



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**SMALL SATELLITE SENSOR AND PROCESSING
ANALYSIS FOR MARITIME DOMAIN AWARENESS**

by

Jeremy A. McGowan

June 2018

Thesis Advisor:

Charles M. Racoosin

Co-Advisor:

Wenschel D. Lan

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 2018	3. REPORT TYPE AND DATES COVERED Master's thesis		
4. TITLE AND SUBTITLE SMALL SATELLITE SENSOR AND PROCESSING ANALYSIS FOR MARITIME DOMAIN AWARENESS			5. FUNDING NUMBERS W8A83	
6. AUTHOR(S) Jeremy A. McGowan				
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) The potential feasibility for on-board image processing on small satellites was investigated to meet rapid revisit requirements for Maritime Domain Awareness (MDA). Hardware and software solutions for on-board processing were explored. Resolution requirements for use of satellite imagery to both determine the type and class of a ship were investigated. Current small satellite imaging sensor technology was discussed to meet this resolution requirement. Current unclassified ship detection and classification software developed by Space and Naval Warfare Systems Command (SPAWAR) was researched and discussed as a possible solution and also to gain an understanding of current image processing software capabilities and shortcomings for identifying and classifying ships. Consequently, the state of the art for satellite digital image processing was investigated along with computer vision and machine learning techniques. MATLAB was used to simulate the effectiveness of current forms of these techniques on high-resolution satellite imagery for certain ship classes. Processing hardware and artificial intelligence image processing algorithms for detecting and classifying ships within images were found not yet suited to full automation and hosting on a small satellite. Further research and development efforts are needed in this area.				
14. SUBJECT TERMS digital image processing, small satellites, artificial intelligence, computer vision, machine learning, GPU, TPU, image compression, algorithms, image sensor, deep neural networks, convolutional neural networks, support vector machines			15. NUMBER OF PAGES 103	
			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.

**SMALL SATELLITE SENSOR AND PROCESSING ANALYSIS FOR
MARITIME DOMAIN AWARENESS**

Jeremy A. McGowan
Lieutenant, United States Navy
BS, University of California - Davis, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

**NAVAL POSTGRADUATE SCHOOL
June 2018**

Approved by: Charles M. Racoosin
Advisor

Wenschel D. Lan
Co-Advisor

James H. Newman
Chair, Department of Space Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The potential feasibility for on-board image processing on small satellites was investigated to meet rapid revisit requirements for Maritime Domain Awareness (MDA). Hardware and software solutions for on-board processing were explored. Resolution requirements for use of satellite imagery to both determine the type and class of a ship were investigated. Current small satellite imaging sensor technology was discussed to meet this resolution requirement. Current unclassified ship detection and classification software developed by Space and Naval Warfare Systems Command (SPAWAR) was researched and discussed as a possible solution and also to gain an understanding of current image processing software capabilities and shortcomings for identifying and classifying ships. Consequently, the state of the art for satellite digital image processing was investigated along with computer vision and machine learning techniques. MATLAB was used to simulate the effectiveness of current forms of these techniques on high-resolution satellite imagery for certain ship classes. Processing hardware and artificial intelligence image processing algorithms for detecting and classifying ships within images were found not yet suited to full automation and hosting on a small satellite. Further research and development efforts are needed in this area.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	BACKGROUND	1
B.	THESIS ORGANIZATION.....	2
C.	SENSOR IMAGING TECHNOLOGIES OF INTEREST	3
D.	SMALLSATS, CUBESATS, AND INDUSTRY PROCESSING HARDWARE DEVELOPMENTS.....	9
E.	RESOLUTION ANALYSIS FOR SHIP DETECTION, IDENTIFICATION, AND ANALYSIS	19
II.	SATELLITE IMAGE PROCESSING INDUSTRY OVERVIEW	21
A.	BACKGROUND	21
B.	PROCESSING SYNOPSIS.....	21
C.	FPGA TECHNOLOGY IN SOFTWARE-DEFINED RADIOS FOR SMALLSATS.....	22
D.	IMAGE PROCESSING INDUSTRY SOFTWARE PROGRAMS	23
	1. RAPIER SDS for Image Processing.....	23
	2. DigitalGlobe Atmospheric Compensation (AComp)	31
E.	PROCESSING MODE OPTIONS AND ANALYSIS	33
III.	ARTIFICIAL INTELLIGENCE AND DIGITAL IMAGE PROCESSING FOR SMALL SATELLITE MARITIME DOMAIN AWARENESS	37
A.	BACKGROUND	37
B.	COMPUTER VISION AND MACHINE LEARNING	38
C.	IMAGE SEGMENTATION	41
D.	CONVOLUTIONAL NEURAL NETWORKS, SUPPORT VECTOR MACHINES AND DEEP LEARNING	43
E.	SUPPORT VECTOR MACHINES.....	48
F.	FEATURE ENGINEERING, EXTRACTION, AND SELECTION FOR MACHINE LEARNING	50
IV.	MDA CLASSIFICATION MODELING IN MATLAB, LIMITATIONS, AND CONSIDERATIONS.....	53
A.	INTRODUCTION AND SCOPE	53
B.	MATLAB SIMULATION SETUP.....	53
C.	MATLAB RESULTS.....	55
	1. MATLAB Image Processing and Feature Analysis.....	55

2.	Alexnet, Resnet, Inception-v3, and VGG-16 Comparisons and SVM Tests	61
V.	CONCLUSION	67
A.	SIGNIFICANT CONTRIBUTIONS.....	67
B.	RECOMMENDATIONS FOR FUTURE WORK.....	68
	APPENDIX A. SATELLITE SENSOR FIELD OF REGARD CALCULATION.....	69
	APPENDIX B. SUPPORT VECTOR MATH	73
	APPENDIX C. MATLAB IMAGE THESIS PROCESSING	79
	LIST OF REFERENCES	81
	INITIAL DISTRIBUTION LIST	87

LIST OF FIGURES

Figure 1.	Satellite geometry for solving for “X” in terms of “R” and “ Θ ”	5
Figure 2.	Illustration of different satellite sensor types. Source: [6].	6
Figure 3.	SAR image of busy port. Source: [8].	8
Figure 4.	Nvidia Tegra X1 system-on-chip. Source: [11].	11
Figure 5.	Relative performance/watt comparison for GPU, CPU, and TPU. Source: [20].	13
Figure 6.	Harris Corporation SpaceView imaging payloads graphical depiction. Source [25].	15
Figure 7.	Nadir GSD (m) vs. satellite altitude (km) for Harris SmallSat EO imaging payload for staring mode. Source: [24].	17
Figure 8.	Off-nadir GSD geometry. Source: [26].	17
Figure 9.	DigitalGlobe visible band imagery (left) vs. SWIR band imagery (right). Source: [27].	18
Figure 10.	DigitalGlobe imagery: one-meter quality imagery (IKONOS) from Naval Station Norfolk vs. half-meter quality imagery (Worldview-3) at San Diego naval base (right). Source: [33], [34], respectively.	20
Figure 11.	RAPIER SDS tip sheet output. Source: [40].	26
Figure 12.	RAPIER SDS detecting ships in an image. Source: [40].	27
Figure 13.	RAPIER SDS detection options command window. Source: [42].	28
Figure 14.	Haze moving over the East China Sea in NASA satellite imagery from February 25, 2004. Source: [47].	32
Figure 15.	Artificial intelligence categories and subcategories. Source: [49].	38
Figure 16.	Illustration of classification using nearest class mean. Source: [52].	40
Figure 17.	3 x 3 mask for feature extraction	42
Figure 18.	3 x 3 mask for horizontal line detection.	43
Figure 19.	Basic neural network. Source: [55].	45

Figure 20.	CNN pooling stage illustration. Source: [51].	47
Figure 21.	Support vector machines hyperplane and margin separation construction. Source: [56]	50
Figure 22.	Color (left) and gray-level (right) image of LCS docked at San Diego Naval Base. Source: [34], adapted from [34], respectively.	55
Figure 23.	Gray level histogram for Figure 23	56
Figure 24.	MATLAB image transformation from RGB image to binary grayscale. Source: [60], adapted from [60], respectively.	57
Figure 25.	Black and white image of LCS docked at San Diego naval base using threshold of 0.4 (left) vs. using threshold computed from graythresh function (right). Adapted from [34].	57
Figure 26.	Naval Station Norfolk DigitalGlobe snapshot of carrier in port. Source: [34]	58
Figure 27.	A cropped image of carrier in port at Naval Station Norfolk. Source: [34]	59
Figure 28.	Illustration of why it is better to crop an image before resizing for inputting into a CNN. Source: [34]	59
Figure 29.	Independence Class LCS docked at San Diego naval base. Source: [34]	60
Figure 30.	Multi-class classification (one-decision making step)	61
Figure 31.	SVM trained in MATLAB apps window training graph for ship width (ft) vs ship length (ft)	64
Figure 32.	SVM trained for various model types	65
Figure 33.	Image for FOR calculation with coordinate system used	70
Figure 34.	Support Vector Idea. Source: [56]	74

LIST OF TABLES

Table 1.	Ship image dimensions	54
Table 2.	Coding for the position of ship superstructure in MATLAB.....	61

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AFRL	Air Force Research Laboratory
AOD	Aerosol Optical Depth
AI	Artificial Intelligence
AIS	Automatic Identification System
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
CCD	Charge-Coupled Device
CNN	Convolutional Neural Network
COSMIAC	Configurable Space Microsystems Innovations and Applications Center (University of New Mexico)
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
DNN	Deep Neural Network
DoD	Department of Defense
DSP	Digital Signal Processor
EO	Electro-Optical
GOTS	Government Off-The-Shelf
GPU	Graphics Processing Unit
GSD	Ground Separation Distance
FOR	Field of Regard
FPGA	Field Programmable Gate Array
HTML	Hypertext Markup Language
I/O	Input/Output
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
KML	Keyhole Markup Language
KMZ	Keyhole Markup Language Zipped
MDA	Maritime Domain Awareness
MWIR	Medium-Wave Infrared
NIIRS	National Image Interpretability Rating

OTH-G	Over-the-Horizon Gold
SAR	Synthetic Aperture Radar
SATA	Serial Advanced Technology Attachment
SoC	System on a Chip
SRAM	Static Random-Access Memory
SVM	Support Vector Machine
SWIR	Short-Wave Infrared
TCPED	Tasking, Collection, Processing, Exploitation, and Dissemination
TOA	Top-of-Atmosphere
TPU	Tensor Processing Unit
VMOC	Virtual Missions Operations Center

I. INTRODUCTION

A. BACKGROUND

The subject of maritime domain awareness (MDA) is a broad concept and is defined as the “effective understanding of anything associated with the maritime domain that could impact the security, safety, economy, or environment of the United States” [1]. In general, the volume of maritime assets has significantly increased while the data processing capability to gather and analyze this information has increased only slightly [2].

The need for increased maritime domain awareness is ever-present, as in such incidents reported in the press that countries, including Russia, have turned off the automatic identification system (AIS) on their ships to continue shipping banned items to states such as North Korea. Some of these exchanges can be rather brief and last for minutes or even under a minute. AIS is an identification system ships used to transmit ship and navigational information on VHF frequencies [3]. Some of the information includes the type of ship, ship’s position, course over ground, ship destination and estimated time of arrival, and short safety related messages [3]. Therefore, it creates a maritime situational awareness picture through an AIS electronic display. This display shows all AIS equipped ships in their respective geographic positions.

Additionally, the ability to have a persistent MDA framework in such areas in the Pacific Ocean is driving great interest in exploring aerial and perhaps satellite solutions to collect information on so-called “dark targets.” Dark targets are ships that are not transmitting on AIS. Although AIS and Satellite AIS (S-AIS) have provided greater MDA, several issues such as gaps in coverage, data latency, errors in track association, the rise of AIS spoofing and the fact that a majority of small vessels are not equipped with AIS transponders have highlighted the need for a solution that is persistent.

Possible solutions to increasing persistent MDA not dependent on AIS are varied, increasing, and synergistic. Increasing technical capabilities in electro-optical (EO) satellite sensor technology, current trends in big data analytics, and better detection

algorithms to name a few, show great promise in helping to solve the issue. A cursory analysis of current trends in satellite launch reveals that there has been a substantial rise in the deployment of SmallSats in low earth orbit (LEO). The key benefit of such substantial rise in SmallSats in LEO will be to increase timeliness and ultimately yield near-continuous and tactically relevant coverage of areas of interest. This significant increase in imagery collects has created a wave of big data to process. The principal issue as a result is how to take the vast amounts of raw data, which could be raw images this thesis is concerned with, and extract and deliver information such as ship detections and classifications to ground stations and users [2].

B. THESIS ORGANIZATION

This thesis is organized first to give an overview of satellite EO sensor fundamentals and then discusses SmallSat image processing hardware and the accompanying software that makes more robust image processing on board a SmallSat possible. Because of the complexity demanded of a complete on board image processing solution that can intelligently identify objects within EO sensor imagery, certain graphics processing accelerator chips will be discussed to meet this requirement. This hardware is able to speed up the processing of images whereby certain objects within the image are detected and identified. The aim of performing this processing on board the satellite is to relieve the need to do such processing by way of a ground station. A second-order effect of this is to also minimize the total time spent from processing the image to using the image information extracted to make MDA decisions based on this processing loop.

Additionally, an overview is given of a particular product line of SmallSat EO imaging sensors that can satisfy the resolution requirement needed to allow such a SmallSat imaging processing machine to perform classification of sea vessels. A brief overview of imagery resolution grades is then detailed in support of this idea. The second chapter focuses on the most developed and tested current hardware and software that can be implemented in SmallSat image processing. First, a certain type of hardware is discussed, then certain software programs to achieve ship detection and identification are presented and discussed. Lastly, in light of this, certain processing modes that address the

amount of on-board satellite image processing performed versus ground processing are proposed. The third chapter in this thesis will discuss further the particulars of the algorithms and machine intelligence needed on board a SmallSat to meet the MDA classification capability for sea vessels of interest. An important basic algorithm in image processing is detecting lines. By detecting lines, shapes and objects can be detected geometrically. Any type of a distinct feature of an object could also be an identifying marker. The fourth chapter deals with the limits of classification algorithms and proposes a general approach and recommendations for dealing with current shortcomings in MDA classification ability. A ship classifier is built in MATLAB of limited ship identification extent to demonstrate a level of abstraction whose general idea could be expanded further to a larger ship database to allow for general classification of many more ship classes. Processing hardware and artificial intelligence image processing algorithms for detecting and classifying ships within images were found not yet fully suited to full automation and hosting on a SmallSat.

C. SENSOR IMAGING TECHNOLOGIES OF INTEREST

The major types of sensor technologies for remote sensing by satellites can be broken into electro-optical, thermal infrared, radar detection and ranging, and light detection and ranging. This thesis will be primarily concerned with analyzing EO imagery solutions and, to a much less extent, Synthetic Aperture Radar (SAR) solutions. For optical imaging, magnification is defined as the ratio of the image size to the object size. We typically use resolution rather than magnification. An important physical principle relevant to EO remote sensing is the wave character of light. The wave characteristics of light which include wavelength and frequency, determine how the light interacts with particulates in air. Certain wavelengths of light scatter, reflect, and transmit differently through matter, and so what a corresponding imaging sensor receives from different wavelength light can affect image quality. A property of a wave including a water wave is that when it hits a sharp solid corner, it manages to spread its energy around that corner. Likewise, light behaves the same way and can diffuse around such sharp boundaries [4]. This can be verified from everyday experience as one turns on a light in a room and how the amount of lightening around a hallway corner is changed. By

way of studying a single-slit diffraction pattern while considering a circular aperture or slit, the Raleigh criterion can be obtained as equation 1, which is as follows [4, eq. (1)]:

$$\Delta\theta = 1.22 * \lambda/\text{diameter} \quad (1)$$

where λ is wavelength and $\Delta\theta$ is the angular resolution. This result, together with the following geometrical analysis, will lead to the resolution equation for a circular aperture for an EO satellite, assuming an isosceles triangle with its base representing a resolution distance x , with the angle extending the equal sides being “ θ ” and the height of the triangle being range or “ R ,” $\tan(\theta/2) = x/2R$. Resolution quality for an EO sensor is important in obtaining the ability to resolve objects that may be close in an image. For MDA, this means more details can be seen. Assuming “ θ ” is close to zero, it can be shown $\theta \sim x/R$ as we can take the first term in the infinite series for the tangent function in this case. This is reasonable to assume for most remote sensing systems as the range is significantly greater than the resolution. Figure 1 gives a visual depiction of this geometry. Substituting “ x ” for the ground separation distance (GSD) and the right side of the Rayleigh equation for θ yield equation 2, which is true for a nadir look angle for a satellite [4, eq. (2)]:

$$\text{GSD} = 1.22 * (\lambda/\text{diameter}) * R \quad (2)$$

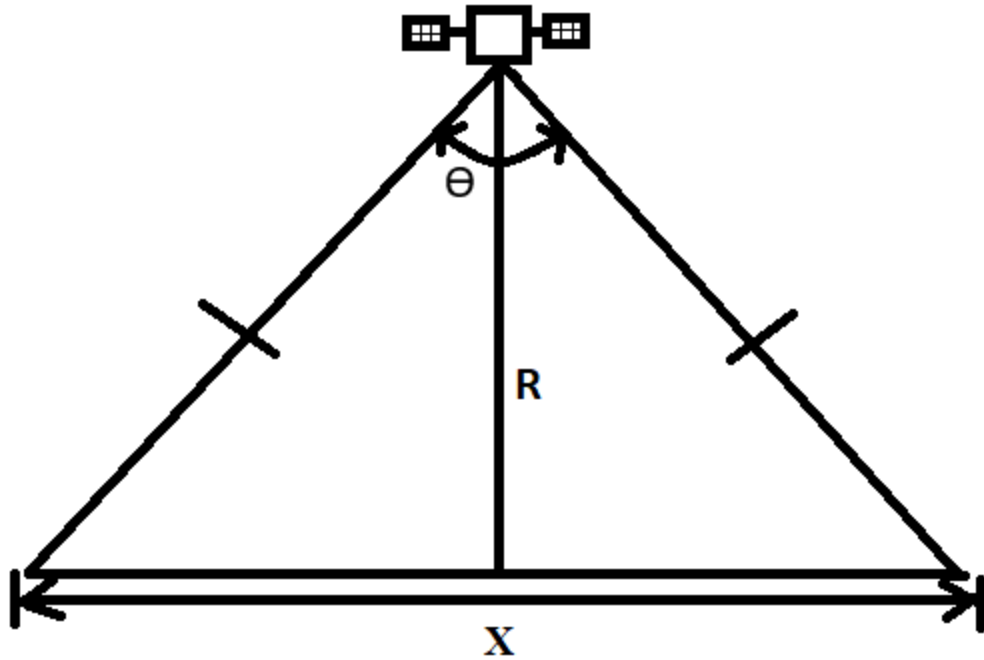


Figure 1. Satellite geometry for solving for “X” in terms of “R” and “ Θ ”

Additionally, optical sensors can be divided into three types. Those types are framing sensors, pushbroom scanners, and whiskbroom scanners. A framing system will snap an image and that image will be stored on film or charge-coupled device (CCD). Skybox Imaging’s SkySat-1 was the first one-meter resolution system to use a framing focal plane, meaning it had the ability to resolve ground features separated in distance by one meter or greater [4]. A cross-track imaging system will utilize a motor which turns a scanning mirror to implement a cross-track scanner or whiskbroom system [5]. The output of the scanning mirror would then feed into a detector [5]. The last imaging scanner system to be considered is an along-track scanner or pushbroom scanner. In this type of imaging system, the pixels in a linear detector for the satellite payload map onto a ground strip that comprises the full instantaneous field of view [5]. The forward motion of the satellite then images strip by strip, hence the name pushbroom [5]. These various sensors can be seen in Figure 2. A derivation for a pointing sensor’s field of regard (FOR) is detailed in Appendix A. This is the maximum possible area that a pointing

sensor could image based on the satellite's altitude and its maximum off-nadir pointing angle.

