EMERGING TECH CONFERENCE – Edge Intelligence

Volume 02, 2023, Page 43 – 49

Proceedings of Emerging Tech Conference:
Edge Intelligence 2023

Fish Shape Alignment Based on Deformable Shape Tracking Suite of Tools

Nikos Petrellis[1], Georgios Keramidas[2], Christos Antonopoulos[1] and Nikolaos Voros[1]
[1] *Electrical and Computer Engineering, University of Peloponnese, Patras, Greece*
[2] *School of Informatics, Aristotle University of Thessaloniki, Greece*
npetrellis@uop.gr, gkeramidas@csd.auth.gr, ch.antonop@uop.gr, voros@uop.gr

Abstract

Fish morphological feature extraction based on shape alignment is used to estimate the dimensions, detect malformations, locate body parts like the eyes or the gills, classify fish orientation and species, etc. The Ensemble of Regression Trees machine learning approach is employed and specifically, the Deformable Shape Tracking package is adapted for fish shape alignment. Fish farms can benefit from the proposed approach for fish health assessment, fish growth and feeding needs estimation and harvest time selection. It can also be used for fish monitoring in open seas. Eighteen (18) landmarks are used to define the shape of a fish with an accuracy of approximately 95%. The training and testing were conducted using a custom dataset with low quality underwater images displaying seabream fish. The novelty of this approach is that the customized DEST package is implemented on the reconfigurable hardware of an embedded platform to support hardware acceleration and real time operation.[1]

1 Introduction

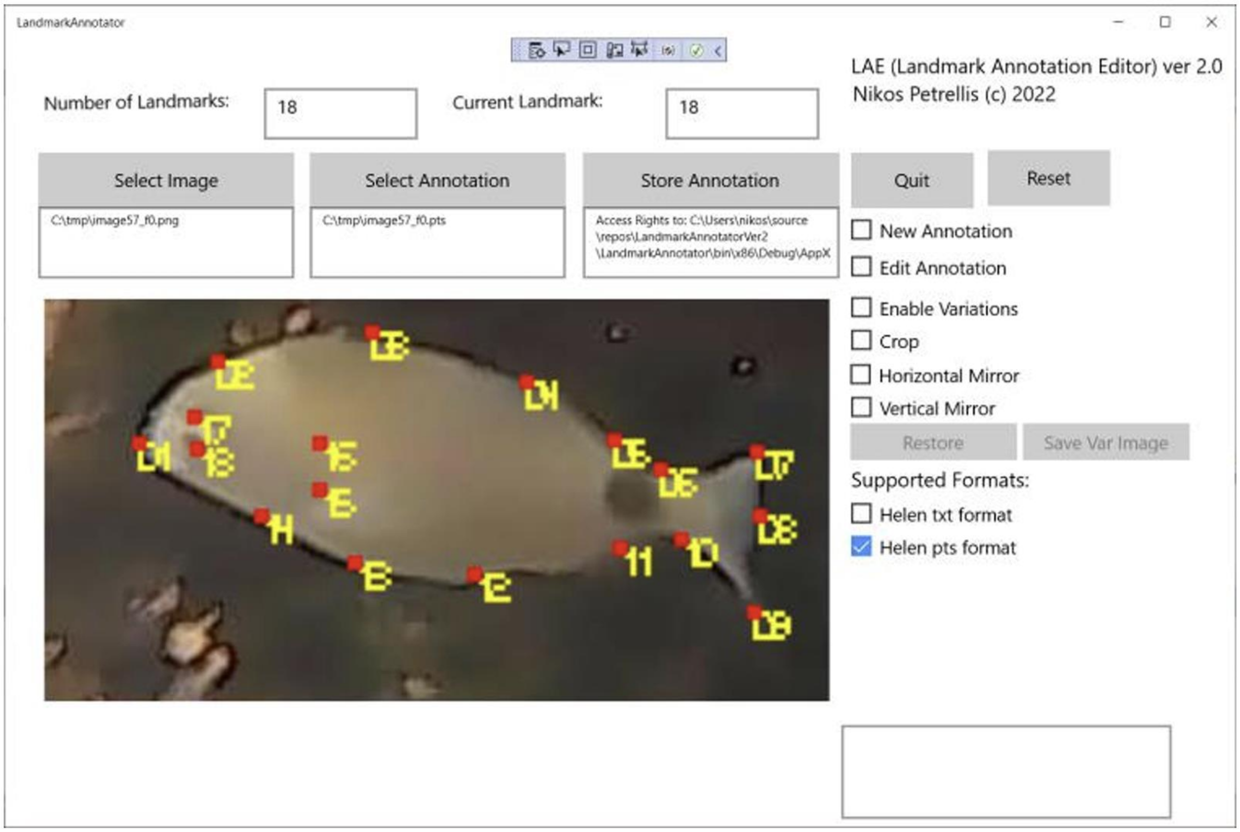
The morphological features that have to be measured daily in an aquaculture include the fish size, its mass, eye diameter, eye and gill color. The fish size and mass can determine when the harvest should be performed as well as the feeding needs. The eye and gill features as well as malformations in the shape of a fish can provide indications about its health, the welfare, etc. Measuring fish dimensions, weight, etc, was until recently performed manually, in an invasive way since fish had to be taken out of the water. The shape can also be used to track fish and interpret its behavior, classify fish species, observe fish populations and these procedures are also important for fish monitoring outside fish cultures (e.g., in rivers or the open sea).

