water

MDPI

Article

An Application of Fish Detection Based on Eye Search with Artificial Vision and Artificial Neural Networks

Ángel J. Rico-Díaz [1,2,*], Juan R. Rabuñal [1,2], Marcos Gestal [1,4], Omar A. Mures [1] and Jerónimo Puertas [3]

[1] Department of Computer Science and Information Technologies, Campus de Elviña, University of A Coruña, 15071 A Coruña, Spain; juan.rabunal@udc.es (J.R.R.); mgestal@udc.es (M.G.); omar.alvarez@udc.es (O.A.M.)
[2] Centre for Technological Innovation in Building and Civil Engineering (CITEEC), University of A Coruña. Campus de Elviña, 15071 A Coruña, Spain
[3] Department of Hydraulic Engineering, ETSECCP, Campus de Elviña, University of A Coruña, 15071 A Coruña, Spain; jpuertas@udc.es
[4] Research Centre in Information and Communications (CITIC), Campus de Elviña s/n, 15071 A Coruña, Spain
* Correspondence: angel.rico@udc.es

Received: 15 September 2020; Accepted: 23 October 2020; Published: 27 October 2020

Abstract: A fish can be detected by means of artificial vision techniques, without human intervention or handling the fish. This work presents an application for detecting moving fish in water by artificial vision based on the detection of a fish's eye in the image, using the Hough algorithm and a Feed-Forward network. In addition, this method of detection is combined with stereo image recording, creating a disparity map to estimate the size of the detected fish. The accuracy and precision of this approach has been tested in several assays with living fish. This technique is a non-invasive method working in real-time and it can be carried out with low cost. Furthermore, it could find application in aquariums, fish farm management and to count the number of fish which swim through a fishway. In a fish farm it is important to know how the size of the fish evolves in order to plan the feeding and when to be able to catch fish. Our methodology allows fish to be detected and their size and weight estimated as they move underwater, engaging in natural behavior.

Keywords: computer-vision; Hough transformation; artificial neural networks; fish-size; stereovision; eye-detection

1. Introduction

Object detection has been a much-discussed topic in artificial vision studies, since a good identification of the required object provides the basis for collecting information via image processing in a system of this type. In the proposed work, the object to be located is a live fish that is submerged in water and where the conditions of both light and visibility may be variable. The detection of fish in their natural environment using artificial vision allows the setting aside of techniques which are potentially invasive for the fish, such as the use of sensors [1], subjecting the specimens to stress, or other traditional techniques such as direct vision that involves continuous supervision by an operator. Our technique can be used in several applications, ranging from monitoring the growth of fish in aquaculture in order to adapt their feeding, to estimating differences in body size and condition of migratory fish moving through fish passageways.

The use of video cameras is one of the most powerful methods to detect objects as it provides a great deal of information and is also one of the cheapest methods. To detect objects under water there are several techniques, one widely used of which is acoustic technology [2,3]. One of the great advantages of acoustic technology is the possibility of detecting objects at many distance ranges, but the biggest limitation is obtaining or detecting the real size of the object. Another limitation is the speed of acquisition, since the devices require a transmission time and signal processing that limits their application to objects that do not move quickly. Acoustic technology is used to detect schools of fish, large fish or slow-moving fish [4]. However, if the objective is to detect an individual fish, of small or medium size, and it is also necessary to know its size, acoustic technology is not the most appropriate. Using video cameras, it is possible to obtain much information and at high speed. The main problem is that underwater the image deteriorates greatly, due to numerous factors, such as turbidity, floating particles, bubbles, and the attenuation of light at different depths. There has been research that tries to restore or improve the quality of images obtained underwater using different techniques. [5–7].

Due to the poor visibility and undesired effects such as aquatic snow, object detection using cameras in underwater environments is considered a challenging task. Artificial Vision techniques for underwater image processing are current, mainly in robotics, for the development of autonomous navigation of underwater robots [8]. The objective is to detect fish that are usually in motion, even at great speed, and once detected to estimate their size. For this, it is proposed to use a stereoscopic camera under water, from which two videos are obtained at speeds of 25–30 frames per second, which allows the fish to be captured even in motion. One of the images is used to apply artificial vision techniques and to be able to detect if there is a fish (or several) and, once detected, the other image (stereo) is able to calculate the distance at which the object is located and estimate its size. This will mean an advance in applications for fish farms, where they can automatically calculate the growth in size of their school of fish. A fish farm is usually in the sea, at a certain distance from the coast, and occupies a space closed by a net, so the fish inside cannot escape. It is important to know how the size of the individual fish is progressing to know when to proceed with their extraction. With the application proposed in this paper, through one or several submerged stereo cameras running continuously, detection of the fish that pass in front of the camera and estimation of their size is permitted. Over time, a computer program can automatically obtain (recording in a database) the means of the fish sizes and automatically visualize the evolution of the average size of the population. To date, there is no system that allows this to be done automatically, and this paper shows how to apply and combine artificial vision techniques that allow their use in these applications with a certain degree of efficiency.

By improving the fish detection process, the accuracy of applications can be increased in order to estimate the number of fish passing through a given point and their sizes in quarantine tanks, fish farms, aquariums, etc., to calculate trajectories for the optimization of building vertical slot fishways [9], and to analyze invasive species [10].

Most of the techniques used in artificial vision need prior knowledge of the fish species to be detected [11,12] and of the background where the image will be captured, which restricts the possibilities of detection in a more open and more real scenario. Thus, detection can be performed using neural networks [13], snakes [14], level sets [11] and patterns [15], etc. In the following paper, the authors use deep learning with a convolutional Artificial Neural Network (ANN) (CNN) to detect fishes, but in clean water and with a high luminosity [16]. In [17] the authors use artificial vision to detect the fish using low visibility underwater videos, using background subtraction methods, filtering the images and a postprocessing technique, obtaining an accuracy of 60%. The main idea of our paper is to use ANN combined with artificial vision techniques to detect the fish and increase the accuracy of the detection process.

This study presents a new application for fish detection by using underwater cameras requiring less knowledge in the first stage about the species to be detected. The basis of this technique is the search for the fisheye in the image, and, from this detection, other techniques may subsequently be applied, aimed at finding out the shape of the fish and performing other analyses.

The proposed technique was first evaluated in a controlled environment, and later experiments were conducted in a real scenario with fish of different size and under changing conditions of light and turbidity of water. For the video acquisition, two submersible synchronized cameras were used to allow stereoscopic capture and thus be able to perform a measurement of the fish, once detected.

Section 2 of this paper describes two techniques for fish detection: first, the technique of eye detection using the Hough algorithm, and second, eye detection using an Artificial Neural Network. We combine both techniques to obtain a better detection capacity against the different conditions that can occur: turbidity, particles in suspension in the water, bubbles, and different positions and movements of the fish. The performed detection is applied in Section 3, to estimate the size of the fish. Section 4 presents the analyses and results of the technique. Finally, Section 5 includes our conclusions, and possible future improvements are discussed.

2. Materials and Methods

In order to be able to measure fish length or weight, the first step is their detection via the input images. To achieve this objective, several image filtering techniques are applied, as well as background subtraction models, contour detection and cascade classifiers, so method combinations can be additive and complementary. First, the background subtraction model is detailed. After obtaining a rough idea of where the fish is positioned in the images as a result of background subtraction, a validation technique is applied, ensuring that the discovered objects are indeed fish.

The first fish detection technique that will be analyzed is based on fisheye detection using the Hough Transform. Since this approach is highly sensitive to noise in the input images, a second fish detection approach using cascade classifiers is proposed. This approach does not perform fisheye detection, but instead attempts to detect the fish shape and texture. This makes the proposed system more robust against noise and possible false positives. In addition, this technique also poses the opportunity for fish species recognition. In parallel to this technique, a trained ANN is obtained to detect whether there is a fisheye in an image and also a fish-like shape. In this way, combining both techniques, detection capacity is improved.

Before applying these techniques, the image is pre-processed with different filtering techniques (greyscale, gaussian blur, mean shift), in order to improve the characteristics of the image, and in the next step a background subtraction method is employed (as in [17]). In Figure 1 we show the steps of the algorithm.

