Applied Soft Computing Journal 84 (2019) 105716

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Oil spill segmentation in SAR images using convolutional neural networks. A comparative analysis with clustering and logistic regression algorithms.

Diego Cantorna [a], Carlos Dafonte [a,*], Alfonso Iglesias [a], Bernardino Arcay [a]

[a] CITIC - Department of Computer Science, University of A Coruña, Spain

ARTICLE INFO

Article history:
Received 31 August 2018
Received in revised form 27 May 2019
Accepted 13 August 2019
Available online 26 August 2019

Keywords:
SAR
Remote sensing
Oil spill
Image segmentation
Deep learning

ABSTRACT

Synthetic aperture radar (SAR) images are a valuable source of information for the detection of marine oil spills. For their effective analysis, it is important to have segmentation algorithms that can delimit possible oil spill areas. This article addresses the application of clustering, logistic regression and convolutional neural network algorithms for the detection of oil spills in Envisat and Sentinel-1 satellite images. Large oil spills do not occur frequently so that the identification of a pixel as oil is relatively uncommon. Metrics based on Precision–Recall curves have been employed because they are useful for problems with an imbalance in the number of samples from the classes. Although logistic regression and clustering algorithms can be considered useful for oil spill segmentation, the combination of convolutional techniques and neural networks achieves the best results with low computing time. A convolutional neural network has been integrated into a decision support system in order to facilitate decision-making and data analysis of possible oil spill events.

1. Introduction

Oil spills that occur as a result of accidents, intentional dumping, or natural causes, are harmful to marine ecosystems and sectors like fishing or tourism [1]. It is important to have tools for the detection and analysis of this problem.

Diverse studies have highlighted the usefulness of Synthetic Aperture Radar images (SAR) obtained by satellites for the detection of oils spills [2,3]. A recent study on Deepwater Horizon spill analysed several SAR images on shoreline regions and concluded that most days oil spills are detectable on these images [4]. These sensors can scan wide areas, are not affected by clouds and do not depend on solar illumination so that they remain operative at night.

To obtain the images, the satellite emits microwaves and subsequently detects the reflected waves. Oil spills affect the marine surface so that the energy reflected towards the satellite is less and appears in images as dark areas. The image analysis is further complicated with phenomena, generally called look-alikes, which have a similar effect and occur in areas with low wind or algae [5,6].

SAR images present an apparently random pattern of dark and light pixels called speckle. This is caused by interferences to the waves received by the sensor that can follow slightly different paths for the zone corresponding to the same pixel [7].

The detection of oil spills using SAR images can be divided into the following stages: preprocessing of the images, segmentation, and classification of dark spots which potentially correspond to oil spills [2]. Each stage may consist of other additional stages, which may vary according to the concrete algorithm that is being considered. For our analysis, we have worked with masks of the oil areas: this information is used to train the algorithms and to carry out a numerical evaluation with different empirical discrepancy metrics. The metrics are more closely studied in subsequent sections.

The detection and management of oil spills is a complex problem in which diverse sources of information and actors come into play. Systems to facilitate decision-making or data analysis have been proposed which integrate technologies like geographical information systems or intelligent hybrid systems [3,8–10].

This article addresses the stage of dark spot segmentation which is considered critical given that the other stages are dependent on it. This particular problem has been addressed in other works with different algorithms.

Several works considered thresholding algorithms combined with other techniques: detection of oil slicks from moving ships using a thresholding algorithm combined with active contour models, adaptive thresholding using an estimation of the wind

* Corresponding author.
E-mail addresses: diego.cantorna@udc.es (D. Cantorna), dafonte@udc.es (C. Dafonte), alfonso.iglesias@udc.es (A. Iglesias), bernardino.arcay@udc.es (B. Arcay).

https://doi.org/10.1016/j.asoc.2019.105716
1568-4946/

optimized with parallel computing techniques, adaptive thresholding combined with edge detection and region growing segmentation, thresholding combined with classical and fuzzy clustering algorithms, and segmentation by means of thresholds that are calculated from a model based on Gamma distributions [11–15].

In [16], the authors use Simulated Annealing, Genetic Algorithms, and Particle Swarm Optimization techniques. Other techniques used in previous works include adaptive edge and texture clustering, integration of region growing segmentation with GLRT (Generalized Likelihood Ratio Test) detection theory, and support vector machines [17–19].

In [20], an energy minimization process is combined with a fuzzy connectivity between pixels. The algorithm requires an external initialization to establish a pixel of the oil spill that will be segmented.

Other widely used algorithms are neural networks, sometimes also combined with other algorithms: the Otsu 2D algorithm combined with morphological filtering and a neural network, a combination of a Weibull multiplicative model for image contrast enhancement and pulse-coupled neural networks for image segmentation, and neural networks combined with image contrast enhancement using histogram stretching and speckle filtering [21–23].

The segmentation of oil spills in SAR images is a complex problem; at present, no existing algorithm is able to detect all the areas of a all types of spills without generating false positives.

It is possible to identify segmentation problems that are common to various algorithms. For instance, the segmentation of oil spills with low environment contrast is not always achieved in a satisfying manner: it is often difficult to detect all the main areas of the spill without generating an excessive amount of false positives. These could indeed distort the shape of the detected spill or even hide it completely if the false positives occupy an extensive area around the spill. Some algorithms also present difficulties to generate compact segmentations in spill areas where the texture is not homogeneous.

The comparison of the published results of various authors is complex given the diversity of methodologies and evaluation criteria employed, the relatively small number of confirmed spills and because the datasets used for validation are different.

This article evaluates a range of clustering, logistic regression and convolutional neural network algorithms applied to Envisat and Sentinel-1 satellites. Spills of various shapes and sizes have been considered in order to reflect the case of large accidental spills as well as spills of a smaller size as a result of intentional dumping or through natural causes.

2. Materials and methods

2.1. Methodology

The general scheme used for the tests that have been carried out are described in this section. The details relating to images and algorithms are described in greater detail in subsequent sections.

The available image set have been divided into training, calibration and validation sets.

The training set is used to adjust the parameters of each algorithm. In the case of logistic regression and convolutional neural networks, algorithms exist to optimize some of their parameters such as the weight connections in a convolutional neural network.

To avoid the problem of overfitting, n-fold cross validation has been considered, which involves dividing the training set into n disjoint subsets, the adjusting or training of the algorithm in n-1 of the aforementioned subsets and their evaluation in the other subset [24]. This process is repeated n times and each time a different subset is used to evaluate the results. Finally, the mean error from the validation subset is calculated, for the n tests. Once the configuration for the hyperparameters of the algorithm has been selected, training can begin using the complete training set.

In this way it is possible to estimate the generalization error of an algorithm at the same time as reducing the risk of overfitting and take advantage of all the training data to adjust the parameters.

Table 1
Features of the images.

Satellite	Mode	Swath	Spatial resolution
Envisat	Wide Swath	400 km	150 m × 150 m
Sentinel-1	Interferometric Wide Swath	250 km	5 m × 20 m

2.2. Dataset

SAR images from Envisat and Sentinel-1 satellites of the European Space Agency (ESA) have been used to evaluate the algorithms. Both cases deal with C-band radar images with VV polarization which are considered appropriate for the detection of oil spills [11].

Envisat was launched in 2002 with the objective of measuring atmosphere, oceans, ice and land. Among its instruments was ASAR (Advanced Synthetic Aperture Radar) from which the images analysed in this article proceed. In 2012, the satellite ceased to be operative having provided a vast quantity of valuable data for studies such as this one, for example, the detailed images of the Prestige oil spill [25]. The ASAR sensor had various modes to acquire data [26]. This work uses Wide Swath mode which allows the monitoring of large areas of the sea because it offers wide spatial coverage (400 km swath).

Sentinel-1 is a constellation of two satellites used to monitor both land and ocean and is an attempt to give continuity to the SAR data from the previous ERS-2 and Envisat missions. Sentinel-1 has also various modes of acquisition. Images in Interferometric Wide Swath mode are used in this study which provide wide spatial coverage (250 km swath) and high spatial resolution (5 m × 20 m).

Table 1 shows the principal features of the images used from each satellite.

