Ship Classification from Multi-Spectral Satellite Imaging by Convolutional Neural Networks

Raffaele Grasso
NATO STO CMRE, La Spezia, Italy
raffaele.grasso@cmre.nato.int

***Abstract*—This work investigates the use of convolutional neural networks for classifying ship targets from images acquired by the Multi-Spectral Instrument sensor on board Sentinel-2 satellites. An automatic procedure, requiring a minimum amount of supervision, is applied to extract labeled target images which are used for training. The data set consists of top of the atmosphere reflectance images in three visible channels and one near-infrared band. The performance of the classifier is evaluated by the receiver operating characteristic curve and the area under the curve statistics. The results show good classification performance with area under the curve greater than 0.95. Future work will be focused on investigating the impact of image atmospheric corrections and on comparing with other methods.**

***Index Terms*—Machine Learning, Ship classification, Satellite imaging**

I. Introduction

In the context of maritime situational awareness (MSA), automatic ship target detection and classification from space-borne multi-spectral images is still an open problem. The authors of a recent literature review on the topic [1], analyzing more than 100 papers in the period from 1978 to 2017, found several criticalities and propose some guidelines for researchers aimed at studying new methods. In particular, most of the reviewed studies using supervised machine learning techniques, are based on small training data sets, cover a limited range of acquisition conditions, and omit suitable quantitative performance assessment making the comparison among methods impossible. Moreover, a fraction of the authors use preprocessed data from public databases with degraded spatial, spectral and radiometric information. The lack of use of multi-modal target data sets in terms of sensor diversity, multi-temporal acquisitions and multi-spectral channels is an additional limitation of these studies. Furthermore, no rigorous studies were conducted to evaluate the impact that the sensor characteristics, the sea state, the atmosphere, the ship kinematic state, and the ship reflectivity properties have on the detection and classification performance.

Taking into account the future research guidelines in [1], this paper aims at evaluating the effectiveness and the flexibility of convolutional neural networks (CNN) classifiers [2] [3] on a well refined training data set, trying to overcome some of the above drawbacks. A data set of labeled ship targets that is representative for a wide range of environmental conditions, viewing geometries and ship types has been collected by automatically processing Sentinel 2 multi-spectral instrument (MSI) data. An automatic target detection algorithm [4] is used to extract target images which are associated to automatic identification system (AIS) data in order to extract multi-spectral ship signatures with attached AIS text attributes and kinematic information. The data are not atmospherically corrected, so the impact of the atmosphere on the classification performance is not investigated in this paper.

The data set is used to asses the performance of CNN classifiers which are able to directly process an input image without the preliminary step of extracting image features, such as in support vector machines (SVMs) which are the state of the art in satellite multi-spectral image classification, as reported in [1]. A CNN is able to jointly perform feature extraction and classification by a series of multi-dimensional and multi-scale spatial filter banks which are optimized by minimizing a loss function on the training set. The network can be also considered as a supervised method to extract image features which can be used later by a different classification algorithm.

Several examples of classifiers will be provided such as a basic ship/non-ship binary classifier and a classifier to discriminate ships navigating at a speed greater than a given threshold. Classifier performance is evaluated by the receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC) statistics which are computed on training, validation and test sets so as to estimate the behavior of the system in case of unforeseen data and check the presence of over-fitting. In general, the performance achieved by the classifiers is good, with estimated AUC greater than 0.95.

The paper is organized as follows. Section II describes AIS and MSI sensors, the procedure for data labeling and the data set used to train the classifiers, while section III briefly introduces ship classification and CNNs. Section IV provides the results, showing the performance metrics, while section V draws the conclusion and suggests future research directions.

II. Sensors and data sets

This work makes use of data from two sources: the AIS and the MSI. Data from the two systems are fused to automatically extract training data sets by using the procedure detailed below.

A. The AIS

The AIS [5] is a self-reporting system, used by vessel traffic services, that allows vessels to broadcast their identification code (the Maritime Mobile Service Identity (MMSI) number),

978-9-0827-9703-9/19/$31.00

characteristics (e.g. ship type, size, navigational status), position, speed, course over ground, destination and estimated time of arrival. AIS messages are periodically broadcast and they are received by other vessels equipped with AIS transceivers, as well as by ground stations and satellite platforms. Historical data bases of AIS messages are available for vessel traffic analysis and are exploited in this work to implement the labeling procedure.

