

Ship Classification in TerraSAR-X Images With Convolutional Neural Networks

Carlos Bentes¹, *Member, IEEE*, Domenico Velotto², *Member, IEEE*, and Björn Tings³

Abstract—Synthetic aperture radar (SAR) is an important instrument for oceanographic observations, providing detailed information of oceans' surface and artificial floating structures. Due to advances in SAR technology and deployment of new SAR satellites, an increasing amount of data is available, and the development of efficient classification systems based on deep learning is possible. A deep neural network has improved the state of the art in classification tasks of optical images, but its use in SAR classification problems has been less exploited. In this paper, a full workflow for SAR maritime targets detection and classification on TerraSAR-X high-resolution image is presented, and convolutional neural networks (CNNs) recently proposed in the literature are cross evaluated on a common data set composed of five maritime classes, namely, cargo, tanker, windmill, platform, and harbor structure. Based on experiments and tests, a multiple input resolution CNN model is proposed and its performance is evaluated. Our results indicate that CNNs are efficient models to perform maritime target classification in SAR images, and the combination of different input resolutions in the CNN model improves its ability to derive features, increasing the overall classification score.

Index Terms—Neural networks, object detection, synthetic aperture radar.

I. INTRODUCTION

DETECTION of metallic targets and estimation of environmental metocean parameters in oceanographic synthetic aperture radar (SAR) images have helped to deal with a large variety of applications, such as ocean pollution monitoring, natural oil slick detection, ship detection for illegal fishing monitoring, and disaster risk assessment [1]–[5].

The SAR system is widely recognized as a key monitoring tool since it is capable of providing on a large scale detailed information of the ocean surface as well as man-made floating structures [6]. Some, but not limited to, man-made floating structures are ships and oil platforms, which are typically responsible for a larger radar backscatter energy compared with the surrounding sea surface. Targets can be detected using variations of the constant false alarm rate (CFAR) algorithm [7]–[9]. Moreover, SAR data contain much more information than signal amplitude. In fact, the signal phase recorded by the sensor can be exploited to generate multiple azimuth synthetic apertures in

which the ocean clutter will become decorrelated and the ships become easily detectable [10], [11]. The recent generation of SAR satellites, e.g., RADARSAT-2, TerraSAR-X, etc., is capable of sending and receiving signals with different polarization, i.e., PolSAR, hence providing even greater information about the observed scene, allowing tailored polarimetric ship detectors to outperform the standard CFAR [12], [13]. Nevertheless, none of the aforementioned detectors is capable of distinguishing between different types of objects, and an additional classification step is needed to perform detection-label associations.

Many researchers have approached the detection-label association issue in SAR images by employing the concept of a neural network (NN). This powerful tool has been available for many years and a limited list of SAR maritime applications solved with this method can be found in [14]–[17] and references therein. In these cases, a shallow NN architecture (consisting of one input layer, one hidden layer, and one output layer) has been adopted, with the additional extraction of handcrafted features, e.g., object intensity, contour, area, texture, etc.

Recent advances in NN mathematical formulation and computation optimization have made it possible to train deep architectures in an efficient way [18]–[20], boosting the use of deep neural networks (DNNs) in image classification problems. DNNs are able to learn complex representations from the raw input data, without the need of handcrafted features. Since satellite's SAR image visual interpretation and understanding is not straightforward even for humans, the application of DNN techniques to SAR classification problems has been limited, or at least less exploited compared to the optical image classification problems. In [21], a DNN architecture is used to perform land cover classification using spaceborne high-resolution SAR images. In [22], a deep convolutional neural network (CNN) is applied to the problem of terrestrial target classification using airborne SAR data.

Our initial results and tests in the use of DNN for classification of SAR maritime targets were shown in [23], and an application of convolutional networks to distinguish between ships and icebergs was presented in [24]. To the best of our knowledge, the results in [23] and [24] were the first showing the capabilities of DNN architectures for SAR maritime applications with the focus on ship classification.

In this paper, a full workflow for SAR maritime targets detection and classification on TerraSAR-X high-resolution data is presented. The developed algorithm ingests a SAR ground range detected product and provides in output five classes of maritime targets: cargo ship, tanker ship, windmill, platform, and harbor

Manuscript received May 1, 2017; revised August 14, 2017; accepted October 21, 2017. Date of publication November 16, 2017; date of current version January 11, 2018. (Corresponding author: Carlos Bentes.)

Guest Editor: M. Migliaccio.

C. Bentes is with Data Analysis, Software Technology and Application Center (STACC), Tartu, Tartumaa 51003, Estonia (e-mail: carlos.bentes@tum.de).

D. Velotto and B. Tings are with the German Aerospace Center, Remote Sensing Technology Institute, Bremen 28199, Germany (e-mail: domenico.velotto@dlr.de; bjoern.tings@dlr.de).

Digital Object Identifier 10.1109/JOE.2017.2767106

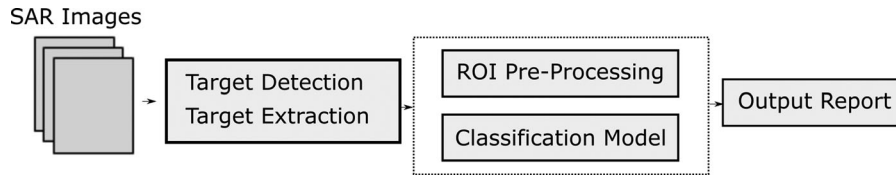


Fig. 1. Target classification pipeline. Targets are detected and extracted from SAR images using CFAR. A log-transformation and segmentation is applied to normalize the input images, and a CNN model is used to predict the target label.

structure. The choice of these five classes is motivated by the following facts.

