Comparison of Methodologies for Detecting Feeding Activity in Aquatic Environment

Gonçalo Adolfo
Instituto Superior Técnico
Lisbon, Portugal
Email: goncaloadolfo20@gmail.com

Helena Sofia
Instituto Superior Técnico
Lisbon, Portugal
Email: hsofiapinto@gmail.com

Alexandre Bernardino
Instituto Superior Técnico
Lisbon, Portugal
Email: alexandre.bernardino@edu.ulisboa.pt

***Abstract*—Feeding is one of the main activities in the aquatic context. This project was developed in the Lisbon Oceanarium domain, in which monitoring it is especially important. The automation of its detection is not only useful to help monitor the activity but also for investigation and perception of species behavior. Due to its importance to Lisbon Oceanarium, we focused on sharks and manta rays. In this project, several methodologies were applied to automatically detect feeding activity, based only on frames filmed by a static camera. Feeding activity is detected using a Convolutional Neural Network (CNN), that automatically concludes patterns in the image itself, or using methods based on motion variability or aggregation since the beginning of this activity is described by changes in these two features. The amount of motion is defined using frame subtraction methods or the Optical Flow algorithm. Aggregation is defined by applying the Delaunay Triangulation algorithm forming a triangular mesh over the fish detected in a given image. For its implementation and evaluation, several videos were filmed at Lisbon Oceanarium. To evaluate each of the approaches, several metrics are extracted such as accuracy, precision, and recall.**

I. Introduction

The animal world, namely the aquatic environment, is based on behaviors within and outside of observable normal patterns. Thus, it is essential to detect and measure these behaviors, to help biologists, both in terms of research and in terms of control and monitoring. This is the essence of this project, incorporated in the context of the Lisbon Oceanarium. From the set of observable behaviors, feeding behavior stands out. It is considered especially interesting because of its impact on production costs and water quality. Underfeeding leads to aggressive behavior while overfeeding leads to food waste (more costs) and the uneaten food/fish feces interferes with water quality. This activity is usually controlled based on the observer's experience, which may be subjective since many factors can contribute to fish appetite: physiological, nutritional, environmental.

Automatic feeding activity detection helps to carry out its management but also to understand the behavior of species. On one hand, the detection of this activity at an unexpected time may indicate that species are being poorly fed, or even other problems. On the other hand, the detection of a fish outside the feeding zone during this activity indicates that this fish is not interested in food. The traditional way to identify these behaviors is based on visual inspection which is very time-consuming and subjective. Additionally, the habitat conditions insert some challenges: species variability, which implies different feeding activity definitions; the size of some tanks, such as the main tank of Lisbon Oceanarium, making it difficult for the camera to cover all its range; the presence of habitat components such as rocks and fauna.

Computer science areas, such as computer vision and machine learning, have evolved in recent years which allows the implementation of a system with the capability of recognizing activities. The approaches in [2], [8] used an accelerometer on each fish of interest, and certain activities were detected based on the collected values. Broell et al [2] (2013) focused on detecting the following set of activities: swimming, feeding, and escaping. They proposed a signal processing system based on the analysis of time series features, related to the acceleration in each dimension. The idea was to identify which features could have identical values within the same activity but different between different activities. Zhang et al. [8](2019) solves a similar problem but focuses on activities related to sharks: swimming, resting, feeding, and non-deterministic movement. Similar to the previous method, it uses time series of the value of the overall dynamic body acceleration (ODBA). Given a set of example time series for each activity (2-second segments), three different models of deep learning were trained, more specifically Convolutional Neural Networks (CNN). These approaches, despite achieving good results, are considered invasive for the species in question, and therefore not supported by biologists.

Zhou et al. developed several projects [12], [10], [11] within the scope of the feeding activity. Initially, in [12], the goal was to detect the feeding activity through the analysis of the level of aggregation of the shoal, since it is usually higher during this period. In [10], one more index was used: Snatch Intensity of Fish Feeding Behavior (SIFFB). In this method, it was argued that fish usually eat close to the surface, and during the feeding period, the surface texture changes substantially due to the intensive movements of the fish. Finally, in article [11], an innovative idea was described to identify the appetite of the fish present in a given image. A convolutional neural network (CNN) was trained based on several images at different levels of appetite.

