Fish Behavior Detection through Video Frames and Trajectories

Gonçalo Adolfo
goncalo.adolfo@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2021

Abstract

Detecting automatically fish behaviors can be helpful to monitor fish in tanks, which can save considerable time for biologists. In this project, we developed a system capable of detecting abnormal behaviors, feeding periods, and also interesting moments to be analyzed by biologists. Due to its importance to Lisbon Oceanarium, we focused on sharks and manta rays. The system relies on video frames, and fish trajectories which are represented in a features vector format. To detect abnormal behaviors, we resorted to clustering approaches or a switching vector model, and several classifiers were trained to detect interesting moments. Additionally, it is possible to define a set of species-specific rules regarding the extracted features. Feeding periods are detected using a convolutional neural network, or based on aggregation/motion variability. To evaluate each of the approaches, several metrics are extracted such as accuracy, precision, and recall.
Keywords: Machine Learning, Behaviors, Computer Vision, Trajectories, Aquatic Species

1. 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, in order to help biologists, both in terms of research and in terms of control and monitoring. This is the essence of this thesis, incorporated in the context of the Lisbon Oceanarium. Fish behaviors themselves can be seen with different levels of abstraction. The focus can be to detect something out of the ordinary or to detect explicitly known activities such as resting, feeding, swimming, among others. The abnormal behaviors can serve as an alarm for something of interesting to analyze and the known activities can help to understand fish patterns through the day (or other levels of granularity).

From the set of observable behaviors, feeding behavior is one that 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.

This thesis is fundamental for the work of biologists at the Lisbon Oceanarium, as it helps them to monitor fishes, preventing in advance certain problems that these species may have in their habitat, preserving the habitat stability. An abnormal behavior can have different meanings: fish disease, problems in water quality, poor habitat integration, or poor integration with other species. Whatever the cause, the traditional way to identify these behaviors is based on visual inspection by marine biologists. This is considered, by many, very time-consuming, dependent on the biologist's experience and also invasive due to the biologist presence. A possible alternative would be to resort to technology and install cameras that could record the different environments, for further analysis. In this case, it would be possible to continuously analyze, instead of having to be analyzed in person in the period of interest. However, the amount of data produced daily would be huge requiring considerable human effort to process it, in addition to the storage space needed to store all the produced videos.

Another aspect to consider is the fact that the oceanarium, namely the Lisbon Oceanarium, contains several tanks with different species. For this reason, biologists cannot be present in all the tanks at the same time. On the other hand, even with several installed cameras, the amount of information to be processed would be substantial and incomparable to the rate at which it is generated. This thesis also reinforces the additional assistance to biologists in solving these problems. Hence, a system capable of helping biologists to monitor fish behaviors is needed.

Computer science areas, such as computer vision and machine learning, have evolved in recent years which allows the implementation of a system in a cheap way, without requiring more expensive technology (e.g. acoustic technology). For this reason, this thesis is motivated by the use of these technologies at the service of the needs of biologists and Lisbon Oceanarium, in order to help monitoring the aquatic environment. Fish habitat may reflect some challenges, namely: the depth of the tank which can affect the notion of movement (speed, curvature, etc); species variability, which implies different abnormality definitions and feeding methodologies; the size of some tanks, such as the main tank of Lisbon Oceanarium, making it difficult for the camera to cover all its range and consequently limiting the space in which fish are detected; presence of habitat components such as rocks and fauna, which can interrupt the fish's tracking.

2. Objectives

This thesis 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 objective is to develop a system capable of helping biologists, managing to produce a set of key episodes to be analyzed. This essentially includes three behaviors: abnormal, interesting and feeding.

One of the factors that can influence the definition of interesting moments and what might be abnormal is the species. The variability that exists in the oceanarium causes difficulties because it is possible that a given criterion is a factor of interest for one species but not for another. As a result, we decided to focus on sharks and mantas due to the fact that they are considered a focus species for biologists, and also because they have similarities in these behaviors. Additionally, they are visually distinct from the others. These species inhabit the main tank, which is the oceanarium's tank with the largest area, which further enhances the complexity of the task of analyzing behaviors only using human vision.

Typically, these focus species do not have major changes in their movement. For this reason, the following criteria can be indicators of an episode of interest: static for a significant time, direction and speed sudden changes, high speed, too many direction changes, manta rays swimming on the bottom, and sharks aggregation. Figure 1 illustrates an example of an interesting episode: a shark with sudden direction and speed change. An identification of an episode of this nature, by one of these indicators, should provide an alert enabling the biologist to analyze it in more detail. Within the interesting episodes, a special behavior was defined as abnormal whose definition is the deviation from normal considering all the detected samples from a given species.

