



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

## Segmentation of Oil Spills on Side-Looking Airborne Radar imagery with Autoencoders

**Citation for published version:**

Gallego, A-J, Gil, P, Pertusa, A & Fisher, R 2018, 'Segmentation of Oil Spills on Side-Looking Airborne Radar imagery with Autoencoders', *Sensors*, vol. 18. <https://doi.org/10.3390/s18030797>

**Digital Object Identifier (DOI):**

[10.3390/s18030797](https://doi.org/10.3390/s18030797)

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Peer reviewed version

**Published In:**

Sensors

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



Article

# Segmentation of Oil Spills on Side-Looking Airborne Radar imagery with Autoencoders <sup>§</sup>

Antonio-Javier Gallego <sup>1,3\*</sup>, Pablo Gil <sup>2,3</sup>, Antonio Pertusa <sup>1,3</sup>, and Robert B. Fisher <sup>4</sup>

<sup>1</sup> Pattern Recognition and Artificial Intelligence Group, Department of Software and Computing Systems, University of Alicante; {jgallego, pertusa}@dlsi.ua.es

<sup>2</sup> Automation, Robotics and Computer Vision Group, Department of Physics, Systems Engineering and Signal Theory, University of Alicante; pablo.gil@ua.es

<sup>3</sup> Computer Science Research Institute, University of Alicante, Alicante, Spain

<sup>4</sup> School of Informatics, University of Edinburgh, Edinburgh, UK; rbf@inf.ed.ac.uk

\* Correspondence: antonio.gallego@ua.es; Tel.: +349-65-903772

Academic Editor: name

Version February 28, 2018 submitted to *Sensors*; Typeset by L<sup>A</sup>T<sub>E</sub>X using class file mdpi.cls

**Abstract:** In this work, we use deep neural autoencoders to segment oil spills from Side-Looking Airborne Radar (SLAR) imagery. Synthetic Aperture Radar (SAR) has been much exploited for ocean surface monitoring, especially for oil pollution detection, but few approaches in the literature use SLAR. Our sensor consists of two SAR antennas mounted on an aircraft, enabling a quicker response than satellite sensors for emergency services when an oil spill occurs. Experiments on TERMA radar were carried out to detect oil spills on Spanish coasts using deep selectional autoencoders and RED-nets. Different configurations of these networks were extensively evaluated and the best topology significantly outperformed previous approaches, correctly detecting 100% of the spills and obtaining an  $F_1$  score of 93.01% at the pixel level. The proposed autoencoders perform accurately in SLAR imagery that has artifacts and noise caused by the aircraft maneuvers, in different weather conditions and with the presence of look-alikes due to natural phenomena such as shoals of fish and seaweed.

**Keywords:** Oil spill detection; Side-Looking Airborne Radar; Neural networks; Supervised learning; Radar detection

## 1. Introduction

A quick response from governments is required in situations of marine pollution due to oil spills [1]. When an oil slick is detected, the authorities activate the emergency protocols in order to control the environmental impact and the ecological damage in the sea. The most relevant technologies and spaceborne sensors for oil-spill sensing are described in [2] and [3,4]. CleanSeaNet is an example of a monitoring service of oil spills and vessels provided by the European Maritime Safety Agency (EMSA). Governments use mainly two kinds of sensors to carry out the monitoring of the sea surface: Synthetic Aperture Radar (SAR) installed on satellites (ERS-1/2, JERS-1, Envisat ASAR, RADARSAT-1, RADARSAT-2, COSMO-SkyMed, Sentinel-1, Sentinel-2, ALOS-2, TerraSAR-X among others) as in CleanSeaNet, and Side-Looking Airborne Radar (SLAR) or another airborne miniaturized radar as in [5]. Both sensors can be used for oil slick detection.

The SLAR used in this work is a SAR mounted on aircraft instead of a satellite and it has two radar antennas. SLAR and SAR sensors have some differences as mentioned in [6]. On the one hand, SLAR has a range and resolution smaller than SAR and, consequently, the complexity in the detection is higher due to the lower details in the acquired image. However, SLAR does not depend

30 on the orbit because it is mounted on an aircraft, and therefore it has a better response time than  
31 SAR. As aircraft can modify their altitude and flight path during signal acquisition, SLAR images  
32 have different perspective and scale. In addition, these images have artifacts and noisy areas caused  
33 by the aircraft motion (turns, slips, etc.) and by the location of the two SLAR antennas under the  
34 aircraft wings. These artifacts and types of noise are not present in SAR images in which speckle  
35 (with granular appearance) is the most common noise.

36 The oil-spill detection strategies using SAR can be categorized into two groups. The first  
37 contains all the approaches that use the raw signals of the radar as well as polarimetric (PolSAR) or  
38 interferometric features (InSAR), and so forth to discriminate the oil slicks [7,8]. The second includes  
39 the methods that use intensity images as a representation of the backscattering coefficient of the signal  
40 [9,10]. In addition, some works such as [11] combine image and polarimetric features extracted from  
41 oil spills and look-alikes in order to discriminate between both targets.

42 In the state of the art there are many works which address oil spill detection using machine  
43 learning techniques. These methods include Tree Forests [12], Support Vector Machines (SVM)  
44 [13,14], Generalized Linear Models (GLM) and Boosting trees among others, as in [15,16] where  
45 both a Bayesian classifier and several evolutionary algorithms were used to select image features  
46 for classifying oil spills and look-alikes. Neural networks have also been used for this task, using as  
47 input different features from radar images characterizing a candidate oil slick [17,18]. The choice of  
48 the classifier architecture is dependent on the problem and when the features are properly selected  
49 there are no significant differences in the results, as shown in [19] with PolSAR data.

50 In some previous works, image processing and computer vision algorithms were used to  
51 automatically extract features and segment regions from radar images. These data can be fed to a  
52 network such as in [20], in which two neural networks were proposed, one to detect dark formations  
53 and another to classify them as oil slicks or look-alikes. In the past, neural network architectures  
54 typically had only three layers (input, hidden and output) as in [21], where a Multilayer Perceptron  
55 (MLP) and the Radial Basis Function (RBF) networks were used. The classification can be performed  
56 at two levels of detail: classification of pixels representing oil slicks when the number of images in  
57 the dataset is small but they have a high resolution [22] or scenarios where the dataset contains many  
58 images [11].

59 More recently, many approaches based on deep learning techniques have been proposed  
60 to increase the success rate in image classification tasks. The main motivation of using deep  
61 convolutional neural networks (CNN) is their ability to extract suitable features for the task at hand, as  
62 it is very difficult to properly select the features that can allow us to discriminate between oil spills and  
63 other natural phenomena due to the similarity of their representations as dark areas in the image. In  
64 this line, Chen *et al.* [23] selected and optimized the PolSAR features reducing the feature dimensions  
65 used as input of the classifier to distinguish oil spill and biogenic look-alikes through layer-wise  
66 unsupervised pre-training. For this task they use Stacked AutoEncoders (SAE, autoencoders with  
67 multiple layers) and Deep Belief Networks (DBN). In addition, Guo *et al.* [24] proposed a CNN to  
68 identify dark areas in SAR images as crude oil (oil slick), plant oil and oil emulsion (both look-alikes),  
69 reaching average success rates of 91% vs the 80% of a traditional neural network. In all these works,  
70 authors used SAR imagery.

71 There are many oil slick detection methods that use SAR imagery as input. However, it is  
72 uncommon to find detection methods using SLAR imagery. Two recent works in this line were  
73 presented in [6] and [25]. The first one is based on traditional image segmentation techniques whereas  
74 the second one uses Recurrent Neural Networks (RNN) to perform the detection. Two decades ago,  
75 Ziemke [26] already proposed a RNN using SAR images for oil spill detection, showing robustness to  
76 variations in both weather conditions and illumination changes.

77 Unlike the previous works using SLAR, we propose an approach that is able to detect oil  
78 slicks even in the presence of look-alikes. Our method, which is an extension of a previous study  
79 presented in [27], is focused on solving the oil-slick region segmentation problem using deep learning

80 techniques, particularly denoising autoencoders using Convolutional Neural Networks as encoder  
81 and decoder functions.

