Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 33, Num. 2 (2026) 370-377

RESEARCH ARTICLE

Livestock Fish Larvae Counting using DETR and YOLO based Deep Networks

Contagem de Larvas de Peixe em Cativeiro Usando Redes Profundas Baseadas em DETR e YOLO

Daniel Ortega de Carvalho[1], Luiz Felipe Teodoro Monteiro[1], Fernanda Marques Bazilio[2], Gabriel Toshio Hirokawa Higa[1]*, Hemerson Pistori[1, 3]

Abstract: Counting fish larvae is an important yet demanding task in aquaculture. In this work, we evaluate four neural networks, including convolutional and transformer-based architectures, in different sizes, in this counting task. We present a new annotated image dataset with less data collection requirements than preceding works, with images of spotted sorubim and dourado larvae. We also combined the neural networks with a tiling technique, in order to improve the counting performance, and performed a comprehensive experimental evaluation on the methods. We achieve a MAPE of 4.46% (± 4.70) with an extra large real-time detection transformer, and 4.71% (± 4.98) with a medium-sized YOLOv8. The results indicate the feasibility of using deep learning to count spotted sorubim and dourado larvae, which may become an important tool to enhance the efficiency of aquaculture farms that deal with these species.
Keywords: machine learning — transformers — convolutional neural networks — fish larvae — fish counting

Resumo: A contagem de larvas de peixe é uma tarefa importante, mas exigente, na aquicultura. Neste trabalho avaliamos quatro redes neurais, incluindo arquiteturas convolucionais e baseadas em transformers, em diferentes tamanhos, na referida tarefa. Apresentamos um novo conjunto de imagens anotadas com menos requisitos de coleta do que aqueles de trabalhos precedentes, com imagens de larvas de pintado e dourado. Também combinamos as redes neurais com uma técnica de ladrilhagem, para melhorar a contagem, e fizemos uma avaliação experimental abrangente dos métodos. Alcançamos uma MAPE de 4,46% ($\pm 4,70$) com um transformer extra grande para detecção em tempo real, e 4,71% ($\pm 4,98$) com YOLOv8 de tamanho médio. Os resultados indicam que é possível usar aprendizado profundo para contar larvas de pintado e dourado, o que pode se tornar uma ferramenta importante para melhorar a eficiência de estabelecimentos aquaculturais que lidem com essas espécies.
Palavras-Chave: aprendizado de máquina — transformers — redes neurais convolucionais — larvas de peixe — contagem de peixes

[1] *Inovisão, Universidade Católica Dom Bosco (UCDB), Brazil*
[2] *Agropeixe Ltda.*
[3] *Universidade Federal de Mato Grosso do Sul (UFMS), Brazil*
***Corresponding author**: gabrieltoshio03@gmail.com
DOI: http://dx.doi.org/10.22456/2175-2745.150926 • **Received:** 14/10/2025 • **Accepted:** 03/11/2025

1. Introduction

Counting fish larvae is an important task in aquaculture, allowing for monitoring growth and development of the fish from early stages. Traditionally, fingerling identification is performed manually, which means that it is usually slow and error-prone. It can also compromise the development of the animals. On the other hand, the use of automation is increasing in the aquaculture industry. By using new techniques such as artificial intelligence and, more specifically, machine learning, it is possible to enhance monitoring of the fish and of their habitats, and even to manipulate them [1]. In this regard, the advancement of image processing techniques has brought about new approaches for automating larvae counting, which may lead to greater agility in the processes of aquaculture [2], including fish larvae counting. If a high enough performance with lower resource requirements is possible, models that satisfactorily count fish larvae may become a tool to reduce the time required by the task, as well as its costs, leading to a more efficient and sustainable production [3].

In this work, we evaluate State-of-the-Art deep neural networks for object detection, including transformer-based architectures, used to count larvae in high resolution images, some of them containing high density of individuals, along with a tiling technique. The smallest MAPE achieved in this work was 4.46 ($\pm$4.70) with an extra large real-time detection transformer. Differently from other works, such as that by Costa *et al.* [4], we use data that adequately represents real-life scenarios, instead of increasing procedural requirements in order to enhance image capturing conditions. Thus, the main contribution of this paper is the combination of object detection neural networks with a tiling technique in order to count fish larvae of two species (namely, dourado and spotted sorubim), a problem that has not yet been widely studied. Furthermore, for this evaluation an image dataset was created and a comprehensive experimental evaluation was carried out.