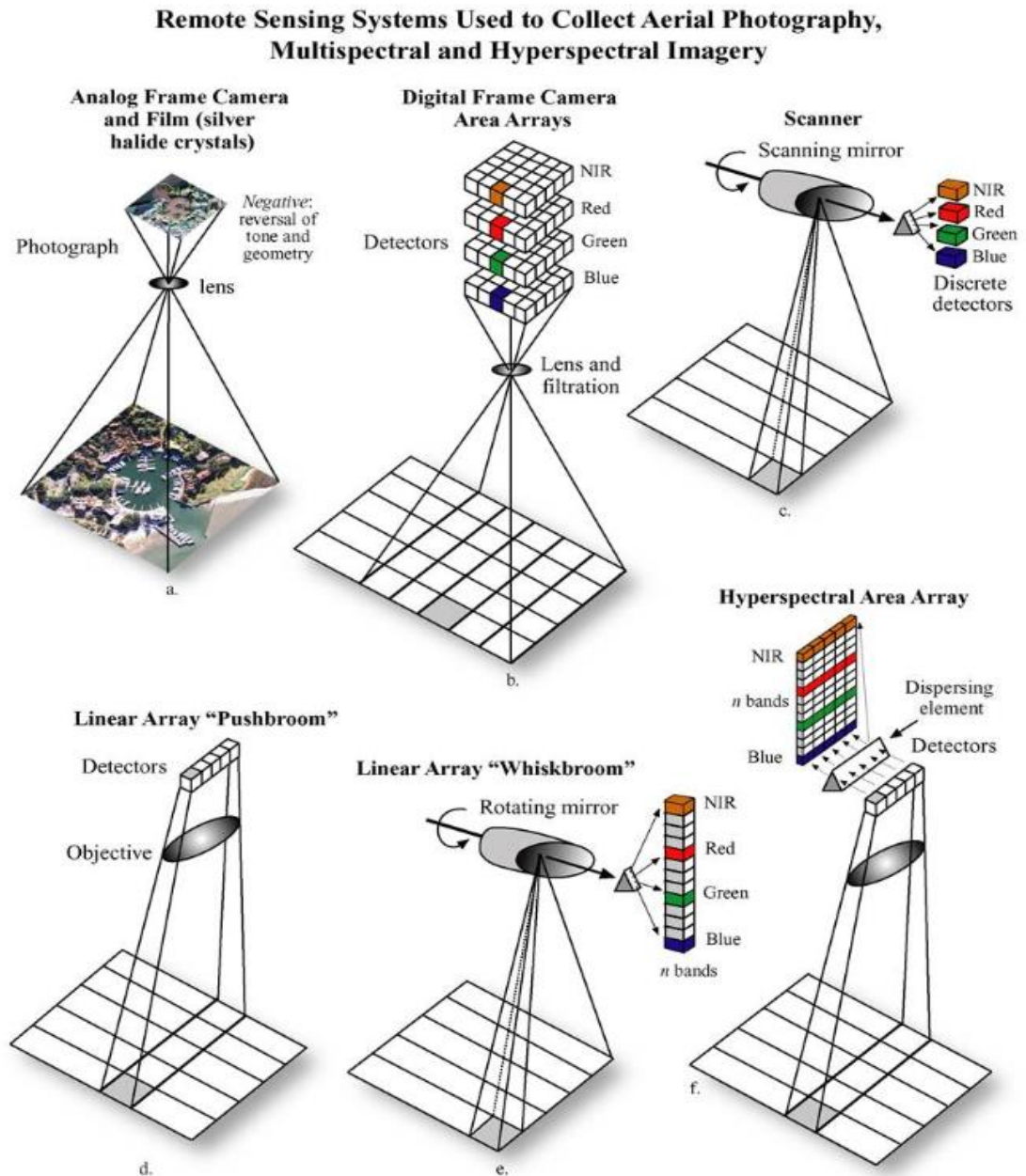


Figure 2. Illustration of different satellite sensor types. Source: [6].

For MDA, though, various factors such as weather and time of day make EO imaging by satellites less efficient. Therefore, for a complete around-the-clock and all weather imaging constellation of small satellites, synthetic aperture radar (SAR) becomes useful for satellite imagery as EO sensors rely on the reflectance of light from the surface of the Earth to work as well as the assumption of limited cloud cover to work. Clouds scatter light, thereby interfering with EO remote satellite sensors. However, to a certain degree during the day algorithms to process EO imagery can greatly improve the usability of the imagery for MDA. A more robust network of sensors for MDA may consider using SAR in addition to EO, but the primary focus of this thesis is achieving MDA through EO.

Nonetheless, SAR is an active instrument, generates its own illumination which is coherent (all the light is in phase and thus is anti-dispersive), and the return signal contains both phase and amplitude, which is useful in applications such as topographic mapping and coherent change detection. SAR employs a short physical antenna along with data recording and processing techniques to simulate a longer antenna, which results in a narrow beamwidth without requiring a short operating wavelength or a high frequency.

For ship detection using SAR, one to three meters is widely accepted by experts as necessary for general detection with one meter being the transition into identification for the ship. At one to three meters, major geometric structures of most ships are discernible while at one meter and below, specific features of ships are discernible enough to permit some degree of classification. The main application of using SAR is for detection, and SAR has unique features in a MDA setting. In particular, the “main feature of ships in SAR images is a bright backscattering,” with the backscattering signal from the sea generally being lower, since electromagnetic radiation incident on the sea is generally directly opposite in direction [7]. This has the result that ships show up as bright spots in SAR images [7]. That is, SAR imagery suffers from speckle noise, which is a result of multiple coherent reflections from ground objects and material [8]. Two types of reflection can result in MDA due to sea state. Specular reflection results from a calm sea state as the sea presents a mirror-like surface for the incident radar energy [8]. A

rough sea state causes the incident radar energy to be scattered in all directions [8]. Corner reflection for incident radar energy off ships causes ships to appear bright in SAR imagery [8]. This can be seen from Figure 3. Therefore, ships in a relatively calm sea state are easily detectable as bright spots amongst a dark sea [8].

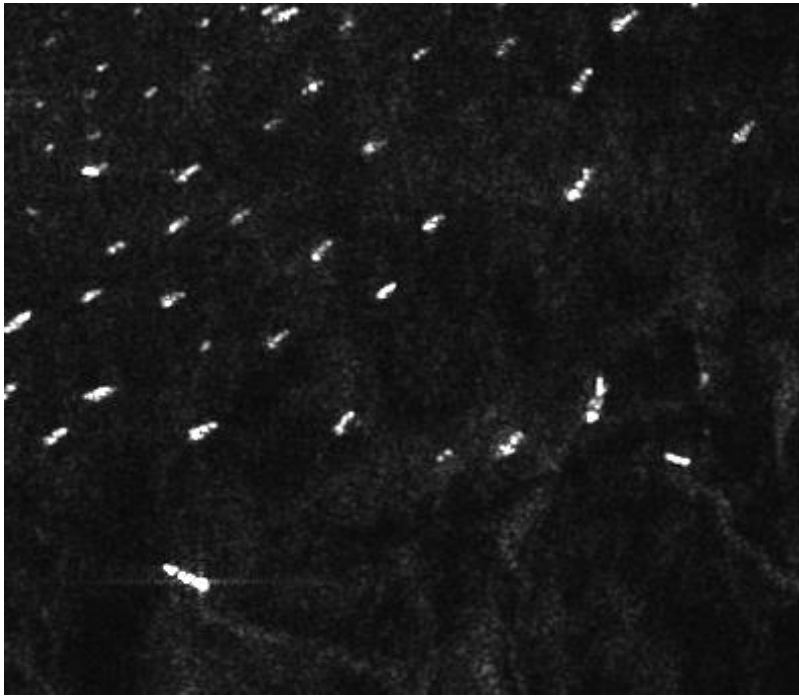


Figure 3. SAR image of busy port. Source: [8].

Analysis of SAR imagery is also quite useful in wake detection and therefore ship detection along with various measurements for the ships such as length, location, speed, heading, and wake type. Using any of these defining characteristics can lead to further specificity on the type of ship that is being imaged. However, SAR in its native format is less useful for classifying maritime assets such as ships, but can be made to appear like electro-optical imagery according to the Advanced Analysis Systems Branch at Space and Naval Warfare (SPAWAR) Systems Center Pacific [9]. This distortion for SAR imagery can as previously stated be attributed to speckle noise but image processing algorithms exist to clean the imagery up. Yet, SAR imaging remains a gold standard for ship detection because of its all-weather and around-the-clock capabilities.

D. SMALLSATS, CUBESATS, AND INDUSTRY PROCESSING HARDWARE DEVELOPMENTS

SmallSats have a mass less than 180 kg. This category is further divided into subcategories: minisatellites at 100–180 kg, microsatellites at 10–100 kg, nanosatellites at 1–10 kg, picosatellites at 0.01-1 kg, and femto satellites at 0.001-0.01 kg [10]. Further, CubeSats are a class of nanosatellites that use a standard size and form factor and were originally developed in 1999 to provide a platform for education and space exploration [10]. The base unit is a 1U CubeSat, which measures 10 cm by 10 cm by 10 cm [10]. Larger sizes are available such as a 6U, which would measure 20 cm by 10 cm by 30 cm [10]. A 12U would then measure 20 cm by 20 cm by 30 cm [10].

Further, the deployment of SmallSats typically occurs in Low Earth Orbit (LEO). The prominent advantages of a LEO orbit for MDA image gathering include increased resolution, higher revisit rates for individual satellites versus being in Medium Earth Orbit (MEO), and shortened signaling paths from LEO to ground versus MEO to ground. However, the lower orbit suffers from some readily apparent disadvantages in providing for persistent MDA. Many SmallSats in numerous orbital planes are needed to provide for persistent MDA in numerous orbital planes.

Additionally, downlinking and processing of MDA data from satellites in such a constellation must be timely and expeditious for real-time actions to be intelligently executed in response to the continuous MDA data updates in areas of interest. Sensors, for example, in a sea-borne area of interest, in close proximity to satellite collections of interest, may need to formulate real-time decisions based upon this chain of remote sensing image collection in proximity to those sensors. Therefore, an important concern for timely actionable intelligence within this construct would be investigating any catalysts to speed up the processing of data in this network. Specifically, an important question is how much on-board image processing can be done on SmallSats versus at ground sites, to speed up this chain of data processing and dissemination ending in action by an end user or sensor. With a robust enough processing capability on board SmallSats, data delivered from SmallSats to ground users can become more real-time actionable as a result of minimizing ground processing requirements. The workflow for ultimately

allowing a ground user to employ satellite imagery intelligence from start to finish is a cycle of tasking, collection, processing, exploitation, and dissemination (TCPED).

With this general overview in mind, the level of detail that SmallSats provide for MDA must be considered. A theoretically more advanced satellite may be able to image objects, process images of those objects, and then produce files on those objects, detailing information such as how long the object is, how wide the object is, what type of object it is (which may depend on length and width), and the heading of the object. All these MDA satellite image processing mechanisms theoretically are possible to integrate together given enough advancement in the state-of-the-art for SmallSat payloads. Additional information may be gained as well from full motion video processing on board, such as object velocity and other motion dependent features inherent with the object or surrounding the object. For the U.S. Navy, developing this sort of system has importance for over-the-horizon targeting.

Currently, there is work to build on-board classification means for SmallSats using artificial intelligence (AI). In particular, a well-known company leading the way into a new world of AI machines is Nvidia. The Air Force Research Laboratory (AFRL) in Albuquerque, New Mexico, has shown interest in using Nvidia chips to usher in a new wave of satellite capability. “The Nvidia Tegra X1 (TX1) is a credit-card size, system-on-a-chip (SoC) that contains an entire suite of processing hardware” from input to output and uses “Nvidia’s graphics processing unit (GPU) architecture and Compute Unified Device architecture (CUDA) parallel computing platform” to provide deep learning [11]. Deep learning is a form of machine learning that uses artificial neural networks (ANNs) to learn features from various forms of visual and written data representations without the use of predefined human analysis methods and direct human interaction with the deep learning framework [12]. It is depicted in Figure 4. Also, software for the SoC is available to perform “image recognition, object detection and location, and image segmentation” [11]. Research in implementing GPU processing on board a SmallSat was started in 2016 by COSMIAC Research Center at the University of New Mexico along with partners Stinger Ghaffarian Technologies Inc. and the AFRL [11].

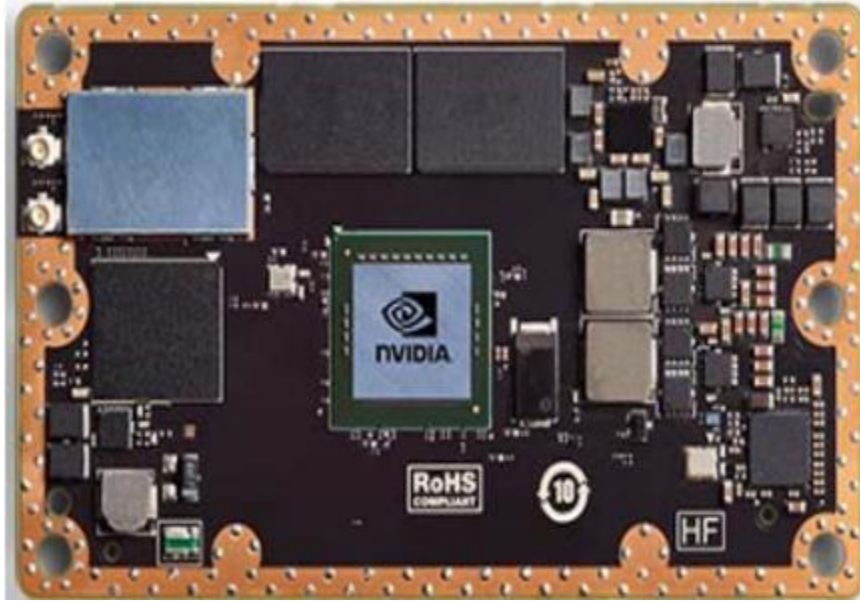


Figure 4. Nvidia Tegra X1 system-on-chip. Source: [11].

The Nvidia TX1 can be mounted on a 7 x 7 inch printed circuit board (PCB) containing the actual I/O device ports [11]. That together was called the Nvidia Jetson TX1 development board [11]. Nvidia's Deep Learning GPU Training System (DIGITS) can be used to train the development board for inference on the ground before it is deployed on a SmallSat [11]. In other words, this board can implement a level of artificial intelligence to perform machine learning and computer vision functions, such as image object classification. Machine learning is a type of learning that allows for machines to learn from experience and is a branch of AI [13]. Computer Vision deals with an AI system gaining a visual understanding through imagery and perception [14]. In fact, the CUDA Deep Neural Network (cuDNN) package can be installed on it [11]. In this respect, this is promising technology that simply needs to be tested and whose proof-of-concept needs to be confirmed.

Using convolutional neural networks (CNNs) such as GoogLeNet and AlexNet and, most recently ResNet, objects in images can be recognized and classified. CNNs are artificial neural networks that are used to detect and classify objects within images. A project called ImageNet "is a large database of images designed for use in visual object recognition software research" [15]. "The ImageNet project holds an annual software

contest called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where software programs compete to correctly classify and detect objects and scenes” [15]. These software programs can be further studied and used as part of an MDA classification scheme for on-board SmallSat image processing. More sophisticated CNNs with better ability to identify objects within images being built could perhaps lead to technology to better enable MDA object identification.

A recent invention by Google Brain, which is the deep learning artificial intelligence research team at Google, promises to greatly increase the performance for processing neural network decisions. The team introduced the Tensor Processing Unit (TPU) in 2016 as a custom Application Specific Integrated Circuit (ASIC) to be used in their data centers to stave off possible data center growth projected because of people using voice search for three minutes per day using speech recognition Deep Neural Networks. Specifically, TPUs accelerate the inference phase of neural networks. The inference phase is the second stage after the neural network has been trained. It is the stage at which the neural network gives answers to tasks after having been trained on a data set. This stage deals with new data in the context of already knowing how similar data maps to certain output. Further, a TPU ASIC consumes 40W when running and, through having being built on 28 nm processing technology, the TPU can be used “as an external accelerator card that fits into a SATA hard disk slot” [16]. The sort of sizing and power requirements would make deployment on a SmallSat feasible. This technology will lead to better hardware for AI to be used for MDA purposes such as object identification. Specifically, TPU technology could be implemented on board SmallSats one day to accelerate the image processing on board. In the meantime, this technology could be used for image processing at ground stations.

Additionally, the software that Google Brain has developed to work with machine intelligence, TensorFlow, is open-source software that uses “data flow graphs for scalable machine learning” [17]. The software for TensorFlow was built using the C++ programming language, but user-interface (UI) applications for TensorFlow can be developed using C++ or Python [18]. Other company AI engines such as Caffe from Nvidia are notable advances in this area as well but “Google’s AI engine is regarded by

some as the world's most advanced" [19]. However, the continued use of all the AI tech Google Inc. has developed and implemented is dependent on Google's willingness to continue sharing its technology. The impetus for making TensorFlow open-source software though is for outside developers to expand the application programming language set to include other languages such as Java, JavaScript, and GoogleGo [18]. In short, implementing TPU technology on-board a SmallSat one day is promising as the Google Brain team found that the "TPU delivered 15–30X higher performance and 30–80X higher performance-per-watt than contemporary CPUs and GPUs" [16]. Figure 5 illustrates the relative performance of the TPU against CPUs as well as GPUs. TPUs are at work helping power better search query performance for Google and products such as Google Image Search, Google Photos and Google Cloud Vision API [20]. For MDA, these performance gains in technological ability could translate into faster and better MDA identification.

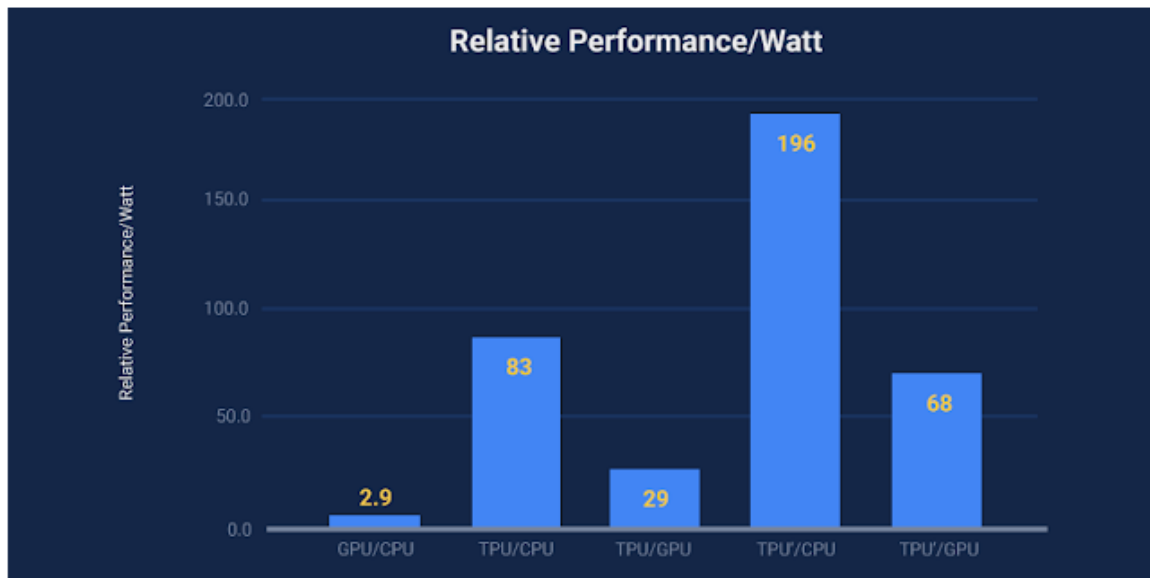


Figure 5. Relative performance/watt comparison for GPU, CPU, and TPU.
Source: [20].

Adding the ability of users to directly task SmallSats creates additional complexity by requiring the processing of user requests in real-time to the satellite

network. Additional software and hardware designed and tested for this purpose would be needed. The reason for doing so would be synergistic in nature with direct-tasking being married with on board image processing. A real-time SmallSat tasking framework would be achieved utilizing increased artificial intelligence on-board for a real-time satellite imagery TCPED cycle. This in turn would enhance the ability of ground units to leverage remote sensing for actionable effects. This ability to task SmallSats in real-time is an object of design for companies such as the Satrec Initiative for the SpaceEye-X system, which is a half-meter resolution remote sensing satellite system weighing 400 kg [21]. In Satrec Initiative's framework, the customer can use "authentication & encryption keys to command imaging and download missions to the satellite" with "the direct tasking function implemented on a space-grade Radiation Tolerant (RT) series Field Programmable Gate Array (FPGA), which is located at the Telemetry and Telecommand (TMTC) module" [21]. The "sequence of direct tasking" for the SpaceEye-X satellite system involves seven steps due to the customer allocation scheme implemented [21]. That is, it is possible to allocate customer tasking requests via the allocation of encryption and authentication keys sent from the mission control station to the customer Ground Station (G/S) [21]. Allowing for a fully automated tasking phase of the TCPED cycle will allow for the time for the TCPED cycle to be reduced to support a more real time delivery capability to end users.

For the commercial SmallSat industry, the last few years have seen the emergence of many startup companies. Companies such as Pumpkin Inc., AAC Microtec, and Adcole Corporation of Maryland offer SmallSat payload and bus components necessary to make SmallSat MDA possible. In particular, U.S. Army Space and Missile Defense Command (US SMDC) has sponsored the Kestrel Eye program for development by Adcole Corporation. By leveraging COTS technology, the total cost per 50 kg SmallSat is less than \$2 million. The sensor aboard a KestrelEye has a 1.5-meter resolution for its Block 2M version. The vision for this satellite is to provide a direct downlink to the warfighter without a need for relaying [22]. The raw image would be returned to the warfighter with a tasking-to-dissemination cycle of ten minutes or less [23].

