2016 23rd International Conference on Pattern Recognition (ICPR)

Underwater Target Classification in Synthetic Aperture Sonar Imagery Using Deep Convolutional Neural Networks

David P. Williams

***Abstract*—Deep convolutional neural networks are used to perform underwater target classification in synthetic aperture sonar (SAS) imagery. The deep networks are learned using a massive database of real, measured sonar data collected at sea during different expeditions in various geographical locations. A novel training procedure is developed specially for the data from this new sensor modality in order to augment the amount of training data available for learning and to avoid overfitting. The deep networks learned are employed for several binary classification tasks in which different classes of objects in real sonar data are to be discriminated. The proposed deep approach consistently achieves superior performance to a traditional feature-based classifier that we had relied on previously.**

I. Introduction

Underwater mines and unexploded ordnance are a nefarious legacy of past conflicts that today pose threats to civilian enterprises in waters around the globe. One principal objective of mine countermeasures (MCM) operations is the detection of suspicious objects on the seafloor and the classification of each as an object of interest (*e.g.*, a mine) or not. Increasingly, the mine searches executed for these tasks are conducted with an autonomous underwater vehicle (AUV) that is equipped with sonar sensors capable of producing very high, centimeter-resolution acoustic imagery of the seafloor. In this paper, sonar data collected at sea is used in conjunction with a deep convolutional neural network to perform underwater object classification.

Many different feature-based classification approaches have been employed for underwater target classification [1]–[3], with these traditional methods all relying on special, hand-crafted features tailored to and developed for the particular task in question. This means that the clues useful for discriminating between two classes of objects are necessarily predetermined, limited by the human subject-matter-expert designing the features. However, deep convolutional neural networks [4] automatically learn filters, which play the role of features in a sense, as part of the training process. As a result, a deep neural network has the freedom to uncover more useful characteristics ostensibly "hidden" in the data.

Another drawback of classification approaches based on traditional "shallow" architectures, such as support vector machines [5], is that the methods eventually hit a performance plateau beyond which the addition of more training data cannot improve. In contrast, networks with deep architectures characterized by a nested functional structure that engenders highly nonlinear decision surfaces can continue to improve their performance as more training data is made available. The cumulative nature of MCM data-gathering operations, in which the total body of sonar data collected continues to increase with each new expedition at sea, makes deep approaches a natural fit.

The great capacity of deep neural networks, when paired with vast amounts of data and sufficient computational resources, can translate into excellent classification success, as evidenced by the state-of-the-art performance achieved in diverse domains from image recognition [6] and cancer screening [7] to the board game Go [8]. Our main contribution is a new application of deep convolutional neural networks to data from a novel sensor modality, namely sonar.

The remainder of this paper is organized as follows. Sec. II describes the deep convolutional neural network architecture and training procedure developed for use with sonar imagery. Experimental results for three different binary classification tasks are presented in Sec. III. Concluding remarks and directions for future research are provided in Sec. IV.

II. Deep Convolutional Neural Networks for Sonar Imagery

The basic algorithmic machinery of deep convolutional neural networks that we use [9] is already well-established, so more focus is devoted here to the data preparation procedures and network architecture design that we have developed specifically for classification tasks with sonar imagery.

A. Data Preparation

Synthetic aperture sonar (SAS) works by coherently summing received acoustic signals of overlapping elements in an array, and it provides an order-of-magnitude improvement in resolution over simple (real aperture) side-scan sonar data [10]. The resulting high-resolution SAS imagery provides a detailed view of the seafloor that makes detection of proud targets (*e.g.*, mines) possible.

All sonar data used in this work was collected at sea by CMRE's MUSCLE AUV. This experimental, state-of-the-art AUV is a 21-inch diameter vehicle from Bluefin that is equipped with a SAS system developed by Thales. The center frequency of the SAS is 300 kHz, and the bandwidth is 60 kHz. The system enables the formation of high-resolution sonar imagery with a theoretical along-track resolution of 2.5 cm, and a theoretical range resolution of 1.5 cm, usually out to a range of 150 m.

