ISSN print: 1224-5984
ISSN online: 2248-3942

Rom Biotechnol Lett. 2023; 28(4): 4021–4032
doi: 10.25083/rbl/28.4/4021.4032

Received for publication, April, 08,2024
Accepted, June, 06, 2024

Original article

Development of Machine Learning Techniques for the Multiclass Semantic Segmentation of Underwater Images

D. ANITHA[1,*], C. RANI[2], K. SUGANYA DEVI[3], S.P. VALLI[4]

[1]Department of Computer Science and Engineering (CSE), University College of Engineering, Panruti, India.
[2]Department of CSE, Government College of Engineering, Bodinayakkanur, India.
[3]Department of CSE, National Institute of Technology Silchar, Assam, India.
[4]Department of CSE, B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India.

Abstract

The Neptune grass (NG) is an endemic plant of the Mediterranean Sea. Successive studies carried out show a significant decrease in their presence in this sea. Concern for the conservation of NG implies being able to monitor its evolution and thus be able to determine if its presence is increasing or decreasing in seabeds. Nowadays, expensive and limited methods are used to carry out this follow-up, which often also involves manual intervention. The work carried out aims to automatically determine the type of seabed from underwater images, performing semantic segmentation using deep neural networks. In this work, a deep neural network has been implemented to carry out a semantic segmentation of images of the seabed, offering improvements with respect to the techniques currently used to obtain this information. The developed neural network allows for distinguishing in the images of the seabed the areas of NG, rock, and sand. The possibility of identifying other elements present, such as dead NG, both the plant itself and its loose leaves scattered on the bottom, has also been explored in this work.

Keywords Neptune grass, Deep Neural Networks, underwater images, seabed.

✉ *Corresponding author: D. Anitha, Department of Computer Science and Engineering (CSE), University College of Engineering, Panruti, India E-mail: anithacsephd@gmail.com

Introduction

The NG is an aquatic plant endemic to the Mediterranean and of great ecological value. It protects the coast from erosion; it is the base of the ecosystem for a multitude of marine species; it also oxygenates the water and captures CO2, making it an element that mitigates climate change. Despite being a protected species and carrying out actions for its conservation, NG continues to decline due to trawling, boat anchors, pollution, climate change, etc. [1–4]. In order to carry out control and recovery, it is necessary to be able to monitor the evolution of NG meadows. This monitoring is currently being done by:

- Divers: So it is a very slow and expensive task.
- Satellite images: detection problems in deep water.
- SONAR: commonly used to make bathymetric maps.
- Autonomous underwater vehicle (AUV) equipped with sensors.

The work carried out identifies the type of seabed from underwater images. This identification consists of indicating in each pixel of the image what it is; in this case, it will be indicated in each pixel if it is NG, sand, rock, etc. This way of classifying images is called multiclass semantic segmentation. To achieve this semantic segmentation, artificial intelligence (AI) will be used, specifically a deep neural network. It is intended to automate the detection of NG in an AUV, which will make maps of the seabed with which to monitor NG. This work is an evolution of a work [5–8]. The improvement consists of carrying out multiclass semantic segmentation; that is, the previous work only identified in the image if there is NG or not in an area. With this work, it is possible to identify, in addition to NG, other types of elements, such as if the floor is made of sand or rock [9].

Figure 1 shows an example of monoclass semantic segmentation and another of multiclass, where the original image has been colored to identify the different types of background. Figure 1A(a) shows the monoclass classification [10–11], where the green color represents areas of NG and the rest of the uncolored image indicates that it is not NG. Figure 1A(b) is the case of multiclass semantic segmentation carried out in this work. Where, in addition to the green color for the NG, the rock is colored red, the sand is yellow, and what has not been identified has not been colored.

Being able to identify, in addition to the NG, the type of soil can help to see the evolution of the growth or regression of the NG depending on the terrain. Thus, in addition to being able to help take actions in certain areas, it also serves to monitor their results.

Literature Review

In 1989, M. Doherty proposed using a segmentation approach to object detection on sonar images [12]. He observes that the pixels associated with a target do not have the same statistical distribution (in terms of gray levels) as the pixels associated with the seabed. He then proposes to carry out an appropriate thresholding of the images to highlight the echoes. In order to validate the detection, it sets up a search for the shadows associated with these echoes using a set of smoothing and averaging operations. These premises show what will be one of the approaches most commonly used in object detection over the following decades, namely a segmentation of the image into three classes: the echo of a target, the shadow, and the bottom. In 1995, M. Bello introduced random Markov fields in this context and demonstrated that they are suitable for such segmentation [13]. These results push M. Mignotte and C. Collet to deepen this method [14–17]. However, due to the significant calculation times and current computer capacities, it was necessary to wait until 2003 and the study proposed by S. Reed and colleagues to obtain an efficient algorithm [18–19]. Using a priori spatial information (on target sizes and geometric signatures), a detection-oriented Markov field model is then developed to segment the image into these three classes. In 2014, [20] proposed to review this method by applying the graph-cuts method [21] to it to further accelerate the convergence of results. Another approach initiated by B. Calder in [22, 23] proposes to use a stochastic Bayesian model in order to classify each pixel of an image. For this, it carries out Bayesian modelling of the data and uses a Gibbs field to model the targets. The approach is considered robust but very computationally intensive. F. Maussang et al. [24] have also invested in the statistical approach to segment sonar images and, in particular, SAS images [25–26]. These methods are based on the relationship between the mean and standard deviation of the Rayleigh distribution of gray levels in sonar images. By modelling the responses of the background using a Weibull law, they observe that the mean and the standard deviation are linked by a multiplicative constant. However, for echo and shadow areas, the authors note that this relationship is no longer as strict. They then propose to take advantage of this by applying two thresholds in a plan defined by the mean and the standard deviation. These then make it possible to extract the areas of echoes and shadows. Salience detection methods are now commonly used and propose seeing an object as an anomaly in a textured region. They seek to model the differences between a given bottom zone and its vicinity. For example, in [27], L. Linett is based on the fact that background reverberations can be modelled

