

Image Segmentation Applied to Multi-species Phenotyping in Fish Farming

Fabrício Martins Batista(✉) and José Remo F. Brega

Sao Paulo State University (UNESP), Bauru, Brazil
{fabricio.batista,remo.brega}@unesp.br
https://sgcd.fc.unesp.br/#!/liv

Abstract. Fish farming has been gaining prominence in recent years, almost doubling world production in a ten-year period. Fish farming products are an important source of protein in coastal or insular countries, as they are the most abundant natural resource in these regions and are part of the daily diet of their populations. Recent technological advances allow the evolution of the practice of fish farming through new tools such as Artificial Intelligence and Internet of Things. Among the tasks that stand out is the phenotyping of animals raised in captivity during various stages of growth to assess the interaction of the species with the environment or even the prevalence of certain characteristics after successive reproductive selections. Phenotyping is usually performed manually through a digital image taken by an expert and post-processed in specialized measurement software using an object of known size as a reference in the image. Based on this problem, this work proposes a Computer Vision System to automate the phenotyping of two common species in Brazilian fish farming: *Piaractus Mesopotamicus (Pacu)* and *Colossoma macropomum (Tambaqui)*. The results indicate a positive correlation between the measurements performed by humans and the proposed Computer Vision system, presenting itself as a viable alternative to accelerate the process of collecting information for reproductive selection in commercial fish farming.

Keywords: Precision Agriculture · Artificial Intelligence · Genetic Selection · Image Segmentation · Computer Vision

1 Introduction

World aquaculture production grew at an average rate of 5.3% per year in the period from 2001 to 2018, amounting to 82.1 million tonnes (US$ 250.1 billion) of aquatic animals in 2018. In Latin America, aquaculture significantly advanced in the global ranking [1].

The main group of native fish species produced in Brazil is composed of Serrasalmidae species (tambaqui *Colossoma macropomum*, pacu *Piaractus mesopotamicus*, pirapitinga *Piaractus brachypomys*, and their interspecific hybrids),

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14813, pp. 96–111, 2024.
https://doi.org/10.1007/978-3-031-64605-8_7

corresponding to about 30% of the national production, estimated at 529.6 thousand tonnes in 2019 [2].

In addition to the applications of Internet of Things (IoT), Computer Vision, Image Processing, and Artificial Intelligence are also highlighted, with various applications such as disease detection [3], species classification [4], quality estimation [5], counting, and phenotyping (extraction of physical traits of animals) [6].

For the present work, data from the Pacu species (scientifically called *Piaractus mesopotamicus*) and Tambaqui (scientifically called *Colossoma macropomum*) were used, characterized by being a freshwater fish and belonging to the same family as the piranhas.

A partnership was made with the Laboratory of Genetics in Aquaculture and Conservation (LaGeAC) at Unesp Jaboticabal to initially carry out experiments related to data collection about the animals. Thus, for this study, imagetic data from the fish were used to build the automatic measurement system.

2 Theoretical Foundation

2.1 Phenotyping

Phenotyping can be defined as the process of extracting specific measurements from certain animals, such as weighing or measuring specific regions of interest like head size, total length, or height. These data are used by researchers or producers to assess the species' interaction with the environment or the prevalence of certain traits after successive reproductive selections [7].

This type of process is manual and quite costly, subject to biases from the researchers responsible for performing the measurements. As the number of animals can be quite large, carrying out this type of process manually becomes unfeasible in commercial applications.

2.2 Reproductive Selection

Reproductive selection, in turn, aims to identify and select individuals that have desirable traits such as disease resistance or total weight achieved after maturation. Through this process, it is possible to achieve genetic refinement with the individuals [8].

Reproductive selection for genetic improvement essentially focuses on choosing individuals that reach higher weights or have a high daily weight gain. However, in the context of commercial fish farming in Brazil, genetic improvement is not yet widely practiced due to the difficulties and intensive manual work required for carrying out such programs on a large scale. To solve this problem, it is necessary to find ways to eliminate some of the manual work involved in the process.

This work presents a method to achieve this goal and collect granular information on individuals of the species for incorporation into genetic improvement

programs for *Piaractus Mesopotamicus (Pacu)* and *Colossoma macropomum (Tambaqui)*.