A typical "scene-level" SAS image, from which objects would be detected and classified, is shown in Fig. 1. Over 14 million pixels comprise the image, whose pixel values

The author is with NATO STO Centre for Maritime Research and Experimentation (CMRE), 19126 La Spezia (SP), Italy. david.williams@cmre.nato.int

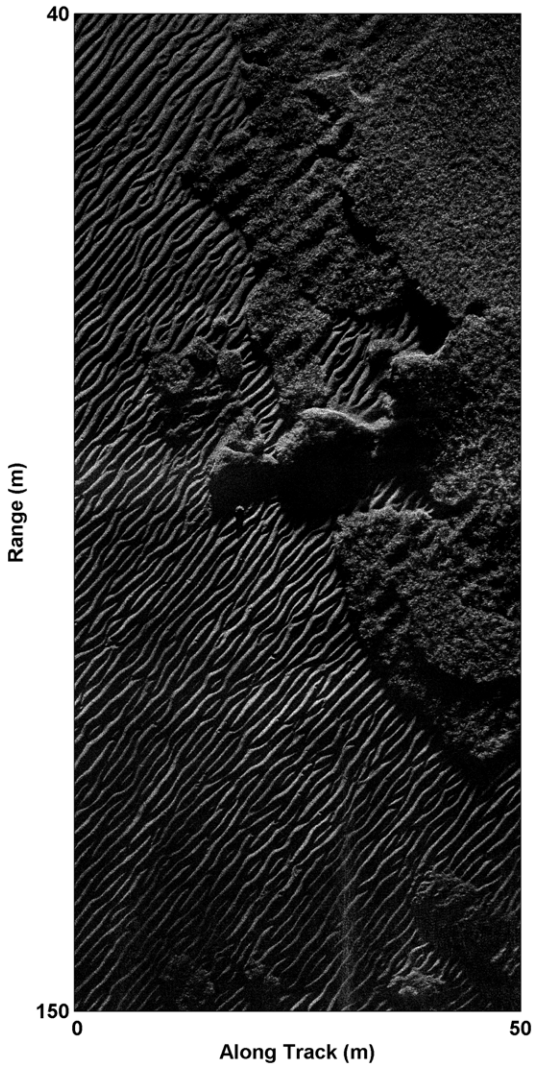


Fig. 1. A typical wide-area, "scene-level" SAS image characterized by sand ripples, vegetation, and rocks.

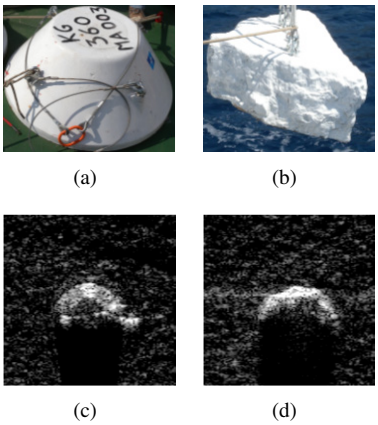


Fig. 2. Photographs of (a) a truncated cone and (b) a calibrated rock, and corresponding SAS "mugshots" of (c) the truncated cone and (d) the calibrated rock. Each SAS mugshot covers an area of 2.505 m $\times$ 2.505 m; the x-axis is along track, the y-axis is range.

correspond to the intensity of the acoustic signal returns at each position in the scene. The default area covered by a pixel in these images is 1.5 cm in the range direction and 2.5 cm in the along-track direction. Bilinear interpolation was used to convert the image pixels into squares covering 1.5 cm in each direction. The scene-level SAS images were also normalized such that all pixel values were in $[0, 1]$.

For this work, a cascaded, integral-image-based detection algorithm [11] is applied to scene-level SAS images like this in order to generate a set of potential objects of interest that will then be examined further in a subsequent classification stage. For each alarm from the detection stage, a "mugshot" of the object is extracted from the scene-level SAS image. Specifically, a 2.505 m $\times$ 2.505 m SAS mugshot, centered around the detection location, is extracted. This size was carefully chosen based on the expected sizes of objects of interest. Importantly, the mugshot size was chosen to be larger than the object sizes so that the full object would be contained within the mugshot. Moreover, since it is known, based on the geometry of the data-collection process, that an object proud of the seafloor will cast an acoustic shadow behind (*i.e.,* further in range) the object, the mugshot size was made large enough to capture some of this shadow information as well; shadow shape is often exploited in hand-crafted features.

Example SAS mugshots of two objects considered in the experiments conducted later are shown, along with their photographs, in Fig. 2. The mugshot images, each 167 pixels $\times$ 167 pixels, are the inputs to the deep convolutional neural networks. A tradeoff exists with the sizes of the (mugshot) imagery: larger images mean more contextual information is retained, but at the expense of more intensive subsequent processing with the deep networks.