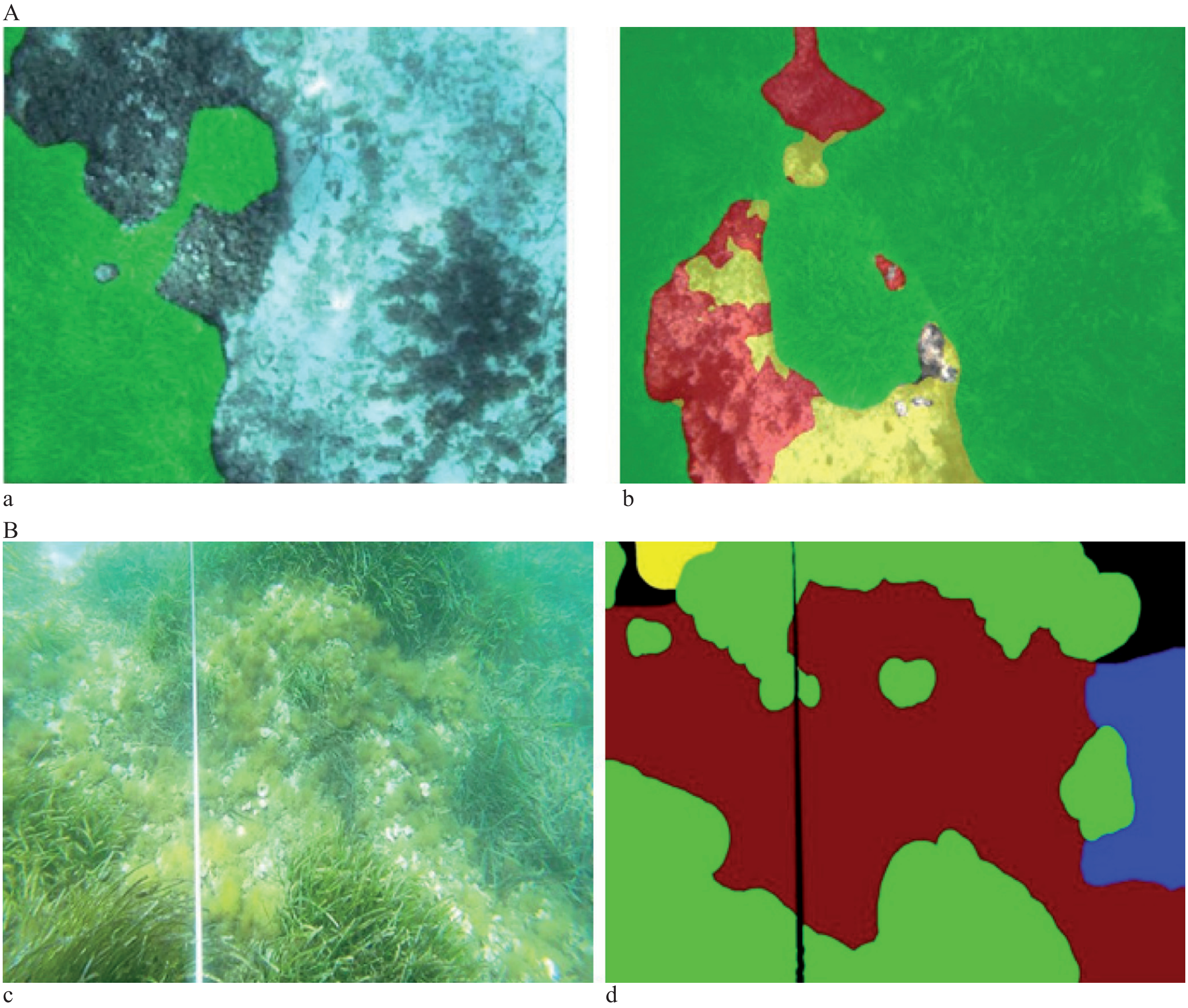


Fig. 1: A. Comparison of monoclass (a) and multiclass (b) semantic segmentation; B. Labeling of images.

by a fractional Brownian process [28], whose distribution is described by its fractal dimension. He then considers that a region whose fractal dimension is different from these neighbours can contain an object. In [29], L. Attalah uses Shannon entropy to detect salient regions. Indeed, he finds that regions containing shadows or echoes have a higher entropy than simple seabed areas. Thus, the author proposes to detect objects via the detection of peaks in the entropy function. In [30], the sonar image is broken down into blocks (of the typical size of an underwater mine, i.e., of the order of a meter), and the author proposes to calculate the correlation, according to the columns, between the different blocks from the same region. According to the author, underwater mines will then have a high correlation value, and a simple threshold would make it possible to detect them. In [30], we propose a salience detection algorithm whose calculations are done through the concept of integral image. The latter makes it possible to achieve detection speeds close to real-time on SAS images, despite their very large dimensions. From a set of sliding windows of varying sizes, the author proposes to estimate zones of echoes and shadows using a cascading architecture. The simplicity and speed of the algorithm are its main assets, which allow it to be one of the few that can be implemented on board AUVs. The classification based on models is entirely based on a priori knowledge that we have about the objects to be detected. This information then makes it possible to create a model for each target that one wishes to highlight. Such approaches are useful for classifying targets on which we do not have many examples (i.e., no learning is possible). In the literature on ATR, many methods have been proposed, and all rely exclusively on model matching; we can cite, for example, [31]. Only the implementations are then different, with a wide variety of algorithms. Some of these approaches propose to use contour-based matching algorithms like those of S. Reed [18]. Its algorithm allows cooperation between active contours to extract the echo and the shadow of the targets jointly in a complex environment. Dempster Shafer's decision theory is then used to classify observed detections against known patterns.

Development of the proposed work

In this work, the necessary neural network has been implemented to achieve the semantic segmentation of the images of the seabed. To achieve this, the Tensor flow libraries have been used, and the programming has been done in Python. The present code is an evolution of a work to detect NG [2], which in turn is an adaptation of another work to detect roads [7]. In addition to the programming part of the neural network, the data for the training sessions had to be prepared and the necessary metrics defined to validate the results and verify the quality of the models obtained.

Data collection

For this work, 302 images obtained from recordings made by cameras mounted on an AUV have been used. The images are of the seabed of the Balearic Islands, especially Mallorca. These images are taken with different lighting conditions and water turbulence; this variety makes the trained network more versatile. Although the original images may have different dimensions, before carrying out the training, they are resized to 480x360, which is the size that the implemented neural network supports.

The images have been divided into two collections:

- 242 images for training (80%).
- 60 images for the tests (20%).

All the images had to be labelled as well. This labelling consists of manually performing a semantic segmentation for each image; in this way, we will have the desired results that the neural network needs to train and validate. To carry out the labelling of an image, a color must be defined for each class that is to be identified. Once the classes are defined, each pixel of the original image must be colored with its corresponding color. Figure 1B shows an example of how an original figure (c) is edited to obtain a labelled figure (d).

The colors that have been defined for each class to be identified in the underwater images are shown in Table 1.

Architecture used

To implement multiclass segmentation, I have based myself on a multilayer neural network, in which the input of the network is an RGB color image to which several transformations are applied, as shown in Figure 2, to achieve multiclass semantic segmentation [2].