(a) **(b)**

Figure 1. An illustration of the sampling process used to obtain the larva bowls that were photographed, along with some of the bowls available for this purpose and utilized in this study.

2. Related Works

In aquaculture, many works have been done on counting using deep learning, both regarding adult fish and their larvae, and also other aquatic animals. For instance, Khai *et al.* [5] proposed the use of a region-based convolutional neural network (R-CNN) with a ResNet101 backbone to identify and count fish in underwater photos. The experiment achieved an accuracy of 95.30% on the validation dataset. Yu *et al.* [6] proposed a deep learning model for fish counting called MAN, which was developed based on multiple modules and an attention mechanism. The authors achieved a counting precision of 97.12% in different aquaculture environments. Finally, Li *et al.* [7] proposed a lightweight neural network for fry fish counting, reporting an MAE of 3.33.

Some works have been done on larvae counting of different aquatic animals. Kakehi *et al.* [8] proposed the use of deep learning to identify oyster larvae, with an ultimate goal of counting them, reporting an f-measure of 86.4%. Hu *et al.* [9] proposed the use of deep neural networks for density estimation to count white shrimp larvae, reporting a MAPE of 2.18%. Liu *et al.* [10] proposed Shrimpseed_Net, a neural network for counting shrimp seed, reporting a MAE of 17.28. Rothschild *et al.* [11] proposed the use of an adapted YOLOv5 model for crustacean larvae counting in different phases.

Some of the fish counting works focus in other early development stages, such as fry fish [7] and fingerlings [12], or simply on juvenile fish [13]. Among the works that focused on fish larvae, the ones by Costa *et al.* [14, 4] employed a set of manually annotated images containing over 6000 tilapia larvae and achieved an accuracy of 98.5% and a mean absolute error of 1.43. In this case, this was achieved by using a complex image capturing apparatus that included a light table and Petri dishes.

The applicability of the proposed approaches in real-life scenarios and use cases can be considered a general concern within the field of automated fish counting. In some of the related works mentioned here, data collection involved steps aimed at enhancing image capturing conditions, in order to increase the performance of deep learning models. In this work, we evaluate four object detection neural network architectures (plus size variants) in the task of spotted sorubim and dourado larvae counting, with images captured by smartphone and less data collection requirements. We proceed by using an image tiling technique and also evaluate different tiling scales, in order to increase detection performance.

3. Materials and Methods

3.1 Image Dataset

For this work, a new image dataset was created[1]. It contains 162 images of small trays with spotted sorubim (*Pseudoplatystoma corruscans*), dourado (*Salminus brasiliensis*) and streaked prochilod (*Prochilodus lineatus*) larvae. Images were collected in Projeto Pacu, an aquaculture company that specializes in large scale fish reproduction located in Terenos, Mato Grosso do Sul, Brazil. Data collection was performed by sampling, according to the following procedure. In virtue of the business process, all larvae are stored in hatcheries, from which a sample is collected using a 50 ml falcon tube. The sample is then transferred to one of a variety of common bowls dedicated to this purpose. Figure 1a illustrates the process of sampling with a tube and Figure 1b shows some of the bowls used in this study.

For image acquisition, a short video of the bowl is taken after the sampling process. Then, one frame per video, and one frame per sample therefore, is selected based on its visual quality. The main criteria for selection is occurrence and quantity of blurred sections. After being selected, the images were annotated with bounding boxes for object detection with only one generic class, fish, for spotted sorubim and dourado larvae. Given the manual annotation with bounding boxes, the number of larvae in the bowl was also obtained as the number of bounding boxes, which was taken as the ground truth for larvae counting. In this work, we focus on these

[1]The dataset is available on request to the corresponding author.

two species. Larvae of streak prochilod were not annotated. This option was due mainly to the fact that the visual aspect of these larvae differs from that of the other ones, in that it tends to be transparent. A close-up example can be seen in Figure 2, along with examples of other species. Given that this is so, it is reasonable to assume that it will require a more specific procedure. Furthermore, this species is used as fish food in dourado production, and while counting them may be useful, it is not the focus of this study. Therefore, in this work streaked prochilod larvae were considered noise, and shall be dealt with in future works.