A review of computer vision applications for aquacultures can be found in (Zion, Computers and Electronics in Agriculture, 2012). Applications for fish and egg counting, size measurement, estimation of the fish mass and gender, identification of species and for fish behavior monitoring, are presented in (Zion, Computers and Electronics in Agriculture, 2012). Estimation of fish freshness from photographs and videos captured in a lab environment with controlled light exposure is reviewed in (Franceschelli et al, MDPI Sensors, 2021). The various sensors (biosensors, electric nose and tongue, colorimetric sensor array, spectroscopy, etc)

---

[1] This work has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No 872614 -SMART4ALL: Self-sustained Cross-Border Customized Cyberphysical System Experiments for Capacity Building among European Stakeholder

that can be used for this purpose are presented in this paper. Fish classification in 12 classes based on Fast Regional-Convolutional Neural Networks (Fast R-CNNs) is described in (Li et al, Oceans, 2015). In (Lekunberri et al, Ecological Infromatics, 2022) various tuna species are counted and classified from image frames that display fish on a conveyor belt with 70% accuracy. Mask Regional-Convolutional Neural Network (Mask R-CNN) (He et al, 2023) and ResNet50V2 neural network architectures are employed to measure tuna fish sizes ranging from 23cm to 62cm. Low resolution images are also used in (Sun et al, CISP-BMEI, 2016) for fish recognition with 78% precision. Sonar imaging is used in (Martignac, Fish and Fisheries, 2015) for fish morphology and swimming behavior estimation. A size estimation error ranging from 2%-8% was measured for fish with size between 40cm and 90cm. The fish size and shape are measured using 18 landmarks in (Alsmadi et al.,Journal of Computer Science, 2010) using 3-layer artificial neural network.



LAE (Landmark Annotation Editor) ver 2.0
Nikos Petrellis (c) 2022

Number of Landmarks: 18
Current Landmark: 18

C:\tmp\image57_f0.png
C:\tmp\image57_f0.pts
Access Rights to: C:\Users\nikos\source\repos\LandmarkAnnotatorVer2\LandmarkAnnotator\bin\x86\Debug\AppX

☐ New Annotation
☐ Edit Annotation
☐ Enable Variations
☐ Crop
☐ Horizontal Mirror
☐ Vertical Mirror

Supported Formats:
☐ Helen txt format
☑ Helen pts format

Figure 1: Fish landmark annotation in the LAE editor

In the approach described here, it is assumed that the fish is already detected and the bounding box of the fish is available as a separate patch from the frame it was derived. Fish detection methods like the one described in (Fish Detection, 2023) can be used for this purpose. Moreover, it is assumed that all the fish that are subject to shape alignment have the same orientation. The fish detection model of (Fish Detection, 2023) can be trained to recognize e.g., only fish in a horizontal position facing at a specific direction. A lightweight shape orientation classification based on Principal Component Analysis (PCA) can be used as described in (Lendave, 2023) to classify the orientation of the fish shape. Then, the corresponding shape alignment model trained for this orientation can be employed for shape alignment.