Table 1. Correspondence of classes and colors

Name	Description	Color
NG-a	Living NG Kill	🟩
NG-d	dead NG leaves	🟦
Rock	clean rocky bottom	🟥
rock-d	Rocky bottom with algae or other elements	🟫
sandy	clean sand background	🟨
Sand-d	Sandy bottom with algae or other elements	🟫
matte-s	Dead NG clump on sand bottom	🟪
Matte-r	Dead NG clump on rock bottom	🟦
Background	Fund that does not correspond to any of the classes to be classified or that has not been identified	

In Figure 2, the number above the layers represents the dimensions of the feature maps that are used in that layer. The number below the layer represents the number of feature maps that layer has; in cases where an X is shown, it corresponds to the number of classes to be identified.

The layer structure is divided into two phases:

Encoder: in this first encoding phase, the characteristics of the image are extracted using convolution and pool layers.

Decoder: in the second phase of decoding, the image is reconstructed by means of transposed convolution; in this way, the classification of each pixel of the image is obtained.

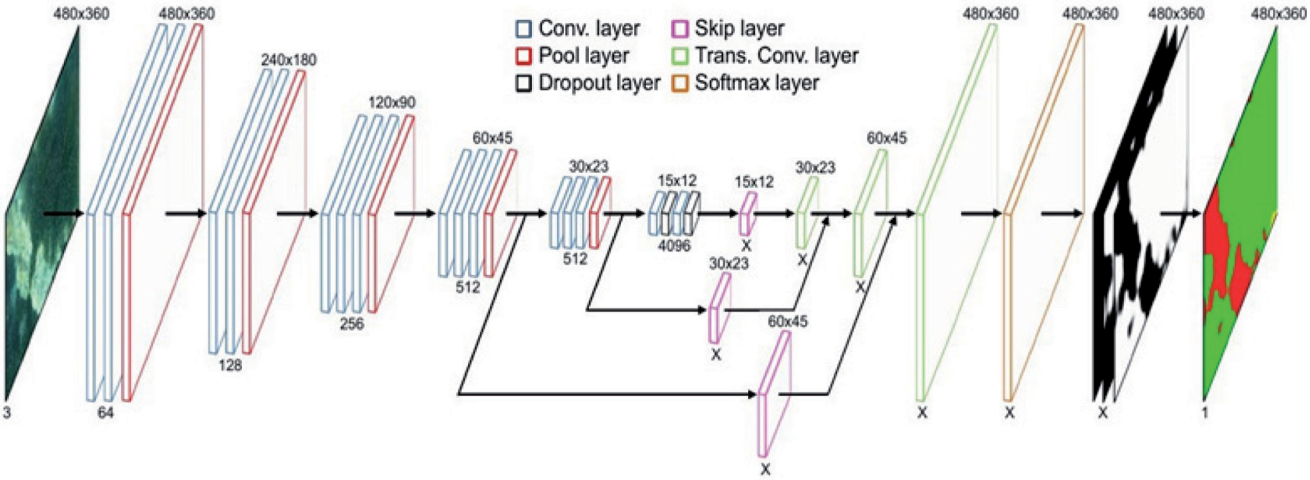


Fig. 2. Layered architecture of the neural network used

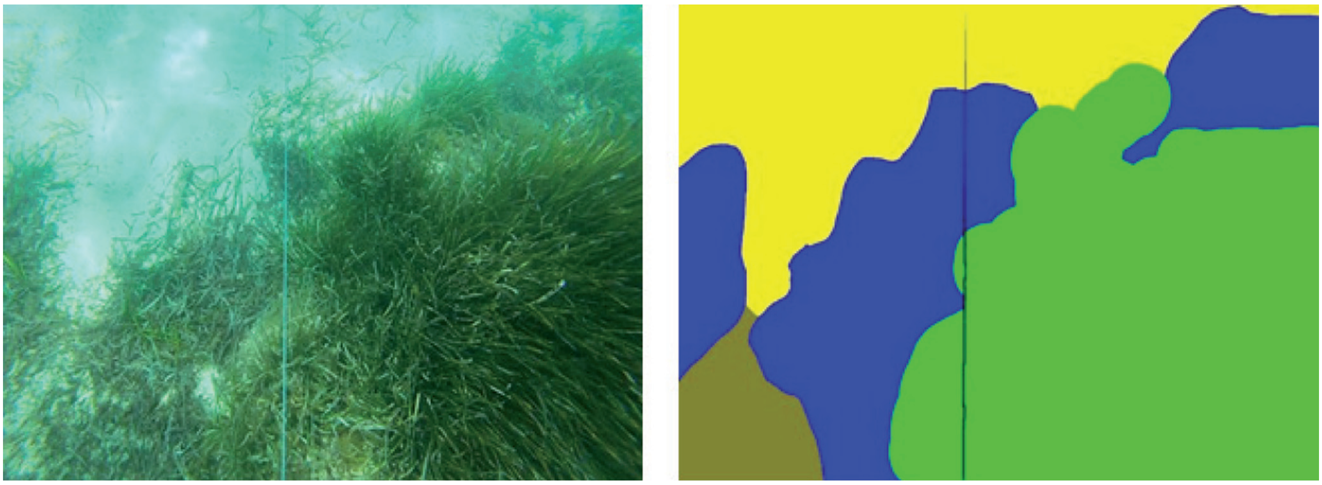


Fig. 3. Image labelling for training

Encoder

The dimensions of the images to be processed will be 480x360 pixels in RGB. In the first stage, the image is received and processed by the first convolution layer. This first stage consists of 2 layers of 64 convolutions and a layer that performs the Max_Pool to reduce the size of the features by half (240x180 pixels). The second stage again consists of two convolution layers, although now the number of convolutions is doubled. When the convolutions are finished, a Max Pool is applied again to reduce the dimensions by half again (120x90 pixels).

The third and fourth stages follow the same procedure by adding a third convolution layer, doubling the number of convolutions, and halving the dimensions. In the fifth stage, the number of convolutions is maintained and the dimensions are reduced by half. In the sixth stage, in addition to the convolution layers, some dropout layers are included that are only used during training to avoid overfitting. At this stage the feature maps have been reduced to a size of 15x12.