82 The rest of the paper is structured as follows: Section 2 introduces background on autoencoders,  
83 Section 3 presents the proposed method, followed by the dataset description in Section 4, the  
84 evaluation in Section 5, and finally the conclusions and future work in Section 6.

## 85 2. Autoencoder architecture

86 *Autoencoders* were proposed decades ago by Hinton and Zemel [28], and since then they have  
87 been an active research field [29]. Autoencoders consist of feed-forward neural networks trained to  
88 reconstruct their input, that is, the input and the output must be exactly the same. This problem  
89 may seem trivial as their goal is to learn the identity function  $f(x) = x$ , but in practice we impose  
90 some restrictions in order to force it to generate a compressed intermediate representation. This is  
91 achieved by using intermediate layers with a size smaller than the inlet layer. This bottleneck forces  
92 the network to extract the most representative characteristics of the sample that allow its subsequent  
93 reconstruction, thus generating a meaningful intermediate representation.

94 Figure 1 shows a graphical scheme of an autoencoder. This type of network is divided in  
95 two parts, the first part (called the *encoder*) receives the input and creates a latent (or encoded)  
96 representation of it, and the second part (the *decoder*) takes this intermediate representation and  
97 tries to reconstruct the input. Formally speaking, given an input  $x$ , the network must minimize the  
98 divergence  $L(x, g(f(x)))$ , where  $f$  and  $g$  represent the encoder and decoder functions, respectively.  
99 The encoder function provides a meaningful compact representation which might be of great interest  
100 as regards feature learning or dimensionality reduction [30].

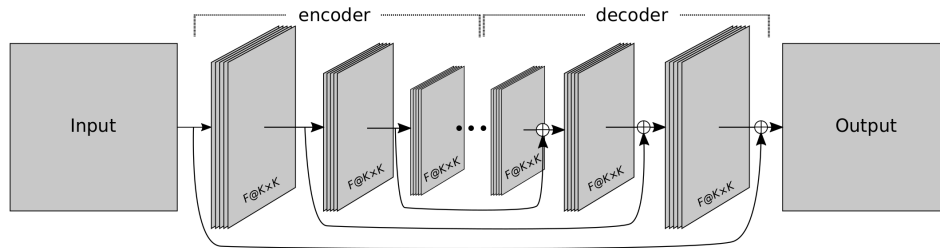
101 Some variations of autoencoders have been proposed in the literature to solve other kind of  
102 problems. For example, *denoising autoencoders* [31] are an extension trained to reconstruct the input  
103  $x$  from a corrupted version (usually generated using Gaussian noise) of it, denoted as  $\hat{x}$ . Thus,  
104 these networks are trained to minimize the divergence  $L(x, g(f(\hat{x})))$ , therefore they not only focus  
105 on copying the input but also on removing the noise [31–33].

106 Autoencoders, and particularly denoising autoencoders, have been successfully used in many  
107 fields such as music, character recognition or medical image segmentation, but in addition they  
108 are currently being used in remote sensing to perform recognition and scene classification. For  
109 example, Zhao *et al.* [34] combined Stacked Autoencoder (SAE) and Extreme Machine Learning (ELM)  
110 techniques for target recognition from raw data of High-Resolution Range Profile (HRRP) acquired  
111 from three different aircraft, achieving a faster time response than other deep learning models. Others  
112 authors such as Kang *et al.* [35] used 23 baseline features and three-patch Local Binary Pattern (LBP)  
113 features that were cascaded and fed into a SAE for recognition of 10-class SAR targets. In addition,  
114 Liang *et al.* [36] presented a classification method based on Stacked Denoising Autoencoders (SDAE)  
115 in order to classify pixels of scenes (acquired from GF-1 high resolution satellite) into forest, grass,  
116 water, crop, mountains, etc.

117 In this paper we propose to use autoencoders which receive as input the signal of SLAR sensors  
118 and return as output the areas detected as oil spills.

## 119 3. Proposed method

120 Based on the idea of *denoising autoencoders*, we use a type of segmentation autoencoder as  
121 proposed in [37] but specifically designed for oil spill detection. In this case, we do not aim to learn  
122 the identity function as autoencoders do, nor an underlying error as in denoising autoencoders, but  
123 rather a codification that maintains only those input pixels that we select as relevant. This is achieved  
124 by modifying the training function so that the input is not mapped identically at the output. Instead,  
125 we train it with a ground truth of the input image pixels that we want to select. From here on, we  
126 will refer to this model as *Selectional AutoEncoder* (SelAE). The SelAE is trained to perform a function



**Figure 1.** Example of a RED-Net topology. The number of layers can change according to the chosen topology. The symbol  $\oplus$  denotes element-wise sum of feature maps.  $F$  represents the number of selected filters and  $(K \times K)$  the size of the kernel.

127 such that  $s : \mathbb{R}^{(w \times h)} \rightarrow [0, 1]^{(w \times h)}$ , or in other words, a binary map over a  $w \times h$  image that preserves  
 128 the input shape and outputs the decision in the range of  $[0, 1]$ .

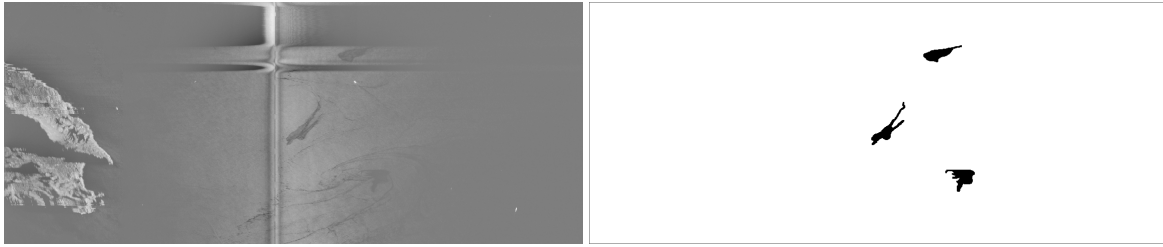
129 Following the autoencoder scheme, the network is divided into encoding and decoding stages,  
 130 where the encoder and decoder functions can be seen as a translator between the input, the  
 131 intermediate representation, and the desired segmentation. The topology of an SelAE can be quite  
 132 varied. However, we have considered only convolutional models because they have been applied  
 133 with great success to many kinds of problems with structured data, such as images, video, or audio,  
 134 demonstrating a performance that is close (or even superior in some cases) to the human level [38].

135 The topology of the network consists of a series of convolutional plus *Max Pooling* layers until  
 136 reaching an intermediate layer in which the encoded representation of the input is attained. It then  
 137 follows a series of convolutional plus upsampling layers that generates the output image with the  
 138 same input size. All layers have *Batch Normalization* [39] and *Dropout* [40], and use *ReLU* as activation  
 139 function [41].

140 The last layer consists of a set of neurons with sigmoid activations that predict a value in the  
 141 range of  $[0, 1]$ . Those pixels whose selection value exceeds the *selectional* level  $\delta$  – which can be seen  
 142 as a threshold – are considered to belong to an oil spill, whereas the others are discarded.

143 In addition, in this work we incorporate into this architecture a series of residual connections  
 144 as proposed in [42]. This type of topology, called RED-Net (Very deep Residual Encoder-Decoder  
 145 Network), includes residual connections from each encoding layer to its analogous decoding layer  
 146 (see Figure 1), which facilitates convergence and leads to better results. Moreover, down-sampling  
 147 is performed by convolutions using stride, instead of resorting to pooling layers. Up-sampling  
 148 is achieved through transposed convolution layers, which perform the inverse operation to a  
 149 convolution, to increase rather than decrease the resolution of the output.

150 We applied a *grid-search* technique [43] in order to find the network architecture with the best  
 151 configuration of layers and hyperparameters (filters of each convolution, the size of the kernel, and  
 152 the dropout value). The results of this experimentation are included in Section 5.4, although we  
 153 anticipate the best topologies for each network in Table 1.



