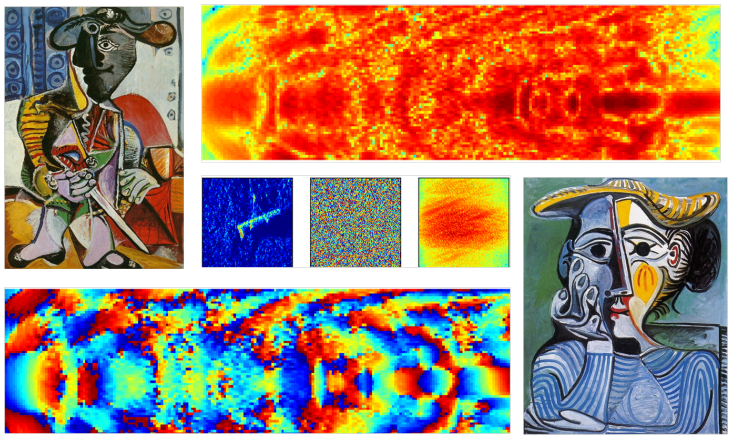
Final Report
SERDP Project MR18-1444

Cubist-Inspired Deep Learning with Sonar
for UXO Detection and Classification



David P. Williams

david.williams@cmre.nato.int

NATO STO Centre for Maritime Research and Experimentation (CMRE)

16 September 2019

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
21-01-2019	SERDP Final Report	11/09/2017 – 21/01/2019

4. TITLE AND SUBTITLE

Cubist-Inspired Deep Learning with Sonar for UXO Detection and Classification

5a. CONTRACT NUMBER
W74RDV72490702, W74RDV80675566

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Williams, David P.

5d. PROJECT NUMBER
MR18-1444

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NATO STO Centre for Maritime Research and Experimentation(CMRE)
Viale San Bartolomeo 400
19126 La Spezia (SP)
Italy

8. PERFORMING ORGANIZATION REPORT NUMBER
MR18-1444

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Strategic Environmental Research and Development Program
4800 Mark Center Drive, Suite 17D08
Alexandria, VA 22350-3605

10. SPONSOR/MONITOR'S ACRONYM(S)
SERDP

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
MR18-1444

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The objective of this project was to demonstrate a proof-of-concept regarding the feasibility of using convolutional neural networks (CNNs) for unexploded ordnance (UXO) classification. Within this context, one main strand of work focused on assessing the applicability of two forms of transfer learning for the task of underwater object classification: target-concept transfer and sensor transfer. The use of transfer learning would allow data collected during mine countermeasures operations, and from different but similar sensors, to be leveraged for the UXO problem. The other main task of the project was to develop a CNN framework that could exploit multiple representations of sonar data simultaneously. The idea underlying the use of multiple representations (derived from the same raw data) is that complementary classification clues would be made accessible in different representations. Experiments involving measured sonar data demonstrated the successful fulfillment of the project objectives.

15. SUBJECT TERMS

Unexploded Ordnance (UXO), Sonar, Convolutional Neural Networks (CNNs), Detection, Classification, Transfer Learning, Multiple Representations

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			David P. Williams
UNLCASS	UNLCASS	UNLCASS	UNLCASS	62	**19b. TELEPHONE NUMBER** *(include area code)* +39 0187-527-439

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

List of Acronyms

ACT	Allied Command Transformation
APL-UW	Applied Physics Laboratory - University of Washington
ATR	Automatic Target Recognition
AUC	Area Under the Curve
AUV	Autonomous Underwater Vehicle
BB	Broadband
BRAC	Base Realignment and Closure
CMRE	Centre for Maritime Research and Experimentation
CNN	Convolutional Neural Network
DEU	Diver Evaluation Unit
DOD	Department of Defense
FUDS	Formerly Used Defense Sites
GAMEX	Greek Ariadne Mine Experiment
GPU	Graphics Processing Unit
HF	High Frequency
LF	Low Frequency
MANEX	Multinational Autonomous Experiment
MCM	Mine Countermeasures
ML	Machine Learning
MR	Munitions Response
MUSCLE	Mine-Hunting UUV for Shallow-Water Covert Littoral Expeditions
NATO	North Atlantic Treaty Organization
NSMEX	North Sea Mine Experiment
NSWC-PCD	Naval Surface Warfare Center - Panama City Division
ONMEX	Olives Noires Mine Experiment
ONR	Office of Naval Research
PC SWAT	Personal Computer Shallow Water Acoustics Tool-set
PSU-ARL	Penn State University Applied Research Laboratory
ReLU	Rectified Linear Unit
RGB	Red-Green-Blue
RMS	Root Mean Square
ROC	Receiver Operating Characteristic
RVM	Relevance Vector Machine
SAR	Synthetic Aperture Radar
SAS	Synthetic Aperture Sonar
SEED	SERDP Exploratory Development
SERDP	Strategic Environmental Research and Development Program
STO	Science and Technology Organization
SVM	Support Vector Machine
TIER	Target-in-the-Environment Response
TJMEX	Trident Juncture Mine Experiment
TP-T	Target Practice - Tracer
TREX	Target and Reverberation Experiment
UUV	Unmanned Underwater Vehicle
UXO	Unexploded Ordnance
VGG	Visual Geometry Group
WTD-71	Wehrtechnische Dienststelle für Schiffe und Marinewaffen, Maritime Technologie und Forschung

Keywords

Unexploded Ordnance (UXO), Sonar, Convolutional Neural Networks (CNNs), Detection, Classification, Transfer Learning, Multiple Representations, High-Frequency Synthetic Aperture Sonar (SAS), Low-Frequency Acoustic Color.

Acknowledgments

The author thanks the following entities:

- the Strategic Environmental Research and Development Program (SERDP) for funding this project;
- the NATO Allied Command Transformation (ACT) for funding the MUSCLE data collection and processing;
- Holger Schmaljohann at Wehrtechnische Dienststelle für Schiffe und Marinewaffen, Maritime Technologie und Forschung (WTD-71) for providing the SeaOtter data;
- Isaac Gerg at the Penn State University Applied Research Laboratory (PSU-ARL) for assisting with the manual labeling of the SeaOtter data, and for running the experiments with the VGG16 network;
- Kevin Williams at the Applied Physics Laboratory - University of Washington (APL-UW) for providing the TREX13 and TIERSWAT data;
- the estate of Pablo Picasso / Artists Rights Society (ARS), New York, for granting permission to reproduce the paintings on the cover page, “Le Matador” and “Femme au Chapeau Jaune (Jacqueline).”

Abstract

The objective of this project was to demonstrate a proof-of-concept regarding the feasibility of using convolutional neural networks (CNNs) for unexploded ordnance (UXO) classification. Within this context, one main strand of work focused on assessing the applicability of two forms of transfer learning for the task of underwater object classification: target-concept transfer and sensor transfer. The use of transfer learning would allow data collected during mine countermeasures operations, and from different but similar sensors, to be leveraged for the UXO problem. The other main task of the project was to develop a CNN framework that could exploit multiple representations of sonar data simultaneously. The idea underlying the use of multiple representations (derived from the same raw data) is that complementary classification clues would be made accessible in different representations.

Using real measured sonar data collected at sea, the feasibility of the two forms of transfer learning were successfully demonstrated. To address the second task, a novel CNN-based classification framework for multiple-representation data was developed. More specifically, a flexible approach that naturally extends to any number of representations was made by fusing the disparate representations at the penultimate dense layer of the CNN. Importantly, it was demonstrated that successful classification could be achieved with notoriously "data-hungry" CNNs – even when only limited training data were available – by leveraging domain expertise in the design of the networks. Employing CNNs with orders of magnitude fewer parameters than are typically used resulted in robust classifiers that generalized well even to objects not seen during training. The benefit of drawing on an ensemble of predictions obtained via various forms of multiple representations – in terms of input data representations, isometries, frequency sub-bands, aspect sub-apertures, multiple object views, and unique CNN architectures – was also illustrated. In particular, it was shown how this approach can greatly reduce the false alarm rate while maintaining the requisite high detection probability that UXO remediation demands.

Collectively, these results show that the proof-of-concept can be deemed to have been fully achieved. That is, the groundwork has now been laid for a comprehensive classification framework that could incorporate multiple data representations, from both high-frequency and low-frequency sonar data, as well as from different sensors, in order to improve underwater UXO remediation efforts. Additionally, the research has opened another promising avenue for future follow-on work.

The main thrust of the proposed follow-on work is to leverage the developed CNN framework in conjunction with specially controlled experiments to learn principled, explainable classification features that can be tied directly to the wave phenomena of the physics involved. This plan envisions a symbiotic feedback loop in which features uncovered by the CNN – and related to specific object characteristics – help inform the understanding of the problem's physics, and vice versa. Both components would contribute to enabling more efficient UXO remediation strategies from a more rigorous scientific perspective.

Chapter 1

Introduction

1.1. OBJECTIVE

An unfortunate legacy of former military activities at sites designated for base realignment and closure (BRAC) and at Formerly Used Defense Sites (FUDS) is the contamination of aquatic environments with military munitions. In the United States, more than 400 underwater sites, spanning an area in excess of 10 million acres, potentially contain such munitions [1]. The presence of these munitions is a serious threat to both humans and the environment, so remediation is necessary. But the return of these contaminated waters to public use is contingent upon the analysis and assessment of wide-area and detailed underwater surveys. Therefore, the Department of Defense (DOD) has an express need for the development of technologies that will enable the detection and classification, at high probability, of military munitions found at underwater sites.

The objective of this project is to develop a novel classification framework for unexploded ordnance (UXO) that exploits synthetic aperture sonar (SAS) data. The new algorithms are based on deep-learning techniques, specifically deep convolutional neural networks (CNNs) [2]. The successful development of this approach should enable the attainment of higher probabilities of detection and classification, at much lower false alarm rates, than is possible with existing approaches. As a result, the application of these machine-learning algorithms to sonar data collected at potentially contaminated underwater sites can guide remediation efforts to effect savings. Specifically, because fewer resources will be spent investigating harmless clutter, the cost of remediation should decrease substantially.

1.2. TECHNICAL APPROACH

"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 CNNs, when paired with vast amounts of data and sufficient computational resources, has translated into state-of-the-art performance in diverse domains. In this project, new deep CNNs are developed for UXO classification. The inspiration for this work, perhaps surprisingly, is the avant-garde movement known as Cubism [3] that revolutionized the art world in the early 1900s. A hallmark of this style of painting and sculpture was the depiction of multiple perspectives simultaneously. In an analogous manner, the new deep CNNs developed incorporate multiple representations of the data – *e.g.,* SAS magnitude imagery, phase imagery, frequency spectrum data – simultaneously. The key is that these alternative representations make certain relevant information *accessible*

and therefore exploitable by a CNN. In the standard image domain, many of the salient clues for classification would effectively remain hidden. To leverage our prior deep-learning research in mine countermeasures (MCM), a significant component of this project also involves exploring the feasibility of two types of transfer learning for CNNs – between sensors (operating at different frequencies) and between target classes (mines to UXO).

1.3. BENEFITS

This project addresses the DOD's need for robust detection and classification approaches for underwater UXO. The scientific community will benefit from the development of a deep-learning framework that, as a by-product, also automatically uncovers valuable classification features (via the learned CNN filters). The result of this work also has the potential to form a foundation for follow-on efforts that would seek to unify high-frequency and low-frequency sonar-data-based classification approaches.

1.4. ORGANIZATION OF REPORT

This report summarizes the research conducted under SERDP SEED Project MR18-1444, from September 2017 to January 2019.

The remainder of the report is organized as follows. Chapter 2 provides a high-level overview of the work executed during the project. Chapter 3 presents the relevant technical details about the CNN algorithm, which is then used for the experiments in the remainder of the report. Chapter 4 describes experiments conducted using high-frequency sonar data. Chapter 5 briefly summarizes the contributions and findings of the project, and then outlines several avenues for future follow-on research. A separate appendix presents experiments that exploited low-frequency sonar data.

Chapter 2

Overview of Work

This chapter provides a high-level overview of the research conducted in this project. More specifically, we explain the logic behind the progression of the work, and how the results of one task informed the approach to the subsequent ones.

In previous work [4] prior to this project, we had developed CNNs for SAS (magnitude) imagery by using a large database of MCM sonar data collected by CMRE's MUSCLE autonomous underwater vehicle (AUV). Work in the current project began by leveraging this existing technology developed for MCM to verify that CNN transfer learning was possible with sonar imagery. First we demonstrated that the previously trained CNNs could be re-used by modifying the target concept of the networks, from various mine shapes to cylindrical objects (as a proxy for UXO).

We then proceeded to demonstrate a second type of CNN transfer learning, involving transfer between both target-concept and sensor: training a CNN using SAS data from one sensor (MUSCLE sensor and mine targets), and adapting it to enable inference on SAS data from a different sensor (SeaOtter Mk II sensor and UXO targets). However, the SeaOtter data, which was the only high-frequency sonar data containing UXO to which we had access, had not been "curated" for classification studies. This meant that we first had to perform detection on the wide-area imagery to generate a list of objects of interest, and also manually label (*i.e.,* ground-truth) the data set. The Mondrian detector [5], a flexible general-purpose object-detection technique designed to be robust to unreliable acoustic shadows (as partially buried UXO objects would produce), proved suitable for the detection task.

Underwater UXO surveys are expensive, and properly curating the collected data is also time-consuming. From these transfer-learning studies, it was established that one can indeed take CNNs trained for the MCM problem and adapt them to the UXO problem. The main implication of this finding is that one can leverage larger MCM databases, including those collected by multiple similar yet distinct sensors, to help inform UXO classification efforts. For this reason, we continued our research using the much larger, richer MCM data set available to us, with the knowledge that the findings would be directly applicable to UXO data as well.

The next strand of work explored the use of alternative data representations. SAS imagery is complex-valued, but typically only the magnitude image is ever used for classification. We sought to examine whether discriminatory information also exists in the phase image, and whether this could be uncovered automatically by a CNN. We indeed established that the phase information present in complex high-frequency SAS imagery could be exploited for successfully discriminating man-made targets from naturally occurring clutter. This result was important because it suggested that one could employ CNNs with data representations for which a human would have difficulty defining features manually, such as acoustic-color data. Moreover,

this finding also preserved the possibility of using multiple data representations simultaneously in a CNN, with the key being that complementary classification clues would be accessible (by the CNN) in different data representations.

During the course of analyzing the results of the aforementioned experiments, ways to make the CNN architectures more efficient were identified. So moving forward, a new set of smaller CNNs were designed, which would allow markedly faster training, reduce data requirements, and improve interpretability. Importantly, these changes would also make the use of *ensembles* of CNNs feasible.

The next major body of work examined using multiple representations of data in the context of CNNs. We explored 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 – *i.e.,* distance-preserving transformations – of a given data representation, and (iii) intermediate representations arising from unique CNN architectures. Taken together, these variants were able to produce excellent classification performance while relying on orders of magnitude fewer free parameters than used in typical CNNs, thereby reducing training data requirements.

But the most significant output from this particular study was the creation of a principled way to combine disparate information sources in a CNN framework. As such, it established a blueprint for how one could fuse multiple representations, and even high-frequency and low-frequency data products, together in a cohesive, unified manner. Moreover, the framework is flexible and can be seamlessly extended to accommodate additional data products from other sensors, or auxiliary information from other sources, as they become available. Essentially, designing such a framework was the ultimate goal of this project.

All of the aforementioned work was conducted on real, measured high-frequency SAS data collected at sea. By this point in the project, all of the tasks set forth in the original project proposal had been completed. But the various insights and findings from the studies had opened several tantalizing research avenues to pursue for classification with *low-frequency* sonar data. Additionally, we had a desire to demonstrate the efficacy of the developed algorithms on more substantial sets of *UXO* data specifically. Therefore, we sought and obtained low-frequency UXO data from colleagues in the research community and then used it to conduct additional CNN studies.

Leveraging all of the knowledge gained during the project, we first designed efficient CNNs for low-frequency acoustic-color data, a representation that expresses target strength as a function of object aspect and frequency. We showed that it was possible, using only limited amounts of this sonar data, to design and train efficient networks with low capacity that avoid overfitting and generalize robustly, even to new objects not seen during training. Importantly, this result showed how the CNN-based approach could obviate the challenging process of manually defining features for acoustic-color data. Classification performance was also examined, for several different CNN architectures, as a function of object type, orientation, range, and input-data aspect span.

Next, we demonstrated that partitioning the acoustic-color data into multi-band and multi-aspect subsets was more useful than treating the full frequency band and full aspect span as a single data product. This form of multiple representations instills robustness by relying on an ensemble of classifier predictions, which improves overall classification performance, but more crucially for the UXO problem, reduces the false alarm rate associated with the target detection probability at or near unity.