Among the measurements obtained by the constructed system are total height (th), total length (tl), standard length (sl), pelvic width (pl), head height (hh), head width (hl), as well as the area in pixels of each region obtained according to Fig. 1.

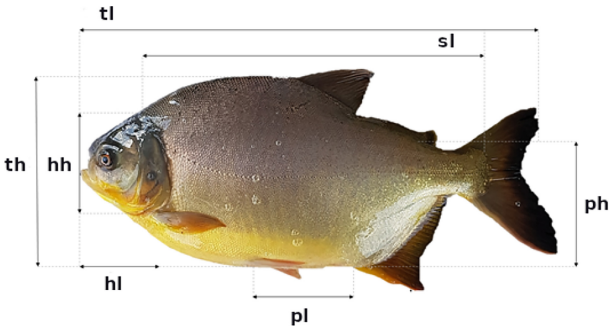

Fig. 1. Measurements to be obtained by the proposed system in this work.

2.3 Convolutional Neural Networks and the R-CNN Family of Methods

Convolutional Neural Networks (CNNs) present an important technique for object detection. Before their re-introduction in the early 2010's the dominant approaches consisted of two main techniques: HOG [9] and SIFT [10]. The techniques used a grid of gradients to highlight edges to act as feature extractors and then were combined with classifiers such as SVMs [11]. After the success of CNNs with AlexNet in 2012 scaling up the LeNet architecture [12] by using a GPGPU parallelized implementation of CNNs, a promising idea would be to apply the feature extraction capabilities of CNNs to the object detection problem. R-CNN [13] was one of the first movements in this direction.

2.4 R-CNN

R-CNN is a multi-step prediction method: selection, extraction, and classification. The name R-CNN comes from the division of images into regions, from the English "Region-based Convolutional Neural Networks".

The image division occurs through the selective search technique [14]. In the first step of the algorithm, about 2000 regions are extracted from the image.

Each of these regions is then resized to be compatible with the input dimensions of pre-trained CNNs on ImageNet. Specifically, the authors conducted two experiments, one using AlexNet [15] and VGG [16] as feature extractors.

After the feature extraction step using CNNs, the obtained feature maps were fed to multiple linear SVMs using the One-vs-All strategy, i.e., one linear SVM for each class of interest in the problem.

The detection of the object bounding box is modeled as a regression problem from the feature maps extracted from the initial regions. From the description of the architecture, it is not difficult to see that these are operations with extremely high computational cost. The architecture of the R-CNN is depicted in Fig. 2.

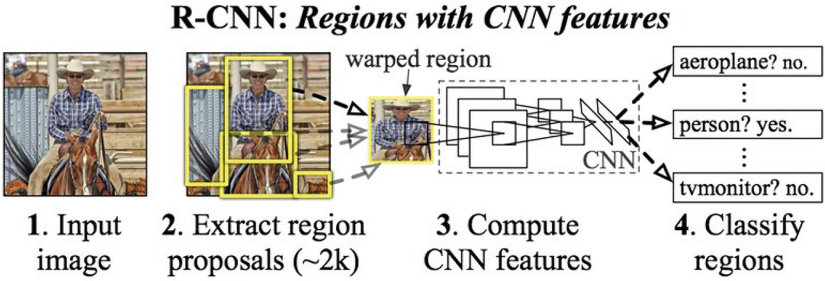

Fig. 2. R-CNN Architecture

After all, it is necessary to apply the convolution at least 2000 times, one for each region extracted by the CNN, without considering the cost of the SVM also applied to each of the regions.

In addition to that, a large part of the regions extracted in the first step overlap, which leads to redundant calculations.

Despite this, object detection observed a leap in performance with this approach. The issue of computational cost would be addressed in subsequent works.

2.5 Fast R-CNN

The main bottleneck of R-CNN was the amount of redundant calculations. To avoid this problem and speed up R-CNNs, SPPNet [17] performed the CNN application only once for the entire image and then performed region selection operations on the feature map, resulting in sub-regions of interest.

This way, it was possible to avoid redundant calculations. However, the approach of SPPNet, like R-CNN, was a multi-step prediction system: selection, extraction, and classification using SVMs.

