Classification of Ships Using Real and Simulated Data in a Convolutional Neural Network

Nina Ødegaard
and Atle Onar Knapskog
Norwegian Defence Research Establishment (FFI)
PB 25
2027 Kjeller
Norway
Nina.Odegaard@ffi.no
Atle-Onar.Knapskog@ffi.no

Christian Cochin
and Jean-Christophe Louvigne
DGA Maîtrise de l'information
BP 7
35998 Rennes CEDEX 9
France
Christian.cochin@intradef.gouv.fr
Jean-christophe.louvigne@intradef.gouv.fr

***Abstract*—Convolutional neural networks (CNNs) have recently been applied successfully in large scale image classification competitions for photographs found on the Internet. As our brains are able to recognize objects in the images, there must be some regularities in the data that a neural network can utilize. These regularities are difficult to find an explicit set of rules for. However, by using a CNN and the backpropagation algorithm for learning, the neural network can learn to pick up on the features in the images that are characteristic for each class. Also, data regularities that are not visually obvious to us can be learned. CNNs are particularly useful for classifying data containing some spatial structure, like photographs and speech. In this paper, the technique is tested on SAR images of ships in harbour. The tests indicate that CNNs are promising methods for discriminating between targets in SAR images. However, the false alarm rate is quite high when introducing confusers in the tests. A big challenge in the development of target classification algorithms, especially in the case of SAR, is the lack of real data. This paper also describes tests using simulated SAR images of the same target classes as the real data in order to fill this data gap. The simulated images are made with the MOCEM software (developed by DGA), based on CAD models of the targets. The tests performed here indicate that simulated data can indeed be helpful in training a convolutional neural network to classify real SAR images.**

I. Introduction

Target classification in SAR images is still an ongoing research topic, and many algorithms have been tested on various data sets. Often the input data to the classifier is a collection of some form of handcrafted features. In the general machine learning community, there has recently been renewed interest in a subclass of pattern recognition methods called deep learning. One advantage of these methods is that the algorithm can figure out for itself what the useful information in the data is, as opposed to earlier methods where the features to be used had to be manually chosen. The abundance of labelled data plus the increase in computation power can be accredited for the renewed interest in this topic. These methods have shown very good results in classification of objects in photographs. They have also been applied successfully, both alone and in combination with other classifiers, to SAR images from the MSTAR dataset [1], [2], [3]. In this paper, CNNs are tested on a data set from the PicoSAR radar. The data set contains a collection of small ships in Oslo harbour, imaged in X band.

The paper is divided into two main parts. The paper first describes more specifically the application of CNNs to the data set and shows that the method can often separate the targets in the test without the need for any handcrafted features. When confusers are introduced, however, the false alarm rate is quite high. Then simulated SAR images made with the MOCEM software are added to the training data, and it is shown that classification by CNN can benefit from this. The aim of this paper is not to build a complete ATR system, or to say that CNNs are the solution to the ATR problem, but rather to see if CNNs can successfully separate different target classes without manually chosen features. Therefore no attention is being paid to methods for automatic segmentation, azimuth angle estimation, possible target occlusion etc. The trials done here indicate that CNNs and also deep learning in general may be a way forward in order to eliminate the feature extraction step in the ATR chain, and also that simulated images may fill the data gap one often faces.

II. Deep learning and Convolutional Neural Networks

Our brains can recognize an abundance of different objects in photographs, but to come up with an explicit set of rules to describe the objects that can be implemented in a computer (in terms of the pixel magnitudes), has shown to be very hard. Deep learning, in which the algorithm tries to resemble processes that take place in the human brain, have recently received renewed interest. This is due to the increase in available labelled data and the increase in computational power. Also, [4] showed in 2006 that it is possible to efficiently train neural networks deeper than just a few layers, something that was earlier not shown to be successful. It is argued in [5] that classic classifiers such as support vector machines, nearest neighbour, neural nets with one or two hidden layers etc. are in fact often too shallow, and therefore do not possess enough expressive power to separate the classes unless the