- 1) Among the ship types, we have selected cargo and tanker because these represent the majority of ship types worldwide and are the most difficult to discriminate when it comes to SAR classification [25].
- 2) Among other maritime targets, we have selected windmill and oil platform structures because with the increase of offshore wind energy production and deep water oil/gas extraction, the number of these structures is increasing and an updated chart with their exact location is not available.
- 3) Last but not the least, we have included the class harbor structure since with the economic growth of port economy new structures are being built, but are not included in medium-resolution worldwide landmask databases.

The focus of this paper is the exploitation of CNN techniques to solve the classification problem at hand, while the detection itself is kept simple and based on standard CFAR algorithm. Note that CNN could be used to directly detect each class of target of interest without the *a priori* CFAR detections, but in our opinion the scalability and adaptation of this approach to other SAR sensors is limited. For this reason, this paper deals with the direct training of the CNNs based on the extracted patches of SAR images relative to the five classes of targets.

The data set is composed of real targets extracted from a large set of X-band StripMap SAR images, covering different incidence angles, sea state conditions, and target orientations. The corresponding ground truth has been obtained exploiting different sources, as, for example, automatic identification system (AIS) to identify cargo and tanker ships, offshore rigs historical database to identify platforms, and nautical charts to identify harbor and windmill structures.

In summary, the major contributions of this research are as follows.

- 1) The performance of our initial CNN model proposed in [23], hereafter CNN-A, is improved by adding a nonlinear data normalization function in preprocessing.
- 2) A baseline classifier based on principal component analyses (PCA) and support vector machine (SVM) is developed and used as a comparison measure.
- 3) CNN-A and three additional CNN models recently proposed in the literature, CNN-B [26], CNN-C [27] and CNN-D [28], are cross evaluated on the same data set and compared with the PCA+SVM baseline classifier.
- 4) Based on several experiments and tests performed to investigate the effects of the input SAR image resolution on the CNN model performance, a final multiple input

resolution CNN model (CNN-MR) is proposed and its performance is evaluated.

This paper is organized as follows. Section II presents the overview of the implemented workflow and its building blocks. Section III describes the classification data set characteristics, formation, and augmentation. In Section IV, the achieved classification performance of the baseline classifier, i.e., PCA+SVM, is first introduced, followed by analyses of the capabilities of the five considered CNNs to solve the problems at hand. Finally, Section V presents our conclusions and future work directions.

II. SYNTHETIC APERTURE RADAR MARITIME TARGETS CLASSIFICATION WORKFLOW

The overall architecture of the classification system is presented in Fig. 1.

It is composed of two main building blocks (i.e., target detection and extraction, and classification model), where the initial input is a TerraSAR-X high-resolution SAR ground range detected product, and the output is the predicted label for each target extracted. The details of each building block are provided in Sections II-A and II-B.

A. Target Detection and Extraction

The SAR input image is first calibrated to sigma zero (σ_0) and processed to detect all targets and obtain their centroids in image coordinates. A full description of the detection process is out of the scope of this paper. The general working idea of the CFAR algorithm applied to SAR data to detect ships can be found in [29], while the TerraSAR-X adapted CFAR version and parametrization is fully described in [7] and the accuracy and methodology on the ship shape and geographical features extraction in [30]. For all detected targets, a region of interest I_{ROI} is extracted from the targets centroid position.

The size of every I_{ROI} is fixed to 128×128 pixels, with a fixed pixel spacing of 2.5 m. This corresponds to a region of 320×320 m around the target's centroid, and corresponds to a compromise between the expected target size and the minimum distance between targets.

The choice of having a fixed 2.5-m pixel spacing is imposed by two factors. 1) In this way, we take into account the slightly different pixel spacing TerraSAR-X data provided for the different StripMap beams (between 1.25 and 1.75 m). 2) With finer pixel spacing, the SAR chip size containing the targets of interest becomes too big and this will affect the computation and time cost during the training.

If more than one target is present inside the I_{ROI} chip, a simple image processing procedure is designed to mask out all possible

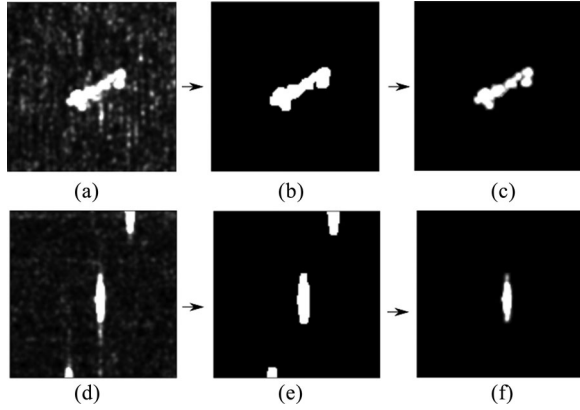


Fig. 2. Target extraction example. (a) Original target and cluttered intensity image. (b) Binary mask after image segmentation and binary morphological operations. (c) Final segmented I_{ROI} showing the target extraction results, where the background clutter is removed. (d) I_{ROI} containing multiple targets. (e) Binary mask. (f) Final segmented I_{ROI} containing the extracted target. Although multiple targets were detected in the intermediate mask, only the central target is preserved.

extra targets (or part of them). Please note that this process applied during the target extraction procedure is of paramount importance. It mitigates the influence of both the presence of additional target and clutter surrounding the target of interest centered in I_{ROI} .