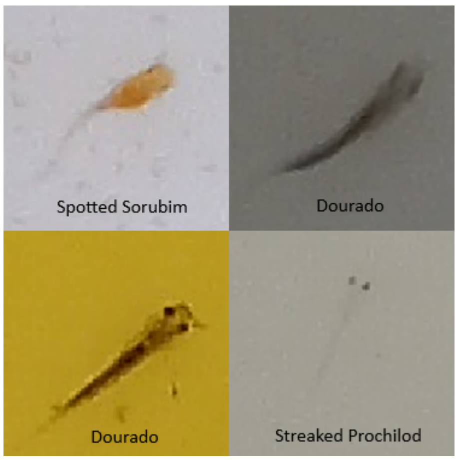

Figure 2. Close-up examples of the larvae used in this study. Streaked prochilods were not counting targets in this study, and it is not intuitively obvious that the same techniques used here will have the same efficacy in counting them as they have in counting spotted sorubim and dourado larvae. Nonetheless, it should be clear from this example that they can be treated as noise.

The resulting images have resolution of (2604, 4624). They were captured by a specialized aquaculture professional using a Samsung SM-A325M smartphone. As described, the data collection process was not made under idealized laboratory conditions, except for the choice of plain bowls and the sampling methodology itself. This leads to the presence of noise in the images, such as dirt, background differences, and the presence of different species in the same sample. In total, there are 13,563 annotated larvae in 162 images. Figure 3 shows the number of annotations per image in the dataset. One should notice that most images contain around 50 annotations, but also that some of them have over 200 annotated larvae. Figure 4 shows some of the images collected in this study.

3.2 Image Tiling

An image tiling strategy was used for processing with the neural networks, mainly due to hardware limitations in two senses. First, regarding the hardware available for this study. Second, and most importantly, regarding limited hardware, such as smartphones, that can be used in real-life scenarios. In this work, two strategies were used for image tiling. The first one consisted in extracting fixed-size fragments of shape

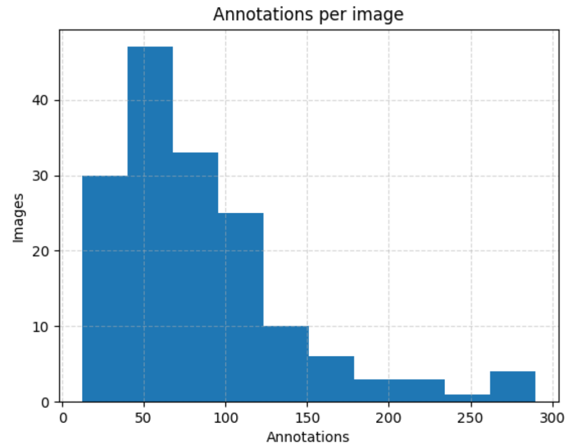

Figure 3. Histogram of the number of annotations per image. Mainly, one should notice that most images have around 50 annotations, but also that some of the images have over 200 annotations.

(640, 640), and was used in the training phase of the neural networks.

The second strategy is to use a scaling factor, which gives the size of the fragments as $ceil(min(H,W)*(s))$, where H and W are the height and the width of the image, respectively, and s is the scaling factor. That is, each fragment is a square whose sides are given by the ceil function of the product between the smallest side of the image and the scaling factor. The image is padded with black pixels, if necessary. This second strategy was used in a second training step, in order to adjust the scaling factor, and was also used during the testing phase. Figure 5 shows an example of fixed-size tiling in an image of the dataset.

Since the tiled images were annotated for object detection, and tiling did not consider whether or not an annotation would span two fragments, a rule was established, according to which an annotation that went beyond the boundaries of the fragment would only be kept if at least 60% of its area was contained within the fragment.

3.3 Deep Learning

In this work, we evaluate four neural network architectures in the task of fish larvae counting, including convolution- and transformer-based architectures. We also test size variations, summing up to nine models evaluated. Next, we comment briefly on each architecture, and present the different versions utilized.

The first architecture evaluated by us is YOLOv8, a state-of-the-art convolutional neural network, focused on accuracy and real-time detection [15]. One of its differentials is that it doesn't use the anchoring method, which is accomplished by directly predicting the center of the object instead of the displacement from a known anchor box. This simplifies the process and speeds up Non-Maximum Suppression (NMS), a complex post-processing step.

