Ecological Informatics 51 (2019) 44–51

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Real-time fish detection in complex backgrounds using probabilistic background modelling

Ahmad Salman[a,*], Salman Maqbool[b], Abdul Hannan Khan[c], Ahsan Jalal[a], Faisal Shafait[a,d]

[a] School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad 44000, Pakistan
[b] School of Mechanical & Manufacturing Engineering, National University of Sciences and Technology, Islamabad 44000, Pakistan
[c] Department of Informatics, Technical University of Munich, Boltzmannstrabe 3, 85748 Garching bei Munchen, Germany
[d] Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad 44000, Pakistan

ARTICLE INFO

Keywords:
Fish sampling
Biomass estimation
Underwater video imagery
Automatic fish detection

ABSTRACT

Computer vision and image processing approaches for automatic underwater fish detection are gaining attention of marine scientists as quicker and low-cost methods for estimating fish biomass and assemblage in oceans and fresh water bodies. However, the main challenge that is encountered in unconstrained underwater imagery is poor luminosity, turbidity, background confusion and foreground camouflage that make conventional approaches compromise on their performance due to missed detections or high false alarm rates. Gaussian Mixture Modelling is a powerful approach to segment foreground fish from the background objects through learning the background pixel distribution. In this paper, we present an algorithm based on Gaussian Mixture Models together with Pixel-Wise Posteriors for fish detection in complex background scenarios. We report the results of our method on the benchmark Complex Background dataset that is extracted from Fish4Knowledge repository. Our proposed method yields an F-score of 84.3%, which is the highest score reported so far on the aforementioned dataset for detecting fish in an unconstrained environment.

1. Introduction

Observation of organisms in their natural environments over long-time periods is critical to estimating their biodiversity. This is especially crucial for endangered species so that effective counter-measures can be planned and executed for their protection. More so, such observation is also necessary to measure the effectiveness of such conservation strategies. For marine ecosystems, marine biologists are interested in the identification of fish species, monitoring their populations, sizes, and other trends. While traditionally destructive sampling methods were used for such studies, the trend has moved towards non-destructive sampling methods (McLaren et al., 2015). Video-based sampling in underwater environments has been used to study fish species (Harvey and Shortis, 1995; Shortis et al., 2009). Such videos collected over long time periods create terabyte-scale of data, of which manual analysis is impractical. Hence, automated and efficient methods based on computer vision algorithms are needed to obtain meaningful statistics from such large datasets. However, automated approaches face challenges in the form of water murkiness, variation in lightning, erratic fish movements, and the movement of aquatic plants in the water.

Generally, video-based automatic fish sampling involves two tasks: (a) fish detection, which discriminates fish from non-fish objects in underwater videos, (b) fish species classification, which identifies the species of the detected fish. Fish sampling approaches can be adopted in either a constrained or an unconstrained environment. Earlier work in this domain mainly focused on constrained sampling. Strachan used colour and shape descriptors for 23 fish species (Strachan, 1993) and to differentiate between a particular fish specie collected from two different sources (Strachan and Kell, 1995). The system worked in a strictly confined environment for caught dead fishes. Harvey and Shortis (1995) proposed an approach where they made the fish flow through a chamber under controlled illumination. They used a stereo-camera to measure fish lengths in such conditions. Storbeck and Daan (2001) used a neural network to classify fish placed on a conveyor belt, with the camera facing perpendicular to the fish samples.

The above described methods work well in constrained environments, but are fine-tuned only towards those settings. Needless to say, they do not perform well in natural unconstrained environments with

* Corresponding author.
E-mail address: ahmad.salman@seecs.edu.pk (A. Salman).

https://doi.org/10.1016/j.ecoinf.2019.02.011
Received 18 September 2018; Received in revised form 19 February 2019; Accepted 22 February 2019
Available online 23 February 2019

freely swimming fish. This led to the development of more powerful and robust methods for the task. Rova et al. (2007) and Spampinato et al. (2010) used texture and shape information to classify fish in natural environments. A variety of machine learning techniques have also been used for the purpose such as Principal Component Analysis (PCA) (Turk and Pentland, 1991) and Linear Discriminant Analysis (LDA) (Mika et al., 1999). More recently, Hsiao et al. (2014) used a combination of both PCA and LDA to extract features and a sparse representation based classifier to detect and classify more than 1000 images of 25 different fish species with 81.1% accuracy. Huang et al. (2015) proposed a hierarchical classification method using Support Vector Machines (SVM) for multi-class classification based on the colour and textural information of fish generating 74.8% accuracy in fish species classification on a dataset having 24,000 images of 15 species. Palazzo et al. (Palazzo and Murabito, 2014) used Efficient Match Kernels (EMK) and Kernel Descriptors (KDES), which are kernel generalizations of the Bag-of-Words (BOW) and Histogram-of-Gradient (HOG) descriptors respectively, for fish species classification in underwater images yielding 84.4% classification accuracy on their dataset of 50,000 images of 10 different species. Spampinato et al. (2014) used Texton features to build a covariance model of the background for fish detection.

Background modelling techniques (Hsiao et al., 2014; Palazzo and Murabito, 2014; Spampinato et al., 2014) for fish detection focus on identifying different features that are used to model background pixels. Any changes in the subsequent frames of the video result in an offset from the expected results of the model. Such pixels can then be categorized as the foreground. Gaussian Mixture Models (GMM) (Stauffer and Grimson, 1999; Zivkovic, 2004) is also a background modelling technique, which builds a probabilistic model of the background using unsupervised generative modelling. Our work builds upon the GMM algorithm (Stauffer and Grimson, 1999) and augments it with Pixel-wise Posteriors (Bibby and Reid, 2008) to obtain better results for fish detection in natural environments.

