Fish Behavior Detection

Gonçalo Adolfo[1][97090]

Instituto Superior Técnico, Portugal
goncalo.adolfo@tecnico.ulisboa.pt
https://tecnico.ulisboa.pt/

Abstract. Automatically detect fish behaviors can be helpful to monitor fishes in tanks, which can save lots of time for biologists. In this report, several methods in the area of detection, tracking, classification, and behavior detection are reviewed and analyzed. A new system is proposed to detect fish behaviors of sharks and manta rays: swallowing air, feeding, lack of interest, and abnormal. It is composed mainly of a behavior detection module and a web application. Initially, this module will be only based on image processing and a set of rules and it is intended to be improved based on machine learning models. The evaluation of the system will take into account detection performance, significance to the user, and processing delay time.

Keywords: Behaviors Detection · Image Processing · Machine Learning · Fish

Table of Contents

1 Introduction

Within the scope of the collaboration with the Lisbon Oceanarium, there were developed several projects such as [13] with the aim of detecting, tracking, and classifying the different fishes among the distinct tanks. Each tank is a habitat for a given set of species, and for that reason, each one represents different challenges regarding the behavior variability. Also, the level of lighting is not constant thought the day and the various tanks do not get the same level of light due to their species and position in the oceanarium. Despite all these challenges, it is now possible to take the information that the given modules can give (fish label, motion track) and decrease the level of abstraction, focusing on trying to understand automatically fish behaviors.

Fish behaviors themselves can also 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 of something of interest to analyze and the known activities can help to understand fish patterns through the day (or other levels of granularity).

Feeding activity is considered especially of interest because of its impact on production costs and water quality. Underfeeding leads to aggressive behaviors 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 observer experience, which can be subjective and many factors can contribute to fish appetite: physiological, nutritional, environmental, and management.

Detection of an abnormal behavior could mean various scenarios: fish disease, problems in water quality, poor habitat integration, or poor integration with other species. Whatever the cause, the traditional way of taking these conclusions is based on visual inspection by marine biologists. This is considered, by many, very time-consuming and highly dependent on the biologist's experience. Another possible approach, more used in the context of open sea systems, is by analyzing footage taken by divers. This method is invasive and is therefore not able to capture behaviors considered normal. One possible solution to overcome these approaches could be the installation of several cameras in the different tanks. In this case, it would be possible to analyze in ”dead times”, instead of having to be done in person in that exact period of interest. However, the amount of data produced daily would be huge and probably only human effort would not be enough.

Biologists cannot be in the various tanks at the same time. Even if there are enough team members to cover the different analysis periods, it consumes lots of time which could be used in another task. Using underwater cameras, the rate at which the footage is produced would be probably greater than the rate at which it is analyzed by humans. Hence, there is a necessity to develop a system capable of helping biologists to monitor the behavior of fishes.

Computer science areas, such as computer vision and machine learning, have evolved a lot in recent years what allows the implementation of a system in a cheap way, without resorting to more expensive technology (e.g. acoustic technology).

1.1 Project Goals

The main objective of the project is to develop a system capable of helping biologists to monitor fishes. The target species are sharks and mantas as they are the ones that cause the least errors in the underlying modules: detection and tracking. The system must be able to detect certain specific activities as well as unusual behavior. According to our conversation with biologist Hugo Batista, it is substantially of interest to detect how many times the bull shark will swallow air at the surface, shark attack behaviors, and identification of fishes that have a low interest during the feeding period. The system should have an interface where it will illustrate the results (detections): activities of interest as well as abnormal behaviors. It must also have the ability to readjust based on the feedback given by the user.

1.2 Outline

In the first part of the document, section 2, several related projects are introduced and analyzed. Posteriorly, a new system is proposed to solve the project goals (section 3). Section 4 explains how the system will be evaluated and section 5 the time planning taking into account the modules that are necessary to develop.

2 Related Work

The identification of activities and the analysis of behaviors come mainly from the modeling of trajectories. Over the years several projects have been developed within the scope of trajectory analysis, and in different contexts: analysis of tourist activities as the example given in [10] by inserting semantic information; modeling of pedestrian/car/cyclist trajectories based on Markov models [2]; methodologies without a concrete domain [11]: clustering trajectories based on shape; other domains: zoology, meteorology, sports game analysis, etc.

In the context of this project, the domain of interest is aquatic species. The related work presented in this section is divided as follows:

- necessary modules for the acquisition of trajectories: fish detection, tracking by several consecutive images and, in certain scenarios, classification of the image of the detected fish;
- detection of abnormal behaviors: many of the applications developed are intended to help biologists, in that sense the detection of something abnormal is highly useful;
- recognition of certain activities to assist monitoring (feeding, swimming, resting, among others).

2.1 Detection Tracking and Classification

This project is being developed following previous projects like [6]. Castelo et al. explored the problem of detection, tracking, and classification of aquatic species within the scope of the Lisbon Oceanarium. Detection is done using background subtraction methodologies. In this type of approach, a background image is estimated by averaging the value of sample images. Subtracting a certain image from the calculated image allows to identify the motion pixels, and using a given threshold to obtain a binary image, where the active pixels are the motion pixels. Through this, it is possible to label the regions that are connected, these being considered the regions of the fish detected at that moment. Using the set of blobs identified in two consecutive images, the association was made through the characteristics of these regions, concerning the position of the centroid and the predominant colors.

The detection module is not perfect, so associations can be lost. To try to deal with this problem, an algorithm (Kalman Filter) is inserted to predict the next positions of a given fish and a heat map is calculated around the estimated position. If there is a region with a similar area and similar colors, it is considered the region of the fish in question, although it has not been detected as a movement zone. In this project, the method used to classify depends on the environment. In the case of being the coral tank, the class associated with a given fish is given by the one with the most overlap between color histograms. On the other scenario, in the main tank, since the fish have very similar colors to the background color, the histogram of oriented gradients (HOG) was chosen: $\theta(x,y) = \arctan(\frac{G_y(x,y)}{G_x(x,y)})$ where G_x and G_y are the image gradients for each dimension. Several linear regressions are used to reconstruct the histogram and the class associated with the model weights that best reconstruct is the one assigned.

