VGG16-based approach for side-scan sonar image analysis

Antoni Jaszcz

Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44100 Gliwice, Poland

Abstract

Side-scan sonar (SSS) images are based on the reflection of the signal from an underwater object. As a result, such data may contain a lot of noise or ambiguous objects to be analyzed. In this paper, we propose a simple system for analyzing such images and classifying objects on them. For this purpose, the convolutional neural network and learning transfer (VGG-16) were used. Such a network model was preceded by the process of dividing the sonar image into smaller fragments in order to avoid the omission of objects by reducing the size. The proposed solution was tested on a dedicated database, which made it possible to evaluate the proposal and reach the high accuracy of the used network. The obtained research results were analyzed and discussed due to the possibility of implementing such a model in practice.

Keywords

Side-scan sonar, learning transfer, geospatial data, vgg16, machine learning

1. Introduction

Recent years have brought enormous growth in image analysis through the use of artificial neural networks. In particular, this is due to convolutional neural networks (CNNs) that automatically detect features and perform classifications. However, the use of such networks comes with additional requirements. Before classification, each network must undergo a training process on a dedicated database. For the network to achieve high accuracy scores, it requires an enormous amount of data. Hence, neural networks are called data-hungry algorithms. However, quite often there are situations when the number of data is small and it is difficult to obtain more samples. For this purpose, techniques of augmentation [1] or learning transfer [2] can be used.

An interesting case of such images is the ones gathered underwater [3, 4]. An example is a side-scan sonar (SSS), which is obtained by visualizing the signal being reflected from objects. Such data are exposed to large amounts of noise, hence an important element is the construction of systems based on neural networks and allowing to increase the accuracy of these images [5]. In this paper, the authors proposed a solution based on a generative adversarial network. The image was processed by down-sampling and then recreated with the up-sampling approach. A similar idea was presented in [6], where a new type of such network was modeled and called s^{2}rgan. However, the detection of objects and classification of them is also important. Such analysis can bring information about the state of the seafloor. One approach is to the analysis of smaller parts to find a region of interest and then applied neural networks to find areas [7]. Again in [8], the automatic overlapping and segmentation techniques were developed. Learning transfer based on yolov5 was also used to detect some objects on SSS images [9]. Similar research was conducted by the use of different CNN models for many types of images [10, 11]. Again in [12], the authors used a similar approach, but classify small images. Segmentation, classification tools can be used for tracking bottom [13, 14]. A segmentation of the image can be done by the use of recurrent residual CNN and self-guidance module [15, 16]. A field-programmable gate array for SSS images based on neural networks was proposed in [17]. SSS data are not only used for finding some object but also to combine its feature to create a surface in 3d projection [18]. Another application is underwater communication for compressed SSS transmission [19].

In this paper, we propose a simple system based on automatic splitting SSS images into smaller parts and using them to train VGG-16. As a result, an automatic system for classifying sonar data is modeled. The main contribution of this paper is:

- the methodology for analyzing SSS images,
- the use of pre-trained model called VGG-16 to classify SSS data,
- the method of automatically adding a sample to the database if the probability of belonging is high enough, which allows increasing the number of samples in the database.

2. Methodology

In this section, the proposition of the system to analyze side-scan sonar images is described (Fig. 1). The incoming sonar data are split into smaller fragments and then processed by a convolutional neural network. If the classification result shows with high probability one of

IVUS 2022: 27th International Conference on Information Technology

aj303181@student.polsl.pl (A. Jaszcz)