Fig. 2 illustrates the target extraction process. The detected target is centralized in the I_{ROI} chip [see Fig. 2(a)]. The I_{ROI} chip is segmented and a foreground target indicator binary mask is generated [see Fig. 2(b)] using an adaptive threshold [29]. This binary mask is adjusted using morphological operations of opening and dilation. The size of the morphological kernel is defined empirically to 3×3 pixels, taking into account the quality of the final extracted targets. Alternatively, the morphological kernel size could be considered as a hyperparameter and chosen to minimize the classification error using the training data set [30]. Finally, the obtained binary mask [see Fig. 2(b)] is used to extract the signal of the target of interest centered in I_{ROI} [see Fig. 2(c)], removing outliers.

Fig. 2(d) illustrates the case where more than one target (part of other windmills structures) is present inside the I_{ROI} chip. All regions are detected using a region-growing labeling process, and only the target in the central region is preserved. Similar to Fig. 2(c), Fig. 2(f) shows that the sea clutter has been removed as well as the signal of the extra targets, while the signal of the target of interest is kept.

Due to the high dynamic range of SAR images, an intensity normalization step is needed to ensure that the backscatter intensity associated with all the different classes of target is scaled within the same range. The following nonlinear normalization function was applied to every I_{ROI} pixel [24]

$$N(\sigma_0) = \frac{L(\sigma_0)}{\max L(\sigma_0)} \quad (1)$$

where

$$L(\sigma_0) = \begin{cases} 1 + \log \sigma_0, & \text{if } \sigma_0 > 1 \\ \sigma_0, & \text{if } \sigma_0 \leq 1 \end{cases} \quad (2)$$

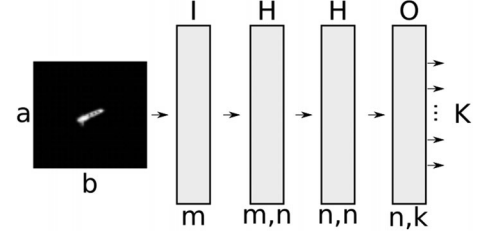


Fig. 3. MPL architecture. The input image of size $a \times b$ is vectorized (row by row) and connected to the input layer I of the network. This illustrative architecture contains two hidden layers H and an output layer O that maps the internal network function to a total of K output classes.

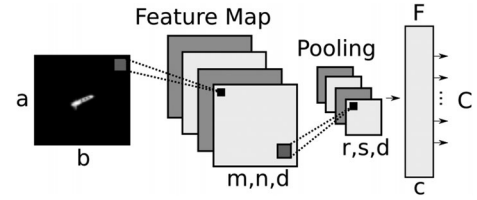


Fig. 4. CNN architecture. The image of size $a \times b$ is connected to the convolutional layer of dimensions $m \times n \times d$. The layer of size $m \times n$ contains the output of the filter kernel. Each layer d is downsampled with a max-pooling operation, resulting in an output field of size $r \times s$. The output of the convolutional layers is an internal feature representation C with dimension $c = r \cdot s \cdot d$.

B. Classification Models

The first and intuitive DNN architecture is the multilayer perceptron (MLP) [31], composed of one input layer I followed by sequential fully connected hidden layers H and one output layer with dimension equal to the number of output classes, as illustrated in Fig. 3.

In this paper, MLP models are not considered and further details on the performances of this type of DNNs for maritime targets classification in SAR images at varying the number of layers and neurons can be found in [23].

The CNN model uses the same basic neuron unit used in the MLP, but the connections between neurons inside the network are optimized to work with images, taking into account the correlation between neighboring pixels.

CNNs are powerful image features extractors, and therefore have been used in many image classification problems, as shown in [19], [32], and [33].

Fig. 4 presents the typical architecture of a CNN, where for graphical simplicity only one general convolution layer that extracts d features maps of size $m \times n$, and one pooling layer that reduces the size of each d features map to $r \times s$ is shown.

An input image of size $a \times b$ is connected to a set of input neurons preserving the 2-D arrangement. These connections define an observation region over the image, and act as a filter window (also known as receptive field).

The filter window (i.e., the blue kernel in Fig. 4) moves along the image, forming a 2-D feature map that corresponds to a image-filter convolution. The result is an NN layer that captures spatially invariant features.

Each convolutional layer is typically followed by a pooling layer to reduce the amount of parameters inside the network and avoid overfitting. In this paper, a max-pooling layer is used, where the output of the pooling operation is defined as the

TABLE I
NETWORK PARAMETERS

Layer	CNN-A [23]	CNN-B [26]	CNN-C [27]	CNN-D [28]
1	Input (128×128)	Input (128×128)	Input (128×128)	Input (128×128)
2	Conv. $32@(2 \times 2)$ - ReLU	Conv. $96@(3 \times 3)$ - ReLU	Conv. $16@(5 \times 5)$ - ReLU	Conv. $18@(9 \times 9)$
3	Conv. $32@(2 \times 2)$ - ReLU	Conv. $96@(3 \times 3)$ - ReLU	Max Pooling (2×2)	Max Pooling (6×6)
4	Max Pooling (2×2)	Max Pooling (2×2)	Conv. $32@(5 \times 5)$ - ReLU	Conv. $36@(5 \times 5)$
5	Conv. $64@(2 \times 2)$	Conv. $256@(3 \times 3)$ - ReLU	Max Pooling (2×2)	Max Pooling (4×4)
6	Conv. $64@(2 \times 2)$	Max Pooling (2×2)	Conv. $64@(6 \times 6)$ - ReLU	Conv. $120@(4 \times 4)$
7	Max Pooling (2×2)	Dense (1000) - Sigmoid	Max Pooling (2×2)	Dense (120)
8	Dense (256) - Sigmoid	Output (5) - Softmax	Conv. $128@(5 \times 5)$ - ReLU	Output (5) - Softmax
9	Output (5) - Softmax	—	Conv. $10@(3 \times 3)$	—
10	—	—	Output (5) - Softmax	—

