ON THE BENEFIT OF MULTIPLE REPRESENTATIONS WITH CONVOLUTIONAL NEURAL NETWORKS FOR IMPROVED TARGET CLASSIFICATION USING SONAR DATA

David P. Williams[a], Ronan Hamon[a] & Isaac D. Gerg[b]

[a]NATO STO Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy
[b]Pennsylvania State University Applied Research Laboratory (PSU-ARL), State College, PA, USA

email: david.williams@cmre.nato.int

Abstract:* *The benefit of using multiple representations of data in the context of convolutional neural networks (CNNs) is demonstrated. We present three variations on this theme of multiple representations, in the form of (i) fundamentally different input data representations obtained from the same raw data, (ii) isometries of a given data representation, and (iii) intermediate representations arising from unique CNN architectures. Taken together, these variants can produce excellent classification performance while relying on orders of magnitude fewer free parameters than used in typical CNNs, thereby reducing training data requirements. The value of this multi-representation approach is demonstrated on a target classification task using real, measured sonar data collected at sea.

Keywords:* *Convolutional neural networks (CNNs), multiple representations, classification, sonar.

1. Introduction

Synthetic aperture sonar (SAS) [1] relies on the coherent processing of acoustic returns to produce high-resolution imagery of underwater environments that can be exploited for object classification and other tasks. But the richness of the *complex-valued* SAS data means there is potentially more exploitable information than what is apparent in the usual image domain. For example, the aspect-dependent nature of sonar returns off objects can be detected in the frequency domain [2], whereas the integration of this information to create (magnitude) imagery effectively obscures this key phenomenon. This insight motivates us to produce multiple data representations *that are fundamentally different in nature* – namely, a sonar magnitude image, phase image, and frequency spectrum – and use them jointly as inputs to a CNN to improve classification performance.

A standard CNN [3] is a sequence of convolutional layers, nonlinear activation functions, and pooling operations that collectively transform input data (*i.e.,* imagery) into a new representation space in which the classes are easily separable. But the alternative *input* data representations we propose could not be "uncovered" naturally by a CNN – *e.g.,* as intermediate-layer representations – because the relevant information is not *accessible* in the usual image domain. CNNs are a natural match for our multi-representation approach because they obviate the extraction of predefined features, which is challenging for difficult-to-interpret alternative data representations, such as sonar or radar *phase* imagery [4, 5].

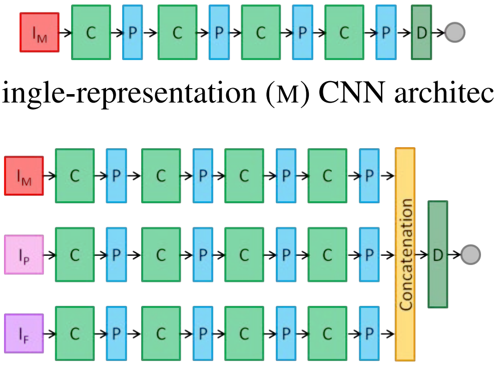
Some prior CNN-based research has focused on incorporating "multi-view" data from disparate sensor modalities [6–9], while other work has decomposed complex-valued data into two representations (*e.g.,* real and imaginary parts) [10, 11]. But our work is the first to derive and successfully exploit multiple input representations from the same raw sensor data. Our unification of the three multi-representation variants under a common theme, and our application to sonar data, are additional novel contributions.

The remainder of this paper is organized as follows. In Sec. 2, we present our multi-representation CNN framework and explain its benefits. Experimental results of the proposed approach on an object classification task using measured sonar data are shown in Sec. 3. Concluding remarks are made in Sec. 4.

