



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**CONVOLUTIONAL NEURAL NETWORKS FOR FEATURE
EXTRACTION AND AUTOMATED TARGET RECOGNITION
IN SYNTHETIC APERTURE RADAR IMAGES**

by

John E. Geldmacher

June 2020

Thesis Advisor:

Co-Advisor:

Second Reader:

Ying Zhao

Walter A. Kendall

Christopher Yerkes,

National Intelligence University

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2020	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE CONVOLUTIONAL NEURAL NETWORKS FOR FEATURE EXTRACTION AND AUTOMATED TARGET RECOGNITION IN SYNTHETIC APERTURE RADAR IMAGES			5. FUNDING NUMBERS	
6. AUTHOR(S) John E. Geldmacher				
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) Advances in the development of deep neural networks and other machine learning (ML) algorithms, combined with ever more powerful hardware and the huge amount of data available on the internet, has led to a revolution in ML research and applications. These advances have massive potential for military applications at the tactical level, particularly in improving situational awareness and speeding kill chains. One opportunity for the application of ML to an existing problem set in the military is in the analysis of Synthetic Aperture Radar (SAR) imagery. Synthetic Aperture Radar imagery is a useful tool for imagery analysts because it is capable of capturing high-resolution images at night and regardless of cloud coverage. There is, however, a limited amount of publicly available SAR data to train a machine learning model. This thesis seeks to demonstrate that transfer learning from a convolutional neural network trained on the ImageNet dataset is effective when retrained on SAR images. It then compares the performance of the neural network to shallow classifiers trained on features extracted from images passed through the neural network. This thesis shows that cross-modality transfer learning from features learned on photographs to SAR images is effective and that shallow classification techniques show improved performance over the baseline neural network in noisy conditions and as training data is reduced.				
14. SUBJECT TERMS machine learning, artificial intelligence, imagery analysis, deep learning, transfer learning, synthetic aperture radar, convolutional neural networks, Synthetic Aperture Radar, SAR			15. NUMBER OF PAGES 67	
			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	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**CONVOLUTIONAL NEURAL NETWORKS FOR FEATURE EXTRACTION
AND AUTOMATED TARGET RECOGNITION IN SYNTHETIC APERTURE
RADAR IMAGES**

John E. Geldmacher
Captain, United States Marine Corps
BS, U.S. Naval Academy, 2012

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN INFORMATION WARFARE
SYSTEMS ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL
June 2020**

Approved by: Ying Zhao
Advisor

Walter A. Kendall
Co-Advisor

Christopher Yerkes
Second Reader

Thomas J. Housel
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Advances in the development of deep neural networks and other machine learning (ML) algorithms, combined with ever more powerful hardware and the huge amount of data available on the internet, has led to a revolution in ML research and applications. These advances have massive potential for military applications at the tactical level, particularly in improving situational awareness and speeding kill chains. One opportunity for the application of ML to an existing problem set in the military is in the analysis of Synthetic Aperture Radar (SAR) imagery. Synthetic Aperture Radar imagery is a useful tool for imagery analysts because it is capable of capturing high-resolution images at night and regardless of cloud coverage. There is, however, a limited amount of publicly available SAR data to train a machine learning model. This thesis seeks to demonstrate that transfer learning from a convolutional neural network trained on the ImageNet dataset is effective when retrained on SAR images. It then compares the performance of the neural network to shallow classifiers trained on features extracted from images passed through the neural network. This thesis shows that cross-modality transfer learning from features learned on photographs to SAR images is effective and that shallow classification techniques show improved performance over the baseline neural network in noisy conditions and as training data is reduced.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	PROBLEM STATEMENT	1
B.	PURPOSE AND RESEARCH QUESTIONS.....	2
C.	THESIS ORGANIZATION	2
II.	TECHNICAL BACKGROUND	5
A.	SYNTHETIC APERTURE RADAR	5
B.	FUNDAMENTALS OF NEURAL NETWORKS.....	7
C.	CONVOLUTIONAL AND POOLING LAYERS.....	10
D.	TRANSFER LEARNING	11
E.	IMAGENET AND ILSVRC.....	12
F.	SHALLOW CLASSIFIERS.....	13
1.	K-Nearest Neighbor	13
2.	Support Vector Machines.....	14
3.	Random Forests	16
G.	RELATED RESEARCH.....	17
III.	METHODOLOGY	21
A.	MSTAR DATASET	21
B.	NETWORK ARCHITECTURE.....	23
C.	TENSORFLOW, KERAS API, AND ORANGE TOOLKIT	25
D.	MEASURES OF PERFORMANCE AND EXPERIMENTAL SETUPS	26
1.	Measures of Performance.....	26
2.	Experiment 1	27
3.	Experiment 2	27
4.	Experiment 3	28
5.	Experiment 4	29
IV.	RESULTS AND ANALYSIS	31
A.	EXPERIMENTAL RESULTS.....	31
1.	Experiment 1	31
2.	Experiment 2	32
3.	Experiment 3	34
4.	Experiment 4	37
B.	ANALYSIS	38

V.	CONCLUSIONS AND FUTURE WORK	41
A.	CONCLUSIONS	41
B.	FUTURE WORK	43
	LIST OF REFERENCES	45
	INITIAL DISTRIBUTION LIST	49

LIST OF FIGURES

Figure 1.	Comparison of EO and SAR Overhead Images. Adapted from Google (n.d.) and Sandia National Laboratories (n.d.).....	6
Figure 2.	Grumman HU-16B Albatross. Source: U.S. Air Force (n.d.).....	7
Figure 3.	The Artificial Neuron. Adapted from Nielsen (2015).	8
Figure 4.	A Basic Neural Network. Source: Nielsen (2015).....	9
Figure 5.	The Convolution Operation. Source: Stewart (2019).	10
Figure 6.	Example Hierarchy of Synsets. Source: Deng (2009).	12
Figure 7.	Example of kNN. Adapted from Theobald (2017).	14
Figure 8.	Example SVM Classifier. Adapted from Theobald (2017).	15
Figure 9.	Non-Linearly Separable Data Transformed into Three-Dimensions. Adapted from Theobald (2017).	15
Figure 10.	Example of Random Forest. Source: Yiu (2019).....	16
Figure 11.	CNN to SVM Methodology. Source: Mufti (2018).....	19
Figure 12.	Example Photographs and MSTAR Images by Class. Adapted from U.S. Air Force (1996), Leahy (1994), and Torin (2008).	22
Figure 13.	VGG16 Architecture. Adapted from Simonyan and Zisserman (2015).....	24
Figure 14.	Modified VGG16 with Partial Transfer Learning and 11-Class Output. Adapted from Rice (2018).	24
Figure 15.	Modified VGG16 with Full Transfer Learning and 11-Class Output. Adapted from Rice (2018).	25
Figure 16.	Multistep Classifier Using a CNN for Feature Extraction. Adapted from Rice (2018).....	25
Figure 17.	Orange Workflow	28
Figure 18.	Comparison of Original Image and 20% Noise Added Image	29
Figure 19.	Comparison of Training Loss and Accuracy	32
Figure 20.	Example Filters from First Convolutional Layer.....	39

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	Number of SAR Images per Class.....	23
Table 2.	Comparison of CNN Performance.....	31
Table 3.	Comparison of Multistep Classifier without Transfer Learning.....	33
Table 4.	Comparison of Multistep Classifier with Partial Transfer Learning	33
Table 5.	Comparison of Multistep Classifier with Full Transfer Learning	34
Table 6.	Comparison of CNN Performance on Low-Noise Images without Retraining.....	35
Table 7.	Comparison of CNN Performance on High-Noise Images without Retraining.....	35
Table 8.	Comparison of Multistep Classifier without Transfer Learning Trained on 20% Noisy Images.....	36
Table 9.	Comparison of Multistep Classifier with Partial Transfer Learning Trained on 20% Noisy Images.....	36
Table 10.	Comparison of Multistep Classifier with Full Transfer Learning Trained on 20% Noisy Images.....	37
Table 11.	Performance of Multistep Classifier with Partial Transfer Learning Trained on 100 Images per Class.....	38
Table 12.	Performance of Multistep Classifier with Partial Transfer Learning Trained on 50 Images per Class.....	38

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

API	application program interface
ATR	Automated Target Recognition
CNN	convolutional neural network
EO	electro-optical
FSU	former Soviet Union
GEOINT	geospatial intelligence
ILSVRC	ImageNet Large Scale Visualization and Recognition Challenge
IMINT	imagery intelligence
IR	infrared
kNN	k-nearest neighbor
ML	machine learning
MSTAR	Moving and Stationary Target Acquisition and Recognition
NGA	National Geospatial-Intelligence Agency
ReLU	rectified linear unit
RF	random forests
RGB	red, green, blue
SAIP	Semi-Automated IMINT Processing
SAR	synthetic aperture radar
SLICY	Sandia Laboratories implementation of cylinders
SVM	support vector machine

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank the people without whom I could not have succeeded during my time at the Naval Postgraduate School. My thanks to Dr. Ying Zhao and Mr. Tony Kendall for working with me as my thesis advisors. They saw potential in my idea and guided me in my research to create this thesis. I'd also like to thank Chris Yerkes for his patience in dealing with my sometimes basic questions about coding and for his inputs in the writing and editing process. I'd also like to thank Michael McCarrin for his assistance in beginning this research as the final project for his Computer Vision class. Thank you also to the reader of this work. I hope it aids you in your own research or in the development of the work below into a fieldable system.

I am also thankful for the cohort with which I have had the pleasure of taking so many classes over the past two years. The comradery and commiseration in the Trident Room and in town will remain my fondest memories of my time in Monterey.

Most importantly, thank you to my wife who patiently listened to me as I babbled about class assignments, technical challenges, and writer's block. Talking through an idea helps me to focus my thoughts in a manner that makes them explainable, even if the person I am talking to has eyes that long ago glazed over in boredom.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PROBLEM STATEMENT

The analysis and classification of targets within imagery captured by aerial and space-based systems provides the Intelligence Community and military geospatial intelligence (GEOINT) personnel with important insights into adversary force dispositions and intentions. Satellite imagery has also entered the mainstream thanks to openly available tools like Google Earth. The high resolution of space-based sensors and common use of overhead imagery in everyday life means that, with the exception of decoys and camouflage, an average person is now reasonably capable of identifying objects in electro-optical (EO) imagery. EO images are, however, limited by cloud coverage and daylight. About half of the time when a satellite in low earth orbit can image a target it is night, necessitating the use of either an infrared (IR) or a synthetic aperture radar (SAR) sensor. SAR has the additional advantage over EO/IR sensors of being all-weather. It is not obscured by cloud, dust or smoke cover that can render EO/IR systems ineffective.

Although a trained analyst is needed to create imagery intelligence (IMINT) products for targeting purposes or to discriminate high fidelity decoys, an untrained person is generally able to identify objects in EO imagery. This is not the case for both IR and SAR images. Even basic analysis of SAR images requires a trained imagery analyst. Scanning an entire swath of SAR imagery is a repetitive and time consuming task that currently requires human expertise, but importantly not creativity. This makes it an ideal problem for machine learning.

Automated target recognition (ATR) seeks to reduce the total workload of analysts so that their effort can be spent on the more human-centric tasks like presenting and explaining intelligence to a decision maker. ATR is also intended to reduce the time from collection to exploitation by screening images at machine speeds rather than manually. SAR ATR is complicated by the available data to train and assess machine learning models. Unlike other image classification tasks that are studied today, such as those to support self-driving vehicles, there is not a large and freely available amount of training data for

researchers. The paucity of training data requires creative solutions to achieve acceptable model performance, particularly with respect to the level needed to support a targeting decision by a military commander.

