



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

---

2022-12

# PRECONDITIONING AND THE APPLICATION OF CONVOLUTIONAL NEURAL NETWORKS TO CLASSIFY MOVING TARGETS IN SAR IMAGERY

Nearing, Heather G.

Monterey, CA; Naval Postgraduate School

---

<https://hdl.handle.net/10945/71517>

---

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

*Downloaded from NPS Archive: Calhoun*



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**PRECONDITIONING AND THE APPLICATION  
OF CONVOLUTIONAL NEURAL NETWORKS TO  
CLASSIFY MOVING TARGETS IN SAR IMAGERY**

by

Heather G. Nearing

December 2022

Thesis Advisor:  
Second Reader:

David A. Garren  
David C. Jenn

**Approved for public release. Distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.				
<b>1. AGENCY USE ONLY</b> (Leave blank)		<b>2. REPORT DATE</b> December 2022		<b>3. REPORT TYPE AND DATES COVERED</b> Master's thesis
<b>4. TITLE AND SUBTITLE</b> PRECONDITIONING AND THE APPLICATION OF CONVOLUTIONAL NEURAL NETWORKS TO CLASSIFY MOVING TARGETS IN SAR IMAGERY			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Heather G. Nearing				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release. Distribution is unlimited.			<b>12b. DISTRIBUTION CODE</b> A	
<b>13. ABSTRACT (maximum 200 words)</b>  Synthetic Aperture Radar (SAR) is a principle that uses transmitted pulses that store and combine scene echoes to build an image that represents the scene reflectivity. SAR systems can be found on a wide variety of platforms to include satellites, aircraft, and more recently, unmanned platforms like the Global Hawk unmanned aerial vehicle. The next step is to process, analyze and classify the SAR data. The use of a convolutional neural network (CNN) to analyze SAR imagery is a viable method to achieve Automatic Target Recognition (ATR) in military applications. The CNN is an artificial neural network that uses convolutional layers to detect certain features in an image. These features correspond to a target of interest and train the CNN to recognize and classify future images. Moving targets present a major challenge to current SAR ATR methods due to the "smearing" effect in the image. Past research has shown that the combination of autofocus techniques and proper training with moving targets improves the accuracy of the CNN at target recognition. The current research includes improvement of the CNN algorithm and preconditioning techniques, as well as a deeper analysis of moving targets with complex motion such as changes to roll, pitch or yaw. The CNN algorithm was developed and verified using computer simulation.				
<b>14. SUBJECT TERMS</b> Synthetic Aperture Radar, SAR, convolutional neural network, CNN, Automatic Target Recognition, ATR, autofocus, machine learning, moving target			<b>15. NUMBER OF PAGES</b> 95	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

**PRECONDITIONING AND THE APPLICATION  
OF CONVOLUTIONAL NEURAL NETWORKS TO  
CLASSIFY MOVING TARGETS IN SAR IMAGERY**

Heather G. Nearing  
Lieutenant, United States Navy  
BS, United States Naval Academy, 2016

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL  
December 2022**

Approved by: David A. Garren  
Advisor

David C. Jenn  
Second Reader

Douglas J. Fouts  
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

## ABSTRACT

Synthetic Aperture Radar (SAR) is a principle that uses transmitted pulses that store and combine scene echoes to build an image that represents the scene reflectivity. SAR systems can be found on a wide variety of platforms to include satellites, aircraft, and more recently, unmanned platforms like the Global Hawk unmanned aerial vehicle. The next step is to process, analyze and classify the SAR data. The use of a convolutional neural network (CNN) to analyze SAR imagery is a viable method to achieve Automatic Target Recognition (ATR) in military applications. The CNN is an artificial neural network that uses convolutional layers to detect certain features in an image. These features correspond to a target of interest and train the CNN to recognize and classify future images. Moving targets present a major challenge to current SAR ATR methods due to the “smearing” effect in the image. Past research has shown that the combination of autofocus techniques and proper training with moving targets improves the accuracy of the CNN at target recognition. The current research includes improvement of the CNN algorithm and preconditioning techniques, as well as a deeper analysis of moving targets with complex motion such as changes to roll, pitch or yaw. The CNN algorithm was developed and verified using computer simulation.



THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>A.</b>	<b>MOTIVATION .....</b>	<b>2</b>
<b>B.</b>	<b>OBJECTIVE .....</b>	<b>3</b>
<b>II.</b>	<b>RADAR IMAGERY BACKGROUND.....</b>	<b>5</b>
<b>A.</b>	<b>SYNTHETIC APERTURE RADAR.....</b>	<b>5</b>
<b>B.</b>	<b>AUTOFOCUS METHODS .....</b>	<b>15</b>
1.	<b>INVERSE FILTERING .....</b>	<b>15</b>
2.	<b>MAP-DRIFT.....</b>	<b>15</b>
3.	<b>PHASE GRADIENT AUTOFOCUS .....</b>	<b>17</b>
4.	<b>ARBITRARY RIGID OBJECT MOTION AUTOFOCUS         (AROMA) .....</b>	<b>19</b>
<b>III.</b>	<b>ARTIFICIAL INTELLIGENCE CONCEPTS.....</b>	<b>25</b>
<b>A.</b>	<b>MACHINE LEARNING .....</b>	<b>25</b>
<b>B.</b>	<b>CONVOLUTIONAL NEURAL NETWORKS.....</b>	<b>26</b>
1.	<b>Convolution Layer .....</b>	<b>27</b>
2.	<b>Pooling Layer .....</b>	<b>28</b>
3.	<b>Fully Connected Layer .....</b>	<b>29</b>
<b>C.</b>	<b>TRAINING THE NETWORK .....</b>	<b>30</b>
<b>IV.</b>	<b>DATASETS .....</b>	<b>33</b>
<b>A.</b>	<b>MSTAR.....</b>	<b>33</b>
<b>B.</b>	<b>SAMPLE.....</b>	<b>35</b>
1.	<b>SAMPLE MODEL.....</b>	<b>36</b>
<b>V.</b>	<b>EXPERIMENTAL DESIGN .....</b>	<b>39</b>
<b>A.</b>	<b>SIMULATION .....</b>	<b>39</b>
1.	<b>Preconditioning .....</b>	<b>39</b>
2.	<b>Method .....</b>	<b>42</b>
3.	<b>Inserting Artificial Phase Error .....</b>	<b>43</b>
4.	<b>Performing Autofocus Using AROMA.....</b>	<b>47</b>
5.	<b>CNN Architecture .....</b>	<b>52</b>
<b>B.</b>	<b>TRAINING AND TESTING.....</b>	<b>53</b>
<b>VI.</b>	<b>RESULTS AND ANALYSIS .....</b>	<b>57</b>
<b>A.</b>	<b>STATIONARY TARGETS BASELINING.....</b>	<b>57</b>

B.	DEFOCUS AND AROMA FOCUSED PERFORMANCE .....	59
C.	MOVING TARGETS .....	62
VII.	CONCLUSION AND FUTURE WORK .....	71
A.	CONCLUSION .....	71
B.	FUTURE WORK .....	73
	LIST OF REFERENCES .....	75
	INITIAL DISTRIBUTION LIST .....	77

## LIST OF FIGURES

Figure 1.	Patch collection geometry. Source: [5].	6
Figure 2.	Cross-Range return along range line. Source: [5].	9
Figure 3.	Synthetic aperture collection along a perpendicular path. Source: [5].	10
Figure 4.	Collection geometry for strip-mapping SAR. Source: [5].	11
Figure 5.	Collection geometry for spotlight-mode SAR. Source: [5].	12
Figure 6.	Three-dimensional SAR collection geometry. Source: [5].	12
Figure 7.	Map-drift technique focus process flow chart. Source: [5].	16
Figure 8.	Phase Gradient Autofocus (PGA) process flow chart. Source: [5].	17
Figure 9.	AROMA process diagram. Source: [4].	23
Figure 10.	Artificial intelligence and data science layered approach. Source: [6].	25
Figure 11.	Pipeline for image classification using a CNN. Source: [9].	27
Figure 12.	Visual depiction of the CNN process to calculate weights. Source: [11].	28
Figure 13.	Average pooling compared to max pooling process. Source: [9].	29
Figure 14.	Target types and comparison between MSTAR and SAMPLE data. Adapted from [3].	37
Figure 15.	Target information for .mat file for each SAMPLE image chip	39
Figure 16.	MATLAB output for complex magnitude image of the M1 Abrams tank using the imwrite() command	41
Figure 17.	SAR Image of 2S1 tank before inserting artificial phase error	44
Figure 18.	SAR Image of 2S1 tank after inserting artificial phase error via Power Law	45
Figure 19.	True artificial phase error vector 1	45
Figure 20.	True artificial phase error vector 2	46
Figure 21.	True elevation angle profile	46

Figure 22.	True path length error profile.....	47
Figure 23.	SAR image of 2S1 tank before performing autofocus via AROMA algorithm with artificially injected phase error to simulate target motion. ....	49
Figure 24.	SAR image of 2S1 tank after performing autofocus via AROMA algorithm. ....	50
Figure 25.	Comparison of 2S1 tank SAR images through precondition, artificial phase injection, and autofocus process. (DR: 50, SM: 100).....	50
Figure 26.	Confusion matrix using AROMA focused images with a DR: 50 and SM: 100. Accuracy 88.10%.....	51
Figure 27.	Comparison of BMP2 tank SAR images through precondition, artificial phase injection, and autofocus process. (DR: 100, SM: 300) .....	51
Figure 28.	Confusion matrix using AROMA focused images with a DR: 100 and SM: 300. Accuracy 55.36%. ....	52
Figure 29.	Portion of CNN algorithm used to create the layer structure.....	53
Figure 30.	Confusion matrix classifying defocused images for a power law of DC 100/SM 300 .....	61
Figure 31.	Confusion matrix after AROMA focus performed on images for a power law of DC 100/SM 300 .....	61

## LIST OF TABLES

Table 1.	AROMA theory primary variables. Source: [4].	21
Table 2.	Common classification tasks and activation functions. Source: [10].	30
Table 3.	Operating conditions for each MSTAR target. Source: [12].	34
Table 4.	Operating conditions for each MSTAR target. Source: [12].	34
Table 5.	Serial number for SAMPLE target vehicle. Source: [3].	36
Table 6.	SAMPLE metadata information. Source: [3].	37
Table 7.	SAMPLE PNG for M1 Abrams tank at 16 degree elevation and 010 degree azimuth center.	40
Table 8.	SAMPLE dataset class size by elevation degree.	42
Table 9.	Number of training and testing images for stationary targets.	54
Table 10.	Number of training and testing images for moving targets.	55
Table 11.	Power law variations and PNG display of smearing effect.	55
Table 12.	Stationary target baseline results with and without M60 target.	57
Table 13.	Visual comparison of M60 and ZSU23 at 17 degree and 15 degree elevation.	58
Table 14.	Defocused verses focused SAR images after CNN classification.	59
Table 15.	Moving target results for first three power law combinations DC 50/SM 100, DC 75/SM 100, and DC 100/SM 100.	63
Table 16.	Moving target results for second set of power law combinations DC 75/SM 150, DC 75/SM 200, and DC 75/SM 300.	64
Table 17.	Moving target results for third set of power law combinations DC 100/SM 200, DC 150/SM 250, and DC 200/SM 500.	65
Table 18.	Visual comparison for defocused and focused ZSU23 target for each power law combination tested during the moving target simulation.	66

Table 19.	Comparison between baseline, oversampling, and undersampling training datasets after CNN classification using various power law equations .....	69
-----------	---	----

## LIST OF ACRONYMS AND ABBREVIATIONS

AFRL	Air Force research laboratory
AROMA	arbitrary rigid object motion autofocus
ATR	automatic target recognition
CAD	computer aided design
CAT	computerized axial tomograph
CNN	convolutional neural network
CPU	central processing unit
CVNN	complex-valued neural network
CW	continuous wave
DL	deep learning
EO	electro-optic
FFT	fast Fourier transform
ISAR	inverse synthetic aperture radar
LFM	linear frequency modulation
ML	machine learning
ML	maximum likelihood
MSTAR	moving and stationary target and recognition
OC	operating conditions
PGA	phase gradient autofocus
QPM	quarter power magnitude
ReLU	rectified linear unit
SAMPLE	synthetic and measured paired labeled experiment
SAR	synthetic aperture radar
SDMS	sensor data management system
STARLOS	SAR target recognition and location system



THIS PAGE INTENTIONALLY LEFT BLANK

## ACKNOWLEDGMENTS

I would like to express my thanks and gratitude to my professor, Dr. David Garren, who took the time to guide and help me through the thesis process. Without his level of knowledge and advice, I would not have accomplished such a task. I also want to thank my second reader, Dr. David Jenn, whose suggestions and thoroughness have been critical to the completion of a successful thesis.

Another sincere thank you to Mr. Benjamin Lewis from the Air Force Research Laboratory, who has assisted me at every step of the way, especially in obtaining the SAR datasets used in this research. His support and timely communication were essential to the efficiency of my research.

Lastly, I send my most humble appreciation to Monsieur Joseph Fourier, whose discoveries have contributed to the very foundation of my education at the Naval Postgraduate School. His development of Fourier analysis is the safety net which allows engineers the ability to sleep soundly at night.

THIS PAGE INTENTIONALLY LEFT BLANK

## I. INTRODUCTION

The synthetic aperture radar principle is used globally for both military and civilian applications. The radar provides imagery from almost any platform, the most common being air and space-borne systems that capture a bird's-eye view of the ground below [1]. The radar return is processed into an image that humans visually examine and apply to everyday problems. The military uses SAR imagery for reconnaissance for example, to study the battlefield terrain or gain information on the adversary. The military also frequently uses optical and infrared instruments for intelligence collection; however, the SAR technique is more flexible because it is not limited by a real aperture wavelength [1]. SAR technique captures and stores the scene echoes along the synthetic aperture, then later combines echoes through a focus algorithm that yields a coherent image [1]. For both optical and infrared sensors, SAR has greater tactical advantage since it is not limited by weather or night-day environment [1]. Since the military operates at all hours of the day, it is necessary for continuous intelligence collection that SAR provides regardless of visibility conditions.

SAR applications include wide area observation and battlefield surveillance from space and aircraft sensors. More recently, SAR improvements include decimeter resolution and classification algorithms [1]. Classification of military targets is growing in importance and necessity to gain a strategic advantage over the adversary [1]. The major limitation of SAR, however, lies in its founding principle of a fixed scene center [1]. A moving object disrupts SAR image processing and complicates classification because it blurs and smears the pixels of the target, sometimes becoming unrecognizable with high-speed or complex maneuvering. There are numerous solutions to improve SAR moving target recognition, the most studied which are inverse SAR (ISAR) mode, bi/multi-static configuration, and improvements to focusing algorithms [1]. For this research, moving target SAR imagery will be simulated by inserting an artificial phase error into stationary SAR images prior to classification.

Target recognition and classification will be conducted using a convolutional neural network (CNN) algorithm. A CNN is a type of artificial neural network that falls under artificial intelligence, machine learning, and deep learning sequentially [2]. Machine

learning (ML) relies on the process of screening large sets of data for classification and uses training data to teach the algorithm what features to prioritize when a new set of data is presented [2]. The features are weighted, and the algorithm can recognize and classify new images [2]. In modern research, image and video processing favor the convolutional neural networks for classification because of networks ability to “learn” features [1]. CNN uses sparse connectivity which is where the hidden layers of neurons are only connected with the previous layer of subset neurons [2]. The architecture consists of three main layers that are convolutional layer, pooling layer, and fully connected layer. The convolutional layer performs the majority computations to convolve features using the filters to calculate a smaller set of weights [2]. The CNN is then trained using backpropagation to create a spatially flipped filter that performs convolution and additional filtering [2]. Next, the pooling layer reduces the spatial dimension of the image and maps the features using either average, max, stochastic, spatial spectral, or multiscale pooling operations [2]. The last layer is the fully connected layer that reconnects the neurons between layers and then makes a prediction based on probability of features belonging to a specific class [2]. Overall, the CNN produces high recognition accuracy when given large sets of training data and therefore a desirable method for conducting automatic target recognition in moving SAR imagery.

## **A. MOTIVATION**

The motivation behind this research is to apply deep learning to classification problem to improve accuracy while reducing the amount of processing and human involvement required for target recognition. While humans can classify images with high levels of accuracy just by the naked eye, they are limited to a single image at a time. The advantage of deep learning is the ability to process massive amounts of data and learn from it [2]. In particular, the neural network uses known images for training and then can recognize new images without the need to collect additional information. Additionally, this research aims to find the limits which the complexity and magnitude of target motion is too great to be focused and accurately identified. This threshold will show the limitations which arbitrary rigid object motion autofocus (AROMA) in conjunction with deep learning is able to assist in target recognition. Chapter V contains the results of this research and ties it back to the motivation discussed in this section.

## B. OBJECTIVE

The objective of this research is to improve preconditioning techniques and analyze the classification accuracy after defocus, focus, and processing through the convolutional neural network. Classification accuracy was tested using the Synthetic and Measured Paired Labeled Experiment (SAMPLE) dataset from the Air Force Research Laboratory (AFRL). The SAMPLE dataset consists of synthetically generated SAR targets with a class size of  $N = 10$ . The advantage of synthetically generated images versus measured data is that there is no background environment that could potentially mask the target of interest. SAMPLE data shows only the vehicle target of interest captured from different aspects and elevations. Without the influence of environmental clutter, the neural network trains only on the target pixels, which has shown improvement in target recognition performance over usage of measured dataset targets [3].

Using the SAMPLE dataset, defocus was conducted by inserting artificial phase error into the pixels of the complex image. The phase error used both translational and more complex movement that was generated from a power law equation in which the spectral density and decay constant were varied to simulate different types of motion. Increasing the spectral density and decreasing the decay constant also increased the complexity of the simulated motion relative to that of smaller spectral density and larger decay constant. The code to inject phase error was also used in the development of AROMA which allowed for simulation of “complicated non-linear rotational and translational target motions” [4]. The goal was to see how different degrees of defocus affected the classification accuracy before and after the images were focused and processed by the CNN. Chapter IV provides further explanation of the code and methods used to simulate moving targets for testing data.

Autofocus was performed on the smeared images using the analysis known as AROMA. AROMA expands upon the concept of phase gradient autofocus (PGA), a method commonly used for performing imagery autofocus. AROMA assists in focusing due to complex non-linear target motion rather than stationary objects that PGA is founded [4]. AROMA analyzes unknown rigid object motion to estimate the rotation angles and translation distances and refocus the target using a maximum likelihood signal-theoretic

analysis [4]. Chapter II explains the process by which AROMA calculates and refocuses smeared images.

The final step, classification, processes the AROMA focused images through the CNN. The MATLAB Deep Learning Toolbox™ was used to generate the CNN. The CNN algorithm consists of 15 layers that contribute the training and validation of the SAR images. The classification results are displayed in a confusion matrix and accuracy percentage is computed for correct class labeling. Chapter II provides a detailed review of the CNN and Chapter IV shows application of the CNN for this research. A summary of the results and recommendations for improved accuracy is contained in Chapter V, and the conclusion and future work are presented in Chapter VI.