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

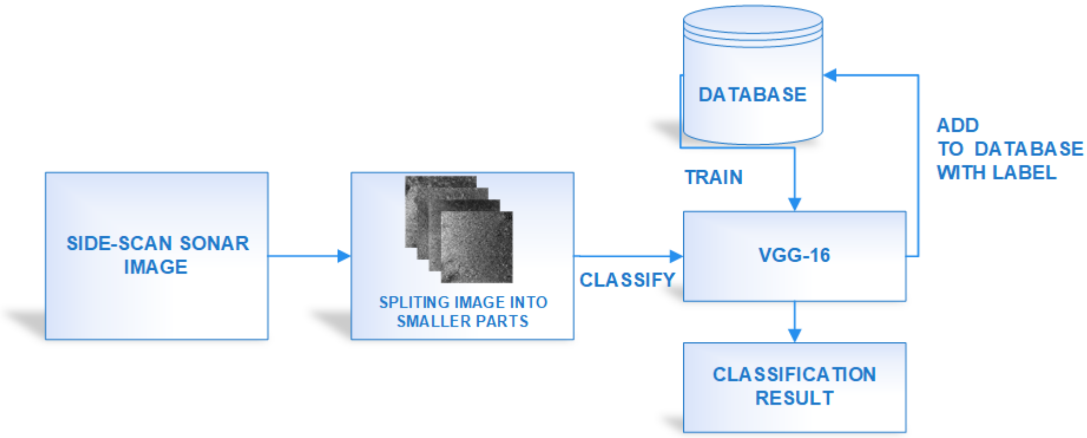


Figure 1: Visualization of the proposed approach based on the pre-trained convolutional network model

the classes (higher than 0.9), the sample is placed in the database and used in the next training.

2.1. Image division

At first, the image is processed. It is done because the sample size has to be reduced to that of the first network layer before further processing. Hence, a large sonar image, when reduced in size, can distort or simplify certain elements. As a result, this may result in much worse learning outcomes and subsequent classification. To prevent this, we propose a simple algorithm for subdividing the sonar image into smaller fragments (see Alg. 1). The main idea is to cut the image in half into two samples. The reason is an area that was not visible to sonar. As a result, two samples are created from one image (presenting the left and right sides of the sonar). Then, the sample is divided by the specified height and passed on. Note that if, after the cutting process, the sample is larger than the first layer of the network, it will be reduced to that size.

Algorithm 1: SSS-image division algorithm

```
Data: Sonar image
Result: Image samples of a given size
while not past the bottom of the image do
    while not past the right edge of the image do
        cut and save a sample of desired shape
          // 256x256 pixels
        move right by the desired amount of
          pixels                    // 100 pixels
    end
    move back to the left edge and shift down by
      the desired amount of pixels         // 100
      pixels
end
```

2.2. CNN

The split image is processed by convolutional neural network. Its structure consists of three different layers: convolutional, pooling, and fully-connected (dense). The convolutional layer change the image I by the filter k (a matrix of size $p \times p$). It is done by the convolutional operator (here marked as $*$) defined as:

$$k * I_{x,y} = \sum_{i=1}^{p} \sum_{j=1}^{p} k_{i,j} \cdot I_{x+i-1,y+j-1}. \qquad (1)$$

The values of filter k are found during the training process. The main task of this image is to modify the image and extract the features on it (therefore the changed image is called a feature map).

The second type is known as pooling and it has one task - reduce the image size. This reduction is based on mathematical functions like $\max(\cdot)$. The operation is understood as a selection of one pixel in a grid that satisfies this function. Of course, the grid is moved over the entire image until the last pixel is covered with it. The minimum size of such a mesh is 2×2.

Third layer is full-connected that presents a classic column of neurons that receives a numerical values and process them to next neurons by the connection. Each connection between two neurons has a weight w and this value is modified in the training process. The mathematical formulation of it is:

$$f\left(\sum_{i=0}^{n-1} w_{i,j} x_i\right), \qquad (2)$$

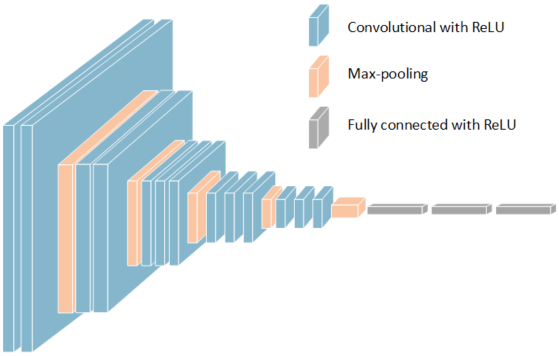


Figure 2: Visualization of the pre-trained model known as VGG16

where $f(\cdot)$ is an activation function, i, j are the indexes of two neurons in adjacent layers.