B. PURPOSE AND RESEARCH QUESTIONS

The primary goal of this thesis is to demonstrate that a transfer learning approach using a model pre-trained on photographs can be applied to achieve high precision and recall rates on Synthetic Aperture Radar (SAR) images of tactically meaningful targets. This thesis will also explore the utility of a two-step classification method where a convolutional neural network is used to extract features from test images for classification by shallow classifiers. Shallow classifiers are machine learning algorithms that do not rely on layers of artificial neurons, rather they use statistical or spatial methods to do regression or assign labels. Classification performance of this multistep method will be compared to the base CNN from which features were extracted as the training dataset is reduced and noise is added to the dataset. This thesis will therefore address the following research questions:

1. Can transfer learning from a model pre-trained on photographs be effective for SAR images?
2. Does using a neural network as a feature extractor for input into a shallow classification algorithm improve model performance?
3. What shallow classification technique provides the greatest benefit to classification accuracy?
4. Does a shallow classification method trained on features extracted from a neural network perform better in noisy conditions or with a small training dataset?

C. THESIS ORGANIZATION

Chapter II discusses the technical background of convolutional neural networks, transfer learning and previous work on SAR ATR. Chapter III describes the MSTAR

dataset, the structure of the neural network, training of the network and how intermediate features were extracted from the model prior to classification. Chapter IV compares the performance of transfer learning approaches and the performance of neural networks to shallow classifiers trained on extracted features. Chapter V looks at how the results address the research questions and areas of future research that could be conducted to further this project.

THIS PAGE INTENTIONALLY LEFT BLANK

II. TECHNICAL BACKGROUND

A. SYNTHETIC APERTURE RADAR

Synthetic Aperture Radar (SAR) is a radar mounted to a moving platform that uses the platform's motion to approximate the effect of a large antenna. The high resolution that can be achieved by creating a radar with an effective aperture much greater in size than is physically possible allows for radar returns to be processed into images similar to what can be achieved with a camera (Skolnik, 1981). SAR imagery provides an important tool for the United States Intelligence Community and military IMINT analysts because of its all-weather, day/night collection capability. Additionally, some wavelengths that SAR imaging systems operate in have a degree of foliage and ground penetrating capability allowing for the detection of buried objects or objects under tree cover that would not be observable by electro-optical sensors.

These important advantages of SAR imaging for IMINT analysts do come with some significant drawbacks inherent to SAR images. Because SAR images are not true optical images they are susceptible to noise generated by constructive and destructive interference between radar reflections that appear as bright or dark spots called "speckle" in the image (Skolnik, 1981). Also, various materials and geometries will reflect the radar pulses differently creating blobs or blurs that can obscure the objects physical dimensions. SAR sensors operate in three primary modes, strip map, spotlight and inverse SAR. In strip map mode the SAR sensor is fixed and collects a swath of radar returns to be processed into an image. While operating in spotlight mode the physical antenna dwells on the desired target as the sensor moves through its path. The longer dwell time on the target allows for higher resolution as well as more average speckle (Skolnik, 1981). The different operational modes of the SAR sensors can change the way a target is represented in the formed SAR image. These issues, as well as problems caused by Doppler shift in moving objects and radar shadows, make the identification and classification of objects in SAR images a difficult and tedious task requiring a well-trained and experienced analyst. Figure 1 demonstrates the difficulties an imagery analyst would face when identifying targets in SAR imagery. The vehicles that are easily recognizable as cars in the EO image

become blurs in SAR. The static display aircraft are also difficult to identify. The wing of the pictured aircraft is only distinguishable by the radar shadow it casts, and the dimensions of the helicopters also become difficult to determine. Figure 2 is a photograph of the same model of aircraft shown in Figure 1 and is representative of the type of photographs that make up the ImageNet database. The ImageNet dataset and transfer learning will be discussed in further detail later in this chapter, but it is clear that the type of features learned on photographs, even of the same equipment as those in SAR images, will be significantly different.

Figure 1. Comparison of EO and SAR Overhead Images. Adapted from Google (n.d.) and Sandia National Laboratories (n.d.).



Pictured are static display aircraft at Kirtland Air Force Base, New Mexico. This comparison demonstrates the way objects in SAR imagery are not always easily identifiable by laymen and require a trained analyst to identify targets.

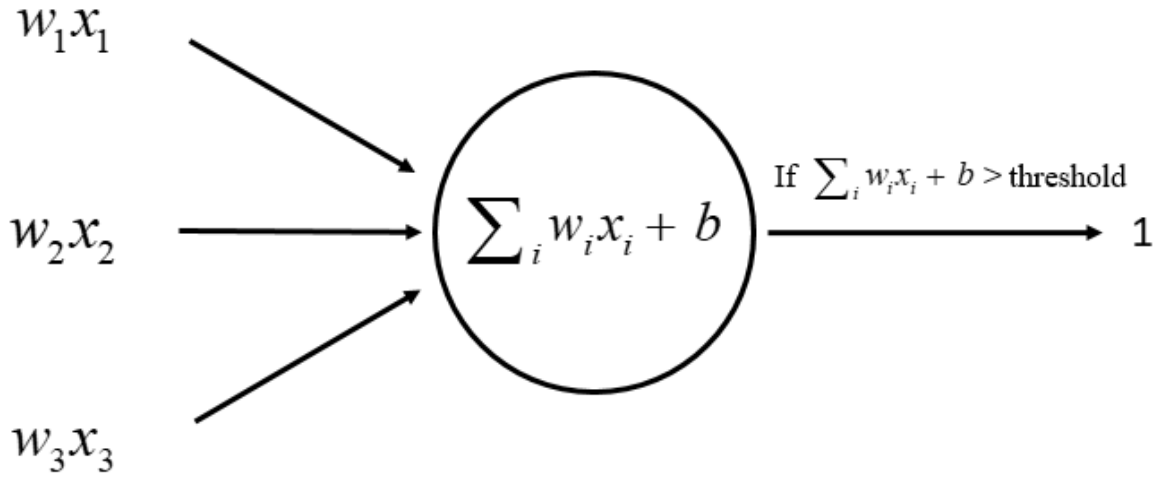
Figure 2. Grumman HU-16B Albatross. Source: U.S. Air Force (n.d.).



B. FUNDAMENTALS OF NEURAL NETWORKS

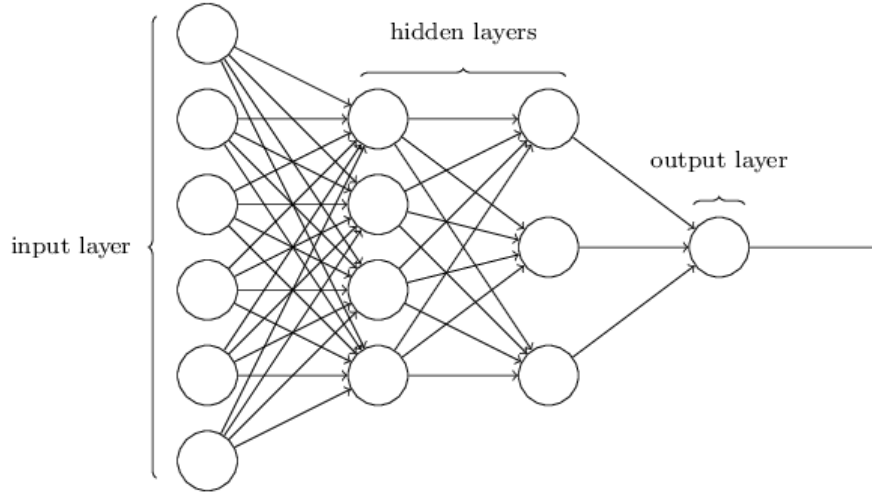
A neural network is, at its most basic, a function that learns to take some input and map it to an output. In image classification this means taking the red, green, blue (RGB) values of each pixel as an input value, performing a function and selecting the strongest output as the most likely class. A neural network is made up of artificial neurons, that similarly to a biological neuron are stimulated by an input and stimulate other neurons to create a response. This biological analogy does not truly hold up on closer inspection, but it provides a framework for thinking about how a neural network behaves. The operation of the most basic form of artificial neuron is shown in Figure 3. The neuron sums weighted inputs then applies an overall bias. The summed inputs and bias are compared to a set threshold value. If the output exceeds the threshold, the output is defined as 1, if it is less than the threshold it is defined as 0 (Nielsen, 2015). This creates a model for making a very simple decision where inputs that are more heavily weighted have greater impact on a binary true or false decision.

Figure 3. The Artificial Neuron. Adapted from Nielsen (2015).



Linking many of these neurons together into layers creates a neural network. To return to the biological model the neurons are linked to each other as in the brain and activations of neurons in turn activate other neurons. One of the common ways of describing the middle layers of a neural network is by calling them hidden layers since the user often has little insight into what the network is doing in between the input and the output that the network provides. Figure 4 depicts a basic 4-layer neural network with 2 hidden layers. The layers in the example neural network are fully connected layers where every neuron in a layer is connected to every other neuron in the next layer. Modern image classification techniques make use of convolutional layers that only connect one portion of a layer to the next. Convolutional layers and convolutional neural networks (CNN) made up of these layers are discussed in the next section.

Figure 4. A Basic Neural Network. Source: Nielsen (2015).



The response of individual neurons in a neural network can be made more precise by applying a more complex activation function than a simple threshold. Rather than a binary on/off response the neuron will activate with a strength based on the activation function. This allows for activations of some neurons to have a greater impact on the neural networks final output if they are more strongly activated than other neurons. The rectified linear unit (ReLU) activation function is a commonly used activation function since it removes negative activations, is computationally cheap, and simplified training by reducing the need for pre-training (Glorot et al., 2011). Equation (1) shows an input/output equation for the rectifier used in the ReLU activation.

$$f(x) = \max(0, x) \quad (1)$$

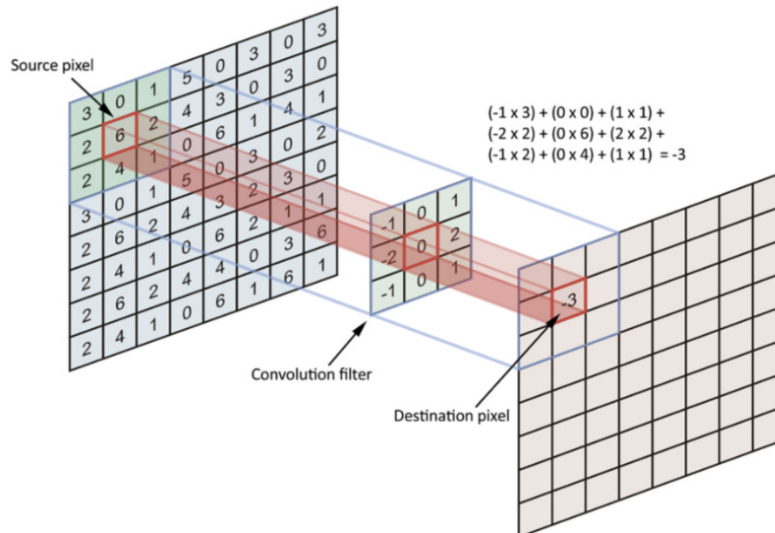
In the output layer two common activations are the sigmoid function and the softmax function. The sigmoid function compresses all outputs to a range between zero and one, allowing for a perceptron's output to be expressed as a probability of belonging to a class when making a binary classification. The softmax function is a generalized form of the sigmoid and allows for a probabilistic assignment of a label when there are more than two classes. The equation for the sigmoid is given in Equation (2).

$$\sigma(x) = \frac{1}{1 + e^{-\sum_i w_i x_i - b}} \quad (2)$$

C. CONVOLUTIONAL AND POOLING LAYERS

One of the most powerful image classification tools to emerge in recent years is the CNN. Convolutional layers provide several advantages over fully connected layers such as shift invariance, and reduced parameterization. In a convolutional layer a kernel of a given height, width, and depth is convolved with an input and the output of that function becomes the input to the next layer. The kernel is then shifted by a chosen stride length and the process is repeated. The height and width of the kernel act as the window while the kernel depth represents the number of filters within the kernel (Stewart, 2019). An example of how the convolution function using a simple 3x3x1 kernel is applied to a region is shown in Figure 5. The convolutional filters provide a means of efficiently describing the local neighborhood of a pixel and can identify meaningful features in an image. These filters are convolved throughout the image slowing them to identify features wherever they appear in an image. Moving the filters across the image and identifying features within a small window allows CNNs to be shift invariant. This is useful since one cannot assume that an object will appear in the same location within an image every time it imaged.

Figure 5. The Convolution Operation. Source: Stewart (2019).



