CONTEXTUAL DETECTION OF ANOMALIES WITHIN HYPERSPECTRAL IMAGES

THESIS

Adam J. Messer, Captain, USAF

AFIT-OR-MS-ENS-11-15

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-OR-MS-ENS-11-15

CONTEXTUAL DETECTION OF ANOMALIES WITHIN HYPERSPECTRAL IMAGES

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Adam J. Messer, BS

Captain, USAF

March 2011

AFIT-OR-MS-ENS-11-15

CONTEXTUAL DETECTION OF ANOMALIES WITHIN HYPERSPECTRAL IMAGES

Adam J. Messer, BS
Captain, USAF

Approved:

//Signed	14 Mar 11
Dr. Kenneth W. Bauer (Chairman)	date
//Signed	14 Mar 11
Maj Mark A. Friend, PhD (Reader)	date

AFIT-OR-MS-ENS-11-15

Abstract

The majority of anomaly detectors in Hyperspectral Imaging use only the statistical aspects of the spectral readings in the image. These detectors fail to use the spatial context that is contained in the images. The use of this information can yield detectors that out perform their spatially myopic counterparts. To demonstrate this, we use an independent component analysis based detector, autonomous global anomaly detector (AutoGAD), developed at AFIT augmented to account for the spatial context of the detected anomalies. Using segmentation algorithms, the anomalies identified are formed into regions. The size and shape of these regions are then used to decide if the region is anomalous or not. A Bayesian Belief Network structure is used to update a posterior probability of the region being anomalous.

Acknowledgments

First off, I'd like to thank my wife who went to great lengths to give me the support, time, and space required to concentrate on my thesis. She is my hero.

I would also like to thank my committee and all the members of the Sensor Fusion Lab. Especially, my advisor, Dr Kenneth Bauer, and my reader, Major Mark Friend, for their guidance and assistance in the entire process.

Adam J Messer

Table of Contents

List of Figures

List of Tables

CONTEXTUAL DETECTION OF ANOMALIES
WITHIN HYPERSPECTRAL IMAGES

1. Introduction

1.1. Background

Recent technological advancements have provided a vast increase in the amount of remote sensing data that is produced. The shear amount of this data surpasses the human capital that can be allocated to analyze the data for useful target information. Hyperspectral images contain massive amounts of data about a scene. Accurate and efficient algorithms are required to identify anomalies in a timely manner for further processing.

Hyperspectral imaging has a wide range of applications within remote sensing, not limited to terrain classification, environmental monitoring, agricultural monitoring, geological exploration, and surveillance (Stein et al., 2002). Within the Department of Defense, the use of hyperspectral remote sensing within the application of surveillance faces growing demand.

1.2. Problem Statement

Current detection algorithms identify anomalous pixels based on the relation of the spectral signature to the background of the image. There are two major types of algorithms for anomaly detection, local and global spectral anomaly detectors. Local spectral anomaly detectors often have increased false positive rates when a small piece of a different background class is surrounded by a separate background class; whereas global spectral anomaly detectors suffer from increase false negative rates when the anomalous pixels lie within the distribution caused by a highly variable background

(Stein et al., 2002). By using the specific context of the anomalies, we show more favorable performance can be attained by using local spatial information with a global detector.

1.3. Methodology

The processing that a human image analyst applies to the detection of targets within an image is very different than the process that is applied through modern anomaly detection algorithms. One primary difference is that a human analyst takes the context of a potential anomaly into account. The likelihood of anomaly being present depends on the size of the anomaly, the type of material, and many other contextual clues.

In this research, we develop an anomaly detection algorithm which utilizes modern detection approaches along with the spatial and spectral context of the identified anomalies. This approach differs from the state of the art because more than just the statistical difference of an anomaly is taken into account to make the final determination of target likelihood.

1.4. Research Objectives

1) Creation of an anomaly detection algorithm post processor that rejects targets based on spatial context
2) Provide ability to augment current anomaly detection algorithms to increase fidelity of results

1.5. Preview

This thesis contains five main sections: an introduction, a literature review, methodology, results, and a discussion. The introduction provides the basic overview that the thesis will follow. The literature review is broken into major areas of HSI research to include HSI basics, anomaly detection, segmentation, and contextual anomaly detection. The methodology section details the process taken to include spatial context into an anomaly detection algorithm. The results section shows the performance of the spatial context sensitive algorithm on hyperspectral images. The final section outlines the contributions our research provides to the anomaly detection discipline.

2. Literature Review

Recent decades have seen advancements to the application of multivariate anomaly detection to hyperspectral imagery. This chapter outlines the significant contributions applicable to contextual anomaly detection. The chapter is organized into five sections: Hyperspectral Imaging (HSI) Basics, Anomaly Detection in HSI, Segmentation, Contextual Anomaly Detection, and Bayesian Belief Networks.

2.1. Hyperspectral Imaging (HSI) Basics

Hyperspectral imaging (HSI) is a subset of the domain of digital imaging. The most basic form of a digital image is a black and white image from a camera. A B&W digital image displays the relative intensity level light in the pixel. By displaying these relative intensities in their spatial relation an image can be viewed. A color image can be thought of three monochromatic images merged together with different wavelength bands being used to represent what our eyes see as red, green, and blue. An ordinary digital camera essentially collects three images. In order to view the color image the three monochromatic images are overlaid with the relative intensity of each color. When a hyperspectral image is created the scene is recorded with up to 250 wavelength bands. These bands normally extend from the visible region (0.4-0.7 μm) into the near infrared region of the electromagnetic spectrum (0.7-2.5 μm) (Landgrebe, 2003). Some hyperspectral sensors are configured to collect mid-wave and long-wave infrared (2.5-15 μm) (Shaw & Manolakis, 2002). Figure 2-1 shows the segment of the electromagnetic spectrum used for hyperspectral imaging. The increased number of collected wavelengths

allow for the comparison of materials that would not be distinguishable with a lower number of collected wavelengths (Shaw & Manolakis, 2002).

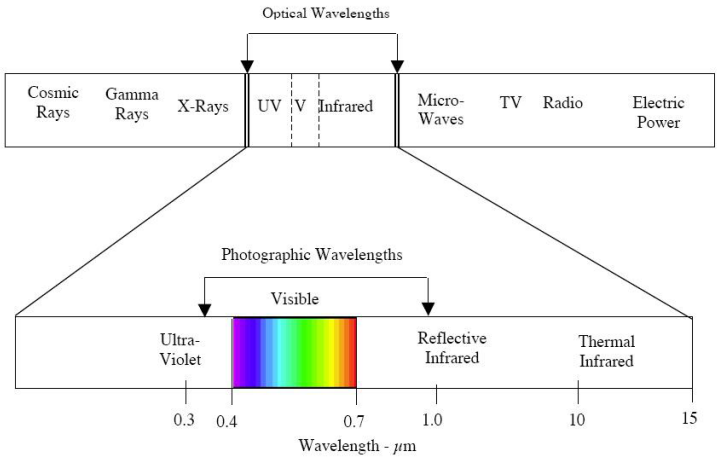


Figure 2-1: The Electromagnetic Spectrum (Landgrebe, 2003)

2.1.1. Data Organization

Hyperspectral Images are traditionally organized into a 'spectral cube.' This format is a 3-dimensional array with height (i), width (j), and wavelength band (k). A spectral sample can be taken from any pixel, depicted in Figure 2-2 as the extraction to the left. Selection of a wavelength allows for a grayscale image to be viewed, depicted with the extraction to the right in Figure 2-2.

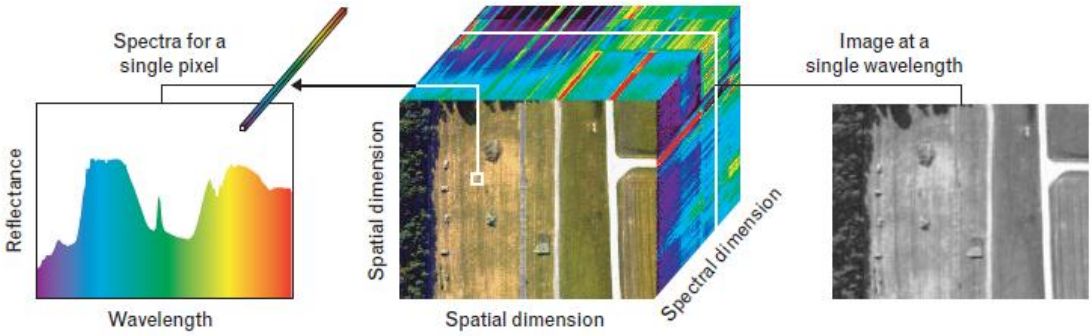


Figure 2-2: Hyperspectral Image Cube (Manolakis, 2003)

During processing, the hyperspectral data cube is normally reorganized to a two-dimensional data array with the rows holding the i*j pixel pairs and the columns holding the k collected spectra, Figure 2-3.

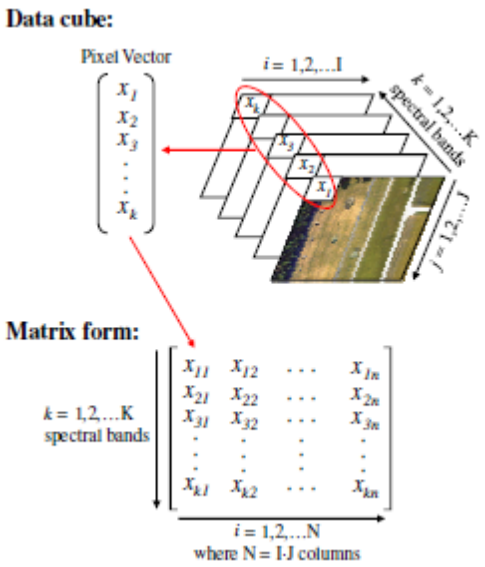


Figure 2-3: Mapping of Pixels to Matrix Form (Miller, 2009)

Once this transformation is complete, the data matrix can be treated as a multivariate database without the loss of the spatial information. This is crucial in the final steps of anomaly detection when the anomalous pixels are mapped back to their original spatial location.

Depending on the characteristics of the sensor, aperture size and altitude, the ground pixel resolution of HSI images varies from a few meters to tens of meters. This directly impacts the type of algorithms that can be executed successfully on the data. When targets are fully resolved in at least one pixel, the spectral signature of that pixel will represent the target. Sub-pixel targets can be very difficult to detect as the target spectra are mixed with some proportion of background spectra. There are many other challenges to the identification of target pixels such as atmospheric interference of the

image, energy scattering and absorption, and viewing and measurement angles (Shaw & Manolakis, 2002).

2.2. Anomaly Detection in HSI

In order to exploit the information provided by HSI, the images must be processed. Historically, this has been done primarily with intense human interaction. The development of anomaly detectors to process the images decreases the commitment on human interaction. Sometimes when an image is collected certain spectral signatures are of interest. If this is the case the detection problem is simpler. The image can be scanned for spectral signatures similar to the ones of interest. Detection of targets within a hyperspectral image without the use of *a priori* knowledge of the target's spectral signatures is more difficult, but more widely applicable, than supervised target detection (Ashton, 1998). In its simplest form anomaly detection is concerned with alerting the analyst that a target or a potential target is contained in the image. Identification of the target class is obtained by subsequent means. Two major classes of anomaly detectors exist: global and local. Each has its own pitfalls. Global spectral anomaly detectors often fail to detect anomalies that resemble the background in the image while local spectral anomaly detectors are susceptible to high false alarms on clutter surrounded by a different background class (Stein et al., 2002).

2.2.1. Anomaly Detection Approaches

The amount of data within a hyperspectral data cube lends itself to techniques that find lower dimensional subspaces that allow for the identification of anomalies. Often the image is oversampled, meaning the spectral signature of the scene can be represented

by a subset of spectral bands (Shaw & Manolakis, 2002). One such approach to anomaly detection through the utilization of lower dimensional subspaces is the application of Principal Component Analysis (PCA). PCA is a multivariate statistical analysis tool that seeks to maximize the amount of variance represented in sequential orthogonal components (Bauer, 2010). The first component holds the maximum amount of variance, the second, orthogonal to the first, holds the maximum amount of the remaining variance, and so on. The use of PCA allows the number of dimensions analyzed to decrease dramatically while still retaining a significant amount of the data variance.

2.2.2. AutoGAD

In his thesis, Johnson uses a variety of techniques to develop an autonomous algorithm for the detection of anomalies in hyperspectral images. Johnson's Autonomous Global Anomaly Detector (AutoGAD) uses a four-phase approach to extract the features that separate anomalies from the rest of the image.

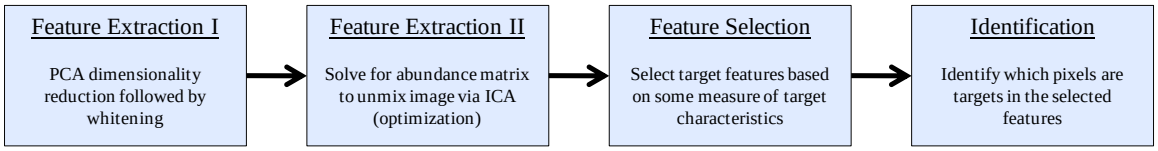


Figure 2-4: Process Flow for Target Detection (Johnson, 2008)

In the first phase, Johnson uses PCA to extract the significant components. In the second phase, those significant components are rotated to become independent using an Independent Component Analysis (ICA) approach. These components contain similar types of objects within the image. For example, ICA may result in three components with roads, vegetation, and targets each represented individually. The third phase selects potential target components from the ICA. The component scores are binned in a histogram and the first bin to contain no observation, the zero bin, is deemed the signal

threshold level. By comparing the noise level to the signal level and the amplitude of the signal level in the component, components are deemed to contain either targets or noise. The fourth step identifies individual pixels in target components that are targets. Target pixels are those that fall above the zero-bin threshold from the third step (Davis, 2009). More information on the third and forth steps of AutoGAD is available in chapter 3.3.

