Received 23 June 2023, accepted 7 July 2023, date of publication 14 July 2023, date of current version 20 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3295693

RESEARCH ARTICLE

Bilinear Pooling With Poisoning Detection Module for Automatic Side Scan Sonar Data Analysis

DAWID POŁAP[1], (Member, IEEE), **ANTONI JASZCZ**[1,3], **NATALIA WAWRZYNIAK**[2], **AND GRZEGORZ ZANIEWICZ**[2]

[1]Faculty of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland
[2]Faculty of Navigation, Maritime University of Szczecin, 70-500 Szczecin, Poland
[3]Marine Technology Ltd., 81-521 Gdynia, Poland

Corresponding author: Dawid Połap (dawid.polap@polsl.pl)

This work was supported in part by the Rector Proquality Grant at the Silesian University of Technology, Poland, under Grant 09/010/RGJ22/0067 and Grant BKM-505/RMS2/2023; and in part by the National Centre for Research and Development (NCBR) of Poland under Grant LIDER/4/0026/L-12/20/NCBR/2021.

ABSTRACT Side-scan sonar (SSS) images are difficult for automatic analysis due to the acoustic measurement parameters as well as the number of different objects that can be distant. In addition, there is a risk that the seabed analysis application may be attacked. For this purpose, we propose a solution based on convolutional neural networks with bilinear pooling in order to achieve higher values of classification accuracy. Bilinear pooling merge data from two networks and return classification results. The first network's branch receives the original image and the second one after applying the superpixel method. This approach allows to focus on different types of features. In addition, we introduced a mechanism of poisoning detection that analyze images and results from the network. For the evaluation process, we used the real SSS images obtained between two water channels in Szczecin city in north-western Poland. The importance of scientific research indicates the accuracy of the analysis as well as the safety of the measurements performed.

INDEX TERMS Classification, convolutional neural networks, machine learning, poisoning detection, side-scan sonar images.

I. INTRODUCTION

At the bottom of a river, sea, or even lake, there may be various objects that constitute a threat and sometimes even prevent safe navigation. Unfortunately, the analysis of the seafloor may involve deep measurements. Various methods can be used to measure it and located objects. For this purpose, the side-scan sonar (SSS) has great interest because of its effectiveness. The idea of operation can be described by emitting an acoustic beam in a plane perpendicular to the direction of sonar movement. As a result of SSS, the obtained images are in high resolution and have the maximum effect of the formation of a hydroacoustic shadow. An additional advantage is the possibility of towing sonar below the thermocline layer, where absorption of acoustic energy is quite common. Despite these advantages, such sonar also has disadvantages. Such problems include the problem of estimating obstacles that may contribute to significant damage to the equipment. The second problem is the difficulty of accurately analyzing the position of the equipment in a horizontal position.

Seafloor images can be very useful in analyzing the state of the bottom, finding some objects like anchors, wrecks, etc. In addition, topographical aspects are important to visualize, classify or map the underwater area. Such activities result in obtaining a better understanding of the underwater environment, among others, for the purposes of navigation or construction of underwater infrastructure. Hence, our research focuses on the possibilities of automatic analysis of such data. However, one scan in some areas can produce many

The associate editor coordinating the review of this manuscript and approving it for publication was Chengpeng Hao.

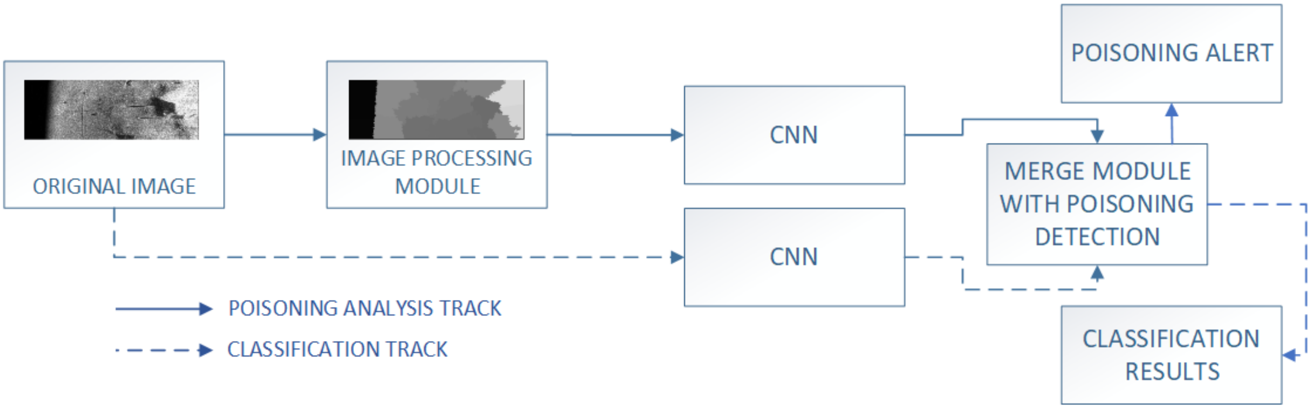


FIGURE 1. Visualization of the SSS image processing in the proposed method. The original data is processed by the image processing module to simplify the contained information. Then a parallel classification of the original sample and the processed one is made. The obtained classification results are taken for analysis in the merge module, where a decision is made on the classification and detection of potential poisoning.

images of high resolution that are merged into one. Moreover, such an image is difficult in terms of automatic analysis. The resolution, the noises, the shadows, and the different shapes are hard to process by classic known algorithms. Therefore, machine learning solutions like convolutional neural networks (CNN) are commonly used to analyze SSS images. It must be noted that large images are not possible to process, so there are two approaches: reduce the size or split it into smaller parts [1], [2].