This forces the freezing of the CNN, which only acts as a feature extractor, leading to lower performance in case of fine-tuning needs for problems that diverge greatly from the original data of the ImageNet dataset.

Based on these issues, the new iteration of R-CNN, Fast R-CNN [18], proposes an architecture with a single step, allowing end-to-end training of the model. Fast R-CNN introduced another element, the RoI (Region of Interest) Pooling layer. The general architecture of the Fast R-CNN is depicted in Fig. 3.

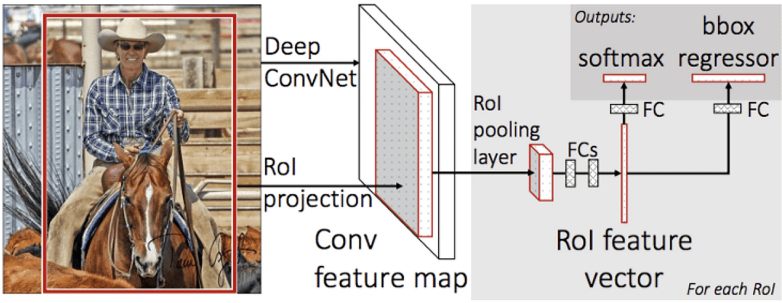

Fig. 3. Fast R-CNN Architecture

This layer is responsible for extracting a fixed-length vector from a region of the feature map, which is then fed to fully connected layers, generating a feature vector that is then sent to two prediction heads, a regressor for the bounding boxes and a softmax classifier. Thus, the training of the architecture now becomes feasible for end-to-end execution, allowing fine-tuning even of the early layers of the feature extractor. The cost function of this architecture is given by the sum of the cost functions of each distinct objective function for the regressor and classifier, thus configuring a multi-objective cost function for the architecture.

2.6 Faster R-CNN

Despite the advances achieved by Fast R-CNN, the operation of region selection through the Selective Search technique was still used. The implementations of this technique are generally based on CPUs, and therefore do not take advantage of the parallel computing power offered by modern GPUs. The average time spent on calculating regions using Selective Search is around 2 s per image, accounting for most of the time spent in the algorithm's application.

The authors of this new iteration of R-CNN, Faster R-CNN [19], argue that it would be possible to implement the Selective Search operations on GPUs. However, this would not properly leverage the improvements in shared calculations made previously. Instead, the authors proposed a new way of selecting important regions in the image through a Region Proposal Network (RPN).

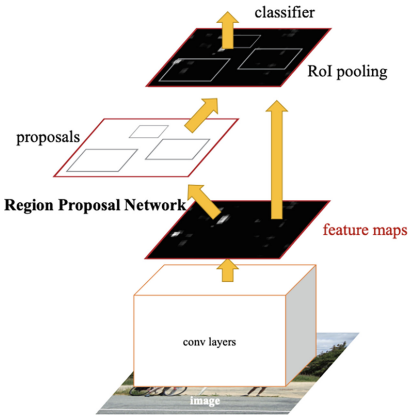

Fig. 4. Region Proposal Network

RPN is a Convolutional Neural Network that takes an image as input, calculates feature maps using the trunk of a pre-trained CNN on the ImageNet dataset, and outputs rectangular regions associated with an "objectness" score. This score represents the probability that the extracted rectangular region contains an object of interest or only the background of an object. The general architecture of an RPN is depicted in Fig. 4.

In this way, it is possible to take advantage of the feature maps that are extracted only once and reuse them for multiple purposes. RPNs produce multiple regions of interest, often redundant and overlapping regions.

To avoid this problem, the Non-Max Suppression technique is used to merge subregions into a single contiguous region when there is a significant overlap rate (greater than 0.7) calculated from the Intersection over Union (IoU) metric.

The calculation of the cost function for training the architecture is the sum of the 4 cost functions for each of the objectives: the regressors of the RPN's regions of interest, the objectness classifier of the region, the bounding box regressor of the Fast R-CNN, and the classifier of the object contained in the identified bounding box.

2.7 Mask R-CNN

Besides object detection and classification, another relevant problem in image processing is semantic segmentation. Semantic segmentation consists of classifying pixels into contiguous regions belonging to an object.

