

COMPLEX INPUT CONVOLUTIONAL NEURAL NETWORKS FOR WIDE ANGLE SAR ATR

Michael Wilmanski^{#*1}, Chris Kreucher^{*2}, & Alfred Hero^{#3}

[#]The University of Michigan

500 S State St, Ann Arbor, MI 48109

¹wilmansk@umich.edu, ²ckreuche@umich.edu, ³hero@umich.edu

^{*}Integrity Applications Incorporated

900 Victors Way, Suite 220, Ann Arbor, MI, 48108

ABSTRACT

To date, automatic target recognition (ATR) techniques in synthetic aperture radar (SAR) imagery have largely focused on features that use only the magnitude part of SAR’s complex valued magnitude-plus-phase history. While such techniques are often very successful, they inherently ignore the significant amount of discriminatory information available in the phase. This paper describes a method for exploiting the complex information for ATR by using a convolutional neural network (CNN) that accepts fully complex input features. We show a performance leap from 87.30% to 99.21% accuracy on real collected wide-angle SAR data with the use of complex features.

Index Terms — Complex-valued Deep Learning, Convolutional Neural Networks, Complex Feature Extraction, Wide-angle SAR

1. INTRODUCTION

This paper describes an approach for fully exploiting complex synthetic aperture radar (SAR) data using a convolutional neural network (CNN). While magnitude-only CNNs have been successful [3-5] for SAR ATR, ignoring phase potentially sacrifices useful information that can aid classification. This motivates our proposed approach that fully exploits complex data. We show experimentally that this complex-input approach provides a substantial classification improvement over magnitude-only inputs for a set of collected wide-angle SAR data.

The paper proceeds as follows. First, we describe the magnitude-only CNN approach to ATR [4]. The input data is analogous to optical grayscale images, allowing CNN techniques developed for image classification to be used without modification.

Second, we describe the proposed CNN that uses complex-valued inputs, based on [1,2]. In this implementation, the first convolutional layer is fully complex. It implements a nonlinearity that produces real values at the output, and the rest of the network is made up of ordinary convolution and pooling layers.

Finally, we show experimental results for each approach that demonstrate that using complex features can aid in classification.

2. CNN CLASSIFIER WITH MAGNITUDE ONLY

Recent CNN-based SAR classification schemes [3-5] use magnitude-detected input data and leverage network designs originally designed for greyscale natural imagery. We briefly describe this approach here in order to contrast it to the fully complex CNN we present later. As such, we focus on the pertinent parts of the CNN implementation, namely the input normalization and CNN topology, which are most distinct from the complex-valued CNN. For more details on magnitude-only CNN ATR implementations see [4].

2.1 Normalization

Before input images are used by the CNN, they are normalized so that pixel values fall between a fixed range, e.g., [-0.5, 0.5], with an image-wide average of 0. Let $p(i, j)$ represent the unnormalized (magnitude-detected, real-valued) pixel values used as input. Let there be N total pixels. Then the normalized feature $p_n(i, j)$ is

$$p_n(i, j) = \frac{p(i, j) - \frac{1}{N} \sum_{i,j} p(i, j)}{\max_{i,j} p(i, j)}. \quad (1)$$

2.2 Network Topology

A CNN consists of several multi-resolution layers, each using a pooling method to pass information forward from layer to layer. A typical CNN topology for SAR image classification is summarized in Table 1 [3]-[5]. The numbers of network layers and filters in Fig. 5 are smaller than many general purpose CNN implementations to protect against overfitting for small size training image datasets typically available in SAR ATR. We implement the pooling layers with the well-established Stochastic Pooling [6] algorithm to further guard against overfitting. The output layer implements the CNN classification stage using the Softmax function.

Layer Type	Image Size	Feature Maps	Kernel Size
Input	80 x 80	1	-
Convolutional	72 x 72	18	9 x 9
Pooling	18 x 18	18	4 x 4
Convolutional	12 x 12	36	7 x 7
Pooling	4 x 4	36	3 x 3
Convolutional	1 x 1	120	4 x 4
Fully Connected	1	120	-
Fully Connected	1	120	-
Output	1	10	-

Table 1: Topology of the magnitude-only CNN.

3. COMPLEX CNN

There are several ways to implement a CNN that uses complex inputs. The main difficulty lies in specification of the activation function, which allows networks to learn highly non-linear functions for classification as well as produce a weight-error gradient used for training.

