Postprint of: Czaplewski B., Dzwonkowski M., A novel approach exploiting properties of convolutional neural networks for vessel movement anomaly detection and classification, ISA TRANSACTIONS (2021), pp. 1-18, DOI: 10.1016/j.isatra.2021.02.030

A Novel Approach Exploiting Properties Of Convolutional Neural Networks For Vessel Movement Anomaly Detection And Classification

Bartosz Czaplewski[1] and Mariusz Dzwonkowski[2]

[1,2]Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Department of Teleinformation Networks, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland
[2]Medical University of Gdańsk, Faculty of Health Sciences, Department of Radiological Informatics and Statistics, Tuwima 15, 80-210 Gdańsk, Poland

[1]bartosz.czaplewski@eti.pg.edu.pl (orcid: 0000-0001-7904-5567, corresponding author)
[2]mard@gumed.edu.pl (orcid: 0000-0003-3580-7448)

Abstract: The article concerns the automation of vessel movement anomaly detection for maritime and coastal traffic safety services. Deep Learning techniques, specifically Convolutional Neural Networks (CNNs), were used to solve this problem. Three variants of the datasets, containing samples of vessel traffic routes in relation to the prohibited area in the form of a grayscale image, were generated. 1458 convolutional neural networks with different structures were trained to find the best structure to classify anomalies. The influence of various parameters of network structures on the overall accuracy of classification was examined. For the best networks, class prediction rates were examined. Activations of selected convolutional layers were studied and visualized to present how the network works in a friendly and understandable way. The best convolutional neural network for detecting vessel movement anomalies has been proposed. The proposed CNN is compared with multiple baseline algorithms trained on the same dataset.

Keywords: convolutional neutral networks, deep learning, anomaly detection, vessel movement anomalies, radar datasets

1. Introduction

Monitoring of the vessel traffic at sea and near the maritime borders is a crucial task of many uniformed services, such as the Border Guard or the Maritime Office. These organizations have to be equipped with technology enabling communication, acquisition, exchange and visualization of data in various operational situations, as well as the classification of vessel behaviors. The amount of data to be analyzed is increasingly overwhelming for human operators. It is common for a single operator to observe multiple screens at the same time. In such conditions, it is very easy to make a mistake, which can be very detrimental. Therefore, it is important to develop techniques to automate some elements of the operator's work. It is necessary to provide automatic processing to synthesize the behaviors of vessels to relieve the operator and present him with ready results of analyses in an understandable and practical way. One such element of the operator's work is the detection and classification of so-called vessel movement anomalies.

A vessel movement anomaly may take various forms. It may be a situation in which a ship significantly deviates from the set course or a situation in which two ships are getting close enough for smuggling. It may also be a situation where the ship is moving in the wrong direction or is entering a restricted area. This article discusses vessel route anomalies relevant to the Maritime Regional Unit of the Polish Border Guard. As a result of the dialogue with the Border Guard, a list of vessel anomalies of interest for to the employees of the border guard was determined. Although the list of the defined anomalies was very long, the following anomalies were selected for this stage of the research: the vessel is far away from the area, the vessel is near the area and it will not enter the area, the vessel is near the area and it will enter the area, the vessel is inside the area. The area can be understood as a fairway or a restricted area.

An important limitation to consider when designing methods for detecting or classifying anomalies is that the data from the Automatic Identification System (AIS) is not trustworthy. In case of malicious event, such as smuggling contraband or military operations, enemy vessels would simply not transmit AIS signals. In addition, nowadays we are aware that AIS is very susceptible to AIS spoofing and AIS poisoning attacks [1]. These attacks are designed to fabricate false AIS signals that provide a false location of a vessel, e.g. to cover up the fact of hijacking a ship. These attacks can lead to misinformation or diversion. For the above reasons, the Border Guard is not interested in detecting anomalies based on AIS data. On this basis, there is a critical need for new anomaly detection methods, which process the data obtained from radars.

This article presents an innovative solution to the problem of classification of vessel movement anomalies using Convolutional Neural Networks (CNNs). To the best of the authors' knowledge, this is the first work on this subject using Deep Learning techniques. The motivation to use this technique is the remarkable achievements of convolutional neural networks in solving problems of image classification. The goal of the paper is to use the object detection approach for images to classify vessel movement anomalies obtained from radar data.

The novelty and contribution of this paper is listed below.

- An innovative approach of using a convolutional neural network for the purpose of vessel movement anomaly detection and classification.
- New synthetic radar datasets for the proposed CNN training. The generated datasets (originally saved as data vectors) were visualized as images.
- In-depth study leading to the selection of the best CNN structure (from a pool of 1458 trained variants) for the considered classification of vessel movement anomalies.
- Detection of anomalies and their further classification into groups of anomalies (defined by the cooperating the Maritime Unit of the Polish Border Guard) in order to study the decision accuracy of the proposed CNN and cover different scenarios of practical application.
- Investigation of the impact of changes in various network structure hyperparameters and learning hyperparameters on the overall classification accuracy.
- Comparison of the proposed method with other machine learning methods using the same training datasets.

- Use of heatmaps of layer activations at various depths in the CNN to explain and visualize how the proposed networks work for better interpretability.

The structure of this article can be summarized as follows. Related work is described in Section 2. Section 3 presents the research methodology. The generation of new datasets is described in Section 4. Section 5 describes the tested network structures and the learning parameters used. An analysis of the overall accuracy as well as specific class prediction rates of the trained networks is presented in Section 6. Section 7 describes the structure of the proposed network. A comparison of the proposed CNN and other machine learning algorithms trained on the same dataset is presented in Section 8. An analysis of activations of selected layers and a discussion on interpretability are included in Section 9. Conclusions are included in Section 10.

2. Related Work

Previous work on the detection of vessel movement anomalies can mostly be divided into four groups: methods based on Support Vector Machines (SVM), methods based on Neural Networks (classic NN, i.e. shallow and fully connected), methods based on Gaussian Mixture Models (GMM), and methods based on Bayesian Networks (BN).

An example of SVM-based methods is presented in [2]. Li et al. proposed a SVM-based method that extracts high abstract movement features, such as turning left or looping, from the vessel's path and then clusters these features into so-called motives. The authors showed that the SVM trained on these high abstract features can classify better than a SVM trained on time series of geographic locations. However, SVM-based methods suffer from limitation to binary classes, impeded communication of the learned models, inability of partial assignment and significant computational complexity.

Examples of NN-based methods can be found in [3–5]. NNs can be used to detect anomalies in a manner that a vector containing a series of geographical coordinates is introduced as the input, and the network maps it to the appropriate output class. Rhodes et al. proposed a method of clustering normal vessel speeds using a fuzzy neural network. A vessel speed that was not considered normal by the network was considered an anomaly. Rhodes et al. also proposed a method for anomaly detection based on a NN by predicting the future movement of the ship. However, classification models taught by classical neural networks are difficult to interpret by a human operator.

Examples of GMM-based methods can be found in [6–8]. The goal of a GMM is to present clustering of training data and its ability to spread across a multi-dimensional space. Kraiman et al. presented a method for clustering of vessel data in high-dimensional feature space using a GMM as a statistical model of normal data. In that method, learned GMM and Bayesian analysis is used for anomaly detection. Laxhammar et al. compared the anomaly detection performance of a GMM and a Kernel Density Estimator (KDE), which is a generalization of a GMM, on AIS data and stochastically generated anomalous paths. For each method, they measured the number of iterations required to correctly classify the path as an anomaly. Laxhammar et al. also proposed an unsupervised data clustering of normal vessel traffic patterns, described in terms of instantaneous course, speed and position of vessels. The GMM is applied as the cluster model with the clustering algorithm being the greedy version

of the Expectation-Maximization method. The presented results showed that the most prominent anomalies are vessels crossing sea lanes, vessels moving in close proximity to sea lanes and vessels moving in the opposite direction of sea lanes. Unfortunately, the GMM's anomaly threshold highly influences the system usability for anomaly detection. Thus, to increase the efficiency of the system, the operator should be able to dynamically set the appropriate threshold value during system runtime. A GMM and KDE are also used in the core of the Traffic Route Extraction and Anomaly Detection (TREAD) method for vessel movement anomaly detection [9].

Examples of BN-based methods can be found in [10,11]. Mascaroa et al. developed an anomaly detection method using Bayesian Networks trained on real-world AIS data and complementary data, obtaining both static and dynamic models of Bayesian Network. The authors also showed that a dynamic and static model can be combined to improve the overall anomaly detection performance. Johansson and Falkman proposed a different method for detection of anomalous vessel behavior based on a BN. This method was tested on synthetic data and can successfully be utilized to detect single parameter anomalies, for instance, vessel speeding. The advantages of a BN are that these networks are easy to examine and verify, it is easy to introduce expert knowledge to the model, and they are easily interpreted by a human. However, according to Johansson and Falkman, detecting anomalies with a BN is not an ideal solution. The method does not detect all of the desired anomalies, unless the detection threshold is raised, causing a vast amount of false positives.

There are papers [12,13] in which deep learning techniques were used to solve such problems as recognition of ship outlines in images, vessel type classification or post processing of radar data for object detection. However, none of these works address the problem of vessel movement anomalies.

To the best of the authors' knowledge this paper is the first work on vessel movement anomaly detection using deep learning techniques, in particular, convolutional neural networks (CNNs). It is difficult or even impossible to compare our work with methods that detect vessel movement anomalies based on, e.g. type, category, speed or any other vessel parameters obtained from the AIS system due to the different interpretation of anomalies for vessel movement as well as the very different input data. Moreover, the AIS system is viewed by the Maritime Unit of Border Guard as an unreliable source of information in the event of anomaly detection. The use of deep learning techniques on data obtained from radars allowed us not only to successfully detect anomalies, but also to further classify them into anomaly groups.

In the past, deep learning techniques have often been discarded in these applications due to difficulties of human interpretation of the learned model. Thanks to research like the Heatmapping project [14,15] this is no longer true. This article also attempts to explain how the network works through visualization of the network "thinking".

3. Proposed Methodology

Before conducting the research described in this article, a prototype of anomaly detection software, that worked in an analytical manner in accordance with the given rules and without using machine learning, had been programmed. Unfortunately, that solution did

not meet the authors' expectations for the following reasons. Firstly, that approach turned out to be immensely slow and inadequately scalable in terms of the computing resources for a large number of observed objects, which is the case with the Border Guard monitoring the entire coast. Secondly, such approach was a closed implementation and did not allow for an easy addition of new anomaly definitions without extra programming, without increasing the complexity of the system and without increasing the required computing resources. Therefore, a need arose to develop a new approach, which is presented in this article.

The purpose of the proposed approach is the detection and classification of vessel movement anomalies using convolutional neural networks. The reasons for the use of convolutional neural network are as follows.

- There is no need to develop any analytical model or feature selection. A CNN uses a convolution of the image and filters to learn features by itself and passes them to deeper layers until the network achieves an output which is invariant to geometrical distortions in input images.
- Once trained, a convolutional neural network can report a decision very quickly. The reply is almost instantaneous – within a few milliseconds on a simple PC.
- An existing CNN can always learn new anomaly classes. A scenario with four classes of anomalies was considered in this paper but in the future we intend to increase their number significantly, including more non-linear problems. A CNN is well suited for an increasing number of classes as it handles multi-class problems by producing probabilities for each class.
- A CNN classifier for multiple classes can be trained in one go.
- The aim of the study was to build a single network to detect multiple complex anomalies with the possibility to add new anomaly classes without changing the algorithm or the structure of the system.
- The architecture of a CNN inherently supports parallelization. NVidia CUDA technology can be used to significantly increase both the training and the decision time.
- A CNN analyzes the local spatial coherence of images. This feature not only reduces the number of operations needed to process an image (by operating on patches of meaningful pixels), but also adequately reflects the real life interpretation of the input data.
- It is easy to interpret a CNN's workflow with the use of heatmaps of layers' activations. This is a direct counterpart to an MRI scan of a human brain.