Another advantage of convolutional layers is the reduced parameterization required to train the convolutional filters. Since even a relatively small 128x128 pixel RGB image requires a network with 49,152 inputs, if each pixel input is connected to every neuron in the next layer and if each subsequent hidden layer is also fully connected the number of parameters would quickly grow to the point of computational infeasibility. The use of a convolutional layer limits the number of weights that need to be saved and updated with each training iteration to the number of parameters within the kernel multiplied by the depth of the previous layer. Additionally, local connectivity between layers means that a single strong activation from a neuron is not propagated across the entire next layer as would occur with a fully connected layer.

Convolutional layers are often followed by pooling layers as a further means to reduce the feature space of each layer (Stewart, 2019). Two primary pooling techniques are max pooling and average pooling. In max pooling a pooling filter of a given size is stepped through the layer and the greatest value within the filter becomes the output. In average pooling the average of the values within the filter are the output. Pooling layers work to create denser representations of the previous layers and reduce the computational requirements for the CNN.

D. TRANSFER LEARNING

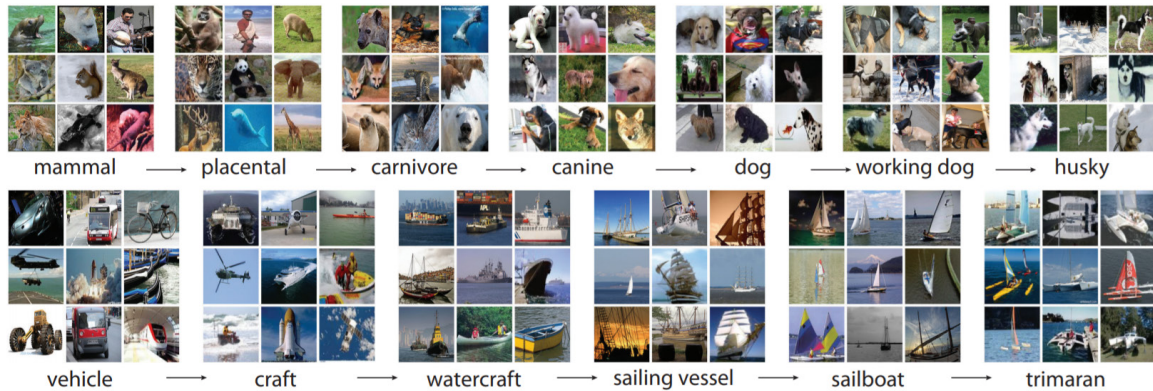
CNNs require a very large amount of data to train an accurate model and it is not uncommon for datasets with tens or even hundreds of thousands of images to be needed. Transfer learning presents one possible solution when training a CNN on a limited dataset by leveraging knowledge from a previously learned source task to aid in learning a new target task (Pan & Yang, 2010). In an image classification problem transfer learning works by training a CNN on a very large number of images and freezing the parameters for a certain number of layers before training further layers and the final classification layer (Kang & He, 2016). Low and mid-level features are likely common across even dissimilar datasets which allows the model to leverage these pre-learned features to reduce the training time and training data. Alternatively, one could attempt to transfer all the weights and biases and only retrain the final model output layer, or a more limited model top. While

this could have the advantage of being computationally cheaper and faster, it is more likely to run into the problem of negative transfer. Transfer learning requires that the source and target tasks not be too dissimilar. If the source task, such as classifying full color photographs, is drastically different from the target task, like classifying overhead SAR images, the transfer learning method may handicap the model performance (Pan & Yang, 2010).

E. IMAGENET AND ILSVRC

ImageNet is an open source labeled image database organized in a branching hierarchical method of “synonym sets” or “synsets.” For example, the “tank” synset is found in a tree going from vehicle to wheeled vehicle to self-propelled vehicle to armored vehicle to tank. Figure 6 provides another example of how images are grouped together. As of the writing of this thesis the ImageNet database consists of over 14 million labeled images organized into over 21,000 synsets. ImageNet is intended to function as a training and benchmarking resource for image recognition tasks, and due to the relationship with the WordNet lexical database could also be used to strength semantic understanding of scenes (Deng et al., 2009).

Figure 6. Example Hierarchy of Synsets. Source: Deng (2009).



The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is one of the leading challenges for computer vision tasks. Started in 2010, the ILSVRC classification and localization task consists of a subset of 1000 non-overlapping synsets for which

researchers train classifiers and submit their generated labels for unlabeled test images. The transfer learning approach explored in this thesis uses a model pre-trained on the ILSVRC-2014 dataset that consisted of 1.2 million images across the 1000 classes (Russakovsky et al., 2015).

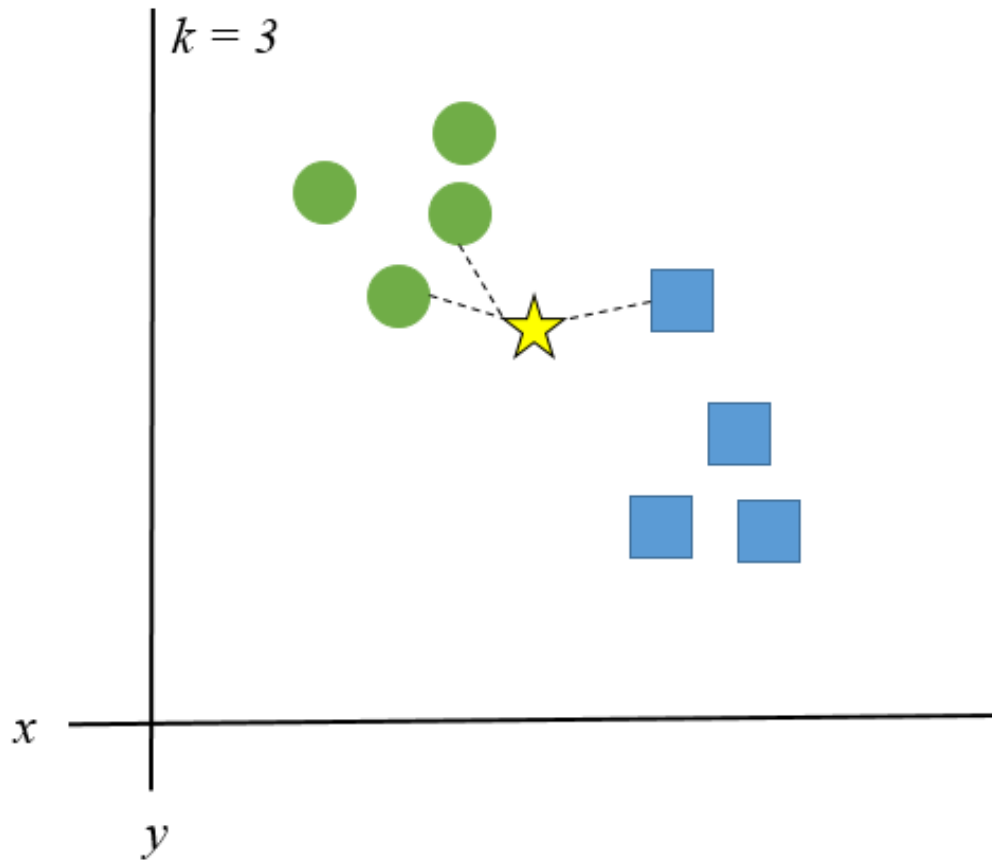
F. SHALLOW CLASSIFIERS

The use of multilayered neural networks for machine learning is also called deep learning. Shallow machine learning techniques do not rely on hidden layers to learn features and classify new data points. This often reduces the computational resources needed to implement shallow machine learning methods, but also often reduces the flexibility to deal with a problem like image classification where the possible feature space for an image is functionally infinite. The three “shallow” techniques used in this thesis are k-nearest neighbor (kNN), support vector machines (SVM) and random forests (RF).

1. K-Nearest Neighbor

K-nearest neighbor (kNN) is a classification algorithm where k represents the number of neighboring points that the classifier will examine when making a determination about what label to assign the new data point. When a data point is classified the classifier measures the distance from the new data point to the k nearest neighbors and assigns the data point the label that corresponds to the label of the largest number of neighboring points as shown in Figure 7. Weights can also be applied to the distances measured between the test point and neighboring points to increase the value of nearby neighbors when making a classification

Figure 7. Example of kNN. Adapted from Theobald (2017).



For this example, k is set to three. If an unweighted classifier is trying to assign the star to a specific class, it would assign it to the green circle class because of the three nearest objects two are green circles.

2. Support Vector Machines

Support vector machines shares many similarities with a logistic regression but rather than attempting to minimize the distance from all data points to the hyperplane that divides classes, the goal is to maximize the distance from the hyperplane to the nearest point in the divided clusters as shown in Figure 8. This reduces sensitivity to outliers and reduces the likelihood of misclassifying data points (Gandhi, 2018). In the case of non-linearly separable like that shown in Figure 9, the “kernel trick” can be applied to map the data into a higher dimensional space where a hyperplane may be found to separate the data.

Figure 8. Example SVM Classifier. Adapted from Theobald (2017).

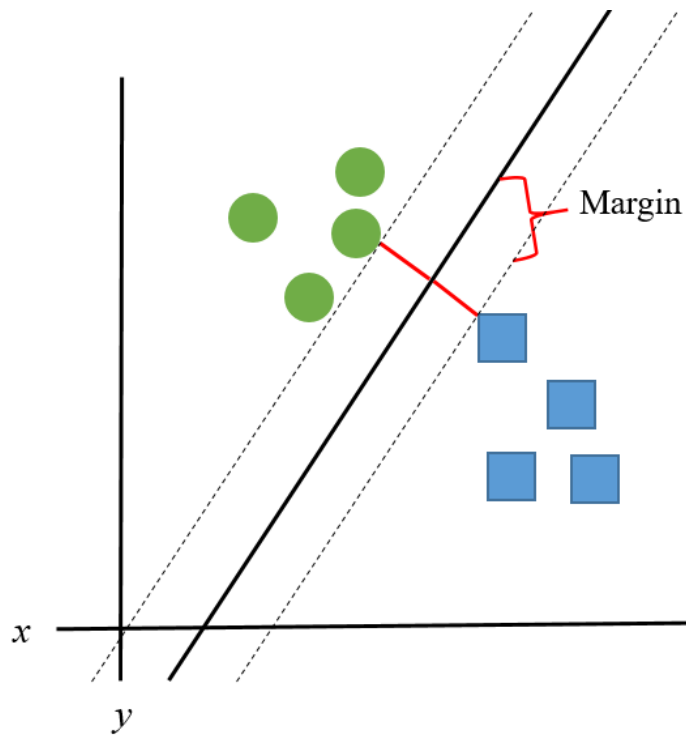
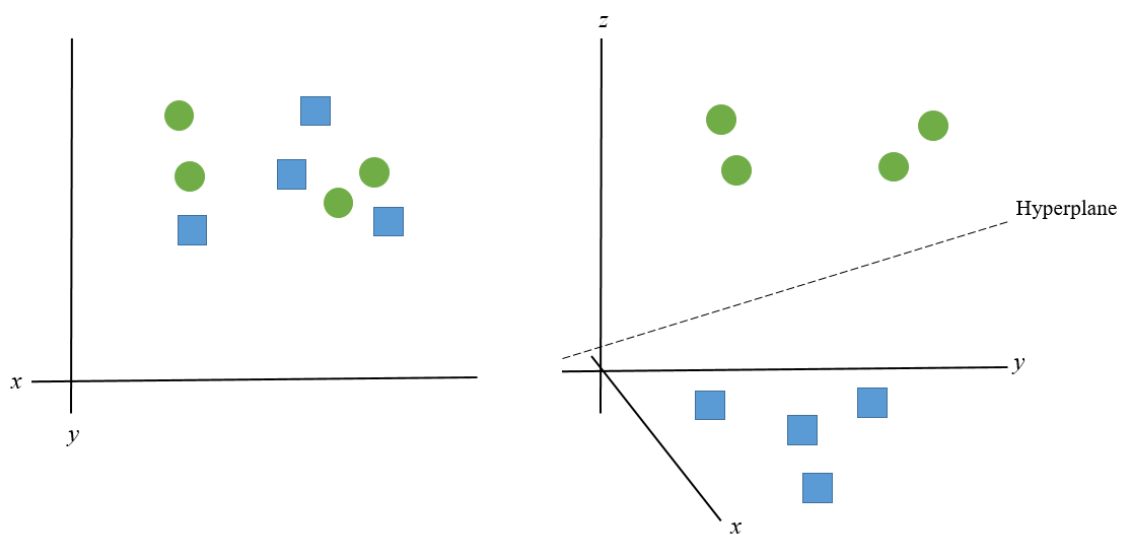


