Deep Learning applied to Underwater Mine Warfare

Killian DENOS, Mathieu RAVAUT, Antoine FAGETTE, LIM Hock-Siong
Thales Research & Technology Singapore
Thales Solutions Asia Pte Ltd
Singapore
{killian.denos, antoine.fagette}@asia.thalesgroup.com

***Abstract* — In this article we are addressing the problem of automatic detection and classification of underwater mines on images generated by a Synthetic Aperture Sonar (SAS). To tackle this problem, we are investigating the use of Machine Learning techniques, in particular Deep Learning. Using this method we faced two challenges, (i) the availability of a sufficient amount of training data to learn the classification model and (ii) the design of the deep learning pipeline suited for this one-class classification problem. Our contributions in this paper are, first the synthetic generation of realistic image datasets for the training of our Machine Learning algorithm, and second the research and development of a novel Deep Learning approach for automatic underwater mines classification using sonar images. The combination of these two contributions offers a new pipeline of operation for Mine Counter Measure Automatic Target Recognition (MCM ATR) systems.**

Keywords—Deep Learning; synthetic training dataset; underwater mine; sonar; synthetic aperture sonar

I. INTRODUCTION

This paper focuses on a Mine Counter Measure Automatic Target Recognition (MCM ATR) system. The goal of such a system is to detect and classify underwater mines lying on or tethered to the seabed. The detection and classification is done on the signal coming from a sonar, the latest technology investigated in this paper being a Synthetic Aperture Sonar (SAS) which can be embedded, for example, on an Autonomous Underwater Vehicle (AUV). To complete this task, we investigate the use of Machine Learning techniques. However, this defense-related field of application poses a fundamental challenge for any method based on this technology which is the lack of training data.

The most common mean to detect and classify underwater mines is based on the analysis of images processed from the sonar signal. Such images are gray-level orthographic top views of the seabed with shadows cast by the sonar. The traditional pipeline, as described on **Fig. 2**, is composed of (i) a detection part dedicated to find any object casting a shadow, and (ii) a classification part aiming at determining whether the detected shadow is cast by a mine or by any other random object that can be found on the seabed (e.g. rocks, wrecks or other landscape features). While the detection part is usually carried out by a shadow detection image processing algorithm, the classification part can be tackled with a feature extraction and Multi-Layer Perceptron (MLP) approach [1]. The latter relies on synthetic databases of mine shadow masks for their training.

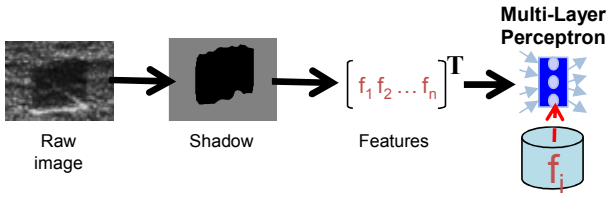

Fig. 2. Shadow detection and classification

Given the recent development in the field of image detection and classification, we decided to use a Deep Learning approach in order to enhance the performances of existing ATR systems. Using this method we faced two challenges, (i) the availability of a sufficient amount of training data to learn the classification model and (ii) the design of the deep learning pipeline. Our contribution in this study is the design and the implementation of a new pipeline of operation as described on **Fig. 1**. This pipeline is composed of 4 main building blocks:

- the first building block is dedicated to the generation of synthetic images to setup the learning database of the object to be detected. This database is therefore composed of snippets of SAS images of mines;
- the second building block is focusing on the one-class classification problem of learning what is the object we want to detect (i.e. the mine) without knowing the background. This task is tackled by an Auto-Encoder

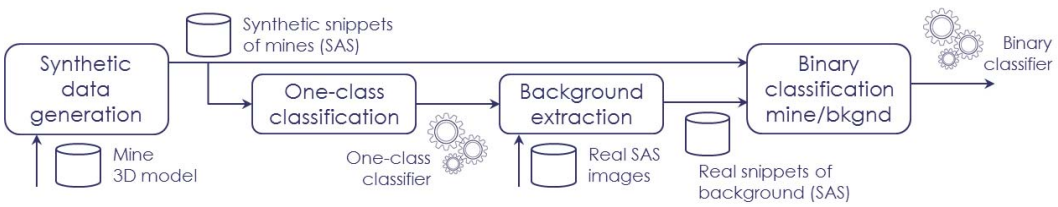

Fig. 1. Proposed pipeline of operation

(AE), takes as input the database produced by the previous building block and outputs a trained AE;