Recent advances in machine learning include employing deep neural networks to extract abstract features from input fish data using colour and texture information. The main motivation behind using these architectures is to learn highly nonlinear and complex data distributions representing fish in underwater video imagery. Moniruzzaman et al. (2017) presented a fish detection accuracy of 65.2% on a dataset of 93 images with fish instances using a deep convolutional neural network. Similarly, various deep architectures are used in (Moniruzzaman et al., 2017; Salman et al., 2016; Siddiqui et al., 2017) producing an average accuracy of 98.43%, 94.3% and 89.95% respectively on fish species classification task using several dedicated fish datasets consisting of 20,000 to 30,000 still images of multiple fish species. The main problem faced in using such complex systems is computational complexity, which bars them in real-time deployment, and the necessary requirement of very large annotated datasets to be used in training. Our main contributions reported in this paper are:

- A data-agnostic approach to fish detection in natural underwater environments, which obtains state-of-the-art results for the task, while maintaining real-time performance.
- *Re*-annotation of the Fish4Knowledge Complex Background dataset, described later in Section 2.1, which previously omitted a significant amount of fish.
- Evaluation and comparison over a larger dataset than the previous state-of-the-art method (Spampinato et al., 2014).

The rest of this paper is organized as follows: We discuss our proposed method and the dataset used for evaluation in Section 2. Section 3 presents the results for our method, followed by a discussion on the outcomes and validity of our method in Section 4. Finally, we conclude our paper and list potential future work directions in Section 5.

Table 1
Fish4Knowledge Complex Background Dataset statistics.

	Category	Number of videos	Number of annotated frames
	Blurred	3	122
	Camouflage foreground	2	108
	Complex background	3	147
	Crowded	3	116
	Dynamic background	2	96
	Hybrid	2	90
	Luminosity variations	2	202
Total	7	17	881

2. Material and methods

2.1. Dataset

Fish4Knowledge (Boom et al., 2014) is a large dataset consisting of underwater videos captured in Taiwan's coral reefs. Our work focuses on a small subset of the Fish4Knowledge dataset, containing 17 videos broadly categorized in seven different categories. Hereafter in this paper, we refer to this subset as the Fish4Knowledge Complex Background (FCB) dataset. Each of the the seven dataset categories present a unique challenge for detection algorithms including *Blurred* imagery, *Camouflage foreground* scenes where it is hard to distinguish fish from the background, *Crowded* video data involving dense gathering of fish in larger numbers in each image, *Complex background* videos showing rich background with colourful objects and coral reefs on seabed, *Dynamic background* videos depicting moving aquatic plants and seaweed, *Hybrid* data is a combination of complex and dynamic background and finally *Luminosity variation*, which contains videos with changing light beams due to surface water disturbances. Table 1 provides an overview of the dataset categories, and Fig. 1 shows sample frames and the annotation used for the dataset.

The videos are a mix of 320 × 240 and 640 × 480 resolution, with frame rates at 5 FPS and 24 FPS respectively. In total, there are 881 annotated frames, which make the task challenging especially for the recent state-of-the-art deep learning (LeCun et al., 2015; Schmidhuber, 2015) methods, which require a large amount of annotated data for effective training. Hence, we resort to traditional computer vision approaches to tackle the problem without compromising the accuracy of fish detection results. Additionally, the low resolution makes the detection task more difficult as the video imagery induces more pixel distortion and noise in the already complex settings.

Fig. 1 shows some sample images from each category, where the last figure represents the segmentation mask for the annotation present in the dataset.

One of the primary concerns over this dataset was the inconsistent annotation scheme previously used for annotating the fish. A lot of frames were annotated such that even prominently visible fish were not labeled. Similarly, a fish would be annotated in a particular frame but not annotated in the previous or the next frame despite being undergoing only small changes in its position. This discrepancy is illustrated in Fig. 2. To remove this inconsistency, we manually re-annotated all the labeled frames to include all the visible fish using Adobe®Photoshop©. Fig. 2 depicts this re-annotation.

2.2. Methodology

Adaptive background subtraction is a popular method to cater for scenes with dynamic background (Stauffer and Grimson, 1999). Such methods work by creating a background model and then constantly updating it over time. While this works well for dynamic backgrounds, adaptive background subtraction has a drawback that it tends to fail if the object of interest is moving slowly or becomes stationary for some