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/RADAR.2016.7485270

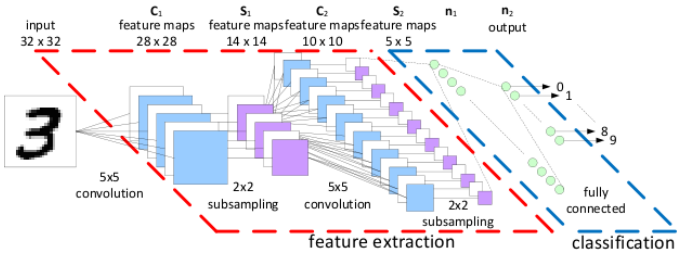

Fig. 1. Illustration of an example layout of a CNN, proposed by [7] for handwritten digits recognition

Fig. 2. The different ships used in this study

features themselves are already so good that they have more or less "done the job". A deep architecture on the other hand, is said to have the ability to learn highly nonlinear regularities in the data with hardly any preprocessing. A CNN is one example of a deep architecture, provided it has enough layers. CNNs are highly suitable to data containing some spatial structure, like photographs and speech. CNNs are said to be reminiscent of simple and complex cells in the primary visual cortex [6]. They have been quite successful in classifying handwritten digits [7]. Also, in the yearly held large scale image classification competition ImageNet more and more of the competing research groups are using CNNs. In 2014, all the groups in the top 9 ranking was using some version of CNNs. CNNs mark an exception to the earlier limitation of backpropagation training, in that it was possible to train them using many layers also earlier than the work in [4]. It is argued that this is caused by the pooling of weights, which in effect limits the number of parameters in the structure. A limited number of parameters in turn does not lead to exploding or vanishing gradients in the backpropagation algorithm, which has appeared to be the problem for other kinds of deep neural networks [8], [9]. It also limits the difference often observed between training and test error, and it keeps the computation time down. One of the motivations for testing a CNN for classification in SAR images is to skip the step of feature selection. Most classification algorithms rely on some kind of manually selected features for input. This step is often a very time consuming part of the design of an ATR system. It is also perhaps the most critical step for the performance of feature based methods, in that only features that can actually separate between the classes are useful. It has been shown that if the features are good, the type of classifier used later on is not that important [10], [11]. Several dimension reduction methods (like PCA, ICA etc.) can be applied to eliminate features that are not contributing to the separation of the classes, however, a list of potential features have to be made manually to start from. One other method that does not rely on manual feature extraction is direct image correlation, but this method has shown to have its limitations.

Figure 1 shows an example layout of a CNN. This particular layout was proposed by [7] to be used in classifying handwritten digits from the MNIST data set. The net used in the experiments in this paper differs a bit from the one in the illustration. The list below shows the parameters used for each layer in the net:

1) Input layer: 304 x 84
2) Convolutional layer: 6 x 5 x 5
3) Subsampling layer: 6 x 2 x 2
4) Convolutional layer: 16 x 5 x 5
5) Subsampling layer: 16 x 2 x 2
6) Convolutional layer: 120 x 18 x 74
7) Fully connected: 84 x 1
8) Output layer: 2/6

The error function to be minimized during training is the Mean Squared Error (MSE). Further, the backpropagation method used here is the stochastic gradient descent using Levenberg-Marquardt with no calculation of the Hessian. The choice of number of layers, number of nodes, backpropagtion method and other parameters may seem a bit arbitrary. There are, however, some heuristics on the subject. [12] has made an overview of best practices. Many other parameter settings could have been tried in this study, but limited time prevents the trial of all possible parameter combinations.

III. Data set

A. Radar data

The data set used in this experiment comes from the radar PicoSAR and was collected by FFI over Oslo harbour in several campaigns between 2009 and 2012. The radar was purchased by FFI from Selex in 2007. It operates in X band and was installed on board a helicopter during the collections. The actual targets used in this paper were not the primary focus of the collections, and the data set is therefore not complete, i.e. it does not cover all intervals in azimuth and elevation angle as one would wish for ATR development. Figure 2 shows the different ships in this study. The data set consists of more than 850 SAR images of these ships, plus some random patches in the images to be used as confusers. Figure 3 shows the azimuth and elevation angles of the available SAR images.