- the third building block aims at extracting real SAS images of the background in order to build a database of background images. This task is performed thanks to our previously trained AE running on real SAS images captured during a first dive of our AUV on a particular area. The output of this building block is a database of real snippets of the background;
- the last building block looks after the training of a binary classifier in order to properly detect and classify mines versus the background. This classification is performed using a Deep Convolutional Neural Network (DCNN). The training of this DCNN takes as input the synthetic mine images dataset created by the first building block and the real background images dataset extracted by the third building block.

Much as the first and second building blocks can be executed before or at the start of the first dive of the AUV embedding the SAS, the third and fourth building block are relying on the capture of real images during the first moments of the dive in order to complete a proper training of the whole pipeline that will be valid for the area being explored.

This paper is organized as follows. First, the synthetic generation of datasets for the training and testing of our Machine Learning algorithm is described in Section II, and the research and development of a novel Deep Learning approach for automatic underwater mines detection and classification using sonar images is detailed in Section III. We then present our results in Section IV and draw our conclusions and prospects for future work in Section V.

II. Synthetic Image generation

The quality and quantity of data in a training set greatly impacts the quality of the classification model to be derived. Due to the sensitivity of the mine warfare domain, the amount of available data is very limited. To overcome this paucity, we decided to generate synthetic data.

In this section, we first describe different approaches for synthetic image generation, and tackle the matter of representativeness and variability. Then, we explain the characteristics of the two datasets that we produce. Finally, we detail the method to create the data.

A. Approach, Representativeness and Variability

First, several options are available to produce synthetic data. On the one hand one can use algorithms to generate synthetic data from real data. These methods include deforming the original data and applying different transformations, or using 3D models of the object of interest to create a partially synthetic data from a real one [2]. On the other hand, it is possible to spawn fully synthetic data. One of the main advantages of such an approach is that the ground truth is readily available and reliable. We studied two approaches: optical and sonar rendering [3][4]. The latter requires high acoustic processing skills and a lot of simulation knowledge and power. We opted for the optical option – in which we generate photo-realistic sonar pictures – for the following reasons: it enables the use of well-developed image processing tools, several open-source 3D rendering engines are available, and it does not require highly complex acoustic simulation tools.

Using synthetic data, we have a full control on the number of images we can generate. Therefore, we can optimize this number so that our training dataset in particular is as diverse as possible yet as small and representative as possible. To address that optimization problem, we focus on three components: (i) the viewing conditions (i.e. the sonar), (ii) the mine and (iii) the seabed. Currently, we can produce data with several shapes of mines such as cylinder, truncated cone or more complex ones, and the sonar we are focusing on is a synthetic aperture side-view sonar which is capable of producing multi-aspect views. However, for this study, we used only cylinder-shaped mines and mono-aspect images.

To deal with representativeness, we consider the following capture parameters: range, attitude of the mine with respect to the sonar and nature of the seabed landscape. As shown on **Fig. 3**, the sonar has a limited range which is a combination of its altitude with respect to the seabed and the aperture of the sonar. Moreover, the mine has two degrees of freedom –rotation with respect to the Z axis (altitude) and rotation to follow the curves of the seabed (see **Fig. 4**) –. In addition to the previous points, inputs from sonar experts drove us to consider some geographical features of the seabed: sandy in most cases – which means flat –, covered with rocks or displaying ripples such as in Singapore – see **Fig. 4** –.

