NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DATA AUGMENTATION FOR SYNTHETIC APERTURE RADAR USING ALPHA BLENDING AND DEEP LAYER TRAINING

by

Alexander W. Denton

March 2023

Thesis Advisor:	David A. Garren
Co-Advisor:	Brij N. Agrawal
Second Reader:	Jae Jun Kim

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE DATA AUGMENTATION FOR SYNTHETIC APERTURE RADAR USING ALPHA BLENDING AND DEEP LAYER TRAINING	**5. FUNDING NUMBERS**
6. AUTHOR(S) Alexander W. Denton	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES 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.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Human-based object detection in synthetic aperture RADAR (SAR) imagery is complex and technical, laboriously slow but time critical—the perfect application for machine learning (ML). Training an ML network for object detection requires very large image datasets with imbedded objects that are accurately and precisely labeled. Unfortunately, no such SAR datasets exist. Therefore, this paper proposes a method to synthesize wide field of view (FOV) SAR images by combining two existing datasets: SAMPLE, which is composed of both real and synthetic single-object chips, and MSTAR Clutter, which is composed of real wide-FOV SAR images. Synthetic objects are extracted from SAMPLE using threshold-based segmentation before being alpha-blended onto patches from MSTAR Clutter. To validate the novel synthesis method, individual object chips are created and classified using a simple convolutional neural network (CNN); testing is performed against the measured SAMPLE subset. A novel technique is also developed to investigate training activity in deep layers. The proposed data augmentation technique produces a 17% increase in the accuracy of measured SAR image classification. This improvement shows that any residual artifacts from segmentation and blending do not negatively affect ML, which is promising for future use in wide-area SAR synthesis.

14. SUBJECT TERMS artificial intelligence, AI, machine learning, ML, convolutional neural network, CNN, data augmentation, AConvNet, SAMPLE, MSTAR, RADAR, synthetic aperture radar (SAR), alpha blending, low-resolution, wide area search, background, tip-and-cue, field of view, FOV, feature detection, object detection, target detection, object proposal, target proposal, synthetic dataset, swath, area search, discovery, patch, chip	**15. NUMBER OF PAGES** 125
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

DATA AUGMENTATION FOR SYNTHETIC APERTURE RADAR USING ALPHA BLENDING AND DEEP LAYER TRAINING

Alexander W. Denton
Major, United States Air Force
BS, United States Air Force Academy, 2008
MS, University of Alabama, 2017

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
March 2023

Approved by: David A. Garren
Advisor

Brij N. Agrawal
Co-Advisor

Jae Jun Kim
Second Reader

James H. Newman
Chair, Space Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Human-based object detection in synthetic aperture RADAR (SAR) imagery is complex and technical, laboriously slow but time critical—the perfect application for machine learning (ML). Training an ML network for object detection requires very large image datasets with imbedded objects that are accurately and precisely labeled. Unfortunately, no such SAR datasets exist. Therefore, this paper proposes a method to synthesize wide field of view (FOV) SAR images by combining two existing datasets: SAMPLE, which is composed of both real and synthetic single-object chips, and MSTAR Clutter, which is composed of real wide-FOV SAR images. Synthetic objects are extracted from SAMPLE using threshold-based segmentation before being alpha-blended onto patches from MSTAR Clutter. To validate the novel synthesis method, individual object chips are created and classified using a simple convolutional neural network (CNN); testing is performed against the measured SAMPLE subset. A novel technique is also developed to investigate training activity in deep layers. The proposed data augmentation technique produces a 17% increase in the accuracy of measured SAR image classification. This improvement shows that any residual artifacts from segmentation and blending do not negatively affect ML, which is promising for future use in wide-area SAR synthesis.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

2D	two-dimensional
AI	artificial intelligence
ATR	automatic target recognition
CAD	computer aided drafting
CFAR	constant false alarm rate
CNN	convolutional neural network
CT	computerized tomography
DL	deep learning
EEG	electroencephalogram
EO	electro-optical
EOC	extended operating conditions
FFT	fast Fourier transform
FOV	field of view
GSD	ground sample distance
IFFT	inverse fast Fourier transform
LLC	layer look chart
ML	machine learning
PCC	percent correct classification
RADAR	radio detection and ranging
RF	radio frequency
RGB	red, green, blue
RGBA	red, green, blue, alpha
RMS	root mean square
SAR	synthetic aperture RADAR
SOC	standard operating conditions
SRC	sparce representation-based classification
SVM	state vector machine

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This work would not have been possible without the additional guidance and support of Dr. John A. Richards of Sandia National Laboratories. Thank you very much.

Additionally, I owe a great deal to my partner, Martha Dowell. Without her endless support and encouragement, I would never have been able to complete such an endeavor. Thank you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Since the early days of flight, aerial imagery has been critical for reconnaissance. The technology progressed from wet-film to electro-optical (EO), from aerial to space-based, but still required a sun-lit and cloud-free view. The synthetic aperture RADAR (SAR) alleviated those problems, collected additional unique types of information, and cemented itself as a crucial data source. But SAR imagery has its own issues. Because of the novel way that the information is collected, SAR images are non-literal; they are fundamentally different than the EO images people are used to interpreting.

The exploitation of SAR imagery has always required special training. Even with such training, SAR images are non-intuitive for humans. This makes the interpretation process very slow. To solve this problem, researchers have spent decades developing so-called automatic target recognition (ATR) processes. While many advances have been made, ATR is still far from ideal. Because it is developed by humans using adaptations of familiar concepts, the scanning and template matching techniques of traditional ATR carry forward many of the limitations of human interpretation. Computer vision—essentially the EO equivalent of ATR—has had similar problems in the past.

Computer vision is now in the midst of a revolution. The advent of so-called artificial intelligence (AI) and, specifically, the subdiscipline of machine learning (ML) has put the template development problem in the hands of neural networks. Newer AI/ML networks have even supplanted the laborious scanning techniques in EO. But SAR is still nonliteral; special training requirements make it less approachable. This makes SAR usage less common and, subsequently, less developed in AI/ML. But SAR ATR can still take advantage of the rapid progress of AI/ML in EO's computer vision field. Rather than retrace the progress of EO, this thesis will adapt several computer vision AI/ML techniques for SAR ATR use.

A. BACKGROUND

There are many ways to approach any problem, and low-resolution SAR search is no exception. This thesis capitalizes on research in other disciplines to make a small leap

forward in the field of ATR. This path and the reasoning behind it can be answered through a series of three smaller questions:

1. Why Apply AI to SAR Imagery?

SAR imagery provides important information in circumstances where EO collection is not available, not possible, or simply not preferred. But the requirement for human-based processing and exploitation of SAR imagery forces long timelines due to differences in phenomenology. These differences are generally detrimental for human observers but can provide unique opportunities for ML. Both the analysis and timeliness limitations can be overcome by removing the common denominator—the human. By using AI/ML, opportunities are created to reach beyond human comprehension and enable rapid interpretation of massive amounts of SAR data.

Expanding human comprehension is the easier of the two tasks for ML. Whereas the red, green, and blue (RGB) channels in an EO image would each produce a readily discernable greyscale image on their own, SAR produces no literal image at all [1]. SAR imaging requires significant processing to convert the spatial frequency domain return into a two-part representation: magnitude and phase. The magnitude portion resembled a greyscale image at first glance but requires months of special training to reliably interpret. Even then, exploitation of magnitude SAR imagery is tiresome and fraught with human error. There are many non-literal artifacts unique to SAR such as layover, backspreading, and resonance. These differences may seem extreme compared to EO imagery processing, but the problems with the phase portion are much worse. In fact, to the human observer, the phase representation of SAR looks like noise and is usually discarded! But unique details like range are buried in the SAR phase data [1]. An AI algorithm has no external experience, no confirmation bias, no fear of failure, and no EO experience holding it back. It learns to interpret magnitude and phase in lockstep. An AI/ML can be taught what no human ever could—to detect objects in phase data.

The second task—speeding up processing and exploitation—is the issue this study will address. An ML model, being a computer algorithm, can search large images very quickly. However, it can only do so reliably if it has been well-trained. The key to quality

training is having a large, diverse training dataset. Unfortunately, there is no such dataset for SAR. The same phenomenology differences that plague the human observer affect the ML development community. Much like humans experienced with EO have difficulty interpreting SAR, the datasets and augmentation techniques developed for EO imagery do not directly translate to SAR. While EO has enjoyed the rapid development of many compatible data augmentation techniques and complex ML models, the lack of datasets has prevented any similar surge for SAR development. Without large or diverse datasets capable of effectively training an ML model, novel augmentation techniques are required to expand the existing SAR datasets.

AI has the potential to greatly increase the utility of SAR imagery not only in the widely studied object classification tasks but also in the critical object detection tasks. In fact, the object detection task has immense potential utility because it allows rapid interpretation of vast amounts of data. By expanding the limited SAR training datasets, this study will take one of many small steps toward realizing that goal.

2. Why Create a Data Augmentation Workflow?

Supervised training of an AI model requires very large datasets with labeled objects. Publicly available SAR datasets, such as the measured MSTAR [2] and synthetic SAMPLE [3], are too small for robust ML training. Furthermore, there are no large SAR datasets of wide field of view (FOV) scenes with labeled objects. Though AI has shown promise classifying narrow, high-resolution SAR images [4], [5], these classification methods fail in field trials because they do not address the object detection problem [6]. Wide-area SAR search still relies on older techniques like constant false alarm rate (CFAR) [7]. AI has the potential to dramatically improve the speed and accuracy of wide-area SAR search if larger datasets are available. Data augmentation, an umbrella term for a large category of techniques, can be used to rapidly expand the training effectiveness of AI datasets [8].

There are several reasons why large SAR datasets do not exist. Because SAR imagery is self-illuminated, high-resolution collections require a tradeoff between long dwell times, high power, or narrow-FOV. This is fundamentally a sensor management

problem [9]. Chip images suitable for object classification can be collected by high resolution SAR spot imagery [9], [10], [11], but these spot images do not provide coverage sufficient for wide-area search and detection. [12]. Because the nature of SAR imagery requires scene illumination and protracted sensing, construction of a SAR image requires considerable power and time [13]. This necessitates that large area search imagery be collected at a resolution too coarse for most classifiers to function.

Low-resolution SAR images with a wide-FOV could still be used in a tip-and-cue construct. An AI/ML object detector using less resource-intensive, low-resolution, wide-area SAR swath images could tip recollection of specific, narrow-FOV chips. Afterward, a human analyst or second AI could classify that smaller set of high-resolution chips. This study works to enable such a tip-and-cue workflow by expanding the SAMPLE dataset using low-resolution data augmentation. This, in turn, will enable fabrication of synthetic, wide-FOV, low-resolution SAR images suitable for training object detectors.

Which data augmentation techniques should be used? There are many existing data augmentation techniques from which to choose [8]; unfortunately, most are specifically designed for three-channel EO images and do not translate well to SAR. For example, geometric transformations such as flips and rotations are not useful because the azimuth is fixed during SAR image formation. The shadows are always cast away from the sensor (in broadside collection) and the sensor can always be modeled on the same side of the reference frame [1]. Flips and rotations are, therefore, neither physically realistic nor representative. If flips or rotations are used to augment a SAR dataset, they will add confusing variability in the shadows and layover. Because EO data augmentation methods like this are largely nontransferable, novel techniques are required to enable robust training with small SAR datasets. This thesis develops an all-new data augmentation technique utilizing segmentation and alpha blending to change the background texture. This new technique is evaluated on a simple classification network but is intended to enable the creation of wide-area synthetic SAR image.

3. Why Use a Simple Classification CNN?

The field of AI/ML has been improving at an exponential rate in the last decade [4], [14]. Many nuanced architectures have increased the robustness of convolutional neural network (CNN) classifiers by crosslinking layers. Single-stage detectors like YOLO simultaneously detect and classify in elaborate, fully trainable backbone-neck-head configurations [15], [16], [17]. This study, however, chooses to use the simple AConvNet classification CNN [18] for several reasons. Each of those terms—simple, classification, CNN—can be unpacked to explain the selection of AConvNet.

First, the use of a simple, shallow, consistent architecture makes the results more approachable. AI/ML, being the intersection of many disciplines, is inherently complex. The main issue to be solved is *overfitting*: "when a network learns a function with very high variance such as to perfectly model the training data" [8]. This problem can be addressed by modifying the architecture, modifying the training data, or injecting functional solutions like dropout and transfer learning [8]. Architectures can be statistically modeled, but the effects of data augmentation and functional solutions depend on complex network behavior and generally requires empirical testing. Simple architectures serve as an analog for first-principles analysis during these empirical tests, allowing wholistic behaviors to be observed in relative isolation. A simple architecture like AConvNet, which uses the ReLU activation function with only five trainable convolution layers, reduces the degrees of freedom inherent in the network. Additionally, because AConvNet is intended for SAR datasets, it provides a strong performance baseline in the literature [18].

Second, the use of a classification network simplifies the problem being solved and verifies the data augmentation technique for future use. While there would be more immediate operational utility in training a single-stage detector to find objects in wide-area SAR, those networks must find *and* classify objects—two separate problems. In fact, two-stage object detectors like RCNN use a CNN for classification [19]. Verifying a data augmentation technique for classification use does, therefore, have direct applicability in that use case. Either way, without a wide-area SAR dataset of labeled objects, there is no way to train either a single-stage or two-stage object detector. Therefore, given a prepared dataset of object chips, classification is the only task that can be directly applied.

Furthermore, when testing a data augmentation technique, a simple classification task allows verification of performance enhancement.

Finally, the "C" in CNN stands for *convolution*, an applied kernel method that creates filter windows. Convolution methods are applied to imagery AI/ML tasks to simplify earlier, fully connected networks by creating a sparce connection that reduces the number of trainable parameters between layers [19]. Non-linear down-sampling in pooling layers also greatly reduces the number of trainable parameters [19]. AConvNet uses both techniques and, more importantly, no others. By using AConvNet, whose building blocks are restricted to these two forms, the results of this study can be more easily generalized to other use cases.

B. RESEARCH PROBLEM

AI/ML techniques should be applied to SAR ATR in order to enable wide-FOV search. However, all existing SAR datasets are either narrow-FOV, unlabeled, or too small for robust training. Because collection of wide-area SAR images is very costly, and because labeling of real-world targets might affect the releasability of such images, the best solution is to synthesize a training dataset. There is currently no mechanism available to synthesize labeled, wide-area SAR images suitable to this task.

C. RESEARCH QUESTION

This thesis answers three primary questions to develop a wide-area SAR synthesis technique:

1. Can deep layer training be achieved using coarse pseudo-resolution images?
2. Can datasets be expanded using background substitution?
3. Is the developed augmentation technique suitable for synthesizing wide-area SAR images?

The first two questions apply directly to the mechanisms of proposed two-stage data augmentation technique—reduced resolution and content manipulation. This

innovative technique combines two threads of research in a manner than enhances the effectiveness of each. There are justifiable concerns that the two-stage technique might create artifacts that degrade the training effectiveness, so the third question focuses on ensuring that no detrimental additions are made.

D. HYPOTHESIS

This thesis answers three questions and starts with a hypothesis for each. The first question regards "deep layer training"—a new term based on the hypothesized training mechanism of coarse resolution images. It has been shown that coarse resolution SAR images are still able to train AI networks to a limited degree [20], [21], [22]. This thesis theorizes that this training capacity exists in a CNN because the coarse images and fine images appear very similar after pooling layers. Therefore, the first hypothesis is that reducing the resolution using pooling-layer-like kernel methods enables those layers to be bypassed. Furthermore, the coarse data can be designed to bypass until a targeted layer by controlling the pixel dimensions of the coarse image. This deep layer training effect is shown by examining the training activity level in each layer and comparing the effects of both coarse and fine images.

The second question asks if datasets can be expanded by using background substitution. The background substitution mechanism developed in this thesis applies EO techniques to magnitude SAR imagery. The hypothesis is that new backgrounds increase the variance of the background area and prevent the CNN from training on background features. This forces the CNN to focus on object features, thus improving accuracy.

The third question asks if the developed technique is suitable for wide-area SAR synthesis, specifically regarding artifacts from augmentation. It is very difficult for a human observer to say what features and artifacts are significant to a CNN. Artifacts, if present, would likely become the primary training feature of the CNN. Therefore, the hypothesis is that detectible artifacts that would make the method unsuitable for wide-FOV synthesis would also cause decreased accuracy in narrow-FOV testing. If test accuracy improves or remains the same in this narrow-FOV testing, then any artifacts are not detrimentally impacting training. If artifacts are non-detrimental—because a human

observer cannot say definitively that they are absent—then the data augmentation technique is suitable.

E. SCOPE AND BENEFITS OF THIS RESEARCH

In order to apply state-of-the-art EO AI/ML object-detection techniques to SAR, larger, labeled, wide-FOV datasets are needed. This thesis develops a novel two-stage data augmentation technique that will enable the creation of such wide-FOV datasets. Due to the lack of labeled wide-FOV datasets, there is no way to determine the validity of this two-stage data augmentation in the wide-FOV context. Therefore, this thesis synthesizes single-object, narrow-FOV images and tests those against real-world SAR chips. Success with narrow-FOV images provides the necessary support to continue development with wide-FOV image synthesis.

Using the AConvNet classifier as an analog for the classifier in two-stage AI/ML object detector, and using the two-stage classifier as a pretense for use in a single-stage network, the results of this paper will support future development using wide-area images synthesized with the same augmentation techniques. A verified, data augmentation method capable of synthesizing wide-FOV SAR scenes will directly support the future implementation of state-of-the-art AI/ML object detection methods.

F. HOW THIS THESIS IS STRUCTURED

As stated previously, this thesis develops a novel two-stage data augmentation technique for SAR. Using this technique to synthesize narrow-FOV images and train a CNN supports continued development with wide-FOV image synthesis. Because the augmentation method is novel, this thesis is structured in a building-block approach for both development and testing, as described in the following paragraphs.

Before any of the technical work, Chapter II provides contextual information on EO, SAR, and AI/ML that supports the two-stage augmentation technique. The two data augmentation stages, which are interleaved in practice, involve decreasing the resolution and overlaying objects onto new background images. For its part, the augmented resolution

is a continuation of earlier reduced-resolution SAR augmentation work. Therefore, Chapter II is rounded out by a discussion of these previous low-resolution SAR works.

Chapter III provides the heavy lifting for this thesis's technical aspects. The first section describes how the SAMPLE dataset is prepared for use with AConvNet. The second section outlines development of this thesis's novel data augmentation technique. The third section defines and describes AI-specific charts and figures of merit that are used to measure findings. Last, the fourth section lays out the details of how each validating test is performed and what results are expected.

Chapter IV investigates the results of each test in a continuous manner. While results could have been presented after each test is described, combining the results in a single chapter helps contextualize the support that each test builds. By the end of the constructive series of tests, the new data augmentation technique is shown to provide a 10-percentage point accuracy improvement (17% improvement over initial performance) when tested against the measured SAMPLE subset.

Finally, Chapter V summarizes the findings and presents conclusions in a concise manner. In the end, the accuracy improvement against real narrow-FOV SAR images show that the proposed augmentation method is promising for wide-FOV SAR image synthesis.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

This chapter performs three foundational tasks for the novel two-stage data augmentation that is developed in Chapter III: building a base of EO and SAR characteristics, outlining the relevant aspects of AI/ML, and describing previous research. First, Section A contrasts relevant aspects of EO and SAR phenomenology. Entire books are written on these subjects, but only four aspects are relevant to the augmentation technique in this thesis. The second section, comprising both Section B and C, is AI/ML specific. Again, despite the diversity of the discipline, only the relevant computer-vision methods are explored. Last, because this paper's two-stage data augmentation method uses previous low-resolution research in SAR as well as techniques from EO, Section D reviews previous and supporting work in these areas.

A. DIFFERENCES BETWEEN EO AND SAR

This study relies on magnitude-only SAR imagery that has been processed into standardized grayscale images. After processing, SAR imagery can look very similar to panchromatic EO images, which leads to confusion. From this grayscale starting point it is easy to lose track of the fundamental differences between EO and SAR imagery. Several of these differences should be kept in mind when using processed SAR imagery for AI/ML training: illumination source, coherent collection duration, surface reflectivity, and image formation.

1. Illumination Source

First, the illumination sources of EO and SAR images are fundamentally different. Unless using a flash bulb in the dark, EO images experience ambient lighting. SAR images, on the other hand, are always illuminated by a point-source. Additionally, the backscattered radio frequency (RF) energy must be converted and orthorectified to form an image. Orthorectification, low depression angles, and a flat-ground assumption make SAR images appear to have been taken from directly overhead [1]. This gives SAR shadows special

significance as back-projections of the objects' profiles [20], which add to the learnable information for an AI/ML model.

Additionally, ideal SAR shadows are perfectly black (zero value)—there is no backscatter return because no RF energy reaches or reflects from behind an object. Any apparent shadow illumination in SAR is caused by aliasing, speckle, orthographic projections, or artifacts of image formation like streaking [1], while EO shadows can have objects dimly illuminated by ambient light. Also, some shadow boundary confusion arises from variations in EO surface *color*, which has no analog in SAR. Last, SAR shadows are always orthogonal to the flight vector [1]. EO shadows are generally cast away from sun and result in so much variation that shadow geometry is generally ignored.

These significant aspects of SAR shadows—back projection, ideal black shadows, and aspect angle—make most EO data augmentation techniques non-transferable. EO images lack the first two properties, so no EO data augmentation techniques exist to exploit them. But these aspects are unique and crucial to SAR, and some studies have successfully classified using only the shadow [20]. Furthermore, SAR's fixed aspect angle makes many geometric transformations like flips and rotations counterproductive. Additional shadow angles would add confusing variations that do not represent the operating conditions of SAR.

This study includes the full object return and shadow to ensure the CNN has access to any trainable data that may be present. AConvNet does not enforce shadow inclusion, but because this study uses a different dataset, no direct comparisons can be made between the AConvNet results and those of this study.

2. Coherent Collection Duration

The second major difference between EO and SAR imagery is the element of time. EO, being ambiently illuminated, cares mainly about exposure (shutter duration). Longer exposure means more photons on the sensor. SAR produces its own illumination and cares about fast and slow time—fast time referring to the speed of light traveling out-and-back in the range direction, slow time referring to the cross-range speed of the sensor [1]. Essentially radar ranging, fast time necessitates a radar "chirp" to resolve high-resolution

images. Went multiple pulses are compiled, this ranging effect creates a 3D return which, when projected into the slant plane, creates layover, backspreading, and overlay—holographic features that make SAR *very* unlike grayscale EO [1]. AI models trained on grayscale EO have very little transferability to SAR due to these fast-time differences.

Slow time, limited by platform velocity and duty cycle, creates the *synthetic* aperture which allows higher cross-range SAR resolution by integration. Slow time may seem similar to shutter duration but is vitally different. If an EO sensor is moving, longer shutter times make can make images blurry. A long-exposure EO sensor needs to be stationary to increase resolution. Conversely, the synthetic aperture of SAR requires sensor movement. Longer transits give wider apertures and higher potential resolutions. A stationary radar source cannot image a scene at all! Because the backscatter reflectivity of objects changes with aspect angle during transit, and because these aspect-dependent variations can manifest in phase data, the phase portion of SAR is potentially useful. In fact, several studies have demonstrated the utility of phase data [21].

Despite these findings and despite AConvNet being structured to optionally accept phase data, it is not used in this study for several reasons. Aside from the fact that phase data is not usually transmitted away from the sensor platform for processing because of bandwidth limitations, the primary reason for omitting phase in this study is uncertainty with the EO-based kernel operations and alpha blending techniques used in data augmentation. It is unknown and beyond the scope of this study to determine whether phase data can be treated in this way.

3. Surface Reflectivity

The third major difference between EO and SAR is surface reflectivity of the objects in a scene. Due to the differences in wavelength and illumination, RF scattering tends to be specular and EO scattering tends to be diffuse. EO images experience surface reflectivity in two scale-dependent ways—color and texture. Because of the wide incident bandwidth of visible light, different wavelengths might be absorbed or scattered differently, resulting in the appearance of color. Similar phenomenon on larger features create texture layered with the underlying color. These abundant EO features play a

significant role in AI/ML and, therefore, in EO data augmentation techniques. However, a clear line-of-sight and sun illumination is required for these passive EO sensors. The long wavelengths of the self-illuminating SAR sensor can penetrate clouds, making it a critical "day/night, all-weather" capability. That capability comes with complications for AI/ML.

The narrow frequency range of a chirp from a SAR emitter removes the possibility of color or texture in the EO sense but introduces new material and range detection capabilities. A SAR chirp is narrowband, and those few frequencies are combined to resolve range, preventing the possibility of color and the multiple channels that many AI/ML models depend on. Still, the apparent similarity to panchromatic imagery can cause analysts to make heuristic errors and apply AI/ML models incorrectly to SAR problems. Additionally, the large wavelengths of SAR radar chirps experience surface textures on a larger scale than the visible spectrum. Backscattering is very faint, so the strong returns of corner reflectors create glaring resonance. These faint, coherent backscatter returns also tend to randomly interfere with each other. The characteristic speckle created by this interference manifests as obscuring noise [1]—a feature not seen in standard EO imagery. Despite the lack of color or texture, and despite the bright points of glare and streaks of resonance, there are still potentially valuable distinctions for AI/ML. Without color or fine texture, the brightness and pattern of SAR returns can describe the material itself—a useful trait when searching for large man-made or metallic objects. Consistently bright returns can also become key points for scattering centers analysis. Additionally, as described previously, the elevation data inherent in the back-projected shadow regions adds particularly significant information to SAR images of large vehicles [20].