B. The MSI

The MSI is a pushbroom multi-spectral passive imaging sensor on board the European Space Agency (ESA) Sentinel 2 satellite constellation used for Earth observation [6]. The constellation consists of two polar orbiting satellites, 2A and 2B, placed on the same orbit and phased at $180°$ to each other. The revisit time with two satellites is 10 days at the equator and 2 to 3 days at mid-latitude. The latitude coverage is between $56°$S and $84°$N and the swath width is $290Km$.

The MSI sensor acquires the solar radiation from the Earth surface and the atmosphere in 13 spectral bands, four bands at spatial resolution $\Delta R = 10m$, six bands at $\Delta R = 20m$ and three bands at $\Delta R = 60m$, with a radiometric resolution of $12bits$. The channels at $\Delta R = 10m$ are the most interesting ones for ship classification. Table I [7] shows the characteristics of these channels for the MSI on board Sentinel 2A and 2B, including the central wavelength, λ_0, the spectral resolution, $\Delta\lambda$, the calibration reference radiance, L_{ref}, and the signal to noise ratio at the reference radiance, $SNR@L_{ref}$. The first 3 channels cover the visible spectrum in the blue, green and red bands, respectively, while the fourth channel covers a band in the near infrared (NIR) range. The data

TABLE I
SENTINEL 2 MSI SENSOR CHANNELS

Band number	**S2A**		**S2B**		$\frac{L_{ref}}{(Wm^{-2}sr^{-1}\mu m^{-1})}$	$SNR@L_{ref}$
	$\lambda_0(nm)$	$\Delta\lambda(nm)$	$\lambda_0(nm)$	$\Delta\lambda(nm)$		
2	492.4	98	492.1	98	128	154
3	559.8	45	559	46	128	168
4	664.6	38	664.9	39	108	142
8	832.8	145	832.9	133	103	174

at $10m$ resolution are provided as image tiles (or granules) on a geo-located uniform grid of 10800 by 10800 samples for each spectral channel and are freely available through the Copernicus Science Hub (CSH) web site [8].

C. The labeling procedure

The CSH provides an application program interface (API) to search and download historical MSI images. Given a list of vessels of interest (VoI), identified by the MMSI codes, their AIS historical tracks in a given temporal window (e.g. 2 years) are used to define the spatial boundary of the search in the CSH. The series of images retrieved from the CSH are expected to contain the VoI as well as other ships.

The data set of labeled ship targets is built by associating AIS tracks to image contacts obtained by an automatic object detector. The detector is based on mathematical morphology non-linear spatial filters and it is described in [4]. The detection is performed on the NIR band to reduce the interference of the signal from the sea water body. A sub-image of 46 by 46 samples is extracted around the center of mass (CoM) of the detected object, for each channel at $\Delta R = 10m$. The CoM Universal Transverse Mercator (UTM) (x, y) position is then associated to an AIS track using a minimum distance criterion with a threshold on the maximum distance equal to $300m$. The sub-images are then labeled by using the data in the AIS message of the associated tracks. The information associated to each ship contact include the MMSI, the ship type and size, the speed over ground (SOG), the course over ground (COG), the position of the ship at the image acquisition time (linearly interpolated), the navigational status and the ship draught. A final visual inspection is performed to refine the data set and label by a "no-ship" flag the residual false alarms due to association errors.

The final dataset consists of about 6000 ship targets and about 2000 non-ship objects. Figure 1 shows on the left the type of ship targets while on the right the navigational status flag. The data set is representative for targets of type "Cargo" and "Tanker" with about 3000 and 2000 contacts per type, respectively. These two classes are broad and a finer categorization is possible as shown in the next sections. As far as the navigational status concerns, ships under way using engine are roughly 3800 while approximately 2000 ships are observed at anchor. Figure 2 shows the distribution of target

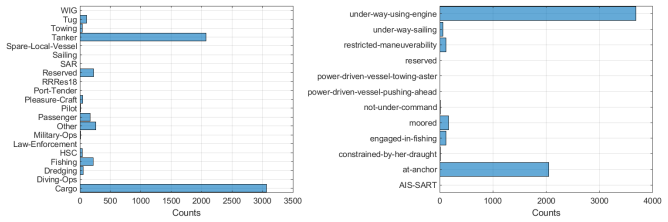


Fig. 1. Ship type and navigational status of targets detected from the Sentinel 2 MSI sensor and associated to AIS tracks

length, and width. The data set is representative for a wide range of ship lengths ranging from few meters to $400m$. Figure 3 displays the spatial distribution of the associated ship