2. Multi-Representation Classification

2.1. CNN Design

We carefully design 4 CNN architectures, whose common schematic is shown in Fig. 1. The key design choice when using multiple *disparate* input representations is how and when to merge them within the CNN. Because the representations capture fundamentally different physics, it is important to *not* simply treat the different data representations as separate channels – as is done for three-channel RGB optical images – because the responses will destructively interfere after the first convolutional layer. Instead, the representations should be kept distinct until a late (*i.e.,* deep) stage of the CNN. Here we propose an elegantly simple solution that is extensible for various applications and types of data. Specifically, when multiple input representations (*viz.* sonar magnitude image, phase image, and frequency spectrum) are used, the respective data products from the CNN transformations are concatenated at the penultimate layer (*cf.* Fig. 1).



(a) Single-representation (M) CNN architecture

(b) Multi-representation (MPF) CNN architecture

Figure 1: Proposed CNN architectures for (a) single and (b) multiple representation inputs. *C*, *P*, and *D* denote convolution *blocks* (comprising one or more convolutional layers), pooling layers, and dense layers, respectively.

Table 1: CNN architecture details

CNN Label	Filter Sizes (Pixels Per Side)	Pooling Factors
A	[4] [3] [3] [4]	4, 4, 2, 4
B	[8] [6] [4] [5]	4, 4, 2, 2
C	$\begin{bmatrix}6\\3\end{bmatrix} \begin{bmatrix}6\\3\end{bmatrix} \begin{bmatrix}6\\3\end{bmatrix} \begin{bmatrix}6\\3\end{bmatrix}$	4, 2, 2, 4
D	$\begin{bmatrix}8\\7\\5\end{bmatrix} \begin{bmatrix}8\\7\\5\end{bmatrix} \begin{bmatrix}7\\7\\5\end{bmatrix} \begin{bmatrix}8\\7\\5\end{bmatrix}$	2, 2, 2, 2

More specific details about the architectures designed for this work are provided in Table 1. Each data input representation to the CNNs is assumed to be 267 pixels by 267 pixels. Each CNN contains 4 convolution blocks; each block contains a specific number of convolutional layers (equal to the number of rows in Table 1's filter-size column). Each filter is square, and only 4 filters are used in each convolutional layer. ReLU activations are used after each convolutional layer, while a sigmoid activation is used at the output. All pooling layers use average pooling (rather than max-pooling) because the former approach has been observed [12, 13] to better handle the speckle phenomenon that characterizes sonar imagery. The design of the architecture (and specifically the final pooling layer) ensures that the dense layer always contains 4 nodes per input data representation. The capacities of the CNNs are intentionally kept so low in order to scrutinize the classification power of small networks when faced with limited training data.

2.2. Data Processing

A large database of scene-level SAS images (each of which typically spans 50 m × 110 m of seafloor) was collected by CMRE's MUSCLE autonomous underwater vehicle (AUV) during 13 sea expeditions conducted between 2007 and 2017 in various geographical locations. The center frequency of the SAS is 300 kHz and the bandwidth is 60 kHz, providing centimeter-level resolution sonar imagery.

The data from each expedition varies greatly in terms of seafloor composition (*e.g.,* sand, mud, vegetation), clutter types and densities, target types, image quality, and environmental complexity. Data from the 8 oldest expeditions are treated as training data for learning the CNNs, while the data from the 5 most recent expeditions are used as *distinct* test sets (owing to their diverse characteristics). The Mondrian detection algorithm [14] is applied to the complex scene-level sonar images, with this resulting in a set of image "chips" of objects to be classified as targets (class 1) or clutter (class 0) by the CNNs. The target class corresponds to man-made objects (usually mimicking mine shapes) that were purposely deployed prior to the surveys. A summary of the data sets is shown in Table 2, where it can be seen that the number of target examples in the training set is extremely limited.