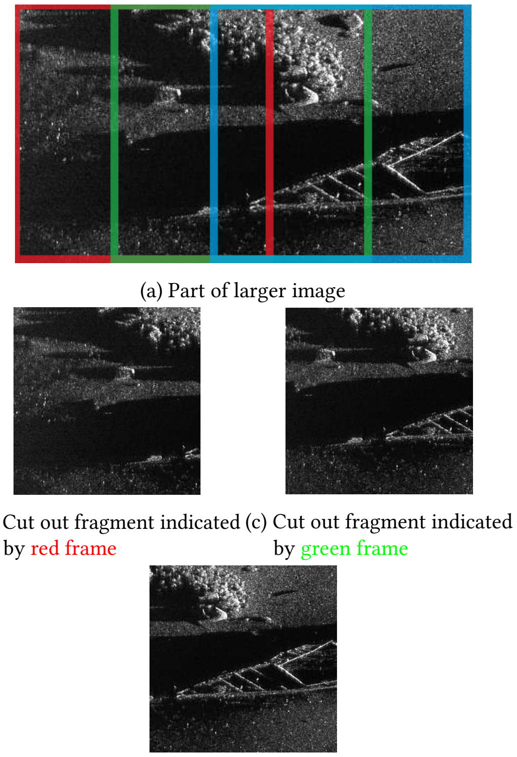
The main reason for using CNN for image recognition is feature extraction. The more consecutive layers the neural network has, the more abstract features can be considered. Basic features extracted by the model first are generally lines along the axis of the image, so features such as vertical, horizontal and diagonal lines. Then some more advanced characteristics are discovered, like shapes. For example rectangles, circles, straight lines, etc. The deeper the model goes, the more complex and abstract newly extracted features become, to the point, that we as humans do not even consider them or are unaware of them.

In the proposed methodology, we propose using a pre-trained model VGG-16 [20] that is presented in Fig. 2.

The problem of object recognition can be universalised to some extent in terms of object detection. That's why pretrained models are commonly used. Such models are trained to extract unique features related to the sought objects. The most popular database for training those models is ImageNet database, containing over a million manually labeled images of thousand classes. By incorporating in such pretrained models, training time can be greatly reduced and focus of the training can be shifted towards detection of specific objects, rather than feature extraction. Some of the ImageNet pretrained models include:

- VGG-family
- ResNet50
- Inception V3
- Xception

In our experiments, we chose VGG-16 because it is most commonly used model, which means that result comparison to other models in the field can be easier. It also incorporates samples assessed with certainty above certain threshold, which benefited our research greatly.



(a) Part of larger image

(b) Cut out fragment indicated by red frame

(c) Cut out fragment indicated by green frame

(d) Cut out fragment indicated by blue frame

Figure 3: Windowing technique visualised on a small part of an image

3. Experiments

In this section, the database, CNN configuration, obtained results and discussion are presented.

3.1. Database and data preparation

In this paper, side-scan sonar images of a river floor were used. The data were gathered between two water channels in north-western Poland. In order for a deep learning model to identify objects on a riverbed, target samples were required. To achieve that, the objects had to be hand-picked and manually classified. To facilitate the process of doing so, the images were automatically split into smaller parts of a fixed size of 256 by 256 pixels. It is worth mentioning, that those parts could overlap with one another. In this paper, the next two samples in the same row were 100 pixels afar from each other, as well as considered row (each following row was 100 pixels lower). This can be presented by a simple image division algorithm shown in Alg. 1. The windowing technique has been visualised in Fig. 3.