maximum activation value inside the pooling kernel window. It is common practice to add a dense fully connected layer to the last convolution or pooling layer. The neurons output of the dense layer represents scores which, in turn, are converted to the probability of belonging to each designed class by the final output layer, using a softmax operation [33].

This paper considers four different CNN models and their original architectures to solve the classification problem at hand: CNN-A, from our previous work presented in [23]; CNN-B, by Ding *et al.*, presented in [26]; CNN-C, by Chen *et al.*, presented in [27]; and CNN-D, by Wilmanski *et al.*, presented in [28]. Table I summarizes the architectures and network parameters of the four CNNs selected. As can be seen, the CNNs have different shapes, i.e., number of layers up to ten (see CNN-C), and are characterized by a diversity of networks settings, e.g., convolutional kernel size, number of convolutional filters, pooling size, etc., but they all share the same input and output size.

The proposed CNN-MR takes into account the effect of the input SAR image resolution on the CNN model performance, and uses different resolution versions of the same I_{ROI} in a three-channel input. The channels are organized as follows: Channel A is connected to the original I_{ROI} with resolution of 3[m]; Channel B is connected to an I_{ROI} with resolution of 12[m]; and Channel C is connected to an I_{ROI} version with 24[m] resolution. The different resolutions are obtained by increasing the number of looks while keeping the pixel spacing fixed at 2.5[m] (and therefore all input channels have the same size of 128×128 pixels). The three-channel input contains different levels of feature details and speckle noise, and the combination of all channels allows the network to produce better feature representations.

The CNN-MR architecture is depicted in Fig. 5.

III. CLASSIFICATION DATA SET

The classification data set is composed of target samples extracted from TerraSAR-X, multilook ground range detected images. A total of 75 StripMap images were used. The

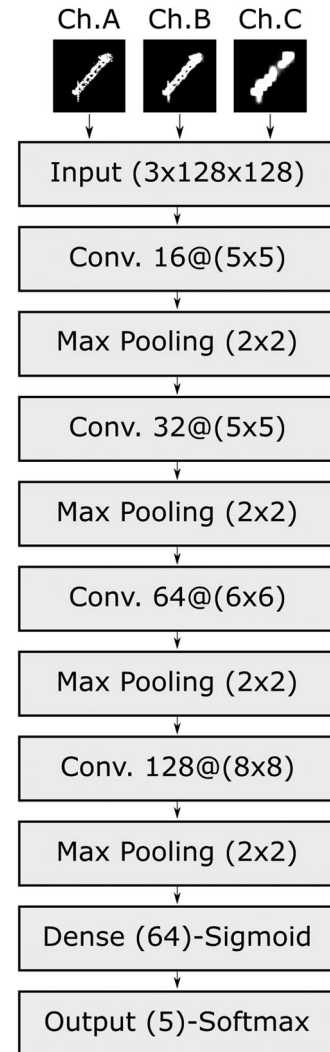


Fig. 5. CNN-MR model. This convolutional network with multiresolution input is designed to improve feature extraction from different representations of the input image. Channel A is connected to the original 3-m resolution image, channel B to the 12-m resolution, and channel C to the 24-m resolution. All channels have a fixed pixel spacing of 2.5-m and image size of 128×128 .

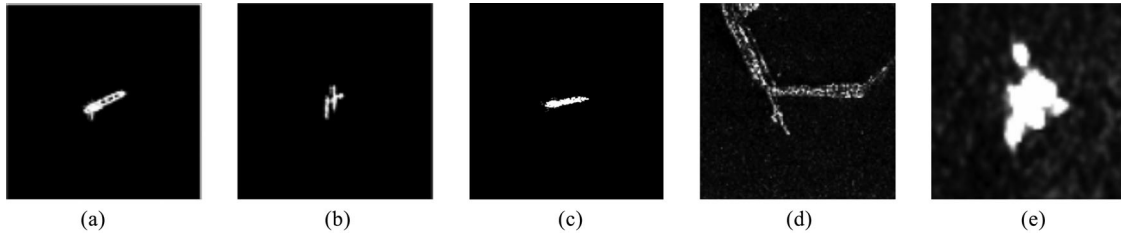


Fig. 6. Example of target classes: (a) cargo; (b) windmill; (c) tanker, (d) harbor; (e) oil platform.

TABLE II
PCA+SVM BASELINE CLASSIFIER

Label	Precision	Recall	f1-score
cargo	0.61	0.73	0.67
harbor	0.93	0.95	0.94
platform	0.98	0.98	0.98
tanker	0.76	0.58	0.66
windmill	0.84	0.85	0.84
Avg. total	0.82	0.82	0.82

TerraSAR-X StripMap images have a ground resolution of approximately 3 m, and were acquired with different incidence angles (ranging from 20° to 45°) and different polarizations (sample distribution: 73% for HH mode, 27% for VV mode). A total of 683 target samples were extracted from the satellite images, which contained a large variety of targets (sample distribution: 68% of ships, 32% of other structures) in different target orientations, moving speeds, and sizes. The diversity of target samples is necessary to allow the classification system to capture the underlying target model and better generalize to new samples in an operational scenario. The effects of strong sea clutter, bright ship's wake, and moving target distortions (e.g., smearing effect) are present in the data set, making the classification a hard task. Although not performed here, target focusing and state-of-the-art techniques such as target super-resolution [34], [35] could potentially improve the classification system performances, and this work is left for future research.

