Applied Intelligence (2021) 51:5809–5821
https://doi.org/10.1007/s10489-020-02155-8

Multi-stream fish detection in unconstrained underwater videos by the fusion of two convolutional neural network detectors

Abdelouahid Ben Tamou[1,2] · **Abdesslam Benzinou**[1] · **Kamal Nasreddine**[1]

Accepted: 15 December 2020 / Published online: 16 January 2021

Abstract

Recently, marine biologists have begun using underwater videos to study species diversity and fish abundance. These techniques generate a large amount of visual data. Automatic analysis using image processing is therefore necessary, since manual processing is time-consuming and labor-intensive. However, there are numerous challenges to implementing the automatic processing of underwater images: for example, high luminosity variation, limited visibility, complex background, free movement of fish, and high diversity of fish species. In this paper, we propose two new fusion approaches that exploit two convolutional neural network (CNN) streams to merge both appearance and motion information for automatic fish detection. These approaches consist of two Faster R-CNN models that share either the same region proposal network or the same classifier. We significantly improve the fish detection performances on the LifeClef 2015 Fish benchmark dataset not only compared with the classic Faster R-CNN but also with all the state-of-the-art approaches. The best F-score and mAP measures are **83.16%** and **73.69%**, respectively.

Keywords Fish detection · Underwater videos · Deep learning · Convolutional neural networks · Multi-stream

1 Introduction

Approximately 71% of the earth's surface is covered by oceans, but only 5% of oceans have been explored.[1] There are currently 230,000 known marine species, including around 20,000 fish species, although the number of species found in oceans is much greater. Fish is one of the important resources for humans, especially as food; they are fished or farmed in ponds or cages in the ocean (aquaculture) by commercial and subsistence fishers.

In recent years, underwater video cameras have been extensively used to explore oceans and study biodiversity. These non-destructive systems do not perturb the environment and generate a large amount of visual data that are usable at any time. However, the manual analysis of these videos is labor-intensive, time-consuming, and costly. Consequently, automatic processing is required. Nevertheless, visual underwater data present several numerous challenges to computer vision. On the one hand, the videos are usually of low quality, the luminosity may change suddenly, the visibility is limited due to turbidity, and the complex coral backgrounds sometimes change rapidly due to moving aquatic plants. On the other hand, in underwater fish videos, the fish move in three dimensions or can hide behind algae and rocks. Fish detection is also hampered by fish overlapping or camouflage with the background.

In general, automatic fish classification involves two phases: 1) fish detection to detect and discriminate fish from the background; and 2) fish species recognition to identify the species of each detected fish. In this paper, we focus on the issue of automatic fish detection in an unconstrained underwater environment. Automatic fish detection is necessary in order to recognize fish species, track them, and study their behavior.

In the literature, different approaches have been proposed for automatic fish detection in the open sea. Early studies implement background modeling such as the adaptive

[1] https://www.noaa.gov/oceans-coasts

✉ Abdesslam Benzinou
benzinou@enib.fr

Abdelouahid Ben Tamou
bentamou@enib.fr

[1] ENIB, UMR CNRS 6285 LabSTICC, Brest, 29238, France

[2] LRIT-CNRST URAC 29, Faculty of Sciences, Mohammed V University in Rabat, Rabat, Morocco

Gaussian mixture model (GMM) [1]. Spampinato et al. [2] modeled the background by combining two algorithms: moving average detection and GMM. Hsiao et al. [3] proposed motion-based fish detection in videos, while modeling the background using GMM. Fish motion is detected when a certain region of the frame does not fit into the trained background model. These approaches are based on hand-crafted methods and are regarded as shallow learning [4], which means that they are unable to model underwater environments under different scenarios: complex backgrounds with complex textures, background movements, transient and abrupt luminosity changes, poor visibility, low contrast, fish camouflage with the background due to color and texture similarity, and samples of crowded fish. Examples of these scenarios are shown in Fig. 1.

Recently, several works have developed algorithms based on convolutional neural networks (CNNs) for various visual tasks on account of their remarkable performance records. Deep CNNs [5] are capable of modeling nonlinear data due to their powerful architecture, which has many learned layers. Recent works applied existing classical CNN-based detectors to automatic fish detection. Li et al. [6] applied Fast R-CNN to underwater fish images, achieving a mean average precision (mAP) of 81.4% on the LifeClef 2014 dataset[2] of 12 fish species. In another study [7], they also accelerated fish detection using the Faster R-CNN detector with ZFNet as the base network and reached a mAP of 82.7% on the same dataset. They improved the mAP by 7.25% [8] using Faster R-CNN with PVANet [9] as the base network. Mandal et al. [10] used Faster R-CNN to detect 50 fish and crustacean species in multiple beaches and estuaries in Australia and achieved a mAP of 82.4%. Zhuang et al. [11] proposed the use of a single shot detector (SSD). Shi et al. [12] presented FFDet, which uses SSD and combines features extracted from different layers. They achieved a mAP of 62.83% for 7,514 fish examples taken from the SeaClef dataset.[3] Sung et al. [13] used the YOLO (you only look once) detector and achieved an average accuracy of 65.2% for 93 underwater fish images. Other works introduced hybrid approaches using CNNs and hand-crafted methods. Jäger et al. [14] used background subtraction to generate bounding box proposals; CNN then extracted features from each proposal to feed a binary support vector machine (SVM) to be classified into fish or background classes. Zhang et al. [15] proposed an unsupervised underwater fish detection. First, they automatically generated and labeled region proposals using motion flow segmentation and selective searches. CNN was then used to classify the proposals as fish or background. Salman et al. [16] used a hybrid system to feed a Faster R-CNN detector by fusing GMM output with the optical flow and grayscale image. They achieved an F-score of 87.44% on the Fish4Knowledge Complex Scenes dataset and 80.02% on the LifeClef 2015 Fish (LCF-15) dataset.[4]

Several studies have proposed fusion strategies for different computer vision tasks. These fusions integrate information from multiple modalities (e.g., RGB, depth, infrared, audio) or multiple spaces (e.g. spatial, temporal). Fusion aims to merge important information extracted from individual modalities or spaces to better solve a given problem. Another advantage is that it also improves prediction robustness using complementary information to ensure that the system remains operational even if one information source is lost. We can divide the fusion approaches into three main categories: early, late, and hybrid fusions. Early fusion [17, 18] involves input- or feature-level fusion. In input-level fusion, multiple types of raw data are concatenated before feeding a one-stream CNN [17]. In feature-level fusion, multiple CNN streams are used to extract features from each information source [18]; the features are then merged to feed a classifier. Early fusion involves only one learning phase. Since the features are merged from the start, early fusion yields a rich feature representation, although its large dimension increases the overfitting risk. Late fusion [19] is applied at the decision level (e.g., classification, detection, regression) by fusing multiple decisions issued from multiple CNN streams. Decisions are fused using a variety of fusion mechanisms: for example, average, maximum, or sum of scores [20]; voting schemes [21, 22]; non-maximum suppression (NMS) [19]; and learned models such as SVM and extreme learning machine (ELM) [23]. Late fusion, however, has the disadvantage of requiring separate supervised learning for each stream. Finally, hybrid fusion [24, 25] is a combination of both early and late fusion schemes, aiming to exploit the advantages of both methods in a common framework. To the best of our knowledge, the existing CNN-based approaches used for live fish detection are all based on the one-stream network. Only the work of Salman et al. [16] proposed a fusion strategy based on a one-stream network via input-level fusion.