After obtaining samples, they were then manually classified into 3 categories, depending on what could be seen in the picture. Those categories are:

1. object - anything distinguishable as a larger object laying on a riverbed. That includes ship/vehicle wrecks (Fig. 4), logs and pipes (Fig. 5), etc. In other words, single objects of considerable size appeared in the river.

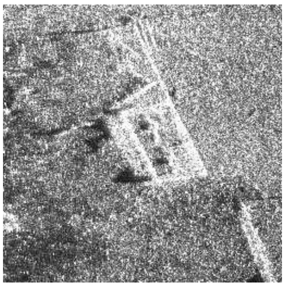


Figure 4: Possible wreck of a car (object class)

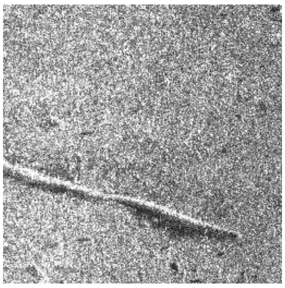


Figure 5: Log or pipe (object class)

2. sand - a plain surface of the riverbed, on which nothing particular can be picked out. An example of such a sample is shown in Fig 6

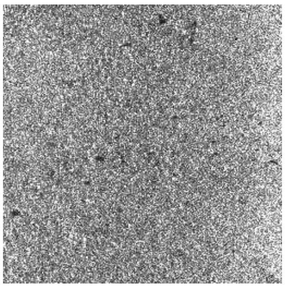


Figure 6: Sand (sand class)

3. rubble - a plain surface of the riverbed with a visible and considerable amount of debris. The goal for adding this class into consideration was to make the model able to detect cluttered areas and distinguish them from the clear ones.

In total, there were 665 samples in the final database, 55 of which were classified as $objects$, 352 as $sand$ and 257 as $rubble$. Next, 226 samples were randomly chosen and put into validation group. Class distribution of those objects can be read from Fig. 8, and is as follows:

- 36 $object$ samples,
- 124 $sand$ samples,
- 63 $rubble$ samples.

The rest of 439 samples formed the training set for the neural network.

It is crucial to mention, that due to the nature of side-scan sonar images, some samples turned out to be inadequate. There was a concern, that they would bring nothing more, than confusion to the model. As a result, it was decided that if a sample is too confusing (in terms of which class it shall belong to), contains a considerable amount of the boat's passage area (a thick, black line stretching along the edges of a sonar image and crossing its center, an example of which can be seen in Fig. 7) or if its quality was concerning (the image was greatly distorted), it should be removed from the dataset.

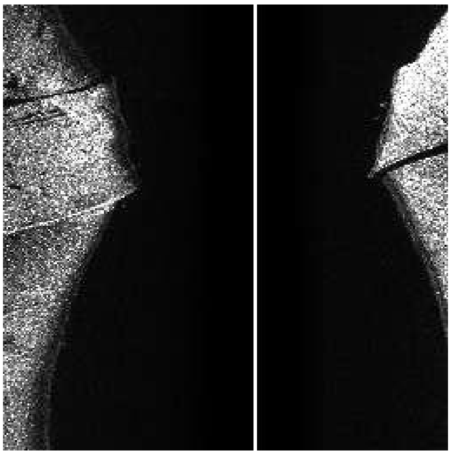


Figure 7: An example of a sample, that was removed

In the experiments, pre-trained VGG-16 model connected to the dense neural network with input augmentation was used. The structure of the model is displayed below:

1. *Augmentation layer* - random horizontal and vertical flip and random rotation
2. *VGG-16 layer*
3. *deep neural layer*
    - flatten layer (),

- dense layer, 50 neurons, *ReLU* activation,
- dense layer, 20 neurons, *ReLU* activation,
- dropout layer (threshold: 0.5)