B. Preprocessing

The ships are found in a complex harbour environment, and automatic segementation of the targets is difficult. We have

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/RADAR.2016.7485270

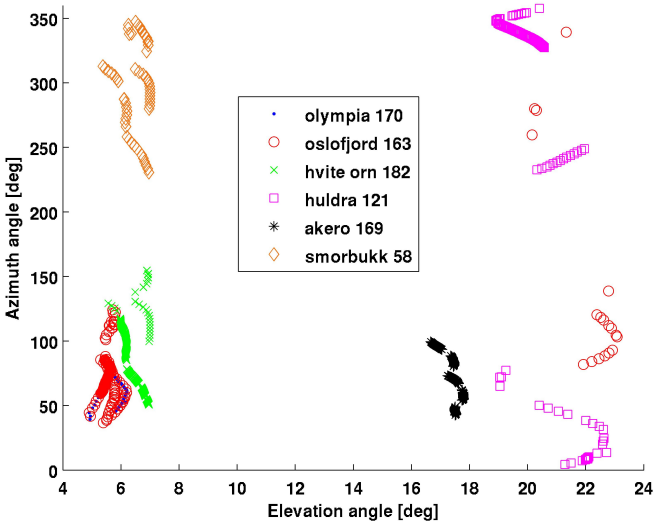

Fig. 3. Available manually segmented SAR images used in this study. Legend denotes number of images. Azimuth angle is defined as 0 from port, 90 from aft etc.

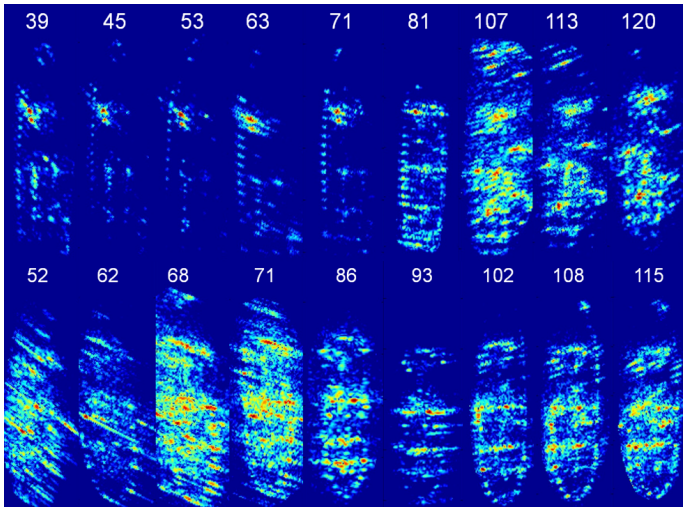

Fig. 4. Examples of SAR images used in this study. Azimuth angle in degrees noted on top. Top row: Oslofjord. Bottom row: Hvite Ørn.

therefore manually segmented out the ships from the images. There are a few uncertainties in this data set:

- Multipath and overlay in the images may be caused by the other structures at the pier. Therefore too much or too little is sometimes included in the manual cut.
- As the ships are not perfectly stationary at the pier, some defocus may occur. The segmented images are therefore autofocused using PGA before they are used in the net. The autofocusing does not always work, but it ensures that major unwanted regularities in the different classes are removed.
- Also there is some uncertainty as to the exact orientation of the targets. The ships are assumed to be perfectly aligned with the pier when the azimuth angle is calculated, but we know that the ships are often a bit skewed. The images are rotated to have the bow upwards before classification.

Figure 4 shows examples of some of the SAR images in the study. The azimuth angle is noted on top.

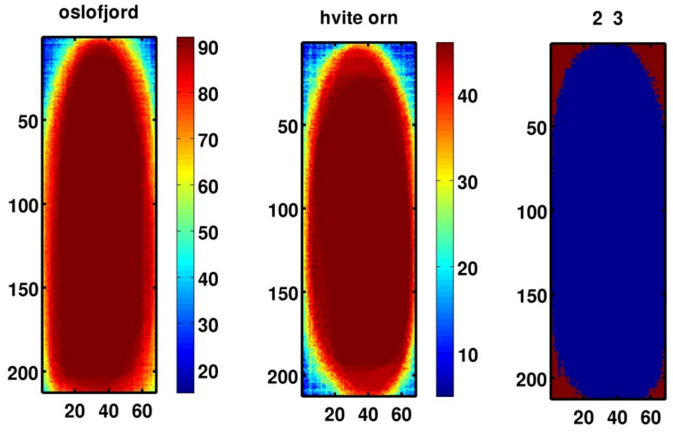

Fig. 5. Example of how the masks for the different classes are created. The colour coding in the first two pictures are the number of images that cover the region. The last picture shows the resulting mask when the coverage of the mean of the number of images for the two ships of the same length are used.