The benefit of executing additional survey passes to obtain multiple views of objects was also quantified, and the optimal look separations (from the initial look) for the two-view and

three-view cases were established. This result is valuable for UXO remediation, where multiple surveys are acceptable because time constraints are less of an issue than in MCM operations.

By exploiting the general multiple-representation CNN framework that had been developed earlier in the project, we next demonstrated the power of using four data representations derived from dual-band, complex-valued acoustic-color data. The impacts of data representation and training environment on classification performance were also studied.

Lastly, we reflected on the comprehensive body of work completed to formulate several promising avenues for follow-on research that would build on insights from this project. The main ideas to pursue revolve around a proposed framework for creating *explainable* CNN-based features that can be tied directly to the physics of the wave phenomena involved. Moreover, this same approach could also be used to establish when an increase in scattering-model complexity is warranted, and to determine whether simulated data can be confidently used interchangeably with real data for classifier training.

Chapter 3

Convolutional Neural Networks

3.1. INTRODUCTION

Convolutional neural networks (CNNs) have recently achieved state-of-the-art performance on a wide range of image classification tasks [6–8] and they are quickly becoming the preferred image-classification method for any domain characterized by vast amounts of labeled data. And even in remote-sensing applications, where the sensor imagery available is typically limited, CNNs are also increasingly being employed [9–18].

The ascendance and supremacy of CNNs for image classification tasks has largely been due to the increased availability of data and processing power. Given sufficient amounts of these two key resources, a CNN will almost certainly outperform traditional "shallow" machine learning (ML) approaches. This observation is driven by the fact that traditional shallow classifiers eventually hit a performance plateau: beyond a certain point, incorporating more data ceases to improve performance. In contrast, a CNN can continue to improve as more data is made available during the training phase. This relationship is shown in the cartoon in Fig. 3.1.

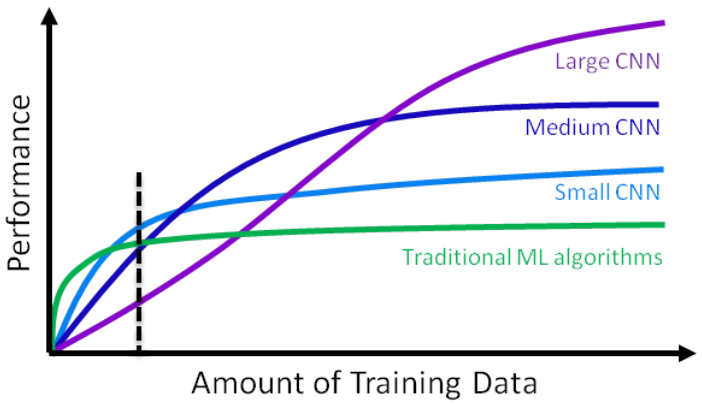


Fig. 3.1. General relationship between the amount of training data available and the classification performance on test data, for CNNs of different sizes and traditional ML algorithms. When a relatively modest amount of training data is available (*e.g.,* at the dashed vertical line), a smaller CNN (as well as traditional ML algorithms) can outperform larger CNNs. As the amount of data increases, larger CNNs with greater capacity can perform better. Relative to the sizes of popular CNNs used in the ML field, UXO sonar data sets are currently in the regime of extremely limited data.

The scaling of CNN performance with training data is due to the richness of the CNN architecture, and specifically its enormous *capacity*, which permits more sophisticated decision surfaces than shallow classifiers. Loosely speaking, the capacity of a CNN is dictated by the number of free trainable parameters in the model. The most popular architectures of famous "named" CNN families, such as VGG-Net [19], Inception [20], or ResNet [21], typically have between 10^6 and 10^8 free parameters. For comparison, commonly used shallow classification approaches, such as a support vector machine (SVM) [22] or relevance vector machine (RVM) [23], rarely have more than 10^2 or 10^3 free parameters, and can even have on the order of only 10 parameters.

The standard refrain about CNNs is that they require enormous amounts of data. But this lament elides subtle, but key, technical issues. The crucial quantity in determining the success of CNNs for classification tasks is not the amount of training data *per se*, but rather the relationship between training data and network capacity. As the number of free trainable parameters in the CNN model grows, so too do the training data requirements. If there is an insufficient amount of training data to support the model's complexity (*i.e.,* too many parameters), the model will *overfit* the training data, and performance on the test data will suffer. Therefore, when faced with limited training data, it is imperative to constrain the CNN's capacity by employing smaller networks. This notion is illustrated in the cartoon in Fig. 3.2.

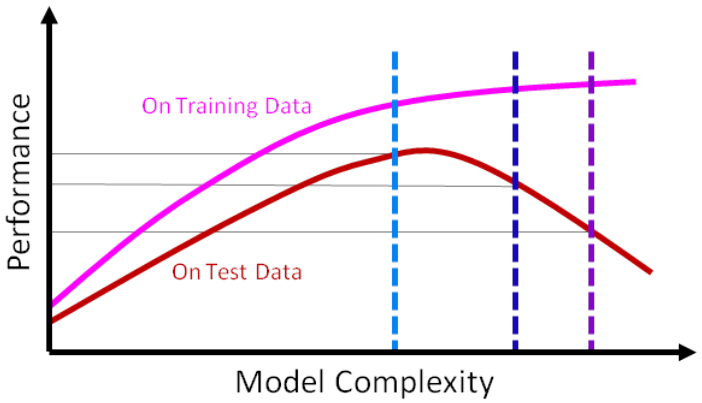


Fig. 3.2. General relationship between CNN model complexity and classification performance on training data and test data. The curves reflect the case for the amount of training data marked by the dashed line in Fig. 3.1; the colors of the vertical dashed lines here associate with that previous cartoon as well. An increase in CNN complexity will enable better performance on the training data, but if an insufficient amount of data exists, overfitting will occur. As a result, performance on the test data will suffer.

3.2. A GENTLE OVERVIEW

A CNN is a sophisticated classification algorithm whose power derives from its great representational capacity. The standard architecture of a CNN consists of sequences of convolutional layers, nonlinear activations, and pooling layers, that are then followed by one or more fully-connected "dense" layers, and a final prediction. The output of one layer is the input to the subsequent layer, with this nested functional structure – in conjunction with the nonlinear activation functions – enabling highly complex decision surfaces. The input to a CNN is an image,

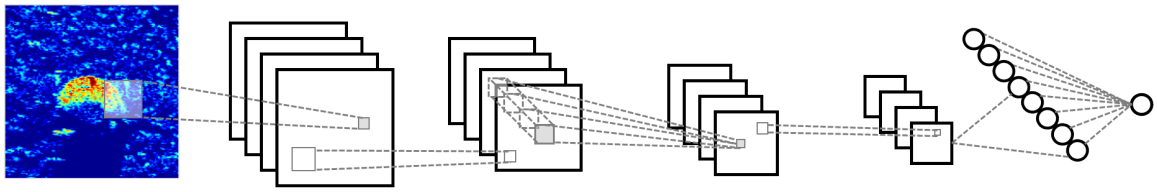


Fig. 3.3. Schematic showing the intermediate data representations that would result from a basic CNN architecture consisting of the input image (a SAS "chip"), a convolutional layer with 4 filters, a pooling layer, a second convolutional layer with 4 filters, another pooling layer, a fully-connected ("dense") layer with 8 nodes, and the final class probability output. The representations at the dense layer can be viewed as conventional features.

and the outputs are the probabilities of belonging to each class under consideration (*e.g.,* targets and clutter). A schematic representation of this basic architecture is shown in Figure 3.3.

Training a CNN simply means learning the filters, and associated bias (*i.e.,* offset) terms, of the convolutional layers.[1] (There are no parameters associated with the pooling layers.) Instead of using pre-defined filters manually created by a human, the CNN automatically *learns* what the filters should be from the data, and in effect, the most useful bases in which to represent the data. That is, CNNs obviate the extraction of predefined, human-engineered features, which can be challenging for difficult-to-interpret data representations (*e.g.,* acoustic color). This property makes CNNs particularly appealing, as it allows the algorithm to uncover clues beyond those enumerated by human intuition.

The filters transform the input data (imagery) into a new representation space. This new representation is simply the output of convolving the filters, also known as kernels, with the input image. The intermediate representations of the data in each CNN layer are usually referred to as *feature maps*. (When there are multiple feature maps in a layer, the associated filters will be three-dimensional tensors.)

With successive convolutional layers, the level of data abstraction and (highly nonlinear) decision-surface complexity increases. (The number of convolutional layers employed, known as the *depth* of the network, also plays a major role in determining the complexity of the CNN.) The data representation at the penultimate dense layer can be viewed and treated as features, in a manner analogous to the features that are extracted – from original image data, for use as inputs – for a shallow classifier. In essence, what a CNN does is learn a transformation to map an input image into a new representation space in which the classes are easily separable.

During the training process, the model seeks to minimize an objective (or "loss") function expressing the classification error on the training data under consideration. At each training iteration, the model parameters are updated by some variant of gradient descent in order to get closer to this objective. Because there can be huge numbers of free model parameters to be learned, it is necessary to have a correspondingly large set of training data to avoid overfitting. In turn, training a CNN "from scratch" can take considerable time, even with high-throughput computational resources like graphics processing units (GPUs). Since the amount of data is always increasing (*e.g.,* with each data-collection survey), CNNs are a convenient way to leverage and exploit the fruits of these investments fully.

Designing a CNN architecture is often considered an art form, and domain expertise is vital to creating successful networks. Assuming a standard "vanilla" architecture, the key attributes that a designer must decide are the number of convolutional layers to employ, the number of

[1]The fully-connected layers' weights are also learned during training, but these can be treated as special convolutional filters of size 1×1.

filters to be learned in each convolutional layer, and the sizes of those filters. The designer must also decide where in the architecture to place pooling layers (which encourage translation invariance), as well as the pooling factors to use in those layers, and the manner in which the pooling is performed (*e.g.,* taking an average or the maximum value in each region). The size of the input data, the manner in which the input data is normalized, the number of nodes in the dense layer, and the mathematical form of the activation functions must also be specified. The training process entails several additional choices, including the exact form of the objective function (which will not be convex), the optimization algorithm and various parameter settings therein (such as learning rate and batch size), and data augmentation strategies to employ, among other things. Classification performance can be very sensitive to many of these design choices.

3.3. MATHEMATICAL DETAILS

Having given this qualitative overview of CNNs, we now define the most important quantities more precisely for the binary classification case with single-channel input images. The following presents the equations that govern the operations in each layer of a CNN, parameter training, and how one obtains final classification predictions for an input image.

Let $\boldsymbol{x}$ be an input image of an object associated with class label $y \in \{0, 1\}$, where $y = 1$ indicates a target and $y = 0$ indicates clutter.

Let $\boldsymbol{a}_i^{(\ell-1)}$ be the ith feature map in the $(\ell - 1)$th convolutional layer. Let $\boldsymbol{w}_{ij}^{(\ell)}$ denote the convolutional kernel connecting the ith input feature map to the jth output feature map in the ℓth convolutional layer, and let $b_j^{(\ell)}$ denote the bias in the jth feature map in the ℓth convolutional layer.

The response of the jth output feature map at the ℓth convolutional layer is obtained as

$$\boldsymbol{z}_j^{(\ell)} = \sum_i \boldsymbol{a}_i^{(\ell-1)} * \boldsymbol{w}_{ij}^{(\ell)} + b_j^{(\ell)} \tag{3.1}$$

$$\boldsymbol{a}_j^{(\ell)} = \sigma(\boldsymbol{z}_j^{(\ell)}), \tag{3.2}$$

where $*$ denotes the convolution operation, and $\sigma(\cdot)$ is the nonlinear activation function. At the CNN input, there is only one feature map: $\boldsymbol{a}_i^{(0)} = \boldsymbol{x}$.

Two nonlinear activation functions used in this work are the sigmoid function and the rectified linear unit (ReLU), which are given by

$$\sigma(\boldsymbol{z}(r, c)) = (1 + \exp\{-\boldsymbol{z}(r, c)\})^{-1} \tag{3.3}$$

$$\sigma(\boldsymbol{z}(r, c)) = \max\{0, \boldsymbol{z}(r, c)\}, \tag{3.4}$$

respectively; (r, c) specifies a row and column position in a feature map.

Pooling layers reduce the spatial size of feature maps in accordance with pooling factors, f_R and f_C, specified for each dimension. Two common types of pooling are average pooling and max pooling. In average pooling, a sub-region of a feature map is mapped to its average value,

$$\boldsymbol{a}_j^{(\ell)}(r, c) = \frac{1}{f_R f_C} \sum_{u=1}^{f_R} \sum_{v=1}^{f_C} \boldsymbol{a}_j^{(\ell)}(r + u - \lceil f_R/2 \rceil, c + v - \lceil f_C/2 \rceil), \tag{3.5}$$

where $\lceil \cdot \rceil$ is the mathematical ceiling. In max pooling, a sub-region of a feature map is mapped to its maximum value,

$$\boldsymbol{a}_j^{(\ell)}(r, c) = \max_{1 \le u \le f_R, 1 \le v \le f_C} \boldsymbol{a}_j^{(\ell)}(r + u - \lceil f_R/2 \rceil, c + v - \lceil f_C/2 \rceil). \tag{3.6}$$

At the final output layer L of the CNN, the softmax function gives the probability of the input image $\boldsymbol{x}$ belonging to each class y_i,

$$p(y_i|\boldsymbol{z}^{(L)}) = \frac{\exp\left\{z_i^{(L)}\right\}}{\sum_{j=0}^{1}\exp\left\{z_j^{(L)}\right\}}. \tag{3.7}$$

The binary-cross-entropy loss function for one training sample, $\{\boldsymbol{x}; y\}$, is defined as

$$J(\boldsymbol{w}, \boldsymbol{b}) = -y\log p(y|\boldsymbol{z}^{(L)}; \boldsymbol{w}, \boldsymbol{b}) - (1-y)\log(1 - p(y|\boldsymbol{z}^{(L)}; \boldsymbol{w}, \boldsymbol{b})). \tag{3.8}$$

When a "mini-batch" of B training samples is employed, the loss function simply becomes the mean loss over the B samples.

Mini-batch gradient descent updates the parameter values according to

$$\boldsymbol{w} = \boldsymbol{w} - \eta\frac{\partial J(\boldsymbol{w}, \boldsymbol{b})}{\partial \boldsymbol{w}} \tag{3.9}$$

$$\boldsymbol{b} = \boldsymbol{b} - \eta\frac{\partial J(\boldsymbol{w}, \boldsymbol{b})}{\partial \boldsymbol{b}}, \tag{3.10}$$

where η is the learning rate that controls the size of the update step, and $J(\boldsymbol{w}, \boldsymbol{b})$ is the loss function of the mini-batch. At each iteration of the training procedure, a new mini-batch of training samples is randomly selected.

The RMSprop optimizer [24] is simply a "fancy" way to make the learning rate adaptive; it normalizes the learning rate for a parameter by a running average of the magnitudes of recent gradients for that parameter.

The backpropagation algorithm [25], which relies on the chain rule from calculus, is used to compute the necessary partial derivatives in a CNN in a computationally efficient manner. The details are beyond the scope of this report.

Chapter 4

Experiments With High-Frequency Sonar

In this chapter, we describe several CNN experiments conducted with high-frequency sonar data. First, in Sec. 4.1, we describe a data set that is used across multiple experiments. Then two forms of transfer-learning are studied in Sec. 4.2. Sec. 4.3 describes constructing CNNs with an unconventional data representation, namely phase imagery. Then the use of multiple representations is examined in Sec. 4.4.

4.1. DATA

Synthetic aperture sonar (SAS) [26] 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. A large database of scene-level SAS images, each of which typically spans 50 m in the along-track direction and 110 m in the range direction, had been collected by CMRE's MUSCLE AUV during thirteen 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. The imagery has an along-track resolution of 2.5 cm and a range resolution of 1.5 cm.

Prior to the surveys, mine-like object shapes including cylinders, truncated cones, wedges, and other man-made objects had been purposely deployed. This allowed detailed target ground-truth information to be formulated. The data of the different expeditions vary greatly in terms of seafloor composition and cover (*e.g.,* sand, mud, vegetation), clutter types and densities, target types, image quality, and environmental complexity. Data from the eight oldest expeditions were treated as training data for learning CNNs in various experiments, while the data from the five most recent expeditions were reserved for use as *distinct* test sets (owing to their diverse characteristics). A summary of the MUSCLE data surveys associated with the potential test data sets is given in Table 4.1.