Harris Corporation offers a line of SmallSat EO payload sensors under its SpaceView product line. The different models available cover satellite classes from NanoSat to MicroSat to SmallSat [24]. The SpaceView 35 or SV-35 is a SmallSat EO payload sensor in the MicroSat class [24]. It is depicted in Figure 6. As a general rule of thumb, a GSD of one meter is necessary to begin to determine the class of a ship, and the SV-35 satisfied this at an altitude of 500 km. The payload mass for the SV-35 is 20 kg – 35 kg and it has an imaging power requirement of 70 W to 100 W [24]. The aperture is .35 m, which is a little over a foot at around 13 inches [24]. The best resolution occurs in the staring mode where the lower bound for GSD for a given altitude is determined [24]. For the SV-35 at 500 km, the GSD at nadir is 0.7 m for staring and 1 m for scanning [24]. A staring sensor's geometry is a two-dimensional layout of image detectors, while a scanning sensor is either a zero-dimensional or one-dimensional layout of detectors as was seen in Figure 2.

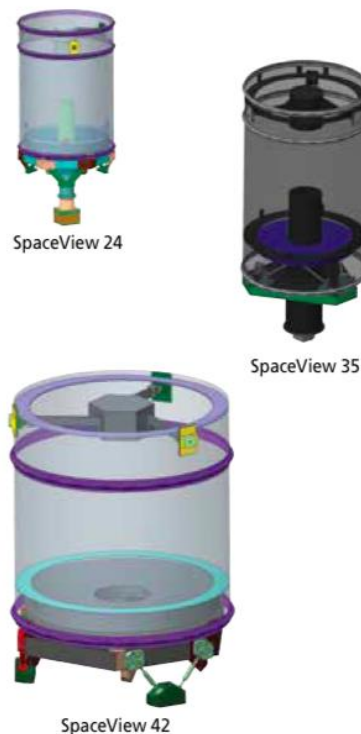


Figure 6. Harris Corporation SpaceView imaging payloads graphical depiction. Source [25].

Figure 7 illustrates the general linear dependence of nadir GSD in meters for the staring sensor as altitude in kilometers is increased. The trend for each model of sensor from SV-24 to SV-70 is linear, which agrees with remote sensing theory. An MDA SmallSat constellation that warrants operation in such a way to allow for ship classification can easily operate at almost all altitudes specified in Figure 7 for the SV-35. Figure 7 is for nadir pointing of the sensor. A sensor that can point off-nadir will have an oblique look at the ground and the GSD will get worse for larger and larger look angles as the distance to the target increases. The look angle for a satellite is the angle from a perpendicular line from the satellite to the center of the Earth and the line from the satellite to the imaging point on the surface of the Earth. Because of this oblique geometry, which is illustrated in Figure 8 where “L” is the look angle and “H” is the distance from the sensor to the off-nadir imaging area, the GSD can be calculated as a rather complicated expression shown in equation 3 as the geometric mean of the projection onto the ground of the x- and y-directions of the sensor footprint [26, eq. (3)]. That equation is based on the geometry of Figure 8, where ISEL “is the angle between the line from the ground target to the satellite and the line tangent to the earth at the ground target” and is known as the “imaging satellite elevation angle” [26]. The angle β is “the angle between the x direction of the digital array and the line perpendicular from the line-of-sight” [26]. The variable H is the slant range (from satellite to ground target) and D is “the diameter of the optics aperture” of the satellite sensor [26]. The variable λ is the wavelength for the optical wave. The variables GSS_{optics_x} and GSS_{optics_y} are the “optic ground spot sizes” in the x and y-directions, which depends on their respective point spread functions in their respective directions [26].

$$GSD_{optics} = \sqrt{(GSS_{optics_x} GSS_{optics_y})} = \lambda \frac{H}{D} \left\{ \left[\cos^2(\beta) + \left(\frac{\sin(\beta)}{\sin(ISEL)} \right)^2 \right] \times \left[\sin^2(\beta) + \left(\frac{\cos(\beta)}{\sin(ISEL)} \right)^2 \right] \right\}^{1/4} \quad (3)$$

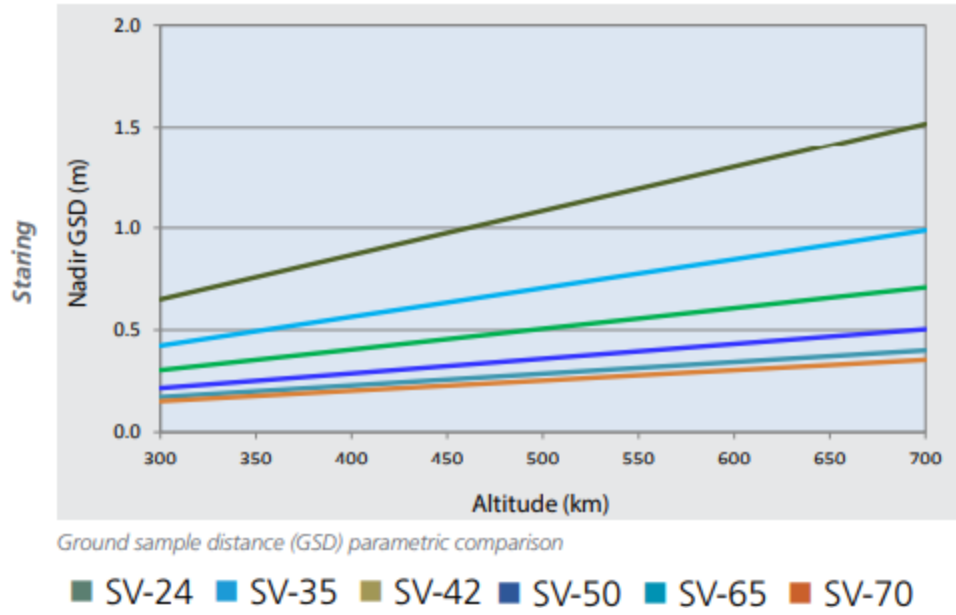


Figure 7. Nadir GSD (m) vs. satellite altitude (km) for Harris SmallSat EO imaging payload for staring mode. Source: [24].

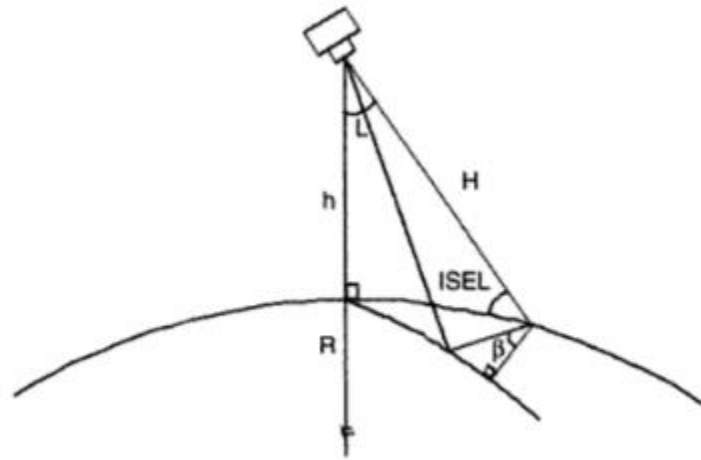


Figure 8. Off-nadir GSD geometry. Source: [26].

For comparison, if better ground imaging resolution is needed the SV-65 imaging payload supports a greater resolution at 0.3 m – 0.45 m with a greater payload weight of 110 kg – 150 kg and bigger aperture at 0.65 m [24]. The payload imaging power is 245 W – 305 W though, which is approximately twice that of SV-35 [24]. Therefore, there is

a tradeoff in using these higher resolution sensors at the cost of a bigger payload and power consumption. The SV-50 imaging payload aperture is 0.5 m with similar increases in payload mass and imaging power, but the advantage of these packages is added Short-wave Infrared/Medium-wave Infrared (SWIR/MWIR) capability [24]. SWIR light is reflective in nature, not visible to the human eyes, and SWIR images are not in color. Figure 9 illustrates how using the SWIR mode can cut through smoke to get a clearer picture of the ground below.

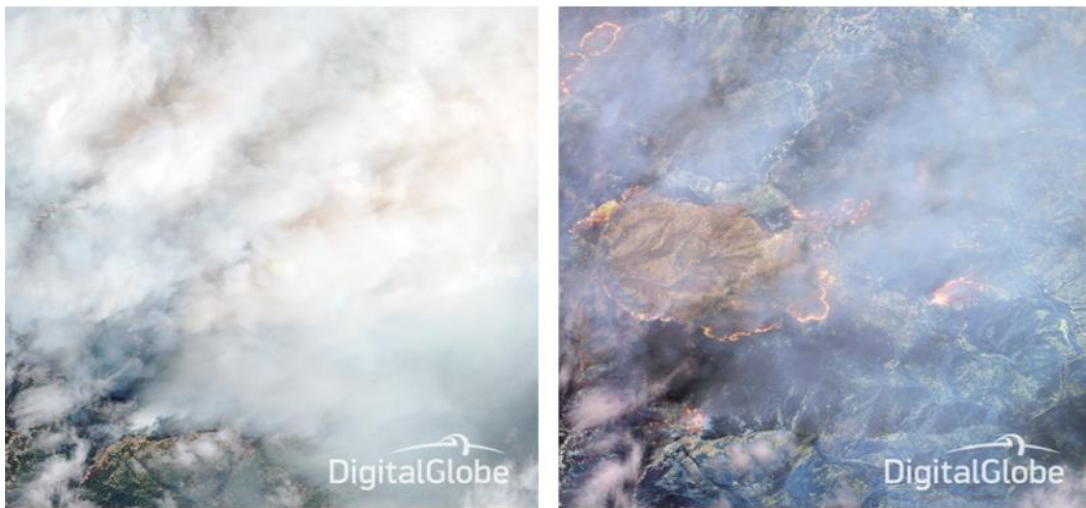


Figure 9. DigitalGlobe visible band imagery (left) vs. SWIR band imagery (right).
Source: [27].

The advantages of SWIR include a high-resolution imaging capability, the ability to provide for imagery in day and night, penetration through thin clouds, covert illumination, the ability to see through smoke, and greater visibility in the presence of atmospheric aerosols. With SWIR imagery for WorldView-3, DigitalGlobe can locate and classify man-made materials and geological minerals on the surface of the Earth using 7.5-meter commercial SWIR resolution. The 3.7-meter resolution for SWIR is reserved for United States government (USG) use. As resolution comes down even more in future years for the USG SWIR product payload, SWIR will become a much more attractive option for typing and classification of maritime assets [28].

A level of detail about whether it may be a warship, cargo ship, oil tanker, or cruise ship is a type determination. To classify a ship is to describe the warship, cargo ship, or oil tanker class within a given type. That is, a warship can be classified as an Arleigh Burke-class destroyer and an oil tanker as a Panamax or Suezmax. The most detailed level of specificity is to determine the class of the ship, which is inherently the hardest to determine from satellite imagery.

Additionally, while going from whiskbroom scanning to pushbroom scanning, the radiometric and spatial resolution improves [29]. However, scanning systems perform less well in this respect compared to staring sensors for all altitudes. This is principally why a staring system would be chosen for MDA object classification purposes. Specifically, the SV-35 will be used as a representative sensor for this study because of fits well within the desired resolution, size, and power constraints. Therefore, the SV-35 is the reasonable and recommended sensor to use in STK modeling of a satellite constellation capable of providing an adequate level of resolution at LEO to allow for classification of notable MDA assets such as ships in general.

E. RESOLUTION ANALYSIS FOR SHIP DETECTION, IDENTIFICATION, AND ANALYSIS

The National Image Interpretability Rating Scale (NIIRS) is used “to define and measure the quality of images and performance of imaging systems” [30]. The NIIRS rating scale ranges from zero to nine, with a rating level of zero being the worst to nine being the best. A rating level of eight (NIIRS 8) is a rather impressive level of detail, where it is possible to identify grille detailing and/or the license plate on a truck, identify windscreen wipers on a vehicle and count individual lambs [30]. This rating of imagery is most likely to originate from low-altitude aerial imagery [31]. A rating level of five (NIIRS 5) for EO imagery allows identification of Cruiser class warships, which tend to be about 173 meters in length and have a beam length of 55 feet [32]. NIIRS 5 falls in the range of 0.75 to 1.2 m for ground resolution [30]. Destroyers tend to have similar dimensions as Cruisers, so sub-meter imagery is generally where ship classification and identification for EO imagery begins to be possible. A veteran image analyst is the reference point for these scales, so an assumption for any automated MDA object

classification is that computerized image analysis would be able to provide a comparable level of identification when moving into a fully automated chain of processing satellite images. Figure 10 illustrates the difference between approximately one-meter and half-meter quality imagery [29]. The biggest noticeable difference when comparing the images in Figure 10 is that the superstructure components for the amphibious assault ship are much more resolved in the picture on the right, allowing for structures within ship structures to be identified. This imagery shows how moving into sub-meter satellite imagery will ideally allow for resolving many details or features for ships, which would be useful in classifying ships.



Figure 10. DigitalGlobe imagery: one-meter quality imagery (IKONOS) from Naval Station Norfolk vs. half-meter quality imagery (Worldview-3) at San Diego naval base (right). Source: [33], [34], respectively.

II. SATELLITE IMAGE PROCESSING INDUSTRY OVERVIEW

A. BACKGROUND

Satellite image processing for MDA ship detection and typing has been in development since the early 2000s. One system, in particular, referred to by the acronym RAPIER, relies heavily on Computer Vision digital image processing techniques to analyze the geometric structure of ships to arrive at the type of ships being captured in a particular EO, Infrared (IR), or SAR satellite image. The use of Field Programmable Gate Array (FPGA) technology has garnered recent attention for use in Software-Defined Radios (SDRs) for small satellite communication systems [35]. FPGA technology is a type of digital technology that can be programmed to implement digital logic to meet one's needs. For image processing, using FPGA SDR technology on board SmallSats would allow for processing upgrades via software upgrade files being sent to the SmallSats. This allows more flexibility for SmallSats to upgrade to new algorithms, for example, which would help with offloading processed data to ground sites and the user to meet MDA needs.

B. PROCESSING SYNOPSIS

To date, satellite imaging payloads mostly operate in a store-and-forward mode. The satellite captures the image of interest, and then when it can establish a connection to a ground site to link to, it sends the data. With an increased data volume for higher spatial resolution imagery needed to identify and classify maritime objects of interest, coupled with the need to utilize the limited onboard resources of small satellites, image compression is an important consideration for payload image processing units to produce smaller downlink files for ground sites and ground users to process and use. This has the additional benefit of lowering the downlink communications bandwidth required. A simpler modulation scheme could then be used to downlink data for a given bit error rate while also lowering power transmission requirements.

For small satellite, downlinking a reduced set of image files and in particular geographic information system (GIS) files, for possible further processing and/or use is

desirable. Some notable examples of GIS file formats would be keyhole markup language (KML), Environmental Systems Research Institute (ESRI) Shapefile (SHP), and Google Earth Placemark File (KMZ) [36]. By reducing the file sizes necessary for transmission from a SmallSat to the ground station or user, transmission bandwidth is more efficiently used, and the complete set of information necessary for MDA is received faster because transmission time is reduced.

C. FPGA TECHNOLOGY IN SOFTWARE-DEFINED RADIOS FOR SMALLSATS

Advances in semiconductor technology relating to high-efficiency gains in SWaP (Size, Weight, and Power), are making small satellite use for MDA increasingly attractive. In particular, the use of Field Programmable Gate Arrays (FPGA) is attractive as their use allows for parallel processing, which reduces the number of clock cycles needed to perform a set of instructions [37]. Parallel processing is a form of computer processing whereby operations and instructions are executed at the same time in parallel by different physical branches in the computer hardware. It is akin to having two people clean a house instead of one. The housework gets done faster. In this case, the functions needed to be accomplished in hardware are done quicker, which makes FPGAs an attractive option for onboard processing for a SmallSat. In particular, Microsemi Corporation offers the ProASIC3 line of FPGA technology. The Military ProASIC3/EL features an advanced reduced instruction set computer machine (ARM) Cortex-M1 soft processor that is fully implemented in the FPGA. It also features firm-error immunity which makes it resistance to strikes from high-energy neutron generated in the upper atmosphere [38]. At LEO orbits, firm error immunity would be applicable and of positive use for on-board processing for SmallSats. This is a problem for non-flash based FPGAs such as static random-access memory (SRAM) FPGAs. Flash is a non-volatile type of memory that holds the configuration patterns of the FPGA and is ready once powered on [37]. Non-flash based technology like SRAM FPGA technology then must use an external source of non-volatile memory to load its configuration. This requires more time to load the configuration and therefore reduces performance. Likewise, an additional benefit of flash-based FPGA technology compared to SRAM FPGA technology is that

since there is no need to load external configuration data at system power-up components such as electrically erasable programmable read-only memory (EEPROM) chips or microcontrollers are not needed [39]. Programming of Microsemi FPGAs is accomplished with Microsemi FlashPro programming system, which is a combination of Microsemi's FlashPro software and hardware programmers [37]. This system supports in-system programming (ISP), which means the FPGAs can be programmed when the complete system or SmallSat is in place [37].

FPGA technology could be used in accelerating the next-generation of deep neural networking software for usage on-board SmallSats. In particular, for convolutional neural networks (CNNs), FPGA technology can implement the most commonly used layers for CNNs, which are convolution layers, pooling layers, rectified linear layers, and inner product or fully connected layers.

D. IMAGE PROCESSING INDUSTRY SOFTWARE PROGRAMS

The main program of interest to be discussed is a software program called rapid image exploitation resource (RAPIER) ship detection system (SDS). Additionally, a program called AComp developed by DigitalGlobe is worth mentioning as it addresses some algorithms to process daytime images that may even be unusable in their raw format. RAPIER SDS is a good case study in determining the current technological capacity to detect ships and determine types of ships from high-resolution EO, SAR, and IR satellite imagery.

1. RAPIER SDS for Image Processing

a. RAPIER SDS Overview

A software program of interest developed by SPAWAR Space and Naval Warfare Systems Command in San Diego shows some promise in processing EO/IR imagery as well as SAR imagery from satellites. SPAWAR's RAPIER SDS is a government off-the-shelf (GOTS) ship detection system that was built upon more than "one million lines of open source code" along with a "\$25 million prior investment" [40]. The project dates back to 2001 when the program code length was much less. The source code is primarily

C++ and Python and a minority of C, XML, and other programming languages [40]. RAPIER SDS uses a suite of advanced image processing algorithms to process an image's EO panchromatic data and IR data as well as SAR data to measure and determine the type of ship from high-resolution commercial satellite imagery [41]. The algorithms perform land masking, the detection and removal of clouds, and glint and noise removal as mitigating techniques to improve detection of probably ships in an image [40].

In addition, RAPIER SDS is a command-line application tool [9]. RAPIER SDS takes an image with ships in it and is able to detect theoretically as many ships as the user may want. However, 10,000 ship detections may be the program limit [9]. It primarily employs methods of Computer Vision along with Fourier Transforms to extract geometric and photometric features from satellite imagery. Algorithms can be developed to detect roads, vehicles, airplanes, and people on the ground. RAPIER SDS also possesses a DoD 5000 Series Technical Readiness Level (TRL) 8 [41]. This TRL level means the system has completed testing and is ready to deploy. However, RAPIER SDS is ready to be deployed to a ground system but not on a satellite [9]. Therefore, further development is needed in this regard for fielding a fully automated capability similar to RAPIER SDS on board a SmallSat. A ground processing framework using RAPIER SDS though is ready for use.

b. RAPIER SDS File Formatting and Sizing

Some of the key benefits of the RAPIER SDS include the ability to process images automatically with minimal user input, the ability to handle NITF, TFRD 4.3, and Geo-Tiff formats from EO, SAR, multispectral, and IR imagery; the ability to output to KMZ, OTH-G, HTML, XML, and RIT; and the ability of users to modify existing algorithms and to create new algorithms by leveraging the open source software of RAPIER SDS [41]. RAPIER SDS is a tool for the image analyst to automatically detect targets or analyze images as well a tool for the image scientist to develop new image processing algorithms to plug into the RAPIER SDS [42].

Of the available output formats, KMZ and HTML are highlighted as popular user-friendly formats, as KMZ files are compatible with Google Earth for overlaying images, and “tip sheets” are generated in HTML. KMZ is a “file format that is used to display geographic data” using a case-sensitive “tag-based structure with nested elements and attributes” [43]. It is strictly structured according to the KML Reference and consists of a main KML file along with supporting files altogether packaged into a single unit called a KMZ archive using a Zip utility [43]. A KMZ archive undergoes compression as well which typically results in a 10:1 compression result depending on the contents of the KML file [43]. Thus, a 20 Kbyte KML file can be served with a 2 Kbyte KMZ archive file. The “tip sheet” shown in Figure 11 includes an image chip of the vessel, the ship type with a confidence percentage determined with the classification algorithm, a geolocation (latitude/longitude), a true heading determined by both a bow/stern curvature algorithm and wake processing algorithm, a compass showing the degree orientation of the ship, a length and width column, and an error ellipse column [42]. The tip sheet is organized by ship size for ease of browsing [40]. In terms of file size for an example input image into RAPIER SDS that has 336 detections, the HTML format tip sheet would be around 391 KB, the XML format tip sheet would be about 8,448 KB, and the KMZ format tip sheet would be about 650 KB [9]. Images of up to 4 GB in size can be processed [9].