IV. Classification results

A. Preparations

To prevent the net from utilizing regularities in the images that are not really a part of the data, such as differences between classes in the manual segmentation, masks are placed over the images. This approach will also make sure that the net cannot utilize it if one of the ships is more skewed at the pier than the rest, and therefore has some parts of the image not occupied by any data. In that case, data in the same pixels in the other images will also be removed. Common masks are made for all classes having the same length. The reason why one mask is not used for all the classes, is that we think that the length of the ship is a feature that will be used in any classification scheme, and so it is no use in removing this information. Also, by using a mask that will fit only the smallest ship on all the images, a lot of useful information about the larger ships will be removed. One could argue that other methods for creating the mask would be better, and perhaps be more realistic for an operational system. The generation of such a mask is illustrated in figure 5. Occlusion is not taken into account in these tests. Images where less than about 75 % of the ship is visible, are discarded. The pixel values are the log intensity values, and no processing have been done to them except normalising each image to have mean 0 and standard deviation 1, which is standard procedure in neural networks.

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/RADAR.2016.7485270

Trial #	Ships	Azimuth	Decim	# runs	Min MCR
1	OF/HO	0 - 360	1	22	0.05
2	HU/AK	0 - 360	1	101	0.0
3	OF/HO	40 - 80	1	17	0.0
4	OF/HO	40 - 80	2	16	0.03
5	OF/HO	40 - 80	3	20	0.06
6	OF/HO	40 - 80	4	19	1.0
7	OF/HO	100 - 170	1	16	0.27
8	All	0 - 360	1	9	0.02

TABLE I
VARIOUS CLASSIFICATION RESULTS FOR CNN. THE DECISIONS ARE FORCED. MCR = MISCLASSIFICATION RATE. OF = OSLOFJORD. HO = HVITE ØRN. AK = AKERØ. HU = HULDRA.

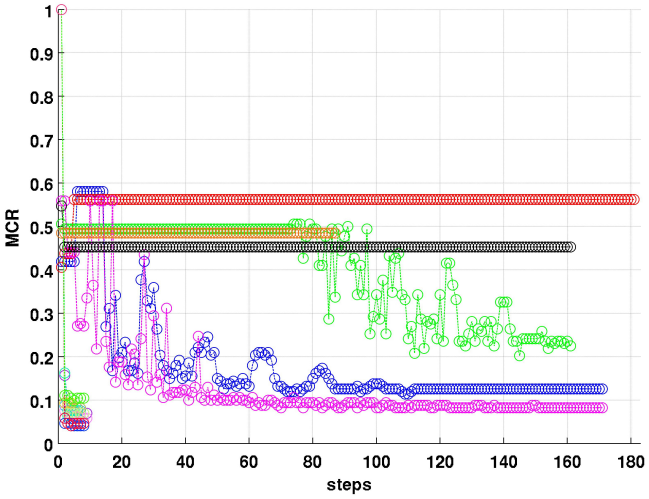

Fig. 6. Behaviour of many training runs using the same parameter set

B. Results

As the error function to be minimized in the backprojection weight learning algorithm can be multi modal, the training process often gets stuck in local minima. There are several tricks reported in the literature to lower the chances of this happening [12], many of which have not been tested in this study. The training will still have to be run several times in order to observe its behaviour. The initial weights, the actual test and training samples, plus the order in which the training samples are presented to the net are randomized for each training run. The results from many trials with different parameters can be found in table I. First, only ships with the same length are trained by the net. The pairs Oslofjord/Hvite Ørn and Akerø/Huldra are thus tested independently. This was done because when the ships have different lengths, it seems likely that the net will utilize the extra data in the longer ships to separate them from the ships of other lengths.