Each extracted target is stored in a processing list P_{list} . The classification model is trained with elements from P_{list} from which ground truth data are available (extracted with colocated AIS or identified manually using nautical charts). Detections containing targets that do not belong to the set of interest (cargo, tanker, windmill, platform, harbor) are removed from P_{list} . Fig. 6 illustrates the different target classes extracted from SAR images.

Classification of static targets, such as windmills structures and oil platforms, can be improved using multitemporal context analysis (using multiple images acquired at different times over one same region). Similarly, geometrical association rules such as target and land proximity can add valuable information for the classification system. Although very useful for real applications systems, the use of spatial-temporal context and other *a priori* information is not considered in this paper, and the normalized target's signal extracted from one SAR image is used as the unique source of information.

The set of targets P_{list} is artificially enlarged to balance the number of samples per class and to reduce model overfitting. This process is performed by stratified sampling the targets $I_{\text{ROI}} \in P_{\text{list}}$ (with replacement), and by applying a set of label-preserving transformations (with uniform probability), as follows:

- 1) horizontal and vertical reflections;
- 2) image rotation, with rotation angle sampled from -10° to 10° ;
- 3) image translation, with translation value sampled from -5 to 5 pixels;
- 4) noise addition by randomly setting input pixels to zero.

This set of operations augments the classification data set to a final size of 1500 samples (300 samples per class). The classification data set is divided into two sets: the training data set D_{train} (with 1200 samples) and the test data set D_{test} (with 300 samples).

IV. RESULTS AND DISCUSSION

The networks are implemented in Python code using Numpy, OpenCV, Scikit-Learn, and Theano libraries as building blocks [36]–[41]. For all network architectures investigated here, the input vector is designed to match the size of the training samples, and therefore has a dimension of $I = 128 \cdot 128 = 16384$ elements.

The desired output consists of classes $c_{\text{target}} \in \{\text{cargo, harbor, platform, tanker, windmill}\}$, encoded in a one-hot binary code of 5 bits. The output layer is implemented using the softmax operation, mapping the output vector of the last hidden layer to the five-elements class vector [42].

A. Baseline Classifier

For evaluation purposes, a baseline classifier was built using PCA followed by an SVM model. As shown in [43], the PCA+SVM model is an efficient target classifier in operational scenarios. The PCA+SVM hyperparameters are selected using grid-search and fivefold cross validation (best model from cross validation uses a Gaussian radial basis function (RBF) kernel with coefficient $\gamma = 0.001$, and penalty parameter $C = 100$).

The classification accuracy of the baseline classifier is evaluated in terms of the single metrics precision (the ratio of true positives and predicted positives) and recall (the ratio of true positives and all positives samples). However, it is a common practice to combine these metrics into one, using the harmonic mean between precision and recall. This metric is called

TABLE III
CNN RESULTS

(a) CNN-A				(b) CNN-B			
Label	Precision	Recall	f1-score	Label	Precision	Recall	f1-score
cargo	0.74	0.67	0.70	cargo	0.63	0.55	0.59
harbor	1.00	1.00	1.00	harbor	1.00	1.00	1.00
platform	0.97	0.97	0.97	platform	1.00	0.93	0.97
tanker	0.74	0.80	0.77	tanker	0.70	0.64	0.67
windmill	0.92	0.93	0.93	windmill	0.78	0.98	0.87
Avg. total	0.87	0.87	0.87	Avg. total	0.82	0.82	0.82

(c) CNN-C				(d) CNN-D			
Label	Precision	Recall	f1-score	Label	Precision	Recall	f1-score
cargo	0.76	0.65	0.70	cargo	0.81	0.70	0.75
harbor	0.98	0.98	0.98	harbor	0.98	0.97	0.97
platform	0.98	0.98	0.98	platform	1.00	0.98	0.99
tanker	0.73	0.78	0.76	tanker	0.76	0.85	0.80
windmill	0.91	0.98	0.94	windmill	0.90	0.95	0.93
Avg. total	0.87	0.88	0.87	Avg. total	0.89	0.89	0.89

f1-score, and provides a convenient basis for comparison between classifiers. The f1-score is then given as

$$\text{f1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

The results of the first experiment are summarized in Table II, where precision, recall, and f1-score achieved by the PCA+SVM baseline classifier on our test data set are listed for the considered five classes of maritime targets.

These results show that the PCA+SVM classifier has a low f1-score for the classes cargo and tanker (with scores of 0.67 and 0.66, respectively) but perform well with the other classes. This is expected, as harbor and platforms are considered easy classes (due to their distinct shape and size characteristics), while cargo and tanker are challenge classes (they have similar shape and sizes across different samples in the data set).

B. Convolutional Networks

The second experiment consist in training from scratch the four CNNs models presented early in Table I. All four CNNs models were trained with stochastic gradient descent (SGD) optimizer [44], with learning rate of 0.001. The batch size is set to 32, and the number of epochs is adjusted using the early stopping method (that is, the training procedure stops when the network error on a validation set stops decreasing) to reduce overfitting. Precision, recall, and f1-score achieved by CNN-A, CNN-B, CNN-C, and CNN-D on our test data set are listed in Table III (a)–(d), respectively.

