Two-Stage Convolutional Neural Network for Ship and Spill Detection Using SLAR Images

Mario Nieto-Hidalgo, Antonio-Javier Gallego, Pablo Gil, *Senior Member, IEEE*, and Antonio Pertusa

***Abstract*— This paper presents a system for the detection of ships and oil spills using side-looking airborne radar (SLAR) images. The proposed method employs a two-stage architecture composed of three pairs of convolutional neural networks (CNNs). Each pair of networks is trained to recognize a single class (ship, oil spill, and coast) by following two steps: a first network performs a coarse detection, and then, a second specialized CNN obtains the precise localization of the pixels belonging to each class. After classification, a postprocessing stage is performed by applying a morphological opening filter in order to eliminate small look-alikes, and removing those oil spills and ships that are surrounded by a minimum amount of coast. Data augmentation is performed to increase the number of samples, owing to the difficulty involved in obtaining a sufficient number of correctly labeled SLAR images. The proposed method is evaluated and compared to a single multiclass CNN architecture and to previous state-of-the-art methods using accuracy, precision, recall, F-measure, and intersection over union. The results show that the proposed method is efficient and competitive, and outperforms the approaches previously used for this task.**

***Index Terms*— Neural networks, oil spill detection, radar detection, side-looking airborne radar (SLAR), supervised learning.**

I. Introduction

THE presence of an oil slick on the sea surface requires early detection in order for active emergency protocols focused on controlling the environmental impact and ecological damage to be carried out. It is also necessary for governments to identify illegal boats in order to impose sanctions. Detection and monitoring are usually performed using two principal kinds of sensors: the synthetic aperture radar (SAR) installed on satellites and the side-looking airborne radar (SLAR) mounted on the aircrafts.

Both sensors allow governments to monitor a wide marine area 24 h a day. Since SAR has a greater range and resolution than SLAR and can also work in all weather, SLAR can monitor any place at any time whenever weather conditions allow aircraft to fly, in contrast with SAR which depends on its orbit.

Other sensors that are not based on microwaves are also used for oil spill detection from a high altitude, such as the optical or visible spectrum and infrared (IR). IR has visibility limitations in adverse weather conditions (clouds, rain, and so on), whereas visible sensors do not work well at night. Other sensors, such as ultraviolet and the microwave radiometer, are usually used to measure the thickness and volume of the spills (not to detect them), since SAR and SLAR cannot generally discriminate thickness, as mentioned on the work by Leifer *et al.* [1], in which the behavior and specifications of the most relevant spaceborne sensors for oil spill remote sensing are also discussed.

SAR and SLAR have also been used for ship detection. Both sensors make it possible to observe man-made metallic targets on the sea. Ship detection systems can be useful to identify the ships fishing in unauthorized waters, outside trade routes (illegal traffic), and close to oil slicks. In the latter case, the ship could potentially be considered as the source of the oil spill.

Both SLAR and SAR images represent oil spills as dark spots on the marine surface. However, some ocean phenomena (low-wind area, surge, and so on), natural activities (coral reef, phytoplankton blooms, fish and algae banks, and so on), and human actions may also cause dark regions. These dark spots are known as look-alikes. The implementation of automatic detection methods to discriminate between look-alikes and oil spills is an important challenge for remote sensing when the input data are SLAR or SAR. Moreover, ships in these kinds of images are represented as bright spots. However, small islands or islets, sea conditions (waves, shoals of fish, and so on), and coast–sea contours can complicate the detection of ships because all of them cause noisy bright spots.

There are two ways to approach the oil spill detection problem: by using multipolarization features to study the characterization of the slick [2], [3] or, as mentioned in this paper, by using the intensity image obtained from a scatter signal without considering the parameters of the image acquisition and formation processes.

As it is shown in [4] and [5], the majority of the methods employed for oil spill detection using intensity SAR imagery are based on the image processing techniques. Automatic ship

Manuscript received October 23, 2017; revised January 10, 2018; accepted February 28, 2018. This work was supported in part by the Spanish Government's Ministry of Economy, Industry, and Competitiveness under Project RTC-2014-1863-8 and in part by Babcock MCS Spain under Project INAER4-14Y (IDI-20141234). *(Corresponding author: Antonio-Javier Gallego.)*

M. Nieto-Hidalgo is with the Computer Science Research Institute, University of Alicante, 03690 Alicante, Spain.

A.-J. Gallego and A. Pertusa are with the Pattern Recognition and Artificial Intelligence Group, Department of Software and Computing Systems, University of Alicante, 03690 Alicante, Spain, and also with the Computer Science Research Institute, University of Alicante, 03690 Alicante, Spain (e-mail: ajgallego@gmail.com).

P. Gil is with the Automation, Robotics and Computer Vision Group, Department of Physics, Systems Engineering and Signal Theory, University of Alicante, 03690 Alicante, Spain, and also with the Computer Science Research Institute, University of Alicante, 03690 Alicante, Spain.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2812619

detection with SAR has been widely studied and was most recently reviewed in [6]–[8]. When compared to the large amount of feasibility analyses for both SAR-based oil spill and ship detection, very few research studies use SLAR for the same purpose.

An important difference between SAR and SLAR images is due to how the radar is mounted on satellites or aircrafts, respectively. SLAR antenna changes its position and orientation according to the direction and the turns made by the maneuvers of the aircraft. Therefore, in SLAR, the observer's perspective is not fixed and it causes more noise than SAR. In our case, the aircraft is a fixed-wing aircraft EADS-CASA CN 235-300 equipped with a TERMA SLAR 9000 with two antennas under the wings, one on each side and perpendicular to the flight direction. This causes scan failures where there is no intersection in the field of view. An example of these failures is produced in the central zone below the aircraft, registered as measurement errors and represented as artifacts or noise in the SLAR image.

Another example is when the aircraft turns causing that one of the two antennas points to the sky and, consequently, registers errors in the acquired SLAR image. These problems cause simultaneous dark and bright spots, which can be confused in the detection process with oil spills, coasts, ships, or other targets. Therefore, it is more common that the noise pixels are present in SLAR than in SAR, hindering the detection process.

It is currently difficult to find the works that address the detection of oil spills through the use of SLAR images. A previous method in this line was presented in [9] in which a system to detect oil spills using recurrent neural networks (RNN) was proposed. The RNN took several adjacent image rows as input, obtaining a test accuracy of 97%.

In this paper, we propose the use of a two-stage convolutional neural network (CNN) for the task of detecting and locating ships, oil spills, and coasts using SLAR sensor data. CNNs are multilayer architectures designed to extract high-level representations of a given input. They have dramatically improved the state of the art as regards image, video, speech, and audio recognition tasks [10]–[12] due to their ability to perform suitable feature transformations for the task at hand. In most computer vision tasks, such as image segmentation [13], [14] CNNs, clearly outperform traditional approaches.

The proposed architecture uses a combination of two-stage CNNs, each of which is specialized in the detection of a type of target, in order to increase the classification accuracy. This architecture provides a coarse detection of the targets over a wide area followed by a per pixel detection to finely locate the targets. This technique increases both the accuracy and the time performance, as the fine detection stage is only executed in the areas in which a target has been previously detected by the coarse stage. In order to overcome the limited amount of data, the proposed data augmentation process and the combination of binary classifiers using the one-vs-rest strategy provide better results with few training data. As shown in the evaluation results, the presented approach experimentally outperformed previous state-of-the-art methods based on image processing and traditional machine learning techniques which use hand-engineered features.

The rest of this paper is organized as follows. Section II provides a brief review of the state of the art as regards the automatic detection of oil spills and vessels using SAR and SLAR sensors data. Section III details the two-stage CNN architecture proposed. Section IV describes the metrics used to evaluate our method, provides a description of the data set used for the experimentation and also presents the experiments, along with a discussion of the results. Finally, Section V shows our conclusion and future work.

II. Background

Until 2010, the majority of methods employed for oil spill detection using SAR images were based on three steps: region segmentation focused on dark spot detection, slick feature extraction, and spot classification, as stated by Brekke and Solberg [4]. Years later, the same authors introduced the regularization of covariance matrices to decrease the number of false positives (FPs) using statistical classifiers and support vector machines (SVMs) [5].

One of the pioneering works as regards discrimination between oil spills and look-alikes was that of Topouzelis *et al.* [15], who proposed a feature vector composed of ten features based on the area, shape, and colors of the instances. The results obtained when using a multilayer perceptron (MLP) with the proposed feature vector yielded a discrimination accuracy of 89%.

More recently, Xu *et al.* [16] compared certain machine learning techniques, such as SVM, generalized linear models, boosting trees, linear discriminant analysis, and MLP among others, taking SAR images from RADARSAT-1 as input. These classifiers were used to predict two classes (oil spills and look-alikes) using 15 features. We used the receiver operating characteristic (ROC) curve and specificity to measure the goodness of their method, in addition to employing cross-validation for the bias-reduced estimation of performance measures.