Table 4.1. MUSCLE test data surveys

Data Set Code	Name of Sea Experiment	Survey Dates (months / year)	Survey Location	Survey Area (km^2)
MAN2	MANEX	9-10 / 2014	Bonassola, Italy	38.9
NSM1	NSMEX	5 / 2015	Ostend, Belgium	22.6
TJM1	TJMEX	10 / 2015	Cartagena, Spain	55.5
ONM1	ONMEX	9 / 2016	Hyères, France	43.8
GAM1	GAMEX	3-4 / 2017	Patras, Greece	19.4

The Mondrian detection algorithm [5] was applied to all complex scene-level sonar images in the database.[1] This algorithm is a flexible, general-purpose object-detection method that can reliably detect objects over the wide range of potential UXO sizes and shapes. Importantly, the approach is robust even when the object response does not have a strong (acoustic) shadow associated with it, a case that arises with partially buried objects.

The detector results in a set of image "chips" of objects to be classified as targets (class 1) or clutter (class 0) by the CNN classifiers; each chip spans 5.025 m $\times$ 5.025 m, larger than the requisite CNN input size, to facilitate data augmentation. An example detection result on a SAS image scene collected by the MUSCLE AUV is shown in Fig. 4.1.

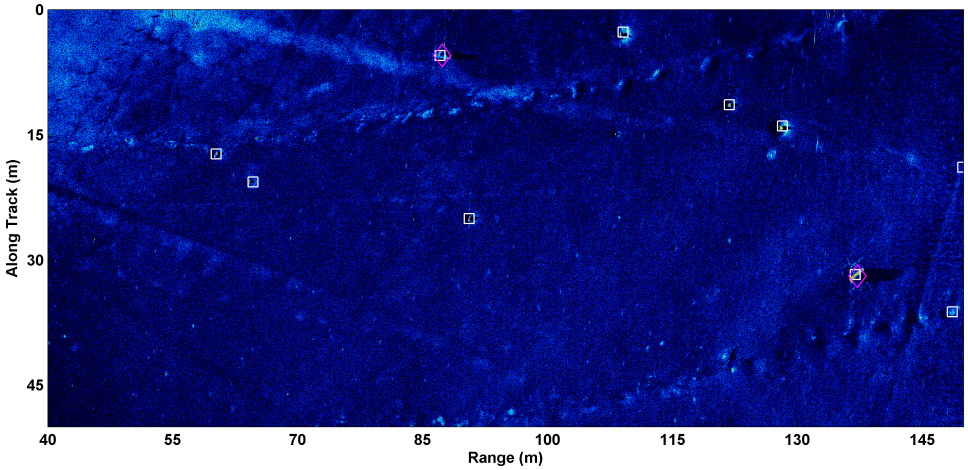


Fig. 4.1. Typical MUSCLE SAS image scene with alarms generated by the Mondrian detection algorithm marked by white boxes and true ground-truth targets marked by magenta diamonds.

All CNNs designed in the experiments in this chapter share the following characteristics. The image chips (or "images" derived from them) are the inputs to the CNNs. Each data representation input to a CNN is assumed to be 267 pixels $\times$ 267 pixels. For SAS magnitude and phase imagery, each pixel spans 1.5 cm in each dimension. All pooling layers use average pooling, rather than max pooling, because the former approach has been observed [27] to better handle the speckle phenomenon that characterizes sonar imagery. The outputs of the CNN's final layer are the (softmax) probabilities of an image belonging to each class (target or clutter). For CNN training, no attempt was made to optimize the learning rate or batch size.

Our main interest in this chapter is to examine the ability of the CNN-based approach to successfully perform *classification*. Therefore, when presenting receiver operating characteristic (ROC) curves, the experiments assume that all targets were successfully flagged in the detection stage. Thus, the maximum possible area under the ROC curve (AUC) [28], a scalar summary measure of performance, is always unity. (A perfect classifier would have an AUC of unity.) To obtain the overall probability of both successful detection and correct classification, the vertical axis of the ROC curves would need to be scaled by the proportion of targets actually found in the detection stage. Finally, a note about nomenclature: On all figures showing ROC curves, the term "probability of detection" is used in the classical signal-detection-theory sense [29] of discriminating a target from clutter; thus, the quantity being measured is in fact the performance of the *classification* stage.

[1] While the experiments recounted in this chapter were being conducted, the Mondrian detection algorithm was still in development. Therefore, different versions of the algorithm (and different parameter settings) were used for the experiments in each main section within this chapter. For this reason, the number of images in the data sets varies; performance comparisons should not be made across experiments in separate sections.

4.2. TRANSFER LEARNING

In remote-sensing applications, data collections can be time-consuming and prohibitively expensive. Moreover, when a new sensor is introduced, there is a desire to be able to still leverage historical data collected by similar predecessor systems. In general, it is not feasible to wait for the execution of numerous onerous data collections before being able to accurately assess the utility of the new system. For these reasons, the idea of transfer learning is appealing.

In the context of CNNs, transfer learning entails taking a CNN trained for one task and re-using it for a related but distinct task. The basic idea exploited in this transfer is that CNN filters that capture lower-level features, such as edges and corners, are useful in a wide range of object-recognition tasks. As a result, one can intelligently initialize (early layers of) a CNN with weights learned for a different task, and then refine with a modest amount of data related to the new task at hand. This approach greatly reduces training time, but also lightens training-data requirements.

Our objective is to demonstrate that data from multiple similar sonar sensors can be collectively exploited to address related automatic target recognition (ATR) tasks. We previously developed a set of CNNs for mine classification tasks using MUSCLE SAS data. Leveraging this work, we here demonstrate the feasibility of two types of transfer learning for the task of underwater object classification: target-concept transfer and sensor transfer. Specifically, we modify the target concept of the networks, from mines to UXO (or UXO proxies), so that the objective is to successfully discriminate UXO – rather than mines – from clutter. The second type of CNN transfer learning we demonstrate involves transfer between sensors: training a CNN using SAS data from one sensor, and adapting it to enable inference on SAS data from a different sensor, operating at a different frequency band.

4.2.1. CNNs for Transfer Learning

In this study, we leverage six CNNs previously trained for a mine recognition task. The input data for each CNN is a 4.005 m $\times$ 4.005 m SAS magnitude chip. The common architecture is a sequence of alternating convolutional layers and pooling layers, followed by a single fully-connected dense layer, and then the output (prediction). Each CNN is distinguished by the number of convolutional layers, the number and size of the filters, and the pooling factors employed. Sigmoid activation functions are used. The details of these CNN architectures, and the total number of free parameters each CNN must learn, are summarized in Table 4.2.

Table 4.2. Previously-developed CNN architectures for magnitude imagery

CNN Label	Convolutional Layers	Filters Per Layer	Filter Sizes (Pixels Per Side)	Pooling Factors	Number of Parameters
A	3	8 10 12	16 8 5	4 4 2	10800
B	3	8 10 12	8 6 6	4 4 2	8344
C	4	2 3 4 5	18 16 14 12	2 2 2 2	7682
D	5	4 6 8 10 12	8 7 7 5 3	2 2 2 2 2	7506
E	5	10 10 10 10 10	4 3 4 4 3	2 2 2 2 2	5712
F	6	10 10 10 10 10 10	2 2 3 3 2 2	2 2 2 2 2 2	3602

4.2.2. Target-Concept Transfer-Learning

We first explore the idea of target-concept transfer-learning by altering the classes of objects considered targets and clutter. The objective is to refine already-trained CNNs so that they properly discriminate the new object classes without the burden of training the networks "from scratch."

4.2.2.1. Experimental Set-Up

Previously, SAS data collected by the MUSCLE AUV during eight sea experiments had been used to train six CNNs in which the target class consisted of mine-like object shapes including cylinders, truncated cones, wedges, and other man-made objects. All other alarms were assigned to the clutter class.

For these transfer-learning experiments, MUSCLE SAS data from three different sea experiments (not used in the earlier training process) were considered as test data. To effect a transfer-learning scenario, the target class was modified to consist of only cylindrical objects (as a surrogate for UXO, which typically takes this approximate shape). All other objects (including the objects previously treated as targets) were considered to belong to the clutter class. A summary of the data set details before and after this "relabeling" procedure is shown in Table 4.3.

Table 4.3. Data set details of target-concept transfer-learning experiments

Data Set	Before Relabeling		After Relabeling	
	Number of		Number of	
	Targets	Clutter	Targets	Clutter
Training Data	2672	15165	553	17284
TJM1	–	–	59	2350
ONM1	–	–	11	295
GAM1	–	–	25	338

With the training data "relabeled" in this manner, CNN refinement was undertaken to update the previously-trained parameters. Specifically, CNN re-training was performed using mini-batch gradient descent with a learning rate of $\eta = 0.001$, in conjunction with a binary-cross-entropy loss function. A batch size of $B = 64$ 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; to this end, the samples for each class's half-batch were chosen from a random pool of size $0.6B$, which allowed an element of stochasticity to be retained. Data augmentation that respected the inviolable geometry of the sonar data-collection procedure was employed during training; this meant a random range translation $i_{tx} \in [0, 0.5 \text{ m}]$, along-track translation $i_{ty} \in [-0.5 \text{ m}, 0.5 \text{ m}]$, and along-track reflection $i_{ry} \in \{0, 1\}$ was applied to each sonar image chip selected for the batch.

Transfer-learning refinement was executed for 2000 epochs, where one epoch was defined to correspond to an update from a single batch, not a full pass through the entire data set. (Defining an epoch on this smaller "timescale" allows changes to be monitored at a finer level.) For the first 1000 epochs of re-training (equivalent to approximately 3 full passes through the data set), the clutter class data were constrained to be drawn from the subset of objects that were

treated as targets in the original training, but as clutter in the refinement phase. For the second 1000 epochs, the clutter class data were allowed to be drawn from all alarms assigned a clutter label for the re-training. This procedure was undertaken to examine whether it was sufficient to focus the refinement on "un-learning" the newly labeled clutter cases, or if considering the full set of clutter cases was necessary.

4.2.2.2. Experimental Results

The results of the target-concept transfer-learning experiments are summarized in Tables 4.4-4.6 in terms of the AUC. Specifically, the AUC is shown for each of the six CNNs considered, for each of the three test data sets, at epoch 0 (*i.e.,* before any refinement had transpired), at epoch 1000, and at epoch 2000.

Table 4.4. Target-concept transfer results on TJM1 data set

CNN Label	AUC at Epoch 0	AUC at Epoch 1000	AUC at Epoch 2000
A	0.825	0.659	0.984
B	0.841	0.846	0.975
C	0.839	0.878	0.943
D	0.845	0.957	0.966
E	0.832	0.826	0.962
F	0.842	0.825	0.961

Table 4.5. Target-concept transfer results on ONM1 data set

CNN Label	AUC at Epoch 0	AUC at Epoch 1000	AUC at Epoch 2000
A	0.774	0.737	0.898
B	0.799	0.782	0.939
C	0.694	0.870	0.990
D	0.738	0.844	0.915
E	0.797	0.674	0.932
F	0.709	0.773	0.904

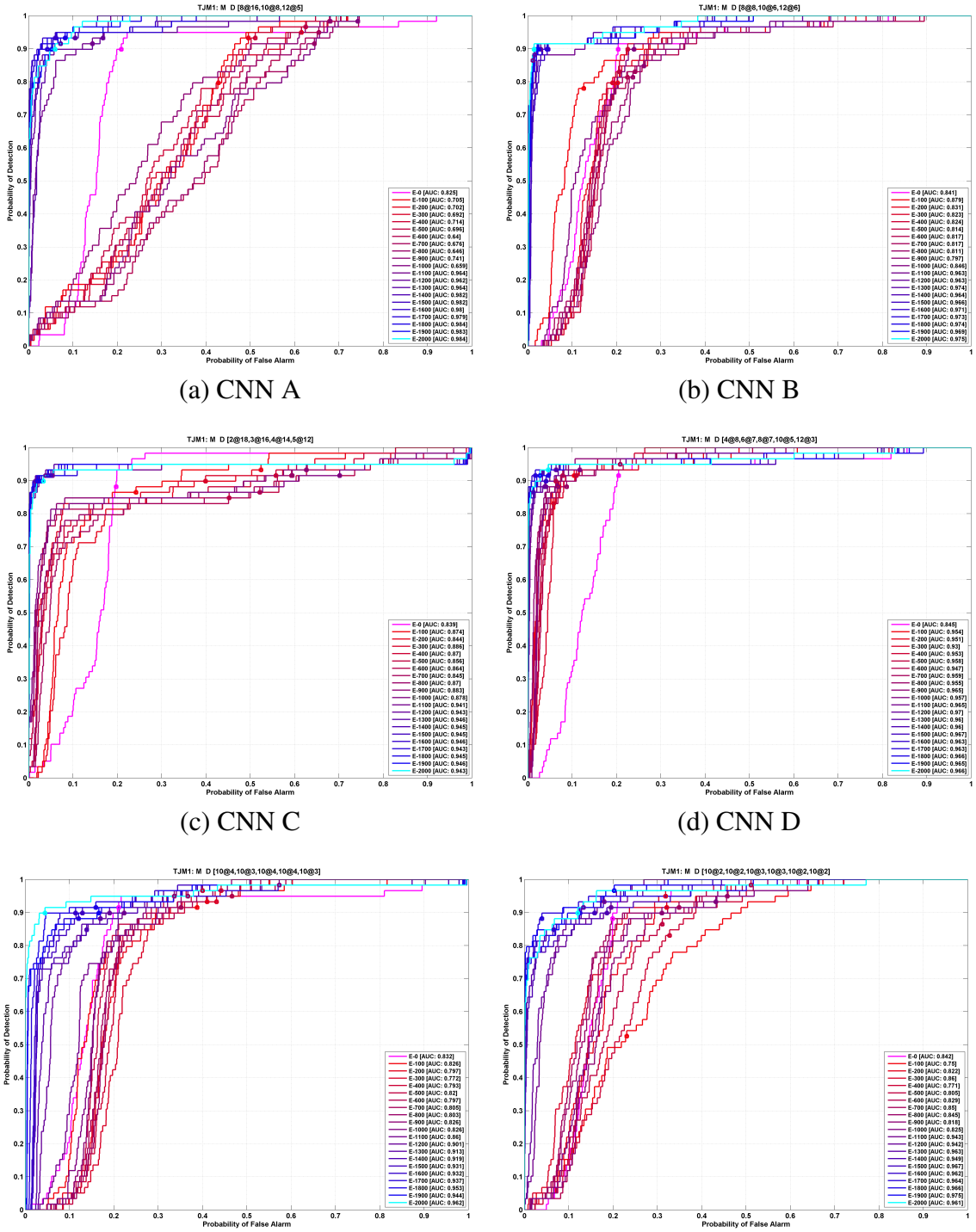
Table 4.6. Target-concept transfer results on GAM1 data set

CNN Label	AUC at Epoch 0	AUC at Epoch 1000	AUC at Epoch 2000
A	0.633	0.829	0.851
B	0.467	0.788	0.840
C	0.398	0.753	0.732
D	0.479	0.842	0.837
E	0.496	0.780	0.748
F	0.554	0.836	0.791

As can be seen from the tables, before refinement has commenced, performance is quite poor because the CNNs had been trained to treat certain objects as targets, which were then considered as clutter during the test phase. Upon refining the CNN parameters with the new labeling rubric, performance improved dramatically. It can be noted that considering the full set of clutter cases was needed to achieve the best performance, indicating that "un-learning" the altered-label cases alone was insufficient.

This phenomenon can also be observed in Fig. 4.2, which shows the evolution of the performance in terms of full ROC curves as a function of refinement epoch, for the six CNNs, on the TJM1 test data set. In fact, there is markedly different behavior during the first stage of refinement (*i.e.,* epochs 1-1000) for the different CNNs. For example, compared to the case of no refinement (*i.e.,* epoch 0), CNN A performs worse, CNN B performs about the same, CNN C performs better at low false alarm rates but worse at high false alarm rates, and CNN D performs better. But during the second stage of refinement (*i.e.,* epochs 1001-2000), when both "regular" clutter and the newly "relabeled" clutter are included as class 0 training data, all of the CNN classifiers become superior.

The results of these experiments show that target-concept transfer is indeed feasible. The valuable implication is that one can start with a CNN trained for an MCM task, and simply refine with a modest amount of UXO data to effectively transfer the network to the new application. This leveraging of related data can effect cost savings by reducing the amount of (UXO) training data, and hence expensive data surveys, required.



(a) CNN A (b) CNN B

(c) CNN C (d) CNN D

(e) CNN E (f) CNN F

Fig. 4.2. Performance on the TJM1 test data set of CNNs A-F in (a)-(f) for the target-concept transfer-learning experiments. The initial performance before any refinement is shown in magenta; the final performance after 2000 epochs of refinement is shown in cyan. In between, the performance curves progress from red to blue with increasing epoch. (Zoom on the electronic version to read the legend and axes.)