A successful activation function must be both non-linear and differentiable. The most common modern activation functions are rectified linear units (ReLU) [7] and the close variants (e.g. Leak-ReLU [8], PReLU [9]). Neither these nor classic activation functions such as Sigmoid and hyperbolic tangent are differentiable for complex inputs. Some researchers suggest splitting real and imaginary parts and processing each as a real number [10,11] or close variations [12]. Others suggest activation functions that ignore magnitude [13]. The problem with split activation functions is that phase information is overly distorted. The issue with ignoring magnitude is that magnitude contains significant information, defeating the point of a complex CNN, which attempts to make use of all available information.

These facts are illustrated in Figure 2 below, which shows the magnitude and phase information of an example from the GOTCHA dataset [14]. It is clear that both magnitude and phase contain structure, which suggests there is discriminatory information in both parts.

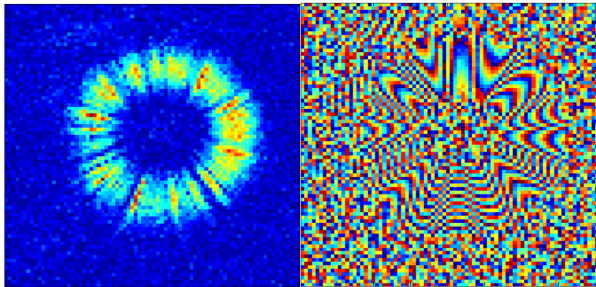


Figure 1: Magnitude (L) and phase (R) of tophat1

Our proposed complex-valued CNN is similar to an approach described in [1,2]. It is distinguished from these earlier efforts in that we do not use the absolute value function as the nonlinearity after the first hidden layer.

In our complex-valued CNN implementation, only the first layer is complex-valued and provides a nonlinearity that

produces real values. The rest of the network is made up of ordinary convolution and pooling layers. The complex layer has two filters for every feature map produced by the first hidden layer. Let A_i denote the matrix resulting from convolving the real part of our complex input image with the first filter of the i^{th} node and let B_i represent the matrix resulting from convolving the imaginary part of the complex input with the second filter of the i^{th} node. Our nonlinearity function for the i^{th} node is

$$f(A_i, B_i) = \sqrt{A_i^2 + B_i^2} \quad (2)$$

The error gradients for training use the partial derivatives

$$\frac{\partial f(A_i, B_i)}{\partial A_i} = \frac{A_i}{f(A_i, B_i)} \quad \text{and} \quad \frac{\partial f(A_i, B_i)}{\partial B_i} = \frac{B_i}{f(A_i, B_i)} \quad (3)$$

The feature maps produced here look distinctly different from a magnitude-only CNN, as shown in Figure 2 below.

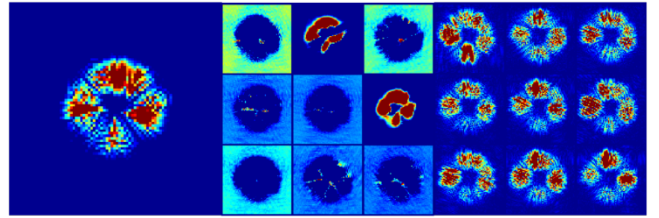


Figure 2: Detected image of a 5'' quad trihedral labeled stri01_1 (L), feature maps of first hidden layer for a greyscale CNN (C), feature maps of first hidden layer for a complex-input CNN (R)

3.1 Normalization

Normalization is handled differently in the proposed complex-valued CNN because subtracting the mean pixel value from every individual pixel distorts the phase information. Instead we use a scaling normalization

$$p_n(i, j) = p(i, j) / \max_{i, j} |p(i, j)|. \quad (4)$$

3.2 Network Topology

In contrast to the magnitude-only CNN described above, the first hidden layer is now a complex convolutional layer, as described above. Since the output of the complex convolutional layer is real-valued, the succeeding layers can be normal convolutional and pooling layers. See Figure 5 for the detailed complex CNN anatomy.