2.3. Segmentation

Segmentation in image processing is the separation of the image into two or more regions. Many techniques and approaches exist for the partitioning. Bieniecki and Grabowski present a possible approach to the segmentation of anomalies beyond that of just grouping neighboring pixels. Their process could quickly group the anomalies into a set of like anomalies, i.e. all the trucks in a group, all the tarps, etc. (Bieniecki & Grabowski, 2004). The application of their algorithm is beyond the scope of this thesis since our aim is to interact with a global anomaly detector.

A simple neighboring pixel approach for image segmentation is applied as a post-process on the anomaly detection mask of our output. Pixels are joined into regions based on their relationships to neighboring pixels. This is accomplished using the ‘regionprops’ function built into the MATLAB image processing toolbox (Mathworks, 2010). An example, of this method applied to a simple black and white image, Figure 2-5, shows 12 segmentable objects.

Figure 2-5: Segmentation Example

2.4. Contextual Anomaly Detection

Most anomaly detection methods are purely statistical based on a single point compared either to the local or global background (Chandola, Banerjee, & Kumar, 2009). The goal of contextual anomaly detection is to use a set of contextual attributes to distinguish anomalous pixels. A simple example of a contextual attribute is the size of the anomaly. If the application of HSI dictates that anomalies should be relatively small, a large region of contiguous anomalous pixels can likely be ignored. A human image analyst continually uses such contextual clues in order to process images. Anomaly detection results can be improved by utilizing the data that is contextually contained within the image.

Chandola et al. make the observation that there has been relatively little research done in the realm of contextual, or conditional, anomaly detection (Chandola et al., 2009). This could possibly be due to the very specific nature of what can be called contextual attributes within a multivariate data set. Based on literature available, none

take into account the actual spatial characteristics of an anomalous region (size, shape, location, e.g.). Some of the literature assumes that the data comes with a contextual identification of the variables (Song, Wu, Jermaine, & Ranka, 2007). Sans this identification, Xiang Wang and Davidson make the astute observation that identifying the context of an anomaly becomes part of the overall problem of anomaly detection (Xiang Wang & Davidson, 2009). The rest of this section addresses three contextual anomaly detection techniques present in the literature. Following the summary of the techniques a comparison is provided.

2.4.1. Conditional Anomaly Detection (Song et al., 2007)

In their paper, Conditional Anomaly Detection, Song et al. approach the problem of conditional anomaly detection by first defining two sets of attributes: environmental attributes and indicator attributes. This identification is done using specific knowledge of the data set and can be viewed as an input to the model. Song et al. then use the technique of maximum likelihood estimator (MLE) to find a model that can be applied to a set of data.

Song et al. use this MLE in a learning algorithm on a set of historical data to generate a parameter set. This parameter set can then be applied to new sets of data that are assumed to be from the same population as the historical data set.

They put forth an example, Figure 2-6, where the indicator attribute is the number of fevers and the environmental attribute is the maximum daily temperature. Without conditional anomaly detection only Point A could be detected. With the introduction of the environmental attribute Point B can also be seen as an anomaly.

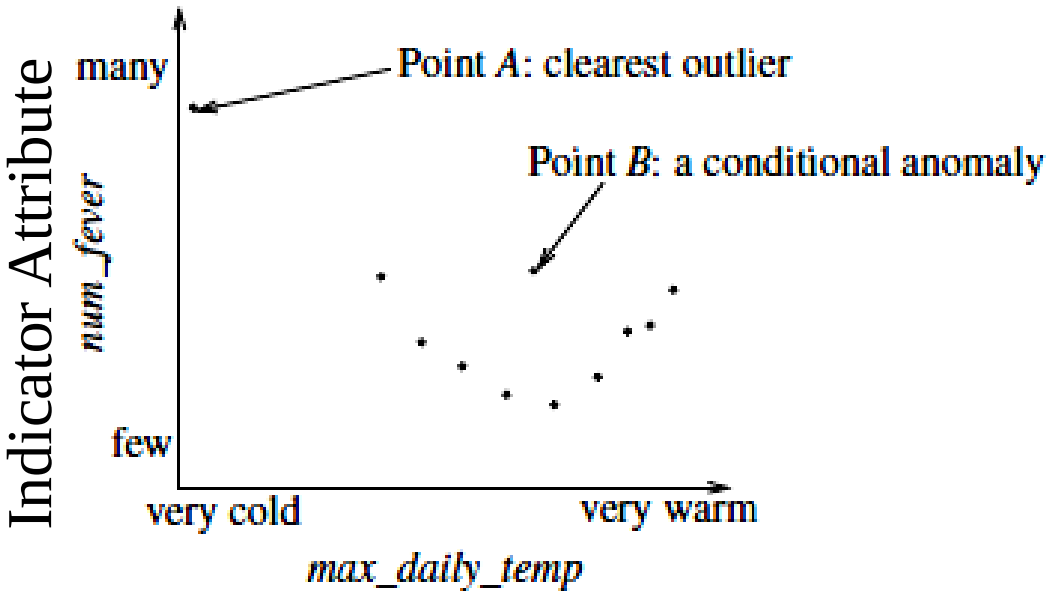


Figure 2-6: Conditional Anomaly Detection Example

The approach that Song et al take for conditional anomaly detection has a very limited application to our desire for automatic target detection. The environmental and indicator attributes are not only difficult to identify but change dependent on image and target characteristics. The approach of using information from the anomalies themselves, other than the statistics of their spectral distribution, could provide a benefit to HSI anomaly detection as we include size or location as environmental indicators.

2.4.2. Detection of Subpixel Anomalies in Multispectral Infrared Imagery Using an Adaptive Bayesian Classifier (Ashton, 1998)

An ingenious way to include contextual information is to compare a pixel to its surrounding pixels. In his paper, Ashton is interested in detecting anomalies in multispectral images that do not resolve to a complete pixel. This is a challenge for many anomaly detectors because the statistics of an anomalous pixel could only partly be filled

with anomalous spectral signatures. The statistics of the pixel will appear to be very close to the background statistics. In order to increase performance, Ashton presents a process to compare a pixel to its four adjacent pixels. He calls the groups of pixels cliques. Figure 2-7 shows the organization of Ashton's cliques. This information is used in a Bayesian process to update the probabilities that the pixel is anomalous.

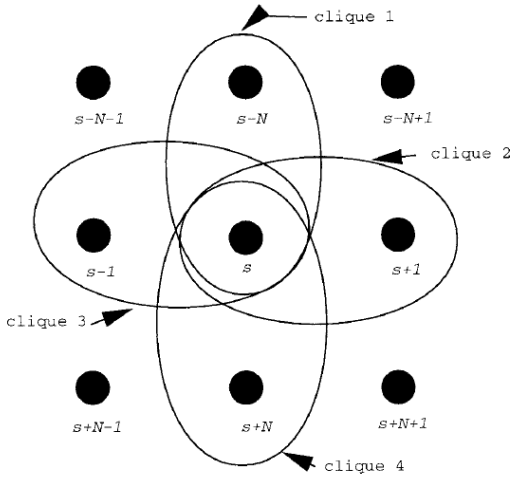


Figure 2-7: Two-pixel clique membership of pixel s (Ashton, 1998)

Ashton's approach to including the context of a pixel into the detection algorithm works in as a local approach; meaning that anomalies are declared from just a subsection of the image. This works well with images containing subpixel targets, with highly resolved targets some anomalous pixels would be surrounded by other anomalous pixels limiting the application of the algorithm.

2.4.3. On Local Spatial Outliers (Sun & Chawla, 2004)

Sun and Chawla feel that the stability of the area around an outlier affects the probability of it actually being an anomaly. If the area around an anomaly is highly unstable, then the anomaly may be from the same distribution versus another distribution. In a stable area an apparent anomaly is more likely to be truly an anomaly. Sun and

Chawla define a measure of a point's relation to the surrounding points: Spatial Local Outlier Measure (SLOM). SLOM combines information about a pixel's relation to its local neighborhood with the volatility of a pixel's neighborhood. Sun and Chawla state that a pixel with a high value in the neighborhood is a good candidate for an anomaly while the same pixel value in a region of other high values may not be anomalous. To calculate SLOM two components are combined: the measure of a point's relation to its neighborhood and a measure of the pixel's neighborhood volatility.

According to Sun and Chawla, the SLOM approach increases the ability to detect local outliers while suppressing the report of global outliers. This comes at the cost of moving away from a global detection algorithm. Both subparts of the SLOM calculation must be calculated for each point in the image. Sun and Chawla's approach only includes a single non-spatial variable; this would require the expansion of the approach to multivariate in order for it to be applied to HSI.

2.4.4. Summary of Current Practices

The majority of the current practices within anomaly detection rely solely on the statistical nature of the points. Within the realm of HSI, this is disregarding a vast amount of data that is present in the spatial information. A few anomaly detectors, like those above, attempt to include some context into the process, but bring additional issues. This thesis exploits the statistical output of an anomaly detector, specifically AutoGAD, using spatial information to improve the performance of the detector. Further, we introduce the approach of declaring anomalous regions of pixels versus declaring anomalous pixels. Anomalies are selected on the context of the region in which they are present.

2.5. Bayesian Belief Networks

Decision structures for the inclusion of spatial context with the statistical nature of a region are numerous. One we have selected to apply is Bayesian Belief Networks (BBN). BBN, also called causal networks or just belief nets, are an approach to discriminant decision theory when the parameters of the probability distributions are unknown (Duda, Hart, & Stork, 2001). Instead of requiring complete probability knowledge of a system, BBN only requires the knowledge of the causal relationships between the variables. By utilizing the knowledge of the causal relationships and Bayes Rule, the posterior probabilities can be found using the evidence present in child nodes. This simplifies the calculations as the conditional probabilities for every relationship are not required, only those that have a causal relationship. Probability relations between unconnected nodes are done using the rules of probability. An example may ease the understanding of the benefits of using the BBN structure. Our network, Figure 2-8, contains four Boolean nodes (Bauer, 2011).

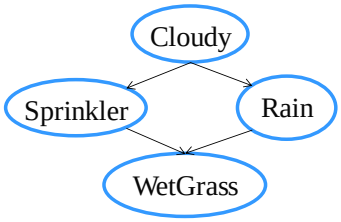


Figure 2-8: Example Bayesian Belief Network

In our BBN, we are interested in the connection between the weather and the wetness of the grass. The parent node ‘cloudy’ has two dependent children nodes, ‘sprinkler’ and ‘rain’. These both share ‘wet grass’ as a child node. These relationships imply a structure. For example, the state of ‘clouds’ directly impacts whether it rained or the sprinklers were used. Suppose we want to know the probability of it being cloudy

given the grass is wet. Using the conditional distributions in Table 2-1, we can calculate the probability relatively simply using Equation 2-1.

Cloudy	P(Cloudy)	P(No Clouds)
Prob	0.5	0.5

	P(Sprinkler)	P(No Sprinkler)
Cloudy	0.5	0.5
Not Cloudy	0.9	0.1

	P(Rain)	P(No Rain)
Cloudy	0.8	0.2
Not Cloudy	0.2	0.8

		P(Wet Grass)	P(Dry Grass)
No Sprinkler	No Rain	1	0
Sprinkler	No Rain	0.1	0.9
No Sprinkler	Rain	0.1	0.9
Sprinkler	Rain	0.01	0.99

Table 2-1: Example Conditional Probabilities

$$P\ \ Cloudy \mid Wet\,Grass\ = \frac{\sum_{s,r} p\ \ c,r,s,w}{\sum_{s,r,c} p\ \ c,r,s,w} = \frac{\sum_r p(w \mid r,s) \sum_s p\ \ r|c\ \ p\ \ s|c\ \ p(c)}{\sum_c \sum_r p(w \mid r,s) \sum_c p\ \ r|c\ \ p\ \ s|c\ \ p(c)} \quad \textbf{(2-1)}$$

Using the equations we find the probability that it was cloudy is 0.5758. This was done without complete calculations of the conditional probability table for ‘cloudy’ given ‘wet grass.’ This simplification of calculations becomes critical once we begin training our network with observational data.

3. Methodology

The methodology chapter outlines the process we propose to include spatial context in the anomaly decision. Our approach relies heavily on the work done previously to develop AutoGAD (Johnson, 2008). This chapter consists of seven sections: Proposed Algorithm, HYDICE HSI Images, Nature of AutoGAD output, Contextual Region Information, Contextual Thresholding, Bayesian Belief Network (BBN) Formulation, and Test and Training Image Results.

3.1. Proposed Algorithm

The proposed algorithm is designed as post processor to the current AutoGAD algorithm. We will cover each step in more detail in this chapter.

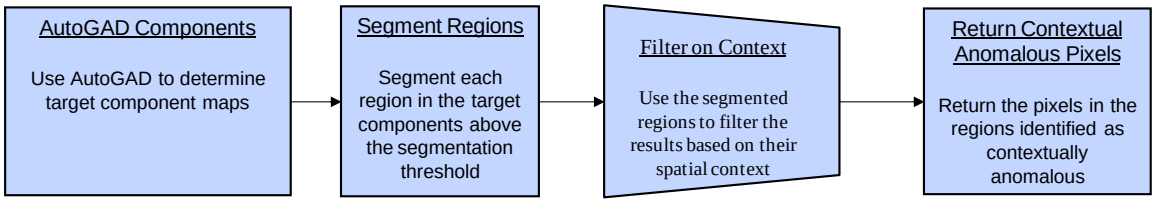


Figure 3-1: Proposed Algorithm Flow

3.1.1. AutoGAD

Our proposed algorithm begins with the output from AutoGAD after the identification of the target components. Essentially this allows us to start with components that already have a bias towards highlighting target regions.

3.1.2. Segmentation

In the segmentation process each component is segmented individually. This allows for overlapping segments to be individually considered. If a road component

overlapped with tanks on a road there would be the possibility of eliminating the target region because it was too large even though it contained the tank regions. Next, the components are normalized to each of their zero-detection histogram method thresholds. This allows for a direct comparison between components. Regions are identified with pixels above a normalized value, 0.8 for example. A simple segmentation algorithm is imposed that groups neighboring pixels utilizing the built-in MATLAB 'regionprops' function.