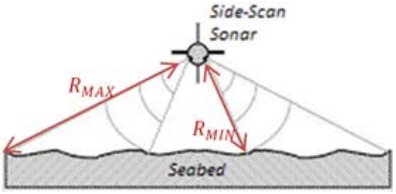

Fig. 3. Sonar range

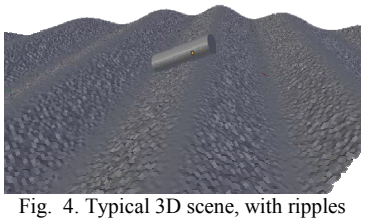
Fig. 4. Typical 3D scene, with ripples

In order to optimize the variability of the datasets, we can tweak the aforementioned parameters and in particular impose changes on the macro-shaping and the micro-shaping of the seabed, the lighting and the burying of the mine.

B. Datasets Characteristics

Two datasets need to be generated, the training dataset and the test dataset that we call operational dataset throughout this article. On first thought, the most crucial dataset to be generated is the training dataset. However, due to the scarcity

of real data we decided to generate synthetic images for our operational dataset in order to demonstrate the concept and the capability of our pipeline. Understandably, the datasets are generated according to the representativeness and the variability described earlier.

1) The training dataset

The purpose of the training dataset is for the Auto-Encoder (AE) to learn the concept of the object of interest, i.e. the system composed of the mine and its shadow in this paper. As the training dataset is fed to an unsupervised machine learning algorithm, the system must fill most of the picture while the background covers a minor part of it. The background aforementioned is built using white noise. We also tried to use random background textures but the results were not conclusive and white noise appears as the most efficient way to prevent the AE from learning anything from it. The pictures contained in this training dataset are therefore snippets of 112-by-112 pixels. This sizing is derived from the size of the mine, the range of the sonar and its resolution. An example of these training snippets can be seen on **Fig. 5**. The main difference between the operational dataset and the training dataset is that the latter does not attempt at modeling a realistic background.

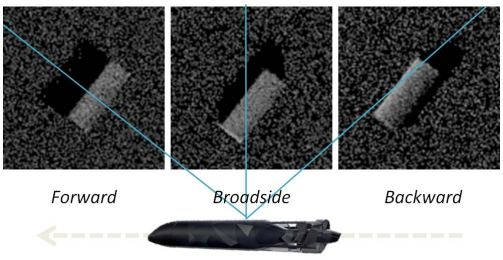

Fig. 5. Example of training snippets

2) The operational dataset

As opposed to the training dataset, the operational dataset has to represent what is captured on the field. It must be utterly realistic. Accordingly, the mines occupy a very small part of the picture while the background is prominent. An example is displayed on **Fig. 6**. The background is therefore quite more important to model. While the training background is composed of white noise, the operational one can be very various in terms of shape. This last parameter will impact the performances of the algorithm. Indeed, a flat sandy seabed leads to great detection performances while it is far more difficult on a rocky seabed or a sea bottom with ripples. Typically, distinguishing the mines and the rocks is a complex task.

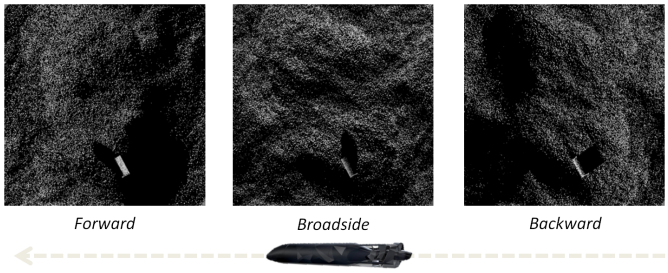

Fig. 6. Example of operational pictures

3) Multi-view

It was mentioned that the sonar technology we work with is capable of multi-aspect. This means that the sonar is capable of capturing snapshots of the sea bottom from several view angles (forward, broadside and backward in this paper) at the same time. This capability is described in **Fig. 5** and **Fig. 6**. Interestingly, this technology enables direct data fusion thanks to different views of the same object. All our datasets are generated in multi-view.