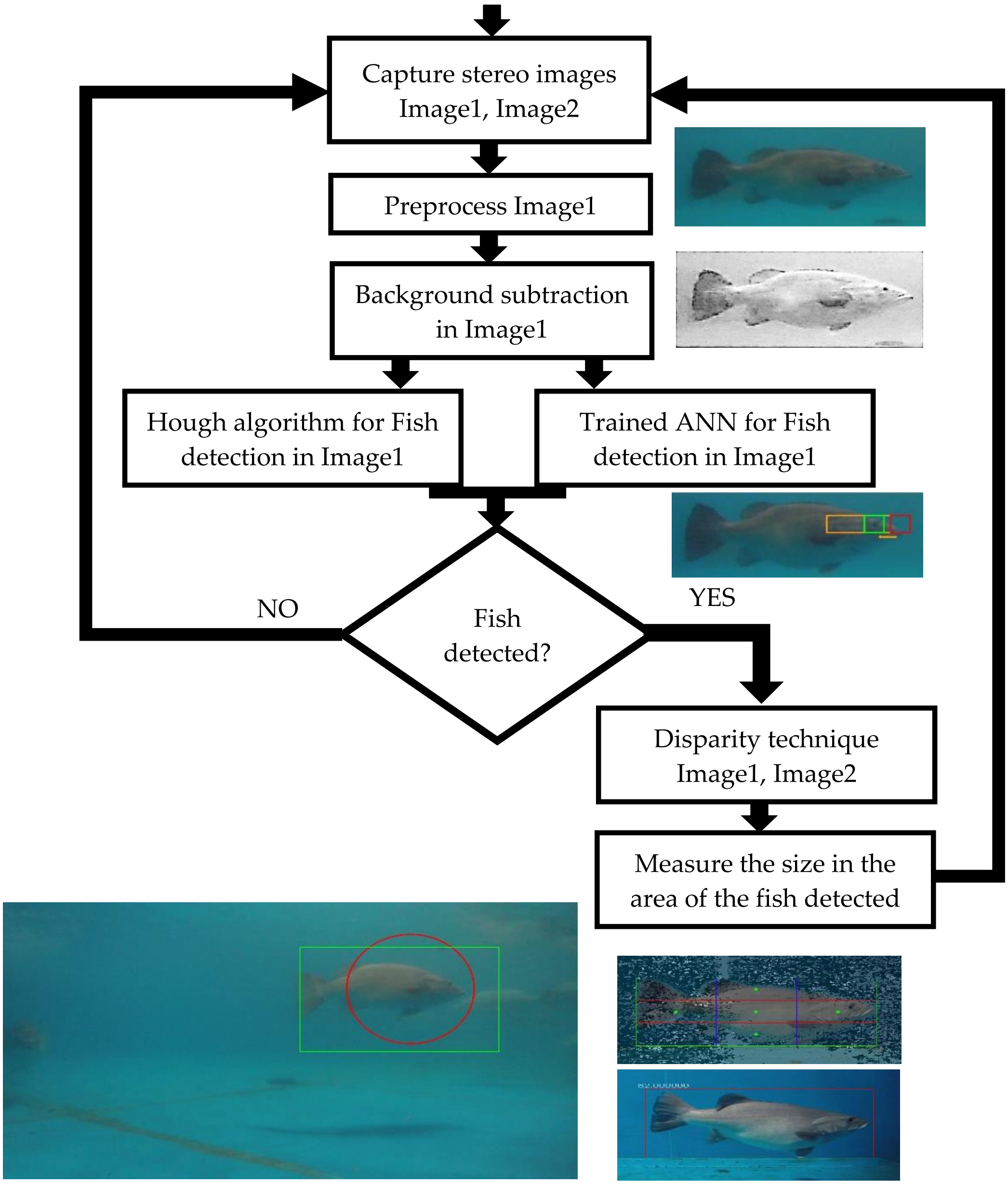

Capture stereo images
Image1, Image2
Preprocess Image1
Background subtraction in Image1
Hough algorithm for Fish detection in Image1
Trained ANN for Fish detection in Image1
Fish detected?
NO
YES
Disparity technique Image1, Image2
Measure the size in the area of the fish detected

Figure 1. Steps of the proposed algorithm and examples of each step.

2.1. *Image Filtering*

Since the images from our test scenarios are substantially noisy and the test environment is also of a difficult nature (bubbles, turbidity, etc.), several image filtering techniques are used to try to enhance the processed images and ease the work of background subtraction and detection techniques. As Appendix A details, the first step is the conversion of the image colors from RGB (Red-Green-Blue) to grayscale (as can be seen in Equation (A1)). Then, in order to reduce noise and improve the robustness of the background subtraction technique, a Gaussian Blur filter is applied to the input images. This filter uses a Gaussian matrix as its underlying kernel, which produces output pixel values only upon the pixel values in its neighborhood, as processed by the convolution kernel (Equation (A2)).

Since the input images are too noisy and complex to obtain good circle matches with the Hough Transform, several steps have to be applied to the input images before fisheyes can be detected. The first step computes a Mean Shift Filter (Equation (A3)). In the second step, the edge contrast of an image is enhanced. In this way the Hough transform is made more robust, An unsharp mask is also applied to the images in order to achieve this objective. The sharpening process utilizes a slightly blurred version of the original image (Gaussian blur), which is subsequently subtracted from the original to detect its edges, effectively creating an unsharp mask. Contrast is then increased, along with these edges (Equation (A4)). Finally, since this type of filter can cause artifacts on edge borders which may lead to double edges, the Laplacian of the original image is also added to the sharpened image (Equations (A5) and (A6))

2.2. Background Subtraction

As performance is key in our application for detecting fish in real time, the first implemented background subtraction method is the simplest and fastest. This approach uses the Gaussian Blur filter in order to reduce noise in the input images.

Next, this blurred image (img_1 in Equation (1)) is compared to the background image (img_2 in Equation (1)) using absolute differences. The following equation is employed for this purpose:

$$AbsDiff(x,y) = \left| img_1(x,y) - img_2(x,y) \right| \tag{1}$$

Once the absolute difference is computed, it can be used to infer which pixels belong to the background and which pixels belong to a moving object. Even when using the previously mentioned filters, noise still presents a problem, in the form of false positives. In order to discard as many false positives as possible, a threshold filter was used:

$$Thresh(x,y) = \begin{cases} maxVal & \text{if } (img(x,y) \ > \ t \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

After disregarding small differences as a result of applying the aforementioned filter, the matches can be expanded to fill holes in the detected objects. This phase involves the dilation of the image with the detected objects, using a 3 × 3 rectangular structuring element as the shape of the pixel neighborhood, over which the maximum will be chosen:

$$Dilate(x,y) \ = \max_{(x',y'):element(x',y')\neq 0} Thresh(x+x', y+y') \tag{3}$$

After the dilation process, the resulting image is run through a contour detection algorithm [18]. The algorithm will detect the object contours by border following, in an image with edge pixels that can be the result of, for example, a Canny edge detector [19], yielding a binary image with the detected borders as a result. This algorithm is able to distinguish between interior boundaries and exterior boundaries of zero regions (holes).

In Figure 2, we can see contours of the two mentioned types; exterior contours are represented by dashed lines, and interior contours by dotted lines:

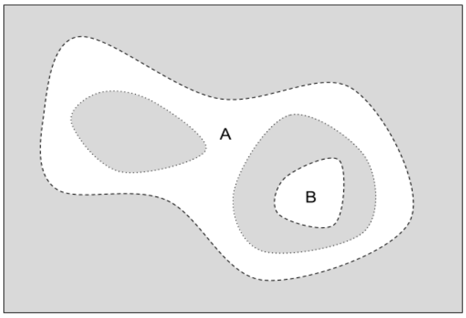


Figure 2. Image with exterior and interior contours.

After the object contours are calculated, the minimal up-right bounding rectangle of the detected object can be easily computed. This yields the potential Region of Interest (ROI) of the object. Since the ROI is not guaranteed to contain a fish, it is first validated as a fish, and only when estimated to enclose a fish is the measurement algorithm applied for estimation.

2.3. Hough Eye Detection

As mentioned in the previous section, once a ROI is obtained with a potential fish candidate, further validation that the detected object is a fish is needed. The first method designed to achieve this objective attempts to identify the eye of the fish in order to distinguish fishes from noise, bubbles, vegetation or other types of marine life. This method is deterministic, meaning that no training is required (in contrast to other methods, such as Artificial Neural Networks), since it mainly relies on the detection of circles using the Hough Transform.

The first steps that are performed in order to try to detect fisheyes deal with the enhancing of the input image in order for the Hough Transform to work better. As mentioned in an earlier section, Mean Shift Filtering is applied to reduce noise and an Unsharp Mask to increase edge contrast.

In addition, before applying the Hough Transform to the image, a Maximally Stable Extremal Regions (MSER) extractor [20] is used to detect binary large objects (BLOBs), a group of connected pixels in an image. This step is necessary due to the issues that the Hough Transform has with noisy and low contrast images. Unlike MSER, it is robust against blur and scale, and useful when it comes to processing images acquired through real-time sources, such as a camera submerged in a fish tank. It reduces the number of false positives of the Hough Transform, which processes the resulting binary image with only the detected white BLOBs against a black background. MSER could also be used to detect the circles pertaining to the fish's eyes, but one would need to measure the parameters of the eye shape in order to detect candidates in advance.