3.1.3. Filter on Context

In contrast to other anomaly detectors that identify anomalous pixels solely on their relationship to the other pixels in the image, our method uses the characteristics of a region of pixels to identify if the region is anomalous. Either contextual thresholding or the BBN structure can be used to identify the anomalous regions. In contextual thresholding, regions are eliminated from the target class when one of their contextual indicators lies beyond the threshold. In the BBN structure, a region is eliminated if the posterior probability of the region is below some threshold. We use the threshold of 0.6 as a functioning level. This threshold could be used in future research to determine the best setting.

3.1.4. Return Contextual Anomalous Pixels

After segmentation and the identification of anomalous regions, the algorithm returns the pixels present in the anomalous regions. Pixels are not identified as anomalous purely on their own values, but also on the characteristics of the region in which they are present.

3.2. HYDICE HSI Images

The HYDICE (Hyperspectral Digital Imagery Collection Equipment) sensor is an airborne Hyperspectral Imaging (HSI) Sensor. This study focuses on 14 images from the Airborne Remote Sensing (ARES) dataset taken with the HYDICE sensor. Two collection events from 1995 are used in this thesis: Forest Radiance I and Desert Radiance II. Table 3-1 shows the available images, their characteristics, and the assigned set for training, test, and validation. After elimination of atmospheric absorption bands, there are 145 bands remaining. The training set is required for the training of the Bayes Net conditional probabilities. For AutoGAD and Contextual Thresholding the training and test set are combined since there is no training for those algorithms. The validation set is excluded from algorithm training and parameter selection to allow for the observation of unturned performance. It should be noted that the number of neighborhood pixels (not including target) is a count of pixels that are not counted as target or non-target pixels. These pixels are neighbors to the target pixels and may be sub-pixel target pixels. In order to control for this the pixels are counted in any performance metrics. True color images and truth maps are available in Appendix A.

Table 3-1:ARES Image Characteristics

Set	HYDICE IMAGES	*PROPERTIES*							
		Size	**Bands**	**Number of Pixels**	**Target Pixels**	**Number of Neighborhood pixels (not including target)**	**Total Targets**	**Scene Type**	**Altitude**
Training	**ARES1F**	191x160	210	30560	1007	973	10	Forest	5,000' AGL
	ARES2F	312x152	210	47424	307	1221	30	Forest	5,000' AGL
	ARES3D	156x156	210	24336	438	155	4	Desert	10,000' AGL
	ARES5	355x150	210	53250	585	1041	15	Forest	5,000' AGL
Test	**ARES2D**	215x104	210	22360	523	1942	46	Desert	5,000' AGL
	ARES3F	226x136	210	30736	145	314	20	Forest	5,000' AGL
	ARES4F	205x80	210	16400	109	339	29	Forest	5,000' AGL
Validation	**ARES1C**	203x108	210	21924	0	0	0	Forest	5,000' AGL
	ARES1D	291x199	210	57909	235	437	6	Desert	5,000' AGL
	ARES2C	124x198	210	24552	0	0	0	Forest	5,000' AGL
	ARES4	460x78	210	35880	882	1524	15	Desert	5,000' AGL
	ARES5F	470x156	210	73320	1077	1664	45	Forest	5,000' AGL
	ARES6D_10kFT	215x77	210	16555	144	221	13	Desert	10,000' AGL
	ARES7F_10kFT	161x88	210	14168	384	292	12	Forest	10,000' AGL

3.3. Nature of AutoGAD output

The components that AutoGAD identifies as target components often contain a collection of segmentable groups of pixels. The nature of these segments represents a unique feature within the image. When this identification is accurate the identification of a target reflects the actual shape of the real world object at the sensor's resolution. When this identification is incorrect the algorithm is identifying a collection of non-targets that seem anomalous. In the environment defined for this thesis, a generally target sparse environment, the statistics for target and non-target pixel react characteristically different. AutoGAD exploits this characteristic difference to attempt to correctly identify target components. In order to identify target maps, AutoGAD applies a dual filter of having both a SNR ratio above 2 dB and a max pixel score greater than 10. However, AutoGAD functions completely without the spatial information of the component maps. Each pixel

is identified to be an anomaly solely on the nature of the independent components and how the pixel scores.

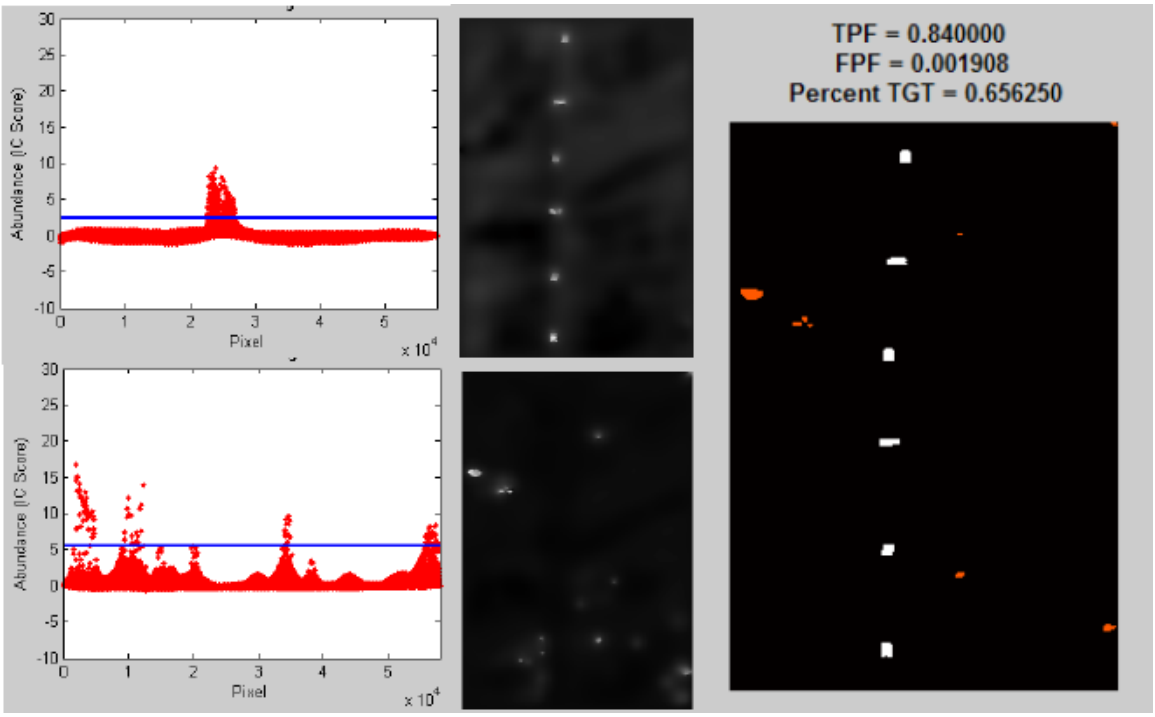


Figure 3-2: ARES 1D: Target Identification (Johnson, 2008)

Returning the statistical information to the spatial realm allows for the plotting of the component maps. These maps display the same information as the component score scatter plots plus the spatial relation of the points. Each Independent Component has a different map, see 3.3.1.

3.3.1. Normalization of Component Scores

In order to compare the output of different components from ICA, they need to be normalized to the same scale. Johnson showed that the threshold value for each component to identify target pixels varies (Johnson, 2008). To remedy this, in our algorithm the output of each component is normalized using the zero-bin threshold level.

Each pixel score is divided by the threshold for the component. Subsequently, pixels with normalized scores above one are identified as targets on all components.

Finding a means to exploit the spatial differences contextually demands a means to measure the difference. For example, a pixel with a normalized value of 0.8 in a region filled with highly anomalous pixels may be anomalous itself; however, the same pixel with non-anomalous neighbors may just be noise. This is the problem that Sun and Chawla attacked with their SLOM measure (Sun & Chawla, 2004). The method that we propose for this is the application of mean intensity of the region to account for the statistical measure of the anomalous region.

3.3.2. Segmentation of AutoGAD Output

In order to produce spatial regions from AutoGAD output the normalized component scores are thresholded to a value. Thresholding at one results in the same regions that AutoGAD would identify originally. By lowering the threshold larger regions of potential targets can be identified. This allows the inclusion of pixels on the low side of the zero-detection method technique threshold. A defining characteristic of target maps becomes apparent when compared to non-target components. The contours within a target map expand at a much slower rate than the contours on a non-target map as the threshold of segmentation is lowered. Figure 3-3 shows the results when the segmentation threshold is lowered on three different components, two background and one target. The two background components display regions that grow rapidly as lower thresholds are segmented. The target component, however, displays uniform size as the threshold is lowered. This characteristic difference between background regions and

target regions is exploited using the mean intensity of the region to measure it is determine its statistical anomaly decision.

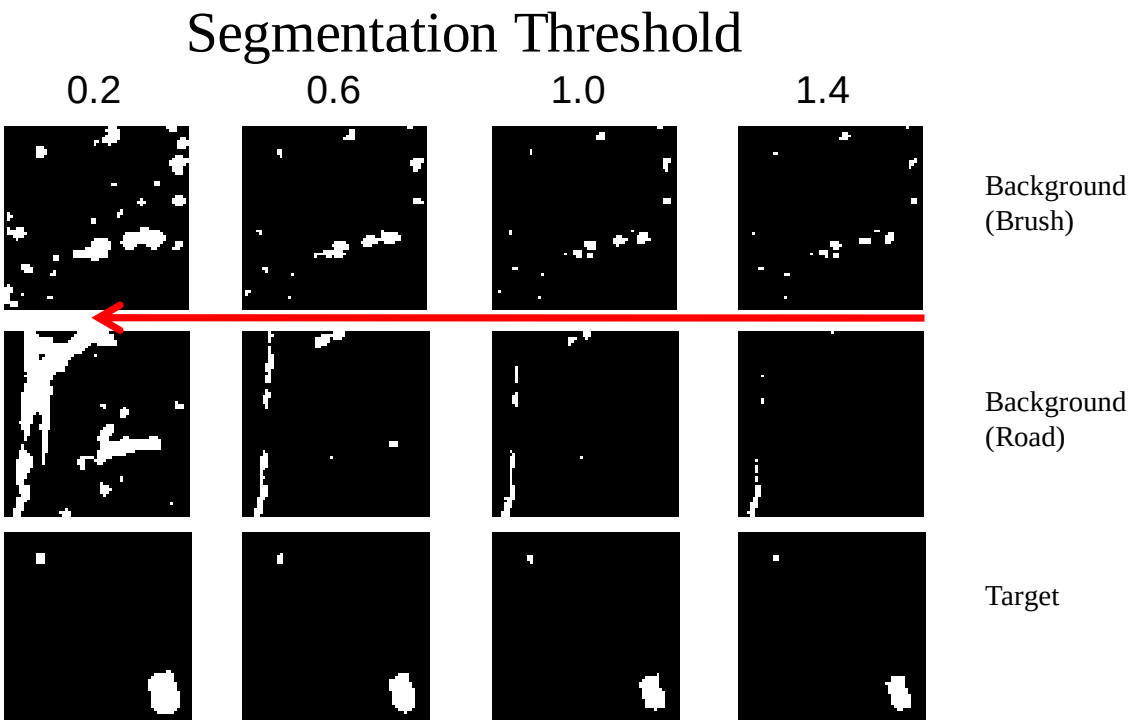


Figure 3-3: Regions from Varying Threshold on Different Component

3.4. Contextual Region Information

After segmentation, a number of contextual values can be extracted from the anomalous pixels. For the remainder of this section the following segmented region, Figure 3-4 and Table 3-2, of 7 pixels is used as an example:

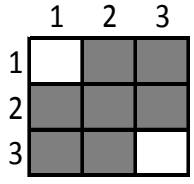


Figure 3-4: Example Region

Pixel	X	Y
1	1	2
2	1	3
3	2	1
4	2	2
5	2	3
6	3	1
7	3	2

Table 3-2: Example Region Pixel Locations

With the regions supplied from the segmentation of the components we propose using four measures of the regions, three spatial and one statistical. The three spatial measures, area, aspect ratio, and bulbosity, measure the size and shape of the potential anomalous regions. The forth measure, mean intensity, uses AutoGAD's pixels scores to determine the statistical value for a potential anomalous region.

3.4.1. Area

Assuming that the sensor altitude is much greater than the difference in altitude of the objects being sensed, the area of each pixel is relatively equal. Therefore, the number of pixels flagged as targets within a segmented object is relative to the actual area of the real world object. This is often measured with ground sample distance and can vary from a meter to above ten meters.

Since the amount of area for a region can only be approximated to the level of pixel resolution, the area of a target in an image is highly variable. An image taken with a space asset will display different characteristics then an image taken with an airborne asset. The images used for thesis are all taken from approximately 5,000-10,000 feet above ground level using the HYDICE sensor.

Area is highly dependent on the type of sensor and the means of employment. As the altitude of the sensors increases the area of each pixel is increased. In the extreme,

targets within an image may not be fully resolved within a pixel, meaning that only a portion of a pixel represents the target, i.e. subpixel. This results in low bound area being useless as a contextual marker when sub-pixel targets are present. The level of appropriate thresholding on area depends on the operating characteristics at the time of image collection.

3.4.2. Aspect Ratio

The aspect ratio of an object is defined as the ratio between its length and width. This is calculated using the first and second principal components. This forces the longest axis possible on the region to be the major axis and the perpendicular axis to be the minor axis. As a result the value of aspect ratio is always greater than or equal to one. In our example region, Figure 3-5, the value of the first principal component is 3.879 measured 45° off vertical and the minor axis length is 2.43 measured -45° off vertical. The resulting aspect ratio is approximately 1.6. The ratio of the principal components supplies the aspect ratio since the ratio is unit-less and we do not need the actual length of the major and minor axes.