## II. RADAR IMAGERY BACKGROUND

### A. SYNTHETIC APERTURE RADAR

As briefly described in Chapter I, synthetic aperture radar is a valuable method for collection of imagery that is unrestricted by night or weather conditions. The use of radar waves to produce an image first emerged around 1980 with the spotlight-mode SAR [5]. This mode was able to reproduce images that could nearly compete with optical sensors of the time, leading to the growth of SAR imaging technology for remote sensing [5].

The fundamentals of SAR are based on the transmission and reception of a radar waveform pulse [5]. Waveforms are sent out in a series of pulses that illuminate a desired target [5]. The target reflects back the pulses which are demodulated and processed to form an image [5]. The direction of these pulses from the radar antenna transmitter to the target scene is defined as the range and the cross-range direction is defined to be with the ground plane and orthogonal to the range direction [5]. Now that a ground reference is established, the next step is to describe the radiation from the airborne platform as it illuminates a ground target.

The continuous wave (CW) burst at radian frequency ( $\omega_o$ ) is a common radar waveform that is commonly used for target illumination, as it is comprised of the envelope function and carrier cosine wave [5]. For a single scattering point, the geometry from the patch center to the platform is described in Figure 1 and shows the ground range, slant range and depression angle. The time interval for the CW waveform is from  $\frac{-\tau_b}{2} < t < \frac{\tau_b}{2}$ .

The signal received by the radar for a single scatterer is [5]:

$$r_b(t) = A |g(u)| \cos(\omega_o(t - \tau_o - \tau(u)) + \angle g(u)) b(t - \tau_o - \tau(u)) \quad (1.1)$$

The time delay amount is  $\tau_o + \tau(u)$ ,  $g(u)$  is change in phase of waveform, and  $b(t - \tau_o - \tau(u))$  is the envelope function from (2.1), which is the time between CW burst pulses [5]. Equation (2.2) shows the relationship between slant and ground range, where  $u$  is the slant range,  $y$  is the distance to the scatterer, and  $\psi$  is the depression angle.



$$u = y \cos(\psi) \quad (1.2)$$

Ultimately, the range resolution is calculated in terms of the slant range, so it is necessary to understand the two SAR image coordinate systems.

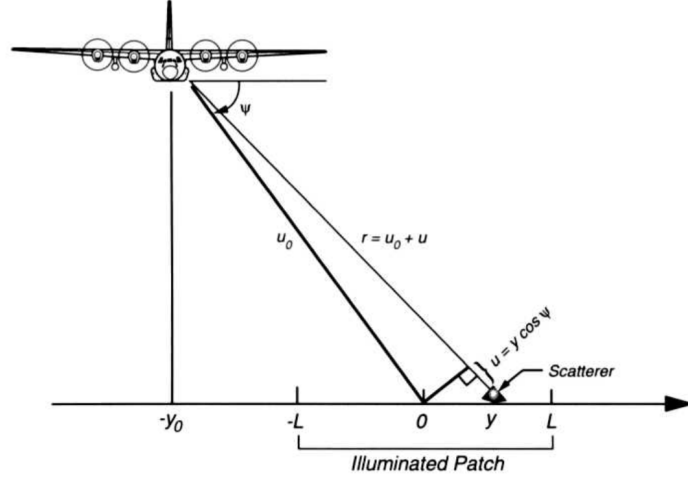


Figure 1. Patch collection geometry. Source: [5].

The total waveform return signal is given as the integral or density function from  $-L$  to  $L$  which is the ground range illuminated patch [5]. The patch propagation time  $\tau_p$  is incorporated into the two-way propagation delay to represent the returned signal interval [5]. The burst waveform duration  $\tau_b$  is chosen to be much smaller than  $\tau_p$  to resolve targets that are close range to each other [5]. The final resolution depends on the effective duration,  $\tau_e$ , for the CW burst pulse envelope given in (2.3). If two strong targets are close together, then the return waveform will have overlapping echoes, which are difficult to resolve [5]. The total effective bandwidth for the CW pulse is solved using the Fourier transform and is given in (2.4).

$$\tau_e = \frac{\int_{-\infty}^{\infty} f(t) dt}{f(0)} \quad (1.3)$$

$$B_e = \frac{1}{2\pi} \frac{\int_{-\infty}^{\infty} F(\omega) d\omega}{F(0)} \quad (1.4)$$

The returned signal is then mixed with the in-phase and quadrature sinusoids at the same frequency and sent through a low-pass filter [5]. The final complex signal, denoted by the overbar, after mixing is given in (2.5) as an integration from one half the propagation patch time  $\tau_p$  over the two channel outputs. Equation (2.6) gives the equivalent output written as a convolution.

$$\bar{r}_b(t) = a_0 \int_{-\frac{\tau_p}{2}}^{\frac{\tau_p}{2}} g\left(\frac{c}{2}\tau\right) \exp(-j\omega_0 \tau) b(t-\tau) d\tau \quad (1.5)$$

$$\bar{r}_b(t) = a_0 g(t) \odot b(t) \quad (1.6)$$

Equation (2.7) and (2.8) give the resolution in terms of  $\tau_e$  and  $B_e$ .

$$\rho_u = \frac{c\tau_e}{2} \quad (1.7)$$

$$\rho_u = \frac{c}{2B_e} \quad (1.8)$$

Another method to extracting the reflected signal from the return signal is to analyze (2.5) as a Fourier transform equation. The frequency domain equation is shown below in (2.9) and has two parts that are  $B(u)$  and  $G(U)$ . The  $U$  in (2.10), the spatial frequency variable, is essential to describe the spatial frequency offset that is used for spotlight-mode SAR cross-range resolution.

$$R_b(\omega) = a_1 G\left[\frac{2}{c}(\omega + \omega_0)\right] B(\omega) \quad (1.9)$$

$$U = \frac{2}{c}\omega \quad (1.10)$$

The CW burst as shown above requires minimal manipulation to recover the reflectivity profile; however, the long transmission time yields smaller bandwidth and range resolution [5]. One of the solutions to improve range resolution is a stretched waveform, such as the linear FM (LFM) chirp that is commonly used in imaging radars [5]. For an LFM chirp, the return echo is shown in (2.11), which integrates from near to far slant range,  $u$ .

$$r_c(t) = A \mathbf{Re} \left\{ \int_{-u_1}^{u_1} g(u) \exp \left[ j \left[ \omega_0 (t - \tau_0 - \tau(u)) + \alpha (t - \tau_0 - (u))^2 \right] \right] du \right\} \quad (1.11)$$

The bandwidth for an LFM chirp is in (2.12) and is in terms of chirp rate or  $\tau_c$ .

$$B_e = \frac{\alpha \tau_c}{\pi} \quad (1.12)$$

The final reflectivity  $g(t)$  is estimated from (2.13) by deconvolving  $s(t)$  and  $r_c(t)$ , or by using a process called deramping [5]. The deramping process is used for spotlight-mode SAR and mixes the return signal with in-phase and quadrature chirps, low-pass filters the mixer output, and is transferred into the Fourier domain [5]. In order to perform the Fourier transform the mixer output, it is assumed that the  $\tau_c$  chirp duration is larger than  $\tau_p$  the patch duration or  $\tau_c \gg \tau_p$  [5].

$$r_c(t + \tau_0) = A_1 \mathbf{Re} \left\{ \int_{-\tau_1}^{\tau_1} g_1(\tau) s(t - \tau) d\tau \right\} \quad (1.13)$$

After demodulation, the scene reflectivity or  $g(u)$  gives the spatial frequency  $U$  and resolution in terms of slant range from (2.14) [5]. The spatial frequency  $Y$  and resolution for the ground range is calculated from (2.16).

$$U = \frac{2}{c} \omega = \frac{2}{c} \left( \omega_0 + 2\alpha(t - \tau_0) \right) \quad (1.14)$$

$$\varrho_u = \frac{c}{2B_c} \quad (1.15)$$

$$Y = \frac{2 \cos \psi}{c} \omega = \frac{2 \cos \psi}{c} \left( \omega_0 + 2\alpha(t - \tau_0) \right) \quad (1.16)$$

$$\varrho_y = \frac{c}{2B_c \cos \psi_c} \quad (1.17)$$

Spotlight-mode SAR resolves both the range and cross-range dimensions via the deramp process to detect target separation and form an image [5]. The cross-range problem occurs when the separation between echoes overlaps and makes it difficult to resolve the

scene [5]. Figure 2 shows the cross-range problem where targets in the same range line separated in cross range are unable to be distinguished in a single return.

A real-aperture imaging radar can overcome the cross-range problem by sending pulses that are equal to the cross-range beam pattern width in order to maintain continuous coverage [5]. This method, however, is impractical because the radar antenna size increases for higher resolution. To achieve a cross-resolution of one-meter would require an unrealistically sized antenna [5].

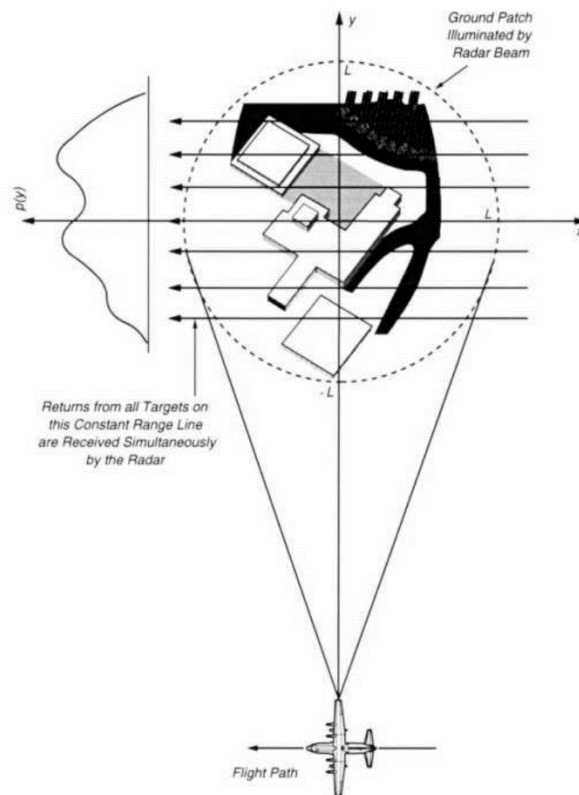


Figure 2. Cross-Range return along range line. Source: [5].

A synthetic aperture is another method that overcomes the cross-range problem and produces high resolution images without an extremely large physical antenna, as with a real aperture [5]. SAR is based on a concept called aperture synthesis that collects scene

data from multiple viewing angles in order to reconstruct the reflectivity scene [5]. A synthetic aperture is the path that the aircraft flies and transmits and receives the radar pulses [5]. The antenna mainbeam typically follows the patch center as the aircraft flies a perpendicular path as shown in Figure 3.

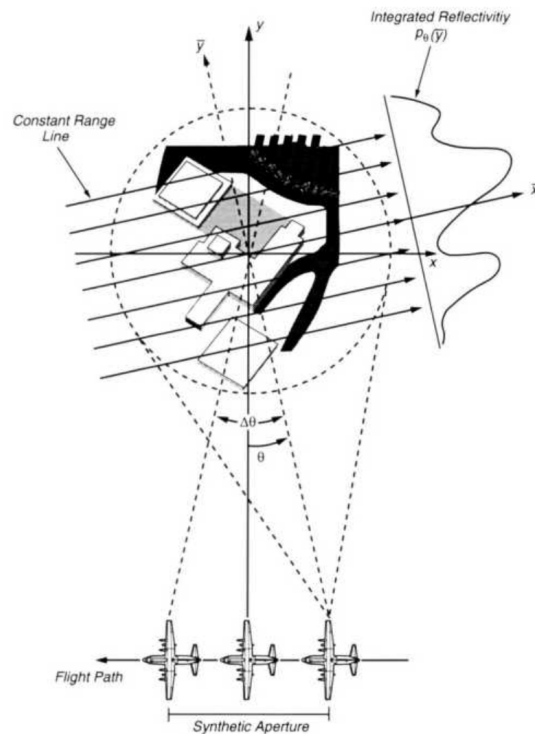


Figure 3. Synthetic aperture collection along a perpendicular path. Source: [5].

Two fundamental modes for SAR data collection are strip-mapping and spot-light geometry [5]. Strip-map SAR dates to the 1950s where an aircraft would aim the antenna orthogonal to the flight path while transmitting and receiving multiple pulses [5]. The longer the flight path, the better the resolution [5]. The pulses are transmitted along the flight path at one-half the antenna width spacing [5]. Figure 4 shows the collection geometry for strip-mapping SAR. This ensures there is no aliasing and produces a better

cross-range resolution [5]. Strip-map SAR is convenient because it is easy to adjust the antenna size and PRF to meet the minimum transmission spacing [5].

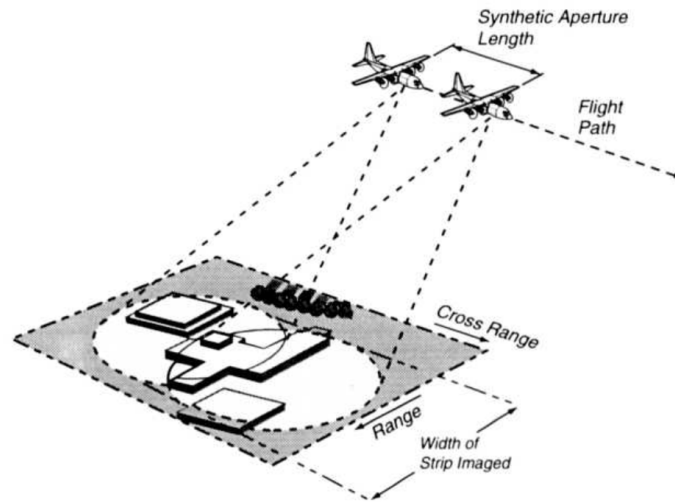


Figure 4. Collection geometry for strip-mapping SAR. Source: [5].

For small area patches, spotlight-mode SAR produces better resolution since the radar is slewed to fix on the illuminated patch while the aircraft is on its flight path [5]. Figure 5 shows the collection geometry for spotlight-mode SAR, which depicts how the radar continually illuminates the scene like that of a stage spotlight. Figure 6 shows the three-dimensional SAR collection geometry.

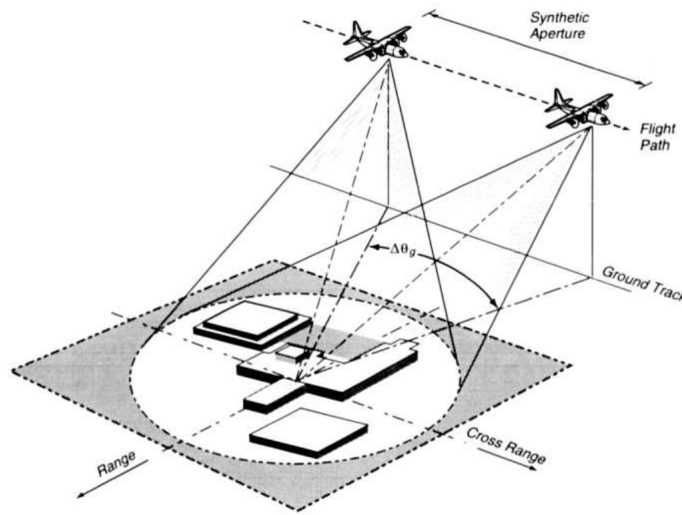


Figure 5. Collection geometry for spotlight-mode SAR. Source: [5].

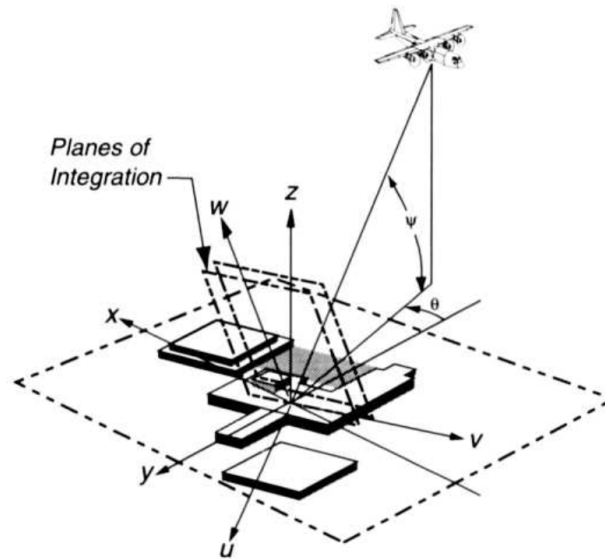


Figure 6. Three-dimensional SAR collection geometry. Source: [5].

For a three-dimensional spotlight-mode SAR collection, the reflected scene is obtained by using the tomographic framework and projection slice theorem [5]. Tomographic framework is commonly used in the medical field for CAT scans and x-ray imaging [5]. The framework produces a layered image using a series of thin slices from multiple viewing angles to recreate a patient’s body scan [5]. The projection-slice theorem

relates the Fourier transform to the viewing angle along the path which the radar scans [5]. For two-dimensional Fourier transform projections are swept to a polar raster 2D Fourier space then the inverse Fourier transfer is taken to get the spatial-frequency data that creates the final tomographic image [5]. Unlike medical tomography, SAR imaging captures scene that may contain elevated targets such as buildings and land features [5]. Elevation requires a third-dimension framework to accurately reproduce reflected scenes [5]. From Figure 6, one can see that the geometry is moved into three-dimensions where  $g(x, y, z)$  is integrated to yield the new projection function that is an angular orientation of  $(\theta, \psi)$ . Just as in 2D space, the Fourier transform of  $g(x, y, z)$  is applied for three dimensions to extract the pulses [5].

The collection of pulses determines the surface of the final image as Jakowatz refers to as a “ribbon surface” shape [5]. The most common method for collection is the straight-line flight, which naturally collects radar return data within a slant plane [5]. The SAR image is a combination of the slant plane and ground plane properties that are converted from polar raster to Cartesian raster data [5]. The final images can be described in resolution for range and cross range values [5]. The resolution depends upon the details of various quantities, including the spatial-frequency data, bandwidth, flight path geometry, and radar wavelength [5]. Equation (2.18) and (2.19) give the slant plane range and cross-range resolutions from the extent of the spatial frequencies  $\Delta Y'$  and  $\Delta X'$ .

$$\rho_{y'} = \frac{2\pi}{\Delta Y'} = \frac{c}{2B_c} \quad (1.18)$$

$$\rho_{x'} = \rho_x = \frac{2\pi}{\Delta X'} = \frac{\lambda}{2\Delta\theta} \quad (1.19)$$

Equation (2.20) and (2.21) give the ground plane range and cross-range resolutions.

$$\rho_y = \frac{2\pi}{\Delta Y} = \frac{c}{2B_c \cos\psi} \quad (1.20)$$

$$\rho_x = \frac{2\pi}{\Delta X} = \frac{\lambda}{2\Delta\theta} \quad (1.21)$$



Building on the concept of resolution, it is possible to calculate the maximum image dimensions. Ideally, a SAR collection would have the highest resolution of the largest scene area as possible [5]. The main challenge to achieving high resolutions is the ability to sample in a nearly straight trajectory and at a rate fast enough that the image can be reconstructed [5]. The maximum image dimensions rely on the sampling speed that the radar collects information [5]. Equation (2.22) and (2.23) give the maximum image dimensions based on the number of samples  $(N_x, N_y)$ .

$$D_x = \frac{2\pi N_x}{\Delta X'} = \frac{\lambda N_x}{2\Delta\theta} \quad (1.22)$$

$$D_y = \frac{2\pi N_y}{\Delta Y'} = \frac{cN_y}{2B} \quad (1.23)$$

Another challenge to achieving higher resolution is the SAR holographic and speckle properties. Spotlight-mode SAR imitates a hologram where the aperture size greatly affects the resolution, and its “phase history is sufficient to reconstruct and image of the entire scene, with resolution proportional to the size of the piece used.” [5] The reflectivity of a SAR image is similar to that of the optical reflection that a hologram builds an image [5]. Both methods produce bright and dark speckle patterns in the image due to the reflection of surface features and objects [5]. When the return is processed and scaled, the effects magnify and can sometimes appear smeared as a result of the smoothing during image processing [5].