4. EXPERIMENTAL RESULTS

We use wide angle SAR data from the GOTCHA collect as in [14]. The dataset contains a set of civilian vehicles and a set of reflectors for target discrimination challenges. The vehicle set contains many moving targets and often has multiple targets per chip. To avoid multiple and or moving targets, that would require using motion-compensation techniques, we use the basic backprojection function

included in the GOTCHA distribution to synthesize composite images of the reflector set. We exclude SAR chips under the labels tophat2a and tophat2b because the targets are moving, and we exclude chips under label tophat_2_3 because the resolution is different and there are multiple targets per chip. We combine all instances of like shapes into single classes. The result is 651 complex chips of 80x80 pixels divided into 6 classes. We then split the set into 80% training and 20% testing sets of 525 and 126 chips respectively. The number of instances of each class is proportional between the training and testing sets.

We trained the CNN’s using stochastic mini-batches of size 35, resulting in 15 iterations per epoch. Each test involved training for 130 epochs (1950 iterations), tracking the loss value on the training set for another 10 epochs, then choosing the epoch that resulted in the lowest training set loss as the version of the network to test on.

4.1 Magnitude-only CNN

For the magnitude-only CNN the training set accuracy is 100%, while the testing set accuracy is 87.30%. Table 2 shows the testing set confusion matrix.

		PREDICTED						%CORR
		brud	dihed	scr1	stri	tophat	tri	
ACTUAL	brud	6	0	0	0	0	0	100
	dihed	0	6	0	0	0	0	100
	scr1	0	0	5	0	1	0	83.3
	stri	0	0	0	10	1	1	83.3
	tophat	0	0	0	2	10	0	83.3
	tri	0	0	0	0	10	74	88.1

Table 2: Confusion matrix for the magnitude-only CNN

An interesting metric is the entropy of the output prediction between classes, especially when comparing chips classified correctly and chips classified incorrectly. Low entropy among correctly classified chips is desirable, as it means that the network made a correct prediction with high certainty. Consequently, high entropy among misclassified chips is desirable as it means that incorrect predictions were made with low certainty and this is potentially detectable to the user. For probability distribution $p(x)$, the entropy is defined as

$$E(p(x)) = - \sum p(x) \log(p(x)) \quad (5)$$

For our 6-class scenario, total uncertainty corresponds to an entropy value of 1.8; a 50/50 split between two classes is ~0.7. Figure 3 compares the entropy of correct predictions to incorrect predictions for the magnitude-only case.

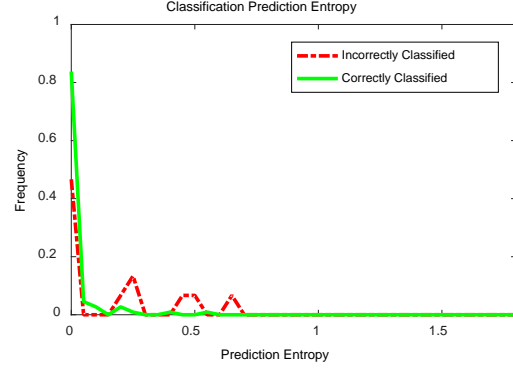


Figure 3: Output entropy for our magnitude-only CNN

4.2 Complex CNN

Accuracy on the training set is 100%, while testing accuracy is 99.21%. Table 3 shows the testing set confusion matrix and Figure 4 shows the histogram of prediction entropy.

		PREDICTED						%CORR
		brud	dihed	scr1	stri	tophat	tri	
ACTUAL	brud	6	0	0	0	0	0	100
	dihed	0	6	0	0	0	0	100
	scr1	0	0	6	0	0	0	100
	stri	0	0	0	12	0	0	100
	tophat	0	0	0	1	11	0	91.7
	tri	0	0	0	0	0	84	100

Table 3: Confusion matrix for our complex CNN

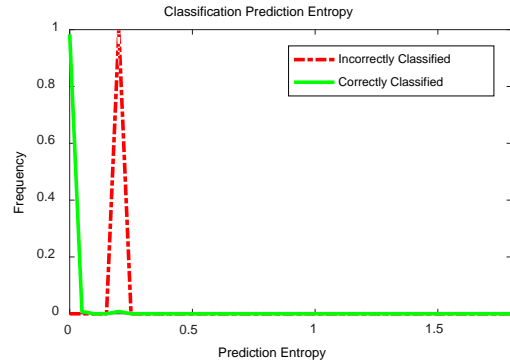


Figure 4: Output entropy for our complex CNN

5. CONCLUSION

Figure 1 shows that there is structure to the phase in the area of the target, suggesting that there is discriminatory information to glean from it. From this, we expect that making use of the phase information could lead to better results, as seen in the complex CNN experiment.