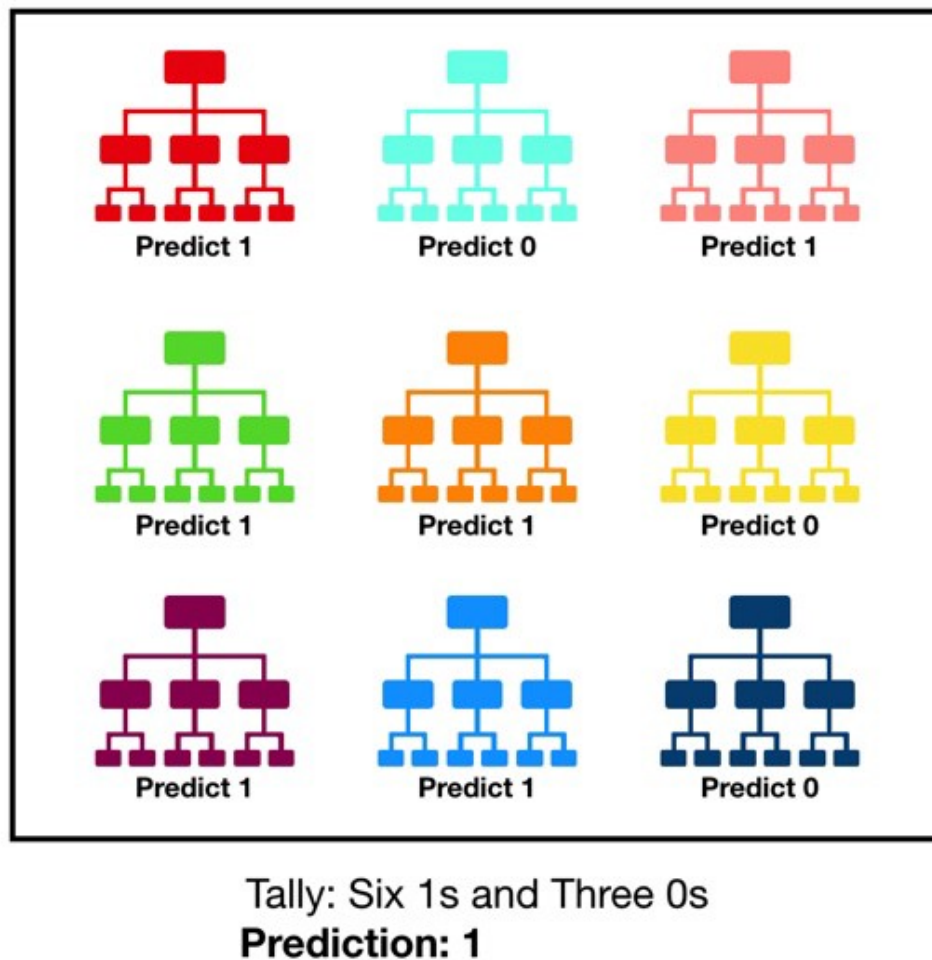
Figure 9. Non-Linearly Separable Data Transformed into Three-Dimensions. Adapted from Theobald (2017).



3. Random Forests

The final shallow classification algorithm used in this thesis is random forests. Random forests leverage decision trees that use binary splits to create branching decisions. In a random forest multiple different decision trees are run on a randomly selected subset of features to predict the class label. The decision trees also make splits based on different binary decisions. As demonstrated in Figure 10, the model then compares the predicted labels generated by the individual decision trees and the majority label is then assigned to the data point. By spreading the classification across multiple trees the random forest compensates for error in any single decision tree (Yiu, 2019).

Figure 10. Example of Random Forest. Source: Yiu (2019).



G. RELATED RESEARCH

Due its availability and ease of access for researchers, the Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset has become the standard dataset for SAR image classification research. The MSTAR dataset is described in greater detail in the methodology section. Initial work on the SAR ATR problem was conducted to support the Defense Advanced Research Projects Agency's Semi-Automated IMINT Processing (SAIP) project which grew out of the operational experience of hunting for Iraqi mobile ballistic missile launchers in the Gulf War. A template matching method pursued by the Lincoln Laboratories reported good results but required the construction of 72 templates per target class covering all orientations of the target (Novak et al., 1997). Research supporting the SAIP project was eventually able to produce 95.8% accuracy on the 10-class MSTAR dataset. There was, however, a notable reduction in performance when faced with certain real-world situations such as having a self-propelled artillery piece deployed in a revetment or a tank that has additional armor applied to it. Lincoln Laboratories was able to overcome these situations through the creation of additional templates (Novak, 2000). Since this research were in support of the SAIP program some data explored is not publically available today. SAR ATR research of the public MSTAR dataset using shallow classification methods also generally produced good results. An approach using an SVM classifier for a three-class subset of MSTAR targets with two confuser vehicles reported 93.4% accuracy (Bryant & Garber, 1999). An SVM method proposed by Zhao achieved 91% accuracy in a three-class test (Zhao & Principe, 2001). while a Bayesian classifier reported 95.05% accuracy in a 10-class test (O'Sullivan et al., 2001).

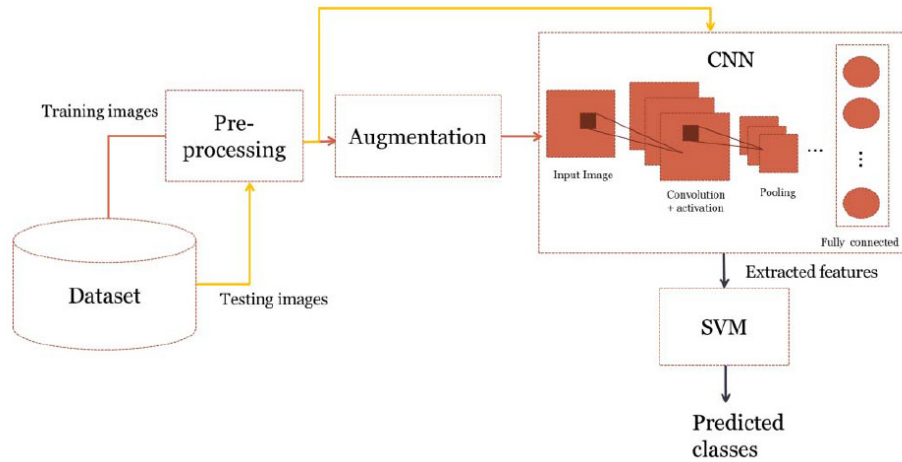
In recent years, the work on classification of SAR imagery has focused on CNNs. In 2015, Morgan showed that a relatively small CNN could achieve 92.1% accuracy across 10-classes of the MSTAR dataset roughly in line with the shallow methods previously explored. Morgan's method also showed that a network trained on nine of the MSTAR target classes could be retrained to include a tenth class 10–20 times faster than training a 10-class classifier from scratch. The ability to more easily adapt the model to changes in target sets represents an advantage over shallow classification techniques (Morgan, 2015). This is especially valuable in a military ATR context given the fluid nature of military

operations where changes to the order of battle may necessitate updating a deployed ATR system. In order to overcome the limitations caused by the relatively small number of images in the MSTAR dataset researchers explored transfer learning from a CNN pre-trained on simulated SAR images. Artificial SAR images were generated by using ray tracing software and detailed computer aided design models of target systems. They showed that model performance was improved, especially in cases where the amount of training data was reduced (Malmgren-Hansen et al., 2017). The technique of generating simulated SAR images for training could also be valuable in a military SAR ATR context where an insufficient amount of training data for some systems may exist.

Chen and Wang explored several approaches to deep learning and feature extraction for the MSTAR dataset. A method where the convolutional kernels were trained using a sparse auto-encoder using randomly sampled image patches resulted in 84.7% accuracy across 10 classes (Chen & Wang, 2014). The employment of a clever data augmentation scheme that sampled several 88x88 pixel patches from each training image in the MSTAR dataset increased the training dataset to 2700 images per class. This larger dataset allowed a CNN to achieve 99.1% accuracy on average across 10 classes (Wang et al., 2015).

While many CNNs use a softmax activation in the output layer of multi-class classifiers, there is evidence that SVM classifiers can outperform the softmax activation for some classification tasks (Tang, 2013). A multi-step classification process with transfer learning from ImageNet to MSTAR and feature extraction for training an SVM classifier was explored by Mufti et al. in 2018. Their methodology compared the performance of an SVM classifier trained on mid-level feature data extracted from multiple layers from AlexNet, GoogLeNet, and VGG16 neural networks without retraining the feature extracting network (Al Mufti et al., 2018). The basic workflow of this methodology is shown in Figure 11.

Figure 11. CNN to SVM Methodology. Source: Mufti (2018).



In the preprocessing step a center 50x53 pixel chip was extracted to reduce noise around the target. Two augmentations were performed on the training dataset, the mean grey level was subtracted and a Laplacian of the Gaussian filter was performed to emphasize target edges (Al Mufti et al., 2018). Combining these augmentations with the original images tripled the size of the training dataset. They reported 99.1% accuracy when classifying targets based on features extracted from mid-level convolutional layers from AlexNet. The best performance reported by Mufti et al. from the VGG16 architecture was 92.3% from a mid-level convolutional layer, but only 49.2% and 46.3% from features extracted in the last two fully connected layers.

Although not focused on SAR images, the application of transfer learning to remote sensing target detection and classification was previously studied at the Naval Postgraduate School by Lieutenant Katherine Rice in 2018. Rice showed that a CNN classifier trained on a photographic dataset could be retrained to perform remote sensing classification of EO satellite images of ships at sea with a recall of .99 and precision of .98 when retrained on a dataset contained 2000 images. This research supports that a transfer learning approach between modalities is feasible (Rice, 2018). Rice's approach to transfer learning is the foundation for the approach taken in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

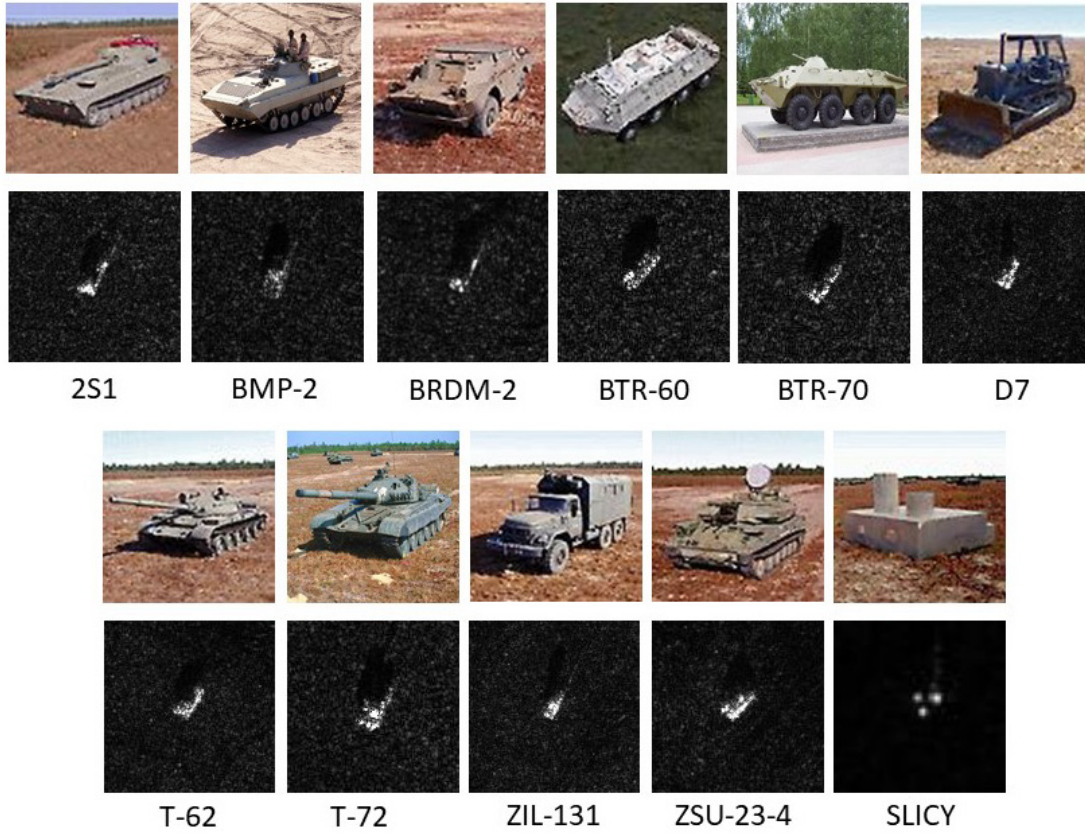
A. MSTAR DATASET

The MSTAR dataset is a publicly available dataset consisting of synthetic aperture radar images of the following 10 classes of military vehicles:

1. 2S1: former Soviet Union (FSU) self-propelled artillery
2. BMP-2: FSU infantry fighting vehicle
3. BRDM-2: FSU armored reconnaissance vehicle
4. BTR-60: FSU armored personnel carrier
5. BTR-70: FSU armored personnel carrier
6. D7: Caterpillar tractor frequently used in combat engineering roles
7. T-62: FSU main battle tank
8. T-72: FSU main battle tank
9. ZIL-131: FSU general purpose 6x6 truck
10. ZSU-23-4: FSU self-propelled anti-aircraft gun

The final class is the Sandia Laboratories implementation of cylinders (SLICY). The SLICY consisting of simple geometric shapes such as cylinders, edge reflectors, and corner reflectors which could be used for calibration of sensors or for modeling the propagation of radar reflections.

Figure 12. Example Photographs and MSTAR Images by Class. Adapted from U.S. Air Force (1996), Leahy (1994), and Torin (2008).



The dataset was collected at a 15-degree and a 17-degrees look angle in September 1995 at the Redstone Arsenal, Huntsville, AL, by the Sandia National Laboratory SAR sensor platform. The sensor was operated in spotlight mode producing images with a 1-foot resolution. The dataset consists of 128x128 pixel greyscale images representing one of the few publicly available SAR datasets and therefore a commonly studied dataset for SAR ATR. The total number of images per class are described in Table 1. Since there is not a consistent number of images per class, 200 images collected from a 17-degree look angle were randomly selected from each class as the training data. The test set consisted of 2700 images collected at a 15-degree look angle and generated by dropping a single image from the 10 vehicle classes and three SLICY images.

Table 1. Number of SAR Images per Class

	17 degree look angle	15 degree look angle
2S1	299	274
BMP-2	233	195
BRDM-2	298	274
BTR-60	256	195
BTR-70	233	196
D7	299	274
T-62	298	273
T-72	232	196
ZIL-131	299	274
ZSU-23-4	299	274
SLICY	274	288
Total	3020	2713

B. NETWORK ARCHITECTURE

The CNN architecture employed in this thesis is a modified VGG16 architecture (Simonyan & Zisserman, 2015). The original VGG16 architecture is shown in in Figure 13. The numbers on the convolutional layers represent the kernel window and depth while the number on the fully connected layers is the number of neurons in the layer. The VGG16 architecture consists of linked convolutional and pooling blocks made up of two or three convolutional layers and a pooling layer before ending with three fully connected layers. The final layer uses a softmax activation to determine the class label. The network employs a 3x3 kernel and a stride of one so that each pixel is the center of a convolutional step.

Figure 13. VGG16 Architecture. Adapted from Simonyan and Zisserman (2015).

Input
3x3 Conv2D - 64
3x3 Conv2D - 64
Max Pooling
3x3 Conv2D - 128
3x3 Conv2D - 128
Max Pooling
3x3 Conv2D - 256
3x3 Conv2D - 256
3x3 Conv2D - 256
Max Pooling
3x3 Conv2D - 512
3x3 Conv2D - 512
3x3 Conv2D - 512
Max Pooling
3x3 Conv2D - 512
3x3 Conv2D - 512
3x3 Conv2D - 512
Max Pooling
Fully-Connected - 4096
Fully-Connected - 4096
Fully-Connected - 1000
Output (softmax)

The base architecture is modified for this thesis as shown in Figures 14 and 15. The model top of three fully connected layers is replaced with a new model top consisting of a fully connected layer, a dropout layer to mitigate overfitting, and two final fully connected layers with a softmax activation for classification. For the partial transfer learning approach the model is initialized with the ImageNet weights and has the first two convolutional/pooling blocks frozen for training to take advantage of the broad feature detection of the pre-trained network (Rice, 2018). This thesis expands on the partial transfer learning approach by exploring a full transfer of weights in the convolutional base. This is achieved by freezing all convolutional and pooling layers and only training the new model top.

Figure 14. Modified VGG16 with Partial Transfer Learning and 11-Class Output. Adapted from Rice (2018).

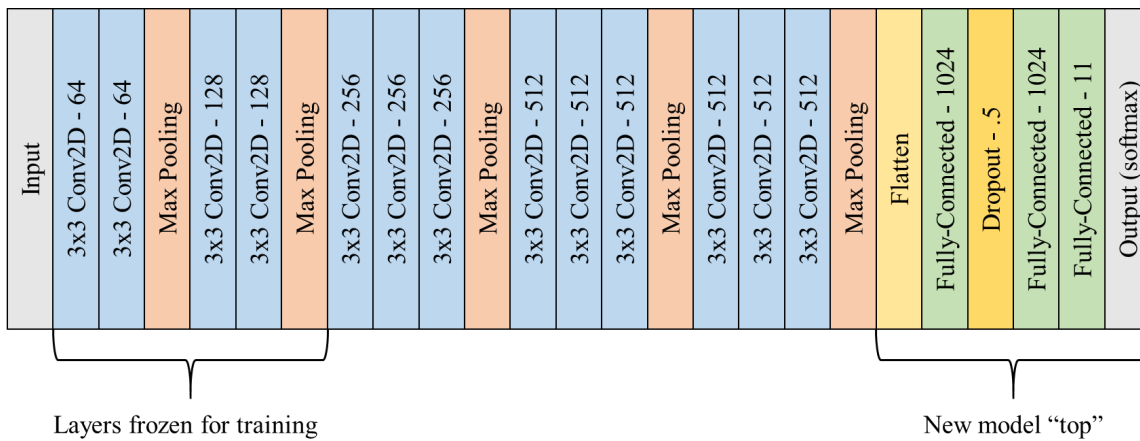


Figure 15. Modified VGG16 with Full Transfer Learning and 11-Class Output. Adapted from Rice (2018).

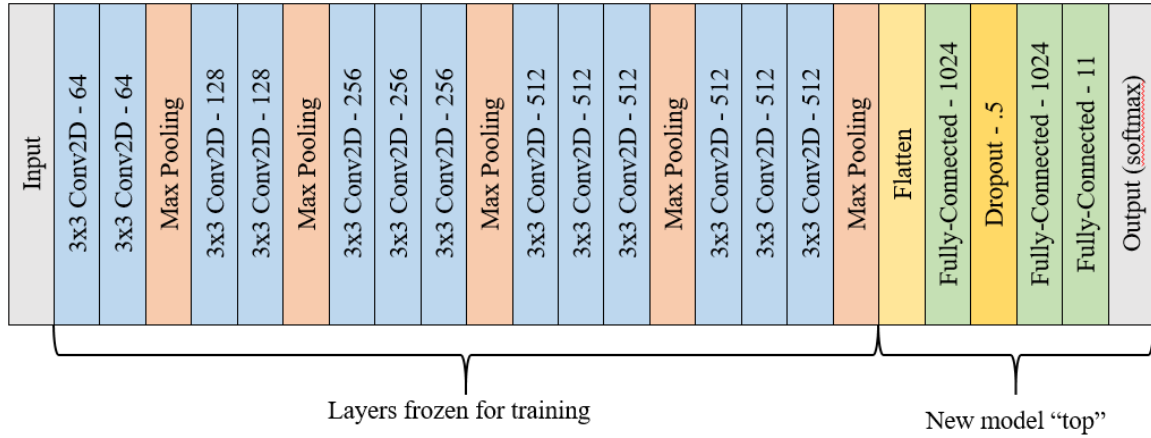
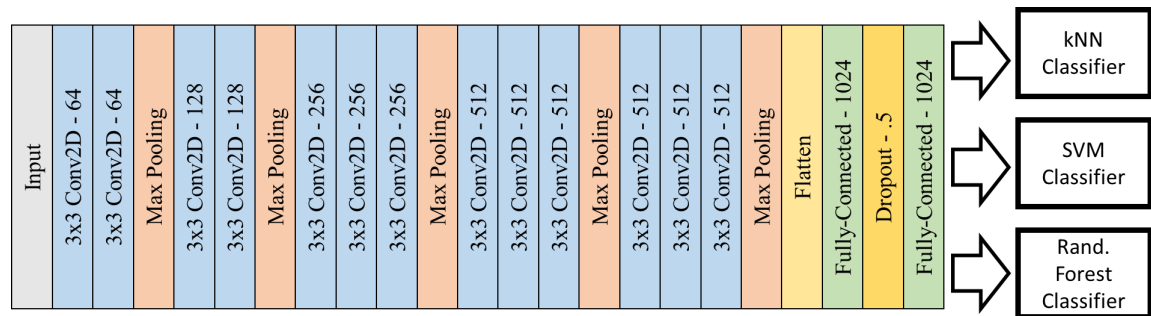


Figure 16 shows process of the multistep methods. A SAR image is passed through the CNN to extract features from the second fully connected layer. The extracted features are saved as a 1024-dimensional vector and used to train the shallow classifiers. Test data is also passed through the CNNs in order to vectorized the images which are then used to evaluate the performance of the shallow classifiers against the softmax activation layer of the base CNN.

Figure 16. Multistep Classifier Using a CNN for Feature Extraction. Adapted from Rice (2018).



C. TENSORFLOW, KERAS API, AND ORANGE TOOLKIT

The previously described models are implemented using the Keras application program interface (API) with TensorFlow as the backend. TensorFlow is an open source

Python machine learning library developed by Google. TensorFlow greatly reduces the programming workload for implementing deep learning. Rather than having to build a CNN from scratch, TensorFlow allows for the creation of models by calling functions to create layers with given parameters. It also provides visualization tools for learning rate, training time, and other useful information in the evaluating the training process of a deep learning model. The Keras API is capable of running on top of other machine learning libraries in addition to TensorFlow and has a prebuilt VGG16 available. The Keras API is a more user-friendly interface with the TensorFlow library. The ImageNet weights available in Keras are ported from Visual Geometry Group at Oxford University that developed the VGG16 architecture for ILSVRC-2014 localization and classification tasks (Simonyan & Zisserman, 2015).

Orange is an open source data science and machine learning toolkit that allows users to easily manipulate data through a graphical user interface. Orange has several built in machine learning algorithms and simplifies the data management and preprocessing requirements to allow users to experiment with approaches to machine learning and data science (Demšar et al., 2004).

