Video-based tracking of fishes in the Lisbon Oceanarium

João Teixeira
Instituto Superior Técnico

H. Sofia Pinto
INESC-ID, IST

Núria Baylina
Oceanário de Lisboa

Alexandre Bernardino
Institute for Systems and Robotics, IST

***Abstract*—There are many challenges on tracking fishes from videos taken from single cameras outside fish tanks, such as frequent occlusion of the targets, the presence of schools, high visual similarities between different individuals, even from different species, sudden changes in motions, and very different light conditions. This paper presents our solution system that detects and tracks fish of different species in tanks in the Oceanário de Lisboa. We make a thorough evaluation of several state-of-the-art detectors in those types of scenarios and improve on previous work addressing detection-tracking-classification on the same scenario. Here we show how the tracking methodology was improved, replacing greedy data association and better tuning of the components. Additionally, we introduce a new step of contrast equalisation to mitigate the challenging illumination conditions of the main tank scenario.**

I. Introduction

This paper focuses on tracking fishes in videos recorded by a single static camera positioned outside the glass walls of the tanks in the Oceanário de Lisboa, an oceanarium that recreates real-life underwater conditions of different maritime habitats. Video based tracking systems can help automate monitoring fishes in tanks, e.g for real-time anomalous behaviour detection. There are many challenges in this task such as the frequent occlusion of the targets, the presence of schools, high visual similarities between different individuals, even from different species, sudden changes in motions, and very different light conditions. Some work has been done in tracking fish in the wild using underwater cameras. For instance the Fish4Knowledge project reported relevant research in topics such as detection, tracking and classification of species [7], [9], [8] in large datasets of videos recorded from multiple underwater locations. However, in an aquarium setting the amount of fish per water volume is typically higher than in open waters, which poses additional challenges to the detection and tracking technology due to frequent crossings and occlusions. The work reported here describes our system that detects and tracks fish of different species in tanks. In particular, we targeted two tanks with very different characteristics: the main tank and the coral reef tank (see Fig. 1). In a previous work [2] we have addressed this problem in a detection-tracking-classification pipeline. Here, we improve that method in several aspects. First, we make a thorough evaluation of several state-of-the-art detectors in those types of scenarios. Second, some parts of the tracking methodology were improved, like replacing greedy data association and better tuning of the components. Additionally, we introduce a new step of contrast equalisation to mitigate the challenging illumination conditions of the main tank scenario (see Fig. 1) . New datasets were manually built in each environment from video sequences of 5 minutes each.

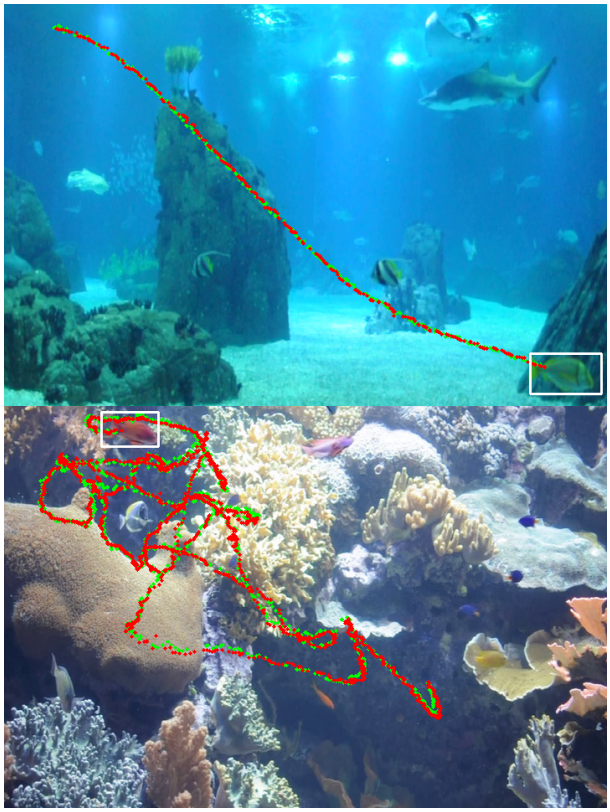

Fig. 1. Example of the tracking results in the two considered scenarios. Top: the main tank. Bottom: the coral reef tank. In green is displayed the ground truth trajectory and in red the estimated trajectory. Note the differences in image content (color diversity, contrast, distance to the observed fish) and fish behaviour (faster and more uncertain motions in the coral reef tank).