TABLE IV
CNN-MR RESULTS

Label	Precision	Recall	f1-score
cargo	0.88	0.83	0.85
harbor	1.00	1.00	1.00
platform	0.98	1.00	0.99
tanker	0.85	0.88	0.87
windmill	1.00	1.00	1.00
Avg. total	0.94	0.94	0.94

CNN-B performs similarly to the baseline classifier. It is able to differentiate the easy classes (harbor, platform, windmill) and fails to correct distinguish between cargo and tanker classes (f1-score of 0.59 and 0.67, respectively). The large number of feature maps in the convolutional layer (see Table I) makes the model too complex, and a larger data set would probably help CNN-B performance. The networks CNN-A, CNN-C, and CNN-D present a better balance between the number of internal parameters and the available classification data. The convolutional layers performed best with rectified linear unit (ReLU) activation functions. For the dense layers, no difference in performance was observed between ReLU and Sigmoid activation functions.

The CNN-MR model (refer to Fig. 5) was trained using multi-resolution versions of the input samples from $I_{\text{ROI}} \in D_{\text{train}}$ generated on-the-fly during the train/test procedure. The CNN-MR results are shown in in Table IV.

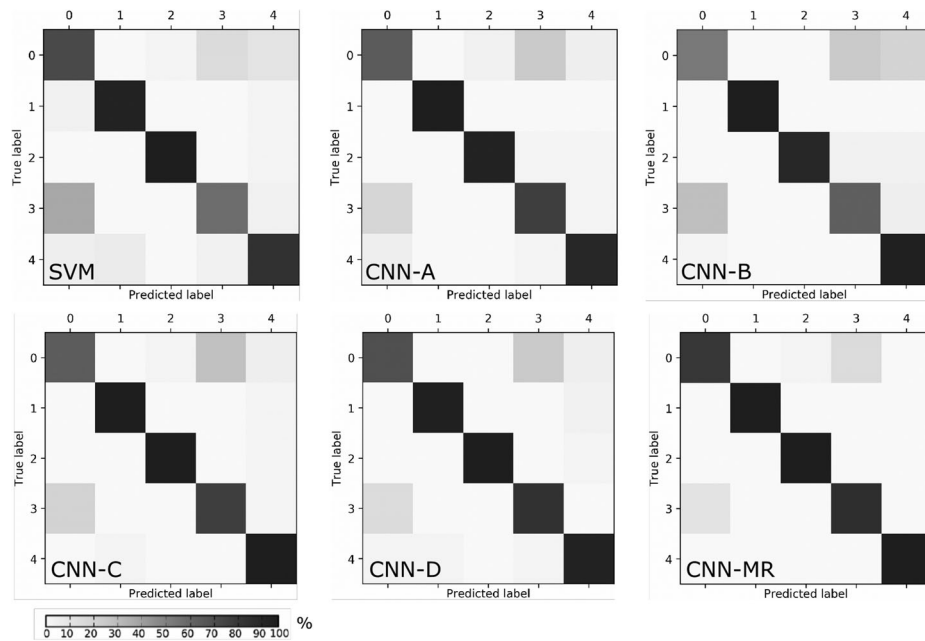


Fig. 7. Confusion matrix: 0 = cargo; 1 = harbor; 2 = platform; 3 = tanker; 4 = windmill.

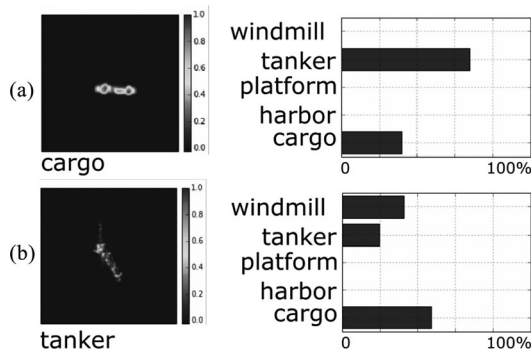


Fig. 8. Example of misclassified targets from the CNN-MR. (a) Cargo ship is classified as a tanker (higher output probability depicted on the right side image). (b) Tanker is classified as a cargo ship.

The CNN-MR obtains a total f1-score of 0.94, with good performance in the task of classifying cargo ships and tankers (f1-score of 0.85 and 0.87, respectively). These results suggest that the CNN network is able to build a better representation of the input image when presented with a variety of target details in the input channel. When the three channels are combined, the CNN-MR is able to learn from each channel the most relevant feature to discriminate between cargo and tanker classes. To get a better intuition of what features the CNN model is learning, more experiments are necessary, and this will be carried out in a future work.

Fig. 7 presents the confusion matrices of all experimented models where, as observed in the f1-scores, the classes cargo and tanker are the ones with highest number of misclassification (off diagonal) entries. Fig. 8 illustrates misclassified examples of cargo ships and tankers, where the output probability from the softmax layer is shown.