4.2.3. Sensor Transfer-Learning

In this next study, we conduct experiments on the topic of sensor transfer-learning. The ultimate objective is to discriminate UXO from clutter, but very limited training data – and specifically target views – are available. This challenge is addressed by making use of two databases of imagery, one collected by the MUSCLE AUV and the other collected by the SeaOtter Mk II AUV. The idea is to begin with CNNs that were trained using MUSCLE data, and refine them using a small amount of data from the SeaOtter, so that the networks can properly classify test data from the new sensor. This approach would eliminate the need for a large set of labeled training data from the new sensor, as would otherwise be required if training the networks "from scratch."

4.2.3.1. Experimental Set-Up

The new database considered here consisted of 476 scene-level SAS images, collectively spanning approximately 2.61 km^2 of seabed, that was collected by the SeaOtter Mk II AUV during a sea expedition conducted in September 2016 in German waters of the Baltic Sea. The center frequency of the SAS is 150 kHz and the bandwidth is 30 kHz. The imagery has an along-track resolution of 2.0 cm and a range resolution of 2.62 cm. The SAS data in this database differs from the MUSCLE database in several notable ways, including the target classes, the types of clutter, seafloor characteristics, and image resolution.

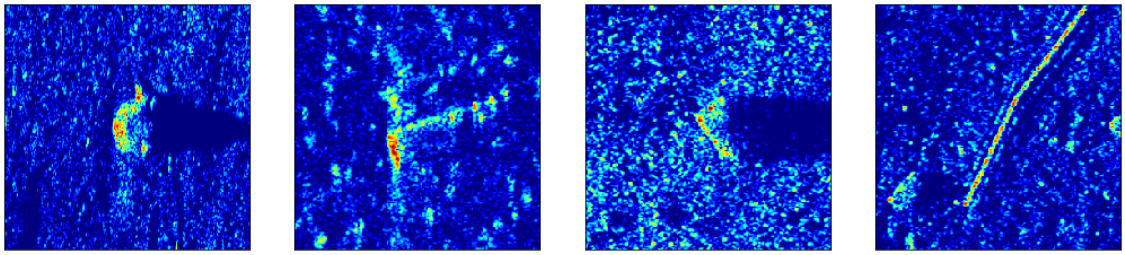
The Mondrian detection algorithm was applied to the scene-level sonar images from the SeaOtter database, with this resulting in sets of image "chips" of objects to be classified as targets or clutter by the CNNs. The survey data was collected in an area that is known to contain real, historical UXO present from World War II, but no proper ground truth was available. Therefore, we manually labeled all of the candidate alarms, assigning suspected UXO to the target class and all other alarms to the clutter class. (Although the resultant classification rates attained may be inaccurate, this approach will still fairly assess the potential of sensor transfer-learning.)

To form relatively balanced training and test data sets from the SeaOtter data, the alarms generated from the port sonar were treated as training data, while the alarms generated from the starboard sonar were treated as test data. Details of the data sets used in these experiments are shown in Table 4.7.

Table 4.7. Details of the data sets for sensor transfer-learning experiments

Sensor Platform	Data Set	Target Class	Number of Targets	Number of Clutter
MUSCLE	Training	Mines	2672	15165
SeaOtter	Training	UXO	65	13746
SeaOtter	Test	UXO	73	15324

Example SAS imagery of targets from the two databases are shown in Fig. 4.3, where the similarities and differences of the data can be observed. The SeaOtter target examples highlight the diversity of object shapes and unreliability of acoustic shadows for detection purposes.



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

Fig. 4.3. SAS imagery of representative targets from (a) the MUSCLE data set and (b)-(d) the SeaOtter data set. Each image spans approximately 5 m $\times$ 5 m; range increases to the right.

We again begin with the six CNNs that had been trained using SAS data collected by the MUSCLE AUV during eight sea experiments, as in Sec. 4.2.2. But then transfer learning is effected by refining these CNNs with the training data set from the SeaOtter AUV. This refinement used the same learning rate, batch size, and data augmentation techniques as in Sec. 4.2.2. To create consistent input data for the CNNs, the slightly lower resolution SeaOtter image chips were re-sampled (via bilinear interpolation) to match the pixel size of the previous MUSCLE training data (*i.e.,* a resolution of 1.5 cm $\times$ 1.5 cm). The same image normalization procedure that had been applied to MUSCLE data was also applied to the SeaOtter data. Transfer-learning refinement was executed for 2000 epochs, where one epoch is again defined to correspond to an update from a single batch, not a full pass through the entire data set.

4.2.3.2. Experimental Results

The results of the sensor transfer-learning experiments are summarized in Table 4.8 in terms of the AUC. Specifically, the AUC is shown for each of the six CNNs considered for the SeaOtter test data set, before any refinement had transpired (*i.e.,* at epoch 0) and with refinement (at epoch 2000).

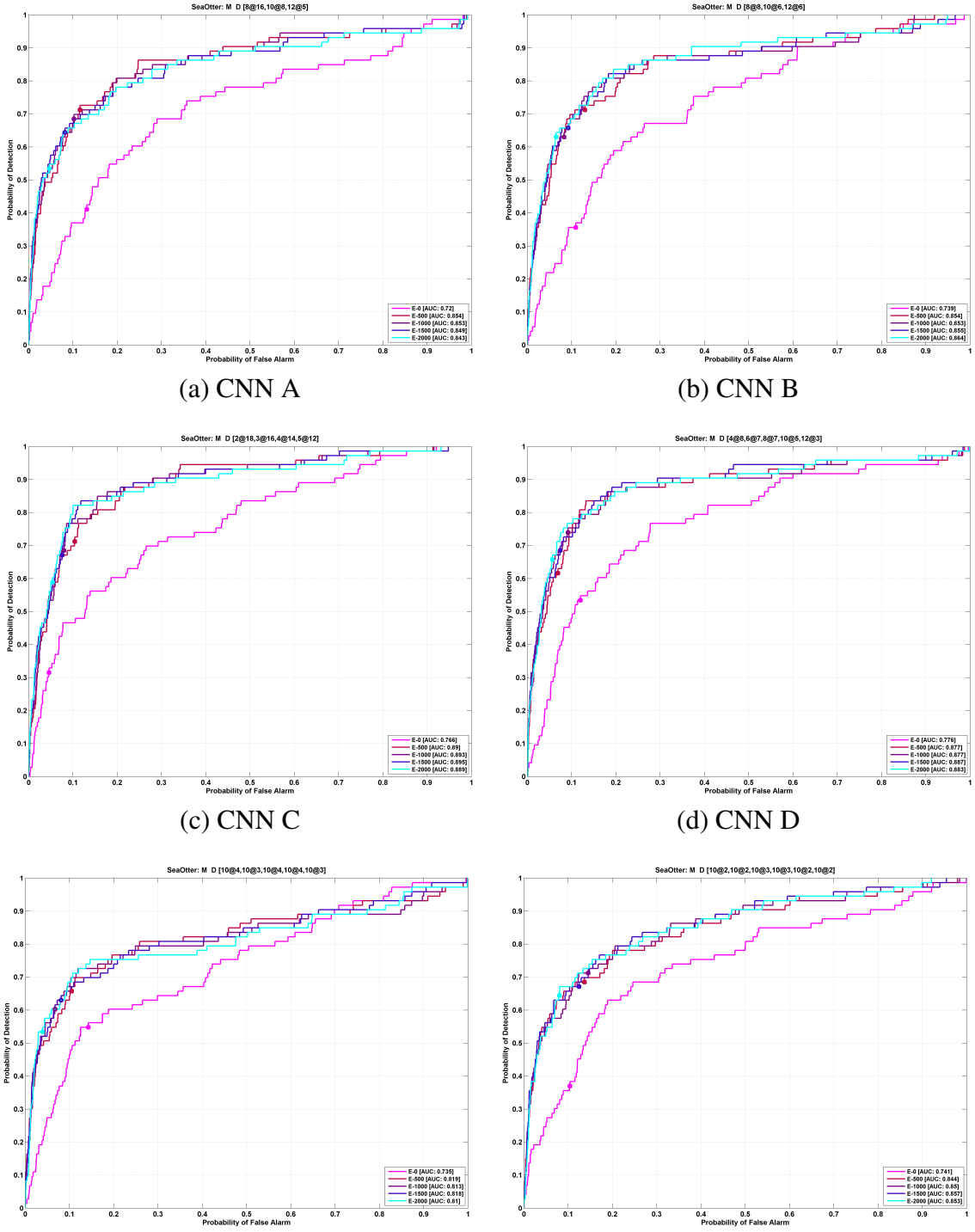
Table 4.8. Sensor transfer results on SeaOtter data set

CNN Label	AUC	
	Without Refinement	With Refinement
A	0.720	0.843
B	0.739	0.864
C	0.766	0.889
D	0.776	0.883
E	0.735	0.810
F	0.741	0.853

Before refinement, the CNNs are still tailored to both the general characteristics of the MUSCLE sensor data and the specific target types (*i.e.,* surrogate mine shapes) used for training. As a result, classification performance on the SeaOtter test data is relatively weak. However, it should be noted that performance is still well above the chance diagonal, suggesting that the CNNs are relatively robust and that the tasks are strongly related.

As the CNNs are exposed to more of the SeaOtter training data during the refinement process, performance improves. This result can be observed in Table 4.8, as well as in Fig. 4.4,

which shows the evolution of the performance in terms of full ROC curves as a function of refinement epoch. In fact, it can be seen that the refinement had more or less converged even after only 500 epochs (equivalent to about 2-3 full passes through the training data set).



(a) CNN A (b) CNN B

(c) CNN C (d) CNN D

(e) CNN E (f) CNN F

Fig. 4.4. Performance on the SeaOtter test data set of CNNs A-F in (a)-(f) for the sensor transfer-learning experiments. The initial performance before any refinement is shown in magenta; the final performance after 2000 epochs of refinement is shown in cyan. In between, the performance curves progress from red to blue with increasing epoch. (Zoom on the electronic version to read the legend and axes.)

Interestingly, relatively small amounts of SeaOtter training data are required to improve performance considerably. For example, only 65 examples of the new target class are available during the refinement process. This underscores the promise of the transfer-learning approach, and the minimal labeled-data requirements involved. Instead, one of the primary factors for success is that the test data and training data of the new sensor are characterized by the same conditions, and drawn from the same underlying statistical distribution.

4.2.4. Summary

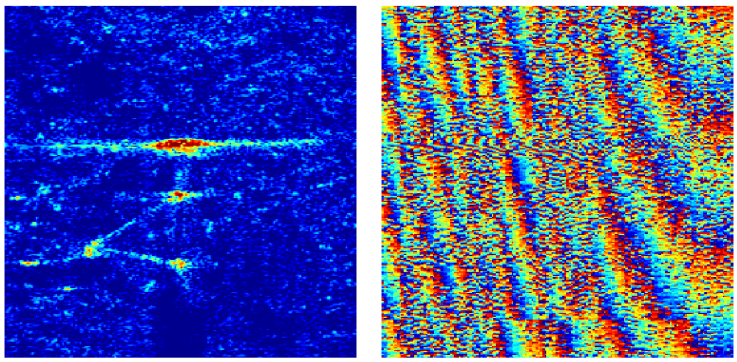
These experiments demonstrated the feasibility of two types of transfer learning for the task of underwater object classification: target-concept transfer and sensor transfer. This transfer learning leveraged CNNs trained for a mine classification task with data from one SAS system, and performed parameter refinement using a small amount of data related to the new task. The success of the transfer learning suggests that CNNs developed on MUSCLE data can be exploited for use with data from other side-looking sonar systems that operate at different (but similar) frequency bands and produce imagery at a slightly different resolution; application to similar yet distinct classification tasks, such as a different target class, is also feasible. In turn, powerful classifiers can be developed relatively quickly with minimal labeled data, thereby reducing the amount of costly data surveys needed with the new sensor, as well as the computation time needed to train a robust classifier.

4.3. ALTERNATIVE REPRESENTATIONS: PHASE

SAS imagery is complex-valued, but typically this data is converted to a (real-valued) magnitude representation that is subsequently used for various signal processing and pattern recognition tasks, such as object classification. The phase information, which is related to the signal travel time, and in turn, the distance traveled, is usually discarded. In this study, we demonstrate that the phase data in SAS imagery contains useful information that can be exploited *on its own* for object classification tasks. This finding upends the conventional wisdom that the phase does not contain useful information for classification.

Complex SAS data is often manipulated to serve various purposes, but seldom is the phase information considered in isolation. (A notable exception is with bathymetric estimation via interferometry [30], where phase differences are exploited to obtain relative height information.) For example, transforming complex data into the Fourier domain enables efficient subband [31, 32] and sub-aperture processing [33, 34] and, with lower-frequency data, the formation of acoustic color representations [35]. But the general view of SAS phase imagery – as a signal with no worthwhile content – is an assumption we challenge.

An example SAS "chip" of an object – an endfire cylinder, with deployment chains attached – is shown in Fig. 4.5. Specifically, both the magnitude and phase images are shown. From visual inspection, it is obvious that the magnitude image contains useful information for classification. Less clear is whether the phase image also contains features or characteristics that can be exploited reliably. The objective of this work is to establish that phase imagery like this does indeed contain information. In order to demonstrate this, we rely on CNNs because of their ability to automatically uncover useful clues for classification via the learned filters. This aspect of CNNs is particularly attractive because, as Fig. 4.5(b) suggests, it is challenging for a human to hand-craft salient features for extraction from phase imagery.



(a) (b)

Fig. 4.5. An object's SAS (a) magnitude image and the corresponding (b) phase image. Range increases down the page.

4.3.1. CNNs for SAS Phase Imagery

In this study, we develop three unique CNNs. The input data for each CNN is a 4.005 m $\times$ 4.005 m SAS *phase* chip. The common architecture is a sequence of alternating convolutional layers and pooling layers, followed by a single fully-connected dense layer, and then the output (prediction). Each CNN is distinguished by the number of convolutional layers, the numbers and

sizes of the filters, and the pooling factors employed. Sigmoid activation functions are used. The details of these CNN architectures are summarized in Table 4.9.

Table 4.9. CNN architectures for SAS phase imagery

CNN Label	Filters Per Layer	Filter Sizes (Pixels Per Side)	Pooling Factors	Number of Parameters
A	8 10 12	16 8 5	4 4 2	10800
B	8 10 12	8 6 6	4 4 2	8344
C	4 6 8 10 12	8 7 7 5 3	2 2 2 2 2	7506

To train these CNNs, the MUSCLE training database comprising eight sea expeditions was used. Testing was then performed using the disjoint set of SAS data from five other sea expeditions. Basic details of these test sets are summarized in Table 4.10.

Table 4.10. Test data set details for phase experiments

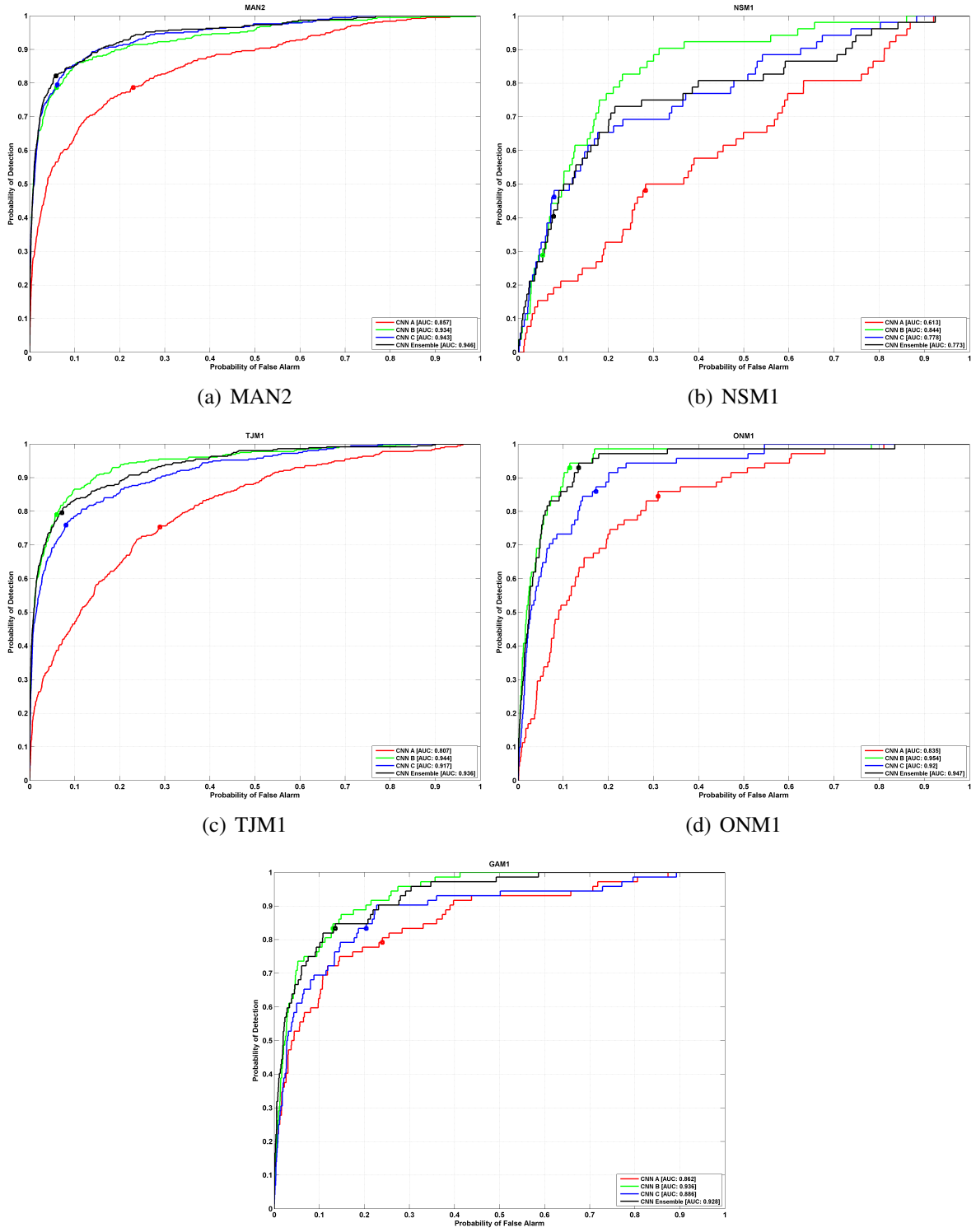
Data Set Code	Number of	
	Targets	Clutter
MAN2	375	77222
NSM1	52	46832
TJM1	357	43847
ONM1	71	23366
GAM1	72	4058

4.3.2. Experimental Results

The results of making (class) predictions using the three CNNs for data from the five test sets are shown in Fig. 4.6, where it can be seen that the classification performance is well above the chance diagonal (in which every prediction is a random coin flip). This result provides strong evidence that there is indeed exploitable classification information contained in the phase images alone. The AUC associated with Fig. 4.6 is also shown in Table 4.11.

