_i), (x_j, y_j)\}$ and min_distance=d

c. Define a new polygon for the head using the two closest points and the center:

new_polygon ← {edge[0], edge[1], (center_x, center_y)}

d. Repeat Step 1 for this new polygon to compute the head bounding box, then normalize:

box_head = (class_id, cx'_norm, cy'_norm, w'_norm, h'_norm)

Step 3: Return box_full and box_head

The above algorithm automates the conversion of polygon labels to bounding boxes. The full-body bounding box is determined by the smallest rectangle enclosing all polygon points, normalized for YOLO input format. Unlike traditional min-max methods, our algorithm leverages the geometric observation that the shortest polygon edge typically connects the shrimp's eyes. By combining this edge with the centroid, we create a new polygon focused on the head region, from which a smaller, more precise bounding box is generated for object detection.

The shortest edge is computed by:

$$\min\ \_distance = min_{1 \le i < j \le n} \sqrt{(xj - xi)^2 + (yj - yi)^2} \quad (4)$$

After identifying the closest point pair, we combine their segment with the full-body centroid to form a new three-point polygon. This polygon targets the head region, enabling the creation of a smaller, more accurate head bounding box. The normalization and YOLO format conversion follow the same procedure as for the full-body box.

The results of label conversion from polygons to full-body and head bounding boxes are visualized in Figure 3. The original polygon (blue) is the manual annotation. From this, two bounding boxes are generated: the full-body bounding box (red dashed line) enclosing the entire polygon, and the head bounding box (pink dashed line) determined by the shortest polygon edge and centroid. The newly constructed head polygon (green) highlights the shrimp's head region. Key features such as the bounding box centers (red and pink stars) and the shortest edge (cyan) are also clearly displayed. **(b)** A sample image with automatically generated labels for multiple postlarvae shrimp. Each shrimp is annotated with a polygon, full-body bounding box, and head bounding box as described in (a). This image demonstrates the scalability of the labeling method to real-world, high-density data.

The BodyBboxShrimp dataset contains YOLO-format files for full-body bounding boxes (outer rectangles), while the HeadBboxShrimp dataset contains YOLO-format files for head bounding boxes (smaller rectangles).

According to our experiments, manual annotation time depends on the label type, with an average shrimp density of 235.82 per image. Specifically, three-point polygon annotation takes about 20 minutes per image, while both full-body and head bounding box annotations take approximately 16 minutes per image.

In contrast, the automatic conversion from polygon to bounding boxes requires only about 0.67 seconds per image. Thus, in terms of efficiency, the automatic conversion method offers significant time savings over manual annotation.

Additionally, manual annotation is prone to subjective errors, while the automated method ensures higher consistency and stability.

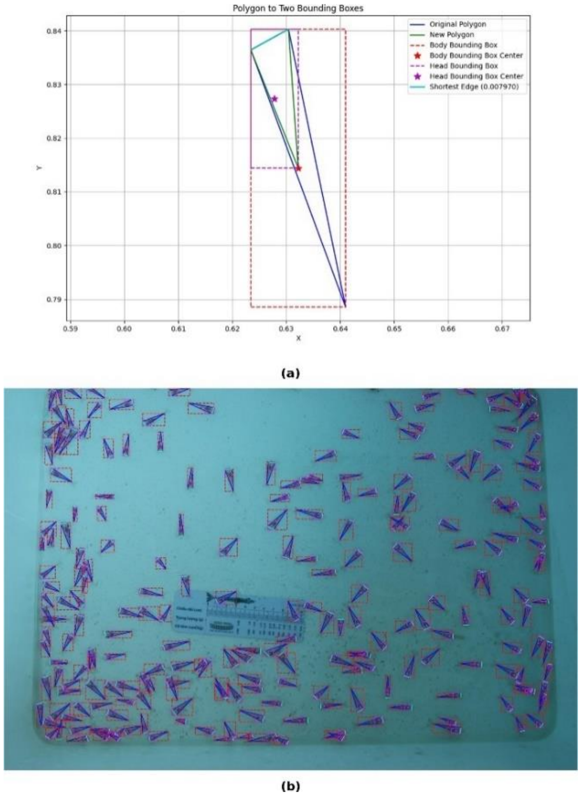

Figure 3. Visualization of Bounding Box Generation for Full Body and Head from Polygon Annotations: a) Geometric Representation, b) Visualization of Generated Full-Body and Head Bounding Boxes on Shrimp Images

To assess the similarity between the two annotation sets, we visualized and calculated the IoU values between manual and automatically converted labels on the same image. Visualization results are shown in Figure 4, where red labels represent manual annotations and yellow labels represent automatically converted annotations.