V. CONCLUSION

This paper presented a complete workflow for SAR maritime targets detection and classification on TerraSAR-X high-resolution images. A simple CFAR target detector is used, and a preprocessing step is applied to reduce spurious signals. The PCA-SVM model has an average f1-score of 0.82, and this value is used as reference to compare the CNNs models. With exception of CNN-B, all other CNN models outperformed our baseline classifier, showing a better f1-score in the discrimination of cargo and tanker classes, and a higher total average f1-score. The results clearly show that the combination of different resolutions in the CNN input layer improves the classification results. The proposed CNN-MR model scored the best classification performance in our data set, with a total average f1-score of 0.94. This implies that CNN can extract better features when presented with SAR images with different number of looks for the same target. Nevertheless, in this paper, the CNN is treated as a black box model, and additional experiments are necessary to understand how the change in image resolution affects the CNN internal activations. The use of additional procedures (e.g., target focusing, target super-resolution) in the preprocessing step can potentially improve the classification results, and will be the focus of future work.

ACKNOWLEDGMENT

TerraSAR-X data were provided by the German Aerospace Center (DLR) via AO accounts OCE2917 and OCE1045.

REFERENCES

- [1] D. Velotto, M. Migliaccio, F. Nunziata, and S. Lehner, "Oil-slick observation using single look complex terrasars-x dual-polarized data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2010, pp. 3684–3687.
- [2] G. Suresh, G. Heygster, G. Bohrmann, C. Melsheimer, and J.-H. Korber, "An automatic detection system for natural oil seep origin estimation

- in SAR images,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2013, pp. 3566–3569.
- [3] S. Singha, D. Velotto, and S. Lehner, “Near real time monitoring of platform sourced pollution using TerraSAR-X over the north sea,” *Mar. Pollut. Bull.*, vol. 86, no. 1, pp. 379–390, 2014.
 - [4] G. Pallotta, M. Vespe, and K. Bryan, “Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction,” *Entropy*, vol. 15, no. 6, pp. 2218–2245, 2013.
 - [5] S. Voigt, T. Kemper, T. Riedlinger, R. Kiefl, K. Scholte, and H. Mehl, “Satellite image analysis for disaster and crisis-management support,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1520–1528, Jun. 2007.
 - [6] C. R. Jackson *et al.*, *Synthetic Aperture Radar Marine User’s Manual*. Washington, DC, USA: U.S Dept. Commerce, 2004.
 - [7] S. Brusch, S. Lehner, T. Fritz, M. Soccorsi, A. Soloviev, and B. van Schie, “Ship surveillance with TerraSAR-X,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 1092–1103, Mar. 2011.
 - [8] G. Gao, L. Liu, L. Zhao, G. Shi, and G. Kuang, “An adaptive and fast CFAR algorithm based on automatic censoring for target detection in high-resolution SAR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 6, pp. 1685–1697, Jun. 2009.
 - [9] A. Frost, R. Ressel, and S. Lehner, “Iceberg detection over northern latitudes using high resolution TerraSAR-X images,” in *Proc. 36th Can. Symp. Remote Sens.*, 2015, pp. 1–8.
 - [10] S. N. Anfinsen and C. Brekke, “Statistical models for constant false alarm rate ship detection with the sublook correlation magnitude,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 5626–5629.
 - [11] A. Marino, M. J. Sanjuan-Ferrer, I. Hajnsek, and K. Ouchi, “Ship detection with spectral analysis of synthetic aperture radar: A comparison of new and well-known algorithms,” *Remote Sens.*, vol. 7, no. 5, pp. 5416–5439, 2015.
 - [12] F. Nunziata, M. Migliaccio, and C. E. Brown, “Reflection symmetry for polarimetric observation of man-made metallic targets at sea,” *IEEE J. Ocean. Eng.*, vol. 37, no. 3, pp. 384–394, Jul. 2012.
 - [13] A. Marino, “A notch filter for ship detection with polarimetric SAR data,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1219–1232, Jun. 2013.
 - [14] P. M. Atkinson and A. Tatnall, “Introduction neural networks in remote sensing,” *Int. J. Remote Sens.*, vol. 18, no. 4, pp. 699–709, 1997.
 - [15] F. Wu, C. Wang, S. Jiang, H. Zhang, and B. Zhang, “Classification of vessels in single-pol COSMO-SkyMed images based on statistical and structural features,” *Remote Sens.*, vol. 7, no. 5, pp. 5511–5533, 2015.
 - [16] D. Velotto, M. Migliaccio, F. Nunziata, and S. Lehner, “Dual-polarized TerraSAR-X data for oil-spill observation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4751–4762, Dec. 2011.
 - [17] A. Makedonas, C. Theoharatos, V. Tsagaris, V. Anastasopoulos, and S. Costicoglou, “Vessel classification in COSMO-SkyMed SAR data using hierarchical feature selection,” *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XL-7/W3, pp. 975–982, 2015.
 - [18] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
 - [19] Y. Bengio, “Learning deep architectures for ai,” *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
 - [20] G. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
 - [21] Q. Lv, Y. Dou, X. Niu, J. Xu, J. Xu, and F. Xia, “Urban land use and land cover classification using remotely sensed SAR data through deep belief networks,” *J. Sensors*, vol. 501, 2015, Art. no. 538063.
 - [22] D. A. Morgan, “Deep convolutional neural networks for ATR from SAR imagery,” in *Proc. SPIE Defense+ Secur.*, 2015, paper 94 750F.
 - [23] C. Bentes, D. Velotto, and S. Lehner, “Target classification in oceanographic SAR images with deep neural networks: Architecture and initial results,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 1–4.
 - [24] C. Bentes, A. Frost, D. Velotto, and B. Tings, “Ship-iceberg discrimination with convolutional neural networks in high resolution SAR images,” in *Proc. 11th Eur. Conf. Synthetic Aperture Radar*, 2016, pp. 1–4.
 - [25] V. Fernandez Arguedas, D. Velotto, B. Tings, H. Greidanus, and C. A. Bentes da Silva, “Ship classification in high and very high resolution satellite SAR imagery,” in *Proc. Security Res. Conf., 11th Future Security*, Berlin, Germany, Sep. 13–14, 2016, pp. 347–354.
 - [26] J. Ding, B. Chen, H. Liu, and M. Huang, “Convolutional neural network with data augmentation for SAR target recognition,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 364–368, Mar. 2016.
 - [27] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, “Target classification using the deep convolutional networks for SAR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.
 - [28] M. Wilmanski, C. Kreucher, and J. Lauer, “Modern approaches in deep learning for SAR ATR,” in *Proc. SPIE Defense+ Security*, 2016, paper 98 430N.
 - [29] D. J. Crisp, “The state-of-the-art in ship detection in synthetic aperture radar imagery,” DTIC Doc., Tech. Rep. DSTO-RR-0272, 2004.
 - [30] B. Tings, C. A. Bentes da Silva, and S. Lehner, “Dynamically adapted ship parameter estimation using TerraSAR-X images,” *Int. J. Remote Sens.*, vol. 37, no. 9, pp. 1990–2015, 2016.
 - [31] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Netw.*, vol. 61, pp. 85–117, 2015.
 - [32] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” in *The Handbook of Brain Theory and Neural Networks*, vol. 3361. Cambridge, MA, USA: MIT Press, 1995, p. 1118.
 - [33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
 - [34] J. H. Ender, “On compressive sensing applied to radar,” *Signal Process.*, vol. 90, no. 5, pp. 1402–1414, 2010.
 - [35] X. X. Zhu, X.-M. Li, and R. Guo, “Compressive sensing for super-resolving SAR imaging to support target detection in coastal zone,” in *Proc. 10th Eur. Conf. Synthetic Aperture Radar*, 2014, pp. 1–4.
 - [36] F. Bastien *et al.*, “Theano: New features and speed improvements,” in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2012, pp. 1–10.
 - [37] J. Bergstra *et al.*, “Theano: A CPU and GPU math expression compiler,” in *Proc. Python Sci. Comput. Conf.*, Jun. 2010, oral presentation.
 - [38] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, “The numpy array: A structure for efficient numerical computation,” *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, 2011.
 - [39] F. Chollet, “Keras,” 2015. [Online]. Available: <https://github.com/fchollet/keras>
 - [40] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
 - [41] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Newton, MA, USA: O’Reilly Media, 2008.
 - [42] J. Bridle, “Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters,” in *Advances in Neural Information Processing Systems 2*. San Mateo, CA, USA: Morgan Kaufmann, 1989, pp. 211–217.
 - [43] Y. Chen, E. Blasch, H. Chen, T. Qian, and G. Chen, “Experimental feature-based SAR ATR performance evaluation under different operational conditions,” in *Proc. SPIE Defense Security Symp.*, 2008, Paper 69 680F.
 - [44] L. Bottou, “Stochastic gradient descent tricks,” in *Neural Networks: Tricks of the Trade*. New York, NY, USA: Springer-Verlag, 2012, pp. 421–436.