In the [9], [16], [19] and [17] approaches, background subtraction methods are also 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 gray-scale image. Additionally, the background image is only updated every 100 seconds. The methods in [19] and [17] use the Multi-Scale Retinex contrast enhancement algorithm before background subtraction is applied. This algorithm enhances the contrast between regions of the image: the original image passes through a Single Scale Retinex: $SSR(x,y) = \log(I(x,y)) - \log(G(x,y,\alpha) \circledast I(x,y))$ where I is the original image and G a 2D Gaussian filter with α scale, $\circledast$ represents the convolution operation. Multi-Scale Retinex is defined as the application of several SSR's with different scales in the Gaussian filter. The work explained in [16] also defines a methodology, an alternative to background subtraction methods, based on Optical Flow, since the focus in this paper is not necessarily on detecting fish regions, but on the analysis of the school itself. A particle grid is placed over the image and between consecutive images, its movement is estimated. The detected movements are assumed to be the movements of the fish. Although this method is not subject to problems such as changes in luminosity, which have an impact on the background and consequently on the obtained motion regions, it presents greater temporal complexity for the motion estimation and is also not resilient to problems such as small movements of background objects.

In [13] and [1], two different approaches are used to detect fishes present in a given image. The first uses a Gaussian Mixture Model while the second uses a YOLO network. The idea behind the first method is to model the intensity of each pixel by a set of Gaussians, and the training works as follows: for a new pixel value, the parameters (average, standard deviation, and associated weight) of the Gaussian that best represents its intensity are updated. Given a new image and the set of Gaussians for each pixel, all pixels in the new image are classified as being part of the background or not. The advantage of this approach is the ability to model small movements of background objects.

The other method (YOLO) is part of the recent field of convolutional neural networks. Despite being applied in several domains, their fame comes mainly in the context of images due to their ability to model structure. In an R-CNN network, the basis is the use of a process called Selective Search: several bounding boxes (of different sizes) are placed over the image domain and are grouped according to similar characteristics, of adjacent bounding boxes, generating a set of proposed regions that serve as input to another convolutional neural network (CNN). The YOLO algorithm follows an approach in which it obtained better training and classification times: it starts by dividing the image into a grid of SxS regions and m bounding boxes are generated in each region; for each bounding box, the network returns as output the probability of belonging to each class and only those with a value greater than x are considered to locate the object. In this sense, with the YOLO algorithm, a single CNN is used to predict bounding boxes and calculate the respective probabilities. The use of this algorithm is more resilient to common problems such as changes in the background and occlusion of fish, but it does not return the fish mask but its bounding box. However, in general, it presents better detection results due to its robustness.

Regarding the tracking module between consecutive images, [1] uses an approach based only on distance while [13] uses the Adaptive Mean Shift algorithm. In the first approach, given a new $f+1$ image, it is checked if there is any close detection in the previous image f. If so, the same identification is associated, otherwise, a new identification is associated. On the other hand, in the other method (AMS) the idea is to move the region of interest (in this case of a certain fish) in the direction of the highest density area until it converges. For this, the binary image of the current instant and the region of interest detected in the previous image is used. With this, the centroid of the zone is calculated given the binary image, the zone is moved towards the centroid, and so on until the successively obtained centroids converge. Finally, it is verified in which blob the final centroid is inserted and the same identification is given. Both approaches do not correct occlusion problems. However, using an image such as the histogram back-projection image makes the second method more robust to this type of problem.

The works described in [13], [8] and [12] also address the problem of classification. In [13], each fish is represented by a vector of texture and shape features. Regarding the texture, statistics of the gray histogram and the co-occurrence matrix are extracted: contrast, energy, correlation, etc. The shape is defined by the histogram (with 30 cells) of the discrete Fourier transform (DFT) on the contour points and the first 20 local minimum of the Curvature Scale Space (CSS) image. The set of example feature vectors is passed to a Principal Component Analysis (PCA) algorithm, which allows reducing the dimensionality by projecting the points on the axes of greater variance. The vectors, of reduced dimensionality, are finally passed as input to a linear discriminant.

The project [8] uses the size of the fish together with the size of the different fines: anal, caudal, dorsal, pelvic, and pectoral. As a classifier, a Support Vector Machine (SVM) is used, being more robust than a linear discriminant since it calculates the hyperplane that best separates the different classes, in other words, with the optimal margin. The real estimation of the size of the fines can be a challenge, but it can be done through semantic segmentation (Mask Neural Networks), as described in [14]. Finally, [12] 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. This type of methodology has the disadvantage of needing a lot of examples, given the high number of parameters. The described method has the particularity of receiving as input the binary image of the fish concatenated with the 3 Red-Green-Blue (RGB) planes. This is calculated by applying the OSTU algorithm to the gray image, to discover the threshold that better divides the intensities present in the fish image (less intraclass variance and greater interclass variance). To enhance the fish region, morphological operations are also applied (opening and closing). The results obtained with deep learning were better compared to previous approaches, according to the accuracy obtained in test sets.

The project described in [7] comes with 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. To not take into account the background pixels of the fine region, the K-Means algorithm is applied. The part of the dorsal that presents greater disturbances is the side that best identifies the shark. To obtain it, first, it extracts the triangle that best characterizes the fine contour and the midpoint (of each side) that is farthest from the real contour is the point belonging to the edge of interest. Then, k key points are identified on this side of the fine contour, in which they will be defined as the boundary descriptor. These key points are the points where there are the biggest disturbances (salient locations) given by: $p(u_i) = \left|D(u_i, m, \sigma) - D(u_{i+1}, m, \sigma)\right|$ where u is a contour point and D a corner response function being m the multiplication factor and σ the standard deviation of a Gaussian filter. Since each shark is defined by a vector of K points, these are first normalized to be invariant in scale and orientation and passed to an algorithm based on graphs, given two sets: one from the current frame and another from the previous frame (or even other images). A connected full graph is built, and the weights between nodes are given by the Euclidean distance. Finally, a minimum-weight perfect matching algorithm is applied to obtain the matches. Under this approach, it is important to highlight the high resolution of the used images, which for certain projects may not be possible.

2.2 Abnormal Behavior Detection

Non-Machine Learning Methods Beyan et al. [3] proposed a method to filter a set of trajectories considered normal from a total set of trajectories. This mechanism is based on the application of several rules that characterize normality. They are applied in the form of a cascade: the total set of trajectories is passed as an entry to the first rule, which in turn filters out a certain part (according to that rule) and the rest are passed as an entry to the next. Trajectories that are not filtered by any of the rules are considered abnormal.