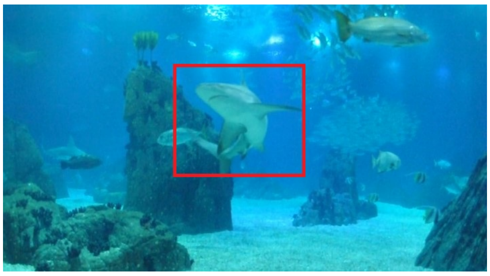

Figure 1: Example of an interesting episode: direction sudden change

Similar to episodes of interest, feeding is also similar in these species. During this period, they tend to aggregate in the feeding area, which does not normally happen except during this behavior. The manta rays do not usually frequent the bottom of the tank, but they are feed in this area through divers (Figure 2). On the other hand, sharks are fed through sticks with food at their tip, closer to the surface (Figure 3).

Finally, in terms of conditions, the system falls within the computer vision field. Several videos of the main tank were filmed in the Lisbon oceanarium, focusing on these behaviors, and using a camera placed outside the tank and in a static position. The developed system was implemented and evaluated using these videos.

Figure 2: Bottom feeding

3. Related Work

Previous Projects

Our project is being developed following previous projects [6, 11]. Castelo et al. explored the problem of detection, tracking, and classification of aquatic species within the scope of the Lisbon Oceanarium. Detection used background subtraction methodologies. Using the set of blobs identified in two consecutive images, the association is made through the features of these regions, both the position of the centroid and the predominant colors.

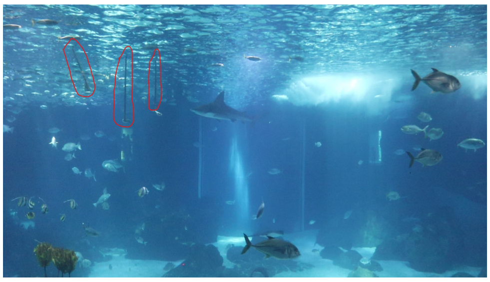

Figure 3: Surface feeding

Santos et al. [11] aimed at improving tracking. The idea was mainly adding robustness to tracking problems: color history to overcome histogram corruption when two fish regions overlap; temporary tracks, that define tracks with low lifetime, to avoid noisy tracks propagation; motion prediction, using Kalman Filter, to recover from missing detection/associations.

Detection Tracking and Classification

In the [9, 14, 17, 15] approaches, background subtraction methods are used. However, an application in [9] allows the user to choose the highest contrast color plane, according to the Red-Green-Blue (RGB) color plane, instead of using the grayscale image. The methods in [17, 15] use the Multi-Scale Retinex contrast enhancement algorithm before background subtraction is applied. In [12] and [1], two different approaches are used to detect fish present in a given image. The first uses a Gaussian Mixture Model while the second uses a YOLO network.

Regarding tracking on consecutive images, in [1] it is used an approach based only on distance, in [12] the Adaptive Mean Shift algorithm, and the project described in [7] proposes the innovative idea of identifying and tracking sharks based only on the characteristics of their dorsal fine as if it were a fingerprint. Two convolutional networks (CNN's) are used: one that detects sharks present in a given image and the second that detects the region of the shark's dorsal fine, given its bounding box.

Research described in [12, 8, 10] also address classification. In [12], each fish is represented by a vector of texture and shape features, and a linear discriminant is used. Following a similar approach, the project [8] uses the size of the fish together with the size of the different fines (anal, caudal, dorsal, pelvic, and pectoral) and a Support Vector Machine (SVM). Finally, [10] uses deep learning for classification between different species. A convolutional neural network (CNN) is trained using a very large dataset of sample images for several species.

Abnormal Behavior Detection

In [3], a rule-based method is proposed to filter out normal trajectories. A series of motion rules are applied in a cascade form, and the trajectories not filtered by any rule are considered abnormal. The method defined in [14], also resorts on motion, but it aims at detecting special events in schools, instead of focusing on certain fish individually. To do so, the kinetic energy is calculated and it is verified if this value is higher than a given specified threshold. It combines two measures regarding the motion vectors, the velocity and the direction angle: $E_{kn} = (D+100)^2 \times (-E)$ where D is the dispersion regarding the motion vectors and E the dispersion concerning velocity in relation to turning angle.