Carlos Bentes (S’13–M’17) was born in Volta Redonda, Brazil, in 1983. He received the B.S.E.E. and M.S.E.E. degrees from the Instituto Tecnológico de Aeronáutica, So Jos dos Campos, Brazil, in 2007 and 2012, respectively. Since 2015, he has been working toward the Ph.D. degree at the Technical University of Munich, Munich, Germany, in the field of remote sensing.

He is a Senior Data Scientist at the Software Technology and Application Center, Tartu, Estonia. His professional interests and research areas encompass artificial intelligence, machine learning, and remote sensing applications.



Domenico Velotto (S'08–M'15) was born in Italy in 1981. He received the M.Sc. degree (five-year legal course of study) in nautical science (curriculum electronic radio navigation and Earth observation systems) from the Università degli Studi di Napoli Parthenope, Naples, Italy, in 2008 and the Dr. Ing. degree in civil, geo and environmental engineering from the Technical University of Munich (TUM), Munich, Germany, in 2016.

He was a Ph.D. student/Young Research Scientist with the synthetic aperture radar oceanography team of the Remote Sensing Technology Institute, German Aerospace Center (DLR), Wessling, Germany, from March 2009 to February 2013. Since March 2013, he has been a Research Engineer at the Maritime Safety and Security Lab, Signal Processing Department (IMF-SAR), DLR, Bremen, Germany. His main research interests include electromagnetic models, Earth observation with SAR polarimetry, image processing, signal processing, machine learning, and data fusion toward maritime surveillance applications.

Dr. Velotto received the 2008 IEEE GRS South Italy Chapter Best Remote Sensing Thesis Award.



Björn Tings was born in Aachen, Germany, in 1987. He received the B.S. degree in scientific programming in combination with the simultaneous qualification in mathematical-technical software development from RWTH Aachen University, Aachen, Germany, in 2010 and the M.S. degree in artificial intelligence from Maastricht University, Maastricht, The Netherlands. He is currently working toward the Ph.D. degree at Helmut Schmidt University, Hamburg, Germany, where he investigates the automatic recognition of ship wake signatures.

Since 2013, he has been a Research Associate in the team of Synthetic Aperture Radar (SAR) Oceanography at the Remote Sensing Technology Institute of German Aerospace Center (DLR), Bremen, Germany. As a part of the team, his research involves the automatic detection and classification of ship signatures on SAR.