In another application, Guerry et al. [19] proposed three fusion architectures known as U-, X-, and Y-fusions to merge RGB and depth images (Fig. 2). U- and X-fusions are late fusions that use two Faster R-CNN models. They fuse detection decisions by NMS placed at the end or in the middle of the architecture. Late fusion ignores the correlation between different data. Y-fusion is an early fusion that uses merged features extracted from two CNNs to feed one region proposal network (RPN) followed by a classification network. Recently, Zhu et al. [18] proposed

[2] https://www.imageclef.org/2014/lifeclef/fish

[3] https://www.imageclef.org/lifeclef/2016/sea

[4] https://www.imageclef.org/lifeclef/2015/fish

Fig. 1 Examples of underwater images from different videos of the LifeClef 2015 Fish benchmark dataset to illustrate the high variation in an unconstrained environment with complex, crowded, or dynamic backgrounds and luminosity variations

a two-stream Faster R-CNN to fuse color and depth images. They used a single RPN with only the depth features to generate depth ROIs and then mapped them onto the RGB features to generate the corresponding RGB-ROIs. Finally, they merged these features to feed a single classifier. Farahnakian and Heikkonen [17] proposed pixel-level fusion as input-level fusion where they concatenated RGB and infrared images to feed a Faster R-CNN.

Inspired by these works, in this paper, we propose two new fusion architectures based on the merging of two CNN streams for two types of data. The first proposed architecture is a hybrid fusion that uses a single RPN shared between two Faster R-CNN models (Fig. 4a). This RPN uses the fusion of features extracted from different information to generate regions of interest (ROIs), thus allowing the RPN to have a richer space and thus better predict ROIs. Each stream uses its own classifier to classify the common ROIs, which results in fewer missed detections. The second proposed architecture is an early fusion that uses two RPNs to feed a shared classifier (Fig. 4b). This classifier is a small CNN that continues the convolution of the merged features to extract the correlation between them. Here, the two RPNs cooperate to generate more precise ROIs.

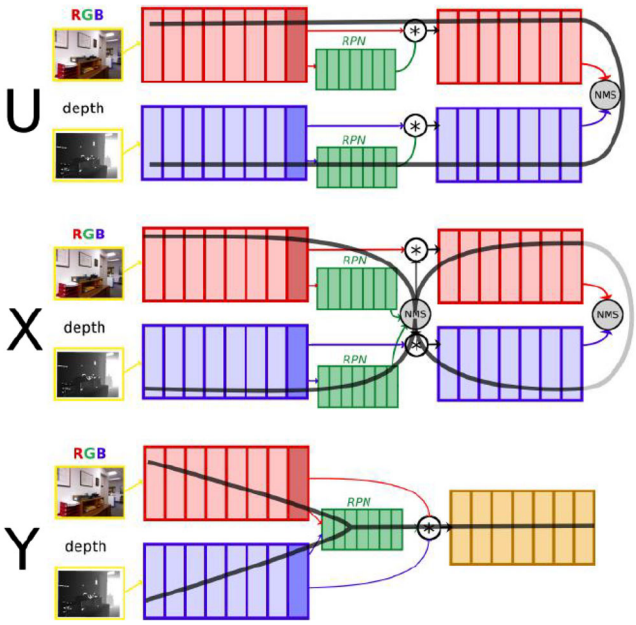

Fig. 2 Different CNN fusion schemes proposed by Guerry et al. [19]. From top to bottom: U-fusion, X-fusion, and Y-fusion

The main contributions of this work are as follows:

- We propose multi-stream fusion for the coral reef fish detection task in unconstrained underwater videos. The color image and motion map are used as the model inputs.
- We propose two new fusion architectures that merge two CNN streams based on two types of data. First, appearance and motion features are extracted separately from the color image and motion map. Then, extracted features are fused to feed a shared RPN (first architecture) or a shared classifier (second architecture).
- The new fusion architectures outperform state-of-the-art fusion architectures on the LCF-15 benchmark dataset, which contains underwater videos in various marine scenes with complex and dynamic backgrounds, crowded fish, and luminosity changes.

The remainder of the paper is organized as follows. Section 2 reviews the different CNN-based approaches for object detection and multi-stream fusion schemes. Section 3 describes the proposed approaches for underwater live fish detection. Section 4 shows the experimental results and makes a comparative study using the LCF15 benchmark dataset. Finally, the conclusion and perspectives are presented in Section 5.

2 Related works

In this section, we review the different approaches proposed for object detection based on CNN. First, we present the most common one-stream CNN-based approaches and then describe the state-of-the-art in multi-stream CNN fusion approaches.

2.1 CNN-based object detection

Girshick et al. [26] were the first to propose CNN for object detection, namely the region-CNN (R-CNN) method using selective searches [27] to generate region proposals from the image. Features are then extracted with CNN for each region to feed the SVM and classify it into foreground/background classes. R-CNN was subsequently improved to Fast R-CNN [28] where CNN extracts feature maps from the whole image. Region proposals are generated by selective searches from these feature maps. For each region, a fixed-length feature vector is extracted using the ROI pooling layer. Each vector is fed into a sequence of fully connected layers to predict the class of the proposed region with a softmax layer. Finally, the localization of the bounding box is predicted and refined using a linear regression layer. Ren et al. [29] presented Faster R-CNN, which combines the RPN with the Fast R-CNN model. RPN is a separate convolution network that is used to simultaneously predict region proposals and scores at each position in the feature maps. Dai et al. [30] proposed the region-based fully convolutional network (R-FCN), which only has convolutional layers but without fully connected layers after ROI pooling.

The aforementioned object detection algorithms are based on a two-stage approach. The first stage generates region proposals and the second recognizes an object in each region. One-stage detectors then merge the two basic stages into a single model to simultaneously take into account the object detection and its localization. OverFeat [31] was the first one-stage detector based on CNN. Redmon et al. [32] proposed the YOLO model, which is a single CNN that directly predicts bounding boxes and class probabilities. It divides input image into grids, and then each grid cell predicts bounding boxes and their corresponding confidence scores. Similar to the YOLO model, Liu et al. [33] developed a single shot multi-box detector (SSD), which uses a set of default anchor boxes with different aspect ratios and scales instead of fixed grids to discretize the output space of the bounding boxes. To handle objects of various sizes, SSD takes feature maps from different convolutional layers to predict the bounding boxes. One-stage detectors are generally faster than two-stage detectors, but they are less efficient. Recently, Lin et al. [34] proposed focal loss to improve the accuracy performance of one-stage detectors by applying a modulating term to the cross entropy loss to focus learning on the hard negative samples. This results in state-of-the-art accuracy and speed.