These fundamental differences allow many AI/ML opportunities but require novel data synthesis and augmentation techniques. Synthetic SAR objects cannot be created in standard 3D graphic rendering software. In SAMPLE, the researchers formed synthetic images using a high-resolution computer-aided drafting (CAD) model, ray-tracing techniques, and simulated material properties [10]. A "data dome" is used to collect the backscatter and synthesize a phase history out of the fast-time and slow-time frequency returns. This phase history (also known as spatial frequency domain return [1]) is then transformed into the image domain. Because the phase history is not provided in the public

dataset, there is no way to merge synthetic objects onto measured returns prior to image formation. Therefore, the augmentation methods in this thesis are restricted to modifying post-collection images, whether synthetic or measured.

4. Image Formation

The fourth difference between EO and SAR, which directly contributes to the data augmentation techniques of this thesis, is the method of image formation. EO images are formed by individual photo elements that correspond to pixels. EO resolution refers to the spacing of those elements on a focal plane. Alternatively, SAR images can be collected through a single-feed antenna. Nonliteral SAR images start as a frequency domain response collected in a phase history (also known as special frequency) on the slant plane [1]. This polar phase history is resampled onto a rectangular grid converted by two-dimensional (2D) Fast Fourier Transform (FFT) to the image domain [1]. The values of the rectangular grid are stored in a numerical array that is displayed as pixels. It makes sense that [22], [23], and [24] create low resolution SAR by Inverse FFT (IFFT) to frequency domain, then using only a fraction near the center frequency to FFT into a lower resolution image. Those authors reasoned that their method is aligned with the way SAR images are formed and is, therefore, a better low-resolution representation of SAR than a Gaussian denoising kernel operation. Their reasoning is sound, but the complexity may not be necessary. Furthermore, the original phase representation cannot be recovered perfectly, so error is introduced. While the single-stage denoising kernel they reference is not helpful [8], the idea of a kernel method is still useful.

Kernel methods like those in the convolution and pooling layers of the CNN assume that the pixels *are* the image. Therefore, when attempting data augmentation for use in a CNN (and for deep layer training), it makes sense to use a kernel method for downsampling to a lower resolution. A denoising Gaussian blur kernel achieves only two of the goals of the downsampling routine. It reduces information density and keeps the object of interest at the original scale. However, as the authors of [21] point out, blurring kernels are a very poor method of synthesizing a low-resolution SAR image. To improve the kernel method, this study uses a two-stage kernel operation instead of single-stage blurring. The first kernel

downsamples to simulate a low-resolution SAR image with a larger pixel spacing. The second kernel uses nearest neighbor interpolation to return the image back to original pixel spacing. This two-stage process realistically removes information from the image, returns the objects of interest to the appropriate scale, and does both without adding artificial smoothing.

B. DATASETS: MSTAR AND SAMPLE

The two most common SAR datasets are MSTAR [2] and SAMPLE [3]. Collected with airborne X-band radar in the late 1990s, MSTAR is a broad dataset of objects (vehicles and targets) and clutter at 0.3m resolution. Together, MSTAR and SAMPLE are the mainstay of development for automated SAR detection techniques. Being measured SAR images, the MSTAR contain real—not ideal—backscattering. Furthermore, the background is composed of real grassy fields with rocks. Because of the unique backgrounds, AI/ML models can classify MSTAR objects based on background association [11], [25], which is a problem. If an AI/ML learns to classify based on features other than the object, then the model will perform poorly in real-world operating conditions when those features are not present. If the associable background features could be removed, the training effectiveness of the dataset would be enhanced. The alpha-blending technique developed in this thesis can do just that. To create wide-area scenes, the paradigm is reversed. Instead of changing backgrounds, the objects can be overlayed into larger scenes.

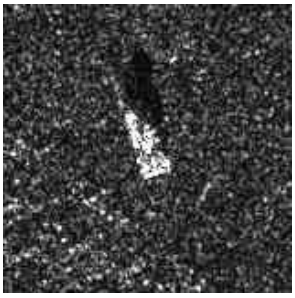
Two overlapping subsets of the MSTAR collection apply to this thesis in different ways. First, indirectly, ten classes of MSTAR vehicle chips—small images scoped to a single object of interest—are used in AConvNet for all training and testing. A different set of ten MSTAR-derived images are used to create SAMPLE, the other most common dataset for SAR. Half of those object classes overlap with the classes in SAMPLE, as seen in Table 1.

Table 1. Object classes in AConvNet and SAMPLE

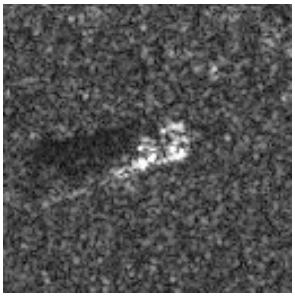
AConvNet (**MSTAR**)	2S1	BMP2	BRDM2	BTR60	BTR70	D7						T72	T62	ZIL131	ZSU234
This Study (**SAMPLE**)	2S1	BMP2			BTR70		M1	M2	~~M35*~~	~~M548*~~	M60	T72			ZSU234

* M35 and M548 classes removed from this study

After geometric translation and rotation, those measured images become part of the "real" SAMPLE subset. Second, the MSTAR Clutter dataset is used as a source of background texture for the alpha-blending operation. An example of an MSTAR chip, its measured SAMPLE corollary, and MSTAR Clutter can be seen in Figure 1.



(a) MSTAR chip of 2S1 used in AConvNet
Azimuth = 343° (from 6 o'clock, clockwise)



(b) Measured SAMPLE chip of same 2S1
Azimuth = 17° (from 0°, counterclockwise)

(c) Wide-area MSTAR Clutter

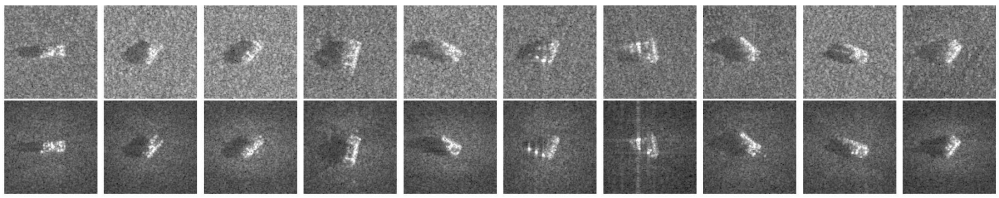
Figure 1. Examples of measured SAR images from MSTAR and SAMPLE. Adapted from [2], [3], and [26], respectively.

SAMPLE, the dataset used in this thesis, is composed of two subsets. The "real" subset of SAMPLE is created directly from the measured MSTAR images by

translationally centering the objects and then rotating them so that the azimuth of illumination is from zero degrees—the right-hand edge of the page [10]. The second subset of "well-matched synthetic" images are then created using high-fidelity 3D CAD models, ray tracing, and data dome collection [10]. Examples of the "truthed CAD models" can be seen in Figure 2(a). A side-by-side comparison of matched synthetic and measured chips can be seen in Figure 2(b). SAMPLE is "ideal for investigating the differences between measured and synthetic SAR imagery" [10].

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

Top row: "A comparison between EO images taken of an M1 during the MSTAR data collect (a, c) and truthed CAD models of the same vehicle (b, d) from two viewpoints. Note that even small details, such as the position of the gun, hatches, and the cable on the side of the vehicle agree well between the two images. Coloration of the CAD models is provided to identify part assemblies and does not denote any specific property that affects the electromagnetic simulation" [10].



Bottom row: "An example of one image of each vehicle in the SAMPLE dataset. Measured MSTAR images are on the top row, and the corresponding synthetic images are on the bottom row….We see that details such as shadows, orientation, and relative return magnitudes are in good agreement" [10].

Figure 2. Example measured and paired synthetic SAR chips. Source: [10]

It is worth noting that SAMPLE contains SAR imagery collected at depression angles of both 15° and 17°. The difference affects the incidence angle with surface features of the object and, thus, the appearance of the object in the formed image. Additionally, the downrange shadow region changes slightly based on geometry. Because MSTAR can be broken down by depression angle in the same way, some studies chose to train on one angle and test on the other. This is mostly a legacy research technique based on the small size of MSTAR. In operations, the depression angle could not be controlled so explicitly. Therefore, to increase the robustness of the model for an operational use case—and to add

data variance—both depression angles are be mixed in this study. There is no control over the proportion of depression angles assigned to either testing or training, which may warrant further study.

Finally, AConvNet and SAMPLE both use a subset of MSTAR objects, but the two sets do not perfectly align. A comparison of the available and used object classes can be found in Table 1. Even though the object classes are not identical between the two studies, the sizes and shapes are largely similar. This study uses the same objects from both the measured and synthetic SAMPLE dataset for comparison. The "real" measured SAMPLE images, being MSTAR images rotated and centered, are used for testing the effectiveness of experimental models trained with synthetic SAMPLE.

C. ARTIFICIAL INTELLIGENCE APPLICATIONS FOR SAR

This thesis exploits AI/ML computer vision techniques for ATR use with non-literal SAR imagery. There are many burgeoning applications for AI/ML; computer vision is the beneficiary of much of that research. Whether two-stage or single-stage, AI/ML computer vision object detection models generally have two broad tasks—detection and classification. These tasks are very similar to the "detect, discriminate, and classify" tasks of SAR Automatic Target Recognition (ATR) systems [27]. The evolution and intersection of these methods are well-documented [4], [9], so this thesis only reviews relevant highlights.

Again, while the ultimate goal of this two-stage data augmentation method is the creation of synthetic wide-area SAR imagery, this thesis is only able to test the classification of SAMPLE chips. Classification is still an important aspect of both an object detection framework and of ATR. Any two-stage or single-stage object detector would eventually have to perform classification of the objects being overlayed here. Showing that synthetic images augmented in this way can improve classification of measured SAR images is an important step supporting their future use in AI/ML-enabled ATR.

1. Automatic Target Recognition and Deep Learning

Automatic Target Recognition has been sought since the beginning of radar in the 1940s [9]. Deep learning (DL) can also trace a lineage to the 1940s artificial neural networks (ANNs) [19]. The 1980s saw the introduction of backpropagation and subsequent statistical DL modeling which inspired model-based ATR and, eventually, the MSTAR project. Early ATR systems relied heavily on human-developed, template-based target representations. Similar methods are still used in some modern AI/ML implementations. The region proposal portion of the two-stage RCNN, SPPnet, and Fast RCNN object detection networks all depend on templates developed with human-dependent processes [19]. Emergent single-stage, full-stack-trainable object detectors like YOLO do not require handcrafted feature detectors. By leveraging backpropagation and regression training techniques instead of proposal and classification to achieve detection, YOLOv1 dramatically increased the speed while reducing background errors by half compared to Fast RCNN [15].

The exclusive use of statistical learning puts single-stage object detectors like YOLO firmly in the second wave of AI [9]. Their simultaneous detection and classification require a paradigm shift from earlier frameworks. ATR traditionally relies on three stages—detect, discriminate, and classify [27]—which are very similar to the two-stage AI/ML object detector's region proposal, feature computation, and classification steps [19]. Both ATR and two-stage AI/ML detection/proposal steps generally rely on sliding-window techniques to scan the entire image as a human analyst might. Techniques like CFAR are common in ATR discrimination, whereas a pairing of trained convolution and pooling layers predominate in feature computation of a two-stage detector's CNN. Indeed, convolution and pooling are the only techniques used in the AConvNet CNN, making it a candidate classifier for a two-stage AI/ML object detector. But single-stage object detectors like YOLO use a different approach. Instead of a sliding window, they divide whole scenes into a grid. Each grid proposes bounding boxes and attempts to classify the boxes. The box center, dimensions, and classification are all trainable by statistical regression. In this way, YOLO has a unique capability to find objects in the context of their

backgrounds [15]. By utilizing context, YOLO is on the threshold of the third wave of AI—contextual reasoning [9].

Single-stage object detectors designed for computer vision tasks could provide a leap in speed and accuracy for SAR ATR. The only thing missing is the dataset. Because the detection functions of architectures like YOLO use supervised training through regression, they require large image datasets with many labeled objects. EO datasets like Pascal VOC and ImageNet contain tens-of-thousands to tens-of-millions of images, tediously labeled by hand. No such public datasets exist for SAR. A data augmentation technique capable of overlaying labeled objects into a wide-area scenes is the ultimate goal of this thesis's methods, but that is beyond the scope of this research. Again, this thesis will only develop single-object chips in order to verify the augmentation method for future use.

2. Classification vs. Object Detection

Continuing with the ATR pattern recognition model—Detect, Discriminate, Classify—it is easy to assume that object detection should be the first milestone of such an ATR project. The framework of a two-stage AI/ML object detector is essentially the same as the ATR model: both classify last. In fact, the methods of detecting and discriminating are the key differences between the ATR and AI/ML models.

Alas, training an AI/ML model requires large datasets and there are no publicly available, labeled, wide-area SAR datasets to train *or test*. What is available are datasets of single-object chips like MSTAR and SAMPLE. Because chips are narrowly scoped to the object of interest, the objects are already isolated; there is no *object detection* step. The available SAMPLE chips are, therefore, ideally suited for testing a classifier.

Furthermore, because the data augmentation technique developed in this thesis is novel, there is no evidence to support its efficacy. It would be difficult, if not impossible, to verify the capability of an integrated object detector if the capability of the classifier is unknown. Thankfully, the classification task can be broken out and examined independently. The first step—the only accessible step—is to develop and test the data

augmentation technique with a classification network like AConvNet. The independent investigation of classification is what this thesis accomplishes.

3. Tip-and-Cue vs. Two-Stage Object Detector

This thesis supposes a tip-and-cue workflow would be used to efficiently search wide-areas with low-resolution SAR before high-resolution recapture. In any tip-and-cue workflow, an initial process identifies and promotes (tip) areas of interest for further investigation (cue). This is actually very similar to how a two-stage AI/ML object detector works. The region proposal stage tips the classifier which then looks at narrowly scoped chips. The tip-and-cue sequence proposed in this thesis is distinct from a two-stage classifier because it does not act on the *same data.* Though this is a sequence of actions and, likely, a sequence of AI/ML networks, the tip-and-cue sequence cannot be integrated into a fully trainable single-stage network because of the mid-stream SAR-recapture requirements. Therefore, any AI/ML object detector using low-resolution SAR would only compose the first step in the tip-and-cue workflow.

4. AConvNet, an All-Convolutional Neural Network for Classification

AConvNet was developed in 2015 as part of an ongoing effort to improve the SAR ATR classification capability of DL networks [18]. The AConvNet authors concluded that the overfitting of SAR images that was experienced in fully connected networks could be overcome with a sparsely connected convolutional network. To study this hypothesis, the authors made *all* trainable layers convolutional. This is where the name *AConvNet*, short for All-Convolutional Neural Network, comes from.

The AConvNet architecture consists of five hidden (deep) layers. As the name implies, all layers contain trainable convolutional filters. The first three layers are also treated with max pooling, as seen in Figure 3. Pooling layers provide non-linear downsampling, which simplifies training and prevents overfitting [19]. This downsampling mechanism is what the pseudo-resolution of this thesis's two-stage data augmentation technique attempts to exploit.

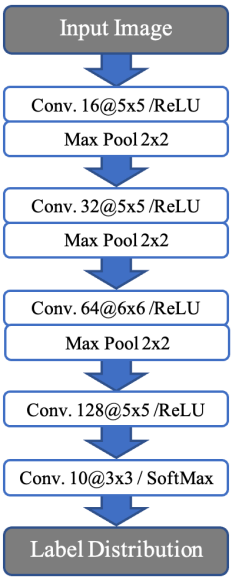


Figure 3. AConvNet architecture with five trainable layers. Adapted from [18].

AConvNet employs minibatch stochastic gradient descent to accelerate training toward a global loss minimum [19]. The ReLU activation function enables deep learning by preventing a vanishing gradient during backpropagation. Training and testing analytics are computed after each epoch, not after each backpropagation training cycle. An epoch concludes when every image has been sent through the network, but images are sent in minibatches of 100. Each minibatch is used to calculate and backpropagate a loss gradient to train the network. Other hyperparameters of AConvNet, such as momentum and learning rate, remain unchanged from the initial work [18].

5. Data Augmentation

Data augmentation is "a data-space solution to the problem of limited data" [8]. In general, data augmentation reduces overfitting by adding variance to the dataset and promoting generalization of the learned parameters (i.e., weights). A taxonomy of image data augmentation techniques is provided in [8] and expanded in Figure 4 to guide this discussion. There are many EO image data augmentation methods and not all are relevant to non-literal SAR imagery. Color space transformations, for instance, do not apply because there are no color channels in SAR. Likewise, geometric flips and rotations are

not applicable due to the self-illuminated collection mechanism in SAR. Many other data augmentation methods do apply, though.

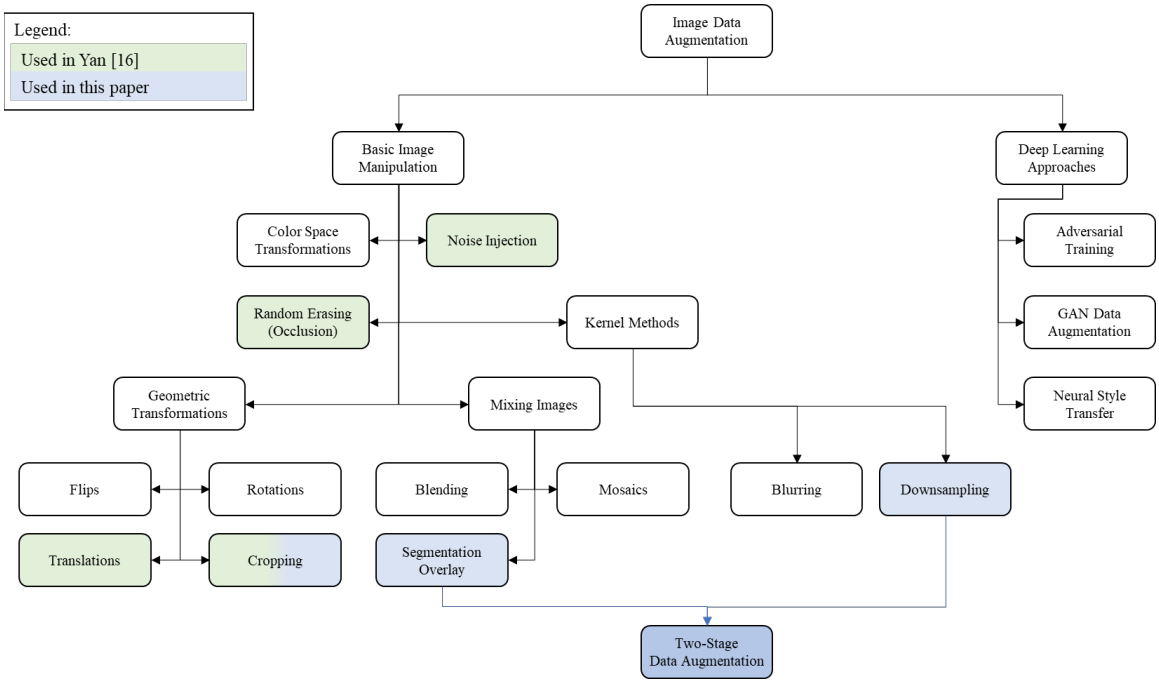


Figure 4. Taxonomy of data augmentation techniques for images. Adapted from [8].

For instance, [18] applies translation and cropping to the dataset used to train AConvNet. These geometric translations are unlikely to introduce errors or artifacts because the pixel values are unchanged. Translation do allow the expansion of the dataset, limited by the cropping size and pixel size of the images. Additionally, [22] defines several extended operating conditions (EOC) in which they explore noise addition, multiple resolutions, and partial occlusions. The noise they add is not Gaussian as [8] suggests, but multiplicative speckle that is more appropriate to SAR imagery. Similarly, resolutions are changed with a method SAR-appropriate method. Rather than creating low-resolutions images by downsampling with kernel filters in the image domain, [22] and [23] downsample in the frequency domain after IFFT and then reform the image with FFT. Finally, partial occlusions in [22] are not achieved through *random* erasing, but by a few

selective erasing patches. While erased regions in EO can be filled with Gaussian noise, the patches in [23] are filled with SAR-appropriate speckle.

The data augmentations in [22] are applied prior to training or testing, but data augmentation can also be applied at runtime. In YOLOv5, for instance, images are treated with flips, rotations, color transformations, blurring, blending, and mosaics at training runtime. This slows training due to extra processing demands but ensures that "images are never presented twice in the same way" [28]. This greatly increases the variance of non-critical features. Since not all these methods apply for SAR use, using YOLOv5 would require some modification. However, the idea of emphasizing critical features by adding variance to non-critical features is worth investigation. In fact, that is the premise of data augmentation in this study.

The underlying assumption in this study is that more *relevant* information about the objects can be learned if the variance of irrelevant background information can exceed the learning capacity of the network. This is especially applicable to an architecture that only performs classification because there are no bounding boxes or segmentations created. The accuracy of a synthetically trained model on measured data then becomes dependent on the quality of the synthetic objects.

6. Dataset Size

Even with data augmentation capable of expanding a dataset, there is no way to accurately predict how large a dataset needs to be to prevent overfitting. As [18] demonstrated with translational data augmentation, more is generally better. While AConvNet's ultimate dataset size is limited by cropping and available pixels, this study's background substitution allows potentially unlimited dataset expansion. This study examines several dataset sizes to observe the effects of dataset expansion, but those results only apply to this architecture and these data augmentation methods. Unless observing the effects of each augmentation in isolation, datasets in this study are created by combining augmented data with the original synthetic dataset. This would not be possible in every circumstance, but in this study the original images provide additional background variations.

7. Hyperparameters

In AI/ML the term "parameter" is used to describe the weights and biases of a CNN—the values that are *trained*. Once trained, parameters compose the model itself [29]. The term "hyperparameter," on the other hand, refers to high-level parameters that govern the model. Hyperparameters are set prior to training and most remain constant during operational use.

Additionally, the "minibatch" hyperparameter is significant in this thesis because if can be used to compare training of differently sized datasets. There is a necessary tradeoff when considering how many images to send through a CNN at once. Training on individual images creates too much variation between runs and gradient descent fails. But whole datasets are too large to be stored in active system memory for use all at once. Instead of using all-or-one, the minibatch allows a subset of images to train the model. Enough minibatches are run during each epoch to ensure that every image in the dataset has contributed to training.

8. Transfer Learning and Pretraining

Aside from data augmentation, there are also non-data-space methods to reduce overfitting including several functional solutions presented in [8]. Dropout, transfer learning, and pretraining are also used in this study to prevent overfitting. Dropout, which is built-in to the AConvNet architecture, zeros out random weights during training to prevent dependence on individual features. Transfer learning and pretraining both present different, alternative datasets to set the model parameters (i.e., weights). In transfer learning, an existing, trained architecture is used. The convolution weights and architecture are retained when switching datasets while other weights are zeroed out. Pretraining differs because it starts with a new architecture and trains it with an existing, large dataset. Then the dataset is switched for retraining. In the case of AConvNet, there are *only* convolution weights, so the nuance between transfer learning and pretraining is lost.

In this study, transfer learning refers to a model trained on synthetic SAMPLE, then retrained on augmented data. This is similar to how an object class would be added to a ready-trained model. Pretraining, in this study, refers to a model trained on augmented data

first, then retrained on synthetic SAMPLE. While this is not traditional pretraining because the initial dataset is not necessarily larger or better than the synthetic SAMPLE, it describes a potential future use case. An AI/ML object detector trained on a large, synthetic, wide-area SAR dataset created with this study's two-stage augmentation could be retrained later using a small dataset of measured wide-area SAR. By observing the training behavior of transfer learning, pretraining, and combined dataset use, this study can make better recommendations for future use cases.

D. FINDING INSPIRATION

This data augmentation technique proposed in this thesis is *novel,* but its component methods are not *new*. In fact, each aspect of the technique draws either directly or inspirationally from previous efforts. The first subsection identifies a thread of research exploring low-resolution SAR imagery as data augmentation. While the current thesis diverges in downsampling technique and dataset, it a continuation of a theme and draws heavily on previous architecture, decisions, and insights. The second subsection describes specific EO augmentation techniques that this thesis adopts for magnitude SAR imagery. The borrowed techniques are described while building justification for their cross-disciplinary application.

1. Varying Resolution

The low-resolution aspect of the proposed data augmentation is developed in response to a thread of research investigating low resolution SAR imagery [22] and multiresolution data augmentation [23], [24]. The thread starts with AConvNet, a five-layer CNN made of all convolution and max pooling layers [18]. This study also relies on AConvNet's straightforward PyTorch implementation [30] and well-documented previous results. Though other ML architectures have surpassed the performance of AConvNet on the MSTAR dataset as seen in [4] and [5], it is easier to examine training behavior when the architecture has fewer degrees of freedom. Since the data augmentation techniques presented herein are hypothesized to affect the behavior of convolutional and pooling layers specifically, it is appropriate to use a well-documented architecture made entirely of convolutional and pooling layers.