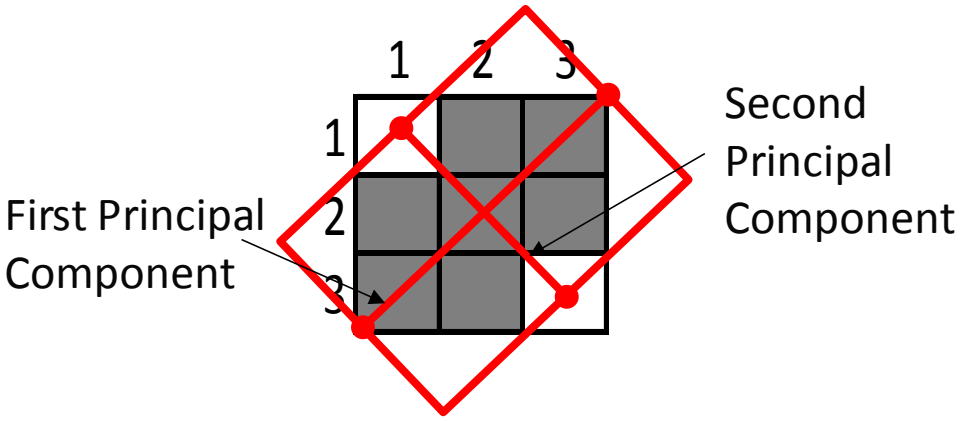


Figure 3-5: Example Region Axes'

The use of aspect ratio to determine the target potential of a region is a based on the majority of targets displaying close to square regions. For example, an anomalous region with an aspect ratio greater than 5:1 for example could be considered highly irregularly shaped and is most likely reflective of a natural object.

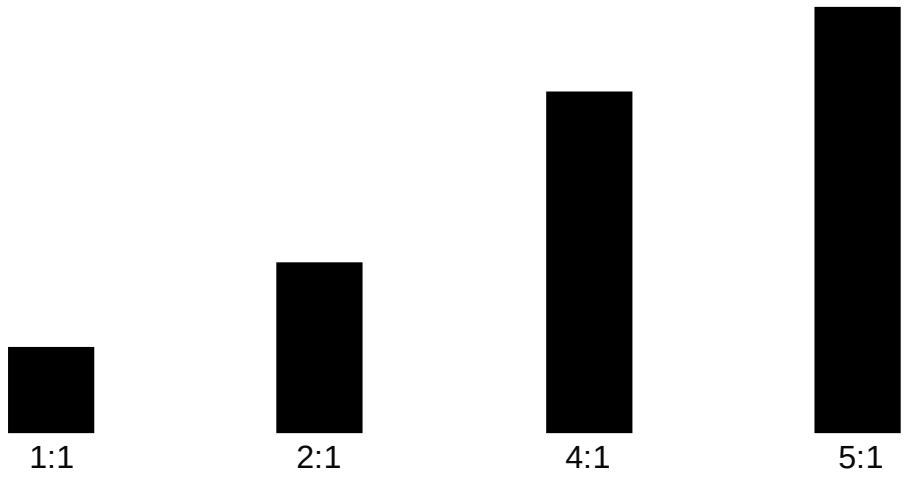


Figure 3-6: Aspect Ratio Examples

3.4.3. Bulbosity

The chaotic nature of natural systems is diametrically different from the order that manmade systems often display. This is clear when two landscapes are compared, one of a natural valley and the other as cityscape. The straight, protruding lines of the cityscape are very different from the rounded edges that mark the natural scene. Figure 3-7 shows an example of the relatively regular shape of the targets versus the bulbous segments of the brush. The manmade objects in the image display straight, perpendicular edges while the natural surroundings display bulbous shapes with few regions of straightness or perpendicularity.

Figure 3-7: ARES4 True Color (Extract)

To calculate the bulbosity of an object the following is proposed:

$$\text{Bulbosity} = \frac{\text{Major Axis} * \text{Minor Axis}}{\text{Area}} \qquad (3\text{-}1)$$

Bulbosity is a measure of how irregular a segmented region presents itself within an image. The method of calculation of bulbosity of an object limits values to being greater than or equal to one. The area resulting from the multiplication of the axes will create a box to include all pixels in the region. Our example region has a bulbosity of 1.3, Figure 3-8.

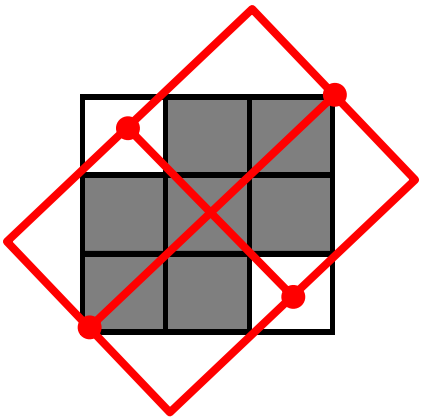


Figure 3-8: Example Region Bulbosity Representation

Figure 3-9 shows another example of bulbosity of a region. This region has a highly level of bulbosity at 1.83

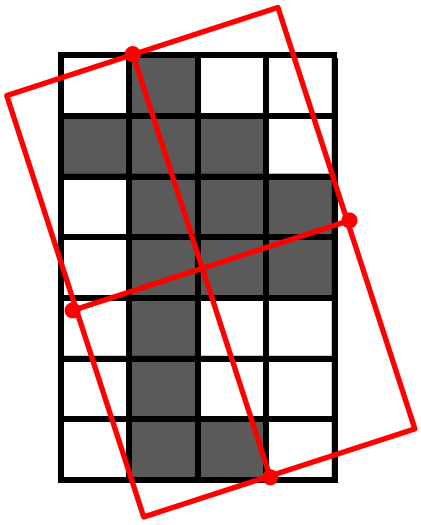


Figure 3-9: Bulbosity Index Example

$$\text{Bulbosity} = \frac{7.37 * 3.4515}{14} = 1.83 \quad \textbf{(3-2)}$$

3.4.4. Mean Intensity

The fourth contextual value that is gleamed from the segmented data is the mean intensity of the region. This is a measure of the regions statistical level of anomalousness. If our example region was built off pixel values found in Figure 3-10 the mean intensity of the region would be 2.79.

	1	2	3
1		2	2.5
2	1.5	3	4
3	2.5	4	

Figure 3-10: Example Region Intensities

This value, since it is built using the AutoGAD component pixel score, is used as a proxy for whether AutoGAD would call the region an anomaly. Each pixel above one

represents a pixel that AutoGAD would have identified originally. When the threshold is lowered some pixels that AutoGAD would not have identified are included. If these pixels are in the region of pixels with high values the region retains a high mean intensity. If the region is of low values, say all between the threshold and 1, the mean intensity is lower.

3.5. Contextual Thresholding

After segmentation and calculation of the context values, the regions can now be filtered based on their contextual spatial and statistical values. We propose four different filters: mean intensity, aspect ratio, minimum area, and bulbosity. The filters eliminate regions that lie beyond the thresholds set for them. For example, the mean intensity threshold could be set to eliminate regions with mean intensities less than one. Each filter is designed to control for different types of errors. The first three filters are attempts to use the spatial context of the regions to remove anomalies that are actually just background pixels while mean intensity relies on the statistical information provided from the regions relation to the image.

1. Minimum Area:

   Filtering on minimum area controls for single pixel noise that may be flagged as anomalous. Using a lower than prescribed threshold than the zero-detection method allows for more pixels to be included in the segmentation regions. The level of this threshold is very dependent on the circumstances of which the image was taken. In certain circumstances a very low, perhaps zero, minimum area threshold would be appropriate, i.e. very low spatial resolution

images. However, this same threshold might be more appropriate as a maximum filter value on images taken from very low altitudes.

2. Aspect Ratio:

This filter acts to eliminate regions with high aspect ratios. The majority of mobile military targets have relatively square shapes. A region displaying a very high aspect ratio is probably a region of road or natural feature that is being picked up. An example of this can be seen in the road component of ARES3D, Figure 3-3.

3. Bulbosity:

The use of the bulbosity index to filter regions is another use the shape of a region to determine its target worthiness. An object with high bulbosity is displaying very irregular shape within the image. This is one indication that the region is of natural means. As stated earlier, target regions should have bulbosity index values close to one, meaning that they are regularly shaped.

4. Mean Intensity:

When the threshold was lowered below the original zero-detection method level we introduced some pixels that were not originally anomalous according to AutoGAD. To control for the risk of false positives, we propose a filter on the mean intensity of the region. This allows for highly anomalous pixels to carry borderline pixels to the point of identification. Conversely, pixels not in a highly anomalous region will again be removed from consideration.

The simple threshold structure gives way to a decision tree depicted in Figure 3-11.

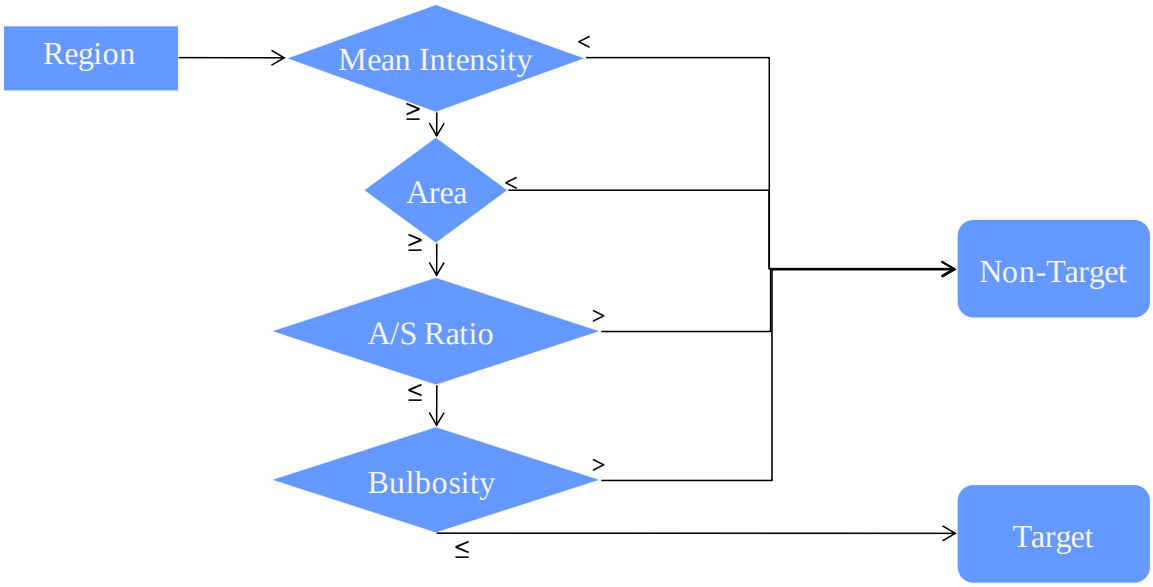


Figure 3-11: Contextual Thresholding Decision Tree

An example of the type of filtering that contextual thresholding can be found in Figure 3-12. The output from AutoGAD contains 15 regions, nine of which are true targets. After contextual thresholding is applied, three false positive regions are removed from the declaration, two for mean intensity and one for aspect ratio.

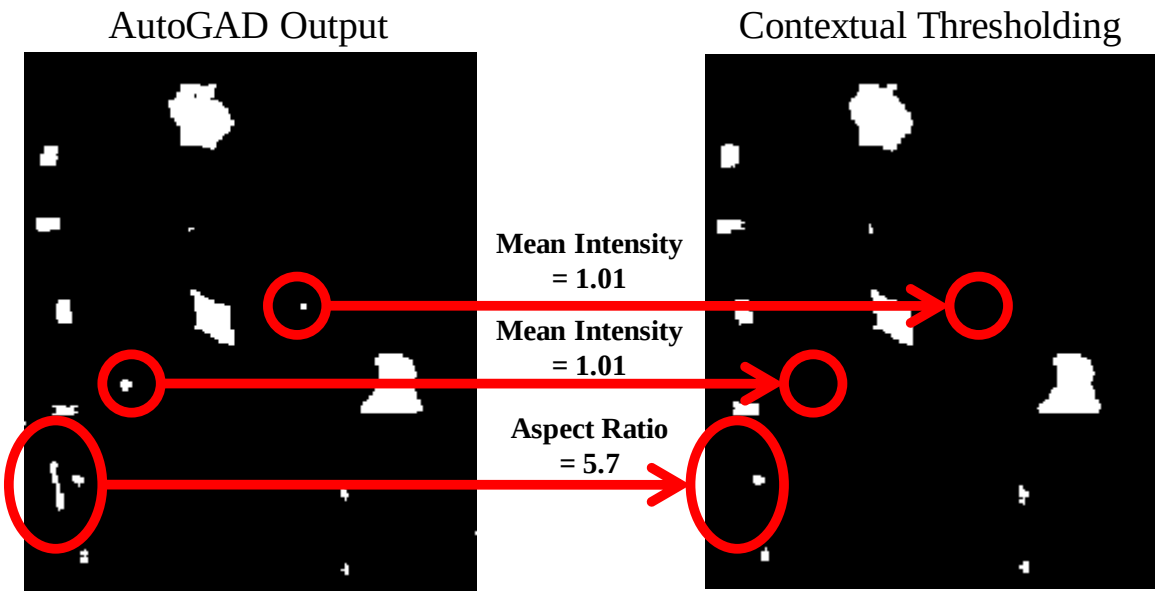


Figure 3-12: Contextual Thresholding Example

3.5.1. Threshold Determination

The proposed algorithm adds five new settings to the settings that AutoGAD already uses (Johnson, 2008). These settings interact to create a new set of identified target pixels. One such example is when a lower segmentation threshold is used the average bulbosity of the regions increases. Simple screening factorial designs for the settings indicates the prescribed settings for AutoGAD from Johnson and the following thresholds for the context thresholding provides superior results:

Table 3-3: Threshold Settings

Segmentation Threshold	0.7
Mean Intensity Threshold	1.1
Aspect Ratio Threshold	3
Minimum Area Threshold	3
Bulbosity Index Threshold	3.5

3.6. Bayesian Belief Network (BBN) Formulation

The approach of using filters on the contextual information of a potential target provides for the exclusion of regions that are clearly outside the target space. This approach rejects all regions that do not fit within all the contextual markers. The risk of rejecting a highly anomalous target just because it is too large or has a high aspect ratio is present. The application of a Bayesian belief network (BBN) to the problem may alleviate the issues with the threshold approach.