(a) (b) (c) (d)

Figure 4. Some of the images that constitute the full dataset. One should notice that, since there was no control for obtaining images in ideal and standardized conditions, the images in the dataset present some degree of variability in lighting conditions and background, as well as dirt and the dark eyes of streaked prochilod larvae.

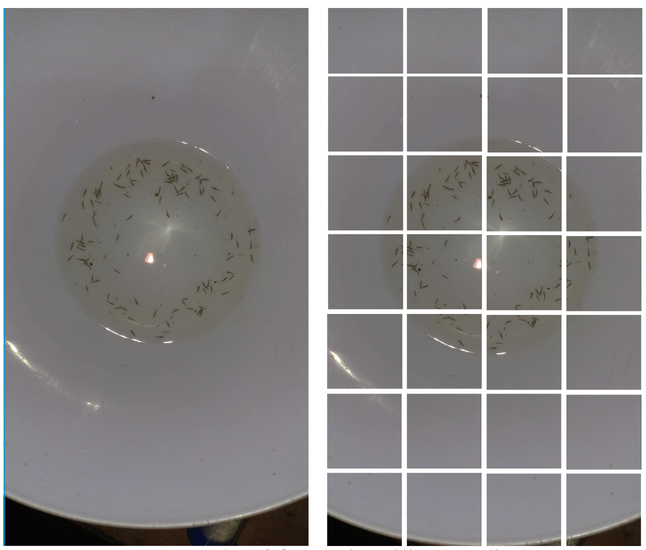

Figure 5. An example of fixed-size tiling applied on an image of the dataset. Another strategy used in this work was tiling according to a scale factor.

Model	Variant	Params. (M)
YOLOv8	nano	3.2
	small	11
	medium	26
	large	44
	extra large	68
RT-DETR	large	32
RT-DETR	extra large	67
DETR-ResNet50		41
Deformable DETR		40

Table 1. Number of parameters (in Millions) for each architecture evaluated in this work, specified by variant (when applicable).

The second architecture we evaluated in this work is the detection transformer (DETR) [16], an object detection network based both on transformers [17, 18] and convolutions. DETR initially uses ResNet, a CNN, as its backbone to extract a set of features from the input image. These features are then passed to an encoder-decoder Transformer, which uses attention mechanisms to capture the global relationships between the features throughout the image.

As a third architecture, we tested the Deformable DETR, an evolution of DETR that introduced a deformable attention mechanism [19]. While DETR uses global attention, considering all positions in the image uniformly, Deformable DETR allows the model to focus dynamically on a set of positions of interest. This is achieved through deformable attention modules that adapt the model's focus area during training. With this strategy, Deformable DETR is expected to achieve higher performance than DETR, while using fewer epochs for training.

Finally, the last architecture evaluated in this work is the real-time detection transformer (RT-DETR) [20]. The RT-DETR (Real-Time DEtection TRansformer) model is an architecture for real-time object detection. It utilizes an encoder and a query selection method that takes into account the accuracy of the overlap between predicted and actual object areas (IoU) to improve its results. It also offers the possibility to adjust the speed at which it processes images to identify objects without the need to go through the training process again.

3.4 Experimental Design

In order to evaluate the Neural Networks for object detection in the proposed approach, the dataset was divided into training (60%), validation (20%) and test (20%) sets, and the experiment followed a three-step procedure. First, the object detection models were trained and validated for fish larvae detection. In this step, all models were fine-tuned without freezing any layers. For training, the maximum number of epochs was set to 100 for the Deformable DETR, and 400 for the RT-DETR and also for the YoloV8 models. Furthermore, early stopping was used with 50 epochs of patience. Regarding Deformable DETR, validation loss values were monitored. For YoloV8 and RT-DETR an average calculated on the mAP values were monitored.