2.2 Multi-stream CNN-based detection

Fusion strategy has been used in several applications, including facial expression recognition [35], audio-visual speech recognition [36], action recognition [37], medical image analysis [22, 38], and clustering [39]. In the object detection field, several works proposed multi-stream CNN fusion, especially for action and person detection. Peng et al. [40] proposed a two-stream Faster R-CNN model for action detection by separately training each stream for two types of data: RGB and optical flow maps. They then introduced a ROI fusion layer to merge the proposals from the two RPNs. At the classification level, they calculated the final detection scores as the average of the softmax scores from both streams and applied the bounding box regressor to the corresponding ROIs of each stream. Guerry et al. [19] proposed three frameworks to fuse color and depth data for people detection using the Faster R-CNN model as illustrated in Fig. 2. The first architecture is U-fusion, which merges the outputs of two detection streams by NMS at the end of training. The second architecture is X-fusion, which fuses the ROIs generated by the RPN of each stream via NMS. In this framework, all ROIs are shared and classified by both streams. Redundant detections are fused by NMS as in U-fusion architecture. The U- and X-fusions described above concern late fusion, and the two streams are structurally independent: both streams are trained separately and only merge at the end of training. Finally, Y-fusion is an early fusion that fuses the features extracted from the two streams to feed one RPN and only one classifier. Recently, Zhu et al. [18] proposed an end-to-end refined two-stream Faster R-CNN model to fuse color and depth to detect lactating sow. First, RGB and depth features are separately extracted using two CNNs. A single RPN then generates the ROIs using only the depth features. The coordinates of generated ROIs were mapped to the RGB image to generate the corresponding RGB ROIs. Finally, two ROIs (by depth and RGB) were merged to feed a single classifier.

3 Proposed approaches

In this paper, we propose two new multi-stream CNN fusion approaches for the efficient detection of moving objects. In these approaches, one CNN stream extracts the appearance features from each color video frame, while the other CNN stream extracts the motion features from successive frames. The input of this stream is composed of two adjacent raw grayscale frames and the corresponding optical flow. This stream aims to detect objects by learning about the relationship between adjacent frames. For image restoration purposes, Yu et al. [41] proposed a similar idea to take advantage of the relationship between infrared handprint images captured at different moments.

In Section 3.1, we describe the inputs of our proposed architectures for moving object detection. In Section 3.2, we present the one-stream Faster R-CNN that forms the basis of

our fusion architectures. Finally, our proposed CNN fusion architectures are provided in Section 3.3.

3.1 Architecture inputs

3.1.1 Color input

Using the right color space is important in automatic detection tasks, especially in underwater videos. The type of input color model can affect the detection performance. We chose the RGB color space, because our architectures were already pretrained using RGB images from the ImageNet dataset. Filter weights are more related to RGB images than other color spaces. Furthermore, in the underwater environment, the visible spectrum is modified by the sea depth. Radiations with higher frequencies are the least absorbed. Thus, red disappears in shallow water (5 m), green at about 50 m, and blue at about 60 m. As a result, deeper seas present blue-green scenes [42]; for this reason, the blue and green components provide much more discriminative information compared to other components of the different color models.

3.1.2 Motion input

We use optical flow to calculate the motion input. Optical flow is a 2D displacement field that describes the apparent motion of objects, surfaces, and contours in a visual scene between two successive frames. It is computed based on the brightness constancy assumption (BCA), which assumes that the brightness of corresponding pixels remains constant in consecutive frames [43]. Optical flow is largely used to separate foreground from background and identify moving objects. It is extensively used in computer vision tasks, including segmentation [44], detection [45], recognition [46], and tracking [47].

Various approaches are proposed to estimate optical flow [48]. In this work, we use total variation (TV) regularization and the robust L^1 norm (TV-L^1) algorithm [49]. This popular and efficient optical flow algorithm is based on a differential method that computes the velocity from spatial and temporal derivatives of the image brightness.

For fish detection, the output of optical flow is used in conjunction with the successive grayscale frames to retain texture and shape information.

3.2 One-stream object detection

To localize the object in video frames, we propose using the Faster R-CNN model [29]. This detector is robust in complex and variable environments, thus making it the most accurate model for object localization [50]. Faster R-CNN consists of three neural networks as illustrated in Fig. 3: base network (backbone), region proposal network (RPN), and classifier network.

The base network or backbone is usually a well-known pretrained CNN such as AlexNet [51], VGG [52], GoogleNet [53], and ResNet [54]. It provides feature maps from the input image. In our work, we use ResNet-50, which is a residual network architecture with ImageNet pretrained weights [55].

RPN is a CNN that takes feature maps generated by the base network as the input and generates a number of bounding boxes known as regions of interest (ROIs), which have a high probability of containing objects of interest. The RPN slides 3×3 spatial windows over the feature maps to produce K proposal boxes known as anchor boxes, which have different aspect ratios and scales to capture the objects of different sizes in the input image. These anchor boxes are fed into two layers: the classification layer classifies each anchor as foreground or background to detect the ROI, and the regression layer generates the ROI bounding box. These ROIs are then passed through the ROI pooling layer to feed the classifier network.

The classifier network is the final layer in the model. It is a fully connected network with two sublayers: the classification layer predicts the final class for each proposed ROI using the softmax activation function, and the regression layer produces bounding box coordinates using linear activation function.

Faster R-CNN has two supervised training techniques: alternative training [56] or approximate joint end-to-end training [57]. In our training process, we use the approximate joint end-to-end strategy via the backpropagation algorithm.

3.3 Multi-stream object detection

We propose two approaches to fuse appearance and motion information. The objective of the fusion is to better detect moving objects using the features extracted from both

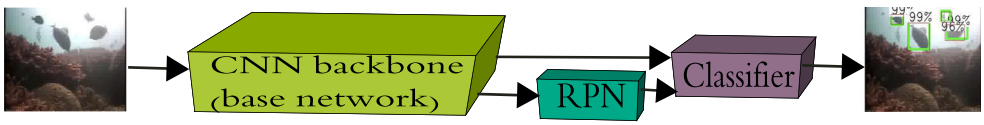

Fig. 3 Faster R-CNN detector with three neural networks: base network, region proposal network (RPN), and classifier network

streams. Figure 4 shows the two proposed approaches for multi-stream object detection.

- **Shared RPN fusion:** In this hybrid fusion framework (Fig. 4a), one RPN is shared between two base networks. The RPN input is the fusion of features extracted from the two base networks, which generates the ROIs. Two classifiers project these ROIs onto the corresponding CNN stream outputs. Finally, a decision-fusion level is placed at the end of the architecture to merge the outputs of the two classifiers and obtain better results. The advantage of this approach is that the RPN has a richer space (appearance and motion) so it can better predict ROIs. Each input stream has its own classifier but operates on common ROIs with fewer missed detections.

