DEMYSTIFYING DEEP CONVOLUTIONAL NEURAL NETWORKS FOR SONAR IMAGE CLASSIFICATION

David P. Williams

NATO STO Centre for Maritime Research and Experimentation (CMRE)
Viale San Bartolomeo 400, 19126 La Spezia (SP), Italy

ABSTRACT

Deep convolutional neural networks (CNNs) are developed to perform underwater target classification in synthetic aperture sonar (SAS) imagery. The deep networks are trained using a huge database of sonar data collected at sea in various geographical locations. The value of CNN ensemble averaging is highlighted, and the feasibility of sonar sensor transfer learning with CNNs is also demonstrated. An analysis that seeks to demystify deep learning suggests tools for understanding how and why the trained CNNs work so well.

***Index Terms*—** Convolutional neural network (CNN), deep learning, classification, synthetic aperture sonar (SAS)

1. INTRODUCTION

"Deep learning" is the generic umbrella term used to denote classification algorithms with architectures characterized by a nested functional structure that engenders highly nonlinear decision surfaces. The great capacity of deep learning algorithms, such as deep convolutional neural networks (CNNs) [1], when paired with vast amounts of data and sufficient computational resources, has translated into state-of-the-art performance in diverse domains from image recognition [2] to cancer screening [3].

Despite this evidence, there remains reluctance in some military communities, such as underwater mine countermeasures (MCM), to embrace deep learning because of an incomplete understanding of how or why the algorithms succeed. Therefore, a principal aim of this work is to help demystify the proverbial "black box" of deep CNNs. We address this objective in the context of underwater target classification using synthetic aperture sonar (SAS) imagery, a sensor modality for which the use of deep learning is incipient.

Given the nascent nature of this algorithm-sensor pairing, the second goal of this work is to experimentally demonstrate how deep CNNs trained carefully on SAS imagery outperform the "shallow" classification method on which we had relied previously. (It should be noted that reusing popular extant CNNs trained on optical imagery for other applications is not feasible because the underlying phenomenology is fundamentally different with the sonar sensor.) In our previous work [4], the first to use deep CNNs with SAS imagery, the experiments addressed a task of limited scope that sought to distinguish between a pair of specific classes of objects using only a single, small CNN. We extend that work here by using deep CNNs for the general, more difficult, binary classification problem seeking to discriminate targets (*i.e.,* mines) from *all* types of clutter. We also learn *multiple* CNNs (that are also more sophisticated than that used previously), which importantly grants the advantages of ensemble averaging.

Another contribution is demonstrating the feasibility of sonar sensor transfer, in which CNNs are trained with data from one sonar but then used to classify (test) data from different sonars operating in different frequency bands, provided that the physics is sufficiently similar (*e.g.,* not low-frequency structural acoustics responses).

The remainder of this paper is organized as follows. Sec. 2 describes the CNNs developed and presents their target classification performance on SAS data. Sec. 3 seeks to elucidate how the CNNs are working, before concluding remarks are made in Sec. 4.

2. DEEP CONVOLUTIONAL NEURAL NETWORKS WITH SONAR DATA

2.1. Data and Training

A huge database of about 5×10^4 scene-level SAS images was collected by CMRE's MUSCLE autonomous underwater vehicle (AUV) during nine sea expeditions conducted between 2007 and 2015 in various geographical locations. The center frequency of the SAS is 300 kHz, and the bandwidth is 60 kHz; the high-resolution sonar imagery has an along-track resolution of 2.5 cm and a range resolution of 1.5 cm. A detection algorithm [5] was applied to the images to obtain "mugshots" of objects of interest, which become the inputs to the CNNs. The data and novel training procedure used for the CNNs are described fully in [4], but are omitted here due to space constraints.

The CNNs designed for this work each consist 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 pre-

Table 1. Architectures of CNNs learned

CNN Label	Conv. Layers	Numbers of Filters	Sizes of Filters	Pooling Factor
A	2	5, 10	16, 7	4
B	3	6, 18, 54	66, 32, 7	2
C	4	6, 6, 6, 6	12, 9, 6, 4	2
D	5	6, 6, 6, 6, 54	10, 8, 7, 4, 3	2

ceding layer, which creates a nested functional structure. The inputs to the initial layer are the SAS mugshots themselves (of size 167 pixels $\times$ 167 pixels). The outputs of the final layer are the probabilities of a mugshot belonging to each class, clutter or target.