D. MEASURES OF PERFORMANCE AND EXPERIMENTAL SETUPS

1. Measures of Performance

The primary methods for measuring the model performance in this thesis are recall and precision. Recall is the proportion of targets that are correctly classified by the model. Recall can also be thought of as the true positive rate. Precision is measure of the percentage of true positives over total positives. High recall indicates that the model correctly classifies targets while high precision indicates that the model does not have many false positives. Both measures have their origins in data retrieval but are a standard way of measuring machine learning performance.

$$\text{Recall} = \frac{\text{True Positives}}{\text{Total Targets in Class}} \quad (3)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{Total Predicted Positives}} \quad (4)$$

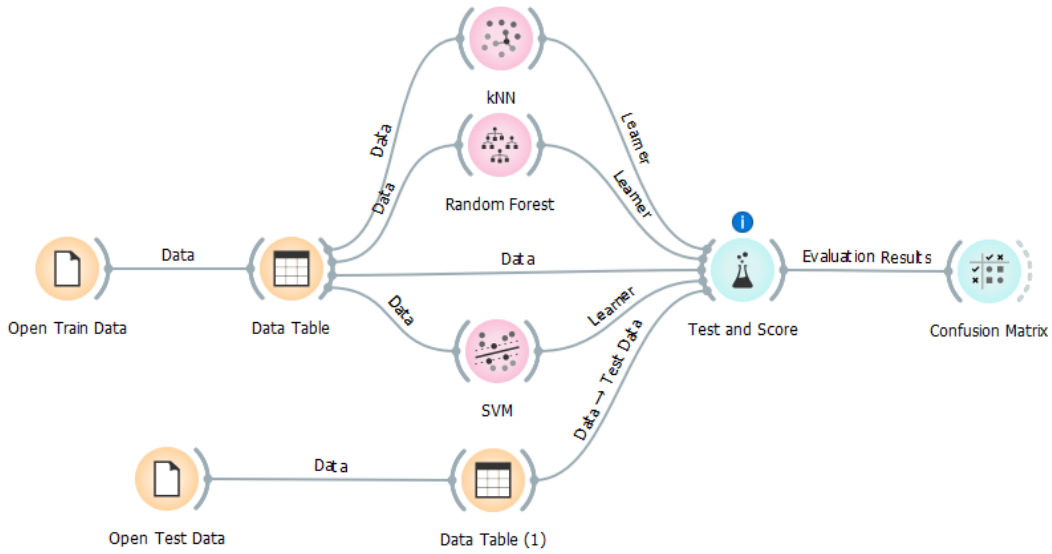
2. Experiment 1

In Experiment 1, the CNN is trained without transfer learning on 200 training images per class with a 20% validation split. The same training dataset is applied to the partial transfer learning and full transfer learning approaches depicted in Figures 13 and 14. These models form the basis for the feature extraction comparisons in follow on experiments.

3. Experiment 2

Experiment 2 explores the utility of CNNs as feature extractors for training shallow classifiers. Features are extracted from the last fully connected layer prior to the final output layer of the CNN as depicted in Figure 16 for both the training and test datasets and are saved as a 1024-dimensional vector representing each image. The vectorized feature representations of the images are run through the Orange workflow pictured in Figure 17. Performance is then compared between the base CNNs and kNN, SVM, and Random Forest classifiers trained and evaluated on the extracted features. For the kNN classifier k was set to 11 and weighted Euclidian distance was used to determine which class label to assign to test images. A sigmoid kernel was used in the SVM classifier, and the random forest consisted of 10 decision trees. The parameters for the shallow classifiers are unchanged in Experiments 3 and 4.

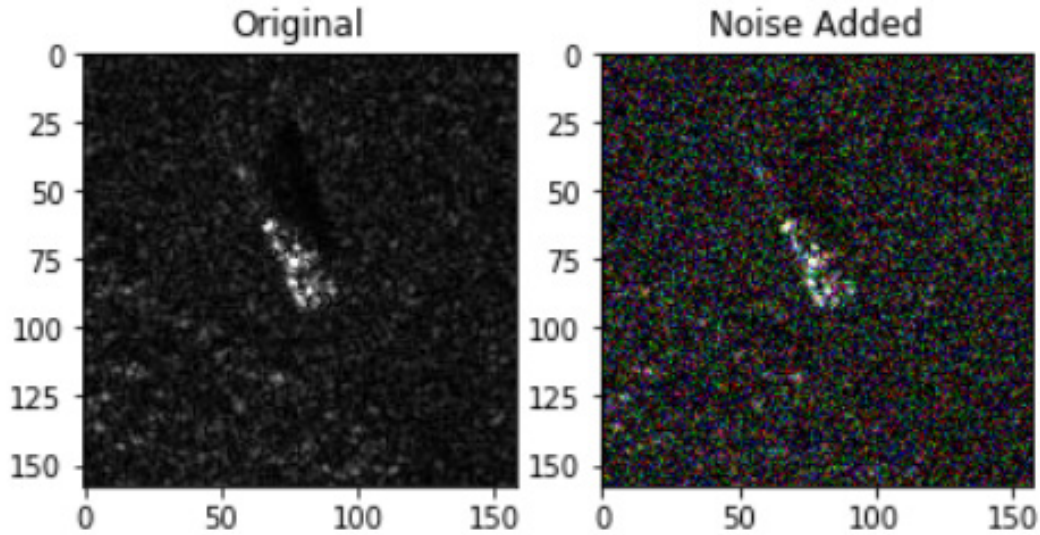
Figure 17. Orange Workflow



4. Experiment 3

For Experiment 3, random Gaussian noise was added to the images from the datasets test and training datasets. Two noise levels were set to a random value up to 5% and 20% of the max pixel value to create low-noise and high-noise tests. An example of the resulting image in the high-noise dataset is shown in Figure 18. The noise added images are then classified by the CNN and multistep process without retraining the CNN in order to test the robustness of the classifiers to noise. The CNNs were retrained on the high-noise images and features extracted from the noise added training dataset were used to train the shallow classifiers.

Figure 18. Comparison of Original Image and 20% Noise Added Image



5. Experiment 4

Experiment 4 seeks to determine how model performance of the partial transfer learning approach is affected as the size of the training dataset is reduced. The training dataset is reduced from 200 images per class by randomly selecting 100 images per class from the total available training data to create a new reduced training set. The process was repeated to select 50 images per class from the total available training dataset for a second reduced training set. The partial transfer learning approach is then applied with the reduced training datasets, and the shallow classifiers are trained as in Experiment 2.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS AND ANALYSIS

A. EXPERIMENTAL RESULTS

1. Experiment 1

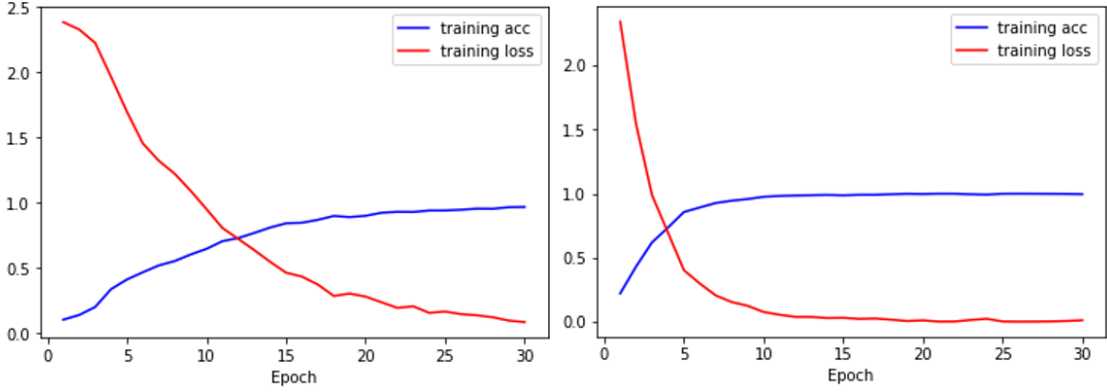
Table 2 compares the performance of the CNN trained without transfer learning on the MSTAR data to the two transfer learning approaches described in Figures 13 and 14. The partial transfer learning approach shows some moderate improvement over the CNN trained exclusively on the MSTAR data. The full transfer learning of all convolutional weights and only retraining the CNN top did not match the performance of the non-transfer learning approach, suggesting some negative transfer occurs in the later convolutional layers. The transfer learning approach also had the advantage of converging much more quickly than the CNN initialized with random weights. Figure 19 shows the training loss and training accuracy of the CNN trained without transfer learning and the partial transfer learning method.

Table 2. Comparison of CNN Performance.

	VGG16 Trained Exclusively on SAR		VGG16 with Partial Transfer Learning		VGG16 with Full Transfer Learning	
Class	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.957	0.954	0.980	0.980	0.883	0.878
2S1	1.000	0.810	0.996	0.945	0.962	0.560
BMP-2	0.949	0.969	0.969	0.954	0.831	0.686
BRDM-2	0.993	1.000	1.000	1.000	0.985	0.978
BTR-60	0.984	0.959	0.995	0.948	0.774	0.933
BTR-70	0.995	0.954	0.950	0.979	0.851	0.764
D7	0.982	0.978	0.996	0.985	0.980	0.886
T-62	0.931	0.842	0.960	0.978	0.914	0.897
T-72	0.956	0.995	0.956	0.995	0.776	0.979
ZIL-131	0.894	0.985	0.996	0.996	0.850	0.974
ZSU-23-4	0.848	1.000	0.968	1.000	0.798	1.000
SLICY	0.997	1.000	0.997	1.000	0.997	1.000

This table compares the performance of the architectures described in Figures 14 and 15

Figure 19. Comparison of Training Loss and Accuracy



The left graph depicts the training accuracy and loss of a CNN initialized with random weights. The right graph shows a model trained with the partial transfer learning method described in Figure 14. The training loss of the transfer learning model descends much more rapidly than the model trained exclusively on the MSTAR dataset. Similarly, the transfer learning method’s accuracy more rapidly approaches 1.

2. Experiment 2

The multistep method using shallow classifiers trained on features extracted from the CNN showed some improvement in recall over the base CNN, particularly in the lowest performing classes in the base CNN like the 2S1 and T-62. The multistep method also saw generally higher performance in precision as well, once again well outperforming the base CNNs, particularly in the low precision classes such as the ZSU-23-4. As shown in Table 3, all shallow classifiers showed improved average performance over the base CNN that was not pre-trained. Tables 4 and 5 show the multistep classifier performance compared to the CNNs pre-trained on ImageNet. The kNN classifier matched the partial transfer learning model’s performance and exceeded the SVM classifier. In the full transfer learning approach the kNN and SVM classifiers exceeded the CNN’s average performance in both precision and recall.

Table 3. Comparison of Multistep Classifier without Transfer Learning

	VGG16 Trained Exclusively on SAR		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + RF	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.957	0.954	0.977	0.976	0.977	0.976	0.973	0.973
2S1	1.000	0.810	1.000	0.905	1.000	0.886	0.996	0.964
BMP-2	0.949	0.969	0.969	0.974	0.959	0.969	0.945	0.969
BRDM-2	0.993	1.000	1.000	1.000	1.000	1.000	1.000	1.000
BTR-60	0.984	0.959	0.983	0.918	0.979	0.974	0.949	0.954
BTR-70	0.995	0.954	0.959	0.969	0.989	0.964	0.984	0.964
D7	0.982	0.978	0.993	0.989	0.993	0.974	0.982	0.996
T-62	0.931	0.842	0.967	0.974	0.950	0.974	0.949	0.967
T-72	0.956	0.995	0.942	1.000	0.961	1.000	0.965	0.995
ZIL-131	0.894	0.985	0.941	0.993	0.951	0.993	0.951	0.989
ZSU-23-4	0.848	1.000	0.972	1.000	0.955	1.000	0.975	1.000
SLICY	0.997	1.000	1.000	1.000	1.000	1.000	0.997	1.000

Table 4. Comparison of Multistep Classifier with Partial Transfer Learning

	VGG16 with Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + RF	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.980	0.980	0.980	0.980	0.976	0.976	0.970	0.970
2S1	0.996	0.945	0.993	0.938	0.996	0.934	0.989	0.963
BMP-2	0.969	0.954	0.940	0.964	0.951	0.897	0.936	0.907
BRDM-2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
BTR-60	0.995	0.948	0.984	0.948	0.979	0.948	0.968	0.928
BTR-70	0.950	0.979	0.974	0.944	0.926	0.959	0.934	0.938
D7	0.996	0.985	0.996	0.989	0.996	0.985	0.974	0.960
T-62	0.960	0.978	0.944	0.985	0.944	0.985	0.961	0.989
T-72	0.956	0.995	0.960	0.995	0.942	0.995	0.928	0.985
ZIL-131	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.963
ZSU-23-4	0.968	1.000	0.978	1.000	0.975	1.000	0.961	1.000
SLICY	0.997	1.000	1.000	1.000	1.000	1.000	0.993	1.000

See Figure 14 for architecture of base CNN and Figure 16 for architecture of multistep classifier.