Mask R-CNN [20] extends the Faster R-CNN architecture by adding a new prediction head responsible for generating a mask for each detected object in parallel with the class and bounding box prediction heads.

This mask prediction head adds little computational cost to the Faster R-CNN, while maintaining the performance gain obtained in previous iterations of the architecture.

In addition to the prediction head, Mask R-CNN also introduced an intermediate layer called ROI Align. The idea of this intermediate layer is to align the spatial regions of the original image with the corresponding regions in the feature map.

As discussed earlier, extracting feature maps through CNNs reduces the spatial dimensions such as height and width of the image and increases the depth of the representation.

To align the region of the original image with a feature map with potentially smaller height and width, the authors used bilinear interpolation. Bilinear interpolation is an operation used in computer vision for image resizing.

The objective function optimized for the additional head of the architecture is a binary problem modeled through a sigmoid function. Thus, the output of this prediction head is a matrix with the spatial dimensions of the original image, and for each pixel, a sigmoid-applied output value between 0 and 1 is obtained.

If the value is greater than 0.5, the pixel is considered in the mask. For each detected object instance in the image, a binary mask is generated in the process.

Mask R-CNN has also shown relevance in another problem: pose estimation. Pose estimation involves classifying the position of a person's joints and identifying the position of each of the 14 key points in the human body.

The success of Mask R-CNN in these two tasks has established the R-CNN family as extremely important methods in the field of computer vision.

3 Development Methodology

The dataset used in this work consists of approximately 1802 animals of the Pacu species and 365 animals of the Tambaqui species. The number of animals is approximate because during handling, some animals end up dying in the process, and the number presented refers to the first collection made. Four regions of interest were annotated in the images: the head, the body, the fins, and the pelvic region.

Due to morphological differences at different growth stages, the images of both species were captured at two different moments, at 15 months and 28 months, in order to make the model invariant to the animals' development stage.

To extract the images, the animals were submerged in a benzocaine solution (0.1 mg/l), identified through an RFID tag, weighed, and placed on a polystyrene base with a ruler for later verification of the model results. After collection, for validation of the measurement system, the images from an independent set not used during training were individually calibrated. Calibration consists of mapping a distance of 1 cm on the ruler in the image to a pixel value using the ImageJ software.

The annotation process was carried out using the Labelbox software by a single annotator and reviewed by a second expert. Incorrectly annotated images were invalidated and corrected until all were correctly annotated.

Four regions of interest were annotated in the images: the head, the body, the fins, and the pelvic region, as shown in Fig. 5. The measurements are initially



Fig. 5. Mask annotated with regions of interest, left: *Colossoma macropomum*, right: *Piaractus mesopotamicus*

extracted in pixels and then mapped to a measurement in millimeters through scaling. A calibration process in pixels was manually performed on each image of the validation dataset to facilitate the validation of the measurement extraction. A ruler was positioned on a styrofoam platform for this calibration.

Specifically, the implementation of Mask R-CNN was adjusted to adhere to the field constraints related to computational cost. To minimize stress on the fish during the measurement process, the trained model needs to be compact and fast enough to run on an edge device as close as possible to the fish tanks.

4 Implementation

The original work of Mask R-CNN employs a 50-layer Convolutional Neural Network called ResNet-50 as the feature extractor.

However, a variant of the ResNet architecture with only 18 layers was used to reduce the computational cost. This effectively reduced the number of floating-point operations (FLOPs), required parameters, and training time to achieve similar results.

During training, data augmentation techniques were applied to the images. The images were vertically and horizontally flipped, rotated by 90° or 270°, and resized at scales of 0.10, 0.25, 0.5, and 0.75 with a probability of 0.5.

These transformations help make the model more tolerant to linear transformations of the input images.

The dataset was randomly divided, stratified to preserve the proportions of different species, into three subsets: training (80%), validation (20%), and 5% of the training set was used as the development set.

The development set was used to tune hyperparameters during experimentation, avoiding the direct use of the validation set and preventing potential overfitting.