The shape alignment applied on the image patch derived from the bounding box of the detected fish, is based on the machine learning (ML) approach called Ensemble of Regression Trees (ERT), presented by Kazemi and Sullivan in (Kazemi and Sullivan, CVPR, 2014) and is exploited in the Deformable Shape Tracking (DEST) suite of applications (DEST, 2023). The DEST package has been exploited for driver drowsiness applications in our previous work (Petrellis et al,VLSI-SoC, 2021). The source code of the DEST library was ported to Ubuntu host computer, Microsoft Visual Studio 2019 as well as Xilinx embedded Field Programmable Gate Array (FPGA) platform to support hardware acceleration. In another previous work (Petrellis, MDPI Appl. Sci., 2021) fish morphological feature measurement was performed for different fish species, using a different approach.

The contribution of this work can be summarized as follows: a) ERT has been used to align 18 landmarks on the fish shape, b) different ERT models can be trained for different fish orientations for more precise alignment, c) hardware acceleration techniques can be directly applied in shape alignment in order to support real time video processing, d) a new Landmark Alignment Editor (LAE) tool has been developed, e) a new dataset has been created and will be made public.

This paper is organized as follows. The ERT background and the use of the DEST package for training and testing fish shape alignment are discussed in Section 2. The experimental results and discussion are presented in Section 3. The conclusions of this work can be found in Section 4.

2 ERT Background and DEST Adaptation

The Ensemble of Regression Trees (ERT) as described in (Kazemi and Sullivan, 2014) is a shape alignment ML method originally developed for facial shape alignment and applications such as driver drowsiness detection, face expression recognition. In this paper, ERT is adapted for fish shape alignment. The ERT method is applied in a fish bounding box. The fish detection method described in (Fish Detection, 2023) is used to get the coordinates of the bounding box of a fish before applying the ERT method. Fish shape is not symmetrical and therefore the fish should have a specific orientation. The fish is allowed to have a tilt of about ±20 degrees. The employed fish detection method can be trained to recognize only fish in a specific direction, otherwise orientation classification methods should be used to verify that a detected fish has an acceptable orientation. In the rest of this paper, we focus on the fish shape alignment stage performed by the ERT method.

In ERT ML method, T_{cs} cascade stages are visited in order to gradually correct a mean shape stored in the trained ERT model. The correction is based on the comparison of the gray level of pairs of reference pixels belonging to a sparse representation of the input image. The default sparse representation of the input image consists of R_p=600 pixels. In face alignment applications, L=68 landmarks are used. In the fish alignment performed in this paper we use L=18 landmarks defined at the positions shown in Fig. 1.

In each cascade stage, T_{rg} binary regression trees with 2^{td}-1 nodes, are visited. In each node the difference in the gray level of a specific pair of reference pixels is compared to a threshold T_h and either the right or the left child node of the regression tree is followed according to whether the difference is higher or not than T_h. For each node of the regression tree, the threshold T_h, the indices of the reference pixels that have to be compared and the next nodes that have to be visited are all stored in the trained ERT model. When a leaf is reached in the regression tree, a correction factor is found that is added to the current position of the landmarks. If the shape S∈ R^{2L}, is defined as the set of the xi pair of