For full-body bounding boxes, the mean IoU (mIoU) reached a high value of approximately 0.86, demonstrating that the automatically converted labels are highly reliable and suitable for training object detection models. However, some discrepancies were observed due to variability in manual annotation positions and sizes, whereas automatic conversion produces more consistent and uniform bounding boxes.

For head bounding boxes, visualization and IoU distribution plots indicated high similarity, with mIoU values around 0.80. Notably, the automatically converted labels effectively and consistently cover the shrimp head region, mainly due to the use of fixed landmarks (the two eyes - the shortest edge) during polygon-to-bounding box conversion. In contrast, manual annotation is more challenging as annotators must subjectively select the optimal position and size for the head bounding box, leading to potential inconsistencies due to human factors.

Overall, the analysis shows that automatic conversion from polygon to bounding box yields stable and high-quality labels, enhancing the efficiency and consistency of the training dataset.

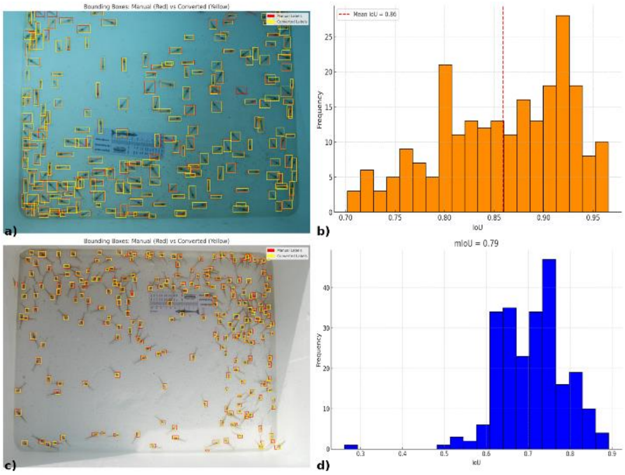

***Figure 4.** Visualization of the Comparison Between Manually Annotated and Automatically Generated Bounding Boxes from Polygon Labels for Two Annotation Types: Full Body and Head of Shrimp*

(a) Illustration of the overlap between manually annotated (red) and automatically generated (yellow) bounding boxes for the full body of shrimp; (b) Histogram of IoU values comparing the two annotation methods for full-body labels; (c) Illustration of the overlap between manually annotated (red) and automatically generated (yellow) bounding boxes for the head of shrimp; (d) Histogram of IoU values comparing the two annotation methods for head labels

3.3. Training and testing

With the datasets converted from polygons, we conducted training and testing on YOLOv11 and YOLOv12 models for object detection and instance segmentation tasks. Both models incorporate attention mechanisms, making them well-suited for detecting small objects such as postlarvae shrimp.

3.3.1. Shrimp counting based on object detection

Model training was performed using YOLO11n.pt and YOLO12n.pt. The bounding box datasets converted from the original polygon-annotated dataset, including BodyBboxShrimp and HeadBboxShrimp, were used for training and testing. The models were then evaluated on a test set consisting of 10 images of postlarvae shrimp collected at a different time, with smaller shrimp sizes. The results were compared with those obtained from models trained and tested on manually labeled datasets, namely HeadBboxShrimpManual and BodyBboxShrimpManual.