The Circle Hough Transform [21,22] provides a more general method that was specifically adapted to detect circles. This method can be used to determine the parameters of a circle when a number of points inside its perimeter are known. A circle with radius *R* and center (*a*,*b*) can be described by the following parametric equations:

$$\begin{aligned} x &= a + R\cos(\theta) \\ y &= b + R\sin(\theta) \end{aligned} \tag{4}$$

For (0 . . . 360) degrees, the points *x* and *y* trace the perimeter of a circle (Figure 3). This is a 3D parameter space, in which the circle parameters can be identified by the intersection of many conic surfaces defined by points on the 2D circle. An accumulator matrix is used for tracking the different intersection points. The true center point will be common to all parameter circles and can be found with a Hough accumulator array.

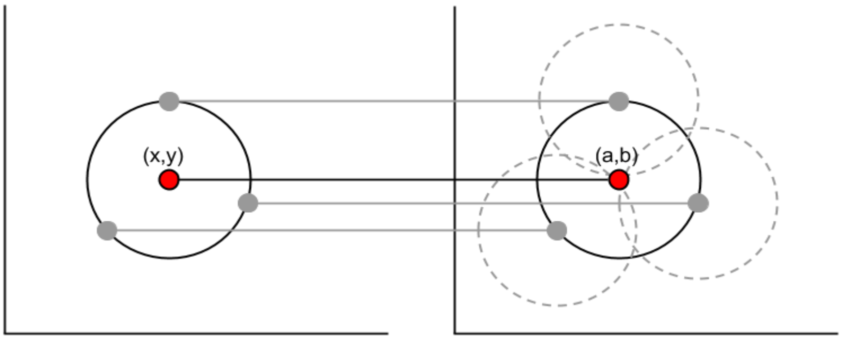


Figure 3. Each point in geometric space (**left**) is the center of a circle in parameter space (**right**). The circles in said space intersect in (a,b) which corresponds to the center (x,y) in geometric space.

Since the radius is not known, the points locus falls on the surface of a cone due to the changing radius *R*. Instead of a circle, each point on the perimeter of the geometric space circle produces a cone surface. Thus, the vector (*a*,*b*,*R*) corresponds to the accumulation cell where most cone surfaces intersect. After the candidate circles are obtained, the researchers still cannot be certain whether they are fisheyes. Further processing is needed to ensure that they are indeed eyes. Besides, if multiple matches are found, only the best should be selected, since our case study needs to measure the fish and the objective is to detect the fish on its side (only one eye is visible from that position). In order to ascertain whether the detected circles are fisheyes or not, the fisheyes in our case study were observed in order to develop a score system in order to classify matched circles. The types of fish that the system was tasked to detect are common fish and looks like the shape in the following pattern (Figure 4). The characteristics of its eyes can be observed.

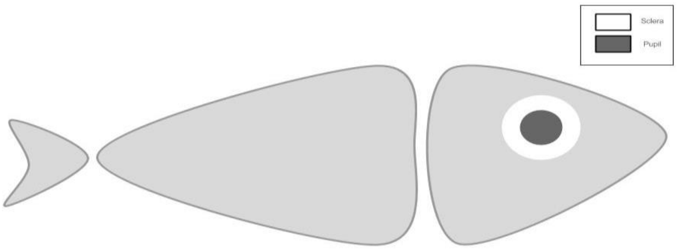


Figure 4. The illustration shows the used features, the sclera and pupil of the fisheye.

Generally, fish eyes have a sclera [23], also known as the white of the eye (Figure 4). This is the opaque, fibrous, protective, outer layer of the eye that contains collagen and elastic fiber. This part has a very characteristic white color that surrounds the pupil of the fisheye. The pupil [24] is a hole located in the center of the eye that allows light to strike the retina. It appears black because of the fact that light rays entering it are either absorbed by the tissues inside the eye directly, or absorbed after diffusing reflections within the eye, which results in them missing the exit of the narrow pupil. The spherical lens of the fisheye protrudes through the pupil opening of the iris. In contrast, humans have a flattened camera-like lens sitting below the iris and pupil. The human iris is therefore adjustable according to light intensity, while the fisheye iris is not.

These characteristics make the fisheye an identifying feature that has the potential to allow us to distinguish fish from other objects. The proposed score system accounts for these two factors, the sclera and pupil of the fisheye. This score system checks the difference between color averages in both regions, since there is usually a high contrast between them as a result of the white and black colors. In order to obtain the scores of the detected candidates and taking into account that different fishes can have different eye features, the user needs to establish several parameters. These parameters include the pupil percentage, sclera percentage, pupil color target and sclera color target. Figure 5 shows what these parameters represent:

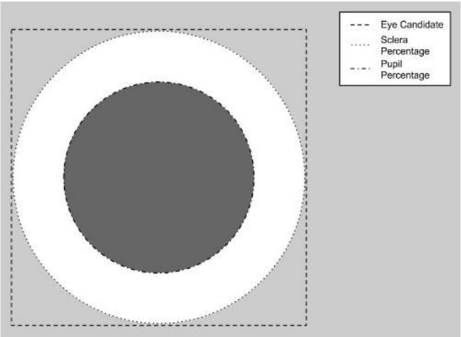


Figure 5. This illustration shows the lines that delimit the sclera and pupil percentages of the candidate match.

The pupil percentage is the percentage of the eye that corresponds to the pupil; the rest of the eye yielding the sclera percentage. The two remaining parameters, the pupil color target, and the sclera color target respectively, account for the color average of the pupil and the sclera of the fish's eye. Once the parameters are chosen depending on the fish type, the next step involves iterating over all the pixels in the candidate match and calculating the color averages of the different fisheye features. Assuming that the pupil and sclera can be modelled with ellipses, this is a matter of checking whether a certain pixel is enclosed in the ellipse defined by the pupil and sclera percentages. This can be ascertained using the following formula:

$$\frac{(x-h)^2}{r_x^2}+\frac{(y-k)^2}{r_y^2}\leq 1 \tag{5}$$

x and y being the pixel position, h and k the position of the ellipse center, r_x the semi-major axis and r_y the semi-minor axis. If the inequality is satisfied, then the pixel will be inside the ellipse. In contrast, if the inequality is not satisfied, then the point will be outside the ellipse. Moreover, knowing if a pixel is inside the sclera or the pupil is simple: if the pixel is inside the two ellipses it will belong to the pupil; in contrast, if the pixel is just inside the outer ellipse, it will belong to the sclera.

As the pixels are classified as part of the sclera or pupil, their color contributes to either the sclera color average or the pupil color average which are calculated using the following Equations:

$$\begin{array}{l}\overline{C}_s=\frac{1}{n_s}\sum\limits_{i=1}^{n_s} img(x_s,y_s)\\ \overline{C}_p=\frac{1}{n_p}\sum\limits_{i=1}^{n_p} img(x_p,y_p)\end{array} \tag{6}$$

n_s and n_p being the number of pixels in the sclera and pupil, img the input image and x_s, y_s, x_p, y_p the positions of the sclera and pupil in the input images. After the average is calculated, it can be compared to the pupil color target or sclera color target and the differences are obtained between the calculated averages and the desired target. If the average error exceeds a certain threshold, the match is discarded; if not, it is temporarily kept. In order to obtain the best match, the candidates are sorted according to their error and the one with the lowest error is chosen as the best match.

Figure 6 shows an example of a correctly matched fisheye in a noisy underwater image. Finally, it can be assumed that the detected object is not a fish or the fish is not in a position suitable for its automatic measurement if an eye match is not present; otherwise it is safe to go ahead and estimate the fish measurements.

The eye detection using the Hough technique has an accuracy rate close to 50% for 9 cm fishes, where the small size of the eye makes it more difficult to locate, whereas for approximately 1 m fishes accuracy rate is over 60%. The processing time is less than one second, which allows in real time processing at least 1 image per s (as can be seen later in the results). However, the 60% adjustment falls short of the end goal. Therefore, it is necessary to implement another technique that allows us to improve this detection capacity; therefore, when this technique is not capable of detecting the fish, Deep Learning is used for its detection, specifically an ANN to classify whether there is a fish present or not. This technique is much more computationally expensive, so it is only applied when the previous algorithm fails.