The difficult trials will therefore be between the ones of the same size, and they are also using the same mask, which forces the net to only utilize the actual values in the same pixel positions. The results of these trials are promising, showing a misclassification rate (MCR) of 0 for Huldra/Akerø when using all the data (test no 2). This may be because the two classes have a limited amount of overlapping azimuth angles, and are different enough not to be confused with each other. Figure 6 shows an example of how MCR can develop for different training runs using the same parameter set. We can see that some of the runs seem to get stuck in local minima, one is stuck for a while, but then manages to get out, and some converge to a good value. None of the runs made it to 0 MCR in this example, though. The MCR is only measured on the test set. Oslofjord/Hvite Ørn does not reach an MCR of 0 when all of the data is used. This may have been solved if we had more data, that could capture all the variation in the class. Instead we try using a finer sectioning in azimuth angle. One would expect that samples from a smaller azimuth angle interval are more similar to each other than samples collected in a wider interval. It is of course a lower limit to the size of these intervals, when the amount of data in them is too small for the net to converge to a good solution. The results when limiting the azimuth angle intervals to 40 - 80 degrees also correspond to these theories (trials no 3 - 6). In these trials, an increasing number in the column called “Decim” (= decimation) means that more and more data is removed from the training set. See table II for an overview of the number of samples used in the various trials. In the other trials, decimation = 1 means that the data set is divided evenly (but randomly) into test and training set. We see that as the azimuth angle interval is limited, the MCR reaches 0, but as we remove more and more data, the performance decreases. In trial no 7, we only have 13 training samples for Oslofjord, and the performance is bad here also. In those cases when we don’t have enough data, simulations may be the solution. [13] suggests generating some distorted versions of the real data in order to increase the training set for handwritten digits recognition. Whether or not that method could work on SAR data is not tested here. Finally the net was trained on data from all the ships, trial no 8 in table I. The confusion matrix corresponding to forced decicion is shown in table III. We see that all the confusion is between Oslofjord and Hvite Ørn. If different masks had not been applied according to length, there may have been more confusion between the other classes also. In an operational system where we have uncertainty about the length of the ship, this result is probably too optimistic.

C. Unknown targets

So far in this paper we have only tried to show that CNNs are capable of finding useful features for separating ships in SAR images, without the need for handcrafted features. However, in an operational system, a classifier has to respond correctly to unknown targets, i.e. targets the net has not been trained to recognize. The desired response for a robust system would be to declare these confusers as “unknown”. The full SAR images contain harbour structures, as well as ships and sea. Random patches were extracted from the same set of SAR images as the segmented images came from. Also the ship classes that the net was not trained on were presented to it as unknown targets. More specifically, a handful of the best nets resulting from trials no 2, 3 and 8 in table I were

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/RADAR.2016.7485270

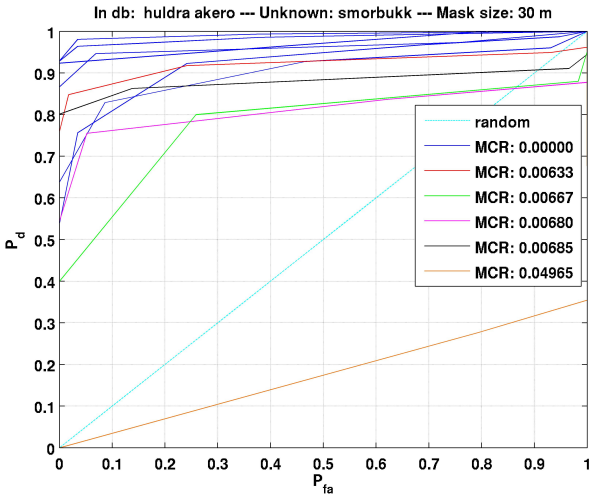

Fig. 7. ROC curves for the unknown target Smørbukk presented to nets only trained on Huldra and Akerø

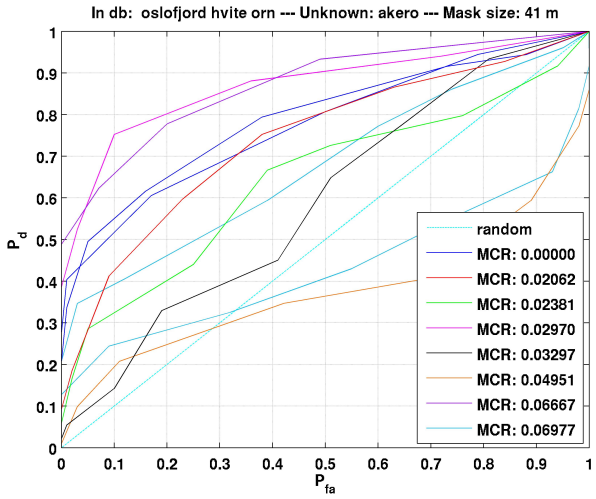

Fig. 8. ROC curves for the unknown target Akerø presented to nets only trained on Oslofjord and Hvite Ørn