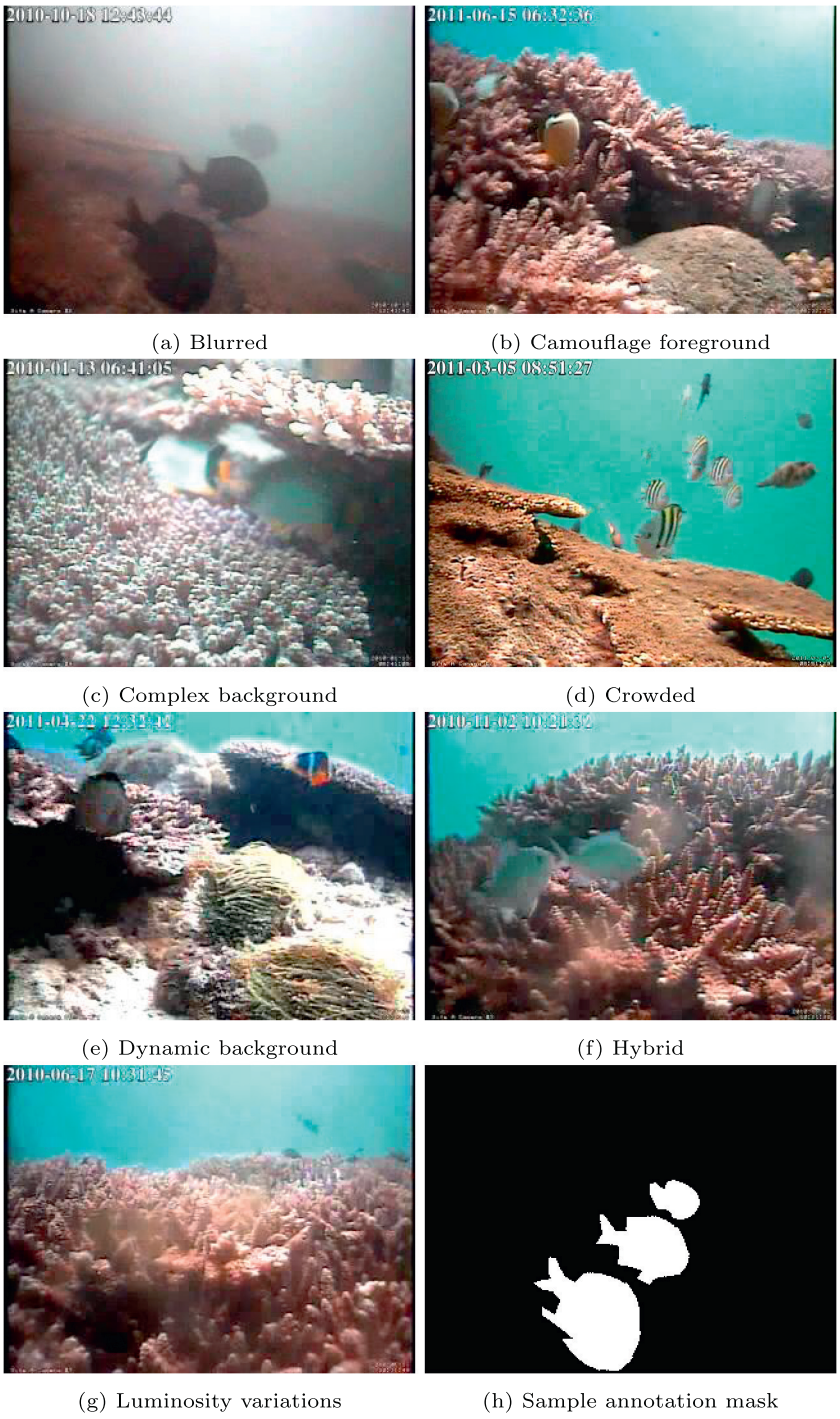

Fig. 1. The Fish4Knowledge Complex Background dataset. Figures (a)-(g) show sample frames from each of the seven dataset categories, while (h) shows an annotated frame to indicate the type of annotation used for the dataset.

time. Therefore, in such cases, the method segments out object of interest partially or, in some cases fails to detect the object at all. In most of the processing pipelines, partially segmented objects are discarded as a post-processing step due to a low pixel count. This increases the false negative count and results in a poor performance overall. Instead, we take care of these partial detections using pixel-wise posteriors (Bibby and Reid, 2008) to improve their segmentation mask.

2.2.1. Gaussian mixture models

We utilize adaptive background subtraction, specifically Grimmson GMM (Stauffer and Grimson, 1999) in our approach, due to its promising accuracy on dynamic background scenes and its real time performance. A GMM is a probabilistic model of the data distribution, represented by multiple individual Gaussian distributions, each characterized by its mean and covariance. These means and variances are learned in an unsupervised manner over an individual pixel distribution. In other words, each pixel value represents a feature of the input data, where feature vectors are obtained by the combination of these pixel values across subsequent video frames; thus, giving us feature vectors equal in number to the number of pixels in a video frame. The GMM then learns a model of the background given these features, where the background is defined as every non-fish entity in the video.

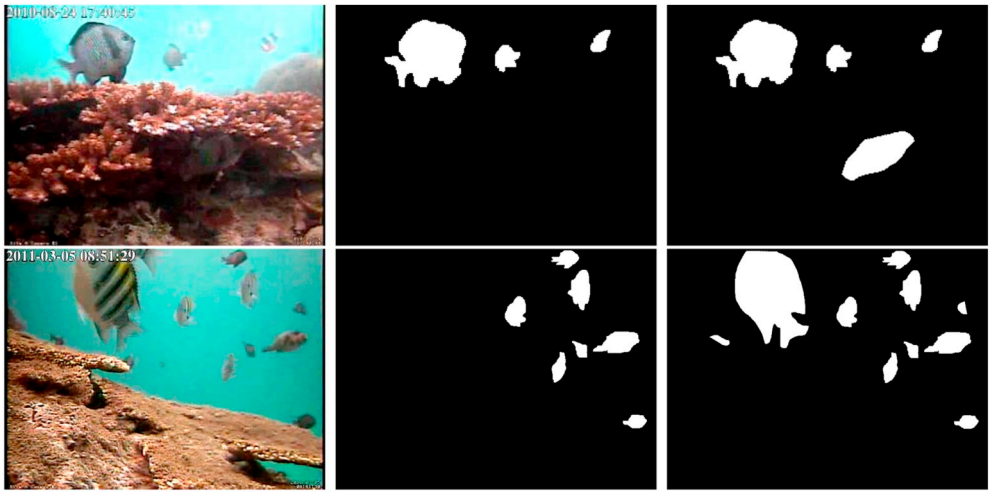

Fig. 2. The inconsistencies in the Fish4Knowledge Complex Background dataset. The left column depicts frame number 249 from the *Camouflage foreground* video 1 and frame number 700 from the *Crowded* video 3. The middle column shows the original ground-truth annotation available for the two images. The right column shows the same frames re-annotated with new fish instances clearly visible to human eye in the original coloured images. As noticed, the masks for the bottom-right and top-right frames are missing in the original annotations in one frame while being present in the other, despite no significant changes in between the frames for the two fish. Our re-annotation corrects this discrepancy.