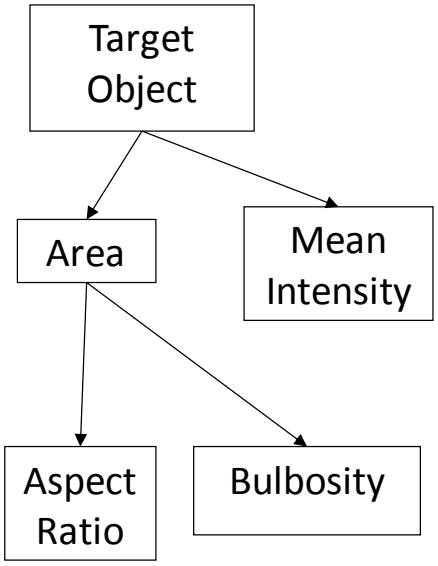


Figure 3-13: Bayesian Belief Network for Context

The Bayesian Network structure in Figure 3-13 shows the structure assumed for using context to update the posterior target probability. By using the images in the training set, the learned conditional distributions of the nodes can be derived. The 'Target Object' node is modeled as a two-level discrete node (Target and Non-target). The contextual nodes are then modeled with four-levels. This allows for adequate binning of the test images regions. Area is a parent node of aspect ratio and bulbosity due to the constant spatial pixel resolution of the sensor. Larger objects are represented at a better relative resolution to small objects. For example, a small object of a single pixel can only have an aspect ratio and bulbosity of one.

With the structure of the Bayes Net formed we can now form the calculation required for the posterior probability of a region being a target:

$$p(\text{Target} \mid \text{Context}) = \frac{p(A, As, B, MI \mid T) * p(T)}{\sum_{i=NT,T} p(A, AS, B, MI \mid T) * p(i)} = \frac{p(As, B \mid A, T) * P(A \mid T) * p(MI \mid T) * p(T)}{\sum_{i=NT,T} p(As, B \mid A, T) * P(A \mid T) * p(MI \mid T) * p(i)} \quad \textbf{(3-3)}$$

Where A=Area; As=Aspect Ratio; B=Bulbosity; MI=Mean Intensity; T=Target, and NT=Non-target.

3.6.1. Selection of BBN Discrete Node Thresholds

The regions that are selected from the training images must be assigned to four categories for the use of discrete conditional distributions. The levels of these thresholds need to be set to provide a discernable difference between the target and non-target regions. We propose the use of quantiles to set the thresholds for the various contextual nodes:

Table 3-4: BBN Discrete Level Thresholds

Level	Quantile
1	10%
2	50%
3	90%
4	100%

This would allow for a simple rule to establish the training data required to estimate the conditional probabilities in BBN. The performance of these thresholds could be investigated with later research.

3.7. Test and Training Image Results

The collection of ARES images available for experimentation is randomly broken into training, test and validation sets. This allows for a set of images that are expressly used to train the BBN conditional probabilities and subsequent images to test the settings

independent of a set of images later used to validate the techniques. Since the contextual thresholding approach does not require the training of a classifier, the training set includes the test images also. Therefore the performance of the algorithm on those sets can be examined, whereas the performance of the BBN on the training images should be higher than can be expected on non-training images.

Training Images	Test Images	Validation Images
ARES1F	ARES2D	ARES1C
ARES2F	ARES3F	ARES1D
ARES3D	ARES4F	ARES2C
ARES5		ARES4
		ARES5F
		ARES6D_10kFT
		ARES7F_10kFT

3.7.1. BBN Conditional Probabilities

Segmentation of the AutoGAD output of the training images results in approximately 300 regions at the previously mentioned AutoGAD settings. These regions represent non-target and target regions about equally. This implies a prior probability of target for AutoGAD of about 0.50. A non-target region is defined as a region containing less than 25% target pixels. This is used to control for the presence of a large region containing a single target pixel being identified as a target region. Once the training regions are collected, the quantiles are used to establish the thresholds for the four discrete levels of the conditional distributions. The thresholds are upper limits for assignment to that level. Any value over the third threshold is assigned to the fourth category. The thresholds determined from the training images are found in Figure 3-14.

Threshold	A/S Ratio	Bulbosity	Area	Mean Intensity
1	1.00	1.33	1.00	0.84
2	1.18	1.33	4.00	0.93
3	1.76	1.38	13.00	1.81

Figure 3-14: Contextual Node Quintile Thresholds from Training Images

The conditional distributions of the nodes are then learned using the MLE estimates within the Bayes Net toolbox (Murphy, 2007). The values for the conditional distributions from the training images are found in Figure 3-15 to Figure 3-18.

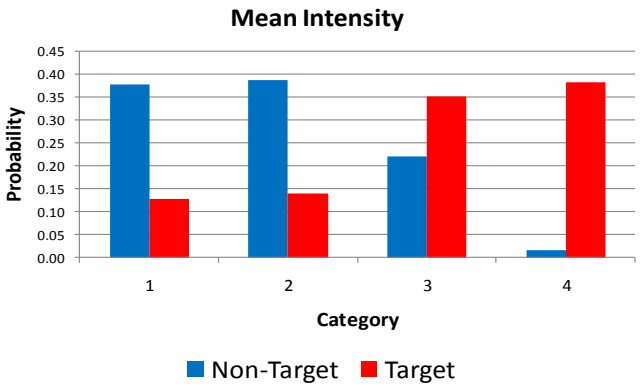


Figure 3-15: Mean Intensity Conditional Distributions from Training Images

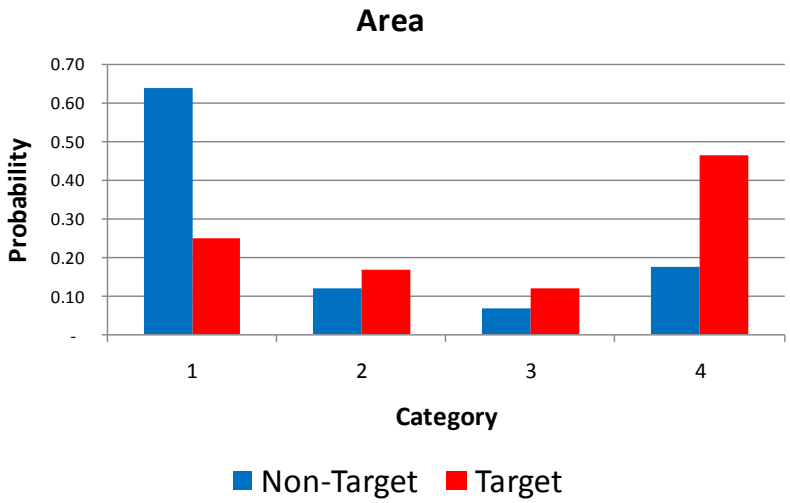


Figure 3-16: Area Conditional Distributions from Training Images

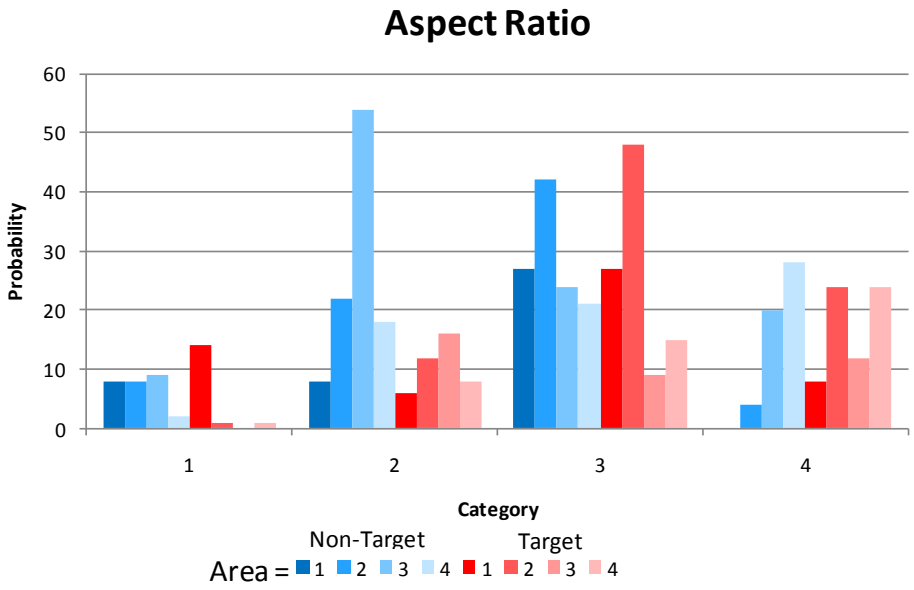


Figure 3-17: Aspect Ratio Conditional Distributions from Training Images

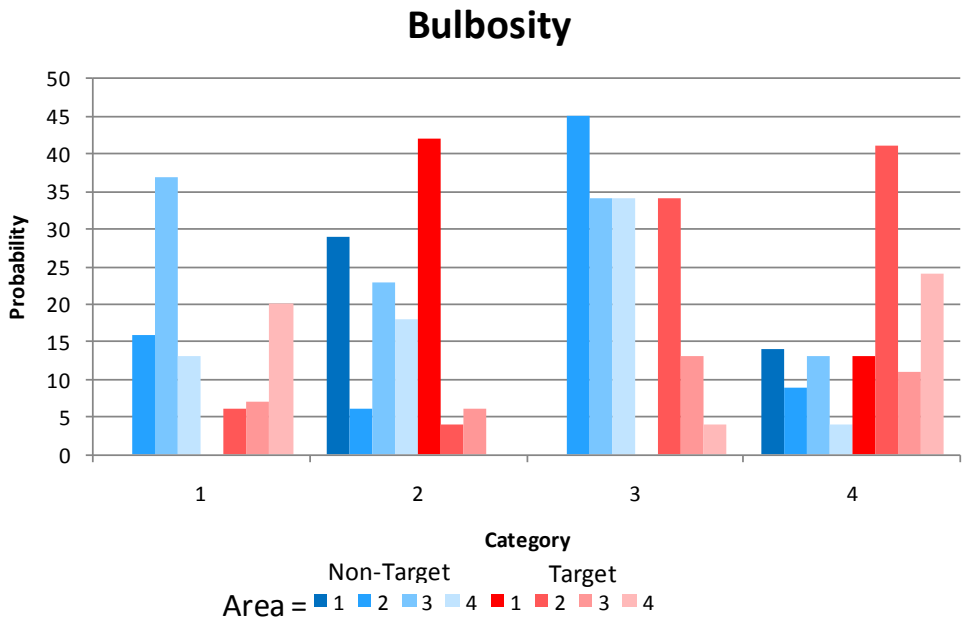


Figure 3-18: Bulbosity Conditional Distributions from Training Images

3.7.2. Performance on Training and Testing Images

The performance of the algorithms is assessed at two different levels: pixel and region. By moving to declaring targets in the region-space the pixel level metrics are insufficient. No longer are we simply examining each pixel to determine anomalousness.

Instead, an entire region is identified as an anomaly or not. It is more vital to look at the regions anomalies to examine the performance of a spatial context sensitive algorithm. We include pixel metrics for the backwards comparison of the algorithms.

The True Positive Fraction is the proportion of true target pixels identified as targets. This is calculated with the fraction true positive over the sum of true positive and false negative, equation 3-4. We desire this metric to approach one.

$$TPF = \frac{TP}{TP + FN} \tag{3-4}$$

The False Positive Fraction is the proportion of falsely identified pixels. This is calculated using the fraction of false positive over the sum of false positives and true negatives, equation 3-5. We desire this metric to approach zero.

$$FPF = \frac{FP}{FP + TN} \tag{3-5}$$

The performance of the spatial context sensitive algorithms seems increase the TPFs while controlling, or lowering, the FPF levels. The evidence of this can be seen in the results section of appendix A. The output from AutoGAD has a stochastic element caused by the fastICA algorithm that is perpetuated through our algorithms. To control for this the images were run through the algorithms a total of 75 times to allow for the determination of average performance. Variance analysis is done in section 4-5. In the charts showing performance, the best performing algorithm is highlighted for each image.

	Average Performance over 75 replications	TPF			FPF		
		AUTOGAD	Cont Thres	BBN	AUTOGAD	Cont Thres	BBN
Training	ARES1F	0.934	0.955	0.955	0.003	0.001	0.003
	ARES2F	0.928	0.947	0.948	0.001	0.001	0.002
	ARES3D	0.940	0.954	0.954	0.014	0.011	0.017
	ARES5	0.859	0.880	0.913	0.000	0.000	0.000
Testing	ARES2D	0.892	0.899	0.899	0.000	0.000	0.000
	ARES3F	0.804	0.808	0.804	0.002	0.002	0.002
	ARES4F	0.723	0.726	0.694	0.005	0.009	0.006

Figure 3-19: Pixel Performance on Training and Testing Images

Since the spatial context sensitive algorithms are segmenting the regions within the images, the same true and false identification structure needs to be applied using the regions instead of pixels. To do this we propose two metrics: percentage of regions identified that are actually targets and the percent of target regions not identified. Both of these metrics begin with the segmentation of the truth information of the image. A region is determined to contain a target if greater than 25% of the pixels within the region are true targets. This prevents the marking of a large region as true when only a small percentage of the pixels are true targets. The percent regions true is the ratio of the number of regions that are true to the total number of regions identified, equation 3-6. We desire this metric to approach one.

$$\% \text{ Regions True} = \frac{\text{True Regions}}{\text{Num Regions}} \tag{3-6}$$

To determine the percent of targets missed we simply divide the number of target regions not detected by the total number of target regions within an image, equation 3-7. We desire this metric to approach zero.

$$\% \text{ Targets Missed} = \frac{\text{Num Regions Missed}}{\text{Num Target Regions}} \tag{3-7}$$

	Average Performance over 75 replications	% of Regions True			% of Targets Missed		
		AUTOGAD	Cont Thres	BBN	AUTOGAD	Cont Thres	BBN
Training	ARES1F	53%	75%	66%	0%	0%	0%
	ARES2F	76%	93%	85%	3%	3%	3%
	ARES3D	16%	23%	24%	0%	0%	0%
	ARES5	84%	93%	93%	0%	13%	4%
Testing	ARES2D	100%	100%	100%	11%	13%	15%
	ARES3F	61%	67%	70%	22%	23%	23%
	ARES4F	80%	77%	79%	28%	28%	31%

Figure 3-20: Region Performance on Training and Testing Images

4. Results and Analysis

The results from the two spatial context sensitive methods, contextual thresholding and Bayesian Belief Network (BBN), are compared with the performance of AutoGAD (Johnson, 2008). The validation set consists of the following seven ARES images:

Table 4-1: Validation Images.