C. Generation Method

The core of the image generation problem is how to generate sonar-like pictures. As mentioned earlier, we do not simulate acoustic sonar data but we work with a sonar picture coming from a 3D model. As the sonar signal is a one dimension intensity point plotted on a two-dimensions map, it was decided that the generated data would be gray-scaled. In order to perform this generation, we describe here several elements: mine and shadow, seabed and rendering. For the design and rendering of the scene, we use the open-source 3D rendering software Blender [5].

First, we designed or bought several 3D models of mines. The initial step is to add sonar noise to those models. Indeed, the rendering of a perfect model of mine would not be realistic. In order to achieve our goal, we shape the mine with micro-deformations – built with a white noise – that will not affect the general shape of the object but will enable a sonar clutter when rendered – see **Fig. 7** and **Fig. 8**–. Using Blender, the micro-shaping is done using several displacement modifiers and a noise texture. The material used for the mines is to be reflective to have an echo, but also mainly diffusive so that most of the mine is in the dark. On real sonar pictures, it is very difficult to catch the mine because only the shadow really pops out. After this process, we lay the mine on the seafloor.

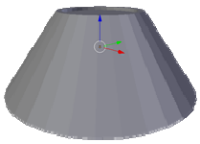

Fig. 7. Truncated cone mine model

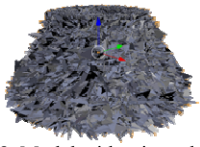

Fig. 8. Model with micro-shaping

Subsequently, the design of the shadow has to be done properly. As we do not use a sonar simulation tool but a rendering engine, a light source has to cast the shadow. Along the path of the sonar, all the lights rays are parallels. However, there is only one departure point for all the range. It means that, for two similar mine objects, the one close to the sonar will have a short shadow whereas the one far away will have a long shadow. Furthermore, two objects in the same position and at an equal range along the sonar path will have exactly the same shadow, as illustrated by **Fig. 9** and **Fig. 10**.

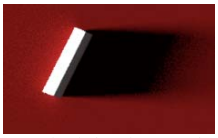

Fig. 9. Principle of sonar shadow at high range

Fig. 10. Principle of sonar shadow at low range

In order to obtain the right lighting, we use a powerful light source located infinitely far from the scene. The orientation of the rays is then set to match the range we are working with. This method works for the training dataset. However, it does not for the operational dataset. Indeed, we need the rays to be parallel along the path, but not along the range. We then designed and used a light source that fits those needs. Moreover, we set the power of the source in order for the render to be photorealistic.

Regarding the generation of the seabed, several aspects come into light: shaping of the macro-features such as ripples, rocks, wrecks, sand or mud, but also design of the sonar clutter, and variability. For the latter, we achieve it thanks to a randomization of the parameters of shaping strength, density and size of rocks, or level of noise. **Fig. 11** illustrates the micro-shaping which enables the clutter. **Fig. 12** presents a ripples macro-shaping, and **Fig. 13** a curvy sandy bottom macro-shaping. In Blender, this shaping is done using displacement modifiers associated with a noise texture for the clutter, and with different modifiers in the macro case. As the seabed is not supposed to reflect any signals, the material is diffusive with a dark color.

Fig. 11. Seabed micro-shaped

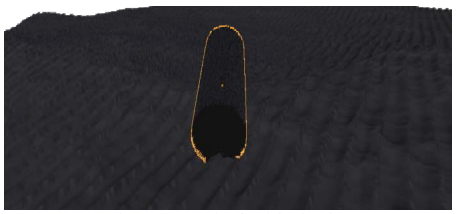

Fig. 12. Seabed with ripples

Fig. 13. Curvy seabed

With the 3D scene ready and properly designed, rendering is done according to the real capturing conditions and parameters which are orthographic mode, correct resolution and multi-aspect angle.

We perform a post-processing phase which includes the decrease of the contrast to match the sonar limitations and the multi-aspect geo-referencing fusion. Additionally, we generate the ground truth as the binary image of the system segmented as well as the bounding box encompassing the system.

The main limitation of our generation method that was raised by our sonar experts lies in the generation of the sonar echo produced by the mine itself. Indeed, using an optical renderer does not allow us to produce a reliable sonar echo for this kind of object. However, this limitation is for the moment ignored as our tests show that our pipeline is actually learning the shadow concept rather than the mine itself which is very beneficial in the sense that, nowadays, it actually gives most of the classification hints to the professional sonar operators to detect and identify underwater mines.