For the selection of parameters and the training of the algorithms, subimages of smaller size have been used to reduce the computing load of the process. A training set different to the final test set has been used for the logistic regression and convolutional neural networks which automatically adjust parameters. In this way it is possible to estimate the error of an algorithm by applying it to data with which it has not been trained [24]. To evaluate algorithms we work with motifs, so that the number of images is multiplied by the number of motifs. The information relating to the oil spills present in the images has been obtained from diverse sources [25,27–33]. Since recently, we also collaborate with the Spanish Maritime Safety Agency (SASEMAR), which allowed us to access a dataset of in situ verified oil spills.

2.3. Preprocessing of data

The land zones in the images were eliminated at the preprocessing stage as they are not necessary at the segmentation stage. In this way the algorithms can be executed more efficiently and artificial distortions can be avoided during the evaluation process. As such, pixels representing the land can be avoided when calculating the corresponding metrics. The land zones were eliminated

using the land/sea mask options of the SNAP (Sentinel Application Platform) tool [34]. The land masks used in the previous work were employed for the Envisat images [15].

In order for the algorithms to function without change for Envisat and Sentinel-1 images, preprocessing of Envisat images was applied so that the grey levels in the pixels would have a similar distribution to the Sentinel-1 images. To achieve this, images of both satellites from the training set were used; distinct and visually homogeneous zones were selected and the average and standard deviation was calculated.

The images used in this process were extracted from the training set; they were not used in other training or evaluation steps [24]. As such, we can avoid artificial distortions in the evaluation process, which could arise if a statistical measure, for example the mean value, was computed from data in the test set and used in the training process.

Finally, the values of each pixel were adjusted in all the Envisat images from all image sets, except for land regions, so that the mean and standard deviation corresponds with those of Sentinel, by applying the following function which returns the new grey level value for the original grey level x:

$$f(x) = \frac{(x - \mu_{Envisat}) * \sigma_{Sentinel}}{\sigma_{Envisat}} + \mu_{Sentinel}$$

$$\mu_{Sentinel} = 96.490$$

$$\sigma_{Sentinel} = 24.808$$

$$\mu_{Envisat} = 1.151e + 009$$

$$\sigma_{Envisar} = 7.449e + 006,$$

where μ_s and σ_s indicate the mean and standard deviation of grey level in the satellite s images.

There are other normalization options that could have been used, for example normalization of all pixel values in all images to a fixed interval [0,1]. In this way, all the pixel values would be bounded and there would be no negative values in the normalized images. This normalization option can be useful in several image processing applications.

The pixel value distribution in SAR images is not uniform across the full range of possible pixel values, and the majority of pixel values concentrates in a relatively small value range. If we had used a normalization to a fixed interval using the maximum allowed pixel value, the normalized pixels values could be distributed around a different mean value and with a different standard deviation for Envisat and Sentinel-1 images. In order to avoid this problem, we used the previously explained normalization expression.

In this way, the segmentation algorithms can be used without adjustments on both kind of images. If a further specific normalization is needed for an algorithm, it can be done at a later stage for this algorithm. For example, it could be useful to normalize pixel values to 0 mean for the input image of a convolutional neural network, but we decided to make this decision only after performing the cross validation tests.

2.4. Segmentation algorithms

Image segmentation is a process which attempts to divide an image into regions or objects of interest for a specific application [35], which is, in this case, possible oil spills.

Tests of clustering, logistic regression and convolutional neural network algorithms were carried out for this article. Images from the training dataset were used to adjust the parameters of each algorithm.

The features of each algorithm are described as follows.

2.4.1. Training and test datasets

For all the segmentation algorithms we used the same training and test datasets.

Both sets consist of images of 2048 $\times$ 2048 pixels, a standard selection based on previous experiments [15] as well as on the resolution features of the new Sentinel-1 satellite. The size of the used images is compatible with the resolution of the Envisat and Sentinel-1 satellites, and allows the inclusion of areas with large oil spills, such as those caused by the Prestige accident [25].

The training set consists of 107 images of 2048 $\times$ 2048 pixels. The images of the training set were used to train and adjust the algorithms.

In the case of the logistic regression and convolutional neural networks algorithms, which have an automatic learning mechanism, we used the data of this images set for the training.

In order to train the convolutional neural network, this training set is divided into smaller subimages, i.e. regions of an image. More concretely, we used 630 subimages of 128 $\times$ 128 pixels: this size is adequate for the elements that we wish to detect, experimentally determined, and it allows us to reduce the computational load of the entire process.

For the adjustment of the clustering algorithms, we used the data of the training set and information from previous works [15].

The test set consists of 23 images of 2048 $\times$ 2048 pixels. These images were only used for the final comparison of the results of each algorithm. In further sections, we analyse the results by considering the complete test set, and in addition we show, for a sample of images, detailed results that were obtained by segmenting them with the best selected algorithm.

2.4.2. Clustering algorithms

Clustering algorithms are unsupervised learning techniques that attempt to find a partition in a dataset so that the differences between elements assigned to a cluster are less than for elements in different clusters [24]. The amount of clusters can be selected according to previous information of the application domain, through an interactive process in which the number of clusters is varied and the result analysed visually, or by using variants of the algorithms that allow us to automatically optimize the number of clusters. As a reference to compare results, this work uses classical variants of the clustering algorithms; the optimization of the number of clusters was based on previous works [15].

Classic (hard) clustering K-means [36], and fuzzy clustering algorithms Fuzzy C-means (FCM) [36] and Spatially Constrained Kernelized Fuzzy C-means (SKFCM) [37] have been tested.

With classic (hard) clustering, each element is associated with a unique cluster. With fuzzy clustering, each element of a set is assigned a degree of membership to each cluster. If a hard output is required, each element can be assigned to a cluster so that it has a greater degree of membership. These algorithms, in general, attempt to minimize an objective function whose principal parameters are degrees of membership and those parameters that determine the shape and localization of the clusters [36].

We used the classical K-means algorithm [36], defined the number of classes and used the Euclidean distance as a metric; this has allowed us to define a reference level that enables us to compare with previous results and with experimental modifications of the new results.

The FCM algorithm is based on K-means and also uses a euclidean distance; however, it uses fuzzy membership degrees for the clusters. This is considered to be more realistic when the data presents ambiguities and the limits between clusters are not clearly defined [36].

The SKFCM algorithm is based on the FCM but uses a distance metric based on a kernel and, in addition, includes a spatial penalty term to compensate for the inhomogeneities in the intensity values [37]. We experimented with the images of the training

set and finally selected a Gaussian Radial basis function (GRBF) kernel with standard deviation 100. Gaussian kernels are often used in various applications because of their ability to eliminate high frequency noise [35], and have been successfully applied in oil segmentation tasks [15].

Preprocessing, as has been explained earlier, is carried out prior to applying the clustering algorithms by filtering the images and labelling the pixels in land areas to avoid their segmentation. The input of clustering algorithms involves the crude pixels from these preprocessed images.

2.4.3. Logistic regression algorithm

The logistic regression algorithm is widely used for diverse classification problems given their simplicity and efficiency [24].

Following the initial stage of preprocessing, the logistic regression algorithm is applied directly on the pixels. The input of the algorithm is the value of a specific pixel and the output an estimation of the probability that the said pixel corresponds with a possible oil slick.

To avoid problems of overfitting, we considered the possibility of applying a regularization factor, selected at the stage of cross validation, which is described in a subsequent section. We also considered the possibility of carrying out an automatic adjustment to compensate for the differences in the probabilities of the classes for oil and background.

2.4.4. Convolutional neural network algorithms

Convolutional neural networks with supervised learning are algorithms which attempt to approximate a function from a training set in which different samples are provided with the desired output for each one. Subsymbolic techniques, such as neural networks, have been used in many applications, for example, in the segmentation of oil spills in images [21,22], or to optimize fishing resources using data proceeding from satellites [38,39].

Feedforward neural networks comprise a set of processing elements, also called neurons, organized into layers with connections from the elements of a layer to elements in the following layers without there being connections towards elements in the previous layers. The depth of a network indicates the number of layers it has. The term deep learning is related to this concept and is used to indicate networks with a sufficient number of layers [40–42].

The use of very flexible models with a large number of adjustable parameters can lead to the risk of overfitting. This problem is related to an excessive adjustment to specific training data including variations of a random character which cannot be extrapolated to new data.

Various regularization techniques have been considered to resolve this problem; these techniques attempt to improve the generalization capacity of the model to handle new data [42]. One of these is called dropout which involves the random elimination of neurons when training, in order to avoid excessive co-adaptation between different neurons [43].