  We investigate three techniques of decision fusion: NMS, ELM, and SVM. NMS is used to reduce redundant detection boxes by conserving the best detection box with the highest score and removing other detection boxes that largely overlap. Nevertheless, we combine the output scores of both streams to feed a designed ELM network or SVM classifier to reclassify each region into fish and non-fish classes. We chose ELM and SVM, because their training processes do not rely on the backpropagation algorithm, which is extremely time-consuming. Moreover, ELM and SVM are efficient in the classification task with good training speeds.
- **Shared classifier fusion:** This early fusion framework (Fig. 4b) shares a single classifier but uses two RPNs corresponding exclusively to the appearance and motion streams. The appearance RPN proposes regions based on appearance, and the motion RPN generates regions based on motion. The NMS placed after the two RPNs allows ROI sharing in order to choose only the best. Then, the classifier projects these ROIs onto the merged features extracted from the two base networks. This technique allows us to obtain a richer space for the classifier and extract the correlation between the merged features. This architecture has a single classifier with fewer parameters to optimize. Further, the two RPNs cooperate to generate more precise ROIs.

4 Experiments

In Section 4.1, we present the LCF-15 benchmark dataset. To evaluate the detection performances, we use two measures presented in Section 4.2. Sections 4.3 and 4.4 evaluate the shared RPN and the shared classifier, respectively. Finally, we compare our findings with the state-of-the-art methods in Section 4.5. We used computer systems equipped with Intel Core-i5 processors with Geforce GTX 1050 Ti GPU and 2 Go GPU memory. We implemented the proposed approach in python using Keras with the TensorFlow library backend and TV-L[1] algorithm for optical flow.[5]

4.1 LifeClef 2015 fish (LCF-15) benchmark Dataset

LCF-15 is an underwater video dataset derived from the European project F (F4K) [58]. F4K is a large dataset with over 700,000 unconstrained underwater videos collected over a period of 5 years from the world's largest fish biodiversity environment in Taiwan with over 3,000 fish species.

The LCF-15 dataset is used for fish detection and species recognition. The training set includes 20 videos that are manually labeled and agreed by two specialists. The test set features 73 annotated videos with challenging underwater images and videos marked by noisy and blurry environments, complex and dynamic backgrounds with rich coral reefs and moving plants, poor and varying illumination conditions, and crowded fish. Figure 1 shows some underwater video frames extracted from the LCF15 dataset to illustrate the high variability in the underwater environment (background, luminosity, and visibility) and fish characteristics (texture, color, size, and shape).

In our experiments, we use input images measuring 640× 480 pixels. For RPN, we consider four different scales (32, 64, 128, 256) each with four different aspect ratios (1:1, 1:2, 2:1, $2/\sqrt{2}$: $2/\sqrt{2}$) to generate 16 anchor boxes.

4.2 Evaluation metrics

We use two measures to evaluate the detections obtained using the proposed approaches: mean average precision (mAP) and F-score. These two measures are defined by recall (R) and precision (P) scores as follows:

$$mAP = \int_0^1 P(R)dR \tag{1}$$

$$F = \frac{2 \times P \times R}{P + R} \tag{2}$$

where:

$$P = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{3}$$

$$R = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{4}$$

These measures are calculated based on the intersection over union (IoU) metric, which is the ratio between the intersection and union of the ground truth and the detected

[5] https://github.com/vinthony

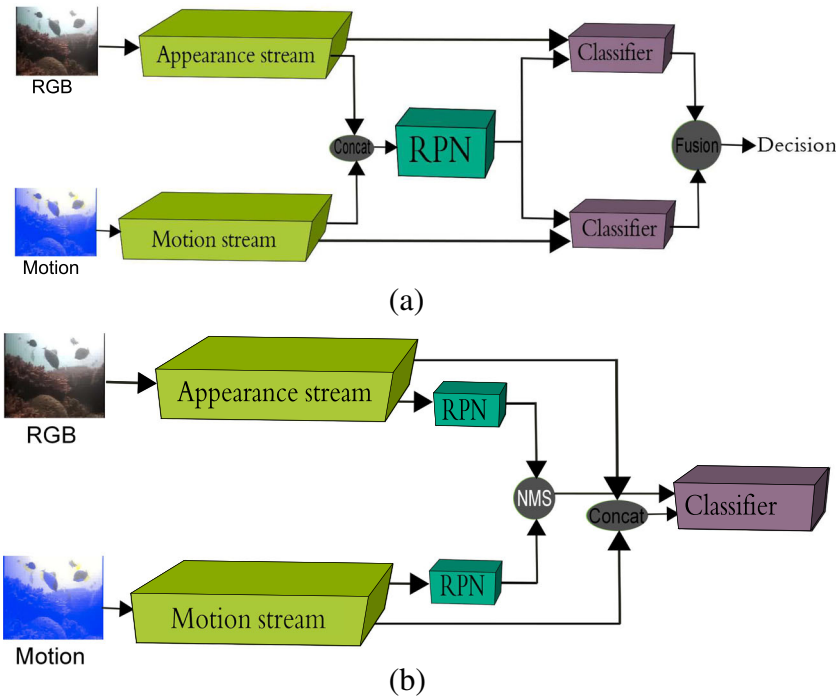

Fig. 4 Proposed fusion approaches. **a** Shared region proposal network (RPN) fusion using one RPN and two classifiers. **b** Shared classifier fusion using two RPNs and a unique classifier

fish boxes. For good fish detection, we consider the case when $IoU \geq 0.5$.

4.3 Shared RPN framework

First, we consider the shared RPN framework (Fig. 4a). Before investigating the final fish detection, we will analyze the behavior of the shared RPN.

4.3.1 Stand-alone RPN *versus* shared RPN

To better understand the effect of fusion on the RPN result, we begin by comparing the performances of the "shared RPN" with the stand-alone RPN, which operates with individual inputs (classic Faster R-CNN [29]). For this purpose, we consider two one-stream Faster R-CNN models trained for only one type of information (RGB or motion). Likewise, the classification is performed using the same information without merging. Table 1 compares the performance of the fish detection of the different schemes using the LCF-15 dataset. For the shared RPN fusion, the table only lists the results of each stream before the decision fusion.

Table 1 shows that our RPN performs better than the stand-alone RPN, which is only trained with appearance or motion information. This is due to the richer space of our RPN, which proposes more confident regions. We also note that motion models are more efficient than appearance models. The optical flow algorithm produces more ROIs due to its sensitivity to every motion in the image or luminosity changes. Moreover, the motion features are more relevant; in addition to spatial information in grayscale frames, they also capture temporal information such as fish movements, luminosity variations, and background changes.

Figure 5 shows examples of classifier outputs for each one-stream Faster R-CNN and each stream of the shared RPN model. Interestingly, in the first two rows, our RPN proposes new ROIs. The RPNs of the one-stream Faster R-CNN model cannot propose confident ROIs for these fish. Nevertheless, our RPN proposes them for both classifiers. Therefore, they would be well classified by at least one of the classifiers. Another important observation is that the shared RPN can remove false positive regions (third row), which increases the precision. However, it can also remove true regions (last row), which decreases the recall.