The foreground corresponds to the fish, which are to be detected. The GMM model for the background is given as:

$$S_M = \left\{ w_j, \mu_j, \sum_j \right\} \quad for\ j = 1\ to\ M_g \tag{1}$$

where μj and Σj represent the mean and covariance matrices of the *jth* feature vector corresponding to the pixel j. w_j are the learned weight vectors, which indicate the contribution of the individual mono-Gaussians. M_g represents the number of individual Gaussians, where $M_g = 3$ in our case. The probability that a pixel x belongs to the background model SM in the t^{th} frame is given by:

$$p(x_t) = \sum_{j=1}^{M_g} w_j \eta \left(x_t \mid \mu_j, \sum_j \right) \tag{2}$$

where

$$\eta \left(x \mid \mu_j, \sum_j \right) = \frac{1}{(2\pi)^{D/2} |\sum_j| 1/2} \; exp \left(- \frac{(x - \mu_j)^T \sum_j -1 (x - \mu_j)}{2} \right) \tag{3}$$

here, D is the dimensionality of the feature vector. Eq. (3) is the standard mono-Gaussian distribution of a pixel x given the mean and variance. A higher probability in Eq. (2) indicates that a particular pixel corresponds to the background. In contrast, a low probability value indicates that the pixel significantly deviates from the background model i.e. there was a significant change in the pixel value, indicating motion. We used the bgslibrary[1] (Sobral, 2013) for the GMM implementation. The value for the number of mono-Gaussians M_g was varied between 2 and 5, while the learning rate, which must be tuned to avoid local minima of loss function in training the GMM, was chosen in the range of 0.001 to 0.01. In our case, $M_g = 3$ and learning rate of 0.005 were selected to yield the best performance.

The background subtraction outputs segmentation masks, which correspond to moving objects; in our case, fish. We perform an opening (erosion followed by dilation) operation with a 3×3 kernel on the obtained segmentation masks to filter noise and remove small non-fish moving objects. After the opening operation, we perform Connected Component Analysis (Samet and Tamminen, 1988) to obtain individual blobs representative of fish detections. We ignore the blobs with very small size in terms of pixel count as a second filter for noise and non-fish objects. For each blob, we determine its bounding box and apply a 20% padding to each side to get an inflated blob. The padding is applied to ensure that there is enough background information for the next steps in our algorithm. Concurrently, the original image is compressed to 5-bit colour values instead of 8-bit, which significantly decreases histogram size from 256^3 to 32^3. This has a positive effect on the algorithm run time. We then obtain the pixels corresponding to the inflated blob in the compressed version of our subject frame.

[1] https://github.com/andrewssobral/bgslibrary.

2.2.2. Pixel-wise posteriors

To address the limitations of adaptive background subtraction, we use Pixel-wise Posteriors (Bibby and Reid, 2008) to improve the segmentation results from the background subtraction. This is especially useful for slow moving objects, which are detected partially by background subtraction algorithms. These partial detections can not be ignored since object detection is often only the first step in such an analysis pipeline. Inaccurate detection often leads to an inaccurate analysis. In such cases, the slow moving objects can be of even more importance such as those in the case of endangered species showing signs of illness. The following notations are used in this algorithm:

- x: Pixel's spatial position in the object coordinate frame.
- y: Pixel's intensity (In our experiments, it corresponds to RGB value).
- $\mathbf{W}(x, \mathbf{p})$: Warp with parameters $\mathbf{p}$.
- $M = \{Mf, Mb\}$: Model's parameter either foreground or background.
- $P(y|M_f)$: Foreground model over pixel values y.
- $P(y|M_b)$: Background model over pixel values y.
- $\mathbf{C}$: The contour that segments the foreground from background.
- $\mathbf{\Phi}(x)$: Shape kernel.
- $\mathbf{\Omega} = \{\mathbf{\Omega_f}, \mathbf{\Omega_b}\}$: Pixels in the object frame $[\{x_0, y_0\}, \ldots, \{x_N, y_N\}]$, which is partitioned into foreground pixels $\mathbf{\Omega_f}$ and background pixels $\mathbf{\Omega_b}$.

Fig. 4 shows a generative model, which is used to represent the image creation process of the posterior analysis. The model considers image as a bag-of-pixels and can, given the model M, the shape $\mathbf{\Phi}$ and the location $\mathbf{p}$, be used to sample pixels $\{x, y\}$. Although the final image would not look like the actual foreground/background image to the naked eye due to jumbling of pixels but the colour distributions corresponding to the foreground/background regions $\mathbf{\Omega_f}/\mathbf{\Omega_b}$ would match the models $P(y|M_f)$ and $P(y|M_b)$ respectively. Due to this simplicity of this model that gives more invariability to the viewpoint and allows 3D objects to be tracked robustly without having to model their specific 3D structure. The joint distribution for a single pixel given by the model in Fig. 4 is:

$$P(x, y, \mathbf{\Phi}, \mathbf{p}, M) = P(x \mid \mathbf{\Phi}, \mathbf{p}, M) P(y \mid M) P(M) P(\mathbf{\Phi}) P(\mathbf{p}) \tag{4}$$

Now, we divide eq. (4) by $P(y) = \Sigma_M P(y|M)P(M)$ to give:

$$P(x, \mathbf{\Phi}, \mathbf{p}, M \mid y) = P(x \mid \mathbf{\Phi}, \mathbf{p}, M) P(M \mid y) P(\mathbf{\Phi}) P(\mathbf{p}) \tag{5}$$