Another consideration during SAR image processing is the phase error that is inevitable when considering the SAR platform motion and the sending and receiving of numerous pulses while airborne. The two main types of phase error are quadratic and high frequency [5]. The next section introduces different autofocus methods that are used to correct the phase error within SAR imagery.

## B. AUTOFOCUS METHODS

This section explores the different types of autofocus methods commonly used during image processing and classification. The different methods have advantages and disadvantages when focusing SAR imagery.

### 1. INVERSE FILTERING

Inverse filtering is a type of autofocus that attempts to correct for phase error that is unavoidable during SAR collection [5]. This technique is centered on a single point target and uses the inverse Fourier transform to estimate the effect of the phase error [5]. One drawback to using a single point target is that SAR images may contain more than one point target at the scene center [5]. In this case, inverse filtering performs focus based on the strongest point target in the image [5]. Equation (2.24) gives the corrected range-compressed data as function of the uncorrected data  $g_e$ , the estimated phase error  $\phi(m)$ , and the residual phase function  $\angle a$ . The indices  $k$  and  $m$  indicate range line and aperture position respectively. The final corrected image from (2.25), is the inverse fast Fourier transform of  $g_e$ , which is convolution of the original image with the residual blurring function  $b_r(n)$ . The index  $n$  indicates the cross-range image-domain number.

$$g_c(k, m) = \bar{g}_e(k, m) e^{-j\hat{\phi}(m)} = \bar{g}_e(k, m) e^{-j(\phi(m) + \phi_r(m) + \angle a)} \quad (1.24)$$

$$g_c(k, n) = IFFT_m \left\{ \bar{g}_c(k, m) \right\} = e^{-j\angle a} g(k, n) \odot b_r(n) \quad (1.25)$$

$$b_r(n) = IFFT_m \left\{ e^{-j\phi_r(m)} \right\} \quad (1.26)$$

Overall, inverse filtering performs well when there is a strong point target and little surrounding clutter in the scene [5]. The process is simple and outputs clear, focused images with few streaks. The downside to using inverse filtering is its ability to handle multiple point targets and clutter that increase streaking and defocus in the final image [5].

### 2. MAP-DRIFT

Map-drift is another type of phase error correction technique that is a sub-aperture process [5]. Map-drift uses the entire image rather than a point target to estimate the phase

error [5]. The phase error is represented as a polynomial function that often better represents the dynamic properties of phase error [5]. The model assumes that a quadratic phase error exists in the corrupted SAR image [5]. From there, the corrupted image is split into two in the aperture domain. The equal split means that the half-aperture images will be shifted linearly in the cross-range direction [5]. From here, the cross-correlation between the half-aperture peak and the sub-pixel location. The quadratic coefficient  $b$  is calculated from the cross-correlation, and a total phase error function is output [5]. The phase error function is given in (2.27). The process for focusing using map-drift technique is shown in the flow chart in Figure 7.

$$\phi(m) = bm^2 \quad (1.27)$$

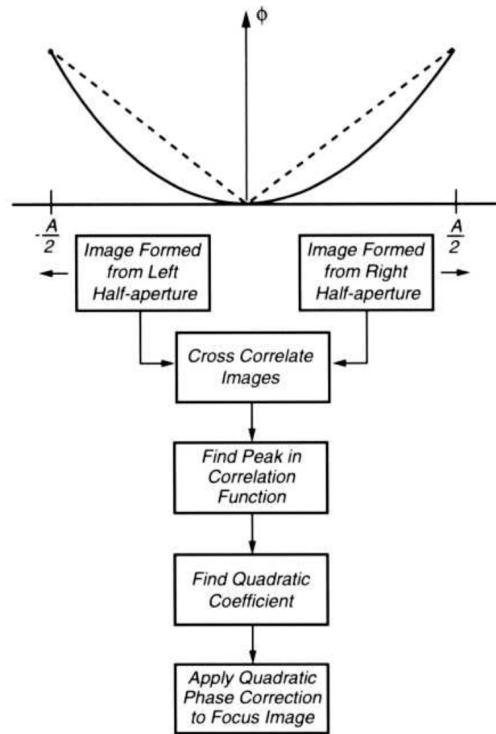


Figure 7. Map-drift technique focus process flow chart. Source: [5].

The disadvantage of using map-drift autofocus occurs when dealing with high order phase error [5]. This is due to the initial quadratic assumption that map-drift calculates the

entire phase error curve [5]. A better method for calculating higher order error is called phase gradient autofocus, which will be discussed next [5].

### 3. PHASE GRADIENT AUTOFOCUS

Phase gradient autofocus (PGA) is a maximum likelihood estimation process that averages the phase error across the image [5]. This differs from inverse filtering and map-drift techniques, which use a point target reference and parametric basis of phase error estimation respectively [5]. Instead of taking the entire average to estimate phase error for a single target, PGA combines “defocus information from a multiplicity of image targets with an optimality that is rooted in the tenets of statistical estimation theory” [5]. The autofocus process is performed in four steps which are circular shifting, windowing, phase difference estimation, and iterative correction [5]. Figure 8 shows the PGA process and steps.

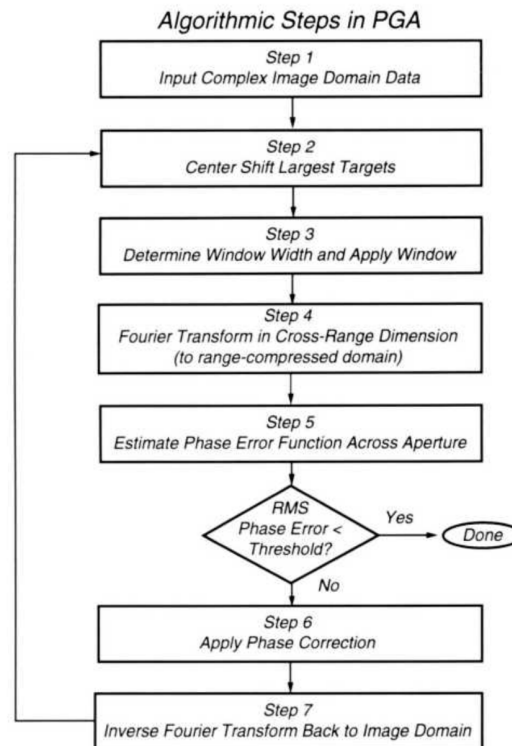


Figure 8. Phase Gradient Autofocus (PGA) process flow chart. Source: [5].

The first step is circular shifting also known as center shifting. This is accomplished through the PGA algorithm that looks at each range line in an image and selects the strongest target to be center shifted [5]. The purpose is to use only the strongest targets to create an average phase error estimate [5]. After the target is selected, the algorithm removes the linear phase component to isolate the phase error and integrate over the circularly shifted image [5].

Next, the second step is windowing which removes any clutter from circularly shifted image. A rectangular window is applied to the circularly shifted image to remove data that is not within the window [5]. The window also applies weights to any of the data that falls within the window [5]. The window size is selected based on non-coherent averaging that sums the magnitude of the image cross-range of the circularly shifted image [5]. This process creates a compressed image from which the PGA algorithm estimates phase error [5].

The third step, phase difference estimation, decompresses the image using the 1D Fourier transformation on each range line to extract the target reflectivity and noise term [5]. The PGA algorithm estimates the phase error from these two terms [5]. The noise term is assumed to be Gaussian which allows extraction of the target signal. The phase estimation is generated two ways [5]. The first approach is maximum likelihood (ML) estimation of all  $N \times M$  data points. The second approach, more popular, is to estimate the phase difference between two adjacent pulses [5].

The fourth step is iterative correction, which involves the repetition of the ML estimation on the refocused image to reduce target smearing. This process is iterated until the energy of the phase error function falls below a given threshold [5]. Each iteration improves image focus by reducing smearing and restoring pixels [5].

Jakowatz summarizes the advantages of PGA over inverse filtering in four key points that are: 1) does not require large point target, 2) uses larger portion of image data, 3) phase estimation is grounded in estimation theory, and 4) uses iterative process to refine phase estimation [5].

#### 4. ARBITRARY RIGID OBJECT MOTION AUTOFOCUS (AROMA)

Another autofocus method, proposed by Dr. David Garren from the Naval Postgraduate School, builds on the concept of PGA. The analysis known as Arbitrary Rigid Object Motion Autofocus (AROMA) also uses maximum likelihood to estimate the angles of rotation in moving targets SAR [4]. As discussed in Chapter I, AROMA better addresses the complex motion that moving targets exhibit compared to stationary objects.

Moving targets present a challenge to SAR target recognition due to the “smearing” effect which primarily occurs in the cross-range direction [4]. Most of the current research on autofocus methods assumes that the target velocity and acceleration are constant for both translational and rotational motions [4]. AROMA performs autofocus for complicated non-linear rotational and translation motion [4]. The analysis is related to PGA through its use of 3D generalization and a ML phase error estimation [4].

The AROMA model assumes that the target is a rigid body with arbitrary temporal profiles for translation and rotation relative to the radar [4]. The coordinate system for the model is based on the coordinate system where  $z = 0$  [4]. For the radar model, the pulse is defined as spherical coordinates with a given center frequency ( $f_c$ ) that is translated into a spatial frequency ( $\varrho_c$ ) [4]. The first pulse is defined in (2.28) and the second pulse is defined in (2.30).

$$R_1 = \tilde{A}_0 \exp(j\omega_1) + N_1 \quad (1.28)$$

$$\omega_1 \equiv 2\pi\varrho_c \{r_1 + x\cos(\theta_1)\cos(\phi_1) + y\cos(\theta_1)\sin(\phi_1)\} \quad (1.29)$$

$$R_2 = \tilde{A}_0 \exp(j\omega_2) + N_2 \quad (1.30)$$

$$\omega_2 \equiv 2\pi\varrho_c \{r_2 + x\cos(\theta_2)\cos(\phi_2) + y\cos(\theta_2)\sin(\phi_2)\} \quad (1.31)$$

The next assumption provides that the amplitude of the radar return pulses is constant, since the change between the pulses is small [4].

The second coordinate system to consider is the reference frame for the stationary target at the center of the target motion [4]. The coordinate system is a function of radial

distance, elevation angle and azimuthal angle. The coordinate systems for each of these angles is given in (2.32), (2.33), and (2.34) respectively.

$$r_0(t) \equiv \sqrt{\{X_p(t)\}^2 + \{Y_p(t)\}^2 + \{Z_p(t)\}^2} \quad (1.32)$$

$$\theta_0(t) = \arctan\left(\frac{Z_p(t)}{\sqrt{\{X_p(t)\}^2 + \{Y_p(t)\}^2}}\right) \quad (1.33)$$

$$\phi_0(t) = \arctan\left(\frac{Y_p(t)}{X_p(t)}\right) \quad (1.34)$$

The final phase change is estimated using a linear approach to find the difference in distance between successive pulses [4]. The linearized form of the phase estimation is given in (2.35).

$$\Delta\omega(t) = 2\pi Q_c \{ \Delta\zeta(t) + x \Delta\mu(t) + y \Delta\nu(t) \} \quad (1.35)$$

$$\Delta\mu(t) \equiv \mu_2(t) - \mu_1(t) = \cos(\theta_2(t))\cos(\phi_2(t)) - \cos(\theta_1(t))\cos(\phi_1(t)) \quad (1.36)$$

$$\Delta\nu(t) \equiv \nu_2(t) - \nu_1(t) = \cos(\theta_2(t))\sin(\phi_2(t)) - \cos(\theta_1(t))\sin(\phi_1(t)) \quad (1.37)$$

$$\Delta\zeta(t) \equiv \zeta_2(t) - \zeta_1(t) = r_2(t) - r_1(t) \quad (1.38)$$

The phase estimation for  $(\zeta, \mu, \nu)$  is calculated using the ML techniques, physical signal model, and integration along the synthetic aperture [4]. As with PGA, the strongest scattering center is selected for estimation on each range line. Down-range position is  $x_k$  and cross-range coordinate is the line  $k$  [4]. The final model is given in (2.39) and Table 1 describes the variables for the AROMA theory to assist in this background chapter.

$$\Delta\omega_k(t) \equiv 2\pi Q_c \{ \Delta\zeta(t) + x_k \Delta\mu(t) + \tilde{y}(x_k) \Delta\nu(t) \} \quad (1.39)$$

Table 1. AROMA theory primary variables. Source: [4].

Variable	Meaning
$j$	imaginary unit = $\sqrt{-1}$
$c$	speed of light
$f_c$	radar centre frequency
$\rho_c$	radar centre spatial frequency = $2f_c/c$
$t$	slow-time
$n$	radar pulse index
$r(t)$	instantaneous radial distance to the radar
$\theta(t)$	instantaneous elevation angle to radar
$\phi(t)$	instantaneous azimuthal angle to radar
$\zeta(t), \zeta_n$	radial error function
$\mu(t), \mu_n$	first direction cosine error function
$\nu(t), \nu_n$	second direction cosine error function
$\Delta\zeta(t), \Delta\zeta_n$	change in $\zeta$ for successive pulses
$\Delta\mu(t), \Delta\mu_n$	change in $\mu$ for successive pulses
$\Delta\nu(t), \Delta\nu_n$	change in $\nu$ for successive pulses
$x_k$	ground down-range indexed by $k$
$y_\ell$	ground cross-range indexed by $\ell$
$\tilde{y}(x_k)$	value of $y_\ell$ for dominant scatterer on $x_k$
$\eta_n$	cross-range spatial frequency at pulse $n$
$g(x_k, y_\ell)$	complex-valued image domain function
$G(x_k, \eta_n), G_{k,n}$	1D Fourier transform of $g(x_k, y_\ell)$
$G_k$	first function $G(x_k, \eta_n)$ of a pair
$H_k$	second function $G(x_k, \eta_{n+1})$ of a pair
$z_k$	vector of $G_k$ and $H_k$
$Q_k$	covariance matrix for $z_k$
$\sigma_A^2$	variance of the target signal process
$\sigma_N^2$	variance of the clutter-plus-noise process
$p(z_k   \Delta\zeta, \Delta\mu, \Delta\nu)$	conditional PDF
$Z$	vector of $K$ -independent range lines
$\Psi_{p,q,n}$	interference product-moment function

The next process explains the estimation of the phase error. As established, AROMA uses a ML estimate for the error functions [4]. AROMA takes the complex image for down-range and cross-range and calculates the 1D discrete inverse Fourier transform [4]. A scattering center is then defined and the properties at the scattering centers are estimated using a Gaussian density function [4]. The PDF corresponding to the differences  $\{\Delta\zeta, \Delta\mu, \Delta\nu\}$  is based on the PGA derivation [4]. The theory assumes that  $A_k$  is independent of  $N_1, k$  and  $N_2, k$  and each range line has the same statistical properties [4].



Under these assumptions, a covariance matrix in (2.40) is formed from the variance and clutter-plus-noise processes.

$$Q_k = \begin{bmatrix} \sigma_Z^2 + \sigma_N^2 & \sigma_A^2 \exp(-j \Delta \omega_k) \\ \sigma_A^2 \exp(j \Delta \omega_k) & \sigma_A^2 + \sigma_N^2 \end{bmatrix} \quad (1.40)$$

Using Equation (2.40), a new conditional PDF for the index  $k$  is defined and then maximised to estimate  $\{\Delta\zeta, \Delta\mu, \Delta\nu\}$ . For multiple dimensions, AROMA continues to optimize and calculate the partial derivatives of the conditional PDF as a log function in order to solve for  $\{\Delta\zeta, \Delta\mu, \Delta\nu\}$  [4]. The equations are solved using a linear approximation. The consequence of using a linear approximation is that aliasing issues occur [4]. Garren explains, however, that if the temporal interval between radar pulses is sufficiently small and small rotation and translation between pulses occurs, then aliasing does not pose a significant issue to phase approximation [4]. The estimated value between each pulse pair is calculated and the mean is removed for each phase term. The leftover function is integrated along the synthetic aperture to solve for the phase terms. The error between phase terms is added up for each iteration to find the total accumulated error [4].

There are two limitations of AROMA that Garren identifies. The first which is due to the linear approximation that “limits the effective changes in the rotational angle between any two successive radar pulses” [4]. However, this limitation does not apply for large total target rotation angles with the integration of numerous pulses. The second limitation is the occlusion of some target scattering centers during SAR collection. This limitation does not seriously impact the accuracy, since the issues rarely “cause any unacceptable divergence of the estimator” in AROMA overall accuracy [4]. The resulting error estimates are ML based and do not need to be averaged. The major disadvantage of ML process is that AROMA has more difficulty processing sudden motions in error estimation [4].

The last step is to use the error estimates to refocus the target. Refocusing is accomplished by using the error estimation and applying target refocus to numerous iterations as the focus is improved [4]. Target refocusing is performed by computing the phase correction offsets of the defocused target and then applied to each iteration [4]. The

phase factor and phase shift are computed. A 1D Fourier transform is then applied to yield a refocused target image chip for the current iteration [4]. The chip is then used as the input to the next stage of iterations [4]. Figure 9 shows the AROMA process flow chart for each stage of error estimation.

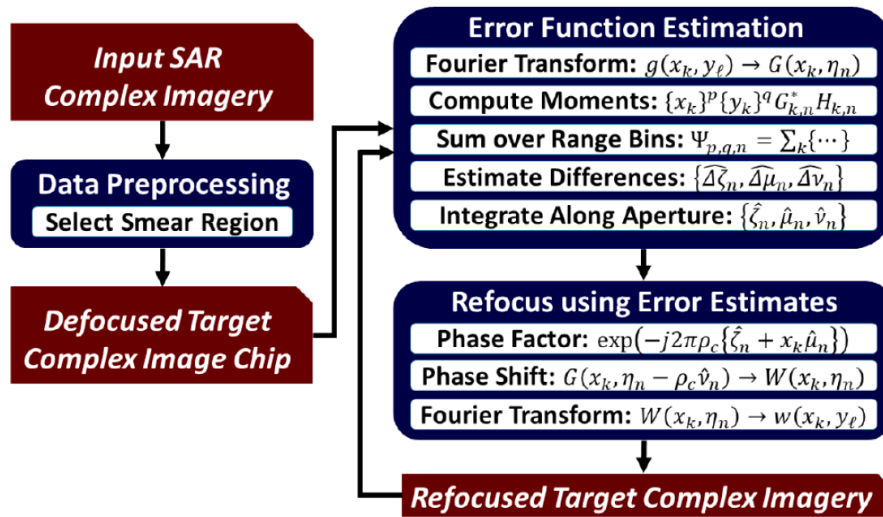


Figure 9. AROMA process diagram. Source: [4].