B. Network Architecture

Three binary classification experiments were considered in this work, the difference among them being which objects were treated as belonging to each class. However, the same deep network architecture was used in all experiments. Specifically, a 10-layer convolutional neural network was designed, with this consisting of alternating layers of convolution and pooling operations, followed by a fully-connected layer, and a final fully-connected output layer. The inputs to a given layer are the outputs from the preceding layer, which creates a nested functional structure. The inputs to the initial layer are the SAS mugshots themselves. The outputs of the final layer are the probabilities of a mugshot belonging to each class.

There were four sets of convolution layers and pooling layers. Each convolutional layer and fully-connected layer used a sigmoid activation function, $\sigma(x) = (1 + e^{-x})^{-1}$. Each pooling layer, which effectively downsamples, used pure averaging rather than the commonly used max-pooling approach because the former should be more robust when dealing with the speckle-like nature of sonar imagery; the averaging was performed always using patch sizes (in pixels) of 2×2 (*i.e.,* downsampling by a factor of 2 in each direction). Each convolutional layer is associated with a fixed number

of filters (*i.e.,* kernels) of predefined size. The number of filters in the four convolutional layers were 6, 8, 10, and 24, respectively. The sizes (in pixels) of the filters in the four convolutional layers were 32×32, 17×17, 7×7, and 5×5, respectively. A stride of 1 was used for all convolutional and pooling layers. The input to the fully-connected layer is a 216-dimensional feature vector, formed from vectorizing and concatenating the previous layer's output maps.

C. Network Training

The training process of the deep network learns the parameters of the model, which for the convolutional layers are the filters and associated bias terms. (There are no parameters associated with the pooling layers.) The model seeks to minimize the standard classification error on the training data under consideration. However, the amount of SAS data available for training is much too large to hold in computer memory at once, so a novel procedure is developed for training. At each training iteration, the following procedure is executed to create a batch of 50 mugshots for updating (improving) the network's parameters.

One scene-level image is randomly selected from the database of available training data images that are known to contain at least one object from class 0. Then 25 mugshots for class 0, each 2.505 m $\times$ 2.505 m (167 pixels $\times$ 167 pixels), are extracted from the scene-level image. The centers of these mugshots are randomly selected from the pixels associated with class 0 objects, which are defined as pixels within 0.51 m (34 pixels) of the objects' center pixels (from the detection stage). Since there are far fewer than 25 objects (of interest) in a given scene-level image, it is guaranteed that multiple mugshots in the batch will correspond to the same object (but differing by small translations). This effect is desired in order to improve the translation-invariance of the deep network, which is important in case the object locations from the detection stage are not perfectly centered. To further increase robustness, each mugshot is also flipped about the range axis – *i.e.,* "mirrored" – with a probability of 0.5. Mirroring mugshots in this manner respects the unique geometry and physics of the target-sensor relationship; an arbitrary rotation of a mugshot would not. This whole process is then repeated for class 1 mugshots, *mutatis mutandis*. The set of 50 mugshots, 25 from each class, is then used to update the network parameters via gradient descent. At the subsequent training iteration, this entire procedure is repeated using different (randomly selected) scene-level images. For the experiments shown here, training consisted of 10^5 such iterations.

Since this work is, to our knowledge, the first to consider using deep convolutional neural networks for SAS data, a few lessons learned experimentally are worth noting for the sake of fellow researchers. It has been found to be extremely important that the distribution of the two classes in each batch is kept balanced. We also believe the network avoids overfitting as a result of randomly selecting a new batch of training data at each iteration. The mirroring and small translations are also thought to improve robustness and avoid overfitting. For deciding whether the network has been trained sufficiently, useful alternatives to using a validation set of data are to examine the filters (and verify that they are generally smooth) and to evaluate a set of mugshots created from mirroring and minor translations (and verify that the predictions' variation is small).