II. Methods

The tracking system is composed of three main modules (see Fig. 2): contrast equalisation, detection and tracking .

A contrast equalisation method is used to improve image quality in the main tank videos. The approach used was inspired by the one in [4], as it seemed suitable to apply to our conditions. We convert the frame from its original color space RGB to CIELAB and apply the CLAHE operation to the L channel, that represents the lightness of the color. After the

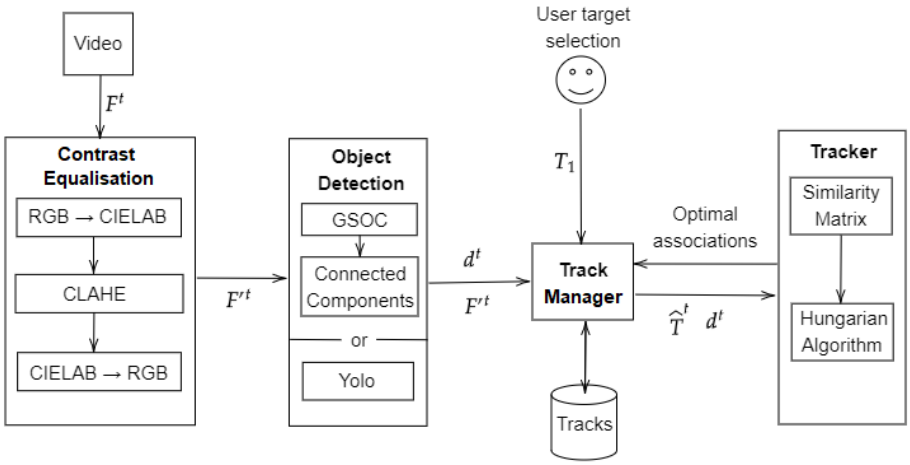

Fig. 2. An overview of the system's architecture.

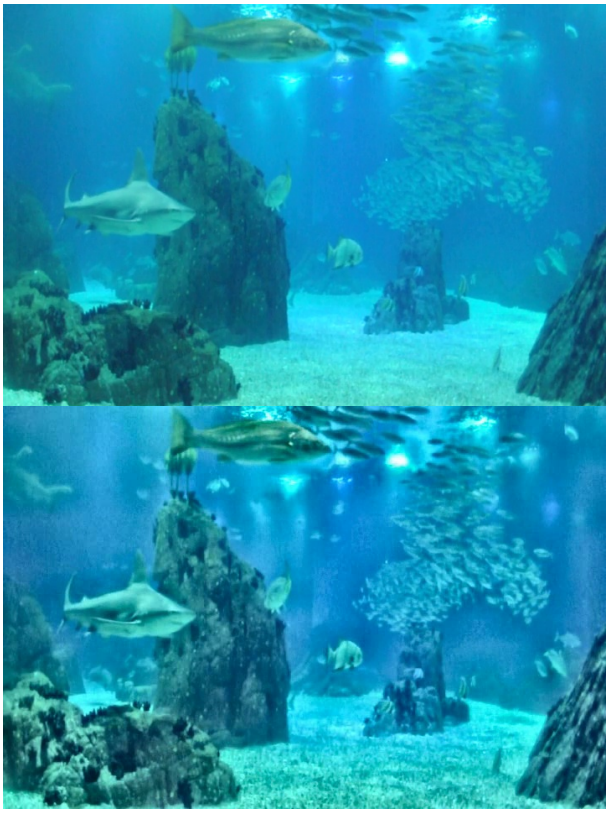

Fig. 3. Result of applying the contrast equalisation technique to a frame from the video of the main tank. Top: original; bottom: after equalisation.

transformation, we convert the frame back to RGB to be used in the following modules. Fig 3 shows the result of applying this technique to one of the frames.