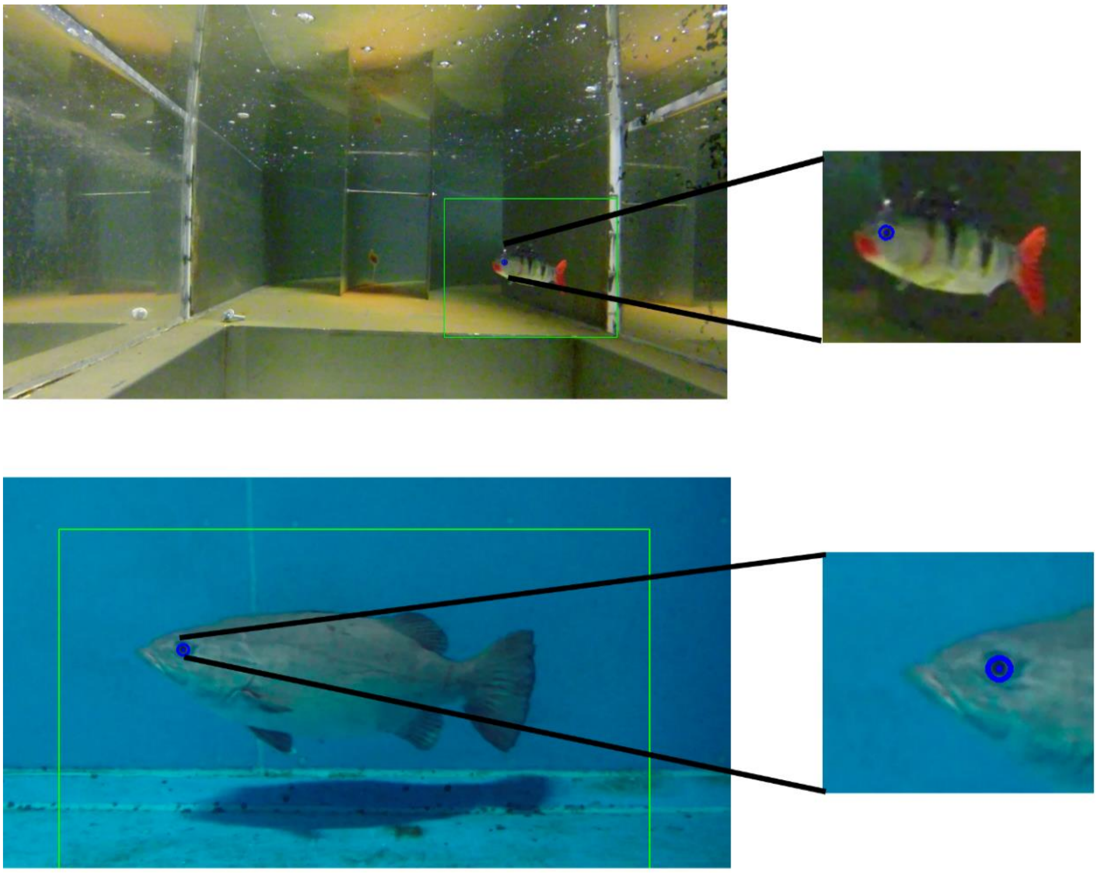


Figure 6. This figure shows a correctly matched fisheye (and zoom) which is highlighted in blue. The ROI in which the fish was located is marked with a green rectangle.

2.4. Fish Detection Using a Feed-Forward Artificial Neural Network

In this part of the work, we use the Deep Learning technique; a Feed-Forward Artificial Neural Network [25] is used for fish eye detection as a classification problem. The Artificial Neural Network (ANN) will indicate whether there is a fish in the image.

The structure of a feed-forward neural network is divided into the input layer, the intermediate layer, and the output layer. The intermediate layer is a multilayer of hidden layers. In this work the intermediate layer will have only one hidden layer in order to reduce the execution time.

Using ANNs requires a training process as first step; in this work the input of the training is a dataset of images classified as “eye” or “no eye” (see images in Figure A3 in Appendix B). All of these samples are derived from full images of fish within the tank and cropped to the desired size. In this work used images width and height are 100 px to avoid too high training times (with this size, training with an ANN requires 3–4 h).

Figure A3 shows a subsample of the images used for the ANN training. The full dataset is composed of 69 eyes images and 103 non-eyes images. These cropped images are used as input samples to the network, so the network will have an input element for each pixel of the image (10,000 PE). A percentage of 85% of these images are used during the training (146 images) and the rest are used in the validation phase (26).

We perform several trials with different architectures and configuration parameters. For example, architectures with 500, 1000, 2000, 5000 and 7500 neurons in the hidden layer were tested. The best results were achieved using 2000 neurons in the hidden layer and showing the samples’ 7500 epochs to the ANN.

With this configuration, ANN was able to predict 21 of 26 of the validation images (80%) and 145 of 146 of the training images (99.31%).

Once the network is trained, it can be applied by means of a sliding window technique to full images used only for the test proposed (they were not used to extract cropped images, neither in test nor validation phases).

Good results are achieved with this technique although too many false positives are detected, so a post-processing phase is required. Furthermore, the main disadvantage of this process is the time required, since it involves the cropping of the original image into segments of 100 × 100 pixels and executing an ANN of considerable dimensions with each, which entails a considerable cost in terms of execution time and computational requirements (more than 10 s). This is the main reason of using only the ANN once the Hough transformation algorithm has not detected the fish.

3. Size Estimation

After performing detection (with Hough algorithm or with Deep Learning technique (ANN)), the silhouette can be located by edge detection techniques. In this case, the information from the resulting image is used after subtracting the background in order to detect the contours appearing in the image and to locate the one containing the detected eye.

With the aim of using this information to estimate the size of the fish, the technique described in [26] was applied. This technique uses two synchronized submerged cameras, creating a stereo vision system that allows the generation of a disparity map of the scenario. The cameras used in this study are two GoPro 3+ Black, with an angle of view of 107 degrees. To submerge them, they are introduced in a watertight housing which maintains the axes of the cameras as parallel and at the same height, and the optical axes are separated by 3.5 cm from each other.

The first step of the technique involves performing a calibration of the cameras, for which a template of coplanar points is used and an OpenCV calibration algorithm is applied based on [27–29]. Through this process, the camera calibration matrices are generated, and the camera pattern is obtained.

The second step consists of calculating the depth. This calculation requires the calibration information from the previous step in order to perform the transformations (remove distortions, displacements, etc.) required in the acquired images, the disparity recorded by the two cameras, so that they can locate the same points in the images of both cameras. This search for matching points between the two images is made using the Block Matching Artificial Vision algorithm [30] and, once found, a disparity map of the image can be generated. The disparity is the difference in the *x*-axis coordinates of each point between the two images and is inversely proportional to the depth. This is a costly process and, with the aim of reducing the processing time, it is applied only to the region where a fisheye was detected. To estimate the size of the fish, the technique used in [31] is applied, which uses the disparity map and the calibration information (Figure 7).

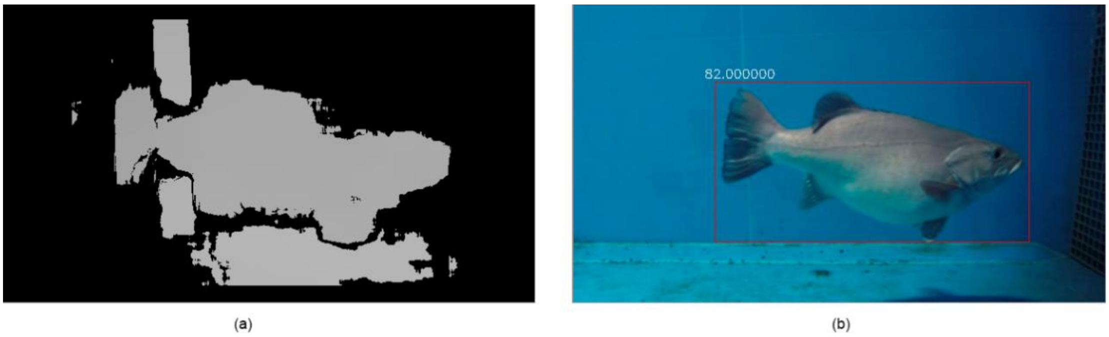


Figure 7. (**a**) Disparity map for detection (**b**) estimation of the fish size (length obtained is 82 cm; real length is 80.9 cm).

The application of the detection technique proposed in this study leads to a decrease of the percentage of false positives, thereby the estimation of the fish size is more accurate, since the size is calculated only at the time when the fish's eye is detected.

4. Results

To check the system, a database of test videos recorded at different locations was used. Videos made where the environment is more controlled are available, as they were recorded in a laboratory of the Centre for Technological Innovation in Building and Civil Engineering (CITEEC), along with other videos recorded in environments more similar to real scenarios, such as the fish tanks from the Finisterrae Aquarium, A Coruña. Over 3000 images were analyzed.

The performance tests include benchmarks that test system efficiency at multiple combinations with fish, no fish, several fishes, different positions of the fish, different combinations of turbidity and luminosity, bubbles, etc., while the robustness benchmarks test the correctness of the proposed method in terms of the detection and measurement of fish. The system that has been used to test the developed software is a medium-performance computer with the following hardware:

- Intel Core i7-3770 CPU (4 cores, 8 threads)
- NVIDIA GT 640 graphics card
- 16 GB of 1600 MHz DDR3 RAM
- WDC WD10EZRX Hard Disk

The operating system used to run the tests was Windows 7 × 64, the compiler used to build the code was VC++ v110 and the OpenCV version was 3.0.

The execution of the technique was tested in real time. The graphs, with the execution times of the proposed technique in two videos, are shown as follows. Figure 8 shows the system tested against the Video 1 scene. This scene contains a single test small fish (9 cm of length), which swims at different depths. Moreover, this video contains high amounts of noise, bubbles, and changing light, which in turn present a serious challenge for our detectors. The calculation of the disparity map of the detected fish is not too complex since the fish has bright colors and several identifying features, which allows the algorithm to obtain good results. The performance results of the second test can be seen in Figure 9, where the system is tested against the Video 2 scene. This is a more complex scene, since there are multiple fish swimming in the recorded tank and the fish are 1 m in length.