II. Problem Statement and Objectives

This project was implemented in partnership with the Lisbon Oceanarium. In this institution, as in many others, biologists analyze the behavior of species visually in real-time. This causes limitations in terms of time management and it can be subjective due to the biologist's experience. For these reasons, the main objective was to develop a system capable of helping biologists manage the main tank, identifying automatically the feeding activity. As previously mentioned, it is especially important to support the feeding control but also to analyze and investigate the species behavior.

Two previous projects [3], [6] were implemented, that explored the detection, tracking, and classification of fish on two tanks of the Lisbon Oceanarium: main tank and coral tank. This project tries to complement the following works on the activity recognition field, more concretely, on the feeding activity. Since there is some diversity in how the different species are fed, we decided to focus on sharks and manta rays as they are species that need special attention from biologists, according to our conversations with biologist Hugo Batista. These inhabit the main tank of the oceanarium, which is the tank with the largest area, highlighting the importance of a system capable of assisting with its monitoring.

Feeding activity is similar in these species. During this period, they tend to aggregate in the feeding area, which does not normally happen except during this behavior. On one hand, manta rays do not usually frequent the bottom of the tank, but they are fed in this area through divers (Figure 1b). Sharks, on the other hand, are fed through sticks with food at their tip, closer to the surface (Figure 1a).

The implemented system falls within the computer vision field. Several videos of the main tank were filmed in the Lisbon Oceanarium, focusing on the feeding activity, and using a camera placed outside the tank and in a static position. The developed system was implemented and evaluated using these videos. Several approaches were experimented to detect automatically the feeding activity, to verify which one is more appropriate to our domain.

In this project, four different methods were trained: a convolutional neural network (CNN), a motion variability approach through active pixels identification or Optical Flow, and an aggregation variability approach. All these methodologies do not resort to fish detection and tracking information but only to the video frame, except the aggregation-based method. These approaches receive a video as input and, through video frames processing, return the timestamps of the feeding activity as output.

III. Motion Variability

Fish have an internal clock. When they feel it is feeding time, they start to get together at the feeding zone. For that reason, the level of motion near this zone starts to increase, as we can state from Figure 1b, which illustrates a frame during the feeding activity at the bottom of the tank. As we can see, several fish from different species, including manta rays, aggregate near the divers to get some food. This logic is not so visible for the surface feeding. At this zone, the level of motion is by default high because of the fish shoals that usually swim there. Additionally, the species that are fed at the surface, such as sharks, are more orderly which does not cause a motion difference as verified at the bottom.

This approach takes advantage of this motion variability to identify the feeding periods. To do so, we start by characterizing the level of motion on each frame. There are two different ways of describing motion: number of active pixels or Optical Flow average magnitude. Regardless of how to quantify it, the approach is based on a motion threshold to classify each frame as being part of feeding activity or not. When several consecutive frames are related to feeding then we consider it as a feeding period. This approach is only based on the video frames and does not resort to fish information.

Identifying the motion pixels is most of the time a preliminary step to identify blobs in a given image, and consecutively detect objects, as performed in [5], [9], [12], [10]. In this project, it is used only as a measure of motion, so we can identify a transition to feeding activity. It is characterized for the following steps:

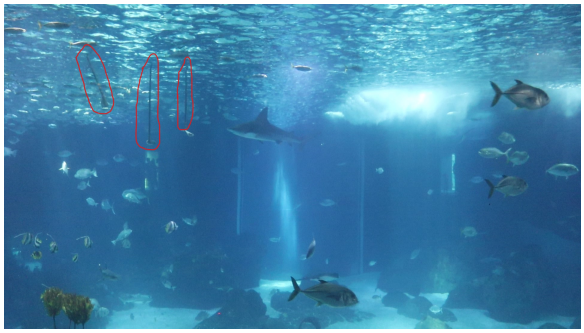
1) transform the frame to a grayscale frame (one single plane)
2) subtract the current frame (f_t) with the previous frame (f_{t-1}) obtaining the difference frame (d_t): $d_t = |f_t - f_{t-1}|$
3) apply a binarization threshold (t): pixels with a high difference from the previous frame are considered motion pixels

$$m(x,y) = \begin{cases} 1, & \text{if } d(x,y) \geq t \\ 0, & \text{otherwise} \end{cases}$$

With the binary motion frame (m_t), it is possible to quantify the level of motion as the number of motion pixels. Figure 2 illustrates an example of a motion frame during the feeding activity. It is possible to apply this logic in a specific part of the frame, as we also can conclude from the figure. The idea is to evidence the feeding zone instead of focusing on the entire frame plane.