Batch normalization regularization [44] has also been considered to normalize the input of one or more network layers.

Dataset augmentation is a technique for improving the generalization capacity of a classifier by generating modified versions of the dataset applying transformations which do not affect the classes [42]. In the tests, rotations and zoom of the data have been carried out.

The input of the tested convolutional neural networks is a representation of the pixels in the preprocessed image. The output of the convolutional neural networks is another image with the same dimensions as the input, in which each pixel is an estimation of the probability that it corresponds with an area of oil.

For the neurons in the output layer, a sigmoid function was used which is enclosed in the interval [0,1] and can be directly interpreted as an estimation of the probability of oil. For the rest of the layers, the sigmoid, hyperbolic tangent and ReLU (rectified linear unit) [40] activation functions were considered.

To train the convolutional neural networks, the training set is divided into smaller subimages to reduce the computing load of the process. The size of the subimages are selected through cross validation [24] which is explained in a subsequent section. To select the subimages for the training, random blocks of the size indicated are selected for each image in the training set. Homogeneous as well as non-homogeneous zones (zones which include pixels labelled as oil and others as background). Given oil pixels are less common than background pixels, the classes are sampled trying to balance the number of samples per class. A totally random sample could result in there being no oil pixels to select for training.

To train the convolutional neural networks, stochastic gradient descent (SGD) [42] and RMSProp [45] optimizers where tested. The binary cross entropy loss function which measures the distance between probability distributions was used and is considered appropriate for problems where the output is an estimation of probability [46].

Tests have been carried out with convolutional neural networks which are considered effective for processing data with grid topology as is the case of the images [42]. Convolutional neural networks have been applied successfully for image processing tasks [47,48].

For this article we have worked with fully convolutional neural networks without applying any subsampling step which would reduce the spatial resolution of the images. In this way, the network output can be directly interpreted as a classification for each specific pixel of the input image. In addition to the conventional convolutional layers, depthwise separable convolutional layers have been employed for their computational efficiency and because they require less adjustable parameters [46].

To select specific values for the hyperparameters such as the number of network layers, a stage of cross validation was carried out, to be explained subsequently.

Factors exist such as network depth or the use of regularization techniques which can distort output probabilities [49]. It is possible for an algorithm to have a good discrimination capacity but the estimated probabilities can differ substantially to the real probabilities. It is held that a classification algorithm is calibrated for a data sample for which the algorithm estimates a probability p, if the expected proportion of positives is similar to p [50].

In order to interpret the convolutional neural networks output as an estimation of the probability of oil, calibration of output probabilities is carried out using the histogram binning technique. This is a non-parametric method which involves dividing non-calibrated probabilities into disjoint intervals and assigning a calibrated probability to each interval [49,51]. To obtain the calibration data, a specific calibration dataset has been used which has not been used for training or for the evaluation of the convolutional neural networks.

Fig. 1 illustrates the process of land zones elimination for an image of the training set; this image is then segmented by the selected convolutional neural network, to which a probability threshold of 0.9 is applied.

Convolutional neural networks can automatically learn useful representations for a specific task. Each convolutional filter can learn to respond to some relevant feature. The filters in deeper layers represent more abstract features using combinations learnt from previous layer outputs [42]. This capability can be useful in complex tasks like oil spill detection, reducing the need for specific manual feature definition and selection.

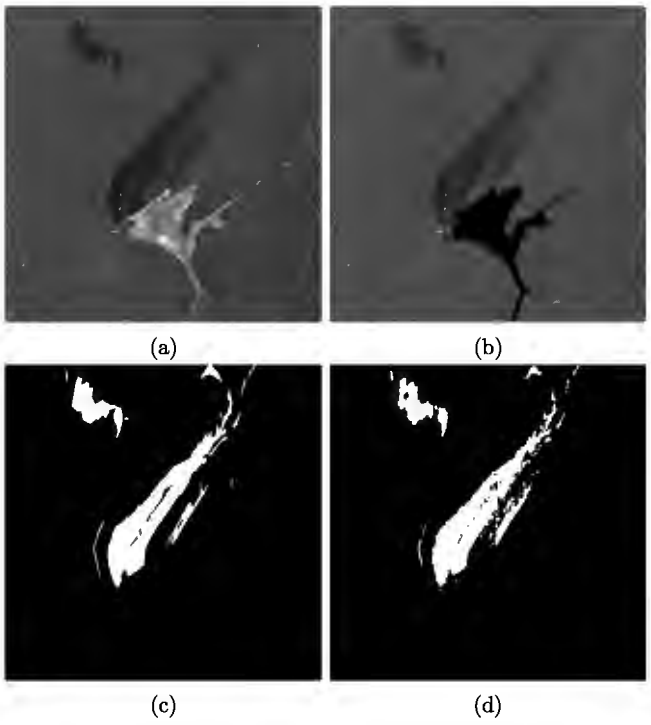

Fig. 1. Application of the preprocessing and segmentation stages a to a Sentinel-1 image: (a) Original image. (b) Preprocessed image with land zones eliminated. (c) Mask. (d) Segmentation.

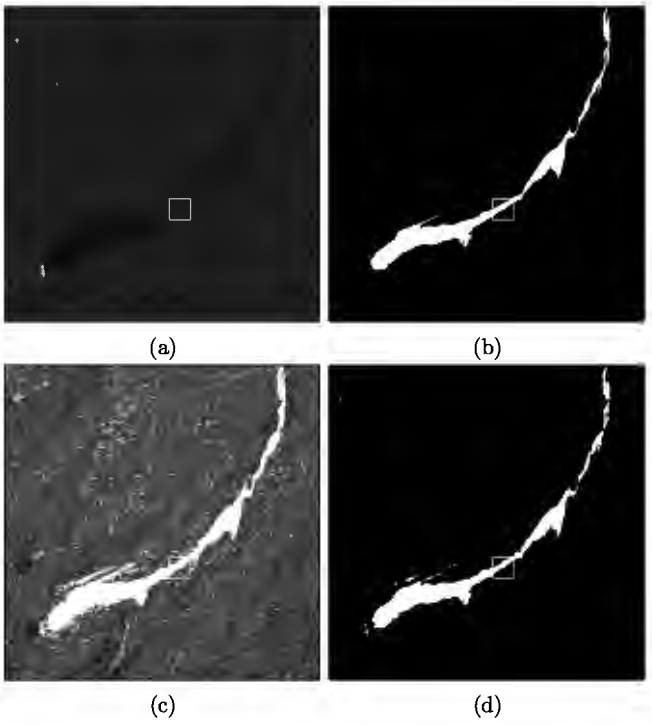

Fig. 2. Application of the convolutional neural network to a Sentinel-1 image from the test set: (a) Original image. (b) Mask. (c) Convolutional neural network output. (d) Segmentation.

Sometimes neural networks are described as black boxes, which could learn complex mappings from input space to output space, but whose internal representations are difficult to analyse using high level concepts. Nevertheless, in the specific case of convolutional neural networks it is possible to take advantage of its particular structure oriented to the processing of grid like structures to show a representation of the features learnt [46].

In order to illustrate the activation pattern of the convolutional neural network filters we selected a region of a Sentinel-1 image from the test set which includes oil and background pixels. Fig. 2 shows the selected region in the input image, mask, convolutional neural network output, and binary segmented image obtained after applying a probability threshold of 0.9 to the network output.

Fig. 3 shows a representation of the activation pattern of the convolutional neural network filters for the selected Sentinel-1 input image. It shows differences in the filter responses that indicate specific filter sensibilities for distinct patterns or textures in the image.

2.5. Evaluation method

To evaluate the results of segmentation algorithms diverse empirical discrepancy methods have been used. This type of metric is based on comparing the results of an algorithm with a reference image, called a mask. To create the masks for the Sentinel-1 images, the zones of possible oil spills are selected manually using the polygon selection options of the SNAP (Sentinel Application Platform) tool [34]. The masks used in the previous work were employed for the Envisat images [15].

Image segmentation is a complex process and it can be difficult to resume the results under one measure, even if it is a measure composed by others [52]. As such, we have decided to calculate various measures of discrepancy and combine them with a visual analysis of the results.

Table 2
Confusion matrix.