ARES1C
ARES1D
ARES2C
ARES4
ARES5F
ARES6D_10kFT
ARES7F_10kFT

Two of the images used in validation are at a higher altitude than the majority of the images. This will allow for the examination of the techniques at extended operating conditions.

This chapter consists of four sections: Validation Images Results, Insights, BBN Node Influence and Variance Analysis.

4.1. Validation Images Results

Both of the techniques experience an appreciable decrease to the number of target regions being reported. This often comes without a decrease in the number of targets that are identified correctly. This is significant since each region reported must be examined further to identify the object. ARES1C is significant since it does not have any targets and AutoGAD does not select any components for targets. Therefore, the post-process is

unable to continue and the results match those of AutoGAD. ARES1C and ARES2C both contain no target pixels.

Average Performance over 75 replications		TPF			FPF		
		AUTOGAD	Cont Thres	BBN	AUTOGAD	Cont Thres	BBN
Validation	ARES1C	No Target Pixels			0.000	0.000	0.000
	ARES1D	0.862	0.890	0.899	0.009	0.004	0.009
	ARES2C	No Target Pixels			0.010	0.010	0.010
	ARES4	0.693	0.718	0.726	0.007	0.009	0.008
	ARES5F	0.603	0.574	0.644	0.000	0.000	0.000
	ARES6D_10kFT	0.668	0.745	0.737	0.012	0.011	0.014
	ARES7F_10kFT	0.862	0.885	0.884	0.013	0.015	0.016

Figure 4-1: Pixel Performance on Validation Images

The performance of the contextual thresholding is promising. By lowering the threshold on the AutoGAD identified components, new and larger target regions can be identified. However, on some images, the thresholds eliminate true positive targets. This is due to the threshold of a single contextual marker eliminating the region.

Average Performance over 75 replications		% of Regions True			% of Targets Missed		
		AUTOGAD	Cont Thres	BBN	AUTOGAD	Cont Thres	BBN
Validation	ARES1C	No Target Regions			No Target Regions		
	ARES1D	5.2%	20.4%	16.7%	0%	0%	0%
	ARES2C	No Target Regions			No Target Regions		
	ARES4	48.4%	50.0%	57.6%	19%	20%	18%
	ARES5F	88.1%	92.0%	91.6%	14%	15%	15%
	ARES6D_10kFT	36.7%	43.1%	47.9%	22%	22%	23%
	ARES7F_10kFT	39.8%	49.1%	43.8%	0%	0%	0%

Figure 4-2: Region Performance on Validation Images

4.2. Insights

To display the potential of the spatial context sensitive approach we examine the results from ARES1D more closely. The desert scene contains a large amount of 'noise' in the form of off-road tire tracks and shrubbery, Figure 4-3.

Figure 4-3: ARES1D True Color and Truth Images

AutoGAD identifies many of these points as false positives, Figure 4-4. This is due to the pixel level identification nature of AutoGAD. Segmentation of these target regions allows the contextual thresholding approach to eliminate some of the noise. The striking improvement that the BBN approach provides can be attributed to the BBN weighing all the spatial context information simultaneously instead of examining each measure separately as contextual thresholding does. This allows the BBN to eliminate more false positive regions than the other two methods. BBN effectively eliminates 80% of the false positive regions reported by AutoGAD. The presence of the artifact line traveling vertically through the image may be controlled with a mixture of contextual thresholding and BBN techniques. Another aspect of the algorithms is the ability of contextual thresholding to eliminate the sensor artifact line down the center of the image that the other two algorithms failed to eliminate. This seems to indicate that a mixture of the two context sensitive algorithms might yield the highest performance.

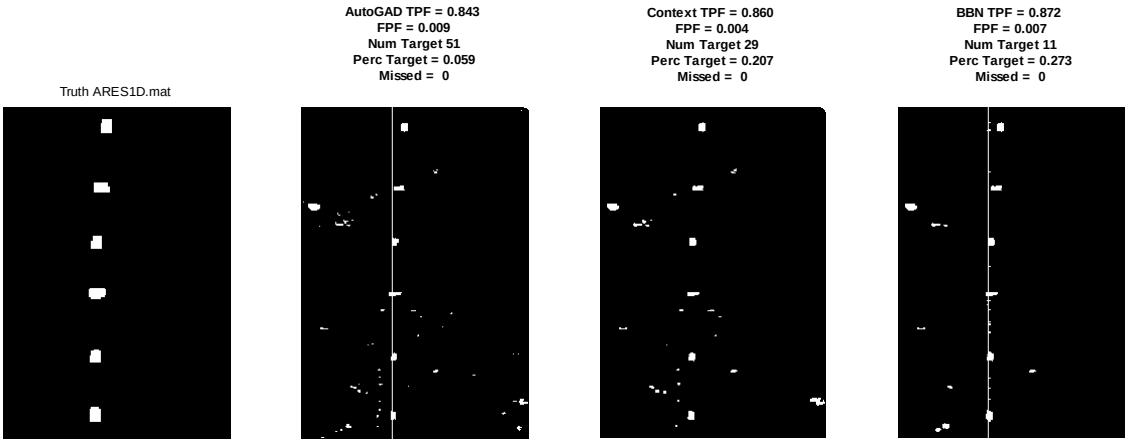


Figure 4-4: ARES1D Performance

4.3. BBN Node Influence

The structure of the BBN allows for the examination of the influence of the different contextual nodes. For example, investigating the impact that the inclusion of the area of a region has on the algorithm's ability to classify regions. To facilitate this, the algorithm was run withholding the information from the BBN for the node in question. As can be seen in Table 4-2, the impact of the mean intensity node is the dominating node. Table 4-2 displays the average performance on the two sets of data without the information for a node. This is in line with expectation given the amount of information that is contained in the mean intensity. Mean intensity supplies the most information to the algorithm for the identification of anomalous regions. The spatial context nodes seem to each impact the algorithm in roughly equal amounts. This seems to imply that each contextual node contributes to the identification of regions where the other nodes are indecisive. The spatial context nodes seem to function more as false positive mitigation than true positive selectors.

Table 4-2: Algorithm Performance without Each Node

	Training/Testing				Validating			
	TPF	FPF	% of Regions True	% of Targets Missed	TPF	FPF	% of Regions True	% of Targets Missed
Full BBN	0.875	0.003	78.8%	11.8%	0.864	0.003	79.5%	9.9%
-Aspect Ratio	0.874	0.004	77.6%	12.1%	0.863	0.004	78.1%	12.2%
-Bulbosity	0.867	0.002	82.7%	14.5%	0.855	0.003	83.2%	18.6%
-Mean Intensity	0.347	0.002	72.0%	59.9%	0.391	0.002	75.6%	66.0%
-Area	0.855	0.003	81.5%	15.7%	0.842	0.003	81.8%	17.2%

4.4. Variance Analysis

Since there is a stochastic nature to the output that our algorithms receive from AutoGAD we need to analyze the variance of the algorithm performance on the images. The performance of AutoGAD on the training varies due to the randomness inherent in the fastICA technique and therefore BBN conditional probabilities are also random. This randomness is perpetuated into the conditional probabilities and into the posterior probability calculations. To investigate the impact of this randomness the algorithm was re-ran on the image sets 75 times. The results are in Table 4-3 below. There seems to be increased amount of variance in the BBN values. This may be a result of the small amount of training data used in determining the conditional distributions. Using a larger training set may decrease this variance as the conditional probability distributions become more stable.

Table 4-3: Variance of Responses

	Variance Performance over 75 replications	AUTOGAD					Contextual Thresholding					Bayesian Belief Net				
		TPF	FPF	Number of Regions	Missed Targets	Percent Target	TPF	FPF	Number of Regions	Missed Targets	Percent Target	TPF	FPF	Number of Regions	Missed Targets	Percent Target
Training	ARES1F	3.08E-31	1.20E-35	0	0	7.89E-31	4.93E-32	1.88E-37	0	0	0	2.56E-06	1.92E-07	2.45	0.03	5.50E-03
	ARES2F	4.61E-05	2.05E-07	8.60	0	0.00	1.80E-05	1.41E-07	1.79	0	1.24E-03	2.64E-05	1.08E-06	10.04	0.12	5.42E-03
	ARES3D	1.05E-06	9.02E-07	6.92	0	2.95E-04	1.05E-06	4.95E-07	1.32	0	2.34E-04	1.05E-06	5.21E-06	16.62	0.00E+00	3.12E-03
	ARES5	5.61E-06	4.72E-10	0.05	0	3.88E-06	2.15E-07	6.61E-39	0.00	0	4.44E-31	4.77E-04	2.25E-08	0.64	4.36E-01	3.61E-04
Testing	ARES2D	3.30E-05	0.00	0	0	0.00	1.49E-04	3.67E-08	0.23	0.23	0	1.19E-05	1.75E-07	0.25	0.22	4.60E-05
	ARES3F	2.11E-04	7.72E-07	14.89	0.41	0.01	2.95E-04	7.57E-07	18.60	0.25	0.02	5.11E-04	1.03E-06	11.49	0.34	0.01
	ARES4F	8.24E-05	1.09E-07	0.45	0	2.91E-05	2.36E-04	2.13E-06	0.18	0.18	1.71E-05	1.28E-03	6.65E-06	1.95	0.95	5.79E-04
Validation	ARES1C	NaN	0	0	0	NaN	NaN	0.00E+00	0	0	NaN	NaN	0	0	0	NaN
	ARES1D	6.12E-04	1.43E-08	0.44	0	8.23E-05	1.46E-03	1.07E-08	0.88	0	4.03E-05	1.16E-03	2.37E-06	70.47	0	6.21E-03
	ARES2C	NaN	3.56E-06	2.44	0	0.00	NaN	2.48E-06	0.15	0	0.00	NaN	1.63E-05	6.22	0	0.00
	ARES4	5.35E-06	1.28E-08	0.19	0.16	6.06E-05	1.37E-06	1.36E-08	0.03	0	1.16E-05	1.63E-04	7.70E-07	2.86	0.21	2.17E-03
	ARES5F	1.58E-04	1.12E-08	2.79	0.72	1.31E-03	8.92E-03	1.96E-09	0.19	2.02	9.51E-05	4.74E-04	1.32E-07	2.28	0.72	8.09E-04
	RES6D_10k	2.06E-04	1.39E-06	3.81	0.16	3.60E-04	8.14E-05	1.91E-06	1.80	0.16	1.33E-03	9.30E-05	1.13E-05	13.66	0.01	7.67E-03
	RES7F_10k	1.53E-05	3.94E-08	3.13	0	5.07E-04	3.15E-05	2.67E-07	1.13	0	4.51E-04	3.37E-05	3.00E-06	1.25	0	3.71E-04

5. Discussion

5.1. Research Contribution

1. Showed the benefit of using spatial context to find anomalies in HSI
2. Created region level performance metrics to augment pixel level performance metrics to determine the performance of HSI anomaly detection algorithms
3. Created a context sensitive post-processor for AutoGAD for the identification of anomalies in HSI.

5.2. Limitations

The development of the algorithm as a post processor for AutoGAD is only due to the nature of AutoGAD output. Each pixel in AutoGAD is assigned a ‘score’ for each component within the image. The segmentation of each component allows for the differentiation of different objects within the image.

Further, AutoGAD was tuned for the entire set of images. Therefore, appreciable improvement to the performance of the algorithm is significant.

5.3. Conclusion

This study has shown that the reintroduction of spatial context information into an anomaly detection algorithm can provide increased performance. The benefit of the information can come at very little computational cost with the use of simple segmentation algorithms while supplying great increases to the performance of the algorithms. We feel that this approach can be adapted to future anomaly detectors to

control the number of regions requiring inspection and increasing the number of true regions found.

5.4. Further Research

1. Further refinement of the settings to find the optimal settings
2. Using the nature of the segmented objects to aid in determining the target characteristics of a component in AutoGAD
3. Use a fusion structure to fuse other detectors with spatial context.