III. Experimental Results

Eleven major sea expeditions have been conducted by CMRE using the MUSCLE AUV from 2008 through 2015 in various geographical locations – Baltic Sea, Mediterranean Sea, North Sea – that have different seafloor characteristics (mud, sand, ripples, vegetation, rocks, *etc.*). In each expedition, specific objects were deployed on the seafloor before conducting sonar surveys over the area with the AUV, so accurate ground-truth information is possessed. The objects deployed included dummy mine shapes, more realistic mine-like targets, other man-made objects, and calibrated rocks. The resulting raw sonar data that were collected were processed into scene-level SAS images. Table I summarizes the number of scene-level SAS images obtained in each sea expedition, and how each data set was exploited for the ensuing classification experiments here. Collectively, the images in the data sets cover over 350 km^2 of seafloor. (Many of the scene-level images do not contain any deployed objects.)

TABLE I
Data Set Details

Data Set	Sea Expedition		How Data Used	Number of Scene-Level Images
	Year	Location		
COL2	2008	Latvia	Training	8941
CAT1	2009	Italy	Training	2858
CAT2	2009	Italy	Training	3084
AMI1	2010	Italy	Training	3103
ARI1	2011	Italy	Training	8951
ARI2	2012	Italy	Training	5596
SPM1	2013	Spain	Training	5686
MAN1	2013	Italy	Training	6516
MAN2	2014	Italy	Testing	6162
NSM1	2015	Belgium	Testing	4109
TJM1	2015	Spain	Testing	10039

We consider three binary classification experiments, the difference being which objects are considered to belong to each class. Experiment A seeks to discriminate targets (class 1) from non-targets (class 0). The target class consists of cylinders, truncated cones, and wedge-shaped objects, all of which mimic common mine types. The non-target class consists of specially calibrated rocks and various man-made objects (whose size and shape are similar to targets) including a washing machine, a cylindrical diving bottle, and a weighted duffel bag, among others. Experiment B seeks to discriminate truncated cones (class 1) from the calibrated rocks (class 0); the latter class of objects resembles the former from certain target-sensor aspects. Experiment C seeks to discriminate the mine type known as mantas (class 1) from truncated cones (class 0); optically, these two classes of objects are very similar.

For each experiment, a unique deep convolutional neural network is learned in isolation. (Alternatively, a single network can be learned and then kept fixed, with additional parameter learning allowed only for the fully connected layers, but we wish to observe the model differences learned in the different experiments.) All experiments draw their training data from the same set of 8 sea expeditions, and evaluate using test data from the 3 other sea expeditions, as specified in Table I. After applying the aforementioned detection algorithm, the number of mugshots in each *test* set that belong to each class in each experiment are summarized in Table II. (The numbers of objects detected in the training sets are comparable.)

TABLE II
NUMBER OF MUGSHOT IMAGES IN TEST SETS

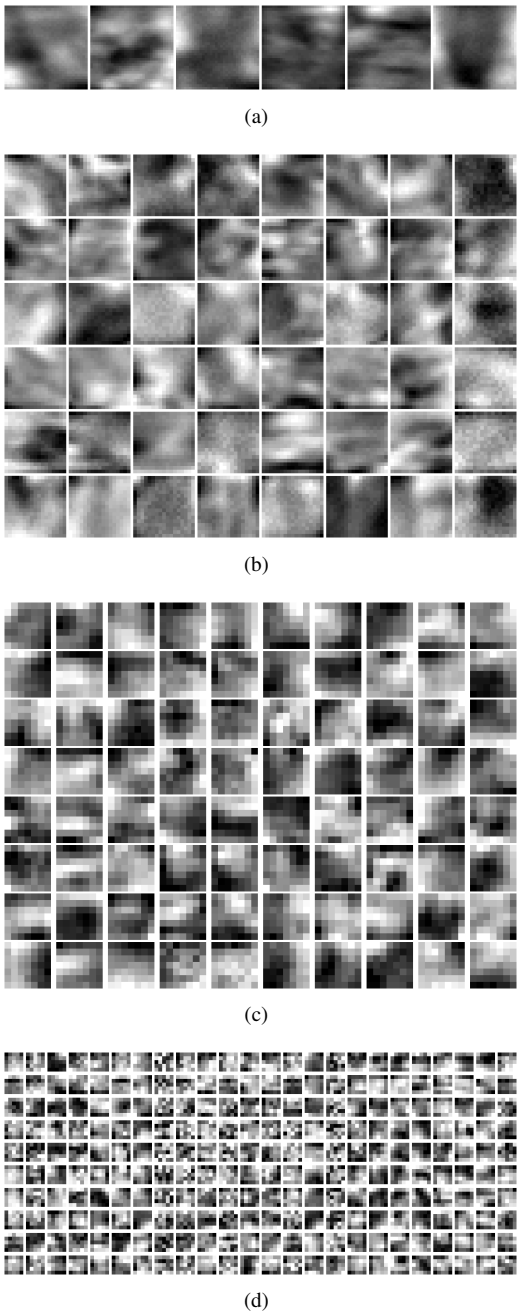
Data Set	Experiment A		Experiment B		Experiment C	
	Class 1	Class 0	Class 1	Class 0	Class 1	Class 0
MAN2	207	118	148	61	79	148
NSM1	77	10	52	10	112	52
TJM1	316	36	195	36	73	195