The object detection module locates the fish in each frame. We evaluate two types of detectors: background subtraction (AGMM [13], KNN [14], Sigma-Delta [5], ViBE [1], PBAS [3], Lobster [10], PAWCS [11], SubSENSE [12] and

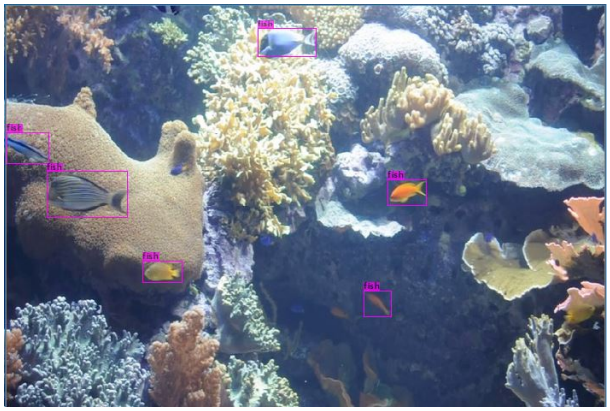

Fig. 4. The bounding boxes (in pink) predicted by YOLO detector.

GSOC[1]) and one deep learning detector (YOLO [6]). The background subtraction algorithms output a segmented frame where the pixels are classified as either background or foreground. After that, a Connected Components algorithm is applied to combine neighbouring foreground pixels into a detected object, and then the bounding boxes of each object are obtained. The best overall background subtraction algorithm for our environments, according to our tests, discussed in the following chapter, was the GSOC. The deep learning detector directly outputs the list of the detected objects as bounding boxes. An example of the bounding boxes predicted by YOLO is shown in Fig. 4.

The final module is related to the tracking step. We pair each object d_j from the list of detected objects with an existing track $\hat{T}_k$. Tracking is performed through data association using colour and position features. The color similarity S_c is computed according to

$$S_c(\hat{T}_k, d_j) = 1 - D(H_{\hat{T}_k}, H_{d_j}) \quad (1)$$

[1]https://github.com/opencv/opencv_contrib

where D is the distance between the two histograms, $H_{\hat{T}_k}$ and H_{d_j}, that is computed according to

$$D(H_{\hat{T}_k}, H_{d_j}) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_{\hat{T}_k} \bar{H}_{d_j} N^2}} \sum_I \sqrt{H_{\hat{T}_k}(I) H_{d_j}(I)}} \quad (2)$$

where N is the number of bins. For the position similarity S_p, the center of the bounding boxes is considered, and is calculated as

$$S_p(\hat{T}_k, d_j) = 1 - \frac{\sqrt{(\hat{T}_{k_x} - d_{j_x})^2 + (\hat{T}_{k_y} - d_{j_y})^2}}{D_{max}} \quad (3)$$

where D_{max} is the maximum Euclidean distance between two pixels in the frame. After the association step, a verification of the results is performed in order to prevent the tracked objects to be matched with detected objects that are too distant or very different in color. To achieve this, a color threshold (c_{thr}) and a position threshold (p_{thr}) are used to regulate the associations, discarding the ones that exceed those limits. As this module is highly dependent on the results from the detection module, we need to account for possible missed objects. Therefore, the system allows the tracks to not have a corresponding detection in every frame.

We have performed improvements to related work in three main aspects: (i) a colour history descriptor to make the tracking more robust to appearance variation, (ii) temporary track management to add and remove new tracks with added robustness, and (iii) movement prediction with custom models of measurement and process noises learned from the data in each scenario. Regarding the first improvement, the color history is achieved by computing an average between the color histogram of the full track ($H_{\hat{T}_k}$) and the histogram of the detection (H_{d_j}), according to

$$H^t_{\hat{T}_k} = (1 - \gamma) \cdot H^{t-1}_{\hat{T}_k} + \gamma \cdot H_{d_j} \quad (4)$$

In a second improvement, we distinguish between temporary and permanent tracks. Every track that is created, is first classified as temporary. Only after a few consecutive successful associations it may change to a permanent track. The association step is divided in two phases: first we associate the permanent tracks to detected objects, and then we perform a second association step between the temporary tracks and the remaining detected objects. In Fig. 5 we can see an example where temporary tracks improve the tracking quality. The last improvement was carried out through movement prediction using the Kalman Filter. The algorithm was tuned for each tank separately given the distinct kinds of fishes that exhibit different motion behaviours. To better approximate the models, the trajectories that had been manually selected for the evaluation were used to compute the Kalman Filter parameters. These parameters included the covariance matrices of the measurement noise and process noise. In Fig. 6, we can see an example of how movement prediction can help the tracking.