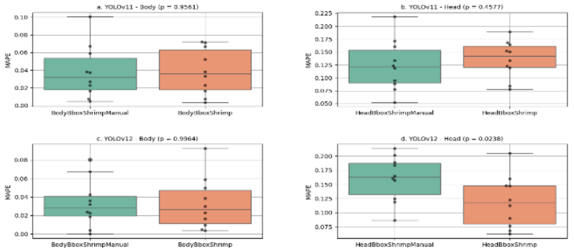

Figure 5. *Comparison of MAPE Errors in Shrimp Counting Using Manual and Automatically Generated Bounding Boxes from Polygon Labels for Full-Body and Head Annotations, After Training with YOLOv11 and YOLOv12.*
(a) YOLOv11 – full-body annotation; (b) YOLOv11 – head annotation; (c) YOLOv12 – full-body annotation; (d) YOLOv12 – head annotation

Based on the statistical analysis of MAPE boxplots for shrimp counting (Figure 5), the following observations can be made regarding the conversion from manual to automatic labeling:

- **Full-body:** Results show no statistically significant difference between the two labeling methods for both YOLOv11 (p = 0.9561) and YOLOv12 (p = 0.9964). The high p-values (>0.05) indicate that the prediction accuracy for shrimp body size is equivalent between manual and automatic labeling. This demonstrates that automatic label conversion does not compromise data quality for full-body detection.

- **Head:** However, there are notable differences. With YOLOv11, the p-value = 0.4577 still indicates no significant difference, but with YOLOv12, the p-value = 0.0238 (<0.05) suggests a statistically significant difference between the two methods. This result may be influenced by the test set, which contains postlarvae shrimp of smaller sizes.

We conducted a comparative evaluation of the effectiveness of automatic label conversion versus manual labeling for both bounding box types (head and full-body). The results are presented in Table 1.

Table 1. *Performance Comparison of Labeling Methods Using YOLOv11 and YOLOv12 Models in the Object Detection Task*

Model	Annotation Style	mAP50	MAPE
YOLOv11	Head Bounding Box Converted from Polygon	**80%**	13.43%
YOLOv12	Head Bounding Box Converted from Polygon	79%	**11.72%**
YOLOv11	Head Bounding Box Manually Annotated	77%	15.14%
YOLOv12	Head Bounding Box Manually Annotated	78%	12.20%
YOLOv11	Full-Body Bounding Box Converted from Polygon	**88.2%**	3.83%
YOLOv12	Full-Body Bounding Box Converted from Polygon	87.5%	**3.26%**
YOLOv11	Full-Body Bounding Box Manually Labeled	84.8%	3.80%
YOLOv12	Full-Body Bounding Box Manually Annotated	84.2%	3.27%
YOLOv5m6 [10]	Full-Body Bounding Box Manually Annotated	-	5.7%

a. Analysis of mAP50 performance by labeling method

From Table 1, the model's performance on automatically converted datasets is higher than on manually labeled datasets. The mAP50 of YOLOv11 increased significantly from 84.8% to 88.2% (+3.4%) when evaluated on the automatically converted dataset. Similarly, YOLOv12's mAP50 increased from 84.2% to 87.5% (+3.3%). Thus, automatic label conversion improves mAP50 by 3.3–3.4% compared to manual labeling, and this result is stable across different models.

Similarly, for the head bounding box dataset converted from polygons, mAP50 values are also higher (80% for YOLOv11 and 79% for YOLOv12) than for manual annotation (77% and 78%). The consistent results across both models and both bounding box types demonstrate the stability and reliability of the automatic conversion method.

The improvement in mAP50 indicates that automatic conversion helps eliminate human subjectivity in labeling, resulting in more uniform and precise bounding boxes and thus improved model recognition.

b. Counting results by labeling method

When training on automatically converted datasets, the error rate is lower than when training on manually labeled datasets for both full-body and head bounding boxes. For full-body, the automatic method yields low and comparable error rates (MAPE: 3.26% for YOLOv12 and 3.83% for YOLOv11) to the manual method (MAPE: 3.27% for YOLOv12 and 3.80% for YOLOv11). For head bounding boxes, after training YOLOv11 on the automatically converted dataset, the error rate is 13.43%, lower than the manual method (15.14%), and similarly for YOLOv12 (11.72% vs. 12.20%).