Appendix A. Image Characteristics and Results

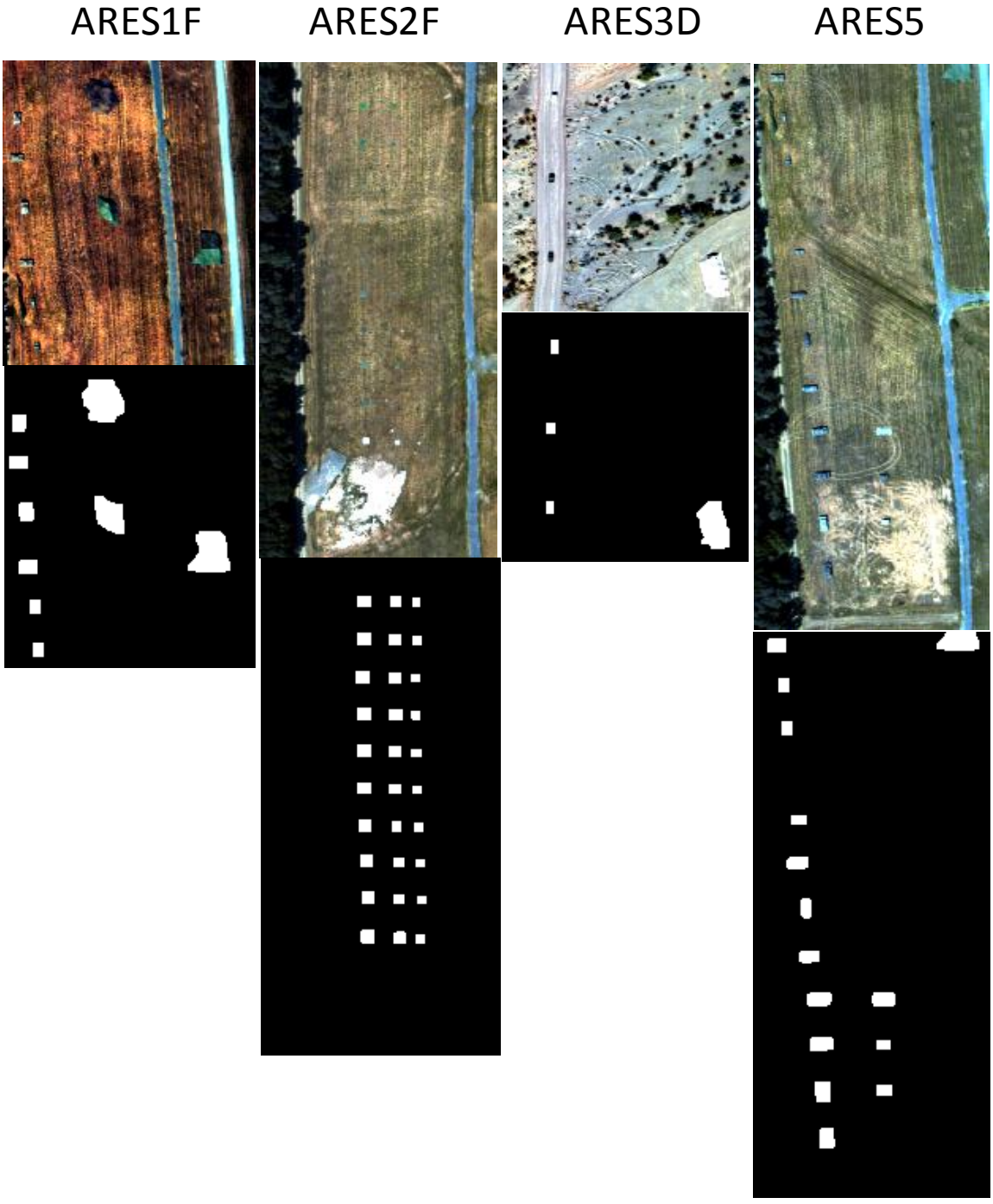


Figure A-1: Training Set True Color Images and Truth Maps

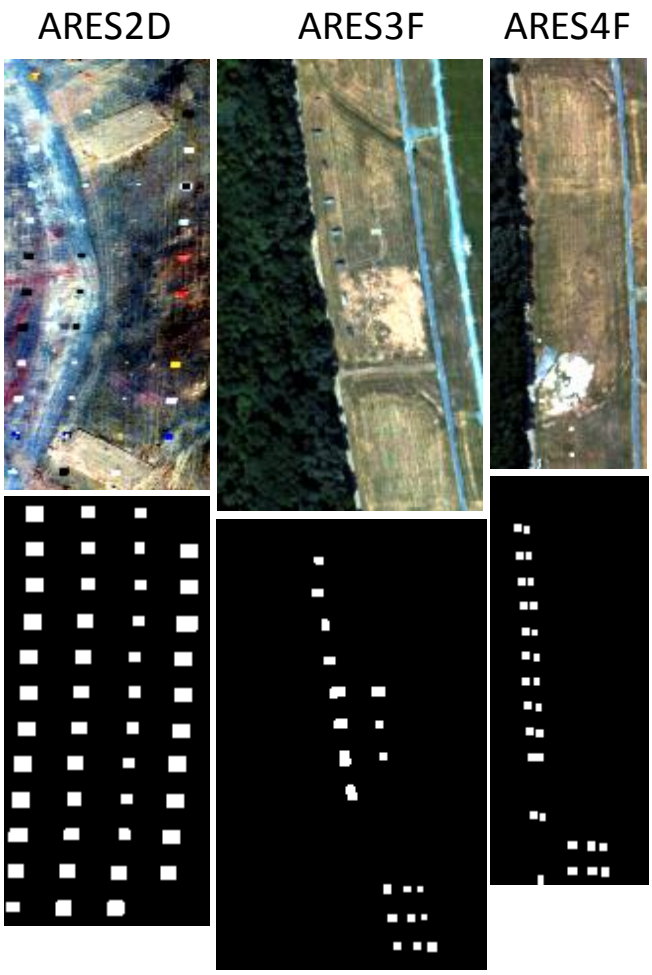


Figure A-2: Testing Set True Color Images and Truth Maps

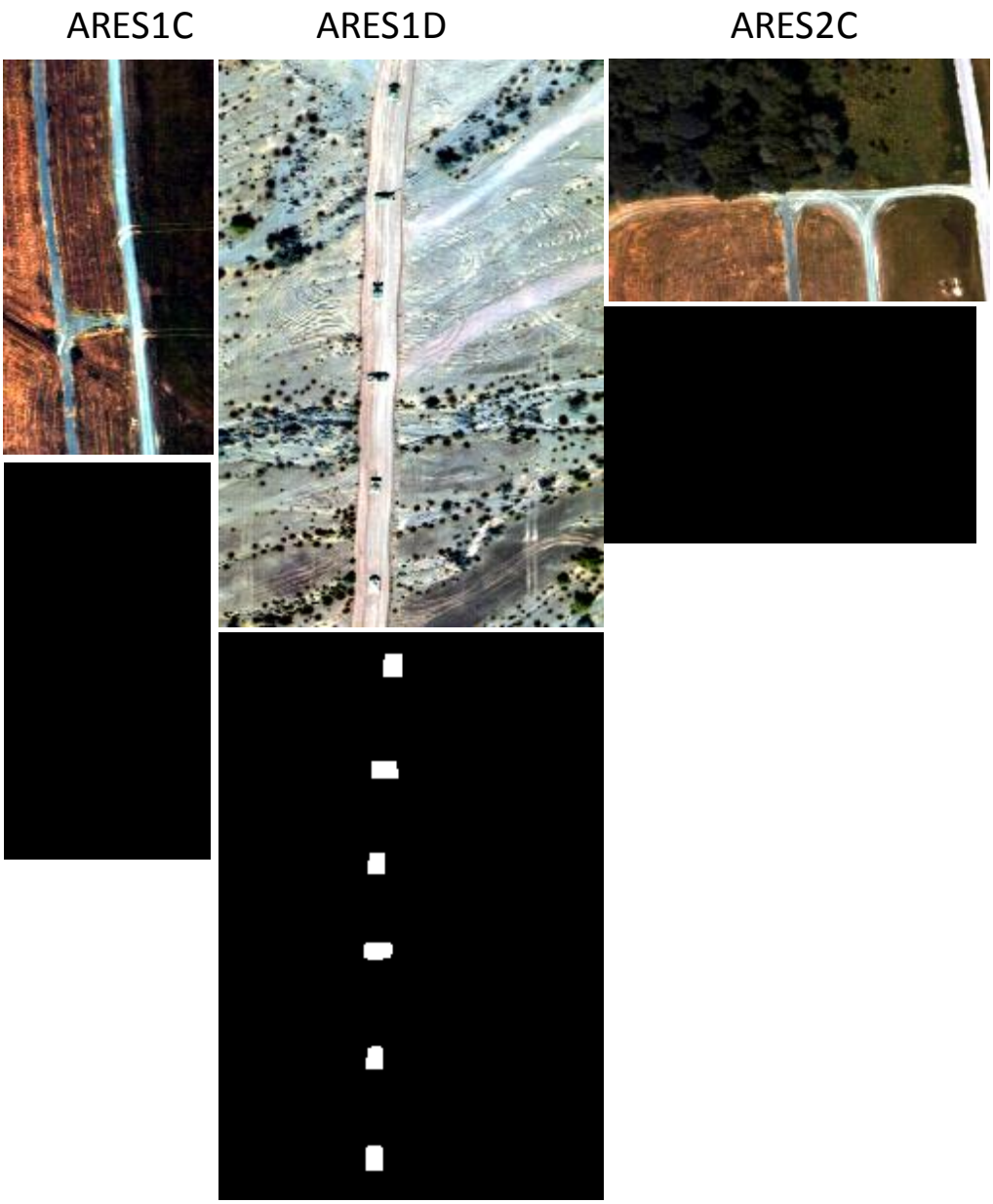


Figure A-3: Validation Set True Color Images and Truth Maps (1 of 2)

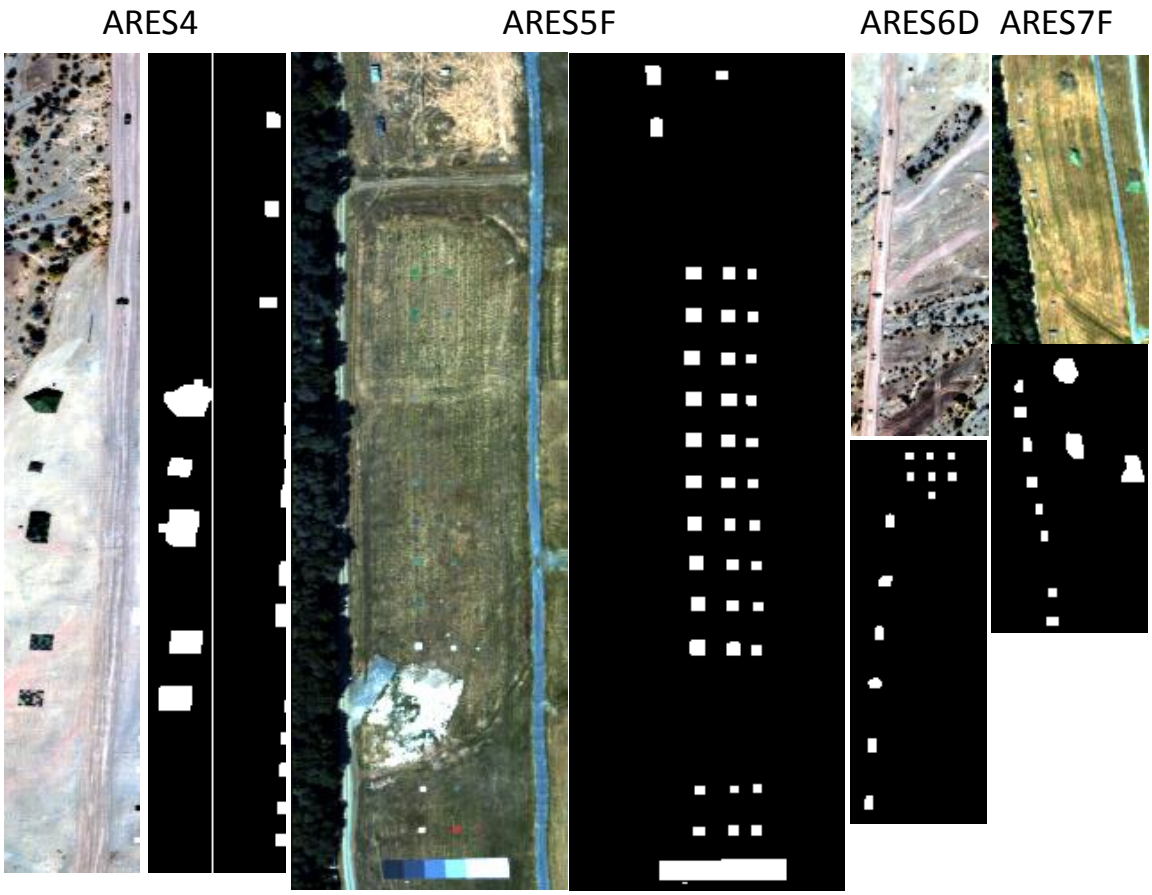


Figure A-4: Validation Set True Color Images and Truth Maps (2 of 2)