**Figure 2.** Example of a SLAR sequence from our dataset (left) and its corresponding ground truth (right) with the oil spills labeled at the pixel level. The SLAR image shows an island on the left side, a vertical zone of noise caused by junction of the signal from the two antennas of TERMA radar, and two horizontal bands of noise at the top produced by aircraft maneuvers.

**Table 1.** Best architectures found after the grid-search process.

Autoencoder type:	SelAE	RED-Net
Input image size:	256x256 px	384x384 px
Number of layers:	4	6
Residual connections:	No	Yes
Filters per layer:	128	128
Kernel size:	5x5	5x5
Down-sampling:	MaxPool(2x2)	Stride(2x2)
Dropout (%):	0	0
Selectional threshold $\delta$ :	0.5	0.8

### 154 3.1. Training stage

155 As autoencoders are feed-forward networks, they can be trained by using conventional  
 156 optimization algorithms such as gradient descent. In this case, the tuning of the network parameters  
 157 is performed by means of stochastic gradient descent [44] considering the adaptive learning rate  
 158 proposed by Zeiler [45]. The loss function (usually called *reconstruction loss* in autoencoders) can  
 159 be defined as the squared error between the ground truth and the generated output. In this case,  
 160 we use the cross-entropy loss function to perform the optimization of the network weights during a  
 161 maximum of 100 epochs, with a mini-batch size of 8 samples. The training process is stopped if the  
 162 loss does not decrease during 10 epochs.

163 In order to train the network we generated a ground truth marking those pixels of the SLAR  
 164 input images which correspond to oil spills. Figure 2 shows an example of a SLAR sequence (left)  
 165 and its corresponding ground truth (right) with the oil spills labeled in black.

166 In this work, the network is fed with the raw data and the ground truth segmentations, so it  
 167 must learn to discriminate the areas with oil spills from the rest of the data. That is, no preprocessing  
 168 is performed on these images to eliminate the noise, as happens in other approaches such as in [46],  
 169 nor any postprocessing is done to refine the detection.

170 The next section details all the information about the dataset and the SLAR images used.

## 171 4. Dataset

172 In order to validate the effectiveness of the proposed method, we used a dataset containing  
 173 38 flight sequences supplied by the Spanish Maritime Safety and Rescue Agency (SASEMAR).  
 174 SASEMAR is the public authority responsible for monitoring the Exclusive Economic Zones (EEZ)  
 175 in Spain and its procedures are based on reports from the European Maritime Safety Agency (EMSA).

176 The data provided by the SLAR sensor of each of these sequences was digitized in images with a  
177 resolution of  $1,150 \times 481$  pixels.

178 The SLAR samples were acquired by a TERMA SLAR-9000 mounted on a variant of the  
179 EADS-CASA CN-235-300 aircraft for search-and-rescue missions (see Figure 3). This aircraft model  
180 reaches a maximum cruise speed of 236 kn, a flight range of 1,565 nmi and around 2,700 nmi with  
181 and without payload, respectively. Its flight endurance is close to 9 hours. The SLAR samples are  
182 digitalized as 8-bit integers due to the constraints of the monitoring equipment installed on the  
183 aircraft. Our autoencoder architecture uses as input these SLAR images in the same format in which  
184 they were generated by the TERMA software.

185 The dataset was captured by the aircraft on Spanish coasts at an approximate average altitude  
186 of 3,271 feet (although the most common altitude for our missions was around 4,550 feet) and with a  
187 wind speed ranging between 0 and 32 kn, the most usual being 14 kn.



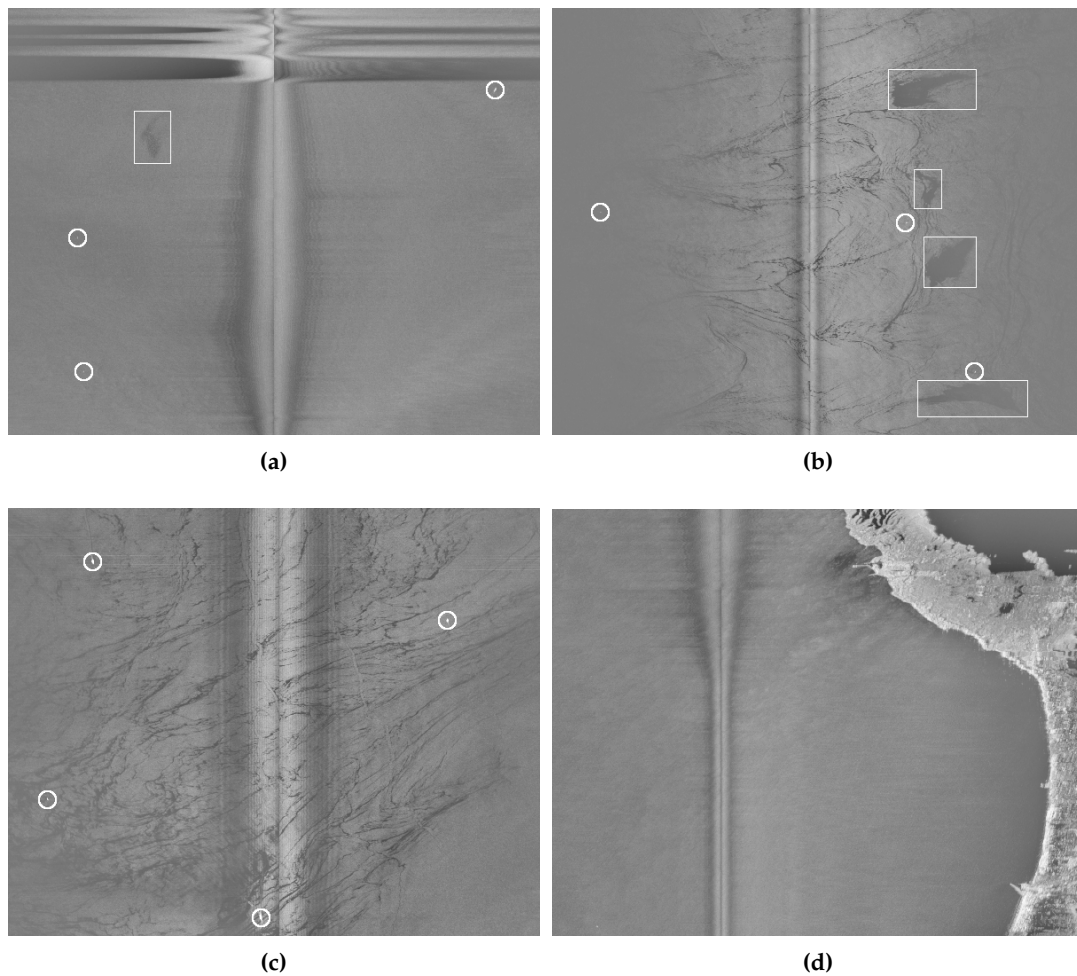
**Figure 3.** SASEMAR 102 (Variant of CN-235-300 aircraft model for search-and-rescue missions) used to obtain the SLAR sequences, manufactured by EADS-CASA.

188 As stated before, for the ground truth we used a binary mask for each SLAR image, delimiting  
189 the pixels corresponding to oil spills. It is important to note that this labeling is performed at a pixel  
190 level since the goal is to evaluate both the detection and the precise location of the spills. This way  
191 we can provide relevant information such as the position, the size and the shape of oil slicks in order  
192 to track them.

193 Figure 4 shows four examples of SLAR images from our dataset. They contain several oil spills  
194 (marked with a bounding box in Figures 4a and 4b), along with other types of artifacts such as boats  
195 (small bright points), coast (Figure 4d), look-alikes, and noise. Figures 4b and 4c contain many  
196 examples of look-alikes, with elongated shapes that are very similar to those of actual spills. All  
197 figures show a central band of noise, which is produced by the union of the information from the two  
198 SLAR sensors. In addition, the upper part of Figures 4a and 4d shows the noise generated by turning  
199 maneuvers of the airplane and the effect produced when the aircraft changes its altitude, respectively.