The current trend in oil spill detection is that of using approaches focused on artificial intelligence techniques. Marghany [17] therefore presented a method based on a genetic algorithm which used data set images acquired by RADARSAT-2 operating in ScanSAR Narrow single-beam mode. The method achieved an oil spill detection of 90%, although the number of samples was small as only data obtained during three days were used for evaluation.

Mera *et al.* [18] used a database with 47 SAR images acquired by ENVISAT to test their approach based on moment invariants. These images were then used to characterize the shapes of candidate regions to be considered as oil spills, after which two classifiers, an MLP with three layers (9, 11, and 2 neurons) and a classification and regression tree, were used to distinguish between two classes: look-alikes and oil spills. The database was composed of 155 instances of look-alikes and 80 oil spills. We used 70% of the samples for training and both classes were balanced.

Singha *et al.* [19] presented a method with which to classify oil spills and look-alikes. It was tested using images from two satellites, ENVISAT (35 images) and RADARSAT-2 (83 images), captured between 2009 and 2012 in CleanSeaNet. The images contained multiple look-alikes and oil spills. We carried out two tests. The results of ENVISAT test were 135 instances of oil spills and 805 look-alikes, whereas those of the RADARSAT-2 test were 226 instances of oil spills and 4923 look-alikes. The first test achieved accuracies of 61.48% and 89.44%, whereas the second achieved 54.43% and 94.86% for oil spills and look-alikes, respectively. However, we did not explain how the training and test phases were designed. Another issue of the approach in question is that the classes were unbalanced.

Guo and Zhang [20] presented another approach that could be used to discriminate between oil spills and other phenomena of a similar appearance. The method defined nine different shapes using a total of 50 eigenvalues (such as ratio, saturation, Hu moments, and so on), which were selected using differential evolution feature selection from the 95 shapes originally computed. They later used two different classifiers: a traditional neural network and a deep neural network. The method attained accuracies of 94% and 84%, respectively, with both classifiers. In work in question, the authors used 20 SAR images captured from ERS-1, ERS-2, and ENVISAT representing 833 instances of look-alikes and 222 oil spills.

Similar to oil spill detection, most of the methods are employed to detect ships use SAR images taken from satellites. Schwegmann *et al.* [21] used this kind of images and applied a low-threshold constant false alarm rate (CFAR) to identify ship candidates, after which they proposed a Haar-like feature extraction. The features obtained in this step were then fed into an adaptable cascade classifier. This obtained an accuracy of 89.38%. Another similar work based on CFAR that proposed an intensity space for ship detection in high resolution-SAR was presented in [22].

A further ship detection work by Wang and Chen [23] showed how a calculation of the local optimal window map was performed using multiple scales of the local contrast measure. The authors then went on to calculate the local variance weighted information entropy of each window and apply a mean-thresholding to detect targets, obtaining an accuracy of 100% with both homogeneous and heterogeneous backgrounds and of 75% with strong background noise.

Studies regarding inshore ship detection, which is a more complex task owing to the great similarity between the gray and textured features of a ship and the harbor, are also beginning to appear. Zhai *et al.* [24] presented an approach that employed saliency map and superpixel segmentation to discriminate ships. In this line, another method based on hierarchical saliency used by Wang *et al.* [25] was inspired by the multilayer selective cognition property of the human visual systems. The authors used random-forest techniques for classification. They evaluated their approach with only four HD-SAR images, achieving success rates of between 88% and 95%.

A method for detecting and removing land areas in order to improve ship detection was presented by Ji *et al.* [26]. The authors first performed a downsampling to make potential ships the size of 1 pixel and then used a median filter to remove all remaining ships. Water and land were subsequently distinguished by using a double thresholding. The first threshold was used globally to determine whether there was land in the image, and the second for the roughly detection of land areas. Finally, morphology operations were applied to remove noise and fill in holes.

While numerous works have been proposed with the intention of separately detecting oil spills or ships in SAR imagery, we are unaware of any that uses SLAR images for both of these tasks. This is an important innovation, because SLAR images have specific characteristics that distinguish them from those of SAR such as a much lower resolution, noise owing to bad signal scattering depending on the flight altitude, or data that are missing as a result of occlusions and aircraft maneuvers. SLAR images correspond to flight sequences of different durations which, when digitized, are transformed into images with the same size independently of the flight duration. Therefore, a pixel in an image can represent different scales, which produces deformations in objects depending on the speed and the altitude of the plane. This fact increases the object detection difficulty when using traditional methods. Furthermore, SLAR images have a lot of noise and artifacts that are similar to the detection targets. These factors make the task of oil spill and ship detection more difficult, thus leading to a low accuracy when traditional methods are used, as discussed in this section. However, CNN is invariant to these kind of transformations. In addition, by applying a data augmentation process, we increase the robustness against these kinds of transformations. In this paper, we propose a new method based on CNNs for the detection not only of ships and oil spills but also other targets such as coasts. This is done by using SLAR images digitized from scattering data as mentioned earlier. The proposed method applies a two-stage CNN in order to overcome the limitations of other approaches when they are used with SLAR rather than SAR images.

III. Method

The proposed methodology is described in this section. First, we design a multiclass CNN with which to obtain baseline results. A two-stage CNN that performs a coarse detection and a refined pixelwise classification is then proposed in order to improve accuracy.

Various considerations were taken into account when designing the different networks. As recommended in [27], we have included *dropout* layers so as to reduce overfitting. This technique randomly drops units along with their connections throughout the training phase. *Max Pooling* [28] layers have additionally been used in order to reduce both computation time and the number of parameters required for the next layers while controlling overfitting and providing translation invariance. This filter is a form of nonlinear downsampling that partitions the input image into a set of nonoverlapping rectangles and yields the maximum value for each of these subregions. We have also added *Batch Normalization* [29] layers, since they help speed up training and improve the overall success rate. This technique enables a normalization

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I

DESCRIPTION OF THE FOUR-CLASS NETWORK AND THE TWO-STAGE CNN ARCHITECTURE. CONV(F × K × K) DENOTES A CONVOLUTIONAL LAYER WITH F FILTERS AND A KERNEL OF $K \times K$, MAXPOOL ($K \times K$) A MAX POOLING FILTER WITH A KERNEL OF $K \times K$, AND *FC(n)* AN FC LAYER WITH n NEURONS

4-class	Conv(64x3x3)
	MaxPool(2x2)
		Conv(32x3x3)
		MaxPool(2x2)
			Conv(32x3x3)
			MaxPool(2x2)
				Conv(64x3x3)	FC(128)	FC(4)	

Two-stage	CNN specialized in the detection of ships						

	Coarse	Conv(32x3x3)
		MaxPool(2x2)
			Conv(16x3x3)
			MaxPool(2x2)
				Conv(16x3x3)	FC(128)	FC(1)	

	Pixel	Conv(32x3x3)
		MaxPool(2x2)
			Conv(16x3x3)
			MaxPool(2x2)
				FC(16)	FC(1)		

	CNN specialized in the detection of spills						

	Coarse	Conv(16x3x3)
		MaxPool(2x2)
			Conv(32x3x3)
			MaxPool(2x2)
				Conv(64x3x3)
				MaxPool(2x2)
					Conv(16x3x3)	FC(128)	FC(1)

	Pixel	Conv(32x3x3)
		MaxPool(2x2)
			Conv(16x3x3)
			MaxPool(2x2)
				Conv(16x3x3)	FC(32)	FC(1)	

	CNN specialized in the detection of coasts						

	Coarse	Conv(16x3x3)
		MaxPool(2x2)
			Conv(32x3x3)
			MaxPool(2x2)
				Conv(64x3x3)
				MaxPool(2x2)
					Conv(16x3x3)	FC(128)	FC(1)

	Pixel	Conv(32x3x3)
		MaxPool(2x2)
			Conv(16x3x3)
			MaxPool(2x2)
				Conv(16x3x3)	FC(32)	FC(1)	