Table 1 Comparison of fish detection performances (in percentage) between the stand-alone Faster R-CNN and our approach of "shared RPN fusion"

Approach	Input	F-score	mAP
One-stream	RGB	77.82	64.71
	Motion	78.78	67.49
Shared RPN fusion	RGB	79.47	67.04
	Motion	80.22	70.50

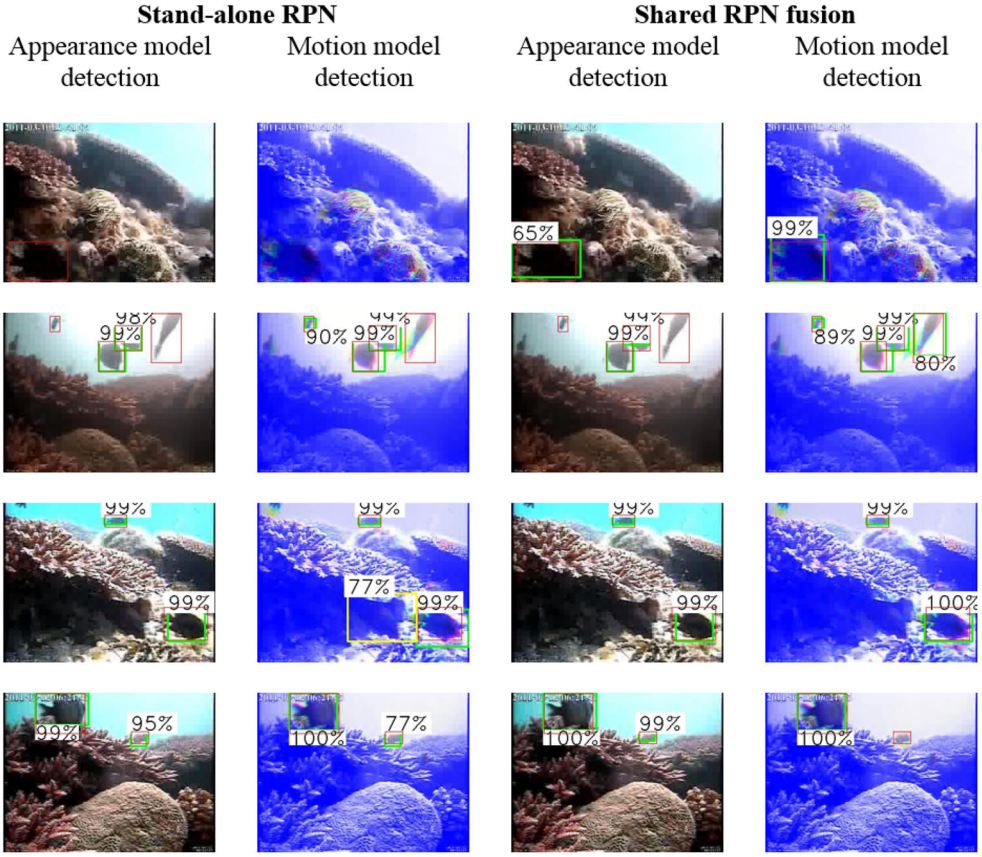

Fig. 5 Examples of predictions with one-stream Faster R-CNN [29] and shared RPN fusion. From left to right: the first two columns are the classifier outputs of one-stream Faster R-CNN trained with RGB or motion images. The last two columns are the appearance and motion classifier outputs of the shared RPN. Red boxes indicate ground truth boxes, green boxes correctly detected fish, and yellow boxes false positive detection boxes

4.3.2 Evaluation of decision fusion techniques

At the end of our shared RPN architecture, we use a decision fusion operation to merge classifier outputs and improve performances. Here, we test three techniques: NMS, SVM, and ELM. The fusion results are shown in Table 2, while a few examples of the different decision fusion techniques are illustrated in Fig. 6.

Based on Table 2 and Fig. 6, we observe that the three decision fusion techniques achieve better F-scores compared to the appearance or motion stream alone. The NMS technique has better performances than ELM and SVM. NMS accumulates detection boxes from the two classifiers for better detection, which increases the recall (see first and second row, Fig. 6). In the third row of Fig. 6, NMS reorders the boxes by score and preserves those with the highest score. Consequently, NMS fusion increases the mAP. However, with this technique, there are numerous false detection boxes (fourth row), which decreases the precision. Nevertheless, ELM and SVM yield prediction results with fewer false positives, although some true positive boxes are removed. Finally, the best F-score measure (**83.16%**) and mAP (**73.69%**) are obtained using the NMS technique.

Table 2 Fish detection performances (in percentage) for the shared RPN fusion approach with various fusion techniques using the LCF-15 dataset

	Technique	F-score	mAP
Shared RPN	RGB	79.47	67.04
	Motion (Mo)	80.22	70.50
	NMS (RGB+Mo)	83.16	73.69
	SVM (RGB+Mo)	81.59	70.08
	ELM (RGB+Mo)	81.83	70.57

4.4 Shared classifier framework

The shared classifier fusion strategy is less efficient compared to the shared RPN fusion. We can only reach an F-score of 74.12% and mAP of 62.85% on the LCF-15 dataset. The difficulty with this architecture is that fish detection in an unconstrained environment is a complicated task, so the feature space using RGB and motion data is vast for a single classifier.

Figure 7 presents the precision-recall performance curves for both the shared RPN and the shared classifier fusion techniques. As we can see, the shared classifier approach has a better precision than the shared RPN due to fewer false

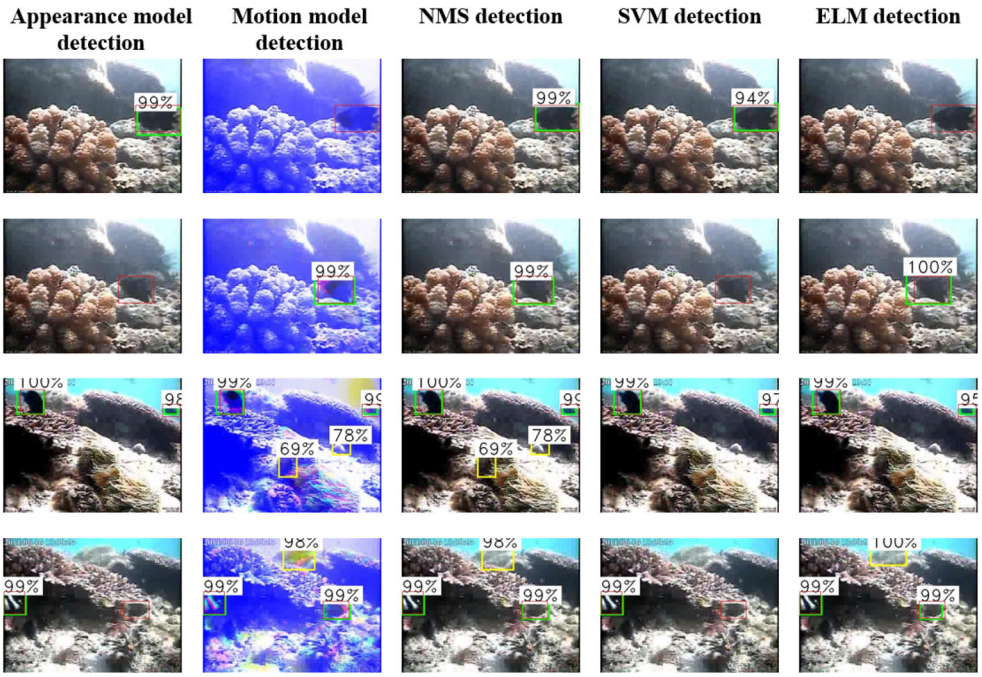

Fig. 6 Examples of predictions with the shared RPN fusion framework using different decision fusion techniques. From left to right: appearance classifier outputs of the shared RPN, motion classifier outputs of the shared RPN, NMS fusion, SVM, and ELM. Red boxes indicate ground truth boxes, green boxes correctly detected fish, and yellow boxes false positive detection boxes