The fish, when caught by the camera's ray of vision, do not have many changes in direction. This is the basic idea of the proposed rules. It is argued that usually there are no more than three changes between primitive movements: horizontal movement, vertical movement, and stationary movement. Each of the rules models the possible set of moves giving a total of $3+(3\times2)+(3\times2\times2) = 21$ rules. Each filter, and for each of the primitive movements, is also associated with the concept of "search area" (estimated in the training phase and analyzing example trajectories). The trajectory must satisfy the following conditions:

- all the centroids of the bounding boxes of the fish, during a certain movement, must be within a certain area ("search area");
- the centroid in a given frame f must be within an area given the position detected in frame $f-1$.

The results illustrate that it has a low false-positive rate. However, most of the trajectories did not pass any rule, giving a high rate of false negatives. That said, it has not shown good results when it comes to detecting abnormal trajectories but it can work as a preliminary method for another approach.

The method defined in [16] has the objective of detecting special events in schools, instead of focusing on certain fish individually. The underlying idea is that the fish in a school does not disperse much in terms of movement. The proposed kinetic energy model, according to physics, represents the energy that a given object presents given its movements. It combines two measures of the dispersion that exist concerning 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. Given several example images, a threshold is estimated for each measure: dispersion of movement, dispersion of velocity/angle, and energy value. When these conditions are met for n consecutive images, an alarm is given.

The features of the school are modeled by two histograms: a 2D histogram of the motion vectors (in x and y), and a 2D histogram of the velocity concerning the direction angle. These are normalized and the entropy value is extracted: $E = -\sum_{i}^{m_1}\sum_{j}^{m_2} p_{xy}(i,j)\log_2(p_{xy}(i,j))$ where p_{xy} is the probability in a given cell, m_1 and m_2 are the number of bins for each feature. This measure is associated as a measure of uncertainty/chaos. The higher the entropy value, the more dispersed the movement features of the school fish are, indicating something special. This approach, as it uses background subtraction methodologies to detect fish, is very sensitive to occlusions and unions of different fish regions. For this reason, it also proposes a method based on Optical Flow, as explained in the previous section.

Machine Learning Methods 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 the [13] approach, 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). This algorithm is applied to each set of trajectories of each species since different species have different motion patterns.

Trajectories can contain a large number of points, and not all of which are necessary to describe correctly, which can lead to a long processing time. To try to solve this problem, before being used as input to the clustering algorithm, they are pre-processed with the Douglass-Peucker algorithm. This presents the goal of removing points from the trajectory that has no impact on its original form. This is done by recursively dividing the trajectory. It consists of the following steps:

1. mark first and the last point as "to keep";
2. obtain the furthest point from the line segment AB, with A initially being the starting point of the trajectory and B being the last point;
3. if the distance is greater than ϵ then that point is marked as "to keep" and the algorithm returns to step 2 but with the line segments A-FurthestPoint and FurthestPoint-B. Otherwise, all points in the middle are discarded.

The IKMeans clustering algorithm itself also tries to deal with this issue. It takes advantage of the Haar Wavelet decomposition, which allows representing a time series of the trajectory in lower resolutions, preserving the original information/form. The idea of the IKMeans algorithm is to apply the traditional KMeans algorithm in the set of time series of the trajectories at increasing levels of detail. This approach usually converges soon at low levels of resolution. However, even if that does not happen, the centroids of one level will be the centroids of the next level, allowing them to converge more quickly in the next level. Overall, better results are achieved in terms of clusters quality and processing time than the traditional algorithm.

The method described in [4] also uses a clustering algorithm but to train a hierarchical classifier. The points at a given level of the tree are obtained as follows:

1. feature extraction (subset) and Principal Component Analysis (PCA) application;

2. clustering and outlier detection;
3. feature selection: increase features subset;
4. return to step 1.

Despite being a model whose training is based on clustering, the example trajectories must have an associated class. The previous points are repeated until eventually, the accuracy goes down with the increase of features space. When this happens, the clusters obtained that are pure (only normal trajectories or only abnormal trajectories) are maintained at this level. The trajectories belonging to the remaining clusters are used to estimate the clusters of the next level of the tree and, for this, the steps mentioned are repeated but with different input. In this way, each level of the tree will have a set of associated pure clusters and considers different subsets of features. Given this tree the classification is made as follows:

1. for each level, the closest cluster to the new trajectory is obtained (given its feature vector for the level in question);
2. it is verified to which class each of the closest clusters are associated;
3. the new trajectory is classified as normal if there is a pure cluster of normal trajectories as the closest cluster (in at least 1 of the levels).

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. Several models were evaluated, namely: Naive Bayes, K-Nearest Neighbors, Linear Regression, and a Random Forest. Within the models described, the Naive Bayes probability model was the one that achieved better results. Since the input of these models is the image itself, the usage of deep learning was expected. However, the decision must have been derived from the reduced set of example data. Both approaches obtained similar values in terms of evaluation despite the [4] method being tested on more robust datasets, including datasets from other domains. Additionally, it takes into account that the trajectories may not be perfect: an interpolation is applied to derive certain missing points. The other approaches do not take this problem into account, as well as the fact that abnormal events can be quite fast. In this case, in longer trajectories, these events can be canceled by the normality of the remaining segments.

2.3 Activity Recognition

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

The inspection activity is detected when at least one fish approaches the net sufficiently and is counted by the number of images in which it occurs.

The midpoint of the network is manually marked as a reference point by the user. For each image, the horizontal distance between the center of mass of each fish and the reference point is calculated. Other statistical metrics are also extracted (to analyze certain correlations with the activities): center of mass of the school, an average of the horizontal/vertical distance, and average velocity. With this method, it was possible to obtain conclusions about the activities in the different conditions, given the simplicity of the environment (especially the shape of the tank and the fact that the network is static and marked by the user). Additionally, despite being defined in the article, it is not explicit how to detect the biting activity.

The approaches in [5] and [15] both place an accelerometer on each fish of interest, and it is based on collected values that they identify certain activities. Broell et al [5] (2013) detect the following set of activities: swimming, feeding, escaping. To this end, they propose a signal processing system, based on the analysis of time series characteristics related to the acceleration in each dimension. Based on sample data for the various activities, multiple features were calculated and in several domains: frequency (spectral and wavelet analysis), probability (mean, maximum, variance), and time (autocorrelation, integrals, and derivatives). The idea would be to identify which features can present identical values within the same activity but different between different activities. They managed to conclude the following set of points:

- the standard deviation feature of the magnitude of acceleration $MA = \sqrt{x^2 + y^2 + z^2}$ was the most efficient to distinguish between the swimming activity and the others, presenting significantly lower values;
- to distinguish between feeding and escaping activities, 6 features were used: 4 of them are associated with the comparison of values between the x and y dimensions (standard deviation, maximum value, acceleration range, and root mean square) and the remaining 2 are correlation values (autocorrelation of the acceleration norm, spearman correlation between x and y dimensions).