In each experiment, we compare three different classification approaches. Two approaches employ the learned deep convolutional neural network; these are denoted "Proposed" in subsequent figure legends. The first version uses the network as is; the second variant also mirrors each mugshot and takes the average of the network predictions (on the original and mirrored mugshot) as the final prediction. The third approach uses a modified version of a relevance vector machine [12], with the classifier parameters directly weighting a small set of traditional features [13] that have previously been found to characterize various attributes of the objects well. This method is denoted in subsequent figure legends as "Detection Score," and it represents our previous best-performing classifier for these tasks.

A. *Experiment* A*: Targets Versus Non-Targets*

The convolutional filters learned by the deep neural network in Experiment A are shown in Fig. 3. Due to space constraints, the convolutional filters learned by the deep network in Experiments B and C are not shown, but each set exhibits interesting differences owing to the different nature of the objects being considered. (Future work will examine in greater detail the elements of the objects that each filter is keying on by masking small regions of the mugshots as in [14].)

For Experiment A, the performance of the competing approaches on the three test data sets in terms of receiver operating characteristic (ROC) curves and area under the ROC curve (AUC) – a summary measure between 0 and 1 – are shown in Fig. 4. (A higher AUC is better.) As can be seen from Fig. 4, the proposed deep networks are far superior to the traditional feature-based approach. This can be directly attributed to the much greater capacity and complexity of the deep networks, whereas the feature-based approach is limited by its relatively tiny number of parameters.



(a) (b) (c) (d)

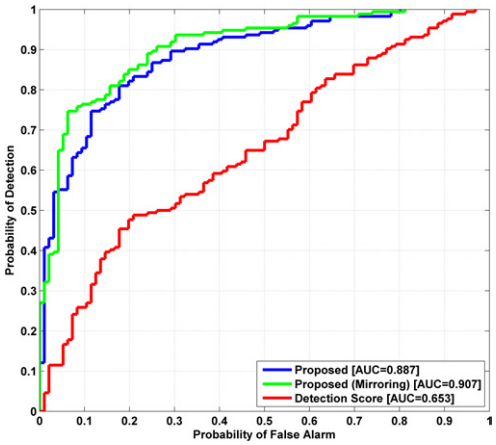
Fig. 3. Learned filters for the (a) first, (b) second, (c) third, and (d) fourth convolutional layers of the network in Experiment A.

B. *Experiment* B*: Truncated Cones Versus Rocks*

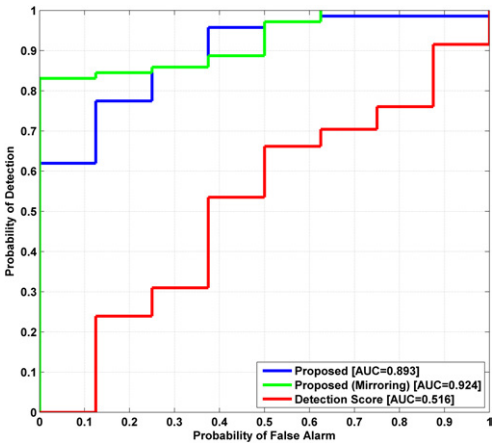
For Experiment B, the performance of the competing approaches on the three test data sets are shown in Fig. 5. Again, the deep networks perform much better than the traditional feature-based approach.