	Background (algorithm)	Oil (algorithm)
Background (mask)	TN	FP
Oil (mask)	FN	TP

We can consider the final output of a segmentation algorithm as a binary classification for each pixel of the image so that the *positive* class indicates oil and the *negative* indicates background. The following values can be defined:

- True Positives (TP): oil pixels correctly classified.
- True Negatives (TN): background pixels correctly classified.
- False Positives (FP): background pixels incorrectly classified as oil.
- False Negatives (FN): oil pixels incorrectly classified as background.

These values can be represented using a confusion matrix (see Table 2).

Distinct metrics can be defined from these values:

- Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$
- Precision $= \frac{TP}{TP+FP}$
- Recall or TPR (True Positive Rate) $= \frac{TP}{TP+FN}$
- F1 $= \frac{2*Precision*Recall}{Precision+Recall}$
- FPR (False Positive Rate) $= \frac{FP}{FP+TN}$
- Jaccard index J $= \frac{TP}{TP+FP+FN}$

Precision can be considered a measure of exactness (of the pixels classified as oil, how many are classified correctly) and Recall as a measure of completeness (of the total of pixels which are truly oil, how many have been correctly classified) [53]. The metric F1 is the harmonic mean of Precision and Recall [54].

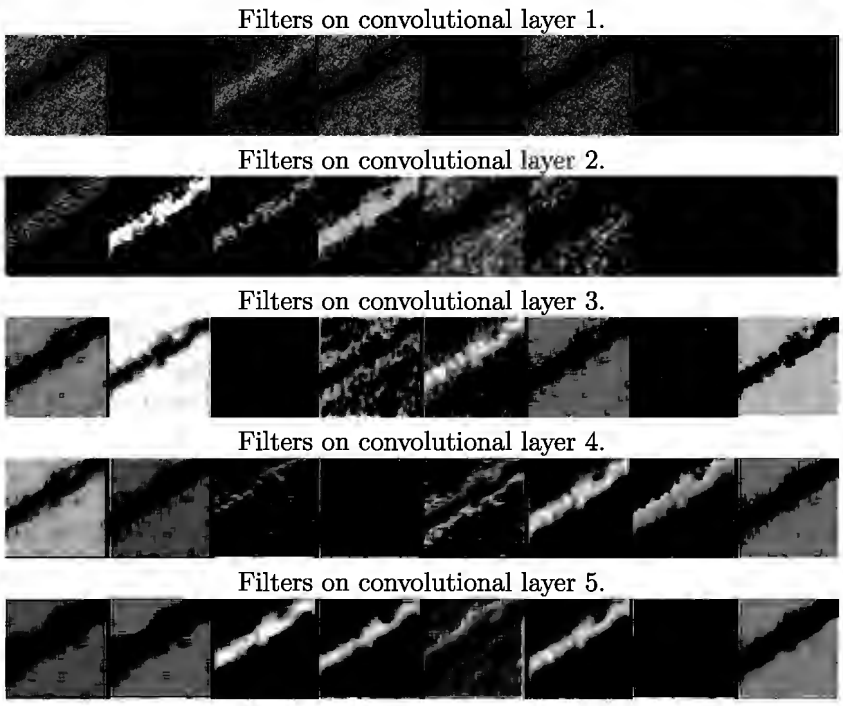

Fig. 3. Filter activation pattern obtained after applying the convolutional neural network to the selected region of a Sentinel-1 image from the test set. Each row shows the filters of a convolutional layer. The upper row represents the first convolutional layer in the network.

The Jaccard index estimates the likelihood of a pixel corresponding to an oil zone, if it is not correctly classified as background [55].

If the number of samples in one of the classes is considerably greater than that of another, it is necessary to take into account that some metrics can be deceptive. For example, if the positive class occurs only once each ten times, an algorithm which always predicts the negative class will have an accuracy of 0.9, even though the algorithm has no utility.

In these cases, it may be more useful to use Precision–Recall curves which graphically illustrate both metrics for distinct thresholds in the algorithm output [53,56]. For example, if the algorithm output is a probability, a threshold of 0.1 can be selected and the corresponding Precision and Recall calculated, with which a point on the curve can be obtained; this process would be repeated for different thresholds.

The selection of a threshold is, at first, in the hands of the user of the algorithm which can be varied depending on specific needs. However, even in this case, the Precision–Recall curves can provide valuable information for the evaluation of algorithms. For example, it could be that an algorithm always has a high Precision for any specific Recall and so it is better (according to these metrics) for any threshold that may be selected.

ROC curves is another method of evaluation which shows how the TPR metric varies in function of FPR [54]. The TPR (True Positive Rate) is equivalent to the metric Recall. The FPR value (False Positive Rate) indicates how many background pixels have been incorrectly classified as oil. The area under the ROC curve (AUC) is a commonly used measure to summarize the information of a ROC curve and indicates the capacity of an algorithm to distinguish between classes.

2.6. Cross validation and parameter selection

There are training algorithms that can automatically adjust the parameters of logistic regression and convolutional neural network models. However there are hyperparameters, such as the number of layers in an convolutional neural network, which are not optimized during training. The technique of cross validation is used to select the values of hyperparameters.

Table 3
Parameters of the logistic regression algorithm in the tests with 5-fold cross validation.

	LogReg 1	LogReg 2	LogReg 3	LogReg 4	LogReg 5
Preprocessing–Normalization	**Yes**	No	Yes	No	No
Preprocessing–Filtering	**Median** **7 × 7**	Median 7 × 7	Median 5 × 5	Median 5 × 5	Median 3 × 3
Probability adjustment	**Yes**	Yes	Yes	No	No
Regularization factor	**1.705**	1.816	1.380	1.426	0.885
Precision	**0.916** ± **0.057**	0.908 ± 0.065	0.883 ± 0.084	0.880 ± 0.079	0.794 ± 0.128

Tests have been carried out with 5-fold cross validation to conduct a random search in parameter space as has previously been described.

The stage of preprocessing in addition to the segmentation stage have been taken into account during the search process. We have carried out tests on whether or not to use a filter to reduce image noise, and whether or not to normalize the mean of the images to reduce the variability between different images.

All the tests at this stage were carried out with the training dataset. The validation dataset was only used at the final stage once the hyperparameter values for each algorithm were selected.

A total of 50 tests were carried out for the logistic regression algorithm. On the basis of the cross validation tests carried out for this algorithm, the configuration selected consists of a mean normalization stage, 7 × 7 median filter, automatic adjusting of probabilities for oil and background classes and the regularization factor indicated in Table 3 (see also Fig. 4).

The adjustment of convolutional neural networks is quite complex because of the number of parameters that they have

Table 4
Parameters of the convolutional neural networks selected during the tests with 5-fold cross validation.

	ConvNet 1	ConvNet 2	ConvNet 3	ConvNet 4	ConvNet 5
Preprocessing-Normalization	**Yes**	Yes	Yes	Yes	Yes
Preprocessing-Filtering	**–**	Median 3×3	Median 3×3	–	–
Subimages size	**128**	128	128	128	128
Convolutional layers	**6**	5	4	3	3
Depthwise separable convolution	**Yes**	Yes	Yes	Yes	Yes
Filters per layer	**8**	8	4	8	8
Convolutional kernel size	**5**	7	9	9	7
Activation function	**ReLU**	ReLU	ReLU	ReLU	tanh
Weight initialization	$\mathcal{N}(0.06, 0.35)$	$\mathcal{N}(0.08, 0.39)$	$\mathcal{N}(0.01, 0.17)$	$\mathcal{N}(0.07, 0.28)$	$\mathcal{N}(0.09, 0.35)$
Bias initialization	**0**	0	1	0	0
Optimizer	**RMSprop**	RMSprop	RMSprop	RMSprop	RMSprop
Learning rate	**0.005**	0.001	0.001	0.001	0.005
Batch size	**7**	4	4	10	6
Precision	**0.945 ± 0.029**	0.940 ± 0.034	0.937 ± 0.037	0.936 ± 0.035	0.932 ± 0.040

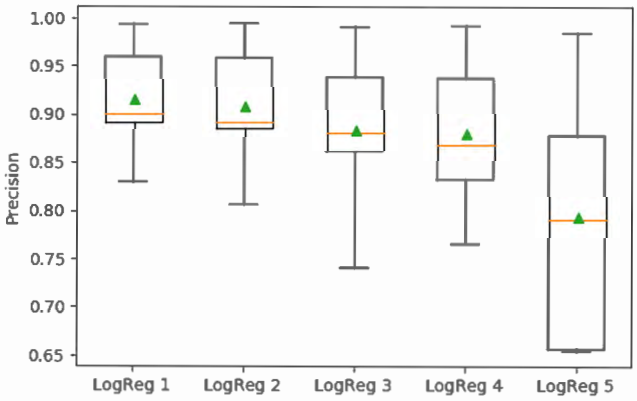

Fig. 4. Evaluation of the logistic regression algorithm selected during the tests with 5-fold cross validation.