(U) Output Type: HTML “Tip Sheet”

Note: Underlined and bolded heading values indicate higher confidence.

Ship ID	Image Chip	Ship Type (Confidence)	Latitude / Longitude	True Heading (°)	Compass	Length (m) / Width (m)	Error Ellipse: semimajor axis (m)/ semiminor axis (m)/ azimuth (°)
Ship 4	 Length/Width Lines	Tanker (84 %)	35.51761 / 139.96410	<u>22.37</u> / <u>202.37</u> (75%) (Bow/Stern Algorithm)		225.25 / 49.18	53.09 / 46.39 / 301.48
Ship 12	 Length/Width Lines	UNKNOWN	35.24581 / 139.78450	<u>1.96</u> / <u>181.96</u> (70%) (Wake Algorithm)		121.71 / 33.44	0.00 / 0.00 / 0.00

Figure 11. RAPIER SDS tip sheet output. Source: [40].

c. RAPIER SDS Image Processing Software Interface

Figure 12 illustrates the program in use for a vast number of targets. Each maritime detection is outlined with a yellow square. Figure 13 shows a command window that allows the RAPIER SDS user to set options for the size of objects to be detected from the image, the maximum amount of tips to acquire and buttons corresponding to whether the image being processed is for EO/IR or SAR. This command window still exists today with slight modifications. Today there is an option for detections to be set as points on the targets or rectangles on the targets [9]. The output files get written out to the local file system after the image is processed [9].



Figure 12. RAPIER SDS detecting ships in an image. Source: [40].

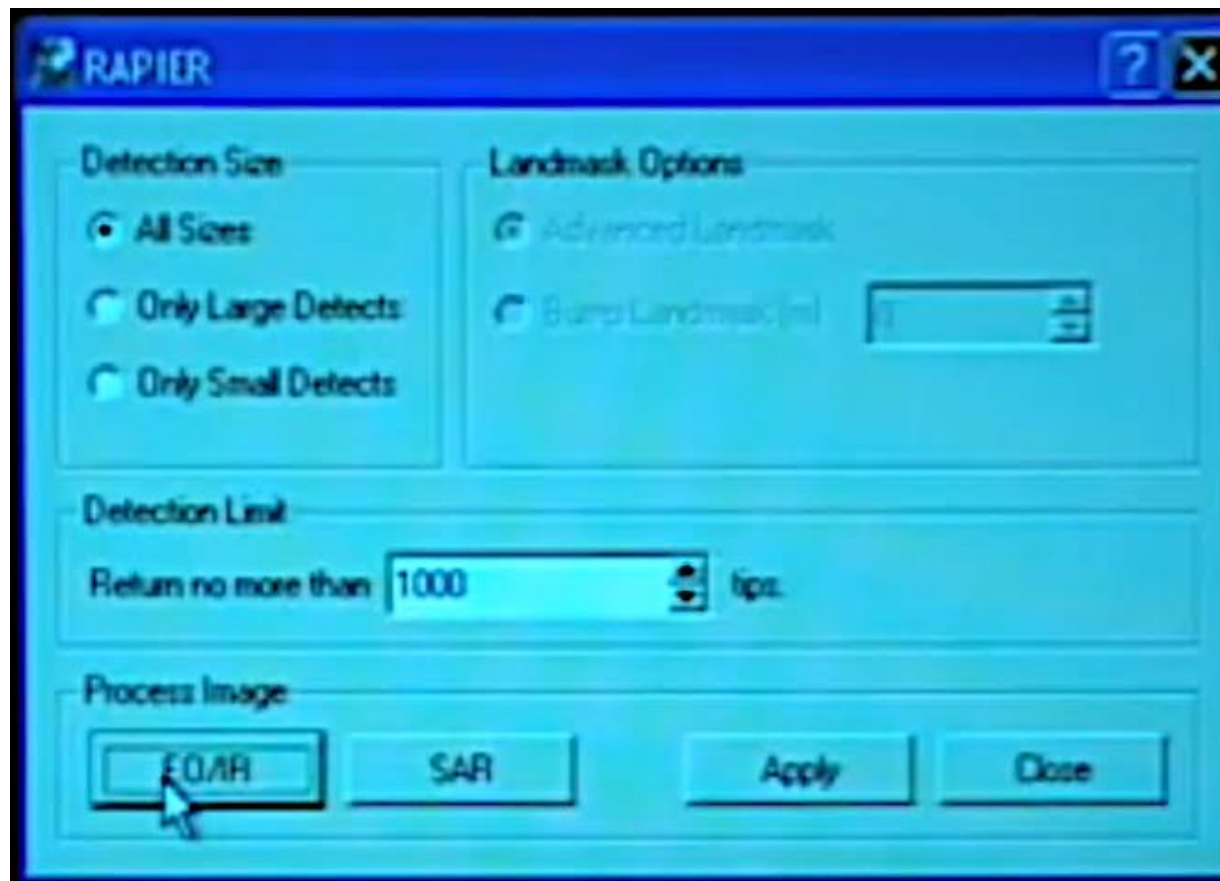


Figure 13. RAPIER SDS detection options command window. Source: [42].

d. RAPIER SDS Detection Challenges

The algorithms for RAPIER SDS produce a lot of false detections for EO imagery versus SAR imagery [9]. The maritime domain does present ship detection challenges for EO imagery in the form of sun glint, whitecaps, and ship wakes [42]. Sun glint is a physical phenomenon whereby light that is reflecting from the ocean surface does so at an angle that matches the angle by which the satellite sensor is viewing it [42]. It is a type of specular reflection and is present when the surface of the water is relatively calm, creating a smooth surface much like a mirror does, which causes the light reflection off the surface to be very regular and not diffuse [42]. Whitecaps are regions of ocean waves where water movement is chaotic and where water can move choppily in random directions [44]. The height and shape of whitecaps vary with the amount of time the wind blows over the water, the intensity with which the wind blows over the water, and the surface area of the ocean in which the wind comes into contact with [44]. Large, frothy white caps are indicative of wind that has blown over water long and hard [44]. A ship wake is the water froth generated in the stern trail of a moving ship and can also be used to detect ships. Detecting this signature can be a challenge in the open ocean with various sea states. Sun glint is a concern because it can cause sections of satellite imagery to be blotted out and therefore lower detections of ships. Whitecaps can appear as ships and so produce false detections.

In addition, the SPAWAR RAPIER team has taken steps to mitigate noisy backgrounds producing false detects by developing algorithms that characterize that background and then change the detection algorithms to find small ships on that background [42]. RAPIER SDS is also capable of detecting the presence of wakes in an image, and from this information the ship's heading can be determined easily enough [42]. If no wake is present, the bow and stern curvatures can be analyzed and when the location of the bow is determined, the ship's heading can be determined [42]. Characteristics of the ship such as length, width, and deck features such as domes, containers, and hatches, can be used to determine the ship type [42]. "Applications for

RAPIER SDS include port, coastal, and open ocean surveillance, monitoring of fishing zones, assistance for search and rescue, drug and human trafficking monitoring, and merchant vessel tracking” [42].

e. RAPIER SDS Calibration Method

RAPIER SDS is calibrated through the importation of the geometric properties of thousands of vessels from publicly available databases of vessel registries. Initial classification is performed by grouping ships according to their length and width and length/width ratio [42]. The program is currently in use operationally in its current ground processing form, but because it is a program that operates on a computer, it is feasible for RAPIER SDS’s algorithmic framework, but not exactly RAPIER SDS to be implemented on a small size satellite as on-board memory could be loaded with the program inside a processor unit. This is not outside the range of technical possibility.

f. The Future of RAPIER SDS

Some of the realities of the RAPIER SDS tool today (2018) are that it has been set aside without further development for about five years [9]. The limits of what Computer Vision, with its many geometric analysis processes and digital image processing algorithms based on Fourier Transformations, is bumping up against a ceiling in its ability to further classify, rather than typing, ships. About 1,000 labeled images for a particular ship class in a hypothetically more advanced system based on machine learning would probably be adequate for a proof-of-concept for a machine learning algorithm to classify ships [9]. For an operational system, 5,000 to 10,000 training images would be needed for training for classifying a particular ship class [9]. All of this training for an operational system can be done on the ground.

To overcome the limits of applying computer vision techniques alone for ship classification, the RAPIER SDS team is using synthetic imagery 3-D modeling of ships in many environmental conditions to simulate different look angles of the same ship in 1000 different images [9]. Prospects for using RAPIER SDS as it is on a satellite do not seem to be so promising. Apparently, RAPIER SDS has been developed in a limited scope to be deployed to desktop machines [9]. In any case, in its current form RAPIER

SDS cannot be deployed on a satellite but with new development efforts and reconfiguring and refactoring of RAPIER SDS, the possibility for a modified version on-board a satellite remains possible with increased funding for those development efforts. Lastly, one major conclusion agreed upon within this thesis and with the RAPIER SDS team is that multiple algorithmic and sensor processing paths to achieve a classification on a ship are the most promising [9]. At the very least using one sensor, multiple machine learning algorithms and their determinations for classification of a ship can be combined in a sequence or parallel with other methods like computer vision methods to arrive at an aggregate determination, which may be more accurate. For example, the inability of a particular support vector machine (SVM) to classify an object can be mitigated by using additional algorithms after the SVM to avoid wrong classifications or inability to classify. An SVM is a type of machine learning algorithm used to classify a new unlabeled or unclassified sample into a particular class based on previous training of that SVM. This idea will be discussed further in later detail in Chapter three.

2. DigitalGlobe Atmospheric Compensation (AComp)

DigitalGlobe has an atmospheric compensation computer program that corrects for the effects of light-wave scattering due to particles in the atmosphere. It does this by recognizing that the quality of a digital image is affected at three levels of reflectance [45]. Specifically, these are top-of-atmosphere (TOA) reflectance, Rayleigh Scattering, and surface reflectance [45]. TOA results from incoming sunlight from the sun being reflected off the atmosphere which never gets reflected from the surface. Rayleigh scattering results from light being scattered by particles in the atmosphere. DigitalGlobe's AComp proprietary algorithm promises to clean up satellite imagery to the point that what is left is a purely surface reflectance image [45]. The main sources of unwanted reflectance come from water vapor, haze, and other particulates in the atmosphere [45]. The strength of the algorithm lies in using atmospheric aerosol information from the imagery being collected. However, AComp also corrects for absorption effects [46]. How DigitalGlobe also differentiates itself from other algorithms said to also reduce these effects is that the AComp algorithm analyzes the Aerosol Optical Depth (AOD) [46].

AOD is a dimensionless number that represents the absorption and scattering of light through a column of air due to aerosol particles such as dust, haze, and smoke [46].

In order to achieve MDA via SmallSats in certain parts of the world where massive industrialization is happening along coastline cities, an algorithm such as AComp is important to cut through smoke and haze being produced from industrial pollution. Figure 14 illustrates the massive amounts of haze that move around the East China Sea. For anyone who has been forward deployed in the East Asian region in contemporary times, blue clear skies are hard to come by, thus highlighting the need for an algorithm that is able to filter out pollution effects from satellite imagery.

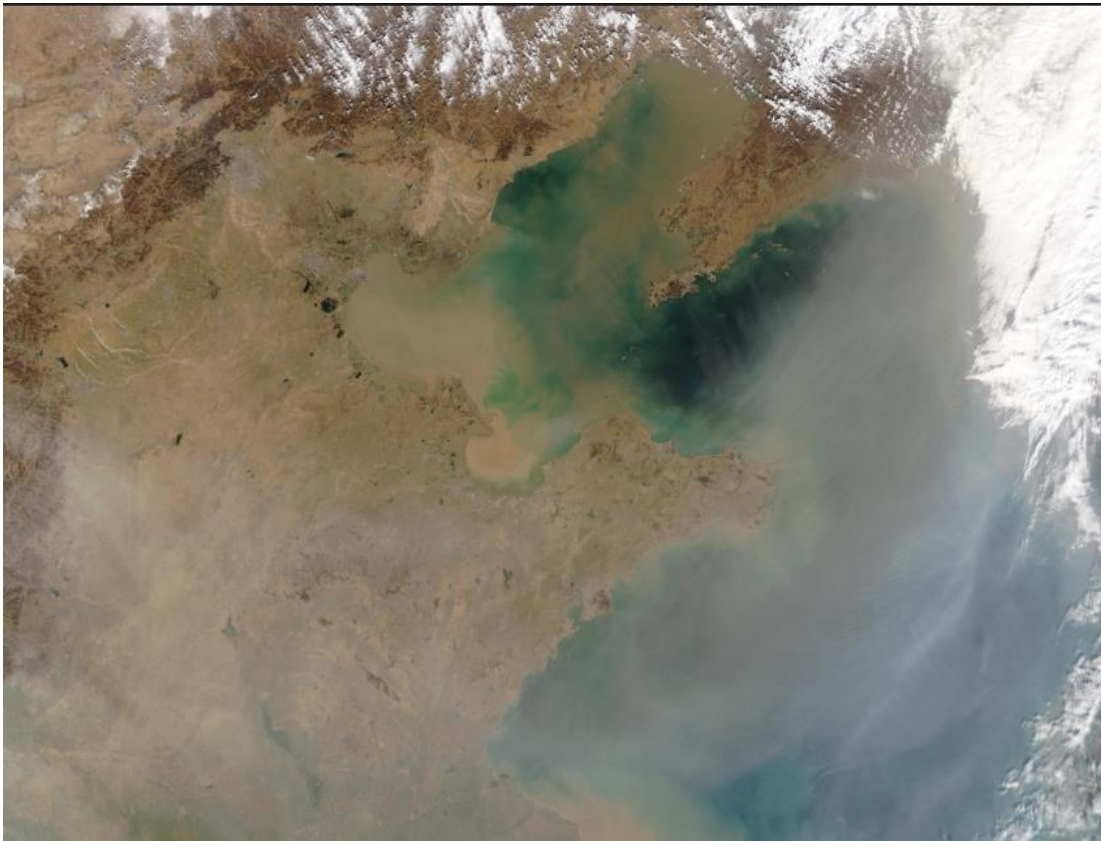


Figure 14. Haze moving over the East China Sea in NASA satellite imagery from February 25, 2004. Source: [47].

E. PROCESSING MODE OPTIONS AND ANALYSIS

There are three primary modes for processing satellite imagery. The first involves taking the image and transmitting the image in whole or in part to a ground station. This is the current state of affairs in satellite image processing and involves the most image analysis and processing by human analysts in order for information to be derived from the image to be used for real-world decisions or operations. This also requires the largest volume of data to be delivered to the ground processing elements. The second mode of processing would involve sending something like RAPIER SDS's tip sheet down to a ground station after on-board satellite processing. The information in the image would be processed to give a set of detections and ships of interest for a maritime setting along with ship type, heading, and other object/ship parameters that could be uploaded by a human-in-the-loop to another system for action, which may involve additional transmission via another satellite hop to the ultimate user of that finally processed data. Without implementing a more advanced framework for on-board satellite image processing that could not only determine types of ships in an image but also classes of ships, this framework would be needed. The third framework would implement a full satellite on-board image processing suite for object classification that would downlink a properly formatted data stream to a user terminal on the ground for immediate action by the user terminal based on that data input.

The first mode is proven but adds valuable time to any real-time processing cycle driving real-time action based on that data processed. In fact, this is the longest processing mode as human analysts are performing much of the image analysis. The second mode may be acceptable depending on the time requirement and the location of the processing station where the downlink is occurring in relation to the end user who takes action based on the final processed product.

A current commercial example for operating a ground processing network of SmallSats is Planet's Mission 1 ground architecture. The autonomous nature of its ground sites where images are downlinked and "temporarily buffered locally on a server co-located with a receive antenna, and then added to an Internet upload queue" to cloud storage over a secured Internet connection for customer consumption, does demonstrate a

level of processing automation that is promising as a proof-of-concept in completing one part in establishing a fully automated image processing chain [48]. However, this method is constrained by the availability of downlink opportunities driven by the ground architecture. Once uploaded to the cloud the “images are individually de-mosaiced, color-corrected, flat-fielded, ortho-rectified,” and cloud cover within the image is masked so that customers receive a series of quality metrics [48]. Another interesting feature to Planet’s Mission 1 constellation is that its satellites will store imagery from one pass and the imagery will only be deleted until the next pass when the imagery is” confirmed as having successfully entered the processing pipeline” [48]. That is, the imagery is held for the next pass regardless of the state of processing. The possibility for failure in receiving the current pass imagery and being able to appropriately process that imagery does exist, so a similar architecture could also build this redundancy in as well or perhaps cross-link imagery for other purposes and use by ground users. That is, there could be a failure in a particular satellite to downlink imagery files due to various factors, such as the ground station becoming inoperable in the area or downlink file corruption.

In any case, the third mode of processing is the most technically challenging today as several of the methods to implement such an architecture on-board a SmallSat are in a proof-of-concept stage as highlighted by the Nvidia-based development board discussed earlier in this thesis. However, by implementing such an architecture, ground stations for payload processing would not be needed, which could result in significant cost savings in implementation. The disadvantages of satellite ground stations include susceptibility to destruction, the need to possibly obtain foreign government approval for placing a satellite ground station on its soil, the need to provide for some level of security for these stations which may include physical security measures, and other various operating costs to name a few. The advantage of using ground stations primarily is that various proven avenues for processing satellite imagery can be executed at these ground stations. Instead of putting all the processing complexity on board a SmallSat to accomplish classification of a maritime asset, a ground station operator within the field of view of the satellite can be present to click through options in a program like RAPIER SDS, or image analysts can further process the imagery using their expertise.

In the next chapter, the fundamentals and basic of machines intelligently being able to extract and identify geometric and photometric features of objects in imagery will be discussed to become familiar with a possible way forward in advancing SmallSat on-board image processing to a point where it is autonomous. Specific mathematical procedures for classification will be discussed. In particular, methods specific to machines classifying objects in imagery will be of analytical importance.

THIS PAGE INTENTIONALLY LEFT BLANK

III. ARTIFICIAL INTELLIGENCE AND DIGITAL IMAGE PROCESSING FOR SMALL SATELLITE MARITIME DOMAIN AWARENESS

A. BACKGROUND

The field of artificial intelligence (AI) is not new. It has been studied and worked on since the 1960s. Advances in algorithms, computing technology, electronic manufacturing processes, and a greater number of experts in this field have recently reached a critical mass. Commercial AI applications are sprouting up at a brisk pace from targeted online advertising, self-driving cars, order kiosks, natural language processors, and most important to this thesis work, in smart image sensor technology. As Figure 15 illustrates, the field is wide-ranging and technically complex. Since the early 2000s, one of the biggest buzzwords in Silicon Valley has been AI. Silicon Valley and most notably companies such as Alphabet Inc., Nvidia Corporation, and Facebook have been working in this area at a brisk pace. The commercial ecosystem for artificial intelligence processing hardware and software is firmly in place and further advancements within this ecosystem are poised to accelerate through the collective synergy of the efforts of the companies within this ecosystem.

Together with the advancements in satellite sensor and antenna technology coming from various U.S. Defense contractors, and most notably in the SmallSat sensor segment, a fully automated satellite image processing framework is not long off in the future as TRL 4 and 5 levels have been reached as demonstrated . Commercial demand for this AI processing technology on-board SmallSats will certainly be there as the need for maritime sea traffic reports and other economic indicators extractable from satellite sensor technology transitions into near real-time.

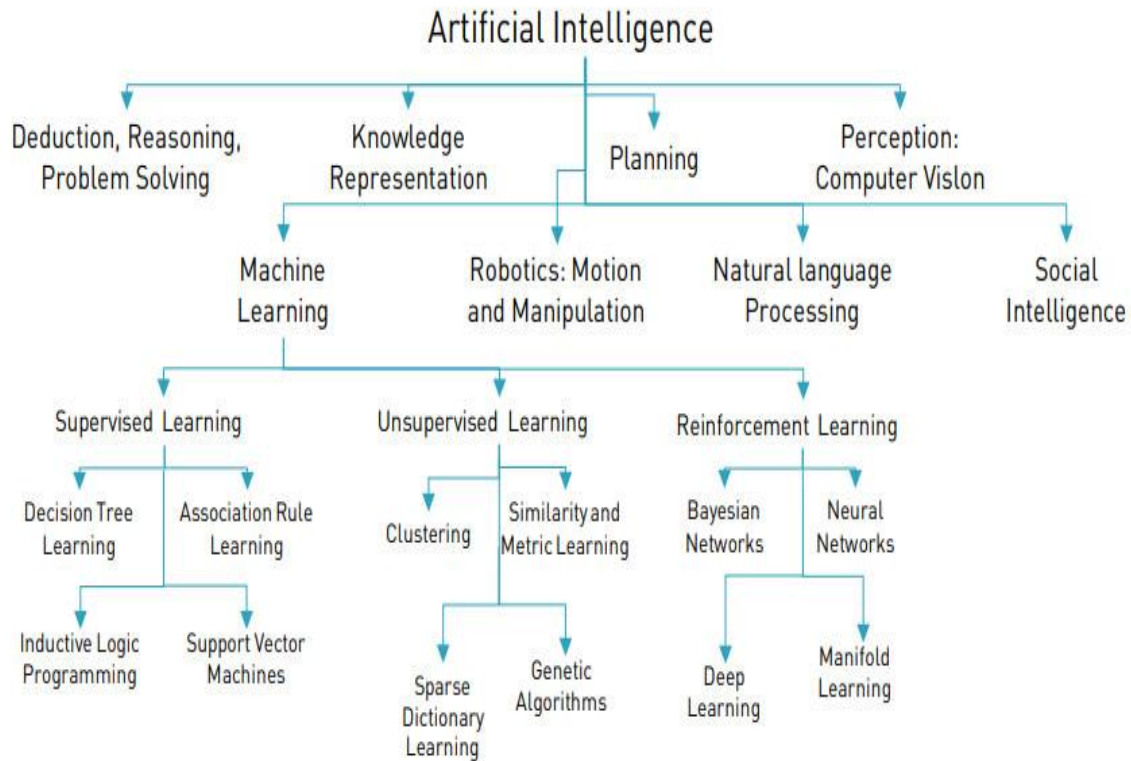


Figure 15. Artificial intelligence categories and subcategories. Source: [49].