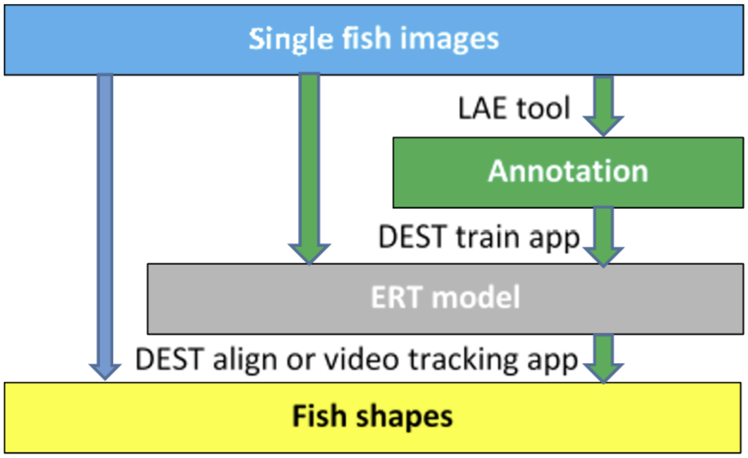


Figure 2: Training and testing fish shape alignment

coordinates: $S = \{x_0 ,x_1 ,\ldots x_{L,}\}$, the ERT algorithm reads the mean shape *S* from the trained model. The, current shape estimate in the regressor t is $\hat{S}^{(t)}$ (t=1,.., T_{cs}) and the transition to the next cascade stage, requires the estimation of a correction factor r_t. This correction factor rt is added to $\hat{S}^{(t)}$ in order to generate the updated shape in the next regressor $\hat{S}^{(t+1)}$. As already mentioned, the r_t value is determined at the leaves of the regression trees that are visited. The regression factors r_t are determined during training using a gradient tree boosting algorithm with a sum of square error loss. During training, the values of rt are determined by the triplet: $\left(I_{\pi_i}, \hat{S}^{(t)}, \Delta\, \hat{S}_i^{(t)}\right)$ where I_{π_i} , $0 \le \pi_i < N$ is the image π_i in the set of N training images and $\hat{S}_i^{(t)}$ is the mean shape derived from all training image with $i \neq \pi_i$. If S_{π_i} is the shape in the π_i image, the residual $\Delta\, \hat{S}_i^{(t+1)}$ in the regressor r_{t+1} is estimated as: $\Delta\, \hat{S}_i^{(t+1)} = S_{\pi_i} - \hat{S}_i^{(t+1)}$. More details can be found in (Kazemi and Sullivan, 2014).

The procedure followed to train and test the fish shape alignment performed from the adapted DEST tools is shown in Fig. 2. During training the Landmark Annotator Editor (LAE) that has been developed, is used to annotate the L=18 landmarks in the images generated by the bounding boxes that display a single fish (see Fig. 1). In the LAE editor, the number of landmarks can be defined and a new annotation can start or an existing one can be modified easily by moving landmarks with a single mouse click. The annotations can be stored in a format compatible with the DEST tools. The LAE editor also offers dataset augmentation services (image mirroring, cropping). The resulting (image, annotation) pairs can be split in training and test images.

The DEST suite of tools (Deformable Shape Tracking, 2023) is developed in C++ exploits the Eigen math library and contains applications for ERT model training (dest_train), shape alignment in single photographs (dest_align), shape alignment in video frames (dest_video_tracking), etc. This software package has been ported to Ubuntu Intel i5/i7 environment as described in (Petrellis et al, 2021) for a driver drowsiness monitoring application. From Ubuntu it was easy to port applications like dest_video_tracking to the Xilinx Vitis environment in order to accelerate in hardware the frame processing rate on an embedded platform (Xilinx ZynqMP Ultrascale+, ZCU102). In the context of porting DEST to Ubuntu, software was restructured and time consuming Eigen calls were replaced by optimized high speed C code achieving a software acceleration in the order of 240 times. Hardware acceleration on the other hand, leads to a further reduction of frame processing latency by 60%. In the context of this work, the DEST suite of tools has also been ported to Microsoft Visual Studio 2019 for operations in Windows environment. In fish shape alignment applications with real time constraints, the accelerated platforms in Ubuntu and ZynqMP Ultrascale+ can be exploited for high speed frame processing. With hardware acceleration and only 18 landmarks as is the case of the fish shape alignment described in this paper, the frame processing latency is estimated between 10 and 15 ms