positives, although its recall is low. Nevertheless, the shared RPN increases the recall from 60.12% to 76.03% without greatly reducing precision.

4.5 Comparison with the state-of-the-art

In this section, we compare our proposed architectures with the state-of-the-art approaches. Table 3 reports the comparative results on the LCF-15 benchmark dataset, showing that input-level fusion based on one-stream Faster R-CNN performs better than classic Faster R-CNN. Salman et al. [16] proposed the fusion of GMM, optical flow, and grayscale images. However, GMM has many disadvantages: the segmentation results are not robust to noise or sensitive to luminosity variations and other environmental factors such as aquatic plant movements, ocean currents, and camera shaking. Feature-level fusion aims to merge features from two streams to feed a single classifier, although this fusion strategy produces poor performances compared to other strategies. Late fusion (or decision-level fusion) performs better than early fusion. Finally, our hybrid fusion performs the best. Unlike early fusion, late and hybrid fusions contain two classifiers, which improve the performances.

Figure 8 illustrates examples of U- and X-fusion predictions. An important advantage of X-fusion is its ability to detect fish that cannot detected by one-stream networks (second row). This architecture shares individual ROIs at the intermediate stage. Thus, the one-stream network can propose a ROI but cannot identify the fish inside it. With the addition of a second stream, however, the network can identify fish, even though the stream did not propose this region as a ROI. Unfortunately, this technique also generates new false detection boxes (third row). In this case, the ROI was well classified by one stream, but after sharing ROIs, the second stream could not classify it. The

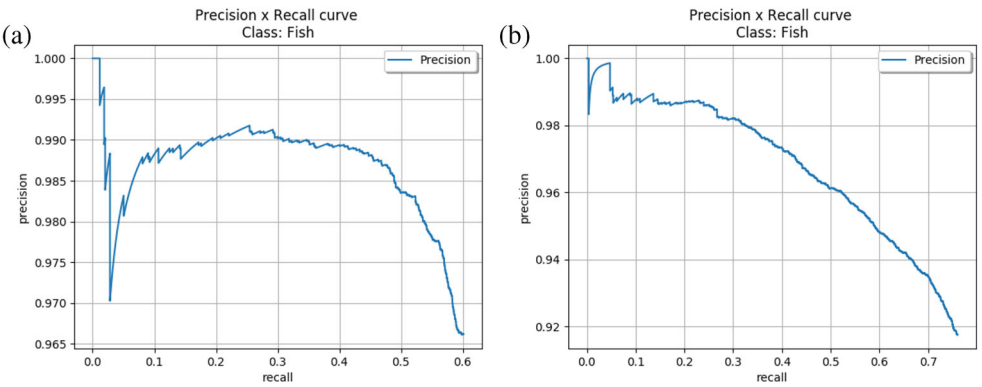

Fig. 7 Precision-recall curves for both proposed approaches **(a)** Shared classifi er **(b)** Shared RPN

Table 3 Fish detection performances (in percentage) compared to the state-of-the-art approaches on the LCF-15 dataset

Fusion approach	Technique		F-score	mAP	Architecture
Faster R-CNN classic	RGB [29]		77.82	64.71	
Early fusion	Input-level fusion	Motion [17]	78.78	67.49	One-stream
		Salman et al. [16]	80.02	-	
	Feature-level fusion	Y-fusion [19]	71.72	61.85	Mutli-streams
		Zhu et al. [18]	70.73	61.48	
		Shared Classifier (ours)	74.12	62.85	
Late fusion	U-fusion [19]		82.24	71.88	
	X-fusion [19]		82.14	71.83	
Hybrid fusion	Shared RPN (ours)		83.16	73.69	

intermediate NMS only keeps ROIs with the highest scores and removes other regions. Therefore, a few false detection boxes with low scores were removed (fourth row), although several true detection boxes were also removed (fifth row).

U-fusion does not use this ROI sharing technique before the classification step, although it reorders by score the detection boxes placed at the end of the architecture in the NMS. As X-fusion also uses an NMS at the end, it generally

