Detection of Key Points for a Rainbow Trout in Underwater Video Surveillance System

Maksim Pavlov, Aleksey Marakhtanov, Dmitry Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
mpavlov@cs.petrsu.ru, marahtanov@yandex.ru, dkorzun@cs.karelia.ru

***Abstract*—Video Surveillance Systems (VSS) are progressing towards the industry digitalization. In this demo, we apply the Artificial Intelligence (AI) technology to detection of key points for a rainbow trout. The well-known YOLO-Pose neural network is typically used for human pose recognition in video recordings. We use this neural network for detection of 8 key point for an individual fish. Our constructed dataset includes 10 500 images for approximately 30 thousand fishes. The YOLO-Pose network is trained on the dataset using our combined loss function. Our early experiment shows that that detection works fine even in muddy underwater conditions.**

I. INTRODUCTION

Rainbow trout belongs the salmon family. The fish is actively farmed in natural environment and growth in Recirculating Aquaculture System (RAS) fisheries. Recently, the Machine Learning (ML) technology has been increasingly used in fish farm industry [1]. Development of an aquaculture underwater Video Surveillance System (VSS) requires specific information about an individual rainbow trout: their location in the frame and a set of particular key points for further analysis and monitoring. In this work-in-progress paper, we consider the experience of applying these methods to detection of eight key points of rainbow trout.

When developing a VSS, the detection of fish key points must take into account that a live object in the frame is always moving. The flexibility of its body as it moves can be a problem for accurate detection. The main contribution of our research is the experimental proof that such an underwater VSS can be developed using well-known ML-based recognition algorithms and our proposed method. Implemented by solutions' results in video data can achieve results comparable with their human equivalents.

The rest of the aper is organized as follows. Section II introduces the problem of object detection based on key points. Section III describes our proposed method for key points detection based on the well-known Yolo-Pose neural network. Section IV provides our experimental study of the proposed method. Section V summarizes the key findings of this study.

II. OBJECT DETECTION BASED ON KEY POINTS

The task of detecting objects and recognition a certain set of points on them is not new for computer vision. There is a whole set of specific methods and strategies. The most common at the moment is Deeppose [5]. It is Human pose estimation based on deep learning. Such methods can usually be divided into two main ideas:

- Top-down. It localizes objects in the image with subsequent determination of their key points. This approach is characterized by a high accuracy but with a heavy time-consuming. The complexity increases linearly with the increase in the number of objects in the image.
- Bottom-up. It definition of key points of all objects in the image with subsequent grouping of points by objects. This approach is characterized by a less accuracy but more faster and with difficult post-processing and grouping predicted points by objects.

Among the currently available solutions, we have chosen the YOLO-Pose neural network based on YOLOv5. It is the first focused attempt to solve the problem of two-dimensional pose estimation without a probability map and with a focus on getting rid of the various non-standardized post-processing currently used [6]. One could say that it has brought together the advantages of the bottom-up and top-down methods. It is based on the top-down method, but without the complex post-processing. By linking anchor boxes and object points, there is no need for additional grouping of points by objects. It also uses standard Object Detection post-processing instead of complex post-processing, including Non Max Suppression at the pixel level and various clustering algorithms. It is making Yolo-Pose a great neural network for real-time Underwater Video Surveillance System. The approach itself is robust due to end-to-end learning without independent post-processing.

In selecting a neural network for developing the VSS, we would suggest to consider the following criteria:

- Transferability. Possibility to use a neural network to train on your own class of objects and key points.
- Ability to use your own dataset for training, converting annotations to the required format.
- The speed of the neural network, its size and number of parameters. The performance of the developed solution must be sufficient to work in real-time condition.
- Accuracy of the chosen solution. In many tasks, an accuracy error of even a couple of pixels can be critical. For example, it is important for estimate fish weight and size. Especially it is important to keep this in mind because people today tend to rely more and more on the neural networks. The developed VSS has to meet all user expectations.

- Having pretrained weights on a different dataset. Our practice and experience shows that it is better to start training with pretrained rather than with random weights. Preferably on another dataset with more object classes.

III. Detection Method for Rainbow Trout

Important to remember that the performance of the Underwater VSS also depends on the number of key points. During the solving this problem, we selected eight key points for rainbow trout to estimate their position in the frame (Fig. 1).