The application dest_train has been used to train the ERT model for fish shape alignment based on the dataset created by the LAE editor. The trained ERT model can be used in real time when an input image frame is analyzed. The fish shape is determined by the L=18 landmarks that are defined as follows (see Fig. 1): landmark No. 1 is the fish mouth, landmarks 2-5 are the upper part of the contour, landmarks 6-10 are defined in each salient of the caudal fin, landmarks 11-14 make the bottom part of the contour, landmarks 14-16 denote the position of the gill and finally, landmarks 17-18 denote the location of the fish eye.

The position of the eye and the gills can be used to analyze the color of these body parts for disease diagnosis. The fish shape and its malformations can also reveal information about the fish health, variety, welfare, growing conditions, etc. The distance between landmarks 1 and 8 can be used to estimate the relative length of the fish while landmarks 3 and 13 can be used to estimate the height.

3 Experimental Results-Discussion

The ERT model was trained using 300 fish photographs. The orientation of the fish was horizontal, facing left. The contrast of the fish images was low as shown in Fig. 1. The images were selected on purpose with low quality in order to test the developed shape alignment method under worst case conditions. Concerning the resolution of the dataset images, the width ranges between 60 and 400 pixels, while the height between 20 and 300 pixels. The test set consisted of 100 fish images. Seven ERT models have been trained as shown in Table 1.

Model	T_{cs}	T_{rg}	Frame Process Latency (MS VS2019)	Normalized error	Fish Length estimation err.
M1 (default)	10	500	344 ms	4.81%	6.09%
M2	8	500	281 ms	5.16%	6.69%
M3	10	400	310 ms	4.81%	6.11%
M4	8	400	264 ms	4.91%	6.22%
M5	6	400	156 ms	6.21%	6.40%
M6	8	300	211 ms	5.06%	6.42%
M7	6	300	200 ms	6.41%	7.50%

Table 1: Comparison in the speed and error of the trained models

In the default model M1, T_{cs}=10 cascade stages have been employed with T_{rg}=500 regression trees in each stage. The cascade stages have been reduced to 8 in M2 while the regression trees have been reduced to

400 in M3. Both cascade stages and regression trees have been reduced to 8 and 400, respectively, in M4. The cascade stages have been further reduced to 6 in M5 along with 400 trees. In M6, 8 cascade stages have been used with only 300 trees in each cascade stage. Finally, in M7, 6 cascade stages with 300 trees are used. The frame processing latency is the one listed in the 4th column of Table 1 and is measured in a Microsoft Visual Studio 2019 environment. The frame processing latency is almost proportional to $T_{cs} \cdot T_{rg}$. In an Ubuntu Intel Core i5-9500 CPU @3.00GHz, 6 core processor with 16GB RAM environment, the frame processing latency for M1 is less than 0.4us while this latency for M7 is less than 0.2us. On an embedded ZCU102 platform the corresponding latencies for M1 and M7 are less than 15ms and 8ms, respectively. The normalized relative error is estimated as the mean relative Euclidean distance between the estimated landmark position and the one annotated with LAE as ground truth. In the estimation of the normalized error the width and height distances are divided by the image width and height, respectively. As can be seen from the 5th column of Table 1, the normalized error in M3 is the same with M1 although the M3 latency is 10% shorter than M1. The mean error in M6 (5%) is quite close to M1 while its latency is 39% smaller. The estimation error in the fish length from landmarks No. 1 and 8, is listed in the last column of Table 1, for each ERT model. Compared with referenced approaches, (Lekunberri, et al., 2022) estimate tuna size with a standard deviation between 0.328 and 0.396, while (Martignac et al., 2015) estimate fish length with an error between 2% and 9%. Generally, we can state that our approach can achieve a high accuracy with a very low latency.