Using a machine learning solution for analyzing images is one of the popular approaches due to its high accuracy. It can be seen in the example of performed tasks by neural networks like segmentation and classification. The use of segmentation allows to locate and extract the important object. One such example is to use mask recurrent CNN that generates an object mask which is a region of interest of found object [3]. A similar solution is based on deep CNN with a recurrent network that processes SSS images and returns segmentation outputs prediction. These results are processed by a self-guidance module that decides if the image is a segmentation prediction or ground truth.

Mentioned segmentation and denoising images are some of the basic tasks in analyzing SSS images. Segmented areas can be used in further analysis like identification and classification. Similar to the earlier task, machine learning is one of the most commonly used tools. However, it should be noted that in many cases it is not just a simple classifier but with some modification or additional modules that improve the performance of the algorithm. Except for neural approaches, meta-estimators like the Adaboost cascade can be used to classify some objects. An example of such research was dedicated shipwreck identification, where different fractal texture features were extracted and then classify [4]. Classification with CNNs is facilitated quite often by the possibility of using learning transfer. It is an over-trained network model on large databases, which only gets further trained, mainly of the last layer. Yolov3 is one of the over-trained models that has been used to classify objects on SSS images [5]. Under the proposed method, the authors added also a k-means clustering algorithm to reset the prior frame parameters. It helps lower the time computational of the proposal. Another learning transfer model was used in the real-time application of analyzing SSS images [1]. Similar tools were shown in [6], where both mentioned pre-trained models were used to identify a strategy that distinguishes three zones in the processed image – two acoustic and one dead between them. For the classification of the reef, mud, and sand waves a CNN module with residual/dense blocks was applied to reach a high accuracy level [2]. In this study, the methods analyze small parts extracted from the SSS measurement of the sea bottom. Again collaborative learning transfer for obtaining CNN's parameters was introduced to improve the time needed to train and improve the accuracy [7]. Automatic sonar analysis system is based mainly on a few steps: image preprocessing including normalization and georeferencing, then different algorithms for detection and classification of found areas [8]. Other models use different ideas to extract important information, for instance, temporal correlation features [9]. The authors applied encoder decoder model for mapping purposes.

Besides the processing of SSS images, there are also aspects of security of solutions based on machine learning. In the case of image classifiers, poisoning attacks are very common and hard to detect. Poisoning involves changing labels, which causes the classifier to learn to recognize individual classes incorrectly. In addition, the image may contain some noise or additional features (invisible to humans) that will cause incorrect training and subsequent classification [10]. Recent attempts at such attacks have also been widened by using a separate network to find adversarial patches that cause a bug. However, some research on that matter is made. The current state of the detection method is based on adding authentication and provenance [11]. Another approach is to introduce a specific index for each object using training datasets and compare it to find the differences [12]. These types of attacks are becoming more and more popular in the case of applications that use machine learning methods. In order to ensure correct operation as well as data security, it is important to implement elements that detect and prevent poisoning attacks.

Based on this motivation, we propose a solution based on analyzing parts of SSS images. Incoming image is processed by two similar CNNs, the first one takes a processed image by superpixel techniques and the second one original sample. The results of classification from both networks are gathered by merge module for analyzing it. This step compares the obtained results and decides if there is a potential poisoning of the image. It is done by applying a probabilistic module that processes the probability of belonging to all classes indicated by networks. If there is no poisoning attack, the classification result is returned. In the other case, the system returns information about the attack and deletes the sample from the database. It is also an extension of our previous study [13]. The main contributions of our proposal are:

- bilinear pooling in CNN for SSS images classification,
- two separate CNNs process one image to eliminate potential poisoning attack,
- merge module based on probability analysis to increase the security of the implemented machine learning solution in real-time scenarios.

II. DESIGNED METHODOLOGY

Our proposition is based on processing SSS images and returning classification information and information if there is a poisoning attack. We describe the proposition in the form of individual stages: image processing module, bilinear convolutional neural network, and final merge module with poisoning detection tool. A simplified visualization of the proposal is shown in Fig. 1.

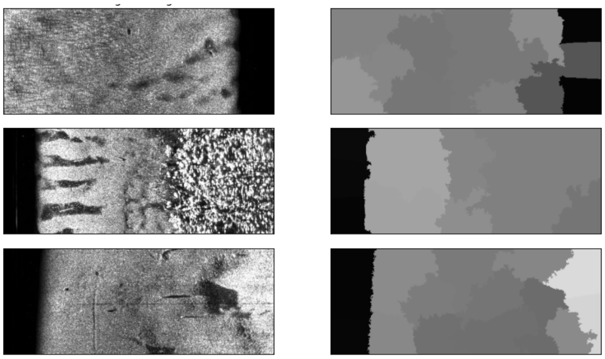


FIGURE 2. Original samples collected while measuring the bottom of the river (on the left) and corresponding superpixel forms (on the right).