C. *Experiment* C*: Mantas Verus Truncated Cones*

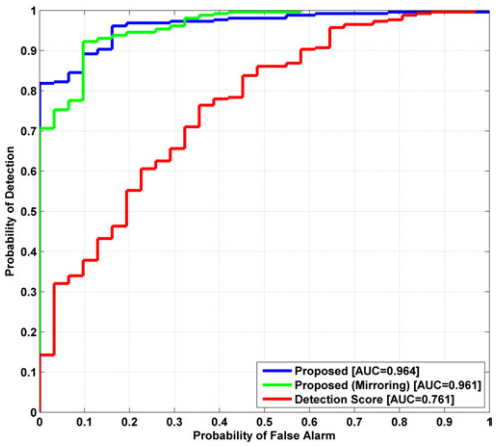
For Experiment C, the performance of the competing approaches on the three test data sets are shown in Fig. 6. As before, the deep networks are consistently superior to the traditional feature-based approach.



(a)

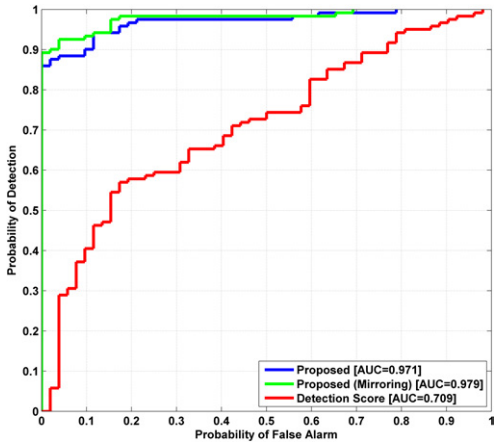


(b)

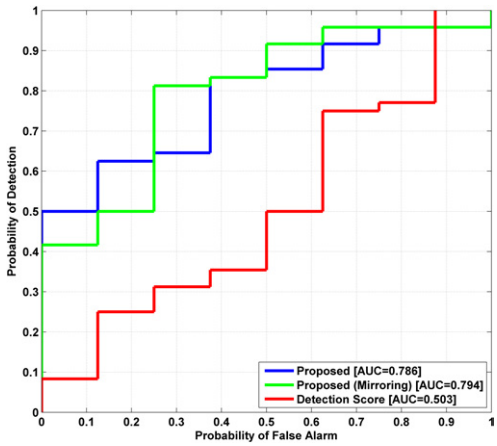


(c)

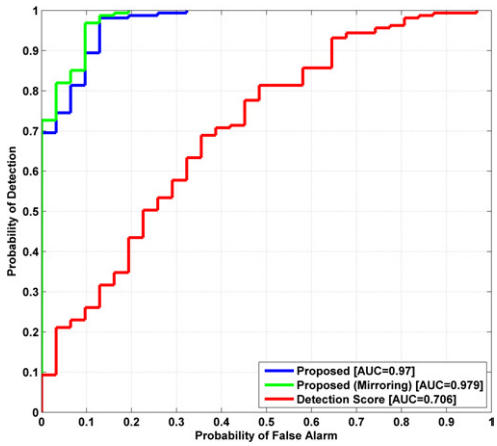
Fig. 4. Experiment A classification performance, discriminating targets from non-targets, on data from (a) MAN2, (b) NSM1, and (c) TJM1.



(a)



(b)

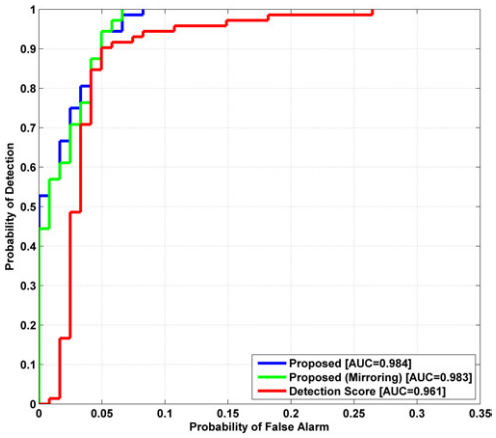


(c)

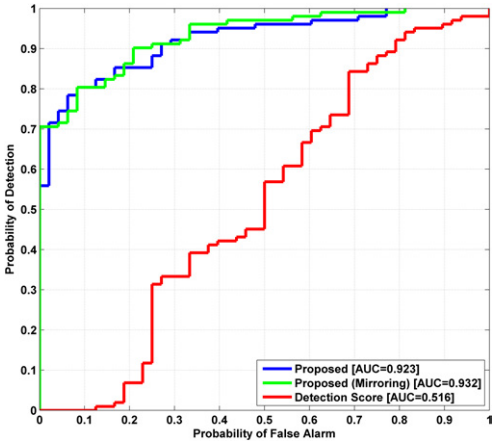
Fig. 5. Experiment B classification performance, discriminating truncated cones from calibrated rocks, on data from (a) MAN2, (b) NSM1, and (c) TJM1.