The experiment was conducted using an NVIDIA RTX 2060 GPU with 8 GB of RAM, and an Intel i5 9600f CPU with 6 physical cores. The experiment ran for 350 epochs and took approximately 22 h.

The optimized metric was Intersection over Union (IoU). At the end of each epoch, the IoU was calculated for the development set. During the experiments, training was interrupted several times to optimize the learning rate, which was the hyperparameter with the greatest sensitivity in the model, using the performance on the development set as the decision parameter.

IoU was calculated using the area obtained by the model A and the expected (annotated) area B with the formula $A \cap B / A \cup B$. A IoU value of 1.0 indicates that the annotated area exactly overlaps with the area obtained as the output by the model.

Based on the outputs obtained by Mask R-CNN, the bounding box was used to calculate the morphometric measurements of the fish in pixels, and the segmented masks were used to calculate the area in pixels of the regions of interest.

For evaluating the measurement results obtained by the model, two metrics were assessed: Pearson Correlation and Mean Squared Error (MSE) between the measurements obtained by the model and the manual measurements performed by an expert.

5 Results

5.1 Model Training

The model was trained for 350 epochs, with a learning rate fixed at 10^{-5}. Table 1 shows the values of IoU for Precision and Recall on the test set calculated for the bounding boxes.

The indices are divided into intervals and different detection scales. IoU indices with a value of -1 in the table indicate that there were no object detections in the respective scale.

Table 2 shows the values of IoU for Precision and Recall on the test set calculated for segmentation.

The loss curves of the model are presented in Fig. 6. The total loss is given by the sum of all cost functions $L = L_{cls} + L_{box} + L_{mask} + L_{objectness}$, which corresponds to the multi-objective function optimized by Mask R-CNN. The optimization had significant variations until epoch 150. After this point, the model was further optimized for 200 epochs because the IoU metrics continued to improve with the optimization focused on finer details of the images.

Table 1. IoU Table for Bounding Boxes

Metric	IoU/Size	Value
Average Precision (AP)	0.50:0.95/all	0.941
Average Precision (AP)	0.50/all	0.995
Average Precision (AP)	0.75/all	0.995
Average Precision (AP)	0.50:0.95/small	−1.000
Average Precision (AP)	0.50:0.95/medium	0.869
Average Precision (AP)	0.50:0.95/large	0.953
Average Recall (AR)	0.50:0.95/all	0.960
Average Recall (AR)	0.50:0.95/small	−1.000
Average Recall (AR)	0.50:0.95/medium	0.906
Average Recall (AR)	0.50:0.95/large	0.964

Table 2. IoU Table for Segmentation

Metric	IoU/Size	Value
Average Precision (AP)	0.50:0.95/all	0.899
Average Precision (AP)	0.50/all	0.995
Average Precision (AP)	0.75/all	0.995
Average Precision (AP)	0.50:0.95/small	−1.000
Average Precision (AP)	0.50:0.95/medium	0.879
Average Precision (AP)	0.50:0.95/large	0.921
Average Recall (AR)	0.50:0.95/all	0.919
Average Recall (AR)	0.50:0.95/small	−1.000
Average Recall (AR)	0.50:0.95/medium	0.909
Average Recall (AR)	0.50:0.95/large	0.926

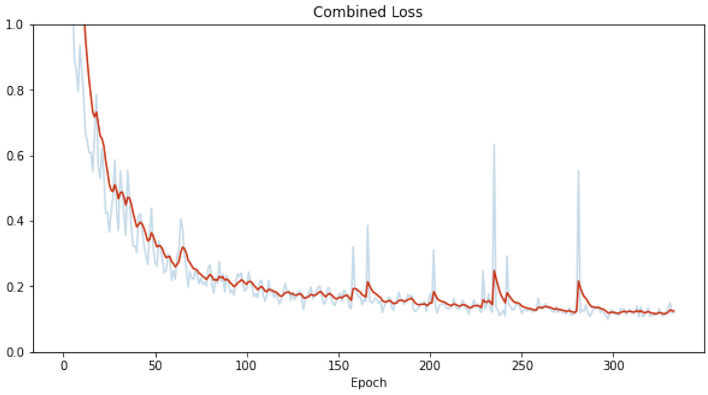

Fig. 6. Multi-objective Cost Function Loss of Mask R-CNN

5.2 Phenotyping *Piaractus mesopotamicus*

Due to the image collection process being carried out by different teams and at different stages of animal development, in order to ensure the quality of the results reported in this work, the images in the validation set used to evaluate the model were individually calibrated.