A. IMAGE PROCESSING MODULE

The classic approach assumes that the sample is reduced to a certain size of the input layer of neural networks. However, such a sample may be subject to modifications or attacks. For this purpose, we suggest adding an image processing module that will simplify the image and ignore any possible modifications. To make it possible to remove the specific information about pixels, but retain the global shapes, a superpixel technique can be applied. By definition, a superpixel is a group of pixels that has the same characteristic (for instance intensity or saturation). Assume, we have image I of size $w \times h$ and each pixel is represented in the CIELab color model. The most common algorithm is called SLIC [14], [15] and it is based on the number of superpixels K and the image that contains N pixels. The idea assumes that an image is split into a grid of smaller intervals ($S = \sqrt{N/K}$), wherein the center of each will be a superpixel center. At first, centers are selected as $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ (where $k = [1, K]$ in each interval, l_k, a_k, b_k are CIELab color components known as brightness, the color value from green to magenta, the color value from blue to yellow, and the other two components represent pixel coordinates). According to the assumptions of the algorithm, it does not use the Euclidean norm to determine the distance, but the value is defined as:

$$D_s = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} + \frac{m}{S}\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}, \quad (1)$$

where the value is calculated between two centers defined as $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ and $C_i = [l_i, a_i, b_i, x_i, y_i]^T$, m is a parameter to control the compactness and it is assumed to take $m = 10$ [14]. The choice of measure is defined by the CIElab color space, where the perceptual limit of the color distance quite often outweighs their similarity.

After sampling and selecting centers, we move the centers to a location with a lower gradient relative to the grid of size 3 × 3. Then each pixel is associated with the closest cluster center. After all, pixels have been assigned to the clusters, a new center is determined as the average vector of all pixels in that cluster. This operation is performed until it converges.

Such an approach allows modifying of incoming SSS image and processing to their superpixel form. An example of created samples is shown in Fig. 2.

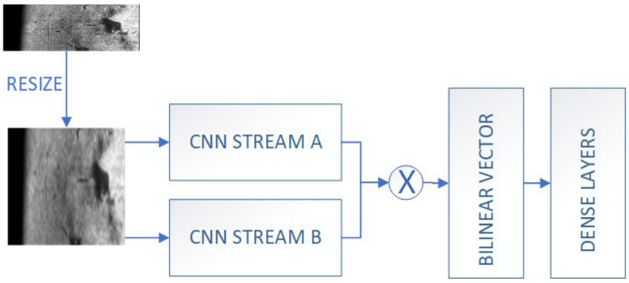


FIGURE 3. Bilinear CNN for SSS images where streams are based on two pretrained VGG16 and bilinear layer with dense ones.

B. BILINEAR POOLING

The next stage is to process the image by bilinear CNN. The network has a similar construction to the classic CNN because of the layer types like convolutional, pooling, and dense. However, two modifications are made. The first one is to duplicate the convolutions/pooling blocks (called streams) and bilinear layer that gathers results from both streams (see Fig. 3). From the mathematical side, the bilinear CNN can be described as:

$$f_{A,B} = \mathcal{P}\left(f_A^T \cdot f_B\right), \quad (2)$$

where f_A and f_B are the feature maps gained from streams A and B. The feature maps are merged by the use of the matrix outer product and then the average pooling (here the function $\mathcal{P}(\cdot)$ is the pooling operation) is used to reshape the results to a bilinear vector. Then this vector can be processed by the dense layers.

In our proposition, the network structure is based on the pre-trained model VGG16 with categorical cross-entropy as a loss function. The idea is to use two of these networks. The first one will be trained with the original samples, and the second with the images after superpixel processing. Each of these networks will return the vector of probability that the evaluated sample belongs to a specific class. These vectors are processed further. In that stage, the classification results cannot be considered classified as there is a chance that an attack has occurred.

C. MERGE MODULE WITH POISONING DETECTION TOOL

Both neural networks return the probability vector of the analyzed sample belonging to each class, which can be represented as ξ_A and ξ_B where $\xi_{j,A} = \left[p_0^j, p_1^j, \ldots, p_{n-1}^j\right]_A$ (j is the number of samples, n is the number of classes). Let us mark the network trained with superpixel images as A, and the second one B. The sample classification result will indicate the highest probability in the vector ξ_B. However, an attack of poisoning may have occurred. Hence, it is important to verify that the classification results are correct.

There are two scenarios - the database was poisoned during training or a sample during the classifying phase. In the case of the first situation, after each training iteration, the model should be checked. The basic parameter that could be used is the accuracy of both models. However, these two accuracies cannot be compared to each other as training does not have to be identical in both cases. Hence, the detection of poisoning relies on the analysis of samples in the testing dataset D_T that do not take part in the training process.

The mechanism is based on comparing the results from the current and previous iterations (marked as $t-1$ and t). The comparison is made on a test dataset, so all samples will be evaluated as:

$$\sum_{i=0}^{|D_T|-1} F\left(\xi_{A,i}^t, \xi_{B,i}^t\right) \geq \sum_{i=0}^{|D_T|-1} F(\xi_{A,i}^{t-1}, \xi_{B,i}^{t-1}), \tag{3}$$

where a function $F(\cdot)$ F is the similarity index, defined as:

$$F\left(\xi_A, \xi_B\right) = \begin{cases} 1 & \text{if } index(\max(\xi_A)) == index(\max(\xi_B)) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

If Eq. (3) is not met, it means that the classification result for samples from D_T has deteriorated. Then there is a possibility that the set has been poisoned.