Based on the above-mentioned properties, one can see that convolutional neural networks are an ideal tool for the designated approach, which is discussed in detail below.

The input of the proposed method is a data vector containing data about the restricted zone (number of vertices and geographic coordinates of the vertices) and data about the vessel's movement (sequence of 10 consecutive vessel positions). The input data is processed and converted into an image. The output of the method is the decision whether a given vessel movement is normal (anomaly class 0) or is an anomaly, and if so, to classify this anomaly into one of three classes (anomaly classes 1–3).

The flowchart for the proposed method is shown in Figure 1. The vessel movement data coming from radars are stored in the surveillance system database. The restricted area is defined by the operator and its definition is stored in the surveillance system database. Before the data is fed from the database into the convolutional neural network, it is converted into an image. The convolutional neural network classifies the input data into one of the anomaly classes. Class 0 means normal vessel movement, i.e. no anomalies. The decision on anomaly detection and the decision on the anomaly class, in case of its detection, are transferred back to the surveillance system, to the presentation module. Classifier prediction scores can also be transferred to the presentation module, so that the operator of the surveillance system has full information about the certainty of the decisions.

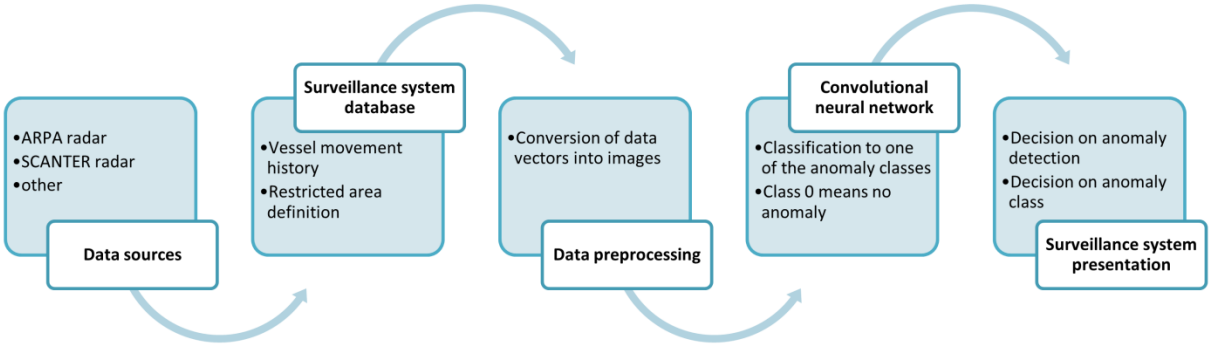

Fig.1. The proposed approach flowchart.

The research can be presented as a sequence of six detailed tasks: 1. Generation of new datasets containing anomalies of various classes. 2. Training of convolutional neural networks of various structures. 3. Searching for the best shape of the network based on the accuracy of the trained networks, as well as an analysis of rates of specific class predictions for the best networks. 4. Determining the structure of the proposed network for the considered problem and future research. 5. Comparison with multiple baseline algorithms using the same dataset. 6. Analysis of layer activations for explanation and interpretation of the learned model.

The first task was to write an application to generate random forbidden areas and random routes of vessels in relation to the area. The routes were drawn in such a way as to obtain the appropriate classes of vessel movement anomalies. Each instance of a vessel route was visualized in the form of a grayscale image. The idea was to use the achievements of image recognition through deep learning. Three variants of datasets were produced for three classification scenarios. Details are described in Section 4.

The second task was to use the previously created datasets to train 1458 networks of various structures. The trained networks differed in the number of convolutional layers, the number and size of filters in the convolutional layers, the number of fully connected layers, the types of functions in the activation layers, and the types of normalization layers. The goal of training such a large number of networks was to find a favorable network shape to solve the classification problem under consideration. Details are described in Section 5.

The third task was to evaluate the 1458 trained networks. The evaluation criterion was the resulting accuracy of the network. The impact of individual variable elements of the

network structure on their resulting classification accuracy was examined. Based on the obtained results, the best structures of networks for the three dataset variants were selected. Next, more detailed analysis of the accuracy of the networks was conducted. Rates for all possible class predictions were calculated for the selected best network structures. The greatest difficulty in the considered classification scenarios and the problems for visualization of the samples in the datasets were identified. The results are presented in Section 6.

The fourth task was to choose the structure of the proposed convolutional neural network for the considered problem. The intention is that this proposed network will be the starting point for future research using real-life data from radar devices. Details are described in Section 7.

The fifth task was to compare the proposed CNN with multiple baseline algorithms using the same dataset. The comparison includes accuracy, training time, decision time, and interpretability. The proposed CNN and the second best method are compared using ROC curves and confusion matrices. Details are described in Section 8.

The sixth task was to perform detailed tests of the activations of selected network layers were performed using heatmaps. Deep learning methods are often accused of being incomprehensible to humans. Hence, the purpose of this study was to explain and understand the performance of the trained networks. The reasoning of the best network for the most complex dataset variant was visualized. Heatmaps of layer activations at various depths in the network were presented. The results are presented in Section 9.

For the generation of new datasets, an application was written in the C# language in the .NET framework. As the programming language for training and validation of the convolutional neural networks and their further analyses the MATLAB 2018a with Deep Learning Toolbox was selected. The calculations were made utilizing the CUDA technology and a Nvidia GeForce GTX 1080 Ti graphics card.

4. New Datasets

For the purpose of the research, datasets for the training and validation of convolutional neural networks were generated. The data characteristics were determined from cooperation with the Maritime Regional Unit of the Polish Border Guard. An important assumption was that the data were to be geographic coordinates obtained only from radars. Data from the AIS system are of little interest for the border guard for many reasons. Firstly, persons of interest to the border guard will always turn off AIS transmitters during illegal activities, whether smuggling, hijacking, intelligence, etc. Secondly, cases of AIS spoofing are widely known, which completely undermine the credibility of data from the AIS system.

As a result of the cooperation, a list of vessel movement anomalies interesting for the operators of the border guard was determined. Although the list of defined anomalies was very long, only a group of anomalies related to the movement of the object relative to the designated area was selected at this stage of the research. The designated area may be, e.g. a fairway or a restricted area, depending on the circumstances. Hence, examples of anomalies in this group may be the following situations: the ship is too close to the area, the ship is in the area, the ship enters the area, the ship leaves the area, etc.

An application was written in the C# language in the .NET framework. The application generates random restricted areas and random routes of vessels in relation to the area. The aim of the research was to utilize the significant achievements of object detection in images through deep learning, thus, the data was to be presented as an image. In order to facilitate processing, the data was visualized in the form of one-channel grayscale PNG files.

Restricted areas were randomly generated closed figures with different numbers of vertices. Various appearances for the areas were considered, such as figures filled with solid gray with visible edges, figures filled with solid gray without edges, or empty figures with only visible edges. After preliminary testing, it was found that the best results were obtained for figures filled with solid gray with no visible edges, thus such were used for the remainder of the study.

In order to draw the path of the vessel, a series of geographical positions were randomly generated. The shape of the path was pseudo-random, but the generator was written in such a way as to obtain four types of shapes: straight path, arc path, zigzag path, combined path from two other types. To illustrate the variable speed of vessels, the distance between successive positions was randomized from the given range. Various appearances for the travelled routes were considered. Paths with a highlighted head, i.e. the youngest position on the path, or without a visible head, were considered. Furthermore, the path could have been a solid color or darkening in grayscale with distance. After preliminary testing, it was found that the best results were obtained for the path with a visible head and the color changing in grayscale from white for the most recent position to black for the oldest position.

Three variants of the anomaly classification were defined, which differ in the number of classes. Variant A covered a classification scenario in which there were only two classes of anomalies: “class 0–2” which means that the vessel is outside the area, and “class 3” which means that the vessel is inside the area. Variant B included a classification scenario in which there were three classes of anomalies: “class 0” – the vessel is far from the area; “class 1–2” – the vessel is near the area but still outside of it; “class 3” – the vessel is inside the area. While, the most complex variant, C, included a scenario in which there were four classes: “class 0” – the vessel is far away from the area, “class 1” – the vessel is near the area and will not enter the area; “class 2” – the vessel is near the area and will enter the area; “class 3” – the vessel is inside the area.

It should be noted that the datasets in variants A and B are only a prelude to our research. The reader should pay the greatest attention to the dataset in variant C as it is closest to our target anomaly list, which is described in Section 8.

The routes were drawn in such a way as to obtain the above-mentioned four classes of vessel movement anomalies. The threshold of proximity to the forbidden area was set to 0.05 degrees in the decimal degrees format. For each of the 4 classes of anomalies, 5 groups of areas with the number of vertices varied from 4 to 8 were generated. For each group of areas, 4 subgroups of paths with 4 previously mentioned types of shapes were generated. In each subgroup, 1000 paths of a given anomaly class were generated. In total, $4 \times 5 \times 4 \times 1000 = 80{,}000$ images were generated. The resulting images were sorted to get three variants of datasets for three classification scenarios. Examples of images for various anomaly classes in three variants of datasets are shown in Figures 2 to 4.

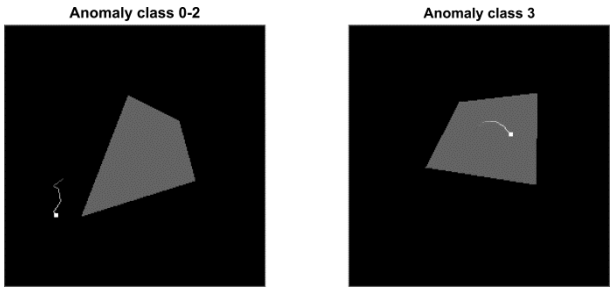

Fig.2. Examples of class 0–2 and class 3 anomalies in the variant A dataset.

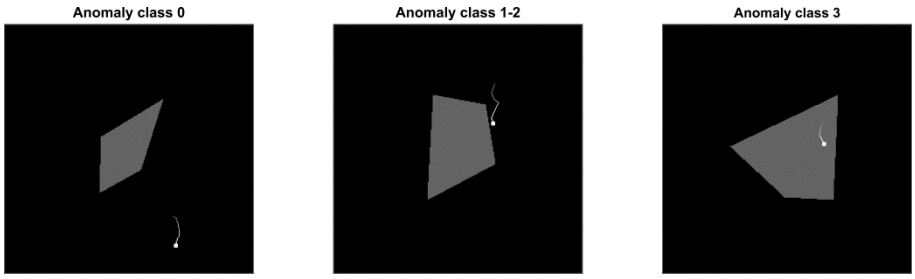

Fig.3. Examples of class 0, class 1–2 and class 3 anomalies in the variant B dataset.