After the neural networks were trained, an inference hyperparameter tuning step took place, in which the confidence thresholds and the tiling scales were combined for each neural network. Several values were iteratively tested, where each iteration was carried out on a smaller range of values, with increasing precision. This second phase was carried out on the training set. Additionally, in order to reduce the time required for computation, we downscaled the image resolution by a factor of 0.5. Since most of the images in the dataset were in vertical orientations, we also used the following data augmentation techniques: rotation between 0°and 360°without changing image orientation, clockwise rotation between 0°and 90°changing image orientation, and horizontal and vertical mirroring. In order to choose a combination of hyperparameters, we calculated the MAE, MAPE and RMSE for each combination of hyperparameters, applied min-max normalization and chose the combination that yielded the smallest sum. This same procedure is used to order the results in the following Section.

Finally, the models were tested with the hyperparameters chosen for each model in the preceding step. In order to evaluate the results, we used the Analysis of Variance (ANOVA), followed by TukeyHSD, by considering each image a repetition, both at 5% significance threshold. Given this procedure, the hypothesis test was applied on the absolute difference and on the absolute percentage difference (with which MAE and MAPE are calculated, respectively). We also calculated the coefficient of determination (R^2), and generated scatterplots.

4. Results and Discussion

Table 2 shows the confidence thresholds and the tiling scales chosen in the post-training hyperparameter tuning step for each model evaluated in this work. It is possible to see that both the confidence threshold and the tiling scale varied considerably, ranging from 25% up to 70%, and from 20% up to 50%, respectively.

Table 3 shows average MAE, MAPE and RMSE results in the test set, along with their standard deviations, for each architecture evaluated, ordered by smallest sum of normalized means. Table 3 also shows the results of the TukeyHSD in compact letter display format. One can see that rtdetr-x achieved the smallest average MAE and MAPE on the test set, with the smallest standard deviation on the later, and second smallest on the former. However, yolov8m achieved the smallest average RMSE, and also the smallest sum of normalized means. The ANOVA test was indicative of statistically significant difference for both MAE and MAPE, with $p = 0.00144$ for MAE, and $p = 0.00164$ for MAPE. On the other hand, the post-hoc TukeyHSD applied did not indicate that either yolov8m or rtdetr-x differed from the other models, exception made to the detr-resnet-50.

Model	Conf. threshold	Tiling scale
yolov8n	45%	40%
yolov8s	40%	50%
yolov8m	40%	50%
yolov8l	50%	30%
yolov8x	45%	30%
rtdetr-l	70%	20%
rtdetr-x	50%	50%
detr-resnet-50	40%	30%
deformable-detr	25%	20%

Table 2. Confidence thresholds and tiling scales chosen for each model in the hyperparameter tuning step.

Figure 6 shows the scatterplots of test predictions, compared to the ground truth counting, along with the coefficient of determination. These plots, along with Table 3, show that most models achieved an R^2 close to 1, which indicates that most models achieved a strong linear correlation. The nano variant of the YOLOv8 architectures achieved an R^2 higher than larger variants, albeit smaller than the R^2 achieved by the medium version. The detection performed by the nano variant is show in Figure 7. As indicated by the error measurements, the Deformable DETR and the DETR with ResNet50 backbone showed a smaller R^2 value, which is indicative of an inferior adjustment. Finally, the medium version of YOLOv8 and the extra large version of RT-DETR showed the highest R^2 values. Visually, one can see that they also presented the smallest deviation from the best fit line, especially when higher counting numbers are considered.

The results achieved were below those found by Costa *et al.* [4]. Arguably, the main reason for this is that in their case the images were taken in an idealized environment, with more clean backgrounds, higher contrast between larvae and background, better illumination, and more clean water. Their procedure, however, imposed additional requirements for image collection, and is less suitable for real-life use cases. In Figure 8, it is possible to see a close-up example that illustrates how dirt can be present in the water, in color and format that are similar to that of fish larvae, and act as noise for the neural networks. The same can be argued regarding the work by Fernandes *et al.* [12], who reported recall rates of 99% in the detection of serrasalmid fingerlings. In this case, the fish were in a more developed stage, but the image capturing was also highly standardized. This indicates that the standardization of conditions such as background and lighting can, of course, improve results and even, from the point of view of evaluation, replicability. On the other hand, this standardization also imposes additional requirements for image collection,