The task of detecting trajectories related to something abnormal can also be seen as a problem of outlier detection or even a classification problem. In [12], a clustering algorithm (IKMeans) is applied over the entire set of trajectories and the clusters with few samples are considered to be trajectories of interest (when compared to the total number). Clustering is also used in the method described in [4], but to train a hierarchical classifier. The [1] approach also trains and evaluates various classifiers to classify trajectories as normal or abnormal. However, what is passed as input in this method is an image with all the trajectories of the fish detected in a 10s window, instead of motion feature vectors. Several models were evaluated, namely: Naive Bayes, K-Nearest Neighbors, Linear Regression, and a Random Forest.

Activity Recognition

Papadakis et al. [9] developed a system capable of monitoring a set of fish tanks simultaneously. The project aimed to observe fish behaviors in the following environment: a net was placed in each tank to separate its area into two zones, and all fish were placed in only one of the regions. The tanks had different fish densities and the net in different states. The focus behaviors were inspection and biting.

The approaches in [5, 13] used an accelerometer on each fish of interest, and certain activities were detected based on the collected values. Broell et al [5] (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. [13](2019) solves a similar problem but focuses on activities related to sharks: swimming, resting, feeding, and non-deterministic movement. Similarly 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 are trained, more specifically convolutional neural networks.

Zhou et al. developed several projects [17, 15, 16] within the scope of the feeding activity. Initially, in [17], the goal was to detect the feeding activity through the analysis of the level of aggregation of the school, since it is usually higher during this period. In [15], one more index is used: Snatch Intensity of Fish Feeding Behavior (SIFFB). In this method, it is 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 [16], an innovative idea is described to identify the appetite of the fish present in a given image. A convolutional neural network (CNN) is trained based on several images at different levels of appetite.

4. Implementation

Pre-Processing

Trajectories are not perfect. Detection and tracking algorithms can also have errors. It is possible to miss the detection of some fish in a given instant, and it is also possible to fail its tracking especially if the fishes are very aggregated. In the pre-processing phase we try to overcome some of these problems.

An interpolation was the applied solution to fill the gaps that a trajectory may have. Two types of interpolation were implemented: linear as used in work [4], and also newton. This is applied for both axes independently: x position and y position. Figure 4 illustrates an example of the application of the different methods of interpolation. The generated synthetic gap is surrounded by red dots. We can verify how each of the approaches predicts the missing position points. As expected, the newton interpolation takes more into account than the two edge points, since the coefficients are calculated with Newton's divided differences, so the predicted line is not straight. The downside of this polynomial is the Runge's phenomenon which can be slightly attenuated using the nearest points of the gap as interpolation points.

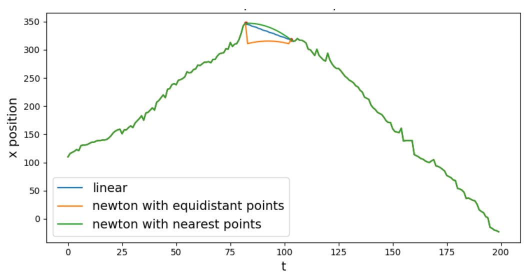

Figure 4: Interpolation methods comparison

The trajectory points can have noise due to instability on the detection task. Most of the time, the points are characterized by the center of the bounding box detection and this can be noisy because of the image plane and fish aspect ratio changes. One of the possible solutions is the application of moving average techniques to smooth the position time series. The exponential moving average was the chosen filter, so closer data points can have a higher impact. Figure 5 also complement this problem in a trajectory path visualization. The left frame in the figure draws the trajectory using the original positions. On the other side, the right frame draws the trajectory after applying the exponential moving average filter through the x-axis and y-axis positions. The trajectory after the filtering process is significantly more smooth, which will reflect also on the feature extraction phase, and consequently in the following behavior detection processes.

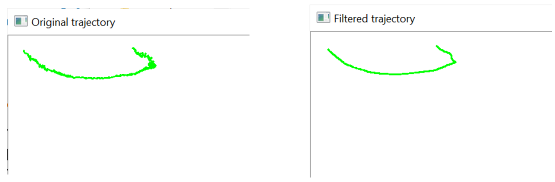

Figure 5: Filtering illustration: path before and after the filtering process

In the context of our domain, trajectory segmentation was a block that was essentially used as preprocessing to identify **interesting episodes**, so the different partitions could be classified instead of the whole trajectory. It is derived from the application of the Douglass Peucker algorithm regarding the position, speed, and angle time series of a given trajectory. Figure 6 illustrates a frame with the different trajectory segments drawn with a different color, resulting from the trajectory segmentation. Using this method, we probably won't cut the trajectory in the middle of some important moment.