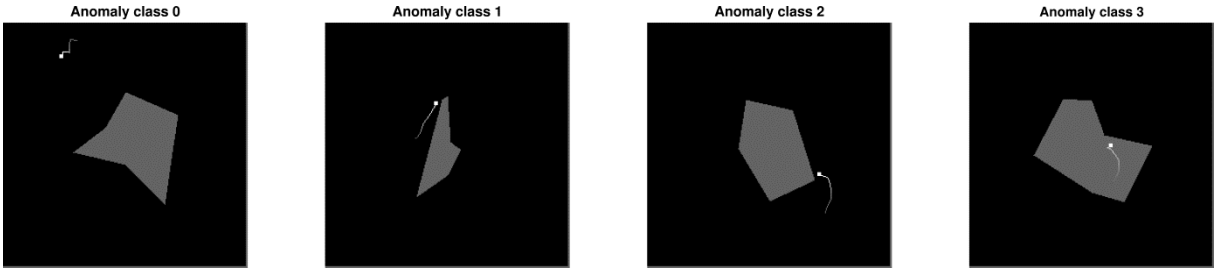

Fig.4. Examples of class 0, class 1, class 2 and class 3 anomalies in the variant C dataset.

5. Network Structures and Learning Parameters

All three variants of datasets described in Section 4 were used to train convolutional neural networks of various structures. The goal was to find a favorable network shape to solve the considered classification problem.

Firstly, an adequate architecture backbone and the considered hyperparameters were selected. Hyperparameters for CNN fall into two groups: network structure parameters and learning parameters. In the network structure parameters group, the following hyperparameters can be distinguished: number of hidden layers, types of layers, number of filters, filter size, activation functions, etc. In the learning parameters group the following hyperparameters can be distinguished: learning rate, momentum, number of epochs, batch size, etc.

Then a number of architectures were evaluated. We tuned hyperparameters and evaluated them empirically. The empirical approach, instead of the analytical one, is a known and recognized approach in innovative works using convolutional neural networks. Such approach is applicable if no one has previously presented a similar solution to the problem under consideration, which could be used as a reference point. If the work is pioneering in its application, there is no more direct approach.

Each network structure consists of a convolutional part, a fully connected part and an output classification layer. It was decided to investigate networks differing in the number of convolutional layers, the number and size of filters in the convolutional layers, the number of fully connected layers, the types of functions in the activation layers, and whether different normalization layers are present. All variable network structure parameters are described in detail in Table I. Since the images in the datasets are relatively simple, no additional pre-processing of the input images was applied. The pixel values of images sized 256×256 were directly fed into the input of the network. The structure of the networks in this study is shown in Figure 5.

Tab.I. Network structure parameters.

Structure parameter	Considered values
Input layer	
Size of input images	256×256 pixels
Convolutional part	
The number of convolutional layers	2, 3, or 4
The number of filters in the first convolution layer	16, 32 or 64
The number of filters in deeper convolution layers	8, 16, or 32
Filter size in all of the convolution layers	5×5, 7×7 or 9×9
Stride value in the first convolutional layer	2 both vertically and horizontally
Stride value in deeper convolutional layers	1 both vertically and horizontally
Zero padding in convolutional layers	to match the size of the output to the size of the input
Batch normalization layers	present or not present
Activation layers	ReLu function, clipped ReLu function, or leaky ReLU function with scale 0.01
Channel-wise local response normalization layers	present or not present; 9 channels per element
Fully connected part	
The number of fully connected layers	1 or 2
The number of neurons in each of the fully connected layers	100
Activation layer	ReLu function, clipped ReLu function, or leaky ReLU function with scale 0.01
Output layer	
The number of output neurons	2, 3 or 4; equal to the number of classes
Activation layer	Softmax function
Classification output	Cross-entropy loss function

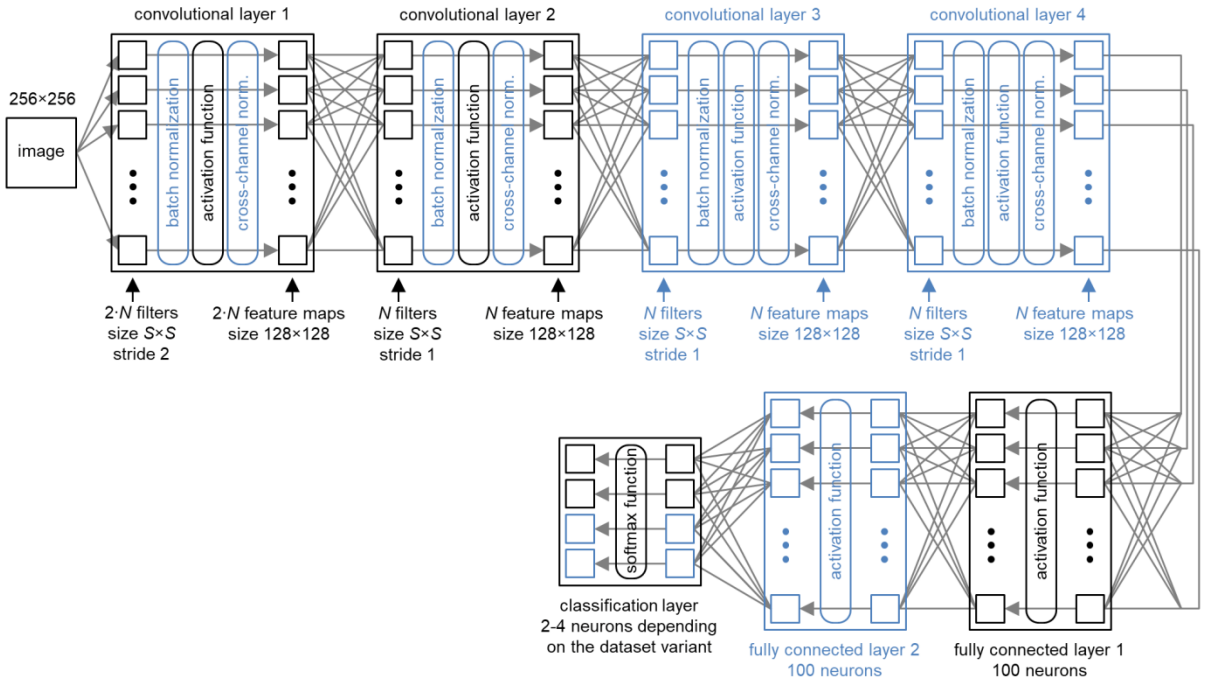

Fig.5. Variable structure of the CNNs in the research (blue elements are optional and not present in every structure, black elements are mandatory).

The CNNs were trained via supervised learning. The dataset was shuffled once for each of the CNNs. Then, 75% of the dataset was allocated to training, and the rest was allocated to validation. By shuffling the data, we reduce the variance in learning results and ensure that the learned model is general and not overfitted. In cases where data is sorted by its classes and types, as in this paper, shuffling the data is a common solution. The purpose of shuffling is to ensure that training set and validation set are representative of the overall distribution of the data. Moreover, when using the batch gradient descent algorithm, shuffling the data is even more important because it avoids the risk that the batch is not representative of the overall dataset. In theory, the data could be shuffled even after each epoch to avoid the risk of bad batches, but in these studies shuffling was limited to once per training. Data shuffling causes each data point in the training set to have an independent impact on the model without being biased by the data points before them.

All learning parameters are described in detail in Table II. The training settings were chosen empirically and the goal was to achieve precise training in a reasonable time of a few months for all of the combinations of network structures.

Tab.II. Learning parameters.

Learning parameter	Value
solving algorithm	SGDM (stochastic gradient descent with momentum optimizer)
momentum	0.9
initial learn rate	0.01
learn rate schedule	none
learn rate drop factor	0.1

learn rate drop period	10
L2 regularization	0.0001
max epochs	200
mini-batch size	128
validation frequency	234
validation patience	5
verbose	1
verbose frequency	twice per epoch
weight learn rate factor in convolutional layers	0.001
bias learn rate factor in convolutional layers	0.002

Taking into consideration all of the possible variable parameters of the network structures, the total number of trained networks was 1458. The uninterrupted training and validation of the above 1458 networks of different structures lasted about 3 months. Checking a wider range of parameters can only benefit the final results. As a result, we were able to show in the paper the influence of individual parameters of the network structure on the final classification efficiency in Section 6.

Structure parameters for the best networks for the three training dataset variants are presented in Section 6. Furthermore, structure parameters for the final proposed method are shown in Section 7, moreover, they are presented in the syntax of the Matlab's Deep Learning Toolbox, to facilitate the reproduction of our experiment.

6. Classification Accuracy

Each variant of the network structure, shown in Figure 5, was tested to present how each of them works. From the changes in the architecture, we can obtain changes in: classification accuracy, training time, decision time and the number of learnable parameters. However, in real use, only classification accuracy and decision time are of great importance. As the differences in decision time for different network architectures are negligible, classification accuracy was selected as the main criterion. Ultimately, high classification accuracy is a prerequisite for the actual implementation of the proposed method. This section shows how changes to the network architecture affect the overall accuracy. The impact of architectural changes on the other characteristics mentioned above was omitted, as this would unnecessarily lengthen the article and would be irrelevant.

The overall accuracy of the convolutional neural networks was defined according to the following formula:

$$a = \frac{\sum_{k=1}^{C} v_{k,k}}{|V|} \tag{1}$$

where a is the overall accuracy of the CNN, C is the number of classes, $v_{k,k}$ is the number of validation data samples of the k-th class which were predicted as the k-th class by the CNN, and $|V|$ is the cardinality of the validation dataset.

Figure 6 shows the overall accuracy of CNNs with different structures for three variants of the datasets used for training and validation. It can be seen that for each dataset

variant, some of the studied network structures did not converge at all and the accuracy remained at 0.5 for 2 classes, 0.33 for 3 classes, and 0.25 for 4 classes. Apart from these cases, the networks quickly acquired the ability to classify with varying levels of effectiveness. The best overall accuracy was 0.9821 in the case of the dataset variant of 2 classes, 0.9523 in the case of 3 classes, and 0.8546 in the case of 4 classes. The network structure parameters that achieved these best results are detailed in Table III. These CNN structures were studied further at the end of this section.

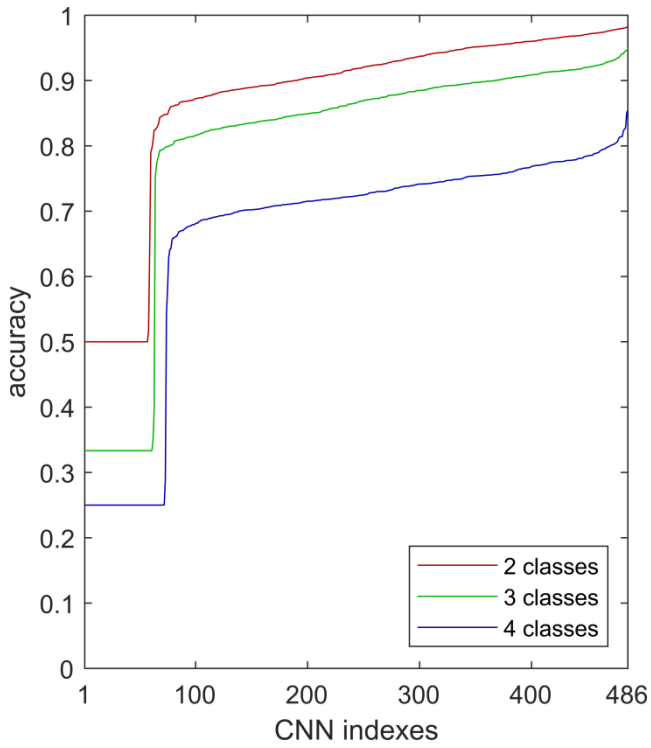

Fig.6. Overall accuracy of CNNs for different variants of datasets.