III. Machine Learning Algorithm

Since their achievements in worldwide competitions such as ImageNet, deep neural networks have been of tremendous interest for object detection problems in image processing. Our algorithm pipeline is divided in three consecutive phases corresponding to the second, third and fourth building blocks of the pipeline of operation described in Section I. The first phase, described in Section III.A, is dedicated to the one-class classification problem using a Denoising Auto-Encoder (DAE). The second phase, detailed in Section III.B, is looking at the extraction of the background images using our trained DAE. Finally the third phase, presented in Section III.C, is focusing on the supervised binary classification problem using a DCNN.

A. One-class classification

In the first part, our algorithm learns the concept of a mine and its shadow. Indeed, in our mine detection problem, what is not a mine could be anything else, and it goes as far as imagination does (e.g. sandy seabed, rocks or wrecks). On the other hand, the number of different types of mines to be detected is lesser and therefore easier to model.

Auto-encoders are a very specific type of neural networks which force the output to be as close as possible to the input [6]. A dimension reduction with sets of convolution and pooling layers is performed, creating higher-level features theoretically capable of reconstructing the input. Auto-encoders have an egg timer shape. To improve learning, we corrupt the input images by adding noise (implemented by a random Gaussian noise) and suppressing random features (e.g. replacing the pixel value by 0 following a particular destruction ratio). Such auto-encoders are called Denoising Auto-Encoders (DAE, [7]). Indeed, while reconstructing the image, since they have only learned higher-level features, they won't reproduce the noise or the black pixels. Noise and feature deletion help the network find a better quality higher level representation of the input. **Fig. 14** presents an example of corruption and reconstruction the auto-encoder we use.

Here, the size of the bottleneck layer is equal to approximately 1.05% of the input layer's size.

The network was trained using the training dataset described in Section II.B.1) for 15 epochs with 5000 training images and a batch size of 2.

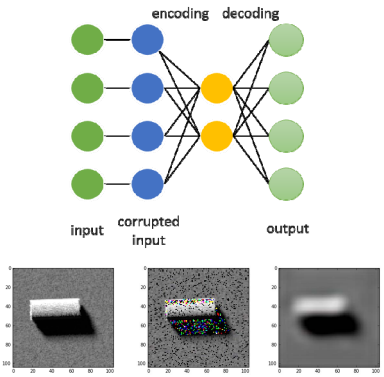

Fig. 14. Training snippet, corruption and reconstruction

B. *Background extraction*

Once our DAE is trained, the idea is to use it in order to discriminate areas of the operational images that are surely composed of the background versus those for which the DAE seems to recognize a threat.

When feeding an operational image to our trained DAE, we can compute for each pixel the error of reconstruction made by our auto-encoder between the input snippet centered on that given pixel and the corresponding output snippet produced. The auto-encoder only knows how to properly rebuild mines and their shadows. So, if there is a mine on that snippet, the auto-encoder should be performing well, meaning that the reconstruction should be accurate, and therefore that the reconstruction error should be low. We chose the mean square error (MSE) as a quantitative descriptor of reconstruction quality between the input and the output of the auto-encoder. This process leads to the derivation of a heat map of mean square errors on the full test image. In theory, the lower the MSE is, the higher the chance that there is a mine around the given pixel. However, plain gray backgrounds for instance are easily reconstructed by any model, and therefore produce a low MSE. Hence, we also check if the reconstruction MSE is in the same range as the one resulting from the validation set. In the end, we combine those three criteria, (i) global minimum, (ii) known MSE range and (iii) local minima to highlight the level of confidence that the DAE outputs regarding the presence of a threat or not.