In short, this method has a total of 7 parameters: a threshold for each of the mentioned features. These values were estimated in a different set from those used for analysis and in a brute force way: go through a set of values and check which ones presented the best division between activities. The division precision is associated with the parameter weight for the final decision, which can be interpreted as a confidence value in the parameter in question. With the first parameter, the signal processing method first identifies "fast start" events (feeding and escaping). To do this, it is used a sliding window covering each 1 second period of the acceleration norm, and it is calculated the respective standard deviation. When the condition is verified, then the remaining parameters are used to distinguish between feeding and escaping. Finally, in this project, the impact of frequency was also studied. The lower the frequency, the useful characteristics/thresholds vary and worse results are obtained. However, in certain scenarios, it can be advantageous in terms of energy costs. It is a context-dependent tradeoff.

Zhang et al. [15](2019) solves a similar problem but focuses on activities related to sharks: swimming, resting, feeding, and non-deterministic movement. For this, and similarly to the previous method, it uses time series of the value of the overall dynamic body acceleration (ODBA), calculated by adding the absolute value of the acceleration in each of the dimensions. 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:

- Shark VGG1: two convolution layers followed by a max-pooling, repeated 2x and always using twice the filters of the previous convolutions. After the layers related to features extraction, the output is full connected to a layer with 1000 neurons which in turn is full connected to a layer with 100 neurons. Finally, an output layer with a softmax activation function which the number of neurons is relative to the number of activities (4);
- SharkVGG2: the previous model uses 1x5 kernels. In this version, the same layers are used for feature extraction with different kernels (1x3, 1x5, 1x7). The output, using the 3 kernels, is concatenated and passed as input to the first hidden layer. In other words, the layers related to feature extraction are applied with different kernels "in parallel". The network architecture remains;
- SharkInception: uses the concept of inception. In this case, the feature extraction layers are not replicated "in parallel" but in each convolution block, before being passed to the max-pooling layer. The network architecture remains the same.

The idea behind the application of several kernels is to allow processing at different levels of detail in parallel. The results shown in the article demonstrate that all networks went into overfitting since the accuracy obtained in the training was much higher than that obtained in the test set. It should also be noted that approaches based on this type of sensors are considered invasive by certain specialists and are easily lost from the animal in question, causing additional economic costs.

Zhou et al. developed several projects ([19], [17] and [18]) within the scope of the feeding activity. Initially, in [19], the objective was to detect the feeding activity through the analysis of the level of aggregation of the school, since normally it is more aggregated during this period. This measure is described using a value called the Flocking Index of Fish Feeding Behavior (FIFFB). For its calculation, fish are first detected in a given image, and their center of mass is calculated. Given the set of centroids, 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 points not yet covered;
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 adding 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.

$$FIFFB = \frac{\sum_{i}^{n}(L1_i + L2_i + L3_i)}{n} \tag{1}$$

The idea of this approach possibly works best in scenarios where the species in question are not usually aggregated, so the difference is greater during the feeding period. In [17], one more index is used: Snatch Intensity of Fish Feeding Behavior (SIFFB). In this method, it is argued that fish normally eat close to the surface, and during the feeding period, the surface texture changes substantially due to the intensive movements of the fish. The texture is represented by the gray level co-occurrence matrix (GLCM). This matrix is a $p \times p$ matrix where p is the number of possible intensity values. For each position (i, j), the corresponding value defines how many times these values occur together given an offset in terms of x and y. The SIFFB index value is defined as the contrast value which is higher during the feeding period. It is given by: $\sum_{i}^{p}\sum_{j}^{p}(i-j)^2 P(i,j)$ where i and j are intensity values and P the GLCM. The work proposed in this article also proposes a methodology, based on a model that combines a fuzzy inference system and a neuronal network, to identify when the feeding should be stopped. The input of this model is the two described indexes and the output is a binary value on to stop or not.

Finally, in article [18], an innovative idea is described to identify the appetite of the fish present in a given image. For this, a convolutional neural network (CNN) is trained. Several images at different levels of appetite are necessary for the model to achieve good performance. To have more training examples, the dataset was increased using transformations of rotation, scale, and translation. The idea in this approach is that the model itself learns image patterns associated with each level of hunger. The network architecture consists of two series of convolution layers (5x5 kernel) and max-pooling (2x2 kernel), followed by two hidden layers (with 120 and 84 neurons respectively) and an output layer (with 4 neurons representative of the various hunger intensities). This approach may have worse performance when several species are detected by the camera, like in environments considered more complex. Additionally, the measurement of appetite itself can be considered subjective, from specialist to specialist.

2.4 Summary and Conclusions

Detection Background subtraction methodologies can be simpler to implement. However, they are very sensitive to environmental changes. Gaussian mixture models can deal with small background movements once it can model different pixel intensities. Finally, Convolutional Neural Networks are the most robust to this problem but it needs a huge amount of data to achieve good performance.

Tracking The base approach for tracking is position-based. This can be very sensitive to occlusions derived from fish crossing movements. More features can be used to make this module more robust namely: color, shape, and texture features. Motion prediction can also be useful to recover from missing detection or occlusions during a few frames. In situations where the actual prediction and full track are not important, optical flow can be a good choice to have an idea of the general movement (for example using a particle grid).

Classification To classify the fish frame, there are two options: feature extraction plus a model/set of rules or an approach based on neural networks. Convolution Neural Networks usually achieve better performance but need lots of data. The traditional feature extraction approach can also achieve good results, resorting to color, shape, and texture features.

Behavior Detection Table 1 defines a summary for the behavior related projects. From that the following conclusions can be extracted:

- the rule-based approach ([3]) gave a substantial amount of false abnormal behaviors due to the rules coverage and the different species taken into account;
- [16] method has the disadvantage of focusing on shoal behavior instead of fishes individually;
- clustering approaches ([13], [4]) are applied considering all data points of the trajectory, so the interval related to the abnormal behavior can not be gathered. Additionally, the normality of the most part of the trajectory can cancel some fast-start abnormal event;
- projects that used sensors ([5], [15]) are considered invasive and these are easily lost causing more costs.