Figure 6: Trajectory segments using position epsilon=30, speed epsilon=2 and angle epsilon=50

Feature Extraction

Most of the features that are extracted are also described in the work in [4]. Overall there are features related to the motion, fish orientation, trajectory sparsity, and regions analysis. In total, the following features are calculated:

- velocity that can be calculated as $v = \frac{\Delta p}{\Delta t}$, and described as the position variation over time;
- acceleration that can be calculated as $a = \frac{\Delta v}{\Delta t}$ and described as the velocity variation over time;
- turning angle that can be calculated as $\theta = \arctan(p_t - p_{t-1})$, where $p_t - p_{t-1}$ is the motion vector;
- curvature that can be calculated as $k = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}$, where x', y' and x'', y'' are the first and second derivatives respectively;
- centered distance that can be calculated as $cd = d(p_t, \mu)$, where p_t is the position on instant t, and μ is the mean position of the trajectory;
- normalized bounding box that can be calculated as $nbb = \frac{w}{h}$, where w and h are the width and height of the bounding box respectively.

4.1. Anomaly Detection

Clustering-based

This approach follows the logic also used in the work [4], but instead of using the positioning to group the trajectories, we use the features described in the previous section. First, all the trajectories are transformed into vector format and a clustering algorithm is applied using all these vectors. Looking at the output clusters, the groups that have few samples when compared with the total number of samples, are considered outlier groups. Two different clustering algorithms were applied: KMeans and DBScan. KMeans has the disadvantage of needing the specification of the number of clusters, so a specification of a percentage threshold is needed, to label each cluster as being or not an outlier cluster. On the other hand, DBScan has already the notion of outlier.

Switching Vector Model

The switching vector model was used to model pedestrian trajectories in work [2]. In this project, we try to use it to model the fish trajectories. In summary, it tries to model the different motion vectors for each image plane region, and also the transition between those fields. The image plane is described by a grid of nodes, where each region contains k motion vectors (as the ones drawn in Figure 7) and a transition matrix. It is essentially characterized by the grid size, the number of fields, and two parameters that impact the fields update during training: δ which defines the neighborhood distance, and α that defines how attenuated distant vectors are.

In the anomaly detection task, we can specify a minimum threshold for the joint probability. Given a new trajectory, the joint probability is calculated and if its value is lower than the specified threshold it is considered as an outlier.

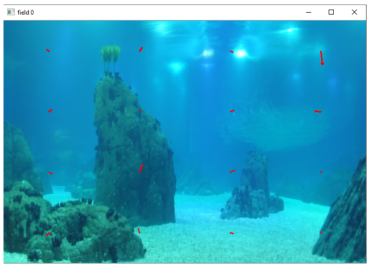

Figure 7: First field vectors after training

4.2. Interesting Episodes Detection

Machine Learning Models

Taking into account all the trajectories, these were transformed into a vector format and passed as input to a pre-processing pipeline before being transmitted to the model itself. The goal is to be able to learn interesting trajectories feature patterns, to have the ability to generalize to other not known trajectory samples. Figure 8 complements this explanation. We can verify the three described phases: extraction of features from the trajectory, vectors pre-processing phase, and finally the model training phase.

Several models were experimented. We tried to explore the different classifier families: decision trees and random forest from a logic perspective, naive bayes from a probabilistic side, and support vector machines and k-nearest neighbors from a similarity view. Relatively to the pre-processing approaches, several methods were experimented to improve the model performance: normalization, class balance, principal component analysis, ANOVA feature selection, and correlation removal. Almost all these methods are already implemented, so we resorted to the *sklearn* library which is one of the known names on the machine learning libraries field.

Rule-based

The rule validation is done through the feature time series that are extracted from the trajectory: velocity, acceleration, turning angle, curvature, normalized bounding box, and regions information. Each rule is defined by an interval of values and