In the case when the models are trained, a sample can be verified by simple conditions:

$$\begin{cases} \text{if} & index(\max(\xi_A)) == index(\max(\xi_B)) \\ & \text{return classification result,} \\ \text{else} & \text{return information about potential poisoning,} \end{cases} \tag{5}$$

where $index(\cdot)$ returns the index of parameters (here, it will be an index of class where the probability is the highest). This condition means, that if the result from both networks indicates the same class, then there is no poisoning. However, if the classes are different, then it means that the sample could be manipulated. Then, the alert about a potential poisoning attack is returned.

III. EXPERIMENTS

In this section, we describe the data collected in Szczecin in Poland (see Fig. 4). Then we describe the analysis of used CNN models to classify original images and superpixel ones. In the last part of this section, we analyzed the poisoning rate to its detection during the training phase and validation.

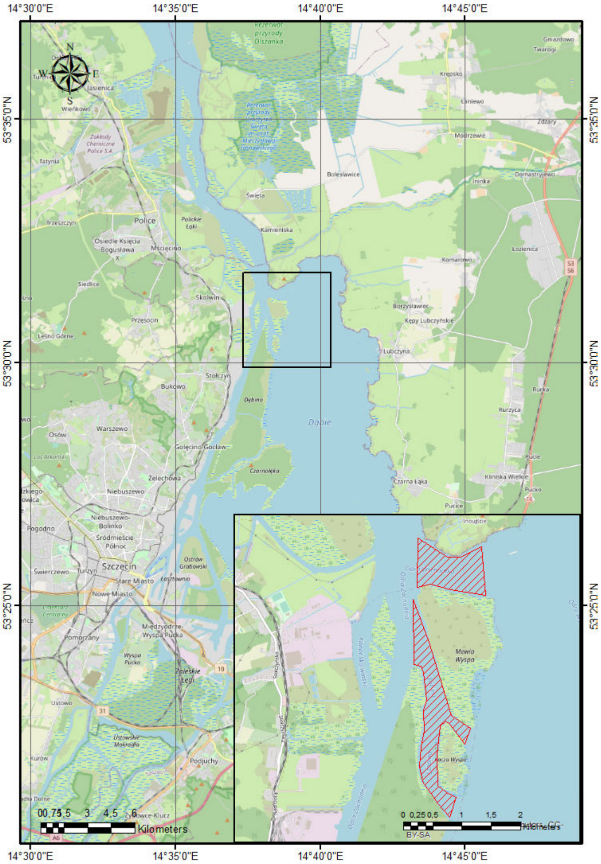


FIGURE 4. Map of Szczecin Water Node with an inset map of the surveyed region [ArcGis, Edge Tech and Open Street Map].

System was implemented on the Intel Core i7-8750H with 24GB RAM and NVIDIA GeForce GTX 1050 Ti. As the programming language, we used Python with the use of the TensorFlow library for neural networks. During the initial state of research, we analyzed the selection of network architecture. First, we focused on creating a new model that will be trained from the beginning. However, the learning efficiency was not very high. Hence, we eventually used the VGG-16 model.

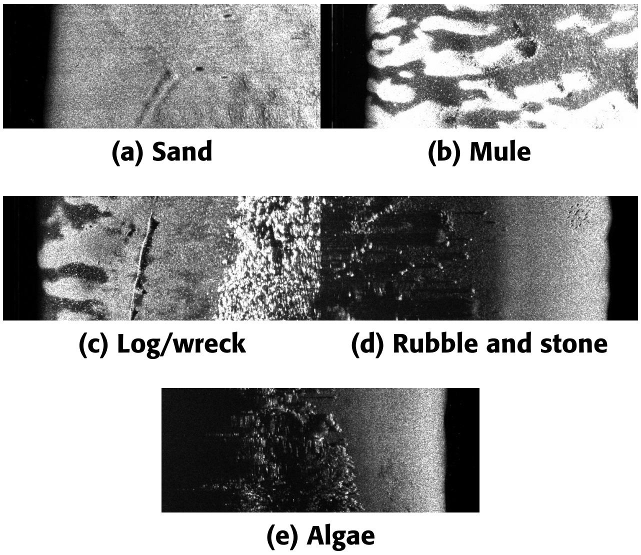

(a) Sand
(b) Mule
(c) Log/wreck
(d) Rubble and stone
(e) Algae

FIGURE 5. Representatives of object classes in sonar images.

A. DATA COLLECTING

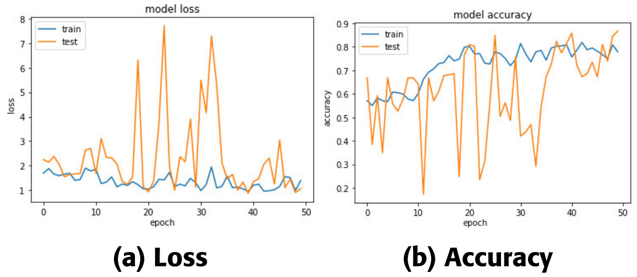
The survey area is located within the city of Szczecin (north-western Poland). The first part of the area is the Czapina and Babina Canal, which is a natural connection of Dabie Lake with the Oder River harbor basin. The basin is classified as inland waters. The second area is the anchorage on the Inski Nurt. The total area of the surveyed area is approximately 53000 m2 (Fig. 4).