THIS PAGE INTENTIONALLY LEFT BLANK

### III. ARTIFICIAL INTELLIGENCE CONCEPTS

#### A. MACHINE LEARNING

The rise in popularity of artificial intelligence (AI) for technological purposes has led to a formal defining of the terms associated with this field such as machine learning (ML) or deep learning (DL). AI is a “field focused on automating intellectual tasks normally performed by humans” while “ML and DL are specific methods of achieving this goal” [6] ML and DL methods can solve complicated tasks that require a pattern recognition algorithm. Figure 10 is a visual representation of the subset layers that make up artificial intelligence and data science.

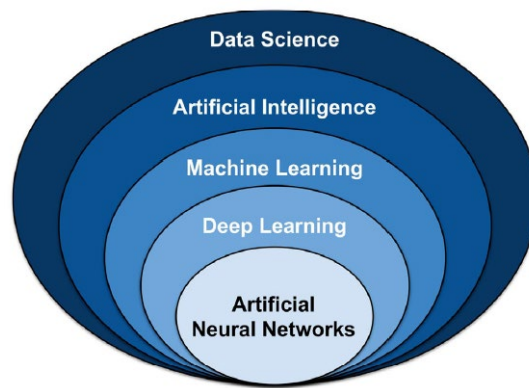


Figure 10. Artificial intelligence and data science layered approach. Source: [6].

Machine learning differs from conventional computer programming because the algorithms learn to solve tasks using past analysis based on features and weights [6]. The weights are derived from the first encounter and are used to produce a new algorithm to predict future datasets [6]. There are four main types of learning which are supervised, unsupervised, semi-supervised, and reinforced learning [6]. Supervised learning is based on patterns in training dataset that are mapped to features in a target [6]. The outcome is compared to a validation set to verify the performance of the algorithm [6]. Unsupervised learning also detects patterns for labeling, however, the patterns may not exist or be defined

for any certain features [6]. Semi-supervised learning is similar to supervised learning, but only a small subset of data is labeled and then used to train the algorithm [6]. The last method, reinforced learning, seeks for a desired outcome rather than a specific answer [6]. The algorithm uses trial and error to model a learning experience rather than data so that it better addresses novel challenges and is not limited by data input [6]. Deep learning is a subset of machine learning that uses multiple layers, such as in a neural network, to learn different levels of image features [7]. Deep learning analysis is complex form of machine learning that mimics the way a human brain learns through the neurons and connections of different layers [8]. The next section will explain the neural network as it is applied to image recognition.

## **B. CONVOLUTIONAL NEURAL NETWORKS**

With the advancement in computing power, machine learning and deep learning rose in popularity and success for labeling large amounts of data [9]. As a subcategory of deep learning, convolutional neural networks (CNNs) have proven to be a highly successful method for image classification [9]. From the early 2000s to now, CNNs have grown in complexity and practical application [9]. Early neural networks used dual-stage processing compared to recent models which use multiple layers, to include the most recent advance called maximum pooling, previously mentioned in Chapter I [9].

As explained in Chapter I, CNNs are a type of artificial neural network that was created to mimic that of visual cortex in the human brain [9]. The visual cortex consists of layers of cells for visual cognition which the artificial neural network seeks to model [9]. The layers in a neural network are stacked together creating a deep learning model [9]. Referring to Chapter I, the three main layers of the neural network are convolutional layer, pooling layer, and fully connected layers [9]. Figure 11 shows the image classification pipeline for a CNN.

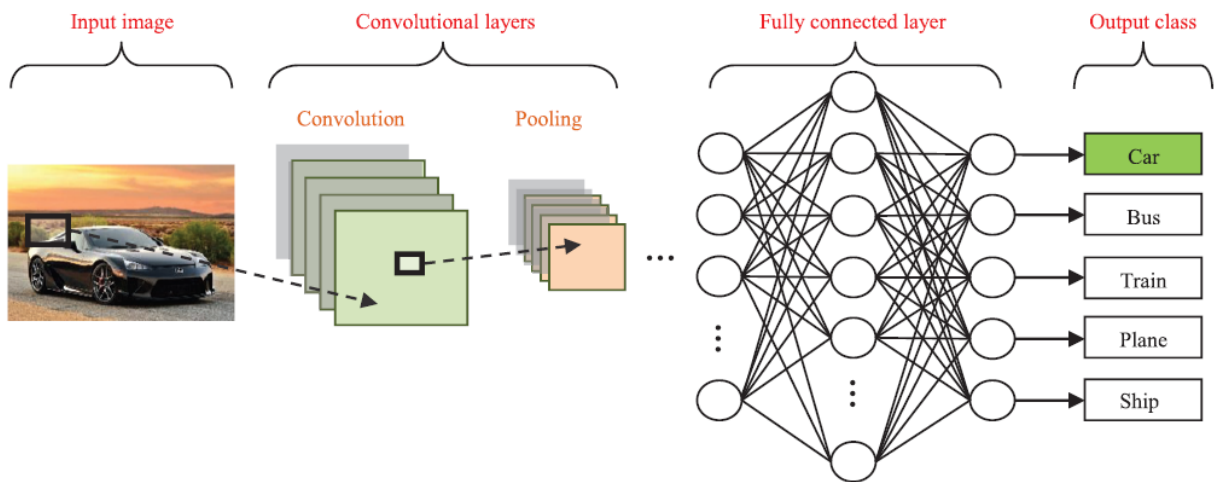


Figure 11. Pipeline for image classification using a CNN. Source: [9].

## 1. Convolution Layer

The convolution layer creates a feature map and calculates training weights that are used for creating the next feature map [9]. This layer learns the features of an input image and creates a map of neurons that is connected to the consequent layers [9]. For each new input, the learned weights are convolved with the feature map to create a new updated feature map [9]. The terminology used for describing a feature map is in terms of kernel and tensor [10]. A kernel is a small number of arrays, which is applied across a tensor [10]. The tensor, an array of numbers, is multiplied with each element of the kernel to create an output value [10]. This output value makes up the feature map that is the foundation of the convolutional layer [10]. The kernels act as feature extractors that are defined by size and number [10]. More importantly, kernels contain the feature patterns that play a role in the training phase of CNN in which the convolutional layer identifies the kernels that best describe a dataset [10]. Figure 12 shows a visual depiction of the convolutional layer where one can see the values for each kernel as it is placed over the input vector and recalculates the new weights.

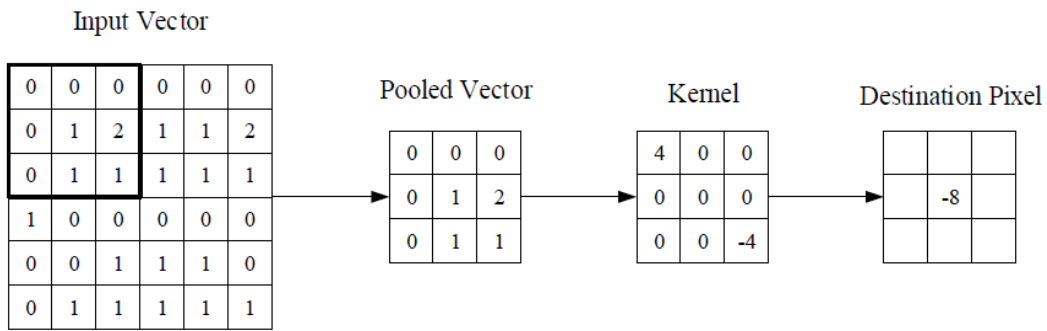


Figure 12. Visual depiction of the CNN process to calculate weights. Source: [11].

## 2. Pooling Layer

The pooling layer function is to “reduce the spatial resolution of the feature maps and thus achieve spatial invariance to input distortions and translates” [9]. Pooling is a type of down sampling that reduces the dimensions of the feature maps to reduce the number of “learnable parameters” without causing distortion to the feature map [9]. Two types of pooling are average and maximum pooling [9]. Average pooling propagates the input average to all neighborhood images in the layer [9]. Average pooling is considered an extreme down sampling that takes the feature map to a size of 1x1 array [9]. This method is typically only used once in the CNN prior to the fully connected layer [10]. Maximum pooling only propagates the maximum value from one layer to the next and gets rid of the other numbers [10]. Maximum pooling is a newer method that is more popular than average pooling [10]. Figure 13 shows how max pooling method calculates the maximum value of a field and average pooling propagates the average of the field for the next layer.

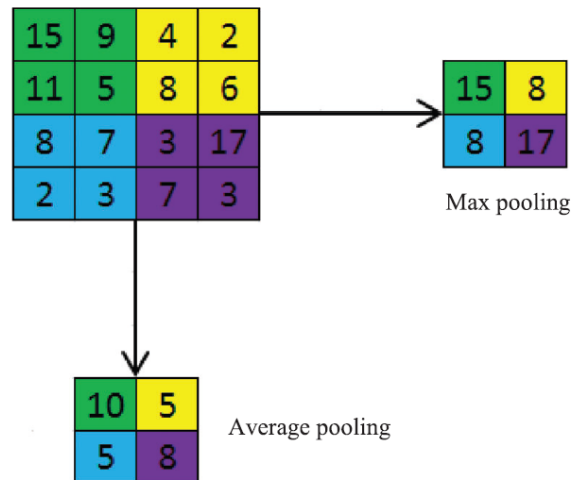


Figure 13. Average pooling compared to max pooling process. Source: [9].

### 3. Fully Connected Layer

The fully connected layers interpret the features from a stack of convolutional and pooling layers for ultimate classification [10]. The fully connected layers contain the final outputs, or probabilities of each class, of the network after the features are extracted and down sampled in the convolution and pooling layers respectively [10]. The parameters for this layer are in terms of number of weights and activation function [10]. The activation function normalizes the output into a probability between 0 and 1 and sums these values [10]. Depending on the classification task required, there are multiple options for choosing the type of activation function, such as sigmoid, softmax, and identity [10]. Table 2 contains a list of the common classification tasks and the activation functions associated with the tasks.



Table 2. Common classification tasks and activation functions. Source: [10].

Task	Last layer activation function
Binary classification	Sigmoid
Multiclass single-class classification	Softmax
Multiclass multiclass classification	Sigmoid
Regression to continuous values	Identity

### C. TRAINING THE NETWORK

Neural networks require training to teach the algorithm how to adjust the weights to maximize performance outcome [10]. The goal of training a network is to find kernels and weights in the fully connected layer that will minimize the predicted and truth classification errors [10]. Backpropagation is the most common algorithm to train the network [9]. Backpropagation “computes the gradient of an objective function to determine how to adjust a network’s parameters” [9] Backpropagation calculates the loss value for chosen kernels and weights, then choses the best performance model [9].

Gradient descent is used in conjunction with backpropagation to update the learnable parameters that are kernels and weights to minimize loss [9]. A gradient for each learnable parameter is created based on learning [9]. The gradient decent shows the direction of the loss function rate [9]. The loss function, also called the cost function, compares the predicted and truth labels for the network [9]. Equation (3.1) defines the gradient in terms of learnable parameter  $w$ , learning rate  $\alpha$ , and loss function  $L$ . The goal of the gradient decent algorithm is to reduce the loss as the learnable parameters are updated, and the gap between predicted and truth labels decreases [9].

$$w := w - \alpha \frac{\delta L}{\delta w} \quad (3.1)$$

The network uses three datasets to perform classification. The datasets are training, validation, and test sets. The training set is used to calculate the loss values and learnable parameters [10]. One issue to avoid with the training set is overfitting [10]. Overfitting occurs when the network starts to memorize the signal noise of the dataset rather than the desired pattern [10]. To ensure overfitting does not occur, the loss should be monitored during training and validation steps [10]. The validation set tests and selects the best training model for classification. The test set is used to evaluate the performance of the end-to-end model.

Training the network plays an important role in the performance of a neural network. As a saying within the machine learning community, “garbage in, garbage out,” the data that is used for each step—training, validation and testing— must be selected to produce the best results and avoid overfitting [10].

THIS PAGE INTENTIONALLY LEFT BLANK

## IV. DATASETS

### A. MSTAR

The foundation of the SAMPLE dataset introduced in Chapter I, was generated using the Air Force Research Laboratory Moving and Stationary Target and Recognition (MSTAR) data set. The MSTAR data set was first collected in September 1995 in Huntsville, Alabama and then again in November 1996 at Eglin Air Force Base, Florida [12]. An X-band SAR radar in one foot resolution spotlight mode was used to collect images of the targets to include articulation, obscuration, and camouflage [12]. Clutter data was also collected for the targets. The 1996 collection expanded the total number of targets from 13 to 15 [13].

The MSTAR public data set contains the clutter, targets, T-72 variants, and mixed targets [13]. The final data set contains 11 target types at 15-, 17-, 30-, and 45 degree depression angles at different aspect angles [13]. The operating condition for each target is provided with the data set and is seen in Table 3 and Table 4. The letters “C”, “S” and the number in the sequence corresponds to the collection number, scene, and depression angle respectively. The clutter data was also taken using a X-band SAR sensor in strip map mode. The clutter library contains 100 full scene images at 15 degree depression for rural and urban backgrounds.

Table 3. Operating conditions for each MSTAR target. Source: [12].

Target Type	Bumper Number	C1S1_15	C1S1_17	C2S1_15	C2S1_16	C2S1_17	C2S1_29	C2S1_30	C2S1_31	C2S1_43	C2S1_44	C2S1_45	C2S2_30	C2S2_45	C2S3_30	C2S3_45
2S1	B01			N		N		N				N				
BMP2	9563	N	N													
BMP2	9566	N	N													
BMP2	C21	N	N													
BRDM2	E71			N		N		N				N			Afs	Afs
BTR60	K10YT7532	N	N													
BTR70	C71	N	N													
D7	92v13015			N		N										
slicy	1			N	N	N	N	N	N	N	N	N				
T62	A51			Cf		Cf										
T72	132	N	N													
T72	812	Cf	Cf													
T72 M	A04			VCf		VCf										
T72 M1	A05			V		V										
T72 M	A07			V		V										
T72 M	A10			V		V										
T72 AV	A32			VCfr		VCfr										
T72 B	A62			VCf		VCf										
T72 B	A63			VCf		VCf										
T72 BE	A64			V		V		V				V			VAth	VAth
T72	S7	N	N													
ZIL131	E12			N		N										
ZSU23/4	D08			N		N		N				N	Atgd	Atgd		

N=Nominal  
 A=Articulation (t=turret, g=gun, h=hatch, f=firing rack, s=sight port, d=dish)  
 C=Configuration (f=fuel barrels, r=reactive armor)  
 V=Version Variant

Table 4. Operating conditions for each MSTAR target. Source: [12].

Target Type	Bumper Number	C1S1_15	C1S1_17	C2S1_15	C2S1_16	C2S1_17	C2S1_29	C2S1_30	C2S1_31	C2S1_43	C2S1_44	C2S1_45	C2S2_30	C2S2_45	C2S3_30	C2S3_45
2S1	B01			274		299		288				303				
BMP2	9563	195	233													
BMP2	9566	196	232													
BMP2	C21	196	233													
BRDM2	E71			274		298		287				303			133	120
BTR60	K10YT7532	195	256													
BTR70	C71	196	233													
D7	92v13015			274		299										
slicy	1			274	286	298	210	288	313	255	312	303				
T62	A51			273		299										
T72	132	196	232													
T72	812	195	231													
T72 M	A04			274		299										
T72 M1	A05			274		299										
T72 M	A07			274		299										
T72 M	A10			271		296										
T72 AV	A32			274		298										
T72 B	A62			274		299										
T72 B	A63			274		299										
T72 B	A63			274		299										
T72 BE	A64			274		299		288				303			133	120
T72	S7	191	228													
ZIL131	E12			274		299										
ZSU23/4	D08			274		299		288				303	118	119		

The last two data sets are the T-72 variants and mixed targets. The T-72 variants were collected using an X-band STARLOS sensor at one-foot resolution spotlight mode [12]. The depression angels for the T-72 variants were 15-, 17-, 30-, and 45 degrees [12]. The mixed targets also used the STARLOS sensor with the same resolution and depression angles. The image chips for the mixed targets were provided as a JPEG file [12]. The MSTAR data sets make up seven CDs that are also publicly available on the Air Force

Research Laboratory (AFRL) Sensor Data Management System (SDMS) website for download [12].

## **B. SAMPLE**

The SAMPLE dataset was used for testing and development of this research. As with Capt. Erik Henegar’s work from 2020, the SAMPLE dataset was chosen over MSTAR since overfitting did not occur while training the CNN [14]. This section covers the background and formation of the SAMPLE dataset from the Air Force Research Laboratory. The SAMPLE dataset was designed specifically for automatic target recognition (ATR) research [3]. The driving factor was to improve on the previous MSTAR twenty years of ATR research [3]. The SAMPLE dataset is more reliable than MSTAR because it standardizes the sensor parameters and configurations [3]. MSTAR contains a limited dataset with varying sensor parameters, target configurations and environmental conditions that do not meet the wide range of operating conditions (OCs) that contribute to broader development of ATR algorithms [3]. SAMPLE dataset uses the SAR imagery from the MSTAR dataset and augments with synthetic data that better represents the OC space [3]. One example is the ability to capture images from various viewing angles and bearings that allows for diversity in the dataset. Especially for CNNs, diverse training data is required to produce better performance. The solution to expanding the OC space is to produce synthetic data that mimics different environments and reflections [3].

There are numerous methods to creating synthetic SAR data such as electromagnetic computation tools and computer aided design (CAD) models [3]. The main challenge, however, is that radar reflections are difficult to model since their reflective properties do not act the same as an electro-optic (EO) model that is often used for creating synthetic SAR data [3]. The SAMPLE data improves on the current MSTAR-based models by better simulating the electromagnetic reflections of the targets. The goal of SAMPLE database is to “bridge the gap between measured and synthetic SAR image.” [3]

## 1. SAMPLE MODEL

The SAMPLE dataset was generated using carefully truthed CAD models and MSTAR SAR images [3]. The merging allowed for the CAD model vehicles to be moved into certain positions that simulated an actual electromagnetic reflection [3]. For each of the various positions, the radar signal was simulated and saved as a SAR image. The images contained elevations between 15 degrees and 17 degrees at different locations. A total of ten vehicle types were chosen for the SAMPLE database and are listed in Table 5 by name and serial number. The CAD models were compared to EO images to validate the truth for each target.

Table 5. Serial number for SAMPLE target vehicle. Source: [3].

Vehicle	2S1	BMP2	BTR70	M1	M2	M35	M548	M60	T72	ZSU23-4
Serial #	B01	9563	C71	0AP00N	MV02GX	T839	C245HAB	3336	812	D08

After removing any discrepancies in the CAD models, the next process was to collect the electromagnetic signatures of the vehicles [3]. The surface material and features were simulated based on the known properties of the vehicles. The background used for all SAR images was a rough surface to create a speckle pattern in the resulting SAR image [3]. A simulated radar pulse and return was recorded for the vehicles over 360 degree azimuth between 14 degree and 18 degree elevation.