Table 2: Summary of sonar data sets

Data Set	Survey Years and Locations	Number of Clutter	 Targets
Training Data	2008–2013 Latvia; Italy; Spain	29280	2912
MAN2	2014 - Bonassola, Italy	14313	404
NSM1	2015 - Ostend, Belgium	6580	113
TJM1	2015 - Cartagena, Spain	1938	351
ONM1	2016 - Hyères, France	111	91
GAM1	2017 - Patras, Greece	157	75

Given a complex sonar image, $z = x + iy$, of an object, the frequency representation used as input to the multi-representation CNNs is $\mathbf{I}_F = \frac{1}{3}(\log_{10}|\mathcal{F}\{z\}| - 8)$, where $\mathcal{F}$ is the 2-d discrete Fourier transform (and the extra constants effect a normalization). The magnitude image representation is given by $\mathbf{I}_M = |z|$, and the phase image representation is the result of the two-argument arc-tangent function, $\mathbf{I}_P = \phi(y,x)$. Each magnitude image and phase image are further normalized so that the pixel values of each are in $[-1,1]$. Each data input representation to the CNNs is 267 pixels by 267 pixels; for the magnitude and phase images, this corresponds to a spatial extent of 4.005 m × 4.005 m. An example of these three input data representations for a target is shown in Fig. 2.

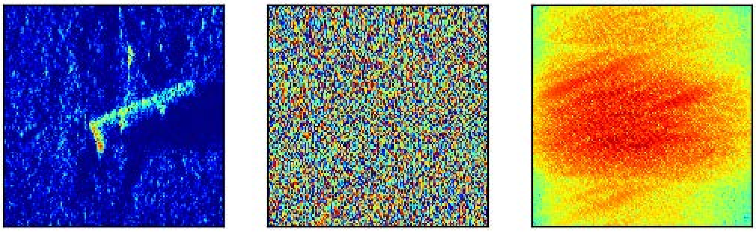


Figure 2: From left to right, a cylindrical target's three data representations: sonar magnitude image, phase image, and frequency spectrum. Strong normal-incidence returns from the cylinder face are visible (as linear features) in the latter.

2.3. CNN Training

CNN training was performed using the RMSprop optimizer with a learning rate of 0.001, in conjunction with a binary-cross-entropy loss function, until the loss on the training set converged. A batch size of 64 was used, with equal numbers selected from each class to combat the severe class-imbalance of the training data. Importance sampling to bias toward choosing challenging training data – as quantified by the Mondrian detection score – was employed. Data augmentation was effected by applying minor random translations in the range and along-track directions, as well as along-track reflections, to each (complex) sonar image chip selected for the batch. No attempt was made to optimize the learning rate or batch size.

Using the architectures described in Sec. 2.1, we train 4 single-representation and 4 multi-representation CNNs, where the former set (denoted M) uses only the sonar magnitude image, while the latter set (denoted MPF) also uses the phase image and frequency spectrum as additional inputs.

3. Experimental Results

We conduct experiments to show the benefit to classification performance, measured in terms of the area under the curve (AUC), of using three multi-representation variants. Specifically, we compare (i) using a set of isometric inputs versus using only image-centered inputs with a CNN, (ii) using an ensemble of CNNs with unique architectures versus using only a single CNN, and (iii) using a set of alternative data representations versus using only sonar magnitude imagery as input to the CNN.

For the first case, demonstrating the value of multiple representations of sonar imagery resulting from isometries of each original input data example, we employ a set of 18 affine transformations that do not violate the physics of the sonar-object geometry. (These operations are performed on the raw complex-valued imagery.) This set is formed from the Cartesian product of range translations $i_{tx} = \{0, 0.25\text{ m}, 0.50\text{ m}\}$, along-track translations $i_{ty} = \{-0.25\text{ m}, 0, 0.25\text{ m}\}$, and along-track reflections $i_{ry} = \{0, 1\}$.

The classification performance of the 4 multi-representation CNNs on the 5 test data sets is shown in Table 3 for when the CNNs are evaluated using object-centered ($\oslash$) input images or the ensemble ($\mathcal{E}$) of 18 isometric inputs. (Bold values indicate the best performance achieved among methods within the table's double vertical lines, on a row-wise basis.) The use of multiple representations in the form of simple isometries clearly provides significant performance gains.