Sonar data was recorded using Edgetech 4125 side scan sonar installed on the Hydrograf XXI survey vessel owned by the Maritime University of Szczecin. The sonar operated in outboard mode, mounted in a vertical axis together with GNSS/RTK satellite positioning receiver. The range of the sonar beam during the recording was set at 75m. The sonar acoustic frequency during recording was 600 kHz. Search profiles were conducted alongside the shores and guaranteed full bottom coverage. Data recording was performed in Discover software. Data were saved to native jsf files. In the data postprocessing stage, gain and TVG values were adjusted for signal correction. Selection of actual test and training data was performed in the Target Logger module of the Discover software during data post-processing in water flow mode.

B. PROCESSING SONAR DATA

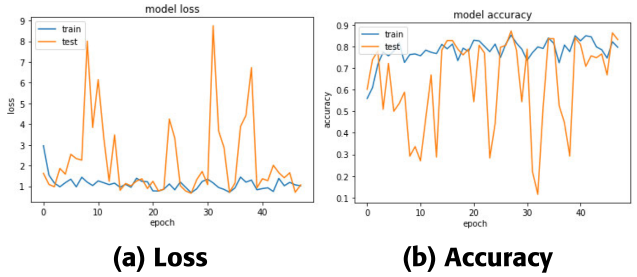
Sonar data were presented as tiff files that contained an image of the bottom and georeference data. A single sample was composed of a sonar image composed of a swim relative to a specified distance. Hence, such a sample was divided into smaller areas relative to the sonar measurement direction (i.e. along the OY axis). A constant cut area of 2 meters was assumed. The sample trimmed in this way was scaled to a network input that was equal to 256 × 256 × 3. Class labels were added manually.

C. CLASSIFICATION RESULTS



(a) Loss
(b) Accuracy

FIGURE 6. Training results for CNN and original images.



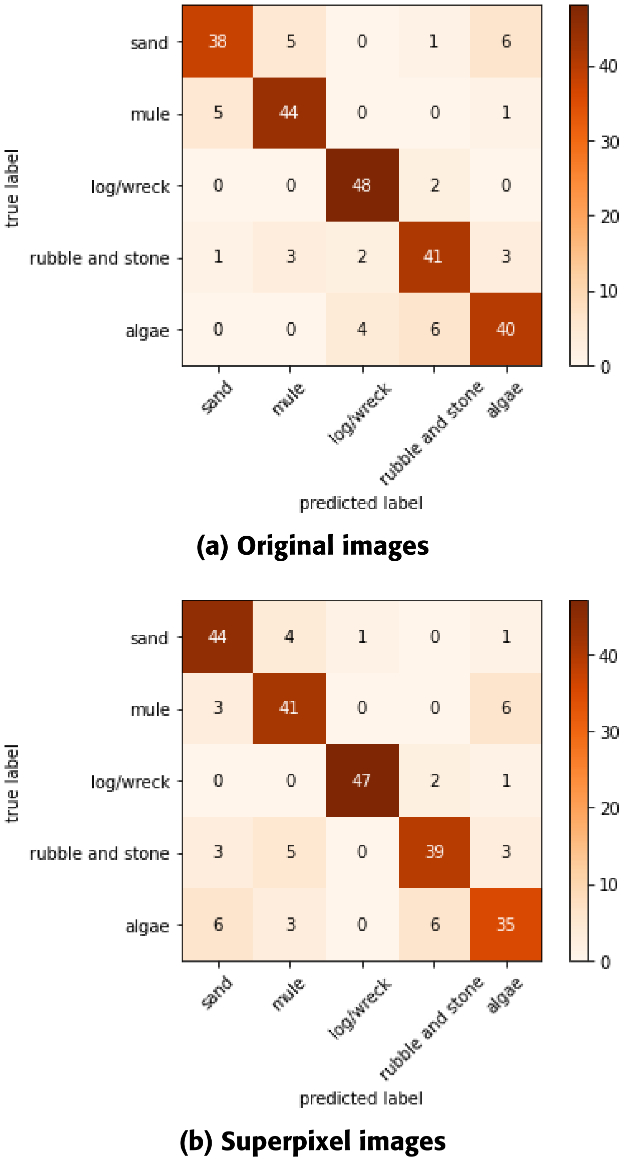
(a) Loss
(b) Accuracy

FIGURE 7. Training results for CNN and superpixel images.