However, the calibration showed low variability in this set, where 1 cm varied from 46.68 to 51.67 pixels (mean of 49.49 pixels). The validation set of measurements for Pacu contained 1031 animals, of which only 979 had valid observed values.

Another important point to highlight is that the images were captured by two devices with different resolutions. Segmentation anomalies were detected in only 9 images of the set, which were excluded from the visualizations and metrics presented below. These images had anomalous values above the upper limit of $\mu + 3 * \sigma$ and below the lower limit $\mu - 3 * \sigma$ of the automatically obtained measurements.

For the measurement in this particular test set, the pelvic measurement was disregarded as the data for this metric had not yet been standardized. Therefore, the measurements of pelvic length (pl) and pelvic height (ph) were not considered in this particular analysis.

Figure 7 presents a scatter plot where the X-axis represents the value estimated by the system and the Y-axis represents the value annotated by a human.

The other measurements are standardized according to Fig. 1. Table 3 presents the metrics obtained from the comparison of automatic measurement with the one observed by human annotators.

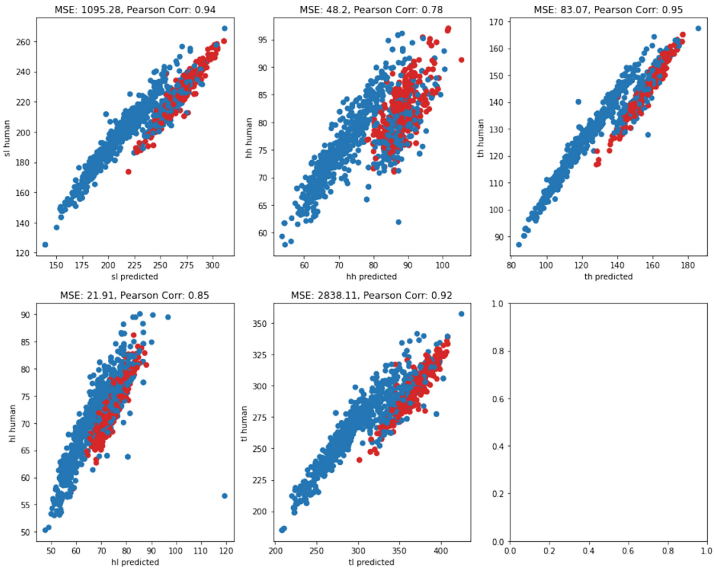

Fig. 7. Scatter Plot X = estimated, Y = observed

The color differentiation in Fig. 7 is due to the fact that the images were captured at different times by devices with different resolutions. Figure 8 shows box plots comparing the values obtained through automatic measurement and human measurement.

Table 3. MSE and Pearson's Coefficient Table

Measure	MSE (mm)	Pearson's Coefficient
tl (Total Length)	2720.83	0.92
th (Total Height)	81.27	0.95
hh (Head Height)	46.47	0.78
hl (Head Length)	22.87	0.85
sl (Standard Length)	1054.11	0.94

Since the calibration was performed individually and the values converted to the same scale, it is possible to observe them on the same graph. However, there is a small deviation in relation to the two groups, also observed in the following graph.

Despite the high correlation (greater than 0.70) in all measurements, the mean squared error for the length measurements tl and sl draws attention.