Also shown in the table is the result of using the ensemble of the 4 CNNs' predictions ($\mathcal{E}(\text{ABCD})$) as the final prediction for each test data point. This second form of multiple representations, exploiting the fact that the unique CNN architectures induce different intermediate representations, also consistently provides benefit over the individual CNNs.

And lastly, to demonstrate the value of employing multiple *input data* representations, Table 4 shows the performance of the single-representation CNNs (M) and the multi-representation CNNs (MPF). For comparison purposes, we also adapted the pre-trained VGG16 net [15] to our problem and (sonar magnitude) data; details of this transfer appear in [16]. The best VGG16 performance achieved (after considerable effort and parameter tuning) is shown in Table 4. The performance of the Mondrian detector, which makes predictions using a set of 5 features with fixed weights (and thus has no free parameters), is included as a baseline "shallow" classification approach.

Table 4 also shows the number of parameters of each method. The ensemble of our 8 CNNs generates predictions after training around 5×10^4 free parameters, which is only 0.3%

of the 1.47×10^7 parameters contained in the VGG16 net. The architecture of the VGG16 net is similar to ours, with alternating convolution blocks and pooling layers. One of the major differences is the number of filters in each convolutional layer; the VGG16 net uses between 64 and 512 filters in each convolutional layer, while our CNNs use only 4 filters in each layer. Despite the handicap of relying on orders of magnitude fewer free parameters, our limited-capacity CNNs collectively perform favorably to the VGG16 net, as can be seen in Table 4. Additionally, with our CNNs, training is more straightforward, free of extensive parameter tuning.

One valuable finding from the course of this work is that the individual branches of the multi-representation CNN should *not* be initialized with weights learned from training in isolation the analogous single-representation CNN. As can be seen in Fig. 3 (a) and (b), the filters that are learned for a single-representation CNN differ from those of the multi-representation CNN branch of the same data representation. We hypothesize that using the single-representation CNN to initialize, as was done in [16], causes training to get stuck in the equivalent loss-space minimum, thereby preventing gains (from the additional representations) from being realized.

Finally, Fig. 4 shows the intermediate responses at each layer of CNN B in the single-representation and multi-representation forms. The drastically distinct responses arising from the disparate input data representations are obvious.

Table 3: AUC for multi-representation (MPF) CNNs using centered ($\oslash$) input images or an ensemble ($\mathcal{E}$) of isometric inputs

Test	CNN A		CNN B		CNN C		CNN D		$\mathcal{E}$(ABCD)	
Data Set	$\oslash$	$\mathcal{E}$	$\oslash$	$\mathcal{E}$	$\oslash$	$\mathcal{E}$	$\oslash$	$\mathcal{E}$	$\oslash$	$\mathcal{E}$
MAN2	0.958	**0.965**	0.969	**0.980**	0.975	**0.979**	0.978	**0.981**	**0.986**	**0.986**
NSM1	0.921	**0.935**	0.944	**0.963**	0.956	**0.975**	0.939	**0.952**	0.976	**0.980**
TJM1	0.985	**0.989**	0.993	**0.996**	0.994	**0.995**	0.993	**0.997**	0.997	**0.998**
ONM1	0.947	**0.962**	0.936	**0.957**	0.983	**0.991**	0.966	**0.985**	0.982	**0.987**
GAM1	0.975	**0.985**	0.992	**0.993**	**0.997**	**0.997**	**0.999**	**0.999**	**0.998**	**0.998**

Table 4: AUC for single-representation (M) and multi-representation (MPF) CNNs and competing methods