Approach	**Behaviors**	**Limitations**
Motion rule-based([3])	Normal/Abnormal	Lots of false abnormal behaviors
Kinetic energy (velocity + turning angle) ([16])	Normal/Abnormal	Shoal based
Clustering and outlier detection ([13],[4])	Normal/Abnormal	Performed on all trajectory
Reference Zones ([9])	Inspecting/Biting	Image plane (2D)
Accelerometer data ([5], [15])	Swimming/Resting/ Feeding/Abnormal	Sensor costs, invasive, easily lost
Flocking and Snatching Index ([19], [17])	Feeding	Species must have a flocking behavior during the feeding period

Table 1. Behavior Detection Summary

3 Proposed Solution

The proposed solution proposes a system capable of detecting certain activities of sharks and mantas as well as abnormal behaviors, considered of interest to biologists to analyze. The system includes a web application where the results will be illustrated and additionally allows the end-user to provide feedback regarding the alerts given. The main module is the behavior detection module. A first approach is only based on a set of rules, acquired from the analysis of certain features during the various behaviors. In a second phase, it is intended to improve performance including machine learning models.

3.1 Behavior Detection Module

The behavior detection module, as the name implies, aims to detect a set of activities considered of interest, called episodes. This module receives a set of trajectories as input and returns a set of episodes (fig.1). A trajectory can be seen as a list of positions on which the fish was detected given by: $p = [(t_1, x_1, y_1), ..., (t_n, x_n, y_n)]$, where n is the total number of frames in which the fish was detected and t is the number of the frame itself. Each episode is characterized by the ID of the fish, its species, the description of the episode, and the initial and final timestamp. The description can be one of the following: “swallowing air”, “feeding”, “lack of interest” and “abnormal”.

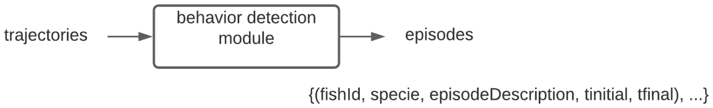

trajectories
behavior detection module
episodes
{(fishId, specie, episodeDescription, tinitial, tfinal), ...}

Fig. 1. Behavior Detection Module

The produced trajectories depend on the performance of the underlying modules: detection and tracking. For this reason, the trajectories received as input may have missing points due to failure in detection or poor association by the tracking module. To try to get around this problem, polynomial interpolation is applied since a polynomial allows to represent most functions satisfactorily. The method expected to be used is Newton’s method. The coefficients of the polynomial are calculated using the divided differences. The interpolating function is given by: $p(x) = y_0 + \sum_i^n (\Delta^i y_0 \prod_j^i (x - x_j))$ where $\Delta^i y_0$ is the split difference of i-th order. The interpolation is done concerning x and y independently. The points used to interpolate are the edges of the missing interval plus $\frac{n-1}{2}$ previous/subsequent points, being n the polynomial degree. Notice that the interpolation is just an approximation, if the tracking algorithm gives too large gaps something has to be improved on this module.

Baseline Approach The baseline approach has the goal of giving an initial idea of the module performance. This method does not use any machine learning models and it is based only on image processing algorithms. In general, the baseline approach defines a set of rules to detect the desired behaviors.

The swallowing air activity is detected based on the distance of the shark center of mass from a reference point (as in [9]), in this case from the surface. If this distance is smaller than a given parameter during n consecutive frames, then this activity is inserted in the result episode set. Otherwise, it is only considered to be noise. The image domain can be seen as a 2D plan. Therefore, the threshold parameter should be dynamic in a way that it should have a higher value when the shark is further away from the camera. An idea of the depth can be obtained based on the bounding box area. This will be larger when closer to the sensor.

Relatively to the feeding behavior, it is recognized based on the aggregation level. Sharks usually meet at the surface when they feel it is feeding time and manta rays follow the same logic but not at the surface. Additionally, these species normally do not swim in shoal so the aggregation level can be a good indicative factor of the feeding period. The Delaunay triangulation algorithm (as used in [19] and described in 2.3) is applied to the set of sharks and manta rays centroids and the flocking index is calculated based on the perimeters of the triangles (equation 1). When this value is smaller than a given parameter, it means that the set of sharks, or manta rays, detected on the given frame are very close to each other. Similar to the previous activity, when this is noticed for n consecutive frames this activity is saved into the results set, for every fish in the aggregation region.

When the feeding behavior is detected, the aggregation center of mass is calculated and the distances from each centroid to this point. If there is any point further away from the aggregation centroid, it is considered that the fish relative to that point is not interested during the feeding process. This is determined by finding out if any of the distances 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.

Sharks and manta rays usually have constant direction and velocity. Therefore, to detect abnormal behaviors these features are used. Velocity is calculated based on the norm of the difference among consecutive positions and the turning angle is the angle between its line segment and the x-axis. This is calculated for several example images and many fishes. A normality histogram is defined for the sharks and another for the manta rays. This histogram is a 2D normalized histogram that have into account the velocity with the turning angle. Given a new detection, the velocity and angle are calculated, and their probability is retrieved from the histogram. If this probability is smaller than a given parameter, then it could be indicative of something abnormal. This is inserted as a new episode if it is verified for several n consecutive frames.

Fig.2 illustrates the underlying flow. It starts by trying to infer if the given image is related to a feeding period. If it is, then the module identifies fishes with a possible lack of interest. Otherwise, it tries to identify if any shark is near the surface indicating the swallowing air behavior. In none of the activities was detected it finally checks if there is anything abnormal according to velocity and turning angle.