There is a key difference between the setup of this study and that of AConvNet, however. While AConvNet relied on the fully measured MSTAR dataset, this study uses SAMPLE [10]. Because this study aims to improve the training efficacy of synthesized SAR images, the synthetic subset of SAMPLE is used for augmentation, training, and validation. The measured subset is reserved for testing.

On the heels of AConvNet, [22] explores the behavior of AConvNet and numerous other AI models using MSTAR at lower resolutions. They show that limited training is still occurring with only low-level details. In addition to simply reducing MSTAR's resolution, [23] and [24] extended the idea by creating an augmented dataset with multiple resolutions. All three studies showed better accuracy in standard operating conditions (SOC) with AConvNet than in a state vector machine (SVM) or sparce representation-based classification (SRC) model.

While [22], [23], and [24] use a rigorous IFFT technique to create lower resolution images, this thesis assumes that high-level details can be removed with a kernel method. Additionally, this thesis assumes that the observed remaining effectiveness of low-level data also applies to the synthetic SAMPLE dataset. To show that multiresolution data can improve the accuracy of a CNN, this study explores the effect of augmenting SAMPLE with both specific resolutions and a combined multiresolution dataset.

2. Changing the Background

In addition to reducing the apparent resolution, this study has a second stage of data augmentation—changing the background. This enables two distinct use cases: deep layer training and synthetic wide-area image creation. The former is used in this study with the hopes that its verification will enable the latter. Because true wide-area SAR images are not collected at 0.3 m resolution, the hope is that an object detector trained with multiple synthetic resolutions will be able to function at *or better* than lowest training resolution. Of course, that study will have to be saved for future work.

Most people today would understand what it means to "photoshop someone into a picture." While Adobe Photoshop is a product with a wide range of capabilities, when someone uses the name as a deonym for adding objects to a scene, they are talking about

cropping and pasting. These are automation software versions of processes called segmentation and alpha blending. Usually, this process is envisioned as adding the object to another scene. Indeed, adding the object to the scene would be the case for synthesizing wide-area SAR images. But from the objects' frame of reference, their background is being changed. Adding a new background to an object is the case for this thesis where the goal is chip classification.

a. Segmentation Techniques

Segmentation is an automated process for determining the classification of each pixel in an image. To clarify, object and pixel classification are distinct ideas that share a title. Whereas the object classification performed in this thesis is a process of determining what type (class) *of object* is in a scene, pixel classification determines what type (class) *of area* each pixel is part of within the larger scene. For instance, segmentation might classify pixels as either object, shadow, or background without attempting to classify the type of object or composition of the background. When only single objects are desired, segmentation is usually depicted as a border drawn around the object of interest. When the encircled area is viewed in isolation as a binary image, it is referred to as a *segmentation mask*. This border can then be used to "crop" the object out of the image during the alpha blending step, described later. There have been many techniques to automate the segmentation process with varying degrees of success. Many types of regions—such as shadows—have indistinct borders, complicating matters. Furthermore, the problem becomes more difficult when an image only has a single, grayscale channel. Whereas colored red-green-blue (RGB) three-channel images have layered pixel intensities to interpret, grayscale images are flat.

There have been many robust efforts to create high precision segmentations of both grayscale and SAR images using either explicit algorithms or AI/ML. Some are based on 3D CAD models and show real promise for creating synthetic SAR overlays [31]. Many others rely on histogram thresholding [32], [33], [34]. Otsu thresholding [35], a well-established method that has been integrated into software packages such as OpenCV, can be used to find suitable thresholds in each image of a varied dataset. For SAMPLE,

however, the levels are consistent throughout the dataset and a manually set threshold is adequate. There is no requirement for dynamic threshold recalculation for each image, nor is there a need for the higher precision segmentation offered by more rigorous methods. Higher precision masks would be degraded during downsampling, making high precision segmentation a poor investment for this study. In future work, a high precision segmentation mask might enable the two-stage data augmentation technique to be applied at higher resolutions.

b. Alpha Blending

The second portion of the automated "crop and paste" idea, after segmentation, is accomplished with a process called alpha blending. The term *alpha* comes from the computer graphics practice of adding a transparency channel to RGB images [36]. The new, four-channel, red-green-blue-alpha (RGBA) images can be dynamically overlayed in RGB environments, such as a computer monitor, by using the transparency layer to blend the pixels with the underlaying graphics. The alpha layer, when viewed as a binary image in isolation, is called an *alpha mask*. In this study, the alpha mask is derived from the segmentation mask described in the previous section.

In grayscale images the alpha blending process becomes slightly more complex. With only a single channel, the alpha mask cannot remain a separate and distinct feature of the digital image; the images must be irreversibly integrated. In this study, the alpha blending processing is fully integrated into the augmentation workflow.

E. SUMMARY OF LITERATURE

This chapter has provided a basis of understanding for the remaining investigation. First, significant differences between EO and SAR imagery were detailed. These differences explain the difficulties in SAR interpretation and why existing AI/ML data augmentation technique cannot be directly transferred to SAR use. Second, the MSTAR and SAMPLE datasets were introduced. MSTAR is significant because it was the original dataset used with the AConvNet architecture. Though SAMPLE is an MSTAR derivative, the two datasets have a few significant differences. Third, many aspects of AI/ML are

developed for the reader. The discussion started as a comparison of object detection and ATR in a historical context, but quickly narrows into details specific to this investigation. Finally, the Literature Review chapter was rounded out by a discussion of recent research that provided inspiration and context for the data augmentation technique developed in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

This chapter performs three key functions of this thesis: preparing the dataset and architecture, building the augmentation routine, and setting up the tests. First, in A. Preparations the SAMPLE dataset and MSTAR Clutter dataset are standardized and sorted. One hyperparameter of AConvNet CNN, which is otherwise unchanged, is changed to accept magnitude-only data. Second, in B. Data Augmentation the novel data augmentation routine is constructed. The processes of segmentation and alpha blending are examined in depth. Third, in C. Charts and Figures of Merit precisely those things are laid out. The AI/ML field has specific ways of measuring and comparing different architectures and those may not be common knowledge to readers from outside the field. This section also introduces a novel means of investigating training activity in each layer of a CNN—a unique tool designed for this thesis that has great utility.

Last, in D. Validating the Workflow and E. Implementation the actual tests performed in this thesis are constructed. The reasoning behind the tests, their sequence, and the presentations of results are spelled out. Separating setups from results in this way might lead to some consternation, but this is done with careful reason. Test setup constitutes the *methodology* of testing; test *results* build a sequential body of evidence to support the final conclusions. Presenting the results sequentially—without breaking for setup and methodology—keeps results close at hand.

A. PREPARATIONS

Two major changes are made to the decibel representations of the public SAMPLE dataset prior to data augmentation: removing two classes of objects and cropping to 88x88 pixels. The dataset is then divided and hyperparameters are set.

A Log10 transformation is applied by the SAMPLE team during creation of the *decibel* dataset, and the same procedure is applied to the MSTAR Clutter dataset for use as backgrounds.

1. Removing M35 and M548

The first preparation made to the SAMPLE dataset is the removal of two object classes: M35 and M548. Very strong returns in those classes create glares and resonance in the formed images as seen in Figure 5(a). These glares cause the simple threshold segmentation routine to misidentify the shadow region as seen in Figure 5(c). Configuring the data augmentation workflow was a recursive process, so this complication was discovered after segmentation of a small subset of SAMPLE. After this discovery, the M35 and M548 classes are removed prior to augmenting the entire dataset. After removing these two classes, the synthetic and measured subsets of SAMPLE each contain 1088 images.

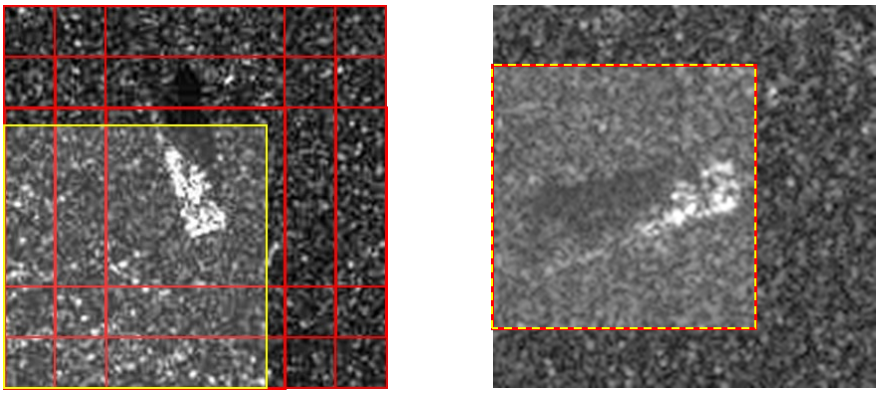
(a) Original SAMPLE (b) Object Alpha Mask (c) Shadow Alpha Mask

Figure 5. Examples of glares and resonance in class M35

2. Cropping to 88x88 Pixels for AConvNet

The AConvNet architecture is designed to accept 88x88 pixel input images. Some cropping is necessary because the images in the MSTAR dataset are inconsistently shaped. Additionally, [18] uses a sliding window cropping mechanism to add translation data augmentation to the training dataset as shown in Figure 6(a). Unfortunately, their cropping does not enforce shadow inclusion, a factor that is significant in SAR imagery [20]. Furthermore, images in the prepared AConvNet datasets are cropped to 94x94 pixels then further reduced to 88x88 pixels at runtime using a random sampling technique. This enhances the translational data augmentation effect but does not necessarily ensure uniqueness.

To remove translational variation while ensuring results of this study are as comparable as possible to the original AConvNet, the 88x88 image size constraint is applied to the 128x128 SAMPLE dataset in a fixed position. Because this study explores a novel augmentation technique, translation would add uncontrolled variability and is not applied. Cropping in this workflow is consistently centered in the middle of the vertical axis and on left edge of the horizontal axis, as shown in Figure 6(b).



(a) AConvNet: Example of translational cropping and random sampling

(b) This study: Fixed cropping position includes shadow

Red shows prepared cropping area; yellow shows final 88x88 area

Figure 6. Cropping to 88x88 pixels using AConvNet geometric translation in MSTAR (a) and this thesis's fixed cropping in SAMPLE (b)

It is worth pointing out that cropping position is not *specified* as the center-left edge. This study's data augmentation workflow is actually more complex and generally applicable than simply selecting a fixed position in the image. Segmentation masks for the object and shadow are created and combined *before* cropping. Once combined, the spatial center of the masks is used to determine the center of the 88x88 cropping area. If that area exceeds the bounds of the 128x128 chip, then it is shifted to remain within the chip bounds. The consistent center-left placement of the crop area is coincidental because of the consistent object and shadow placement from SAMPLE's translations and rotations.

3. Expanding the Dataset

The fixed cropping position in this study prevents the expansion of the dataset in a manner consistent with AConvNet's geometric translation. The authors in [18] report that they expand their representation of each object to 2700 images (total of 27,000 images in the dataset); implying roughly five representations of each image. In the PyTorch implementation [30] there is no mechanism to restrict the datasets in this way. Either way, their method is ultimately limited by pixels in the image.

In this study, the background substitution capability of the two-stage data augmentation workflow could easily be used to expand the dataset to an unlimited number of images. With no way to judge the ideal dataset size, this would quickly become a resource constraint issue. Instead, this study uses limited-size datasets to explore the potential benefits of expanding the dataset by changing the background and resolution independently.

For this thesis, each pseudo-resolution of the 1,088 synthetic SAMPLE images is created five times. This results in five subsets of 1,088 images labeled with alphabetical suffixes *a* through *e.* The full set for each pseudo-resolution therefore contains 5,440 images.

4. Log10 Transformation

This study's chips come from the decimal representations of the SAMPLE dataset [3] which has already been pretreated using the logarithm and normalization techniques recommended by the MSTAR team [2]. The logarithmic transform corrects issues relating to low dynamic range in SAR scenes. Normalization simply converts image pixel values from 256 discreet gray levels to a 0-to-1 decimal scale. The same logarithm and normalization techniques are applied in this study to the MSTAR Clutter dataset used for clutter background patches. Using the same treatments ensures that clutter scenes used for background texture have similar histogram levels to the chips. In operational use, any AI/ML trained on such treated images would need to prepare new images in the same manner to ensure consistent results.

It is worth noting that "No image preprocessing algorithm is applied to the SAR images [in AConvNet]" [18]. Therefore, this is another deviation from the original AConvNet that prevents direct comparison of results. That study does not preprocess the magnitude imagery because it also uses the phase channel. This thesis, however, is using an imagery-specific alpha blending technique that is not directly applicable to the phase representation. Overlaying objects with their phase representation might be possible through signal blending in the phase history (also known as special frequency) domain [1], but is beyond the scope of this thesis's proof-of-concept. There are still good reasons to preprocess SAR collections into non-literal images. Primarily, this is the format most often seen by end-users. Unless the AI/ML is implemented on the collection platform or full phase history is transmitted to the processing center, the AI/ML only has access to magnitude-only images. Luckily, conversion from phase history to the image domain actually improves CNN results [9].

5. Training, Validation, and Test Sets

For this thesis, all data augmentation operations are performed on the synthetic subset of SAMPLE, divided 80/20 into training and validation sets. The split is applied at the object level to ensure an even representation of each object in both the training and validation sets.

The test set is composed of the measured SAMPLE subset. Because the dataset is so small, testing is performed against the entire 1088 image test set after each epoch to prevent variation from random sampling. Results of testing are for monitoring and evaluation purposes only and are never fed back into the model.

6. Hyperparameters

The hyperparameters from the original AConvNet [18] are maintained except for "channels" which is reduced from two to one in order to accommodate the magnitude-only datasets in this study. Additionally, in [18] and this thesis, minibatches of 100 images are used. This consistent minibatch size enables the training rate comparison between dissimilarly sized datasets.

B. DATA AUGMENTATION WORKFLOW

Recalling the image data augmentation taxonomy chart in Figure 4, it is easy to see that the techniques used in this thesis are not new to the field of AI/ML. What sets this study apart from other kernel methods or image mixing routines is their combined use. The downsampling kernel method is similar to max pooling layers. This similarity allows information to bypass the initial layers somewhat, creating the observed deep layer training behavior. Meanwhile, the segmentation and alpha blending augmentations change the background of the image and enable continued training without overfitting.

This thesis interleaves the two augmentation techniques in a continuous workflow. Performing segmentation at the highest resolution produces the smoothest border while performing alpha blending at the lowest resolution simplifies the border transition. Border transition requires gradation and could insert detectible artifacts into images, but at lower resolution the lack of high-level detail greatly reduces the risk of small artifact transfer. An overview of the workflow can be seen in Figure 7. Details of each operation are included later in this section.

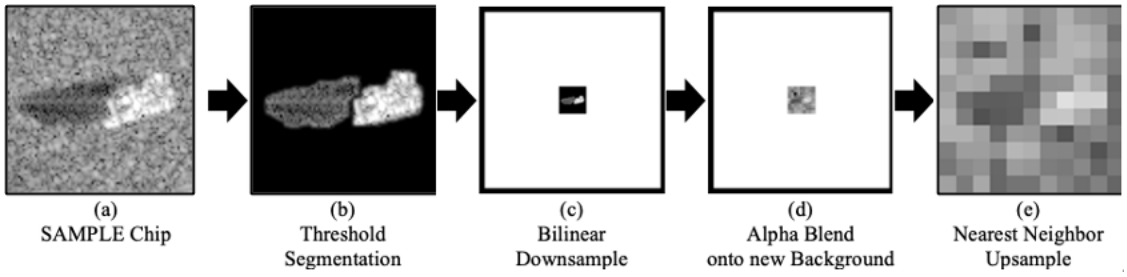


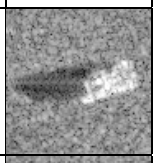
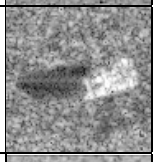
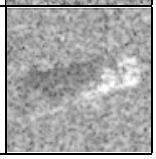
Figure 7. Two-stage data augmentation workflow showing interleaved kernel operations and background substitution

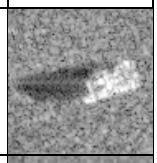
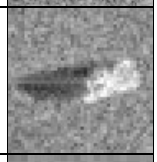
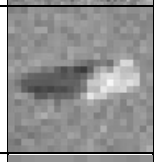
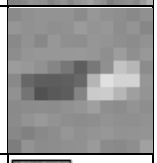
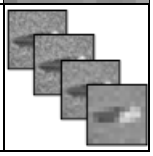
Varying the resolution and background produces several datasets for training and testing. Table 2 explains the naming convention applied to the generated datasets. Note that the resolution designations for datasets are not literal. Lower pseudo-resolutions are created by integer downsampling ratios. The baseline 0.3m resolution is derived from SAMPLE's range and cross range resolution values of 0.3047m, which is the ground sample distance (GSD). Datasets S03 and M03 represent the control set and test set,

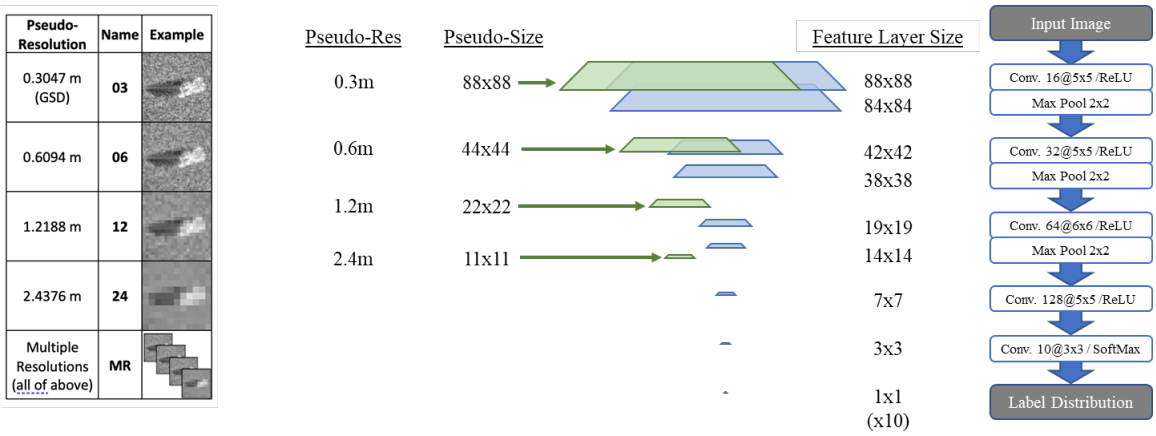
respectively. Both S03 and M03 datasets are prepared by cropping but are otherwise unmodified.

Together, the downsampling and background substitution allow bypassing of pooling layers in the CNN. The theoretical injection point of modified data is that position in the CNN where the feature layers are transitioning through the pseudo-size of the augmented images, as seen in Figure 8. For instance, the C12 dataset is downsampled to a pseudo-size of 22x22 pixels. Though it is resampled back to 88x88 pixels prior to training input, its lower pseudo-resolution has a reduced training effect as it bypasses the first two deep layers of the CNN. After this injection point, the training effect is essentially the same as unmodified S03 data.

Table 2. Dataset title nomenclature

Source or Background	Name	Example
Synthetic (SAMPLE **[3]**)	**S**	
Noise	**N**	
Clutter (MSTAR Clutter **[26]**)	**C**	
Measured (MSTAR-derived in SAMPLE **[3]**)	**M**	

Pseudo- Resolution	Name	Example
0.3047 m (GSD)	**03**	
0.6094 m	**06**	
1.2188 m	**12**	
2.4376 m	**24**	
Multiple Resolutions (all the above)	**MR**	



Augmented images bypass pooling layers until the pseudo=size matches the feature layer size

Figure 8. Mechanism of deep layer training. Architecture (right) adapted from [18].

1. Alpha Masks from Segmentation

The first step in this thesis's interleaved, two-stage data augmentation technique is the creation of alpha masks using threshold segmentation. An alpha mask is an image processing tool that specifies the transparency of certain image areas. In this thesis, threshold segmentation is used to create an alpha mask which defines the object area to keep; the original background will be transparent and discarded. To get the most accurate segmentation, the alpha mask is defined in fine-resolution chips. These masks are later applied in coarse resolution to reduce artifacts around at the line of transition from segmented object to background. The threshold segmentation method is discussed in this section. Explanation of the full alpha blending procedure is reserved until subsection 4. Alpha Blending, because the process requires a new background.

As previously discussed, there are many ways to segment a grayscale image. Luckily, SAMPLE has consistent brightness levels and background textures. Additionally, once the chips are downsampled, even high-precision segmentation masks become pixelated and degraded. For the limited purposes of this test, a manually specified, observation-based threshold is adequate to initialize the segmentation routine.

Segmentation borders are found by a series of dilation and erosion kernels. Objects are segmented using their histogram's mean value as a threshold. The shadow is segmented

by inverting the image and thresholding with 133% of the new mean value. Examples of each can be seen in Figure 9. This mean offset and the kernel size are derived empirically based on the busiest clutter scenes. In future work with non-SAMPLE data sources, a more robust method could provide more reliable segmentation and improve the data augmentation technique.

(a) SAMPLE chip (b) Object mask (c) Shadow mask (d) Combined mask

Figure 9. Examples of object mask (b), shadow mask (c), and combined mask (d) created using threshold segmentation

2. Kernel Downsampling

The second stage of the interleaved process is the application of the first of two kernel methods as seen in Figure 7(c). This first kernel method removes high-level information from the image with bilinear downsampling. Later, another kernel will upsample images back to their original scale.

Kernel techniques are the same mechanism used in convolution and pooling layers and are, therefore, generally not used for data augmentation [8]. However, this study does not use a blurring kernel as assumed in [8] and [23]. Instead, a two-stage kernel operation simulates a SAR image originally formed at a lower GSD being upsampled to bring objects of interest to the same scale at which the CNN's feature layers are trained.

In [22], [23], and [24] the authors use an IFFT to revert magnitude images back to the frequency domain before resampling at coarser resolution. Their technique is consistent with non-literal SAR image formation where frequency domain data is converted by FFT to the image domain, cast onto a numerical array, and then displayed as pixels. However, the convolution and pooling layers in a CNN use simple kernel techniques. Therefore, this study uses a similar kernel mechanism to bypass these convolution and pooling layers.

Since rigorous simulation of low-resolution SAR imagery is not strictly necessary to remove high-level information, a kernel technique is sufficient.

Execution of the two-stage kernel technique begins by removing high-level information through bilinear downsampling [37], [38]. The kernel operation removes high-level detail by averaging small bright/dark spots into their pixel neighborhoods. The simulated coarse resolution is the *pseudo-resolution*. The minimum size of the image after downsampling to the pseudo-resolution is the *pseudo-size* of the image; see Figure 7(c), Bilinear Downsample.

To be able to return the images to their original dimensions, the kernel size for downsampling and upsampling are necessarily the same. This kernel size is determined by the ratio of image pixel dimensions. Importantly, the upsampling ratio *must* be an integer. Without an integer, fractional upsampling would translate objects toward the upper-left corner of the image. Therefore, the available downsampling ratios are determined by the image pseudo-resolutions, which are driven by the pseudo-size and, fundamentally, by the factors of 88. If different pseudo-resolutions are desired, modifications would have to be made to the augmentation routine and/or CNN input size.

Hypothetically, because the kernel process used in both downsampling and upsampling is so similar to the convolution and pooling layers of the CNN, the augmented images should pass through several initial layers largely unchanged, as depicted back in Figure 8. Inserting augmented training data into deep layers through the normal training pathways prevents the need for handcrafted feature layers. This allows augmented images to disperse throughout the CNN uniformly and allows full-stack backpropagation training.

The main difference between the kernel process and the CNN's convolution and pooling layers is the CNN's trainable weights. In the data augmentation process, the bilinear downsampling operation uses linear weights based on the size of the kernel. In the CNN, the convolution weights vary based on backpropagation. The max pooling kernels remain fixed and nonlinear. If the pooling layers were instead using bilinear downsampling, and the convolution layers were all ones (not filtering), then the augmented images would be recreated when the layer size matched the pseudo-size of the image.

Based on this, the dataset's training effect should insert at the point where the layer size equates to dataset's pseudo-size, as seen earlier in Figure 8.

3. Background Texture

Prior to the background substitution step, a background must be generated. Two background textures are explored in this thesis: noise with speckle and clutter. If noise is as effective as clutter, then AI/ML object detectors could be easily pretrained with randomly scattered objects on a noise background. The clutter background method, on the other hand, requires a set of wide-area SAR images that are parametrically similar to the object chips. Thankfully, MSTAR Clutter has the same parameters as the MSTAR chips and, subsequently, the SAMPLE chips. The creation and implementation of these backgrounds is described in the remainder of this subsection.

a. Noise

A randomly created noise background is simple to create and adds variation to the dataset. Upon inspection of histograms from dataset S03, the original background histogram distribution is more similar to a flipped Rayleigh distribution than either random or Gaussian as shown in Figure 10. Therefore, a Rayleigh distribution of noise is created, scaled, and flipped about the x-axis. To approximate the original background histogram, a mean value of 152 and variance of 788.88 is used. Noise backgrounds are created at the lower pseudo-size of their intended host chip to make noise representative at that scale and prevent downsample blurring. Before introducing the segmented object from the host chip using alpha blending, salt-and-pepper effect is added to simulate background. The overall effect is very similar to the original synthetic background upon either histogram or visual inspection, as seen in Figure 11.

b. Clutter

The second background texture is clutter. Clutter is derived from measured SAR data from the MSTAR Clutter dataset [26]. The clutter dataset contains natural textures of terrain and foliage without man-made objects, making it perfect for this application. Variation is ensured by cutting patches from random locations in random clutter images

(scenes) for each background, as seen in Figure 12. Linear histogram matching is performed to ensure background and shadow intensities from the clutter scene are scaled to match those of the original SAMPLE chip. As seen in Figure 13, histogram matching is achieved by finding the mean value of the shadow and background regions in both the chip and clutter scene. Individual pixel values in the clutter patch are then linearly modified so that their shadow and background mean values align with those of the chip. After histogram matching, patches are bilinearly downsampled in the same manner as their paired chip object prior to alpha blending. This downsampling ensures that background textures are representative at the object's scale.