Table 4.11. Classification performance for phase experiments

Data Set Code	AUC			
	CNN A	CNN B	CNN C	Ensemble
MAN2	0.857	0.934	0.943	0.946
NSM1	0.613	0.844	0.778	0.773
TJM1	0.807	0.944	0.917	0.936
ONM1	0.835	0.954	0.920	0.947
GAM1	0.862	0.936	0.886	0.928



(a) MAN2 (b) NSM1

(c) TJM1 (d) ONM1

(e) GAM1

Fig. 4.6. Classification performance using SAS phase imagery, on five test data sets with three different CNNs and the ensemble. The operating point corresponding to a prediction threshold of 0.5 is marked, by a circle, on each curve.

Interestingly, classification performance was markedly worse on the NSM1 data set. This data was collected in the North Sea where there were very strong currents. As a result, the sonar-equipped AUV was rarely able to maintain an ideal linear trajectory during data collection, and SAS processing image-formation was more challenging. The *magnitude* imagery from this data set is often blurry, and shadows cast by objects are typically not well-defined. We hypothesize that this factor also causes the phase imagery to lack strong structure in the shadows, thereby eliminating exploitable classification clues.

It can also be noted that the only difference between CNN A and CNN B is the filter size, where CNN B uses smaller filters. As can be seen in the results, CNN B outperforms CNN A, providing mild evidence that smaller filters are preferable. There are also fewer parameters to learn, which makes training easier (*i.e.,* faster). These insights will inform our design of new networks moving forward.

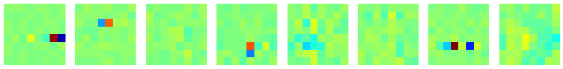
4.3.3. Analysis

We seek to better understand the reasons for the classification success on the phase imagery. Because CNN B consistently achieved the best classification performance, we investigate its filters in more detail. Fig. 4.7 shows the learned filters of the three convolutional layers of CNN B. (For a given convolutional layer, each filter uses an identical color scale in which the color green corresponds to zero, warmer colors are positively valued, and cooler colors are negatively valued.)

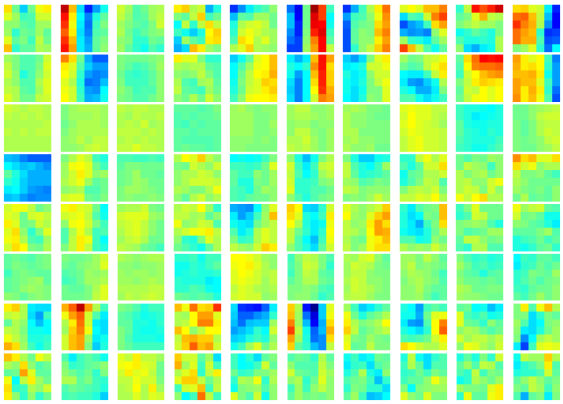
From the figure, it can be observed that many of the filters are essentially zero-valued, meaning they are effectively "dead nodes." That is, pruning these filters from the network would have no impact on predictions. This insight suggests that more efficient CNNs could be constructed in which fewer filters are employed in each layer. This modification would also reduce training time. Thus, moving forward, we will design new CNNs with this result in mind.

It can also be seen that the purpose of the first convolutional layer's filters is ostensibly to locate vertical or horizontal gradients in the input phase imagery. This insight should be contrasted with the finding in [4], which studied CNNs for which the input imagery was SAS *magnitude* images. In that work, it was discovered that the first convolutional layer's filters were effectively acting as bandpass filters (vis-à-vis pixel values) to segment the images into highlight, shadow, and background regions.

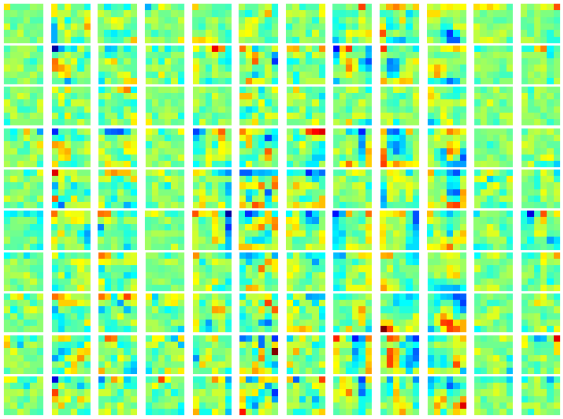
More complicated structural features in the imagery can be isolated when multiple convolutional layers (with nonlinear activation functions) are nested. After the convolutional and pooling layers, the three-dimensional output tensor is "flattened" into a vector to accommodate a fully-connected layer. However, the vector of weights of the fully-connected layer can be reshaped to recover the spatial form destroyed by the flattening. These weights of CNN B are shown in Fig. 4.8. When presented in this format, one can associate and visualize the relative importance of each spatial region, or "receptive field," of a generic input image. For this CNN, it can be seen that there is not a single dominant region that drives all predictions. Rather, various components will, in general, have the capacity to influence the final class predictions.



(a)



(b)



(c)

Fig. 4.7. For CNN B, the filters of the (a) first convolutional layer, (b) second convolutional layer, and (c) third convolutional layer. (Three-dimensional filters in the latter two layers are grouped columnwise.)

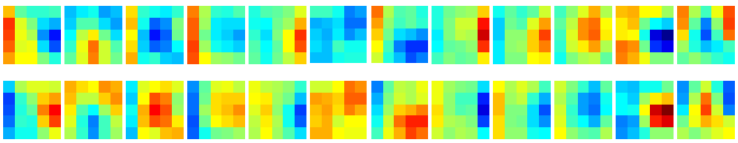


Fig. 4.8. For CNN B, the weights of the fully-connected layer, displayed such that spatial form is retained (*i.e.,* the "flattening" step has been inverted). N.B. For display purposes, the weights have been rotated counterclockwise 90° so that range now increases to the right.

Next, we examine some intermediate representations of CNN B for a few specific phase images from the test set. Specifically, we choose to show the intermediate representation of the imagery at the second convolutional layer (prior to evaluation with the sigmoid activation function) because this representation seems to contain the most interpretable structure. In a sense, two convolutional layers are required to transform the phase imagery into a form that humans can readily comprehend. This should be contrasted with a SAS *magnitude* image, which requires no transformations (*i.e.,* convolutional layers) to be understandable.

We also show the contributions of each spatial location (receptive field) to the final classification predictions. These contributions are the result of inner products between the fully-connected layer's weights and the image responses input to those nodes. That is, these contributions would be summed, added to a bias term, and then passed through a sigmoid function, in order to obtain the final probabilities of belonging to each class. By showing the individual contributions, one can better observe which components of the (phase) image drive the final prediction. (In this series of figures, green corresponds to a value of zero, warmer colors are positively valued, and cooler colors are negatively valued.)

In Fig. 4.9, we consider a cylindrical target located in a sand ripple field. In the SAS magnitude image, the object's highlight blends in with the background. From the SAS phase image, it is difficult to glean much information. However, by the second convolutional layer, significant structure has been uncovered, as evidenced in Fig. 4.9(d). For example, some of the filters are effectively delineating the shadow region; the somewhat remarkable thing is that this product has been produced from considering only the *phase* image. It is this sort of feature unearthing that partially explains why the phase can be exploited for target classification. (Results like this also suggest that an additional potential use of phase information can be image segmentation.)

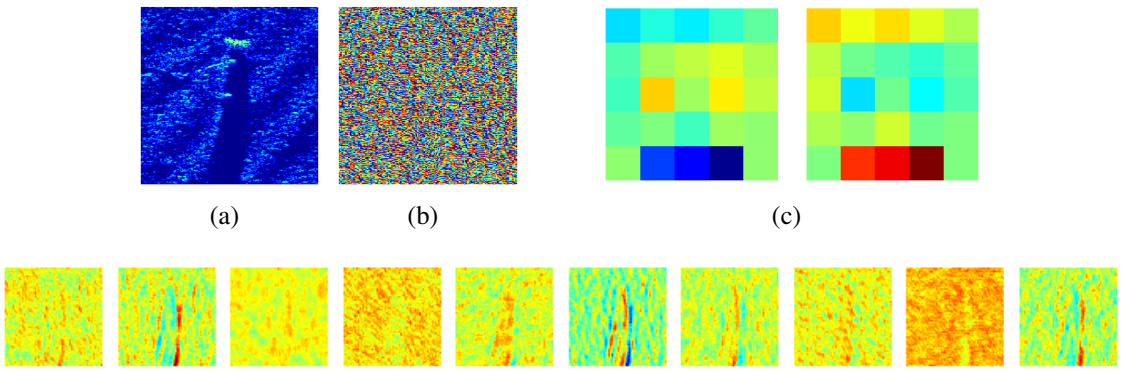


Fig. 4.9. SAS imagery of a cylinder (target) in a sand ripple field, in the form of its (a) magnitude image and the corresponding (b) phase image. The phase image is the input to the CNN; the magnitude image is shown only for reference. (c) Each receptive field's contribution to the final prediction of belonging to the clutter class (left) and target class (right). The target was classified correctly. (d) The intermediate representation of the phase image after the second convolutional layer, but prior to the activation function, of CNN B.

In Figs. 4.10 and 4.11, we consider two alarms from the clutter class. In these two cases, the intermediate representations seem to transform the input phase imagery into recognizable textures associated with specific orientations. At earlier layers of the CNN, closer to the original input imagery, the image abstractions are still difficult to understand, while at later layers of the CNN, the meaningfulness of the abstractions again becomes obscured.

In all three figures, it can be observed how different receptive fields influence the predic-

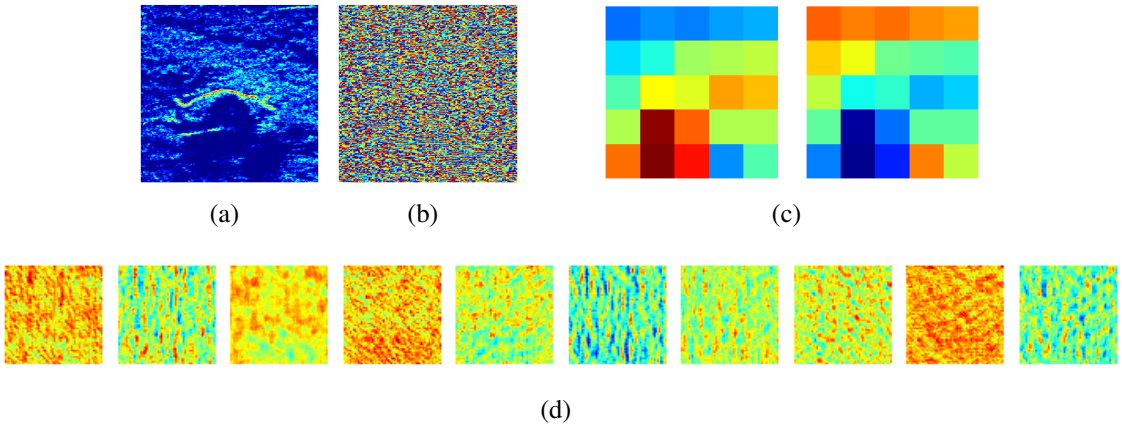


Fig. 4.10. SAS imagery of a clutter object in the form of its (a) magnitude image and the corresponding (b) phase image. The phase image is the input to the CNN; the magnitude image is shown only for reference. (c) Each receptive field's contribution to the final prediction of belonging to the clutter class (left) and target class (right). The clutter was classified correctly. (d) The intermediate representation of the phase image after the second convolutional layer, but prior to the activation function, of CNN B.

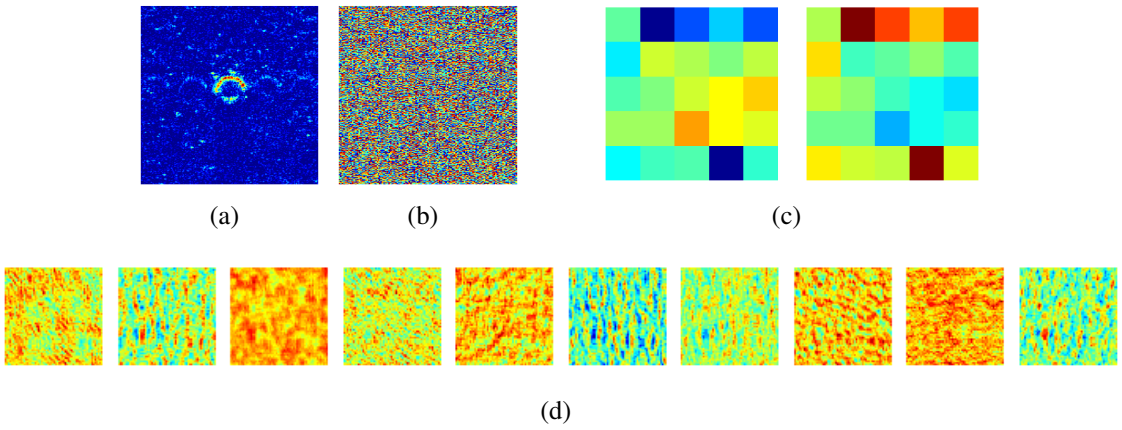


Fig. 4.11. SAS imagery of a clutter object in the form of its (a) magnitude image and the corresponding (b) phase image. The phase image is the input to the CNN; the magnitude image is shown only for reference. (c) Each receptive field's contribution to the final prediction of belonging to the clutter class (left) and target class (right). The clutter was classified incorrectly. (d) The intermediate representation of the phase image after the second convolutional layer, but prior to the activation function, of CNN B.

tions to different extents. However, the region in which one would expect to see a shadow (in the magnitude image) – due to the geometry shared by the sensor and proud object – does consistently have a significant impact on the predictions.

Based on these and other preliminary analyses, the information being exploited in the phase imagery for classification appears to arise when the pixel values deviate from a uniform distribution, and more specifically, form non-random spatial structure in the phase. Based on the physics involved, where the phase is tightly coupled to the distance traveled by the sonar signal, this scenario can manifest for different reasons. In [36], spatial correlation in the phase of synthetic aperture *radar* (SAR) imagery was found to be present because of strong reflectors, processing artifacts, and homogeneous surfaces. In [37], evidence of phase structure was seen in SAR images from very strong combined scatterers and their side lobes. In the underwater

domain, a discontinuity in the gradient of the phase can be an indication of an abrupt bathymetric change, and the presence of an object proud of the seafloor. Additionally, an acoustically smooth object may produce a structured phase image with a pixel distribution that is not uniform. We hypothesize that another potential source of structure occurs in shadow regions, where the signal levels are so low that deterministic self-noise of the sonar system itself may be visible in the phase.