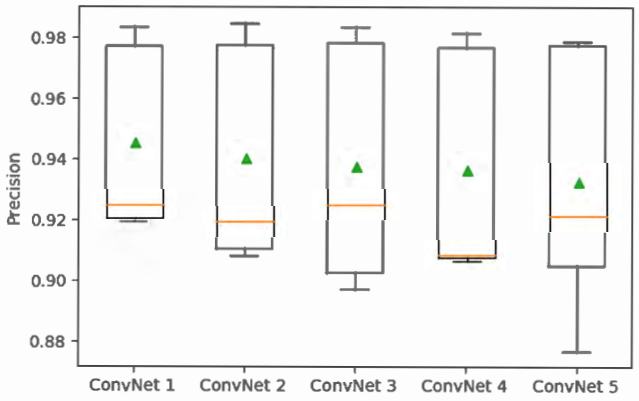

Fig. 5. Evaluation of the convolutional neural networks selected during the tests with 5-fold cross validation.

and the time they require for training. For this, an exploration of parameter space has been carried out in various stages:

- Initial stage: determine a number of training epochs which are needed for a general estimation of error for a network. Tests with 25, 50 and 75 epochs were carried out. No appreciable difference was observed above 50 epochs. Given that training time is an important consideration to be able to test a sufficiently large number of different configurations, we opted for the use of 50 epochs. 3 fold cross validation (instead of 5 fold cross validation) was used to speed up this stage.
- Candidate selection stage: 100 tests were conducted with 50 epochs for each one. 3 fold cross validation was used to speed up this stage (instead of 5 fold cross validation).
- Training stage with the best candidates: the best candidates from the previous stage were selected and fine adjustments to the hyperparameters were made, training for more epochs (100 instead of 50) using 5 fold cross validation.
- Finally, the network which obtained the best results in the previous stage was selected and was then trained with all the images from the training dataset.

On the basis of the cross validation tests carried out for convolutional neural networks, the configuration indicated in Table 4 was selected which has the highest Precision (see also Fig. 5).

During the test phase, we considered the possibility of varying the parameters of Table 4. The indicated configurations provided the best results. For testing we used convolutional neural networks with one same amount of filters for each convolutional layer. It would be interesting to allow a different number of filters per layer, but we decided to simplify the selection of these values for the sake of the experiment (which is already extensive and computationally complex). The training process in itself causes some network elements to *die* when using ReLU-type activation functions: the weights acquire values that result in a gradient of the cost function 0 for any input [57]. This regularization mechanism in the training process itself carries out a selection of the network architecture that hardly differs from previous selections.

In accordance with the selected configuration, a mean normalization of the images was conducted at the preprocessing stage without applying a noise filter. The learning algorithm used is RMSProp with a learning rate of 0.005. 630 subimages of 128×128 pixels were used for the training process. The selected convolutional neural network has 6 convolutional layers with 8 5×5 convolutional filters in each layer. In all the convolutional layers zero-padding was used [46], causing the spatial extension of the output of a convolutional layer to be the same as that of its input. The activation function is of the ReLU type, except for the output layer, which has a sigmoid-type activation.

Table 5 indicates the regularization techniques used for the convolutional neural network training.

Fig. 6 shows the architecture of the convolutional neural network.

Table 5
Regularization techniques applied for the convolutional neural networks selected during the tests with 5-fold cross validation.

	ConvNet1	ConvNet2	ConvNet3	ConvNet4	ConvNet5
Label smoothing	**0.100**	0.150	–	0.100	0.010
Weight range limit	**[−3,3]**	–	[−2,2]	[−2,2]	–
Dropout on input layer	–	–	0.191	0.1	0.200
Dropout on first conv. layer	**0.124**	0.066	0.174	0.3	–
Dropout on other conv. layers	**0.180**	0.168	0.095	0.3	–
Batch normalization	**No**	No	Yes	No	No
Dataset augmentation — rotation range	±32.6°	±15.9°	±34.1°	±16.3°	±18.1°
Dataset augmentation — zoom range	±23.4%	±16.3%	±13.6%	±24.6%	±28.6%

Table 6
Results of the algorithms in the 5-fold cross validation stage and in the test set.

Algorithm	5-fold precision	Test precision	Test time
kMeans	–	0.807 ± 0.163	23 ± 8 seg.
SKFCM	–	0.825 ± 0.146	365 ± 129 seg.
FCM	–	0.849 ± 0.132	122 ± 5 seg.
Logistic regression	0.916 ± 0.057	0.910 ± 0.065	3 seg.
Convolutional neural network	0.945 ± 0.029	0.946 ± 0.051	27 ± 1 seg.
Convolutional neural network (GPU)	"	"	19 ± 1 seg.

Table 7
Confusion matrix of the convolutional neural network considering all images in the test set, and confusion matrixes normalized by rows or columns.

	Conf. matrix		C.m. norm. rows		C.m. norm. cols	
	Back. (alg.)	Oil (alg.)	Back. (alg.)	Oil (alg.)	Back. (alg.)	Oil (alg.)
Back. (mask)	86 742 728	1 128 462	0.987	0.013	0.994	0.143
Oil (mask)	536 216	6 781 903	0.073	0.927	0.006	0.857

3. Results

A number of clusters equal to ten was considered for the clustering algorithms. Two clusters were considered sufficient at the beginning given that we were only interested in distinguishing oil from background areas. However, if the image contains oil slicks of a limited size these may not be distinguished if a sufficient number of clusters is not considered.

All of the tests were carried out on the same computer with the following characteristics: Intel(R) Core(TM) i5-4200M CPU 2.50 GHz, 8 GB RAM, NVidia GeForce 740M 2 GB graphics card.

Table 6 indicates the Precision and average processing time for each algorithm in the test set. For the logistic regression and convolutional neural network algorithms, we also indicate the values obtained in the tests with 5-fold cross validation.

The convolutional neural network has the best mean Precision and can directly provide a classification of oil versus background areas and appears to be the most adequate selection. An implementation of this algorithm was made with the NVIDIA CUDA platform [58] to take advantage of its parallel computing power. This implementation reduces the average computing time from 27 to 19 s.

Fig. 7 shows the Precision–Recall and ROC of the convolutional neural network, calculated for all the data of the final test dataset.

To obtain a binary classification it suffices to select a probability threshold and apply it directly to the output of the convolutional neural network. The selection of a threshold is equivalent to setting a point from the corresponding Precision–Recall curve.

To illustrate the results, we have chosen an oil probability of 0.9. The pixels, therefore, have been chosen so that the convolutional neural network considers that the probability is higher than the said threshold. This threshold obtains a Precision of 0.857, a Recall of 0.927, a F1 value of 0.891, a Jaccard index of 0.803 and 98.3% of pixels that are correctly classified.

The threshold is the same for all the images so that the coherence of the algorithm can be illustrated on the labelling of the pixels; on using the same threshold one can, for example, establish an automatic mechanism which would order a set of images for processing according to its probability of containing oil slicks, to be subsequently revised by an operator or by another algorithm.

Table 7 shows the corresponding confusion matrix considering all images in the test set, and also shows the same data normalized by rows or columns respectively.

Images 8 and onward display the results of the convolutional neural network applied to a range of images from the Envisat and Sentinel-1 satellites. For each input image, the mask used to evaluate results is shown as well as the binary image that results from the application of the selected threshold to the segmentation.

In general the algorithm outputs are similar to the corresponding masks. Nevertheless there are some false positives and negatives. For example, there are false positives in the upper part of image 8, and false negatives in the thin linear part of the oil slick in the lower right part of image 10. These issues indicate a difficulty in distinguishing small slicks from background pixels when the contrast between the slick and the surrounding pixels is too low.