By applying the previously mentioned steps on all frames of a given video, we can produce a timeseries of the number of motion pixels (Figure 3). As we can see from both Figures (2 and 3) there are some noise regarding the motion pixels identification, which consequently affects the timeseries. To highlight the bi-modal property of the motion level, an exponential moving average filter is applied to the original timeseries. It is characterized for giving more importance to near values. The new value for a given timestamp t is defined as the weighted (α) average of the values in a given window of size n: $y(t) = \sum_{k=0}^{n} (1-\alpha)^k x[t-k]$. As we can see in Figure 3, a more clear separation is visible between feeding activity and the normal period after the filter application.

We also provide an alternative to defining motion levels using an Optical Flow algorithm. This algorithm allows estimating motion for a grid of particles, given the current frame



(a) Feeding at the surface

(b) Feeding at the bottom

Fig. 1: Different feeding scenarios.

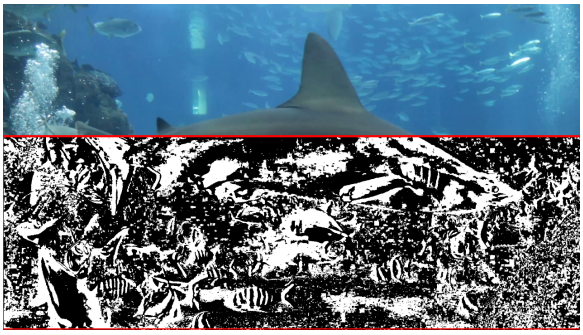


Fig. 2: Motion pixels for a given frame during the feeding activity.

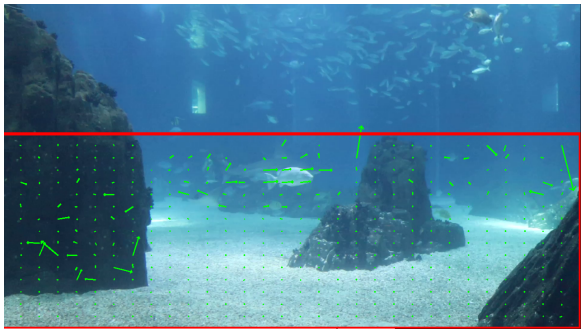


Fig. 4: Motion estimation using Optical Flow.

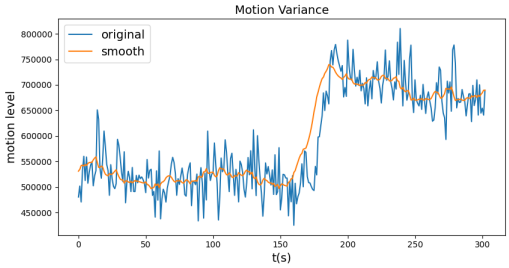


Fig. 3: Number of motion pixels over a training video.

and the previous frame. Figure 4 illustrates an example of the resulting motion vectors after the application of the algorithm. Based on this approach, the motion level is considered as the average magnitude of all output vectors. The remaining logic is the same as the active pixels method, considering this new way of defining the level of motion. Similar to the active pixels method, it is also possible to define a specific region instead of the frame as a whole.

In summary, the idea of this approach is to identify transitions between the overall motion level on the feeding zone. Both motion estimation methods provide a clear separation between these two states (normal and feeding), according to a training video. When analyzing the motion timeseries, the chosen value for the threshold of the active pixels was 394K and for the Optical Flow magnitude was 2.8, considering only about 60% of the height of the frame and a frame resolution of 1920x1080. When using whole image plane, the Optical Flow threshold stayed the same at a value of 2.8, but since the focus region is larger the threshold of the active pixels changed to 625K.