Each convolutional layer and fully-connected layer employs a sigmoid activation function before the result is passed to the subsequent layer. Each pooling layer, which effectively downsamples, uses pure averaging rather than the commonly used max-pooling approach because the former is more robust when dealing with the speckle properties of sonar imagery. The training process of a CNN learns the parameters of the model, which for the convolutional layers are the filters (a.k.a. kernels) and associated bias terms. (There are no parameters associated with the pooling layers.)

The details of the specific CNN architectures developed in this work are summarized in Table 1; the filters are square and the size (of a side) is given in pixels. Each CNN was designed with different filter *sizes* (at the initial convolutional layer) to encourage firing on unique physical elements of the objects, with an eye toward later ensemble fusion of predictions. Although the CNNs are relatively small, their capacities are still sufficiently large to perform well on the task; in fact, beneficially, the small networks meant "dropout" [6] was not needed to prevent overfitting.

2.2. Evaluation and Results

Training was performed using data from the first eight sea expeditions, while the ninth was used for test (evaluation) purposes. The performance of the CNNs on the test data set in terms of receiver operating characteristic (ROC) curves and area under the ROC curve (AUC) is shown in Fig. 1. To improve the robustness of the CNNs' predictions, each test mugshot was mirrored about the range axis and the final prediction was taken to be the average of the predictions on the two (mirror image) mugshots.

Also shown in the figure is the result from a "shallow" approach that uses a modified version of a relevance vector machine [7], with the classifier parameters directly weighting a small set of traditional features [8] that have previously been found to characterize various attributes of the objects well.

Because CNNs with different architectures can learn very different representations, it can be advantageous to reduce the variance by averaging the predictions of multiple CNNs [9].

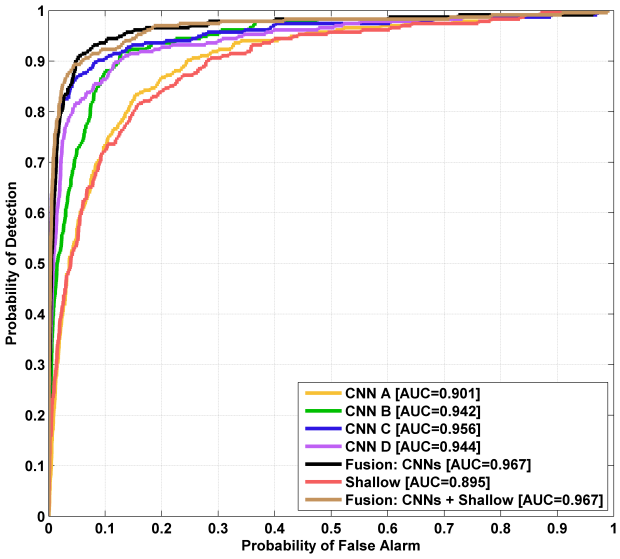


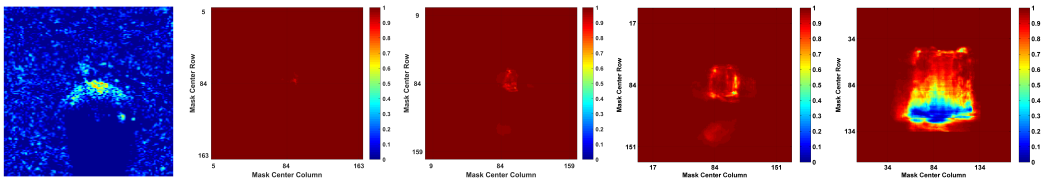
Fig. 1. Classification performance on the test data set.

This type of ensemble "fusion" was performed, both with and without including the shallow approach's predictions in the average. From Fig. 1, it can be seen that each of the CNNs outperforms the shallow approach, but that the fusion of CNNs achieves even superior performance. This insight importantly removes the burden of having to design a single "best" CNN architecture with optimal parameters (*e.g.,* numbers of and sizes of filters).

To explore the feasibility of sensor transfer learning, we also take the CNNs trained using full-resolution mugshots but then simulate lower-resolution *test* mugshot imagery (by downsampling by a factor in each image dimension). The performance that results from submitting the lower-resolution test imagery to the CNNs is shown in Table 2. The results suggest that sonar transfer learning is indeed possible, with gradual degradation in performance; this implies that the CNNs trained on MUSCLE sensor data are relevant for exploitation with data collected by a wide range of other lower-resolution side-scan sonars (whose resolution is up to 8 times poorer in both along-track and range). The results also hint at the CNN architectures that are fully exploiting high-resolution information, and conversely, which are more suitable for successful sensor transfer.