200 From the 38 flight sequences, 22 contain examples of oil spills and 4 of look-alikes. Within these  
201 examples, the spots only represent 0.32% of the pixels in the image, which creates a very unbalanced  
202 dataset. To evaluate the method properly in the presence of unbalanced data we use the  $F_1$  and in  
203 addition other metrics described in section 5.1.

204 In all the experiments we used an  $n$ -fold cross validation (with  $n = 5$ ), which yields a better  
205 Monte-Carlo estimate than when solely performing the tests with a single random partition [47]. Our  
206 dataset was consequently divided into  $n$  mutually exclusive sub-sets, using the data of each flight  
207 sequence only in one partition and maintaining the percentage of samples for each class. For each  
208 fold, we used one of the partitions for test (20% of the samples) and the rest for training (80%).



**Figure 4.** Examples of SLAR images from our dataset showing oil spills (marked with a bounding box), ships (small bright points marked with circles), look-alikes (elongated shapes in figures (b) 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 (a)), and an example of coast (on the right side of figure (d)). Figures (c) and (d) do not contain any example of oil spills, however they have other artifacts that can lead to confusion.

209 For tuning the hyperparameters (see Section 5.4) the training partition was divided into two,  
 210 assigning 10% of these samples for validation and the rest (70%) for training. The classifier was  
 211 trained and evaluated  $n$  times using these sets, after which the average results plus the standard  
 212 deviation  $\sigma$  were reported.

## 213 5. Evaluation

214 This section shows the experiments performed. First, we describe the metrics used for the  
 215 evaluation, followed by the augmentation methodology and the type of normalization applied. Then  
 216 we present the best hyperparameters found by the grid-search process and finally, the results obtained  
 217 by the proposed method.



218 The following experiments were made on an SGI ICE XA system<sup>1</sup> with two 2.1 GHz, 18-core  
 219 Intel(R) Xeon E5-2695 (Broadwell) and 256GB RAM. The computational resources from this machine  
 220 were mainly exploited to parallelize the grid-search process in order to explore several network  
 221 configurations.

### 222 5.1. Evaluation metrics

223 Three evaluation metrics widely used for this kind of tasks were chosen to evaluate the  
 224 performance of the proposed method: Precision, Recall, and  $F_1$ , which can be defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

$$F_1 = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FN} + \text{FP}} \quad (3)$$

225 where TP (True Positives) denotes the number of correctly detected targets (pixels), FN (False  
 226 Negatives) the number of non-detected or missed targets, and FP (False Positives or false alarms)  
 227 the number of incorrectly detected targets.

228 It should be noted that the  $F_1$  metric is suitable for unbalanced datasets but it is not the most  
 229 adequate for this task since it measures the precision of the results at the pixel level but not whether  
 230 the algorithm has detected the spill or not. For this reason, we also use the Intersection over Union  
 231 (IoU) for evaluation, measuring whether the algorithm correctly detects all the spills present in the  
 232 image and also how well it detects their size and location.

233 In order to calculate the IoU, we map each object proposal ( $p$ ) to the ground-truth ( $g$ ) bounding  
 234 box with which it has a maximum IoU overlap. Bounding boxes are calculated to include the groups  
 235 of pixels (or *blobs*) marked as 1 in the network output after applying the selectional threshold or in  
 236 the ground-truth images. A detection is considered as TP if the area of overlap ( $a_o$ ) ratio between the  
 237 predicted bounding box ( $B_p$ ) and the ground-truth bounding box ( $B_g$ ) exceeds a certain threshold ( $\lambda$ )  
 238 according to the following equation:

$$a_o = \frac{\text{area}(B_p \cap B_g)}{\text{area}(B_p \cup B_g)} \quad (4)$$

$$TP = a_o > \lambda$$

239 where  $\text{area}(B_p \cap B_g)$  depicts the intersection between the object proposal and the ground truth  
 240 bounding box, and  $\text{area}(B_p \cup B_g)$  depicts its union. By convention, we use a threshold of  $\lambda = 0.5$   
 241 to set a TP candidate.

### 242 5.2. Normalization

243 Initially we conducted an experiment to determine the best type of normalization for the task  
 244 at hand. The literature cites different ways to normalize the data to feed a network [48,49], but the  
 245 most appropriate technique depends on the particular problem. The most common normalization  
 246 methods are:

---

<sup>1</sup> Cirrus UK National Tier-2 HPC Service at EPCC (<http://www.cirrus.ac.uk>).

$$Z_{standard} = \frac{M - \text{mean}(M)}{\text{std}(M)} \quad (5)$$

$$Z_{min-max} = \frac{M - \min(M)}{\max(M) - \min(M)} \quad (6)$$

$$Z_{mean} = M - \text{mean}(M) \quad (7)$$

$$Z_{norm} = M/255 \quad (8)$$

247 where  $M$  is the input matrix containing the raw image pixels from the training set. For the  
 248 normalization of the test set we used the same mean, deviation, max, and min values calculated  
 249 for the training set. It is also important to note that the range of values obtained depends on the  
 250 equation used, however this is not an issue since the configuration of the network allows it, and as  
 251 stated before, this can lead to a better result.

252 We evaluated these types of normalization on the two networks, including the option of not  
 253 normalizing the data. For this we considered a base configuration (with 32 filters per layer, a kernel  
 254 size of  $3 \times 3$ , a dropout of 0.25, and a *selectional* threshold  $\delta$  of 0.5), and then we varied the input size  
 255 (subsampling the input images to 128x128px and 256x256px) and the number of hidden layers of each  
 256 network (from 4 to 8), in order to obtain an statistically significant average result. The networks were  
 257 trained using a data augmentation of 20 (see Section 5.3) on the training set, and for the evaluation  
 258 we used the validation set.

259 The results of this experiment (in terms of  $F_1$ , see Equation 3) are shown in Table 2, where each  
 260 cell shows the average of 30 experiments (6 network configurations per 5 folds). As can be seen, the  
 261 best  $F_1$  for the two types of networks are obtained using the standard normalization, followed by  
 262 the mean norm. The type of data normalization considerably affects the result obtained, since the  
 263 differences in some cases reach up to 25%. For this reason, in the following experiments we use the  
 264 standard normalization.

**Table 2.** Average  $F_1$  (%) plus  $\sigma$  when applying different types of normalization on the input data, and without normalization.

	None	$Z_{Standard}$	$Z_{min-max}$	$Z_{mean}$	$Z_{norm}$
<b>SelAE</b>	54.33 $\pm$ 2.23	<b>70.02</b> $\pm$ 1.26	44.65 $\pm$ 3.14	69.84 $\pm$ 1.67	44.10 $\pm$ 3.57
<b>RED-Net</b>	65.25 $\pm$ 1.97	<b>75.12</b> $\pm$ 1.07	53.66 $\pm$ 2.75	74.91 $\pm$ 1.35	59.67 $\pm$ 2.91

### 265 5.3. Data augmentation

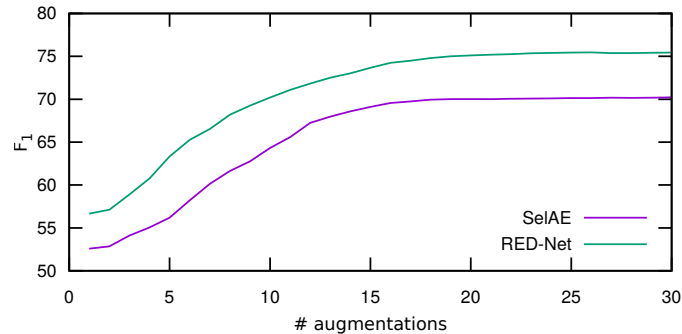
266 Data augmentation is applied in order to artificially increase the size of the training set [49,50].  
 267 As the experimental results show, augmentation systematically improves the accuracy.