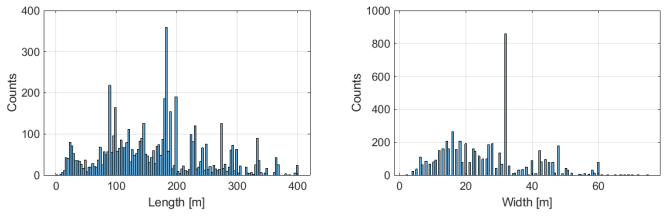


Fig. 2. Ship parameters of targets detected from the Sentinel 2 MSI sensor and associated to AIS tracks

contacts at global level as well as over sub-regions where the contacts are distributed along typical sea lanes.

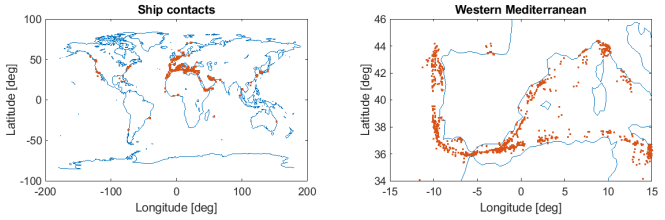


Fig. 3. Spatial distribution of targets detected from the Sentinel 2 MSI sensor and associated to AIS tracks

III. Ship classification

A. Classification tasks and training data sets

In order to show the flexibility of the proposed approach, a set of example classifiers are implemented by training a CNN on the labeled target signatures described in section II. The tasks performed by the considered classifiers include:

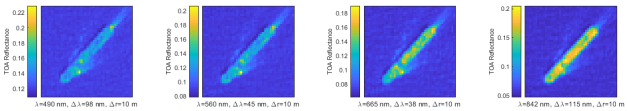
- a binary ship/no-ship low level classifier,
- a binary low-speed classifier,
- a 4 class COG quadrant classifier and
- a 4 class ship type classifier.

The ship/no-ship classifier is a low level classifier that is used to discriminate between ship targets and non-ship targets like clouds. The other ones are applied afterward to understand some additional properties of the target, acquiring a rough idea of the navigational status (an information that is correlated to the presence of a ship wake) and the identity.

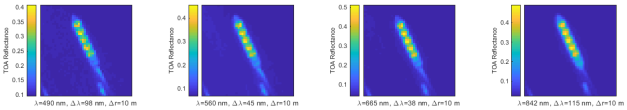
Specific training sets are built for each task. Concerning the low level classifier, 2000 "no-ship" targets and 2000 "ship" targets were extracted from the main data set. Then 50% of the samples of each class were assigned to the training set, 25% to the validation set and 25% to the test set. Regarding the low speed classifier, a data set of ship targets longer than $200m$ was first extracted. This set was split in two classes of 1100 samples for each class by using a $1Knot$ threshold on the SOG. Finally, 70% of the samples were assigned to the training set, 15% to the validation set and the remaining 15% to the test set. For the COG quadrant classification task, 250 ship targets for each quadrant (Q1 = Northern-Eastern, Q2 = Northern-Western, Q3 = Southern-Western and Q4 = South-Eastern quadrants) were picked at random. Then 60% of the samples were assigned to the training set, 20% to the validation set and the other 20% to the test set. The data set for ship type classification was extracted by searching for targets in a given list of vessels. In particular, four groups of ships are taken into account:

- a fleet of cargo ships of a given shipping company,
- a group of liquid natural gas (LNG) tankers of the spherical Moss type,
- a fleet of roll-on/roll-off (RORO) vehicle carrier ships and
- a set of very large crude carriers (VLCC) and ultra large crude carriers (ULCC) oil tankers.

Ships in each group share similar structural characteristics and spectral signatures (see Fig. 4). The size of each group is of 75, 33, 99 and 104 target images, respectively. The 30% of the targets were assigned to the training set, 35% to the validation set and the other 35% to the test set. The training set is finally augmented as described in section III-C to reduce the effects of small sample over-fitting phenomena. Due to the small size of this data set the results obtained on the ship type classifier are not definitive. Further investigations will be conducted in the future by collecting additional samples.



(a) Cargo



(b) LNG Moss-Spherical

Fig. 4. Ship target spectral signatures from the Sentinel 2 MSI sensor. For each ship type and from left to right, the figure displays the target in the blue, green, red and NIR channels.