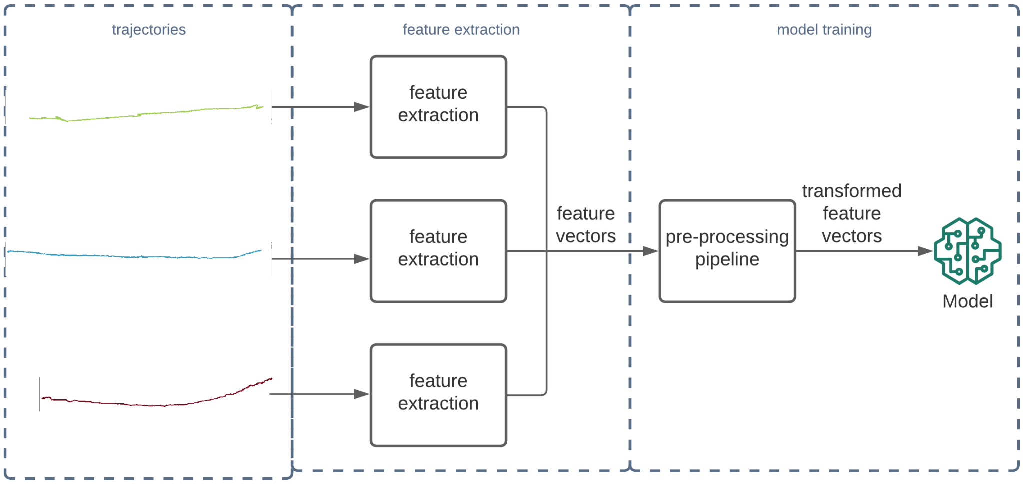

Figure 8: Machine learning model training process

a minimum duration. The feature value has to be within the specified interval for at least n consecutive frames, where n is the specified minimum duration. When this is detected, a highlight moment is saved. Figure 9 shows an example of a highlighted moment detected on the turning angle feature for a given trajectory, using an interval value of 60 to 120, which reflects a significant ascent to the surface.

As an additional feature, it is also possible to specify a single threshold and use a minimum or maximum function. A single threshold is also what is needed relatively for region-based features. We can specify a maximum duration allowed for a given region. For example, manta rays do not usually frequent the bottom part of the tank, so we could specify a rule to trigger an event when that happens. The same logic can be applied to the transition between regions.

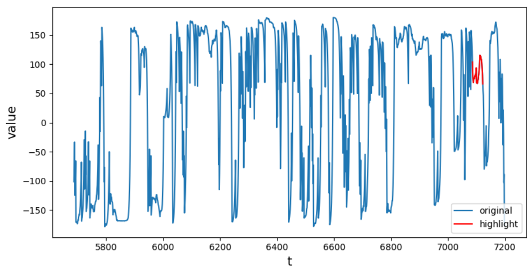

Figure 9: Highlight moment regarding turning angle feature

4.3. Feeding Period Detection

Convolutional Neural Network

Resorting on Convolutional Neural Networks (CNN) is one of the possible solutions to detect feeding periods. Using this type of model, the frame can be directly passed as input. Internally, a feature vector will be calculated using convolutional and pooling layers. This vector is then passed to a set of full connected layers to make the decision of being a feeding frame or not.

Several training videos were filmed for this behavior, so all frames from these videos can be used to train a model of this nature, so it can be able to learn image patterns for each of the classes. The work in [16] also uses a neural network, but with a sensibly different goal: measure the feeding hungry intensity. However, its architecture was used as baseline for this project.

Motion-based Approaches

The feeding activity is characterized by the change on motion intensity around the feeding region. An alternative approach to the machine learning method is to be able to measure the amount of motion, and verify if this difference is visible between the feeding period and the normal period. We implemented two different ways of measuring the level of motion: apply consecutive frame subtraction and a threshold binarization to detect the number of active pixels (Figure 10); apply optical flow algorithm and calculate the average magnitude of the vectors (Figure 11).

Aggregation-based Approach

Similar to the logic around the motion measurement, fishes also tend to aggregate more during feeding. To measure the level of aggregation of the detected fishes in a given frame we use the approach explained in work [17], which applies the delaunay

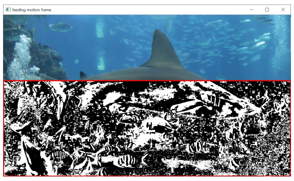

Figure 10: Active pixels

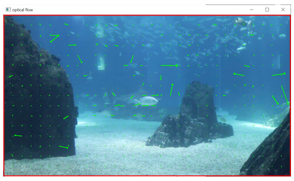

Figure 11: Optical flow

triangulation on the fish's centroids that will form a triangular mesh between those points. However, we do not take into account fishes that seem to be far way from the center of mass (median position of all points). These outlier fishes are considered to not being interested on the feeding activity, which can be a motif of alert and a advantage of this approach. Figure 13 illustrates the resulting mesh and outlier from the application of this method on the frame of the Figure 12.