YOLOv12 consistently outperforms YOLOv11 in counting accuracy across all experimental configurations. For head bounding boxes (auto from polygon), YOLOv12 achieves MAPE 11.92% compared to 13.43% for YOLOv11. For manual head annotation, YOLOv11 achieves MAPE 12.40% versus 15.34% for YOLOv12. For full-body (auto from polygon), YOLOv12 achieves MAPE 3.26% versus

3.83% for YOLOv11, a reduction of 0.57 percentage points. Similarly, for manual full-body annotation, YOLOv12 achieves MAPE 3.27% versus 3.80% for YOLOv11, a reduction of 0.53 percentage points. These results demonstrate that improvements in YOLOv12 architecture have significantly reduced counting estimation errors.

c. Comparison of full-body vs. head bounding boxes

Results show that the full-body bounding box method outperforms the head bounding box method. The mAP50 for full-body bounding boxes is significantly higher (84%–88%) than for head bounding boxes (77%–80%) in all cases.

The most notable difference is the much lower counting error rate for full-body detection. For head bounding boxes, MAPE ranges from 11.72% to 15.14% (average ~13.12%), while for full-body bounding boxes, MAPE ranges from 3.26% to 3.83% (average ~3.54%). Thus, full-body detection yields counting accuracy about 3.7 times higher than head detection. The lowest error rate for head detection (11.92%) is still about 3.5 times higher than the highest error rate for full-body detection (3.83%).

This indicates that the full body contains more geometric features, allowing the model to detect more accurately. In contrast, the head region may be occluded, subject to background noise, or lack a clear shape, resulting in lower mAP and counting accuracy.

3.3.2. Shrimp counting based on instance segmentation

To evaluate the quality of datasets automatically converted from polygon to bounding box labels, we further conducted training for the instance segmentation task. Instance segmentation combines object detection and semantic segmentation. However, as shown in the detection task, the head region of postlarvae shrimp does not contain many morphological features, so we did not evaluate the head bounding box datasets (auto/manual) for this task.

The procedure is similar to the detection task: the polygon-annotated dataset is automatically converted to YOLO-format full-body bounding boxes, then further converted to BodyBboxShrimpSeg_Auto with four-point coordinate labels suitable for instance segmentation. Similarly, the manually labeled full-body bounding box dataset is converted to BodyBboxShrimpSeg_Manual. The datasets were trained and tested using YOLO11n-seg.pt and YOLOv12 models for instance segmentation. Results are shown in Table 2.

Table 2. *Performance Comparison of Annotation Methods Using YOLOv11 and YOLOv12 Models in the Instance Segmentation Task*

Model	Dataset	mAP50	MAPE
YOLOv11	Full-Body Bounding Box Manually Annotated	85.4%	3.81%
YOLOv12	Full-Body Bounding Box Manually Annotated	78.9%	4.23%
YOLOv11	Full-Body Bounding Box Converted from Polygon	89.0%	3.78%
YOLOv12	Full-Body Bounding Box Converted from Polygon	79.7%	3.9%
UCBAM[1]	Polygon-Based Annotation	89.2%	3.38%

The results indicate that, similar to the detection task, model performance on automatically converted datasets is higher than on manually labeled datasets. The mAP50 values for YOLOv11 and YOLOv12 (89% and 79.7%) on the automatically converted dataset are 0.8–3.6% higher than on the manual dataset (85.4% and 78.9%).

With our automatic conversion method, model performance for instance segmentation is even better than for detection, with YOLOv11's mAP50 increasing by 1%.

MAPE analysis shows that all models achieve good accuracy in object size estimation, with MAPE ranging from 3.78% to 4.23%. Notably, YOLOv11 always yields lower MAPE than YOLOv12 on the same data, indicating superior size prediction accuracy. Using bounding boxes converted from polygons instead of manual annotation further improves MAPE for both models, with YOLOv11 achieving the lowest error at 3.78%. While the error is slightly higher compared to the UCBAM model (MAPE 3.38%) on manually polygon-annotated data [1], this demonstrates that automatic conversion from polygons to bounding boxes achieves high accuracy in object size estimation, improving model estimation and reducing errors in practical applications requiring high precision.