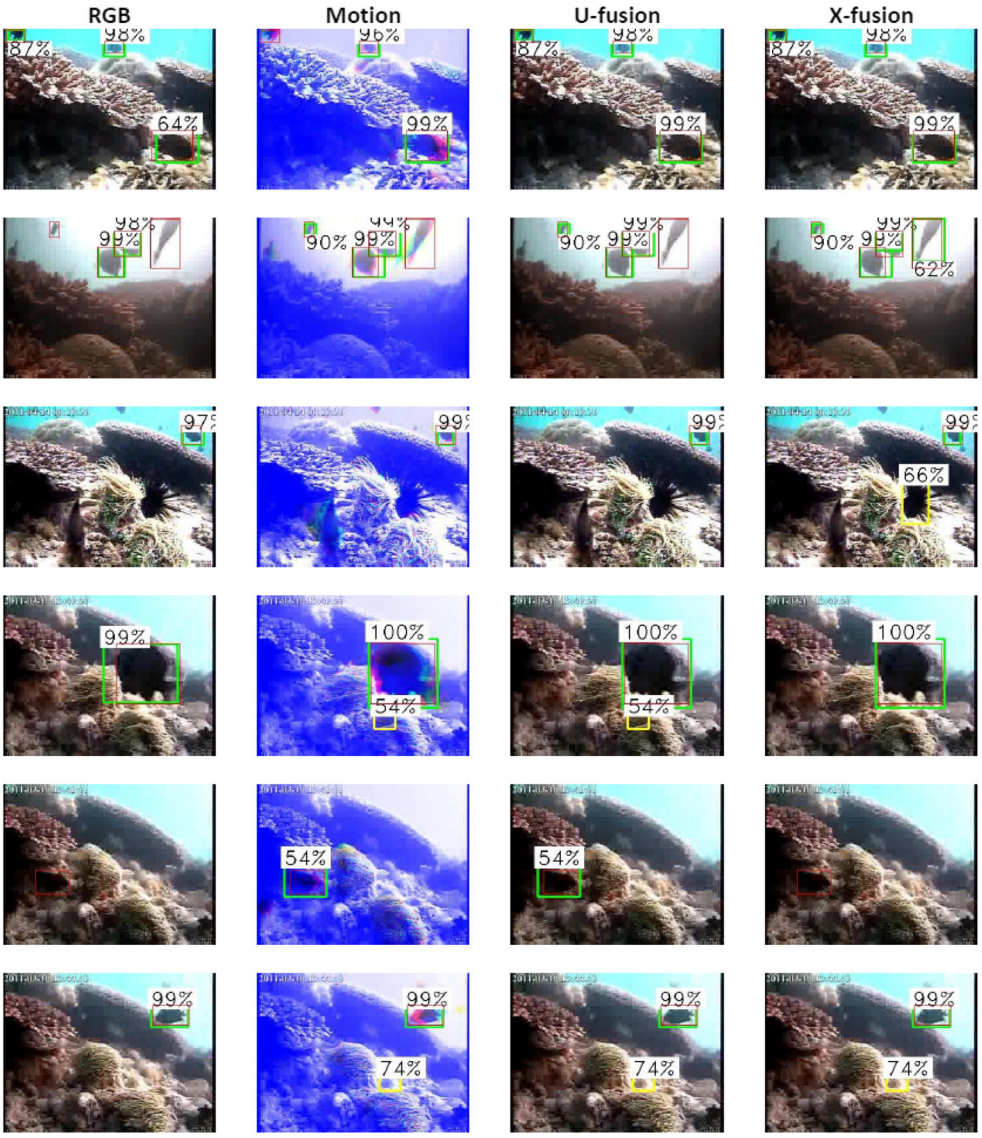

Fig. 8 Examples of predictions with U-fusion and X-fusion. From left to right: appearance and motion classifier outputs, U-fusion, and X-fusion. Red boxes indicate ground truth boxes, green boxes correctly detected fish, and yellow boxes false positive detection boxes

shows a similar behavior to U-fusion. Figure 8 shows that detection boxes from the two streams are accumulated for better detection, although false detection boxes are also present (last row), which decreases the precision of the method.

Unlike our shared RPN approach, U- and X-fusion do not simultaneously train the two streams; they are instead trained independently and only merged at the end of training. Our approach allows the RPN to learn from the two feature spaces (appearance and motion) in order to better propose ROIs.

Feature-level fusion strategies with a single classifier are less efficient than late and hybrid fusions with two classifiers. Unlike our shared classifier architecture, Y-fusion and the approach of Zhu et al. [18] use only one RPN and one classifier. The feature space in Y-fusion is thus larger for both the RPN and the classifier; it is also larger for the classifier in [18], which makes the training highly sensitive. In [18], RPN only uses one type of data to generate ROIs. Consequently, it does not use complementary information, which could be relevant in order to propose more reliable regions. These three architectures are early fusions: they fuse fish features extracted from two streams. The dimension of merged features becomes large, which increases the overfitting risk.

We conclude that multi-stream late and hybrid fusions with two classifiers significantly improve the performances of automatic fish detection, especially for our shared RPN hybrid architecture. We achieve an F-score of **83.16%** and mAP of **73.69%** for shared RPN, whereas U- and X-fusions have F-scores of 82.24% and 82.14% and mAPs of 71.88% and 71.83%, respectively. Consequently, we outperform the state-of-the-art methods.

5 Conclusion

In this paper, we propose multi-stream fusion approaches for the detection of underwater moving objects. These approaches are based on the merging of two Faster R-CNN models that share the same RPN or classifier with the objective to improve the performances of object localization by fusing information extracted from multi-modalities or multi-spaces. We applied these new fusion architectures to fish detection in underwater videos captured in unconstrained environments. We used raw RGB video frames to capture the appearance features and optical flow combined with two successive raw grayscale frames to incorporate motion information into the feature space. Experiments on the LifeClef 2015 Fish benchmark dataset demonstrated that our shared RPN fusion approach performs better than the shared classifier fusion and that its performances exceed state-of-the-art methods for automatic fish detection tasks.

As demonstrated, the NMS gives the best results among the decision fusion techniques but accumulates false positives. Our future work aims to improve the NMS by incorporating previous predictions. We also intend to extend our fusion architectures to other detection tasks by using different modalities such as depth, infrared, and radar. In the fish detection task, we aim to construct a fully automatic system that combines fish detection with species classification in the presence of a large number of fish species.

Acknowledgements The authors would like to thank the Région Bretagne for financial support.

References

1. Zivkovic Z (2004) Improved adaptive Gaussian mixture model for background subtraction. In: Proceedings of the 17th international conference on pattern recognition. ICPR 2004, vol 2. IEEE, pp 28–31
2. Spampinato C, Chen-Burger YH, Nadarajan G, Fisher RB (2008) Detecting, tracking and counting fish in low quality unconstrained underwater videos. VISAPP 1(2):514–519
3. Hsiao YH, Chen CC, Lin SI, Lin FP (2014) Real-world underwater fish recognition and identification, using sparse representation. Ecol Inf 23:13–21
4. Bengio Y (2009) Learning deep architectures for AI. Now Publishers Inc.
5. LeCun Y, Bengio Y (1995) Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), 1995
6. Li X, Shang M, Qin H, Chen L (2015) Fast accurate fish detection and recognition of underwater images with fast r-cnn. In: OCEANS 2015-MTS/IEEE Washington. IEEE, pp 1–5
7. Li X, Shang M, Hao J, Yang Z (2016) Accelerating fish detection and recognition by sharing CNNs with objectness learning. In: OCEANS 2016-Shanghai. IEEE, pp 1–5
8. Li X, Tang Y, Gao T (2017) Deep but lightweight neural networks for fish detection. In: OCEANS 2017-Aberdeen. IEEE, pp 1–5
9. Hong S, Roh B, Kim KH, Cheon Y, Park M (2016) PVANet: lightweight deep neural networks for real-time object detection. arXiv:1611.08588
10. Mandal R, Connolly RM, Schlacher TA, Stantic B (2018) Assessing fish abundance from underwater video using deep neural networks. In: 2018 international joint conference on neural networks (IJCNN). IEEE, pp 1–6
11. Zhuang P, Xing L, Liu Y, Guo S, Qiao Y (2017) Marine animal detection and recognition with advanced deep learning models. In: CLEF (Working Notes)
12. Shi C, Jia C, Chen Z (2018) FFDet: a fully convolutional network for coral reef fish detection by layer fusion. In: 2018 IEEE visual communications and image processing (VCIP). IEEE, pp 1–4
13. Sung M, Yu SC, Girdhar Y (2017) Vision based real-time fish detection using convolutional neural network. In: OCEANS 2017-Aberdeen. IEEE, pp 1–6
14. Jäger J, Rodner E, Denzler J, Wolff V, Fricke-Neuderth K (2016) SeaCLEF 2016: object proposal classification for fish detection