IV. Convolutional Neural Network (CNN)

Convolutional Neural Networks is a neuronal network class that usually give good results on the image processing domain. It incorporates internally on its architecture feature extraction layers, composed essentially by convolutional layers that apply the convolutional operations, and pooling layers for down sampling. Resorting on this model is one of the possible solutions to detect feeding periods, in which the frames are directly passed as input. Internally, a feature vector is calculated and passed to a set of full connected layers to decide between being a feeding frame or not. Similar to the approach explained in the previous section, when several consecutive frames are classified as feeding frames it is considered a feeding period.

The work in [11] also uses a neural network of this nature, but with a sensibly different goal: measure the feeding hungry intensity of the fish present in a given frame. Its architecture was used as a baseline architecture for our project. However, it has one single output neuron since it is a binary classification problem. Figure 5 illustrates this architecture:

- input layer that receives a low resolution frame as input (80x50)
- two convolutional/max-pooling layers with 5 and 10 kernels respectively, relu as activation function, and a 2x2 window for the max-pooling operation
- two full connected layers with 120 and 84 neurons respectively, relu as activation function, no dropout
- a single output neuron to decide between feeding frame and normal frame

Several videos were filmed in the main tank of the Lisbon Oceanarium, for both focus feeding scenarios: bottom and surface. Two datasets were built transforming all its frames into low-resolution frames (80x50). These were separated into a training and a testing set following an 80-20 division percentage. Consecutive frames were inserted into the same set, instead of using a randomization approach, in order to have more reliable results.

CNN parameters can have a significant impact on the performance, either in terms of architecture or in terms of hyper-parameters. Tables I and II contain a summary of the set of parameters that were tuned. On one hand, regarding architecture, we decided to vary the number of convolutional/pooling layers, the number of hidden layers, the different number of filters, and the number of neurons on each hidden layer. On the other hand, we experimented different learning rate and dropout values, as well as different activation and error functions.

A grid search approach was applied to verify the most suitable values for each parameter and each feeding dataset. The training samples were split into a training set and validation set, and several models were trained and evaluated. For the bottom-feeding dataset it was possible to conclude the following points:

- architecture: best results were achieved using one convolutional layer and two hidden layers or vice versa;
- hyper-parameters: best results were achieved using a learning rate of 0.001 and the mean squared error as error function;
- the model with the best performance (accuracy of 94%) on the validation had the following parameters: number of convolutional layers (1), number of hidden layers (2), number of neurons per hidden layer (120), number of applied filters (5), learning rate (0.001), dropout rate (0%), activation function (relu), error function (mean squared error).

For the surface dataset, according to hyper-parameters, there was no significant difference between the achieved values. However, more differential results were observed regarding architecture. When using only 1 hidden layer, the best results seem to be getting together with 80 neurons per layer. On the other hand, using 2 hidden layers worked better using 120 neurons on each. Additionally, as a general rule, using only 1 hidden layer also gives better results. However, despite all these patterns, 78% of accuracy (best accuracy) was achieved using a combination of 2 hidden layers with 120 neurons each and 1 convolutional layer, and 5 filters. Regarding the number of filters, if we change the number of filters to 15 the accuracy decreases 9%.

TABLE I: Hyper-parameters description

parameter	description
learning rate (0.01, 0.001)	step size at each iteration
dropout rate (None, 0.2)	percentage of neurons that are ignored
activation function (relu, sigmoid)	defines the output of a neuron
error function	measure of prediction (mean squared, cross entropy)

TABLE II: Architectural parameters description

parameter	description
#convolutional layers (1,2)	layer that applies a set of filters
#hidden layers (1,2)	layer of mathematical functions that produce a given output
#neurons per hidden layer (80,120)	mathematical function units
#filters (5,15)	matrices that slide over the image (convolution)

V. Aggregation-based

Similar to the logic around the motion measurement, fish also tend to aggregate during feeding. To measure the level of aggregation, of the detected fish in a given frame, we used the approach explained in work [12]. It uses the Delaunay Triangulation [4] defining a triangular mesh between the detected fish, to calculate a measure named as Flocking Index of Fish Feeding Behavior (FIFFB). The Delaunay Triangulation algorithm is applied to originate a set of triangles that interconnect the points. This algorithm allows obtaining the triangles that maximize the minimum angle of all triangles. It is defined by the following points:

1) a point p_1 is selected randomly from the set of points not covered yet
2) the first edge is formed with the closest point p_2
3) the third chosen point is the point that forms the circumference with the smallest radius and that satisfies the Delaunay rule: no point can be within circumferences formed by other triangles

With the obtained triangles, the FIFFB value is described by the sum of the perimeter of all triangles ((1) where n is the total number of triangles and L the length of each side). The lower this value, the greater the level of aggregation. The analysis of the value of this index also makes it possible to indirectly conclude the level of appetite: the longer the fish take to return to their "normal" aggregation level, the longer the feeding period and in turn the greater the appetite.

$$FIFFB = \frac{\sum_i^n (L1_i + L2_i + L3_i)}{n} \quad (1)$$

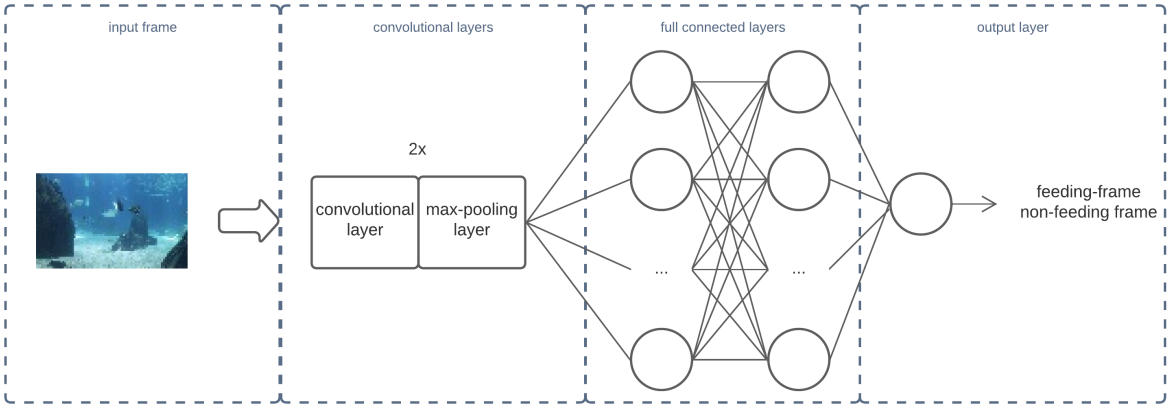


Fig. 5: Baseline CNN architecture.

This approach, as we can see from the previous explanation, is dependent of the fish detections, or trajectories if tracking is available. This information is passed as input to the Delaunay Triangulation which will form the mesh that will be used to calculate the flocking index. We performed three additions to this approach:

1) trajectories interpolation to fill possible gaps and miss detections
2) identification of outlier detections to identify fish with lack of interest
3) divide the flocking index value by the number of detections, so that more aggregated fish do not have a higher value than few fish farther apart.

Two types of interpolation were implemented: linear as used in work [1], and also newton [7]. This is applied for both axes independently: x position and y position. Linear interpolation assumes that the speed during the missing period is constant. The value for one axis (x or y) for a given instant t can be calculated as

$$y(t) = y_0 + (t - t_0)\frac{y_1 - y_0}{t_1 - t_0},$$

where t_0, t_1 are the timestamps of the gap edges positions, and y_0, y_1 are the position values (x or y) for those points. On the other hand, newton interpolation takes more points into account (interpolation points). Using $k+1$ interpolation points, defining a polynomial of degree k, the value for an instant t can be calculated as

$$N(t) = \sum_{j=0}^{k} a_j n_j(t),$$

where a_j is the jth coefficient and n_j is the newton basis function. Based on this type of interpolation, there are two different ways of choosing the interpolation points:

- equidistant points among the position time series
- nearest points to the gap edges

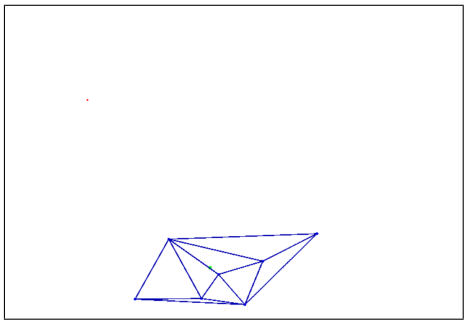


Fig. 6: Outliers and outuput mesh for an example frame.