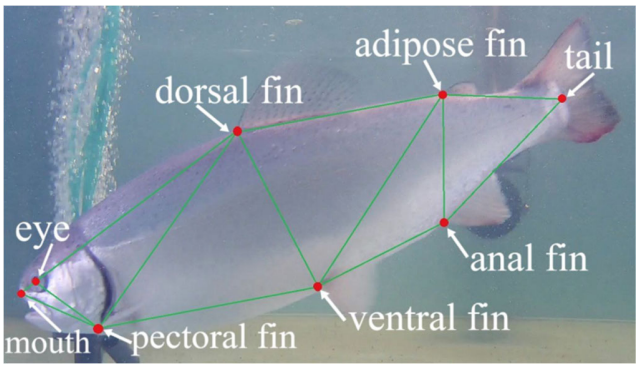

Fig. 1. Rainbow trout key points

Using the knowledge of an area with a fish and it is eight points, it is possible to predict and monitor an extensive class of life history parameters of an individual. Such parameters can include weight and size of fish [2, 3], speed and behavioral characteristics [4], detection of anomalies of growth, defects and diseases based the images of distinct body parts. This information will improve the efficiency and speed of selection and allow breeders to determine the better fish to maintain their broodstocks.

All the data were prepared and tested as part of hardware-software complex for monitoring of aquaculture tank life-cycle - "Fish Grow Platform". The constructed dataset contains 10500 images, 30597 rainbow trout individuals, with 19112 of them having key points.

For several reasons, the main loss function for training is Object Key points Similarly [6], because it allows the estimation of both visible and hidden object points:

$$\text{OKS} = \frac{\sum_{i\in[0,N-1]}\left(\exp\left(\frac{-d_i^2}{2s^2k_i^2}\right)\sigma(\vartheta_i>0)\right)}{\sum_{i\in[0,N-1]}\sigma(\vartheta_i>0)}, \text{ where}$$

d_i^2 - the Euclidean distance between the i-th predicted and target point;

ϑ_i - visibility flag of the i-th point in the annotation;

σ - boolean function that returns 1 if the content is true, and 0 if false;

s - the scale of the object, the square root of its area;

k_i – set of constants describing the distributed deviation in the annotation for the i-th point.

This function is exponential and is not emission-sensitive. Therefore, points predicted far away from the target objects cannot be pulled in. It is also quite difficult to adjust for your own class of objects. And a calculation of exact coefficient for each point (k_i) is difficult and time-consuming. We decided at the beginning, for 25% of the training time, to use another loss function emphasizing those very large errors - MSE:

$$\text{MSE} = \frac{\sum_{i\in[0,N-1]}(d_i^2\sigma(\vartheta_i>0))}{\sum_{i\in[0,N-1]}\sigma(\vartheta_i>0)}, \text{ where}$$

d_i^2 - the Euclidean distance between the i-th predicted and target point;

ϑ_i - visibility flag of the i-th point in the annotation;

σ - boolean function that returns 1 if the content is true, and 0 if false.

This is possible because these two functions are similar in their desire for the beginning of coordinates (Fig. 2). By this combining of two functions, we have been to eliminate outliers in based learning and to simplify the adjustment of the loss function based on OKS, while maintaining the level of accuracy of the predictions. To achieve sufficient accuracy, it is no longer necessary to precisely calculate the distributional deviation coefficients for each point (k_i), it is possible to use the approximations to the values obtained from empirical analysis.

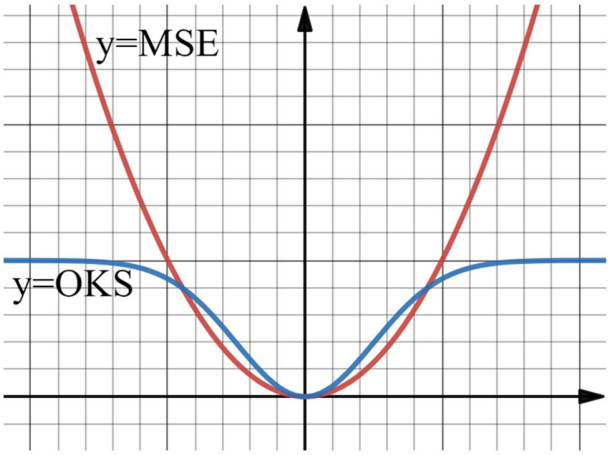

Fig 2. Graph of loss functions without scaling factors

We had to make the following changes in YOLO-Pose to use in our proposed method:

1) Reducing the number of tensors used for point prediction, as a result of moving from 17 human to 8 rainbow trout points;

2) Selecting the best model weights based on a point prediction improvement of 5%, mAP 0.5:0.95 with a contribution of 85% and mAP 0.5 at 10%.