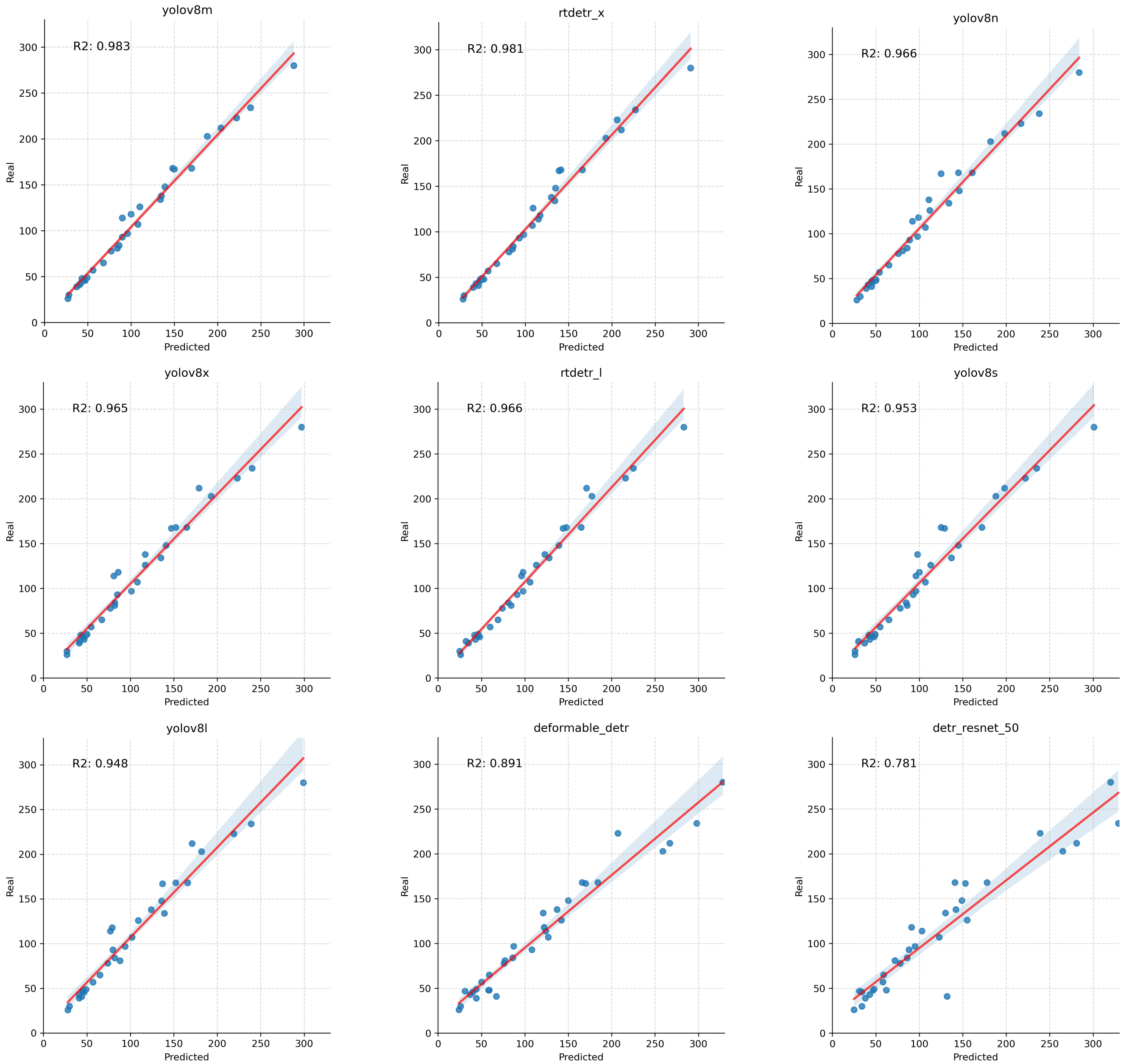

Figure 6. Scatterplots of ground truth counting and predictions in the test set for each model evaluated in this study, along with their coefficients of determination.

which might be undesirable – in some cases, it might hinder the usage of the system.

As discussed by Costa *et al.* [14] (although for tilapia larvae; we claim that these remarks do apply to the species we are working with as well), even in controlled, close-to-ideal, environments, the are some intrinsic difficulties to the problem. For instance, larvae agglomerations are an important issue, which is actually intensified by the less restrictive sampling used here. Another difficulty is that the shape of the larvae can change with their movement. Finally, depending on the species and on the equipment used (in our case, depending on the bowl used), certain segments of the larvae may not be clearly distinguishable from the background. Once again, this is intensified in a real-world scenario, since elements such as dirt come into play.