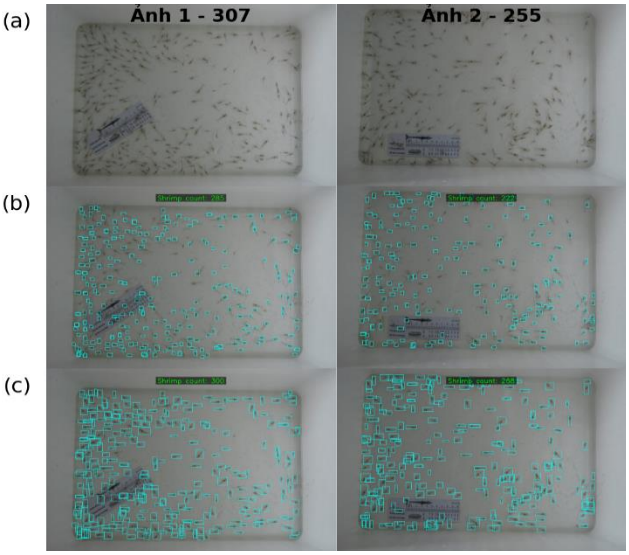

Figure 6. *Visualization of Shrimp Counting Tests: (a) Input Image, (b) Testing YOLOv11 Model Trained on HeadBboxShrimp, (c) Testing YOLOv12 Model Trained on BodyBboxShrimp*

4. Discussion

The proposed method for automatic conversion from three-point polygon labels to derived datasets for various computer vision tasks - such as object detection and semantic segmentation in shrimp quantification - has produced promising results. This process not only optimizes costs in designing diverse datasets for research and practical applications but also eliminates human subjectivity in labeling. However, several issues remain for further discussion:

4.1. High mAP50 but low MAPE

In model testing, there is a trade-off between mAP50 and counting accuracy. For object detection, YOLOv11 excels in object detection (mAP50 up to 88.2%) but tends to have higher counting errors (MAPE 3.83%), while YOLOv12, though slightly lower in detection performance (87.5%), achieves a lower counting error (MAPE 3.25%) when using auto-converted data.

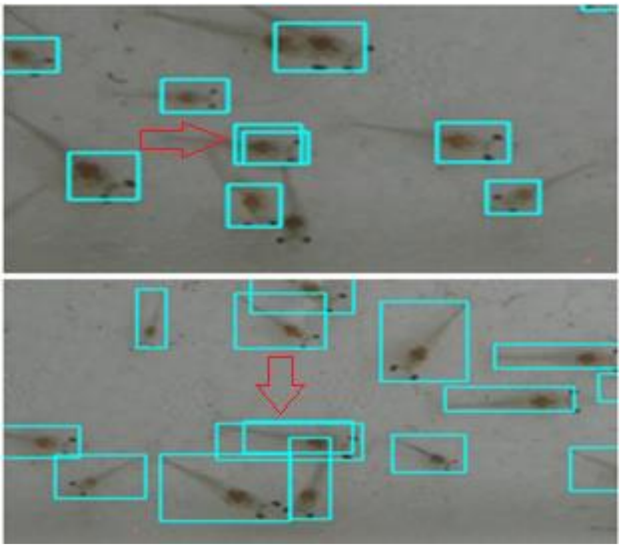

Figure 7. *The Phenomenon of Counting the Same Object Twice*

Visualization of counting results shows that some shrimp are counted twice (Figure 7), which explains the trade-off in YOLOv11 due to two main reasons. First, the IoU threshold in the Non-Maximum Suppression (NMS) algorithm may not be optimal for small objects like postlarvae shrimp. NMS helps remove overlapping bounding boxes, but if the confidence scores do not accurately reflect true object presence, NMS may not effectively eliminate duplicates, leading to counting errors [20], [21]. Second, head bounding boxes are often small and close together, tending to overlap in dense regions. While the model detects more boxes (increasing mAP50), duplicates in detection lead to higher MAPE.