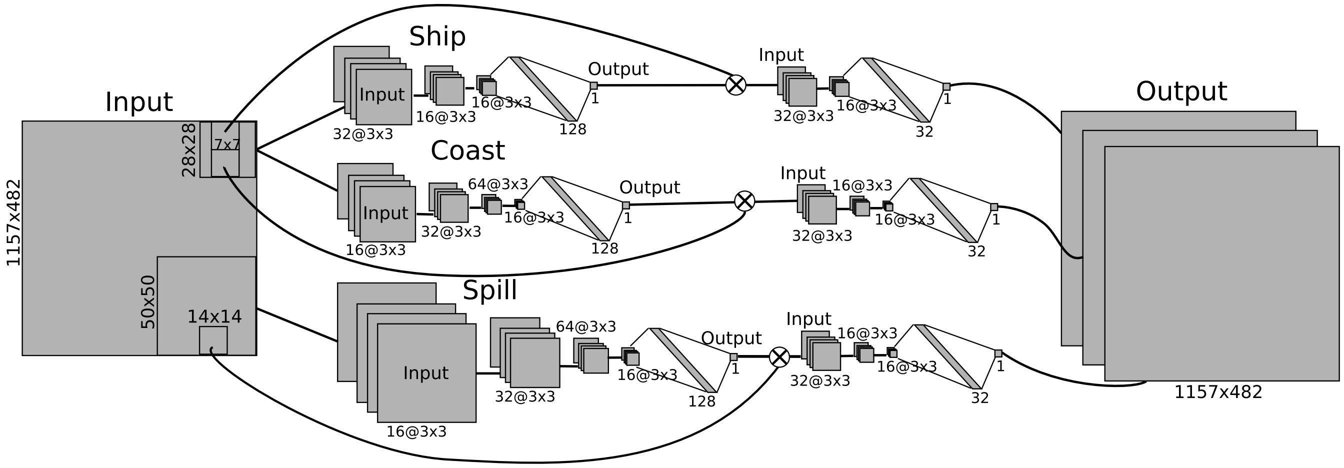


Fig. 1. Layout of the two-stage CNN showing the three different pairs of CNNs to classify ships, coasts, and oil spills.

process of the weights learned by the different layers after each training minibatch. Finally, we have chosen *rectified linear unit* (ReLU) [30] as the activation function, since it is computationally efficient and enhances the gradient propagation throughout the training phase, thus avoiding vanishing and exploding gradient problems.

A. Four-Class CNN

In order to test the benefits of the proposed approach, a single CNN that performs a multiclass classification has been designed. This baseline CNN performs a pixelwise classification using 28 × 28 pixel windows and outputs a one hot vector representing the class of the central pixel, that can be either water, oil spill, coast, or ship. We define the central pixel of a $n \times n$ window as the pixel located at position $i = j = \lceil n/2 \rceil$.

This network contains four convolutions and three max pooling layers. The first convolution layer has 64 filters, the second and the third 32, and the fourth 64. The max pooling layer uses a 2 × 2 kernel. Two fully connected layers are stacked at the top of the network, one with 128 neurons and the last with four. We incorporated dropout in all the layers and used ReLU as a activation function for the entire network, except for the last layer which uses a softmax to return the final prediction. The complete configuration of this network is detailed in Table I. For clarity, the dropout layers are not shown in Table I.

The parameters and layers of this network have been tuned to get a high accuracy. Details of this process can be found in Section IV-D and the results for this network are shown in Section IV-E.

B. Two-Stage CNN Architecture

As a more accurate alternative, we propose the use of a two-stage CNN in order to perform the classification of water, spills, coasts, and ships, employing the one-vs-rest multiclass strategy to overcome the limited amount of training data. Fig. 1 shows the outline of the proposed architecture. The main idea is to create pairs of specialized networks to classify each class.

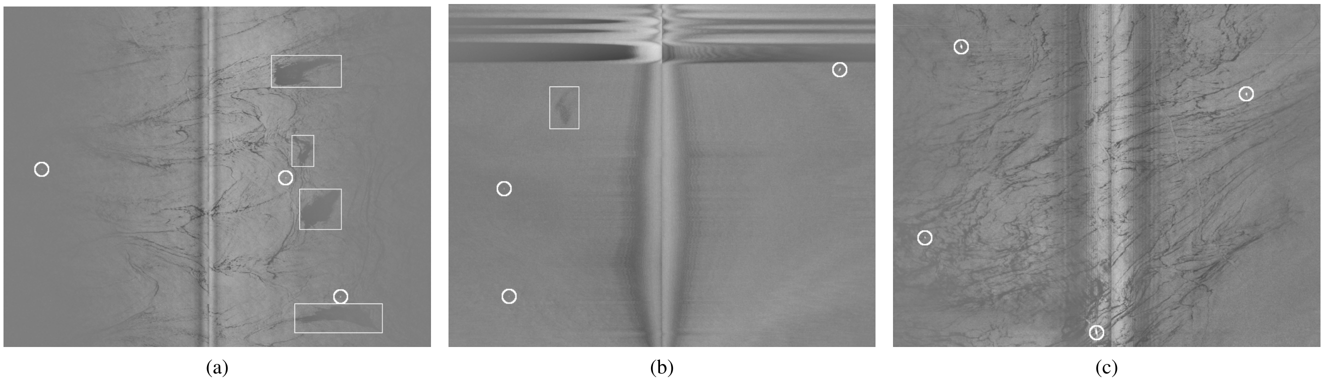


Fig. 2. Examples of SLAR images from our dataset showing oil spills (marked with a bounding box), ships (marked with circles), look-alikes (elongated shapes in figures (a) and (c)), the noise produced by the sensor (the central vertical band that appears in all the images) and the aircraft maneuvers (the horizontal bands that appear in the upper part of figure (b)).

The first network of each pair performs a coarse detection of that class, and a second network performs a more precise detection at a pixel level. The second network is only activated when the first network detects the presence of the target class in the input image. This operation is denoted by the symbol ⊗ in Fig. 1. The aim of this approach is, on the one hand, to improve efficiency because a pixelwise classification is a costly operation and, on the other hand, to improve accuracy. Both goals are accomplished by using a total of six CNNs, two for each class (we assume that the water class is the area not detected by any of the CNNs).

In the first stage, the CNN takes an input image and outputs 1 if there are any pixels from the current class within that region. The second stage uses the images classified as 1 in the previous stage, to perform a pixelwise classification, outputting 1 if the center pixel belongs to that class. We have used an overlapping of half the size of the window to divide the input image into windows. In the second step, when we extract the windows from the previous image, we obtain a border of pixels that cannot be classified. This is solved by instead extracting those windows directly from the original image.

The network topology that shown in Fig. 1 is the configuration that yielded the best results performing a grid-search of its parameters (see Section IV-D). It can be seen that the number of kernels used for the coarse detection of oil spill and coast increases from 16 to 64, whereas the corresponding kernel size for the ship location decreases from 32 to 16. This difference in the subnetwork parameters can be explained for the complexity of the samples to be classified. Oil spills and coasts have more variability than ships, which are small points with high gradients. However, coast and spills are larger and they present a higher variability in shape and gray levels. Therefore, it is necessary a larger amount of filters in the last layers, which are those that analyze the high-level representations.

The last stage of each CNN consists of a fully connected (FC) layer which collects the output of the convolutional part and performs the final classification. To feed the FC layer with the results of the CNN layers, it is necessary to carry out a flattening operation. This operation transforms the output of the CNN layers into a 1-D feature vector to be used by the FC layer for the final classification.

The output of the two-stage CNN is a combined three-channel image, in which each channel stores the output of each class. Pixels marked as 0 in all the three channels are considered to be water. Since we are interested only in the detection of ships and oil spills, information classified as coast will be used to refine the classification of these two classes.

The complete configuration of these networks is detailed in Table I. We incorporated dropout and batch normalization in all the layers. We also used ReLU as an activation function for the entire network, with the exception of the last layer for which we used a sigmoid because it obtained better experimental results. These details are not indicated in the table for the sake of clarity.

The configuration parameters of the network were experimentally adjusted and guided by the intrinsic characteristics of the targets to be detected.

Ships are shown as a small bright regions in SLAR images, as shown in Fig. 2. The main problem that arises when detecting ships is that other floating objects, such as fish farms or small islands, are also shown as the small bright areas. We have, therefore, used a window size that is larger than the size of the boats in order to take advantage of the information in the context of the image. The two-stage CNN shown in the first rows of Table I is used to detect the presence of a ship. It first searches in a 28 × 28 pixel region and, if the network finds a ships, it then enables the second network by using a 7 × 7 pixel region. In these topologies, we use more filters in the first layer, because the ships are small and more information can thus be obtained in fine detail.

Oil spills are shown as homogeneous dark areas of different sizes, as shown in Fig. 2(a) and (b). In the absence of wind, some sea bottom areas are also represented as dark zones, as occurs with some algae formations. These phenomena are called look-alikes and it is difficult to differentiate them from oil spills as can be observed in Fig. 2(a) and (c). As oil spill areas are larger than ships, we trained the CNN using 50 × 50 pixel regions. In this case, the third layer is that which has more filters because the oil spills are larger.

The coast is shown as a large bright area with some shadows depicting terrain elevations. These are also characterized by a very high contrast gradient at the edges of the coast. It is for this reason that small islets may often be confused with ships.

In this case, the same CNN used to detect spills is employed, but it is trained with different samples and 28 × 28 pixel input images are used.