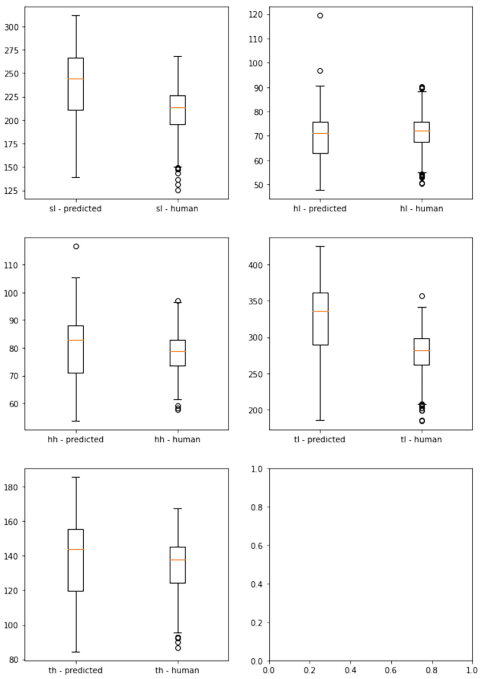

Fig. 8. Box-Whisker Plot - Observed x Estimated

Through Fig. 8, it is possible to observe that there is greater variability as well as a higher number of outliers in the two length measurements, tl and sl, manually collected by human annotators.

This phenomenon was attributed to differences in the manual measurement methodology. Therefore, the benefits obtained from the application of an automatic measurement system go beyond simply automating the process.

The measurement system eliminates the differences in methodology between different human annotators, bringing not only automation but also standardization of the results and greater reliability of the information produced in this type of operation.

5.3 *Colossoma macropomum* Phenotyping

Figure 9 presents a scatter plot where the X-axis represents the value estimated by the system and the Y-axis represents the value annotated by a human. The capture device used was standardized for a 12-megapixel digital camera integrated into the Motorola One Action smartphone, configured at maximum resolution and autofocus. During the collection of this sample, the distance between the camera and the styrofoam platform was standardized at 80 cm. Following the previous experiment, the other measurements are standardized according to Fig. 1. In order to ensure the quality of the results reported in this work, the images in the validation set were individually calibrated.

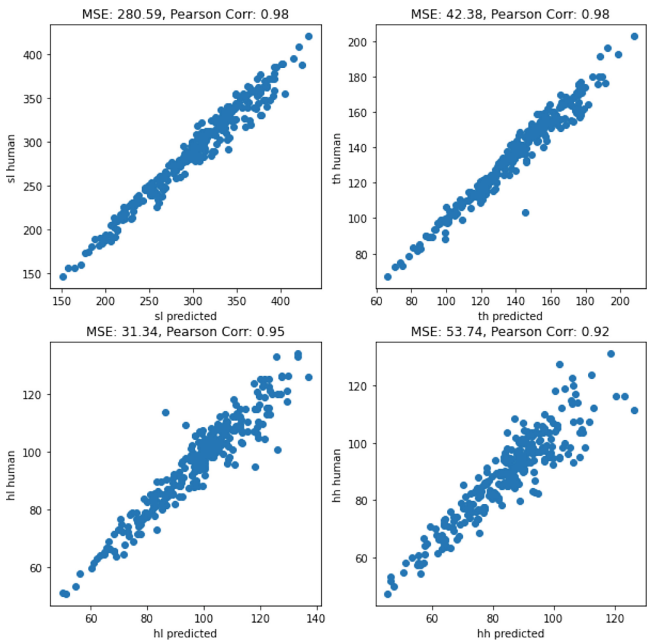

Fig. 9. Scatter Plot X = estimated, Y = observed

Table 4. MSE and Pearson's Coefficient Table

Measure	MSE (mm)	Pearson's Coefficient
th (Total Height)	42.38	0.98
sl (Standard Length)	280.59	0.98
hh (Head Height)	53.74	0.92
hl (Head Length)	31.34	0.95

Table 4 presents the metrics obtained from the comparison of automatic measurement with the one observed by human annotators.

Figure 10 compares the measurements made by a human with those estimated by the algorithm using a box-whisker plot. The calibration also showed low variability in this set, where 1 cm ranged from 45.5 to 51.0 pixels (average of 48.1 pixels). The validation set for Tambaqui measurements contained 303 animals, of which only 273 had valid observed values.

With the standardization of the collection methodology in this second experiment, it was not possible to observe the existence of outliers.