4. *Output layer* - a dense layer with 3 neurons (one for each output class)

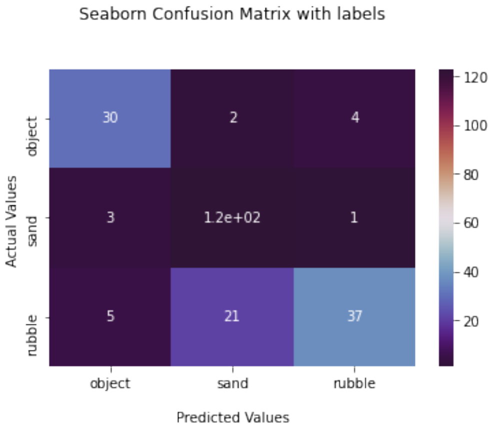


Figure 8: Multi-class confusion matrix

3.2. Results

In the Tab. 1, calculated metrics, such as $accuracy$, $precision$, $specificity$, $recall$ and $f1 - score$ are displayed for each class, as well as their mean value. The values were calculated with following formulas:

- Accuracy:

$$\alpha = \frac{TP + TN}{TP + FP + TN + FN}, \tag{3}$$

- Precision:

$$\psi = \frac{TP}{TP + FP}, \tag{4}$$

- Recall:

$$\rho = \frac{TP}{TP + FN}, \tag{5}$$

- Specificity:

$$\sigma = \frac{TN}{TN + FP}, \tag{6}$$

- F1-Score:

$$\frac{1}{f1} = 0.5 \cdot \left(\frac{1}{\psi} + \frac{1}{\rho}\right), \tag{7}$$

where
TP - true sample predicted as true,
TN - false sample predicted as false,
FP - false sample predicted as true,
FN - true sample predicted as false.
Please note that in this paper, we consider multi-class detection, which also applies to the results. Thus, by true, a correct class assignment is meant. That is, at a given time during assessment, only class is considered $true$ and the others are $false$. That applies for every class and is visible in Fig. 8 and its impact on obtained metrics is displayed in Tab. 1

Table 1
Calculated metrics

	precision	recall	specificity	f1-score
	0.7895	0.8333	0.9579	0.8108
object		accuracy=0.9381		
	0.8425	0.9685	0.7677	0.9011
sand		accuracy=0.8805		
	0.8810	0.5873	0.9693	0.7048
rubble		accuracy=0.8628		
	0.8376	0.7964	0.8983	0.8056
average		accuracy=0.8938		

As seen in the Fig. 8 and Tab 1, the model achieves decent accuracy, when detecting objects. It also performs very well when detecting non-$object$ samples, which is indicated by high specificity. In terms of identifying $sand$, the accuracy has dropped. However, its recall is high. That suggests, that model correctly assigned most of the $sand$ samples. However, it was at cost of precision, which is not great. As can be observed in Fig. 8, 21 of the $rubble$ samples were categorized as $sand$, which indicates, that the model aggressively assesses $sand$-class objects and requires some more balance. As a result, $ruble$'s metrics are all fairly low, accuracy included. All of that implies, that the $ruble$ detection is the weakest link of the model.

In Fig. 9 and Fig. 10, a trade off between classifying train and test group (both derived from aforementioned $traininggroup$, the former counting samples to learn from, and the ladder having samples to validate the outcomes) in terms of $loss$ (9) and $accuracy$ (10) during the subsequent epochs of training is presented.