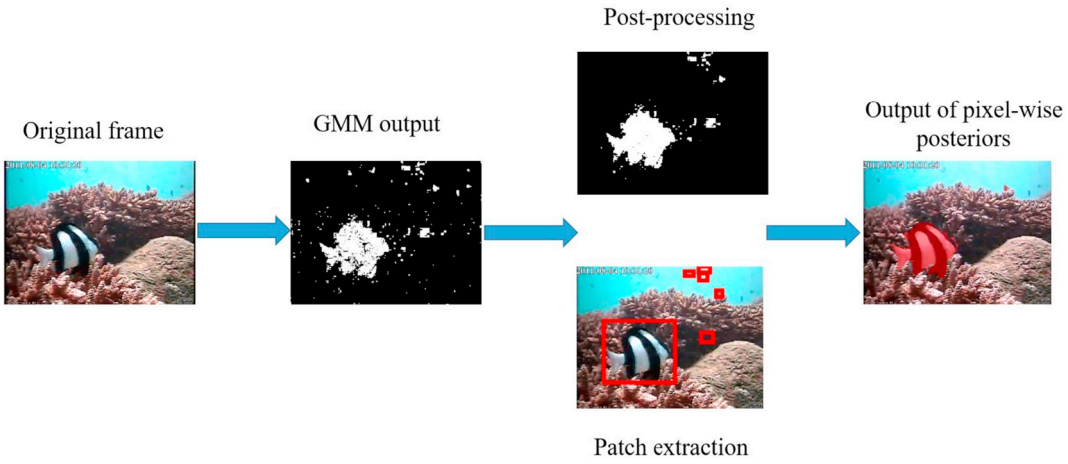

Fig. 3. An overview of our proposed method.

where the term $P(M|\mathbf{y})$ is the pixel-wise posterior, of the models M, given a pixel value y. The Pixel-wise Posteriors model is given by:

$$P(M_j \mid y) = \frac{P(y \mid M_j)P(M_j)}{\sum_{i=f,b} P(y \mid M_i)P(M_i)} \qquad j = f, b \tag{6}$$

Where j represents the foreground or background model index, $P(M_j)$ is probability of a pixel belonging to model j, y is colour value of pixel, $P(y \mid M_j)$ is probability of a particular colour belonging to model j and $P(M_j \mid y)$ is posterior probability of a pixel being part of a model given its colour value.

We apply Pixel-wise posteriors on the inflated blobs as seed and the corresponding extracted pixels from the compressed original image. The method uses both, the model and the colour value of pixels to calculate $P(y \mid M_j)$ by dividing frequency of pixel value y by total number of pixels in M_j. $P(M_j)$ is calculated by dividing pixel count in M_j by the total number of pixels in the image. $\Sigma_{i=f,\ b}P(y \mid M_i)P(M_i)$ can be replaced by $P(y)$.

The results of Pixel-wise Posteriors are concatenated into a single image after another size filtering step to ignore very small blobs. Our algorithm is computationally efficient and gives real-time performance on CPUs, as opposed to the current state-of-the-art (Spampinato et al., 2014), which require the support of graphical processing units (GPU) in conjunction with the CPU. Fig. 3 gives an overview of our algorithm.

3. Results

We use the popular F-measure to evaluate our detection results. The F-measure depends on the Precision and Recall, which themselves are measures of classification performance and are defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{7}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{8}$$

A high precision indicates a low number of false positives while a high recall is synonymous with a small number of false negatives. Usually, these two are competing objectives and maximizing one leads to the other lowering in value. The F-measure thus provides a balanced measure between the two objectives and is obtained as the harmonic mean between the two. It is defined as:

$$\text{F} - \text{measure} = \frac{\text{Precision}^* \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

In addition to our proposed method, we also propose an enhancement to the Grimson GMM (Stauffer and Grimson, 1999) algorithm where we achieve significantly improved results over the GMM implementation used by (Spampinato et al., 2014). GMM-based background subtraction followed by 3×3 opening (erosion followed by dilation) kernel improves the results from 69.92% (Spampinato et al., 2014) to 83.26% in our case. Further, using Pixel-wise Posteriors in addition to the above enhancement, we achieve an 84.28% F-measure, which according to the best of our knowledge, sets the current state-of-the-art on the employed dataset. Additionally, our algorithm is computationally efficient and supports real time performance even on CPU-based systems. The results for both the Enhanced GMM method as well as our primarily proposed approach using Pixel-wise Posteriors[2] are summarized in Table 2. Fig. 5 also shows some of the detection results for our algorithm.

4. Discussion

According to International Union for Conservation of Nature (IUCN), 1414 species of fish are at the risk of extinction. Similarly, 36% of 15,000 known fresh water species are threatened due to habitat loss, industrial pollution, deforestation, climate change and commercial over fishing. Therefore, the need of quick and extended underwater sampling of fish is inevitable to monitor their population size. The availability of advanced computational resources has created opportunities for rapid yet automatic sampling of fish fauna using underwater videos. As opposed to laborious and costly methods of manual sampling, automatic fish detection using efficient machine learning and computer vision techniques is gaining attention of the marine scientists and conservationists due to their ability to generate fast detection results.

The main contribution of this paper is performance improvement over existing solutions for automatic fish detection in unconstrained underwater videos. This is a vital step to estimate fish abundance, biomass and assemblage in any water body. To achieve this aim, we have proposed a novel recipe to combine GMM-based background subtraction with pixel-wise posteriors, a refining step to compensate the shortcomings of GMM. GMM is a machine learning algorithm, which in our case, acts as the foreground segmentation process by learning the first and second order statistics of the background pixels. Estimating this background distribution assumes the background to completely lack foreground objects i.e., fish in the training dataset. In unconstrained underwater videos, it is extremely difficult to extract pure background frames as fish instances may still appear in the scenes of training data. This results in a compromised performance due to either false alarms or miss detections as shown in Fig. 5, third column. To some extant, this problem can be rectified by applying morphological

[2] https://github.com/ahsan856jalal/Fish-Segmentation-using-pixel-wise-Posteriors.git.