C. Postprocessing

A postprocessing stage is carried out on the output of the networks in order to improve the results. The coasts and oil spills detected are postprocessed using an opening morphological operation with a square kernel (K) to eliminate the possible small look-alikes.

A second postprocessing stage is then performed to filter out ships or oil spills that are surrounded by a minimum amount (λ) of coast. This error may appear with certain look-alikes in interior areas, or with very large coastal areas whose color similar to that of the sea. As it can be seen in Section E, this postprocessing helps to improve the accuracy.

Algorithm 1 summarizes all the steps followed by the proposed two-stage network. In this algorithm, $\mathcal{I}$ and $\mathcal{O}$ represent the input and output images, respectively, $\mathcal{O}_n$ accesses the channel n of the output image $\mathcal{O}$ (for clarity, we used *ship*, *coast*, and *spill* as channel names), and the lowercase letters r, c, and b are regions of the image. The function $regions(\mathcal{I}, size)$ returns a list of regions with the given *size* from the input image $\mathcal{I}$, the function $blobs(\mathcal{O}_n)$ returns the set of blobs found in the corresponding channel of the output image, and the symbol $\circ$ (in $\mathcal{I} \circ K$) performs the opening morphological operation on the image $\mathcal{I}$ using the square kernel K.

D. Training Stage

Both architectures proposed in this section were trained in the same way. The learning of the weights was performed by means of the stochastic gradient descent [31], with the consideration of adaptive learning rate proposed by Zeiler [32]. The training lasted a maximum of 200 epochs with *early stopping* when the loss did not decrease during 10 epochs.

In the case of the four-class network, the training was performed using a mini-batch size of 16. In the case of the two-stage network, each part of the network was individually trained and then combined in order to create the architecture described. The mini-batch size was set to 16 samples for the networks in the first part of the architecture and 32 samples for those in the second part. Section IV-D on hyperparameters evaluation shows the experimentation performed to find these training values.

IV. Experiments

This section describes the evaluation metrics used to analyze the performance of the proposed methods, the details of the data set used for the evaluation, and the method used to augment the data in the unbalanced classes. The evaluation of the CNN hyperparameters using a grid search process is then described. Finally, we present and analyze the results obtained when considering the different metrics and compare them with other published results.

Algorithm 1 Two-Stage CNN Algorithm

```
$\mathcal{I} \leftarrow$ input image
$\mathcal{O} \leftarrow 0$
foreach $r \in regions(\mathcal{I}, 28)$ do
    if $CoarseCNN_{Ship}(r) = 1$ then
        foreach $c \in regions(r, 7)$ do
            $\mathcal{O}_{ship} \leftarrow PixelCNN_{Ship}(c)$
        end
    end
    if $CoarseCNN_{Coast}(r) = 1$ then
        foreach $c \in regions(r, 7)$ do
            $\mathcal{O}_{coast} \leftarrow PixelCNN_{Coast}(c)$
        end
    end
end
foreach $r \in regions(\mathcal{I}, 50)$ do
    if $CoarseCNN_{Spill}(r) = 1$ then
        foreach $c \in regions(r, 14)$ do
            $\mathcal{O}_{spill} \leftarrow PixelCNN_{Spill}(c)$
        end
    end
end
$\mathcal{O}_{spill} \leftarrow \mathcal{O}_{spill} \circ K$
$\mathcal{O}_{coast} \leftarrow \mathcal{O}_{coast} \circ K$
foreach $b \in blobs(\mathcal{O}_{ship})$ do
    if $|b \cap \mathcal{O}_{coast}| > \lambda$ then
        $\mathcal{O}_{ship} \leftarrow b \cap 0$
    end
end
foreach $b \in blobs(\mathcal{O}_{spill})$ do
    if $|b \cap \mathcal{O}_{coast}| > \lambda$ then
        $\mathcal{O}_{spill} \leftarrow b \cap 0$
    end
end
```

A. Evaluation Metrics

In order to evaluate the performance of the proposed CNN models, four evaluation metrics widely used for this kind of tasks have been chosen: accuracy, precision, recall, and F-measure ($F1$), which can be defined as

$$\begin{aligned}\text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}\\ \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}}\\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}}\\ \text{F-measure} &= \frac{2\cdot\text{TP}}{2\cdot\text{TP}+\text{FN}+\text{FP}}\end{aligned}$$

where true positives (TPs) denote the number of correctly detected targets, true negatives (TN) denote the number of incorrectly detected targets, false negatives (FN) denote the number of nondetected or missed targets, and FPs or false alarms denote the number of incorrectly detected targets.

The ROC curve is also used to display the results. It is computed by plotting the true positive rate (or sensitivity, equivalent to recall) against the false positive rate (equivalent to 1-specificity) at various threshold settings. The area under the curve (AUC) is also calculated using the trapezoidal rule to measure the goodness of discrimination.

In addition, the metric used to quantitatively evaluate the proposed method with which to locate the three classes considered was intersection over union (IoU) [33]. We map each object proposal (op) onto the ground truth (gt) bounding box (BB) with which it has a maximum IoU overlap. A detection is considered to be TP if the area of overlap (a_o) ratio exceeds a certain threshold according to the following equation:

$$a_o = \frac{\text{area}(B_{\text{op}} \cap B_{\text{gt}})}{\text{area}(B_{\text{op}} \cup B_{\text{gt}})} \tag{1}$$

where area($B_{\text{op}} \cap B_{\text{gt}}$) depicts the intersection between the object proposal and the ground truth BB and area($B_{\text{op}} \cup B_{\text{gt}}$) depicts its union.

B. Data Set Configuration

The data set used as input in the experimentation contains a total of 23 SLAR images supplied by the Spanish Maritime Safety and Rescue Agency (SASEMAR) with a resolution of 1157 × 482 pixels. SASEMAR is the public authority responsible for monitoring the exclusive economic zones in Spain and its procedures are based on reports from the European Maritime Safety Agency. The data set samples were captured at an approximate altitude of 4500 feet, with a flight speed of about 200 Kn, and with a wind speed ranging between 6 and 25 Kn. The features of SLAR images depend on the sampling and digitalizing performed by the TERMA-9000 sensor control software. All data are registered in SLAR images as 8-bit integers due to the constraints of the monitoring equipment installed on EADS-CASA CN 235-300. Our CNN architectures use as input the SLAR images as they are generated by the TERMA software.

As ground truth, we have considered a gray scale mask for each SLAR image, delimiting the pixels in each target class with a different gray value. The five classes considered as targets: ship, oil spill, look-alike, coast, and water. It is important to note that this labeling has been performed at pixel level, since the goal is to evaluate both the detection and the precise location of the points belonging to each class. In this way, we can provide relevant information such as the count of the different elements from each class that are present in the image. Location is also necessary to obtain the coordinates of ships performing illegal activities or the position of oil spills. Furthermore, the points belonging to oil spills can be used to detect the size and shape of spills in order to track them.

Fig. 2 shows samples of SLAR images from our data set. They contain several examples of boats, spills, coasts, and look-alikes, along with the noise generated by this sensor, which depends on the aircraft trajectory and navigation maneuvers. Examples of noises caused under the aircraft and by turning maneuvers are shown in Fig. 2(b) in its central and top area, respectively. The instances of ships are marked with a circle in the image, whereas the spills are indicated with a BB. Fig. 2(a) and (c) contains many examples of look-alikes around the central noise, with elongated shapes that are very similar to those of current instances of spills.

TABLE II

STATISTICS OF THE DATA SET: NUMBER OF INSTANCES OF EACH CLASS, PERCENTAGE OF AREA IN PIXELS, AND AVERAGE SIDE SIZE IN PIXELS OF THE SQUARE BB THAT CONTAINS THOSE AREAS

Class	# instances	% of pixels	Avg. BB side in px.
Ship	72	0.01	1.95
Oil Spill	14	0.32	30.72
Look-alike	172	2.16	22.74
Coast	115	4.76	41.27
Water	393	92.76	98.59

The data set used for the experiments contains 72 instances of ships, 14 oil spills, 172 look-alikes, and 115 blobs of coast (Table II). Most of the image pixels (92.76%) correspond to the water class. This fact is corroborated by observing the average size of the samples in each class, considering the side of its square BB.

For the experimentation, the aforementioned data set has been divided into training, validation, and test sets, using 74% for training, 13% for validation, and 13% for evaluation, respectively. These partitions have been made in order to ensure that representative samples of each of the classes are included. Two fixed sets have also been designed in order to assess how the complexity of the samples taken as input may influence the results. Set 1 is simpler as it does not contain look-alikes, whereas Set 2 contains more complex samples, including look-alikes. Due to the variability in our data set samples, it was necessary to manually select the samples for each set, because some SLAR images did not contain ships or only contain noise. For that reason, we tried to balance the training and testing sets.

The pixels of the input images in both sets are normalized between 0 and 1.