Tab.III. Best network structure parameters for different dataset variants.

Structure parameter	2 classes	3 classes	4 classes
Input layer			
Size of input images	256×256 pixels		
Convolutional part			
The number of convolutional layers	4	4	4
The number of filters in the first convolution layer	32	32	64
The number of filters in deeper convolution layers	16	16	32
Filter size in all of the convolution layers	9×9	9×9	9×9
Stride value in the first convolutional layer	2 both vertically and horizontally		
Stride value in deeper convolutional layers	1 both vertically and horizontally		
Zero padding in convolutional layers	to match the size of the output to the size of the input		

Batch normalization layers	not present	not present	not present
Activation layers	clipped ReLU	leaky ReLU	leaky ReLU
Cross-channel normalization layers	present	present	present
Fully connected part			
The number of fully connected layers	1	1	1
The number of neurons in each of the fully connected layers	100	100	100
Activation layers	clipped ReLU	leaky ReLU	leaky ReLU
Output layer			
The number of output neurons	2	3	4
Activation layer	Softmax	Softmax	Softmax
Classification output	Cross-entropy loss function	Cross-entropy loss function	Cross-entropy loss function
Overall accuracy	0.9821	0.9523	0.8546

Figure 7 shows the overall accuracy of the CNNs for different numbers of convolutional layers. For each dataset, it can be concluded that networks with 4 convolutional layers resulted in the best results, but also the largest number of networks that did not converge.

Figure 8 shows the overall accuracy of the CNNs for different filter sizes in the convolutional layers. For each dataset, it was observed that the larger the size of the filters in the convolutional layers, the greater the accuracy of the network.

Figure 9 shows the overall accuracy of the CNNs for different numbers of filters in the convolutional layers. It can be seen that the number of filters had a negligible impact on the overall accuracy, although the smallest number of networks that did not converge were obtained for the largest number of filters.

Figure 10 shows the overall accuracy of the CNNs for different activation functions. It was observed that the type of activation function used had no effect on the overall accuracy of the network. All three tested activation functions produced very similar results.

Figure 11 shows overall accuracy of the CNNs for different normalizations present in the networks structures. The combination of batch normalization turned on with cross-channel normalization turned on or turned off resulted in almost the same results. However, very interesting results were obtained for the combination of batch normalization turned off with cross-channel normalization turned on. In this case, there was an extremely large number of networks that did not converge, which caused this combination to be initially rejected during the initial study. However, the continuation of the research showed that this normalization is able to give slightly better results than others, but at the expense of long learning time.

Figure 12 shows the overall accuracy of CNNs for different numbers of fully connected layers. For each dataset, it was observed that a single fully connected layer gives the best results. Additionally, a single layer means fewer network parameters to learn.

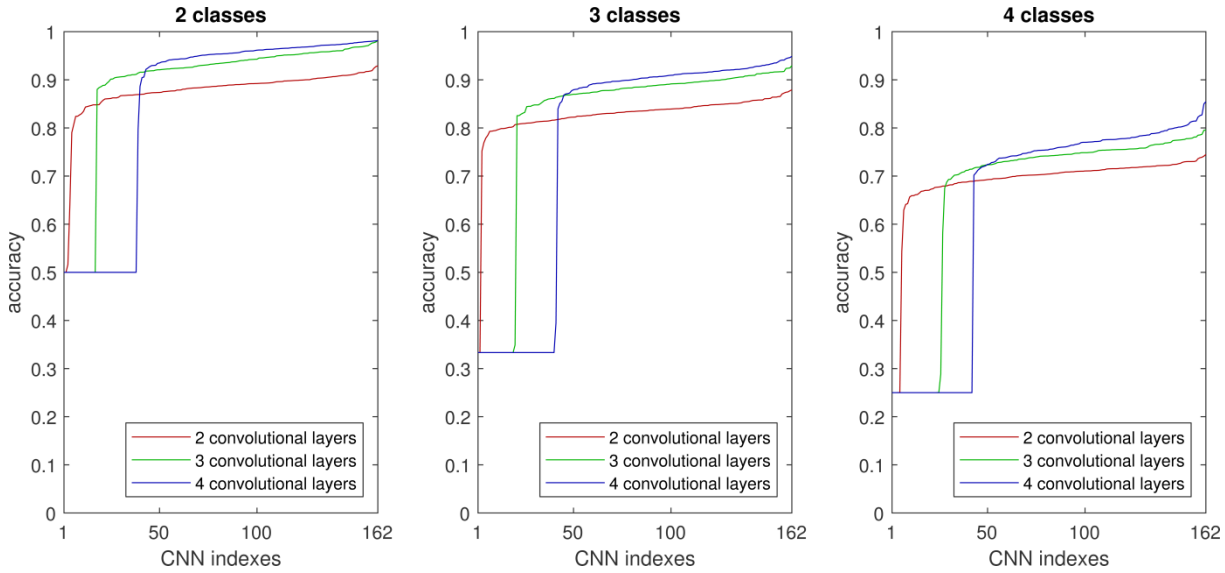

Fig.7. Overall accuracy of CNNs for different numbers of convolutional layers.

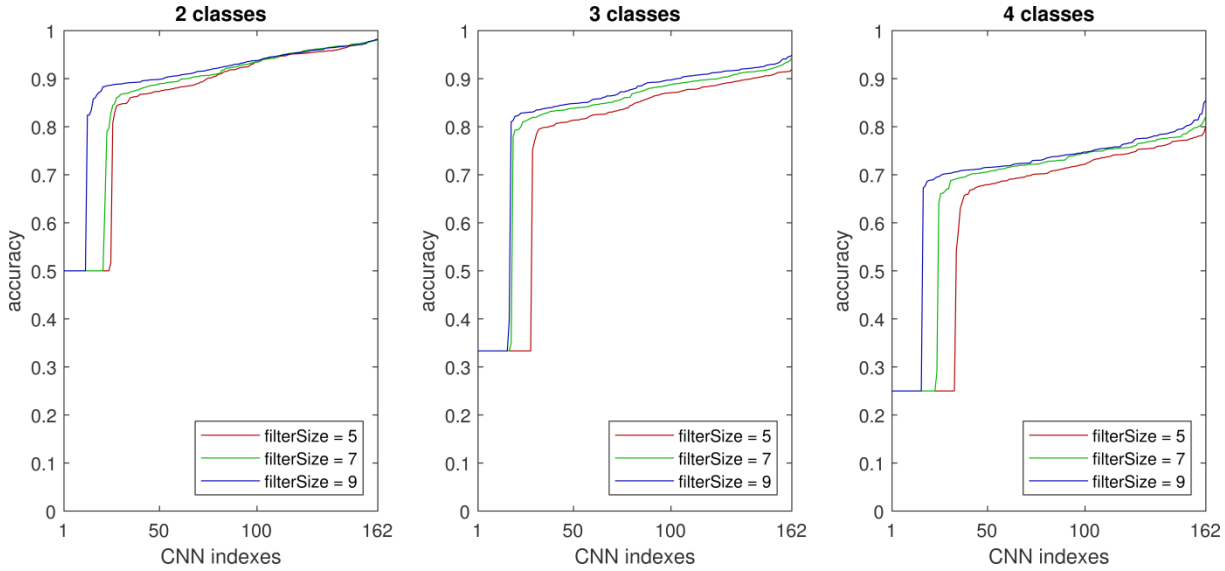

Fig.8. Overall accuracy of CNNs for different filter sizes in convolutional layers.

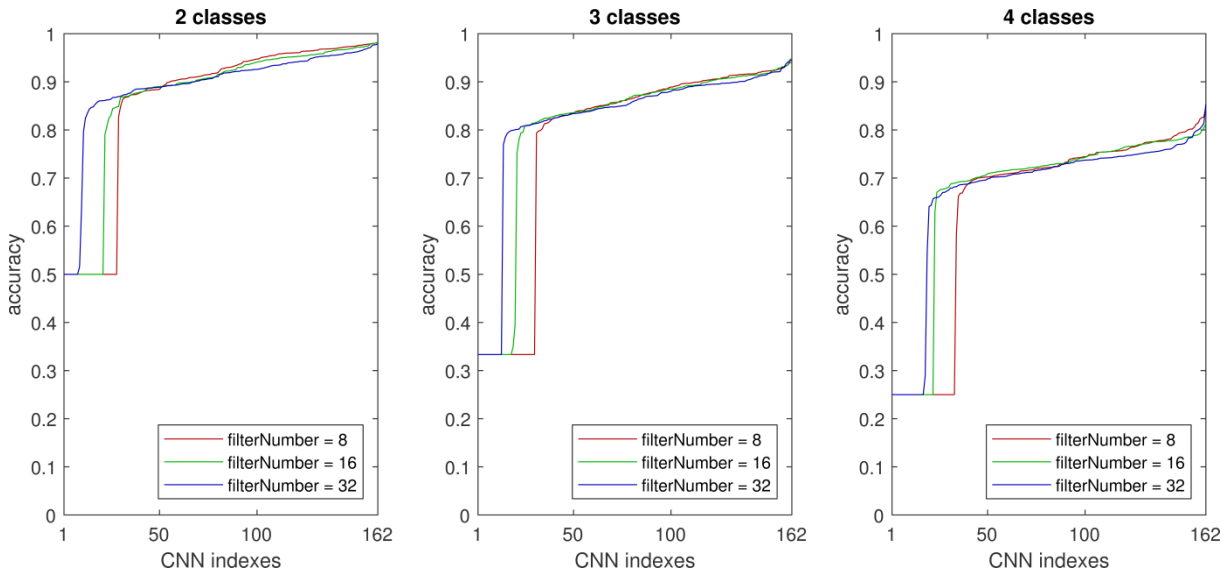

Fig.9. Overall accuracy of CNNs for different numbers of filters in convolutional layers (twice as many in the 1st layer).

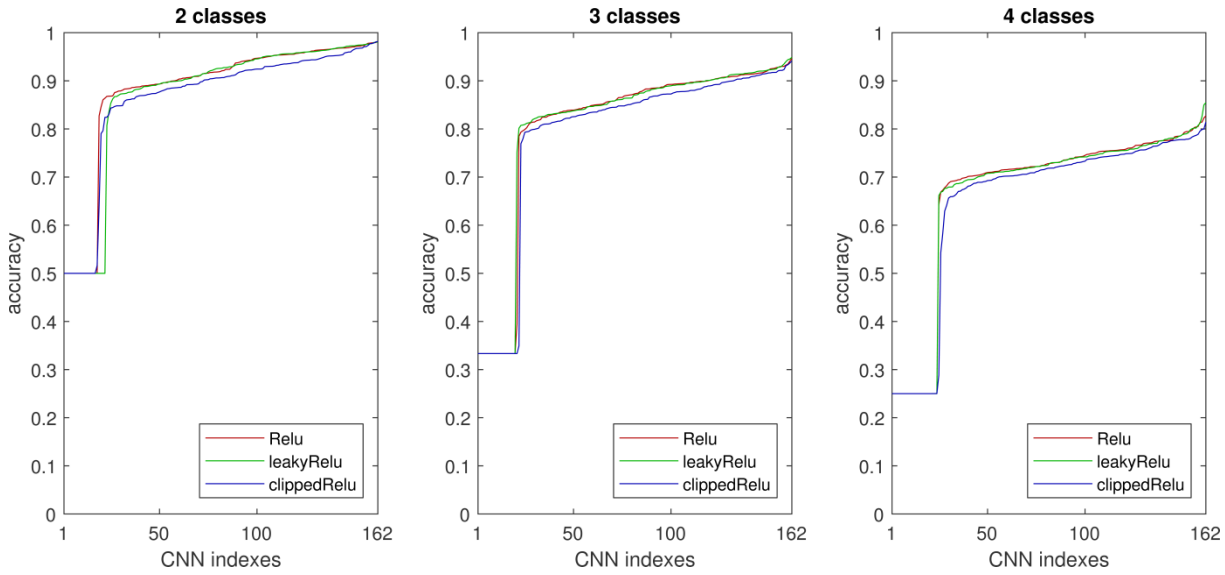

Fig.10. Overall accuracy of CNNs for different activation functions.