Test	CNN A		CNN B		CNN C		CNN D		$\mathcal{E}$(ABCD)			VGG16	Mondrian
Data Set	(M)	(MPF)	(M)	(MPF)	(M)	(MPF)	(M)	(MPF)	(m)	(MPF)	(M, MPF)		
MAN2	0.964	**0.965**	0.979	**0.980**	**0.981**	0.979	0.955	**0.981**	0.984	0.986	**0.987**	0.986	0.928
NSM1	**0.935**	**0.935**	0.938	**0.963**	0.968	**0.975**	0.949	**0.952**	0.977	0.980	0.982	**0.983**	0.892
TJM1	0.985	**0.989**	0.995	**0.996**	**0.998**	0.995	0.993	**0.997**	0.998	0.998	**0.999**	0.996	0.955
ONM1	**0.978**	0.962	**0.987**	0.957	**0.991**	**0.991**	0.968	**0.985**	**0.992**	0.987	0.988	0.958	0.851
GAM1	**0.986**	0.985	0.988	**0.993**	0.994	**0.997**	0.990	**0.999**	**0.999**	0.998	0.998	0.997	0.978
Parameters	629	1885	1509	4525	2485	7453	7877	23629	12500	37492	49992	14715201	0

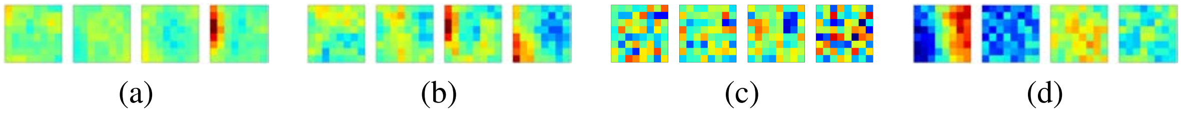


Figure 3: For the single-representation CNN B using the magnitude image as input, (a) the first convolutional layer's filters. For the multi-representation CNN B, the first convolutional layer's filters for the (b) magnitude-image branch, (c) phase-image branch, and (d) frequency-spectrum branch. The filters of a given subfigure use the same colorscale; green corresponds to zero.

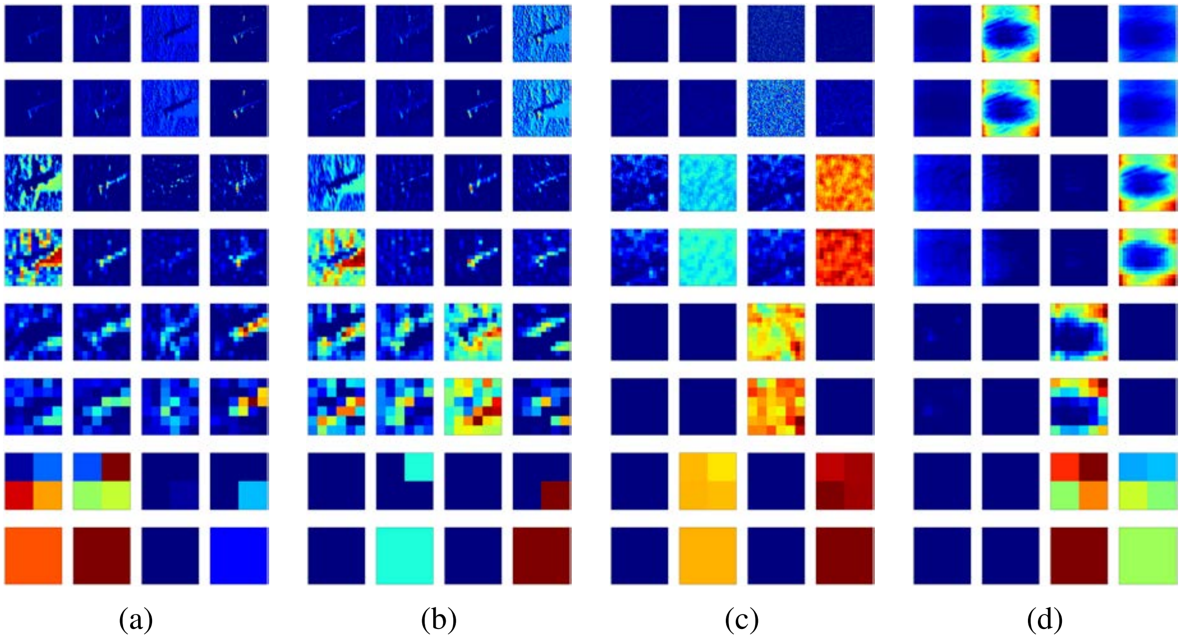


Figure 4: Intermediate CNN layer responses for the data example in Fig. 2. For the single-representation CNN B using the magnitude image as input, (a) the responses after each layer. For the multi-representation CNN B, the responses after each layer, prior to concatenation, for the (b) magnitude-image branch, (c) phase-image branch, and (d) frequency-spectrum branch.