4. Conclusions

The analysis of the bed of any body of water using side-scan sonar requires great image data to be manually reviewed. To narrow the dataset needed to be hand-checked, artificial intelligence can be used to pick out fragments of the images with sought objects (for example, shipwrecks). For this purpose, the use of a pre-trained convolutional neural network (VGG-16) connected to a dense neural network was presented. In the research,

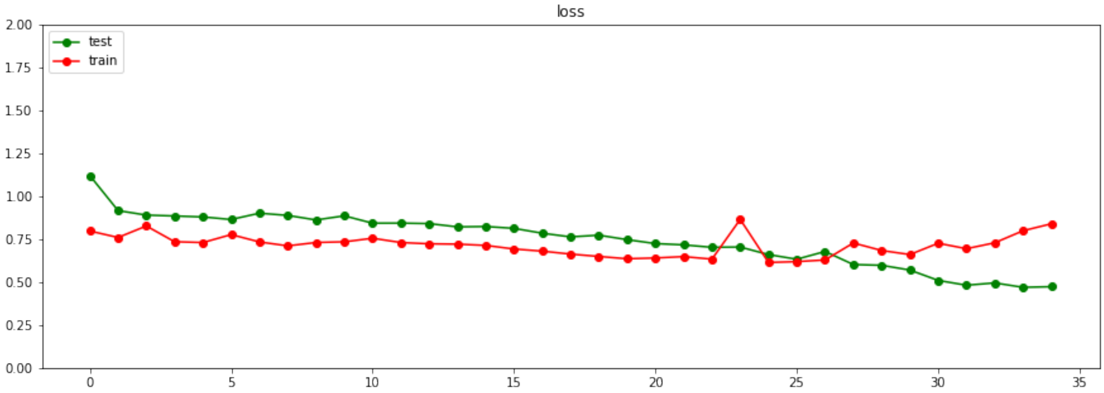


Figure 9: Trade off between $test$ and $train$ loss value during the subsequent epochs of training

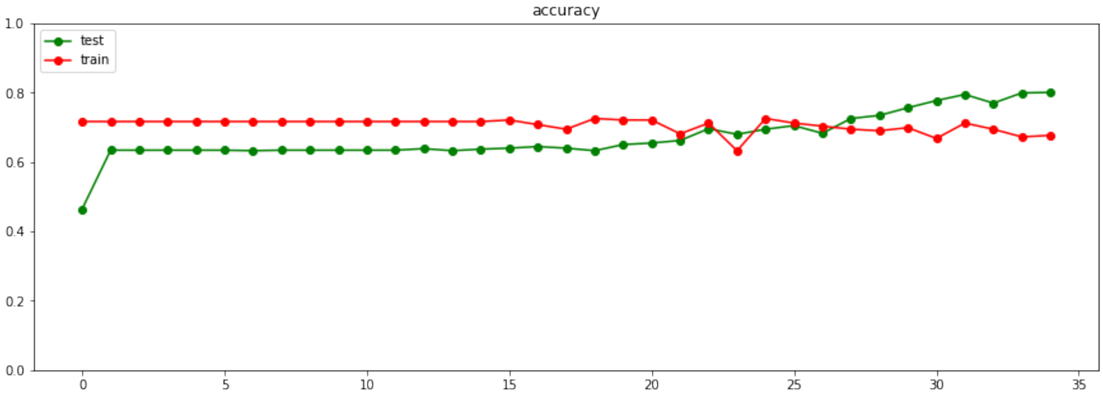


Figure 10: Trade off between $test$ and $train$ accuracy value during the subsequent epochs of training

larger images were divided into target samples and augmented during training (by randomly flipping and rotating them). The obtained results indicate that such a model can satisfactorily distinguish objects of unconventional shapes for the riverbed. It is also suggested, that with some improvements, the model can be used in more advanced riverbed analysis by detecting objects harder to distinguish from plain river's ground such as rocks, water weeds, silt, etc., as well as their percentage in the whole image. The model tested in this paper was aggressive towards classifying samples containing ruble as plain ground. That said, it was not the case with object detection. The reason for that is as mentioned above, little and the hardly-noticeable difference between several, distorted small objects and plain surfaces in computer vision.

Acknowledgements

This work is supported by the Silesian University of Technology by the mentoring project.