3) Combination of two learning loss functions for training: MSE and OKS (Fig. 3).

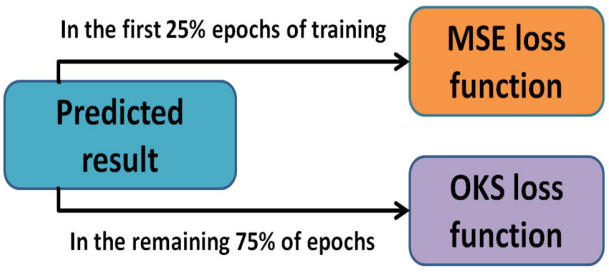

Fig 3. Schematic for choosing a loss function for training in our proposed method

IV. EXPERIMENTS

The selected metrics help to estimate the performance of the neural network in practical terms. In addition, we considered three further estimates of point prediction accuracy, in percentage terms, relative to the scale of the individual in the frame:

$$\text{MSPE kpts} = \frac{100}{n} \times \sum_{i\in[0,N-1]} \frac{d_i^2}{s}$$

$$\text{MAPE kpts} = \frac{100}{n} \times \sum_{i\in[0,N-1]} \frac{|d_i|}{s}, \text{ where}$$

d_i^2 - the Euclidean distance between the i-th predicted and target point;

$|d_i|$ - the difference module of the prediction vector and the target vector for the i-th point;

s - the scale of the object, the square root of its area.

For example, MSPE is the average radius of the circles around the all ideal target points, in which most of the predicted points hit.

As an experiment, we trained three YOLO-Pose neural network models (s6, m6, l6) on a collected dataset for 300 epochs with an input image of 640x640 pixels. The server for data processing has the following specification:

- Intel Core i9-9900K
- Nvidia RTX 2060
- 32 GB RAM

We also tested the performance of the proposed method and learned models on a validation dataset of 750 images that were not participating in neural network training and measured the main metrics for estimating detection accuracy (Table I). Obtained results are comparable to human equivalents in terms of accuracy and detection speed.

Developing a VSS will have to be stable against changing environment conditions. Therefore, we tested the stability of the neural network under above-water conditions that were not present in the training set (Fig. 4) and checked the trained models' performance in conditions of high density of fish in the frame and muddy water conditions (Fig. 5).