3.2.2 Decoder

The decoder contains a map for each class to be classified. As the decoder stages progress, the size of the maps increases until they reach the size of the original image. The skip layer basically consists of a convolution of the feature map to obtain a new map per class. In the seventh stage, it consists of a skip layer, so from the sixth stage onwards, one map per class is obtained. The eighth stage consists of a transposed convolution that increases the size of the image. In the ninth stage, the transposed convolution of the sum of the layers of the previous stage and the skip layer of the fifth stage are performed, increasing the size of the image again. The tenth stage performs the transposed convolution of the sum of the layers of the previous stage and the skip layer of the fourth stage, increasing the size of the original image. In the eleventh stage, the softmax function is applied to normalize the result obtained. Finally, in the last stage, we obtain a map by class. The value of each position on this map represents the probability that this pixel is of being in the class of the map. Therefore, to decide which class each pixel belongs to, the one with the highest value is chosen. As each class is assigned a color at the end, an image is obtained where each pixel is classified.

Training

To calculate the values of the neural network model, a supervised training phase will be carried out. This training consists of adjusting the parameters of the model from a collection of images with their corresponding labelled images. Figure 3 shows how an image (a) has been labelled (b) in order to carry out the training. Each color represents a class; the green is NG, the blue is dead NG, the yellow is sand, and the brown is sand with other elements.

The training consists of processing the images and comparing the obtained result with the desired result using a loss function. By applying the loss function, the model error is obtained, from which the model weights are adjusted. This procedure is repeated several times to progressively adjust the weights. Using an algorithm called back propagation, thanks to the gradient of the loss function, the weights of the model are calculated. To avoid overfitting and improve the result of the model obtained, dropout layers are usually added. Overfitting is a problem that can appear in neural networks. What it causes is that the model obtained after training works correctly for the images used in the training but not for the new ones. What the dropout layers do is randomly deactivate some connections in the neural network. In this way, the trained network is forced to be more versatile, thanks to the fact that the activation of more connections is forced when identifying the characteristics of the image [4] and [5]. Figure 4 shows a scheme of how the

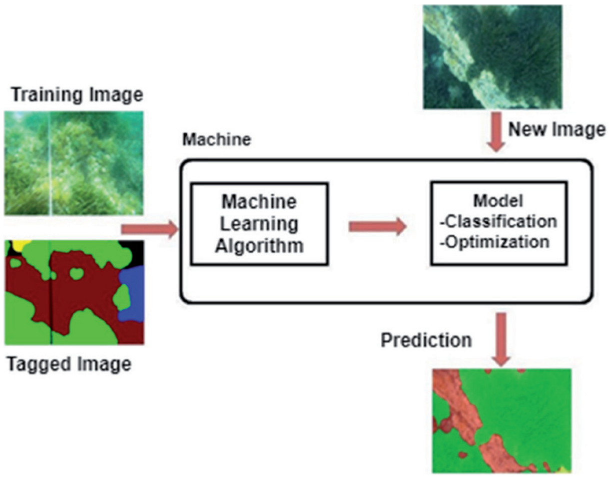

Training Image
Machine
New Image
Machine Learning Algorithm
Model -Classification -Optimization
Tagged Image
Prediction

Fig. 4. Model training scheme

model is trained so that once it is trained, it can perform the inference of new images.

Hyperparameters used in training

There are a number of parameters that are used during training that directly influence the output of the trained model. Some of these parameters are:

Data augmentation consists of increasing the amount of data used during training. To do this, modifications are made to the contrast and brightness of the images used for training, thus increasing the variety of images and reducing overfitting.

Learning Rate: This affects how the model converges to the result during each step. A larger size can achieve the final result sooner, and smaller values obtain more optimal results. Therefore, it is necessary to determine an appropriate value that reaches a compromise between speed and quality of the results.

Number of iterations: defines the number of times to perform the back propagation and train again.

The suitability of these parameters depends on each problem to be solved. In this case, the same ones have been used as in work [2]. Since we are facing the same problem, the same types of images are used, and the same architecture is used simply by increasing the number of classes to be identified. During the training, data augmentation is used, with a learning rate of 1e-05 and 16000 iterations.

Models

Due to the different and varied classes present in the set of images, it is decided to carry out different training sessions, each of them grouping all the classes in a different way. In this way, since the input data for the network is different, different models will be obtained. The groupings made for the different trained models are shown in Table 2.

Because the images labelled as Sand and Sand-d are very similar and there are not enough images for the training to be effective in distinguishing them, it has been decided to group them all in the same Sand class. The same goes for the rock class and rock-d, which have been lumped together under the rock type.

Model I

When carrying out the training of model I, it has been decided to carry out a minimum classification, trying to dispense with the classes that appear less in the images, grouping them with the class that most resembles them. For this reason, the living NG (NG-a) and the dead NG leaves (NG-d) have been grouped into the same class, with all the rocky funds in the rock class and all the sand funds in the sand class. Table 3 shows the classification used for model I.

Table 2. Grouping of classes for the models used

All classes			Model III		Model II		Model I	
Name	Description	Color	Name	Color	Name	Color	Name	Color
NG-a	Living NG Kill	🟩	NG-a	🟩	NG-a	🟩	NG	🟩
NG-d	dead NG leaves	🟦	NG-d	🟦	NG-d	🟦		
Rock	clean rocky bottom	🟥	Rock matte	🟥	Rock	🟥	Rock	🟥
rock-d	Rocky bottom with other elements	🟫						
Matte-r	Dead NG clump on rock bottom	🟦	sandy	🟪				
matte-s	Dead NG clump on sand bottom	🟪			Sand	🟨	Sand	🟨
sandy	clean sand background	🟨	Name NG-a	🟨				
Sand-d	Sand background with other elements	🟫						
Background	Fund that does not correspond to any of the classes to be classified or that has not been identified	⬛	NG-d	⬛	Background	⬛	Background	⬛

Table 3. Classification used in model I

Name	Description	Color
NG	Live NG mat and dead NG leaves	🟩
Rock	Rocky bottom	🟥
Sand	sand background	🟨
Background	Fund that does not correspond to any of the classes to be classified or that has not been identified	⬛

Table 4. Classification used in model II

Name	Description	Color
NG-a	NG bush alive	🟩
NG-d	NG leaves dead	🟦
Rock	Rocky bottom	🟥
Sand	sand background	🟨
Background	Fund that does not correspond to any of the classes to be classified or that has not been identified	⬛

Table 5. Classification used in model III

Name	Description	Color
NG-a	Living NG Kill	🟩
NG-d	dead NG leaves	🟦
Rock	Rocky bottom	🟥
Sand	sand background	🟨
Matte	dead NG bush	🟪
Background	Fund that does not correspond to any of the classes to be classified or that has not been identified	

Model II

When model II was carried out, the classification was increased by one more class with respect to model I. The NG class separates the live NG (NG-a) and the dead NG leaves (NG-d). Table 4 shows the classification used for model II.