B. Overview of CNNs

A deep neural network can be defined as a series of nested input-output function layers where the i^{th} layer processes the output $\boldsymbol{y}_{i-1}$ from the previous one by applying an affine transformation and a non-linear function as follows [3]:

$$\boldsymbol{y}_i = \boldsymbol{g}(\boldsymbol{W}_i \boldsymbol{y}_{i-1} + \boldsymbol{b}_i) \qquad i = 1, \ldots, L, \tag{1}$$

where $\boldsymbol{y}_0$ is the input data vector while $\boldsymbol{y}_L$ is the output one; $\boldsymbol{W}_i$ is a weight matrix, $\boldsymbol{b}_i$ is a bias vector and $\boldsymbol{g}$ is a non linear activation function, such as the rectified linear unit (ReLU), $g(z_{ij}) = max\{0, z_{ij}\}$, applied to the $j-th$ component of the transformed vector $\boldsymbol{z}_i = \boldsymbol{W}_i \boldsymbol{y}_{i-1} + \boldsymbol{b}_i$.

A CNN [2] [3] is a network that jointly optimizes image feature extraction and classification. The network consists of a number of convolutional layers (CL), each layer processes the inputs from the previous layer by a bank of multi-dimensional linear filters followed by bias summation, non-linear activations (e.g. ReLUs) and a pooling layer to down-sample filter outputs by a given rate in order to calculate multi-scale features and simplify the complexity of the network. Being the convolution a linear operator, a CNN layer can be considered equivalent to (1). The convolutional front-end acts as a multi-scale non-linear feature extraction system, similarly to a linear wavelet transform filter bank [9], which learns features at progressively larger scales isolating details of the input image which are semantically meaningful [10].

The outputs of the final CL are lexicographically ordered in a vector and successively processed by a series of fully connected layers (FCL), or dense layers (DL), each followed by a bias summation layer and ReLU activations as well. Typically, the last two layers of a classifier include a DL with a number of outputs equal to the number of classes, followed by a soft-max layer to provide a probability distribution vector

over the classes. The output class is decided, for instance, by applying the max rule on the estimated probabilities.

Some of the layers in the network can be followed by a dropout layer which implements a regularization technique to reduce over-fitting phenomena. The technique consists in multiplying by zero a subset of the previous layer activation outputs, with a given probability [11].

The parameters of the network, $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$, for each layer, are learned by minimizing a given loss function over a set of input-image/output-label pairs. In this work, the adaptive moments (ADAM) stochastic gradient algorithm is chosen as optimizer [12], with a constant learning rate equal to 10^{-3} and a maximum number of epochs between 50 and 200. In order to speed up the training phase, the loss gradient is evaluated at each iteration on a minibatch [3] of training samples of size 100. The loss to be optimized is the cross entropy function for Kmutually exclusive classes which is defined as:

$$H = \sum_{i=1}^{N} \sum_{j=1}^{K} c_{ij} ln(p_{ij}), \tag{2}$$

where N is the number of input/output training samples, c_{ij} is 1 if the i^{th} sample belong to the j^{th} class, 0 otherwise, and p_{ij} is the soft-max (probability) output of the i^{th} sample for the j^{th} class. Alternatively, a weighted cross entropy loss can be considered.

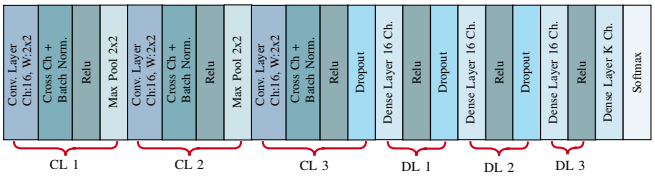


Fig. 5. CNN structure.

In this work, instead of using a pre-trained deep CNN, such as the AlexNet [13], and applying the technique of transfer learning [13] [14] to speed up the training phase, a CNN that can be trained from scratch in a reasonable amount of time (less than 12 hours), even without using a GPU, is preferred in order to test the performance that a simpler not pre-trained network can achieve on the considered data set. Figure 5 shows the architecture of the network used in this work. This is characterized by 3 convolutional layers with 16 filters of size 2×2. The first two CLs are followed by normalization, ReLU activation and max pooling layers, while the last one has a dropout layer in place of the max pooling one. The dropout layer is followed by three DLs with 16 outputs. The first two DLs are followed by a ReLU and dropout. The third is followed by a ReLU that feeds the input of the final classifier DL and a softmax layer. The performance of this basic structure is improved by additional layers (not displayed in Fig. 5) that transfer higher resolution features to the output by using additional DLs and a combination node. The network hyper-parameters (e.g. the size of the convolutional linear filters and the dropout probability) have been set heuristically by a trial and error procedure. Optimization of hyper-parameters in a more rigorous way (e.g. using cross validation) will be investigated in a future work on suitable hardware.