Regarding outlier centroids, these are detected before the flocking index calculation, and are considered the fish with lack of interest. This is determined by finding out if any of the distances, to the center of mass of the feeding behavior, is less than $1.5 \times IQR$ below the first quartile or more than $1.5 \times IQR$ above the third quartile, being $IQR = Q_3 - Q_1$ and Q_1, Q_3 the first and third quartile respectively. The center of mass is defined as the median between all the detected fish centroids, for both dimensions: x and y. Figure 6 illustrates an example of the aggregation-based approach. In this Figure, we can see the detected fish centroids marked as a dot, the output mesh generated by Delaunay Triangulation, the center of mass marked as a green dot, and finally the considered outlier fish marked as a red dot.

VI. Evaluation

A. Bottom Feeding Results

Using the model parameters that gave the best performance, in the case of the Convolutional Neural Network (CNN), and the predefined thresholds in the case of the motion/aggregation variability methods, these approaches were evaluated in a different test set. Both approaches gave good results according to accuracy, precision, and recall metrics having been able to reach 92%, 96% and 87% of accuracy respectively (Table III). However, the optical flow method classified most of the frames

Method	Accuracy	Precision	Recall
CNN	0.90	0.95	0.87
active pixels	0.96	1	0.94
optical flow	0.4	0.65	0.05
aggregation value	0.87	0.80	0.99
active pixels (using region)	0.96	1	0.94
optical flow (using region)	0.4	0.7	0.06

TABLE III: Bottom feeding test set results

as non-feeding frames, which means that the vectors gathered on the test video had a lower average magnitude than the one observed on the training video, even on the feeding period.

One additional experience was made: evaluate the motion-based approach using the region definition ability and errors dispersion in time. As we can see in Table III, the region was not the problem for the miss classification of some of the frames because the same results were sensibly observed. As a final analysis, we also tried to understand if the errors that we were getting were close to the feeding state transition. For the CNN approach, the errors are scattered through time. The same does not happen with the active pixels method. When the feeding period ends, the motion that is detected does not drop instantly, as we can visualize in Figure 3, once the frames that are still close to this state change can still be classified as feeding frames.

B. Surface Feeding Results

Surface feeding is more difficult to notice than bottom feeding. When this event happens on the bottom, a clear change in the number of aggregated species is verified, and also on the level of motion in that region. Surface feeding is hard to visualize. Even as a human, the food sticks are not easily detectable on the frames, only if there is some knowledge about the context. Figure 1a illustrates a frame during the surface feeding period. This example frame has a high resolution which is not the one passed into the network. Even in this resolution, we can conclude that the sticks are hard to verify.

In terms of evaluation, the same methodology was used when comparing to bottom-feeding evaluation. First, we tried to identify the most suitable parameters in terms of hyper-parameters and architecture, as explained in section IV. After that, using the parameters that gave the best performance on a validation set, the model was evaluated on a different test set. Table IV illustrates the resulting confusion matrix. This model was able to identify correctly almost all the non-feeding frames giving a maximum recall close to 1. Relatively to feeding images, the precision was significantly lower obtaining a value of 0.53. As expected, better results were achieved on the bottom-feeding model.

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	4333	3788	8121
	n′	2	5171	5173
	total	4335	8959	

TABLE IV: Resulting confusion matrix for surface feeding

VII. Conclusion

Automatic behavior detection can play a major role, which can save time for biologists, and let them focus on other tasks. Traditionally, it is made by visual inspection, and that requires the biologists to spend considerable time analyzing fishes, and it can be subjective to biologist experience.

In this project, we made a comparison of different approaches to detect the feeding activity in the main tank of the oceanarium, and as described, focusing on sharks and manta rays. This is the tank that presents the focus species, and the most useful to monitor given its dimensions, being complicated to carry out an efficient monitoring only using manual inspection. There are several approaches that can describe this behavior: we can define a threshold according to aggregation or motion variability, or try to model feeding frames patterns through a Convolutional Neural Network.