The work by Alacid and Gil [34] presents a method with which to detect and remove the aforementioned noise areas which can also be seen in Fig. 2 (right). In this paper, we have used the same method to filter images from our data set in order to eliminate those noise areas before running the proposed approach. Noise is not, therefore, considered in either the training process or in the classification stage.

The size of this data set may seem small for a supervised classifier but it should be noted that SLAR images are digitalized as time series in which each scanning time is represented in the image by a certain number of rows. Each SLAR image can, therefore, be divided into a grid and can thus be dealt with individually. In this paper, SLAR images have been cut into regions of 50 or 28 pixels per side (with an overlapping of 25 and 14 pixels) to train the networks in the first stage, and into regions of 14 or 7 pixels per side (with an overlapping of 1 pixel) in the second stage of our architecture, as discussed in Section III. A total of up to 512 566 samples per image are, therefore, generated overall.

Few samples are obtained when considering larger window sizes. For example, for regions with 50 pixels on a side on which 25 pixels overlap, we obtain around 800 samples per

TABLE III

GRID-SEARCH OF THE HYPERPARAMETERS USING THE VALIDATION PARTITION FOR THE DIFFERENT CNNS. THE F-MEASURE (IN PERCENTAGE) IS SHOWN, HIGHLIGHTING THE BEST RESULTS IN BOLD

		Two-stage network						**4-class network**
		Ship		**Spill**		**Coast**		
Dropout %	**Batch Norm**	**Coarse**	**Pixel**	**Coarse**	**Pixel**	**Coarse**	**Pixel**	**Pixel**
0	**No**	94.75	75.38	80.72	45.53	**100.00**	54.88	30.67
	Yes	94.55	78.04	**84.25**	40.59	100.00	59.47	27.44
25	**No**	**97.25**	80.94	65.93	42.37	99.38	49.31	35.18
	Yes	95.78	82.10	83.49	**45.66**	99.38	57.11	21.16
50	**No**	93.56	83.72	70.16	42.94	95.01	60.59	**35.91**
	Yes	96.83	**86.70**	78.27	20.66	94.30	65.04	25.51
75	**No**	29.88	77.87	33.33	43.57	86.83	**74.70**	29.77
	Yes	29.88	81.21	33.33	0.25	95.60	69.98	25.65

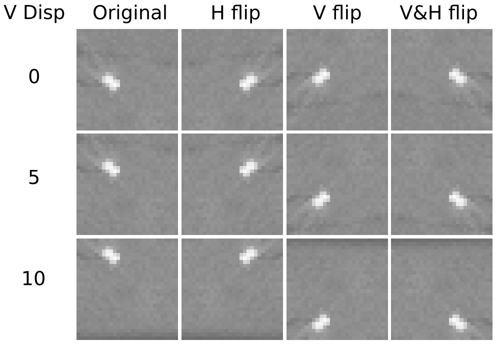


Fig. 3. Data augmentation process showing the displacements and flips. Horizontal displacement is not shown.

image. Moreover, in our data set, more than 90% of the pixels are water, and there are fewer examples of ships and oil spills. We consequently have a very unbalanced data set. This issue has been solved by relying on data augmentation techniques, as described in Section IV-C.

C. Data Augmentation

Data augmentation is applied in order to artificially increase the size of the training set, as indicated in [28] and [35]. As the experimental results show, this process systematically improves the accuracy. In order to augment the data of the unbalanced classes, we focus a window around the area of interest in which the samples of the target class were found, and we then extract samples by moving the window around them and performing horizontal and vertical flips. Fig. 3 presents an example of the data augmentation process, in which an original image and its transformations to obtain synthetic samples are shown.

The data augmentation process is applied to the samples of each class until the size of the class with most samples is attained, thus balancing the number of samples per class and signifying that all the classes contain the same amount of samples after data augmentation. Algorithm 2 describes the process followed to achieve data augmentation. In this algorithm, we denote $X = \{x_1, x_2, \ldots, x_n\}$ as the whole set of samples to be augmented, $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ as the set containing all the classes of X, and X^c as a subset of X containing only the samples of class c. Function $g_{\text{next}}(X)$ returns the next sample in the set X to be augmented and $f_{\text{aug}}(x)$ performs the data augmentation process for the sample x following the procedure described earlier.

Algorithm 2 Data Augmentation

```
$max_c := \arg\max_{c \in \mathcal{C}} |X^c|$
foreach $c \in \mathcal{C}$ do
    while $|X^c| < |X^{max_c}|$ do
        $s := g_{next}(X^c)$
        $X^c := X^c \cup f_{aug}(s)$
    end
end
```

Augmentation was applied to the different training sets with the window sizes considered. However, as the second step of the two-stage architecture works at the pixel level and with a small margin of context, augmentation through translation was not applied in this stage as it would not provide new data.

D. Hyperparameters Evaluation

In order to select the best networks configuration, we have performed a *grid search* [36] on the four-class network and the two-stage CNN. First, we evaluated different window sizes between 7 and 100 pixels per side for the input of the networks (see Fig. 4), eventually selecting the best parameters obtained in each case (previously mentioned in Sections III-A and III-B).

Table III shows the best hyperparameters found after the grid search (marked in bold). It will be noted that the most adequate values of batch normalization and dropout differ in each network. The best parameters in each case were used for the subsequent experiments.

Fig. 5 shows the influence of the mini-batch size (ranging from 8 to 128) on the training process of the four-class network and also on the two parts of the two-stage network. As it can be seen in these results, the best mini-batch size is 16 samples for the four-class network and the coarse networks, and 32 samples for the pixel networks. This difference may be due to the input window size, since for the first part of the two-stage network a larger window size is used, and therefore

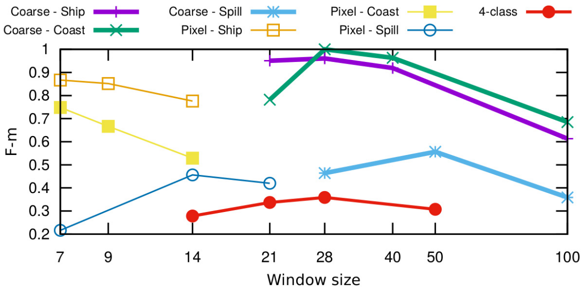


Fig. 4. Influence of the window size on the different networks evaluated.

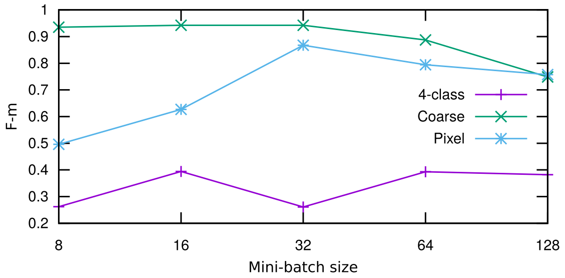


Fig. 5. Influence of the mini-batch size for the different networks evaluated.

it has fewer training samples. The same is true for the four-class network, which uses a window size of 28 × 28 pixel.

We additionally performed an experiment with which to determine the optimum size for the opening kernel used in the postprocessing stage for spill and coast detection. We did this by evaluating sizes from 2 × 2 to 10 × 10 (see Section F), finding that the optimal size to be used is a 7 × 7 square kernel.

E. Results

After the CNNs were trained using the best values of dropout and batch normalization indicated in Table III, we assessed the performance of the proposed methods. This was done by using the two test sets described in Section IV-B without using look-alikes (Set 1) or considering them (Set 2). None of the images in the test sets were used during the training process. One of the images included in Set 2 was particularly difficult because it was full of look-alikes. We ran the four-class and the two-stage networks with these two test sets and then assessed the performance by comparing the results with the ground truth, pixel by pixel.

Table IV shows the results obtained at pixel level by both approximations, with and without postprocessing, for the two test sets. These results are shown for each class using the metrics described in Section IV-A. In the postprocessing stage, a morphological opening filter with a 7 × 7 square kernel is applied to the output. Finally, those pixels classified as spill or ship are processed using a sliding window of 21 × 21 pixel, filtering out those that are surrounded by coast pixels in a fraction greater than 0.3.

As it can be seen in the Table IV, the two-stage approach improves the F-measure result obtained by the four-class network in all the configurations evaluated (with and without postprocessing). In the cases of both ship and coast, the two-stage method is more than 25 points better than the four-class network. This improvement is slightly lower in the case of spills owing to the presence of look-alikes identified as oil spills. Upon comparing Sets 1 and 2, it will be observed that the classifier obtains worse results when considering images with look-alikes. The recall of oil spills after postprocessing was 86.7% and 72.7% for Sets 1 and 2, respectively, showing that our approach is capable of detecting spills but it can be confused by look-alikes, as indicated by the precision measure which is around 65.05% and 52.23%, respectively. A detailed analysis of the results concerning the look-alikes is provided in Table V. It is also observed that how the accuracy of the proposed method improves by an average of 3.6, and in one case by almost 9, after applying the postprocessing. For the two sets and the three classes considered, the two-stage CNN with postprocessing obtains an overall success rate of over 99% and a high recall value.