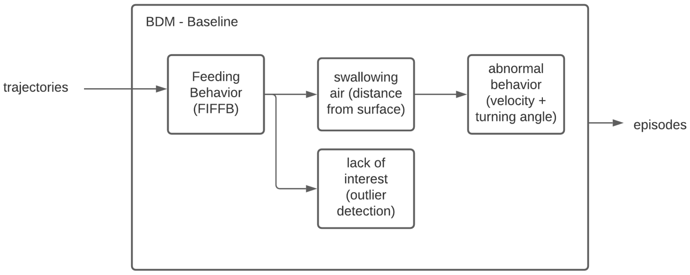

BDM - Baseline
trajectories
Feeding Behavior (FIFFB)
swallowing air (distance from surface)
abnormal behavior (velocity + turning angle)
lack of interest (outlier detection)
episodes

Fig. 2. Baseline Approach

Improvement using Machine Learning The evolution of hardware capabilities and the huge amount of data that is generated daily encourages the usage of machine learning models. In the context of this project, it is pretended to improve the baseline performance training two different models:

- a density-based clustering model to detect feeding behavior
- classification model to predict between feeding, swimming, and abnormal movement activities

DBScan is one of the options that can be used as a density-based clustering algorithm. It is essentially composed of two parameters ϵ and $min\_samples$, and it is described in the following steps:

1. choose a random unvisited point and create a new cluster
2. verify if there are more than $min\_samples$ with smaller distance than ϵ
3. if so, the same cluster is given to those samples and step 2 is repeated for those as well
4. if not it returns to step 1

As the logic described in the previous section, the clustering process is performed on the set of detected centroids relative to a given frame (and separately for sharks and manta rays). If the algorithm finds a cluster with enough cohesion, then it is assumed that it is related to feeding behavior. The detected outliers can be seen as the fishes with a lack of interest.

The pretended classifier should be able to distinguish between swimming, feeding, and abnormal movement behaviors. The training phase (fig.3) is characterized by the following steps:

1. each trajectory is segmented using a t seconds window
2. each segment is converted in two time series: one related to velocity and another related to the turning angle
3. feature extraction: several features are extracted from the time series values (mean, median, standard deviation, minimum, maximum, first quartile, third quartile, and autocorrelation for each time series individually plus correlation between time series)
4. finally the set of feature vectors are fit into the model

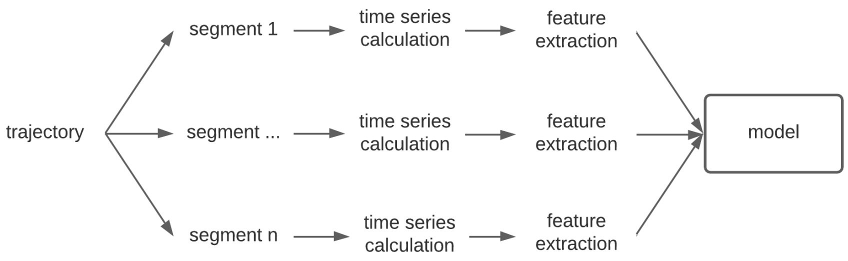


Fig. 3. Classifier Training

Naive Bayes, Support Vector Machine (SVM), and Random Forest are the models intended to be evaluated initially. Each of them represents a different classifier family: probabilistic, similarity-based, and logic respectively. The usage of deep learning will depend on how much data will be possible to gather once this type of model needs a huge amount of examples due to its number of parameters.

3.2 System Architecture

The architecture that supports the project goals and that incorporates the module explained in the previous section is illustrated in fig.4. In general, it can be described by the following flow:

1. the data gathering process reads camera frames and from time to time notifies the server that there is a new video to be processed;
2. the server, every time a new video is available and it was not already processed, sends a request to a worker which will process using the module explained previously;
3. after receiving the results, these are stored in a database which will be consulted by a web service when the web application needs to present the results to the user.

The relationship between the server and the worker nodes can be seen as a publisher-subscriber model. When a worker node is launched, it subscribes to the server to be able to receive new videos to process. Regarding the web application, the main goal is to present the alerts to the users (biologists). They can also give feedback related to each alarm and with this feature, the system can re-train the models when there are many incorrect alarms.

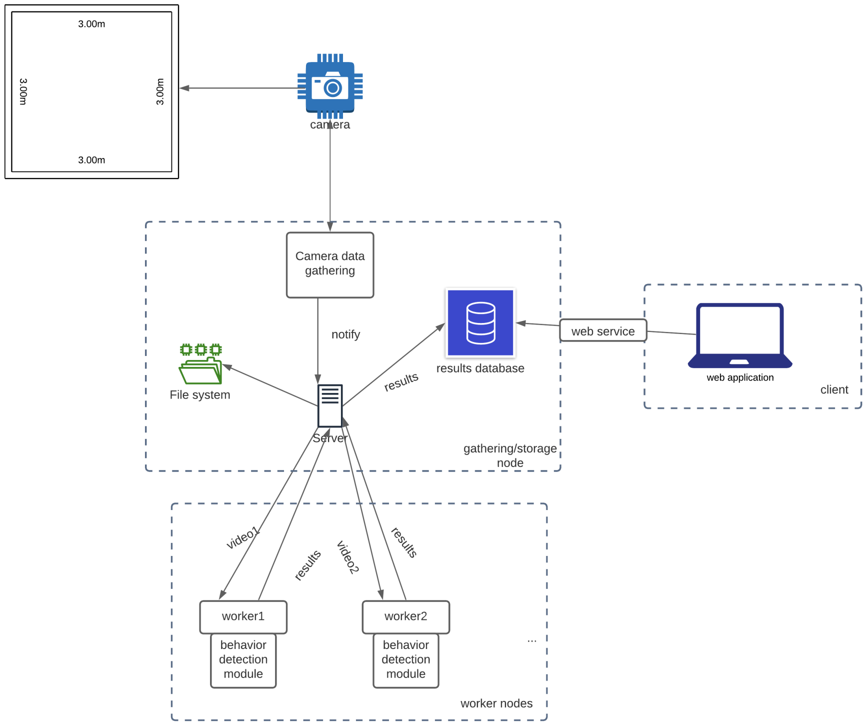


Fig. 4. System architecture

Results Storage Memory nowadays is considered cheap. However, considering the scenario where the system has a camera recording 24h daily, store every single frame can be heavy, even when using compressing algorithms. To overcome this problem, it is avoided the storage of the whole video. Instead, only the frames related to the detected episodes are saved into persistent memory. The metadata is stored in a non-relational database given the simplicity of the data to be stored and the performance guarantees that this type of database can give. The metadata that needs to be stored is represented in fig.1. Beyond the fields returned by the behavior detection module, it is also necessary to have information about the file path to the video segment related to that episode. Therefore, each episode document will have the following fields:

1. fish ID
2. fish specie
3. episode description
4. initial timestamp
5. final timestamp
6. video path