268 To this end, we randomly applied different types of transformations on the original images,  
 269 including horizontal and vertical flips, zoom (in the range [0.5, 1.5] times the size of the image),  
 270 rotations (in the range  $[-10^\circ, 10^\circ]$ ), and shear (between  $[-0.2^\circ, 0.2^\circ]$ ).

271 In order to evaluate the improvement obtained with this augmentation process, we carried out  
 272 an experiment in which we gradually increased the number of random transformations applied to  
 273 each image from our training set, and evaluated it using the validation set. As before, we performed  
 274 this experiment for both architectures fixing the configuration to 32 filters per layer, a kernel size of  $3 \times$   
 275  $3$ , a dropout of 0.25, and a *selectional* threshold  $\delta$  of 0.5, and only varying the input size (subsampling  
 276 the input images to 128x128px and 256x256px) and the number of hidden layers of each network  
 277 (from 4 to 8). The input data was normalized using standard normalization.

278 Figure 5 shows the average results of such experiment, where the horizontal axis indicates the  
 279 augmentation size and the vertical axis the  $F_1$  obtained. As can be seen, for the two models evaluated

280 the highest improvement is obtained at the beginning, after which the results begin to stabilize and  
 281 stop improving after 20 augmentations. For this reason, in the following we set to this value the  
 282 number of augmentations applied to each image.



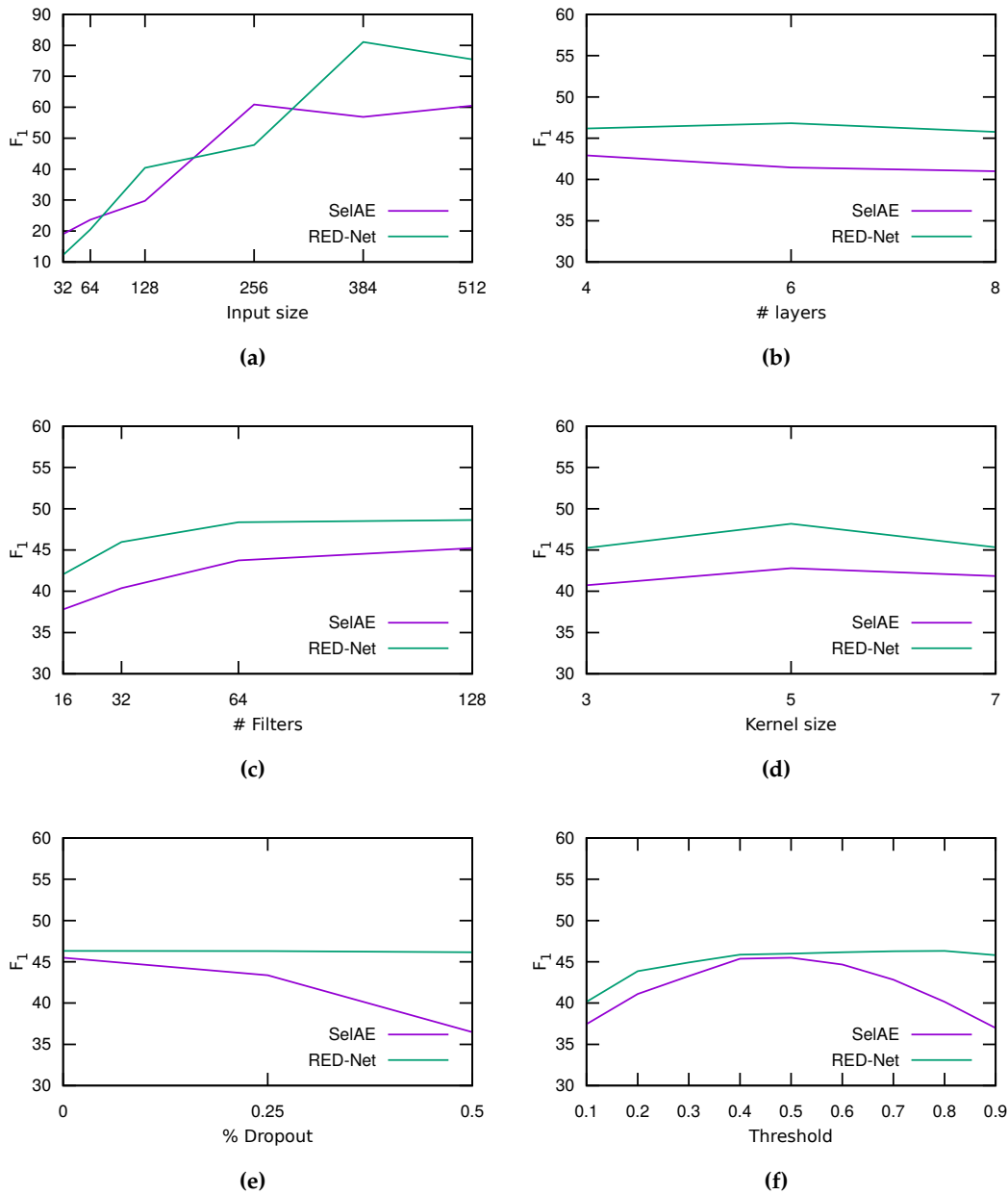
**Figure 5.** Average results of the data augmentation process. The horizontal axis represents the number of augmentations and the vertical axis the average  $F_1$  (in percentage) obtained for each of the networks.

#### 283 5.4. Hyperparameters evaluation

284 In order to select the best hyperparameters for the two types of CNN evaluated, we have  
 285 performed a *grid-search* [43] using the training and validation sets. The configurations evaluated  
 286 include variations in the network input size (from 32px to 512px per side), in the number of layers  
 287 (from 4 to 8), the number of filters (between 16 and 128), the kernel size (between 3 and 7), the  
 288 percentage of dropout (from 0 to 0.5), and the *selectional* threshold  $\delta$  (between 0 and 1). Overall,  
 289 6,480 experiments were made, using 1,296 configurations  $\times$  5 folds. In all cases we applied an  
 290 augmentation of 20 and the standard normalization.

291 Figure 6 shows the results of this experiment. The average  $F_1$  when varying the input size is  
 292 shown in Figure 6a. As can be seen, larger inputs are beneficial for this task. The SelAE architecture  
 293 obtains the higher  $F_1$  with a 256x256px size, whereas the most suitable size for RED-Net is 384x384px.  
 294 Figure 6b shows the results when varying the number of layers. The SelAE architecture obtains the  
 295 best  $F_1$  with four layers, whereas RED-Net requires 6 layers. This may happen because pooling layers  
 296 lose information for the reconstruction, whereas RED-Net mitigates this loss through residual layers.  
 297 Figure 6c shows the average  $F_1$  obtained when varying the number of filters per layer. Using more  
 298 filters increases the  $F_1$ , and this improvement is noticeable from 16 until 64 filters, only increasing  
 299 marginally with 128 filters. Figure 6d shows the average  $F_1$  for the three kernel sizes evaluated, and  
 300 both architectures obtained the best results with 5x5 filters. Figure 6e shows the average  $F_1$  obtained  
 301 by varying the dropout percentage applied to each layer. The best result for both architectures in this  
 302 experiment was obtained without using dropout. The RED-Net results remain stable but they lightly  
 303 worsen when increasing the dropout, whereas with SelAE the  $F_1$  is significantly lower when dropout  
 304 grows. Finally, Figure 6f shows the result by varying the *selectional* threshold  $\delta$ . RED-Net remains  
 305 fairly stable to changes in this value, obtaining its maximum for a threshold of 0.8. SelAE seems to be  
 306 more affected by changes, obtaining better results with an intermediate threshold of 0.5.

307 In conclusion, the final architecture chosen for each network is with 128 filters with 5x5 size  
 308 and without dropout. The SelAE uses an input size of 256x256px, 4 layers, and a threshold of 0.5,  
 309 whereas RED-Net uses 384x384px with 6 layers and a threshold of 0.8. Table 1 shows a summary of  
 310 the topologies that were eventually chosen.