References

[1] G. Chandrashekar, A. Raaza, V. Rajendran, D. Ravikumar, Side scan sonar image augmentation for sediment classification using deep learning based transfer learning approach, Materials Today: Proceedings (2021).

[2] D. Połap, Fuzzy consensus with federated learning method in medical systems, IEEE Access 9 (2021) 150383–150392.

[3] W. Kazimierski, G. Zaniewicz, Determination of process noise for underwater target tracking with forward looking sonar, Remote Sensing 13 (2021) 1014.

[4] N. Wawrzyniak, G. Zaniewicz, Detecting small moving underwater objects using scanning sonar in waterside surveillance and complex security solutions, in: 2016 17th International Radar Symposium (IRS), IEEE, 2016, pp. 1–5.

[5] P. Shen, L. Zhang, M. Wang, G. Yin, Deeper super-resolution generative adversarial network with gradient penalty for sonar image enhancement, Multimedia Tools and Applications 80 (2021) 28087–28107.

[6] H. Song, M. Wang, L. Zhang, Y. Li, Z. Jiang, G. Yin, S2rgan: sonar-image super-resolution based on generative adversarial network, The Visual Computer 37 (2021) 2285–2299.

[7] D. Połap, N. Wawrzyniak, M. Włodarczyk-Sielicka, Side-scan sonar analysis using roi analysis and deep neural networks, IEEE Transactions on Geoscience and Remote Sensing (2022).

[8] X. Shang, J. Zhao, H. Zhang, Automatic overlapping area determination and segmentation for multiple side scan sonar images mosaic, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14 (2021) 2886–2900.

[9] F. Yu, B. He, K. Li, T. Yan, Y. Shen, Q. Wang, M. Wu, Side-scan sonar images segmentation for auv with recurrent residual convolutional neural network module and self-guidance module, Applied Ocean Research 113 (2021) 102608.

[10] W. Yanchen, Sonar image target detection and recognition based on convolution neural network, Mobile Information Systems 2021 (2021).

[11] D. Połap, M. Woźniak, Meta-heuristic as manager in federated learning approaches for image processing purposes, Applied Soft Computing 113 (2021) 107872.

[12] X. Qin, X. Luo, Z. Wu, J. Shang, Optimizing the sediment classification of small side-scan sonar images based on deep learning, IEEE Access 9 (2021) 29416–29428.

[13] X. Qin, X. Luo, Z. Wu, J. Shang, D. Zhao, Deep learning-based high accuracy bottom tracking on 1-d side-scan sonar data, IEEE Geoscience and Remote Sensing Letters 19 (2021) 1–5.

[14] G. Zheng, H. Zhang, Y. Li, J. Zhao, A universal automatic bottom tracking method of side scan sonar data based on semantic segmentation, Remote Sensing 13 (2021) 1945.

[15] Y. Yu, J. Zhao, Q. Gong, C. Huang, G. Zheng, J. Ma, Real-time underwater maritime object detection in side-scan sonar images based on transformer-yolov5, Remote Sensing 13 (2021) 3555.

[16] N. Wawrzyniak, M. Włodarczyk-Sielicka, A. Stateczny, Msis sonar image segmentation method based on underwater viewshed analysis and high-density seabed model, in: 2017 18th International Radar Symposium (IRS), IEEE, 2017, pp. 1–9.

[17] C. Wang, Y. Jiang, K. Wang, F. Wei, A field-programmable gate array system for sonar image recognition based on convolutional neural network, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 235 (2021) 1808–1818.

[18] M. Włodarczyk-Sielicka, I. Bodus-Olkowska, M. Łącka, The process of modelling the elevation surface of a coastal area using the fusion of spatial data from different sensors, Oceanologia (2021).

[19] J. Cui, G. Han, Y. Su, X. Fu, Non-uniform non-orthogonal multicarrier underwater communication for compressed sonar image data transmission, IEEE Transactions on Vehicular Technology 70 (2021) 10133–10145.

[20] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015).