III. Experimental Results

In a first experiment we compare the different background subtraction techniques (Fig. 7). A background subtraction dataset was built by selecting frames from videos of both tanks and manually labelling the pixels as foreground/background, as seen in Fig. 8. To evaluate the algorithms we use the processing speed and the F_1-score metric. The F_1-score is computed from the Precision (P) and Recall (R) values based on the number of True Positives, False Positives and False Negatives obtained from each pixel classification as either background or foreground, according to

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \quad (5)$$

We can see that the algorithm with the best F_1-Score is FgSegNet but looking at the processing speed, we notice that it is very slow. GSOC scored a bit lower for both videos but achieved higher processing speeds. We can also see that all algorithms performed worse in the main tank, which was expected, due to blur, the presence of schools, and more color homogeneity.

Because background subtraction algorithms may take some time to estimate the background model, the beginning of the videos may have many detection errors. To mitigate this problem, we run a second test, where we first use the entire video to build the background model and then evaluate the performance of each algorithm on a second pass through the video. This will give a better assessment of the long term performance of the methods. The results of the two-pass experiment are presented in Fig. 9. In both tanks, having a prior background model, helped improve the score of the top scoring algorithms.

Finally, we evaluated the influence of contrast equalisation in the detection results. This was done only for the main tank, which presents the most challenging conditions. Results are presented in Fig. 10. Overall, GSOC was one of the best algorithms with a good trade-off between speed and F_1-Score in most of the tests.

For the tracking evaluation, a dataset was built by manually selecting the trajectories of the fish throughout the videos. The selected targets include multiple fish species with different types of trajectories that differ in speed, direction and length. The dataset structure is presented in Table I. It was split into two groups: validation and test. First, the validation dataset was used to tune the parameters and select the tracking features to be used with each detector for both tanks. Then, the system is tested against the test dataset using the configurations obtained in the validation step. The tracking configuration is presented in Table II. In Fig. 11, we can see the full tracking pipeline results in the test dataset, both using the chosen background segmentation detector (GSOC) and a deep learning based detector (YOLO). Overall, the best tracking results were achieved by using the GSOC in the coral reef tank and by YOLO in the main tank.

In terms of computational speed, the system was able to achieve processing speeds of around 14 FPS on a machine

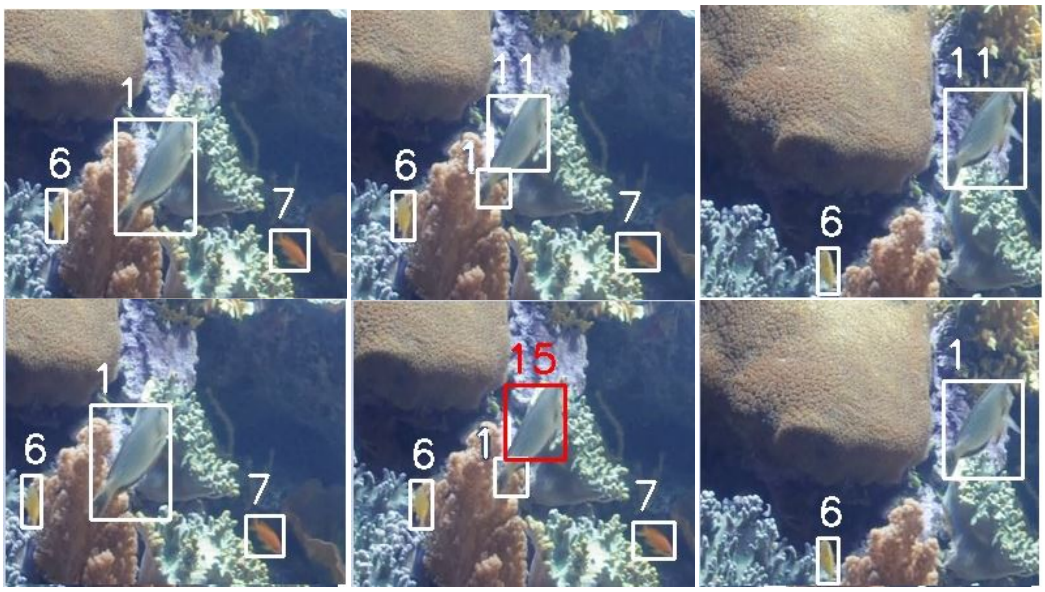

Fig. 5. Example of the use of temporary tracks in the coral reef tank. In the top row, without temporary tracks, the track #1 lost its target as another newly created track got the association. In the bottom row, using the temporary tracks (shown in red), the original track was able to keep tracing the original target.