B. COMPUTER VISION AND MACHINE LEARNING

Computer Vision is pertinent to the research described herein because it involves an automated process whereby features from an image or images are extracted to achieve a visual understanding of objects and their properties within the image set [50]. On the commercial side for satellite imagery, startups such as Orbital Insight, Cape Analytics, Descartes Labs, and Space_Know are looking to leverage this automated intelligent processing of images to deliver informative content to customers.

Various models for classifying an object in an image exist [51]. Some main ones are:

1. Classification using the nearest class mean
2. Classification using nearest neighbors
3. Classification by maximum a posteriori probability

4. Classification using a feedforward artificial neural network
5. Decision trees

How exactly an object in an image gets recognized is a topic worthy of introduction before having a discussion of these five methods of classification. This naturally leads to the idea that any object will contain features. For example, for a ship within an image, some features that have some value to classification may be area, curvature, perimeter, length, or width. Many important features may exist, and choosing the right features to examine in the construction of an artificial neural network for classification is important because such networks may undergo supervised learning, wherein the artificial neural network's accuracy is improved through a training set that decides proper classification based upon these features. An artificial neural network is a type of mathematical network of nodes used to take inputs and produce an output based on the network having already determined weights for nodes in the network based on having been trained on a data set.

Classification using the nearest class mean is a vector comparison method. For a two-dimensional feature plane wherein each class member point is defined by two features, one class may have a distribution of class points in one area of the plane and another class a different area of the plane and so on. There could be many classes, and this method is not limited to simply a Cartesian plane. Figure 16 illustrates this concept. The black "X" is nearest the turquoise "X" for the distributions and so will be classified as belonging to the class of vectors denoted by the color turquoise.

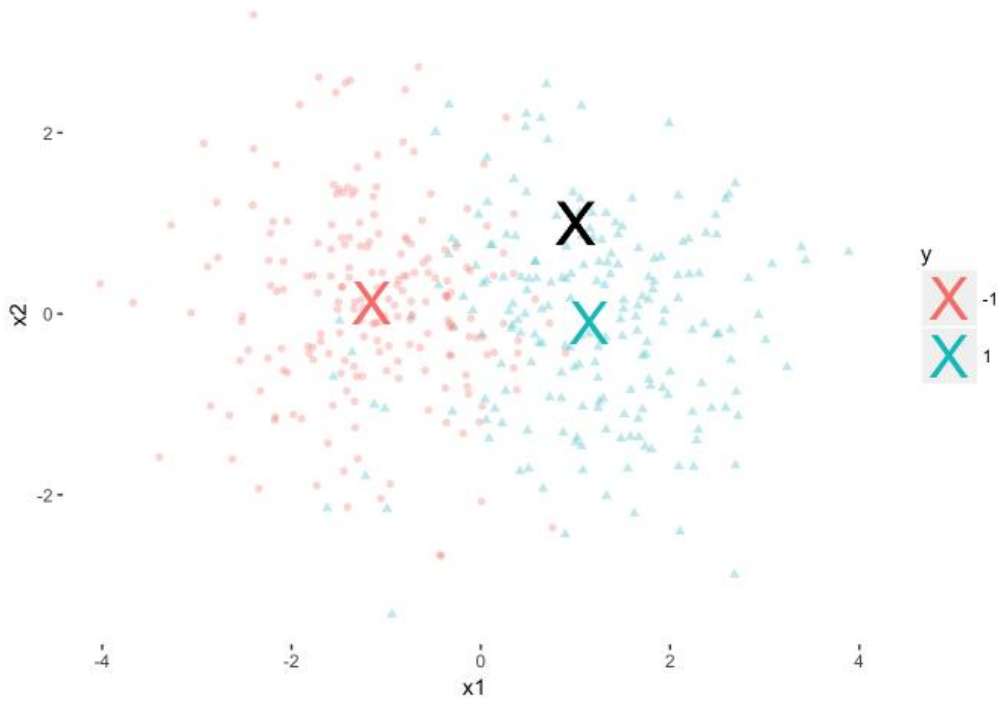


Figure 16. Illustration of classification using nearest class mean. Source: [52].

However, this example is used for simplicity. Each class will have a class mean vector [50]. Therefore, any new feature vector $\mathbf{x}$ can be classified in whichever class whose mean vector is closest to the vector $\mathbf{x}$ [50]. However, variations with class data distributions can be multimodal and have outliers within the class closer to the mean of another class, so that any new feature vectors belonging to that class may be classified incorrectly [50]. To prevent this, a modified scaled Euclidean distance is computed from $\mathbf{x}$ to the class mean vector $\mathbf{x}_c$ that involves computing the standard deviation along each dimension of the class. Equation 4 is the resulting equation [50, eq. (4)]. The letter “c” denotes that the vector is a class centroid, and the letter “i” is a summation index variable to denote a vector within the class. The class centroid can be treated as a constant then for every index.

$$\|\mathbf{x} - \mathbf{x}_c\| = \sqrt{\sum_{i=1,d} ((\mathbf{x}[i] - \mathbf{x}_c[i]) / \sigma_i)^2} \quad (4)$$

Classification using nearest neighbors, which is often abbreviated as kNN, standing for k-nearest nearest neighbors, simply uses the nearest k feature vectors regardless of their labels to classify an unknown feature vector $\mathbf{x}$ [50]. This does not use the same formula for classification using nearest class mean but simply calculates a simple difference between the unknown sample class vector and nearest neighbors of each class. Then, the sample class vector can be classified into a particular class based on choosing the smallest difference. Certain restrictions on what value k can be to prevent ties for classification are listed in certain books on pattern recognition [52]. Classification by a posteriori probability uses Bayesian probability and particularly Bayes Rule which depends on some posteriori probabilities. Using statistics, samples of occurrences can be plotted to form an array or histogram. An underlying probability distribution, whether Poisson, exponential, or normal can then be fitted with this sampled data based upon statistical tests such as the normality test, a chi-square goodness of fit test, and other statistical tests [50]. A Bayesian classifier will classify an object into the class to which it is most likely to belong based on its observed features [50]. Understanding how a convolutional neural network works will serve as a good example of a feedforward artificial neural network (ANN), which will be discussed soon after discussing image segmentation. A feedforward ANN does not include any loopbacks from one decision layer of an ANN to a previous decision layer. Decision trees classify based upon using branching steps with each step testing a quantitative or qualitative value of a feature of interest [50]. In the case for MDA classification, each step of the decision tree can be seen as testing for a qualitative answer based upon quantitative measurements.

C. IMAGE SEGMENTATION

Image segmentation is useful in ignoring large parts of an image that hold no analyst value. For example, in a particular scene of ships on the ocean, the ships are of sole interest for purposes of detecting and classifying these ships. Image segmentation involves treating pixels within an image as having numerical values, which can be used mathematically in image segmentation operations. A jump or no change in one pixel intensity value to a neighboring pixel intensity value is of particular interest. A jump introduces a discontinuity from one pixel intensity value to the next. If two neighboring

pixel intensity values are the same, there is similarity instead of discontinuity. A notable discontinuity is an edge, and so edge detection is an important concept in image detection. In addition, two other discontinuities of importance are lines and points. Before delving, into discontinuities further, it is important to understand some basic image properties. An image is composed of pixels, and pixels can be represented by numbers based on intensity. Therefore, every image can be abstracted to be a matrix of values.

Further, an important tool for extracting information from an image is to use another matrix of values termed a mask. A useful image extraction technique involves computing the sum of products of these two together. By structuring the matrix mask with certain values in certain cells of the mask certain geometric features of an image can be extracted [51]. A simple example of this to look at would be the following 3 x 3 mask in Figure 17.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 17. 3 x 3 mask for feature extraction

For a 3 x 3 pixel area on an image, sliding this mask over the corresponding pixels would yield for the middle cell in the 3 x 3 pixel area, a product value significantly higher than the products for the cells immediately surrounding it results. A homogeneous background in the area of the detector mask is assumed if the sum of product of the mask values and pixel values are to sum to zero [51]. In a MATLAB context, if a 3 x 3 matrix A is the pixel matrix and a 3 x 3 matrix B is the mask matrix, this summing is equivalent to $\text{sum}(A.*B)$. If this mask was slid over a homogenous pixel intensity value matrix containing all 2's, the sum would be computed as: $2 * (-1) * 8 + 2 * 8$, which would give a sum of zero, indicating the background is homogeneous and there is no point there in the image. However, if there was a point in the image, its value could be 4, which in this case

would yield: $2*(-1)*8+4*8 = 16$. Therefore, a non-zero sum indicates a single point discontinuity.

Line detection would follow in a similar vein. For example, to detect a horizontal line, the mask in Figure 18 could be used [51]:

-1	-1	-1
2	2	2
-1	-1	-1

Figure 18. 3 x 3 mask for horizontal line detection

It is not hard to imagine how the set of 2's in Figure 18 could be arranged to extract line detections for a vertical line, a 45 deg. angle line, and a -45 degree angle line [51]. Thus, far, we are assuming all detection schemes to be applicable to gray level differences. That is, we are dealing with a panchromatic image. Software for converting color images to black and white images is readily available. This process is a simplification of the image segmentation algorithm that is used in feature detection software such as RAPIER and others that can be used for MDA purposes.

D. CONVOLUTIONAL NEURAL NETWORKS, SUPPORT VECTOR MACHINES AND DEEP LEARNING

Whether MDA images are processed on board small satellites or at ground sites, convolutional neural networks (CNN) will be important in speeding up the processing component of any operation in which processing maritime objects of interest is important. The power of CNNs lies in their ability to perform image recognition and classification. Pioneering work by Yann LeCun on the LeNet architecture in the 1990s resulted in character recognition tasks such as reading zip codes and digits [53]. Therefore, the importance in developing such architectures for not only small satellite imagery processing but satellite processing in general cannot be understated. In the case

of MDA, building and developing a satellite imagery processing framework with the greatest amount of accuracy possible and with the least amount of operator inputs will open up new possibilities for MDA for both commercial and government entities.

For the purposes of this thesis, detection, identification (typing), and classification of maritime vessels is of interest. CNNs are a category of neural networks that can greatly assist in automating this process. Neural networks are a set of algorithms designed to recognize patterns and have their origin in the way the human brain biology works with neurons to make decisions. Instead of using neurons, neural networks use nodes, which are points where computation happens. These nodes then makeup layers and at each node input data goes in and is combined with a coefficient or weight that acts to decrease or increase the input and present its output to the next set of nodes. These weights change as a CNN is trained and stay the same after the training phase. The output of each node is subsequently connected to each node in the next node layer, whereby it may be used. The most basic neural network simply has “an input layer, a hidden layer, and an output layer” [54]. The nodes of the hidden layers are where the weights are used to multiply against the inputs to the hidden layers. Such a network is depicted in Figure 19.

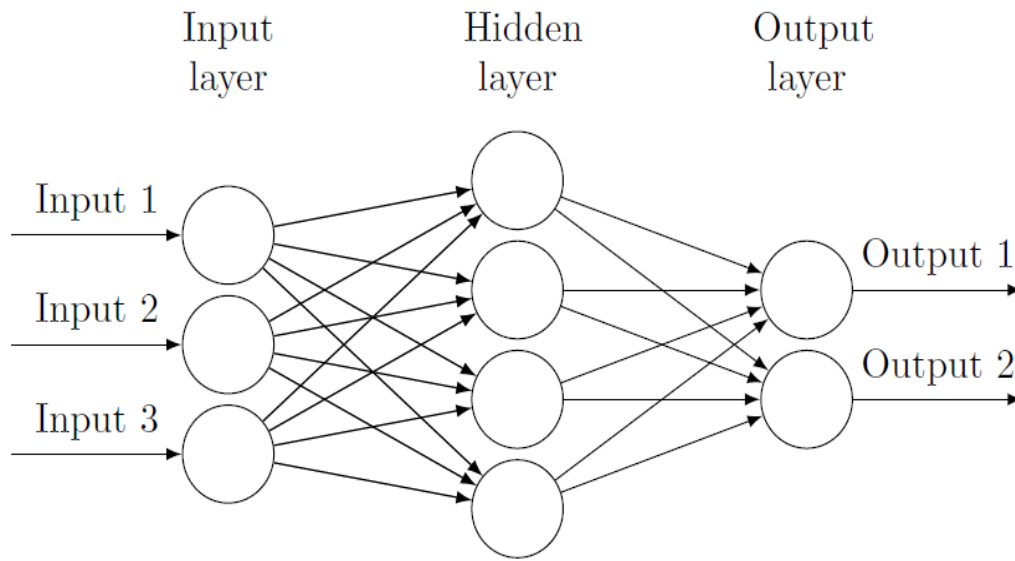


Figure 19. Basic neural network. Source: [55].

The qualification for a neural network to be considered a deep neural network (DNN) is for the neural network to have more than three layers, which means the network possesses more than one hidden layer [54]. Different features are trained for in each layer of the DNN, which form an increasing level of complexity in features trained for by using the outputs from previous layers as inputs into subsequent layers [54]. Altogether, this “makes deep-learning networks capable of handling very large, high-dimensional data sets with billions of parameters that pass through nonlinear functions” [54]. The potential of deep learning to organize photos by such features as having a pigeon sitting in a corner or a child in a grass field is possible through DNNs. That is, objects within a picture can be detected and classified. This is a process of taking unstructured and unlabeled data and outputting sets of structured data [54].

The power of DNNs is that algorithms can be written to use them, and associated hardware can be built to implement their structure to produce systems that can perform expert tasks. For satellite imagery, the knowledge-based expert system that would utilize artificial intelligence then takes the place of the image analyst in determining and classifying object within images. In addition, much like how humans learn with

experience and training, DNNs can be trained on labeled data [54]. In theory, the larger the training set the better the performance of the DNN when applied to unstructured data [54]. The output of a deep-learning network is a set of probabilities associated with possibilities as to what the object identified in the image could be [54].

Convolutional Neural Networks (CNN) involve a type of image convolution with the primary purpose of extracting features from the input image. Because an image can be treated as a matrix array of pixel values, one can use another matrix array to act as a sort of filter to perform a convolution on the image matrix. The filter slides over the image from left to right starting with the top left of the image moving a pixel wide distance with each step. At each step, an element-wise multiplication of the filter matrix with the overlapping pixel area of the image matrix is performed and the sum of that is used as the resulting element in the convolved feature. Based upon the dimensions of the filter matrix, the pixel width and length of the convolved feature matrix is equal in dimensions for this qualitative case being described. After the first row of the convolved feature is obtained through this multiplication process, the filter is shifted down one-pixel length and performs the same horizontal shifting to obtain the second row of the convolved feature matrix, and this process continues until the last element-wise multiplication is performed when the lower right corners of both the image matrix and feature matrix overlap [51].

With different filters, different operations can be performed on an image, which include the identity operation, an edge detection operation, a sharpening operation, a box blur, and a Gaussian blur to name a few [51]. Another name for the convolved feature is feature map. Three variables make up a feature map [51]. The first parameter of note for a feature map is depth and it directly “corresponds to the number of filters used for the convolution operation” [51]. In the previous example noted to describe the convolution operation, it was assumed that only one filter was used. Also, the size of the feature map is dependent on stride, which is the amount horizontally shifted in number of pixels for each element-wise multiplication and resulting addition [51]. As the stride increases, the dimensions of the resulting feature map decrease in proportion [51]. Lastly, for the actual convolution operation, zero-padding can be utilized where the border of the image matrix

may be surrounded with zero-valued pixels, which would result in a bigger feature map when convolved [51]. The term for zero-padding convolution is wide convolution and narrow convolution for image convolution without zero-padding [51].

After the convolution is performed and the feature map is obtained the negative pixel values in the feature map can be zeroed out to introduce non-linearity since real-world data the convolution is performed on is non-linear [51]. For most situations, this simple non-linear processing part is more effective than other non-linear functions that could be used such as tanh or sigmoid [51]. The next step that can be performed is the pooling step, which reduces the dimensions of each feature map using a max, average, or sum [51]. Max pooling in practice works better so this type of pooling tends to be used. Max pooling simply slides a smaller dimension window over the feature map and takes the max pixel value from each grouping of pixels for each slide [51]. Figure 20 illustrates such a max pooling operation.

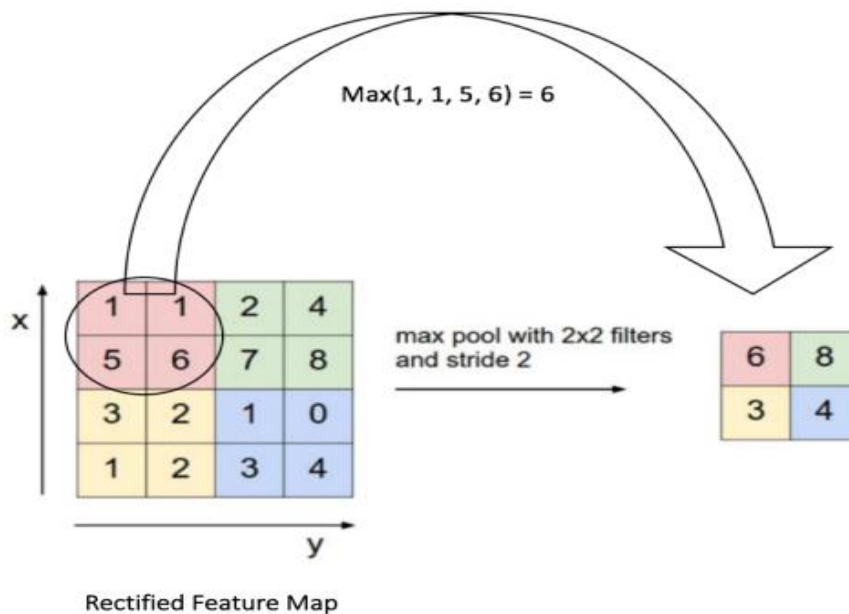


Figure 20. CNN pooling stage illustration. Source: [51].

The final layer of a convolutional neural network is called the fully connected layer. This is the layer where image classification happens and its name “implies that

each neuron in each layer is connected to every neuron in the next layer” [51]. This layer implements a softmax activation function that can be used to convert value into probabilities. For an image, this set of probabilities may be associated with mammals for example. There may be a 0.9 probability that an object in an image is a dog, and a 0.05 probability that it is a cat, and a 0.05 probability it is a raccoon. Each stage from convolution to pooling to the fully connected layer serves important purposes for increasing the general accuracy of the guess and there are multiple iterations of each layer in a real CNN. Oftentimes, the vast complexity of CNNs makes describing why a particular probability was assigned difficult. However, as factors such as weights for the CNN are adjusted in response to the CNN being trained on increasing numbers of images, the accuracy trends upwards and image object recognitions can reach over 90% and in some cases be very close to 100% [51].

In short, CNNs are an important tool for image object classification which can be implemented in recognizing classes of ships with sufficient complexity and with sufficient satellite sensor resolution. Combined with hardware such as FPGA technology, GPU technology, and TPU technology (more distant in the future), the ability of CNNs to contribute to on-board image processing leading to image object classification without further analysis will greatly reduce the processing time involved in the tasking, collecting, processing, exploitation, and dissemination (TCPED) cycle.

E. SUPPORT VECTOR MACHINES

Support vector machines use supervised learning. Supervised learning is grouped into classification and regressions problems, but only classification is of concern in this thesis. At its core, supervised learning entails learning how input data maps to output data and through this being able to predict the output for new input data [56]. Because a set of input data is known to be mapped to a set of corresponding output before predictions are made, this original mapping acts much like a supervisor in determining those predictions [56]. Determining an optimal hyperplane for linearly separable patterns is the key idea behind support vector machines [56]. Mathematically, a hyperplane $\mathbf{V}$ can be defined in a vector space $\mathbf{W}$ as any subspace such that $\mathbf{W}/\mathbf{V}$ is one-dimensional [57]. In layman’s

terms, this means for a 2-dimensional space (a plane) the hyperplane is one-dimensional (a line) or one less than the 2-dimensional space. However, if the patterns are not linearly separable, there is still a method to make the support vector idea work [56]. This is accomplished by transforming the original data to a new higher-dimension space via what is termed the Kernel function [56]. The mathematics for this can be seen in Appendix B.