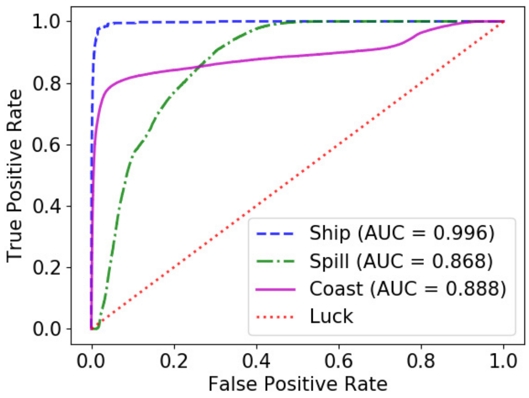


Fig. 6. ROC curves of the proposed two-stage network for ships, oil spills, and coast detection (without including the postprocessing step).

A possible explanation why the two-stage model outperforms the F-measure of the multiclass topology is that the number of samples is relatively small for the four-class network, which requires more parameters to correctly identify 4 different classes. The two-stage version has small networks specialized in each individual class, requiring less training data, and the first level networks are used to improve the results of the second-level models, which are specialized to detect a single class.

Fig. 6 shows the ROC curves calculated for ship, oil spill, and coast outputs of the two-stage network, without the postprocessing step, and applying different threshold levels in order to see how it affects the sensitivity and the specificity of the model. The higher the AUC index the better the discrimination performed by the method. Specifically, and using the traditional academic point system, the AUC for spills and coast curves is in the 0.8–0.9 range, showing a good accuracy, and the AUC for the ship curve is in the 0.9–1 range, so it has an excellent accuracy.

The results for the spill class shown in Table IV were obtained by considering look-alikes as water. However, this type of formations, usually algae or corals, are very difficult to differentiate from true oil spills, even for the human eye. For this reason, the result obtained improves significantly if we consider this class as an oil spill, as shown in Table V

TABLE IV

Comparison of the Results (in Percentages) Obtained at the Pixel Level by Both Approaches, With and Without Post-Processing, and for the Different Test Sets, Indicating the Result by Class

			Set 1			Set 2		
			Ship	**Spill**	**Coast**	**Ship**	**Spill**	**Coast**
Without post-proc.	**4-class**	**Accuracy**	99.55	97.66	95.44	99.48	95.31	93.50
		Precision	51.65	55.01	57.44	50.85	50.95	59.88
		Recall	95.35	79.91	65.83	96.93	70.28	86.93
		F-measure	53.08	58.15	59.87	51.54	50.74	64.41
	Two-stage	**Accuracy**	99.99	98.10	99.05	99.98	95.14	99.42
		Precision	82.02	57.69	95.26	66.55	51.40	91.04
		Recall	91.28	91.13	76.63	95.30	81.54	94.42
		F-measure	86.06	62.56	83.45	74.24	51.52	92.66
With post-proc.	**4-class**	**Accuracy**	99.65	99.03	95.44	99.67	97.95	93.50
		Precision	51.65	58.98	57.44	51.28	51.03	59.88
		Recall	85.22	70.07	65.83	94.52	59.38	86.86
		F-measure	53.07	62.35	59.87	52.40	51.50	64.41
	Two-stage	**Accuracy**	99.99	99.21	99.07	99.99	97.82	99.48
		Precision	87.44	65.05	96.12	73.85	52.23	92.46
		Recall	90.35	86.75	76.61	95.31	72.70	94.08
		F-measure	**88.84**	**71.26**	**83.66**	**81.25**	**53.62**	**93.26**

TABLE V

Comparison of the Results Obtained When Considering and Not Considering Look-Alikes as Oils Spills. For This Experiment, Spill Detection was Only Performed Using Set 2, Because Set 1 Does Not Contain Look-Alikes

Method	Look-alikes	Accuracy	Precision	Recall	F-measure
4-class	**No**	97.95	51.03	59.38	51.50
	Yes	96.64	61.15	60.97	**61.06**
Two-stage	**No**	97.82	52.23	72.70	53.62
	Yes	97.57	72.36	74.40	**73.33**

TABLE VI

Results Obtained Using the IoU Metric and Different Threshold Values

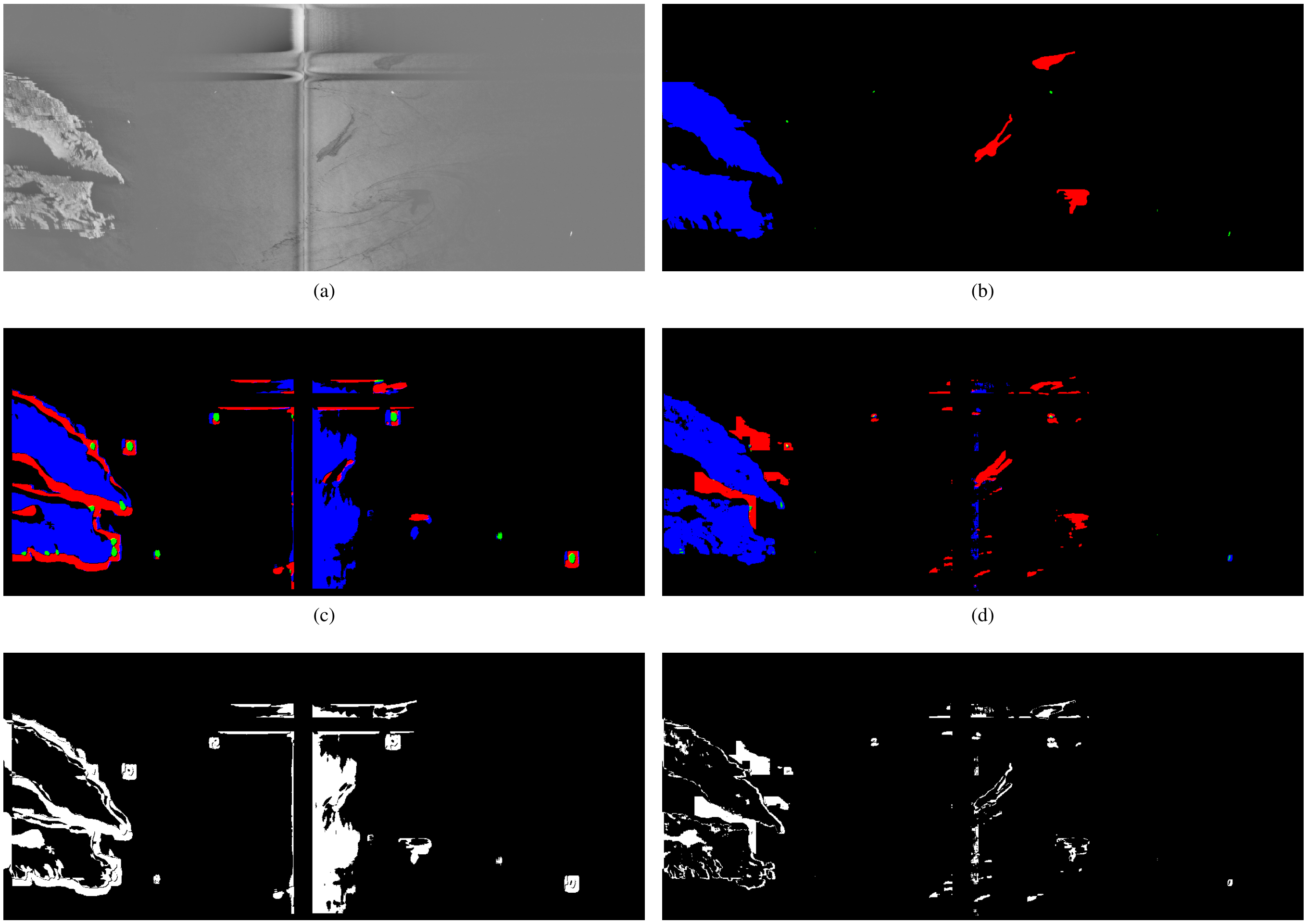
		Set 1			Set 2		
Method	**Threshold**	**Ship**	**Spill**	**Coast**	**Ship**	**Spill**	**Coast**
4-class	0.1	66.67	33.33	66.67	28.57	100.00	66.67
	0.3	11.11	33.33	66.67	7.14	33.33	66.67
	0.5	0.00	0.00	0.00	7.14	0.00	66.67
Two-stage	0.1	94.44	100.00	100.00	92.86	100.00	100.00
	0.3	88.89	66.67	66.67	92.86	66.67	100.00
	0.5	77.78	33.33	66.67	92.86	33.33	66.67

(row “Yes”). In order to perform this experiment, we used the same networks with the same topology and weights, since the number of classes remains the same (look-alikes become part of the Spill class). We thus recommend including them as spills, since in these cases it is better to provide a FP and request a human operator to validate it.