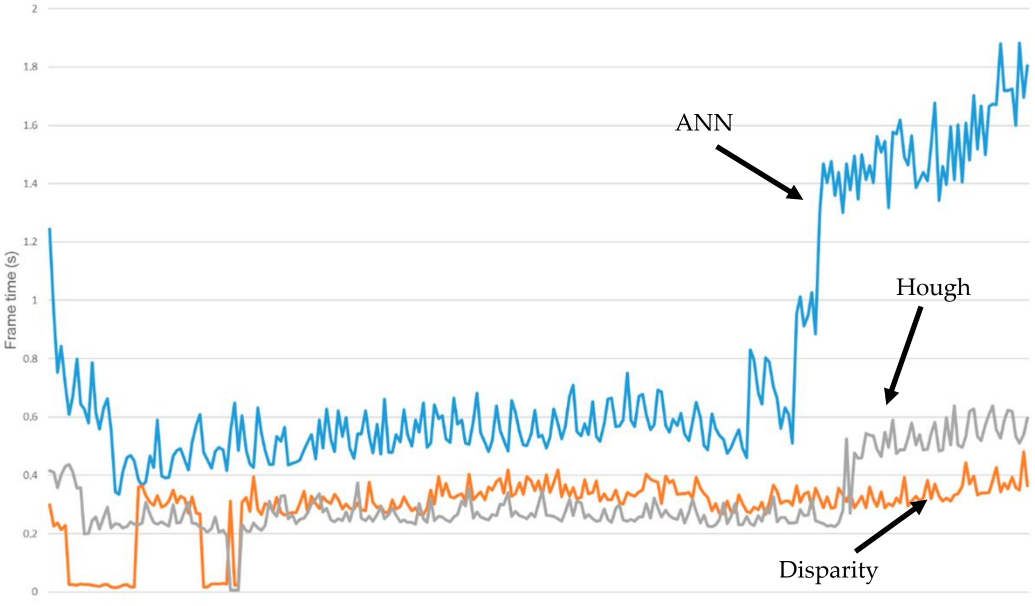


Figure 8. Algorithms execution on Video 1.

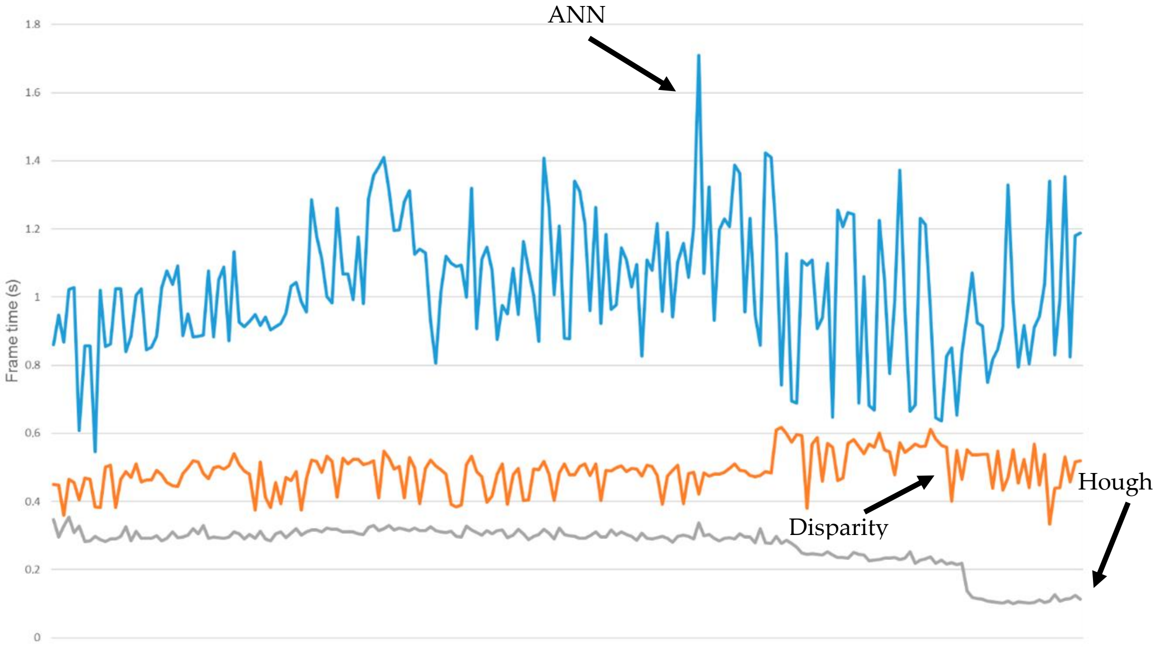


Figure 9. Algorithms execution on Video 2. Blue: ANN; Orange: Disparity; Grey: Hough.

The highest peaks in Figure 8 are produced by changes in light and shadows when the fish is next to the walls of the tank. Thus, the background subtraction has changed, as Figure 10a,b show, and, as Hough algorithm fails to detect the fish, the ANN must analyze the images, which uses more performance time.

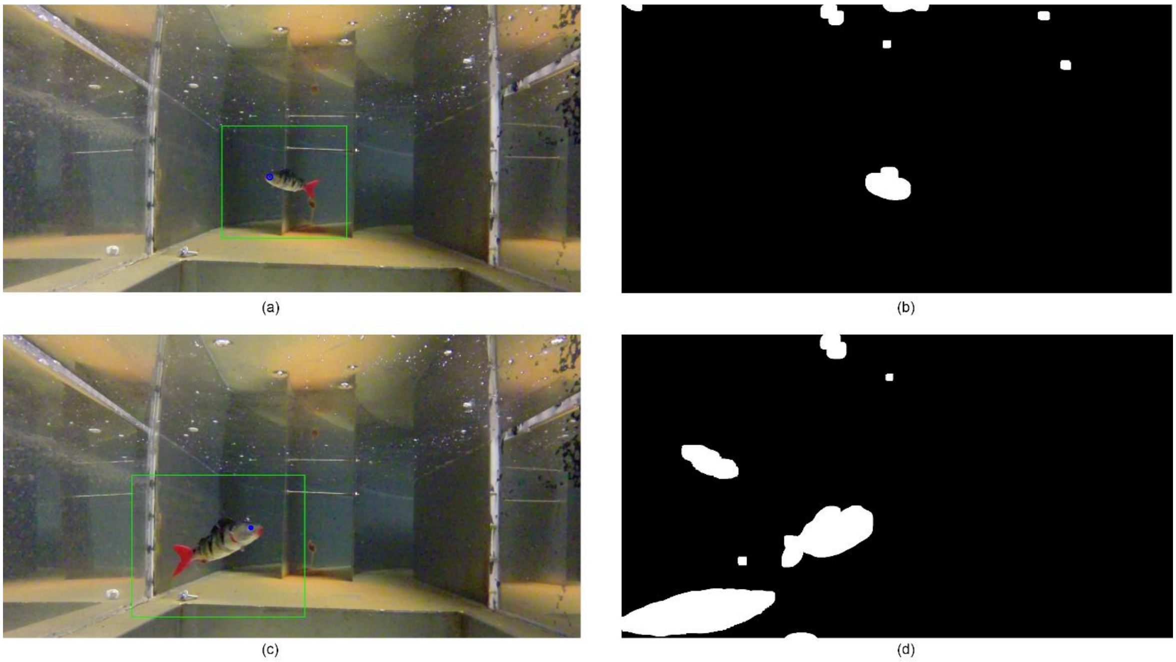


Figure 10. Video 1. (**a**) small fish in the frame; (**b**) background subtraction; (**c**) fish position in frame where the fish is close to walls; (**d**) background subtraction with shadows which create new shapes in the background mask.

Both graphs show that, in order to achieve real-time processing, using only the Hough algorithm allow us to obtain 1 frame per second, but if we want to increase the accuracy when this method does not detect the fish, if we use the ANN then the runtime peaks during processing exceed one second. This is due to the pre-processing which has to be performed to reduce noise, the impact of the changing light, and validation of the results provided by both algorithms (Hough and ANN). Using Deep Learning with the trained ANN, when the Hough algorithm fails, the system has a 74% accuracy rate with the validation set and almost 100% with the training set.

In the tests the specimens were European perch (Perca fluviatilis) and brown trout (Salmo trutta), while in the second test (Figure 11) they were Atlantic wreckfish (Polyprion americanus). About 3000 images have been analyzed in these experiments and the results of the measurements obtained by the system have been compared with those obtained manually.