Model III

In model III, one more class is increased with respect to model II. I added the matte class to represent the mats of dead NG. Table 5 shows the classification used for model III.

Metrics of a neural network

To determine the quality of the results obtained in the classification of the neural network, a series of metrics such as true positive (TP), true negative (TN), false positive (FP), false negative (FN), accuracy, precision, recall, and F1 are used. These metrics basically consist of a series of calculations that are performed on the results in order to obtain objective information on the behaviour of the neural network.

Results

When training with the collections, the values for the models are obtained. Once with the trained models, the inference is made with the validation collections, the confusion matrix, and the metrics of the models.

Model I

Table 6 shows the confusion matrix for model I, and table 7 shows the metrics obtained from the confusion matrix.

As a result of the results, it can be verified that the metrics of both the NG and the rock are relatively good, above 90% in all metrics. As for the sand class, the results are generally good, but it has precision problems (79.3%), which means that the model is usually wrong more or less 1 time out of 5 when it predicts that a pixel is sand. Also, the value of F1 is not very high, which is why it is worth 86.7% since it is affected by precision. The one that obtains the worst results is the background class; the precision is relatively good, but the recall is quite low. This means that when a background

Table 6. Confusion matrix of model I

		Prediction (pixels)			
		NG	Rock	Sand	Background
Real (píxels)	NG	4963283	60011	67662	2264
	Rock	136687	3664330	50693	20764
	Sand	6340	2051	523967	3047
	Background	159546	321024	7412	100000

Table 7. Model I metrics

	Area (%)	Accuracy(%)	Precision(%)	Recall(%)	F_1(%)
NG	51.5227	96.8262	95.0256	98.8242	96.9267
Rock	39.8226	95.2263	91.7257	95.4245	93.5289
Sand	6.9223	99.8261	80.7258	97.1235	88.1261
Background	7.4268	96.1268	79.1260	17.7236	28.4247

Table 8. Normalized confusion matrix of model I

		Prediction			
		NG	Rock	Sand	Background
Real (%)	NG	98.8212	2.6227	2.7219	1.5273
	Rock	5.2241	95.4228	3.0260	2.0270
	Sand	4.8227	1.8231	97.1235	2.0268
	Background	29.5261	55.8252	2.6260	17.7267

is predicted, it is usually correct, but most of the time, when there is a background, it is not detected.

Table 8 shows the normalized confusion matrix, and from it it is clearly seen that most of the time that there is a background class, it is predicted as rock and sand rather than as background. Of the rest of the classes, it is verified that most of the time it is predicted correctly. Figure 5 shows an example of some predictions made from model I. In the first column are the original images (1a) and (2a), and in the second column are the manually labelled images (1b) and (2b). And finally, in the third column, is the prediction made by the model during the inferences (1c) and (2c). This prediction is shown superimposed on the actual image.

In the images in Figure 5, it can be seen that the NG, the rock, and the sand have been detected with enough precision.

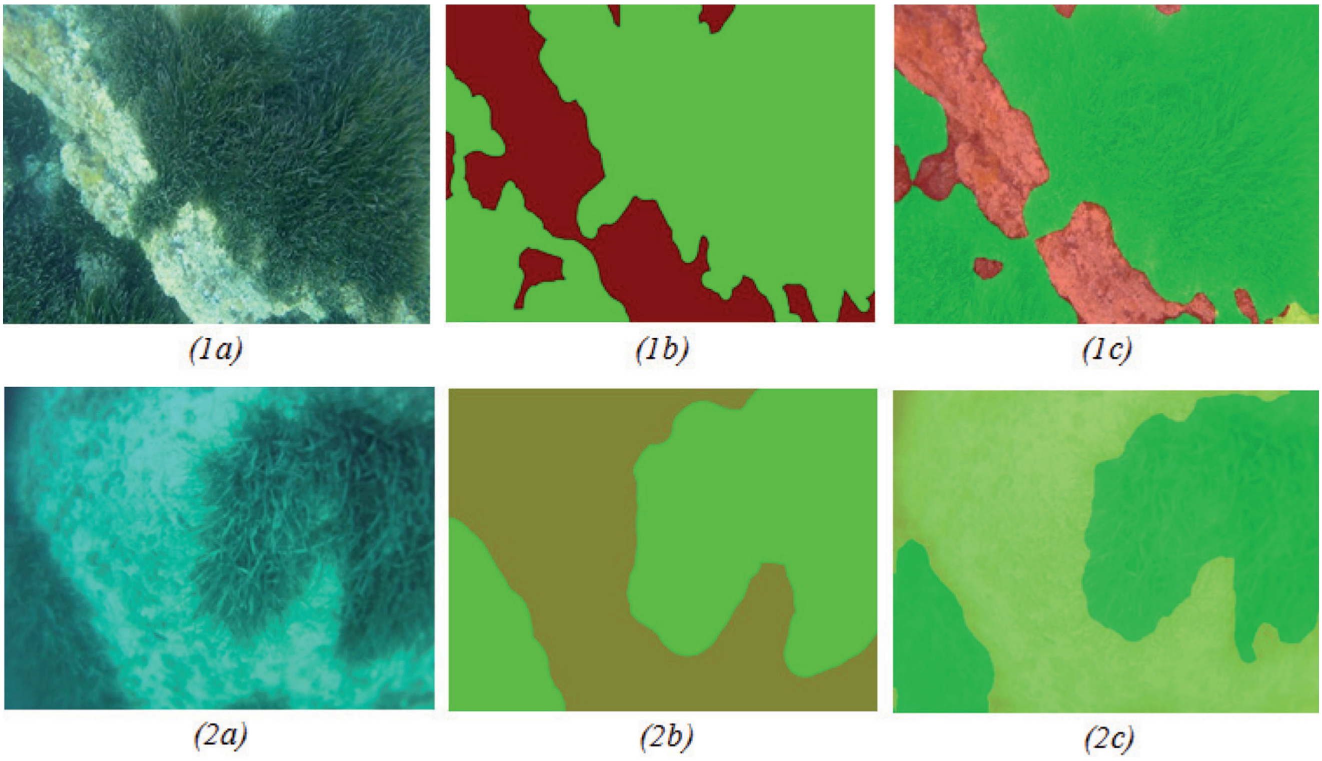
Model II

Table 9 shows the confusion matrix for model II, and table 16 shows the metrics obtained from the confusion matrix.

In this model, one more class is added, distinguishing between living and dead NG. But the number of images containing dead NG is significantly lower than the live ones.

The NG-a, Rock, and Sand have good behaviour, similar to that of the model I. The background class also behaves in a similar way, although it has worsened its behaviour a little more compared to model I. The newly added NG-d class has poor behaviour, similar to Background. Precision, recall, and F1 have values below 20%, which means that this class is not detected very well.