Table 5. Comparison of Multistep Classifier with Full Transfer Learning

	VGG16 with Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + RF	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.883	0.878	0.917	0.912	0.913	0.908	0.842	0.838
2S1	0.962	0.560	0.983	0.634	0.994	0.579	0.880	0.458
BMP-2	0.831	0.686	0.811	0.820	0.792	0.747	0.688	0.670
BRDM-2	0.985	0.978	0.982	0.989	0.964	0.982	0.945	0.941
BTR-60	0.774	0.933	0.871	0.902	0.892	0.851	0.750	0.804
BTR-70	0.851	0.764	0.814	0.851	0.871	0.831	0.723	0.790
D7	0.980	0.886	0.971	0.974	0.974	0.971	0.905	0.872
T-62	0.914	0.897	0.934	0.938	0.934	0.938	0.856	0.875
T-72	0.776	0.979	0.824	0.985	0.841	0.979	0.774	0.897
ZIL-131	0.850	0.974	0.848	0.978	0.831	0.974	0.797	0.850
ZSU-23-4	0.798	1.000	0.916	1.000	0.916	1.000	0.822	0.996
SLICY	0.997	1.000	0.990	1.000	0.993	1.000	0.993	1.000

See Figure 15 for architecture of base CNN and Figure 16 for architecture of multistep classifier.

3. Experiment 3

The sensitivity of CNNs to noise is clearly demonstrated by the drop in CNN performance presented in Table 6 when even low amounts of random noise are introduced. With 20% random noise the models become totally ineffective as shown in Table 7. Although the performance of the full transfer learning method is not acceptable in any real world context, it is the least impacted by noisy images seeing a drop in average recall of .131 as compared to .185 and .281 for the CNN trained exclusively on SAR data and the partial transfer learning approach respectively.

Table 6. Comparison of CNN Performance on Low-Noise Images without Retraining

	VGG16 Trained Exclusively on SAR		VGG16 with Partial Transfer Learning		VGG16 with Full Transfer Learning	
Class	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.854	0.769	0.828	0.699	0.808	0.747
2S1	0.748	0.996	0.471	0.381	0.778	0.604
BMP-2	0.585	0.959	0.663	0.820	0.435	0.892
BRDM-2	1.000	0.110	1.000	0.051	0.957	0.729
BTR-60	0.968	0.778	0.993	0.763	0.702	0.814
BTR-70	0.981	0.267	1.000	0.195	0.900	0.369
D7	0.689	0.996	0.820	0.952	0.660	0.938
T-62	0.973	0.934	0.876	0.835	0.932	0.456
T-72	0.979	0.939	0.944	0.944	0.977	0.667
ZIL-131	0.505	0.938	0.351	0.996	0.630	0.872
ZSU-23-4	0.967	0.641	0.991	0.777	0.920	0.886
SLICY	1.000	0.989	1.000	0.975	0.996	0.986

Table 7. Comparison of CNN Performance on High-Noise Images without Retraining

	VGG16 Trained Exclusively on SAR		VGG16 with Partial Transfer Learning		VGG16 with Full Transfer Learning	
Class	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.177	0.115	0.035	0.169	0.296	0.250
2S1	0.110	0.996	0.197	0.891	0.216	0.469
BMP-2	0.260	0.165	0.000	0.000	0.158	0.675
BRDM-2	0.000	0.000	0.000	0.000	0.111	0.084
BTR-60	0.581	0.093	0.000	0.000	0.199	0.706
BTR-70	1.000	0.010	0.000	0.000	0.429	0.077
D7	0.000	0.000	0.000	0.000	0.600	0.560
T-62	0.000	0.000	0.000	0.000	0.000	0.000
T-72	0.000	0.000	0.000	0.000	0.000	0.000
ZIL-131	0.000	0.000	0.186	0.971	0.545	0.176
ZSU-23-4	0.000	0.000	0.000	0.000	1.000	0.004
SLICY	0.000	0.000	0.000	0.000	0.000	0.000

Tables 8, 9, and 10 show the model performance after retraining on the high-noise dataset. The pre-trained CNNs do not match the performance of the CNN trained exclusively on MSTAR data. The trend in Experiment 2 of improved average accuracy of shallow classifiers trained on features from CNNs holds with noisy data as well. The kNN

and SVM classifiers outperformed the models from which features were extracted in both precision and recall. The addition of noise more negatively impacts the transfer learning approaches compared to a CNN trained from scratch on noisy data.

Table 8. Comparison of Multistep Classifier without Transfer Learning
Trained on 20% Noisy Images

	VGG16 without Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + RF	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.928	0.925	.931	.929	.936	.933	.923	.919
2S1	0.941	0.817	.977	.795	.977	.795	.944	.736
BMP-2	0.727	0.933	.756	.845	.759	.876	.690	.861
BRDM-2	0.953	0.886	.933	.912	.937	.927	.924	.934
BTR-60	0.948	0.845	.907	.902	.923	.871	.873	.887
BTR-70	0.950	0.872	.926	.897	.935	.892	.929	.867
D7	0.961	0.985	.982	.974	.978	.978	.989	.978
T-62	0.938	0.897	.915	.912	.934	.934	.934	.890
T-72	0.936	0.979	.927	.979	.905	.979	.917	.964
ZIL-131	0.897	0.985	.909	.985	.937	.985	.927	.978
ZSU-23-4	0.964	0.978	.954	.985	.951	.989	.947	.978
SLICY	0.993	1.000	.993	.996	.990	1.000	.997	1.000

Table 9. Comparison of Multistep Classifier with Partial Transfer Learning
Trained on 20% Noisy Images

	VGG16 with Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + RF	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.904	0.907	0.915	0.916	0.920	0.922	0.900	0.903
2S1	1.000	0.634	0.963	0.766	0.981	0.758	0.922	0.784
BMP-2	0.720	0.784	0.748	0.825	0.759	0.845	0.783	0.820
BRDM-2	0.928	0.941	0.947	0.912	0.942	0.945	0.928	0.897
BTR-60	0.888	0.897	0.930	0.887	0.903	0.912	0.869	0.887
BTR-70	0.826	0.928	0.873	0.913	0.894	0.908	0.880	0.862
D7	0.967	0.967	0.964	0.967	0.953	0.974	0.923	0.971
T-62	0.948	0.879	0.935	0.897	0.964	0.893	0.911	0.864
T-72	0.846	0.985	0.883	0.964	0.879	0.964	0.879	0.969
ZIL-131	0.914	0.974	0.890	0.974	0.910	0.960	0.871	0.938
ZSU-23-4	0.928	0.989	0.934	0.978	0.937	0.982	0.948	0.938
SLICY	0.983	1.000	1.000	0.996	0.997	1.000	0.990	1.000

See Figure 14 for architecture of base CNN and Figure 16 for architecture of multistep classifier.

Table 10. Comparison of Multistep Classifier with Full Transfer Learning
Trained on 20% Noisy Images

	VGG16 with Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + RF	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.768	0.771	0.784	0.773	0.793	0.785	0.698	0.693
2S1	0.840	0.385	0.847	0.385	0.816	0.407	0.602	0.355
BMP-2	0.502	0.562	0.459	0.608	0.506	0.634	0.369	0.521
BRDM-2	0.796	0.758	0.795	0.780	0.796	0.817	0.716	0.656
BTR-60	0.702	0.825	0.745	0.753	0.731	0.814	0.569	0.701
BTR-70	0.708	0.672	0.696	0.692	0.712	0.672	0.560	0.503
D7	0.892	0.821	0.839	0.861	0.904	0.824	0.794	0.846
T-62	0.708	0.783	0.805	0.699	0.783	0.728	0.635	0.596
T-72	0.729	0.938	0.743	0.949	0.748	0.928	0.699	0.856
ZIL-131	0.824	0.842	0.828	0.821	0.824	0.857	0.800	0.718
ZSU-23-4	0.807	0.890	0.780	0.919	0.794	0.919	0.809	0.806
SLICY	0.944	1.000	0.944	1.000	0.960	1.000	0.946	0.993

See Figure 15 for architecture of base CNN and Figure 16 for architecture of multistep classifier.

4. Experiment 4

Building off of the performance of the partial transfer learning method with shallow classifiers shown in Experiments 1 and 2, Experiment 4 seeks to determine the partial transfer learning method's performance as training data is reduced. Multistep classification methods showed significant improvements over the base model as the size of the training dataset decreased as shown in Tables 11 and 12. When the training data was reduced to 100 images per class the multistep classifiers all performed better on average than the base CNN. Both the kNN and SVM classifiers maintained precision and recall performance above .95 while the base CNN performance dropped to .907 for recall and .904 for precision. When the training dataset was reduced to 50 images per class all multistep classifiers continued to outperform the baseline model and saw much better performance on difficult targets like the BTR-60, BTR-70 and T-62.

Table 11. Performance of Multistep Classifier with Partial Transfer Learning
Trained on 100 Images per Class

	VGG16 with Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + RF	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.918	0.906	0.960	0.959	0.958	0.957	0.929	0.926
2S1	0.963	0.670	1.000	0.875	1.000	0.872	1.000	0.700
BMP-2	0.978	0.902	0.971	0.856	0.977	0.876	0.947	0.830
BRDM-2	0.996	0.967	0.989	0.996	0.996	1.000	0.982	0.989
BTR-60	0.967	0.912	0.907	0.959	0.913	0.974	0.819	0.954
BTR-70	0.989	0.923	0.934	0.949	0.945	0.944	0.899	0.918
D7	0.836	0.912	0.989	0.971	0.974	0.949	0.924	0.938
T-62	0.745	0.882	0.949	0.963	0.925	0.949	0.917	0.938
T-72	0.883	0.964	0.937	0.990	0.937	0.990	0.955	0.969
ZIL-131	0.974	0.828	0.941	0.985	0.964	0.974	0.881	0.952
ZSU-23-4	0.791	1.000	0.938	1.000	0.907	0.996	0.901	1.000
SLICY	0.979	1.000	1.000	1.000	0.997	1.000	0.990	1.000

See Figure 14 for architecture of base CNN and Figure 16 for architecture of multistep classifier.

Table 12. Performance of Multistep Classifier with Partial Transfer Learning
Trained on 50 Images per Class

	VGG16 with Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + RF	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.808	0.770	0.879	0.874	0.867	0.862	0.822	0.806
2S1	0.980	0.707	0.959	0.777	0.958	0.758	0.900	0.722
BMP-2	0.569	0.830	0.706	0.732	0.751	0.716	0.681	0.639
BRDM-2	0.957	0.989	0.993	0.989	0.997	0.993	0.971	0.993
BTR-60	0.935	0.371	0.889	0.742	0.745	0.814	0.707	0.670
BTR-70	0.641	0.605	0.741	0.805	0.775	0.708	0.667	0.667
D7	0.648	0.795	0.840	0.883	0.861	0.842	0.775	0.681
T-62	0.853	0.640	0.883	0.919	0.864	0.868	0.855	0.779
T-72	0.785	0.918	0.923	0.928	0.908	0.908	0.961	0.892
ZIL-131	0.889	0.615	0.888	0.842	0.902	0.879	0.854	0.835
ZSU-23-4	0.626	1.000	0.848	1.000	0.778	1.000	0.678	0.989
SLICY	1.000	1.000	1.000	1.000	1.000	1.000	0.997	1.000

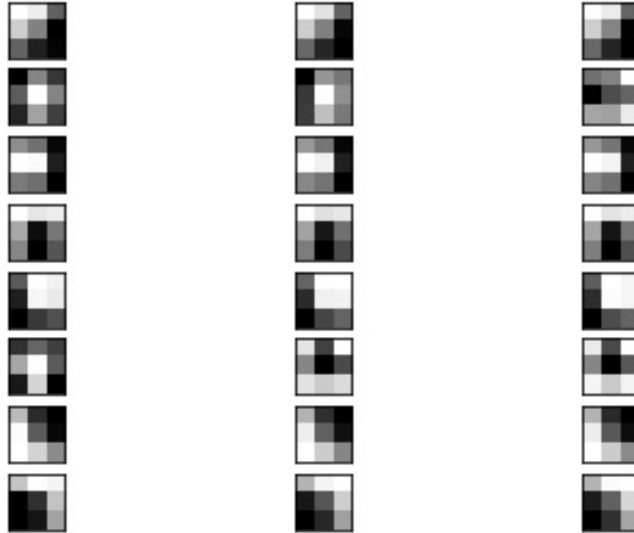
See Figure 14 for architecture of base CNN and Figure 16 for architecture of multistep classifier.