It is also interesting that the entropy of correct predictions decreased when the complex information was exploited. This suggests that the fully-complex approach has more desirable prediction entropy properties.

It is difficult to draw a conclusion for incorrect prediction entropy results since there are fewer incorrect predictions, but the outcomes seem comparable.

Our future work will focus on validating the performance improvement of the proposed complex-valued CNN on a larger database of SAR images.

6. ACKNOWLEDGEMENT

The authors would like to thank Dr. Mark Tygert for helpful discussions about his implementation of neural networks with complex-valued input features. This work was partially supported by ARO Grant W911NF-15-1-0479.

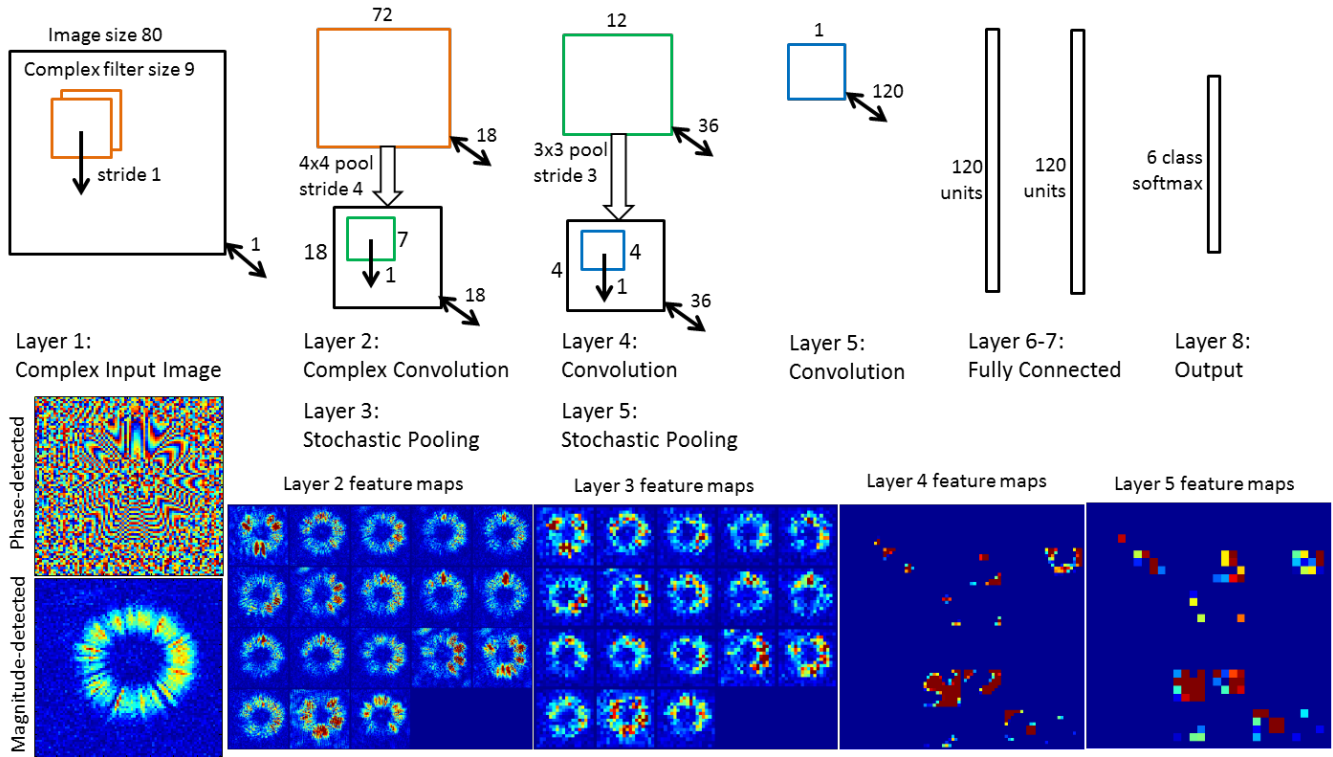


Figure 5: Detailed anatomy of the complex-input CNN; the CNN block diagram was inspired by figure 3 of [15]