4.3.4. Summary

It was demonstrated that the phase information present in complex high-frequency SAS imagery can be exploited for successful object classification. To exploit the information ostensibly hidden in the phase imagery, relatively simple CNNs were trained, "from scratch," on a large database of SAS phase images collected at sea. The filters learned by one of the CNNs were studied, and the intermediate responses from the network for specific input phase images were examined. Hypotheses regarding the sources of information contained in the phase imagery were offered. The results highlight the potential of using CNNs with non-traditional data representations for which manual feature-extraction is challenging, such as acoustic color.

4.4. MULTIPLE REPRESENTATIONS

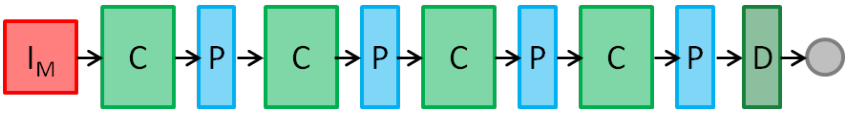
The richness of the *complex-valued* SAS data means there is potentially more exploitable information than what is apparent in the usual image domain. For example, the aspect-dependent nature of sonar returns off objects can be detected in the frequency domain [38], whereas the integration of this information 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 (*i.e.,* power spectral density) – and use them jointly as inputs to a CNN to improve classification performance.

A standard CNN transforms 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 *phase* imagery.

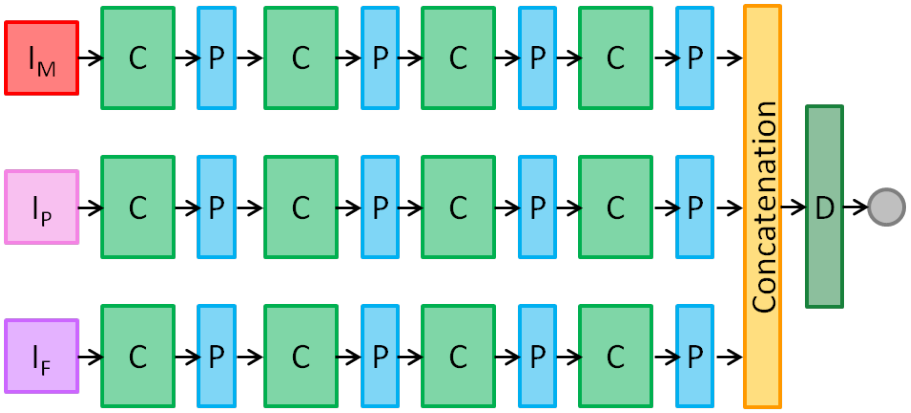
Some prior CNN-based research has focused on incorporating "multi-view" data from disparate sensor modalities [39–43], while other work has decomposed complex-valued data into two representations (*e.g.,* real and imaginary parts) [44, 45]. But our work is the first to derive and successfully exploit multiple input representations from the same raw sensor data.

4.4.1. CNN Design

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 red-green-blue (RGB) optical images – because the responses will destructively interfere after the first convolutional layer. (We observed this experimentally also.) 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, we choose to fuse the respective data products from the CNN transformations by concatenating them at the penultimate layer. A schematic of our proposed multi-representation CNN architecture, and an analogous single-representation architecture, is shown in Fig. 4.12.



(a) Single-representation (M) CNN architecture



(b) Multi-representation (MPF) CNN architecture

Fig. 4.12. Proposed CNN architectures for (a) single and (b) multiple representation inputs. *C*, *P*, and *D* respectively denote convolutional *blocks* (comprising one or more convolutional layers), pooling layers, and dense layers.

For this study, we carefully design four CNNs that share this common architecture. Additional details about the specific CNNs are provided in Table 4.12. It should be noted that the notation in this table differs slightly from previous tables. Here, brackets are used to convey the concept of convolutional *blocks*, because now there can be multiple convolutional layers in between pooling layers, whereas earlier there was always only one convolutional layer before pooling. The convolutional block construct allows deeper networks, and thus greater complexity, without a proportional increase in the number of parameters to learn. These new CNNs should be contrasted with the old, less-efficient CNNs in Table 4.2, which were shallower yet had more parameters to learn.

Table 4.12. Architectures of CNNs trained

CNN Label	Conv. Blocks	Conv. Layers Per Conv. Block	Filters Per Conv. Layer	Filter Sizes (Pixels Per Side)	Pooling Factors	Number of Parameters 1-rep. (M)	Number of Parameters 3-rep. (MPF)
A	4	1	4	[4] [3] [3] [4]	4 4 2 4	629	1885
B	4	1	4	[8] [6] [4] [5]	4 4 2 2	1509	4525
C	4	2	4	[6 6; 3 3] [6 6; 3 3] [6 6; 3 3] [6 6; 3 3]	4 2 2 4	2485	7453
D	4	3	4	[8 8 7; 7 7 5] [8 8 7; 7 7 5] [7 7 7; 5 5 5] [8 8 7; 7 7 5]	2 2 2 2	7877	23629

Each CNN contains 4 convolutional blocks; each block contains a specific number of convolutional layers. Each filter is square, and only 4 filters are used in each convolutional layer. ReLU activations are used after each convolutional layer, while a softmax activation is used at the output. 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.

4.4.2. Data Preparation

The MUSCLE data sets are again used for experiments, as this is the richest data – in terms of size and possible representations – available to us. A summary of the data sets is shown in Table 4.13, where it can be seen that the number of target examples in the training set is still extremely limited.

Table 4.13. Summary of SAS data sets for multi-representation experiments

Data Set	Number of	
	Targets	Clutter
Training Data	2912	29280
MAN2	404	14313
NSM1	113	6580
TJM1	351	1938
ONM1	91	111
GAM1	75	157

Given a complex sonar image, $\boldsymbol{z} = \boldsymbol{x} + i\boldsymbol{y}$, of an object, the frequency representation used as input to the multi-representation CNNs is $\mathbf{I}_F = \left(\log_{10}|\mathcal{F}\{\boldsymbol{z}\}| - 8\right)/3$, where $\mathcal{F}$ is the 2-d discrete Fourier transform (and the extra constants effect a normalization). The magnitude image representation is given by $\mathbf{I}_M = |\boldsymbol{z}|$, and the phase image representation is the result of the two-argument arc-tangent function, $\mathbf{I}_P = \boldsymbol{\phi}(\boldsymbol{y}, \boldsymbol{x})$. 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 $\times$ 267 pixels; for the magnitude and phase images, this corresponds to a spatial extent of 4.005 m $\times$ 4.005 m. An example of these three input data representations for a target is shown in Fig. 4.13.

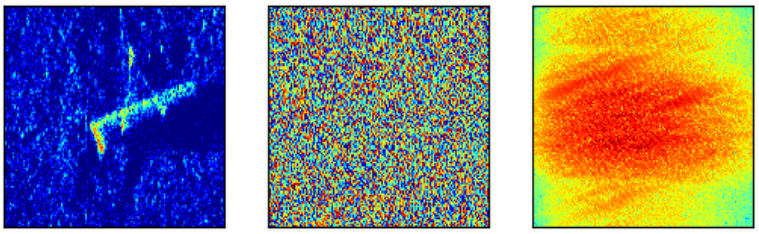


Fig. 4.13. From left to right, a cylindrical target's three data representations: sonar magnitude image, phase image, and frequency spectrum. Strong normal-incidence returns from the cylinder face are visible (as linear features) in the latter. In the magnitude and phase images, range increases to the right.

4.4.3. CNN Training

CNN training was performed using the RMSprop optimizer with a learning rate of $\eta = 0.001$, in conjunction with a binary-cross-entropy loss function, until the loss on the training set converged. A batch size of $B = 64$ 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; to this end, the samples for each batch were chosen from a random pool of size $2B$, which allowed an element of stochasticity to be retained. Data augmentation that respected the inviolable geometry of the sonar data-collection procedure was employed during training; this

meant a random range translation $i_{tx} \in [0, 0.5 \text{ m}]$, along-track translation $i_{ty} \in [-0.5 \text{ m}, 0.5 \text{ m}]$, and along-track reflection $i_{ry} \in \{0, 1\}$ was applied to each (complex) sonar image chip selected for the batch.

Using the architectures described in Sec. 4.4.1, we train four single-representation and four 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.

4.4.4. Experimental Results

We conduct experiments to show the benefit to classification performance, measured in terms of the 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 "centered input" case, denoted $\oslash$ later, in which the detected object is well-centered in the imagery, corresponds to $[i_{tx}, i_{ty}, i_{ry}] = [0, 0, 0]$.

The classification performance of the four single-representation CNNs is shown in Table 4.14, and of the four multi-representation CNNs in Table 4.15. Specifically, for each of the five test data sets, performance is shown 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 between 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 in both sets.

Also shown in the tables is the result of using the ensemble of the four 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 4.16 shows the performance of single-representation CNNs constructed using only sonar magnitude (M) imagery and the multi-representation CNNs (MPF) that also utilize phase imagery and the frequency spectrum as data inputs. (The values in this table correspond to the ensemble of isometric input cases from Table 4.14 and Table 4.15, brought together for easier comparison.)

Table 4.14. AUC for single-representation (M) 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	0.951	**0.964**	0.970	**0.979**	0.974	**0.981**	0.924	**0.955**	0.979	**0.984**
NSM1	0.905	**0.935**	0.916	**0.938**	0.940	**0.968**	0.923	**0.949**	0.965	**0.977**
TJM1	0.978	**0.985**	0.990	**0.995**	0.991	**0.998**	0.987	**0.993**	0.994	**0.998**
ONM1	0.969	**0.978**	0.984	**0.987**	0.989	**0.991**	0.936	**0.968**	0.990	**0.992**
GAM1	0.953	**0.986**	**0.990**	0.988	0.990	**0.994**	0.969	**0.990**	**0.999**	**0.999**

Table 4.15. 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 4.16. AUC for single-representation (M) and multi-representation (MPF) CNNs (all using the ensemble of isometric inputs)

Test	CNN A		CNN B		CNN C		CNN D	
Data Set	(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**
NSM1	**0.935**	**0.935**	0.938	**0.963**	0.968	**0.975**	0.949	**0.952**
TJM1	0.985	**0.989**	0.995	**0.996**	**0.998**	0.995	0.993	**0.997**
ONM1	**0.978**	0.962	**0.987**	0.957	**0.991**	**0.991**	0.968	**0.985**
GAM1	**0.986**	0.985	0.988	**0.993**	0.994	**0.997**	0.990	**0.999**
Parameters	629	1885	1509	4525	2485	7453	7877	23629

In Table 4.17, the performance of the ensemble of the four CNNs, denoted $\mathcal{E}(\text{ABCD})$, is shown for the single-representation cases and the multi-representation cases, as is the performance of the ensemble of those eight CNNs. For comparison purposes, we also adapted the pre-trained VGG16 net [19] to our problem and (sonar magnitude) data. Considerable effort was expended in order to achieve the best performance possible. After initializing the network with the pre-trained (*i.e.,* "off-the-shelf") weights, various approaches (and parameter settings) were explored. Despite using aggressive drop-out [46], the network quickly overfit. The best performance (used in Table 4.17) was obtained by "freezing" all but the final dense layer of weights for one epoch, and thereafter permitting refinement of all weights in the network for a few epochs. The best VGG16 performance achieved is shown in Table 4.17. The performance of the Mondrian detector, which makes predictions using a set of five features with fixed weights (and thus has no free parameters), is included as a baseline "shallow" classification approach.

Table 4.17. AUC for single-representation (M) and multi-representation (MPF) CNN ensembles and competing methods

Test Data Set	$\mathcal{E}$(ABCD) (M)	$\mathcal{E}$(ABCD) (MPF)	$\mathcal{E}$(ABCD) (M, MPF)	VGG16	Mondrian
MAN2	0.984	0.986	**0.987**	0.986	0.928
NSM1	0.977	0.980	0.982	**0.983**	0.892
TJM1	0.998	0.998	**0.999**	0.996	0.955
ONM1	**0.992**	0.987	0.988	0.958	0.851
GAM1	**0.999**	0.998	0.998	0.997	0.978
Parameters	12500	37492	49992	14715201	0

Table 4.17 also shows the number of parameters of each method. The ensemble of our eight 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 convolutional 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 4.17. Additionally, with our CNNs, training is more straightforward, free of extensive parameter tuning.

The full ROC curves associated with the results in Table 4.16 and Table 4.17 are also shown in Figs. 4.14-4.18 for the five different test data sets.

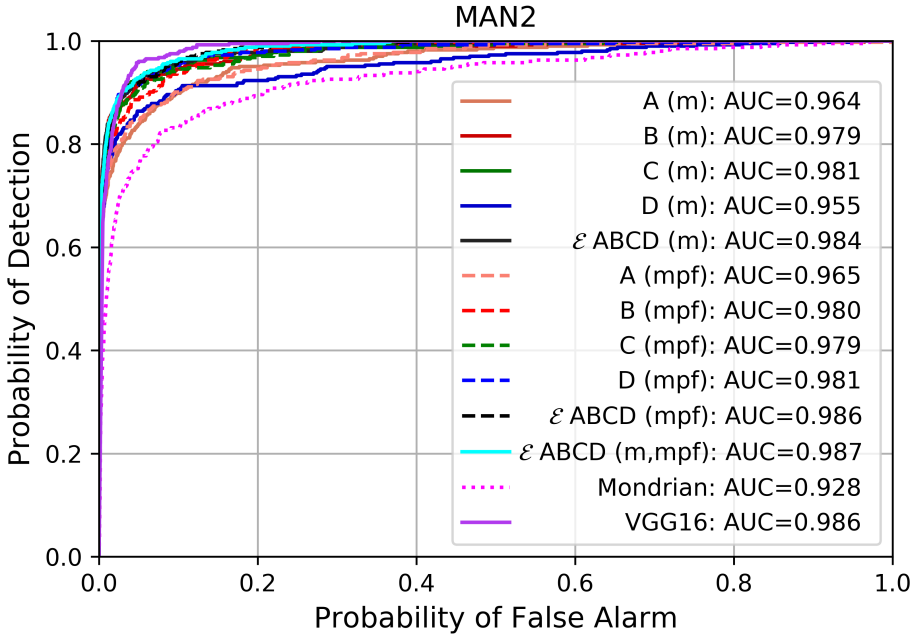


Fig. 4.14. For the MAN2 test data set, classification performance using the various CNNs.

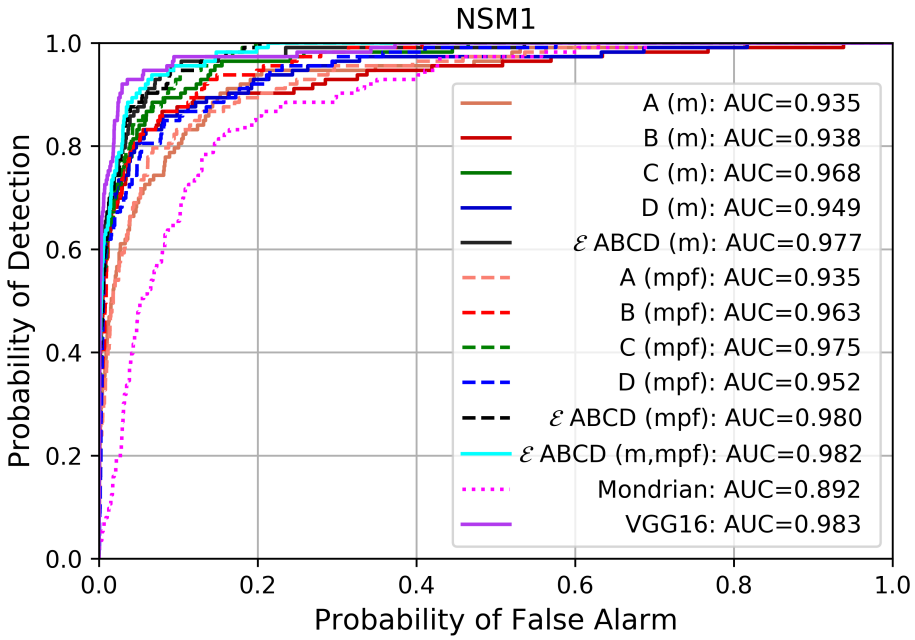


Fig. 4.15. For the NSM1 test data set, classification performance using the various CNNs.