Fig. 15 shows an example of a test image and the resulting heatmap after going through the auto-encoder and the three criteria for thresholding. On this example, one can guess that blue zones are much more likely to be mines. But there is no certitude, since it could as well represent rock or seabed shadows which present artifacts that are too close to the mine and shadow signature for the DAE to discriminate properly. Our labels cover all these possibilities and produce false positives (such as in the red spots on **Fig. 16**). In labeled areas, there is probably a mine. But what matters for this phase is that we do know with a very high level of certainty that, in unlabeled areas (black in the labels map), there are no mines. Especially if these areas were given a high MSE in the heat map. Thus, we can extract snippets from these zones, and we will have examples of what is not a mine, i.e. examples of the unknown negative class that is the background. Building such a negative class of similar volume to the positive one will enable us to transform the mine detection problem into a supervised binary image classification. Therefore, feeding these two classes to a deep convolutional network will fine-tune our mine detector and classifier.

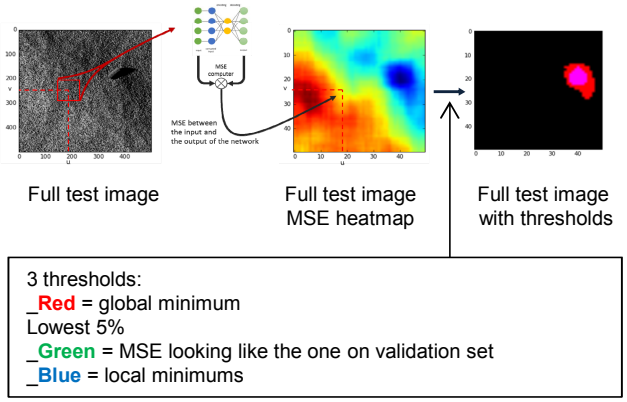

Fig. 15. Test image, resulting heatmap, and labelling

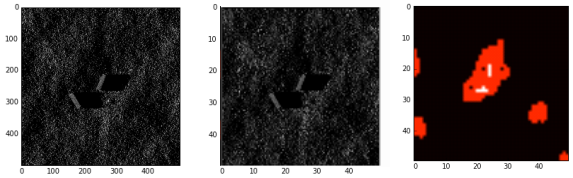
Fig. 16. Other example of labelling on test image

One of the main problems with the negative class, i.e. the background, is that it is too large and too diverse to be properly represented. But building the negative class on the operational images themselves can provide a good sample of the background faced during that specific operation. The chosen negative examples will accurately describe the seabed in that configuration, because they come from this configuration. Thus, this algorithm presents the immense advantage of being adaptable to any theatre of operation.

For this phase, we used a subset of the operational dataset described in Section II.B.2) and fed it to our trained DAE. **Fig. 17** presents some examples of randomly extracted negative images with the process described in this Section. To be noted that this algorithm does not guarantee that there won't be any part of the mine visible on the negative class images.

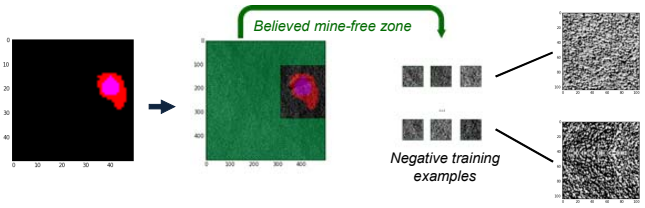

Fig. 17. Example of background extraction

C. *Supervised binary classification*

The architecture deployed for our convolutional neural network was inspired from the ones of famous detection and classification networks such as VGGNets [8]. It is made of

pairs of Convolution layers followed by MaxPooling ones, and completed by 3 fully-connected layers at the end of the network, the last one being activated by a softmax function. Dropout layers were also used to help prevent over-fitting. Training was done using the operational dataset described in Section II.B.2) and for 5 epochs with a batch size equal to 8.

This network returns a binary image with ones where it believes mines are.

IV. Results

In this Section, we present the results obtained by our pipeline on two operational datasets generated as described in Section II.B.2).

A. Evaluation methodology

We decided to use two metrics to measure the performance of our detector, namely the F-score and the localization distance.

The F-score, using surface ratios representing false positives, true positives, false negatives and true negatives, presents two advantages. It is easily apprehended because comprised between 0 and 1, and it takes into account false positives (unlike the accuracy), which are a major problem in all mine detection algorithms.