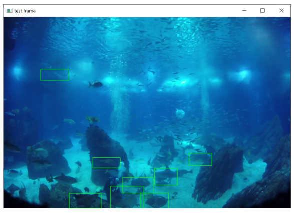

Figure 12: Feeding image

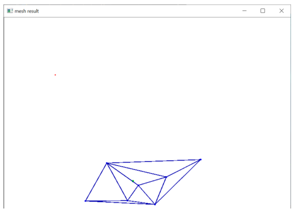

Figure 13: Mesh

5. Results

5.1. Anomaly detection DBScan vs KMeans

To choose the best pipeline to be used with DBScan, a set of experiences using different pre-processing methods was made. The pipeline with the better performance considering silhouette metric: a standard scaler as the normalizer, followed by a feature selector using the ANOVA f-value and finally applying the DBScan cluster algorithm. Resorting on this pipeline, a silhouette of nearly 0.72 was achieved which was significantly higher than the ones obtained utilizing the other pipelines. Only one outlier was detected from the set of trajectories, as we can see in Figure 14 and 15: a shark that does a considerable curvature to the bottom and suddenly changes direction turning back to the origin point of detection. However, if we relax the ϵ value, more samples can be seen as outliers but the silhouette value also decreases.

The DBScan algorithm has the advantage of not being necessary to specify the number of clusters, unlike the KMeans algorithm. Although it is not possible to observe a standing out an elbow, a value of k whose cohesion value starts to stabilize, we could verify that from a value of approximately seven clusters the cohesion starts decreasing at a slower pace. It is possible to visualize the trajectories from clusters 0 and 2 in Figure 16. As we can

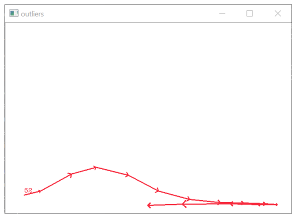

Figure 14: Identified outliers using DBScan

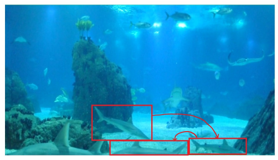

Figure 15: Sequence of frames of the detected outlier

see, there is a very pronounced pattern in each of the clusters. Group 0 tends to keep the same direction (constant turning angle) during all the trajectory. This can also be concluded from Figure 17, which presents the set of features that most differentiate from the other clusters. Group 2 contains the identified outlier using the DBScan algorithm. It also contains two additional trajectories which have the same motion behavior: a sudden direction change turning back to the origin of the trajectory. Although these resulting clusters seem to represent interesting groups according to motion patterns, the overall performance (silhouette) value was sensibly 0.30, thus we believe that the DBScan algorithm is more appropriated for the outlier detection task.

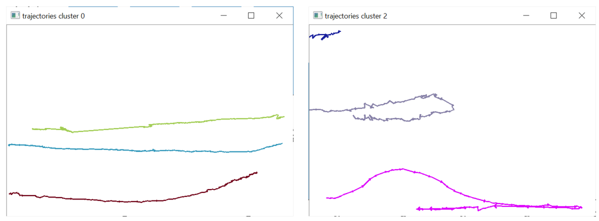

Figure 16: Resulting trajectories from cluster 0 and 2 respectively

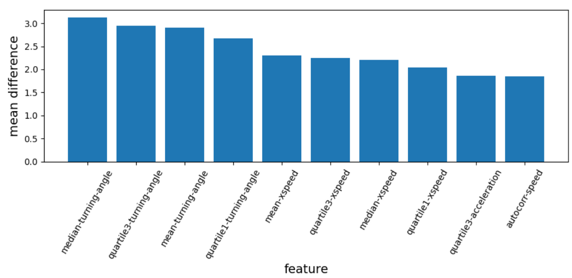

Figure 17: Most characterizing features of cluster 0

5.2. Interesting episodes detection

In the context of Machine Learning, interesting episodes detection task can be seen as a binary classification: it is either classified as interesting or not. Several models, and from different classifiers families, were evaluated to verify which one fits better on this type of data: Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Decision Tree, and Random Forest. Additionally, because of the impact that the pre-processing methods and hyperparameters can have, different pipelines were tested, and the respective tuning for each model applying a grid search.

Precision and recall can also have an impact on the model choice process. In our context, it would be possibly overwhelming for biologists to analyze several false positives. However, it can also be a danger to let some interesting moments pass by. It is a trade-off that has to be agreed to biologists' needs. Figure 18 draws the precision-recall curve for each of the models. This also complements the information from the previous table and consolidates the fact that the Support Vector Machine works better with this data. Table 1 shows the area under the curve values, which illustrate the models that have a better precision-recall overall trade-off.