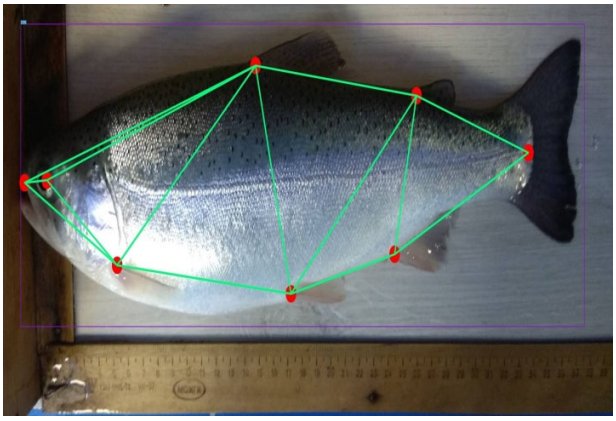

Fig. 4. The result of detection in above-water conditions

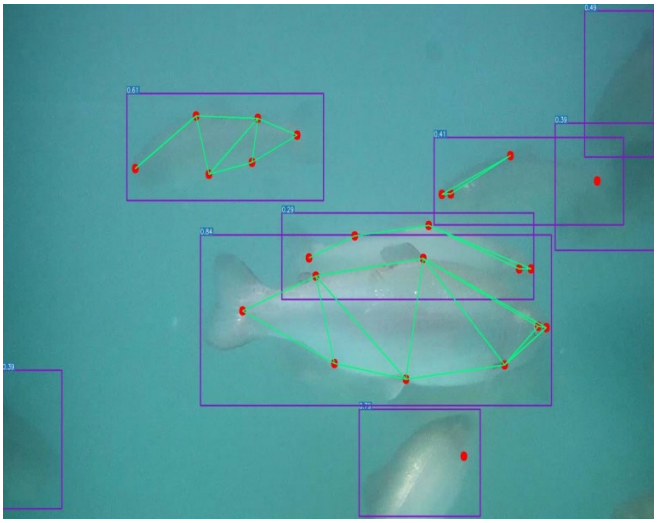

Fig. 5. The result of working in visibility disruptions in muddy water conditions

TABLE I. RESULTS OF THE DEVELOPED SOLUTION

Models	Speed, ms	FLOPS	VRAM, Mb	mAP 0.5	mAP 0.5:0.95	MSPE kpts	MAPE kpts	Max loss kpts
YOLO-Pose s6	8.0	20.2	1841	0.924	0.653	1.46	0.47	12.4
YOLO-Pose m6	12.8	51.9	2049	0.958	0.658	1.21	0.42	9.92
YOLO-Pose l6	23.6	117.1	2401	0.91	0.671	1.12	0.38	7.69

V. Conclusion

The presented demo shows that our proposed method allows achieving performance comparable to human equivalents in your own object research (Table I). The learning neural network is capable of standing up to underwater visual disruption (Fig. 5) and to the high density of fish in the frame. The developed solution shows enough results in both clear and muddy water conditions to use in Underwater Video Surveillance System. It is being actively introduced under the "Fish Grow Platform" to the fish industry in Republic of Karelia and Russian Federation. And we showed that the existing recognition algorithms and technologies can be used in development of such Underwater VSS services in practice

Acknowledgment

This R&D study is implemented within the hardware and software complex for monitoring of aquaculture tank life-cycle "Fish Grow Platform" by LLC "Internet Business Systems" (https://fish-grow.ru). We also extend special thanks to all members of the PetrSU Artificial Intelligence Center team.

References

[1] Marakhtanov A. G., Semenov A. B. Fish Grow Platform. Tsifrovye tekhnologii v obrazovanii, nauke, obshchestve : materialy XV vserossiiskoi nauchno-prakticheskoi konferentsii [Digital technologies in education, science,society : materials of the XV all-russian scientific and practical conference]. Petrozavodsk, 2022, pp.56-59.(in Russian)

[2] Tsarev N. V.,Perminov V. V. Otsenka massy ryby na osnove ee lineinykh razmerov, izmerennykh metodami videoanaliza [Estimation of fish mass based on its linear dimensions measured by video analysis methods]. Tsifrovye tekhnologii v obrazovanii, nauke, obshchestve : materialy XV vserossiiskoi nauchno-prakticheskoi konferentsii [Digital technologies in education, science,society : materials of the XV all-russian scientific and practical conference]. Petrozavodsk, 2022, pp. 112-115 (in Russian)

[3] Freitas M. S. et al. High-throughput Phenotyping by Deep Learning to Include Body Shape in the Breeding Program of Pacu (Piaractus Mesopotamicus). Aquaculture, vol. 562, Elsevier BV, Sept. 2022, p. 738847. https://doi.org/10.1016/j.aquaculture.2022.738847.

[4] Tvirko T. A., Perminov V. V. Opredelenie povedencheskikh parametrov ryby s pomoshch'iu metodov intellektual'nogo analiza videodannykh [Determining fish behavioural parameters using video intelligence methods]. Tsifrovye tekhnologii v obrazovanii, nauke, obshchestve : materialy XV vserossiiskoi nauchno-prakticheskoi konferentsii [Digital technologies in education, science,society : materials of the XV all-russian scientific and practical conference]. Petrozavodsk, 2022, pp. 115-117 (in Russian)

[5] Toshev A., Szegedy C. Deeppose: Human pose estimation via deep neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 2014, pp. 1653-1660.

[6] Maji D. et al. YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition [undefined] (CVPR) Workshops. 2022, pp. 2637-2646.