The “support vectors are the data points that” lie in the “gutters” of the area wherein the hyperplane exists [56]. These gutters are the edges determined by a margin of separation calculated from the hyperplane in order to allow for classification. The purpose of SVMs are to maximize the margin on each side of the hyperplane. In 2-dimensions a line can be used to separate the data points for each class, but for higher dimension hyperplanes are needed [56]. Finding the hyperplane is an exercise in solving by using optimization techniques based on Lagrange multipliers. Figure 21 illustrates the construction of such a hyperplane for a set of two-dimensional data. The blue solid squares in the figure represent one class of data, and the green circles represent another class of data. The red line represents the hyperplane separating the two classes, and the idea is to maximize the distance between edge points of both classes by selecting a hyperplane for which parallel offset lines produce a maximum margin between those edge points. The points on the margin lines (edge points) are the support vectors that are used in the derivation of the hyperplane (see appendix for details).

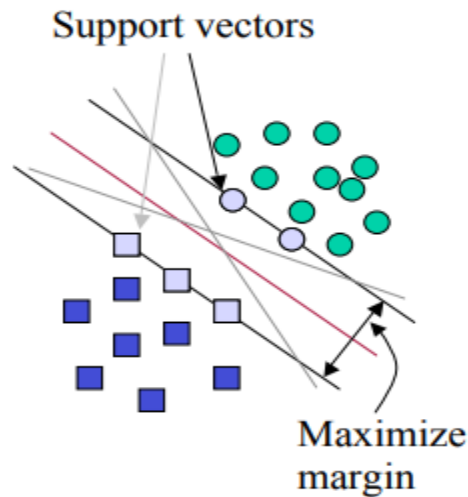


Figure 21. Support vector machines hyperplane and margin separation construction. Source: [56]

Classification redundancies such as using multiple methods of image object classification to arrive at a most likely classification would hypothetically be a very powerful tool as well as highly desirable for image object classification precision. With higher resolution, satellite sensors provide for more precision in seeing features within an image along with supporting more advanced digital image processing algorithms being applicable to object classification within that image. In a general sense, the subsequent use of machine learning to classify objects within those pre-processed images points to a complete automated capability being possibly deployed on-board SmallSats in the future.

F. FEATURE ENGINEERING, EXTRACTION, AND SELECTION FOR MACHINE LEARNING

In order to properly execute machine learning, there are three key steps to be carried out before a CNN or SVM or any other supervised learning or reinforcement machine learning algorithm can be satisfactorily used on test data. These three steps are feature engineering or feature creation, feature extraction, and feature selection. Feature engineering refers to the process of deciding exactly what features are relevant for training a CNN for example [58]. For ships, some basic features that may be trained on could include length, width, length/width, the position of each ship's superstructure,

locations of missile tubes, and the perimeter of the ship as viewed from satellite imagery, etc. However, each type of data will require its own specific and specialized features [58]. After feature engineering, the step of feature extraction must occur. Feature extraction takes the raw data from an image for example and converts it into a vector of feature values [58]. The last step is feature selection, which means a subset of specific features is chosen from the features that have been engineered or extracted [2]. The main purpose of this is to reduce the complexity of the classifier as well as the risk of overfitting and lessen runtime/memory complexity from a large number of features [58]. Lastly, for feature engineering, it is useful to know that features can be created from other features and features can be created from other classifiers [58].

The big takeaway is that SVMs are simply another type of machine-learning software that can be used for remote sensing classification. Programs such as MATLAB can implement the math of SVMs quite readily with just a table insertion into the workspace of MATLAB and by using the classification learner from the Apps menu. Thus, software for this tool already exists in various forms. As recently as 2016, MathWorks introduced its machine learning package within the MATLAB program for example. The applications are used in deciding how a new data point fits into a particular class based upon its feature values. Altogether, the machine learning tools included in MATLAB represent a very powerful tool in making classification decisions. Applications from determining an entity's credit rating, to being utilized as a medical diagnosis aid to indicate the presence or absence of cardiac arrhythmia in a patient, to classifying iris species, and many others indicate that using machine learning to develop a decision structure to classify maritime objects is possible by using the right set of features and the right set of algorithms and classification redundancies [59].

THIS PAGE INTENTIONALLY LEFT BLANK

IV. MDA CLASSIFICATION MODELING IN MATLAB, LIMITATIONS, AND CONSIDERATIONS

A. INTRODUCTION AND SCOPE

The scope of the work in this chapter is to take features gained from image processing and feature extraction and train an SVM to classify based on those features. Also, the limitations of trained CNNs will be explored. It is assumed that feature measurements are already extracted and in a table format for the training of the SVM. In this respect, the aim of this thesis chapter is not to train any particular neural network or SVM to classify ships but to use and build upon features found to be relevant to train MATLAB's machine learning SVM classifiers. There is previous thesis work for MDA related to exactly how to perform image preparation and feature extraction as well as extract feature values [61]. An overview of both the methodology and MATLAB coding capabilities to perform this work will be discussed and summarized. An example case where computer vision feature extraction is assumed, and an SVM is trained and used on those features, will be presented. Additionally, computer vision logic to make the ship classifier construct work will be demonstrated. This is again to emphasize that sequentially using different AI methods for a ship classification programs results in the best classifier model.

B. MATLAB SIMULATION SETUP

The gist of image preparation for feature extraction can be summed up as performing image segmentation techniques, which include image thresholding, image smoothing, edge and line detection by using a Canny edge detector (uses hysteresis thresholding) perhaps, while also first converting the image to grayscale and normalizing intensity values for the pixels within the image. Image smoothing is helpful in obtaining object and background separation in raw images whose histograms do not exhibit the bimodal distribution needed to even begin calculating a reasonable threshold [51]. An averaging mask can be used to perform the image smoothing in some cases. In other cases, multiple thresholding, variable thresholding, and multivariable thresholding (a

method for color images) can be deployed [51]. Multiple thresholding is useful when there are three or more distinct regions in an image that can be separated [51]. Variable thresholding segments the image into different regions, and each region has its own threshold, which is useful when parts of the image are illuminated differently [51]. This method can produce impressive results when applied correctly. Multivariable thresholding refers to color images, where each pixel is defined by three variable intensities corresponding to red, blue, and green [51]. A single threshold will be used for this MATLAB simulation.

For the MATLAB simulation, DigitalGlobe’s EnhancedView Web Hosting Service (EVWHS) was used to acquire imagery. The resolution for this imagery from this repository of Earth remote sensing images is 30-cm. The principal images taken from this repository are of a Littoral Combat Ship (LCS) docked at San Diego naval base, a Ticonderoga-class cruiser, an Arleigh Burke-class destroyer, and an aircraft carrier docked at Naval Station Norfolk, an oil tanker sailing through the Strait of Hormuz, and a container ship. Table 1 summarizes the image sizes used for these particular ship images.

Table 1. Ship image dimensions

Ships	Image Size (pixels)
LCS	213 by 319
Ticonderoga-class cruiser	245 by 675
Arleigh Burke-class destroyer	275 by 638
Aircraft Carrier (uncropped)	903 by 1840
Aircraft Carrier (cropped)	421 by 649
Oil Tanker	903 by 1840
Container Ship	558 by 340

This gives a total of seven images with two being variations of the same ship, which was the aircraft carrier. The simulation is divided into two parts, one in which SVMs are trained on known geometric and photometric features with the trained model being used to classify new slight variations from the exact known feature values for

thenset of ships (a slight variation from a known ship), and the other is using CNNs to classify the ships from their pictures.

C. MATLAB RESULTS

1. MATLAB Image Processing and Feature Analysis

A threshold is simply a digital number (DN) value in the image histogram that allows for an object and background separation. A DN is a numerical value that represents the shade of coloring for a pixel in an image. For gray-level images, a DN value of zero represents black, and a DN value of 255 represents white with varying levels of graying in between. In MATLAB, an image is read into Editor, and the color image is then converted to a gray-level image. Figure 22 shows the result of such a conversion.



Figure 22. Color (left) and gray-level (right) image of LCS docked at San Diego Naval Base. Source: [34], adapted from [34], respectively.

The peak in the histogram plot corresponds to the background as by visual inspection of the gray image in Figure 22 one can see that the background has the blackest DN value and most frequently occurring DN value in the picture. This is represented by the peak in Figure 23. This can also be seen from Figure 25 as all the objects in the picture appear white while the background is black for both pictures. The

particular MATLAB function “im2bw” uses the second input into its argument list to define a threshold below which everything it is made black and anything above it is made white. Therefore, most of the normalized DN values after 0.4 in the graph in Figure 23 correspond to the ship, the dock, and the clouds in the lower right-hand portion of the image. A threshold was chosen by visual inspection of Figure 23 that most of the pixel data comes before a normalized DN value of 0.4. Using MATLAB’s graythresh function, the value computed for the threshold is 0.4667. The resulting black and white image can be seen in Figure 25 using this MATLAB computed threshold as well. Figure 24 illustrates a maritime color image being transformed by MATLAB to a binary grayscale image. The context of using such transformation is to prepare an image feature extraction.

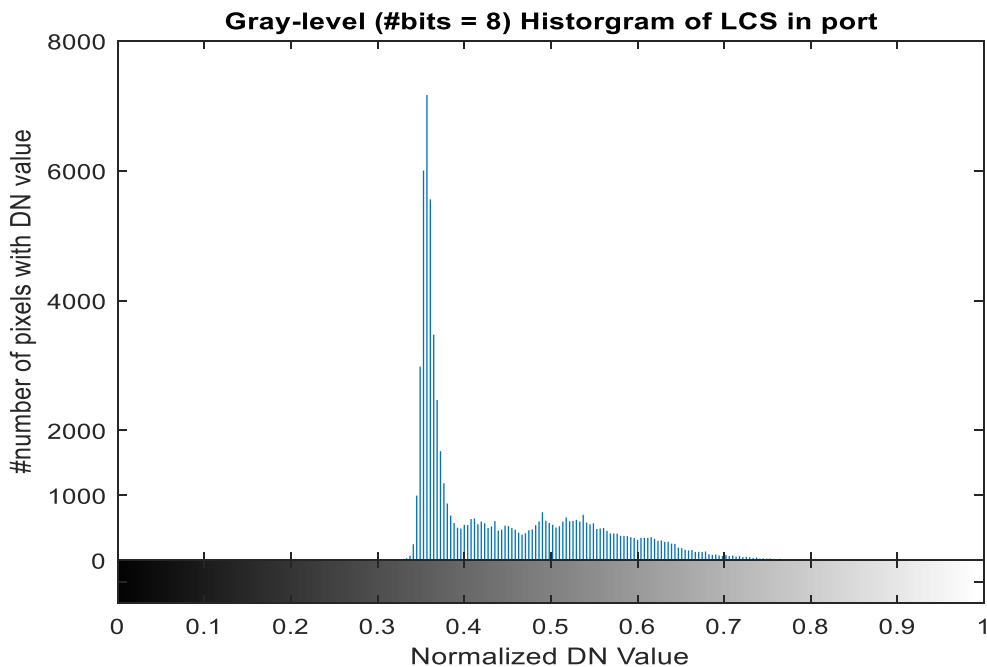


Figure 23. Gray level histogram for Figure 23



Figure 24. MATLAB image transformation from RGB image to binary grayscale.
Source: [60], adapted from [60], respectively.

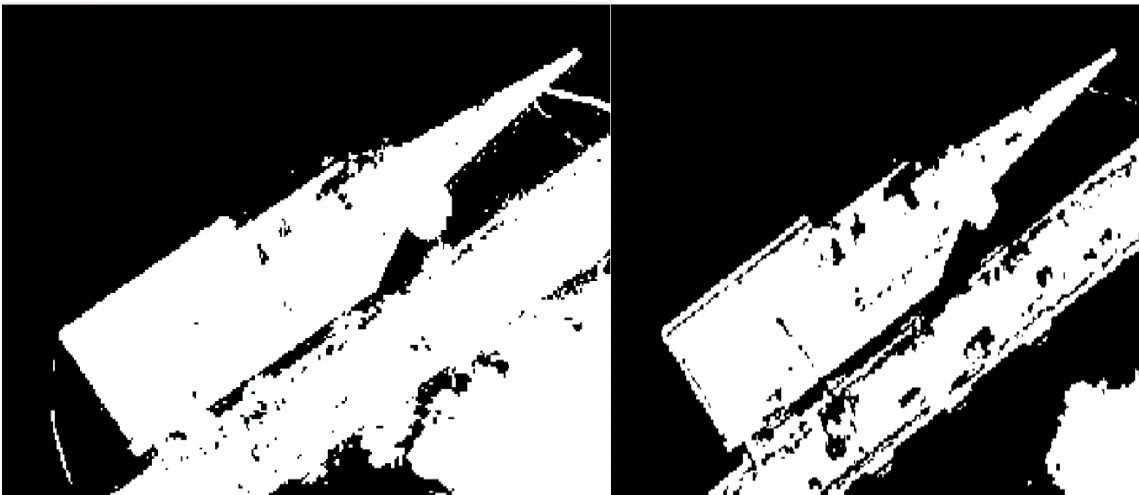


Figure 25. Black and white image of LCS docked at San Diego naval base using threshold of 0.4 (left) vs. using threshold computed from graythresh function (right). Adapted from [34].

The big issue with modeling an MDA classification solution is that it is easy to be lulled into oversimplification. At the same time, simplifying assumptions need to be made for any demonstration. However, image analysis can suffer from completely or mostly ignoring important topics related to image analysis such as image resolution variability effects on classification, the need to resize images to match the size of image

that can be input into a neural network, simplification of thresholding, a very small set of training images used, using a very basic neural network for classification, and other limitations. Higher resolution images contain more information, and the amount of information in an image is directly correlated to a human analyst's ability or a machine's ability to make decisions on what is in that image. Processing an image can help a machine classify an image, but the mechanism that inputs an image into a processing chain, such as a sensor and the distance between the sensor and objects in the image, determines how effective that processing is for classification. The importance of that sensor mechanism cannot be divorced from the processing mechanism.

However, one can build a very simple neural network, ignore resolution concerns so long as imaging sizes are the same, and work from a limited number of images for general types of ships. It is possible because of the generality. Yet, even then the ability to classify in this general sense could be limited to correct classification percentages which are not acceptable and which approach 50% at times for a small single layer neural network [61]. Figures 26–29 will be used in testing the method of using DNNs for classification purposes.

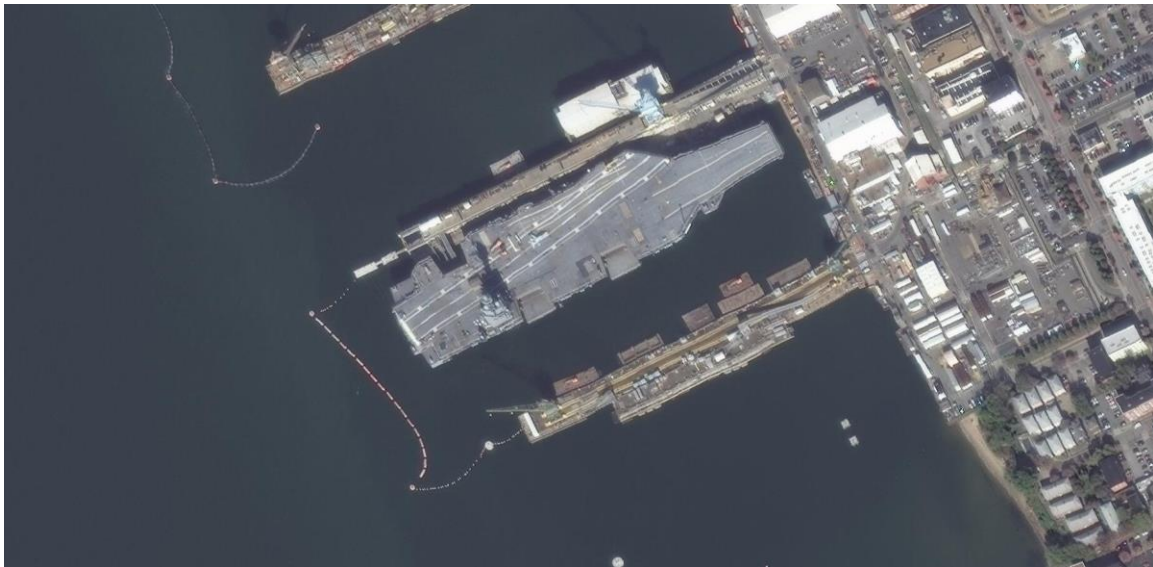


Figure 26. Naval Station Norfolk DigitalGlobe snapshot of carrier in port. Source: [34].



Figure 27. A cropped image of carrier in port at Naval Station Norfolk. Source: [34].

Cropped vs. Uncropped Image Resizing (Left vs. Right)

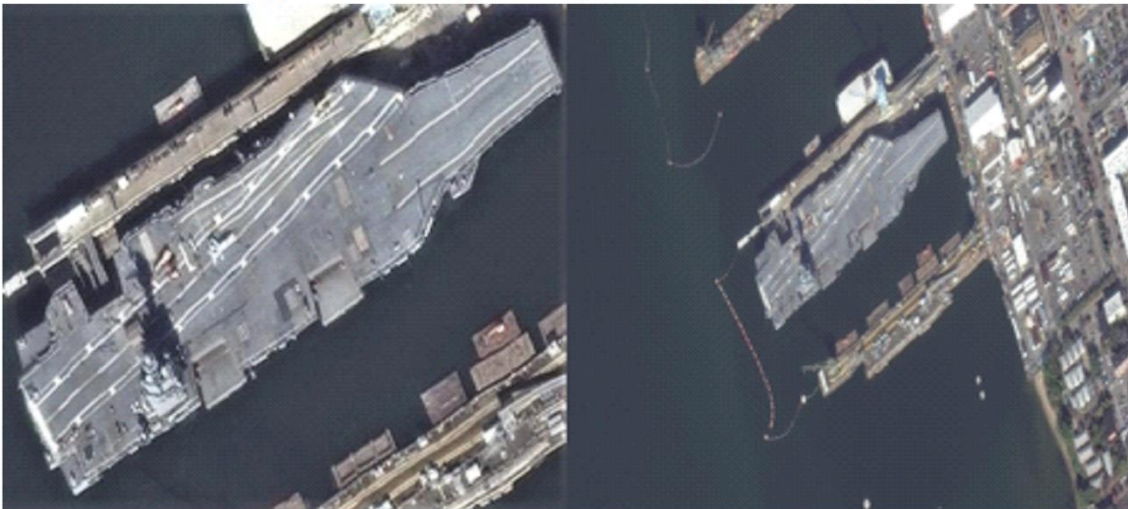


Figure 28. Illustration of why it is better to crop an image before resizing for inputting into a CNN. Source: [34].



Figure 29. Independence Class LCS docked at San Diego naval base. Source: [34].

The position of the ship superstructure was coded as seen in Table 2. That is, the position of the superstructure was given a two-digit number where the first digit indicated the lateral position of the superstructure and the second number indicated the longitudinal position of the superstructure. Also, if the ship has containers it is given a positive value for its “CON” feature value, and if not, it is given a value of zero. Edge detection techniques for the imagery are useful for detecting containers on a ship. This type of classification involves only a one-step logical decision route in learning the ship class rather than a binary method and multi-step decision route of first determining if the object is a ship, then finding out if it is a warship or oil tanker, and then finally ascertaining what specific warship or oil tanker class it is. Such a decision route is illustrated in Figure 30. Together with added MATLAB code relating to testing a ship length criterion (if it is greater than 1000 ft long), the classifier built is able to classify the ships trained for while allowing for small deviations in ship length and width due to measurement error. If the MATLAB program written cannot classify a ship it outputs that the ship is unidentifiable.