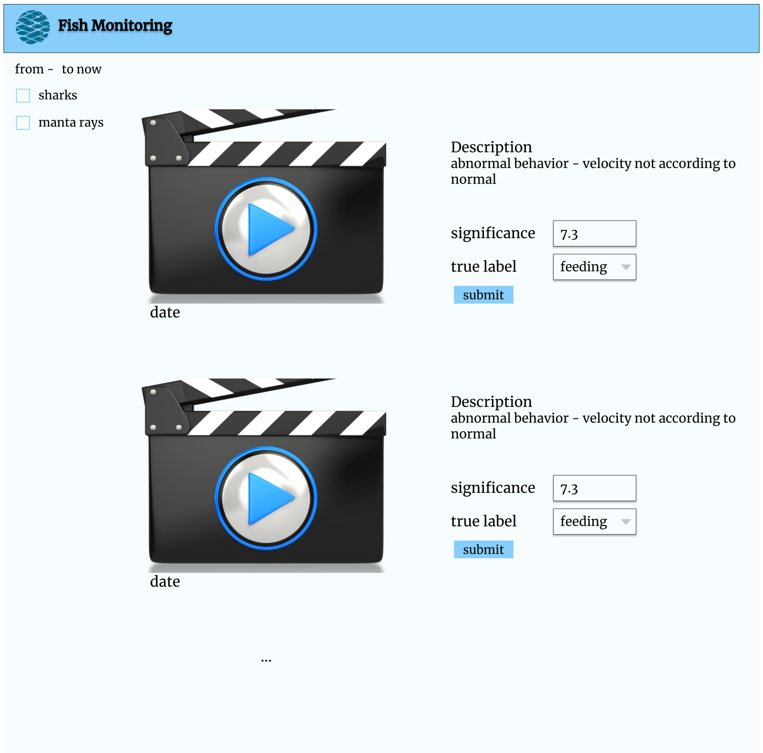
The system will have the capability of re-training the deployed classification model, according to user feedback. To do so, it is necessary to also store all the processed trajectories, at least until the user satisfies all the detected episodes related to a given trajectory. They are stored having a *fish ID* field, as the episode, and the *positions* field which will be an array of position tuples. Additionally, each episode will have a field named "real label" that will be the label assigned by the user. He can also give feedback regarding the episode significance so another field *significance* will be inserted which will be a 0-10 number, being 10 high significance.

Technologies All the server-side software will be developed with Python language due to its growth in popularity and especially because of the wide number of libraries in the data processing, image processing, and artificial intelligence world, namely: *numpy*, *matplotlib*, *pandas*, *openCV*, *scikit-learn*, *keras*, *tensorflow*, among others. The server will be implemented using *Flask* framework, which allows the development of web API's and applications. This server will be responsible for producing the web pages for the web application and for sending videos to the worker nodes. Additionally, the *flask* HTML templates (*jinja* templates) will be using the *bootstrap* library. Finally, *MongoDB* was the chosen database engine for this project, which is a document database recognized for its scalability and flexibility.

3.3 User Interface

The web application will be a single-page app. The main goal is to present the detected episodes to the biologist and also to allow receiving feedback. The fig.5 illustrates a first version of its design. In summary, it will implement the following features:

- demonstrate episode video segment: it should highlight the focus fish and draw its trajectory;
- associate a description to the event according to the described episode labels;
- receive feedback: significance level (used to evaluate the system) and the true label (used to re-train);
- filter by date and species.



Description
abnormal behavior - velocity not according to normal

significance 7.3

true label feeding

Description
abnormal behavior - velocity not according to normal

significance 7.3

true label feeding

Fig. 5. User interface

Model Retraining After deploying a model, its performance can degrade over time. It is mainly related to changes in the environment where the model is operating on. This concept is called Model Drift. In the context of the project, new sharks or manta rays can be inserted into the tank over time, so the initial dataset used to train can contain outdated samples, according to the fishes that are currently on the tank. The dataset will be label by a non-biologist so it can also have misleading classes. The web application makes it possible to take into account the biologist feedback, and the set of wrong label episodes can be used to re-adjust the deployed model.

One of the issues of re-training is the capability of detecting model drift and decide how often should the model be re-trained. In this project, the model should be re-adjusted every time is detected n incorrect classifications. When this is triggered, in the first version, the model starts again the training process using the previous dataset together with the new data. In some contexts, re-train from scratch can be expensive. To overcome that, online learning techniques should be adapted such as:

- SVM case: adjust the current weights only based on the new data

- Random Forest case: adding more decision trees to the ensemble according to only to the new data
- among others, depending on the selected model

4 System Evaluation

The proposed system will be evaluated in several ways:

1. classifiers performance
2. episodes detection performance
3. alarms significance
4. system processing delay

Regarding the performance of the supervised models, metrics such as accuracy(equation 2), precision (equation 3) and recall (equation 4) will be extracted, where tp, tn, fp and fn are the true positives, true negatives, false positives and false negatives respectively. Note that probably the dataset will be unbalanced, therefore only the accuracy will not describe its performance. The precision specifies the true positive rate while recall specifies the fraction of the total amount of positive instances that were well classified. The confusion matrix will also be observed to conclude between which behaviors it has the most errors.

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2}$$

$$precision = \frac{tp}{tp + fp} \tag{3}$$

$$recall = \frac{tp}{tp + fn} \tag{4}$$

The behavior detection module will be evaluated similarly and using the same metrics. If an episode returned as output is within $x\%$ of the time of the real episode, then it is considered a correct episode.

The metrics related to the significance of the alarms allow us to conclude whether the alarms are being useful for the end-user (biologists). The web application interface will concede the ability to give feedback regarding a certain alarm. This feedback will be a value within the 0-10 scale. From the set of collected values, it is possible to extract statistics such as standard deviation and mean: $\frac{1}{n}\sum_{i=1}^{F} f_1 + f_2 + \cdots + f_n$ where F is the total number of received feedback.

Finally, the system will also be evaluated according to the processing delay it will have. Although it is not a real-time system, it is not beneficial to delay the processing of videos and consequently their results to the user. That said, the delay will be described as the maximum number of videos that have been queued for a certain period (e.g.: daily).

5 Planning

Fig.6 illustrates the schedule for the project development (the black lozenges represent milestones). Initially, the goal is to develop the behavior detection module: first the baseline version and after with machine learning-based improvements. By the middle of the project, the system architecture can start being taken into account by developing the user interface, the centralized server, and the database. Finally, a benchmarking process is performed for methods' comparison while the model retraining feature is also started. The last two months of the project are restricted to dissertation writing.