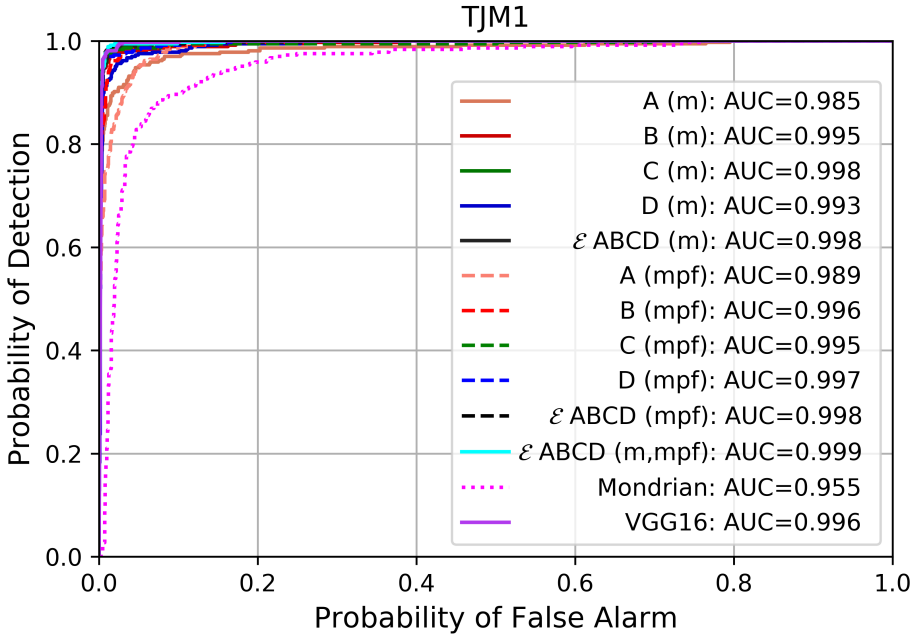


Fig. 4.16. For the TJM1 test data set, classification performance using the various CNNs.

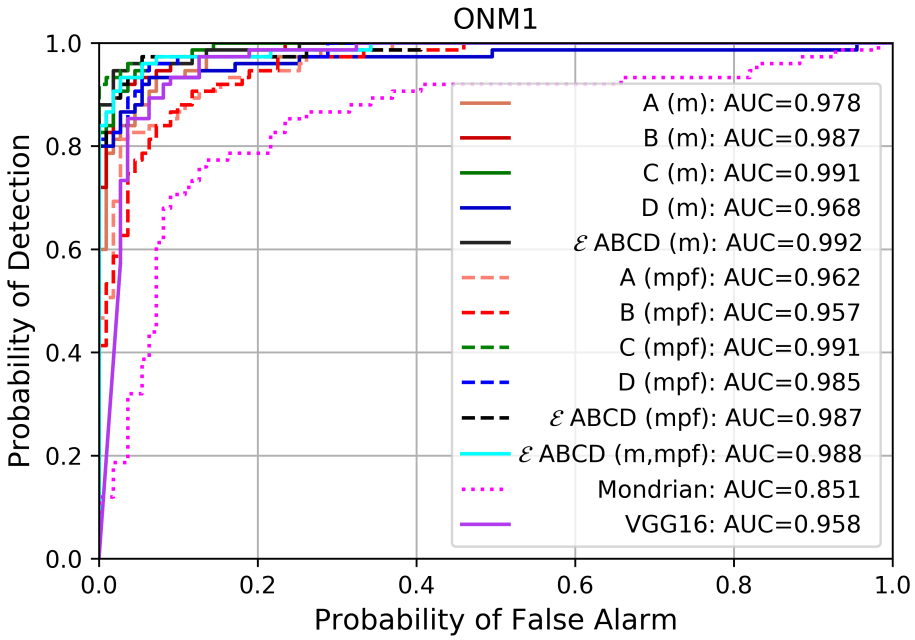


Fig. 4.17. For the ONM1 test data set, classification performance using the various CNNs.

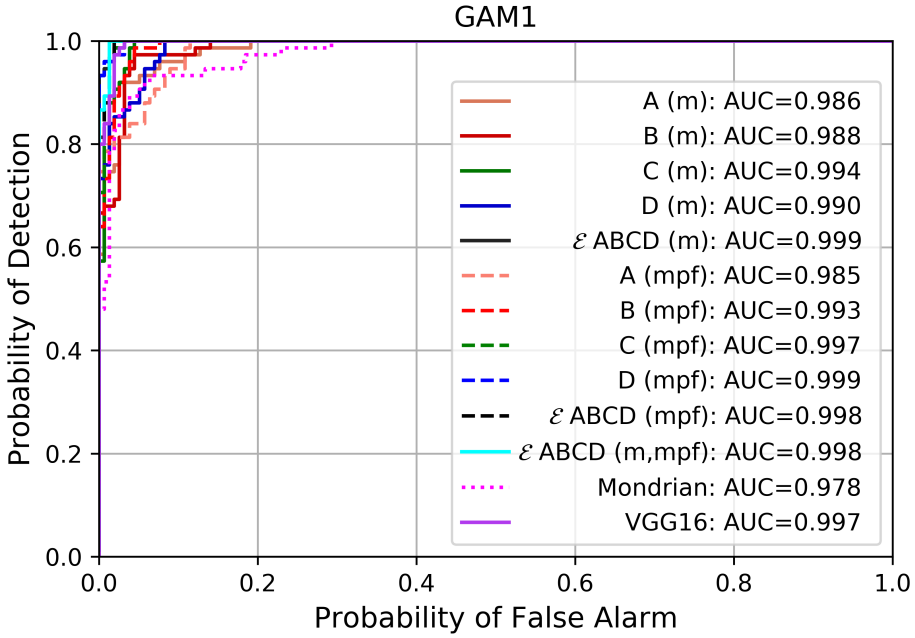
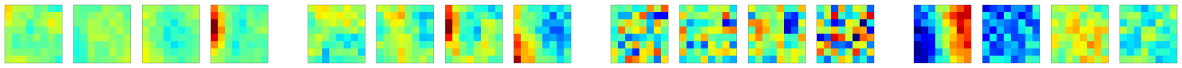


Fig. 4.18. For the GAM1 test data set, classification performance using the various CNNs.

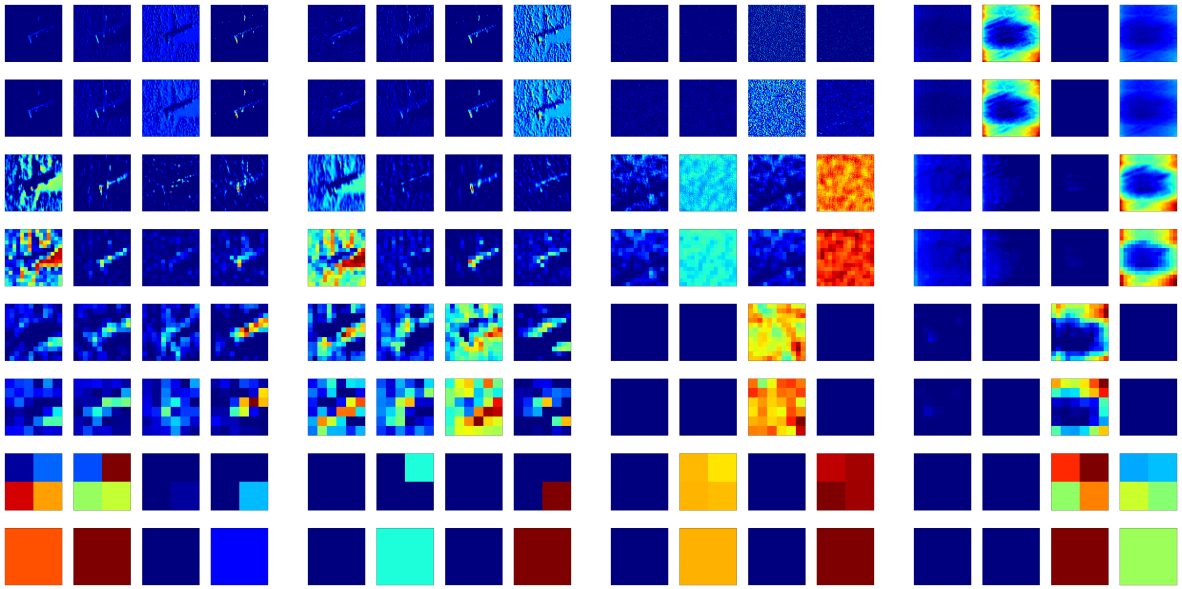
One valuable finding from the course of this study is that the individual branches of the multi-representation CNN should *not* be initialized with weights learned from training in isolation the analogous single-representation CNN. As can be seen in Fig. 4.19(a) and Fig. 4.19(b), the filters that are 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 our earlier experiments, causes training to get stuck in the equivalent loss-space minimum, thereby preventing gains (from the additional representations) from being realized.



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

Fig. 4.19. 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.

Finally, Fig. 4.20 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.



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

Fig. 4.20. Intermediate CNN layer responses for the data example in Fig. 4.13. 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.

To observe the relative amounts of discriminatory information contained in each of the three representations, we also train individual-representation CNNs using the architecture of CNN B. The ROC curves for these cases, along with the AUC values, are shown in Figs. 4.21-4.25 for the different test data sets. In these figures, the "B (mpf)" case corresponds to the *three*-representation input CNN; "$\mathcal{E}$ B (m,p,f)" is the ensemble of the three *single*-representation CNN cases.

It can be seen that the most information is contained in the magnitude representation, followed by the frequency representation, and then the phase representation. This result comports with intuition based on visual examination of the different image representations. However, it can also be seen that training the CNN with the three representations *jointly* – *i.e.,* our proposed framework – is better than training the three single-representation CNNs independently and simply averaging the individual predictions afterward. Because the jointly trained network has all the data available, it can exploit relationships and dependencies among the representations and achieve superior performance. These results provide additional evidence that the proposed multi-representation CNN framework is beneficial.

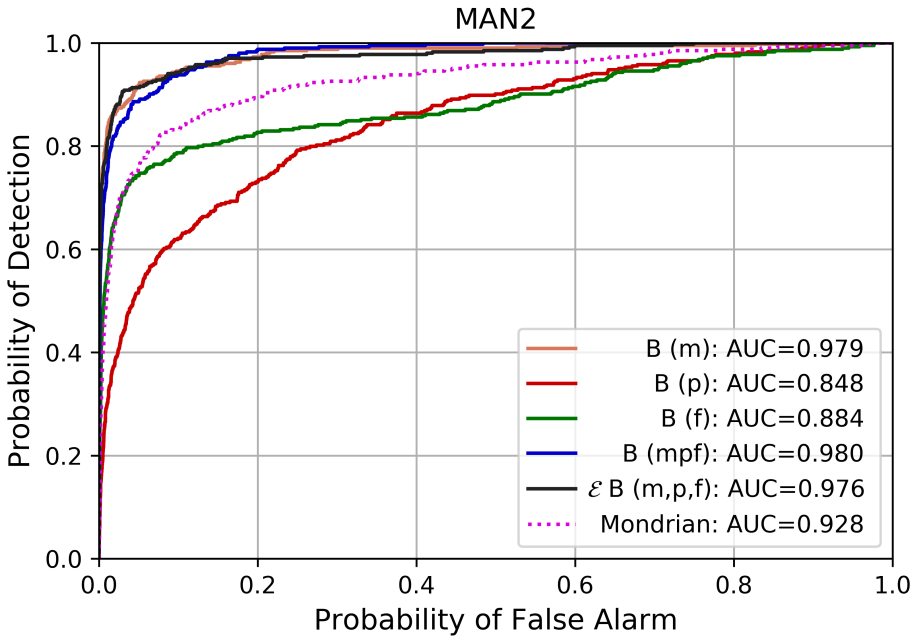


Fig. 4.21. For the MAN2 test data set, classification performance using the architecture of CNN B for different input representations indicated in the legend.

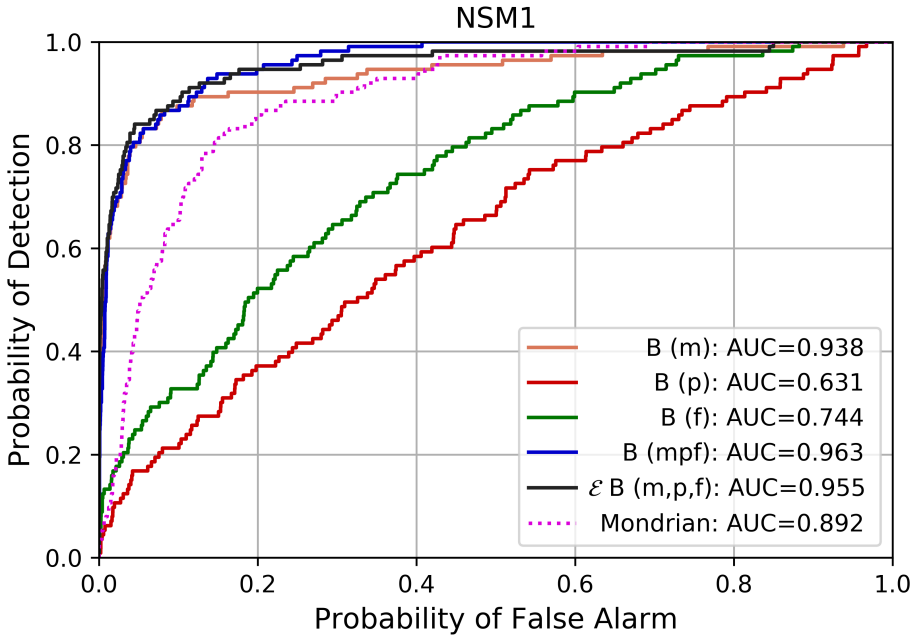


Fig. 4.22. For the NSM1 test data set, classification performance using the architecture of CNN B for different input representations indicated in the legend.

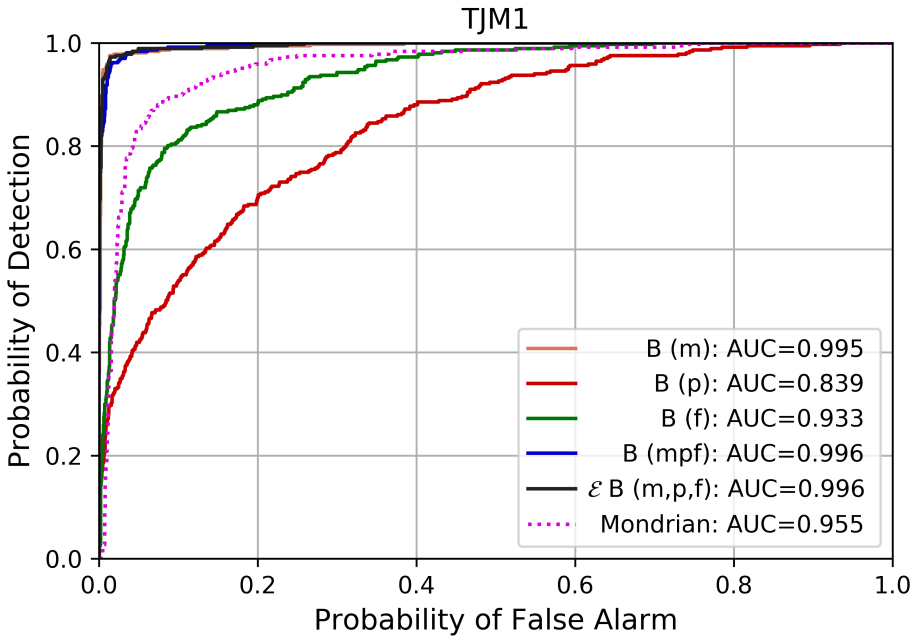


Fig. 4.23. For the TJM1 test data set, classification performance using the architecture of CNN B for different input representations indicated in the legend.

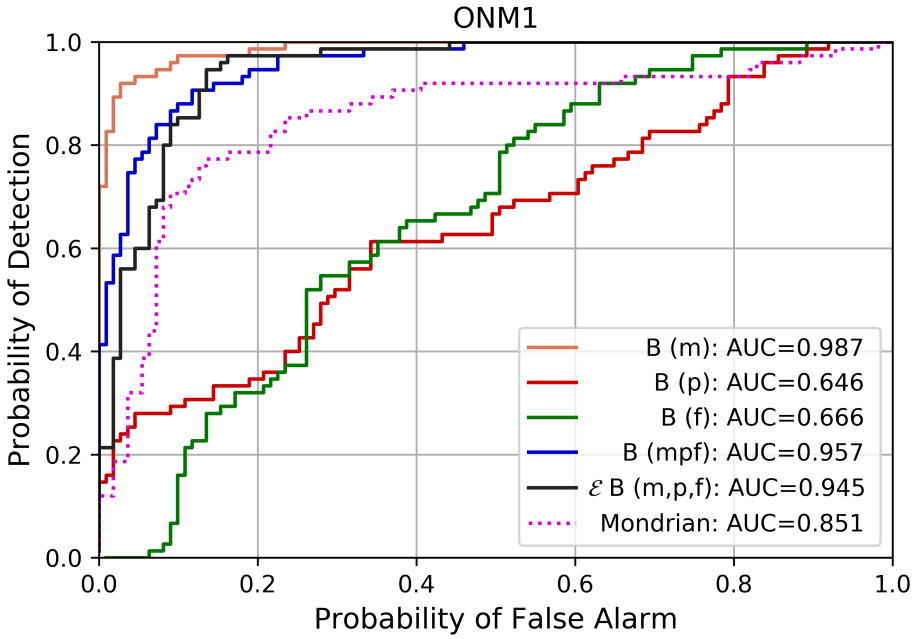


Fig. 4.24. For the ONM1 test data set, classification performance using the architecture of CNN B for different input representations indicated in the legend.