Table 2. Coding for the position of ship superstructure in MATLAB

Position of Ship Superstructure Coding			
Lateral Position		Longitudinal Position	
Right	1	Aft	0
Left	2	Middle	1
Middle	No number	Bow	2

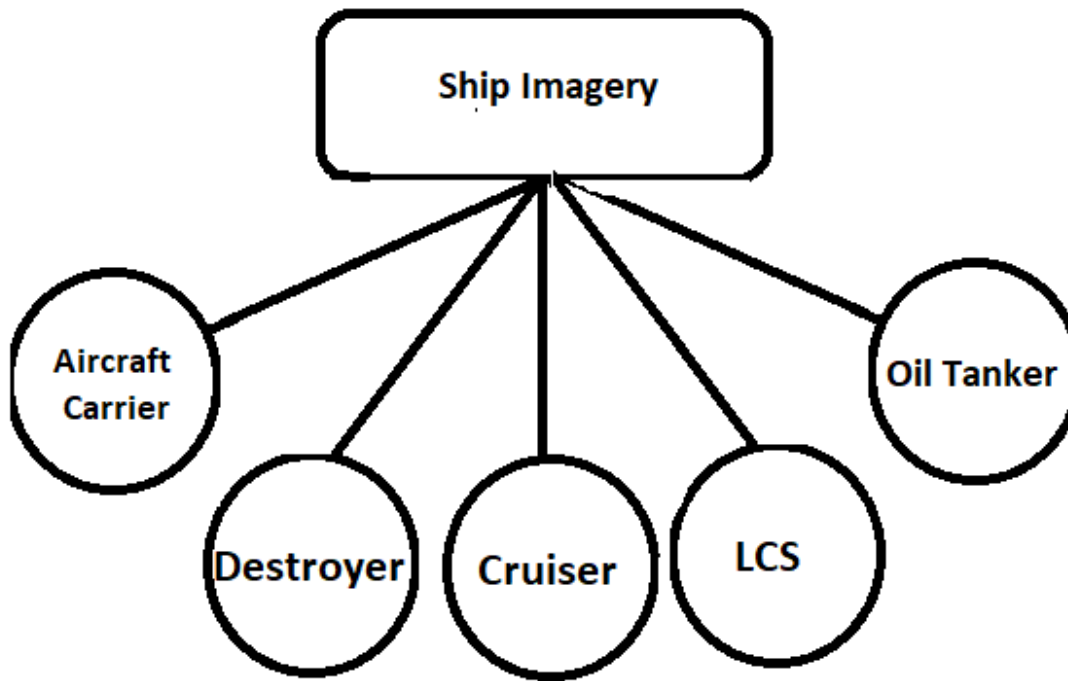


Figure 30. Multi-class classification (one-decision making step)

2. Alexnet, Resnet, Inception-v3, and VGG-16 Comparisons and SVM Results

DNNs like Alexnet, Resnet, Inception-v3, and VGG-16 are software packages that can be loaded onto a computing system to process imagery and identify objects within that imagery. Each one has been trained on databases of imagery and accepts only certain sizes of imagery. The code is short enough in length to use in MATLAB, and with MATLAB type classification software loaded onto a SmallSat, using such DNN

constructs could be used to programmatically implement image processing onboard a SmallSat. These DNNs were used to classify ships to understand their limits without heavily investigating their construction.

Using images from DigitalGlobe's EnhancedView Web Hosting Service along with the ability to crop those images, an already established CNN like AlexNet can be used in MATLAB to determine the type of a ship from that image. These images are the same that were used previously. Cropping the image is important in this case as an image inputted into the Alexnet neural network must be 227 by 227 pixels. Therefore, the act of cropping an image as seen from Figures 26 and 27, can allow for better performance of Alexnet as images with a greater number of pixels as illustrated in Figure 27 suffer much greater resolution loss due to this necessary resizing. Figure 28 illustrates this.

Photographic image resizing and the effects of such is an extensive topic in itself and one which is worth understanding for any image processing applications. Both images were successfully classified in MATLAB as aircraft carriers. However, Alexnet and even Resnet failed to properly classify a destroyer docked at Naval Station Norfolk. They can correctly classify container ships though. Somewhat curiously though, Alexnet classified the Independence Class Littoral Combat Ship in Figure 29 as a slide ruler while Resnet did not much better in classifying it as a "liner," which in this context most likely refers to an Ocean liner. Resnet also classified an oil tanker as a trimaran, which is a yacht with three hulls in parallel. However, it fared better than Alexnet which classified the oil tanker as a lacewing as did VGG-16, which is another DNN that can be used in MATLAB.

More DNNs such as Inception-v3 were used, but wrong classifications for the oil tanker were made. The general justification for this lack of ability in classifying ships is that these DNNs were not trained for such. Therefore, using these DNNs by themselves in their current form will not be sufficient for a ship classification image software architecture. The DNNs simply classify in a broad sense in one step. That is, each DNN appears to place the ability to achieve breadth in identifying objects first, compared to the ability to classify objects further into subcategories. That is, these DNNs may be able to identify an object as a pizza when it is specifically a mushroom pizza. In fact, this was

confirmed in MATLAB using the Inception-v3 DNN that the level of specificity ascertained by the DNN was not a mushroom pizza but just a pizza.

Therefore, in their current form, these DNNs have limited use for deployment on a SmallSat but with a DNN specifically trained for ship classification, a DNN could be built and tested that could potentially greatly expand the ability to classify ships. This would be a somewhat significant undertaking as DNNs like Alexnet require some nontrivial development time. In short, further development is needed in this area before a SmallSat solution for AI image processing using DNNs is deployed. The level of classification ability of DNNs for ship classes is not at a point yet for such.

In terms of the SVM modeling path, it is assumed that the necessary geometric and photometric features have already been determined. That is, a table with geometric and photometric features such as ship length, ship width, ship length/width ratio, and the position of superstructures for ships was generated in MATLAB for various ship classes using vectors for each feature. This is another illustration of where computer vision techniques to determine geometric and photometric features from a ship in an image are used to create features for the SVM to use. Figure 31 show the SVM having been trained in MATLAB's classification learner application for the previously mentioned features. The solid circles indicate correct classifications for five of the ships and the "x's" indicate incorrect classifications for two of the ships based on the trained data set. From Figure 32, the maximum accuracy using fine, medium, and course Gaussian SVMs is 71.4%. Any one of these trained models then can be exported to the MATLAB workspace for use on classifying slightly brand new feature values for an unknown ship of interest. The main drawback of this process is that a vast database of ship class feature values would need to be constructed and trained for.

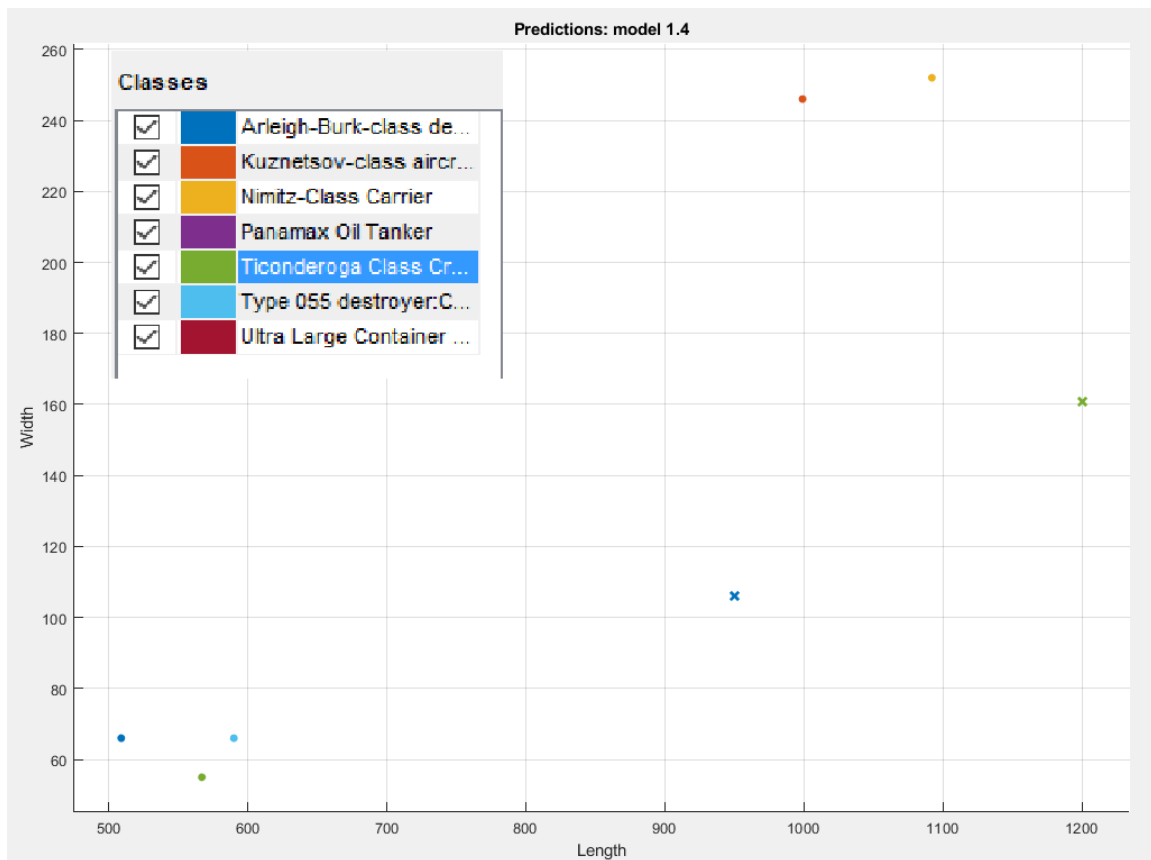


Figure 31. SVM trained in MATLAB apps window training graph for ship width (ft) vs ship length (ft)

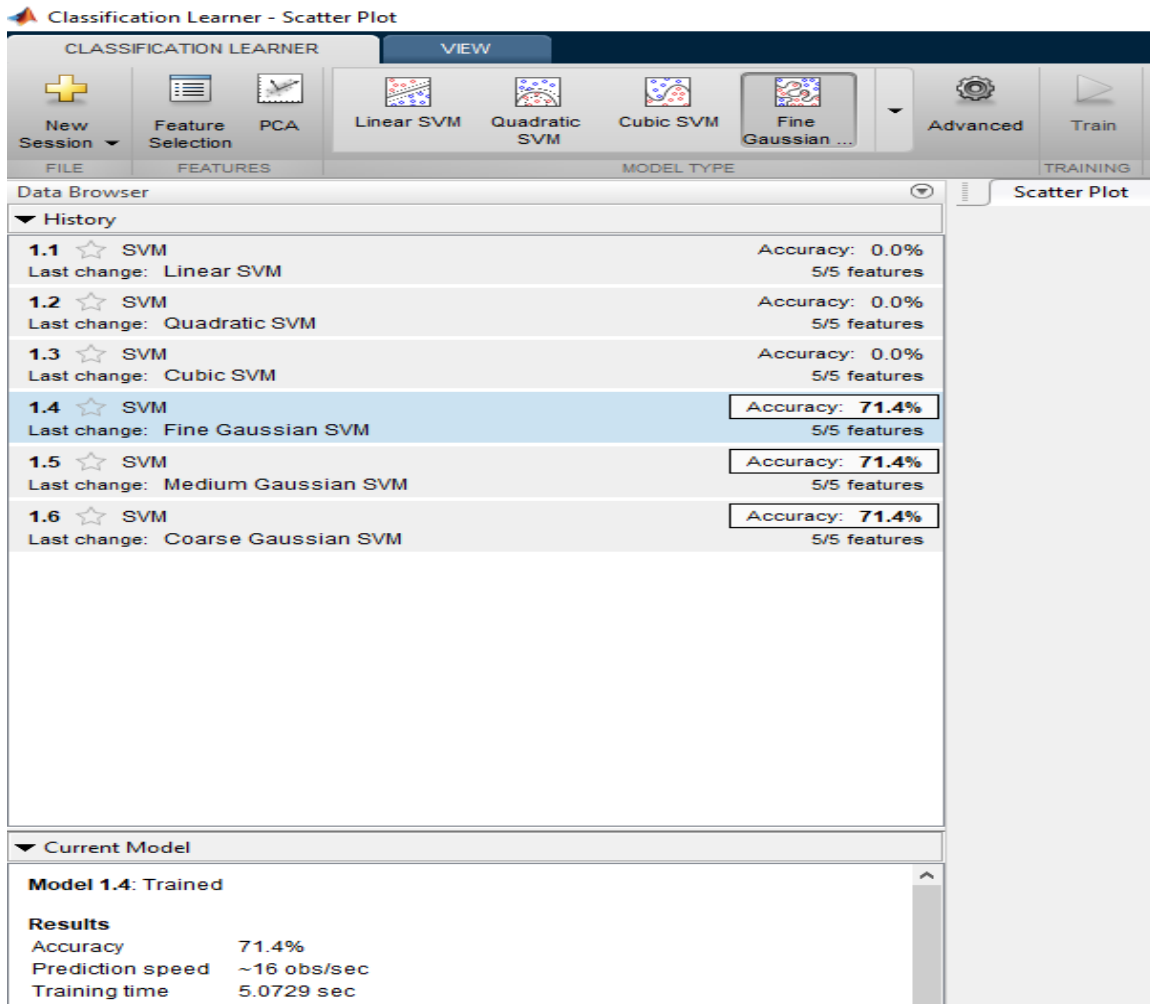


Figure 32. SVM trained for various model types

In conclusion, even after the SVM is run, it may not correctly classify the ship, and through building an SVM for ship classification in MATLAB this was observed. Indeed, a third step was needed to make the model as robust as possible. That step involved checking the ship length again to see if it was greater than 1,000 ft and if it was, the ship was identified as an Ultra Large Container Vessel to prevent misclassification as a Panamax Oil Tanker, and the MATLAB code for such is included in Appendix C.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

The current state-of-the-art in SmallSat EO sensor technology was investigated, and a specific EO sensor built by Harris Corporation was recommended to accomplish a persistent MDA solution that allows for the ability to begin to classify most sea-going vessels of interest. Current state-of-the-art detection and classification hardware and software were also investigated for their capabilities and their ability to be integrated on board SmallSats. Also, different processing methods were outlined with specific levels of capability. Current direct-tasking efforts for SmallSats were highlighted. Artificial intelligence and specific subcategories were discussed to give an overview of current tools to process imagery from a satellite. Specific limitations on these AI methods were demonstrated in MATLAB, and a general method of aggregation of classifier methods was proposed to best handle limitations in any one method. CNNs and SVMs tailored to MDA will need to be both trained and tested. As was elaborated on, SPAWAR Systems Center Pacific is beginning the early stages of such work.

A. SIGNIFICANT CONTRIBUTIONS

A significant contribution in this thesis was showing what is feasible currently for a deployable image processing solution on board SmallSats. Many sources and programs for image processing were investigated. Some were not specifically mentioned because their capabilities were not as advanced as RAPIER and only focused on ship detection. One of the main conclusions that can be drawn is that further development efforts and proof-of-concept for implementing a fully automated TCPED chain for SmallSats is needed, and most specifically, in the area of image processing through artificial intelligence means. Most importantly, it was found that there is not one software program that will allow for classification of ships. The statement that began this thesis rings true. The sensor technology for classification is there, but the processing solution for a completely automated TCPED chain for SmallSats will require many entities working together to develop and field. Specifically, many of the Silicon Valley heavyweights such as Google and Nvidia will play a key role in opening up their technology prowess

alongside DoD contractors and organizations to arrive at a feasible solution that can be deployed. A problem so complex usually warrants such combined efforts. In the near term, some limited capability for on-board processing could be deployed, such as using a ship typing system such as RAPIER SDS.

Another significant contribution in this thesis is showing what a software solution might look like for classifying sea-going vessels. Specifically, what can be done is to integrate computer vision methods, reinforcement learning methods, and supervised learning methods either in a sequential or parallel fashion or combined fashion to minimize classification error. This is an important capability that should receive sustained support so that continuity in advancement is not halted. Systems can age and the knowledge base that once allowed them to progress further can come to a halt.

B. RECOMMENDATIONS FOR FUTURE WORK

One important research area not addressed in great depth in this thesis is the use of other sensor types and platforms such as SAR, IR, and drones to gain further information for MDA. Particularly for SAR sensor research, examining the state-of-the-art in SAR sensory technology would be a first step, followed by cataloging commercial SAR platforms along with their coverage areas and access times. Also, drones could fill in any time gaps in the inability for satellites to detect and identify maritime vessels, and this was not integrated in this discussion. Another application for a drone could be for data storage and forwarding for when a ground station or vessel is ready to receive the information and is within sight.

Another important research area involves the coupling of big data and multisensory data fusion. For example, for a particular part of the open ocean, a database of MDA activity could be maintained on a day-to-day basis. Specific ships and their activities could be tracked, for example, to establish patterns of life. This information could be cross-referenced with satellite classification data to help improve classification confidence.

APPENDIX A. SATELLITE SENSOR FIELD OF REGARD CALCULATION

The calculation of the field of regard for a satellite sensor depends on concisely two variables. These are the pointing angle for the sensor off nadir and the height of the satellite above Earth otherwise known as altitude. The following derivation for the field of regard is an original mathematical construction I developed based upon multivariable calculus:

A further assumption includes modeling the Earth as a sphere with a constant radius. It is known that the Earth though is an oblate spheroid. Further, let “H” represent the sum of the Earth’s radius and the altitude. Figure 33 is a visual depiction of the following mathematical derivation. From this:

$$x^2 + y^2 + (z - H)^2 = R_E^2$$

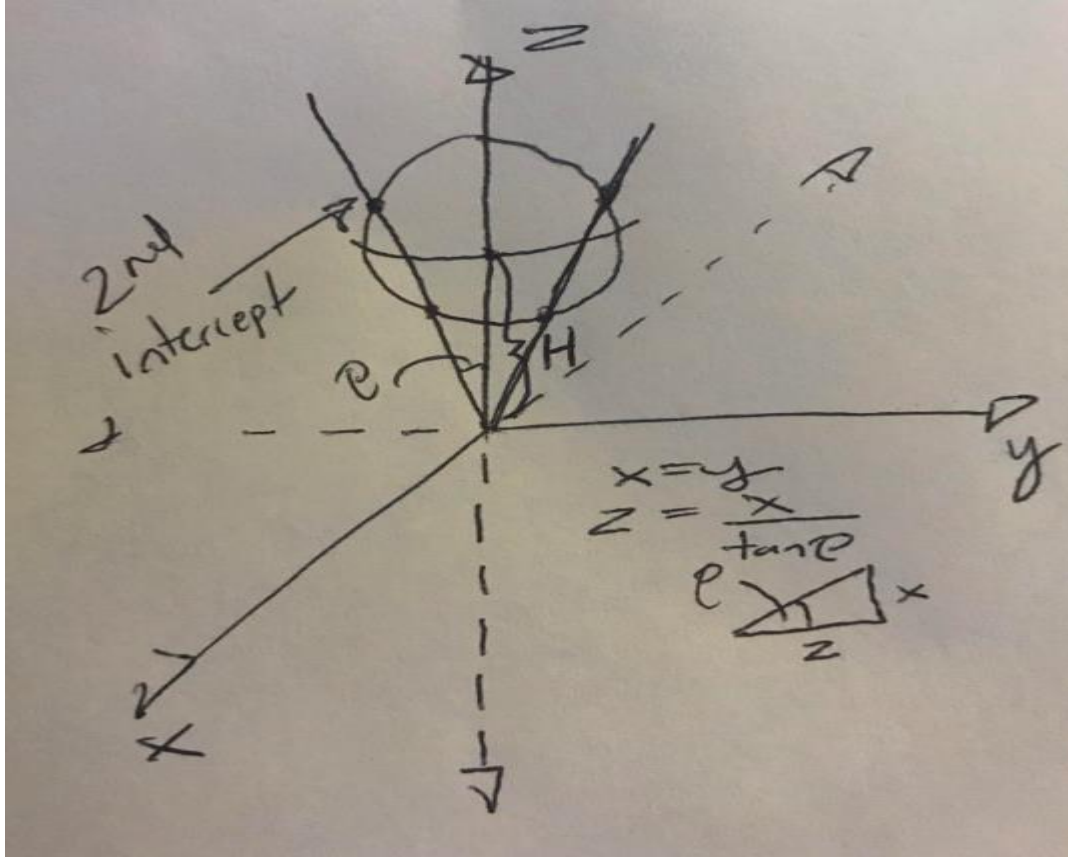


Figure 33. Image for FOR calculation with coordinate system used

This equation can be rewritten as:

$$z = H \pm \sqrt{R_E^2 - x^2 - y^2}$$

The surface area of two-dimensional surface can be expressed as [62]:

$$A(s) = \iint \sqrt{1 + (\partial z / \partial x)^2 + (\partial z / \partial y)^2} dA$$

This can be reduced by appropriate substitutions to:

$$A(s) = \iint r \sqrt{1 + (r^2 / (R_E^2 - r^2))} dr d\theta$$

From Algebra we know that solving a quadratic equation results in the unknown x being expressed as:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Now imagine there is a cone of view that points directly up from the origin of a 3-dimensional space. That cone corresponds to the Field of Regard (FOR) that first intersects the sphere once upon entering and a second time upon leaving the sphere. The first intersection encompasses the FOR.

First, we recognize from the first equation involving the sphere that $x = y$ and $z = x/\tan\varphi$. Let φ be the pointing angle. From these substitutions and solving for x using the quadratic equation, one can show the following:

$$x = \frac{2H / \tan \varphi \pm \sqrt{4H^2 / \tan^2 \varphi - 4(2 + 1 / \tan^2 \varphi)(H^2 - R_E^2)}}{2(2 + 1 / \tan^2 \varphi)}$$

After solving this equation for x ,

$$FOR(x) = \int_0^{2\pi} \int_0^x r \sqrt{R_E^2 / (R_E^2 - r^2)} dr d\theta$$

when H is small and the pointing angle is small as well, the $FOR \approx$ Area of the base of the cone whose height is H .