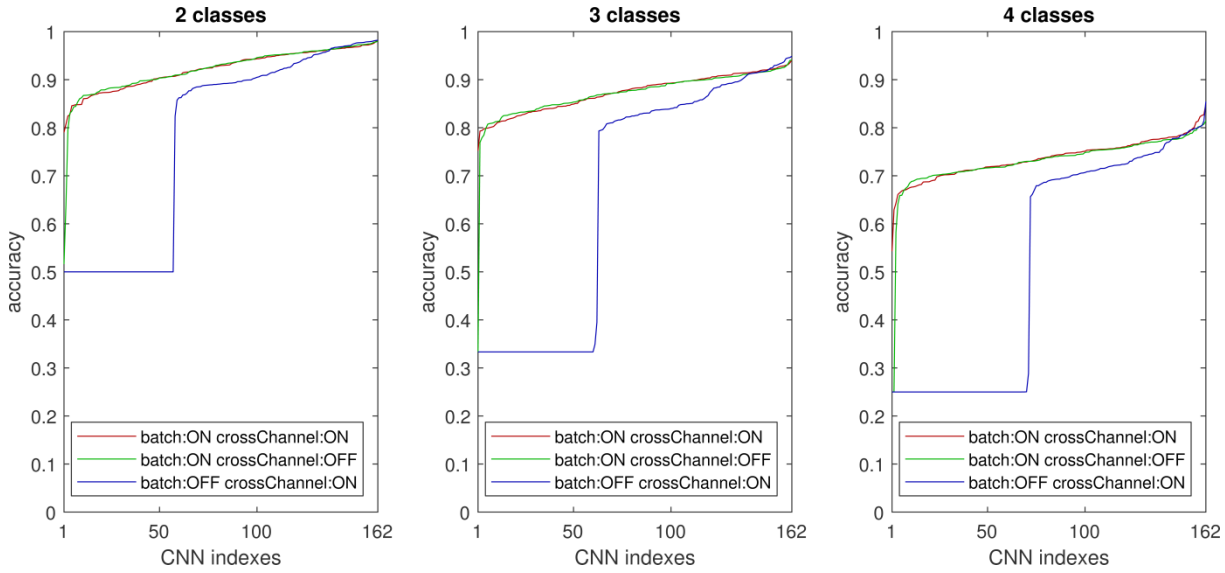

Fig.11. Overall accuracy of CNNs for different combinations of normalizations.

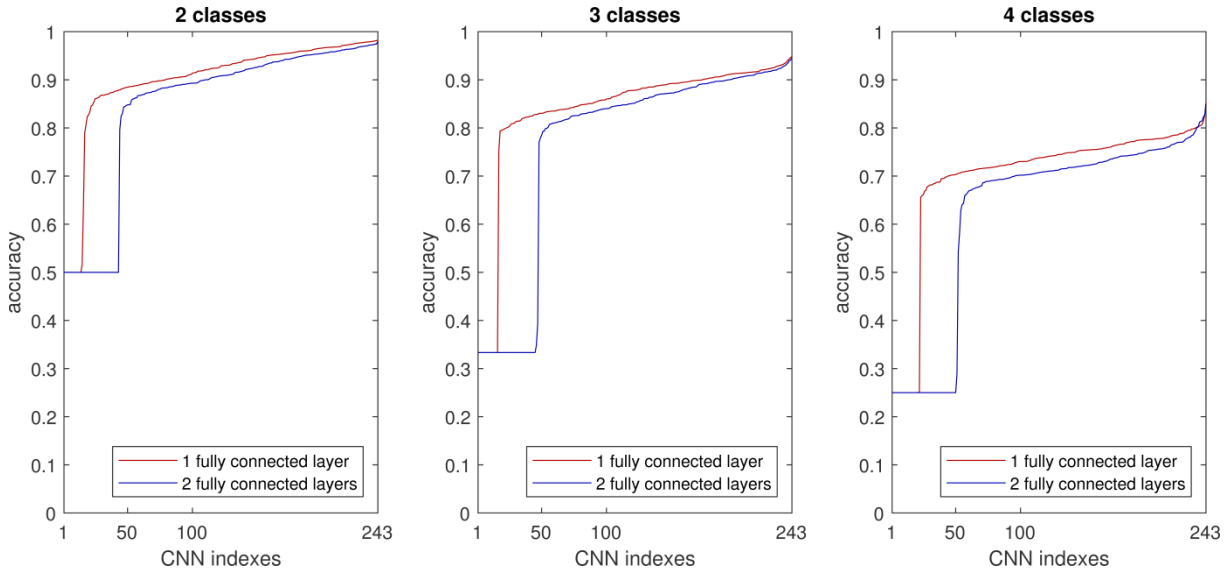

Fig.12. Overall accuracy of CNNs for different numbers of fully connected layers.

The rate of a specific class prediction was defined according to the following formula:

$$r_{p,c} = \frac{v_{p,c}}{|V_c|} \tag{2}$$

where $r_{p,c}$ is the prediction rate of a sample of the c-th class as the p-th class, $v_{p,c}$ is the number of validation data samples of the c-th class which were labeled as the p-th class by the CNN, and $|V_c|$ is the cardinality of a subset of the validation dataset for the c-th class.

The best CNN structures were indicated for each number of classes in Table III. Rates of specific class predictions were calculated for these three networks. The results are presented in Tables IV to VI. The highlighted cells in the tables apply to the correct prediction of a given class, while the rest apply to different types of incorrect predictions.

Tab.IV. Rates of specific class predictions in case of two classes.

$r_{p,c}$	p = "class 0–2"	p = "class 3"
c = "class 0–2"	0.9781	0.0219
c = "class 3"	0.0139	0.9861

Tab.V. Rates of specific class predictions in case of three classes.

$r_{p,c}$	p = "class 0"	p = "class 1–2"	p = "class 3"
c = "class 0"	0.9698	0.0292	0.0010
c = "class 1–2"	0.0478	0.9398	0.0124
c = "class 3"	0.0004	0.0522	0.9474

Tab.VI. Rates of specific class predictions in case of four classes.

$r_{p,c}$	p = "class 0"	p = "class 1"	p = "class 2"	p = "class 3"
c = "class 0"	0.9280	0.0282	0.0404	0.0034
c = "class 1"	0.0164	0.6819	0.2370	0.0647

c = “class 2”	0.0079	0.1269	0.8477	0.0175
c = “class 3”	0.0000	0.0200	0.0192	0.9608

For scenarios with two classes (Table IV) or three classes (Table V) of vessel movement anomalies, the selected CNNs provided very satisfactory results. For each class of anomaly, there is a very high (~ 0.95) rate of correct classification and very low rates of different errors, hence these scenarios can be considered solved.

However, in the scenario with four classes of anomalies (Table VI), the selected CNN classifies class 0 (0.9280) and class 3 (0.9608) very well, has reduced effectiveness for class 2 (0.8477) and is relatively often mistaken for class 1 (0.6819). Additional studies were performed and instances of the same CNNs were trained with similar results. In some cases, the situation for classes 1 and 2 was reversed, but the biggest challenge was always the correct distinction between class 1 or class 2. As a reminder, class 1 means the vessel is near the area, and will not enter the area, and class 2 means the vessel is near the area and will enter the area. Note that both classes apply when the vessel is considered to be close to the area. It was concluded that the problem may be in the form of visualization of the data samples, i.e. information on the direction of the vessel’s movement should be marked differently.

Based on the high rates in the first and the last row in Table VI, it should be understood that the networks trained on the variant C of the dataset will also very effectively classify the data from variants A and B. It can be seen from the contents of this table that the proposed architecture is very well generalized for datasets with a smaller number of classes. This is achieved mainly due to the very high rate of true positive classification in the case of "class 0" and "class 3". As a result, the network trained on a dataset in variant C can be successfully used to distinguish between a smaller number of classes, e.g. for the classification "class 0-2" vs "class 3" or "class 0" vs "class 1-2" vs "class 3". In addition, the output of the proposed method can be presented as a set of probabilities or percentage weights for each class to give the operator more context regarding the final decision in the case of classes that are very similar, e.g. "class 1" and "class 2".

7. Proposed Convolutional Neural Network

Based on the results of the research presented in the earlier sections and the conclusions drawn from them, the resulting structure of the convolutional neural network for the considered classification of vessel movement anomalies was proposed. The structure was presented in Figure 13 as well as in Table VII using the syntax of the Matlab Deep Learning Toolbox. Classification accuracy was 0.9280 for anomaly class 0, 0.6819 for anomaly class 1, 0.8477 for anomaly class 2, and 0.9608 for anomaly class 3. The overall accuracy was 0.8546. The total number of learnable parameters in the network is 52,766,424.

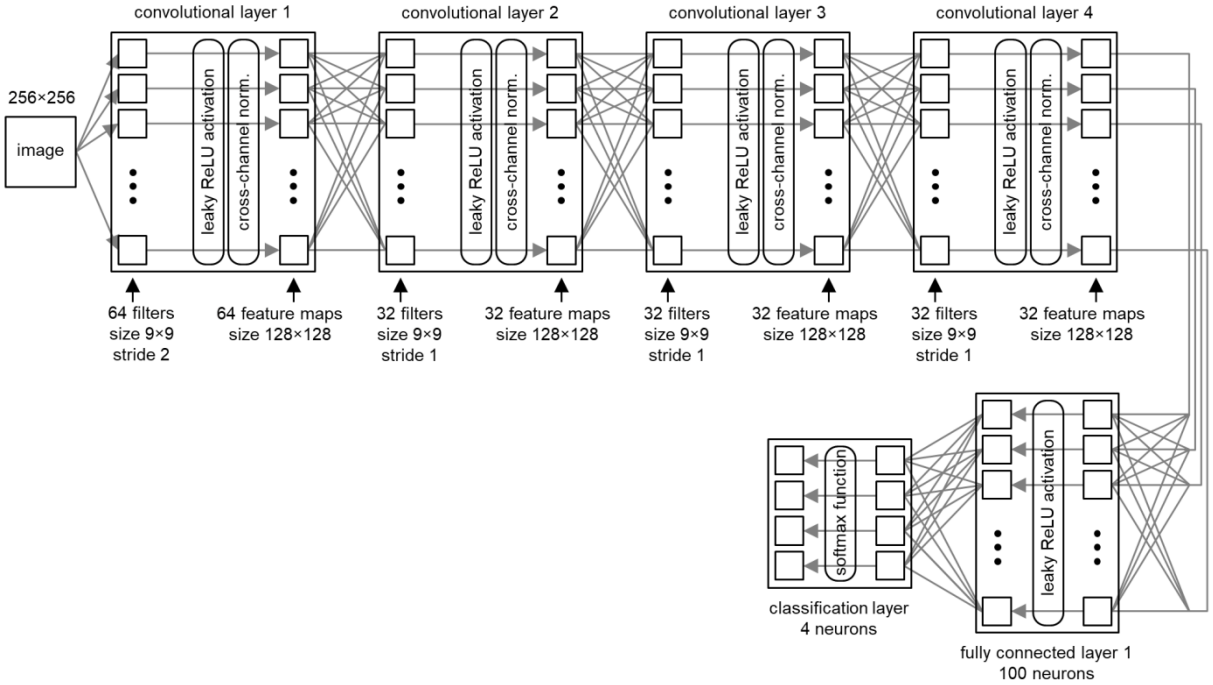

Fig.13. Final structure of the convolutional neural network for the considered classification of vessel movement anomalies.