Defining the true positives for mines is not straightforward. Hence, we decided of a criterion. The reasoning is illustrated by **Fig. 18**.

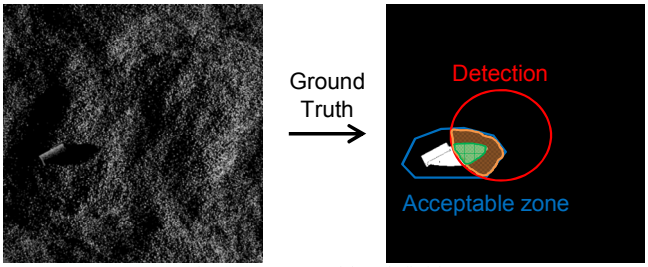

Fig. 18. True Positive definition

Our criteria is based on the combination of (i) the ratio of the area of the intersection between the ground truth and the detection, by the area of the detection – on **Fig. 18**, it corresponds to green over red –, and (ii) the ratio of the area of the intersection between the acceptable zone and the detection, by the area of the detection – on **Fig. 18**, it corresponds to orange plus green over red –. If this criterion is above a fixed threshold, we consider the detection as True Positive.

Furthermore, we compute the second metric as the L2-distance between the detection centroid and the ground truth centroid. This metric is explained on **Fig. 19**, where the distance is represented by the green arrow between the black centroid and the red one.

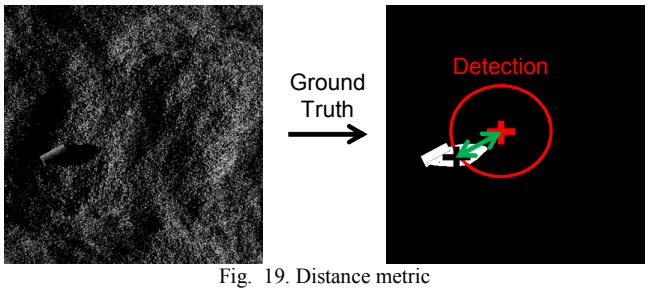

Fig. 19. Distance metric

B. Results

Table 1 is summarizing the values of the metrics obtained on images from a first operational dataset. To generate this dataset, we create a scene that is 25 meters by 25 meters large. The synthetic seabed is a sandy environment with low level of variation on the landscape and without rocks. On this seabed, we then spread randomly between zero and three cylinder mines of 2m length and 50cm radius. Our operational dataset comprises 500 images and **Fig. 20** is an example of these.

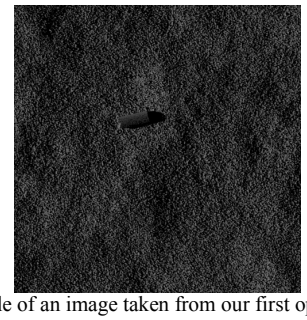

Fig. 20. Example of an image taken from our first operational dataset

These results are quite preliminary as they were obtained after five months of work on a first design of our pipeline. Nonetheless, they are very encouraging as the F-Score is high and the location of the mines is accurate and below the typical size of the cylinder to be found. However, the background used for this operational dataset is quite simple and increasing its complexity is the way forward in order to improve the reliability of our pipeline.

Table 1: Results for the first operational dataset

F-Score	Average L2 Distance (meters)	Standard Deviation L2 Distance (meters)
0.87	0.71	0.48

We conducted a second series of test using a second operational dataset with a more complex background. For this dataset, we create a scene of same size, but this time the synthetic seabed is a sandy environment with higher level of variation on the landscape and rocks. On this seabed we randomly lay zero or one cylinder mine. **Fig. 21** illustrates an image from this dataset that comprises 500 images.