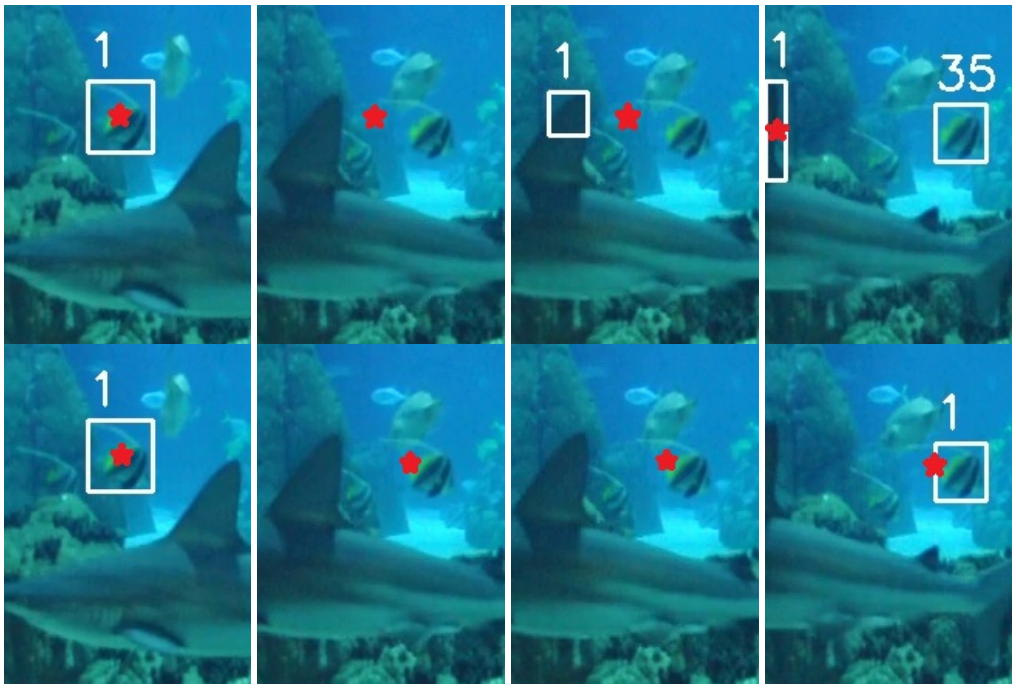

Fig. 6. Example of using Kalman Filter in the main tank for movement prediction for the track #1. The red star is the position considered for the position similarity. In the top row, without using movement prediction, when detection fails the position remains the same in the following frames, allowing incorrect associations afterwards. In the bottom row, using movement prediction, the position keeps being updated until it eventually detects the fish and recovers the original target.