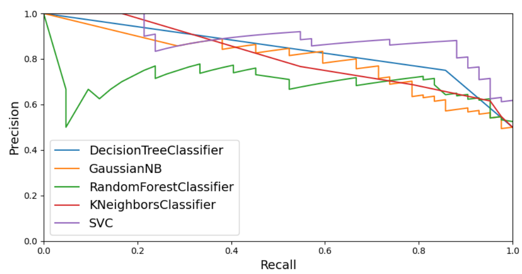

Figure 18: Precision-Recall curves for the different Machine Learning models

Model	AUC
SVM	0.89
KNN	0.81
NB	0.8
DT	0.84
RF	0.7

Table 1: Area Under the Curve (AUC) values

Multiple-model and Segmentation impact

The previous evaluation considers models that are trained using all the samples for both focus species (shark and manta-ray). This experiment aims verifying if separating on different models for each species produces better results. Figure 19 reflects the obtained results. Better performance was achieved modeling using data from both species instead of separating on different models. However, it was not a significant difference and the model for manta rays got a better recall value although the precision has been harmed.

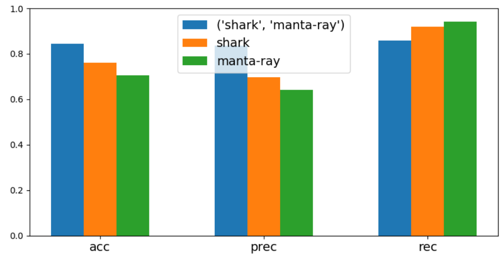

Figure 19: Models performance modeling by species

Segmentation could also be useful for interesting episodes identification. The idea was to segment all the trajectories and to perform classification on each segment instead of the trajectory as a whole. The performance difference was similar to the one observed on modeling by species results: better performance was achieved without using segmentation but the difference was not significant (Figure 20).

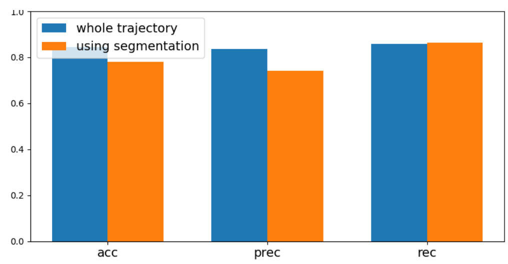

Figure 20: Model performance using segmentation

Switching Model Adaptation

The models that best fit each of the classes were used on the interesting trajectories detection task. Table 2 illustrates the number of correctly classified trajectories on each group. As we can see, the results show us a good precision value but a significantly low recall value, which means that this approach gave a considerable amount of false positives, once the overall accuracy stayed by 58%.

group	correctly classified
normal	17 of a total of 42 (40%)
interesting	24 of a total of 29 (83%)
both	41 of a total of 71 (58%)

Table 2: Vector swicthing model evaluation results

5.3. Feeding Period Detection

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-based methods, these approaches were evaluated in a different test set (bottom feeding). Both approaches gave good results according to accuracy, precision, and recall metrics having been able to reach 92% and 96% of accuracy respectively (Table 3). However, the optical flow method classified most of the frames 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.

Two additional experiences were made: evaluate the motion-based approach using the region definition ability and the CNN in a video filmed with a different camera which was in a different perspective. As we can see in Table 3, the region was not the problem for the miss classification of some of the frames because the same results were sensibly observed. Also, we can conclude that the initial trained model to bottom-feeding classification task could not generalize to another perspective. Contrary to what was observed on the optical flow approach, almost all the frames were classified as feeding frames.

As a final analysis, we also tried to understand if the errors that we were getting were close to the feeding state transition. We could conclude that 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, once the frames that are still close to this state change are still classified as feeding frames.

6. Conclusions

Aquatic life has a lot of density. It can be hard for biologists to track all the fish activity. 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 focused on abnormal behaviors, feeding periods, and interesting moments of sharks and manta rays, due to their importance to Lisbon Oceanarium. Keep track of abnormal behaviors can be mainly important to conclude about diseases, water quality problems, or poor habitat integration. Additionally, feeding can also impact water quality, and it is important to control food waste that can lead to costs or underfeeding that can lead to aggressive behaviors.