Tables 8 and 9 show the confusion matrixes for the Envisat and Sentinel images from the test displayed on Fig. 8 and onward. The tables also show the same data normalized by rows or columns respectively.

The segmentation algorithm has been integrated into a decision support system in order to facilitate decision-making and data analysis of possible oil spill events. The system has a web interface that allows users to search and view images, request asynchronous segmentation of images, and also display the segmentation outputs. The components of the system communicate by means of REST APIs [59] and a queue subsystem implemented in Node.js [60]. The data persistence layer is based on PostGIS [61], which provides geospatial functionality to a PostgreSQL [61] database. This system architecture allows the components to be loosely coupled and facilitate asynchronous communications between them.

4. Conclusions

This article has evaluated different algorithms for the segmentation of oil in SAR images from the Envisat and Sentinel-1 satellites. The images studied show various oil spills of diverse shapes and sizes to try to reflect large oil spills such as the Prestige and other smaller spills, whether they are intentional, accidental or a result of natural causes.

The numerical comparison of the results published by various authors is complicated by the diverse methodologies and the evaluation criteria employed, the relative shortage of confirmed

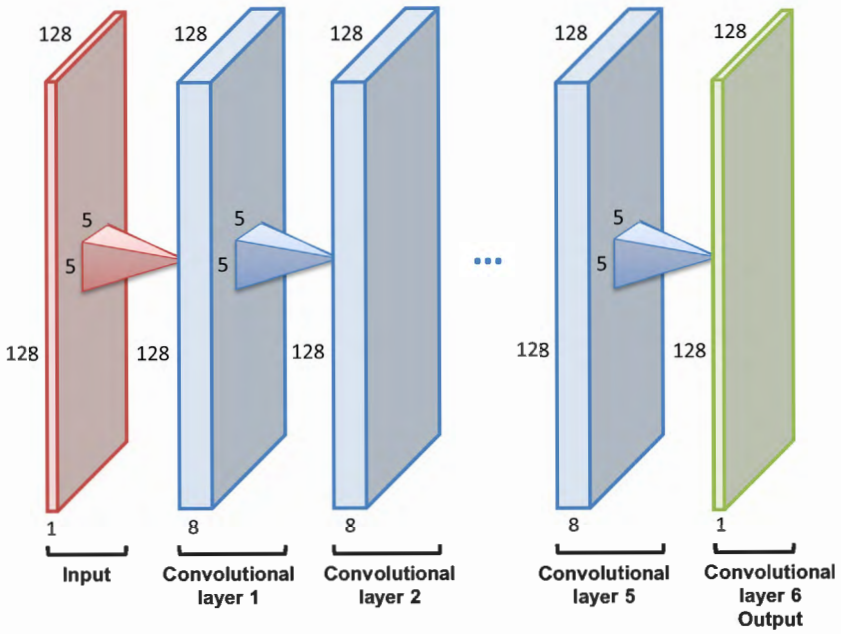

Fig. 6. Architecture of the convolutional neural network.

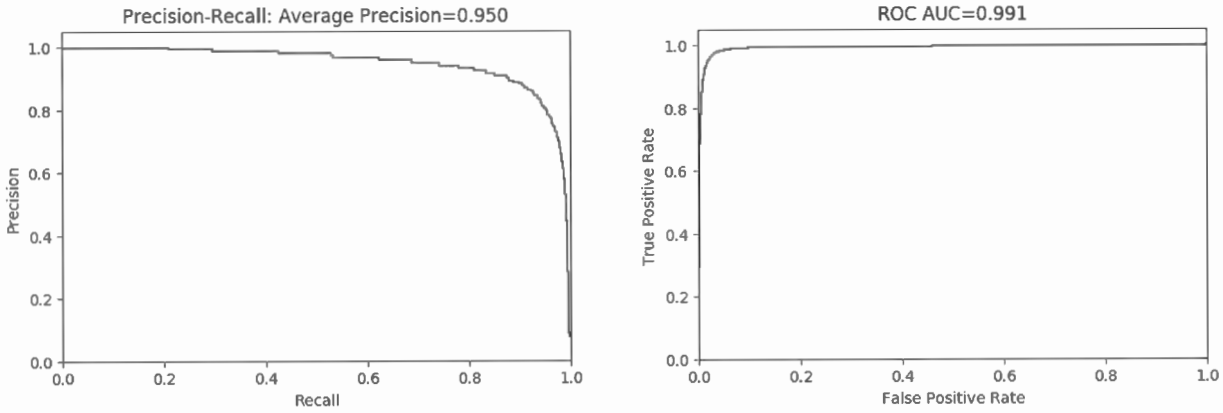

Fig. 7. Evaluation of the convolutional neural network in the test dataset: (a) Precision–Recall curve. (b) ROC curve.

Table 8
Confusion matrixes of the convolutional neural network for individual Envisat images in the test set, and confusion matrixes normalized by rows or columns.

		Conf. matrix		C.m. norm. rows		C.m. norm. cols	
		Back. (alg.)	Oil (alg.)	Back. (alg.)	Oil (alg.)	Back. (alg.)	Oil (alg.)
Im. 8							
	Back. (mask)	3 127 682	88 884	0.972	0.028	0.996	0.325
	Oil (mask)	12 862	184 741	0.065	0.935	0.004	0.675
Im. 9							
	Back. (mask)	3 398 263	4 576	0.999	0.001	0.986	0.006
	Oil (mask)	47 909	743 556	0.061	0.939	0.014	0.994

spills, and because the datasets used for validation are usually different.

The convolutional neural network that was presented in this article allows us to carry out an automatic segmentation and obtain a balance between the detection of the main areas of the considered oil spills and a reduced number of false positives, while reasonably preserving the shape of the oil spill and keeping the computing time to a minimum. We consider this to be an improvement of existing similar algorithms that were analysed in the Introduction section.

For the analysis of the results, various empirical discrepancy metrics were used, calculated on the basis of image masks.

One can expect a relatively high percentage of correctly classified pixels as a result, simply, of the imbalance in the number of oil and background pixels. As such, this metric can be deceptive if not analysed in conjunction with others.

A mean Precision of 0.950 was obtained using a convolutional neural network with low computing time. Different points of the Precision–Recall curve can be selected according to specific needs, determining a balance between Precision and Recall. The

Table 9
Confusion matrixes of the convolutional neural network for individual Sentinel images in the test set, and confusion matrixes normalized by rows or columns.

		Conf. matrix		C.m. norm. rows		C.m. norm. cols	
		Back. (alg.)	Oil (alg.)	Back. (alg.)	Oil (alg.)	Back. (alg.)	Oil (alg.)
Im. 10							
	Back. (mask)	4 102 295	7 887	0.998	0.002	0.997	0.097
	Oil (mask)	10 489	73 633	0.125	0.875	0.003	0.903
Im. 11							
	Back. (mask)	3 637 550	59 677	0.984	0.016	0.993	0.112
	Oil (mask)	26 264	470 813	0.053	0.947	0.007	0.888
Im. 12							
	Back. (mask)	3 892 691	84 485	0.979	0.021	0.997	0.290
	Oil (mask)	10 469	206 659	0.048	0.952	0.003	0.710
Im. 13							
	Back. (mask)	4 113 944	43 107	0.990	0.010	0.999	0.557
	Oil (mask)	2 981	34 272	0.080	0.920	0.001	0.443
Im. 14							
	Back. (mask)	3 264 028	22 642	0.993	0.007	0.981	0.026
	Oil (mask)	63 253	844 381	0.070	0.930	0.019	0.974
Im. 15							
	Back. (mask)	2 981 434	4 125	0.999	0.001	0.973	0.004
	Oil (mask)	84 252	1 124 493	0.070	0.930	0.027	0.996
Im. 16							
	Back. (mask)	4 038 066	12 645	0.997	0.003	0.998	0.085
	Oil (mask)	7 388	136 205	0.051	0.949	0.002	0.915

Fig. 8. Application of the convolutional neural network to an Envisat image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

Fig. 9. Application of the convolutional neural network to an Envisat image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

threshold corresponding to a Precision of 0.857 and a Recall of 0.927 has been used to illustrate the results. With this threshold, the F1 value (harmonic mean of Precision and Recall) is 0.891, the Jaccard index is 0.803 and the percentage of correctly classified pixels is 98.3%.