All models were based on VGG-16 pre-trained model [16], which was extended to a bilinear pooling layer from two streams (like it was shown in Fig. 3). Both models were trained separately for 50 iterations of training by the use of the ADAM algorithm [17]. During training, we used mechanisms that interrupt training if there was a possibility of overtraining the model. The obtained data was augmented to increase the amount of training data. For this purpose, classic methods of data modification were used, based on random methods of image processing, such as rotation by any angle, magnification, stretching, and change of color saturation. It allows the creation of a large database for training purposes from 3000 to 3900 images (augmentation produced 30% more data). This data was split into two subsets: training and testing (80% from the whole images to training, and the remaining 20% were put in testing one). All samples were labeled into five classes: sand, mule, log/wreck, rubble/stone, and algae. In addition, 50 samples for each class were identified, which were not used in the training process, but only to verify the operation of the model. The training results are shown in Fig. 6 and 7. In Fig. 6, the loss and accuracy of the trained model with original images were shown. In the case of the chart of the loss value, the greatest jumps can be observed in the range of 20-35 iterations for test samples. In subsequent iterations, they still occur, but not so drastic. Such jumps are caused by taking the algorithm samples for which the value of the loss function was high, so the fit of the model to the given samples was much worse. However, for the loss for the training set, there are no such drastic jumps. It is worth taking a look at the matter of the accuracy of the trained model. There are small jumps for the training set, but the effect increases as the number of iterations increases. In contrast, training the same classifier with identical parameters, but for a set of images after the superpixel operation, achieved higher accuracy results faster (see Fig. 7). The reason is that the samples were simplified, so the operation of the convolution and pooling layers quite often did not contribute to huge

TABLE 1. Metric values determined for the validation database and trained classifiers.

	Original images			Superpixel images		
	Precision	Recall	F1 score	Precision	Recall	F1 score
sand	0.86	0.76	0.81	0.79	0.88	0.83
mule	0.85	0.88	0.86	0.77	0.82	0.80
log wreck	0.89	0.96	0.92	0.98	0.94	0.96
rubble and stone	0.82	0.82	0.82	0.83	0.78	0.80
algae	0.80	0.80	0.80	0.76	0.70	0.73



(a) Original images

(b) Superpixel images

FIGURE 8. Confusion matrices for testing with 250 samples (50 for each class).

changes in the extracted features. However, there is quite a significant difference in the increments for the test set for the loss function and the accuracy.

For better analysis of the trained models, we prepared a testing set containing 250 images (50 for each class) that were not used in the training process. All of them were used in the verification of trained models. In the case of a model trained with original images, the results were shown in Fig. 8a. The accuracy was reached at 84,4%. The worst results of the classification were achieved in the case of the sand class, where the classifier quite often found features that indicated other classes - mule or algae. A similar test was performed on these 250 images (with superpixels operation) and the results reached 82,4% (see Fig. 8b). In the case of the simplified images, the classification of the samples marked as sand was higher but much worse for the class marked as algae. It is worth noting that the accuracy of such simplified images is very high. In many cases, even a specific pixel arrangement can be a key to indicating a specific class by a neural network. Hence, the obtained results indicate that simplifying the image (by superpixel technique with such a large number of clusters and resizing) allows for the classification of such classes.

For a more accurate analysis of the classifiers, the metric values have been determined and presented in Tab. 1 (precision is calculated as $TP/(TP+FP)$, recall as $(TP/(TP+FN)$, $F1$ score is a harmonic mean of precision and recall, where TP is true positive, FP false positive, FN false negative). With the exception of the two classes representing log/wrecks and sand, the classification of original images returns much better results. Based on this, it can be seen that the selected classes are differentiated by features, not shape. The use of the idea of superpixels causes distortion of the image while preserving, above all, the shape. Hence, classes such as sand obtained better classification metrics for superpixels as the classifier did not focus on feature extraction. However, it can be seen that most of the classes were better classified compared to the original images, which allowed for a high average precision value of 0.844. Unfortunately, these values indicate that the classification analysis of algae is the lowest. It is caused by the problem of the similarity with rubble and stone. Despite these problems, the F1-score value allows for a real assessment of the classifier itself, which is the harmonic mean of the other two measures. The ideal value is 1, which means that the classifier has ideal precision and recall values. In the case of original images, the average value reached 0.842. Again for superpixels, it was 0.824. In both cases, these are high values, which indicate good adaptation to the selected classes.

In order to better assess our solution, we trained the earlier model to the datato data used in [18]. The used in those experiments was collected by USGS and Northern Arizona University field technicians, river guides, and volunteers from a fish monitoring site that spans a 1.6-km canyonbound reach of the Colorado River, located 98-km downstream of Lees Ferry in Marble Canyon, Arizona, directly upstream from the confluence of the Little Colorado River, and covers multiple pool-riffle sequences. Data were collected during five river trips between May 2012 and April 2015. Our model was trained using the fore-mentioned data to classify three, same

TABLE 2. Accuracy comparison of the proposed model with state of art [%].