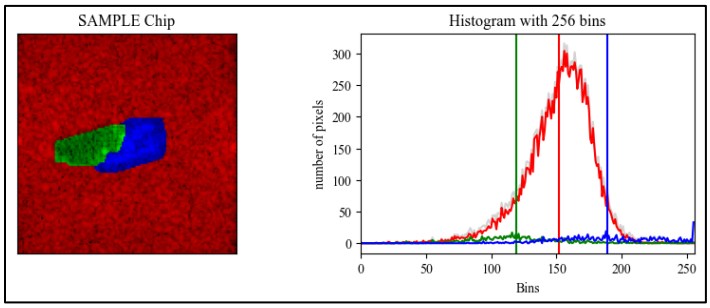


Figure 10. SAMPLE chip background histogram by segmentation area with mean values as vertical lines

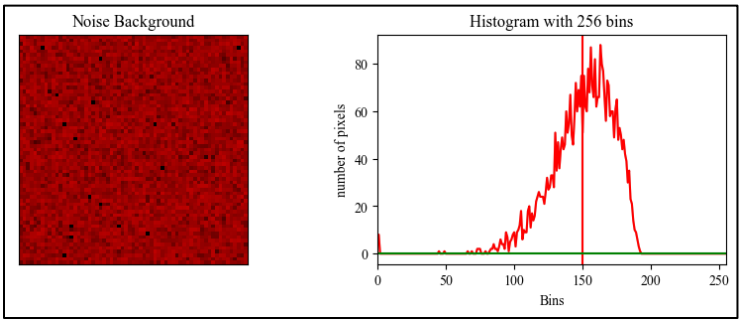


Figure 11. Flipped Rayleigh noise distribution with salt-and-pepper added on a 94x94 pixel background

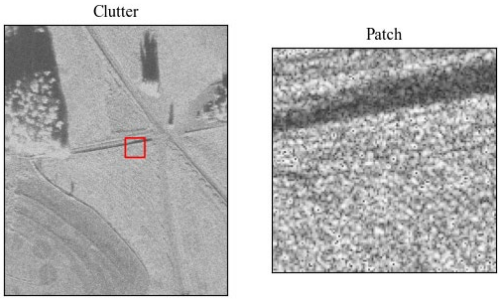
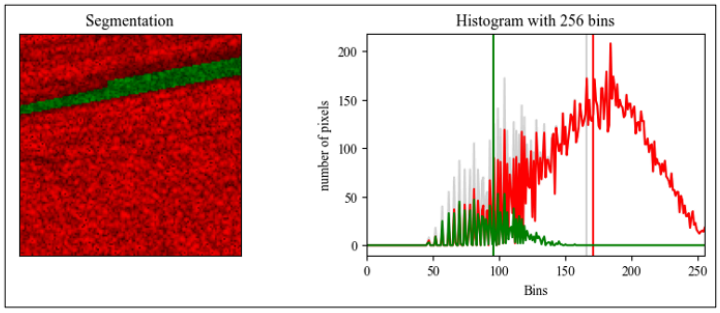
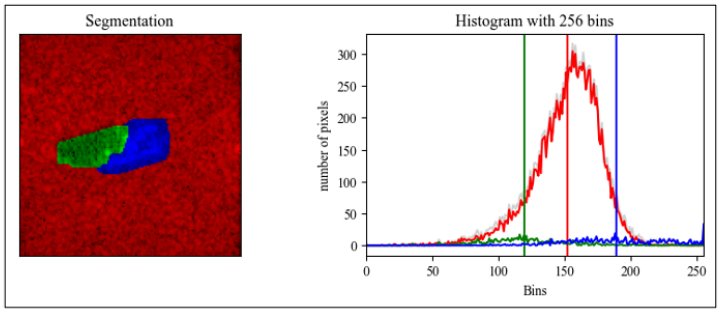


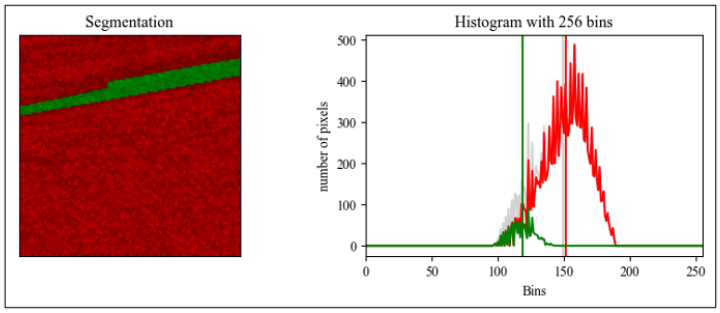
Figure 12. Cutting a patch from an MSTAR Clutter image



(a) MSTAR clutter patch



(b) SAMPLE chip



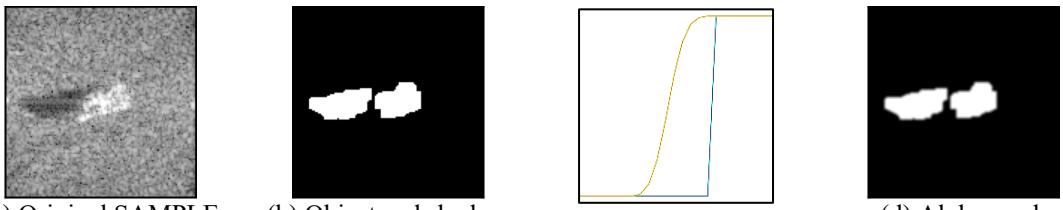
(c) Linear scaling of clutter to match mean of shadow and background in chip

Figure 13. Example of linear histogram matching of clutter patch to SAMPLE chip

4. Alpha Blending

The third stage in the interleaved data augmentation process is the combination of the segmented object with a new background using alpha blending, as seen back in Figure 7(d). As previously described, an alpha mask is an image processing tool that specifies the transparency of certain image areas. Alpha *blending* is the application of the alpha mask to create a new image. In this study, the areas of non-transparency (the areas to keep) in the alpha mask are derived from the enclosed areas created during object and shadow segmentation. The remaining background area of the image is marked as transparent.

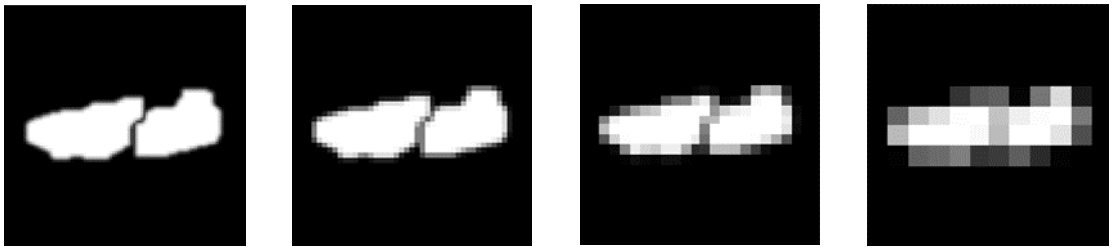
Such a sudden transition at the border of the alpha mask is likely to create detectible high-level artifacts. Because of the higher apparent resolution datasets like C03 and C05, it is important to blend the transition area of the border. This study uses a combination of dilation and sigmoid transition to prevent artifacts, as depicted in Figure 14. A dilation kernel uniformly increases the size of the segmented area, then the sigmoid transition adds a nonlinear gradient for blending.



(a) Original SAMPLE chip | (b) Object and shadow masks | (c) Detail of sigmoid transition of object | (d) Alpha mask with sigmoid transition

Figure 14. Alpha mask created from segmentation and sigmoid transition

In the alpha mask, the sigmoid is entirely outside of the object area and is fully opaque by the object to ensure that details are not lost, as seen in Figure 14(c). Conversely, the sigmoid transition begins at the outer edge of the shadow area to prevent inclusion of any background features outside of the shadow. This also prevents the creation of a bright ring around the shadow. Regardless, most gradation is lost in the 1.2m and 2.4m datasets, as shown in Figure 15.



(a) 0.3m (b) 0.6m (c) 1.2m (d) 2.4m

Figure 15. Example of alpha masks at different pseudo-resolutions after cropping to 88x88 pixels

For single-channel grayscale images, the alpha mask cannot simply be added as a fourth channel (such as RGBA). Instead, the grayscale image must be irreversibly changed. This is a multi-step process involving the alpha mask, the object to be overlayed, and the background image. The binary alpha mask is multiplied into the object, substituting black (zero value) for transparency. An inverse of the alpha mask, called a *beta mask*, is multiplied into the background image [39]. This creates a region of black in the background that will contain the object when the two modified images are added together. The sum of the alpha-treated chip and the beta-treated background create the augmented chip. Example images from this process are displayed in Figure 16.

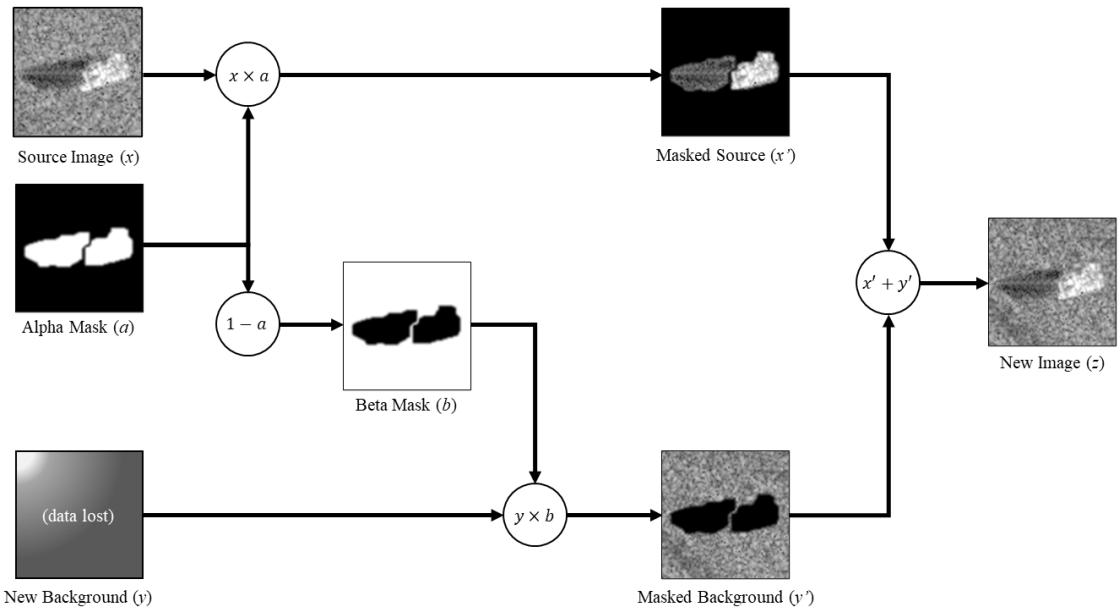


Figure 16. Example of grayscale alpha blending steps

5. Kernel Upsampling

The fourth and final stage of the novel, interleaved data augmentation process is the return of synthetic, coarse images to their objects' original scale using nearest neighbor upsampling. Resampling behavior is common practice for enforcing input image size in both classifiers and two-stage object detectors. Frequently this resampling goes unacknowledged, which could have significant impacts if it creates artificial variation in the convolution layers. This variation could be detrimental to CNN performance if it is not representative of operational use cases. In this study, this upsampling is acknowledged and controlled. More significantly, the upsampling in this thesis returns objects to their original scale, which reduces the feature size variation that convolution filters must learn.

In this second kernel method, the smaller image with a new background is upsampled using nearest neighbor interpolation [37]. The blocky appearance of the resultant image is caused by the direct representation of the pixels in the smaller image without any interpolation assumptions. A detail of nearest neighbor can be found in Figure 17 Significantly, no new information is added with the nearest neighbor method.

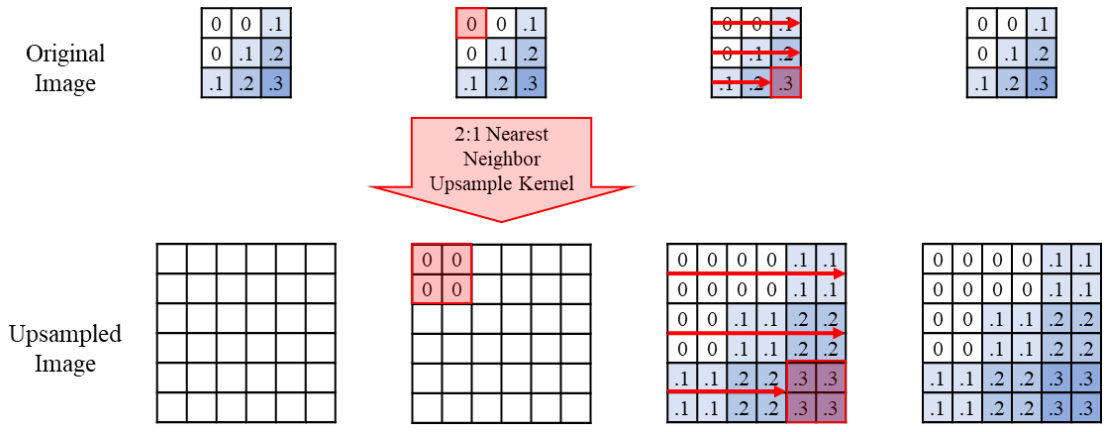


Figure 17. Depiction of 2D nearest neighbor upsample kernel method

By upsampling the newly created images after alpha blending, the interleaved augmentation process has created a synthetic image with fewer fine-resolution object and

background features. Having objects at the same scale in the original and augmented images ensures that the convolution kernel training is consistent if datasets are combined.

C. CHARTS AND FIGURES OF MERIT

The AI/ML field uses terms, charts, and figures of merit that may be unfamiliar to readers from other fields. This section first draws an important distinction between validation and testing. Second, the key figures of merit—accuracy and loss—and their accompanying graphical depictions are explained. Third, this thesis's novel method of observing training activity is developed. Last, this section introduces the minibatch hyperparameter—*hyper* because only the weights and biases are considered *parameters*—allows the observation of training rates.

1. Validation versus Testing

How can an observer tell whether CNN training is valid or the model is overfitting the training data? In essence, a CNN is performing an optimization routine; therefore, it is expected to improve continually until it converges at some optimum solution. But the model is optimizing to a specific, limited set of data—the training dataset. Overfitting occurs when the model begins failing to classify images outside the training dataset.

As alluded to previously in III.A. Preparations, this thesis splits the SAMPLE dataset into three subsets: training, validation, and testing. As the name implies, the *training* dataset is used to train the model. The validation and testing datasets are reserved to measure how effective this training will be on data that is *not* in the training dataset [40]. Sometimes the terms validation and testing are used interchangeably, which leads to a great deal of confusion. In general, the *validation* dataset is a portion of the training dataset that is not used for training, whereas the *test* dataset differs depending on the scenario. Sometimes, as during AI/ML competitions, the test dataset has the same characteristics as the training and validation datasets, but is simply unknown to contestants prior to competition. In such a case, test results would measure performance of the network. Other times, as in this thesis, the test dataset images have different characteristics. In these cases, the test results show compatibility of the model with a wider range of data.

Another key distinction between validation and testing is that validation is concurrent with training while testing is usually done once training is complete. It is necessary to calculate validation loss concurrently because it is the primary indication of overfitting. Test results, on the other hand, are primarily used to show performance or compatibility. Sometimes, as with MSTAR in AConvNet [18], there is too little data to create a true test dataset. In these cases the term *test* is sometimes used to describe the reserved dataset that is evaluated concurrently with training and used to determine overfitting and overall performance. This is more properly validation, but calling it *testing* was standard at the time of AConvNet [18].

In this thesis the measured subset of SAMPLE provides a true test dataset. The measured and synthetic SAR images have characteristic differences; therefore, measures of compatibility are desired. Moreover, because the eventual desire is to create wide-area synthetic SAR imagery to train an object detector for use on measured SAR images, compatibility with the measured subset of SAMPLE is paramount. Therefore, testing is *also* accomplished concurrently with training in this thesis.

With both validation and test information available after each epoch, the only overfitting that truly matters is when the test loss diverges. The divergence of test loss (overfitting measured data) is always earlier than validation loss divergence because of the different image characteristics. For this simple reason, only the training and test lines are depicted on charts in this thesis.

2. Accuracy and Loss

The primary visual tools used to analyze training effectiveness are the accuracy and loss charts. Each chart tracks the respective Figure of merit over all the epochs of training. Accuracy is the percentage of correct classifications and is abbreviated as *PCC* in some references. Accuracy is calculated by dividing the sum of true positives and true negatives by the total samples, as shown in Figure 18. If a model is trainable with a given dataset, the accuracy increases during the beginning of training and, generally, converges toward a maximum accuracy prior to overfitting, as shown in Figure 19(1). At the time AConvNet

was developed, total accuracy was the primary Figure of merit used to compare classification networks.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total}$$

Figure 18. Accuracy calculation

Loss is significant because it is the primary way to detect overfitting. The term "loss" can be confusing because the word is also used to describe signal degradation. When talking about AI/ML training, "loss" refers only to the difference between the correct classification and that predicted by the model. AConvNet uses a cross-entropy loss function to backpropagate error and train the convolution layers [18]. If a model is trainable, the value of the loss function decreases during the beginning of training and converge toward limit around the same epoch as accuracy, as shown in Figure 19(2). If overfitting is occurring the validation or test loss function diverges and grow larger again as seen in Figure 19(3).

Three different portions of the dataset—training, validation, and testing—are each used to calculate accuracy and loss during each epoch, but only training and testing are displayed. Accuracy and loss from the training dataset are displayed on the charts. Training loss is used with stochastic gradient decent for backpropagation training. Finally, results from the measured test dataset, M03, determine the final performance figures of each model.

Because test accuracy and loss can vary above and below a mean, inflection points are difficult to determine. To elucidate the inflection points, this thesis takes a mean of test loss values. The mean of the past three, future three, and present Figure are calculated; the running average is displayed as a trendline.

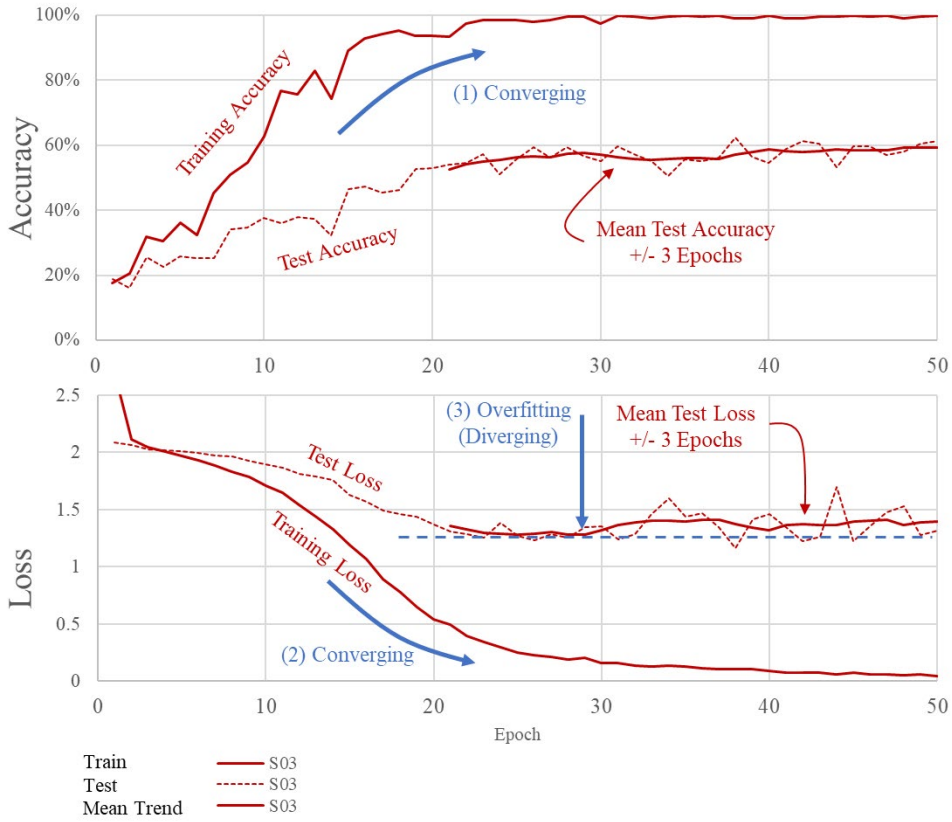


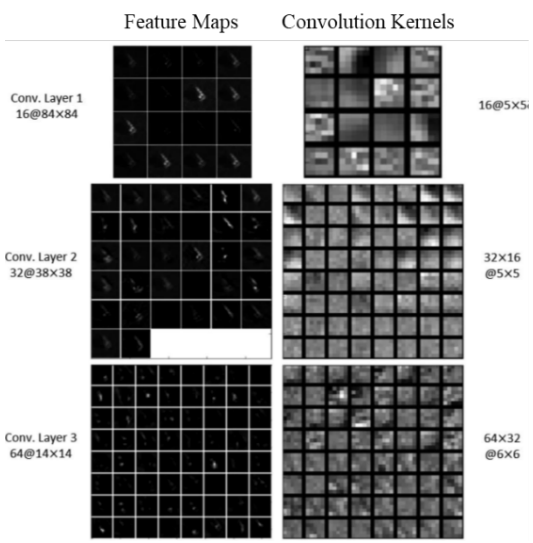
Figure 19. Representative accuracy and loss charts, labeled for clarity

When modifying datasets and training different models, two significant results come from comparing different runs. First is the average final accuracy. While it is standard practice to choose a "best fit" model at some point prior to overfitting, this is misrepresentation of expected CNN performance. In operational use, there is no way to judge if a model is operating at this elevated point. The peak is merely a momentary deviation that favors the dataset. Instead, by tracking the accuracy over each epoch, the average accuracy can be calculated for a set of runs prior to overfitting. In this study, the last ten epochs before overfitting are averaged and the mean is presented as the representative accuracy of the model.

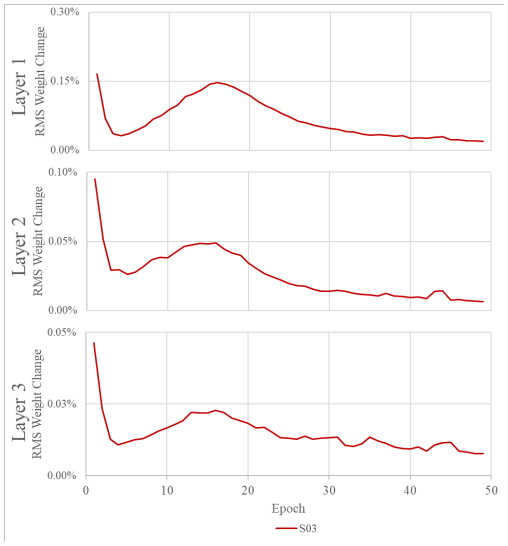
The second result can be found on the loss chart. When the test loss passes an inflection point and begins to climb, the model is overfitting. This means that the model is becoming too narrowly specialized on the training dataset and is losing the ability to classify the test dataset.

3. Layer Look Chart

To determine if deep layer training is improved after removing high-level detail, this study develops a new technique of investigating epoch-to-epoch training in each layer of the CNN. Referred to herein as a Layer Look Chart (LLC), this technique can be thought to extend the CNN's neuron metaphor to an electroencephalogram (EEG). Though it cannot tell you anything about the results of thought, an EEG measures and displays brain activity. Similarly, the LLC measures and displays training activity within the CNN layers. Without this new representation, prior studies have struggled to display or interpret training activity. The AConvNet paper [18] repeats a common practice of displaying grayscale representations of the filter layers at a given epoch, as seen in Figure 20(a). Unfortunately, much like a computerized tomography (CT) scan creating a cross section, representations of feature layers do not provide direct evidence of activity levels or training effect. They are only a cross section of one instance in time. Instead of investigating the layers as grayscale filters, the LLC looks at the change in the collection of weight values from one epoch to another using the root mean square (RMS) technique, as seen in Figure 20(b).



(a) Grayscale convolution kernels (filters) at the last epoch; similar to a CT scan



(b) Layer Look Chart showing training activity throughout epochs; similar to an EEG

Figure 20. Training activity investigation with grayscale filters (a) and the new Layer Look Chart (b). Grayscale source: [18].