in underwater videos. In: CLEF (Working Notes), pp 481–489
15. Zhang D, Kopanas G, Desai C, Chai S, Piacentino M (2016) Unsupervised underwater fish detection fusing flow and objectiveness. In: 2016 IEEE winter applications of computer vision workshops (WACVW). IEEE, pp 1–7
16. Salman A, Siddiqui SA, Shafait F, Mian A, Shortis MR, Khurshid K, Schwanecke U (2019) Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES J Marine Sci
17. Farahnakian F, Heikkonen J (2020) Deep learning based multi-modal fusion architectures for maritime vessel detection. Remote Sens 12(16):2509
18. Zhu X, Chen C, Zheng B, Yang X, Gan H, Zheng C, Xue Y (2020) Automatic recognition of lactating sow postures by refined two-stream RGB-D faster R-CNN. Biosys Eng 189:116–132
19. Guerry J, Le Saux B, Filliat D (2017) "Look at this one" detection sharing between modality-independent classifiers for robotic discovery of people. In: 2017 European conference on mobile robots (ECMR). IEEE, pp 1–6
20. Wang Y, Song J, Wang L, Van Gool L, Hilliges O (2016) Two-stream SR-CNNs for action recognition in videos. In: BMVC
21. Morvant E, Habrard A, Ayache S (2014) Majority vote of diverse classifiers for late fusion. In: Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR). Springer, Berlin, pp 153–162
22. He K, Cao X, Shi Y, Nie D, Gao Y, Shen D (2018) Pelvic organ segmentation using distinctive curve guided fully convolutional networks. IEEE Trans Med Imag 38(2):585–595
23. Monkam P, Qi S, Xu M, Li H, Han F, Teng Y, Qian W (2018) Ensemble learning of multiple-view 3D-CNNs model for micro-nodules identification in CT images. IEEE Access 7:5564–5576
24. Wöllmer M, Weninger F, Knaup T, Schuller B, Sun C, Sagae K, Morency LP (2013) Youtube movie reviews: sentiment analysis in an audio-visual context. IEEE Intell Syst 28(3):46–53
25. Poria S, Cambria E, Howard N, Huang GB, Hussain A (2016) Fusing audio, visual and textual clues for sentiment analysis from multimodal content. Neurocomputing 174:50–59
26. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
27. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW (2013) Selective search for object recognition. Int J Comput Vis 104(2):154–171
28. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
29. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
30. Dai J, Li Y, He K, Sun J (2016) R-fcn: object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387
31. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2013) Overfeat: integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229
32. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
33. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, Cham, pp 21–37
34. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
35. Corneanu CA, Simón MO, Cohn JF, Guerrero SE (2016) Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: history, trends, and affect-related applications. IEEE Trans Pattern Anal Mach Intell 38(8):1548–1568
36. Potamianos G, Neti C, Gravier G, Garg A, Senior AW (2003) Recent advances in the automatic recognition of audiovisual speech. Proc IEEE 91(9):1306–1326
37. Chen C, Jafari R, Kehtarnavaz N (2017) A survey of depth and inertial sensor fusion for human action recognition. Multimedia Tools Appl 76(3):4405–4425
38. James AP, Dasarathy BV (2014) Medical image fusion: a survey of the state of the art. Inf Fus 19:4–19
39. Liu X, Zhu X, Li M, Wang L, Zhu E, Liu T, Gao W (2019) Multiple kernel k k-means with incomplete kernels. IEEE Trans Pattern Anal Mach Intell 42(5):1191–1204
40. Peng X, Schmid C (2016) Multi-region two-stream R-CNN for action detection. In: European conference on computer vision. Springer, Cham, pp 744–759
41. Yu X, Ye X, Gao Q (2020) Infrared handprint image restoration algorithm based on apoptotic mechanism. IEEE Access 8:47334–47343
42. Bianco G, Muzzupappa M, Bruno F, Garcia R, Neumann L (2015) A new color correction method for underwater imaging. Int Arch Photog Remote Sens Spat Inf Sci 40(5):25
43. Horn B, Berthold KP (1981) Schunck. Determining optical flow. Artif Intell 17(1–3):185–203
44. Tsai YH, Yang MH, Black MJ (2016) Video segmentation via object flow. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3899–3908
45. Xu D, Yan Y, Ricci E, Sebe N (2017) Detecting anomalous events in videos by learning deep representations of appearance and motion. Comput Vis Image Understand 156:117–127
46. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576
47. Xiao F, Jae Lee Y (2016) Track and segment: an iterative unsupervised approach for video object proposals. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 933–942
48. Tu Z, Xie W, Zhang D, Poppe R, Veltkamp RC, Li B, Yuan J (2019) A survey of variational and CNN-based optical flow techniques. Sig Process Image Commun 72:9–24
49. Zach C, Pock T, Bischof H (2007) A duality based approach for realtime TV-L 1 optical flow. In: Joint pattern recognition symposium. Springer, Berlin, pp 214–223
50. Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Murphy K (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7310–7311
51. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
52. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
53. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9

54. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
55. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255
56. Zuo Z, Yu K, Zhou Q, Wang X, Li T (2017) Traffic signs detection based on faster r-cnn. In: 2017 IEEE 37th international conference on distributed computing systems workshops (ICDCSW). IEEE, pp 286–288
57. Lei HW, Wang B, Wu HH, Wang AH (2018) Defect detection for polymeric polarizer based on faster R-CNN. J Inf Hid Multimed Sign Process 9:1414–1420
58. Boom BJ, Huang PX, He J, Fisher RB (2012) Supporting ground-truth annotation of image datasets using clustering. In: Proceedings of the 21st international conference on pattern recognition (ICPR2012). IEEE, pp 1542–1545

Abdelouahid Ben Tamou received the Master degree in Computer Science and Telecommunications from Mohammed V University in Rabat Faculty of Science in 2015 Morocco. He is currently pursuing his Ph.D in signal and image processing in LabSTICC Laboratory, Ecole Nationale d'Ingenieurs de Brest, France, and LRIT laboratory, Mohammed V University in Rabat, Morocco. His reserach interests include image processing, computer vision and machine learning in particular for underwater environment applications.

Abdesslam Benzinou received the Ph.D. degree in signal and image processing from Universite de Bretagne Occidentale, Brest, France, in January 2000. In September 2000, he was a Researcher of signal processing for metrology and communication systems with Schlumberger-RMS, Chasseneuil, France. He is currently a Professor with Ecole Nationale d'Ingenieurs de Brest, Brest, where he joined in September 2001. He is leader of the OSE Team, LabSTICC Laboratory. His current research focus on signal/image processing, artificial intelligence and computer vision. He is particularly interested in sea and environment monitoring and surveillance.

Kamal Nasreddine has obtained in 2006 his diploma of electronic engineering from the Lebanese University in Lebanon, Beirut. He received the Ph.D. degree in signal and image processing from the "Université de Bretagne Occidentale", Brest, France, in November 2010. He is currently an Associate Professor in the "Ecole Nationale d'Ingénieurs de Brest", Brest, France. His research fields within the Labsticc lab (CNRS, UMR 6285) concern signal processing and computer vision in particular for biomarine and biosanitary applications.