4. Conclusion

This work demonstrated the benefit of using three forms of multiple representations in the context of CNNs. Taken together, these variants can produce excellent classification performance while relying on orders of magnitude fewer free parameters. Thus, by exploiting alternative techniques that enable us to employ CNNs with limited capacity, we effectively reduce the amount of training data required. This result is valuable for remote-sensing applications, and in particular underwater target classification tasks, where the collection of data (at sea) is prohibitively costly.

REFERENCES

[1] M. Hayes and P. Gough. Broad-band Synthetic Aperture Sonar. *IEEE Journal of Oceanic Engineering*, 17(1):80–94, 1992.

[2] D. Plotnick and T. Marston. Utilization of Aspect Angle Information in Synthetic Aperture Images. *IEEE Trans. on Geoscience and Remote Sensing*, 56(9):5424–5432, 2018.

[3] Y. LeCun, Y. Bengio, and G. Hinton. Deep Learning. *Nature*, 521(7553):436, 2015.

[4] D. Williams. Exploiting Phase Information in Synthetic Aperture Sonar Images for Target Classification. In *Proceedings of the IEEE OCEANS*, pages 1–6, 2018.

[5] M. Alver, S. Atito, and M. Çetin. SAR ATR in the Phase History Domain Using Deep Convolutional Neural Networks. In *Image and Signal Processing for Remote Sensing XXIV*, volume 10789, pages 1078913–1 – 1078913–10, 2018.

[6] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On Deep Multi-view Representation Learning. In *International Conference on Machine Learning*, pages 1083–1092, 2015.

[7] A. Wang, J. Lu, J. Cai, T. Cham, and G. Wang. Large-Margin Multi-modal Deep Learning for RGB-D Object Recognition. *IEEE Trans. on Multimedia*, 17(11):1887–1898, 2015.

[8] Y. Bin, Y. Yang, F. Shen, and X. Xu. Combining Multi-representation for Multimedia Event Detection Using Co-training. *Neurocomputing*, 217:11–18, 2016.

[9] J. Wagner, V. Fischer, M. Herman, and S. Behnke. Multispectral Pedestrian Detection Using Deep Fusion Convolutional Neural Networks. In *ESANN*, pages 509–514, 2016.

[10] J. Muth, S. Uhlich, N. Perraudin, T. Kemp, F. Cardinaux, and Y. Mitsufuji. Improving DNN-based Music Source Separation Using Phase Features. *arXiv preprint arXiv:1807.02710*, 2018.

[11] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury. Deep Learning Convolutional Neural Networks for Radio Identification. *IEEE Comms. Magazine*, 56(9):146–152, 2018.

[12] D. Williams. Underwater Target Classification in Synthetic Aperture Sonar Imagery Using Deep Convolutional Neural Networks. In *Proceedings of ICPR*, 2016.

[13] M. Emigh, B. Marchand, M. Cook, and J. Prater. Supervised Deep Learning Classification for Multi-band Synthetic Aperture Sonar. In *Proceedings of the 4th International Conference on SAS/SAR*, volume 40, pages 140–147, 2018.

[14] D. Williams. The Mondrian Detection Algorithm for Sonar Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):1091–1102, 2018.

[15] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] I. Gerg and D. Williams. Additional Representations for Improving Synthetic Aperture Sonar Classification Using Convolutional Neural Networks. In *Proceedings of the 4th International Conference on SAS/SAR*, volume 40, pages 11–22, 2018.