Table 2. AUC as a function of test-mugshot resolution

	Resolution Factor				
Method	1	2	4	8	16
CNN A	0.901	0.907	0.913	0.903	0.802
CNN B	0.942	0.942	0.942	0.936	0.907
CNN C	0.956	0.956	0.952	0.925	0.870
CNN D	0.944	0.941	0.928	0.901	0.850
Fusion	0.967	0.966	0.961	0.944	0.891



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

Fig. 2. (a) A target mugshot, and the corresponding CNN C predictions of belonging to the target class when background masks of size (b) 9 pixels, (c) 17 pixels, (c) 33 pixels, and (d) 67 pixels are applied at each location in the image. (The axis tick marks indicate the minimum and maximum mask center locations allowed.) The range of each colorbar is $[0, 1]$.

3. ANALYSIS

3.1. Effects of Masking

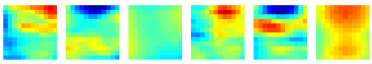
To better understand which elements of an object are driving a CNN's prediction, one can mask small regions of the mugshot as proposed by [10]. However, using a mask that zeroes out regions (as done in [10]) is inappropriate for this sonar data because shadows (where the normalized pixel values are approximately zero) are in fact informative. Therefore, we instead employ a mask that is fashioned from the seafloor background, which should impart minimal influence.

For this study, we consider CNN C that was learned and the target mugshot shown in Fig. 2(a). A square mask of side N pixels is created, where the mask is extracted beginning from the top center of the mugshot itself (which here guarantees a mask of background pixels). This mask is then applied to the mugshot at a specific location, swapping the original mugshot's pixels there with those of the mask. This modified mugshot is then submitted to the CNN to produce a prediction for belonging to each class. This process is repeated by shifting over the original mugshot the mask centered in each valid location (*i.e.,* where the mask does not exceed the bounds of the mugshot). The result is a "heat map" showing the CNN prediction when the mask has been applied at each location. By examining the change in prediction as a function of mask location (and size), one can better understand the key locations (and their spatial extents, generally) in the mugshot that are driving the prediction. Carrying out this procedure using masks with $N = \{9, 17, 33, 67\}$ pixels per side produces the results in Fig. 2. From the figure, it can be seen that the highlight and *adjacent* shadow region impact the predictions the most, and that a sizable mask is required to significantly alter the prediction; the latter fact indicates that a large region of the object is being used by the CNN.

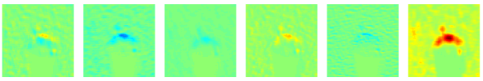
3.2. Interpreting Intermediate CNN Stages

To illustrate the inner machinery of CNNs, we show intermediate data products as the mugshot in Fig. 2(a) passes through the layers of the trained CNN C. In the following, for a given sub-figure, each image in the set uses an identical color scale in which the color green corresponds to zero, warmer colors are positive, and cooler colors are negative. The exceptions are sub-figures showing mugshots or outputs of the (sigmoid) activation function, for which the range of values is $[0, 1]$.

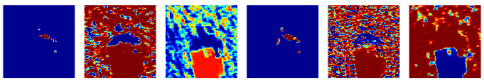
The mugshot is the input to the CNN, and it is convolved with the filters of the first convolutional layer, Fig. 3(a), which produces responses, Fig. 3(b), that are then (in conjunction with unique bias terms) passed through the sigmoid activation function to produce the output in Fig. 3(c).



(a) Filters of convolutional layer 1



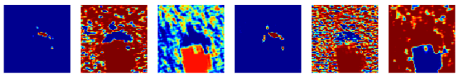
(b) Pre-activation responses of convolutional layer 1



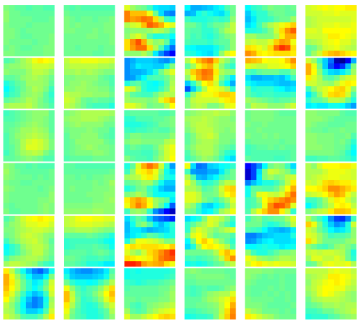
(c) Post-activation responses of convolutional layer 1

Fig. 3. Beginning stages of the CNN. See text for details.