From the normalized confusion matrix in Table 11, it can be seen that most of the times that NG-d should have been predicted were predicted as NG-a. This is due to their great similarity. It can also be seen that many times the NG-d was actually predicted to be sand; this is possibly due to the fact that in the training images, most of the time the NG-d was on sand. Figure 6 shows an example of some predictions made from model II. As in Figure 5, in the first column are the original images (1a) and (2a), in the second the manually labelled ones (1b) and (2b), and finally the prediction made by the model during the inference (1c). and (2c).


(1a) (1b) (1c)

(2a) (2b) (2c)

Figure 5. Model I predictions

Table 9. Confusion matrix of model II

		Prediction (pixels)				
		NG-a	**NG-d**	**Rock**	**Sand**	**Background**
Real (píxeles)	**NG-a**	4981572	496	58423	14502	1586
	NG-d	13678	50568	7361	8014	0
	Rock	135737	0	3746465	3829	153
	Sand	15949	7287	54085	446606	531
	Background	4981572	0	58423	14502	1586

Table 10. Model II metrics

	Area (%)	Accuracy(%)	Precision(%)	Recall(%)	F1(%)
NG-a	51.1263	97.1263	94.7263	99.9263	97.2263
NG-d	1.8263	100.8263	21.3263	20.5263	20.9263
Rock	39.8263	94.9263	89.5263	97.5263	93.3263
Sand	6.9263	100.1263	95.3263	83.3263	88.9263
Background	51.1263	97.1263	94.7263	99.9263	97.2263

Table 11. Normalized confusion matrix of model II

		Prediction				
		NG-a	**NG-d**	**Rock**	**Sand**	**Background**
Real (%)	**NG-a**	99.9263	1.4263	2.6263	1.7263	1.4263
	NG-d	41.0263	20.5263	21.2263	23.0263	1.4263
	Rock	5.2263	1.4263	97.5263	1.5263	1.4263
	Sand	4.5263	6.5263	11.3263	83.3263	1.5263
	Background	99.9263	1.4263	2.6263	1.7263	1.4263

In Figure 6, it can be seen in figure (1c) how sand is no longer detected, as well as in model I, confusing NG-d with sand. On the other hand, the NG and the rock can be seen, and it continues to detect them with enough precision.

4.3 Model III

Table 12 shows the confusion matrix for model III, and table 13 shows the metrics obtained from the confusion matrix.

A new class, Matte (dead NG kill), has been added to this model. For this class, as for the NG-d, it does not appear many times in the images used during training.

The NG-a and Rock and Sand classes, despite having worsened the predictions a bit, continue to behave quite well. The sand class is the one that has worsened the most, although it continues to behave in an acceptable way. The new Matte class does not behave well, as do the NG-d and

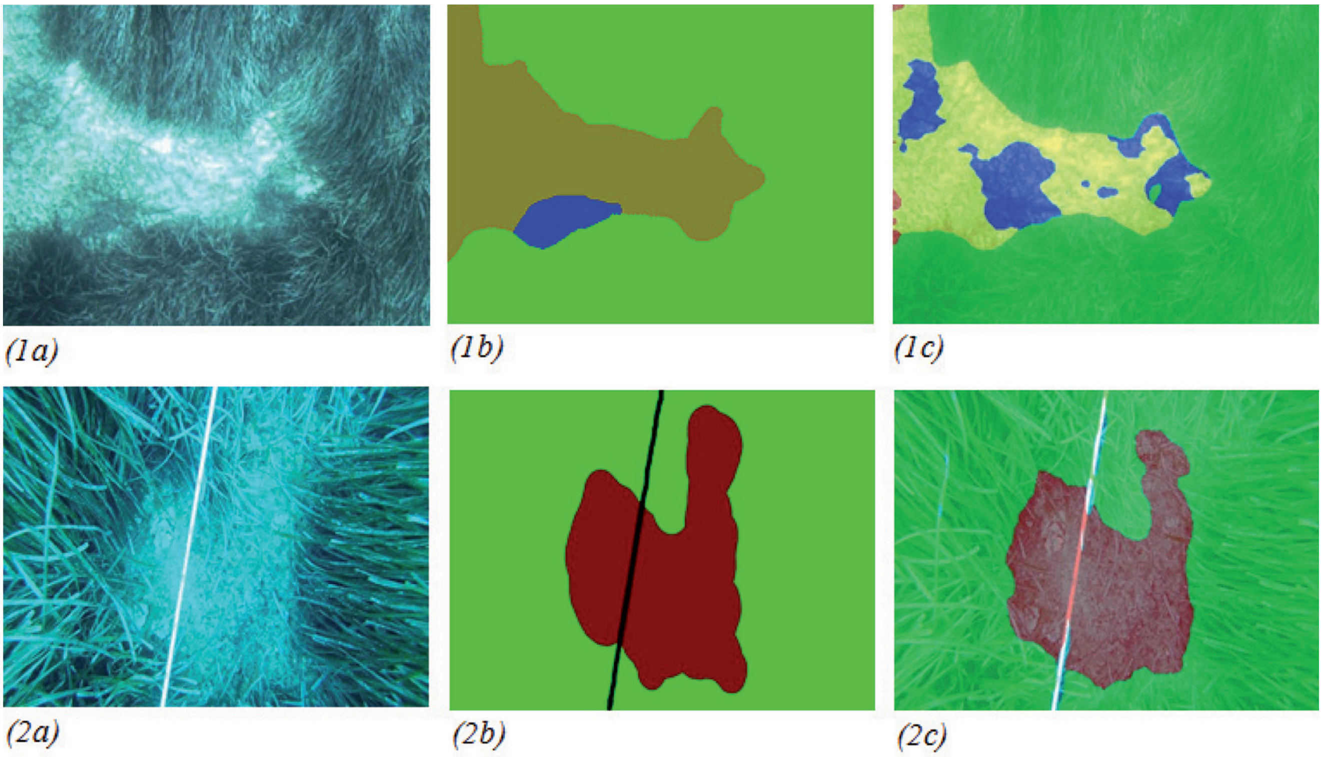


Figure 6. Model II predictions

Table 12. Confusion matrix of model III

		Prediction (pixels)					
		NG-a	NG-d	Rock	Sand	Matte	Background
Real (pixeles)	NG-a	4732028	2173	127895	74669	76	7443
	NG-d	6388	13065	3306	14282	0	209
	Rock	66907	0	3619081	166670	0	40151
	Sand	3833	25656	4074	452823	1625	476
	Matte	4999	23741	0	40620	4176	0
	Background	142627	0	309794	12298	0	134469