Overall all the approaches achieved good performance metrics. Convolutional Neural Networks could model feeding image patterns and achieved 90% of accuracy. The aggregation method could also achieve good performance obtaining 87% of accuracy. The motion approach, based on active pixels identification, also obtained results in the excellent range (96%). On the other hand, using optical flow to this effect was noisier decreasing this value to only 40%. Despite these values, there are several topics that could be explored in the future. It would be interesting to try to understand what image patterns, and image regions, are characterizing each of the classes. Additionally, all the methods suffer from the lack of knowledge of depth information. Another future problem could be focusing on trying to take advantage of depth data, in order to be able to have a more flexible and less restrictive system regarding the position of the camera. Finally, it would be extremely useful to develop an application to be used by biologists. This could allow the analysis of the generated feeding alerts, in order to help biologists to carry out the monitoring of the tanks.

Acknowledgment

We would like to thank Lisbon Oceanarium for the interest and trust in this project, and for letting us film videos from the oceanarium tanks that were key to the development process. A special thank you to Núria Baylina and Hugo Baptista for letting us know more about aquatic life and for helping us to understand what could be important for biologists.

REFERENCES

[1] Cigdem Beyan and Robert B Fisher. Detection of abnormal fish trajectories using a clustering based hierarchical classifier. In *British Machine Vision Conference*, 2013.

[2] Franziska Broell, Takuji Noda, Serena Wright, Paolo Domenici, John Fleng Steffensen, Jean-Pierre Auclair, and Christopher T Taggart. Accelerometer tags: detecting and identifying activities in fish and the effect of sampling frequency. *Journal of Experimental Biology*, 216(7):1255–1264, 2013.

[3] José Castelo, H Sofia Pinto, Alexandre Bernardino, and Núria Baylina. Video based live tracking of fishes in tanks. In *International Conference on Image Analysis and Recognition*, pages 161–173. Springer, 2020.

[4] Der-Tsai Lee and Bruce J Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980.

[5] Vassilis M Papadakis, Ioannis E Papadakis, Fani Lamprianidou, Alexios Glaropoulos, and Maroudio Kentouri. A computer-vision system and methodology for the analysis of fish behavior. *Aquacultural engineering*, 46:53–59, 2012.

[6] Joao Santos. Tracking animals in underwater videos. Master's thesis, Instituto Superior Técnico, Lisbon, 2020.

[7] Hillel Tal-Ezer. High degree polynomial interpolation in newton form. *SIAM journal on scientific and statistical computing*, 12(3):648–667, 1991.

[8] Wenlu Zhang, Anthony Martinez, Emily Nicole Meese, Christopher G Lowe, Yu Yang, and Hen-GeulHenry Yeh. Deep convolutional neural networks for shark behavior analysis. In *IEEE Green Energy and Smart Systems Conference (IGESSC)*, pages 1–6. IEEE, 2019.

[9] Jian Zhao, Zhaobin Gu, Mingming Shi, Huanda Lu, Jianping Li, Mingwei Shen, Zhangying Ye, and Songming Zhu. Spatial behavioral characteristics and statistics-based kinetic energy modeling in special behaviors detection of a shoal of fish in a recirculating aquaculture system. *Computers and Electronics in Agriculture*, pages 271–280, 2016.

[10] Chao Zhou, Kai Lin, Daming Xu, Lan Chen, Qiang Guo, Chuanheng Sun, and Xinting Yang. Near infrared computer vision and neuro-fuzzy model-based feeding decision system for fish in aquaculture. *Computers and Electronics in Agriculture*, 146:114–124, 2018.

[11] Chao Zhou, Daming Xu, Lan Chen, Song Zhang, Chuanheng Sun, Xinting Yang, and Yanbo Wang. Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision. *Aquaculture*, 507:457–465, 2019.

[12] Chao Zhou, Baihai Zhang, Kai Lin, Daming Xu, Caiwen Chen, Xinting Yang, and Chuanheng Sun. Near-infrared imaging to quantify the feeding behavior of fish in aquaculture. *Computers and Electronics in Agriculture*, pages 233–241, 2017.