Although logistic regression and clustering algorithms can be considered useful for oil spill segmentation, the combination of convolutional techniques and neural networks achieves the best results, and also reduces the computing time with respect to the algorithms considered in previous works [15]. The best performing segmentation algorithm has been integrated into a decision support system in order to facilitate decision-making and data analysis of possible oil spill events.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.105716.

Fig. 10. Application of the convolutional neural network to a Sentinel-1 image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

Fig. 11. Application of the convolutional neural network to a Sentinel-1 image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

Fig. 12. Application of the convolutional neural network to a Sentinel-1 image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

Fig. 13. Application of the convolutional neural network to a Sentinel-1 image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

Acknowledgements

The authors would like to thank the Spanish Maritime Safety Agency (SASEMAR) for providing us access to a dataset of in situ verified oil spills.

The authors are also grateful to the European Space Agency (ESA) and the Copernicus Open Access Hub for allowing us to use SAR images and tools.

Fig. 14. Application of the convolutional neural network to a Sentinel-1 image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

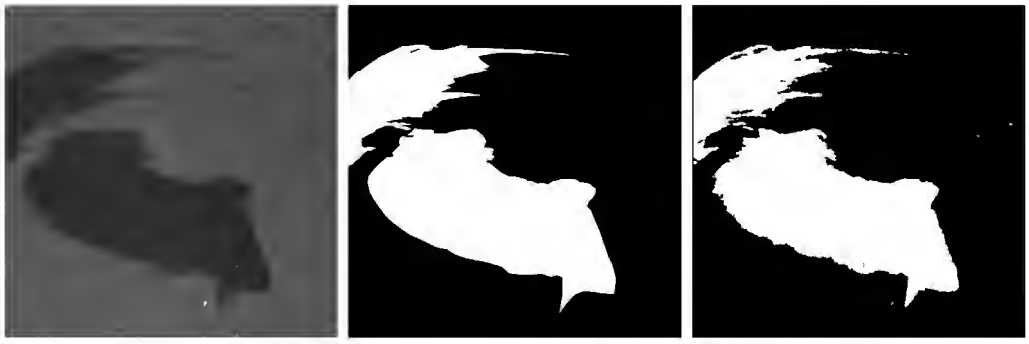

Fig. 15. Application of the convolutional neural network to a Sentinel-1 image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

Fig. 16. Application of the convolutional neural network to a Sentinel-1 image from the test set: (a) Original image. (b) Mask. (b) Segmentation.

Part of this work was supported by the Xunta de Galicia, Spain (Potencial Crecemento ED431B 2018/42 and Centro Singular ED431G/01) and the European Union (European Regional Development Fund-ERDF); we also used IT infrastructure that was acquired through the RTI2018-095076-B-C22 and ESP2016-80079-C2-2-R projects, financed by the Spanish Ministry of Science, Innovation and Universities and the Ministry of Economy, Industry and Competitiveness, Spain.

References

[1] J. Beyer, H.C. Trannum, T. Bakke, P.V. Hodson, T.K. Collier, Environmental effects of the deepwater horizon oil spill: A review, Mar. Pollut. Bull. 110 (1) (2016) 28–51, http://dx.doi.org/10.1016/j.marpolbul.2016.06.027.

[2] A.H. Solberg, Remote sensing of ocean oil-spill pollution, Proc. IEEE 100 (10) (2012) 2931–2945, http://dx.doi.org/10.1109/JPROC.2012.2196250.

[3] P. Genovez, N. Ebecken, C. Freitas, C. Bentz, R. Freitas, Intelligent hybrid system for dark spot detection using SAR data, Expert Syst. Appl. 81 (2017) 384–397, http://dx.doi.org/10.1016/j.eswa.2017.03.037.

[4] O. Garcia-Pineda, J. Holmes, M. Rissing, R. Jones, C. Wobus, J. Svejkovsky, M. Hess, Detection of oil near shorelines during the deepwater horizon oil spill using synthetic aperture radar (SAR), Remote Sens. 9 (6) (2017) http://dx.doi.org/10.3390/rs9060567.

[5] M. Fingas, C.E. Brown, A review of oil spill remote sensing, Sensors 18 (1) (2018) http://dx.doi.org/10.3390/s18010091, 2018.

[6] D. Mera, V. Bolon-canedo, J.M. Cotos, A. Alonso-betanzos, On the use of feature selection to improve the detection of sea oil spills in SAR images, Comput. Geosci. 100 (December 2016) (2017) 166–178, http://dx.doi.org/10.1016/j.cageo.2016.12.013.

[7] T. Lillesand, R. Kiefer, J. Chipman, Remote Sensing and Image Interpretation, Wiley, 2014.

[8] D. Mera, J.M. Cotos, J. Varela-Pet, P. G. Rodríguez, A. Caro, Automatic decision support system based on SAR data for oil spill detection, Comput. Geosci. 72 (2014) 184–191, http://dx.doi.org/10.1016/j.cageo.2014.07.015.

[9] A.J. Davies, M.J. Hope, Bayesian Inference-based environmental decission support systems for oil spill response strategy selection, Elsevier Mar. Pollut. Bull. 96 (2015) 87–102.

[10] G. Zodiatis, M. De Dominicis, L. Perivoliotis, H. Radhakrishnan, E. Georgoudis, M. Sotillo, R.W. Lardner, G. Krokos, D. Bruciaferri, E. Clementi, A. Guarnieri, A. Ribotti, A. Drago, E. Bourma, E. Padorno, P. Daniel, G. Gonzalez, C. Chazot, V. Gouriou, X. Kremer, S. Sofianos, J. Tintore, P. Garreau, N. Pinardi, G. Coppini, R. Lecci, A. Pisano, R. Sorgente, L. Fazioli, D. Soloviev, S. Stylianou, A. Nikolaidis, X. Panayidou, A. Karaolia, A. Gauci, A. Marcati, L. Caiazzo, M. Mancini, The mediterranean decision support system for marine safety dedicated to oil slicks predictions, Deep-Sea Res. Part II 133 (2016) 4–20, http://dx.doi.org/10.1016/j.dsr2.2016.07.014.

[11] D. Mera, J.M. Cotos, J. Varela-Pet, O. Garcia-Pineda, Adaptive thresholding algorithm based on SAR images and wind data to segment oil spills along