After radar collection, the dataset required image formation to create the final SAMPLE SAR database [3]. Each image contains the metadata for target type, radar bandwidth, collection azimuth, elevation, target position, range resolution, and cross-range resolution [3]. The metadata was saved with the pre-existing MSTAR file that also contained the synthetic SAR image [3]. The phase history data and radar pulse locations were pulled from the metadata and run through a Taylor window [3]. The windowing was applied to the synthetic data within the MSTAR file to create a complex image [3]. The complex image was aligned using circular shifting followed by quantization [3]. The image

was quantized into seven levels using a convex hull mask and then aligned with the synthetic image [3]. The final image was saved as a MATLAB .mat file to create the SAMPLE database. The Air Force Research Laboratory (AFRL) released a subset of SAMPLE database for public experimentation which allows access to the .mat files and the metadata for each synthetic image. The metadata provided for each SAMPLE image is shown in Table 6. A comparison with the original MSTAR thumbnail images on the top row and the new SAMPLE images on the bottom row is shown in Figure 14. The publicly released SAMPLE dataset was used in this research for testing the neural network and manipulation of the complex images to test automatic target recognition.

Table 6. SAMPLE metadata information. Source: [3].

Variable	Meaning	Type	Units
azimuth	center azimuth angle	float	degrees
complex_img	complex image data	$128 \times 128$ float	-
elevation	elevation angle	float	degrees
polarization	radar polarization	string	-
range_resolution	range resolution	float	meters
target_name	target name	string	-
xrange_resolution	cross-range resolution	float	meters

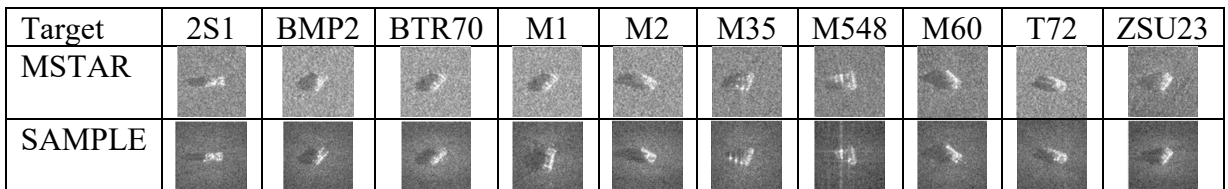


Figure 14. Target types and comparison between MSTAR and SAMPLE data.  
Adapted from [3].



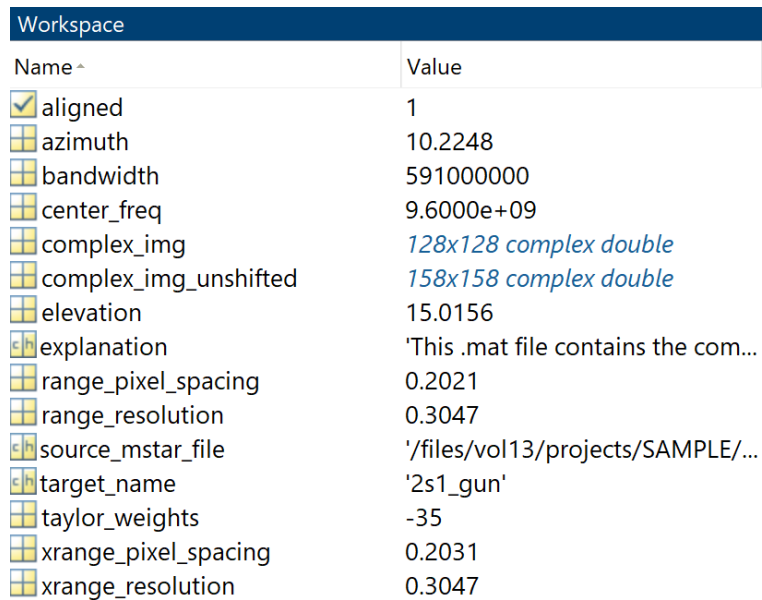
THIS PAGE INTENTIONALLY LEFT BLANK

## V. EXPERIMENTAL DESIGN

### A. SIMULATION

#### 1. Preconditioning

The first steps after receiving the SAMPLE database were to become familiar with the format and metadata followed by preconditioning the SAR images for training the CNN. The SAMPLE data was provided as both a .mat file and PNG image. The database for .mat files contains two versions of the complex image as real and synthetic. The .mat chip for each file contains information that was used to create the image, such as bandwidth, azimuth, and range resolution. Figure 15 shows an example of a 2S1 target information that the .mat file contains when loaded into MATLAB. The .mat format was required for inserting the artificial phase error into the stationary SAR image. Additionally, .mat file format was required for performing the autofocus in the AROMA algorithm. The PNG format was required specifically for training and testing the CNN.



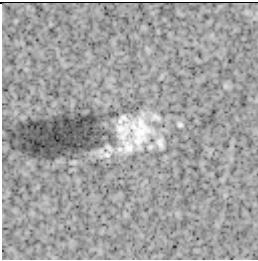
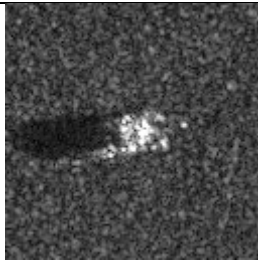
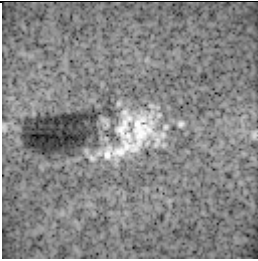
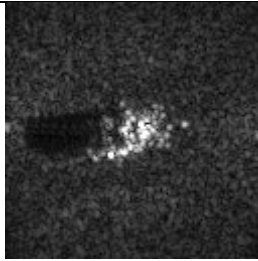
The image shows a screenshot of the MATLAB Workspace window. The title bar is 'Workspace'. Below it is a table with two columns: 'Name' and 'Value'. The table lists various parameters for a target, including 'aligned' (checked), 'azimuth', 'bandwidth', 'center\_freq', 'complex\_img', 'complex\_img\_unshifted', 'elevation', 'explanation', 'range\_pixel\_spacing', 'range\_resolution', 'source\_mstar\_file', 'target\_name', 'taylor\_weights', 'xrange\_pixel\_spacing', and 'xrange\_resolution'. The values are displayed next to each parameter name.

Name	Value
<input checked="" type="checkbox"/> aligned	1
<input type="checkbox"/> azimuth	10.2248
<input type="checkbox"/> bandwidth	591000000
<input type="checkbox"/> center_freq	9.6000e+09
<input type="checkbox"/> complex_img	128x128 complex double
<input type="checkbox"/> complex_img_unshifted	158x158 complex double
<input type="checkbox"/> elevation	15.0156
<input type="checkbox"/> explanation	'This .mat file contains the com...
<input type="checkbox"/> range_pixel_spacing	0.2021
<input type="checkbox"/> range_resolution	0.3047
<input type="checkbox"/> source_mstar_file	'/files/vol13/projects/SAMPLE/...
<input type="checkbox"/> target_name	'2s1_gun'
<input type="checkbox"/> taylor_weights	-35
<input type="checkbox"/> xrange_pixel_spacing	0.2031
<input type="checkbox"/> xrange_resolution	0.3047

Figure 15. Target information for .mat file for each SAMPLE image chip

As with the .mat files, the PNG images also contained real and synthetic libraries. For each library, the images were saved as a *decibel* or *quarter power magnitude (QPM)* format. The difference between the images was the reversible scaling that was applied to the pixel values. The pixel scale for each image was between 0 and 1. *QPM*, or quarter power magnitude, represents the square root of each pixel value that was designed to produce a magnitude image that represented one quarter power. The *decibel* version of the image was a decibel scaling of the magnitude or  $20 \cdot \log_{10}(\text{magnitude})$  and then rescaled back to 0 to 1. Table 7 shows a visual comparison between the decibel and QPM format for real and synthetic of an M1 Abrams tank.

Table 7. SAMPLE PNG for M1 Abrams tank at 16 degree elevation and 010 degree azimuth center.

	Decibel	QPM
REAL		
SYNTHETIC		

For this research, the real .mat and real QPM PNG were used for testing the CNN algorithm. Real data was selected since it modeled a more real-world target than synthetic data when testing for moving targets. The QPM image was selected due to the contrast in grayscale pixel. The contrast between light and dark pixels was more apparent in the QPM images and therefore better suited to train the CNN to look for patterns.

During the first stages of testing, the classification performance using real QPM images with the CNN were unsatisfactory. Upon further investigation, it was found that the QPM images needed additional preconditioning before being processed by the CNN. Chapter VI will explain in detail the issues that occurred using QPM PNG for training and testing the CNN. The final solution required that all of the complex images— stationary, smeared, and focused— to be saved as a magnitude PNG using an `imwrite()` command prior to feeding the images into the CNN. This is due to the way in which MATLAB saves complex images and returns the PNG. The PNG output darkens the background and lightens the reflection of the target. The black and white color contrast is greater than the original QPM image, and therefore must be used to train and test the CNN. Figure 16 shows the output of the complex .mat image after being formatted using `imwrite()`.

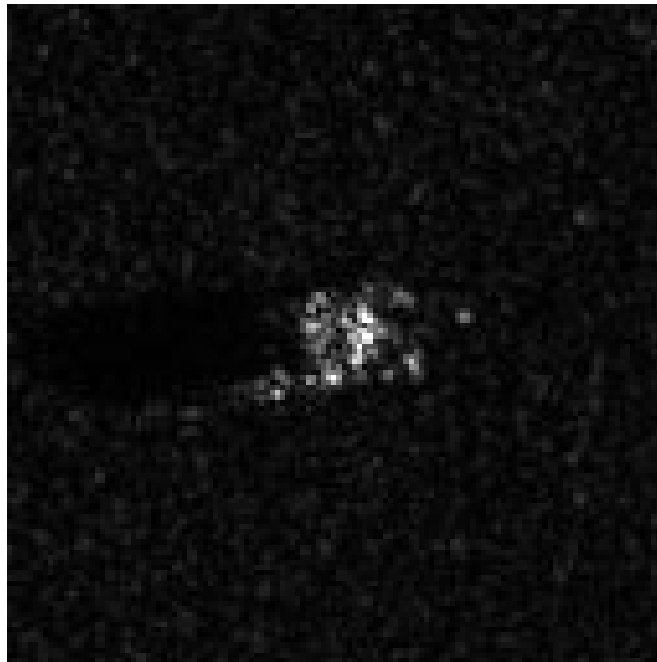


Figure 16. MATLAB output for complex magnitude image of the M1 Abrams tank using the `imwrite()` command

The stationary data provided in the SAMPLE dataset was resaved using `imwrite()` and was used to train the network. All ten types of targets were used during training and testing. All elevations of stationary targets were used for training, while 16 degree elevation

targets were used for testing since every class contained 16 degree images. Additionally, this elevation best represented the medium for the target datasets since the elevation was between 14 degrees and 17 degrees. Table 8 shows the breakdown of the SAMPLE dataset for each target type.

Table 8. SAMPLE dataset class size by elevation degree.

Target	Total Class	14-deg	15-deg	16-deg	17-deg
2S1	174	-	66	50	58
BMP2	107	-	-	55	52
BTR70	92	-	-	43	49
M1	129	26	-	52	51
M2	128	23	-	52	53
M35	129	24	-	52	53
M60	176	-	65	51	60
M548	128	23	-	52	53
T72	108	-	-	56	52
ZSU23	174	-	66	50	58
TOTAL =	1345	96	197	463	539

## 2. Method

The simulation method used the complex image .mat and 128 x 128-pixel SAMPLE images for input into the CNN. The MATLAB Deep Learning Toolbox and Parallel Computing Toolbox were used to construct and run the CNN. Deep Learning Toolbox provided the functions to create specific CNN layers and then training options. Parallel Computing Toolbox was enabled after initial testing to decrease the processing time of classification. The Parallel Computing Toolbox allowed the use of dual central processing units (CPUs) for processing the images through the CNN. The final output of the CNN code was a confusion matrix that displayed the predicted versus actual target classes numerically and color coded. The performance accuracy was given as a percentage correct

out of one-hundred. Both the confusion matrix and accuracy percentage were critical to developing the CNN algorithm because they allowed for direct observations of target trends and CNN performance.

### **3. Inserting Artificial Phase Error**

As discussed in Chapter I, complex target motion was simulated by inserting artificial phase error into the SAR images. Phase error was introduced into the complex image to output a PNG and .mat that contained the smeared pixels. The amount and complexity of the motion was varied throughout testing by adjusting the decay rate and spectrum maximum within the power law computation code. The power law was selected because it was able to simulate complex motion rather than purely translational motion. The complex motion was incorporated by adding an additional decay rate and spectrum maximum values into the phase error equation. In a real target, the complex motion could represent a pitching motion in the downrange direction. The code assumed a center frequency of 1.2 GHz and the default global parameters. The only required input to generate artificial phase error was the complex image format for the desired targets.

The next process for inserting target motion was designed to produce a phase error vector that would be applied to the input image to produce a smearing effect. Additionally, the power law decay constant and spectrum max were manually set prior to running the smear code. The complex images were saved in their class folders and fed into the smear code one at a time. The inverse fast Fourier Transform shift was performed for each complex image to produce a range compressed image. The next process was to compute the artificial power law phase error using the range compressed image. The decay constant and spectrum max were extracted from the AROMA initialized parameters. The transform domain of the phase error was then defined, and the magnitude of the FFT of the phase error was generated with the assumption that the phase error was random from 0 to 360 degrees. Since the phase error is generated at random, the error applied to each image varies for each trial. The corresponding phase error was computed for the right and left half of the image by multiplying the magnitude image by the phase error exponential. The mean of the first derivative was then subtracted from the real-valued phase corresponding phase error. The result was integrated to form the phase, and the truth vectors were generated

along the path length “s” and the elevation angle theta. Figure 17 and Figure 18 show the SAR image of a 2S1 tank before and after inserting the artificial phase error. For this example, the decay rate was set to 50 and the spectrum max set to 100 to simulate a slow-moving target. Figure 19-22 show the artificial phase error vector 1, artificial phase error vector 2, the true elevation angle, and true path length respectively.

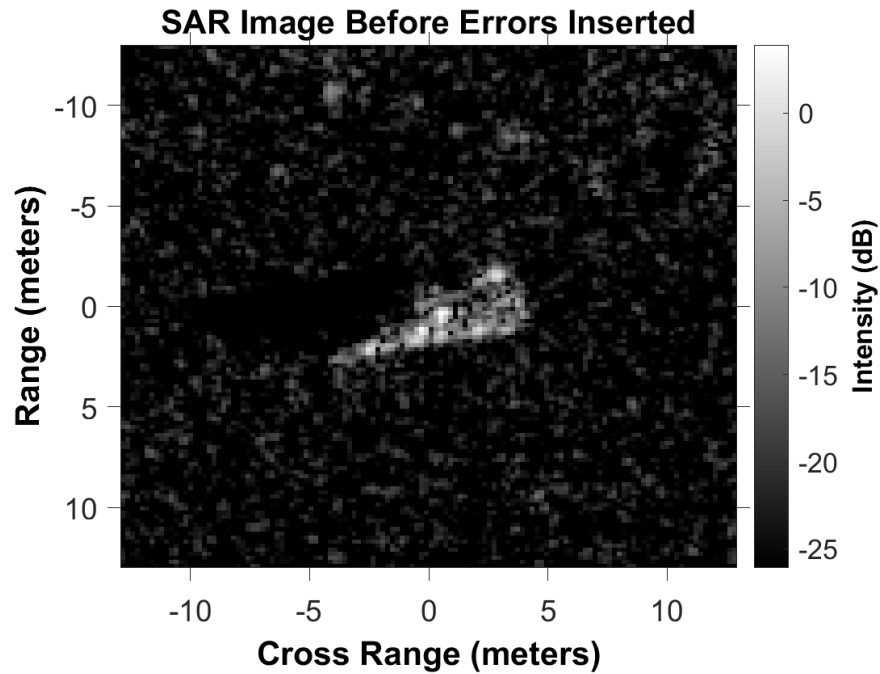


Figure 17. SAR Image of 2S1 tank before inserting artificial phase error

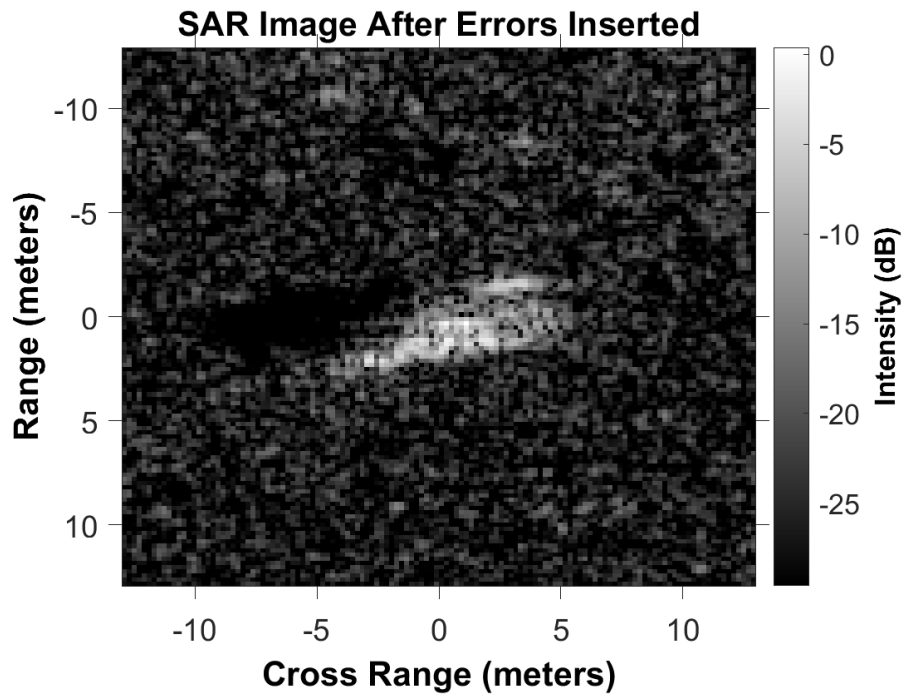


Figure 18. SAR Image of 2S1 tank after inserting artificial phase error via Power Law