Tab.VII. Final structure of the convolutional neural network for the considered classification of vessel movement anomalies.

Index	Layer type	Description	Learnable parameters
1	Image Input	256x256x1 images with 'zerocenter' normalization	
2	Convolution (1st)	64 9x9x1 convolutions with stride [2 2] and padding 'same'	Weights 9×9×1×64 Bias 1×1×64
3	Leaky ReLU	leaky ReLU function with scale 0.01	
4	Cross Channel Normalization	cross channel normalization with 9 channels per element	
5	Convolution (2nd)	32 9x9x64 convolutions with stride [1 1] and padding 'same'	Weights 9×9×64×32 Bias 1×1×32
6	Leaky ReLU	leaky ReLU function with scale 0.01	
7	Cross Channel Normalization	cross channel normalization with 9 channels per element	
8	Convolution (3rd)	32 9x9x32 convolutions with stride [1 1] and padding 'same'	Weights 9×9×32×32 Bias 1×1×32
9	Leaky ReLU	leaky ReLU function with scale 0.01	
10	Cross Channel Normalization	cross channel normalization with 9 channels per element	
11	Convolution (4th)	32 9x9x32 convolutions with stride [1 1] and padding 'same'	Weights 9×9×32×32 Bias 1×1×32
12	Leaky ReLU	leaky ReLU function with scale 0.01	
13	Cross Channel	cross channel normalization with 9	

	Normalization	channels per element	
14	Fully Connected	fully connected layer with 100 neurons	Weights 100×524288 Bias 100×1
15	Leaky ReLU	leaky ReLU function with scale 0.01	
16	Fully Connected	fully connected layer with 4 neurons	Weights 4×100 Bias 4×1
17	Softmax	softmax function	
18	Classification Output	cross-entropy loss function – crossentropyex with 'anomaly0', 'anomaly1', 'anomaly2', and 'anomaly3' classes.	

8. Comparison

In this section, the proposed CNN from the Section 7 is compared with multiple baseline algorithms using the same dataset. The advantages of these methods are discussed in terms of accuracy, training time, decision time, and interpretability. Additionally, we present ways to compare the proposed CNN with other algorithms outside this paper via commonly known tools such as ROC curves and confusion matrices. In the end of this section, we explain why the problem is not linearly separable.

For comparison, we chose the most important dataset for us, i.e. variant C with four classes of anomalies. The choice of this variant of the dataset was dictated by the fact that it is the closest to the real life application for which we are conducting this study. In fact, in the future, the list of anomalies will be expanded to around 20 anomalies. Although the datasets in variants A and B served well in the early stages of research for the initial selection of parameters, comparing the algorithms based on variants A and B would serve purely academic purposes and would be out of touch with practice.

The following algorithms were selected for comparison: decision tree, SVM with different kernel functions (linear, quadratic, cubic, and Gaussian), K Nearest Neighbors (KNN) with different distance metrics (Euclidean, cosine, cubic, and weighted), Ensemble classifiers with different ensemble methods (AdaBoost with decision tree learners, Random forest with decision tree learners, Subspace with discriminant learners, Subspace with nearest neighbor learners, and RUSBoost with decision tree learners). The results of the comparison are presented in Table VIII.

Tab.VIII. Comparison of the proposed CNN with baseline algorithms using the same dataset.

Algorithm	Accuracy	Training time	Decision time	Interpretability
The proposed CNN	0.855	5.97 h	0.0082 s	Easy due to heatmaps; hard otherwise
Decision Tree	0.385	20 s	0.0018 s	Easy
SVM (Linear kernel function)	0.385	1.39 h	0.0047 s	Easy

SVM (Quadratic kernel function)	0.716	2.73 h	0.0278 s	Hard
SVM (Cubic kernel function)	0.535	4.43 h	0.0293 s	Hard
SVM (Gaussian kernel function)	0.571	0.33 h	0.0390 s	Hard
KNN (Euclidean distance metric)	0.480	0.98 h	0.0185 s	Hard
KNN (Cosine distance metric)	0.531	1.04 h	0.0463 s	Hard
KNN (Cubic distance metric)	0.480	2.35 h	0.1068 s	Hard
Weighted KNN (Euclidean distance metric)	0.491	1.42 h	0.0189 s	Hard
Ensemble Classifier (AdaBoost with decision tree learners)	0.378	1.51 h	0.0126 s	Hard
Ensemble Classifier (Random forest with decision tree learners)	0.556	1.58 h	0.1242 s	Hard
Ensemble Classifier (Subspace with discriminant learners)	0.389	1.58 h	0.0195 s	Hard
Ensemble Classifier (Subspace with nearest neighbor learners)	0.503	2.02 h	0.2326 s	Hard
Ensemble Classifier (RUSBoost with decision tree learners)	0.376	2.10 h	0.0121 s	Hard

In terms of the accuracy of classification, which is the primary criteria of comparison, the most effective method is the proposed CNN with an accuracy of 0.855, followed by Quadratic SVM with an accuracy of 0.716. The next five methods are characterized by an accuracy of approx. 0.55, and the rest have an accuracy below 0.5. This clearly shows the supremacy of convolutional neural networks.

In terms of training time, the proposed CNN was found to be the slowest, with a training time of 5.97 h, which was to be expected, followed again by quadratic SVM, with a training time of 2.73 h. However, it should be noticed that this parameter is practically irrelevant, because in practical applications, an already trained and ready network would be used in the system. In addition, one should be aware of the possibility of accelerating deep network training thanks to the ever evolving Nvidia CUDA technology.

In terms of decision time, the proposed CNN is in the top 3 methods with a decision time of 0.0082 s, rivaled only by decision tree with 0.0018 s and linear SVM with 0.0047 s. However, neither decision tree nor linear SVM demonstrated satisfactory classification performance. The remaining methods have a decision time of one or even two orders of magnitude greater than the proposed CNN. This is an extremely important parameter for the use of the detection and classification of vessel movement anomalies in real-time or near real-time.

In terms of interpretability, the proposed CNN is still an attractive solution. The only ones that are very easy to interpret are decision tree and linear SVM due to their extreme simplicity and linearity. CNNs are easy to interpret if heatmaps of layer activations are analyzed, similar to the discussion in section 9. All of the other methods, including the previously mentioned quadratic SVM, are hard to interpret.

Summarizing, the proposed CNN is characterized by the best classification accuracy, with one of the lowest decision times and relatively easy interpretability after heatmap analysis. The only other method that gives at least similar classification accuracy is quadratic SVM. Therefore, only these two methods were considered for comparison analysis. To enable a precise comparison of the proposed CNN with another algorithm outside of this paper, we present the ROC curves in Figure 14 and the confusion matrices in Tables IX and X.

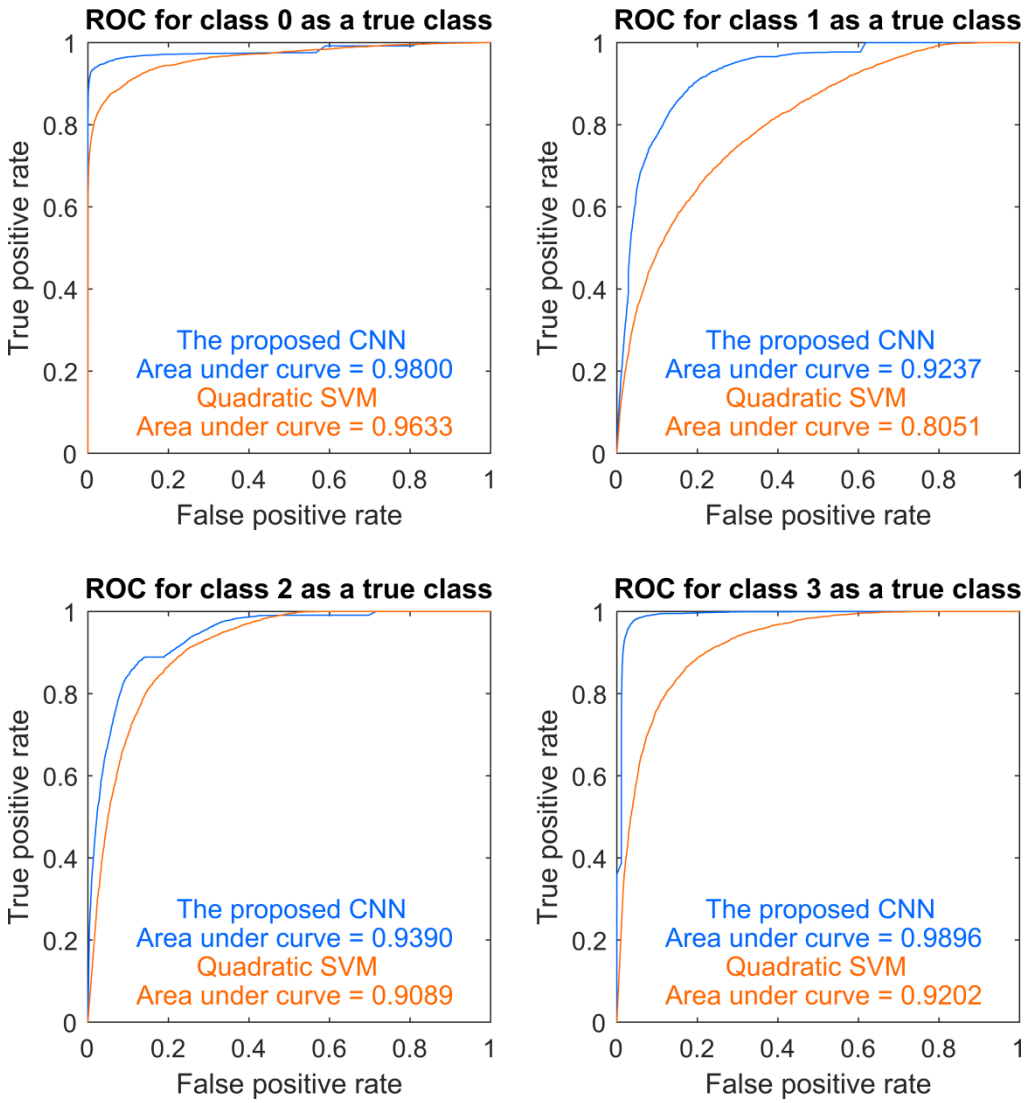

Fig.14. ROC curves for the proposed CNN and SVM with quadratic kernel function.