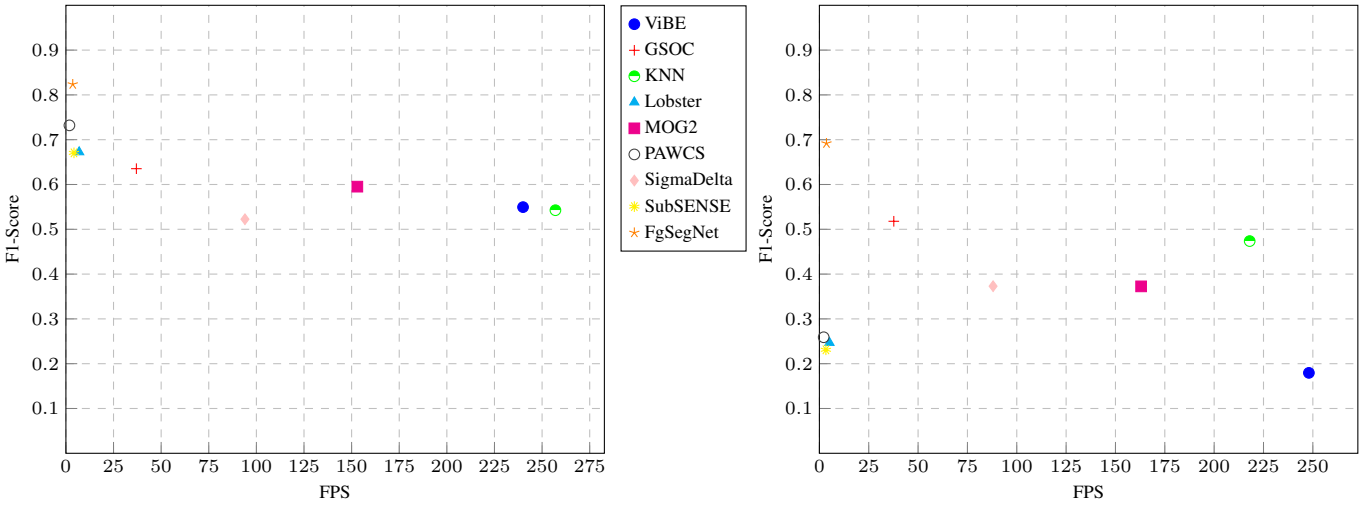

Fig. 7. F_1-score of the background subtraction algorithms in the coral reef tank (left) and the main tank (right).

Fig. 8. Example of a manually labelled frame from the background subtraction dataset.

TABLE I
TRACKING DATASET.

Tank	Trajectories	Min.Length (frames)	Max.Length (frames)	Usage
Main	15	152	827	Validation
Coral reef	17	58	1687	Validation
Main	4	197	385	Test
Coral reef	4	192	264	Test

equipped with an Intel Core i7-8750H @ 2.20GHz CPU and an NVIDIA GeForce GTX 1050 Ti GPU.

IV. CONCLUSIONS

We have presented a video based fish tracking system that was applied to video sequences captured at the Oceanário de Lisboa. We have performed a detailed evaluation of several state-of-the-art detectors in the same type of scenarios. We were able to improve the results of previous work by a thorough choice of the detection algorithms, contrast equalisation, and novel tracking features. There are still some limitations in

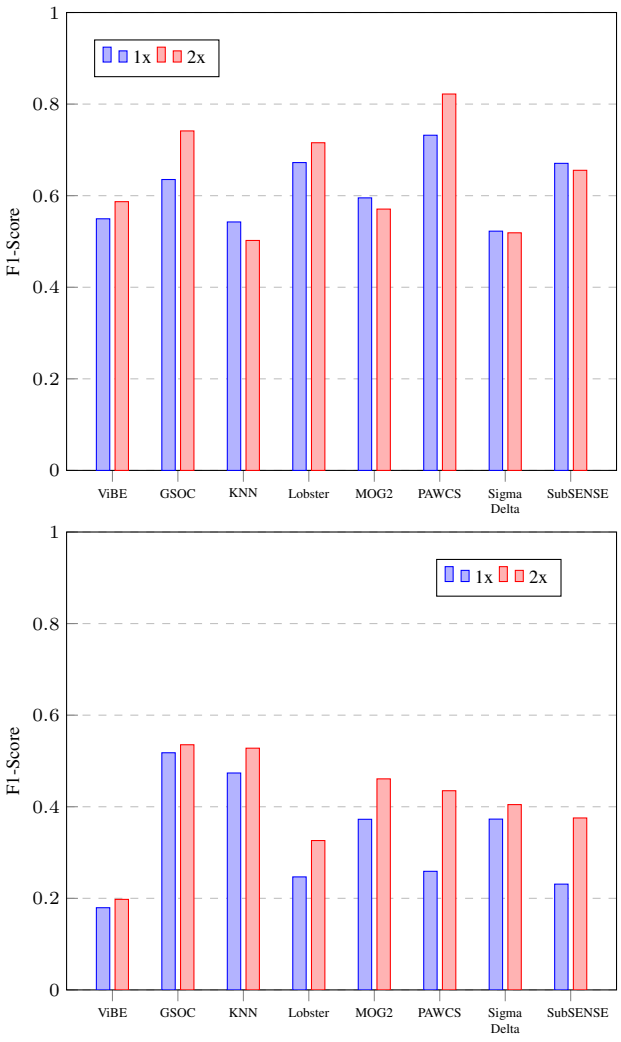

Fig. 9. F1-score comparison for two-pass variants for the coral reef tank on the top and for the main tank on the bottom.