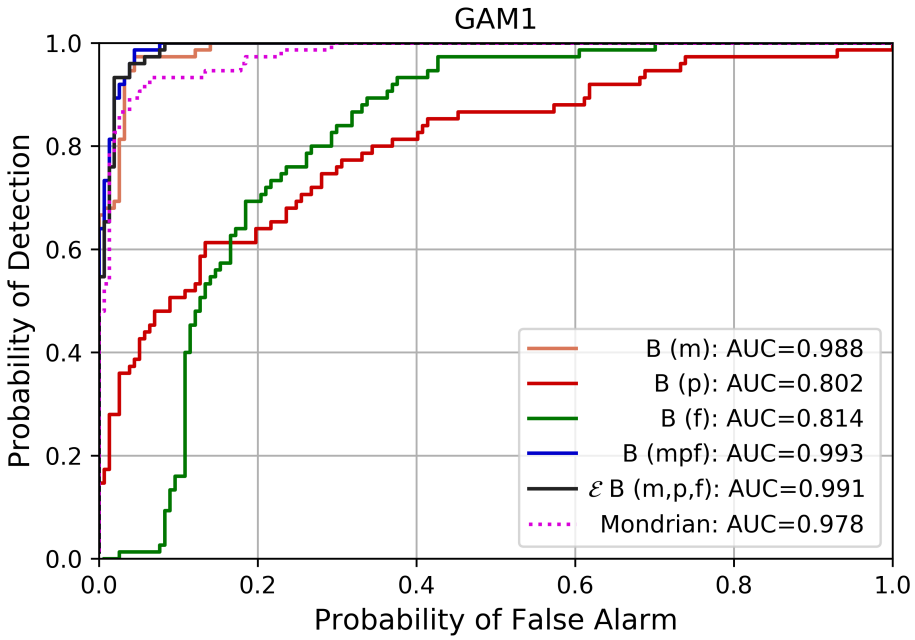


Fig. 4.25. For the GAM1 test data set, classification performance using the architecture of CNN B for different input representations indicated in the legend.

4.4.5. Summary

This study 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 UXO classification tasks, where the collection of data (at sea) is prohibitively costly in terms of both time and money.

But more importantly, a new CNN framework for fusing disparate data representations has been constructed. This approach is flexible and easily extendable to any number of data representations. As such, it provides a convenient unified classification approach capable of exploiting high-frequency and low-frequency sonar data products, as well as information from other sources, simultaneously.

Chapter 5

Conclusion

5.1. SUMMARY OF CONTRIBUTIONS

The overarching objective of this project was to lay the *groundwork* for a classification framework that could incorporate multiple data representations, from both high-frequency and low-frequency sonar data, as well as from different sensors, in order to improve underwater UXO remediation efforts. This goal has been achieved with the newly developed CNN architecture in which disparate representations are fused in the penultimate dense layer of the CNN. The success of this approach has been demonstrated on high-frequency complex-valued SAS data, where this data was used to create magnitude imagery, phase imagery, and a frequency spectrum representation. The framework was also shown to perform well with multiple representations of low-frequency acoustic-color data. By relying on a CNN-based approach, the feature-extraction process has been obviated. This marks a potential breakthrough for acoustic-color data, for which no robust, scalable solution to the feature-extraction problem had previously existed.

Additionally, the feasibility of leveraging related MCM data, and of using data from similar sensors, was also verified via transfer learning. The benefit of drawing on an ensemble of predictions obtained via various forms of multiple representations – in terms of input data representations, isometries, frequency sub-bands, aspect sub-apertures, multiple object views, and unique CNN architectures – was also illustrated. In particular, it was shown how this approach can greatly reduce the false alarm rate while maintaining the requisite high detection probability that UXO remediation demands. Importantly, it was demonstrated that successful classification could be achieved with notoriously "data-hungry" CNNs, even when only limited training data were available, by leveraging domain expertise in the design of the networks. By using CNNs with orders of magnitude fewer parameters than are typically used, the learned classifiers were robust and generalized well. This finding should also translate into significant cost savings by reducing the number and extent of expensive underwater (training) data-collection surveys needed.

5.2. PROPOSED FOLLOW-ON WORK

We have demonstrated a proof-of-concept that CNNs using non-traditional representations of sonar data as input can succeed at UXO classification. The feasibility of the approach has been shown for both high-frequency and low-frequency sonar data. A framework to exploit multiple representations simultaneously was also developed. Collectively, these components now open a promising avenue for follow-on research.

The main objective of the proposed follow-on work is to use the developed CNN framework in conjunction with specially controlled experiments to learn principled, explainable features that can be tied directly to the wave phenomena of the physics involved. The idea is to develop a CNN classifier in which the two classes are *not* UXO and non-UXO, but rather whether or not a specific object has a certain attribute. For example, one experiment would attempt to train a CNN to determine whether an object is solid or hollow. In this case, training data for one class would come from, say, a solid cylinder, while the training data for the other class would come from a hollow cylinder. Importantly, all other variables, such as range and environment, would be kept identical. As a result, any clues that the CNN uncovers must necessarily be due to the single variable that differs (in this example, the interior). Test data could be of the same objects, but at ranges different from that used in training. If the CNN is able to classify the test data correctly, the features from the dense layer of the CNN would necessarily be linked to the attribute under investigation. In the event that the physics is not fully explainable *a priori*, the intermediate representations uncovered by the CNN can lend insight and improve understanding to the wave phenomena involved. This entire process can also be repeated using frequency sub-bands and aspect sub-apertures, so that there is potentially an entire *set* of features – each with even greater specificity – linked to the attribute.

These controlled comparisons can be conducted for a series of attributes, such as object interior (*e.g.,* solid cylinder vs. hollow cylinder), exterior (*e.g.,* howitzer with a collar vs. howitzer without a collar), material composition (*e.g.,* steel UXO vs. aluminum UXO), diameter (*e.g.,* 105 mm UXO vs. 155 mm UXO), length (*e.g.,* 2:1 aspect telephone pole section vs. 5:1 aspect telephone pole section), and burial state (*e.g.,* object proud vs. object buried). (Valuably, all of the aforementioned example experiments can be supported by existing TREX13 data, so no additional data collections would be necessary.) Unique CNNs would be developed for each attribute case. The features derived from the classifiers that can successfully distinguish the two classes under consideration could be retained as explainable features linked to the relevant attribute. When a set of multiple such features was obtained, a new larger CNN that concatenates these controlled-comparison CNNs at the penultimate dense layer could be designed. That is, the multi-representation framework developed in this current project would be leveraged, albeit in a slightly different manner. The end result would be a CNN classifier in which the relative importance, or weight, of each "explainable" feature could be examined for each prediction.

All of the above can also be explored for various CNN architectures, as well as different or multiple data representations, to enable the use of ensembles that were shown to be so powerful in the current project. But the end result would be that every CNN classifier prediction that is made could be explained by a combination of well-understood features tied to specific attributes, frequency bands, aspects, *etc.* Preliminary controlled-comparison experiments we conducted suggest that this route is indeed feasible.

The same idea of controlled comparisons can also be used for two other important studies. In the current project, we demonstrated the robustness and generalizability of the CNN-based classifier even in the face of relatively limited data. As more data is available, performance should continue to improve. For this reason, modeling efforts to create simulated data are worthwhile. The controlled-comparisons approach can be used to establish when an increase in model complexity is necessary. If the CNN cannot distinguish data from two models with different complexity (*e.g.,* number of ray paths [47] or finite-element-method mesh elements [48]), it can be assumed that the simpler model is sufficient.

Underwater data collections are expensive, so the use of simulated data is an attractive route to pursue. But one must first verify that the simulated data is indistinguishable from real data. To this end, the controlled-comparisons approach could also help determine whether

simulated data can be confidently used interchangeably with real data for classifier training. If the CNN cannot distinguish simulated data (one class) from real data (the other class) under the same environmental conditions, it is an indication that the model data has sufficient fidelity. However, if the CNN is able to discriminate the simulated data from the real data, it is a valuable warning that additional model development is necessary. This investigation would provide a more rigorous *quantitative* answer than the qualitative (visual) assessment used in [48]. As such, it would further efforts seeking to reach the eventual goal of confidently training on simulated data and testing on real data.

Carefully controlling the difference between the two classes of data under examination – whether it be object attribute, or model complexity, or data provenance – forces the resulting CNN's final features to necessarily be tied directly to that specific quality. For the case of a certain object attribute, both classification and modeling efforts can be strengthened via a symbiotic feedback loop that alternates between explanations based on rigorous physics and the controlled CNN experiments' features that are produced. So the proposed follow-on work would aim not only to develop strong classifiers for guiding UXO remediation efforts, but also to enable a deeper fundamental understanding of the physics of the problem from a principled scientific perspective. We believe that no other extant approach offers the flexibility, power, and promise of explainability that this CNN-based framework affords.

References

[1] "SERDP/Office of Naval Research Workshop on Acoustic Detection and Classification of UXO in the Underwater Environment," Tech. Rep., SERDP Final Report, September 2013.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[3] A. Gleizes and J. Metzinger, *Du "Cubisme"*, Eugène Figuière Éditeurs, 1912.

[4] D. Williams, "Demystifying deep convolutional neural networks for sonar image classification," in *Proceedings of the Underwater Acoustics Conference*, 2017.

[5] D. Williams, "The Mondrian detection algorithm for sonar imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 1091–1102, 2018.

[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*, pp. 411–418. 2013.

[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] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, "High-resolution SAR image classification via deep convolutional autoencoders," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2351–2355, 2015.

[10] S. Chen, H. Wang, F. Xu, and Y. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4806–4817, 2016.

[11] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[12] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.

[13] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for SAR target recognition," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 364–368, 2016.

[14] Z. Lin, K. Ji, M. Kang, X. Leng, and H. Zou, "Deep convolutional highway unit network for SAR target classification with limited labeled training data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 7, pp. 1091–1095, 2017.

[15] W. Zhao, L. Jiao, W. Ma, J. Zhao, J. Zhao, H. Liu, X. Cao, and S. Yang, "Superpixel-based multiple local CNN for panchromatic and multispectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 4141–4156, 2017.

[16] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7177–7188, 2017.

[17] B. Dekker, S. A. Jacobs, A. S. Kossen, M. C. Kruithof, A. G. Huizing, and M. Geurts, "Gesture recognition with a low power FMCW radar and a deep convolutional neural network," in *European Radar Conference (EURAD)*. IEEE, 2017, pp. 163–166.

[18] 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*. International Society for Optics and Photonics, 2018, vol. 10789, p. 1078913.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

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

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

[24] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[25] 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.

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

[27] D. Williams, "Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks," in *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, 2016.

[28] J. Hanley and B. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, pp. 29–36, 1982.

[29] J. Egan, *Signal Detection Theory and ROC Analysis*, Series in Cognition and Perception. Academic Press, 1975.

[30] R. Hansen, T. Sæbo, K. Gade, and S. Chapman, "Signal processing for AUV based interferometric synthetic aperture sonar," in *Proceedings of IEEE OCEANS*, 2003, vol. 5, pp. 2438–2444.

[31] J. Chanussot, A. Hétet, G. Le Merrer, and E. Tireau, "Multispectral decomposition of synthetic aperture sonar images for speckle reduction," in *Proceedings of the Sixth European Conference on Underwater Acoustics (ECUA)*, 2002, pp. 269–274.

[32] S. Silva, S. Cunha, A. Matos, and N. Cruz, "Sub-band processing of synthetic aperture sonar data," in *Proceedings of IEEE OCEANS*, 2008, pp. 1–8.

[33] T. Marston, J. Kennedy, and P. Marston, "Coherent and semi-coherent processing of limited-aperture circular synthetic aperture (CSAS) data," in *Proceedings of IEEE OCEANS*, 2011, pp. 1–6.

[34] D. Williams and A. Hunter, "Multi-look processing of high-resolution SAS data for improved target detection performance," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 153–157.

[35] K. Williams, S. Kargl, E. Thorsos, D. Burnett, J. Lopes, M. Zampolli, and P. Marston, "Acoustic scattering from a solid aluminum cylinder in contact with a sand sediment: Measurements, modeling, and interpretation," *The Journal of the Acoustical Society of America*, vol. 127, no. 6, pp. 3356–3371, 2010.

[36] D. Petit, L. Soucille, J. Durou, F. Adragna, H. Oriot, and E. Simonetto, "Spatial phase behavior in SAR images," in *SAR Image Analysis, Modeling, and Techniques IV*. International Society for Optics and Photonics, 2002, vol. 4543, pp. 53–64.

[37] M. Soccorsi and M. Datcu, "Phase characterization of polarimetric SAR images," in *SAR Image Analysis, Modeling, and Techniques IX*. International Society for Optics and Photonics, 2007, vol. 6746.

[38] D. Plotnick and T. Marston, "Utilization of aspect angle information in synthetic aperture images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5424–5432, 2018.

[39] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *International Conference on Machine Learning*, 2015, pp. 1083–1092.

[40] A. Wang, J. Lu, J. Cai, T.-J. Cham, and G. Wang, "Large-margin multi-modal deep learning for RGB-D object recognition," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1887–1898, 2015.

[41] Y. Bin, Y. Yang, F. Shen, and X. Xu, "Combining multi-representation for multimedia event detection using co-training," *Neurocomputing*, vol. 217, pp. 11–18, 2016.

[42] J. Wagner, V. Fischer, M. Herman, and S. Behnke, "Multispectral pedestrian detection using deep fusion convolutional neural networks," in *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2016, pp. 509–514.

[43] Y. Yuan, G. Xun, K. Jia, and A. Zhang, "A Multi-view Deep Learning Framework for EEG Seizure Detection," *IEEE Journal of Biomedical and Health Informatics*, 2018.

[44] 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.

[45] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep Learning Convolutional Neural Networks for Radio Identification," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, 2018.

[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[47] L. Owsley and A. España, "Modeling buried target acoustic response by component," Tech. Rep., SERDP Project MR-2324 Final Report, November 2014.

[48] S. Kargl, "Acoustic response of underwater munitions near a sediment interface: Measurement model comparisons and classification schemes," Tech. Rep., SERDP Project MR-2231 Final Report, April 2015.

[49] S. Kargl, K. Williams, T. Marston, J. Kennedy, and J. Lopes, "Acoustic response of unexploded ordnance (UXO) and cylindrical targets," in *Proceedings of IEEE OCEANS*, 2010, pp. 1–5.

[50] D. Plotnick, P. Marston, K. Williams, and A. España, "High frequency backscattering by a solid cylinder with axis tilted relative to a nearby horizontal surface," *The Journal of the Acoustical Society of America*, vol. 137, no. 1, pp. 470–480, 2015.

[51] K. Williams, S. Kargl, and A. España, "TREX13 target experiments and case study: Comparison of aluminum cylinder data to combined finite element/physical acoustics modeling," *The Journal of the Acoustical Society of America*, vol. 136, no. 4, pp. 2111–2111, 2014.

[52] S. Kargl, A. España, K. Williams, J. Kennedy, and J. Lopes, "Scattering from objects at a water-sediment interface: Experiment, high-speed and high-fidelity models, and physical insight," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 3, pp. 632–642, 2015.

[53] A. España, K. Williams, D. Plotnick, and P. Marston, "Acoustic scattering from a water-filled cylindrical shell: Measurements, modeling, and interpretation," *The Journal of the Acoustical Society of America*, vol. 136, no. 1, pp. 109–121, 2014.

[54] M. Azimi-Sadjadi, "Multichannel detection and acoustic color-based classification of underwater UXO in sonar," Tech. Rep., SERDP Project MR-2416 Final Report, September 2015.

[55] J. Hall, M. Azimi-Sadjadi, and S. Kargl, "Underwater UXO classification using matched subspace classifier with synthetic sparse dictionaries," *Proceedings of IEEE OCEANS*, pp. 1–9, 2016.

[56] W. Au, *The sonar of dolphins*, Springer Science & Business Media, 2012.

[57] W. Evans and B. Powell, "Discrimination of different metallic plates by an echolocating delphinid," in *Animal Sonar Systems: Biology and Bionics*, 1967, vol. 78, pp. 363–382.

[58] P. Moore, S. Martin, and L. Dankiewicz, "Investigation of off-axis detection and classification in bottlenosed dolphins," in *Proceedings of IEEE OCEANS*, 2003, vol. 1, pp. 316–319.

[59] G. Sammelmann, "High-frequency images of proud and buried 3D-targets," *Proceedings of IEEE OCEANS*, pp. 266–272, 2003.