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. SUPPORT VECTOR MATH

Suppose we have two sets of data differentiated in writing as “+’s” and “-’s.” as shown in Figure 20 [56]. The data are linearly separable as seen by the black line separating the data and the data are in $\mathbb{R}^2$ [56]. Also, the vector $\mathbf{w}$ is perpendicular to the lines [56]. The vector $\mathbf{u}$ can be projected onto $\mathbf{w}$ by using the dot product of both, and as $\mathbf{u}$ becomes greater in magnitude, eventually this projection is large enough to reach the positive sample boundary [56]. Mathematically we can say [56]:

$$\mathbf{w} \bullet \mathbf{u} \geq c$$

Without loss in generality, this equation can be rewritten as [56]:

$$\mathbf{w} \bullet \mathbf{u} + b \geq 0, \text{ if } c = -b$$

This equation is then our decision rule for when we have a positive sample [56]. We do not know which $\mathbf{w}$ or b to use though, which will require us to choose additional constraints [56].

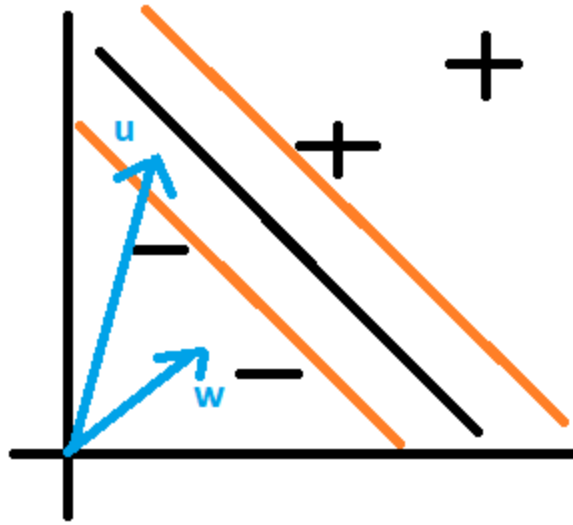


Figure 34. Support Vector Idea. Source: [56].

Suppose we have a positive sample vector $\mathbf{x}_+$ that is known. From this we have [56]:

$$\mathbf{w} \cdot \mathbf{x}_+ + b \geq 1$$

This equation describes what it is to be a positive sample [56]. In a similar fashion for a negative sample vector $\mathbf{x}$ [56]:

$$\mathbf{w} \cdot \mathbf{x}_- + b \leq -1$$

Let y be a unit step function such $y = +1$ for positive samples and $y = -1$ for negative vector samples [56].

$$y(\mathbf{w} \cdot \mathbf{x}_+ + b) \geq 1$$

This equation results for both values of y , being positive or negative [56]. Therefore, in general for a sample vector $\mathbf{x}_i$ the following is true [56]:

$$y(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$$

Subtracting “1” from both sides yields [56]:

$$y(\mathbf{w} \bullet \mathbf{x}_i + b) - 1 \geq 0$$

For samples that end up on the orange lines or in the so-called “gutters,” the equation can be set as an equality [56]:

$$y(\mathbf{w} \bullet \mathbf{x}_i + b) - 1 = 0$$

From a vector perspective, the width of the gap separating the orange lines can be written as [56]:

$$Width = (\mathbf{x}_+ - \mathbf{x}_-) \bullet \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

By using some mathematical manipulation from the last two equations, $\mathbf{x}_+ \bullet \mathbf{w} = 1 - b$ because $y = 1$ in this case [56]. Similarly, $\mathbf{x}_- \bullet \mathbf{w} = -(1+b)$ so that [56]:

$$Width = \frac{2}{\|\mathbf{w}\|}$$

A goal then is to maximize the width which can also be accomplished by minimizing [56]:

$$F(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2$$

The equations for the function “F” and 3rd equation before this together mathematically describe a constrained optimization problem [56]. Mathematically then we can use the method of Lagrange multipliers applied to these functions relating to the two variables “ $\mathbf{w}$ ” and “ b ” [56]. Therefore, in general we have [56]:

$$L(\mathbf{w}, b) = f(\mathbf{w}) - ag(\mathbf{w}, b)$$

We solve for the extrema of this function in this case by solving [56]:

$$\nabla L(\mathbf{w}, b) = \langle \partial L / \partial \mathbf{w}, \partial L / \partial b \rangle$$

$$\nabla L(\mathbf{w}, b) = 0$$

Specifically, $f(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2$ and $g(\mathbf{w}, b) = y(\mathbf{w} \cdot \mathbf{x}_i + b) - 1$. Therefore, we have [56]:

$$L(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum a_i y_i (\mathbf{w} \cdot \mathbf{x}_i + b) + \sum a_i$$

Since $\nabla L(\mathbf{w}, b) = \langle \partial L / \partial \mathbf{w}, \partial L / \partial b \rangle$, we can take the partial derivative of the function L with respect to each variable and set each partial derivative equal to zero to learn some useful properties [56]:

$$\partial L / \partial \mathbf{w} = \mathbf{w} - \sum a_i y_i \mathbf{x}_i = 0 \rightarrow \mathbf{w} = \sum a_i y_i \mathbf{x}_i$$

$$0 = \sum a_i y_i$$

By substituting the last two equations into equation $L(\mathbf{w}, b)$ we can get the maximum value for $L(\mathbf{w}, b)$ [56]:

$$\text{Max } L(a_i) = \sum a_i - \frac{1}{2} \sum a_i a_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

This equation says that maximum of the function is dependent on the inner product or dot product of two different sample vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ [56]. The function is also maximized for nonzero values for the a 's [56]. Looking at the dot product we can see that the function gets maximized when the “x” vectors are close to one another [56]. This makes sense graphically because as the angle between the vectors grows, the width of the channel separating the points shrinks [56]. When y_i and y_j are opposite in value, the function is maximized when the “x” vectors point in the most similar direction or same direction [56].

Linear separation can be achieved by taking the sample vectors in their current vector space and mapping them each through the same transformation function to a higher dimension vector space [56]. Suppose the transformation function is ϕ [56]. The resulting dot product computed in the higher-dimension space would be $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ [56]. The transformation could be of a complicated nature that we need to compute [56]. This is the big challenge but fortunately a function called the Kernel takes the “x” vectors and determines how similar the vectors are in the transformed space such that the transformation function is no longer needed [56]. Therefore, the function in general we want to maximize for the non-linearly separable case is [56]:

$$\text{Max } L(a_i) = \sum a_i - \frac{1}{2} \sum a_i a_j y_i y_j K(\mathbf{x}_i \cdot \mathbf{x}_j)$$

By implementing non-linear kernels that are distinct, different types of SVMs can be defined [56]. That is, an SVM may be polynomially kernel based, sigmoidal kernel based, or Gaussian kernel based [56].

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. MATLAB IMAGE THESIS PROCESSING

Using Alexnet on Figure 18 image to “classify” the ship as an aircraft carrier

MATLAB CODE:

```
deepnet = alexnet
img1 = imread('Carrier2.jpg');
img1 = imresize(img1,[227 227])
figure;
imshow(img1);
pred = classify(deepnet,img1)
```

Figure 24 Histogram Code

```
img1 = imread('lcs.jpg');

I2 = rgb2gray(img1);
one = 1;
I2(I2==0) = one;
I2 = double(I2);
D = size(I2);
D = 255*ones(D);
intensity = I2./D;

figure
imhist(intensity)
x =xlabel('Normalized DN Value');
set(x,'position',get(x,'position')+[-1200,0]);
xtickangle(45)

ylabel('#number of pixels with DN value')
ylim([0 8000])
yticks([0 1000 2000 4000 6000 8000])
yticklabels({'0','1000','2000','4000','6000','8000'})

title('Gray-level (#bits =8) Histogram of LCS in port')
```

SVM features trained for to classify a limited set of ships

```
%in feet
Length = [509;567;1092;590;999;950;1200];
Width = [66;55;252;66;246;106;160.7];
Ratio = Length./Width;

% Position of Superstructure
% Right = 1, Left =2, Middle = no number
% Aft = 0, Middle = 1, Bow = 2

POS = [1;1;10;1;10;0;1];
```

```

% Has containers
CON = [0;0;0;0;0;0;0;100];

Class = {'Arleigh-Burk-class destroyer';'Ticonderoga Class Cruiser';...
'Nimitz-Class Carrier';'Type 055 destroyer:Chinese';...
'Kuznetsov-class aircraft carrier';'Panamax Oil Tanker';...
'Ultra Large Container Vessel'};

T = table(Length,Width,Ratio,POS,CON,Class);

```

Use of SVM trained model to classify an Ultra Large Container Vessel

```

Length = [1001];
Width = [59];
Ratio = [Length/Width];
POS = [0];
[CON] = [1];

newTable = table(Length,Width,Ratio,POS,CON);
if CON == 0
yfit = trainedModel2.predictFcn(newTable)
else if Length >1000
yfit = {'Ultra Large Container Vessel'}
else
yfit = {'Unidentifiable Ship'}
end
end

```

Image resizing code for cropped and uncropped aircraft carrier image

```

I1 = imread('Carrier.jpg');
I = imread('Carrier2.jpg');

I2 = imresize(I, [224 224]);
I3 = imresize(I1,[224 224]);

imshowpair(I2,I3,'montage')
title('Cropped vs. Uncropped Image Resizing (Left vs. Right)','FontSize',24)

```

LIST OF REFERENCES

- [1] “Navy maritime domain awareness concept,” Chief of Naval Operations, U.S. Navy, 29 May 2007. [Online]. Available: www.navy.mil/navydata/cno/Navy_Maritime_Domain_Awareness_Concept_FINAL_2007.pdf
- [2] A. Dupuis and M. Rey, “Leveraging technology trends for maritime domain awareness.” Vanguard, 12 Mar. 2018. [Online]. Available: vanguardcanada.com/2018/03/12/leveraging-technology-trends-for-maritime-domain-awareness/
- [3] S. Bhattacharjee, “Automatic identification system (AIS) integrating and identifying marine communication channels,” Marine Insight. [Online]. Available: <https://www.marineinsight.com/marine-navigation/automatic-identification-system-ais-integrating-and-identifying-marine-communication-channels/>
- [4] R. Olsen, *Remote Sensing from Air and Space*. 2nd ed. Bellingham, WA: Society of Photo-Optical Instrumentation Engineers, 2016
- [5] W.D. Philpot, “Scanners,” Remote Sensing Fundamentals, Fall 2015. [Online]. Available: http://ceeserver.cee.cornell.edu/wdp2/cee6100/6100_monograph/mono_08_F15_scanners%20&%20orbital%20mech.pdf
- [6] “PennState College of Earth and Mineral Sciences: GEOG 480,” Multispectral Remote Sensing Systems. [Online]. Available: <https://www.e-education.psu.edu/geog480/node/494>
- [7] I. Hajnsek, K. Ouchi, M. Armando and M. Sanjuan-Ferrer, “Ship detection with spectral analysis of Synthetic Aperture Radar: A Comparison of New and Well-Known Algorithms,” Remote Sensing. [Online]. Available: <http://www.mdpi.com/2072-4292/7/5/5416/htm>
- [8] “Centre for remote imaging, sensing, and processing (CRISP) National University of Singapore,” SAR Images. [Online]. Available: https://crisp.nus.edu.sg/~research/tutorial/sar_int.htm
- [9] M. Bowes, private communication, Apr. 2018.
- [10] E. Maybrouk, “What are SmallSats and CubeSats?,” Science Instruments, NASA, 26 Feb. 2015. [Online]. Available: www.nasa.gov/content/what-are-smallsats-and-cubesats
- [11] N. Buonaiutoet, et al., “Satellite identification Imaging for small satellites using NVIDIA,” in 31st Annual AIAA/Utah State University (USU) Conference on Small Satellites, 2017. [Online]. Available: <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=3791&context=smallsat>

- [12] "NVIDIA developer," Deep Learning. [Online]. Available: <https://developer.nvidia.com/deep-learning>
- [13] "What is machine learning? A definition." Expert System. [Online]. Available: <http://www.expertsystem.com/machine-learning-definition/>
- [14] "What is computer vision?," The British Machine Vision Association and Society for Pattern Recognition, BMVA. [Online]. Available: www.bmva.org/visionoverview
- [15] S. Das, "CNNs architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more," Medium, 16 Nov. 2017. [Online]. Available: medium.com/@siddharthdas_32104/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5
- [16] K. Sato, "An in-depth look at Google's first tensor processing unit (TPU)," Google Cloud Big Data and Machine Learning Blog, May 12, 2017. [Online]. Available: <https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu>
- [17] "tensorflow." GitHub. [Online]. Available: <https://github.com/tensorflow/tensorflow>
- [18] "tensorflow," TensorFlow.org. Accessed April 10, 2017. [Online]. Available: <https://www.tensorflow.org/>
- [19] C. Metz, "Google just open sourced tensorflow, its artificial intelligence engine," Wired, November 9, 2015. [Online]. Available: <https://www.wired.com/2015/11/google-open-sources-its-artificial-intelligence-engine/>
- [20] N. Jouppi, "Quantifying the performance of the TPU, our first machine learning chip," Google Cloud Platform Blog, Google, 5 Apr. 2017. [Online]. Available: cloudplatform.googleblog.com/2017/04/quantifying-the-performance-of-the-TPU-our-first-machine-learning-chip.html
- [21] S. Park, et al. "Direct tasking operation design for SpaceEye-X system," presented at the SpaceOps Conference AIAA., 2016, pp. 1–5.
- [22] G. Krebs, "Kestrel eye 2M (kestrel eye block 2M)," Gunter's Space Page. [Online]. Available: space.skyrocket.de/doc_sdat/kestrel-eye-2m.htm
- [23] "KE-IIM (Kestrel eye block-II microsatellite)," EoPortal Directory, ESA. [Online]. Available: directory.eoportal.org/web/eoportal/satellite-missions/k/kestrel-eye

- [24] "SpaceView small satellite imaging solutions," Harris, Harris Corporation, 2017. [Online]. Available: https://www.harris.com/sites/default/files/downloads/solutions/55523_spaceview_brochure_4_v9_el_final_web.pdf
- [25] "SpaceView small satellite imaging solutions," Harris, Harris Corporation, 2017. [Online]. Available: <https://www.harris.com/sites/default/files/downloads/solutions/55475-spaceview-24-35-42-datasheet-final-web.pdf>
- [26] R.D. Fiete, "Image quality and LFN/p for remote sensing systems," SPIE Digital Library, Commercial and Government Systems, 6 Apr. 1999. [Online]. Available: www.spiedigitallibrary.org/journalIssue/Download?fullDOI=10.1117/1.602169
- [27] M. Grauss, "House panel wants answers on DigitalGlobe licensure delay," Space News, Digital Globe, 6 June 2016. [Online]. Available: spacenews.com/house-panel-wants-answers-on-digitalglobe-licensure-delay/
- [28] "SWIR imagery," DigitalGlobe, Maxar Technologies, 2018. [Online]. Available: www.digitalglobe.com/products/swir-imagery
- [29] "Sensors." Section 2: Acquiring Remote Sensing Data, Ames Remote. [Online]. Available: www.amesremote.com/section2.htm
- [30] J. Pike, "National Image Interpretability rating scales," FAS Intelligence Resource Program, Federation of American Scientists, 16 Jan. 1998, 7:49:10 AM. [Online]. Available: fas.org/irp/imint/niirs.htm
- [31] "The National Imagery Interpretability Rating Scale (NIIRS)," National Collection of Aerial Photography. [Online]. Available: ncap.org.uk/feature/national-imagery-interpretability-rating-scale-niirs
- [32] "Ticonderoga class: air defense cruiser," Military-Today.com. [Online]. Available: http://www.military-today.com/navy/ticonderoga_class_cruiser.htm
- [33] "IKONOS satellite image of Norfolk Naval Base," Satellite Imaging Corporation. [Online]. Available: <https://www.satimagingcorp.com/gallery/ikonos/ikonos-norfolk-virginia/>
- [34] "EnhancedView web hosting service," DigitalGlobe. [Online]. Available: <https://evwhs.digitalglobe.com/myDigitalGlobe/login>
- [35] K. Varnavas, et al. "The use of field programmable gate arrays (FPGA) in small satellite communications systems," NASA. [Online]. Available: ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150009433.pdf
- [36] "The ultimate list of GIS format – geospatial file extensions," GISGeography, Feb 19, 2018. [Online], Available: <https://gisgeography.com/gis-formats/>

- [37] “FPGA acceleration of convolutional neural networks,” Nallatech. [Online]. Available: www.nallatech.com/fpga-acceleration-convolutional-neural-networks/
- [38] “ProASIC3.” Microsemi, Microsemi Corporation, 2018. [Online]. Available: www.microsemi.com/product-directory/fpgas/1690-proasic3#ds
- [39] S. Knapp, “Field-programmable gate array.” Frequently Asked Questions (FAQ) About Programmable Logic, OptiMagic Inc., 14 May 2000. [Online]. Available: www.fpga-site.com/faq.html
- [40] H. Buck, “RAPid image exploitation resource (RAPIER) ship detection system,” [Online]. Available: <https://community.apan.org/wg/nmio/m/nmiodocs/190770/download>
- [41] RAPIER (RAPid Image Exploitation Resource) ship detection system,” U.S. Navy, July 2010. [Online]. Available: http://www.public.navy.mil/spawar/Pacific/technology/Documents/RAPIER_SDS_TIP_Sheet_-_Final_Draft.pdf
- [42] SSCPacific. YouTube, YouTube, 23 Sept. 2009. [Online]. Available: www.youtube.com/watch?v=GOWVFdgdBWc
- [43] “Keyhole markup language,” Google Developers, November 14, 2013, [Online]. Available: https://developers.google.com/kml/documentation/kml_tut
- [44] T.V. Wilson, “How surfing works,” Adventure, HowStuffWorks. [Online]. Available: adventure.howstuffworks.com/outdoor-activities/water-sports/surfing5.htm
- [45] C. Formeller, “Let’s clear the air: DigitalGlobe atmospheric compensation (AComp), part 2,” DigitalGlobe. [Online]. Available: <http://blog.digitalglobe.com/technologies/lets-clear-the-air-digitalglobe-atmospheric-compensation-acomp-part-2/>
- [46] F. Pacifici, N. Longbotham and W. J. Emery, “The importance of physical quantities for the analysis of multitemporal and multiangular optical very high spatial resolution images,” in IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 10, pp. 6241–6256, Oct. 2014 [open access]
- [47] J. Descloitres, “Haze over Eastern China,” Visible Earth: A catalog of NASA images and animations of our home planet. [Online]. Available: <https://visibleearth.nasa.gov/view.php?id=70122>
- [48] “Flock 1: Planet – flock imaging constellation,” eoPortal Directory. [Online]. Available: <https://directory.eoportal.org/web/eoportal/satellite-missions/f/flock-1>

- [49] A. Nazre and R. Garg, “A deep dive in the venture landscape of artificial intelligence and machine learning,” SlideShare, LinkedIn Corporation, 31 Aug. 2015. [Online]. Available: www.slideshare.net/ajitnazre/deepdive-in-aiml-venture-landscape-by-ajit-nazre-rahul-garg
- [50] G.C. Stockman and L.G. Shapiro. *Computer Vision*, 1st ed. Upper Saddle River, NJ: Prentice-Hall, 2001.
- [51] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002
- [52] “Classification: introduction, nearest centroid and k nearest neighbors,” Store 390: Introduction to Data Science. [Online]. Available: <https://idc9.github.io/stor390/notes/classification/classification.html>
- [53] U. Karn, “An intuitive explanation of convolutional neural networks,” The Data Science Blog, WordPress, 11 Aug. 2016. [Online]. Available: ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
- [54] “Introduction to deep neural networks (deep learning),” DL4J, SkyMind, 2017. [Online]. Available: deeplearning4j.org/neuralnet-overview
- [55] “Neural network graph,” Latex Stack Exchange, Stack Exchange Inc., 2017. [Online]. Available: tex.stackexchange.com/questions/353993/neural-network-graph/354001
- [56] R. Berwick, “An idiot’s guide to support vector machines (SVMs),” CAP 6412 - Advanced Computer Vision, 28 Jul. 2012. [Online]. Available: <http://www.cs.ucf.edu/courses/cap6412/fall2009/papers/Berwick2003.pdf>
- [57] E.W. Weinstein, “Hyperplane,” From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/Hyperplane.html>
- [58] M. Paul, “Feature creation and selection,” Applied Machine Learning, University of Colorado Boulder, 24 Oct. 2017. [Online]. Available: cmci.colorado.edu/classes/INFO-4604/
- [59] “Classification learner,” MathWorks®. [Online]. Available: <https://www.mathworks.com/products/statistics/classification-learner.html>
- [60] “OneAtlas sandbox,” Airbus. [Online]. Available: <https://sandbox.intelligence-airbusds.com/web/>
- [61] “Improving maritime domain awareness using neural networks for target of interest classification,” Naval Postgraduate School Dudley Knox Library. [Online]. Available: <https://calhoun.nps.edu/handle/10945/45252>

- [62] J. Stewart. *Calculus: Early Transcendentals*. 4th ed. Pacific Grove, CA, USA: Brooks/Cole Publishing Company, 1999.

INITIAL DISTRIBUTION LIST

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