**Figure 6.** Average  $F_1$  (%) of the grid-search process when varying (a) the input image size, (b) the number of layers, (c) the number of filters per layer, (d) the kernel size of the convolutional filters, (e) the percentage of dropout, and (f) the *selectional* threshold  $\delta$ .

### 311 5.5. Results

312 Once the best configuration and parameter settings for each network were selected, we evaluated  
 313 the results using the different metrics. Moreover, we compared these results with three *state-of-the-art*  
 314 methods for *oil slick* segmentation in SLAR images:

- 315 • “*Graph-based method*” [51]: It is an adaptation for SLAR images of the method proposed in [52].  
 316 It uses progressive intensity gradients for extracting regions with variable intensity distribution.
- 317 • “*JSEG method*” [51]: It is also an extension to SLAR images of a previous work [53], where the  
 318 input image is quantized according to the number of regions to be segmented. Pixel intensities

319 are replaced by the quantized label building a class-map called J-image. Later, a region-growing  
 320 technique is used to segment the J-image.

- 321 • “*Segmentation guided by saliency maps (SegSM)*” [6]: It first applies a pre-process of the noise  
 322 caused by aircraft movements using Gabor filters and Hough Transform. Then, the saliency  
 323 map is computed and used as seeds of a region-growing process that segments the regions that  
 324 represent oil slicks.

325 Details regarding the implementation and the parameters used in these methods can be found  
 326 in the corresponding references.

327 Table 3 shows the final result obtained with the proposed approach as well as the comparison  
 328 with the state-of-the-art methods using the test set for the evaluation. It should be noted that the test  
 329 set had not been used in previous experiments to avoid adjusting the network architecture for this  
 330 set.

331 The best results were obtained in all cases using the RED-Net architecture, which shows a higher  
 332  $F_1$  (see Equation 3) for all the tested images. On the one hand, the best RED-Net configuration  
 333 increases up to 3.7 % the  $F_1$  of the SelAE autoencoder, and between a 37 % and a 64 % the other  
 334 methods of the state of the art. The SegSM method has a high precision and a low recall, which  
 335 indicates that it accurately detects some parts of the spills but producing many FN. On the other  
 336 hand, both JSEG and Graph-based methods have a high recall and a low precision, since in this case  
 337 they are producing many false positives. The proposed method obtains a more balanced result in the  
 338 detection of oil-spill pixels. This fact can be confirmed by looking at the IoU metric (see Equation  
 339 4), where RED-Net also improved significantly the results with respect to the other methods, which  
 340 indicates a better precision in the detection of the shape and the position of the oil slicks.

**Table 3.** Evaluation results including the standard deviation for the two architectures using the chosen parameters after grid-search.

Model	Precision	Recall	$F_1$	IoU
<b>Graph-based</b>	32.99 ±1.62	97.25 ±0.33	48.28 ±1.87	32.55 ±0.16
<b>JSEG</b>	17.04 ±0.32	92.58 ±0.25	28.73 ±0.46	16.50 ±0.35
<b>SegSM</b>	98.54 ±0.27	39.55 ±1.21	55.78 ±1.18	87.33 ±0.51
<b>SelAE</b>	89.64 ±0.95	88.99 ±0.91	89.31 ±0.93	92.14 ±7.21
<b>RED-Net</b>	93.12 ±0.86	92.92 ±0.84	<b>93.01 ±0.85</b>	100.00 ±0.00

341 Figure 7 shows a graphic representation of the results obtained with the best approach, i.e. the  
 342 RED-Net model. The first column of images shows the original input SLAR images (oil spills are  
 343 marked with a bounding box), and the second column shows the prediction of the network. In the  
 344 images of the second column, white and black areas depict correct detections of sea and oil spills,  
 345 respectively, and red and blue pixels depict FP and FN of oil spills, respectively.

346 These figures help to visualize the accuracy of the proposed model and to understand where  
 347 the errors of each target class occur. As can be seen, wrong detections are typically made only at the  
 348 contours of the oil spills.

349 Figure 7a shows that the proposed method correctly detects the spill even in the presence of noise  
 350 due to look-alikes (biological origin and weather conditions). In Figure 7c we can see a larger spill  
 351 produced by a ship emptying its bilge tanks. This spill is correctly detected and there are only few  
 352 mistakes at the edges. Figure 7e contains coast, but the method do not miss and it correctly detects  
 353 just one small spill at the center. Figure 7g also shows a coast section in the upper-right part, and in  
 354 this case the image contains an airplane turn. In this example, even when the spill is located at the  
 355 center of the noise, the method is able to correctly perform the detection. Finally, the last example in

356 Figure 7i shows an image with high noise (caused by aircraft movements), including coast at the left  
357 side and without any spill. As can be seen, the method correctly concludes that the image does not  
358 contain any spill.

## 359 6. Conclusions and Future Work

360 In this work, we propose to use deep convolutional autoencoders for the detection of oil spills  
361 from SLAR imagery. Two different network topologies have been analyzed, conducting extensive  
362 experiments to get the best type of data normalization, to know the impact of data augmentation on  
363 the results, and to obtain the most suitable hyperparameters for both networks.

364 A dataset with a total of 28 flight sequences was gathered on Spanish coasts using TERMA SLAR  
365 radar, labeling the ground-truth in order to train both selectional autoencoders and RED-Nets. It is  
366 composed of oil spills acquired in a wide variety of sea conditions dependent on weather (i.e. wind  
367 speed) and geographic location as well as of flight conditions such as altitude and type of motion.

368 The proposed approach is able to segment accurately oil slicks despite the presence of other dark  
369 spots such as biogenic look-alikes, low wind which also introduces a lot of look-alikes, and noise  
370 due to bad radar [measurements](#) caused by the aircraft maneuvers. Results show that the RED-Net  
371 achieves an excellent  $F_1$  of 93.01% when evaluating the obtained segmentation at the pixel level. In  
372 addition, by analyzing the precision of the regions found using the Intersection over Union (IoU)  
373 metric, the proposed method correctly detects the 100% of the oil spills, even in the presence of  
374 artifacts and noise caused by the aircraft maneuvers, in different weather conditions and with the  
375 presence of look-alikes.

376 Future work includes increasing the dataset size by adding more labeled samples from  
377 additional missions. In addition, Generative Adversarial Networks (GAN) such as Pix2Pix [54] could  
378 be used to deal with a reduced dataset by generating synthetic samples. [Also, the detected oil slick  
379 locations could be used to initialize oil spill models for better oil spill prediction and response \[55\].  
380 A study correlating the  \$F\_1\$  score with the wind speed or weather conditions could also be useful to  
381 understand to what extent the effectiveness of the proposed method depends on these factors.](#)