C. Mitigation of over-fitting due to small data sets

Over-fitting due to small sample is further reduced by augmenting the training sets. The augmentation procedure consists in applying random geometric and pixel-wise value transformations to the training images to produce new samples. In this work, the image transformations include:

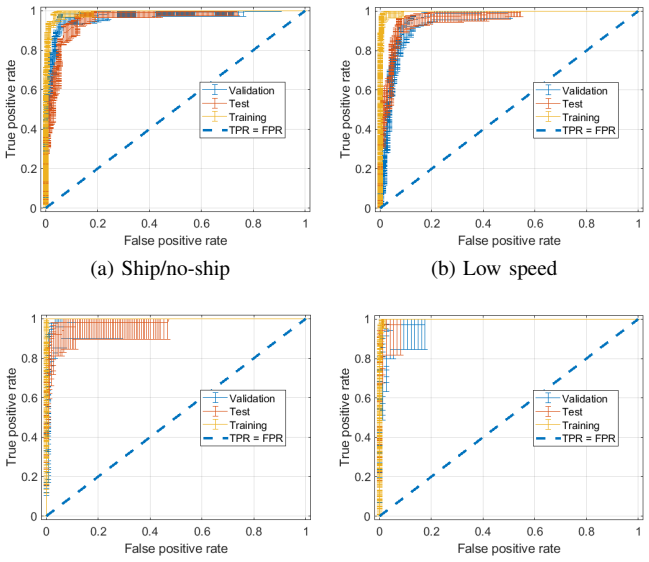
- random translation of a maximum amount of 3 pixels in both spatial directions,
- random spatial scaling with a scale factor between 0.8 and 1.2 on both directions,
- random rotations between 0 and 30 deg (not applied in the case of COG classifier)
- and random scaling of pixel values with a factor between 0.8 and 1.2.

D. Performance evaluation

Binary classifier performance is evaluated by ROC curves which display the classifier true positive rate (TPR) versus the false positive rate (FPR), for several values of the decision threshold applied to the network score of the positive class. For multi-class problems the ROC curve is estimated by taking one class as positive while the remaining ones are joined in a single negative class. The classification score used to estimate the ROC curve in this case is re-computed as $m_i = p_i - max_{j \neq i}(p_j)$, where $i, j = 1, \ldots, K$, i is the index of the chosen positive class, and p_i is the score at the output of the classifier for the i^{th} class. A bootstrap method is then used to estimate ROC curves and AUC confidence intervals allowing testing if the trained classifier statistically outperforms the $TPR = FPR$ classifier with $AUC = 0.5$.

IV. Results

This section shows the performance of low-level and high-level CNN classifiers, which were trained using the data sets described in section III-A. Table II summarize the AUC statistics of the four classifiers, i.e. the AUC point estimation and AUC 95% confidence intervals (CI). Figure 6a shows the ROC curves and the associated 95% CIs of the ship/no-ship low level classifier estimated for the training, validation and test sets. The estimated 95% CI of the AUC is $(0.954, 0.986)$ for the validation set, $(0.940, 0.976)$ for the test set and $(0.985, 0.998)$ for the training set, with AUC point estimation of 0.975, 0.962 and 0.995, respectively. Figure 6b shows the same curves for the low speed classifier. The estimated 95% CI of the AUC is $(0.914, 0.969)$ for the validation set, $(0.940, 0.978)$ for the test set and $(0.995, 0.999)$ for the training set, with AUC point estimation of 0.952, 0.962 and 0.998, respectively. The ROC curves of the 4-class COG quadrant classifier for the three data sets are displayed in Fig. 6c. The graph is relative to the Q1 class taken as positive while the remaining ones are considered as a single negative class as described in section III-D. The estimated 95% CI