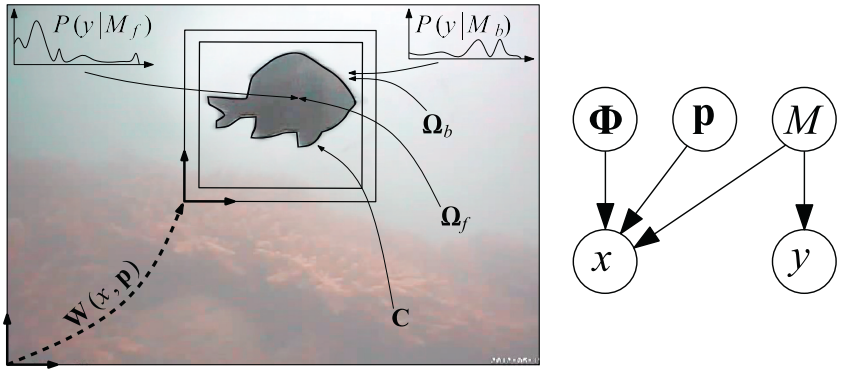

Fig. 4. (Left): Image containing fish as object showing: the contour **C**, the set of foreground pixels Ω_f, the set of background pixels Ω_b, the foreground model $P(y|M_f)$, the background model $P(y|M_b)$ and the warp $\mathbf{W}(x, \mathbf{p})$; (Right): The graphical view of the generative model showing the image as a bag-of-pixels, which gives greater invariability to viewpoint compared with template based tracking of objects.

operations to subside noise and redundancy in the output. Furthermore, due to imperfect training data, GMM sometimes fails to detect stationary or partially moving fish if that fish instance also occurs in the training dataset as it gets confused with the background and results in a fragmented output foreground blobs of the fish. This problem is cured by employing probabilistic modelling of both background and foreground together with realistic object bounding warping in pixel-wise posterior approach. We have modified the output of GMM by adding an opening operation (erosion followed by dilation) to improve its foreground detection results from 69.92% to 83.26%. After that, we performed connected component analysis to get inflated blobs for the foreground objects, which are passed to pixel-wise posteriors module along-with the corresponding extracted pixels from the compressed original image. The improved segmentation of the foreground objects from pixel-wise posteriors are concatenated to a single image after filtering blobs below a certain size.

Table 2 summarizes a comparison between our proposed approach and other techniques, which have been used frequently for object detection tasks in videos. It is evident that our proposed algorithm outperforms all others on FCB dataset on average F-scores. It is worth mentioning here again that we not only established an enhancement to the existing GMM (Stauffer and Grimson, 1999) through morphological image post-processing, but also refined it with pixel-wise posterior analysis. Our proposed approach achieved the best performance in the categories of *Camouflage foreground objects*, *Crowded*, *Dynamic background* and *Luminosity variation*. On the other hand, our enhanced version of GMM outperformed in *Blurred* scenario, although it is ahead of the pixel-wise posterior refinement with a very narrow margin of 0.42%. The main reason behind this anomaly is the appearance of *Blurred* video, where it is extremely difficulty to estimate pixel-wise posteriors due to uniformity in pixel intensity of the entire frame. This uniformity is caused by the severe water murkiness in most of the videos of this category, where foreground fish contour cannot be distinguished from the background pixels and only a slight variation can be observed due to fish movement, which is captured by Enhanced GMM. The pixel-wise posterior refinement of top of Enhanced GMM is causing a very little degradation due to the wrong estimates of fish contours (with the parameter C) that segments the foreground with the background. KDE-RGB (Sheikh and Shah, 2005) algorithm yields the best scores for *Complex background* and *Hybrid* category. This observation is consistent with their original work and can be justified in two ways. First, *Complex background* and *Hybrid* scenes depict mostly complex structure of background seabed representing one structural kind of coral reef. These videos naturally show distinction in colour profiles of foreground fish and background, which works best for KDE-RGB algorithm as it exploits correlation in RGB pixel intensity with neighbouring pixels having close spatial proximity. In other words, colour and texture of foreground and background show good intra-correlation but poor inter-correlation. Thereby, fish instances can be easily detected. Second, there is uniformity in the background pixel distribution as large portions of the frames in these videos are similar in pattern and can be modeled using a single distribution, a method adopted in KDE-RGB technique. These attributes cannot be observed in other video classes hence, KDE-RGB lags as compared to other approaches especially in the *Camouflage foreground objects* and *Dynamic background* category where background is rich and vibrant and therefore, cannot be modeled with a single data distribution. In addition, the texture and colours of the foreground fish and background cannot be distinguished easily (see Fig. 1).

Except Enhanced GMM and our proposed approach, Texton-KDE (Spampinato et al., 2014) shows good performance on the average. In fact, the previous best scores on FCB dataset are reported by this approach. This technique is an improvement to the KDE-RGB, where in addition to the background, it also models the foreground fish using their texture-dependent features in the low-contrast region of the image. This improves the performance in the scenes where it becomes extremely challenging to differentiate fish with the background e.g., in the case of camouflage, fish and background pixels show confusion in texture and similarity in colour. In their original work, the authors limited the FCB dataset to 280 images for the evaluation of their algorithm. We repeat the experiment using their algorithm with our rectified and re-annotated dataset on the same images as are used in their experiment but do not realize a significant improvement. On the other hand, our evaluation dataset comprises of 881 labeled frames and thus demonstrates an increased robustness and accuracy. Our results in Table 2 are consistent with this observation.

We tried morphological enhancement and pixel-wise posterior refinement with other approaches mentioned in Table 2 including KDE-RGB, ZGMM and Texton-KDE with significant improvement but the scores could not exceed our proposed approach. However, the computation time was observed to be at least twice that of ours. The reason behind this outcome is the choice of our GMM implementation (Stauffer and Grimson, 1999), which is simple, effective and produces equally good scores when enhanced with post-processing and pixel-wise posteriors.