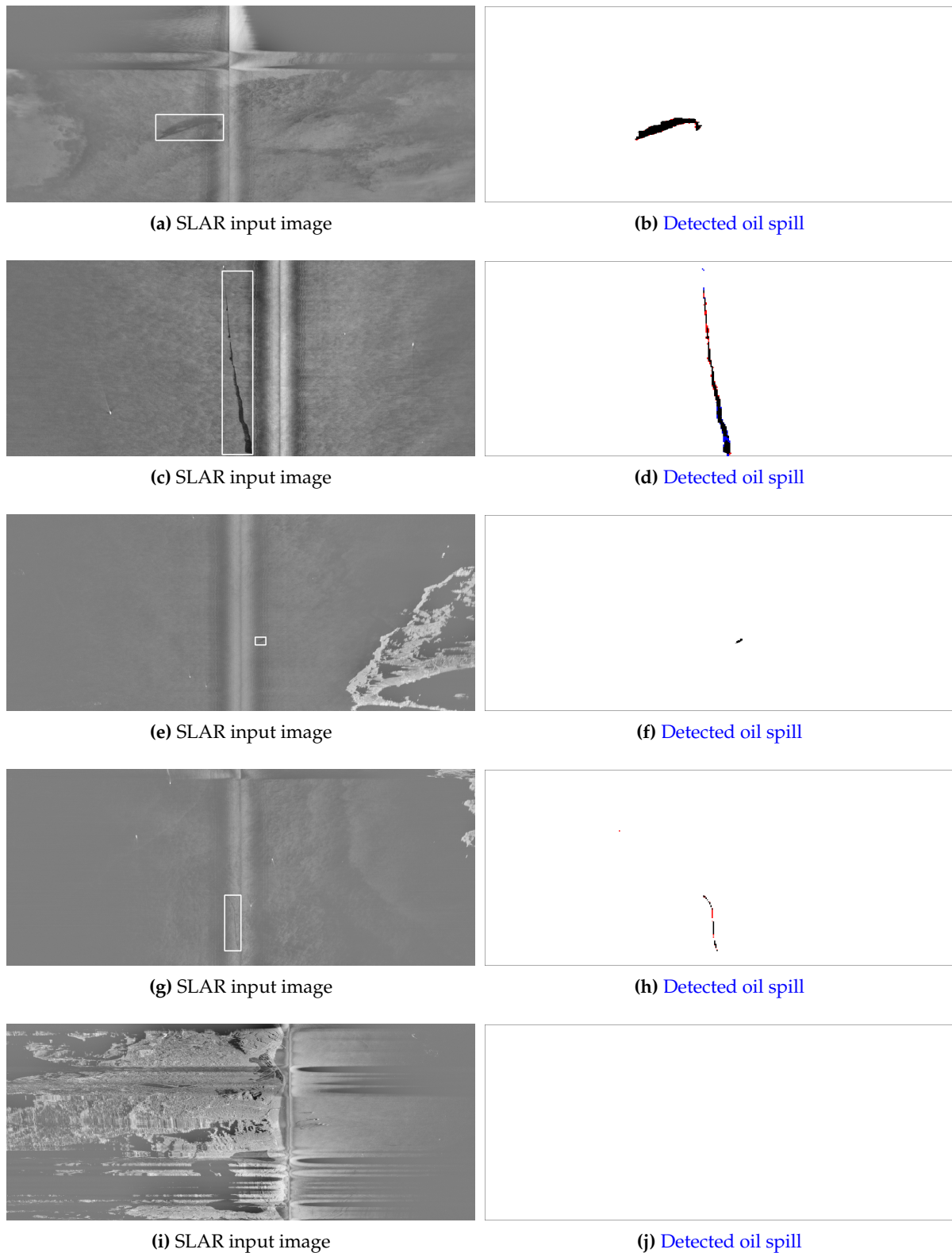
382 **Acknowledgments:** This work was funded by both the Spanish Government's Ministry of Economy, Industry  
383 and Competitiveness and Babcock MCS Spain through the RTC-2014-1863-8 and INAER4-14Y(IDI-20141234)  
384 projects as well as by the grant number 730897 under the HPC-EUROPA3 project, a Research and Innovation  
385 Action supported by the European Commission's Horizon 2020 programme.

386 **Author Contributions:** A-J.G. conceived the proposed method and performed the experiments; R.B.F. helped  
387 with the design of experiment distribution on supercomputer; All authors analyzed the data and results; A-J.G.  
388 and P.G. wrote the paper; A.P. and R.B.F. reviewed and helped with clarifying the paper.

389 **Conflicts of Interest:** The authors declare no conflict of interest.

## 390 Bibliography

- 391 1. Liu, Y.; Macfadyen, A.; Ji, Z.G.; Weisberg, R. *Monitoring and Modeling the Deepwater Horizon Oil Spill: A  
392 Record-Breaking Enterprise*; American Geophysical Union: Washington, D.C., 2011.
- 393 2. Leifer, I.; Lehr, W.J.; Simecek-Beatty, D.; Bradley, E.; Clark, R.; Dennison, P.; Hu, Y.; Matheson, S.; Jones,  
394 C.E.; Holt, B.; Reif, M.; Roberts, D.A.; Svejkovsky, J.; Swayze, G.; Wozencraft, J. State of the art satellite  
395 and airborne marine oil spill remote sensing: Application to the {BP} Deepwater Horizon oil spill. *Remote  
396 Sensing of Environment* **2012**, *124*, 185 – 209.
- 397 3. Fingas, M.; Brown, C. Review of oil spill remote sensing. *Marine Pollution Bulletin* **2014**, *83*, 9–23.
- 398 4. Fingas, M.; Brown, C. A review of oil spill remote sensing. *Sensors* **2018**, *18*, 91:1–18.
- 399 5. Jones, C.; Minchew, B. and Holt, B.; Hensley, S., Studies of the Deepwater Horizon Oil Spill With the  
400 UAVSAR Radar. In *Monitoring and Modeling the Deepwater Horizon Oil Spill: A Record Breaking Enterprise*;  
401 American Geophysical Union: Washington, D.C., 2011; Vol. 195, pp. 33–50.
- 402 6. Gil, P.; Alacid, B. Oil Spill Detection in Terma-Side-Looking Airborne Radar Images Using Image Features  
403 and Region Segmentation. *Sensors* **2018**, *18*, 151:1–15.



**Figure 7.** Results of processing five SLAR input images. The first column shows the original SLAR images (oil spills are marked with a bounding box), and the second column shows the detection results. White and black areas depict correct detections of sea and oil spills, respectively, and red and blue pixels (hard to see because they are few) depict FP and FN of oil spills, respectively.