Decim	# train OF	# test OF	# train HO	# test HO
2	37	66	32	54
3	24	79	21	65
4	12	91	10	76
5	0	103	0	86

TABLE II
NUMBER OF SAMPLES FOR TEST AND TRAINING SET IN THE TRIALS WITH AZIMUTH ANGLE INTERVAL 40 - 80 DEG.

Class	OL	OF	HO	HU	AK	SB	Unknown
OL	85	0	0	0	0	0	0
OF	0	73	8	0	0	0	0
HO	0	2	89	0	0	0	0
HU	0	0	0	60	0	0	0
AK	0	0	0	0	84	0	0
SB	0	0	0	0	0	29	0

TABLE III
CONFUSION MATRIX FOR TRIAL 8 IN TABLE I, CORRESPONDING TO FORCED DECICION. OL = OLYMPIA, SB = SMØRBUKK.

presented with unknown samples. The unknown samples were also masked in the same way as the samples used in the training of the nets. All combinations of unknown classes and mask sizes were tested, to see if a pattern would emerge. The resulting ROC curves show that the false alarm rate is often very high, meaning that the classifier is not robust. However, for some combinations of unknown target and mask size, the false alarm rate is quite good. It seems that generally all unknown classes have the lowest false alarm rate when a mask size of 30 m is used, but even when all parameters are fixed, different nets have very different false alarm rates. Different nets with the same MCR can also have substantially different ROC curves, and it is not always the net with the lowest MCR that has the lowest false alarm rate. No substantial difference in false alarm rate was observed for the random patches than for the unknown ship targets. Figures 7 and 8 show ROC curves for some of the trials, supporting these observations. The rate of correct classification, however, is always high here on the samples that are detected, as shown by the low MCR. The lack of robustness, and the effect of masks on the false alarm rate, is something that must be studied further.

V. SIMULATED SAR IMAGES

Access to real SAR data is always limited. It is therefore interesting to see if simulations can be used as training data together with real data in order to fill the data gaps, especially if the azimuth angle intervals of the net are decreased.

A. *MOCEM software*

The MOCEM software is a tool for simulating SAR images rapidly. It was intended for ATR and training of image analysts. It is developed by Alyotech under a DGA contract [14], [15]. Instead of trying to make a very accurate RCS prediction, the code has a phenomenological approach based on Geometrical Optics and Physical Optics in the last facet. Based on the geometry of and the materials chosen on the CAD model, major scattering mechanisms are located in 3D. The scattering is projected onto slant range to produce an ideal image. Finally a SAR point spread function is applied in correspondence with the radar parameters chosen to produce the final SAR image. The different materials on the CAD model are given separate electromagnetic properties. The backscattered energy of a given point on the target is calculated as a sum of specular and diffuse phenomena. The ratio of these varies depending on the values set for dielectric constant, roughness and the σ^0 curves selected. The CAD models used in this paper has been built in Rhinoceros, based on photographs and line drawings found on the Internet. Figure 9 shows the two CAD models used in this study.

B. *Simulated SAR images*

Some examples of the resulting simulations can be seen in figure 10. Not much time was spent on tuning the simulations

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/RADAR.2016.7485270

Fig. 9. CAD models used in the simulations. Left: Hvite Ørn. Right: Oslofjord.

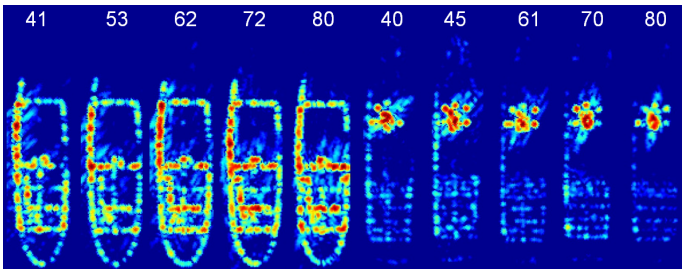

Fig. 10. Simulations made in MOCEM, azimuth angles in degrees noted on top. Left: Hvite Ørn. Right: Oslofjord.

to the real data. We can see that they visually resemble the real data, but there are also large differences. The classification trials will show if we have been able to capture some structure that the net can utilize. When introducing more classes to the test, it will probably be more important to get a better correspondence between the simulations and the real data. In this case, with only two classes to test, it is enough if the simulations of one class are more similar to the real data from that class than from the other, and vice versa. However, the similarity has to be one that is useful to the net.