The computation resources utilized in our work include Intel®Core™-i7 4 GHz CPU with 32 GB RAM and a mechanical hard disk drive operating at 7200 rpm. With these computer specifications, Enhanced GMM takes 150 ms to process one frame and pixel-wise posterior operation takes 200 μs per frame, which is an insignificant computation overhead to the Enhanced GMM. Overall, our approach achieves 6 fps (frames per second) processing speed given that some videos in the FCB dataset are recorded even at 5 fps, which is an acceptable frame rate to observe and analyze underwater videos (Boom et al., 2014).

5. Conclusion

We have proposed an algorithm to improve segmentation of fish in an unconstrained underwater environment using pixel-wise posteriors

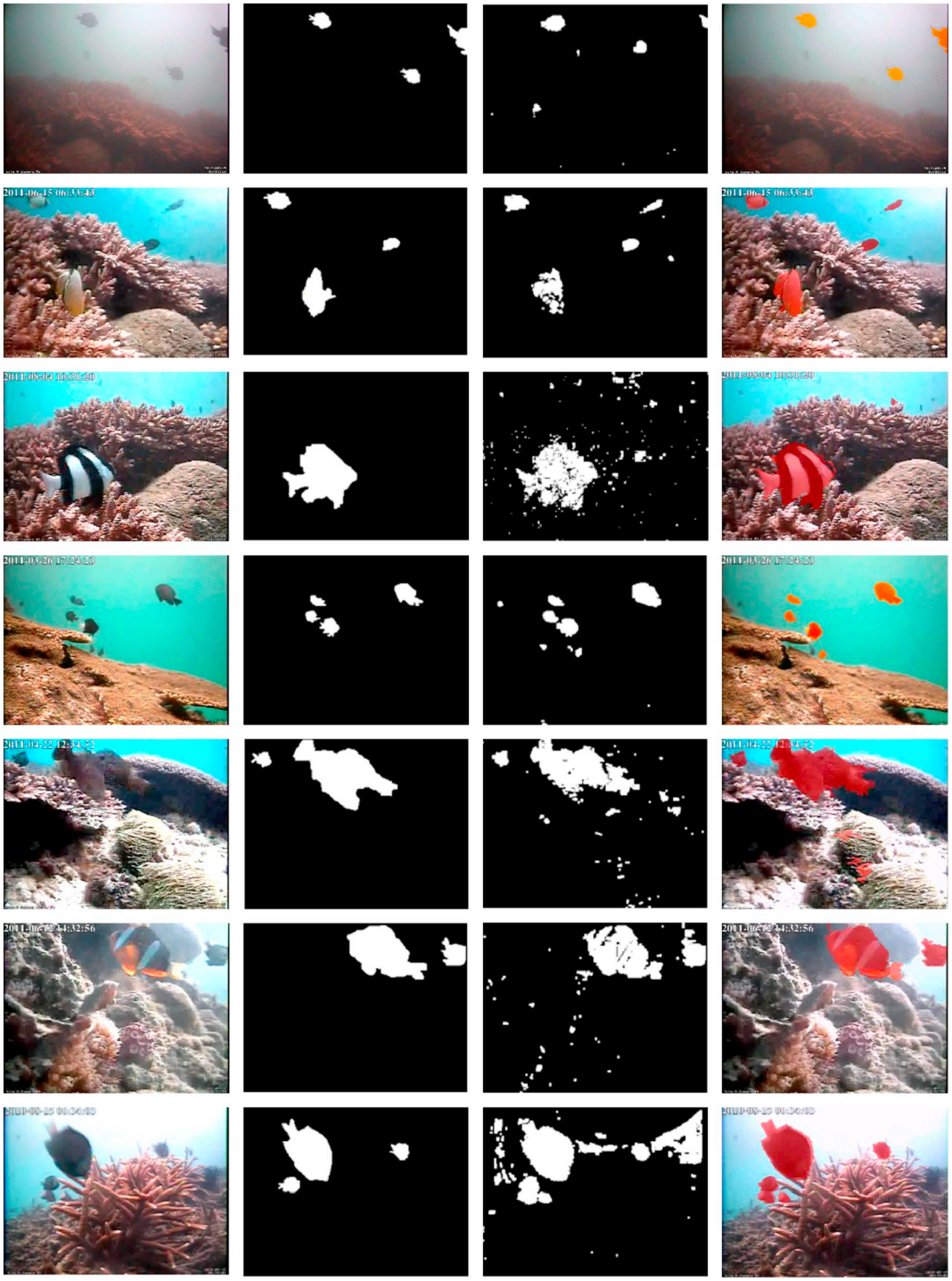

Fig. 5. Detection results on the selected frames from each category of the dataset. Original frames, ground truths, GMM outputs, and the outputs from pixel-wise posterior analysis are shown in the first, second, third and fourth columns respectively.

on adaptive background subtraction from Gaussian mixture model. It is observed that adaptive background subtraction techniques suffer when the object is slowly or partially moving or when video imagery suffers from foreground confusion with the background, which results in incomplete detections. Partially detected fish instances are often discarded before further fish classification tasks due to their low pixel count. This missed detection lowers the overall performance of the system as false negatives increase and the accuracy of the system decreases. The experimental results have shown that the proposed approach has the best average F-score measure in detecting fish under complex background and varying environments, which will lead to better fish classification. Today, deep learning is emerging as the most promising approach in extracting task-specific information from the data utilizing highly nonlinear mathematical models especially suitable for computer vision tasks (LeCun et al., 2015). However, their computational complexity is the biggest obstacle in their deployment to the real-word scenarios. Therefore, in future, we aim to employ deep neural networks in conjunction with the pixel-wise posteriors with highly optimized algorithm that should be capable of yielding results in real-time and with favourable accuracy.