4 Conclusion

The DEST implementation of ERTs was adapted for fish shape alignment and morphological feature extraction. The position of 18 landmarks was estimated with an accuracy of about 95%. Future work will focus on the incorporation of the developed method for morphological feature estimation in a framework where automatic fish detection and tracking will also be supported.

5 References

[1] Alsmadi, M. K., Omar, K. B., Noah, S. A., Almarashdeh, I. (2010), Fish Recognition Based on Robust Features Extraction from Size and Shape Measurements Using Neural Network. Retrieved from https://doi.org/10.3844/jcssp.2010.1088.1094.

[2] Deformable Shape Tracking (DEST) (2023, June). Retrieved from https://github.com/cheind/dest.

[3] Fish detection (2023, June). Retrieved from https://github.com/kwea123/fish_detection.

[4] Franceschelli, L., Berardinelli, A., Dabbou, S., Ragni, L., and Tartagni, M. (2021). Sensing Technology for Fish Freshness and Safety: A Review. MDPI Sensors 21, (p. 1373). Retrieved from https://doi.org/10.3390/s21041373

[5] He, K., Gkioxari, G., Dollár, P., Girshick, R. (2023). Mask R-CNN. Retrieved from https://arxiv.org/abs/1703.06870.

[6] Kazemi, V., and Sullivan, J. (2014, June). One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 867-1874). Retrieved from 10.1109/CVPR.2014.241.

[7] Lekunberri, X., Ruiz, J., Quincoces, I., Dornaika, F., Arganda-Carreras, I., Fernandes, A.A. (2022). Identification and measurement of tropical tuna species in purse seiner catches using computer

vision and deep learning. Ecological https://doi.org/10.1016/j.ecoinf.2021.101495.

[8] Lendave, V. (2023, June). Detecting Orientation of Objects in Image using PCA and OpenCV. Retrieved from https://analyticsindiamag.com/detecting-orientation-of-objects-in-image-using-pca and-opencv/

[9] Li, X., Shang, M., Qin, H., and Chen, L (2015). Fast accurate fish detection and recognition of underwater images with Fast 1R-CNN. Oceans (pp. 1-5). Retrieved from 0.23919/OCEANS.2015.7404464

[10] Martignac, F., Daroux, A., Bagliniere, J.L., Ombredane, D., and Guillard, J. (2015). The use of acoustic cameras in shallow waters: new hydroacoustic tools for monitoring migratory fish population. A review of DIDSON technology. Fish and Fisheries 16 (pp. 486–510). Retrieved from https://doi.org/10.1111/faf.12071

[11] Petrellis, N., Christakos, P., Zogas, S., Mousouliotis, P., Keramidas, G., Voros, N., Antonopoulos, C. (2021). Challenges Towards Hardware Acceleration of the Deformable Shape Tracking Application. In Proceedings of the 2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC). Retrieved from 10.1109/VLSI-SoC53125.2021.9606999.

[12] Petrellis, N. (2021). Measurement of Fish Morphological Features through Image Processing and Deep Learning Techniques. MDPI Appl. Sci. 11 (p. 4416). Retrieved from https://doi.org/10.3390/app11104416.

[13] Petrellis, N. (2021). Measurement of Fish Morphological Features through Image Processing and Deep Learning Techniques. MDPI Appl. Sci. 11 (p. 4416). Retrieved from https://doi.org/10.3390/app11104416.

[14] Sun, X., Shi, J., Dong, J., Wang, X. (2016). Fish recognition from low-resolution underwater images. In Proceedings of 9th IEEE International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) (pp. 471–476). Retrieved from https://doi.org/10.1109/CISP-BMEI.2016.7852757

[15] Zion, B. (2012). The use of computer vision technologies in aquaculture – A review. Computers and Electronics in Agriculture 88 (pp. 125-13). Retrieved from https://doi.org/10.1016/j.compag.2012.07.010.