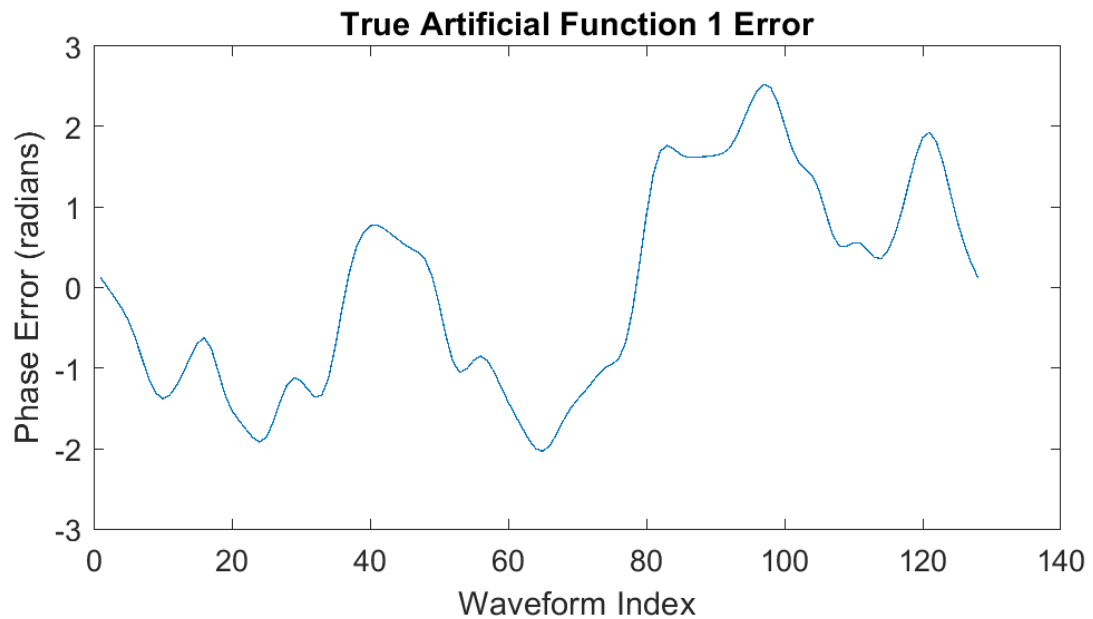


Figure 19. True artificial phase error vector 1



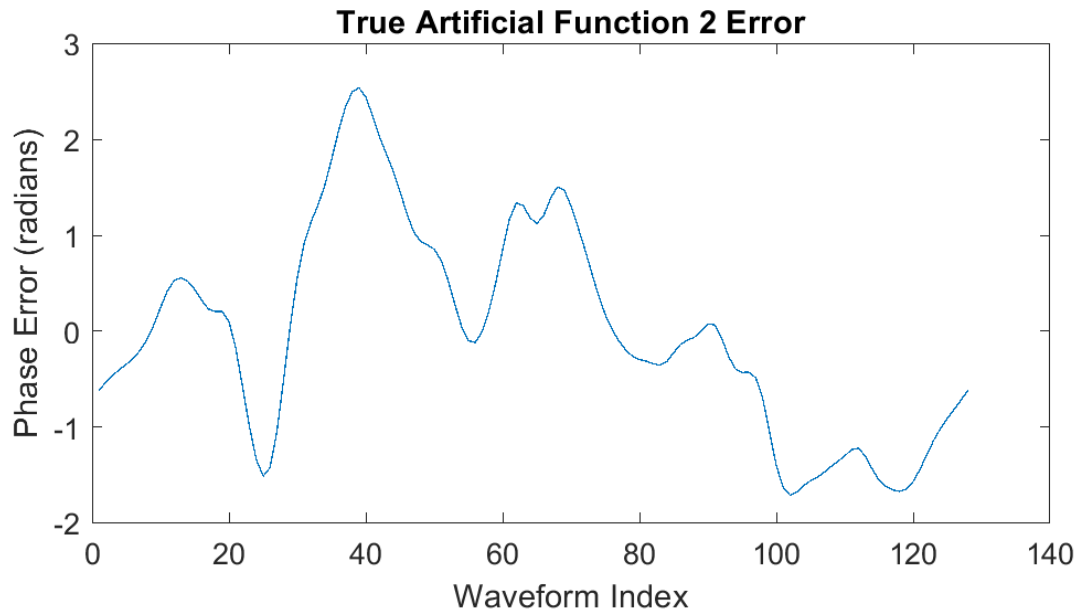


Figure 20. True artificial phase error vector 2

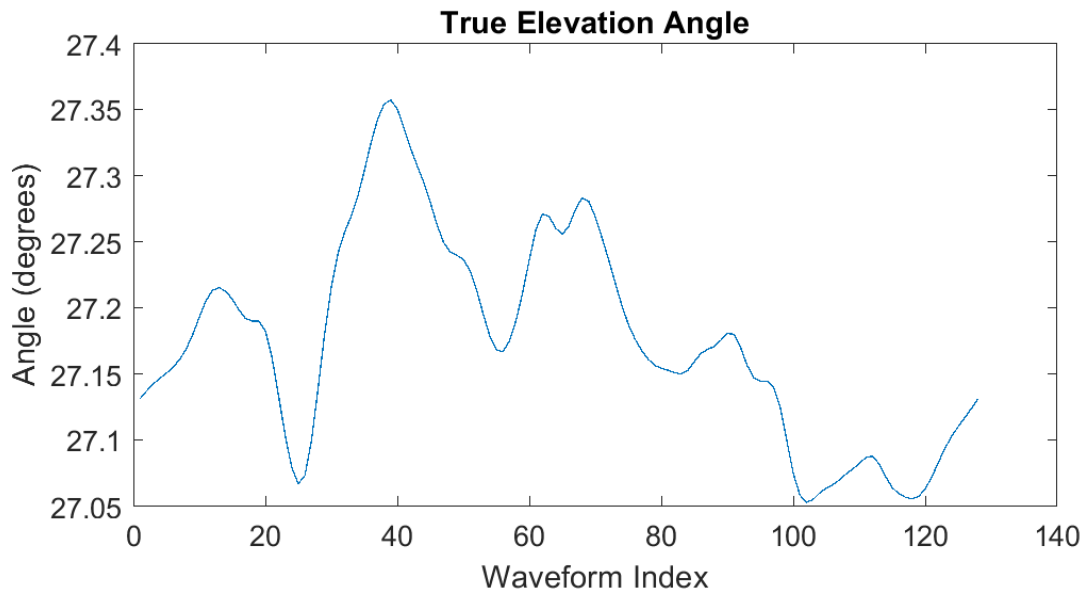


Figure 21. True elevation angle profile

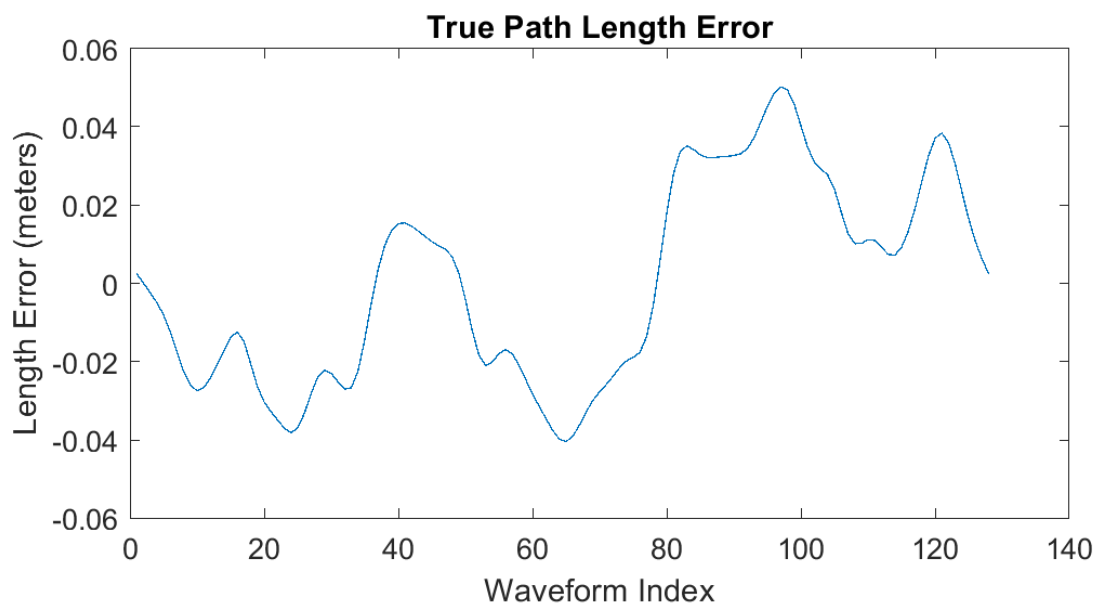


Figure 22. True path length error profile

After the artificial phase error was computed, a separate code was used to insert the error into the SAR image. The linear and constant phase terms were then subtracted from the true phase error. The output yielded the defocused image data which was written to a data structure and saved as a complex .mat file in preparation for autofocus.

#### 4. Performing Autofocus Using AROMA

The next step was to perform autofocus on the smeared images using AROMA. As with the phase error injection, AROMA required the .mat complex image input to perform the necessary calculations for phase error correction. During testing, it was discovered that in order to precondition the data for AROMA, the grid space parameters had to be defined. As AROMA was originally written for a SAR chip image input, the grid space had different dimensions than the SAMPLE dataset that was created. Using the metadata contained in each SAMPLE .mat file, the range and cross range pixel spacing was required to calculate the range and cross range grid size. Similar to the code to produce artificial smearing, the default parameters for ground range mid (30 000 meters), elevation mid (1000 meters), frequency center (1.2 GHz), and waveform speed ( $3 \times 10^8$  meters per second) were kept the same value. Using these default parameters, the spatial frequency range and cross range

grid were computed. The defocused complex image, range and cross range grids, and center frequency were saved into a data structure. The data structure contained the preconditioned SAMPLE data that was properly formatted and ready to be processed by AROMA.

Following preconditioning, the AROMA main driver code initializes the quantities  $\zeta$ ,  $\mu$ , and  $\nu$  then hands off the data for phase error estimation and interpolation [4]. One iteration per image is performed beginning with the phase error estimation. The non-parametric phase error is estimated along the synthetic aperture for a single iteration and the total phase error is then extracted from the resulting structure [4]. Following this step, the maximum intensity pixel on each range line is located and the pixel index value is computed in order to ensure the zero index is in the center of the cross-range position [4]. The image is then converted into the range-compressed and azimuth-spread domain in order to create the phase factor for performing a cross-range shift. The cross-range shift is required so that the dominant pixel on each range line is centered at the middle cross-range position [4]. The image is circularly shifted and a window is applied to extract the desired cross-range pixels. The image is transformed back to the range-compressed and azimuth-spread domain [4]. The algorithm then forms two large subapertures offset by one pixel or less to form relevant interference product matrix. The result is integrated along the aperture length to yield range and cross-range matrices [4]. The real and imaginary components of the product moment vectors and weighted average are computed from the interference product matrix [4]. Further integration along the synthetic aperture then produces a phase error vector for this iteration. The current difference vector is added to the current total difference vector and the mean is subtracted [4]. The new phase difference error vector is integrated to form an estimate of the phase error vector [4]. The mean from the phase error vector is subtracted and is written to the output autofocus structure [4]. A cumulative output phase error vector is created and passed to the next AROMA subroutine.

The final step the AROMA autofocus technique uses a cyclic sinc interpolation to estimate phase error. The cumulative output phase error structure is extracted and provided as input to the subroutine. The total phase error is point multiplied by the range-compressed, azimuth-spread form of the original image [4]. The result is padded and sinc

interpolation is applied [4]. Sinc interpolation provides a shift according to the  $v$  vector and computes the frequency cross-range shift [4]. A loop is created to apply various samples of the sinc function over the cross-range window [4]. The window is applied to the sample points closest to the peak [4]. The analysis creates a  $v$ -compensated range-compressed, azimuth-spread image [4]. The  $v$ -compensated image is point multiplied by a phasor matrix and then transformed to the function domain to form a motion-compensated image [4]. The motion-compensated image data is output to a complex image structure and then saved as a PNG. The output PNGs will provide the testing targets for the CNN. The performance of the CNN will determine how well the autofocus was able to remove the smearing effects caused by target motion within the SAR image. It is predicted that the more complex motion, or larger pixel smear, AROMA will have greater difficulty performing phase error estimation and restoring the image to a stationary representation. Figure 23 and Figure 24 show the SAR image of a 2S1 tank before and after performing autofocus using AROMA. Figure 25 shows the final comparison of stationary, defocused, and focused image.

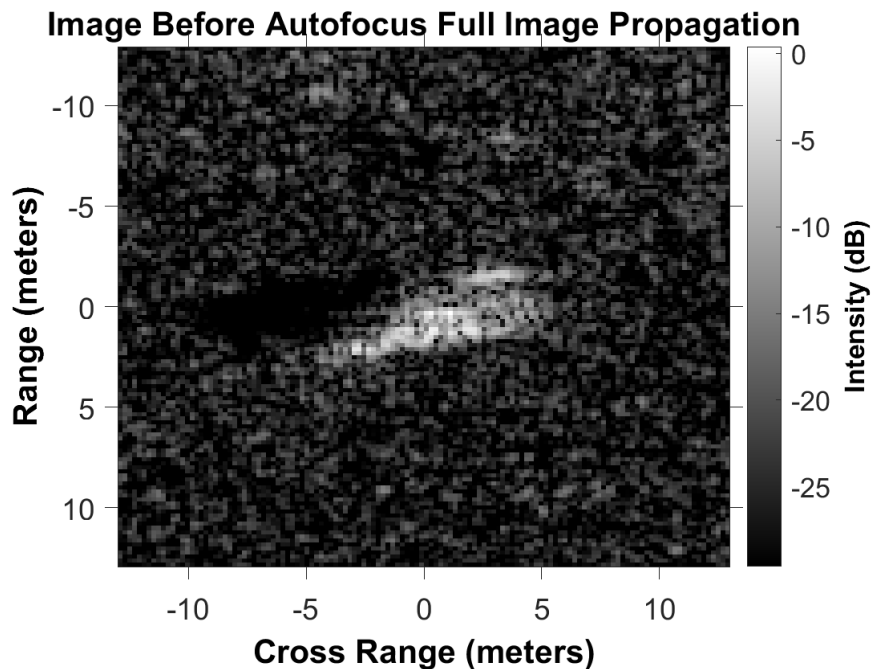


Figure 23. SAR image of 2S1 tank before performing autofocus via AROMA algorithm with artificially injected phase error to simulate target motion.

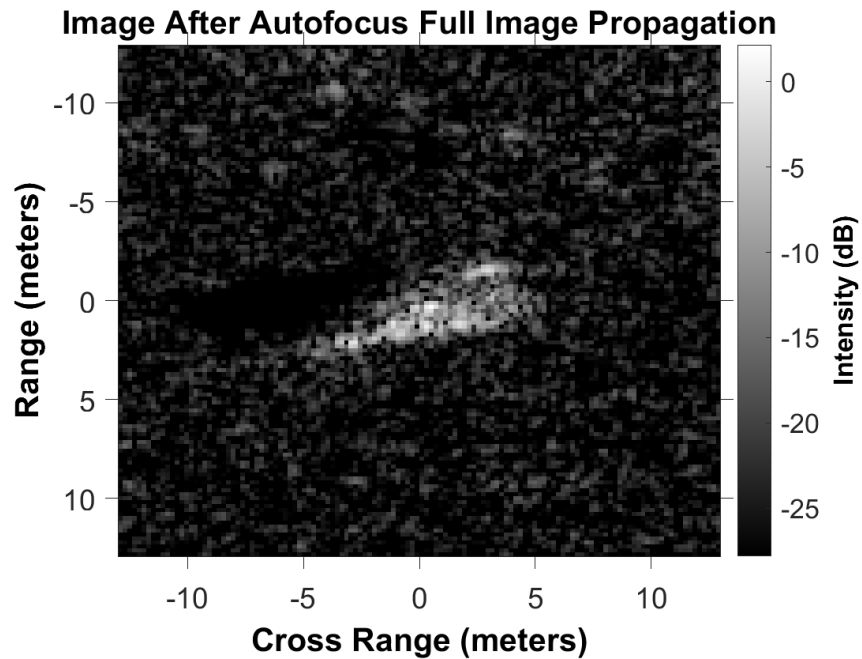


Figure 24. SAR image of 2S1 tank after performing autofocus via AROMA algorithm.

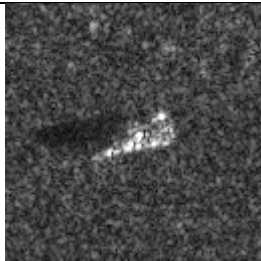
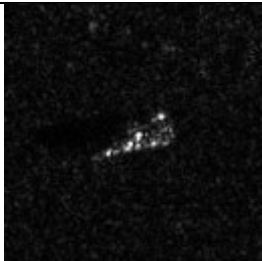
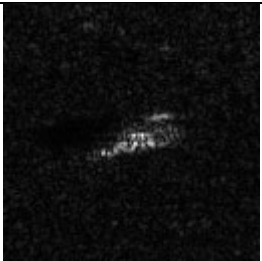
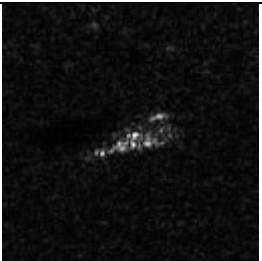
Real QPM Raw	Magnitude Saved	Defocused Power Law	AROMA Focused
			

Figure 25. Comparison of 2S1 tank SAR images through precondition, artificial phase injection, and autofocus process. (DR: 50, SM: 100)

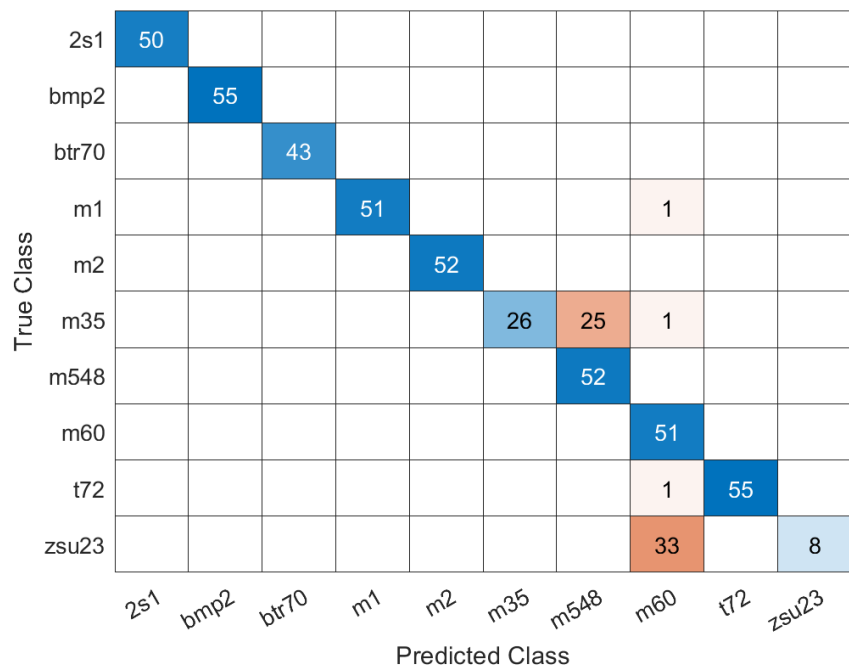


Figure 26. Confusion matrix using AROMA focused images with a DR: 50 and SM: 100. Accuracy 88.10%

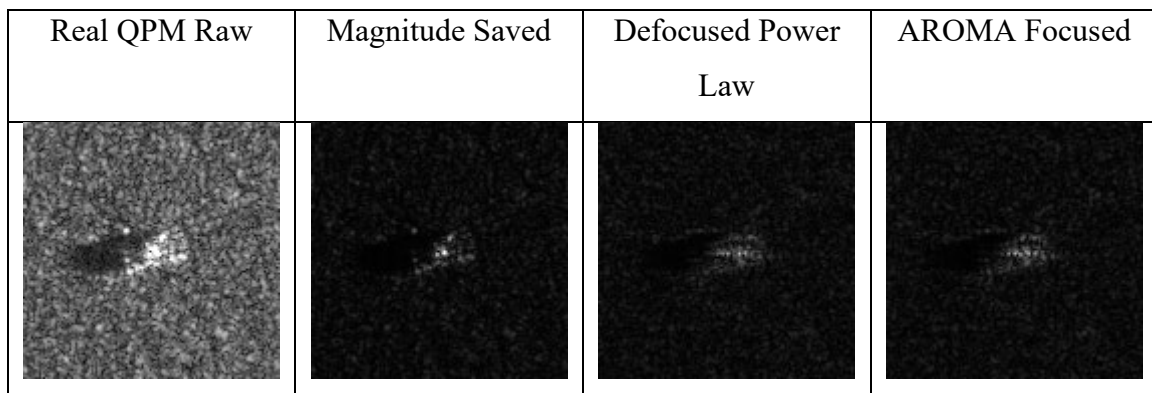


Figure 27. Comparison of BMP2 tank SAR images through precondition, artificial phase injection, and autofocus process. (DR: 100, SM: 300)

True Class \ Predicted Class	2s1	bmp2	btr70	m1	m2	m35	m548	m60	t72	zsu23
2s1	42							5		3
bmp2	4	21		2	15			4		9
btr70	5		24			3		7		4
m1	1			24				16	1	10
m2				1	20	1		25	2	3
m35						23	23	6		
m548						1	51			
m60								50	1	
t72	1			3	1			36	14	1
zsu23	1							34		15

Figure 28. Confusion matrix using AROMA focused images with a DR: 100 and SM: 300. Accuracy 55.36%.