4.2. Performance issues with head bounding box datasets

Results show that both YOLOv11 and YOLOv12, when trained on automatically generated head bounding boxes, achieve higher mAP50 than when trained on manually labeled head bounding boxes (YOLOv11: 77% to 80%; YOLOv12: 78% to 79%). This reflects better recognition accuracy with automatic labeling. However, counting errors remain much higher for head bounding boxes than for full-body bounding boxes, regardless of labeling method.

We believe that the high MAPE is not inherent to the automatic labeling method. The discrepancy in counting results may be due to other factors:

- The test data (PL25) differs significantly in size from the training data (PL28). Thus, models trained on larger head bounding boxes (PL28) may produce boxes covering the entire shrimp body when applied to PL25, causing confusion and missing small heads.

- The training set lacks sufficient diversity in object size and shape, limiting the model's generalization to different developmental stages.

The promising results from automatic bounding box generation provide a basis for further research. We plan to expand the training dataset with more diverse shrimp sizes to improve generalization and reduce undercounting when sample types change. In addition, we will study data augmentation and balancing techniques by size or developmental stage, and employ multi-scale models to simultaneously detect small (PL25 heads) and large objects (PL28 and above).

4.3. YOLOv11 or YOLOv12 for shrimp counting

The spatial attention mechanism in YOLOv11 (C2PSA) enables the model to learn fine spatial features, which are well-suited for shrimp counting, where objects are small, close together, or overlapping. In contrast, YOLOv12 uses Area Attention, dividing the image into large regions, which is faster but may have difficulty distinguishing objects at boundaries. This explains why YOLOv11 achieves higher mAP50 in both detection (88.2%) and segmentation (89%). Moreover, YOLOv11's overlapping attention mechanism is better suited for segmentation, while YOLOv12 maintains higher accuracy for detection. Thus, attention mechanisms play a crucial role in model performance for specific tasks.

5. Conclusion

In this study, we developed and presented a method for converting from polygon labels to full-body and head bounding boxes for postlarvae shrimp. The novelty lies in the proposed conversion algorithm based on the shortest edge principle to create optimal and consistent bounding boxes. This algorithm not only ensures stable training data but also significantly improves YOLO model performance in shrimp detection and counting tasks.

We also introduced a three-point polygon annotation technique, which greatly reduces time and cost compared to traditional manual labeling. The automatic conversion process from polygons to bounding box labels has optimized data processing, minimized errors, and eliminated inconsistencies due to human subjectivity, thereby enhancing the quality of training datasets.

Experimental results, evaluated through mean IoU (mIoU), show that the automatically converted labels are highly consistent with manual labels (mIoU $\geq$ 0.80 for both full-body and head bounding boxes), confirming the effectiveness and reliability of the proposed conversion method. Furthermore, experiments with YOLOv11 and YOLOv12 in detection and segmentation tasks demonstrated that automatically generated data achieves high quality, with YOLOv12 achieving very high accuracy (MAPE only 3.26%) in shrimp counting.

Future research will focus on integrating and optimizing the Non-Maximum Suppression (NMS) algorithm to reduce overcounting in YOLOv11. We will also collect more diverse shrimp data to further improve model performance and generalization in practical applications.

Acknowledgements: We would like to express our sincere gratitude to Nha Trang University for funding this research project (ID: TR2024-13-16). We also thank Nha Trang Marine Research Station Co., Ltd. (NTMaRS), for their support in collecting postlarval shrimp image data. Special thanks to Dr. Le Hoang Thanh from the University of Wollongong, Australia, for providing valuable knowledge and inspiration for this research.

REFERENCES

[1] C. T. Phuong, P. T. K. Ngoan, B. T. H. Minh, M. D. Thao, and T. V. Hoan, "Deep learning research and applications in computer vision: Experiments with shrimp counting problem", in *Proc. 27th National Conference on Information and Communication Technology (VNICT 2024)*, Hanoi, Vietnam, 2024, pp. 410–422.

[2] H. Duan *et al.*, "Shrimp Larvae Counting Based on Improved YOLOv5 Model with Regional Segmentation", *Sensors (Basel)*, vol. 24, no. 19, Sep. 2024, doi: 10.3390/s24196328.