- 404 7. Skrunes, S.; Brekke, C.; Eltoft, T. Characterization of Marine Surface Slicks by Radarsat-2  
405 Multipolarization Features. *IEEE Transactions on Geoscience and Remote Sensing* **2014**, *52*, 5302–5319.
- 406 8. Salberg, A.; Rudjord, O.; Solberg, H. Oil spill detection in hybrid-polarimetric SAR images. *IEEE*  
407 *Transactions on Geoscience and Remote Sensing* **2014**, *52*, 6521–6533.
- 408 9. Brekke, C.; Solberg, A.H.S. Classifiers and Confidence Estimation for Oil Spill Detection in ENVISAT  
409 ASAR Images. *IEEE Geoscience and Remote Sensing Letters* **2008**, *5*, 65–69.
- 410 10. Wu, Y.; He, C.; Liu, Y.; Su, M. A backscattering-suppression-based variational level-set method for  
411 segmentation of SAR oil slick images. *IEEE Journal of Selected Topics in Applied Earth Observations and*  
412 *Remote Sensing* **2017**, *10*, 5485–5494.
- 413 11. Singha, S.; Ressel, R.; Velotto, D.; Lehner, S. A Combination of Traditional and Polarimetric Features  
414 for Oil Spill Detection Using TerraSAR-X. *IEEE Journal of Selected Topics in Applied Earth Observations and*  
415 *Remote Sensing* **2016**, *9*, 4979–4990.
- 416 12. Topouzelis, K.; Psyllos, A. Oil spill feature selection and classification using decision tree forest on SAR  
417 image data. *ISPRS Journal of Photogrammetry and Remote Sensing* **2012**, *68*, 135–143.
- 418 13. Taravat, A.; Oppelt, N. Adaptive Weibull Multiplicative Model and Multilayer Perceptron Neural  
419 Networks for Dark-Spot Detection from SAR Imagery. *Sensors* **2014**, *14*, 22798–22810.
- 420 14. Mera, D.; Bolon-Canedo, V.; Cotos, J.; Alonso-Betanzos, A. On the use of feature selection to improve the  
421 detection of sea oil spills in SAR images. *Computers and Geosciences* **2017**, *100*, 166–178.
- 422 15. Xu, L.; Li, J.; Brenning, A. A comparative study of different classification techniques for marine oil spill  
423 identification using RADARSAT-1 imagery. *Remote Sensing of Environment* **2014**, *141*, 14 – 23.
- 424 16. Chehresa, S.; Amirkhani, A.; Rezaiead, G.; Mosavi, M. Optimum Features Selection for oil Spill Detection  
425 in SAR Image. *Journal of the Indian Society of Remote Sensing* **2016**, *44*, 775–886.
- 426 17. Frate, F.; Petrochi, F.; Lichtenegger, J.; Calabresi, G. Neural networks for oil spill detection using ERS-SAR  
427 data. *IEEE Transactions on Geoscience and Remote Sensing* **2000**, *38*, 2282–2287.
- 428 18. Singha, S.; Bellerby, T.; Trieschmann, O. Satellite oil spill detection using artificial neural networks. *IEEE*  
429 *Journal of selected topixs in applied earth observations and remote sensing* **2013**, *6*, 2355–2363.
- 430 19. Zhang, Y.; Li, Y.; Liang, X.; Tsou, J. Comparison of Oil Spill Classifications using Fully and Compact  
431 Polarimetric SAR Images. *Applied Sciences* **2016**, *7*, 193:1–22.
- 432 20. Topouzelis, K.; Karathanassi, V.; Pavlakis, P.; Rokos, D. Detection and discrimination between oil spills  
433 and look-alike phenomena through neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*  
434 **2007**, *62*, 264–270.
- 435 21. Topouzelis, K.; Karathanassi, V.; Pavlakis, P.; Rokos, D. Dark formation detection using neural networks.  
436 *International Journal of Remote Sensing* **2008**, *29*, 4705–4720.
- 437 22. Taravat, A.; Del Frate, F. Development of band ratioing algorithms and neural networks to detection of  
438 oil spills using Landsat ETM+ data. *EURASIP Journal on Advances in Signal Processing* **2012**, *107*, 1–8.
- 439 23. Chen, G.; Li, Y.; Sun, G.; Zhang, Y. Application of Deep Networks to Oil Spill Detection Using Polarimetric  
440 Synthetic Aperture Radar Images. *Applied Sciences* **2017**, *7*.
- 441 24. Guo, H.; Wu, D.; An, J. Discrimination of Oil Slicks and Lookalikes in Polarimetric SAR Images Using  
442 CNN. *Sensors* **2017**, *17*, 1837:1–20.
- 443 25. Oprea, S.O.; Gil, P.; Mira, D.; Alacid, B. Candidate Oil Spill Detection in SLAR Data - A Recurrent Neural  
444 Network-based Approach. Proceedings of the 6th International Conference on Pattern Recognition  
445 Applications and Methods (ICPRAM), 2017, Vol. 1, pp. 372–377.
- 446 26. Ziemke, T. Radar image segmentation using recurrent artificial neural networks. *Pattern Recognition*  
447 *Letters* **1996**, *17*, 319 – 334.
- 448 27. Alacid, B.; Gallego, A.J.; Gil, P.; Pertusa, A. Oil Slicks Detection in SLAR Images with Autoencoders.  
449 *Proceedings* **2017**, *1*, 820:1–15.
- 450 28. Hinton, G.E.; Zemel, R.S. Autoencoders, Minimum Description Length and Helmholtz Free Energy.  
451 *Advances in Neural Information Processing Systems*, 1994, pp. 3–10.
- 452 29. Baldi, P. Autoencoders, Unsupervised Learning, and Deep Architectures. *JMLR W&CP*, 2012, Vol. 27, pp.  
453 37–49.
- 454 30. Wang, W.; Huang, Y.; Wang, Y.; Wang, L. Generalized Autoencoder: A Neural Network Framework for  
455 Dimensionality Reduction. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)  
456 Workshops, 2014, pp. 490–497.



- 457 31. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked denoising autoencoders: Learning  
458 useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning*  
459 *Research* **2010**, *11*, 3371–3408.
- 460 32. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and Composing Robust Features with  
461 Denoising Autoencoders. Proceedings of the 25th International Conference on Machine Learning; ACM:  
462 New York, NY, USA, 2008; ICML '08, pp. 1096–1103.
- 463 33. Bengio, Y.; Yao, L.; Alain, G.; Vincent, P. Generalized denoising auto-encoders as generative models.  
464 *Advances in Neural Information Processing Systems* **26**, 2013, pp. 899–907.
- 465 34. Zhao, F.; Liu, Y.; Huo, K.; Zhang, S.; Zhang, Z. Radar HRRP Target Recognition Based on Stacked  
466 Autoencoder and Extreme Learning Machine. *Sensors* **2018**, *18*, 173:1–15.
- 467 35. Kang, M.; Ji, K.; Leng, X.; Xing, X.; Zou, H. Synthetic Aperture Radar Target Recognition with Feature  
468 Fusion Based on a Stacked Autoencoder. *Sensors* **2017**, *17*, 192:1–16.
- 469 36. Liang, P.; Shi, W.; Zhang, X. Remote Sensing Image Classification Based on Stacked Denoising  
470 Autoencoder. *Remote Sensing* **2018**, *10*, 16:1–12.
- 471 37. Gallego, A.; Calvo-Zaragoza, J. Staff-line removal with selectional auto-encoders. *Expert Systems with*  
472 *Applications* **2017**, *89*, 138–148.
- 473 38. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
- 474 39. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal  
475 Covariate Shift. *JMLR W&CP* **2015**, *37*.
- 476 40. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to  
477 Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **2014**, *15*, 1929–1958.
- 478 41. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. *Journal of Machine Learning*  
479 *Research (JMLR) W&CP* **2011**, *15*, 315–323.
- 480 42. Mao, X.; Shen, C.; Yang, Y. Image Restoration Using Very Deep Convolutional Encoder-Decoder  
481 Networks with Symmetric Skip Connections. *Advances in Neural Information Processing Systems* **29**:  
482 *Annual Conference on Neural Information Processing Systems 2016*, December 5-10, 2016, Barcelona,  
483 Spain, 2016, pp. 2802–2810.
- 484 43. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning*  
485 *Research* **2012**, *13*, 281–305.
- 486 44. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of*  
487 *COMPSTAT'2010*; Springer, 2010; pp. 177–186.
- 488 45. Zeiler, M.D. ADADELTA: An Adaptive Learning Rate Method. *CoRR* **2012**, *abs/1212.5701*.
- 489 46. Alacid, B.; Gil, P. An approach for SLAR images denoising based on removing regions with low visual  
490 quality for oil spill detection. *SPIE Remote Sensing. International Society for Optics and Photonics*, 2016,  
491 pp. 1000419–1000419.
- 492 47. Kohavi, R. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection.  
493 *Proceedings IJCAI*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1995; Vol. 2, *IJCAI'95*,  
494 pp. 1137–1143.
- 495 48. Shalabi, L.A.; Shaaban, Z.; Kasasbeh, B. Data Mining: A Preprocessing Engine. *Journal of Computer Science*  
496 **2006**, *2*, 735–739.
- 497 49. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural  
498 Networks. *Advances In Neural Information Processing Systems* **2012**, pp. 1–9.
- 499 50. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Return of the Devil in the Details: Delving Deep  
500 into Convolutional Nets. *British Machine Vision Conference*, 2014, pp. 1–11.
- 501 51. Mira, D.; Gil, P. and Alacid, B.; Torres, F. Oil Spill Detection using Segmentation based Approaches.  
502 *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods -*  
503 *Volume 1: ICPRAM,, 2017*, pp. 442–447.
- 504 52. Felzenszwalb, P.; Huttenlocher, D. Efficient Graph-Based Image Segmentation. *International Journal of*  
505 *Computer Vision* **2004**, *59*, 167–181.
- 506 53. Deng, Y.; Manjunath, B.S. Unsupervised segmentation of color-texture regions in images and video. *IEEE*  
507 *Transactions on Pattern Analysis and Machine Intelligence* **2001**, *23*, 800–810.
- 508 54. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial  
509 Networks. *arxiv* **2016**.

510 55. Liu, Y.; Weisberg, R.H.; Hu, C.; Zheng, L. Tracking the Deepwater Horizon Oil Spill: A Modeling  
511 Perspective. *Eos, Transactions American Geophysical Union* **2011**, *92*, 45–46.

512 © 2018 by the authors. Submitted to *Sensors* for possible open access publication under the terms and conditions  
513 of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>)