In this project, it was developed a system capable of detecting these behaviors in the main tank of the oceanarium, and as described, focusing on sharks and manta rays. To detect abnormal and interesting moments we extract several features from each fish trajectory and we try to model this kind of behavior through machine learning models, and a switching vector model. Additionally, the system also has the ability to define a set of rules based on those features. Relatively to feeding, there are several factors 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 the system achieved good performance metrics. The detected outlier seems to also correspond to an interesting moment, given its sudden changes in direction, but it would be good to have a more robust set of trajectories. In terms of its evaluation, a clustering algorithm based on density is considered more appropriate for this task. Regarding interesting moments, all the training models round the 80% of accuracy. The usage of segmentation or multiple-model training did not make any significant difference in terms of achieved results, and the switching vector model adaption obtained poor results regarding this task. Finally, for the feeding behavior, convolutional neural networks could model feeding image patterns and achieved 90% 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%.

There are several topics that could be explored in the future. One of them could be focusing on the clustering approach, but using a fish trajectories big data dataset. Additionally, all the methods suffer from image plane dependency and camera perspective, especially for the trajectories definition which is a sequence of 2D position points. Another future problem could be focusing on fixing or decreasing this dependency. Results coming from the convolutional neural network approach can be hard to understand, it would be also interesting to try to understand what image patterns, and image regions, are characterizing each of the classes. Finally, it would be extremely useful to develop an application to be used by biologists.

References

[1] O. Anas, Y. Wageeh, H. E.-D. Mohamed, A. Fadl, N. ElMasry, A. Nabil, and A. Atia. Detecting abnormal fish behavior using motion trajectories in ubiquitous environments. *Procedia Computer Science*, 175:141–148, 2020.

[2] C. Barata, J. C. Nascimento, and J. S. Marques. Improving a switched vector field model for pedestrian motion analysis. In *International Conference on Advanced Concepts*

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
active pixels (using region)	0.96	1	0.94
optical flow (using region)	0.4	0.7	0.06
CNN using go pro dataset	0.27	0.28	0.86
CNN (surface feeding)	0.71	1	0.53

Table 3: Bottom feeding test set results

for Intelligent Vision Systems, pages 3–13. Springer, 2018.

[3] C. Beyan and R. B. Fisher. A filtering mechanism for normal fish trajectories. In *Proceedings of the International Conference on Pattern Recognition*, pages 2286–2289. IEEE, 2012.

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

[5] F. Broell, T. Noda, S. Wright, P. Domenici, J. F. Steffensen, J.-P. Auclair, and C. 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.

[6] J. Castelo, H. S. Pinto, A. Bernardino, and N. Baylina. Video based live tracking of fishes in tanks. In *International Conference on Image Analysis and Recognition*, pages 161–173. Springer, 2020.

[7] T. G. D. Coleman. *SharkID: A Framework for Automated Individual Shark Identification.* PhD thesis, California State University, Long Beach, 2020.

[8] S. Ogunlana, O. Olabode, S. Oluwadare, and G. Iwasokun. Fish classification using support vector machine. *African Journal of Computing*, 8(2):75–82, 2015.

[9] V. M. Papadakis, I. E. Papadakis, F. Lamprianidou, A. Glaropoulos, and M. Kentouri. A computer-vision system and methodology for the analysis of fish behavior. *Aquacultural engineering*, 46:53–59, 2012.

[10] D. Rathi, S. Jain, and S. Indu. Underwater fish species classification using convolutional neural network and deep learning. In *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–6. IEEE, 2017.

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

[12] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. J. Chen-Burger, R. B. Fisher, and G. Nadarajan. Automatic fish classification for underwater species behavior understanding. In *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pages 45–50, 2010.

[13] W. Zhang, A. Martinez, E. N. Meese, C. G. Lowe, Y. Yang, and H.-G. Yeh. Deep convolutional neural networks for shark behavior analysis. In *IEEE Green Energy and Smart Systems Conference (IGESSC)*, pages 1–6. IEEE, 2019.

[14] J. Zhao, Z. Gu, M. Shi, H. Lu, J. Li, M. Shen, Z. Ye, and S. 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.

[15] C. Zhou, K. Lin, D. Xu, L. Chen, Q. Guo, C. Sun, and X. 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.

[16] C. Zhou, D. Xu, L. Chen, S. Zhang, C. Sun, X. Yang, and Y. Wang. Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision. *Aquaculture*, 507:457–465, 2019.

[17] C. Zhou, B. Zhang, K. Lin, D. Xu, C. Chen, X. Yang, and C. Sun. Near-infrared imaging to quantify the feeding behavior of fish in aquaculture. *Computers and Electronics in Agriculture*, pages 233–241, 2017.