Fig. 7 provides a graphical representation of the results of both approaches. Fig. 7(a) and (b) shows the original SLAR image and its ground truth, respectively. In the ground truth, the green areas depict ships, the blue areas the coast, and the red areas the spills. The results of the four-class and two-stage CNNs after the postprocessing step are shown in Fig. 7(c) and (d), respectively. Fig. 7(e) and (f) shows the respective wrongly classified pixels for each approach after postprocessing. These figures help visualize the accuracy of the two models evaluated and to understand where the errors are caused for each target class. As it can be seen, in both approaches the main mistake made is the confusion between the spill and the coast classes, since both have similar shapes and similar gradient changes on the borders of the regions. What is more, some look-alikes are misclassified around the center of the image.

According to Fig. 7, the proposed topology correctly identifies the different classes. However, when measuring the error that occurs at pixel level, unfair results are obtained (see Table IV), since, although it might appear to make many errors, the two-stage network correctly detects the presence and position of the objects. For this reason, and in order to assess the localization performance, we used the IoU metric [see (1)]. We mapped each object proposal onto its greatest IoU in the ground truth, considering a detection to be TP if the IoU was greater than a certain threshold λ. We evaluated three common values for this threshold λ, 0.5, 0.3, and 0.1, that were adequate for the task at hand considering the reduced size of the ships. The obtained results are shown in Table VI. In this case, the IoU metric shows that the two-stage approach is much more accurate than the four-class CNN in the two test sets and for the three classes considered. The four-class CNN tends to detect larger areas than the actual size of the targets, and the IoU obtained when mapping the detection onto the ground truth is, therefore, usually lower than the threshold. Note that when using $\lambda = 0.1$, our method obtains a 100% of IoU for the spill and coast classes, and 92.86% for ships, respectively.

Table VII shows an analysis of the contribution to the F-measure by each part of the two-stage network. As can be seen, the highest contribution is due to the first part of the network (Coarse), although the second part (Pixel) improves up to 36 points the classification of ships in Set 1. Although the postprocessing stage has a smaller contribution, in some



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 7. Results of processing an SLAR input image and its validation ground truth. Green areas: ships. Blue areas: coast. Red areas: spills. (a) SLAR image. (b) Ground truth. (c) Four-class CNN output after postprocessing. (d) Two-stage CNN output after postprocessing. (e) Four-class CNN error after postprocessing. (f) Two-stage CNN error after postprocessing.

TABLE VII

ANALYSIS OF THE CONTRIBUTION TO THE FINAL RESULT BY EACH PART OF THE TWO-STAGE NETWORK IN TERMS OF F-MEASURE

		Set 1			Set 2		
		Ship	**Spill**	**Coast**	**Ship**	**Spill**	**Coast**
Incremental results	**Coarse**	+49.83	+39.22	+82.85	+50.07	+47.01	+68.67
	Pixel	+36.23	+23.34	+0.6	+24.17	+4.51	+23.99
	Post-proc.	+2.78	+8.7	+0.21	+7.01	+2.1	+0.6
	Total	88.84	71.26	83.66	81.25	53.62	93.26

cases (as for spill detection in Set 1), it increases the final F-measure by about nine points.

F. Comparison

In this section, we compare the proposed two-stage architecture with other methods from Oprea and Alacid [9] for the detection (without localization) of oil spills in SLAR images. Note that unlike the paper cited above, which only performs detection, our method performs both detection and localization. In order to evaluate both approaches under the same conditions, we trained the network with the same data set of SLAR images, using the same training and test partitions used in [9] (which is different from that used in this paper and described in Section IV-B), and with the same metrics (Accuracy, Precision, and Recall).

In [9], different types of classifiers were evaluated: MLP networks, RNN [37], long short-term memory (LSTM) networks [38], bidirectional RNNs (BRNN) [39], and support vector classifiers (SVCs) with a *radial basis function* kernel. Each classifier was evaluated using different configurations of parameters, including the number of neurons on the hidden layers, the number of hidden layers, activation functions, batch size, dropout, and time steps length considered in the RNN.

Table VIII shows a ranking of the best results obtained with each classifier. The details regarding the implementation and parameters used in these methods can be found in the work by Oprea and Alacid [9]. As it can be seen, the proposed two-stage networks (with a kernel size of 7 × 7) outperform the accuracy and precision of all the previous methods evaluated. We have also included the result obtained with a smaller kernel (5 × 5) which gets a higher recall value. As can be seen

TABLE VIII

RANKING OF METHODS FOR DETECTION OF SPILLS: OUR TWO-STAGE CNN APPROACH VERSUS DIFFERENT CLASSIFIERS PRESENTED IN [9]

Model	**Accuracy**	**Precision**	**Recall**
Two-stage CNN ks. 7x7	**98.34**	100.00	88.06
Two-stage CNN ks. 5x5	93.98	70.21	98.51
BRNN 1	97.00	96.32	100.00
BRNN 2	97.00	97.01	99.23
MLP 1	96.82	97.25	98.73
MLP 2	96.62	97.24	98.47
LSTM 1	96.22	96.28	98.98
MLP 3	96.22	96.52	98.73
LSTM 2	96.21	97.22	97.96
BRNN 3	96.20	95.83	99.49
MLP 4	96.02	96.06	98.98
MLP 5	95.63	96.73	97.71
SVC	95.03	94.24	99.74

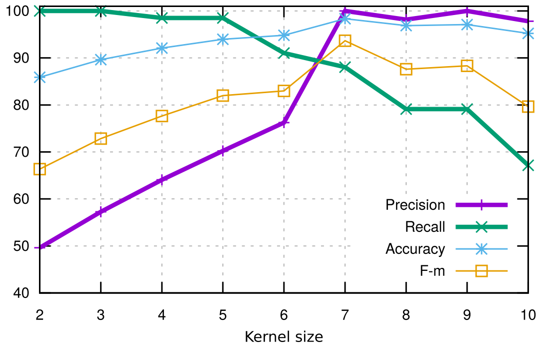


Fig. 8. Evaluation of the kernel size used in the postprocess stage.

in Fig. 8, the larger the kernel size, the higher precision and accuracy is obtained, as more FPs are removed. For smaller kernel sizes, the recall increases but the precision decreases. This allows us to adjust this parameter according to our priority.

G. Runtime Analysis

From a practical point of view, it should be noted that the proposed architecture can be used in applications that require a fast response. For the detection and localization of the three classes considered, the proposed network takes an average of 7.81 ms to process a sweep of the SLAR sensor, whereas the four-class network takes 106.09 ms, without considering the network loading time. In the case of the Two-stage networks, the time is distributed as follows: 5.92 ms for the first part of the network and 1.89 ms for the second part, which execution depends on the result of the first one. These runtimes were obtained using a GeForce GTX 1070 GPU.

The two-stage networks are computationally more efficient than the multiclass version for the following reasons: 1) the calculation of the different networks can be parallelized; 2) the networks from the second stage are only used when a ship, coast, or spill is detected, and given that most pixels belong to sea (92.76% according to Table II), mostly they are not executed; and 3) the networks require less parameters as they are specialized to detect a single class.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a two-stage CNN using a one-vs-rest strategy to perform a coarse detection of large portions of the image, followed by a refined pixelwise classification of the areas detected in the first step. The main contributions of this paper are summarized as follows.

1) To the best of our knowledge, this is the first approach using CNNs for SLAR imagery. These type of images are composed of two radar signals that come from different antennas, so it is also worth mentioning the use of a composite input into CNNs.
2) A two-stage CNN architecture to classify oil spills, ships, and coast is presented. The first stage performs a coarse detection and the second one is used to get the exact pixels of the classes. The proposed setup improves the efficiency and increases the accuracy with respect to a standard multiclass CNN. It is also scalable, as it is easy to add more specialized networks for new targets without requiring to change the other networks.
3) An extensive experimentation is performed in order to adjust both the hyperparameters and the network topologies for this particular problem. A comparison with the state-of-the-art methods for oil spill detection in SLAR imagery shows that the proposed approach outperforms the results of previous works.

When compared with a single pixelwise four-class CNN, the F-measure is significantly improved in the case of the ship and coast classes. There is, however, not much improvement in the case of spill detection, as both approaches are confused by look-alikes. The two-stage CNN obtains an overall accuracy of over 99% and high recall values for the two sets and the three classes considered.