IV. Conclusion

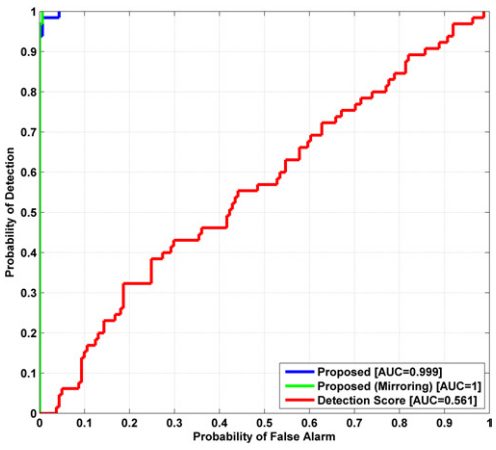
This work demonstrated the tremendous potential of using deep convolutional neural networks on sonar imagery for underwater target classification. A significant jump in performance was achieved over the traditional feature-based classification approach that we had relied on previously. Based on the classification results, it appears that overfitting did not occur despite the huge number of parameters in the deep networks. This can perhaps be attributed to the special training scheme developed, as well as the relatively small number of filters



(a)



(b)



(c)

Fig. 6. Experiment C classification performance, discriminating mantas from truncated cones, on data from (a) MAN2, (b) NSM1, and (c) TJM1.

used in each convolutional layer. Importantly, the proposed training procedure also augmented the amount of training data mugshots available for learning. It was demonstrated that the deep networks were capable of learning valuable differences between similar classes of objects automatically, which may subsequently enhance our understanding of the target-sensor phenomenology underwater.

There are several avenues for future research that will be, or are already being, pursued. Additional experiments that explore different deep network architectures – with different numbers of layers, filters, and filter sizes – are currently in progress. Other ongoing work is also considering the larger, more general classification problem that seeks to discriminate targets from all clutter objects (not only *deployed* clutter objects). The great diversity of naturally occurring clutter types found on the seafloor suggest that considerably longer training times will be required to learn a deep network that performs well on this task. Moreover, larger networks – both deeper and wider – may also be required, though preliminary results that we have already obtained for this challenging task are promising. Additional effort will be devoted to better understanding the object characteristics that the learned network filters are exploiting. An eventual goal is to employ deep networks for performing both detection and classification in a single unified stage on the scene-level SAS images, which should be feasible given the convolutional nature of the networks.

REFERENCES

[1] G. Dobeck, J. Hyland, and L. Smedley, "Automated detection/classification of seamines in sonar imagery," in *Proc. SPIE International Society of Optics*, vol. 3079, 1997, pp. 90–110.

[2] S. Reed, Y. Petillot, and J. Bell, "An automatic approach to the detection and extraction of mine features in sidescan sonar," *IEEE Journal of Oceanic Engineering*, vol. 28, no. 1, pp. 90–105, 2003.

[3] E. Coiras, P. Mignotte, Y. Petillot, J. Bell, and K. Lebart, "Supervised target detection and classification by training on augmented reality data," *IET Radar Sonar Navigation*, vol. 1, no. 1, pp. 83–90, 2007.

[4] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[5] B. Scholkopf and A. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[7] D. Cireşan, A. Giusti, L. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *Medical Image Computing and Computer-Assisted Intervention*, 2013, pp. 411–418.

[8] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[9] R. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Master's thesis, Technical University of Denmark, Lyngby, Denmark, 2012.

[10] M. Hayes and P. Gough, "Broad-band synthetic aperture sonar," *IEEE Journal of Oceanic Engineering*, vol. 17, no. 1, pp. 80–94, 1992.

[11] D. Williams, "Fast target detection in synthetic aperture sonar imagery: A new algorithm and large-scale performance analysis," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 1, pp. 71–92, 2015.

[12] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[13] D. Williams and E. Fakiris, "Exploiting environmental information for improved underwater target classification in sonar imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6284–6297, 2014.

[14] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, 2014, pp. 818–833.