To create a Layer Look Chart, the entire tensor of weights must be saved after each training epoch. Then the elemental difference between each epoch is calculated by subtraction. Because these changes can be positive or negative, RMS is then used to find average training activity level for the entire layer. The LLC displays these changes as a line graph to show the training activity over epochs. This gives important insight into the effects of transfer learning and data augmentation at each layer of the CNN.

There are several features that appear in the LLC, as annotated in Figure 21. Foremost, the line itself represents the training activity level. The actual RMS value of this line is largely irrelevant except when used for comparison within the same layer chart. This comparison might be drawn between different epochs on the same model, or between models. Additionally, each training activity line displays two characteristic shapes: the peak and the tail. Models without a peak do not settle and, therefore, have not learned. After settling, the tail tends to approach a stasis activity level other than zero. If a model ever reached zero on all layers' LLC that would mean it is perfectly fit to the training data—an overfit condition.

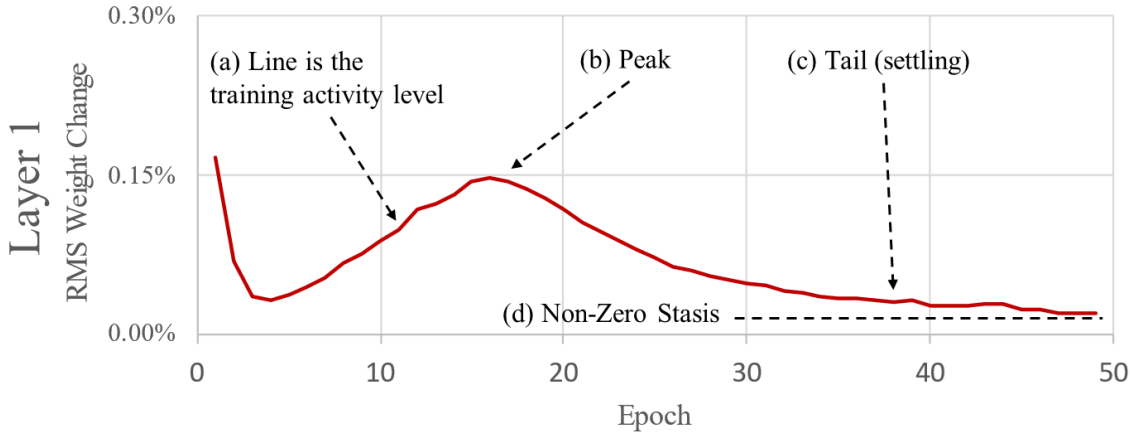
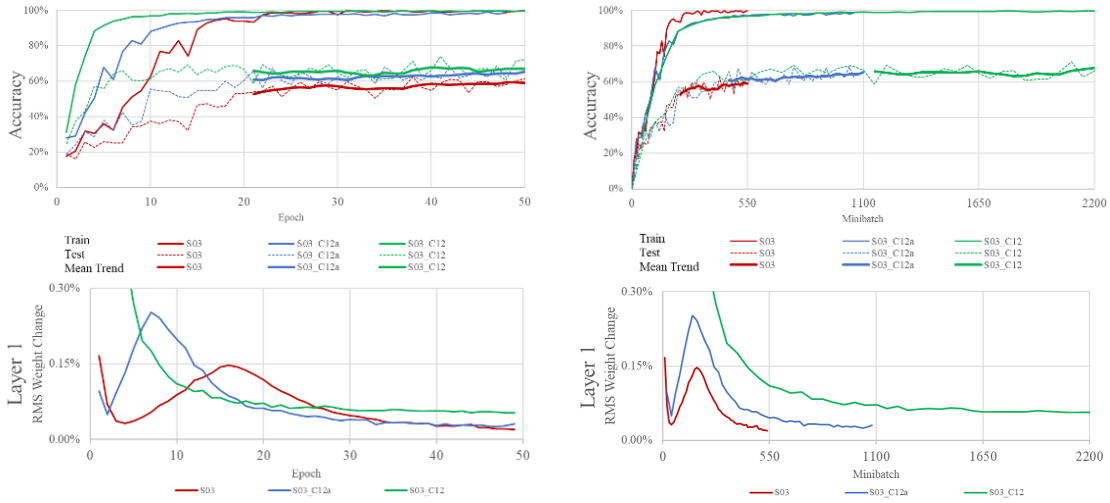


Figure 21. LLC showing key features and characteristic shapes

4. Minibatch

When datasets of different sizes are compared, larger datasets might appear to train faster because there are more minibatches in their epochs. Because the size of the minibatch

is limited by a system's memory and processor, it remains fixed at 100 images throughout this study, regardless of dataset size. Therefore, a dataset with twice as many images has twice as many minibatches each epoch. Training occurs after each minibatch even though analytics are only gathered at the end of an epoch. Therefore, minibatches can be thought of as *training cycles*. To compare training rates, the minibatch provides a more appropriate basis of comparison than epoch, as seen in Figure 22. This study creates of different sizes, so the number of minibatches is used for the x-axis when building comparative accuracy and loss charts.



(a) Accuracy and LLC using Epochs

(b) Accuracy and LLC using Minibatch

Figure 22. Displaying training results with epoch (a) versus minibatch (b)

D. VALIDATING THE WORKFLOW

Before this study's two-stage data augmentation technique can be confidently applied to classification data, the classifier and augmentation methods need to be validated. This section builds a validation process block-by-block.

First, the installation of AConvNet needs to be verified. It is convenient at this point to quickly investigate the effect of the phase channel, transferability to SAMPLE, and the effect of removing AConvNet's random translation augmentation.

Second, the two data augmentation steps are investigated individually and together to see if they truly benefit from being combined. Then the noise and clutter backgrounds are tested against measured chips to determine which to use during implementation. A single, best pseudo-resolution is also chosen as an exemplar for single resolution testing.

1. Using AConvNet

To confidently apply this thesis's novel data augmentation technique to AConvNet [18], a series of brief tests are run. First, the previous PyTorch results will be recreated from [30]. After that, small changes are made step-by-step until SAMPLE [3] can be used without augmentation. Once SAMPLE can be applied to AConvNet, data augmentation tests can be conducted without worrying that observed behaviors might be due to network architecture issues.

a. Recreating AConvNet's Standard Operating Conditions

This quick test simply uses PyTorch [41] to reproduce AConvNet's SOC results as prescribed by the authors [30]. This reproduction insures that PyTorch and AConvNet are installed correctly on the workstation. This test also verifies the AConvNet results from [18] and [30]. To examine the training behavior, the AConvNet code is modified slightly to enable continuous graphing of accuracy and loss on WandB.com [42]. The authors say on their GitHub site [30] that the final SOC accuracy in the PyTorch implementation is 99.34%, slightly higher than the official accuracy of 99.13% at epoch 28 [18]. The PyTorch result is expected in this test. The training cycle is allowed to run for 50 epochs to ensure convergence.

b. Removing the Phase Channel

AConvNet optionally accepts two input channels: one each for magnitude and phase. The original [18] and PyTorch implementation [30] of AConvNet both use MSTAR with two channels: magnitude and phase. This makes sense because phase is significant and trainable for a CNN [21]. However, this study uses single channel, magnitude-only SAR images from SAMPLE to simulated operational implementation.

To directly view the effect of removing the phase channel, the second channel is removed from AConvNet's original MSTAR dataset using their built-in data generation options [30]. Accuracy and loss for both two-channel and phase-only training runs are overlayed on one set of charts. The general training behavior and maximum accuracies are examined.

Based on AConvNet's suggested optimal case, the magnitude-only dataset was only run for 28 epochs. Much later it was discovered that a 50-epoch run was more suitable, but time did not allow this test to be re-accomplished.

c. Using SAMPLE Without Translations

After installing AConvNet and removing the phase channel dependency, the next natural step is to ensure that the model can train on the new SAMPLE baseline dataset. Ideally, this would be a two-step process of changing datasets and removing translation augmentation independently. Unfortunately, there are issues employing the translation augmentation from [30]; mainly that there is no indication of how to restrict dataset expansion to only 2700 images of each object class [18]. Unable to isolate the change to SAMPLE and removal of translation augmentation, this test changes both factors simultaneously.

The translational data augmentation technique is baked into AConvNet's code for dataset loading [30], but can be circumvented by taking advantage of the implementation method. Because the MSTAR images have excessive background and are not uniformly shaped, the researchers in AConvNet crop images down twice—first to 94x94 pixels, then to 88x88 pixels. These cropping procedures are applied in two steps: first during dataset generation (preconditioning), second during dataset loading just prior to training. During dataset generation, AConvNet expands their dataset by cropping with a 94x94 sliding window technique. This expansion is applied to the full dataset prior to splitting it 80/20 into train and validate subsets. During data loading the training subset is further augmented to prevent overfitting using an 88x88 random crop. Leveraging knowledge of these two cropping steps allows two reversible ways to remove the cropping without significantly modifying the original code.

The first option skips dataset generation and crops all new images to 88x88 pixels prior to dataset loading. In this way, the random cropping routine must always choose the entire image without translation. The second option adds a third method in the data loader: train, *train2*, and test. The new *train2* method uses the *test* method's center-cropping data loader while building the training subset in memory. Because the train/test split is originally made with a Boolean-type *is_train* flag, this method requires modifying AConvNet's code to accept a string-type flag. Both methods—presenting 88x88 pixel data and enforcing center cropping—are redundantly implemented.

After circumventing the random translation mechanism, non-augmented SAMPLE datasets can be created. The first two are S03 and M03, based on the original synthetic and measured datasets, respectively. Because these two are part of the larger family of datasets, they also have M35 and M548 objects removed, resulting in 1088 images in each dataset.

Also significant to this study, shadow inclusion is enforced during the cropping of S03 and M03. In this way, this short test also shows the effect of using shadows and objects together. Because the new datasets do not use translation, it is not an apples-to-apples comparison, but it is still interesting.

These two datasets are used to train AConvNet, and the accuracy and loss charts are displayed. The SOC training lines and S03 test lines are also included as they help guide discussion of the results. Note that the SOC and M03 datasets, being measured data themselves, have no measured test dataset for comparison.

2. Two-Stage Data Augmentation

This section contains two sets of tests related to the novel two-stage data augmentation itself. The first test isolates the two stages of data augmentation—reduced resolution and background substitution—to ensure that their proposed combination is mutually beneficial. The second tests are used to select a background texture and pseudo-resolution for the implantation tests.

a. Background, Resolution, or Both?

In the first test, the kernel operations and background replacement operations are examined as independent data augmentations using the S12 and C03 datasets, respectively. The C12 dataset is then used to show the effect of combining both augmentation techniques. Each of the three augmented datasets and the S03 control are used to independently train AConvNet for 50 epochs.

The S12 dataset is created by *not* changing the background after downsampling to a pseudo-resolution of 1.2m. The C03 dataset downsamples with a 1:1 ratio before changing the background. The C12 dataset, of course, runs the entire workflow with a 1.2m pseudo-resolution and changed background.

As an aside, the 1.2m pseudo-resolution was selected because the test runs were recursive; 1.2m was determined to work the best before the final run-through of this comparative test. Similarly, the clutter background was used because it is superior to the noise background. All of that will be shown in Chapter IV. Results subsection A.2.Two-Stage Data Augmentation.

To control for dataset size and observe augmentation effects in isolation, none of these datasets are combined with S03. Therefore, each dataset contains only 1,088 images. The training and test accuracy and loss of all four training runs use the epoch standard x-axis. Each run is continually tested against the M03 dataset. Because the test accuracy and loss are somewhat variable from epoch to epoch, the mean values of ± 3 epochs are overlaid as a trendline after epoch 20 to determine overfitting.

b. Different Backgrounds

What is the training effect when changing the background that the segmented objects are alpha-blended onto? If AConvNet can train just as well with a noise background as a clutter background, then infinite wide-area images could be synthesized by randomly placing segmented objects onto a swath of noise. However, if a clutter background provides better test accuracy, then coherent background information is creating a model which is more robust.

In this second test, the noise and clutter backgrounds are compared to each other during training and continuous testing against M03. Both the best background and the best pseudo-resolution are determined. The S03 results are also included as a control group for comparison.

(1) Comparing Training at Different Pseudo-Resolutions

This test compares eight different training runs—four pseudo-resolutions each of the noise and clutter backgrounds—and a ninth control run of S03. Because of the large volume of data, the accuracy and loss charts cannot be displayed in a single figure. Instead, the runs are broken out by pseudo-resolution. Using the mean-value trendline in the loss chart, an arrow points out the epoch of overfitting (if present) in each dataset.

(2) Comparing Test Accuracy

The test accuracies from the previous experiment are compared to determine the best pseudo-resolution. If overfitting is present, then the test accuracy from that epoch is used. Both noise and clutter are included to further emphasize the selection. The relationship between training and test accuracies is also discussed before testing implementation of the data augmentation.

E. IMPLEMENTATION

The intended use case—synthesizing wide-area SAR imagery for a single-stage object detector—cannot be tested without truth data. Therefore, two supporting implementation modes are tested instead. First, the hypothesized training mechanism of low-resolution imagery, deep layer training, is tested using a transfer learning framework. Second, the training capacity of the augmented data is explored using combined datasets of different sizes and resolutions.

1. Deep Layer Training

The LLC provides a critical tool for observing the training behavior of different datasets. A CNN attempts to train on any data that is presented, so it is impossible for the augmented datasets to completely bypass the initial layers. The intent is for augmented

datasets to show less training activity than the S03 dataset, but observing the effects with isolated datasets is difficult. This is likely because the CNN trains more quickly in the early epochs regardless of the image pseudo-resolution. The difference between initialization weights and epoch 15 trained weights is so large in each case that there is a great deal of training activity. Viewing the different behaviors and judging if deep layer training is possible requires a way to skip this initial flurry of activity.

To assess the possibility of deep layer training and characterize its performance, two things are needed: a way to change the behavior and a way to view the change. The mechanism of change is controlled by presentation of data to the CNN. The augmented datasets are presented to the models either through transfer learning, preconditioning, or combining with S03. The LLC allows inspection of training activity and viewing of change.

a. Dataset Presentation

Training behavior is affected by how the augmented dataset is presented to the CNN. To view these behaviors, the presentation is changed by leveraging transfer learning, preconditioning, and combined data augmentation. As previously stated, this nomenclature is forward-looking. In this thesis, transfer learning and preconditioning are essentially the same, save for the order of dataset presentation. In both transfer learning and preconditioning, the model is fully trained on one dataset, then retrained on a second. The combined data augmentation mode simply combines both previous datasets into one larger dataset. This acts as a control for the deep learning inspection but can also be used to show the effect of data augmentation over the non-augmentation.

To compare the behavior of these three modes, three independent AConvNet models are trained using datasets S03 and C12a in three arrangements. Model 1 represents traditional transfer learning by training on the S03 control dataset and then transitioning to C12a augmented dataset. Model 2 represents preconditioning by transferring from C12a to S03. Model 3 represents traditional data augmentation by combining the C20 and S03 datasets. In Model 3 the original datasets are divided into training and testing sets prior to combination to ensure even representation. A visual depiction of these combinations can be found in Figure 23.

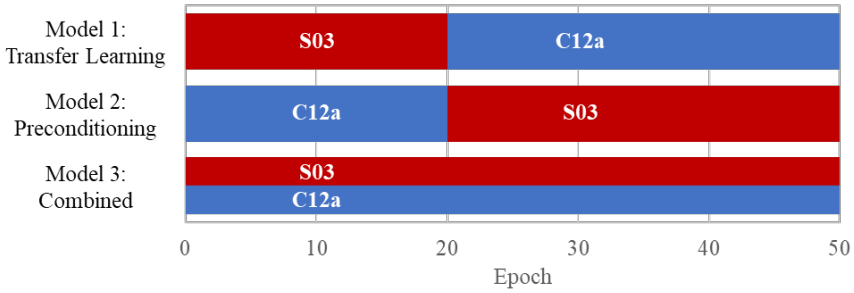


Figure 23. Dataset presentation schedule for deep layer training

The transfer learning techniques used in this test are very straightforward to avoid creating additional variability. Models are trained for 20 epochs on the first dataset, the entire dataset is changed, and then the model continues training for an additional 30 epochs. No layers are reset during the transition, and no layers are prevented from continued training. Transferring datasets after 20 epochs ensures that the peak of training activity in the LCC has been passed, but none of the models' accuracy have converged. The total training length of 50 epochs is based on stabilization of the training accuracy curves of both transfer and preconditioning learning models.

Accuracy and loss charts are used to observe any long-lasting effects from training the first 20 epochs. The test results, as measured against M03, are the most significant portion of these charts. Training and testing lines for S03 and C12a are also included past the transfer point in a subdued, dashed line. These extra lines show what Models 1 and 2 would have done without transfer learning.

b. Training By Layer

The LLC is the primary tool for observing training activity levels. The goal of the three training techniques—transfer learning, preconditioning, and combined data augmentation—is to observe the ability of deep layer training to bypass the initial convolution layers of a CNN. This study's two-stage data augmentation technique borrows from the kernel methods of the convolution and pooling layers. It should be able to bypass initial layers and "inject" its data into the first layer that matches its pseudo-size, as seen in Figure 24. If deep layer training is achieved, the representative C12a dataset should have

less effect on the trainable convolution layers that it is bypassing. This is directly observable on the LLC.

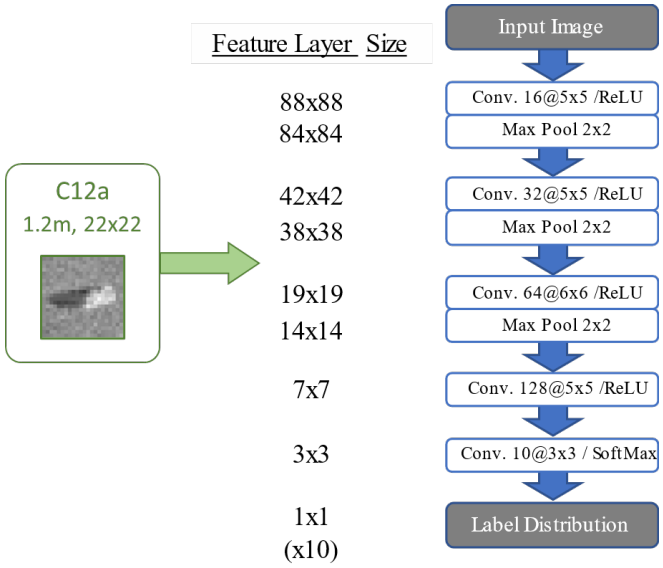


Figure 24. Theoretical injection point of C12a dataset based on pseudo-size. Architecture (right) adapted from [18].

A few behaviors can be predicted based on the assumed bypassing mechanism. First, C12a should induce less training than S03 in the bypassed layers because there is less high-level information available for training. Second, new background information is being presented in both Model 1 and Model 2, so they should both see some retraining after epoch 20. Again, because C12a contains less high-level information, retraining in the bypass layers should be less for Model 2 than Model 1. Third, because accuracy convergence takes longer in C12a, the training behaviors should also appear delayed compared to S03.

Model 3's combined data augmentation does something different. It controls for the effects of transfer learning while still allowing the model to see both datasets. Transfer learning is not the purpose of this study; it is only a tool to elucidate deep-layer training behavior. Model 3 represents the method used in the remainder of this thesis. Because there are more available images in Model 3, the LLC shows more training during early epochs

in each layer. The inclusion of Model 3 in this chart provides the opportunity to see which parts of the isolated datasets influence combined training behavior.

To view the effects of the three implementation techniques, training and test accuracy are compared for all three models in an overlaid accuracy chart. Because the test accuracy is somewhat variable from epoch to epoch, the mean values of ± 3 epochs are overlaid as a trendline to assess comparative test accuracy between models. There is no need to characterize the accuracy of these models, so the overall test accuracy is not calculated. Alone, the accuracy chart does not provide a means of observing deep layer training. The LLC is also required.

The LLC compares the internal training activity levels of all three models. LLC lines for S03 and C12a are also included past the transfer point in a subdued, dashed line. These extra lines show what Models 1 and 2 would have done without transfer learning. The training difference after dataset transfer is also highlighted and labeled for clarity.

Finally, all charts use the minibatch basis of comparison. To control for dataset size, only single-augmentation datasets of 1,088 images are used for transfer learning and preconditioning. However, combining the two datasets into Model 3 means doubling the size of the dataset. Therefore, minibatch is required for behavioral comparison. The charts are truncated to 600 minibatches because the focus is Models 1 and 2, which finish after 550 minibatches. Though Model 3 continues until 1,100 minibatches, the final values of that model are less important than the inter-model comparison.

2. Expanding Datasets with Data Augmentation

The final set of experiments explore the possibility of expanding the SAMPLE dataset to different sizes and with different resolutions. Two-stage data augmentation allows objects of interest to be overlayed onto infinite backgrounds and many pseudo-resolutions, but there is no way to calculate the exact dataset size requirement. The ideal dataset size varies with each AI/ML network and object class. Instead, to explore the possible benefits of dataset expansion, this thesis runs two tests based on backgrounds and multiple resolutions. An overview of the dataset composition is laid out in Table 3, and a visualization of their expansion is displayed in Figure 25 as scaled volumes.

Table 3. Expanded dataset composition

Dataset	Composition	Images	Minibatches per Epoch
S03	S03	1,088	11
S03_C12a	S03, C12a	2,176	22
S03_C12	S03, C12a/b/c/d/e	6,528	54
CMR	S03, C03a, C06a, C12a, C24a	5,440	54
Super_CMR	S03, C03a/b/c/d/e, C06a/b/c/d/e, C12a/b/c/d/e, C24a/b/c/d/e	22,848	229

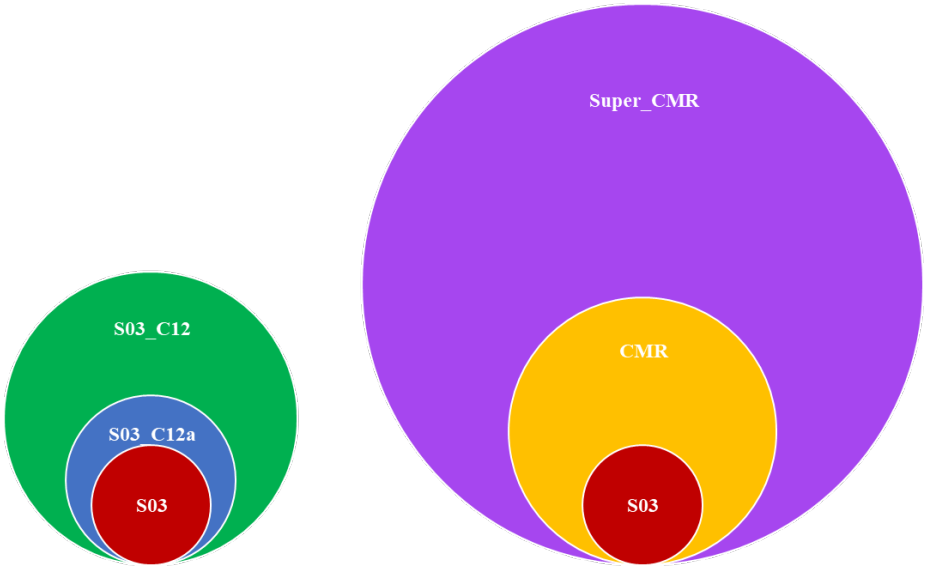


Figure 25. Expanded dataset size represented as volumes (to scale)

a. Single Pseudo-Resolution Dataset Expansion

This test observes the effect of increasing dataset size by changing the background. The pseudo-resolution is held constant at 1.2m during the creation of five groups of 1,088 augmented images, each with a random clutter background. To view the effect of

expansion, the S03 control test is compared to S03_C12a and S03_C12. The first of these combined datasets—S03_C12a—uses only the *C12a* group of 1,088 augmented images for a total size of 2,176 images. The second combined dataset—S03_C12—uses all five groups of C12 for a total size of 6,528 images.

Training and concurrent testing are displayed on accuracy, loss, and LLC charts for 50 epochs and then for 3,300 minibatches, representing all 50 epochs of S03_C12 training. Because the test accuracy and loss are somewhat variable from epoch to epoch, the mean values of +/ 3 epochs are overlaid as a trendline after epoch 20 to determine overfitting.

b. Multiple Pseudo-Resolution Dataset Expansion

The final test continues the multi-resolution idea from [23] by combining S03 with all the pseudo-resolution datasets. The CMR dataset contains S03 and the 1,088-image *a* group of all four pseudo-resolutions for a total of 5,440 images. This size is similar to S03_C12 which has 6,528 images from just S03 and C12 groups. The second dataset in this test is named "Super_CMR" and contains S03 plus the 4,352 images from each full set (of *a, b, c, d,* and *e*) of the four pseudo-resolutions giving a total dataset size of 22,848 images. This size is roughly comparable to the 27,000-image AConvNet SOC dataset created by geometric translation. Unfortunately, AConvNet used only measured data with magnitude and phase so the results cannot be compared.