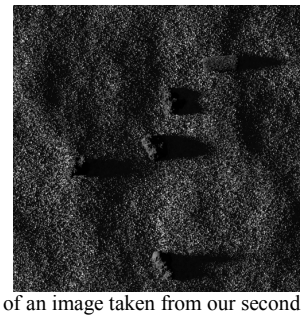

Fig. 21. Example of an image taken from our second operational dataset

Table 2 summarizes the results on this second and more challenging dataset. As expected, the quality of the results is far lower than for the first operational dataset. Hence, we are presenting only the F-Score alongside the values of the True Positives (TP), False Negatives (FN) and False Positives (FP) which explain why the F-Score is low.

Table 2: Results for the second operational dataset

F-Score	True Positives	False Negatives	False Positives
0.26	106	33	566

From the results obtained on the second operational dataset, one can see that the number of FP is sky-rocketing. This is explained by the fact that our pipeline is not discriminating rocks as part of the background. When we take a look at the operational snippets of background extracted for the background dataset, we notice that those snippets do not include any rocks. This is explained by the fact that the DAE having doubts about the regions of the images where rocks are, it is labeling those regions as explained in III.B. Hence it does not allow our pipeline to extract snippets from those areas. To overcome this problem, we decided to orient our work towards semi-supervised learning, taking advantage of those areas labeled as non-background.

V. Conclusion and Future Work

In this article, we have presented our work on Underwater Mine Detection in SAS images. We have setup a new pipeline of operation composed of (i) a synthetic image generator, (ii) a DAE, (iii) a background dataset extractor and (iv) a DCNN as illustrated on **Fig. 1**. Through our tests, we show that this pipeline is very promising as results on a first simple operational dataset are encouraging. However, work needs to be pursued in order to be able to tackle more complex and challenging backgrounds. We foresee a few tasks to be carried out on both the image generation part and the machine learning part.

Following the comments from sonar experts, some work needs to be done to improve the photo-realism of our synthetic images. Typically, the sonar echo generated by the objects on the seabed is not properly designed yet, which will impact the performance of our system when dealing with real data. Furthermore, so far only cylinder mines have been used. We plan to work with other mine shapes. Finally, regarding the synthetic dataset, optimizing its generation and use in order to minimize the number of images to create while maximizing their representativeness and variability is still a task that can be studied and significantly improved.

On the machine learning part, the sonar we are focusing on is capable of generating one-shot multi-view image sets in a single path therefore enabling interesting data fusion capabilities that we plan to explore in order to improve the false alarm and successful detection rates. We also think that the threshold criteria used to discriminate the output of our DAE can be greatly improved to refine the clustering between background and potential threats. Finally, we believe that a semi-supervised approach for the CNN, taking advantage of the potential threat cluster labeled by the DAE, could drastically enhance the classification performances of the whole pipeline.

Obviously, we also need to test our system with real data captured during actual trials. This step only will fully validate our new concept of operation.

Acknowledgment

The authors would like to thank the experts from Thales Underwater Systems for their support and valuable inputs to improve the quality of our synthetic data. Furthermore, this work would have been far more challenging without the excellent contribution of our team of interns: Quentin Chan-Wai-Nam, Alexis Coquoin, François Darmon and Bruno Lecouat.

References

[1] "Automatic classification for MCM systems", I. Quidu, N. Burlet, J.-P. Malkasse, F. Florin, Oceans 2005 Europe, Brest, France, 20-23 June 2005

[2] "On Rendering Synthetic Images for Training an Object Detector", Artem Rozantsev, Vincent Lepetit, Pascal Fua, 2015

[3] "Simulation and 3D Reconstruction of Side-looking Sonar Images", E. Coiras, J.Groen, 2009

[4] "Development of Image Sonar Simulator for Underwater Object Recognition", Jeong-Hwe Gu, Han-Gil Joe1 and Son-Cheol Yu, 2013

[5] www.blender.org

[6] "Auto-association by multi-layer perceptrons and singular value decomposition" H.Bourlard, Y.Kamp, 1988

[7] "Extracting and composing robust features with denoising autoencoders", Pascal Vincent, Hugo Larochelle, Yoshua Bengio, Pierre-Antoine Manzagol, 2008

[8] "Very deep convolutional networks for large-scale image recognition" Karen Simonyan, Andrew Zisserman, 2015