Table 13. Model III metrics

	Area (%)	Accuracy(%)	Precision(%)	Recall(%)	F1(%)
NG-a	51.1261	97.0250	96.7223	97.2225	96.9227
NG-d	1.8253	100.6261	21.0283	36.3217	26.5214
Rock	39.7232	94.1226	90.1214	94.1228	92.1268
Sand	6.2221	98.0224	60.8216	93.6227	73.7260
Matte	51.1234	97.0268	96.7267	97.2223	96.9223
Background	1.8225	100.6262	21.0260	36.3213	26.5217

Table 14. Normalized confusion matrix of model III

		Prediction					
		NG-a	NG-d	Rock	Sand	Matte	Background
Real (%)	NG-a	97.2263	0.0000	3.9263	2.9263	0.0000	1.6263
	NG-d	18.6263	36.3263	10.5263	39.6263	0.0000	2.0263
	Rock	3.4263	0.0000	94.1263	5.7263	0.0000	2.4263
	Sand	2.2263	6.9263	2.5263	93.6263	1.8263	1.5263
	Matte	8.2263	34.7263	1.4263	55.6263	7.1263	1.4263
	Background	24.9263	0.0000	54.0263	3.4263	1.4263	23.3263

Background classes. This is due to the few images that are in the training for these classes.

In the normalized confusion matrix of Table 14, it can be seen that the classes NG-a, Sand, and Rock tend to be correct with the predictions. On the other hand, the background and NG-d classes continue to fail a lot; the NG-d class has improved a bit compared to the II model, but they are still not acceptable values, and in addition to being confused with the NG-d and Sand, it is also confused in the predictions with Matt. The matte class also doesn't do well when mixed up, especially with the sand and NG-d classes. This is due, as for NG-d and background, to the few images that are available to

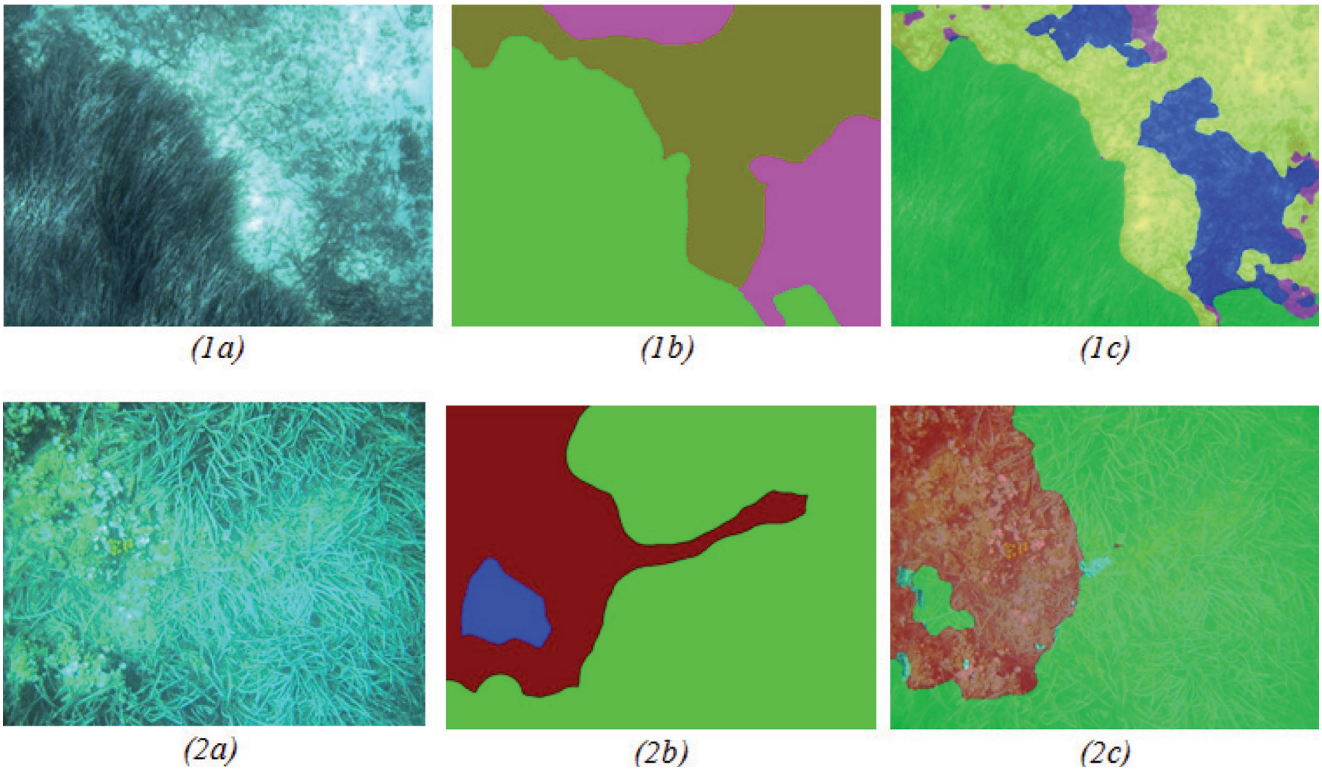


Fig. 7: Model III predictions

train this class. Figure 7 shows an example of some predictions made from model II. As in Figure 5, in the first column are the original images (1a) and (2a), in the second the manually labelled ones (1b) and (2b), and finally the prediction made by the model during the inference (1c). and (2c).

In Figure 17, it can be seen that rock and NG are still detectable, but sand is sometimes confused with other classes, and both NG-d and matte do not quite identify them correctly.

Analysis of the results

The results obtained are relatively satisfactory, especially in the detection of NG, sand, and rock. Although the rest of the classes have not been able to be detected in a reliable way. The biggest reason why this happens is that the number of available images used in the trainings did not reach 250 for all classes. Since most of the images were of NG, it has led to the best detection of this class. Normally, in this type of classification, image collections of several thousand are used for each class that is desired to be able to detect. Not having so many images has favoured a decrease in the performance of the results obtained by the models. Another error that has been introduced in the collection of images has been during labelling. The labelling has been done by hand, and when labelling the boundaries between classes, the contours have not been exhaustively outlined. Also, different people would not mark exactly the same class boundary in an image. The labelling of certain classes has also been complicated, especially with the matte class and with some images where certain areas did not have sufficient quality. In these cases, there are quite a few discrepancies in how two people would classify the areas of the same image. These labelling errors introduce a decrease in the performance of the trained models.