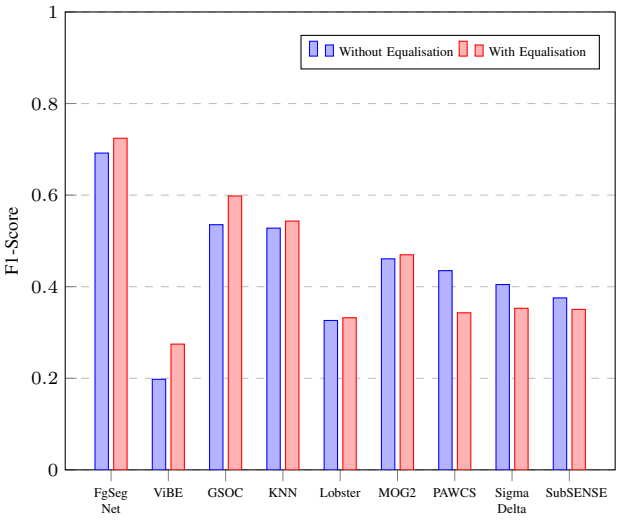

Fig. 10. F1-score of contrast equalisation for the main tank.

TABLE II
THE BEST CONFIGURATIONS ACHIEVED USING BOTH DETECTORS FOR EACH TANK.

		Baseline	
Tank	**Detector**	p_{thr}	c_{thr}
Main	GSOC	65	0.2
Main	Yolo	80	0.3
Coral	GSOC	65	0.3
Coral	Yolo	65	0.2

		Features		
Tank	**Detector**	**Color Hist.**(γ)	**Temp.Tracks**	**Mov.Prediction**
Main	GSOC	0.2	Yes	Yes
Main	Yolo	0.9	Yes	No
Coral	GSOC	0.2	Yes	Yes
Coral	Yolo	0.2	Yes	No

the ability to handle big fish schools swimming around in the main tank and the tracking of very small fish in the coral reef tank, or far away fish in the main tank.

ACKNOWLEDGEMENTS

This work was supported by FCT with the LARSyS Project UIDB/50009/2020, INESC-ID Project UIDB/50021/2020, and project VOAMAIS (PTDC/EEI-AUT/31172/2017, 02/SAICT/2017/31172).

REFERENCES

[1] O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, 2011.

[2] José Castelo, H. Sofia Pinto, Alexandre Bernardino, and Núria Baylina. Video based live tracking of fishes in tanks. In Aurélio Campilho, Fakhri Karray, and Zhou Wang, editors, *Image Analysis and Recognition*, pages 161–173, Cham, 2020. Springer International Publishing.

[3] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–43, 2012.

[4] D. Konovalov, A.Saleh, M. Bradley, M. Sankupellay, S. Marini, and M. Sheaves. Underwater fish detection with weak multi-domain supervision. *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019.

[5] Antoine Manzanera and Julien C. Richefeu. A new motion detection algorithm based on sigma-delta background estimation. *Pattern Recognition Letters*, 28(3):320–328, February 2007.

[6] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[7] Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. *VISAPP (2)*, 2008(514-519):1, 2008.

[8] 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*, ARTEMIS '10, page 45–50, New York, NY, USA, 2010. Association for Computing Machinery.

[9] Concetto Spampinato, Simone Palazzo, Daniela Giordano, Isaak Kavasidis, Fang-Pang Lin, and Yun-Te Lin. Covariance based fish tracking in real-life underwater environment. In *VISAPP (2)*, pages 409–414, 2012.

[10] P. St-Charles and G. Bilodeau. Improving background subtraction using local binary similarity patterns. In *IEEE Winter Conference on Applications of Computer Vision*, pages 509–515, 2014.

[11] P. St-Charles, G. Bilodeau, and R. Bergevin. A self-adjusting approach to change detection based on background word consensus. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 990–997, 2015.

[12] P. St-Charles, G. Bilodeau, and R. Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, 2015.