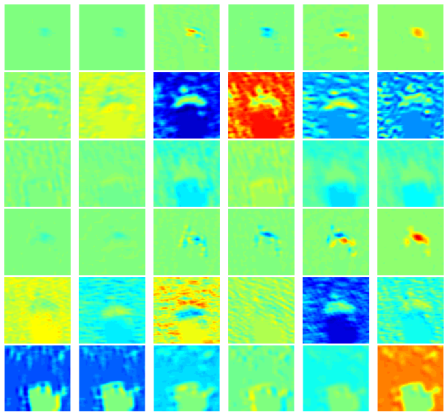
In a CNN, the filters are often a source of mystery and confusion. But examining the filters and the responses can provide insight into what the filters are actually doing. Here, it turns out that the first layer of filters are exploiting characteristics that are often used as shallow classifier features for sonar classification tasks. For example, the second filter is effectively performing binary segmentation into highlight and non-highlight regions. Similarly, the sixth filter is doing the same for shadows. The third filter is essentially segmenting the mugshot into highlight, shadow, and background regions. And the first and fourth filters are isolating strong, localized sonar returns. These same sorts of segmentations are commonly performed [11] in order to derive features, such as shadow length and highlight width, for shallow feature-based classifiers. An advantage of instead relying on CNNs is that no painstaking feature engineering is involved, and one is not constrained *a priori* by a human's imagination and perception of what is important for discrimination.



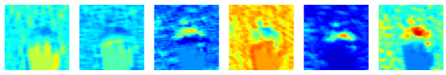
(a) Output of pooling layer 1



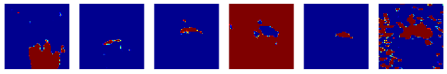
(b) Filters of convolutional layer 2



(c) Individual contributions to responses of convolutional layer 2



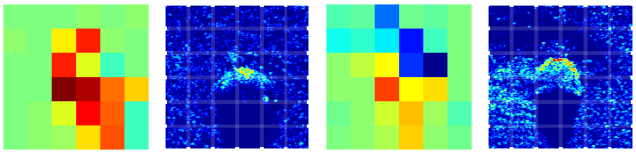
(d) Pre-activation responses of convolutional layer 2



(e) Post-activation responses of convolutional layer 2

Fig. 4. Intermediate stages of the CNN. (Three-dimensional filters are grouped columnwise.) See text for details.

The output in Fig. 3(c) then passes through a pooling layer, producing Fig. 4(a), a third-order tensor in which the third-dimension is referred to as depth. This result is then convolved with the filters of the second convolutional layer, with depth running vertically on the page in Fig. 4(b). By showing the intermediate result of each depth component in Fig. 4(c), one can better understand the function of the second layer of filters. When the third-dimension is collapsed via a summing operation to produce Fig. 4(d), and then passed through the activation function to output Fig. 4(e), it becomes more difficult to comprehend the result because each component (*i.e.,* in depth) is then a combination of (here) six different representations. It can be observed that at the second convolutional layer, the filters are mainly detecting edges at different orientations.



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

Fig. 5. (a) Individual contributions of each spatial region to class 2 prediction for (b) input target mugshot. (c) Individual contributions of each spatial region to class 2 prediction for (d) input clutter mugshot. (White grid lines are added to the mugshots to aid region association.)

As one passes deeper through the CNN – moving further away from the physical object – the level of abstraction increases and it becomes more challenging to interpret the concepts on which the filters are cueing. (These deeper layers are, vitally, the source of CNNs' great capacity and classification proficiency.) But at the final fully-connected layer, one can associate the individual contribution to the final class predictions of each *spatial* region (*i.e.,* receptive field) in the original mugshot. This valuable insight is illustrated in Fig. 5 for two mugshots. These contributions would then be summed and passed through a sigmoid activation function to obtain the probability of belonging to each class.

In Fig. 5(a), it can be observed that both the highlight region and the shadow region are strongly contributing (red colors) to the correct prediction that the mugshot belongs to class 2 (the target class). In Fig. 5(c), one can see that the shadow region is consistent with the target class, but the object highlight is not. By examining this stage of the CNN's output, one can better understand which precise elements of the object are being relied on to make predictions.

Space constraints prevent a more in-depth examination of other layers of the CNN and interpretations of other filters, but the tools proffered here can be used by other researchers to better understand their own data and results. Although CNNs are complex, highly nonlinear models, one can elucidate – through study of masking effects and intermediate outputs – why they may work well for a particular task.

4. CONCLUSION

Deep CNNs were trained using SAS imagery and shown to outperform a shallow approach that had been relied on previously for target classification. An analysis highlighted the feasibility of sensor transfer learning, the benefits of ensemble averaging, and the reasons for the CNNs' performance.

5. REFERENCES

[1] 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.

[2] 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.

[3] 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*, pp. 411–418. 2013.

[4] D. Williams, "Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, December 2016.

[5] 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.

[6] 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, no. 1, pp. 1929–1958, 2014.

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

[8] 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.

[9] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in Neural Information Processing Systems*, 2012, pp. 2843–2851.

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

[11] S. Reed, Y. Petillot, and J. Bell, "Automated approach to classification of mine-like objects in sidescan sonar using highlight and shadow information," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 151, no. 1, pp. 48–56, 2004.