Conclusion

Despite the fact that not all classes have been detected with sufficient reliability, the most relevant classes (NG, Sand, and Rock) have been successfully detected, obtaining in these cases an F1 of around 90%. In addition, the classification problems could be corrected by expanding the collection of images used in the training. The implemented system presents a series of advantages compared to other seabed detection processes, such as, for example, that it allows segmentation for each pixel of the image without suffering loss of information or requiring any type of post-processing, allowing this task to be carried out in real time. The implementation carried out would not only serve to classify images of the seabed; changing the collection of images and the configuration file of the model can be easily adapted to classify other types of images.

References

1. Jiang Q, Chen Y, Wang G, Ji T. A novel deep neural network for noise removal from underwater image. Signal Processing. Sept. 2020;87, doi: 10.1016/j.image.2020.115921.
2. O'Byrne M, Pakrashi V, Schoefs F, Ghosh B. A Stereo-matching technique for recovering 3D information from underwater inspection imagery. Comput. Aided Civ. Infrastruct. Eng. 2018;33:193–208.
3. Coyle R, Hardiman G, Driscoll K.O. Microplastics in the marine environment: A review of their sources, distribution processes, uptake and exchange in ecosystems. Case Stud. Chem. Environ. Eng. 2020;2:100010.
4. Honingh D, Van Emmerik T, Uijttewaal W, Kardhana H, Hoes O, Van de Giesen N. Urban River water level increase through plastic the accumulation at a rack structure. Front. Earth Sci. 2020;8:28.
5. Zhang M, Liu C, Wang S, He Q, Wei Q. Lightweight underwater object detection based on YOLO v4 and multi-scale attentional feature fusion. Remote Sensing. 2021;13(22):4706.
6. Xiao Y, Tian Z, Yu J, Zhang Y, Liu S, Du S, Lan X. A review of object detection based on deep learning. Multimed. Tools Appl. 2020;79:23729–23791.
7. Zhao ZQ, Zheng P, Xu ST, Wu X. Object detection with deep learning: A review. IEEE Trans. Neural Netw. Learn. Syst. 2019;30:3212–3232.
8. Ge P, Chen Y, Wang G, Weng G. A hybrid active contour model based on pre-fitting energy and adaptive functions for fast image segmentation. Pattern Recognit. Lett. 2022;158:71–79.
9. Weng G, Dong B, Lei Y. A level set method based on additive bias correction for image segmentation. Expert Syst. Appl. 2021;185:115633.
10. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal Loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. 2020;42:318–327.
11. Fang W, Wang L, Ren P. Tinier-YOLO: a real-time object detection method for constrained environments. IEEE Access. 2020;81935–1944.
12. Doherty M, Landowski J, Maynard P, Uber G, Fries D, Maltz F. Side scan sonar object classification algorithms. Proceedings of the 6th International Symposium on Unmanned Untethered Submersible Technology; IEEE. 1989;34:417–424.
13. Bello MG, Markov random-field-based anomaly screening algorithm. in SPIE's 1995 Symposium on OE/Aerospace Sensing and Dual Use Photonics. International Society for Optics and Photonics, 1995;34:466–474.

14. Mignotte M, Collet C, Pérez P, Bouthemy P. Markov random field and fuzzy logic modeling in sonar imagery: application to the classification of underwater floor. Computer Vision and Image Understanding. 2000;79(1):4–24.
15. Mignotte M, Collet C, Pérez P, Bouthemy P. Three-class markovian segmentation of high-resolution sonar images. Computer Vision and Image Understanding. 1999;76(3):191–204.
16. Mignotte M, Collet C, Pérez P, Bouthemy P. Statistical model and genetic optimization: application to pattern detection in sonar images. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE. 1998;5:2741–2744,
17. Mignotte M, Collet C, Pérez P, Bouthemy P. Unsupervised markovian segmentation of sonar images. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE. 1997;4:2781–2784.
18. Reed S, Petillot Y, Bell J. An automatic approach to the detection and extraction of mine features in sidescan sonar. IEEE Journal of Oceanic Engineering. 2003;28(1):90–105.
19. Bell J, Petillot Y, Lebart K, Reed S, Coiras E, Mignotte P, Rohou H. Target recognition in synthetic aperture and high resolution sidescan sonar. In: High Resolution Imaging and Target Classification. The Institution of Engineering and Technology Seminar. IET. 2006;99–106.
20. Reed S, Petillot Y, Bell J. Automated approach to classification of mine-like objects in sidescan sonar using highlight and shadow information. IEE Proceedings Radar, Sonar and Navigation. 2004;151(1):48–56.
21. Daniell O, Petillot Y, Reed S, Vazquez J, Frau A. Reducing false alarms in automated target recognition using local sea-floor characteristics. In: Sensor Signal Processing for Defence (SSPD), 2014.
22. Calder B, Linnett L, Carmichael D. Spatial stochastic models for seabed object detection. In: AeroSense'97. International Society for Optics and Photonics. 1997;172–182.
23. Calder B, Linnett L, Carmichael D. Bayesian approach to object detection in sidescan sonar. IEE Proceedings-Vision, Image and Signal Processing. 1998;145(3):221–228.
24. Maussang F, Chanussota J, Histetb A. Automated segmentation of sas images using the mean–standard deviation plane for the detection of underwater mines. In: Proc. of MTS/IEEE Oceans 03 conference, 2003.
25. Maussang F, Chanussot J, Hétet A, Amate M. Mean-standard deviation representation of sonar images for echo detection: Application to sas images. IEEE Journal of Oceanic Engineering. 2007;32(4);956–970.
26. Linnett L, Carmichael D, Clarke S, Tress A. Texture analysis of sidescan sonar data. In: Texture analysis in radar and sonar, IEE Seminar on. 1993;2–1.
27. Bell J M, Linnett L. Simulation and analysis of synthetic sidescan sonar images. IEE Proceedings-radar, sonar and navigation. 1997;144(4):219–226.
28. Atallah L, Shang C, Bates R. Object detection at different resolution in archaeological side-scan sonar images. In: Europe Oceans. IEEE. 2005;1:287–292.
29. Michael TG, Tucker JD. Canonical correlation analysis for coherent change detection in synthetic aperture sonar imagery. Institute of Acoustics Proceedings. 2010;32(4);117–122.
30. Williams DP. Fast target detection in synthetic aperture sonar imagery: A new algorithm and large-scale performance analysis. IEEE Journal of Oceanic Engineering. 40, no. 1; 71–92, 2015.
31. Reed S, Ruiz IT, Capus C, Petillot Y, The fusion of large scale classified side-scan sonar image mosaics IEEE Transactions on Image Processing. 2006;15(7):2049– 2060.