In order to further validate the benefits of using the proposed method, future research could focus on two evaluations. First, it will be important to compare the results of the proposed method with those of other methods, such as direct manual counting (which should also have an associated error). For this end, measurements such as the Bland-Altmann agreement can be used. Second, further data should be collected to form an independent set of images to be used for assessing the performance of the method in a more precise way and with a

Model	MAE	MAPE	RMSE	R^2
yolov8m	5.44±6.81 a	4.71±4.98 a	**8.63±11.96**	**0.983**
rtdetr-x	**5.41±7.52 a**	**4.46±4.70 a**	9.17±13.87	0.981
yolov8n	7.31±10.14 a	6.00±6.53 a	12.37±18.73	0.966
yolov8x	7.81±9.95 a	6.66±6.96 a	12.53±17.87	0.965
rtdetr-l	8.31±9.33 ab	7.45±6.14 ab	12.39±18.12	0.966
yolov8s	8.44±12.12 ab	7.19±8.64 ab	14.61±21.91	0.953
yolov8l	9.72±12.06 ab	7.67±8.26 ab	15.34±21.49	0.948
deformable-detr	14.47±17.10 ab	13.85±12.48 ab	22.19±32.44	0.891
detr-resnet-50	18.47±25.79 b	18.74±38.86 b	31.39±47.74	0.781

Table 3. Average MAE, MAPE, RMSE and R^2 results in the test set, along with their respective standard deviations, for each neural network evaluated in this work. The results are ordered smallest sum of normalized means, according to the procedure described in Section 3.4. TukeyHSD results can be seen in compact letter display format within MAE and MAPE columns.

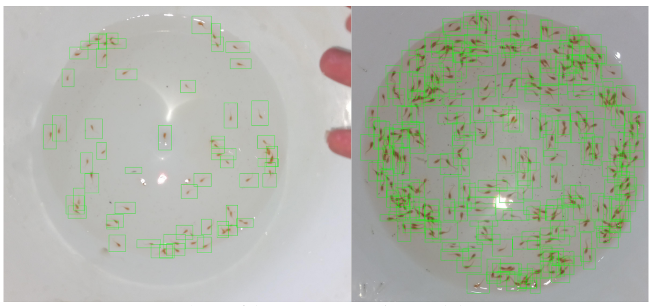

Figure 7. Examples of larvae counting with YOLOv8n. In these images, the percentage errors achieved were 5% and 2.41%.

Figure 8. Larvae detection by YOLOv8n with presence of dirt and other elements that can be misclassified as larvae, in virtue of their color and shape. In this example, the percentage error of the model was 17.31%, which can be considered high, when compared with the MAPE found in this work (see Table 3).

more robust methodology, including external validation.

Finally, one should notice that there remain possible difficulties and limitations regarding the usage of the proposed method in the context of business fish farms. An important issue is that, since the model is trained for specific species, it will require a new data collection, training and evaluation process in order to count larvae of other fish species. Another problem is that it demands a high computational power, and it has not yet been validated for use with portable devices, such as smartphones, thus relying on the connectivity infrastructure of the farm, which may be an issue in some rural areas.

5. Conclusion

In this work, we performed a comparative analysis of different neural network architectures for the task of detecting and counting fish larvae. Using a set of annotated images, we explored the performance of Transformer-based models such as RT-DETR, DETR-ResNet-50 and Deformable-DETR, as well as variants of the YOLOv8 architecture. For the evaluation, we created a new annotated image dataset that imposes less data collection requirements, whose images can be considered more representative of real-life conditions than other existing datasets for similar purposes. We achieved MAPE 4.46 (±4.70) with an extra large RT-DETR, and 4.71 (±4.98) with a medium sized YOLOv8. Further work can focus on increasing the dataset, as well as in evaluating different models and approaches for the task of larvae counting. In order to provide a larvae counting system for aquaculture, it may also be beneficial to develop a procedure for counting fish larvae without the need for sampling techniques.