[3] S. Armalivia, Z. Zainuddin, A. Achmad and M. A. Wicaksono, "Automatic Counting Shrimp Larvae Based You Only Look Once (YOLO)," in *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, Bandung, Indonesia, 2021, pp. 1-4, doi: 10.1109/AIMS52415.2021.9466058.

[4] L. Zhang, X. Zhou, B. Li, H. Zhang, and Q. Duan, "Automatic shrimp counting method using local images and lightweight YOLOv4", *Biosyst Eng*, vol. 220, pp. 39–54, 2022, doi: https://doi.org/10.1016/j.biosystemseng.2022.05.011.

[5] T. Rädsch *et al.*, "Quality Assured: Rethinking Annotation Strategies in Imaging AI BT", in A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., Computer Vision – ECCV 2024, Cham: Springer Nature Switzerland, 2025, pp. 52–69. doi: https://doi.org/10.1007/978-3-031-73229-4_4.

[6] S. Nou, J.-S. Lee, N. Ohyama, and T. Obi, "The improvement of ground truth annotation in public datasets for human detection", *Mach Vis Appl*, vol. 35, no. 3, p. 49, 2024, doi: 10.1007/s00138-024-01527-1.

[7] J. F. Mullen, F. R. Tanner, and P. A. Sallee, "Comparing the Effects of Annotation Type on Machine Learning Detection Performance", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 855–861. doi: 10.1109/CVPRW.2019.00114.

[8] T.-Y. Lin *et al.*, "*Microsoft COCO: Common Objects in Context BT*", in D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Computer Vision – ECCV 2014, Cham: Springer International Publishing, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[9] A. Gupta, P. Dollár, and R. Girshick, "LVIS: A Dataset for Large Vocabulary Instance Segmentation", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Long Beach, CA, USA, 2019, pp. 5351-5359, doi: 10.1109/CVPR.2019.00550.

[10] M. Cordts et al., "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 3213-3223, doi: 10.1109/CVPR.2016.350.

[11] C. Bukas *et al.*, "Robust deep learning based shrimp counting in an industrial farm setting", *J Clean Prod*, vol. 468, p. 143024, 2024, doi: https://doi.org/10.1016/j.jclepro.2024.143024.

[12] S. Asmak, D. Rizaldi, R. Saputra, A. Abseno, V. Hananto, and E. Oktarina, "A Mobile App for Counting Shrimp Larvae Based on the YOLO V5 Method", *Journal of Computer Electronic and Telecommunication*, vol. 5, Dec. 2024, doi: 10.52435/complete.v5i2.647.

[13] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements", *arXiv*, 2024, [Online]. Available: https://arxiv.org/abs/2410.17725.

[14] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN.", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, [Online]. Available: https://arxiv.org/abs/1911.11929.

[15] R. Khanam and M. Hussain, "A Review of YOLOv12: Attention-Based Enhancements vs. Previous Versions", *arXiv,* 2025, [Online]. Available: https://arxiv.org/abs/2504.11995.

[16] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 9992-10002, doi: 10.1109/ICCV48922.2021.00986.

[17] Z. Huang *et al.*, "CCNet: Criss-Cross Attention for Semantic Segmentation", *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 6, pp. 6896–6908, 2023, doi: 10.1109/TPAMI.2020.3007032.

[18] X. Dong *et al.*, "CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows", in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 12114-12124, doi: 10.1109/CVPR52688.2022.01181.

[19] Ultralytics, "Model Training with Ultralytics YOLO", *ultralytics.com*, November, 12, 2023. [Online].Available: https://docs.ultralytics.com/modes/train/ [Accessed: Mar. 03, 2024].

[20] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression.", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6469-6477, doi: 10.1109/CVPR.2017.685.

[21] J. Gilg, T. Teepe, F. Herzog, P. Wolters, and G. Rigoll. "Do We Still Need Non-Maximum Suppression? Accurate Confidence Estimates and Implicit Duplication Modeling with IoU-Aware Calibration", in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024 pp. 4838-4847. doi: 10.1109/WACV57701.2024.00478.