	Our Model	LSQ Model [18]	GMM-4 Model [18]
Sand	84.2	72.6	60.7
Gravel	69.7	27.7	49.0
Boulders	83.1	75.2	79.8

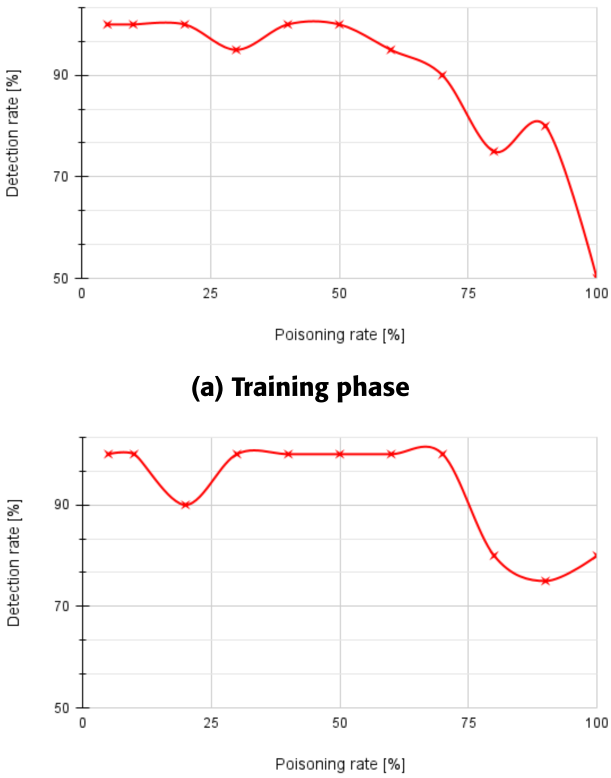
classes chosen in the referred paper [18]. These classes were sand, gravel and boulders. The obtained results are displayed in Tab. 2. Our solution was compared with two different models, the least-squares model (LSQ) and the 4-substrate Gaussian Mixture Model (GMM-4). A big difference in accuracy can be seen especially in the second class, where the accuracy was higher than 20%. It is worth noting that the model we used was already trained and only modified on the last layer to a smaller number of classes. The results also indicate the potential for learning transfer, which allowed to perform an additional training process, but using the previous weights.

As can be seen in Tab. 2, our model performed better with the classification task, achieving higher accuracy than refereed models among all classes. This shows that using sophisticated computer vision for SSS image recognition, like our model, is a valid strategy and should be further explored in the literature.

D. POISONING DETECTION

The next step of our experiments was the analysis of poisoning detection to verify the security of the system. We conducted the tests in two respects: random poisoning of the sets by $\langle 5, 10, 20, \ldots, 100\rangle[\%]$ during the training phase, and in the second test by poisoning samples during the verification phase – when the classifier was ready for implementation. The simulation of a poisoning attack consisted in randomly selecting one of two operations: replacing labels, or randomly modifying pixels on selected samples. The selection of samples was random from a given set of images (50 images for each class in validation case) with rounding up while maintaining uniform poisoning for each class.

The poisoning of the entire dataset consisted of the random selection of samples in the verification base and the use of the mechanism presented in Fig. 1. After every 10 iterations, the model was evaluated through the results from both networks. The results of such measurements are presented in Fig. 9a. The detection efficiency for poisoning half of the dataset was above 90%, which means that it was detected in almost all cases. However, after poisoning more samples, the proposed solution had much lower results. Nextly, we evaluate the detection of attacks on the verification phase itself which results are shown in Fig. 9b. Here, the detection has a good rate even when 75% of sets are poisoned. In the case of poisoning almost the whole dataset, the detection rate decreases. The proposed solution is based on two neural networks indicating a high potential for poisoning detection. It is worth noting that the proposed method requires more computing power because a second identical network is created and trained to classify processed samples. However, despite additional calculations, the use of artificial intelligence methods becomes much safer. Especially when the training process itself goes safely and the trained model is implemented in the production system. It is a solution that is based on a quick evaluation model as it uses the classic neural network classification, but also the classification with simplified samples. This allows for removing modified pixel alignments. At the same time increasing the security of systems based on artificial neural networks.



(a) Training phase

(b) Validation phase

FIGURE 9. The dependence of the poisoning of the collection on its detection.

IV. CONCLUSION

In this paper, a new system of processing and analyzing side-scan images was presented. The proposal was based on using identical bilinear convolutional neural networks, where the first one will process a simplified image (by the apply superpixel technique) and the second one the original one. As a result, two probability vectors (for one image) are returned. This vector is used for comparing purposes. In the case when two probability from both samples indicates different classes, it means that the original image can be poisoned. When both networks returned similar results, there is likely no attack on the data. Our proposition proposes an alternative approach to the use of deep models of neural networks in practical terms, which allows for high accuracy of the network as well as its safety. This proposition was evaluated on real data gathered in selected areas in Poland. Of course, the data has been additionally augmented to increase their amount for the neural network training process. However, the performed tests indicate the high accuracy of the classifiers, where over 80% accuracy was achieved (in the case of both networks). Moreover, the average sample poisoning detection rate based on all simulations performed reached 91%. However, it should be noted that the highest detection rate was achieved with poisoning less than 50% of the dataset. Moreover, the proposed technique increases the number of additional operations: an additional training process is performed and the images are processed beforehand. In future work, we want to focus on decreasing the number of additional calculations in the proposed system.

REFERENCES

[1] Y. Yu, J. Zhao, Q. Gong, C. Huang, G. Zheng, and J. Ma, ''Real-time underwater maritime object detection in side-scan sonar images based on transformer-YOLOv5,'' *Remote Sens.*, vol. 13, no. 18, p. 3555, Sep. 2021.

[2] X. Qin, X. Luo, Z. Wu, and J. Shang, ''Optimizing the sediment classification of small side-scan sonar images based on deep learning,'' *IEEE Access*, vol. 9, pp. 29416–29428, 2021.