6 Conclusions

This project tries to solve the monitoring problem. Biologists do not need to spend too much time controlling fishes in a given tank. The related literature is vast in terms of topics: detection, tracking, classification, abnormal behaviors, feeding behaviors, among others. Relatively to behaviors, there is a special interest in abnormal behaviors and also on feeding behaviors due to their impact and support to biologists. The proposed work aims the development of a system capable of detecting some of the considered behaviors of interest, to filter, to the end-user (biologists), the detected episodes. The system is composed of a behavior detection module which initially will only be based on image processing and several rules. Posteriorly, it will be improved using machine learning approaches. The system will also have a web application where the detected episodes will be shown.

References

1. Omar Anas, Youssef Wageeh, Hussam El-Din Mohamed, Ali Fadl, Noha ElMasry, Ayman Nabil, and Ayman Atia. Detecting abnormal fish behavior using motion trajectories in ubiquitous environments. *Procedia Computer Science*, 175:141–148, 2020.
2. Catarina Barata, Jacinto C Nascimento, and Jorge S Marques. Improving a switched vector field model for pedestrian motion analysis. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 3–13. Springer, 2018.
3. Cigdem Beyan and Robert B Fisher. A filtering mechanism for normal fish trajectories. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 2286–2289. IEEE, 2012.
4. Cigdem Beyan and Robert B Fisher. Detection of abnormal fish trajectories using a clustering based hierarchical classifier. In *BMVC*, 2013.
5. 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.

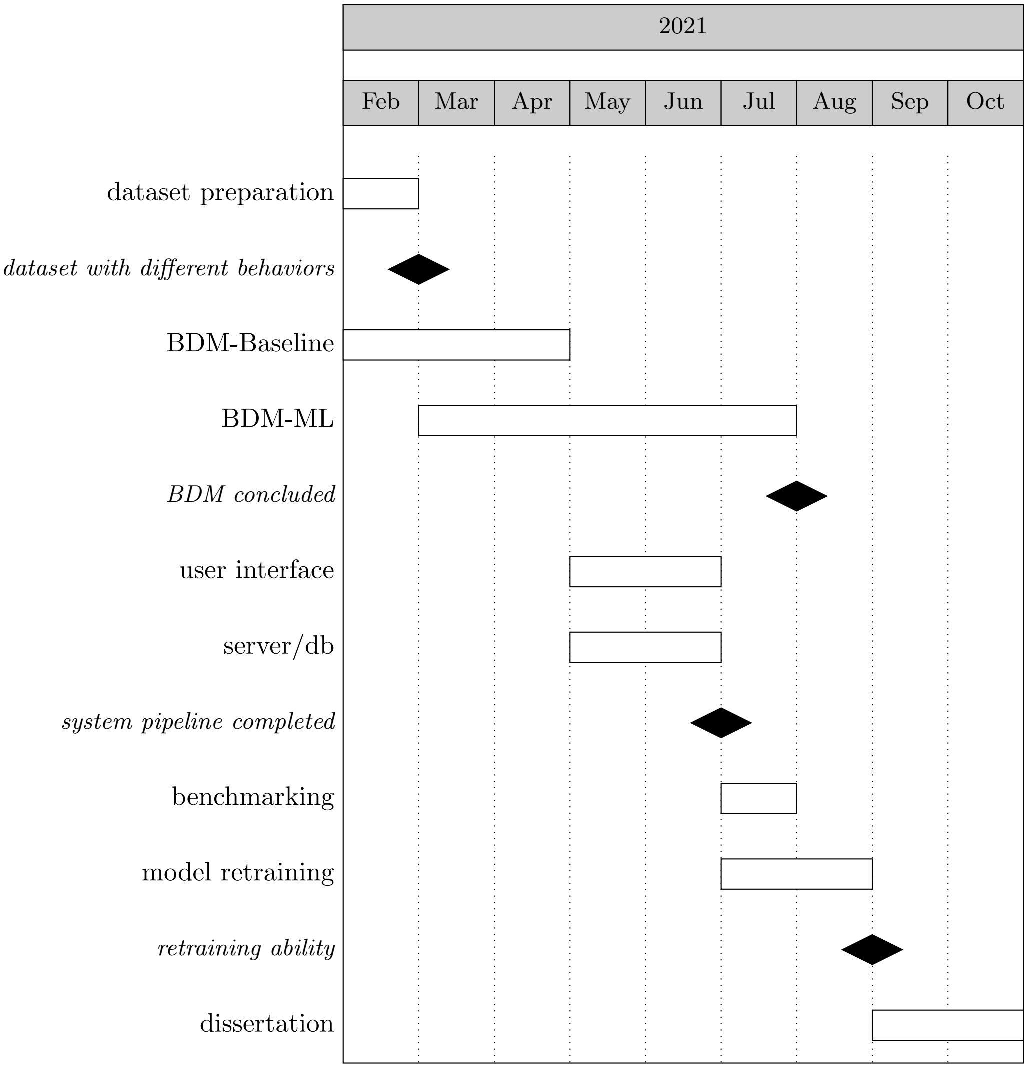


Fig. 6. Project gantt chart

6. 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.
7. Taina Gariglio Dias Coleman. *SharkID: A Framework for Automated Individual Shark Identification.* PhD thesis, California State University, Long Beach, 2020.
8. SO Ogunlana, O Olabode, SAA Oluwadare, and GB Iwasokun. Fish classification using support vector machine. *African Journal of Computing & ICT*, 8(2):75–82, 2015.
9. 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.
10. Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, et al. Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)*, 45(4):1–32, 2013.
11. Telmo JP Pires and Mário AT Figueiredo. Shape-based trajectory clustering. In *ICPRAM*, pages 71–81, 2017.
12. Dhruv Rathi, Sushant 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.
13. Concetto Spampinato, Daniela Giordano, Roberto Di Salvo, Yun-Heh Jessica Chen-Burger, Robert Bob Fisher, and Gayathri 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.
14. Chuang Yu, Xiang Fan, Zhuhua Hu, Xin Xia, Yaochi Zhao, Ruoqing Li, and Yong Bai. Segmentation and measurement scheme for fish morphological features based on mask r-cnn. *Information Processing in Agriculture*, 2020.
15. 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 *2019 IEEE Green Energy and Smart Systems Conference (IGESSC)*, pages 1–6. IEEE, 2019.
16. 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*, 127:271–280, 2016.
17. 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.
18. 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.
19. 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*, 135:233–241, 2017.