Training and concurrent testing are displayed on accuracy and loss charts first for 50 epochs and then for 4,620 minibatches, representing 20 epochs of Super_CMR. Super_CMR is very resource intensive as it performs 231 minibatch training cycles per epoch (more than 20 epochs worth of training in S03). Therefore, it is stopped early at epoch 20—long after convergence. Because the test accuracy and loss are somewhat variable from epoch to epoch, the mean values of ± 3 epochs are overlaid as a trendline after epoch 20 (epoch 10 for Super_CMR) to determine overfitting.

c. Similar Dataset Sizes with Different Compositions: S03_C12 and CMR

This quick review compares the results of the similarly sized S03_C12 and CMR datasets. No new information is created, but new conclusions are drawn. In this case, the

dataset size is roughly controlled while the augmented pseudo-resolution composition is changed.

All charts use minibatches. The similarly sized datasets train at roughly the same rate, but because S03 is still used as a baseline, the epoch method would be misleading about training rates.

d. Accuracy Improvement with Dataset Expansion

This section again compiles information from previous tests. This time the compilation demonstrates the overall potential of dataset expansion using the proposed two-stage data augmentation technique. A quad chart is prepared that combines the volumetric representation of dataset sizes with charted test accuracies. The test accuracies are presented three ways — as a simple bar chart, as a line diagram comparing test accuracy to dataset size, and again as a line diagram with log-scale dataset size. The log-scale diagram is included because the non-log diagram looks suspiciously like a logarithmic pattern. By plotting on a log-scale, it is easy to discern whether or not the trend is logarithmic or approaching a limit.

Additionally, these final classification results can be used to briefly investigate the importance of segmentation artifacts. A potential criticism of alpha blending is the introduction of artifacts that could negatively affect training. If data augmentation adds artifacts, then the AI/ML might treat them as features and use them to learn classifications. This would be similar to the background association issue experienced with MSTAR [11], [25]. Artifacts of alpha blending could include (but are not limited to) mismatched preprocessing intensity levels, background information contained in object mask area, or the detectable outline of the segmentation mask itself. These issues are not directly addressed elsewhere in the thesis because they are very difficult to prove or disprove during chip classification. However, if classification of measured SAR images is *improved* by training with augmented data, then the CNN has not trained on segmentation artifacts. If artifacts are playing a significant role, then the classification accuracy against unaugmented images would decrease.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS

There are two series of tests in this thesis: one that validates the architecture and data augmentation steps, and one that examines two implementations of the data augmentation technique. Because results from the validation tests directly support the findings of the implementation tests, it is cognitively useful to keep these results sequential. The results of each test build continuously; therefore, the setup for the tests are presented previously in the Methodology chapter. To allow easy cross-reference for readers who may want to review the setup for each test, the section headings are paralleled in each chapter.

A. VALIDATING THE WORKFLOW

These first two tests validate the CNN architecture and the data augmentation steps. They are important and necessary to build confidence in later tests. They also introduce crucial findings that support conclusions about the data augmentation's operating principles.

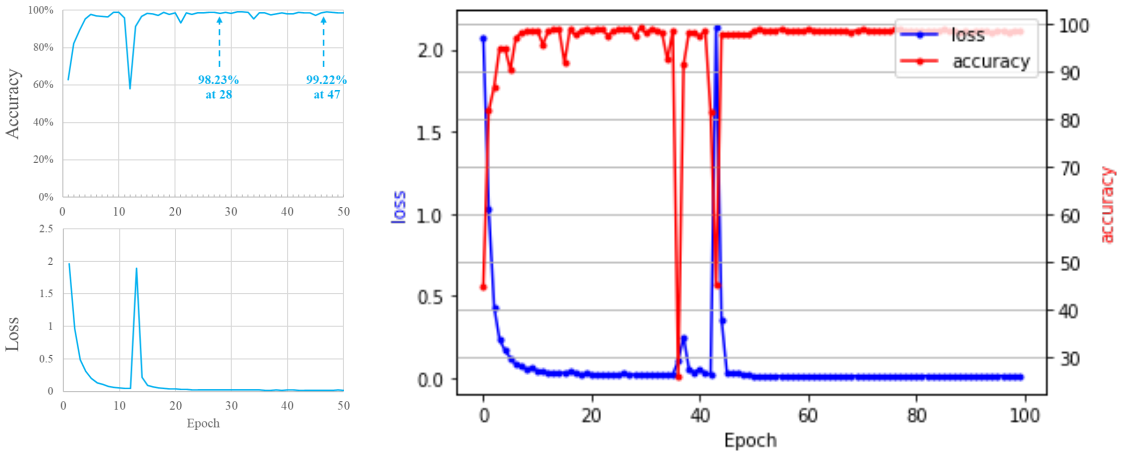
1. Validating AConvNet with SAMPLE

This test performs several important validations. First, it ensures that the PyTorch version of AConvNet [30] has been installed correctly. Then it walks back assumptions of the AConvNet paper [18] until the SAMPLE dataset [3] can be applied.

a. Recreating AConvNet's Standard Operating Conditions

Two results are expected in this quick verification test. The first—verify installation of PyTorch and AConvNet—is successful. Second is verification of AConvNet's results and their expected accuracy of 99.34%. As seen in Figure 26(a), this test produces a 98.23% accuracy at epoch 28 as suggested in [18] and a maximum 99.22% accuracy at epoch 47. Still, this maximum accuracy result is higher than the original 99.13%, so the test result is still reasonable. Additionally, the accuracy is consistently near 98% after epoch 15.

A potential reason for the deviation lies in details of the PyTorch implementation of AConvNet [30]. This implementation lacks any mechanism to constrain the datasets to 2,700 images per object class. Instead, instructions for recreating SOC direct a step-down in chips size prior to dataset generation. Yet the provided code does not implement the change. The results in Figure 26(a) use that provided code and the results do not exactly align with the provided results in Figure 26(b). However, they are largely similar. Furthermore, common practice at the time of [18] said that the highest accuracy should be reported despite the fluctuations. This implies that current results are plausible and that AConvNet has been successfully verified.



(a) Results of SOC recreation

(b) Expected results

Figure 26. Recreating the AConvNet SOC results. Expected results source: [18].

b. Removing Phase Channel

After removing the phase channel from AConvNet's MSTAR dataset, the model is still able to train. As seen in Figure 27, the magnitude-only model reached a maximum of 96.95% at epoch 11 and again at epoch 21. Despite the significant fluctuations in accuracy, loss continued to decrease smoothly from 0.02621 at epoch 10 to just 0.0199 at epoch 20. This shows that the model continued to train though the accuracy did not improve. During this time, the two-channel model's accuracy also fluctuated, but to a lesser degree. These fluctuations prevent comparison of accuracy at the same epoch. Instead, the maximum of

each model prior to epoch 28 is used because the magnitude-only model is only run for 28 epochs. Including phase data allows the two-channel model to reach a maximum of 99.01% accuracy while the magnitude-only model only reaches 96.95%.

Removing the phase data results in a loss of 2.06 percentage points (a 2.08% decrease). This is a notable loss and would move AConvNet back two positions in [5]. However, it is still too difficult to predict the effect of segmentation and alpha-blending on phase representations. Therefore, the current study will proceed with magnitude-only SAR images knowing that there are potential improvements to be made in future work.

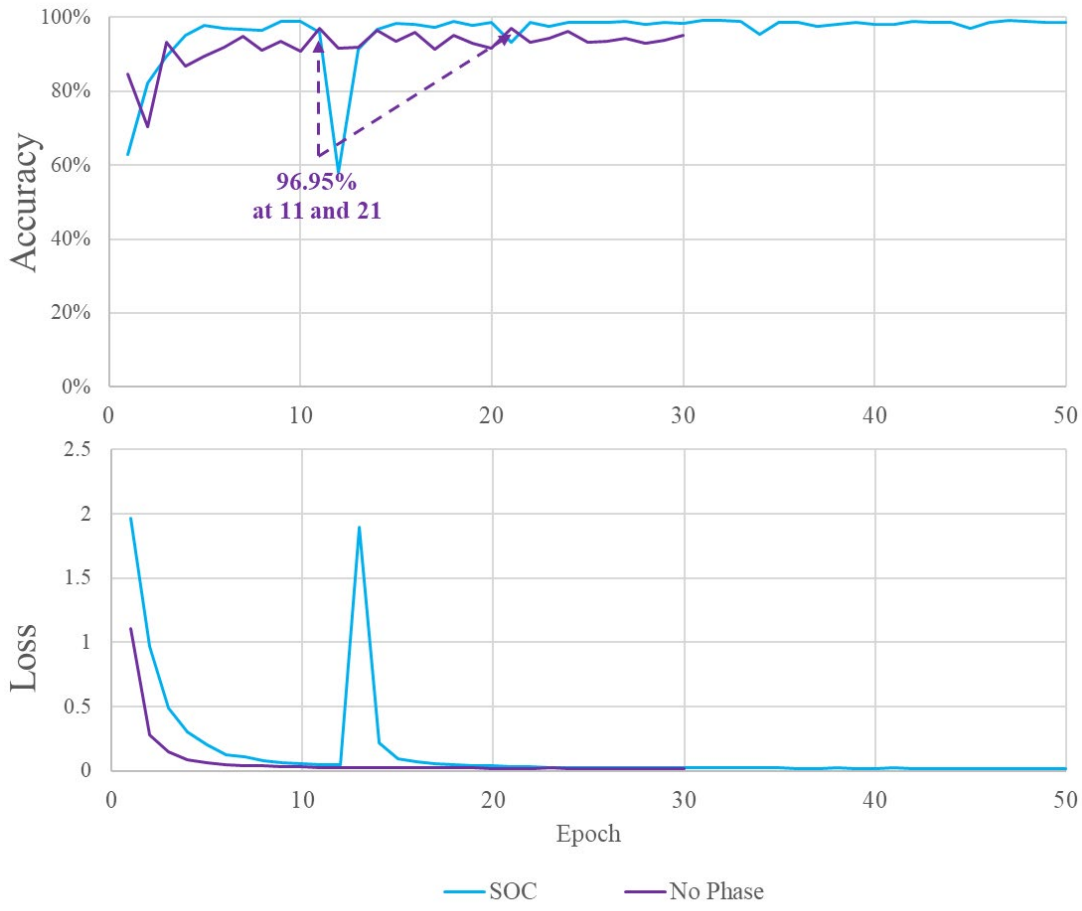


Figure 27. Removing the phase channel from MSTAR dataset

c. Using SAMPLE Without Translations

Accuracy and loss charts for SOC, S03, and M03 are shown in Figure 28 along with the S03 Test lines. The SOC and M03 datasets, being measured data themselves, have no measured test dataset for comparison. Epochs are used for the x-axis despite the difference in dataset sizes. To reiterate, this is not an apples-to-apples comparison. Attempting to standardize based on minibatches would not alleviate issues from different data augmentations, shadow inclusion, and image preconditioning. Rather than attempt to draw that comparison, only two observations can be reliably made from these charts.

First, both the S03 and M03 datasets are able to train AConvNet without translation augmentation. Both datasets converge with high accuracy. The S03 dataset reaches 100% training accuracy after 31 epochs; the M03 dataset reaches 97.5% after 35 epochs. The training loss Figure for both datasets continue to decrease slightly after these points, indicating that overfitting is likely. The magnitude-only M03 dataset achieves lower accuracy than two-channel MSTAR dataset in [18], as expected from the previous test where phase is removed. It is interesting that the magnitude-only M03 dataset, which is derived from MSTAR, reached a higher accuracy than AConvNet's translated MSTAR without phase data. This difference may be due to image preprocessing, as suggested in [9].

Second, the synthetic SAMPLE data trains the model more rapidly than measured SAMPLE data. Several things could have caused this difference: disparities between real and rendered backscatter signatures, differences in background texture, and differences in actual and simulated illumination levels are several likely reasons. Most telling is the fact that S03 reaches a higher training accuracy than M03. S03 Test, which is trained on S03 data and tested using M03 data, keeps pace with M03 until epoch 18 where they diverge. There is something about the synthetic data that makes training easier but eventually unrealistic.

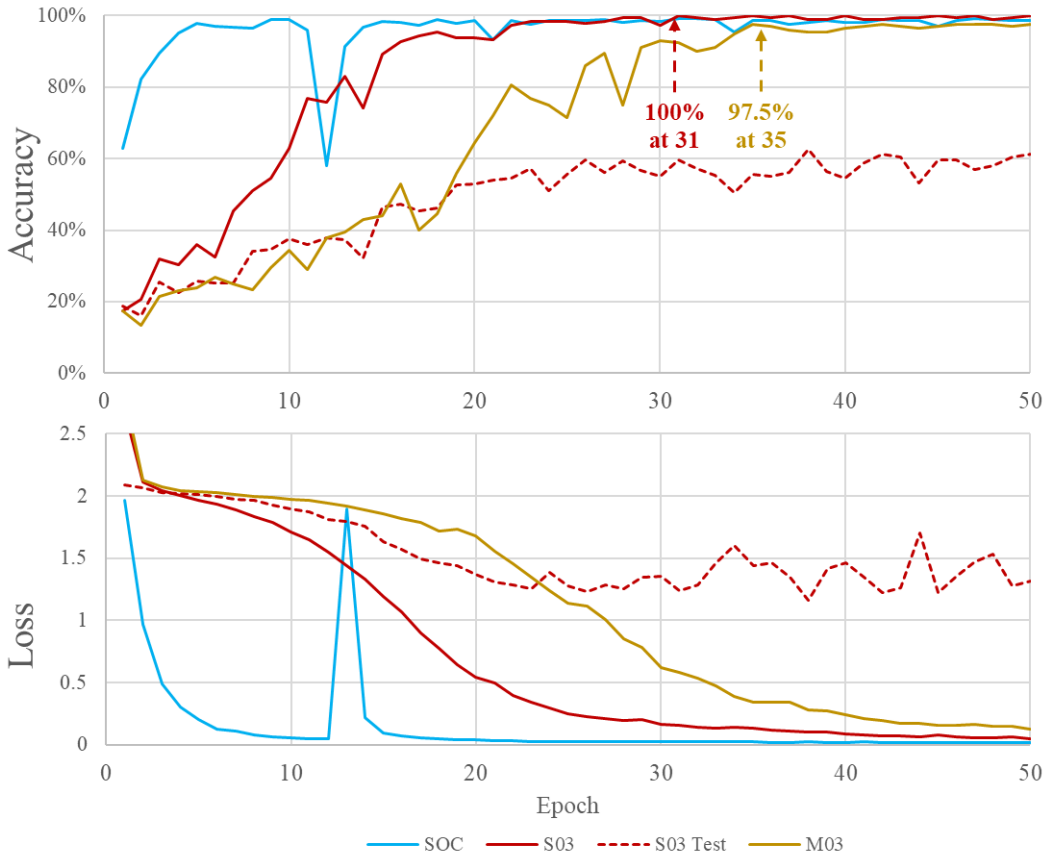


Figure 28. Successfully training AConvNet with SAMPLE data

2. Two-Stage Data Augmentation

This section contains two sets of tests related to the novel two-stage data augmentation itself. The first test isolates the two stages of data augmentation—reduced resolution and background substitution—to ensure that their proposed combination is mutually beneficial. The second tests are used to select a background texture and pseudo-resolution for the implantation tests.

a. Background, Resolution, or Both?

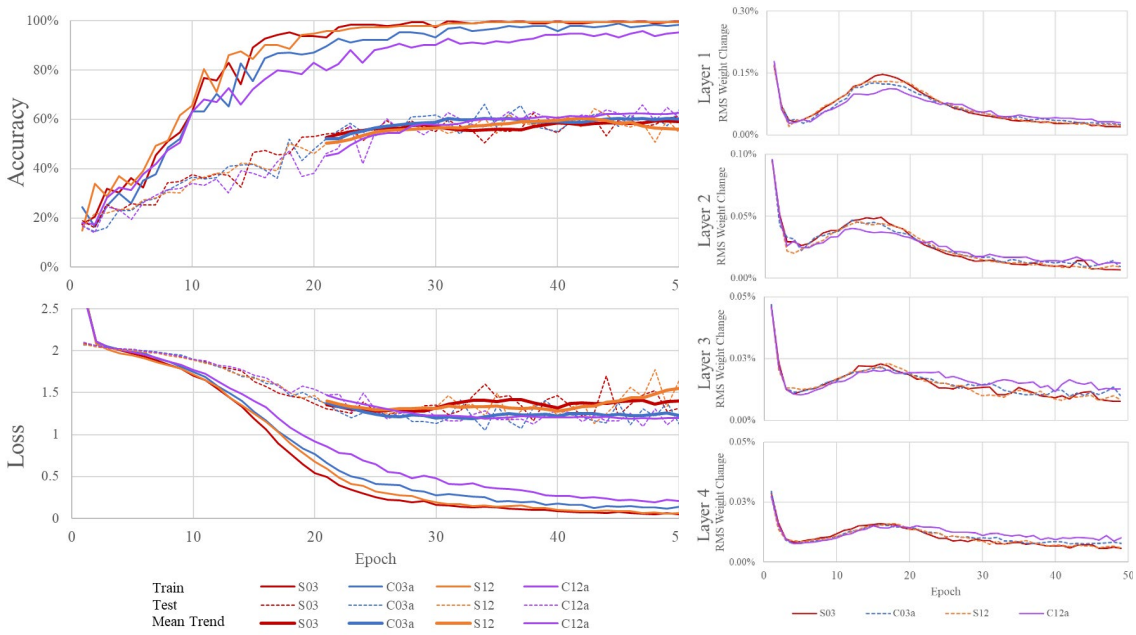
Two backgrounds and two pseudo-resolutions can be matrixed into four independent datasets: S03 with no modification, C03a with only new backgrounds, S12 with only coarse pseudo-resolution, and C12a with both new backgrounds and coarse pseudo-resolutions. To control for dataset expansion, each of these four datasets contains

1,088 images. By comparing the training behavior and results, these four datasets determine if one factor is more significant and if the augmentations benefit from being combined.

Observing Figure 29(a), all three augmented datasets have nearly the same test accuracy as the S03 control, despite variations in training accuracy. Even though the C12a dataset has the lowest training accuracy, it has a marginally higher test accuracy than the other three. Additionally, using both augmentation techniques in C12a reduced loss the most and delayed overfitting the longest.

Changing the resolution between C03a and C12a has less effect than changing the background from either S03 to C03a or from S12 to C12a. This has an easy and expected explanation. While convolution layers see little difference between backgrounds in the S03 dataset, each background in C03a is unique and requires robust rejection mechanisms. In this regard, there is little difference between C03a and C12a. Both have diverse backgrounds that prevent overfitting. Therefore, both C03a and C12a prevent overfitting in the same way. Without overfitting, both continue to train and increase their accuracy compared to S03 and S12.

The improved test accuracy from C03a to C12a is even more interesting because it indicates that the two data augmentation steps perform best when combined. After AConvNet's convolution layers, the max pooling layers downsample the images. The two-stage augmentation of C12a is designed to largely bypass these mechanisms while preventing overfitting. This effect can be seen in Figure 29(b) when the peak difference between C03a and C12a collapses in layer 3. Since C03a only has slightly more high-level object data than C12a, this difference might help explain C12a's improved test accuracy against measured M03. Perhaps the high-level differences between measured and synthetic SAR images are causing unidentified overfitting problems. In other words, the synthetic objects might be wrong in specific ways that reduce their effectiveness. By removing fine detail in the objects, the synthetic representations might become more general. This bodes well for future use in wide-area SAR synthesis as well as configuration variation.



(a) Accuracy and Loss Charts (b) Layer Look Chart

Figure 29. Improved test performance by combining augmentation techniques

Note that S03—the unaltered synthetic SAMPLE dataset—begins to overfit the M03 test set at epoch 28, as indicated by the increasing mean loss curve. Coincidentally, this is the same epoch that [18] describes as AConvNet's best model with MSTAR. Unfortunately, these results cannot be directly correlated because the MSTAR dataset in [18] is expanded using translation augmentation. Based on the number of minibatches, the 28th epoch in [18] has experienced many more training cycles with their larger, translated dataset.

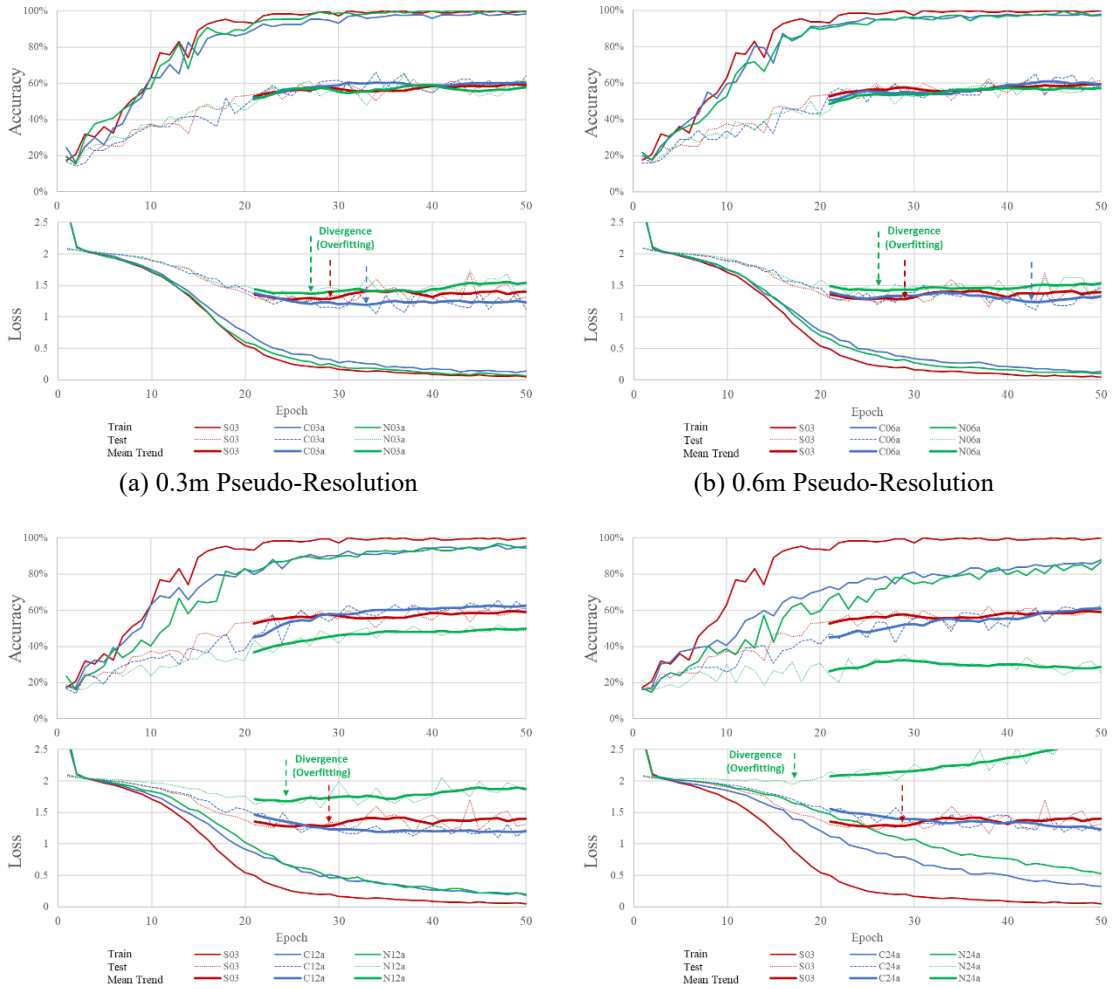
b. Different Backgrounds

This series of tests determines which background and pseudo-resolution should be applied in the implementation tests. It also begins to show evidence for a crucial factor—background variance—that is the driving force behind implementation success.

(1) Comparing Training at Different Pseudo-Resolutions

The results of eight training runs and the control S03 run are displayed in Figure 30. One of the most interesting observations from this test is that training accuracy does

not necessarily correlate to test accuracy as backgrounds are changed. Training accuracy continuously decreases at lower resolutions for both clutter and noise, yet the clutter test accuracy increases! Furthermore, the noise test loss shows more and earlier overfitting than clutter at each pseudo-resolution. Together, these indicate that the coherent clutter background is more difficult for the CNN to learn. The clutter background significantly reduces overfitting compared to both the noise background and the original SAMPLE background. Likely due to increased coherent background variance, the clutter background makes the synthetic SAR imagery more robust and more suitable for use with measured SAR images.



(a) 0.3m Pseudo-Resolution

(b) 0.6m Pseudo-Resolution

(c) 1.2m Pseudo-Resolution

(d) 2.4m Pseudo-Resolution

Figure 30. Comparison of noise and clutter backgrounds at different pseudo-resolutions with overfitting epoch identified

(2) Comparing Test Accuracy

As Figure 31 shows, C12a has the highest mean test accuracy. This is likely due to the layers in which the 1.2m and 2.4m images are injected. Referring back to Figure 24, the 1.2m images inject after the second convolution/pooling layer while the 2.4m images inject after the third. There is no subsequent pooling after the 2.4m images are injected. These 2.4m images may have too little information remaining to train earlier layers, or they may simply be bypassing too much of the CNN. Either way, it seems that the 2.4m pseudo-resolution is insufficient for classification of objects similar to those in the test set under these conditions.