The ratio between true positive and the false positive, for a particular class, is plotted by the Receiver Operating Characteristic (ROC). The ROC curve for a perfect and faultless classifier is placed at right angle to the upper left edge of the graph. A line drawn at an angle of 45 degrees represents the worst case i.e. just a random guess. The overall classifier's quality is measured by the area under the ROC curve. The larger the area under the curve, the better the classification performance. As seen in Figure 14, the proposed CNN has better classification performance than the quadratic SVM, because the ROC curve for the proposed CNN is closer to the top left corner of the plot and has a larger area under the curve for every class.

Tab.IX. Confusion matrix for the proposed CNN.

Overall accuracy = 0.8546		Predicted class				True positive rate	False negative rate
		class 0	class 1	class 2	class 3		
e clas	class 0	0.9280 (4640)	0.0282 (141)	0.0404 (202)	0.0034 (17)	0.9280	0.0720

	class 1	0.0164 (82)	0.6820 (3410)	0.2370 (1185)	0.0646 (323)	0.6820	0.3180
	class 2	0.0078 (39)	0.1270 (635)	0.8478 (4239)	0.0174 (87)	0.8478	0.1522
	class 3	0.0000 (0)	0.0200 (100)	0.0192 (96)	0.9608 (4804)	0.9608	0.0392

Tab.X. Confusion matrix for the SVM with quadratic kernel function.

	Overall accuracy = 0.7156	Predicted class				True positive rate	False negative rate
		class 0	class 1	class 2	class 3		
True class	class 0	0.7878 (3939)	0.0856 (428)	0.1034 (517)	0.0232 (116)	0.7878	0.2122
	class 1	0.0236 (118)	0.4682 (2341)	0.2796 (1398)	0.2286 (1143)	0.4682	0.5318
	class 2	0.0114 (57)	0.0724 (362)	0.8294 (4147)	0.0868 (434)	0.8294	0.1706
	class 3	0.0004 (2)	0.1124 (562)	0.1104 (552)	0.7768 (3884)	0.7768	0.2232

The confusion matrix of a particular classifier allows for a display of the overall performance of the classifier for each class. It is also possible to easily identify the areas for which the classifier performed defectively. The rows and columns of the matrix represent the true class and the prediction class, respectively. Matrix cells contain prediction rates in the form of fractions and the number of classified instances in parentheses. The correspondence of the true class and the predicted class is represented by the diagonal cells. The two additional columns on the right contain the true positive rate and the false negative rate for each true class. As seen in Table IX and X, the proposed CNN has better classification performance than the quadratic SVM due to the higher true positive rates and lower false negative rates.

The results presented in this section proved that the considered classification is not a linear problem. Classes presented as images, as in Figure 4, may appear to be linearly separable, but they are not. Mainly due to the fact that in order to distinguish between class 1 and class 2, it is not enough to analyze the position of the vessel in relation to the forbidden area, but also the direction of its movement. In this way, the change of the output is not proportional to the change of the input. In addition, on the basis of consultations with the Maritime Unit of the Polish Border Guard, we plan to expand the anomaly list in the future by introducing even more nonlinearity. Examples of new vessel movement anomalies for future research are: possible collisions between ships, transit in narrow control sectors reflecting a two-way fairway, an anchored ship has moved too far from the anchor point, floating buoy, exceeding the speed limit, object separation into two objects, merging two objects into one, ships meeting, ships meeting in a restricted area, monitoring the number of objects, and much more. Given the above, the further development of any linear classifier would be a dead end.

Simultaneously, we find convolutional neural networks to be an ideal tool for our further research.

9. Layers' Activations

Deep learning techniques, in particular convolutional neural networks, are characterized by very high predictive power, but are not easily interpretable by humans. Interpreting a nonlinear classifier is important to gain trust into the prediction and to identify potential data selection mistakes. This Section describes an experiment that aims to explain and visualize how the proposed networks work. In this experiment, the activations of selected layers are shown to investigate the learned features by comparing activated locations in the convolutional layers and the corresponding locations in the input images. This is a direct analogy to magnetic resonance imaging of the activity of the human brain in which different regions also activate for different stimulations. If the pixels on the heatmap are bright, it means that the channel in question has been activated in the given location, and the network considers this feature to be important. By analyzing individual heatmaps, we can see with the naked eye what features of the input image are recognized in individual channels of the convolution layers. As input images, examples of four classes of anomalies from variant C dataset were used, which are presented in Figure 4 in Section 4. As the examined CNN, the network with the best classification results for variant C dataset was used, which was described in Section 6 and 7.

Each convolutional layer is made of multiple channels, which are two-dimensional arrays. Typically, each channel identifies one of useful features of the input image. Figure 15 presents the activations of the 1st convolutional layer for a data sample of anomaly class 2. There are 64 channels in the first convolutional layer. Each subplot in Figure 15 shows the normalized output of a different channel. Black/maroon pixels represent weak activations, white/yellow pixels represent strong activations, and orange/red are medium. The bright pixels in the channel's output reflect the localization in the input image for which the channel was activated. Figure 16 shows the output activations from the 21st (first row) and 44th (second row) channel of the 1st convolutional layer for different anomaly classes. By comparing with Figure 4, it can be noticed that the 21st channel activated on the edges, while the 44th channel activated on the outside of the restricted area. The first convolutional layer is usually for learning such simple features of the input image as color or edges.

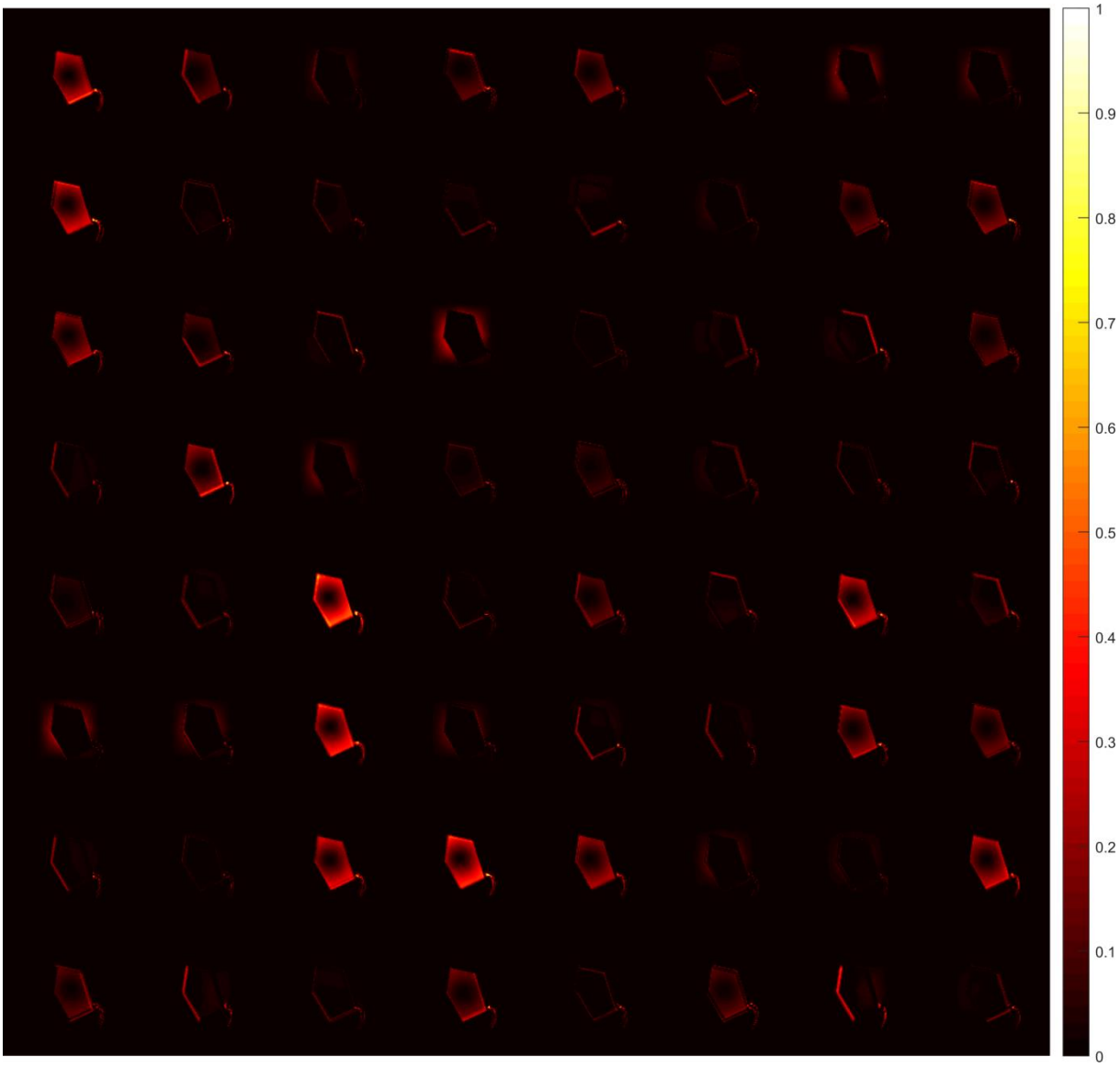

Fig.15. Output activations of 1st convolutional layer for an anomaly class 2 sample.

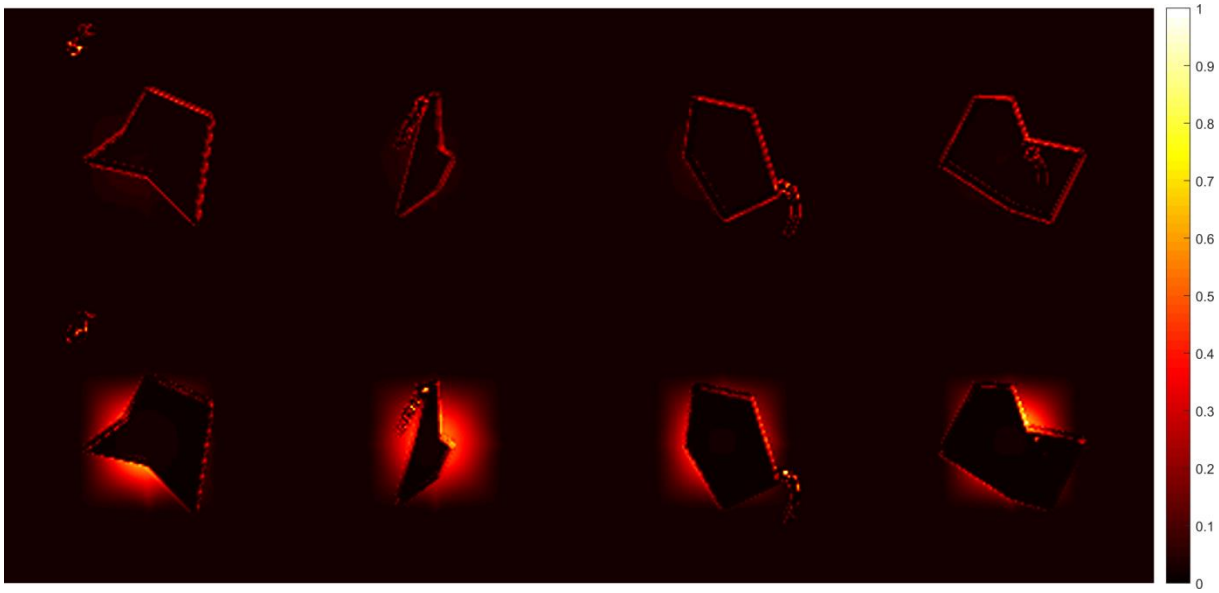

Fig.16. Output activations from 21st (first row) and 44th (second row) channel of 1st convolutional layer for different anomaly classes.