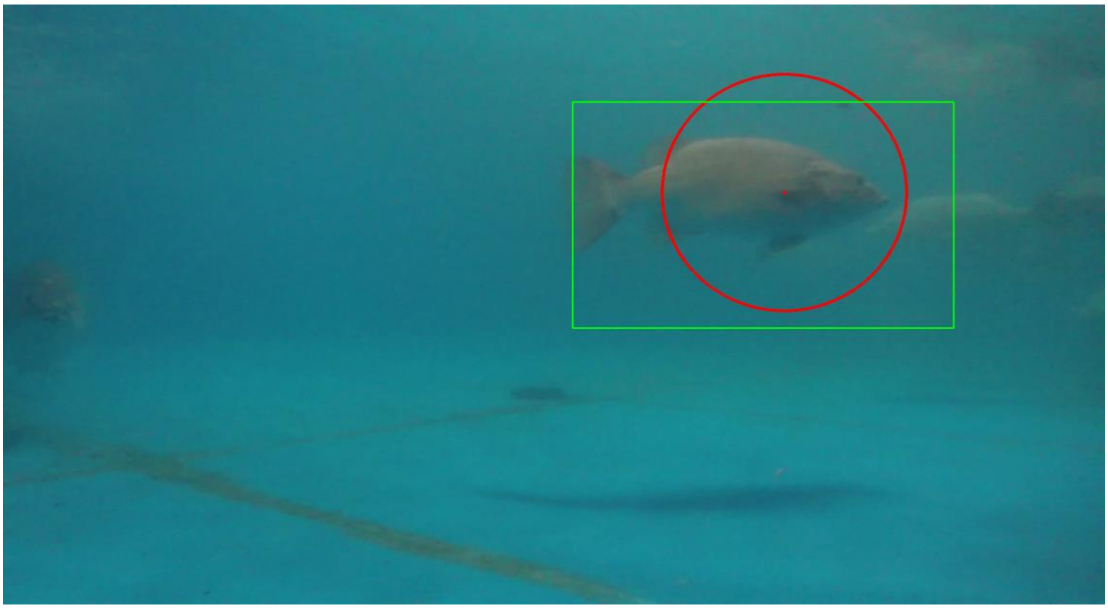


Figure 11. Video 2 with a frame of the fish detection algorithm. correctly detecting a fish in a difficult image. The small differences in the image tones, turbidity, noise and multiple fish in invalid positions present serious challenges that the classifier is able to overcome. The detected fish is highlighted in red, and the candidate ROI in green.

Table 1 shows the results obtained by the method proposed in this paper operating in real situations in a tank with a school of fish of different species. As can be seen, the average size of the fish can be calculated for the different species with good precision: in European perch about 6.8% error over the real fish size, in brown trout about 10.7%, and in Atlantic wreckfish about 7%. As can be seen in the results, the system, combining deep learning with artificial vision techniques, has a detection accuracy of 74% in large fish and over 90% in medium fish. As can be seen in this precision, false negatives are not obtained because the fish is always detected when it appears in any of the images (30 frames per s), which is one of the advantages of this technique; however, there is a false positive ratio, where the algorithm indicates the detection of fish when there are really none. This ratio is higher with large fish mainly due to changes in brightness, bubbles, shadows and noise that tend to cover larger areas of the image.

Table 1. Results obtained with real fish of different species.

Results	European Perch	Brown Trout	Atlantic Wreckfish
Avg. Measured Size (cm)	8.8	6.5	92.6
Std. Dev. Measured Size (cm)	0.8	0.8	8.4
Avg. Absolute Error (cm)	0.6	0.9	6.5
Std. Dev. Absolute Error (cm)	0.6	0.7	5.2
Avg. Relative Error	0.07	0.12	0.07
Std. Dev. Relative Error	0.09	0.09	0.06
True Positives	620	600	182
Detected False Positives	44	31	47
Precision	0.93	0.95	0.74
False Positive Ratio	0.07	0.05	0.26

5. Conclusions

In this paper we show an application of artificial intelligence techniques for detecting fish. The advantage of the method proposed in this study is that there is no need to have in-depth knowledge of the fish species to detect them. However, similarly to other artificial vision techniques, it remains conditional on good image quality and good background subtraction. Other advantages of this technique are that detection is performed in real time and the recording is made in stereo format, and these data can be subsequently used to calculate the size of the fish.

The Hough Transform method, even with the usage of MSER to increase robustness, has difficulties with noisy images and is also highly dependent on light conditions. For this reason, an alternate fish detection method has also been provided, Deep Learning with an ANN. Comparing the Hough Transform method with the use of neural networks, the latter correctly detect the fish in the environment for which they are trained, but when the conditions vary, the ANN should be re-built and re-trained for the new scenario. Other works are based on the shape or color of the fish or background to obtain a segmentation that allows their detection, thereby changing species or conditions, would not make this method particularly useful. On the other hand, comparison with patterns has a major drawback consisting of the need to have enough images of the species to be detected; so our proposal could perform a search among the available patterns, and using both methods (Hough Transformation algorithm and, when it fails, the ANN) we can obtain an accuracy of 74%.

The first step to test the proposed technique was to include more than one species in the same environment so that several specimens of different species may appear in the same frame. Secondly, the tests have been conducted in an environment as close as possible to reality, where turbidity and changes in brightness have a greater impact. The results obtained have been satisfactory; in a real video of 3000 s (50 min) of a school of fish in an aquarium, the method has been able to detect 74% of images in which a fish appeared. This means that it is feasible to carry out a real-time fish detection system (one image per second) that is capable of efficiently detecting fish and subsequently detecting their size. In this way it could be applied in practical cases such as in fish farms or in studies of wild populations, where the objective is to analyze the evolution of the size of a school of fish.

Author Contributions: Conceptualization, Ã.J.R.-D. and O.A.M.; Eye detection: O.A.M. and Ã.J.R.-D.; Size estimation: Ã.J.R.-D. and O.A.M.; writing—original draft preparation, Ã.J.R.-D. and O.A.M.; writing—review and editing, J.R.R., M.G. and J.P.; supervision, J.R.R., M.G. and J.P.; project administration, J.R.R.; funding acquisition, J.R.R. and Ã.J.R.-D. All authors have read and agree to the published version of the manuscript.

Funding: This work was supported by FEDER funds and the Spanish Ministry of Economy and Competitiveness (Ministerio de Economía y Competitividad) (Ref. CGL2012-34688). The authors also acknowledge support from the Spanish Ministry of Education, Culture and Sport (FPI grant Ref. BES-2013-063444). This project was also supported by the Spanish Ministry of Economy and Competitiveness through the project BIA2017-86738-R and through the funding of the unique installation BIOCAI (UNLC08-1E-002, UNLC13-13-3503) and the European Regional Development Funds (FEDER) by the European Union. Additional support was offered by the Consolidation and Structuring of Competitive Research Units—Competitive Reference Groups (ED431C 2018/49) and Accreditation, Structuring, and Improvement of Consolidated Research Units and Singular Centers (ED431G/01), funded by the Ministry of Education, University and Vocational Training of the Xunta de Galicia endowed with EU FEDER funds. Last, the authors also acknowledge research grants from the Ministry of Economy and Competitiveness, MINECO, Spain (FEDER CTQ2016-74881-P).

Acknowledgments: The authors would also like to thank the managers and personnel of the Finisterrae Aquarium of A Coruña for their support, technical assistance and for allowing the unrestricted use of the Finisterrae facilities and of the Centre for Technological Innovation in Building and Civil Engineering (CITEEC).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The first step is the conversion of the image colors from RGB to grayscale, using the following formula:

$$Grayscale = r \times R + g \times G + b \times B, \quad \text{(A1)}$$

For testing, the default values are maintained (r = 0.299, g = 0.587 and b = 0.114) provided by OpenCV [27], as they are valid in these cases. However, in other environments they may need to be varied in order to obtain a higher contrast in the frame, facilitating the following steps.

Next, in order to reduce noise and improve the robustness of the background subtraction technique, a Gaussian Blur filter is applied to the input images. This filter uses a Gaussian matrix as its underlying kernel, which produces output pixel values resulting only upon the pixel values in its neighborhood as processed by the convolution kernel. The following formula represents the 2D Gaussian applied:

$$G_0(x, y) = A^{\left(\frac{-(x-\mu_x)^2}{2\sigma_x^2} + \frac{-(y-\mu_y)^2}{2\sigma_y^2}\right)} \quad \text{(A2)}$$

where A is the amplitude, μ is the mean and σ is the variance per each of the variables x and y. Since the input images are too noisy and complex to obtain good circle matches with the Hough Transform, several steps have to be applied to the input images before fisheyes can be detected with the aforementioned approach.

The first step computes a Mean Shift Filter [32,33] in order to smooth textures, reduce noise and therefore make eye color validation easier and more robust, having an RGB image that consists of n pixels, each of which has three values (r, g, b). Since an image is a distribution of points in a three-dimensional space, a density function can be defined as:

$$f(\vec{x}) = \left\{ \vec{x}_i \middle| i = 1, \ldots, n \right\} \quad \text{(A3)}$$

Local maximum points of the density function can be detected by Mean Shift Analysis as illustrated below. In Figure A1, the dark circles indicate local maximum points.