For the measurement in this particular test set, the pelvis measurement and total length were disregarded because the data for these metrics were not collected by human observers. Therefore, the measurements of pelvis length (pl),

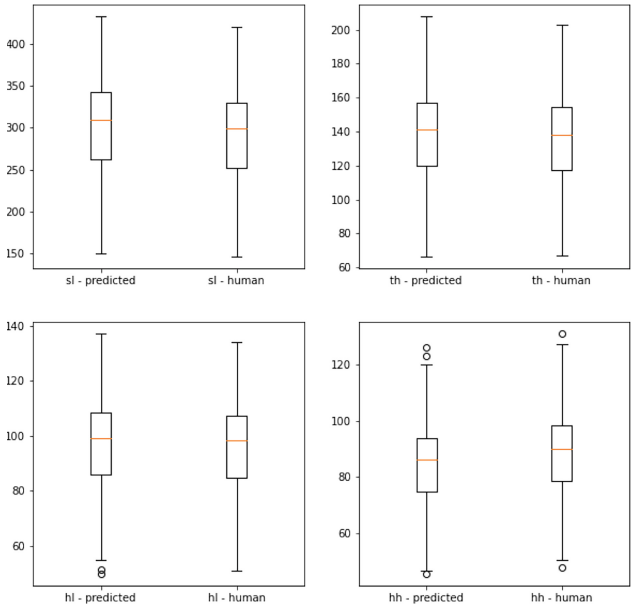

Fig. 10. Box-Whisker Plot - Observed x Estimated

pelvis height (ph), and total length (tl) were not considered in this particular analysis.

However, it is important to consider that in the experiment conducted with *Piaractus mesopotamicus*, the evaluated set is almost four times larger. The scarcity of outliers in this set can also be attributed to this fact.

6 Conclusion

Computer vision has shown great potential for automation of tasks in the context of Precision Fish Farming.

The application of Mask R-CNN in the context of live animal phenotyping, under natural lighting conditions, allows for a lower stress load on the animals in question, as it can be performed closer to their natural habitat, reducing mortality during the handling of such operations.

In addition to reducing the required time, the algorithm is able to standardize the operation, avoiding the variability of measurement methodologies and enabling large-scale phenotyping for genetic improvement programs in fish farming, both in commercial and research aspects.

Furthermore, the created measurement system can be used for the measurement of more than one species simultaneously, as demonstrated by the presented results.

For future work, it is intended to evaluate the possibility of using the obtained pixel area metrics for estimating other values of interest through regressions. Some values can only be obtained by sacrificing the animals, so it is interesting to evaluate the quality of the obtained estimates, avoiding the sacrifice of potential breeders who can generate increasingly genetically refined individuals with high commercial value.

References

1. FAO. The state of world fisheries and aquaculture 2020. Sustainability in action Rome (2020)
2. IBGE (2020). https://cidades.ibge.gov.br/brasil/pesquisa/18/0
3. Malik, S., Kumar, T., Sahoo, A.K.: Image processing techniques for identification of fish disease. In: 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), pp. 55–59 (2017)
4. Cao, Y., Lei, Q., Wei, T., Zhong, H.: A computer vision program that identifies and classifies fish species. In: 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), pp. 384–390 (2021)
5. Taheri-Garavand, A., Fatahi, S., Omid, M., Makino, Y.: Meat quality evaluation based on computer vision technique: a review. Meat Sci. **156**, 183–195 (2019)
6. Wang, G., Hwang, J.-N., Wallace, F., Rose, C.: Multi-scale fish segmentation refinement and missing shape recovery. IEEE Access **7**, 52836–52845 (2019)
7. Blay, C., et al.: Genetic parameters and genome-wide association studies of quality traits characterised using imaging technologies in Rainbow trout, Oncorhynchus mykiss. Front. Genetics **12** (2021)

8. Lhorente, J.P., Araneda, M., Neira, R., Yáñez, J.M.: Advances in genetic improvement for salmon and trout aquaculture: the Chilean situation and prospects. Rev. Aquac. **11**(2), 340–353 (2019)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893 (2005)
10. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
11. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
12. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
13. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, vol. abs/1311.2524 (2013)
14. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS 2012, (Red Hook, NY, USA), pp. 1097–1105. Curran Associates Inc. (2012)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
17. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR, vol. abs/1406.4729 (2014)
18. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015)
19. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR, vol. abs/1506.01497 (2015)
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN (2017). arxiv:1703.06870Comment. Open source; appendix on more results