Table 2
Category-wise F-score results on the FCB dataset. Highest scores are enlisted in bold.

Method / Video	KDE-RGB (Sheikh and Shah, 2005)	ZGMM (Zivkovic, 2004)	ML-BKG (Yao and Odobez, 2007)	Texton-KDE (Spampinato et al., 2014)	Enhanced GMM	Proposed approach
Blurred	92.11	76.91	71.21	93.10	**96.42**	**96.00**
Camouflage foreground objects	53.58	71.13	74.89	82.88	83.44	**84.85**
Complex background	**87.06**	**76.56**	**82.11**	**82.06**	**71.80**	**75.17**
Crowded	82.92	74.81	80.27	84.67	83.99	**84.83**
Dynamic background	59.79	64.87	78.32	76.31	77.59	**78.32**
Hybrid	**84.87**	**76.11**	**72.79**	**83.45**	**81.60**	**82.11**
Luminosity variations	72.43	59.63	83.13	70.10	87.99	**88.71**
Average	**76.10**	**71.43**	**77.53**	**81.79**	**83.26**	**84.28**

References

Bibby, C., Reid, I., 2008. Robust real-time visual tracking using pixel-wise posteriors. Comput. Vision–ECCV (2008), 831–844.

Boom, B.J., He, J., Palazzo, S., Huang, P.X., Beyan, C., Chou, H.-M., Lin, F.-P., Spampinato, C., Fisher, R.B., 2014. A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. Ecol. Inform. 23, 83–97.

Harvey, E., Shortis, M., 1995. A system for stereo-video measurement of sub-tidal organisms. Mar. Technol. Soc. J. 29 (4), 10–22.

Hsiao, Y.-H., Chen, C.-C., Lin, S.-I., Lin, F.-P., 2014. Real-world underwater fish recognition and identification, using sparse representation. Ecol. Inform. 23, 13–21.

Huang, P.X., Boom, B.J., Fisher, R.B., 2015. Hierarchical classification with reject option for live fish recognition. Mach. Vis. Appl. 26 (1), 89–102.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

McLaren, B.W., Langlois, T.J., Harvey, E.S., Shortland-Jones, H., Stevens, R., 2015. A small no-take marine sanctuary provides consistent protection for small-bodied by-catch species, but not for large-bodied, high-risk species. J. Exp. Mar. Biol. Ecol. 471, 153–163.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.-R., 1999. Fisher discriminant analysis with kernels. In: Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop. IEEE, pp. 41–48.

Moniruzzaman, M., Islam, S.M.S., Bennamoun, M., Lavery, P., 2017. Deep learning on underwater marine object detection: A survey. In: International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, pp. 150–160.

Palazzo, S., Murabito, F., 2014. Fish species identification in real-life underwater images. In: Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data. ACM, pp. 13–18.

Rova, A., Mori, G., Dill, L.M., 2007. One fish, two fish, butterfish, trumpeter: recognizing fish in underwater video. In: MVA, pp. 404–407.

Salman, A., Jalal, A., Shafait, F., Mian, A., Shortis, M., Seager, J., Harvey, E., 2016. Fish species classification in unconstrained underwater environments based on deep learning. Limnol. Oceanogr. Methods 14 (9), 570–585.

Samet, H., Tamminen, M., 1988. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. IEEE Trans. Pattern Anal. Mach. Intell. 10 (4), 579–586.

Schmidhuber, J., 2015. Deep learning in neural networks: an overview. Neural Netw. 61, 85–117.

Sheikh, Y., Shah, M., 2005. Bayesian modeling of dynamic scenes for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 27 (11), 1778–1792.

Shortis, M., Harvey, E., Abdo, D., 2009. A review of underwater stereo-image measurement for marine biology and ecology applications. In: Oceanography and Marine Biology. CRC Press, pp. 257–292.

Siddiqui, S.A., Salman, A., Malik, M.I., Shafait, F., Mian, A., Shortis, M.R., Harvey, E.S., 2017. H. editor: Howard Browman, Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data. ICES J. Mar. Sci. 75 (1), 374–389.

Sobral, A., 2013. BGSLibrary: An opencv c++ background subtraction library. In: IX Workshop de Visão Computacional (WVC'2013), Rio de Janeiro, Brazil. URL. https://github.com/andrewssobral/bgslibrary.

Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y.-H.J., Fisher, R.B., Nadarajan, G., 2010. 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. ACM, pp. 45–50.

Spampinato, C., Palazzo, S., Kavasidis, I., 2014. A texton-based kernel density estimation approach for background modeling under extreme conditions. Comput. Vis. Image Underst. 122, 74–83.

Stauffer, C., Grimson, W.E.L., 1999. Adaptive background mixture models for real-time tracking. In: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. Vol. 2. IEEE, pp. 246–252.

Storbeck, F., Daan, B., 2001. Fish species recognition using computer vision and a neural network. Fish. Res. 51 (1), 11–15.

Strachan, N., 1993. Recognition of fish species by colour and shape. Image Vis. Comput. 11 (1), 2–10.

Strachan, N., Kell, L., 1995. A potential method for the differentiation between haddock fish stocks by computer vision using canonical discriminant analysis. ICES J. Mar. Sci. 52 (1), 145–149.

Turk, M., Pentland, A., 1991. Eigenfaces for recognition. J. Cogn. Neurosci. 3 (1), 71–86.

Yao, J., Odobez, J.-M., 2007. Multi-layer background subtraction based on color and texture. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, pp. 1–8.

Zivkovic, Z., 2004. Improved adaptive Gaussian mixture model for background subtraction. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, Vol. 2, IEEE, pp. 28–31.