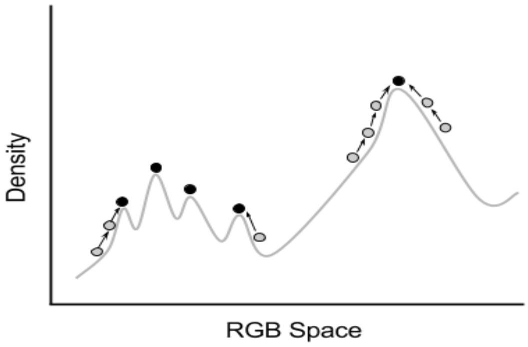


Figure A1. Mean Shift Analysis. dark circles: local maximum points; light circles: local points.

Any point in the three-dimensional space converges to a local maximum, therefore smoothing the image. In addition, position information can also be utilized so that distant objects are not labeled by the same colors. In order to accomplish this, the three dimensional vector (*r*, *g*, *b*) can be expanded to a 5-dimensional vector (*x*, *y*, *r*, *g*, *b*), which includes location information.

Moreover, since the edge contrast of an image should be enhanced, the Hough transform is more robust; an unsharp mask was also applied to the images in order to achieve this objective. The sharpening process utilizes a slightly blurred version of the original image (Gaussian blur), which is subsequently subtracted from the original to detect its edges, effectively creating an unsharp mask. Contrast is then increased, along with these edges, using the aforementioned mask:

$$Out(x,y) = 1.5\ \times\ Input(x,y) - 0.5 \times G(x,y) \tag{A4}$$

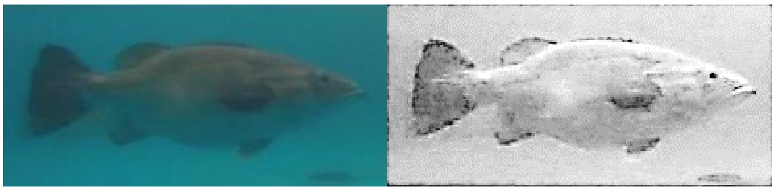
Since this type of filter can cause artifacts on edge borders, which may lead to double edges, the Laplacian of the original image is also added to the sharpened image:

$$Sharp(x,y) = 1.5 \times Input(x,y) - 0.5 \times G(x,y) - W \times (Input(x,y) \times s \times L(x,y)) \tag{A5}$$

W being the weight, *s* the scale and *L*(*x*,*y*) the Laplacian:

$$L(x,y) = \frac{\partial^2 Input}{\partial x^2} + \frac{\partial^2 Input}{\partial y^2} \tag{A6}$$

Figure A2 shows an example of the transformation in greyscale and the image filtering.



input output.

Figure A2. Example of image filtering

Appendix B

Figure A3 shows some of the frames extracted from original images that were used in the training and validation phases of the artificial neural network.