[13] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 28–31. IEEE, 2004.

[14] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773 – 780, 2006.

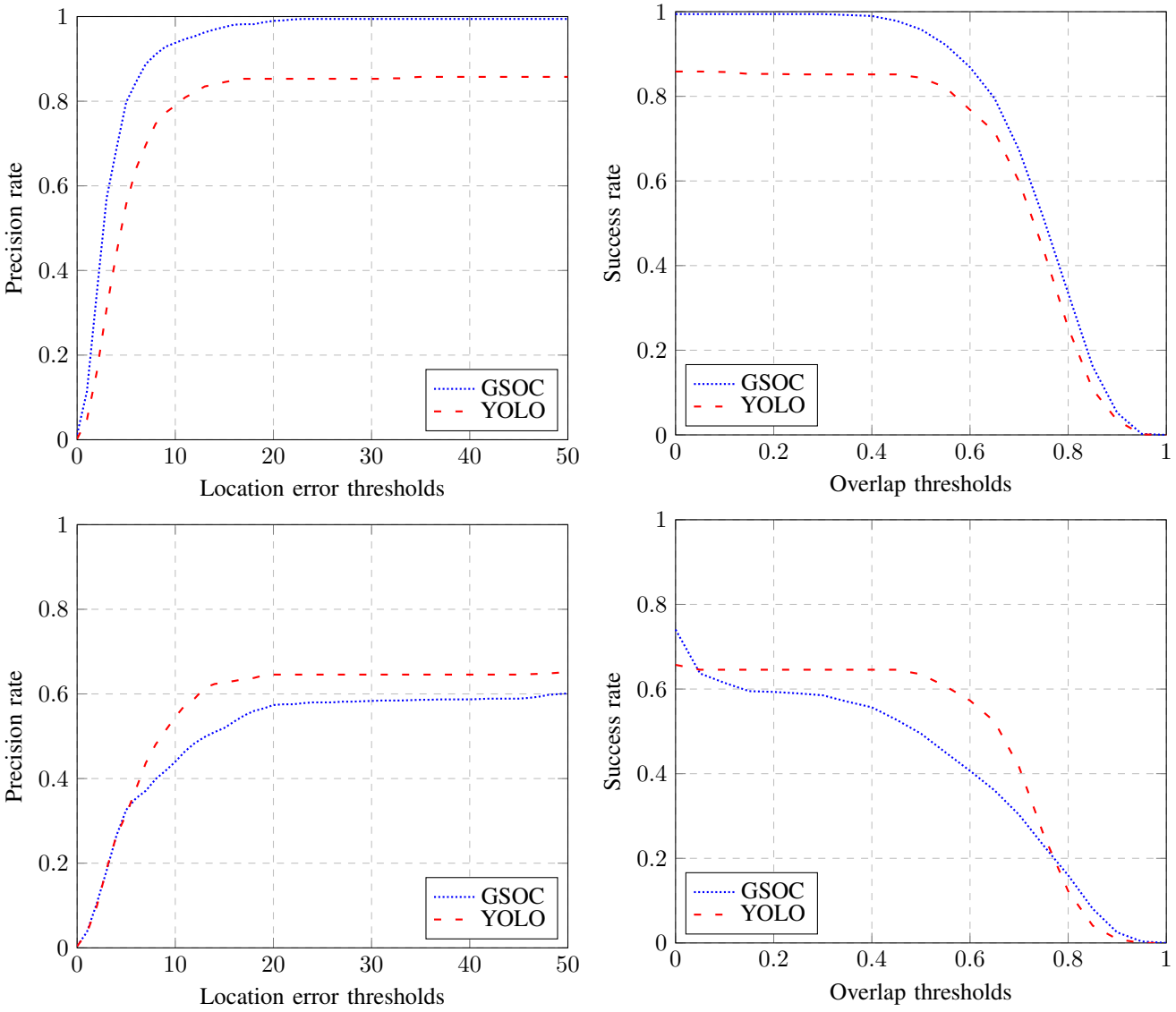

Fig. 11. Precision and Success plots for the system implementations with each object detection technique for the testing dataset in the coral reef tank; Top: coral reef tank; Bottom: main tank; Left: as a function of location error; Right: as a function of overlap.