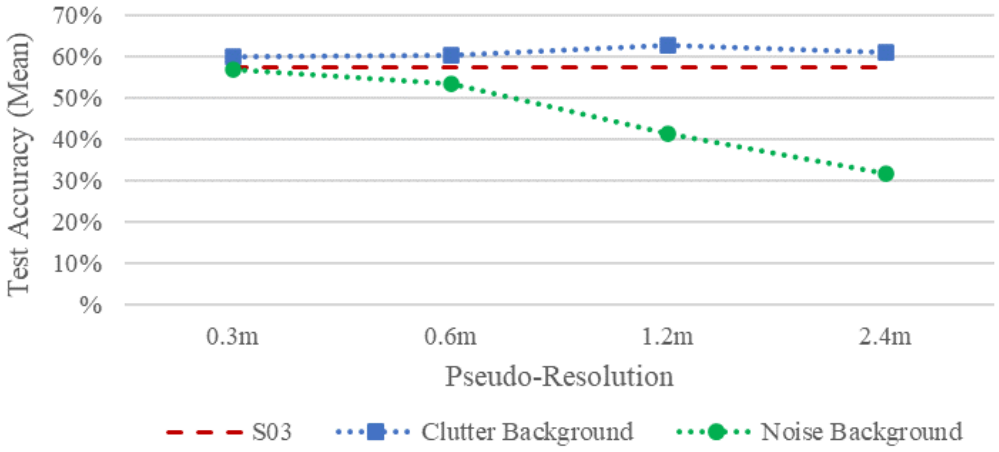


Figure 31. Mean test accuracy of noise and clutter backgrounds before overfitting

Despite the diminishing returns, C24a is still able to train AConvNet as well as C03a and C06a; therefore, all four pseudo-resolutions are still included in the multi-resolution datasets. When only a single pseudo-resolution is needed, C12 is used because it has the highest test accuracy.

Figure 31 also shows that the test accuracy of the noise background does not improve as pseudo-resolution decreases. Instead, the test accuracy of the noise background has similar decreasing returns to earlier low-resolution experiments with MSTAR in [22], [23]. While those results may not be surprising, the fact that all levels of clutter background were able to achieve better accuracy than unmodified SAMPLE is significant. Purely as

data augmentation, whether achieving deep layer training or not, the two-stage augmentation technique is able to improve the training capability of synthetic SAR images by substituting in measured backgrounds.

Furthermore, the sustained test accuracy of the C24a dataset provides a crucial insight for future wide-area SAR synthesis—the classifier works just as well on 2.4m data as on the original S03 data. It works slightly better, in fact. Training accuracy is steadily increasing through 87% after 50 epochs with no indication of overfitting, meaning that it is also able to classify low-resolution (coarse) images. Combined, the equivalent test accuracy and high training accuracy indicate that the proposed data augmentation technique has a good chance of success when used to synthesize wide-area SAR images.

B. IMPLEMENTATION

Finally, two potential implementation scenarios of this thesis's novel data augmentation technique are tested using narrow-FOV SAR chip classification. First, the theorized mechanism of deep layer training is investigated by employing transfer learning concepts and LLC figures. Second, the ability of the background-substitution mechanism to create infinitely large datasets is examined. The success of these expanded datasets provides the crucial support for future synthesis of wide-FOV datasets.

1. Deep Layer Training

In this test, the concepts of transfer learning are used to explore targeted deep layer training. After switching datasets, the deep layer training effect is very clearly observed. The order is also significant. Augmented data prevents overfitting; therefore, when it is presented last, the overall test accuracy is higher.

a. Accuracy and Loss

The first interesting indication from the accuracy and loss charts of Figure 32 is the clear training effect of dataset transfer on Model 1 and Model 2. Training accuracy immediately decreases for Model 1 and increases for Model 2. Similarly, the training loss lines also shift.

The second point of interest is mean test accuracy. Test accuracy for both transfer methods continues to improve after the dataset transfer. This indicates that, with regard to the measured test objects, the trained weights from the first 20 epochs are not spoiled during dataset transfer. Despite the detrimental training accuracy effect from the low-resolution C12a dataset, the test accuracy and test loss continued to improve.

At first these two results may seem counterintuitive, but remember that the training dataset is not only responsible for forward propagation. Results from training are also used to calculate training accuracy and loss before backpropagation. Therefore, it makes sense that the training curves show an immediate effect—the forward propagation weights in epoch 21 were trained on a different dataset! The test dataset, on the other hand, does not change. The weights, for their part, simply continue being refined.

All of the results from the accuracy and loss charts make sense for a transfer learning model. Next, observation of the LLC shows whether or not C12a is able to bypass the first two layers and achieve deep layer training.

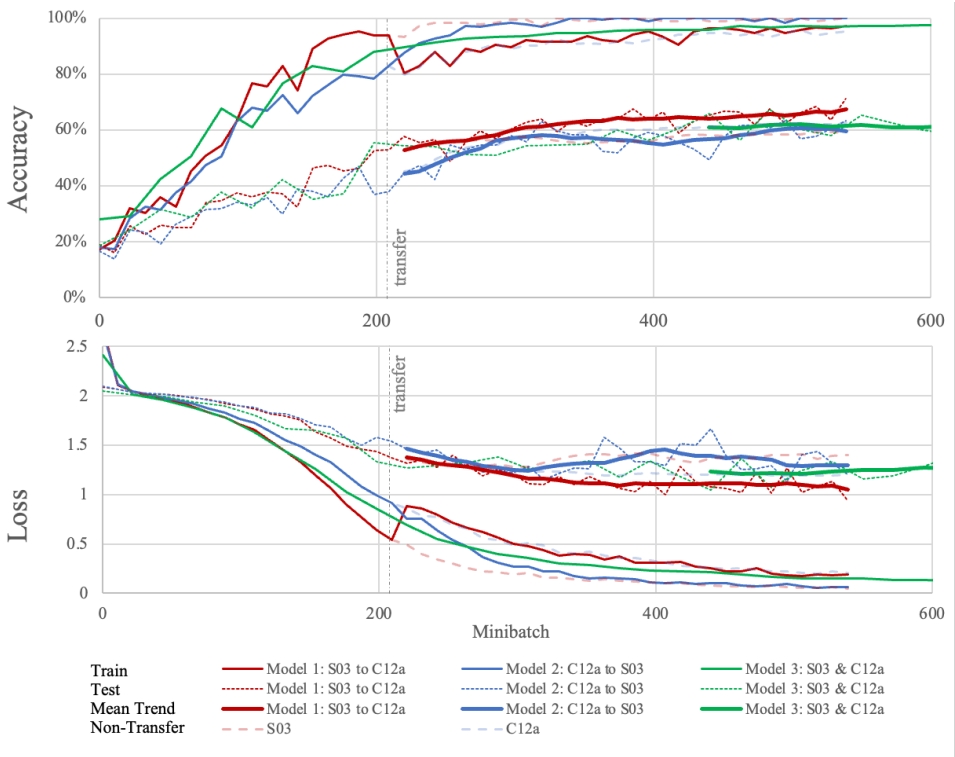


Figure 32. Transfer learning accuracy and loss

b. Layer by Layer Training Analysis

The distinct presence of deep layer training is observable after epoch 20's dataset transfer. Before discussing the evidence of deep layer training, it is necessary to explain the normal indications prior to dataset transfer. All models in Figure 33 display a characteristic "peak" and "tail" in training activity., but Model 1 reaches a higher peak training level than Model 2. Because of C12a's coarse pseudo-resolution, the peaks are noticeably lower than S03 in the bypassed layers 1 and 2. The peak does not occur noticeably sooner than S03, however, because the added variance of the clutter background delays model convergence.

After the dataset transfer, Model 1's C12a shows significantly reduced training in the bypassed layers compared to Model 2's new S03. This shows that deep layer training—the targeted injection of training data into deep layers—is possible. As expected, all layers see some level of retraining because the background texture has changed. However, Model 1's new coarse data is less accessible to the S03-trained layers 1 and 2. Starting at layer 3, the theoretical inject layer of C12a, the amount of retraining is similar between Models 1 and Model 2 and, therefore, between C12a and S03.

Finally, near the end of epoch 50 and minibatch 550, Model 1 training rates are noticeably higher than those of Model 2. This is again due to the clutter variance in the C12a background. Because the clutter backgrounds are all unique, there is not enough correlation for the model to learn those portions of the image. Instead, the training rate converges to a higher statis because the model is stuck forever trying to optimize those regions.

This finding may also explain why there seems to be an upper limit to test accuracy. The clutter background prevents overfitting of the image as a whole, allowing the CNN to continue training and continue improving test accuracy. However, if overfitting is being prevented in the background—which composes most of the image area—that would mask any overfitting in the limited areas of interest. There may be undetected overfitting localized to the synthetic object areas themselves. While background substitution allows

the maximal utility of the synthetic objects, it uncovers a limiting test accuracy based on inherent differences between the synthetic and real objects.

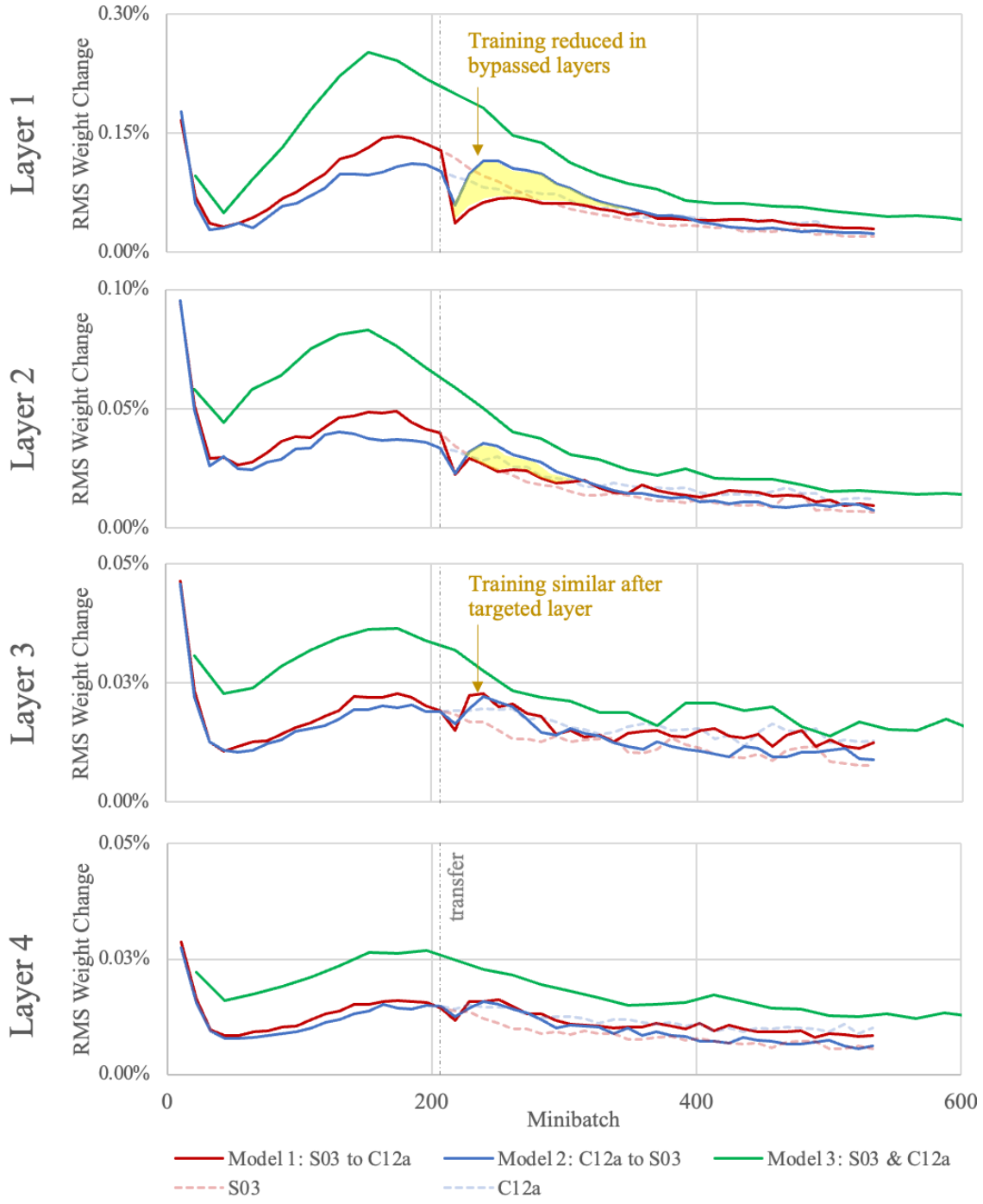


Figure 33. Transfer learning LLC showing bypassing effect of C12a

c. Lingering Effects After Dataset Transfer

A very interesting observation can be made using the subdued continuation lines for S03 and C12a in both Figure 32 and Figure 33; it appears that the training accuracy, training loss, and activity levels coalesce onto the levels that the final dataset would have reached. Model 2 ends with the behavior of S03 despite 20 formative epochs with C12a. There do not appear to be any "preconceived notions" preventing further training. Likewise, even after training on S03 for 20 epochs, the training behavior of Model 1 is most similar to its new C12a at the end of 50 epochs. Model 1 does not gain a "deeper understanding" of the objects' true representations by pretraining on the original images.

The test lines tell a different story, though. While Model 2 assumes the overfitting behavior of the S03 dataset along with the corresponding decrease in test accuracy, Model 1 outperforms its new C12a dataset in both test accuracy and test loss! There may not be a "deeper understanding" inherited from Model 1's time with S03, but the lack of background correlation in C12a seems to sustain valuable training on the objects. Because the objects themselves are unchanged from S03, the delayed overfitting results in more learning of coarse object features and, thus, higher test accuracy.

2. Expanding Datasets with Data Augmentation

This final set of tests examines the ability of the proposed two-stage data augmentation technique to create potentially unlimited datasets. First, a single pseudo-resolution is used with increasing datasets sizes. Second, different pseudo-resolutions are introduced. Results for similarly size datasets are compared to evaluate the potential benefits of using multiple pseudo-resolutions. Finally, the overall improvement is used to show that artifacts of segmentation are insignificant and, therefore, that this data augmentation method is suitable for use synthesizing wide-FOV SAR datasets.

a. Single Pseudo-Resolution Dataset Expansion

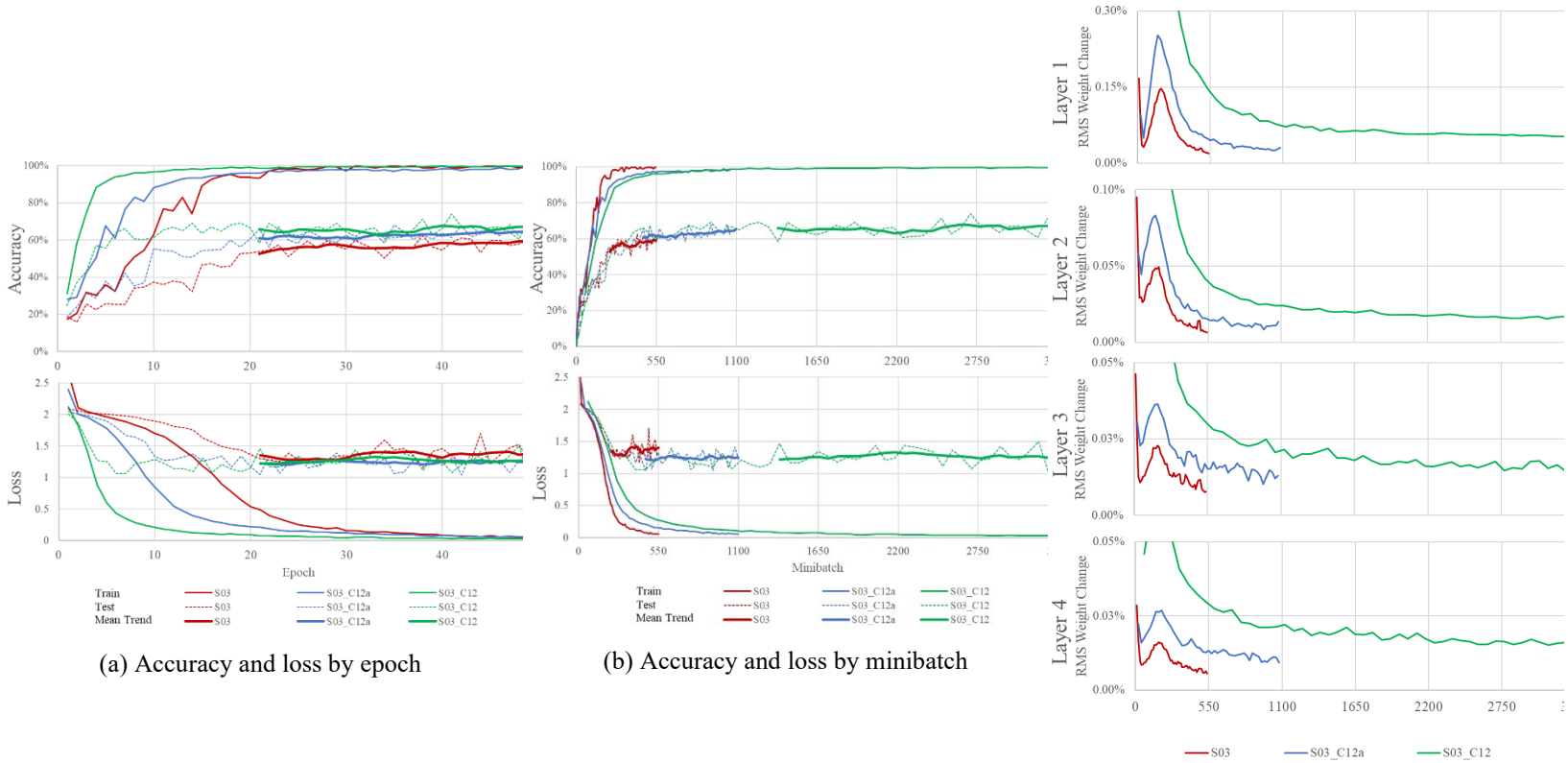
Datasets expanded with non-repeating, coherent backgrounds affect CNN training in many interesting ways, as displayed in Figure 34. Most of these results align with expected training enhancements; others are noteworthy results of increased background

variance. Note that Figure 34(a) uses epochs and Figure 34(b) and Figure 34(c) use minibatches. Epochs are useful for comparing test accuracy and loss between models, but minibatches, which represent training cycles, are necessary for comparing training rates.

First and foremost, the test accuracy increases with increasing dataset size. Comparing the mean test accuracy trendlines from Figure 34(a), it is easy to see stratification based on dataset size. Test accuracy is the most important factor because it determines the utility of using synthetic datasets to detect object in measured imagery. As seen in earlier tests, training accuracy does not necessarily correlate to test accuracy. It is expected that larger datasets with induced variance would train to a lower final accuracy if nothing else about the model changed. In fact, when all training accuracies are fluctuating above 98% they become irrelevant. The test accuracy is most significant. The fact that larger datasets are able to definitively improve test accuracy is significant and encouraging.

Second, larger datasets delay overfitting longer. Looking at test loss in Figure 34(a), neither augmented dataset exhibits obvious divergence or overfitting. Additionally, taking a training cycle viewpoint with Figure 34(b), the larger augmented datasets can continue training without overfitting for *much* longer than S03. As discussed in previous sections, delayed overfitting is likely due to background variance. This delay may allow more training on object characteristics and, therefore, may be the root cause of improved test accuracy.

Third, with training cycles in mind, Figure 34(b) shows a decreased rate of training accuracy improvement with each larger dataset. This is also likely due to variance in the background. Backpropagation in the convolution layers tries to discern correlations between images. While the segmented objects are repeated in each new expansion dataset, every background is random. With more and different backgrounds, the larger datasets take longer to train. This can also be seen in Figure 34(c)'s LLC. Though the peak level of training activity is higher and at nearly the same minibatch as S03, the tail of the larger datasets takes longer to settle to an RMS stasis.



(a) Accuracy and loss by epoch

(b) Accuracy and loss by minibatch

(c) LLC by minibatch

Figure 34. Dataset expansion with single pseudo-resolution

Fourth, related to the delayed LLC settling in Figure 34(c), larger datasets settle to a higher stasis training activity level. This is likely a direct result of the background variance preventing the CNN from learning. More augmented backgrounds mean more variance and larger stasis activity.

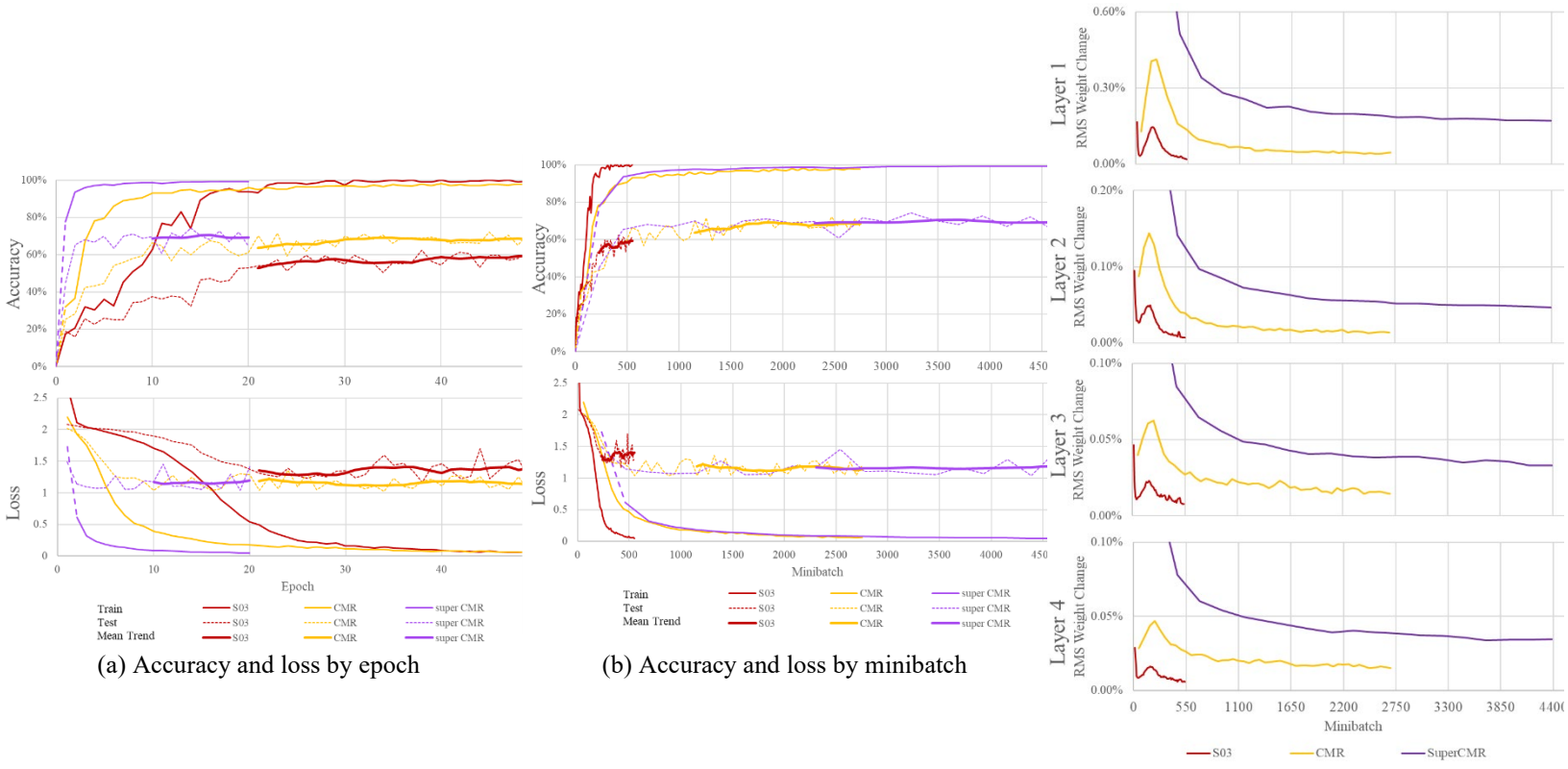
Fifth, larger datasets create higher peak training activity levels. There appears to be a correlation between higher peak training activity and higher test accuracy/lower test loss, but this trend does not hold for training accuracy or loss. As discussed in the first point, more variance does not improve training accuracy. The higher peaks are probably caused by the same background variance driving higher test accuracies.

Sixth, the peak level of training corresponds to the "shoulder" in the accuracy and loss charts. Larger datasets with more background variance do not delay the peak training significantly. This peak is also very consistent throughout the layers; later layers are not delayed while early layers train. The linkage between the peak training activity level and the shoulder in the accuracy and loss curves makes sense because lower training rates mean slower progress in accuracy and loss. The root cause of the peak and the corresponding shoulder is unclear.

b. Multiple Pseudo-Resolution Dataset Expansion

Figure 35 displays features of dataset expansion with the addition of multiple pseudo-resolutions. Figure 35(a) uses epochs and Figure 35(b) and Figure 35(c) use minibatches to compare training rates.

First, again, the test accuracy increases with increasing dataset size. Mean test accuracy again stratifies in Figure 35(a) based on dataset size. Again, training accuracy does not correlate to test accuracy. Larger datasets with more variance have lower final training accuracy despite higher test accuracy. The new observation that *even* larger datasets are able to continually improve test accuracy is significant and encouraging.