VI. Classification results using simulations

The simulations are added only to the training set. The test set consists purely of real data. We gradually increase the ratio of simulated to real data in the training set. The number of simulations is always 26 for each class. The simulated data are only tested for Hvite Ørn/Oslofjord in the azimuth angle interval 40 - 80 degrees to see if they can help the drop in performance seen when the number of samples is too low. Table IV shows the results for the various trials.

The results are promising, and it seems that the addition of simulations can indeed compensate for the drop in performance when the number of samples is low. It seems that the simulations have some of the same regularities as the real data that the net can utilize. If only simulations are in the training set, however, the performance is bad. It may be that some real data are needed to guide the selection of features. When more classes are to be resolved, much more care will probably have to be taken in the simulation step. Alternatively, one could plot the activation kernels of the various layers in the CNN in order to see what features it picks up, as done in [2], and focus the simulation tuning to these areas.

Trial #	Ships	Azimuth	% sim	Decim	# runs	Min MCR
9	OF/HO	40 - 80	43	2	22	0.02
10	OF/HO	40 - 80	54	3	24	0.02
11	OF/HO	40 - 80	70	4	19	0.08
12	OF/HO	40 - 80	100	5	17	0.35

TABLE IV
Classification results using simulated data in a CNN

VII. Conclusion

The trials performed in this paper indicate that deeper neural networks may be a way forward in the ATR field. At least the trials show that the feature extraction step can be handled by algorithms. The set of real SAR data used here is very limited. It does not contain enough classes of the same size. However, similar statements about the use of CNNs have been made by others using the MSTAR set. The trials also indicate that simulated images may be used to fill the data gap often experienced. However, when confusers are introduced, the false alarm rate can be quite high. More extensive studies are thus required, both on the use of CNNs itself, and on the addition of simulations to the training set.

References

[1] S. Wagner, "Combination of convolutional feature extraction and support vector machines for radar ATR," *IEEE 17th International Conference on Information Fusion (FUSION)*, 2014.

[2] D. A. E. Morgan, "Deep convolutional neural networks for ATR from SAR imagery," *Proceedings of SPIE Algorithms for Synthetic Aperture Radar XXII*, 2015.

[3] S. Chen, "SAR target recognition based on deep learning," *International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 541–547, 2014.

[4] G. E. Hinton, S. Osindero, and Y.-W. The, "A fast learning algorithm for deep belief nets," *Neural Computation*, no. 18, pp. 1527–1554, 2006.

[5] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[6] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *Journal of Physiology*, vol. 148, pp. 574–591, 1959.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, pp. 1–46, 1998.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[9] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[10] J. Schiller, "Capabilities, developments and challenges in non-cooperative target identification using radar," *Microwaves, Radar and Remote Sensing Symposium*, pp. 24–27, 2011.

[11] J. Schiller, "Comparing classifier effectiveness," *NATO STO SET-172 Lecture Series on Radar Automatic Target Recognition (ATR) and Non-Cooperative Target Recognition (NCTR)*, 2013.

[12] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, "Efficient backprop," *Neural Networks: Tricks of the trade*, 1998.

[13] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional networks applied to visual document analysis," *Seventh International Conference on Document Analysis and Recognition*, pp. 958–963, 2003.

[14] C. Cochin, P. Pouliguen, B. Delahaye, D. Le Hellard, P. Gosselin, and F. Aubineau, "MOCEM - fast generator of 3D scatterers for radar simulation," *International Radar Conference*, 2009.

[15] C. Cochin, J. C. Louvigne, R. Fabbri, C. Le Barbu, and L. Ferro-Famil, "MOCEM V4 - radar simulation of ship at sea for SAR and ISAR applications," *European Conference on Synthetic Aperture Radar*, 2014.

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1109/RADAR.2016.7485270