Deeper convolutional layers are meant to learn more sophisticated features of the input image. Features of deeper layers are formed by joining features of earlier layers. Figure 17 shows the output activations of the 4th convolutional layer (after activation function) for a data sample of anomaly class 2. The fourth convolutional layer has 32 channels. This is the final convolutional layer of the network. It is followed only by the fully connected layer and the output classification layer. It can be seen that some channels learned to activate on vessels of a class 2 anomaly. As previously, there are a lot of images to analyze, thus only two channels were selected for detailed analysis. Figure 18 shows the output activations from the 7th (first row) and 32nd (second row) channel of the 4th convolutional layer for different anomaly classes. It can be observed that the 32nd channel activated strongly only on vessels far from the forbidden area and did not activate on vessels near or inside the area. The brighter pixels appear only at the location of the vessel from anomaly class 0 in Figure 4. Therefore, it should be understood that the network uses this channel to recognize a class 0 anomaly. Analogously, the 7th channel activates strongly on vessels outside the forbidden area. Hence, it can be considered that the network uses this channel to recognize classes other than class 3.

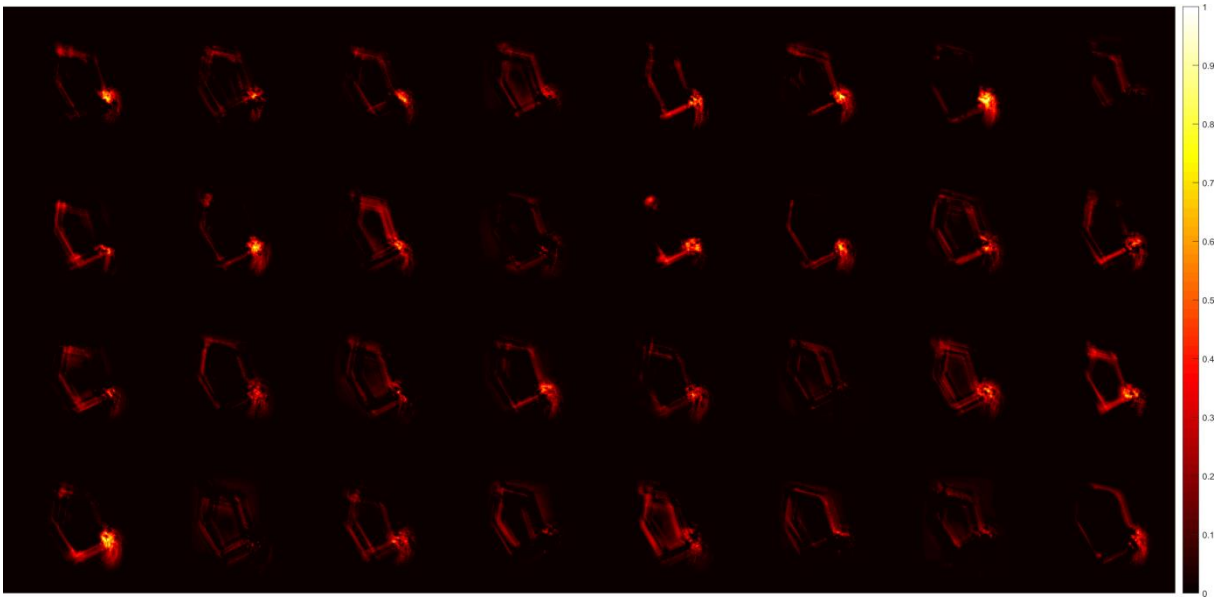

Fig.17. Output activations of 4th convolutional layer for an anomaly class 2 sample.

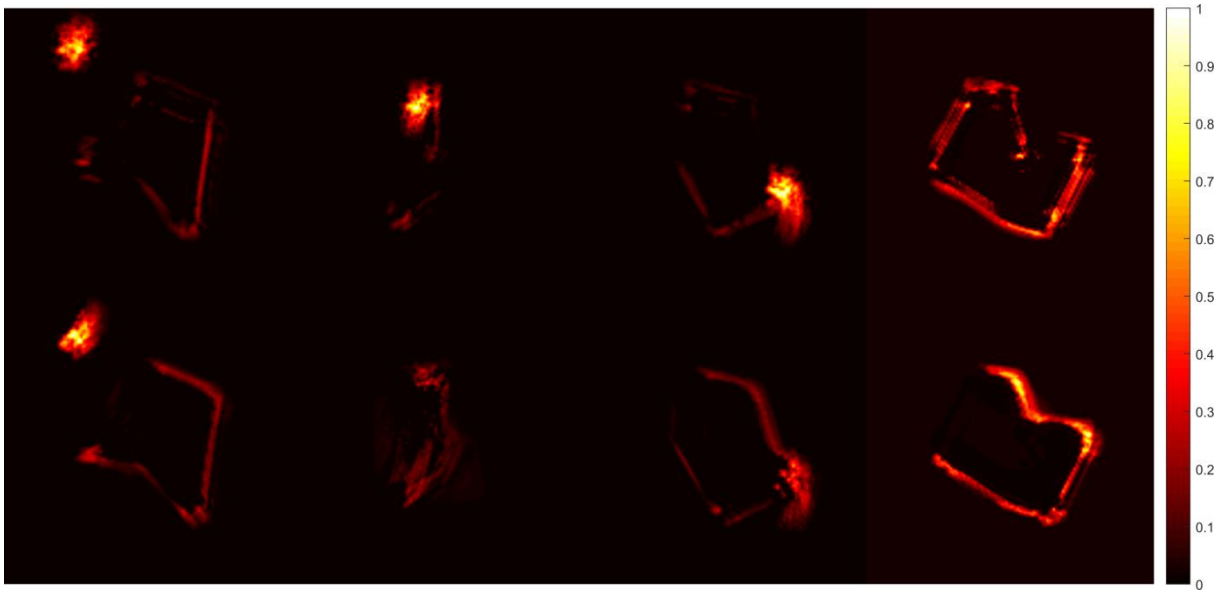

Fig.18. Output activations from 7th (first row) and 32nd (second row) channel of 4th convolutional layer for different anomaly classes.

A similar analysis can be carried out for the other channels of the convolutional layers. Each channel is to provide information about one or more features of a data sample. Then, the goal of the fully connected layer is to implement a function that will provide correct classification results based on the features learned in the convolutional layers.

10. Conclusions

The paper concerns the problem of vessel movement anomaly detection for the services dealing with maritime and coastal traffic safety. The article describes the results of multistage research. In cooperation with the Maritime Regional Unit of the Polish Border Guard, classes of vessel movement anomalies were defined. Software for generating datasets that contain samples of vessel traffic routes in relation to the prohibited area in the form of a grayscale image was written. Three variants of the datasets for different numbers of anomaly classes were generated. 1458 convolutional neural networks with different structures were trained to find the most favorable structure for anomaly classification. The influence of various parameters of network structures on the overall accuracy of classification was examined. Based on the conclusions drawn, the best structures of convolutional neural networks were selected due to the overall accuracy. For the best networks, individual class prediction rates were examined and a problem with data sample visualization was diagnosed, which led to less effective division between classes 1 and 2. The final convolutional neural network for the vessel movement anomaly detection was proposed, for which classification accuracy was 0.9280 for anomaly class 0, 0.6819 for anomaly class 1, 0.8477 for anomaly class 2, and 0.9608 for anomaly class 3. The overall accuracy was 0.8546. The total number of learnable parameters in the network is 52,766,424. For comparison with other machine learning algorithms, ROC curves were drawn and confusion matrices were calculated. Activations of selected convolutional layers were studied and visualized to present how the network works in a friendly and understandable way.

Future work will concern three areas. Firstly, the set of vessel movement anomaly classes will be expanded to include anomalies of relative movement of two objects that would reflect the scenarios of a ship hijacking or a contraband transfer. Secondly, new anomaly classes of unusual paths will be added that reflect the scenarios of a deviating ship or a ship that has moved to some location and then returned to the correct course. Thirdly, the next convolutional neural networks will be taught using real life data rather than artificially generated data.

References:

[1] Wimpenny G. Protecting AIS and the VHF Data Exchange System (VDES) Against Spoofing. 2019 European Navigation Conference (ENC 2019); 2019, Warsaw, Poland.

[2] Li X, Han J, Kim S. Motion-Alert: Automatic anomaly detection in massive moving objects. Proceedings of the 2006 IEEE Intelligence and Security Informatics Conference (ISI 2006); 2006, San Diego, USA, p. 166–177.

[3] Rhodes B, Bomberger N, Seibert M, Waxman A. Maritime situation monitoring and awareness using learning mechanisms. Military Communications Conference; 2005, p. 646–652.

[4] Rhodes BJ, Bomberger NA, Freyman TM, Kreamer W, Kirschner L, L'Italien AC, Mungovan W, Stauffer C, Stolzar L, Waxman AM, Seibert M. SeeCoast: Persistent surveillance and automated scene understanding for ports and coastal areas. Proceedings of SPIE; 2007, vol. 6578, 65781M.1–65781M.12.

[5] Rhodes BJ, Bomberger NA, Zandipour M. Probabilistic Associative Learning of Vessel Motion Patterns at Multiple Scales for Maritime Situation Awareness. The 10th International Conference on Information Fusion; 2007, Quebec, Canada.

[6] Kraiman JB, Arouh SL, Webb ML. Automated anomaly detection processor. Proceedings of SPIE; 2002, vol. 4716, p. 128–137.

[7] Laxhammar R. Anomaly detection for sea surveillance. Proceedings of the 11th International Conference on Information Fusion; 2008, Cologne, Germany, p. 55–62.

[8] Laxhammar R, Falkman G, Sviestins E. Anomaly detection in sea traffic – a comparison of the Gaussian Mixture Model and the Kernel Density Estimator. Proceedings of the 12th IEEE International Conference on Information Fusion; 2009, Seattle, USA, p. 756–763.

[9] Pallotta G, Vespe M, Bryan K. Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction. Entropy; 2013, 15 (6), p. 2218–2245.

[10] Mascaroa S, Nicholson A, Korb K. Anomaly detection in vessel tracks using Bayesian networks, International Journal of Approximate Reasoning; 2014, 55 (1), p. 84–98.

[11] Johansson F, Falkman G. Detection of vessel anomalies – a Bayesian network approach. 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information; 2007, Melbourne, Australia.

[12] Leclerc M, Tharmarasa R, Florea MC, Boury-Brisset AC, Kirubarajan T, Duclos-Hindi´e N. Ship Classification using Deep Learning Techniques for Maritime Target

Tracking. Proceedings of the 21st International Conference on Information Fusion; 2018, Cambridge, UK, p. 10–13.
[13] Protopapadakis E, Voulodimos A, Doulamis A, Doulamis N, Dres D, Bimpas M. Stacked Autoencoders for Outlier Detection in Over-the-Horizon Radar Signals. Computational Intelligence and Neuroscience; 2017, vol. 2017, p. 1–11.
[14] The Heatmapping Project, http://heatmapping.org/; 2019 [accessed 5 March 2019].
[15] Samek W, Montavon G, Vedaldi A, Hansen LK, Müller KR. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer LNCS; 2019, vol. 11700.