## 5. CNN Architecture

Finally, the refocused images were ready for processing through the neural network. The CNN was constructed using 15 layers total with seven convolution layers, seven pooling layers and one fully connected layer. The convolutional layer is further broken into the convolution linear function, batch normalization layer, and ReLU activation function. The convolution linear function creates a 2D convolution layer with a specified filter size of height and width [15]. For this research, a filter size of  $2 \times 2$  was used and then padded at the edges. The middle value of the convolution linear function denotes the number of filters. The number of filters doubled each layer. The batch normalization layer normalizes each channel across a mini-batch which helps reduce the sensitivity to variations within the dataset [16]. The ReLU layer, or rectified linear unit uses a threshold operator to change any input value less than zero to be set to zero [17]. The maximum pooling layer follows the ReLU layer. The max pooling layer divides the data into rectangular pooling regions and outputs the max value for each region [2]. The pooling size for this operation was  $2 \times 2$  with a stride of 2. The stride represents the step

size of the rectangular region as it traverses the data vertically and horizontally. The next layer, or fully connected layer, contains a softmax layer and classification layer. The softmax layer uses a normalized exponential function to perform multi-class generalization problems [18]. The classification layer creates an output layer that holds the name of the loss function during training, the size of the output and the class labels [19]. The layer computes the cross-entropy loss for classification and weighted classification [19]. Since there were ten classes of armored vehicles and tanks, the number assigned to the fullyConnectedLayer was 10. Figure 29 shows a portion of the code used to create the layers.

```
convolution2dLayer(2,16,'Padding','same')
batchNormalizationLayer
reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(2,32,'Padding','same')
batchNormalizationLayer
reluLayer

fullyConnectedLayer(10)
softmaxLayer
classificationLayer];
```

Figure 29. Portion of CNN algorithm used to create the layer structure

## B. TRAINING AND TESTING

The training and testing phase consisted of three simulations to test stationary targets and moving targets. The stationary target simulation was conducted by both training and testing the network with the SAMPLE images without any artificial phase error. This first simulation created a baseline accuracy and ensured that the CNN was properly classifying the targets. Table 9 shows the breakdown for the training and testing number of images.



Table 9. Number of training and testing images for stationary targets.

Target	2S1	BMP2	BTR70	M1	M2	M35	M60	T72
Training	174	107	92	129	128	129	176	128
Testing	50	55	43	52	52	52	51	52

The second simulation compared performance before and after performing AROMA autofocus. The goal was to show that AROMA improved accuracy after conducting autofocus on a defocused image. The results provided a side-by-side comparison of the SAR images as different degrees of phase error was applied to the stationary target. The number of training and testing images used for the simulation is shown in Table 10. A total of four runs were conducted for this simulation to test different phase error. The classification results will be discussed in Chapter VI.

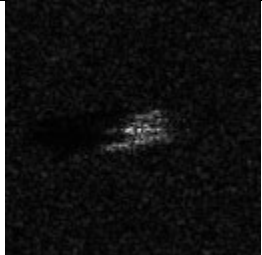
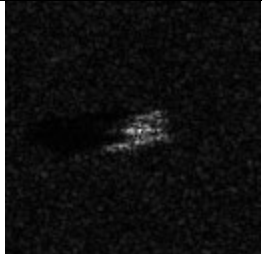
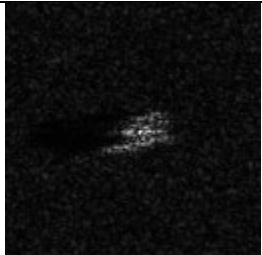
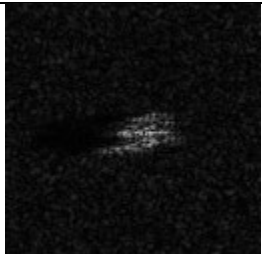
The third simulation tested various degrees of complex motion by adjusting the power law constants during artificial phase error injection. There were nine variations of the power law that were tested. Each variation increased in complexity by increasing the value of the spectrum maximum and decreasing the decay constant of the power law. The power law equation was then applied to the stationary images to create the corresponding phase error. As the intensity of the complex motion was increased, the amount of visible smearing in the images also increased. Table 10 shows the breakdown for the training and testing number images used for the moving target simulation. Table 11 shows the four variations of power law used for testing.

All three simulations created a methodical approach to testing moving target recognition created by artificially injected phase error. Stationary simulation set a baseline level of accuracy for a non-moving target to provide a comparison to that of a moving target. The second simulation provided a first-look into classification results of moving targets. These preliminary findings led to the decision to remove the M60 target from the testing and training during the final classification simulation. The last simulation tested the effects training class imbalance and provided classification results after AROMA autofocusing for nine different power law variations. Results showed that AROMA autofocus technique applied to a defocused image improved classification accuracy after being processed by the CNN.

Table 10. Number of training and testing images for moving targets.

Target	2S1	BMP2	BTR70	M1	M2	M35	M60	M548	T72	ZSU23
Training	174	107	92	129	128	129	176	128	108	174
Testing	50	55	43	52	52	52	51	52	56	50

Table 11. Power law variations and PNG display of smearing effect.

Moving Target Trial	Decay Constant	Spectrum Maximum	PNG of M2
1	50	100	
2	100	150	
3	100	200	
4	100	300	

THIS PAGE INTENTIONALLY LEFT BLANK

## VI. RESULTS AND ANALYSIS

### A. STATIONARY TARGETS BASELINING

The first simulation was conducted to verify that the CNN algorithm was performing as expected. Trials were run on the SAMPLE images in their original state which represented stationary targets. The network was trained using the SAR images captured at 14-, 15-, and 16 degree elevation. Only the 17 degree elevation images for each class were used for testing. As discussed in Chapter V, the accuracy was particularly affected by the M60 target. The first trial shows the stationary trained and stationary tested images with all classes. The second trial shows the same results after removing the M60 target from both training and testing datasets. The results for both stationary target trials are shown in Table 12.

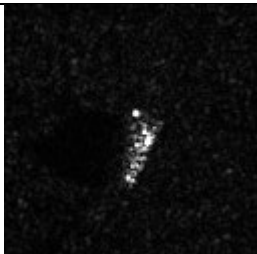
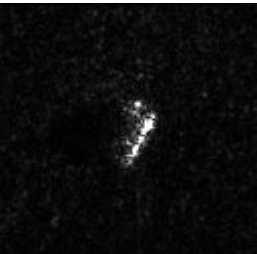
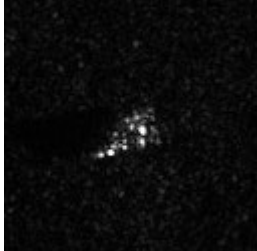
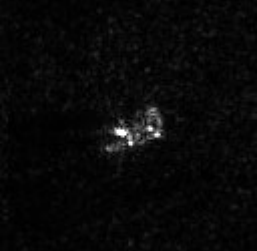
Table 12. Stationary target baseline results with and without M60 target

Target	Trial 1: All Classes			Trial 2: No M60		
	Run 1: Percent Correct	Run 2: Percent Correct	Run 3: Percent Correct	Run 1: Percent Correct	Run 2: Percent Correct	Run 3: Percent Correct
2S1	98.28	96.55	93.10	98.28	94.83	96.55
BMP2	100	100	96.15	98.08	88.46	100
BTR70	100	100	98.04	98.04	98.04	100
M1	96.08	96.08	100	100	94.12	100
M2	96.23	100	96.23	98.11	98.11	98.11
M35	100	98.11	100	100	96.23	100
M548	98.11	100	100	96.23	96.23	100
M60	98.33	75.00	93.33	-	-	-
T72	98.08	100	98.08	98.08	94.23	96.15
ZSU23	0	3.45	13.80	94.83	93.10	100
Average Percent Correct	88.51	86.92	88.87	97.96	94.82	98.98

The results for the first trial with all classes of stationary targets revealed a major concern. The major concern was the poor accuracy for ZSU23 classification. The trial was

run three times to verify the results and accuracy for ZSU23. Both trials, ZSU23 stood out as an outlier with average accuracy of 5.75% correct classification. These results do not compare with the rest of the target classification accuracy. The most likely explanation for the poor classification of ZSU23 is due to its visual shape as it appears in a PNG. The shape, as seen in Table 13, closely resembles the M60 target, which explains a possible reason the neural network classified about half of the targets as a M60 rather than ZSU23 in both runs.

Table 13. Visual comparison of M60 and ZSU23 at 17 degree and 15 degree elevation

Elevation	M60	ZSU23
17 Degrees		
15 Degrees		

Initial testing with stationary targets provided a baseline of classification accuracy for the CNN. The simulation showed that the ZSU23 target was reducing the overall accuracy of the CNN, since the ZSU23 was repeatedly misclassified as an M60. This led to the proposition on whether to eliminate M60 class from the moving target simulations that followed. Since the effects of AROMA autofocus on M60 were not yet explored, the decision was made to keep M60 target in the training and testing datasets for the next simulation. The next simulation, Section 6.2, compared the defocused images and AROMA focused images classification accuracy.

The tests conducted during this section were successful at finding potential classification challenges, such as the M60, and establishing a baseline level of accuracy. The final stationary target classification average accuracy was 97.25% for 9 classes when M60 was removed. This baseline percentage was used in the next section to compare with the moving target classification results.

## B. DEFOCUS AND AROMA FOCUSED PERFORMANCE

The next set of simulations was conducted to test AROMA performance to correct the defocused images after inserting the artificial error. AROMA was predicted to improve classification accuracy after focusing the images and processing through the CNN. The M60 target was included to encompass all targets and to see if AROMA had any effect on correcting the misclassification between the ZSU23 and M60 targets. Four runs were performed with various motion complexity to test how effective AROMA was able to correct phase error. The motion was inserted using the artificial phase error code and varying the spectrum maximum and decay constant in the power law equation. Table 14 shows the run number and the decay constant and spectrum maximum values for the phase error. The table shows a side-by-side comparison of before and after the AROMA autofocus is performed on the smeared image and classified by the CNN.

Table 14. Defocused versus focused SAR images after CNN classification

	Run 1:		Run 2:		Run 3:		Run 4:	
	DC 50/SM 100		DC 75/SM 200		DC 100/SM 200		DC 100/SM 300	
	Defocus: Percent Correct	Focus: Percent Correct	Defocus: Percent Correct	Focus: Percent Correct	Defocus: Percent Correct	Focus: Percent Correct	Defocus: Percent Correct	Focus: Percent Correct
2S1	92.00	96.00	54.00	100	88.00	100	46.00	88.00
BMP2	85.45	90.91	92.59	90.91	80.00	87.27	10.91	70.91
BTR70	83.72	100	83.72	100	83.72	100	25.58	100
M1	84.62	96.15	36.54	76.92	51.92	86.54	17.31	57.69
M2	75.00	94.23	71.15	98.08	40.38	98.08	9.62	38.46
M35	42.31	63.46	63.46	55.80	61.54	75.00	50.00	59.62
M548	100	100	80.80	98.08	100	98.08	84.62	96.15
M60	100	100	90.00	93.33	100	98.04	94.12	92.16
T72	71.43	85.71	82.14	91.07	26.79	82.14	21.43	42.86
ZSU23	20.00	14.00	28.00	20.00	64.00	16.00	16.00	24.00

	Run 1:		Run 2:		Run 3:		Run 4:	
	DC 50/SM 100		DC 75/SM 200		DC 100/SM 200		DC 100/SM 300	
	Defocus: Percent Correct	Focus: Percent Correct	Defocus: Percent Correct	Focus: Percent Correct	Defocus: Percent Correct	Focus: Percent Correct	Defocus: Percent Correct	Focus: Percent Correct
Average Percent Correct	75.45	84.05	68.24	82.42	69.64	84.12	37.56	66.99

The results showed that after performing AROMA autofocusing on defocused images, classification accuracy increased. In order, the classification accuracy increased by 8.6%, 14.18%, 14.48% and 29.43% for each run. Results were as expected due to the increase in smearing effect that causes the target to become unrecognizable and motion too complex to estimate original pixel location. Another finding indicated that many of the targets were being misclassified as an M60 and is evident from the confusion matrixes that show in orange hue along the M60 column indicating incorrect classification. This evidence further reinforces the decision to remove the M60 target from testing and training datasets would benefit classification of the remaining nine targets. The confusion matrixes for the fourth run and power law constants set to 100 decay rate and 300 spectrum maximum shown in Figure 30 and 31 to show how many of the targets were classified as an M60 indicated by the orange boxes.

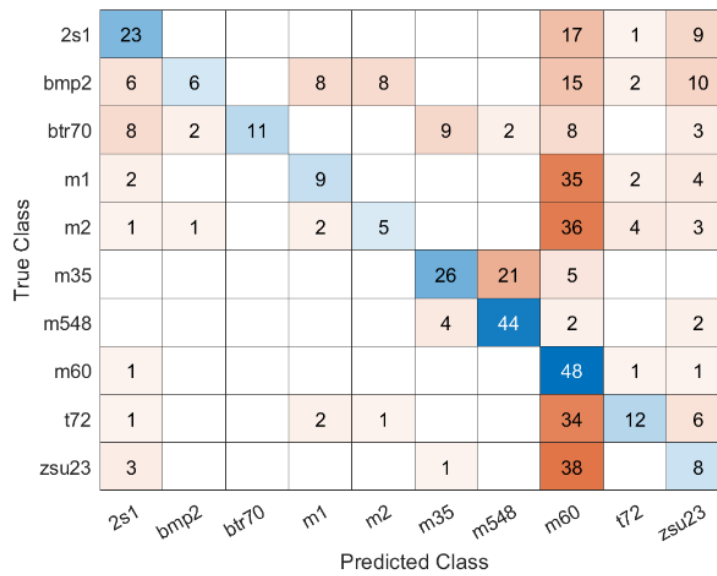


Figure 30. Confusion matrix classifying defocused images for a power law of DC 100/SM 300

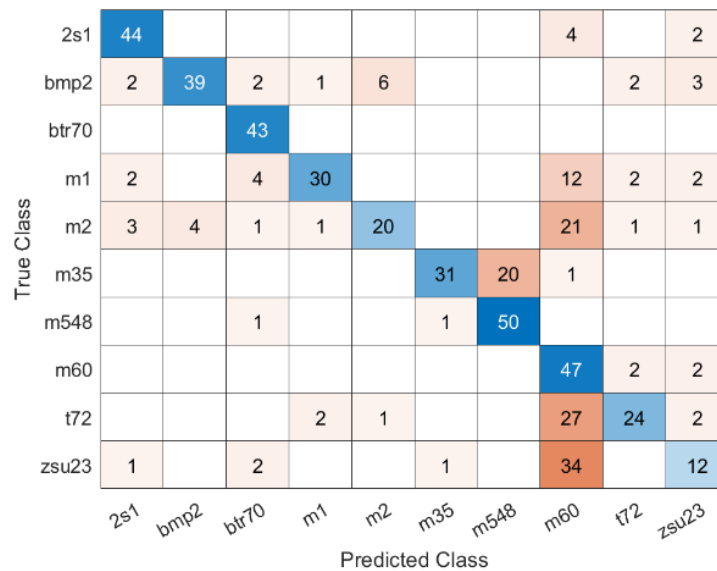


Figure 31. Confusion matrix after AROMA focus performed on images for a power law of DC 100/SM 300



### C. MOVING TARGETS

As presented in Chapter V, the moving target simulation consisted of 18 trials that tested the CNN using different degrees of complex motion and an additional 6 trials to test training class size. A total of nine different power laws were used to create artificial phase error, focused through AROMA techniques, and classified in the CNN. As was discovered in the previous section, several of the targets were getting misclassified as an M60 as the complexity of the motion increased, and particularly the ZSU23. After removing the M60 from the testing, the classification performance improved for the remaining nine targets. In this section, the final simulations to test AROMA autofocus on moving targets was conducted with the omission of M60 target. The results provided a total scope of autofocus and neural network ability to conduct moving target recognition.

The results for the first 18 trials are contained in Tables 15, 16, and 17. Additionally, a visual comparison example between defocused and focused images for the ZSU23 is found in Table 18 with the corresponding power law constants in the left-hand column. In some cases, the defocus effect is more noticeable, especially for the power law of 75 decay constant and 300 spectrum maximum. The decay constant is the number of radar pulses for the spectrum of the phase error autocorrelation to drop to  $\frac{1}{e}$  of its maximum value, while the spectrum maximum affects the amplitude. The more drastic a smear occurs when both the decay constant is small, and the spectrum maximum is large. The complexity of the defocus makes it harder for AROMA to predict the phase error and restore the pixels of the image.