(a) Ship/no-ship (b) Low speed

(c) COG quadrant (d) Ship type

Fig. 6. Classifer ROC curves

of the AUC in this case is $(0.945, 0.996)$ for the validation set, $(0.932, 0.992)$ for the test set and $(0.997, 0.999)$ for the training set, with AUC point estimation of 0.983, 0.974 and 0.998, respectively. Similar results were obtained taking the Q2, Q3 and Q4 quadrants as positive classes. Finally, the performance of the 4-class ship-type classifier is displayed in Fig. 6d for the "Cargo" class taken as positive. The estimated 95% CI of the AUC is $(0.956, 0.996)$ for the validation set, $(0.984, 1)$ for the test set and $(0.994, 0.999)$ for the training set, with AUC point estimation of 0.986, 0.998 and 0.997, respectively. Similar performance is achieved when the other classes are considered as positive. In general, the performance of the proposed CNN structure is good with an estimated AUC greater than 0.95 in all the cases taken into account.

TABLE II
SUMMARY OF AUC STATISTICS

	Training set	**Validation set**	**Test set**
	AUC / 95% CI	**AUC / 95% CI**	**AUC / 95% CI**
Ship/No-ship	0.995 / (0.985, 0.998)	0.975 / (0.954, 0.986)	0.962 / (0.940, 0.976)
Low speed	0.998 / (0.995, 0.999)	0.952 / (0.914, 0.969)	0.962 / (0.940, 0.978)
COG Quadrant	0.998 / (0.997, 0.999)	0.983 / (0.945, 0.996)	0.974 / (0.932, 0.992)
Ship type	0.997 / (0.994, 0.999)	0.986 / (0.956, 0.996)	0.998 / (0.984, 1)

V. CONCLUSION

This paper investigates the performance of CNNs to classify ship targets from the MSI sensor on board the Sentinel 2 satellite. The study is based on a training data set collected by labeling target images by an automatic procedure requiring a minimum amount of supervision. The classifiers considered in the paper as examples show good performance. In particular the ship/non-ship classifier is characterized by an AUC of 0.975 with 95% confidence levels of 0.954 and 0.986 on the validation set. Performance of the same order was observed in high level classifiers and in particular in the classifier for discriminating fleets of sister vessels. This one requires to be further investigated by improving the training set.

In general, CNNs are a flexible and effective method to classify ship targets from space-borne multi-spectral images and it is worth to further investigate this technique according to [1]. In particular, future work will be focused on studying the effects of the atmosphere on classification performance and on augmenting training data sets by using models of the atmospheric radiance to simulate physically observable atmospheric image signals. Moreover, comparison with non-deep learning methods, such as SVMs, will be investigated and a deeper analysis of classification errors will be performed.

VI. ACKNOWLEDGMENT

This work has been funded by the NATO ACT project *Data Knowledge Operational Effectiveness*.

The training data sets contain modified Copernicus Sentinel data from 2016 to 2017.

REFERENCES

[1] U. Kanjir, H. Greidanus, and K. Oštir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote Sensing of Environment*, vol. 207, pp. 1 – 26, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425717306193

[2] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*, 2010, pp. 253–256.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[4] R. Grasso, S. Mirra, A. Baldacci, J. Horstmann, M. Coffin, and M. Jarvis, "Performance assessment of a mathematical morphology ship detection algorithm for sar images through comparison with ais data," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, Nov 2009, pp. 602–607.

[5] ITU, "Technical characteristics for an automatic identification system using time division multiple access in the vhf maritime mobile band," International Telecommunications Union, Tech. Rep. ITU-R M.1371-5, 02 2014.

[6] "ESA-Sentinel-2 mission website," https://sentinel.esa.int/web/sentinel/missions/sentinel-2, accessed: 2018-12-07.

[7] "ESA-Sentinel-2 mission website," https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric, accessed: 2018-12-07.

[8] "Copernicus Science Hub website," https://scihub.copernicus.eu, accessed: 2018-12-07.

[9] S. Mallat, "Understanding deep convolutional networks," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 374 2065, p. 20150203, 2016.

[10] Q. Zhang, Y. Nian Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[12] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[13] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, M. Hasan, B. C. V. Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," *CoRR*, vol. abs/1803.01164, 2018. [Online]. Available: http://arxiv.org/abs/1803.01164

[14] Q. Yang and S. J. Pan, "A survey on transfer learning," *IEEE Transactions on Knowledge & Data Engineering*, vol. 22, pp. 1345–1359, 10 2009. [Online]. Available: doi.ieeecomputersociety.org/10.1109/TKDE.2009.191