[3] Y. Tang, H. Wang, Y. Xiao, W. Gao, and Z. Wang, ''Feature extraction for side scan sonar image based on deep learning,'' in *Proc. 40th Chin. Control Conf. (CCC)*, Jul. 2021, pp. 8416–8421.

[4] L. Dong, ''Shipwreck identification with side scan sonar image based on fractal texture,'' *Mar. Geol. Quaternary Geol.*, vol. 41, no. 4, pp. 232–239, 2021.

[5] C. Sun, L. Wang, N. Wang, and S. Jin, ''Image recognition technology in texture identification of marine sediment sonar image,'' *Complexity*, vol. 2021, pp. 1–8, Mar. 2021.

[6] Y. Li, M. Wu, J. Guo, and Y. Huang, ''A strategy of subsea pipeline identification with sidescan sonar based on YOLOV5 model,'' in *Proc. 21st Int. Conf. Control, Autom. Syst. (ICCAS)*, 2021, pp. 500–505.

[7] Z. Cheng, G. Huo, and H. Li, ''A multi-domain collaborative transfer learning method with multi-scale repeated attention mechanism for underwater side-scan sonar image classification,'' *Remote Sens.*, vol. 14, no. 2, p. 355, Jan. 2022.

[8] F. Langner, C. Knauer, W. Jans, and A. Ebert, ''Side scan sonar image resolution and automatic object detection, classification and identification,'' in *Proc. OCEANS EUROPE*, May 2009, pp. 1–8.

[9] X. Qin, Z. Wu, X. Luo, B. Li, D. Zhao, J. Zhou, M. Wang, H. Wan, and X. Chen, ''Temporal fusion based 1-D sequence semantic segmentation model for automatic precision side scan sonar bottom tracking,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4201816.

[10] I. M. Ahmed and M. Y. Kashmoola, ''Threats on machine learning technique by data poisoning attack: A survey,'' in *Proc. Int. Conf. Adv. Cyber Secur.* Singapore: Springer, 2021, pp. 586–600.

[11] J. W. Stokes, P. England, and K. Kane, ''Preventing machine learning poisoning attacks using authentication and provenance,'' in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2021, pp. 181–188.

[12] T. Chiba, Y. Sei, Y. Tahara, and A. Ohsuga, ''A countermeasure method using poisonous data against poisoning attacks on IoT machine learning,'' *Int. J. Semantic Comput.*, vol. 15, no. 2, pp. 215–240, Jun. 2021.

[13] D. Polap, N. Wawrzyniak, and M. Wlodarczyk-Sielicka, ''Side-scan sonar analysis using ROI analysis and deep neural networks,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4206108.

[14] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, ''Slic superpixels,'' EPFL Tech. Rep. 149300, 2010.

[15] J. Yin, T. Wang, Y. Du, X. Liu, L. Zhou, and J. Yang, ''SLIC superpixel segmentation for polarimetric SAR images,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5201317.

[16] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, pp. 1–9.

[17] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' in *Proc. ICLR*, 2015, pp. 1–15.

[18] D. Hamill, ''Quantifying riverbed sediment using recreational-grade side scan sonar,'' Ph.D. dissertation, ProQuest Dissertations Publishing, Utah State Univ., Logan, UT, USA, 2017, Art. no. 10623435.

DAWID POŁAP (Member, IEEE) received the M.Sc. degree (Hons.) in applied mathematics from the Silesian University of Technology, Gliwice, Poland, in 2017, and the Ph.D. degree from the Czestochowa University of Technology, Czestochowa, Poland, in 2019. He is currently an Adjunct with the Faculty of Applied Mathematics, Silesian University of Technology. He has authored or coauthored more than 100 research papers in international conferences and journals. His main research interests include image processing, intelligence computing, and various aspects of machine learning. He received the ''Diamond Grant'' for the most talented students and a scholarship for exceptional young scientists from the Polish Ministry of Science and Higher Education. He has served as the Editor for *Applied Soft Computing*, *Expert Systems With Applications*, *Sensors*, and *Evolutionary Intelligence*.

ANTONI JASZCZ is currently pursuing the bachelor's diploma degree in computational intelligence and machine learning with the Faculty of Applied Mathematics, Silesian University of Technology, Gliwice, Poland. He is a Participant in a mentoring program for highly talented students from the Rector of the Silesian University of Technology. His current research interests include various aspects of computational intelligence, machine learning, and image processing.

NATALIA WAWRZYNIAK received the M.Sc. degree in computer science and engineering from the Szczecin University of Technology, in 2006, and the Ph.D. degree in computer science, image processing from the West Pomeranian University of Technology, Szczecin, Poland, in 2013. Since 2013, she has been an Associate Professor with the Maritime University of Szczecin, Poland. Her main research interests include spatial data processing, underwater remote sensing, and system design for various inland and marine applications.

GRZEGORZ ZANIEWICZ received the M.Sc. degree in navigation from the Maritime University of Szczecin, Poland, in 2009. Since 2011, he has been a Research Assistant with the Maritime University of Szczecin. Since 2009, he took a part in many research and development projects related to geoinformatics and hydrography. His main research interests include hydrography, sonar target tracking, and underwater data acquisition and processing.