Table 15. Moving target results for first three power law combinations DC 50/SM 100, DC 75/SM 100, and DC 100/SM 100

	DC 50/SM 100			DC 75/SM 100			DC 100/SM 100		
	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct
2S1	88.00	96.00	92.00	100	98.00	99.00	100	100	100.00
BMP2	90.91	96.36	93.64	98.18	92.73	95.46	100	98.18	99.09
BTR70	97.67	90.70	94.19	100	100	100.00	100	100	100.00
M1	94.23	98.08	96.16	100	100	100.00	96.15	100	98.08
M2	84.62	84.62	84.62	96.15	100	98.08	100	96.15	98.08
M35	57.69	55.77	56.73	73.08	84.62	78.85	90.38	94.23	92.31
M548	98.08	98.08	98.08	100	94.23	97.12	98.08	98.08	98.08
T72	96.43	98.21	97.32	100	100	100.00	100	100	100.00
ZSU23	96.00	98.00	97.00	96.00	100	98.00	100	100	100.00
Average Percent Correct	89.29	90.65	89.97	95.93	96.62	96.28	98.29	98.52	98.40

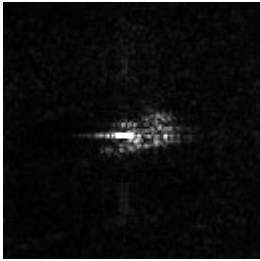
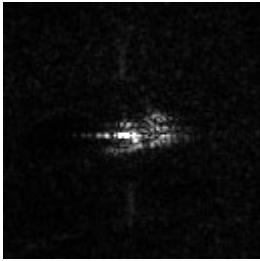
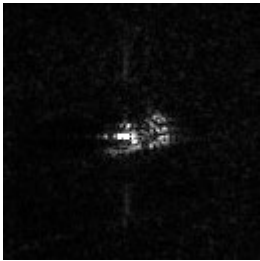
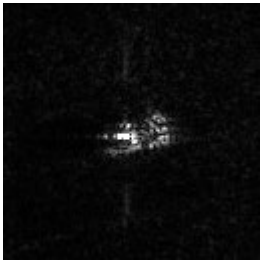
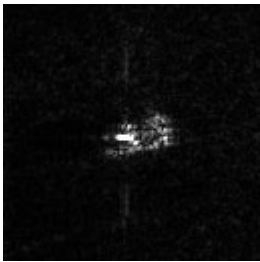
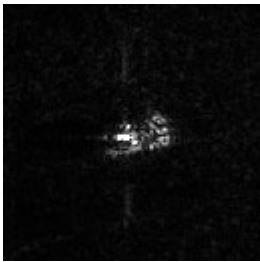
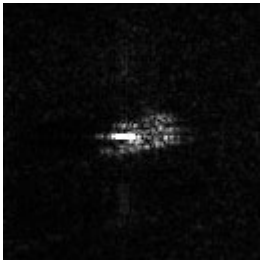
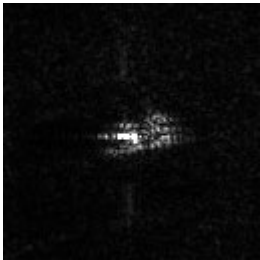
Table 16. Moving target results for second set of power law combinations DC 75/SM 150, DC 75/SM 200, and DC 75/SM 300

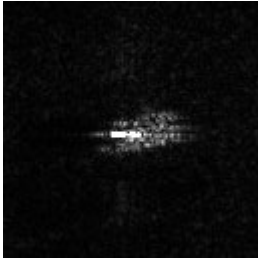
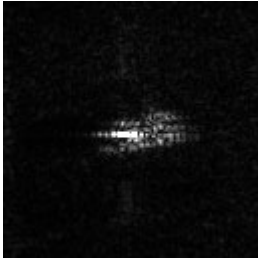
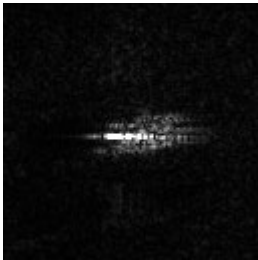
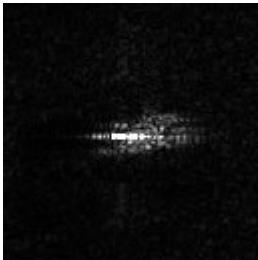
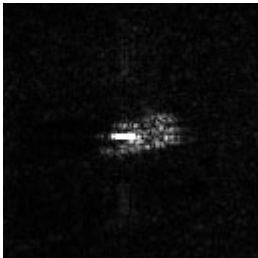
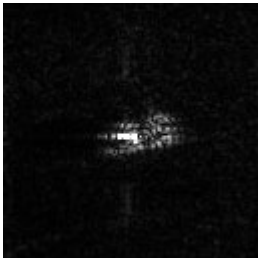
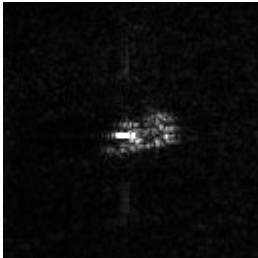
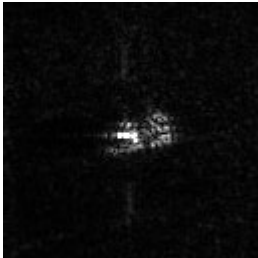
	DC 75/SM 150			DC 75/SM 200			DC 75/SM 300		
	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct
2S1	98.00	96.00	97.00	78.00	86.00	82.00	74.00	68.00	71.00
BMP2	94.55	94.55	94.55	54.55	45.45	50.00	56.36	32.73	44.55
BTR70	100	88.37	94.19	86.05	86.05	86.05	95.35	81.40	88.38
M1	98.08	92.31	95.20	71.15	69.23	70.19	38.46	67.31	52.89
M2	92.31	84.62	88.47	59.62	57.69	58.66	65.38	55.77	60.58
M35	75.00	57.69	66.35	55.77	48.08	51.93	53.85	59.62	56.74
M548	98.08	98.08	98.08	92.31	98.08	95.20	90.38	92.31	91.35
T72	96.43	92.86	94.65	89.29	85.71	87.50	85.71	85.71	85.71
ZSU23	98.00	94.00	96.00	86.00	92.00	89.00	86.00	100	93.00
Average Percent Correct	94.49	88.72	91.61	74.75	74.25	74.50	71.72	71.43	71.57

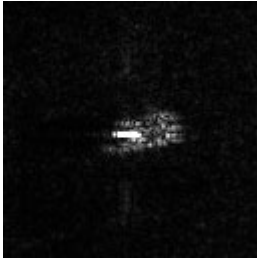
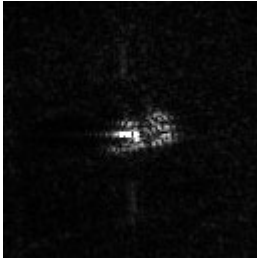
Table 17. Moving target results for third set of power law combinations DC 100/SM 200, DC 150/SM 250, and DC 200/SM 500

	DC 100/SM 200			DC 150/SM 250			DC 200/SM 500		
	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct	Run 1: Percent Correct	Run 2: Percent Correct	Average Percent Correct
2S1	92.00	94.00	93.00	88.00	98.00	93.00	88.00	94.00	91.00
BMP2	85.45	90.91	88.18	94.55	96.36	95.46	72.73	83.64	78.19
BTR70	95.35	90.70	93.03	97.67	95.35	96.51	88.37	93.02	90.70
M1	84.62	94.23	89.43	92.31	98.08	95.20	75.00	86.54	80.77
M2	78.85	88.46	83.66	82.69	90.38	86.54	75.00	75.00	75.00
M35	65.38	67.31	66.35	76.92	78.85	77.89	71.15	69.23	70.19
M548	94.23	98.08	96.16	98.08	98.08	98.08	94.23	100	97.12
T72	87.50	87.50	87.50	92.86	92.86	92.86	78.57	71.43	75.00
ZSU23	90.00	92.00	91.00	92.00	98.00	95.00	88.00	96.00	92.00
Average Percent Correct	85.93	89.24	87.59	90.56	94.00	92.28	81.23	85.43	83.33

Table 18. Visual comparison for defocused and focused ZSU23 target for each power law combination tested during the moving target simulation

ZSU23 Visual Comparison: Defocused and Focused		
Power Law (Decay Rate/Spectrum Max)	Defocused	AROMA Focused
50/100		
75/100		
100/100		
75/150		

ZSU23 Visual Comparison: Defocused and Focused		
Power Law (Decay Rate/Spectrum Max)	Defocused	AROMA Focused
75/200		
75/300		
100/200		
150/250		

<b>ZSU23 Visual Comparison: Defocused and Focused</b>		
Power Law (Decay Rate/Spectrum Max)	Defocused	AROMA Focused
200/500		

During this simulation, a second test was conducted to see if class imbalance affected results. The training dataset size was varied to test whether oversampling or undersampling would affect the CNN classification accuracy. In a journal article written by Mateusz Buda et al., it was found that class imbalance can lead to decrease in performance of CNNs [20]. The recommendation was to oversample to achieve an even class size, which also prevented overfitting [20]. Oversampling involves duplicating samples within a class to increase database size. For this research, this was conducted by randomly selecting various elevation angles of SAMPLE images in each class to make all class sizes equal 176. As predicted, oversampling improved classification accuracy for all tests. Oversampling increased accuracy from baseline average by 4.45%, 14.18%, and 3.35% for the three trials as shown in Table 19.

For undersampling, randomly selected samples were removed from each class to make all class sizes equal 92. The results showed that oversampling and evening class size improved classification accuracy, however, undersampling both increased and decreased accuracy. For all three power laws, the results showed that oversampling performed consistently better than baseline. Further investigation is required to show the effects of undersampling. The likely explanation for undersampling improving classification rather than decreasing is because the SAMPLE dataset is too small to prove Buda et al. class imbalance problem which requires thousands of samples [20]. Table 19 shows the

comparison between baseline, oversampling, and undersampling for three different power laws.

This final simulation to investigate CNN performance and class imbalance proved to be useful at showing how oversampling can lead to increase classification accuracy. For undersampling, the results were inconclusive and further investigation into this theory is required, which was not justifiably explained by Buda et al. in their research.

Table 19. Comparison between baseline, oversampling, and undersampling training datasets after CNN classification using various power law equations

	DC 50/SM 100			DC 75/SM 200			DC 200/SM 500		
	Base Avg	Over sample	Under sample	Base Avg	Over sample	Under sample	Base Avg	Over sample	Under sample
2S1	92.00	98.00	96.00	82.00	96.00	70.00	91.00	96.00	98.00
BMP2	93.64	100	98.18	50.00	85.45	36.36	78.19	80.00	94.55
BTR70	94.19	100	90.70	86.05	93.02	69.77	90.70	86.05	97.67
M1	96.16	98.08	96.15	70.19	88.46	50.00	80.77	82.69	100.00
M2	84.62	94.23	84.62	58.66	94.23	61.54	75.00	78.85	82.69
M35	56.73	65.38	55.77	51.93	57.69	59.62	70.19	73.08	71.15
M548	98.08	98.08	100	95.20	94.23	96.15	97.12	98.08	100.00
T72	97.32	100	98.21	87.50	91.07	80.36	75.00	89.29	83.93
ZSU23	97.00	96.00	94.00	89.00	98.00	96.00	92.00	96.00	96.00
Average Percent Correct	89.97	94.42	90.40	74.50	88.68	68.87	83.33	86.67	91.55



THIS PAGE INTENTIONALLY LEFT BLANK

## VII. CONCLUSION AND FUTURE WORK

### A. CONCLUSION

In an increasingly competitive world, the ability to produce innovative and unique solutions to complex problems becomes an advantage that can make the difference in both military and civilian industries. For synthetic aperture radar, the current challenge to conduct moving target recognition is a valuable function from which the military, especially, can benefit. As introduced, the advantage of SAR over optical sensors is its ability to operate all-weather conditions. This capability, combined with the ability to classify moving targets within SAR imagery, makes SAR a highly desirable and tactical sensor. The use of machine learning techniques, such as a CNN, to conduct moving target recognitions broadens the scope of SAR applications and therefore the resulting intelligence collected from SAR imagery.

This research tested the bounds of AROMA autofocus techniques and the capability to classify targets using the CNN algorithm. Three simulations were conducted to evaluate the overall performance at identifying moving targets. Prior to the simulations, preconditioning of the original SAMPLE images was required to ensure all the formats of the defocused and focused images were in the same format. This discovery was identified early in the development of the CNN algorithm. The preliminary testing yielded poor results well below acceptable accuracy for classifying stationary targets. It was found these results were caused from the format mismatch between the raw SAMPLE data and the smeared and focused data which were saved as a magnitude, complex image PNGs. After saving the processed images using the `imwrite()` command, the background appeared darker, since the SAMPLE data did not contain environment, while the highlights on the vehicle targets were more brilliant and whiter. The solution was to match the image formats by saving the raw SAMPLE data using the MATLAB `imwrite()` command. After ensuring both training and testing data formats matched, the images were ready to be processed by the CNN.

The second finding during the first and second simulations revealed that the M60 target was introducing major classification errors into the other nine targets. During stationary target validation, the ZSU23 was repeatedly getting misclassified as an M60 target. This misclassification was even more evident during the second simulation to test defocus and focus performance. With increasing complex motion inserted into the SAR images, additional targets (especially the M1, M2 and T72) were being misclassified as a M60. The decision to remove the M60 target following the second simulation removed unnecessary errors and the ability to test the remaining 9 targets without the influence of the M60 target. The final third simulation did not use M60 target and provided a better summary of the CNN algorithms performance.

The third preconditioning step required that the training dataset was oversampled to achieve an even class size. Experiments conducted during the third simulation, showed that class imbalance affects the performance. Oversampling the training data to achieve class balance consistently increased classification accuracy. The baseline SAMPLE database class sizes were uneven and performed worse than with oversampling. Undersampling, however, was inconclusive, and both increased and decreased accuracy.

Implementation of these three preconditioning techniques proved to increase total classification performance. This research was also successful at displaying how AROMA processed different degrees of defocus and the resulting classification accuracy. The CNN was then able to quantify the classification after performing AROMA on defocused images. Eighteen different power laws were tested to simulate different types of complex motion. The complex motion was varied by adjusting the decay constant and spectrum maximum. Five of the runs performed below 90% average accuracy which were DC 50/SM 100, DC 75/SM 200, DC 75/SM 300, and DC 200/SM 500. This is likely due to the complexity of the motion when the decay constant is small, and the spectrum maximum is large, that creates a power law wave that oscillates rapidly with a large amplitude. The phase error applied using this type of power law is harder to estimate using AROMA, therefore, the image restoration is not as effective. Further testing is required to determine an exact threshold for the power law at which AROMA is unable to restore a defocused image and reach classification above 90%.

In conclusion, this study quantified the performance of AROMA autofocus on moving targets and offered three preconditioning techniques to improve classification. An efficient process was developed to precondition, insert artificial error, perform autofocus, and classify using a CNN algorithm. This methodical approach allows for a foundation to conduct future research into improving moving target recognition.

## **B. FUTURE WORK**

Further research on moving target recognition and CNN has yet to be uncovered. For this research, the errors caused the M60 target require investigation to provide a complete look at the classification results of ten targets rather than nine targets. Due to the similarity between the M60 target and the ZSU23 target, one possibility is that more data at different angles and elevations needs to be collected on the M60 target to capture any unique features. Similarly, the SAMPLE database provided by the AFRL has since been expanded and is contained in the SAMPLE 2.0 library. Future research using the SAMPLE 2.0 database can broaden the training and testing datasets and a deeper look into the CNN algorithm performance.

Another opportunity for future work would be to explore the class imbalance problem encountered in the last simulation of this research. For any neural network, the training dataset is critical to the overall performance. For this reason, exploration of the effects of training oversampling and undersampling may also lead to improvements in automatic target recognition.

For the CNN, another option for future research is to perform two channel classification. Since the SAMPLE images are saved in a complex format, there is potential to conduct defocus and focus on both channels. The topic of complex-valued neural networks (CVNNs) and deep learning have gained popularity because of the advantages of being able to process both phase and magnitude [21]. This method may yield higher classification accuracy, if applied correctly, using the available code to insert artificial phase error and autofocus.

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF REFERENCES

- [1] P. Berens, “Introduction to Synthetic Aperture Radar (SAR) [RTO-EN-SET-086],” North Atlantic Treaty Organization Science and Technology Organization (NATO S&T), Wachtberg, Germany, 2006.
- [2] N. K. Chauhan and K. Singh, “A Review of Conventional Machine Learning vs Deep Learning,” presented at the International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 2018.
- [3] B. Lewis, T. Scarnati, E. Sudkamp, J. Nehrbass, S. Rosencrantz, and E. Zelnio, “A SAR dataset for ATR development: the Synthetic and Measured Paired Labeled Experiment (SAMPLE),” *Algorithms Synth. Aperture Radar Imag. XXVI*, vol. 10987, May 2019.
- [4] D. A. Garren, “Theory of arbitrary rigid object motion autofocus for non-uniform target rotation and translation,” *IET Radar Sonar Navig.*, vol. 14, no. 11, pp. 1803–1814, 2020.
- [5] C. V. Jakowatz Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*. New York, NY: Springer Science Business Media, Inc, 1996.
- [6] R. Choi, A. Coyner, J. Kalpathy-Cramer, M. Chiang, and J. Campbell, “Introduction to Machine Learning, Neural Networks, and Deep Learning,” *Transl. Vis. Sci. Technol.*, no. 14, Feb. 2020.
- [7] V. Francois-Lavet, P. Henderson, I. Riashat, M. G. Bellemare, and J. Pinaeu, “An Introduction to Deep Reinforcement Learning,” *Found. Trends Mach. Learn.*, vol. 11, 2018.
- [8] R. Ramachandran, D. Rajeev, S. Krishnan, and P. Subarthra, “Deep learning an overview,” *IJAER*, vol. 10, no. 10, pp. 25433–25448, 2015.
- [9] W. Rawat and Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” *Mass. Inst. Technol. Neural Comput.*, no. 29, pp. 2352–2449, 2017.
- [10] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, no. 9, pp. 611–629, Jun. 2018.
- [11] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” in *arXiv*, Dec. 2015.

- [12] A. R. Wise, D. Fitzgerald, and T. Ross, “The Adaptive SAR ATR problem set (AdaptSAPS),” in *Algorithms for SAR Imagery*, Aug. 2003.
- [13] Air Force Research Laboratory, “MSTAR.”  
<https://www.sdms.afrl.af.mil/index.php?collection=mstar> (accessed Jul. 12, 2022).
- [14] E. Henegar, “Applying Convolutional Neural Networks to Identify Moving Targets in SAR Imagery,” Naval Postgraduate School, Monterey, CA, USA, 2021.
- [15] “Convolutional Layer.”  
[https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.convolution2dlayer.html?s\\_tid=srchtitle\\_convolution%20layer\\_1](https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.convolution2dlayer.html?s_tid=srchtitle_convolution%20layer_1) (accessed Oct. 06, 2022).
- [16] “Batch Normalization Layer.”  
[https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.batchnormalizationlayer.html?s\\_tid=doc\\_ta](https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.batchnormalizationlayer.html?s_tid=doc_ta) (accessed Oct. 06, 2022).
- [17] “Rectified Linear Unit (ReLU) layer.”  
[https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.relu.html?searchHighlight=relu%20&s\\_tid=srchtitle\\_relu%20\\_2](https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.relu.html?searchHighlight=relu%20&s_tid=srchtitle_relu%20_2) (accessed Oct. 06, 2022).
- [18] “Softmax Layer.”  
[https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.maxpooling2dlayer.html?searchHighlight=max%20pooling%20layer&s\\_tid=srchtitle\\_max%20pooling%20layer\\_1](https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.maxpooling2dlayer.html?searchHighlight=max%20pooling%20layer&s_tid=srchtitle_max%20pooling%20layer_1) (accessed Oct. 06, 2022).
- [19] “Classification Output Layer.”  
[https://www.mathworks.com/help/deeplearning/ref/classificationlayer.html?s\\_tid=doc\\_ta](https://www.mathworks.com/help/deeplearning/ref/classificationlayer.html?s_tid=doc_ta) (accessed Oct. 06, 2022).
- [20] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Netw.*, no. 106, pp. 249–259, 2018.
- [21] J. Bassey, X. Li, and L. Qian, “A Survey of Complex-Valued Neural Networks,” *arXiv*, vol. 2102, no. 12249, Jan. 2021.

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California





## DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

[WWW.NPS.EDU](http://WWW.NPS.EDU)

---

WHERE SCIENCE MEETS THE ART OF WARFARE