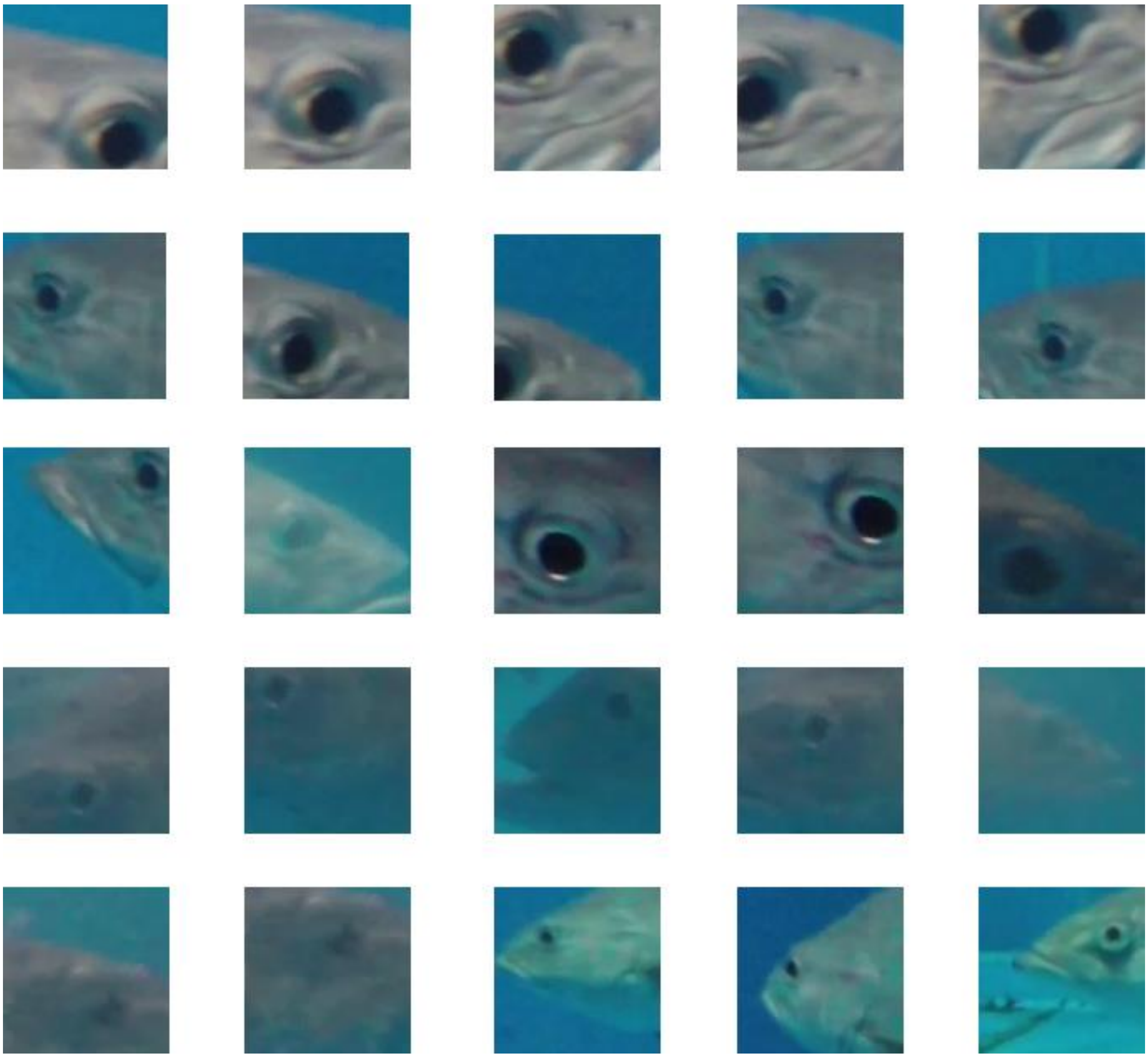


Figure A3. *Cont.*

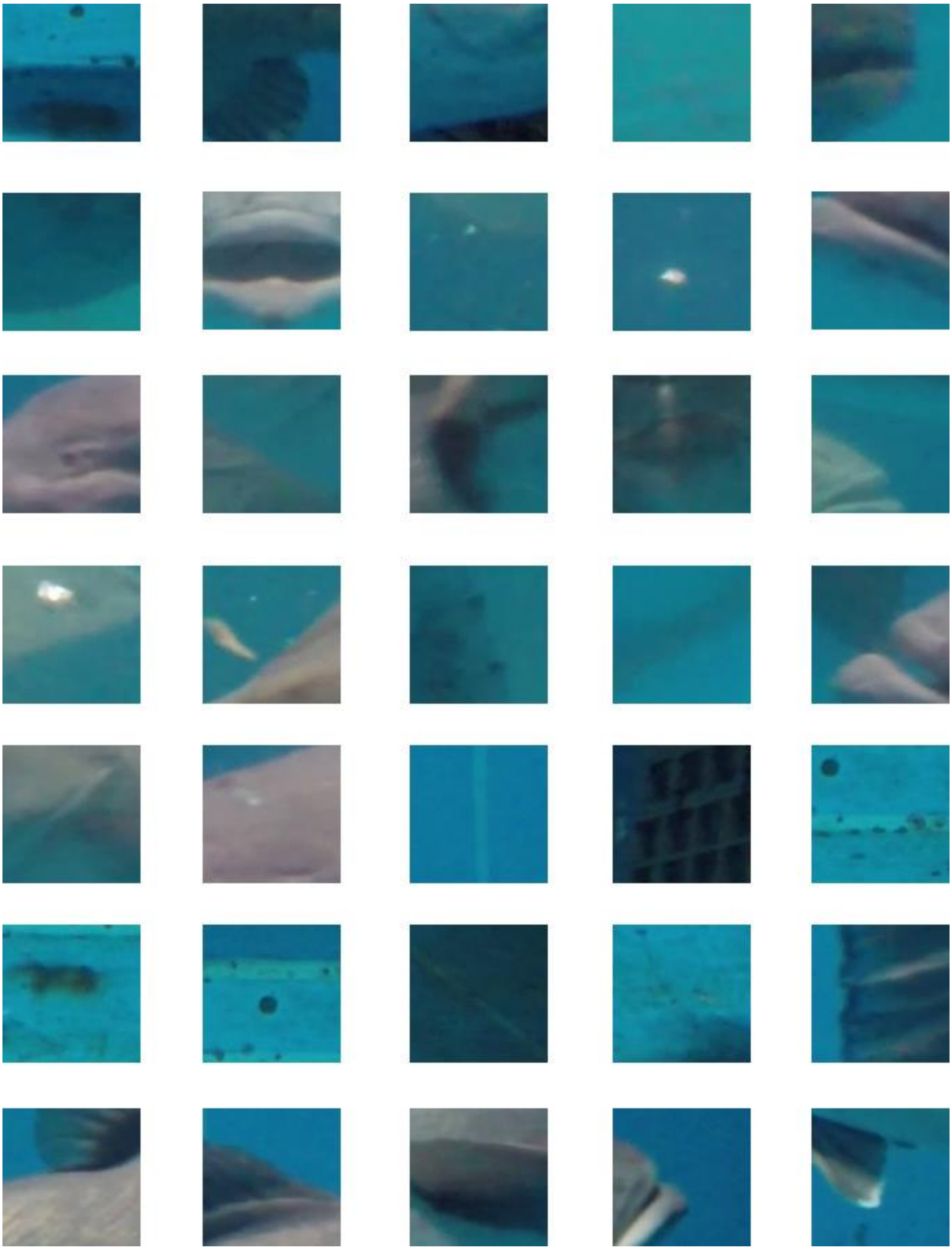


Figure A3. Extract of the dataset used for the ANN training and validation.

References

1. Castro-Santos, T.; Haro, A.; Walk, S. A passive integrated transponder (PIT) tag system for monitoring fishways. *Fish. Res.* **1996**, *28*, 253–261. [CrossRef]
2. Ashraf, M.; Lucas, J. Underwater Object Recognition Techniques Using Ultrasonics. Presented at the IEEE Oceans 94 Osates, Brest, France, 13–16 September 1994.

3. Bakar, S.A.A.; Ong, N.R.; Aziz, M.H.A.; Alcain, J.B.; Haimi, W.M.W.N.; Sauli, Z. Underwater detection by using ultrasonic sensor. In *AIP Conference Proceedings*; AIP Publishing LLC: Melville, NY, USA, 2017; Volume 1985, p. 020305.
4. Ghobrial, M. *Fish Detection Automation from ARIS and DIDSON SONAR Data*; University of Oulu: Oulu, Finland, 2019.
5. Schettini, R.; Corchs, S. Underwater Image Processing: State of the Art of Restoration and Image Enhancement Methods. *EURASIP J. Adv. Signal Process.* **2010**, *2010*, 746052. [CrossRef]
6. Trucco, E.; Olmos-Antillon, A.T. Self-Tuning Underwater Image Restoration. *IEEE J. Ocean. Eng.* **2006**, *31*, 511–519. [CrossRef]
7. Yamashita, A.; Fujii, M.; Kaneko, T. Color Registration of Underwater Images for Underwater Sensing with Consideration of Light Attenuation. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 4570–4575.
8. Lee, D.; Kim, G.; Kim, D.; Myung, H.; Choi, H.-T. Vision-based object detection and tracking for autonomous navigation of underwater robots. *Ocean Eng.* **2012**, *48*, 59–68. [CrossRef]
9. Puertas, J.; Cea, L.; Bermúdez, M.; Pena, L.; Rodríguez, A.; Rabuñal, J.; Balairón, L.; Lara, A.; Aramburu, E. Computer application for the analysis and design of vertical slot fishways in accordance with the requirements of the target species. *Ecol. Eng.* **2011**, *48*, 51–60. [CrossRef]
10. Zhang, Z.; Lee, D.; Zhang, M.; Tippetts, B.; Lillywhite, K. Object recognition algorithm for the automatic identification and removal of invasive fish. *Biosyst. Eng.* **2016**, *145*, 65–75. [CrossRef]
11. Ravanbakhsh, M.; Shortis, M.; Shafait, F.; Mian, A.; Harvey, E.; Seager, J. Automated Fish Detection in Underwater Images Using Shape-Based Level Sets. *Photogramm. Record* **2015**, *30*, 46–62. [CrossRef]
12. Harveya, E.; Cappob, M.; Shortisc, M.; Robsond, S.; Buchanane, J.; Speareb, P. The accuracy and precision of underwater measurements of length and maximum body depth of southern bluefin tuna (Thunnus maccoyii) with a stereo–video camera system. *Fish. Res.* **2003**, *63*, 315–326. [CrossRef]
13. Rodríguez, A.; Bermúdez, M.; Rabuñal, J.; Puertas, J.; Dorado, J.; Pena, L.; Balairón, L. Optical Fish Trajectory Measurement in Fishways through Computer Vision and Artificial Neural Networks. *J. Comput. Civ. Eng.* **2011**, *25*, 291–301. [CrossRef]
14. Shortis, M.R.; Ravanbakskh, M.; Shaifat, F.; Harvey, E.S.; Mian, A.; Seager, J.W.; Culverhouse, P.F.; Cline, D.E.; Edgington, D.R. A review of techniques for the identification and measurement of fish in underwater stereo-video image sequences. In *Videometrics, Range Imaging, and Applications XII; and Automated Visual Inspection*; International Society for Optics and Photonics: Bellingham, WA, USA, 2013; p. 8791.
15. Rico-Diaz, A.J.; Rodríguez, A.; Villares, D.; Rabuñal, J.; Puertas, J.; Pena, L. A Detection System for Vertical Slot Fishways Using Laser Technology and Computer Vision Techniques. In *International Work-Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2015; pp. 218–226.
16. Cui, S.; Zhou, Y.; Wang, Y.; Zhai, L. Fish Detection Using Deep Learning. *Appl. Comput. Intell. Soft Comput.* **2020**, *2020*, 1–13. [CrossRef]
17. Shevchenko, V.; Eerola, T.; Kaarna, A. Fish Detection from Low Visibility Underwater Videos. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018.
18. OPENCV: Open Source Computer Vision. Available online: http://opencv.org (accessed on 26 October 2020).
19. Comaniciu, D.; Meer, P. Mean shift analysis and applications. In Proceedings of the Seventh IEEE International Conference, Munich, Greece, 19 July 1999; pp. 1197–1203.
20. Comaniciu, D.; Meer, P. Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [CrossRef]
21. Suzuki, S.; Abe, K. Topological Structural Analysis of Digitized Binary Images by Border Following. *Comput. Vis. Graph. Image Process.* **1985**, *30*, 32–46. [CrossRef]
22. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI* **1986**, *8*, 679–698. [CrossRef]
23. Nistér, D.; Stewénius, H. Linear Time Maximally Stable Extremal Regions. In Proceedings of the Computer Vision–ECCV 2008, Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 183–196.
24. Illingworth, J.; Kittler, J. The adaptive hough transform. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *9*, 8. [CrossRef] [PubMed]

25. Yuen, H.K.; Princen, J.; Illingworth, J.; Kittler, J. Comparative study of Hough Transform methods for circle finding. *Image Vis. Comput.* **1990**, *8*, 71–77. [CrossRef]
26. Lamb, T.D.; Collin, S.P.; Pugh, E.N. Evolution of the vertebrate eye: Opsins, photoreceptors, retina and eye cup. *Nat. Rev. Neurosci.* **2007**, *8*, 960–976. [CrossRef] [PubMed]
27. Land, M.F.; Nilsson, D.-E. *Animal Eyes*; Oxford University Press: Oxford, UK, 2012.
28. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
29. Rico-Diaz, A.J.; Rodríguez, A.; Puertas, J.; Bermúdez, M. Fish monitoring, sizing and detection using stereovision, laser technology and camputer visio. In *Multi-Core Computer Vision and Image Processing for Intelligent Applications*; Mohan, S., Vani, V., Eds.; IGI Global: Hershey, PA, USA, 2016.
30. Bouguet, J.Y. MATLAB Calibration Tool. Available online: http://www.vision.caltech.edu/bouguetj/calib_doc/ (accessed on 26 October 2020).
31. Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [CrossRef]
32. Je, C.; Park, H.-M. Optimized hierarchical block matching for fast and accurate image registration. *Signal Process. Image Commun.* **2013**, *28*, 779–791. [CrossRef]
33. Rodriguez, A.; Rico-Diaz, A.J.; Rabuñal, J.R.; Puertas, J.; Pena, L. Fish Monitoring and Sizing Using Computer Vision. In *Bioinspired Computation in Artificial Systems, Proceedings of the International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2015, Elche, Spain, 1–5 June 2015*; Ferrández Vicente, J.M., Álvarez-Sánchez, J.R., de la Paz López, F., Toledo-Moreo, F.J., Adeli, H., Eds.; Springer: Cham, Switzerland, 2015; pp. 419–428.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.