Acknowledgements

We dedicate this work to the memory of Professor Dr. Hemerson Pistori, whose example, guidance, and dedication to science have profoundly shaped our academic journeys. His teachings and mentorship continue to inspire us to pursue excellence and innovation, and his legacy lives on in the work of those he guided and inspired.

This work has received financial support from the Dom Bosco Catholic University and from the Foundation for the Support and Development of Education, Science and Technology from the State of Mato Grosso do Sul, FUNDECT. Some of the authors have been awarded with Scholarships from the the Brazilian National Council of Technological and Scientific Development, CNPq and the Coordination for the Improvement of Higher Education Personnel, CAPES. We are especially grateful to the Projeto Pacu Ltda., for allowing and helping with data collection.

Author Contributions

Daniel Ortega de Carvalho: methodology, software, formal analysis, data curation, writing, visualization; Luiz Felipe Teodoro Monteiro: methodology, software, formal analysis, data curation, writing, visualization; Fernanda Marques Bazilio: methodology, data curation; Gabriel Toshio Hirokawa Higa: methodology, formal analysis, writing; Hemerson Pistori: conceptualization, methodology, resources, writing, supervision, project administration, funding acquisition.

References

[1] BARBEDO, J. G. A. A review on the use of computer vision and artificial intelligence for fish recognition, monitoring, and management. *Fishes*, MDPI, v. 7, n. 6, p. 335, 2022.

[2] ZHAO, S. et al. Application of machine learning in intelligent fish aquaculture: A review. *Aquaculture*, Elsevier, v. 540, p. 736724, 2021.

[3] GONÇALVES, D. N. et al. Using a convolutional neural network for fingerling counting: A multi-task learning approach. *Aquaculture*, Elsevier, v. 557, p. 738334, 2022.

[4] COSTA, C. S. et al. Counting tilapia larvae using images captured by smartphones. *Smart Agricultural Technology*, Elsevier, v. 4, p. 100160, 2023.

[5] KHAI, T. H. et al. *Underwater Fish Detection and Counting Using Mask Regional Convolutional Neural Network. Water 2022, 14, 222*. 2022.

[6] YU, X. et al. Counting method for cultured fishes based on multi-modules and attention mechanism. *Aquacultural Engineering*, Elsevier, v. 96, p. 102215, 2022.

[7] LI, W. et al. A lightweight network for portable fry counting devices. *Applied Soft Computing*, Elsevier, p. 110140, 2023.

[8] KAKEHI, S. et al. Identification and counting of pacific oyster crassostrea gigas larvae by object detection using deep learning. *Aquacultural Engineering*, v. 95, p. 102197, 2021. ISSN 0144-8609.

[9] HU, W.-C. et al. A deep-learning-based fast counting methodology using density estimation for counting shrimp larvae. *IEEE Sensors Journal*, v. 23, n. 1, p. 527–535, 2023.

[10] LIU, D. et al. Shrimpseed_net: Counting of shrimp seed using deep learning on smartphones for aquaculture. *IEEE Access*, v. 11, p. 85441–85450, 2023.

[11] ROTHSCHILD, C. et al. Computer vision system for counting crustacean larvae by detection. *Smart Agricultural Technology*, v. 5, p. 100289, 2023. ISSN 2772-3755.

[12] FERNANDES, M. P. et al. Convolutional neural networks in the inspection of serrasalmids (characiformes) fingerlings. *Animals*, v. 14, n. 4, 2024. ISSN 2076-2615.

[13] BABU, K. M. et al. Computer vision in aquaculture: a case study of juvenile fish counting. *Journal of the Royal Society of New Zealand*, Taylor & Francis, v. 53, n. 1, p. 52–68, 2023.

[14] COSTA, C. S. et al. Deep learning applied in fish reproduction for counting larvae in images captured by smartphone. *Aquacultural Engineering*, v. 97, p. 102225, 2022. ISSN 0144-8609.

[15] JOCHER, G.; CHAURASIA, A.; QIU, J. Yolo by ultralytics. *URL: https://github. com/ultralytics/ultralytics*, 2023.

[16] CARION, N. et al. End-to-end object detection with transformers. In: SPRINGER. *European conference on computer vision*. [S.l.], 2020.

[17] VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017.

[18] DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[19] ZHU, X. et al. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[20] LV, W. et al. *DETRs Beat YOLOs on Real-time Object Detection*. 2023.