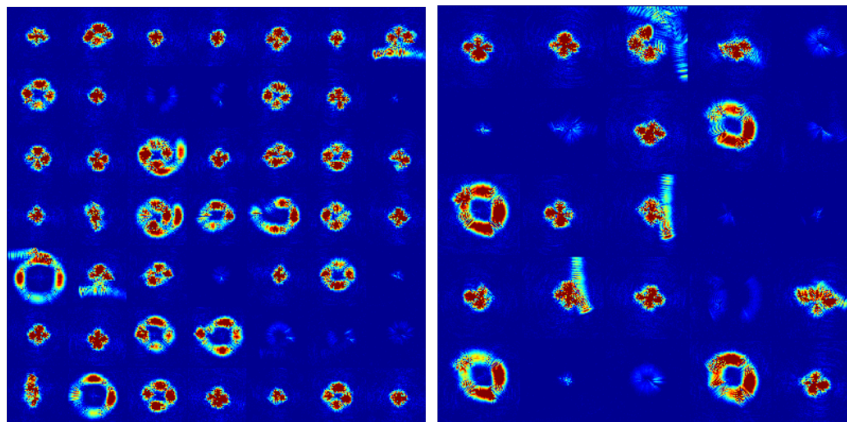


Figure 6: A random subset of training examples (L) and a random subset of testing examples (R) of the reflectors set of [14]; all are magnitude-detected and without normalization

7. REFERENCES

- [1] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino and A. Szlam, "A Mathematical Motivation for Complex-Valued Convolutional Networks", *Neural Computation*, vol. 28, no. 5, pp. 815-825, 2016.
- [2] M. Tygert, A. Szlam, S. Chintala, M. Ranzato, Y. Tian, and W. Zaremba, "Convolutional networks and learning invariant to homogeneous multiplicative scalings," *CoRR*, vol. 1506.08230, 2015.
- [3] D. Morgan, "Deep convolutional neural networks for ATR from SAR imagery", *SPIE Proceedings: Algorithms for Synthetic Aperture Radar Imagery XXII*, vol. 9475, 2015.
- [4] M. Wilmanski, C. Kreucher and J. Lauer, "Modern approaches in deep learning for SAR ATR", *SPIE Proceedings: Algorithms for Synthetic Aperture Radar Imagery XXIII*, vol. 9843, Baltimore, 2016.
- [5] S. Chen, H. Wang, F. Xu and Y. Q. Jin, "Target Classification Using the Deep Convolutional Networks for SAR Images," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4806-4817, 2016.
- [6] M. Zeiler and R. Fergus, "Stochastic Pooling for Regularization of Deep Convolutional Neural Networks", *CoRR*, vol. 1301.3557, 2013.
- [7] K. Jarrett, K. Kavukcuoglu, M. Ranzato and Y. LeCun, "What is the best multi-stage architecture for object recognition?," *2009 IEEE 12th International Conference on Computer Vision*, pp. 2146-2153, Kyoto, 2009.
- [8] A. Maas, A. Hannun and A. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models", *Stanford University Computer Science Department*, 2013.
- [9] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", *CoRR*, vol. 1502.01852, 2015.
- [10] R. Hansch, "Complex-Valued Multi-Layer Perceptrons—An Application to Polarimetric SAR Data", *EUSAR*, vol. 76, no. 9, pp. 1081-1088, 2010.
- [11] R. Haensch and O. Hellwich, "Complex-Valued Convolutional Neural Networks for Object Detection in PolSAR data," *Synthetic Aperture Radar (EUSAR), 2010 8th European Conference on*, pp. 1-4, Aachen, Germany, 2010.
- [12] T. Masters, *Deep Belief Nets in C++ and Cuda C Vol. II : Autoencoding in the Complex Domain*. CreateSpace Independent Publishing Platform, 2015.
- [13] I. Aizenberg, *Complex-valued neural networks with multi-valued neurons*. Berlin: Springer, 2011.
- [14] K. Dungan, J. Ash, J. Nehrbass, J. Parker, L. Gorham and S. Scarborough, "Wide angle SAR data for target discrimination research", *Algorithms for Synthetic Aperture Radar Imagery XIX*, 2012.
- [15] M. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks", *CoRR*, vol. 1311.2901, 2013.