(a) Accuracy and loss by epoch

(b) Accuracy and loss by minibatch

(c) LLC by minibatch

Figure 35. Dataset expansion using multiple pseudo-resolutions

Second, larger datasets delay overfitting longer, as seen in the test loss of Figure 35(a). Delayed overfitting is again likely due to background variance, but resolution variation likely plays a compounding role. The delay likely still allows more training on object characteristics and, therefore, may still be the root cause of improved test accuracy.

Third is a differing observation from single pseudo-resolution. With training cycles in mind, Figure 35(b) shows that the rate of training accuracy is roughly the same in CMR and Super_CMR. While Figure 35(c)'s LCC again shows the tail of the larger dataset taking longer to settle to a higher stasis RMS, the two training accuracy and loss lines are nearly coincident. The reason for this similarity is unclear. Perhaps there is a limit to the training delays that can be induced by variance.

Fourth, again the larger dataset settles to a higher stasis training activity level. This is still likely a direct result of the background variance.

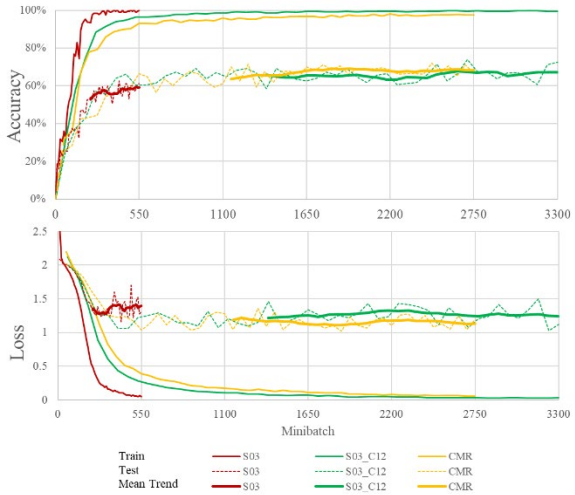
Fifth, larger datasets continue to create higher peak training activity levels. Again, the higher peaks are probably caused by the same background variance driving higher test accuracies.

Sixth, the peak level of training still corresponds to the shoulder in the accuracy and loss charts. Larger datasets still do not delay the peak training significantly, though the peak of Super_CMR occurs within the first epoch. These peaks are still very consistent throughout the layers. The root cause of the peak and the corresponding shoulder is still unclear.

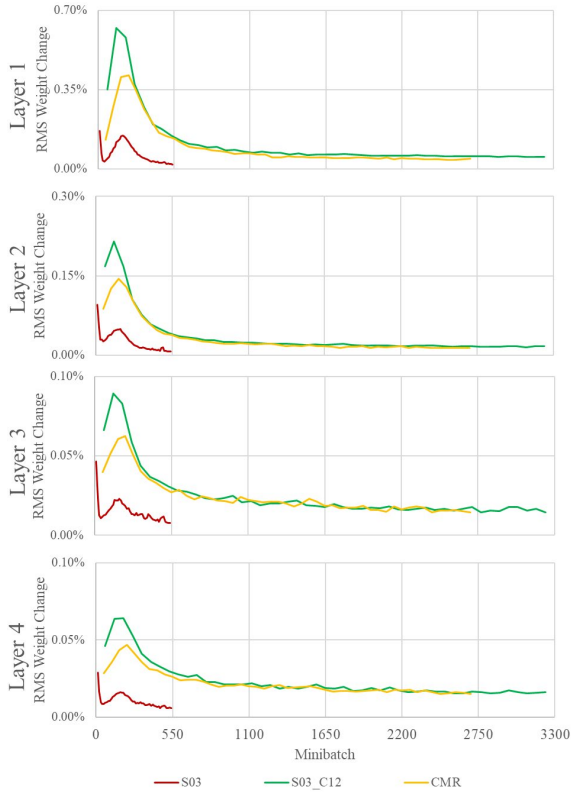
c. Similar Dataset Sizes with Different Compositions: S03_C12 and CMR

Several observations from the previous sections depend on background variance and dataset size. Figure 36, created by combining previous data, shows the effect of combining multiple pseudo-resolutions.

.



(a) Accuracy and loss by minibatch



(b) LLC by minibatch

Figure 36. Charts comparing S03_C12 and CMR, two datasets with approximately 6,000 images

First, looking in Figure 36(a), more pseudo-resolutions in CMR leads to slightly higher test accuracy/lower test loss despite a slightly smaller dataset. This provides good motivation for using multiple pseudo-resolutions in future applications of this data augmentation technique.

Second, looking now in Figure 36(b), the peak training activity for S03_C12 is noticeably higher. The reason for this is unclear. Perhaps the presence of more images with the same pseudo-resolution is encouraging more change.

Third, the peak of CMR is slightly delayed in Figure 36(b). Despite having slightly fewer images with background variance, CMR is showing a longer delay. This may be caused by additional variation from multiple pseudo-resolutions.

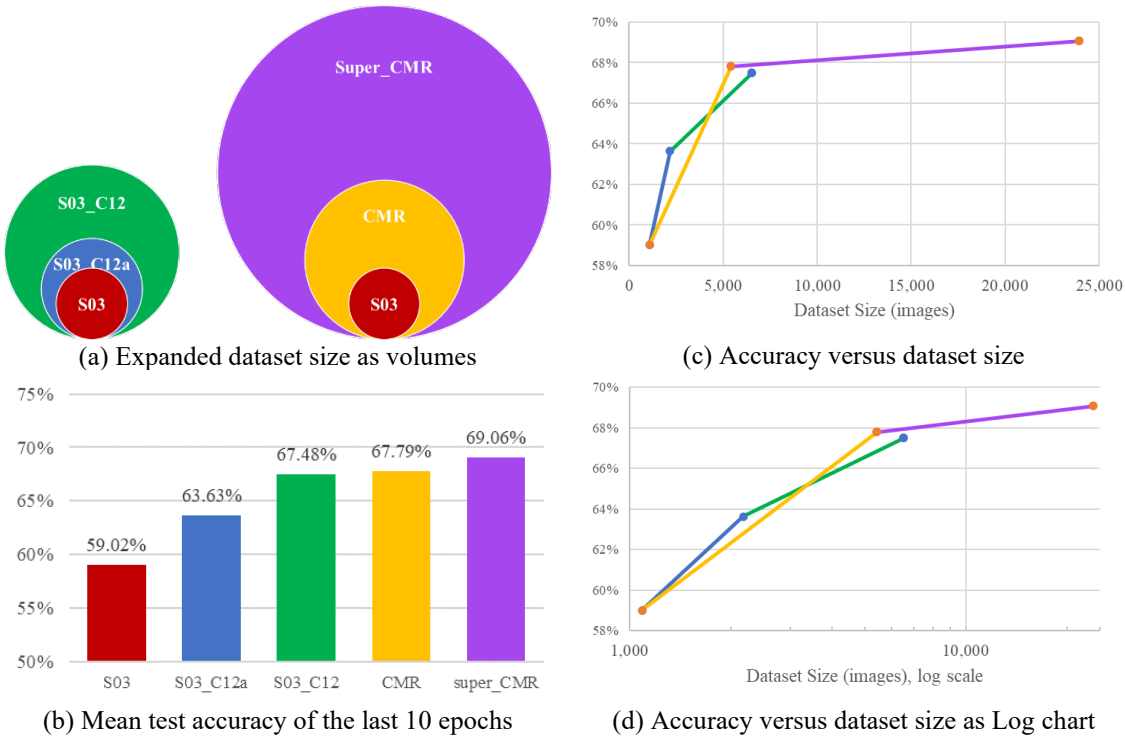
Fourth, the LLC tails of both S03_C12 and CMR settle at same rate and to same stasis. This is likely because they have roughly the same amount of random image backgrounds. Supporting this idea, CMR is a slightly smaller dataset and its stasis is correspondingly slightly lower

d. Accuracy Improvement with Dataset Expansion

The last set of results shows the significant accuracy improvement on measured SAR images provided by expanding datasets with the proposed data augmentation technique. Figure 37 shows the clear and positive effect on the average test accuracy from each of the expanded dataset trials. As seen in Figure 37(b), with Super_CMR the test accuracy of AConvNet increased 10.04 points (a 17% increase) over a model trained only on synthetic SAMPLE.

By plotting these accuracies against the number of images in the datasets, a curve can be approximated as seen in Figure 37(c) and Figure 37(d). No predictive model should be produced from such a small data sample, but trends can be observed. The most significant trend is the apparent existence of an effects limit, as seen in Figure 37(d). There are diminishing returns associated with ever-increasing the dataset size with only this two-stage data augmentation. Not only does accuracy stop increasing rapidly, but computational

demands grow. These factors should be understood and weighed when creating future datasets using the methods proposed in this thesis.



(a) Expanded dataset size as volumes

(c) Accuracy versus dataset size

(b) Mean test accuracy of the last 10 epochs

(d) Accuracy versus dataset size as Log chart

Figure 37. Comparison of mean test accuracy and dataset size

Finally, a review of the increasing test accuracy with increasing data augmentation supports the idea that segmentation artifacts are not significant in CNN training. If present, artifacts have not affected the CNN training in a way that detracts from testing against unaugmented, measured SAR images. If CNN training is reliant upon artifacts such as histogram levels, background texture within the segmentation mask, or identifiable outlines of segmentation areas, then the accuracy against the unaugmented images will decrease. This is clearly not the case. This positive finding encourages future use of the proposed data augmentation technique for synthesizing wide-area SAR imagery.

C. SUMMARY OF FINDINGS

This chapter has revealed several significant findings pertaining to the novel two-stage data augmentation technique proposed in this thesis. These findings and the sections in which they are discussed are summarized in the list below:

4. Coarse-resolution synthetic SAR representations might become more general by removing fine detail (IV.A.2.a, IV.B.1.c).
5. Clutter backgrounds make synthetic SAR imagery more robust and more suitable for use with measured SAR images due to increased coherent background variance (IV.A.2.b(1), IV.B.1.c).
6. Coarse-resolution synthetic imagery with clutter background at all pseudo-resolutions can train AConvNet just as well as the original-resolution synthetic SAMPLE. Conversely, the use of noise backgrounds create progressively worse training results with reduced resolution (IV.A.2.b(2)).
7. Targeted deep layer training is possible (IV.B.1.b).
8. Overfitting in SAR chips can be delayed indefinitely with high levels of coherent background variance (IV.B.1.b, IV.B.1.c).
9. Object regions may become overfit without indication when background regions have untrainably high variance (IV.B.1.b).
10. Expanding dataset size with single pseudo-resolutions increases test accuracy (IV.B.2.a).
11. Expanding dataset size with multiple pseudo-resolutions increases test accuracy more than with single pseudo-resolutions (IV.B.2.c).
12. Synthetic object features seem to impose an upper limit on test accuracy that background substitution alone cannot overcome (IV.B.2.d).
13. Two-stage data augmentation does not create artifacts significant to classification (IV.B.2.d).

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

SAR ATR is a critical capability that could be greatly enhanced with AI/ML object detection, but AI/ML networks require extensive training datasets that are not available for SAR. This thesis proposes a novel data augmentation technique to synthesize and expand wide-FOV, labeled SAR datasets appropriate for such training. Three key research questions are crafted to determine the validity of the proposed method. First, this section verifies AI/ML architecture installation by reproducing the results of previous studies. Next, this section summarizes the implications of this thesis's test results as they apply to the research questions and hypotheses. For a succinct review of test findings please see C. Summary of Findings in the previous chapter.

A. SUPPORTING PREVIOUS WORK

Results from [18] and [30] are supported during recreation of their SOC experiments. Additionally, the idea that the phase channel of SAR is significant [21] is reinforced by removing the phase channel from the original AConvNet dataset. This results in a 2% accuracy decrease when training and validating with measured SAR data. Most significantly, the idea that AI/ML training with SAR images can be enhanced by combining multiple lower resolutions [21], [24] is supported and expanded from SRC use to use with a simple CNN.

B. DEEP LAYER TRAINING

By using transfer learning as a mechanism to explore the influence of lower-pseudo-resolution SAR imagery on different layers of a CNN, it is shown that convolution and max pooling layers can be bypassed somewhat. The noticeable decrease in training activity with low pseudo-resolution data shows that deep layer training—the targeted injection of training data into deep layers—is plausible.

C. EXPANDING DATASET WITH DATA AUGMENTATION

The most significant result of this thesis is the 10 percentage point increase in test accuracy (17% improvement over initial) achieved by using the proposed data augmentation method for dataset expansion. Increasing the background variance by substituting in new coherent backgrounds delayed overfitting indefinitely. Training on the desired object became more robust. This novel, two-stage data augmentation technique significantly improved the capability of the synthetic SAMPLE dataset to train a CNN to detect measured SAR imagery.

D. POTENTIAL FOR WIDE-AREA SAR SYNTHESIS

Regarding the use of the two-stage data augmentation for synthesizing wide-area SAR imagery, there is a potential for detectable artifacts to be created during segmentation. Overall, the improved classification accuracy when testing against measured SAR images has shown that any artifacts—if present—are insignificant. The AConvNet CNN is not reliant upon artifacts for classification. With this finding, supported by the sustained test accuracy of the C12a dataset in isolation and the increasing test accuracy of the expanded datasets, there is good reason to continue investigations into wide-area SAR synthesis.

VI. RECOMMENDATIONS FOR FURTHER STUDY

A. USE MORE COMPLEX AI/ML ARCHITECTURES

This study uses AConvNet [18] precisely because of the combination of simplicity and documented performance against the MSTAR dataset [2]. But the original AConvNet is trained and validated on MSTAR alone. Many of its design considerations deal specifically with the small size of the MSTAR dataset. The behavior of AConvNet may not be representative of the behavior of more complex networks. Unfortunately, without labeled SAR swath imagery to test a synthetic-swath-trained object detector, chip testing is the only suitable substitute. Further research with more advanced classification CNNs and single-stage object detectors is warranted.

B. FINISH THE WIDE-AREA DATA SYNTHESIS TEST

The code is already written that can overlay SAMPLE objects (synthetic or measured) onto wide-FOV backgrounds. The remaining hurdle is testing. How can a single-stage object detector that was trained with these images be tested? One option is to train the architecture and then forward the model weights to a different domain where appropriate data might be available. Of course, that option comes with many complications and roadblocks.

An alternative method might involve modifying the representative size of overlayed objects and determining how different they must be for the object detector to differentiate them. If future researchers' goal is to find objects of a certain size, then a reasonable question would be: "How well does the single-stage object detector do at discriminating objects of different sizes?" A simple scaling factor could change the sizes programmatically before alpha-blending. A confusion matrix could show the result if different scaled objects were given different class names accordingly.

C. INCORPORATE GEOMETRIC TRANSLATIONS, ESPECIALLY AT LOW RESOLUTION

The workflow studied herein synthesizes a large, robust dataset of SAR images. This expansion could be continued and enhanced by layering other data augmentation methods. Shifting the original images one or two pixels left/right or up/down before downsampling would add translational augmentation. The pixel intensities of the low-resolution objects would be varied while maintaining their form. Because the objects become so coarse at low pseudo-resolution, their form is mostly immutable. Applying very slight translation prior to downsampling would allow further augmentation and growth of the low-resolution dataset. This type of translational augmentation could fundamentally change the representation of SAMPLE objects by adding needed variation to the objects themselves. It might even make the objects more general and better suited for training.

D. USE MORE PRECISE SEGMENTATION TECHNIQUES

A more precise segmentation mask would give more reliable overlay at native resolutions. SARBake [31] is very promising but their code has broken dependencies. Other alternatives could also be explored.

E. EXPLORE ALPHA BLENDING WITH PHASE DATA

It is unclear whether phase data can be segmented and alpha blended in the image domain, but it is highly unlikely. The general idea could be used in the spatial frequency domain (also known as phase history) instead. If successful, that work would be a substantial improvement in SAR synthesis in its own right.

F. LOWER RESOLUTION DETECTION

Because most wide-area SAR images are not collected at 0.3m resolution, the hope is that an object detector trained with multiple synthetic resolutions will be able to function at any resolution *as good or better than* the lowest training pseudo-resolution. This could be tested by a using lower resolution test dataset or a mixed resolution test dataset and checking for confusion.

G. EXPLORE THE LIMITS OF DATASET EXPANSION

Researchers cannot seem to pin down a good guideline for dataset sizing—there are just too many variables. With the infinite capacity for background substitution, this thesis's data augmentation technique could be used to explore this question.

H. EXPLORE THE LIMIT OF DELAYED TRAINING

Delayed training due to variance was critical in enhancing test accuracy in this study. These delays are made visible with the LLC, a novel method of viewing training activity. With this new charting method in hand, methods of delaying training could be explored. Perhaps there is a limit to training delays that would explain why CMR and Super_CMR have the same delay despite one being 5 times larger than the other.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] C. V. J. Jakowatz, D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spoltlight-Mode Synthetic Aperture RADAR: A Signals Processing Approach*, 1 ed., Albuquerque, NM, USA: Springer, 1996.

[2] Sandia National Laboratory, "MSTAR overview," Accessed Oct. 1, 2021 [Online] Available: https://www.sdms.afrl.af.mil/index.php?collection=mstar

[3] B. Lewis, "SAMPLE_dataset_public," 27 September 2019. Accessed Oct. 1, 2021 [Online] Available: https://github.com/benjaminlewis-afrl/SAMPLE_dataset_public

[4] O. Kechagias-Stamatis and N. Aouf, "Automatic target recognition on synthetic aperture radar imagery: A survey," *IEEE A&E Systems Magazine,* pp. 56–81, March 2021.

[5] E. Blasch, U. Majumder, E. Zelnio, and V. Velten, "Review of recent advances in AI/ML using the MSTAR data," *Proceedings of SPIE,* vol. 11393, pp. 113930C-113930C-11, May 19, 2020.

[6] E. Zelnio and A. Pavy, "Open set SAR target classification," in *Proceedings Volume 10987, Algorithms for Synthetic Aperture Radar Imagery XXVI*, Baltimore, 2019.

[7] N. Fatih, O. E. Okman, A. Özgür, and M. Çetin, "RmSAT-CFAR: Fast and accurate target detection in radar images," in *SoftwareX 8*, 2017.

[8] C. Shorten and T. M. Khoshgoftaar, "A survey on image data aumentation for deep learning," *Journal of Big Data,* vol. 6, no. 60, pp. 1–48, 2019.

[9] U. Majumder, E. Blasch, and D. Garren, *Deep Learning for Radar and Communications Automatic Target Recognition*, Boston, MA, USA: Artech, 2020.

[10] B. Lewis *et al.*, "A SAR dataset for ATR development: the synthetic and measured paired labeled experiment (SAMPLE)," *Proceedings of SPIE,* vol. 10987, p. 109870H–109870H–16, 2019.

[11] T. Scarnati and B. Lewis, "A deep learning approach to the Synthetic and Measured Paired and Labeled Experiment (SAMPLE) challenge problem," in *Proceedings*

Volume 10987, Algorithms for Synthetic Aperture Radar Imagery XXVI, Baltimore, MD, USA, 2019.

[12] B. Lewis, K. Cai, and C. Bullard, "Adversarial training on SAR images," in *Proceedings Volume 11394, Automatic Target Recognition XXX*, 2020.

[13] D. Massonnet and J.-C. Souyris, *Imaging with Synthetic Aperture Radar*, 1 ed., New York, NY, USA: EPFL Press, 2008

[14] J. Berbraeken, M. Wolting, J. Katzy, J. Koppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Computing Surveys,* vol. 53, no. 2, pp. 30:1-30:33, March 2020.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection [YOLOv1]," University of Washington, Allen Institude for AI, Facebook AI Research, Seattle, WA, USA, 2016.

[16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," University of Washington, Seattle, WA, USA, 2018.

[17] G. Jocher, "YOLOv5," Ultralytics, May 18, 2020 [Online] Available: https://github.com/ultralytics/yolov5

[18] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target cassification using the deep convolutional networks for SAR images," *IEEE transactions on geoscience and remote sensing,* vol. 54, no. 8, pp. 4806–4817, 2016. Available: doi: 10.1109/TGRS.2016.2551720

[19] X. Jiang, A. Hadid, Y. Pang, E. Granger, and X. Feng, Eds., *Deep Learning in Object Detection and Recognition*, Singapore, Rep. of Signapore: Springer, 2019. Available: doi: 10.1007/978-981-10-5152-4

[20] S. Papson and R. M. Narayanan, "Classification via the shadow region in SAR imagery," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 48, no. 2, pp. 969–980, 2012.

[21] K. El-Darymli, P. McGuire, E. W. Gill, D. Power, and C. Moloney, "Characterization and statistical modeling of phase in single-channel synthetic aperture radar imagery," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 51, no. 3, pp. 2071–2092, 2015.

[22] Y. Yan, "Convolutional neural networks based on augmented training samples for synthetic aperture radar target recognition," *Journal of Electronic Imaging,* vol. 27, no. 2, p. 023024–023024, 2018.

[23] B. Ding and G. Wen, "Target recognition of SAR images based on multiresolution representation," *Remote Sensing Letters,* vol. 8, no. 11, pp. 1006–1014, 2017.

[24] X. Tan, M. Zou, and X. He, "Target recognition in SAR images based on multiresolution representations with 2D canonical correlation analysis," *Hindawi Scientific Programming,* vol. 2020, pp. 1–9, 2020.

[25] R. Schumacher and J. Schiller, "Non-cooperative target identification of battlefield targets – classification results based on SAR images," *IEEE International Radar Conference,* vol. 2005, p. 167–172, 2005.

[26] Sandia National Laboratory, "MSTAR public clutter," Sep. 1995. Accessed Oct. 1, 2021 [Online] Available: https://www.sdms.afrl.af.mil/index.php ?collection=mstar&page=clutter

[27] L. M. Novak, G. J. Owirka, and C. M. Netishen, "Performance of a high-resolution polarimetric SAR automatic target recognition system," *The Lincoln Laboratory Journal,* vol. 6, no. 1, pp. 11–24, 1993.

[28] G. Jocher, "YOLOv5 image augmentation," Ultralytics, Nov. 2020. Accessed Dec. 27, 2022 [Online] Available: https://docs.ultralytics.com/FAQ/augmentation/

[29] K. Nyuytiymbiy, "Parameters and hyperparameters in machine learning and deep learning," Medium, 30 December 2020. Accessed Dec. 22, 2022 [Online] Available: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac

[30] J. Park, "AConvNet-pytorch," https://jangsoopark.github.io/, 29 March 2021. Accessed Feb. 08, 2022 [Online] Available: https://github.com/jangsoopark/ AConvNet-pytorch/

[31] D. Malmgren-Hansen and M. Nobel-Jøgensen, "Convolutional neural networks for SAR image segmentation," in *IEEE International Symposium on Signal Processing and Information Technology*, 2015.

[32] L. Jianzhuang, L. Weqing, and T. Yupeng, “Automatic thresholding of gray-level pictures using two-dImensional Otsu method,” in *International Conference on Circuits and Systems*, Xian, China, 1991.

[33] Y. Xiao and Z.-G. Cao, “New entropic thresholding approach using gray-level spatial correlation histogram,” *Optical Engineering,* vol. 49, no. 12, December 2010.

[34] E. Aitnouri, S. Wang, and D. Ziou, “Segmentation of small vehicle targets in SAR images,” *Proceedings of SPIE,* vol. 4726, pp. 35–45, July 25, 2002.

[35] N. Otsu, “A threshold selection method from gray-level historgram,” *IEEE Transationcs on Systems, Man, and Cybernetics,* vol. 9, no. 1, pp. 62–66, January 1979.

[36] T. Porter and T. Duff, “Compositing digital images,” *Computer Graphics,* vol. 18, no. 3, pp. 253–259, 1984.

[37] A. Sharma, “Resizing images using various interpolation techniques,” Medium, Jun. 10, 2021. Accessed Oct. 01, 2021 [Online] Available: https://annmay10.medium.com/resizing-images-using-various-interpolation-techniques-4b99800999f2

[38] OpenCV, “Geometric image transformations,” Dec. 25, 2021. Accessed Jan. 02, 2022 [Online] Available: https://docs.opencv.org/4.5.5/

[39] OpenCV, “Adding (blending) two images using OpenCV,” Jan. 8, 2013. Accessed Dec. 7, 2022 [Online] Available: https://docs.opencv.org/3.4/d5/dc4/tutorial_adding_images.html

[40] J. Browniee, “What is the difference between test and validation datasets?,” Aug. 14, 2020. Accessed Jan. 12, 2023 [Online] Available: https://machinelearningmastery.com/difference-test-validation-datasets/

[41] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” *Advances in Neural Information Processing Systems,* vol. 32, pp. 8024–8035, 2019. Available: doi: 10.48550/arxiv.1912.01703

[42] Weights & Biases, “Weights & Biases,” Accessed Nov. 28, 2022 [Online] Available: https://docs.wandb.ai/

[43] D. E. Dudgeon and R. T. Lacoss, "An overview of automatic target recognition," *The Lincoln Laboratory Journal,* vol. 6, no. 1, pp. 3–10, 1993.

[44] S. Saxena, "Precision vs. recall," Medium, May 11, 2018. Accessed Nov. 1, 2022 [Online] Available: https://medium.com/@shrutisaxena0617/precision-vs-recall-386cf9f89488

[45] J. Solawetz, "How to train YOLOv5 on a custom dataset," Roboflow, June 10, 2020 [Online] Available: https://blog.roboflow.com/how-to-train-yolov5-on-a-custom-dataset/

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California