The proposed two-stage architecture can be considered more complex than the four-class CNN. However, the individual configuration of each subnetwork of the two-stage network is less complex than the four-class CNN. This fact allows getting a better result with few samples and, on the other hand, to improve the efficiency, because the second stage of the network is only executed if the first stage detects the presence of the target. For that reason, both time performance and accuracy were higher with the two-stage based classifier. The combination using one-vs-rest strategy improves the F-measure of the classifier, because it provides specialized networks for each class allowing to overcome the limited amount of samples in the data set. In addition, the two-stage architecture improves the time performance, because the amount of pixels processed by the fine network is reduced.

Experiments using the IoU metric show excellent results for the detection and localization of the three classes. In this case, the two-stage approach performs the detection and localization of ships much more accurately than the four-class approach. This difference is mainly owing to the four-class CNN detected areas, which are much larger than the actual area of the ship, thus making the overlap area lower than the threshold.

The proposed method has been compared with the other state-of-the-art approaches for the same task using SLAR images and outperforms them as regards accuracy. It should also be noted that our method performs both the detection and the localization of the four classes considered using data from an SLAR sensor as a basis, and these are noisier and have less resolution than SAR sensors.

As future work, we would like to combine the information obtained by the SLAR sensors with that of others such as the visible spectrum. Information fusion could solve the limitations of the different sensors, such as the detection of certain types of materials by means of SLAR, or the problems that visible spectrum sensors have at night or in conditions of low visibility.

References

[1] I. Leifer *et al.*, "State of the art satellite and airborne marine oil spill remote sensing: Application to the BP *Deepwater Horizon* oil spill," *Remote Sens. Environ.*, vol. 124, pp. 185–209, Sep. 2012.

[2] S. Skrunes, C. Brekke, and T. Eltoft, "Characterization of marine surface slicks by RADARSAT-2 multipolarization features," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5302–5319, Sep. 2014.

[3] S. Skrunes, C. Brekke, T. Eltoft, and V. Kudryavtsev, "Comparing near-coincident C- and X-band SAR acquisitions of marine oil spills," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 1958–1975, Apr. 2015.

[4] C. Brekke and A. H. S. Solberg, "Oil spill detection by satellite remote sensing," *Remote Sens. Environ.*, vol. 95, no. 1, pp. 1–13, Mar. 2005.

[5] C. Brekke and A. H. S. Solberg, "Classifiers and confidence estimation for oil spill detection in ENVISAT ASAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 1, pp. 65–69, Jan. 2008.

[6] D. J. Crisp, "The state-of-the-art in ship detection in synthetic aperture radar imagery," Dept. Defense, Austral. Government, Tech. Rep., 2004, p. 115.

[7] H. Greidanus and N. Kourti, "Findings of the declims project—Detection and classification of marine traffic from space," in *Proc. SEASAR Adv. SAR Oceanograp. ENVISAT ERS Missions*, 2006.

[8] A. Marino, M. J. Sanjuan-Ferrer, I. Hajnsek, and K. Ouchi, "Ship detection with spectral analysis of synthetic aperture radar: A comparison of new and well-known algorithms," *Remote Sens.*, vol. 7, no. 5, pp. 5416–5439, 2015.

[9] S.-O. Oprea and B. Alacid, "Candidate oil spill detection in SLAR data—A recurrent neural network-based approach," in *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods (ICPRAM)*, vol. 1. 2017, pp. 372–377.

[10] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning—A new frontier in artificial intelligence research," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, Nov. 2010.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[12] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. (Nov. 2013). "Rich feature hierarchies for accurate object detection and semantic segmentation." [Online]. Available: https://arxiv.org/abs/1311.2524

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. (Mar. 2017). "Mask R-CNN." [Online]. Available: https://arxiv.org/abs/1703.06870

[15] K. Topouzelis, V. Karathanassi, P. Pavlakis, and D. Rokos, "Detection and discrimination between oil spills and look-alike phenomena through neural networks," *ISPRS J. Photogram. Remote Sens.*, vol. 62, no. 4, pp. 264–270, 2007.

[16] L. Xu, J. Li, and A. Brenning, "A comparative study of different classification techniques for marine oil spill identification using RADARSAT-1 imagery," *Remote Sens. Environ.*, vol. 141, pp. 14–23, Feb. 2014.

[17] M. Marghany, "Utilization of a genetic algorithm for the automatic detection of oil spill from RADARSAT-2 SAR satellite data," *Marine Pollution Bull.*, vol. 89, nos. 1–2, pp. 20–29, 2014.

[18] D. Mera, J. M. Cotos, J. Varela-Pet, P. G. Rodríguez, and A. Caro, "Automatic decision support system based on SAR data for oil spill detection," *Comput. Geosci.*, vol. 72, pp. 184–191, Nov. 2014.

[19] S. Singha, M. Vespe, and O. Trieschmann, "Automatic synthetic aperture radar based oil spill detection and performance estimation via a semi-automatic operational service benchmark," *Marine Pollution Bull.*, vol. 73, no. 1, pp. 199–209, Aug. 2013.

[20] Y. Guo and H. Z. Zhang, "Oil spill detection using synthetic aperture radar images and feature selection in shape space," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 30, pp. 146–157, Aug. 2014.

[21] C. P. Schwegmann, W. Kleynhans, and B. P. Salmon, "Synthetic aperture radar ship detection using Haar-like features," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 2, pp. 154–158, Feb. 2017.

[22] C. Wang, F. Bi, W. Zhang, and L. Chen, "An intensity-space domain CFAR method for ship detection in HR SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 4, pp. 529–533, Apr. 2017.

[23] X. Wang and C. Chen, "Ship detection for complex background SAR images based on a multiscale variance weighted image entropy method," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 2, pp. 184–187, Feb. 2017.

[24] L. Zhai, Y. Li, and Y. Su, "Inshore ship detection via saliency and context information in high-resolution SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1870–1874, Dec. 2016.

[25] S. Wang, M. Wang, S. Yang, and L. Jiao, "New hierarchical saliency filtering for fast ship detection in high-resolution SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 1, pp. 351–362, Jan. 2017.

[26] K. Ji, X. Leng, Q. Fan, S. Zhou, and H. Zou, "An land masking algorithm for ship detection in SAR images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 925–928.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1–9.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. JMLR W&CP*, vol. 37. 2015, pp. 448–456.

[30] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *J. Mach. Learn. Res.*, vol. 15, no. 4, pp. 315–323, 2011.

[31] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010, pp. 177–186.

[32] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *CoRR*, Dec. 2012.

[33] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.

[34] B. Alacid and P. Gil, "An approach for SLAR images denoising based on removing regions with low visual quality for oil spill detection," *Proc. SPIE*, vol. 10004, pp. 1000419-1–1000419-10, Oct. 2016. [Online]. Available: https://doi.org/10.1117/12.2239257

[35] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.

[36] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[37] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

Mario Nieto-Hidalgo received the bachelor's degree in computer engineering and the master's and Ph.D. degrees in computer science from the University of Alicante, Alicante, Spain, in 2011, 2012, and 2017, respectively

He was with the Department of Computing Technology and the University Institute of Research, University of Alicante, where he was involved in computer vision, gait analysis, ambient assisted living, machine learning, and methodology of science.

Antonio-Javier Gallego received the B.S., M.S. degrees in computer science and the Ph.D. degree in computer science and Artificial Intelligence from the University of Alicante, Alicante, Spain, in 2004 and 2012, respectively.

He is currently an Assistant Professor with the Department of Software and Computing Systems, University of Alicante, Spain. He has authored or coauthored over 25 works in international journals, conferences, books, and book chapters. His research interests include deep learning, pattern recognition, and computer vision.

Pablo Gil (M'12–SM'14) received the B.Sc. degree in computer science engineering and the Ph.D. degree from the University of Alicante, Alicante, Spain, in 1999 and 2008, respectively.

He is currently an Associate Professor with the Department of Physics, Systems Engineering, and Signal Theory, University of Alicante. He has also been a Researcher in over 15 research and development projects funded by Spanish Government agencies and private companies. He has authored or co-authored more than 100 works in international journals, conferences, and book chapters on these topics. His research interests include object recognition, computer vision, 3-D vision, and robotics.

Dr. Gil is a member of the Spanish Automatic Committee of IFAC and a senior member of the IEEE Robotics and Education Societies and the IEEE Sensor Council.

Antonio Pertusa received the B.Sc. degree in computer science engineering and the Ph.D degree from the University of Alicante, Alicante, Spain.

He is currently a full-time Lecturer with the Department of Software and Computer Systems, University of Alicante, where he is also an Assistant Manager with the University Institute of Computing Research. He has authored or coauthored more than 30 works in international journals, conferences, and book chapters, among others. His research interests include deep learning, computer vision, signal processing, and pattern recognition methods.

Dr. Pertusa is a member of the executive committee of the Spanish AERFAI Association.