the northwest coast of the iberian peninsula, Mar. Pollut. Bull. 64 (10) (2012) 2090–2096, http://dx.doi.org/10.1016/j.marpolbul.2012.07.018.
[12] L.W. Mdakane, W. Kleynhans, An image-segmentation-based framework to detect oil slicks from moving vessels in the southern african oceans using SAR imagery, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (6) (2017) 2810–2818, http://dx.doi.org/10.1109/JSTARS.2017.2671403.
[13] F. Yu, W. Sun, J. Li, Y. Zhao, Y. Zhang, G. Chen, An improved otsu method for oil spill detection from SAR images, Oceanologia 59 (3) (2017) 311–317, http://dx.doi.org/10.1016/j.oceano.2017.03.005.
[14] A. El-Zaart, A.A. Ghosn, SAR Images thresholding for oil spill detection, in: 2013 Saudi International Electronics, Communications and Photonics Conference, SIECPC 2013 (i), 2013, pp. 1–5, http://dx.doi.org/10.1109/SIECPC.2013.6550755.
[15] D. Fustes, D. Cantorna, C. Dafonte, B. Arcay, A. Iglesias, M. Manteiga, A cloud-integrated web platform for marine monitoring using GIS and remote sensing. application to oil spill detection through SAR images, Future Gener. Comput. Syst. 34 (2014) 155–160, http://dx.doi.org/10.1016/j.future.2013.09.020.
[16] S.V.T. Manikonda, A. Sheta, A. Katangur, S.A. King, Metaheuristic search algorithms for oil spill detection using sar images, in: 2018 8th International Conference on Computer Science and Information Technology, CSIT 2018, IEEE, 2018, pp. 143–149, http://dx.doi.org/10.1109/CSIT.2018.8486150.
[17] J. Senthil Murugan, V. Parthasarathy, AETC: Segmentation and classification of the oil spills from sar imagery, Environ. Forensics 18 (4) (2017) 258–271, http://dx.doi.org/10.1080/15275922.2017.1368044.
[18] L. Chang, Z.S. Tang, S.H. Chang, Y.L. Chang, A region-based GLRT detection of oil spills in SAR images, Pattern Recognit. Lett. 29 (14) (2008) 1915–1923, http://dx.doi.org/10.1016/j.patrec.2008.05.022.
[19] J. Fan, F. Zhang, D. Zhao, J. Wang, Oil spill monitoring based on SAR remote sensing imagery, Aquat. Procedia 3 (2015) 112–118, http://dx.doi.org/10.1016/j.aqpro.2015.02.234.
[20] P. Ren, M. Xu, Y. Yu, F. Chen, X. Jiang, E. Yang, Energy minimization with one dot Fuzzy initialization for marine oil spill segmentation, IEEE J. Ocean. Eng. PP (2018) 1–14, http://dx.doi.org/10.1109/JOE.2018.2842538.
[21] P. Zhao, X. Yang, Y. Chen, L. Tong, L. He, Feature extraction and classification of ocean oil spill based on sar image, in: International Geoscience and Remote Sensing Symposium (IGARSS), Vol. 2016-Novem, 2016, pp. 1488–1491, http://dx.doi.org/10.1109/IGARSS.2016.7729380.
[22] A. Taravat, D. Latini, F. Del Frate, Fully automatic dark-spot detection from sar imagery with the combination of nonadaptive weibull multiplicative model and pulse-coupled neural networks, IEEE Trans. Geosci. Remote Sens. 52 (5) (2014) 2427–2435, http://dx.doi.org/10.1109/TGRS.2013.2261076.
[23] S. Singha, T.J. Bellerby, O. Trieschmann, Satellite oil spill detection using artificial neural networks, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. (2013) http://dx.doi.org/10.1109/JSTARS.2013.2251864.
[24] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed., in: Springer Series in Statistics, Springer New York, 2009.
[25] ESA, Prestige oil spill, https://earth.esa.int/web/guest/data-access/sample-data/-/asset_publisher/tg8V/content/prestige-oil-spill-galicia-spain-1623 (Online; accessed on 18.12.17).
[26] ESA, Envisat ASAR sensor modes, https://earth.esa.int/web/sppa/mission-performance/esa-missions/envisat/asar/sensor-modes (Online; accessed on 30.12.17).
[27] ESA, Seepage activity in the Gulf of Aden, https://sentinel.esa.int/web/sentinel/news/content/-/asset/_publisher/BZewkR1itkH2/content/seepage-activity-in-the-gulf-of-aden (Online; accessed on 18.4.18).
[28] SkyTruth, Oil leaks in Angola's 'Golden Block', https://www.skytruth.org/2017/06/oil-leaks-in-angolas-golden-block/ (Online; accessed on 14.05.18).
[29] SkyTruth, Oil Spill in the Persian Gulf, https://www.skytruth.org/2017/03/oil-spill-in-the-persian-gulf/ (Online; accessed on 14.05.18).
[30] SkyTruth, Bilge Dump? in Gulf of Mexico, https://www.skytruth.org/2017/03/bilge-dump-in-gulf-of-mexico/ (Online; accessed on 10.06.18).
[31] SkyTruth, ENI - Italian Firm Recently Approved for Offshore Exploration in Alaska - Responsible for Last Week's UK Oil Spill, https://www.skytruth.org/2017/07/eni-italian-firm-recently-approved-for-offshore-exploration-in-alaska-responsible-for-last-weeks-uk-oil-spill/ (Online; accessed on 10.06.18).
[32] M. Mityagina, O. Lavrova, Satellite survey of inner seas: Oil pollution in the black and Caspian seas, Remote Sens. 8 (10) (2016) 875, http://dx.doi.org/10.3390/rs8100875.
[33] K. Topouzelis, S. Singha, Oil spill detection using space-Borne sentinel-1 SAR imagery, in: Oil Spill Science and Technology, second ed., 2016, pp. 387–402, http://dx.doi.org/10.1016/B978-0-12-809413-6.00006-0.
[34] ESA, STEP - science toolbox exploitation platform - SNAP, https://step.esa.int/main/toolboxes/snap/ (Online; accessed on 11.06.18).
[35] R.C. Gonzalez, R.E. Woods, Digital Image Processing, fourth ed., Pearson, 2017.
[36] F. Klawonn, F. Höppner, What is Fuzzy about Fuzzy clustering? understanding and improving the concept of the fuzzifier, in: M.R. Berthold, H. Lenz, E. Bradley, R. Kruse, C. Borgelt (Eds.), Advances in Intelligent Data Analysis V, 5th International Symposium on Intelligent Data Analysis, IDA 2003, Berlin, Germany, August 28-30, 2003, Proceedings, in: Lecture Notes in Computer Science, vol. 2810, Springer, 2003, pp. 254–264.
[37] D.-Q. Zhang, S.-C. Chen, A novel kernelized fuzzy c-means algorithm with application in medical image segmentation, Artif. Intell. Med. 32 (1) (2004) 37–50.
[38] A. Iglesias, B. Arcay, J.M. Cotos, J.A. Taboada, C. Dafonte, A comparison between functional networks and artificial neural networks for the prediction of fishing catches, Neural Comput. Appl. 13 (1) (2004) 24–31, http://dx.doi.org/10.1007/s00521-004-0402-7.
[39] A. Iglesias, C. Dafonte, B. Arcay, J. Cotos, Integration of remote sensing techniques and connectionist models for decision support in fishing catches, Environ. Model. Softw. 22 (2007) 862–870, http://dx.doi.org/10.1016/j.envsoft.2006.05.017.
[40] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[41] G. Zhong, L.-N. Wang, X. Ling, J. Dong, An overview on data representation learning: From traditional feature learning to recent deep learning, J. Financ. Data Sci. 2 (4) (2016) 265–278, http://dx.doi.org/10.1016/j.jfds.2017.05.001.
[42] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958, http://dx.doi.org/10.1214/12-AOS1000, arXiv:1102.4807.
[44] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015, http://dx.doi.org/10.1007/s13398-014-0173-7.2, arXiv:1502.03167.
[45] G. Hinton, Overview of mini-batch gradient descent, http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf (Online; accessed on 22.12.17).
[46] F. Chollet, Deep Learning with Python, Manning, 2017.
[47] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012) 1–9, http://dx.doi.org/10.1016/j.protcy.2014.09.007, arXiv:1102.0183.
[48] G. J. Scott, M. R. England, W. A. Starms, R. A. Marcum, C. H. Davis, Training deep convolutional neural networks for land-cover classification of high resolution imagery, IEEE Geosci. Remote Sens. Lett. 14 (4) (2017) 549–553.
[49] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On Calibration of Modern Neural Networks, CoRR abs/1706.04599. arXiv:1706.04599.
[50] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognit. Lett. 30 (1) (2009) 27–38, http://dx.doi.org/10.1016/j.patrec.2008.08.010.
[51] B. Zadrozny, C. Elkan, Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers, in: Proceedings of the Eighteenth International Conference on Machine Learning, Morgan Kaufmann, 2001, pp. 609–616.
[52] Y.-J. Zhang, A summary of recent progresses for segmentation evaluation, Adv. Image Video Segmentation (2006) 423–440.
[53] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284, http://dx.doi.org/10.1109/TKDE.2008.239, arXiv:1011.1669v3.
[54] S. Muruganandham, Semantic Segmentation of Satellite Images using Deep Learning (Master's thesis), Czech Technical University in Prague & Luleå University of Technology, 2016, p. 94.
[55] CLUSTEVAL, Integrative Clustering Evaluation Framework - Jaccard index, https://clusteval.sdu.dk/1/clustering_quality_measures/12 (Online; accessed on 18.11.18).
[56] V. Mnih, Machine Learning for Aerial Image Labeling (Ph.D. thesis), 2013, p. 109.
[57] C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer International Publishing, 2018.
[58] NVIDIA, NVIDIA - CUDA zone, https://developer.nvidia.com/cuda-zone (Online; accessed on 07.03.18).
[59] R.T. Fielding, Architectural Styles and the Design of Network-based Software Architectures (Ph.D. thesis), University of California, Irvine, 2000, p. AAI9980887.
[60] Node.js, Node.js, https://nodejs.org (Online; accessed on 08.08.18).
[61] PostgreSQL, PostgreSQL, https://www.postgresql.org (Online; accessed on 08.08.18).