B. ANALYSIS

The partial transfer learning approach of transferring low-level features outperformed the full transfer learning approach on both the original and noisy images. While a model trained exclusively on noisy SAR data performed better on noisy images

the performance of the partial transfer learning method is notable on the clean dataset. SAR images are significantly different from the photographs that compose the ImageNet dataset. However, the filters in the early convolutional layer are designed to detect relatively simple features like edges and corners. Figure 20 shows an example of the filters in the first convolutional layer. The features that create strong activations in these filters such as brightness gradients, edges, or corners are not uncommon and could be applied to dissimilar datasets. In later layers the features that the model is attempting to resolve become less transferable which increases the likelihood of negative transfer as seen in Experiment 1.

Figure 20. Example Filters from First Convolutional Layer.



The performance of the kNN classifier compared to the softmax activation for neural network output is notable. Multistep classifier performance on the most difficult classes for the CNN was significantly better in several of the tests. When trained on 50 images the CNN only achieved a recall score of .371 on the BTR-60 class, well below the .742 and .814 of the kNN and SVM classifiers. In the same test the base CNN had recall rates below .65 for the BTR-70, T-62, and ZIL-131, while the kNN and SVM classifiers were able to achieve results for these cases with a recall above .8. There are also large reductions in precision for some classes that are not mirrored in the multistep method. The

nature of the SVM and kNN algorithms may provide greater robustness under the evaluated circumstances when compared to the CNN classifiers. The support vectors and number of neighbors used to classify new data points do not rely on a probability the way a softmax activation does and therefore are less likely to be affected as harshly by a reduced sample size. Since the kNN algorithm calculates the distance to all training data points the computational cost of the kNN algorithm increases with the volume of training data used. The good performance of the kNN method as the training data is reduced as described in this thesis is represents a computationally cheap way to build a future classifier.

Excepting the performance of models on noisy data prior to retraining, all models performed well in both recall and precision on the SLICY target. Performance on the SLICY class is of interest because it demonstrates the model's ability to discriminate a non-valid target from a valid target. All other classes, with the exception of the D7, are former Soviet Union military equipment. Up-armored versions of the D7 and related equipment are often used in combat engineering roles. In a military context this means they are likely to be a valid target. The classification of a SLICY as any other class would indicate the model is accepting a clearly invalid target as a valid target. As demonstrated by the high precision in this class across the experiments valid targets are very infrequently classified as a SLICY and the high recall indicates that the random objects are not being accepted as valid targets.

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

When examining the research questions proposed in this thesis we can make the following observations:

1. Can transfer learning from a model pre-trained on photographs be effective for SAR images?
 - Partial transfer learning of low-level features from photographs to SAR imagery is effective for training a neural network both for classification and feature extraction. Transfer learning of features from early convolutional blocks from a CNN pre-trained on ImageNet showed improved performance over the CNN trained only on MSTAR data. Full transfer of features from the convolutional and pooling layers had lower performance than both the partial transfer learning approach and the CNN trained exclusively on SAR imagery.
2. Does using a neural network as a feature extractor for input into a shallow classification algorithm improve model performance?
3. What shallow classification technique provides the greatest benefit to classification accuracy?
 - A retrained neural network can function as an efficient feature extractor for training a shallow classifier. The performance of the kNN and SVM classifiers trained on features extracted from the retrained CNN exceed the performance of the CNN. Methods such as kNN and SVM classifiers are a potentially useful replacement for the softmax activation common in the final layer of a neural network.

4. Does a shallow classification method trained on features extracted from a neural network perform better in noisy conditions or with a small training dataset?
 - Multistep classification methods using a shallow classifier trained on features extracted from a neural network outperformed the base neural network when tested on noisy data and as the amount of training data decreases. As the training dataset was reduced in size the performance of the shallow classifiers, particularly the kNN and SVM classifiers was less negatively impacted than the retrained CNN.

The National Geospatial-Intelligence Agency manages a GEOINT Professional Certification program. The GPC certificates are available for a fundamentals and 10 specializations, including imagery analysis and geospatial analysis. Tests are scored based on a Modified Angoff method where a subject matter expert determines the probability that a hypothetical minimally competent examinee would answer a question correctly (National Geospatial-Intelligence Agency, 2017). IMINT analysis also often relies on the user's experience and confidence in their own work providing responses such as "possible main battle tank" or "likely BMP-2." The inherently subjective nature by which the GPC is scored, and the confidence based assessments of IMINT analysts make a direct comparison of model performance to expert level difficult to determine.

Both the baseline model employing transfer learning and the shallow classifiers using a neural network as a feature extractor performed with a high degree of accuracy and would be valuable in an operational context as an aid to IMINT analysts. The multistep classification technique also shows that a very large training dataset does not need to be developed and a system could be trained with as few as 100 unaugmented training images per class. This presents an opportunity for an operational SAR ATR model to be updated based on the current units in a specific area of operations or on vehicles that are modified or damaged in a way that prevents accurate classification by the CNN or multistep classifier.

B. FUTURE WORK

This thesis focused on developing and refining techniques to create an accurate classifier of tactical equipment in SAR images based on a limited training dataset. It did not compare the model developed and tested to the current accuracy of a trained imagery analyst. A future thesis could be written on this topic by comparing the time and accuracy of an imagery analyst working alone to classify a number of targets to the model and to an imagery analyst who is aided by an image classifier. As the Marine Corps does not currently have a standard for imagery analyst accuracy this research would also help to establish a baseline for expert level performance that can be compared to future improvements in machine learning for remote sensing applications.

The MSTAR dataset also was not collected using current DOD remote sensors. Future research could be conducted under conditions that more closely mirror the operational environment in which a fully developed model would be fielded. This includes target detection in a complete scene, a task that was not required due to the format of the MSTAR dataset where the target appears centered in a relatively small image. This thesis does, however, provide a technical demonstration that transfer learning from one modality to another has potential to make the development of accurate models with limited training data sets easier.

Further work could also be done to explore a kNN classifier as a replacement for the classification output of a neural network. kNN performance when trained on features extracted from mid-level convolutional layers and from other neural network architectures could be studied for SAR images or in other classification tasks. The optimal size for vectorized images for input to a kNN classifier could also prove a fertile area of study as the distance measurements used in the kNN algorithm become less computationally expensive with smaller vectors.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- Al Mufti, M., Al Hadhrami, E., Taha, B., & Werghi, N. (2018). Automatic target recognition in SAR images: Comparison between pre-trained CNNs in a transfer learning based approach. *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 160–164. <https://doi.org/10.1109/ICAIBD.2018.8396186>
- Bryant, M., & Garber, F. (1999). SVM classifier applied to the MSTAR public data set. *Proc. SPIE, 3721, Algorithms for Synthetic Aperture Radar Imagery VI*. <https://doi.org/10.1117/12.357652>
- Chen, S., & Wang, H. (2014). SAR target recognition based on deep learning. *2014 International Conference on Data Science and Advanced Analytics (DSAA)*, 541–547. <https://doi.org/10.1109/DSAA.2014.7058124>
- Demšar, J., Zupan, B., Leban, G., & Curk, T. (2004). Orange: from experimental machine learning to interactive data mining. In JF. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), *Knowledge discovery in databases: PKDD 2004* (vol. 3202, pp. 537–539). Springer. https://doi.org/10.1007/978-3-540-30116-5_58
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
- Gandhi, R. (2018, June 7). *Support vector machine—Introduction to machine learning algorithms*. Towards Data Science. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
- Google (n.d). [Kirtland Air Force Base static display]. (n.d.). Retrieved April 14, 2020, from <https://earth.google.com/web/@35.05470595,-106.59537596,1623.90184995a,214.4694424d,35y,-142.92018984h,31.60149328t,0r>
- Kang, C., & He, C. (2016). SAR image classification based on the multi-layer network and transfer learning of mid-level representations. *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 1146–1149. <https://doi.org/10.1109/IGARSS.2016.7729290>
- Leahy, T. (1994). [Overhead shot of a Kuwait Army BMP-2 infantry fighting vehicle taking part in Operation VIGILANT WARRIOR in Kuwait]. National Archives. <https://catalog.archives.gov/id/6496793>

- Malmgren-Hansen, D., Kusk, A., Dall, J., Nielsen, A., Enghold, R., & Skriver, H. (2017). Improving SAR automatic target recognition models with transfer learning from simulated data. *IEEE Geoscience and Remote Sensing Letters*, 14(9), 1484–1488. <https://doi.org/10.1109/LGRS.2017.2717486>
- Morgan, D. (2015). Deep convolutional neural networks for ATR from SAR imagery. *Proc. SPIE, 9475, Algorithms for Synthetic Aperture Radar Imagery XXII*. <https://doi.org/10.1117/12.2176558>
- Nielsen, M. (2015). *Neural Networks and Deep Learning*. Determination Press.
- Novak, L. M. (2000). State-of-the-art of SAR automatic target recognition. *Record of the IEEE 2000 International Radar Conference*, 836–843. <https://doi.org/10.1109/RADAR.2000.851944>
- Novak, L. M., Owirka, G. J., Brower, W. S., & Weaver, A. L. (1997). The automatic target-recognition system in SAIP. *The Lincoln Laboratory Journal*, 10(2), 187–202.
- O’Sullivan, J. A., DeVore, M. D., Kedia, V., & Millier, M. I. (2001). SAR ATR performance using a conditionally Gaussian model. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1), 91–108. <https://doi.org/10.1109/7.913670>
- Pan, S., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359. <https://doi.org/10.1109/TKDE.2009.191>
- Rice, K. (2018). *Convolutional neural networks for detection and classification of maritime vessels in electro-optical satellite imagery* [Master’s thesis, Naval Postgraduate School]. NPS Archive: <http://hdl.handle.net/10945/61255>
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Kholsa, A., Bernstein, M., Berg, A., & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115, 211–252. <https://doi.org/10.1007/s11263-015-0816-y>
- Sandia National Laboratories. (n.d.). [SAR Image of Kirtland Airforce Base Static Display]. <https://www.sandia.gov/radar/index.html>
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations 2015*. <https://arxiv.org/abs/1409.1556>
- Skolnik, M. (1981). *Introduction to radar systems* (2nd ed.). McGraw-Hill, Inc.

- Stewart, M. (2019, February 26). *Simple introduction to convolutional neural networks*. Towards Data Science. <https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac>
- Tang, Y. (2013). Deep learning using linear support vector machines. *2013 ICML Challenges in Representation Learning*. <https://arxiv.org/abs/1306.0239>
- Theobald, O. (2017). *Machine learning for absolute beginners*. Scatterplot Press.
- Torin. (2008). [BTR70 in Nizhny Novgorod]. Wikimedia.org. https://commons.wikimedia.org/wiki/File:BTR70_002.jpg
- United States Air Force. (n.d.). [Grumman HU-16B Albatross at the National Museum of the United States Air Force]. <https://www.nationalmuseum.af.mil/Upcoming/Photos/igphoto/2000574422/>
- United States Air Force. (1996). *Moving and stationary target acquisition and recognition public dataset* [Data set]. United States Air Force. <https://www.sdms.af.mil/index.php?collection=mstar>
- Wang, H., Chen, S., Xu, F., & Jin, Y.-Q. (2015). *Application of deep-learning algorithms to MSTAR data*. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS).
- Xavier Glorot, Antoine Bordes, & Yoshua Bengio. (2011). Deep sparse rectifier neural networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR*, 315–323. <http://proceedings.mlr.press/v15/glorot11a.html>
- Yiu, T. (2019, June 12). *Understanding random forest: how the algorithm works and why it is so effective*. Towards Data Science. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2#f3c8>
- Zhao, Q., & Principe, J. (2001). Support vector machines for SAR automatic target recognition. *IEEE Transactions on Aerospace and Electronic Systems*, 37(2), 643–654.

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