Training Images

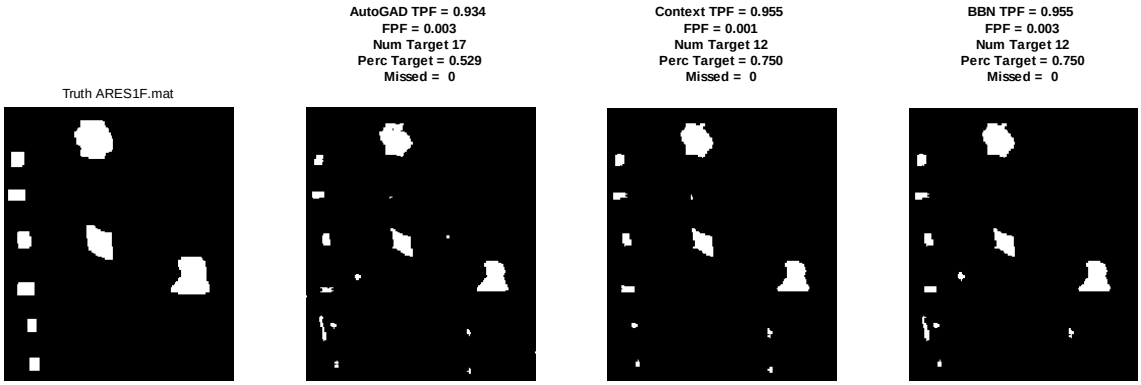


Figure A-5: ARES1F Performance

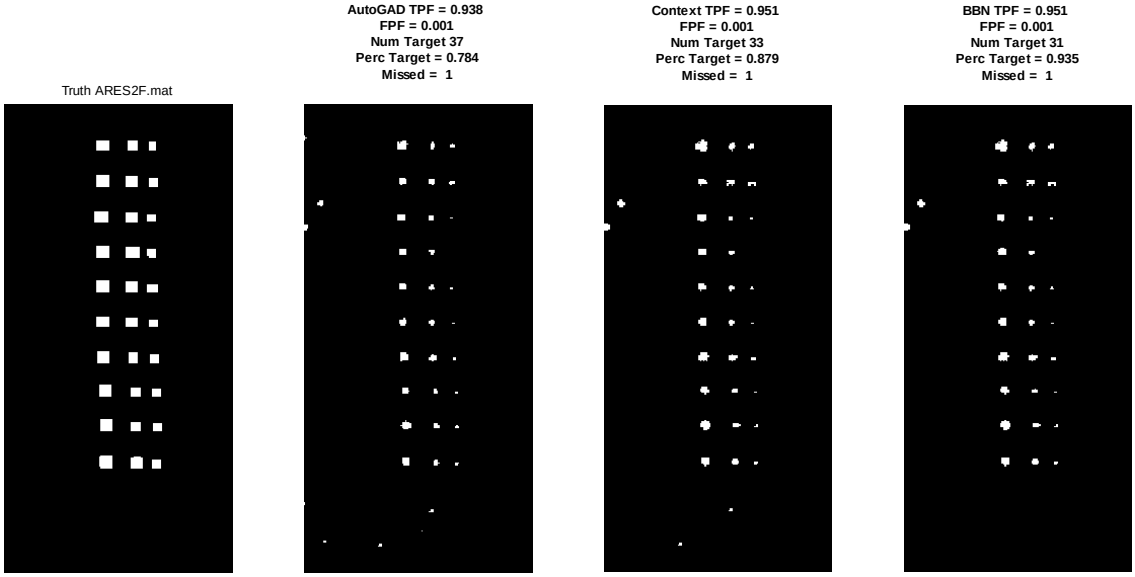


Figure A-6: ARES2F Performance

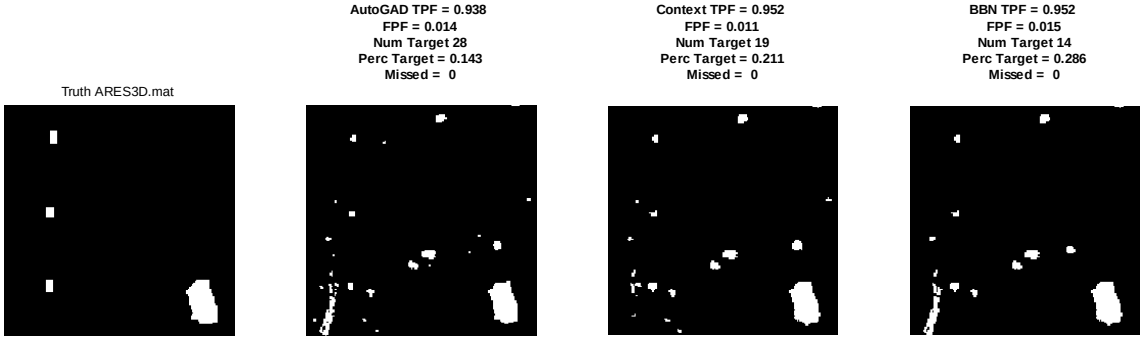


Figure A-7: ARES3D Performance

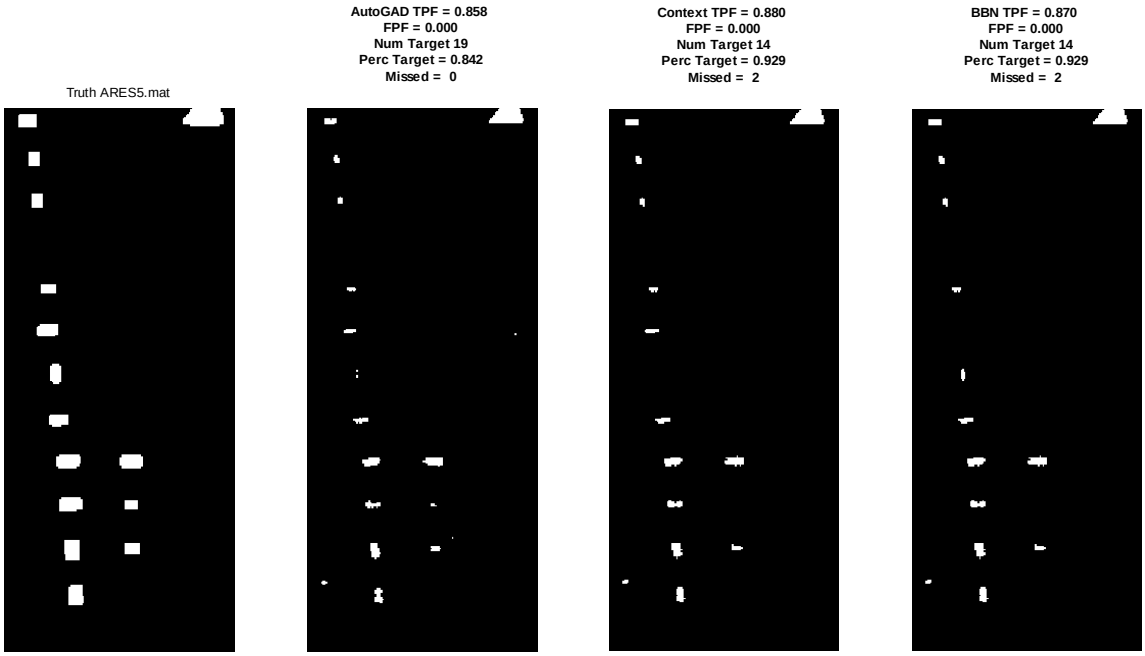


Figure A-8: ARES5 Performance

Testing Images

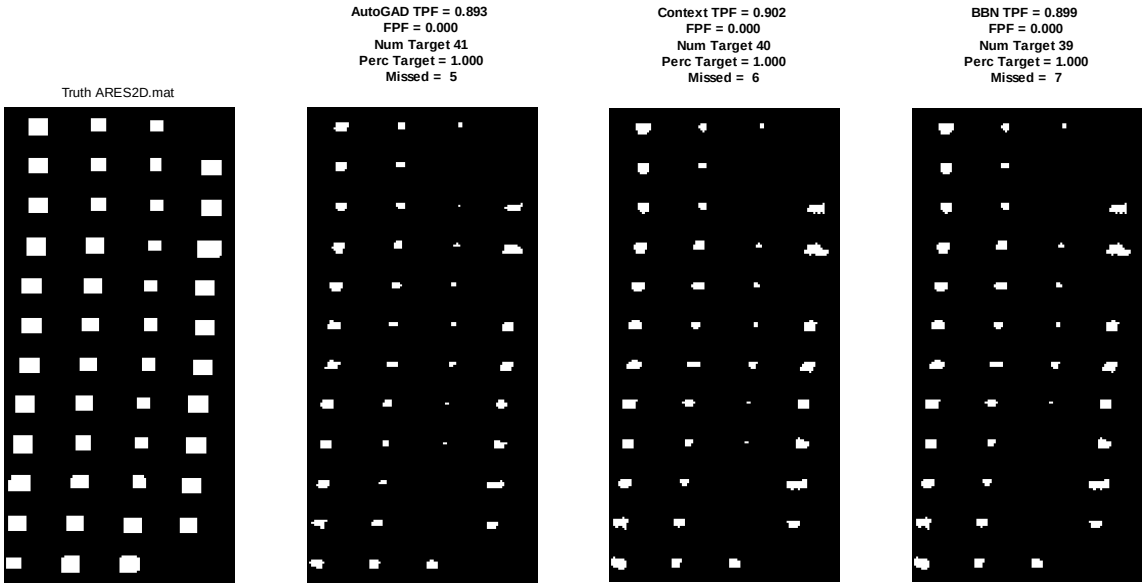


Figure A-9: ARES2D Performance

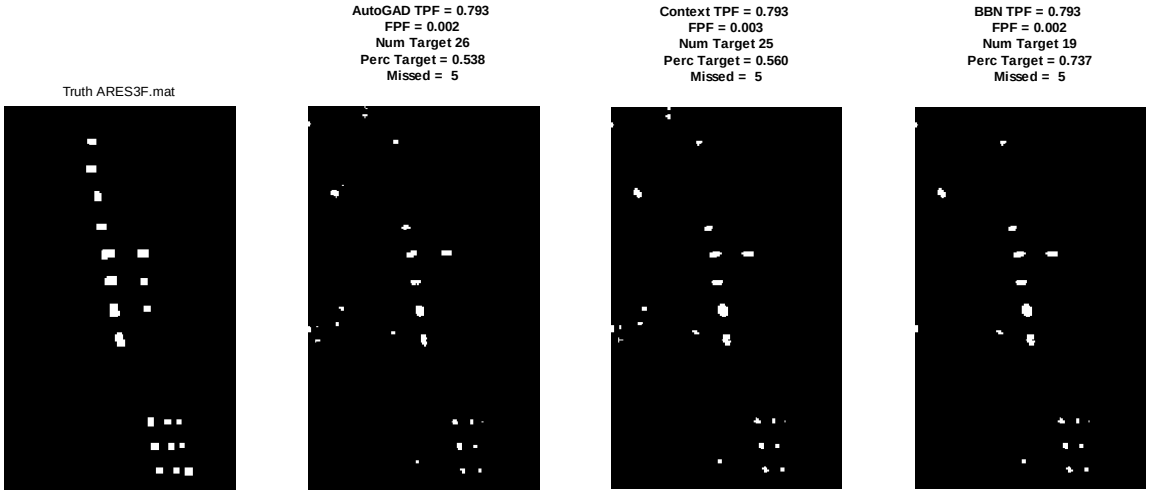


Figure A-10: ARES3F Performance

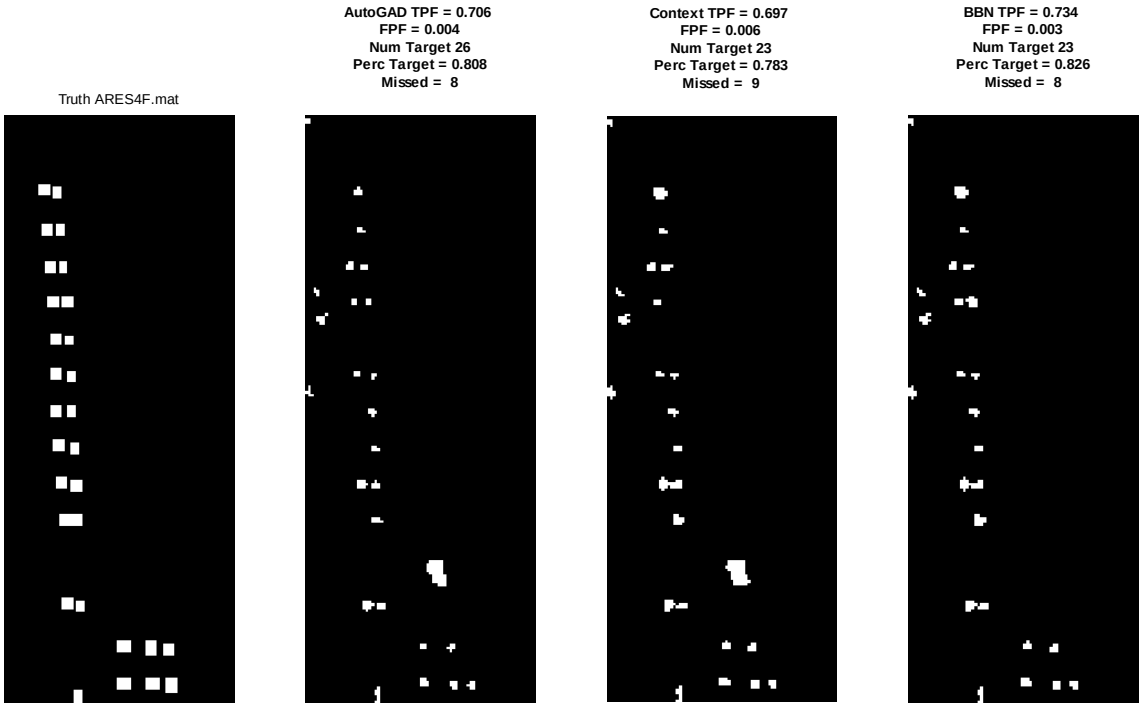


Figure A-11: ARES4F Performance

Validation Images

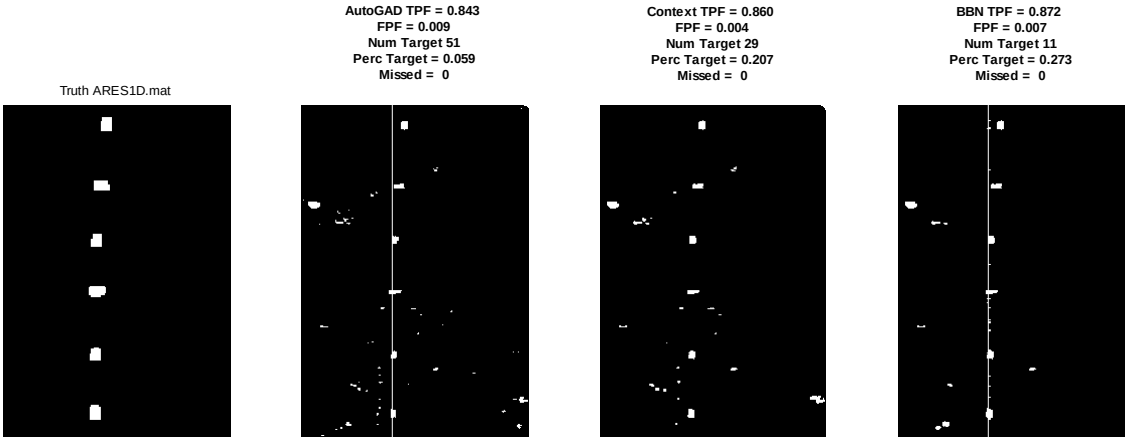


Figure A-12: ARES1D Performance

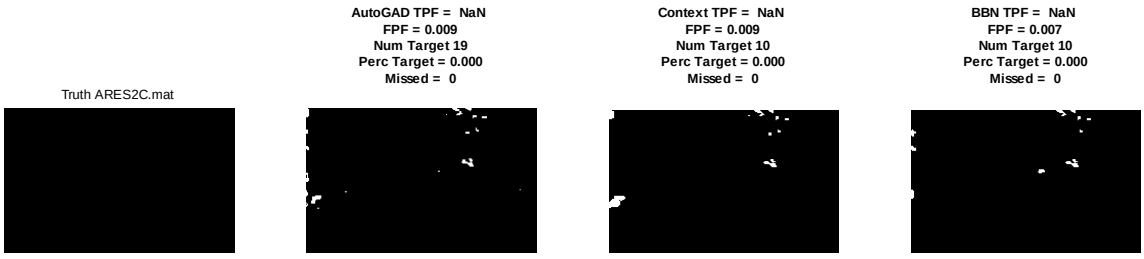


Figure A-13: ARES2C Performance

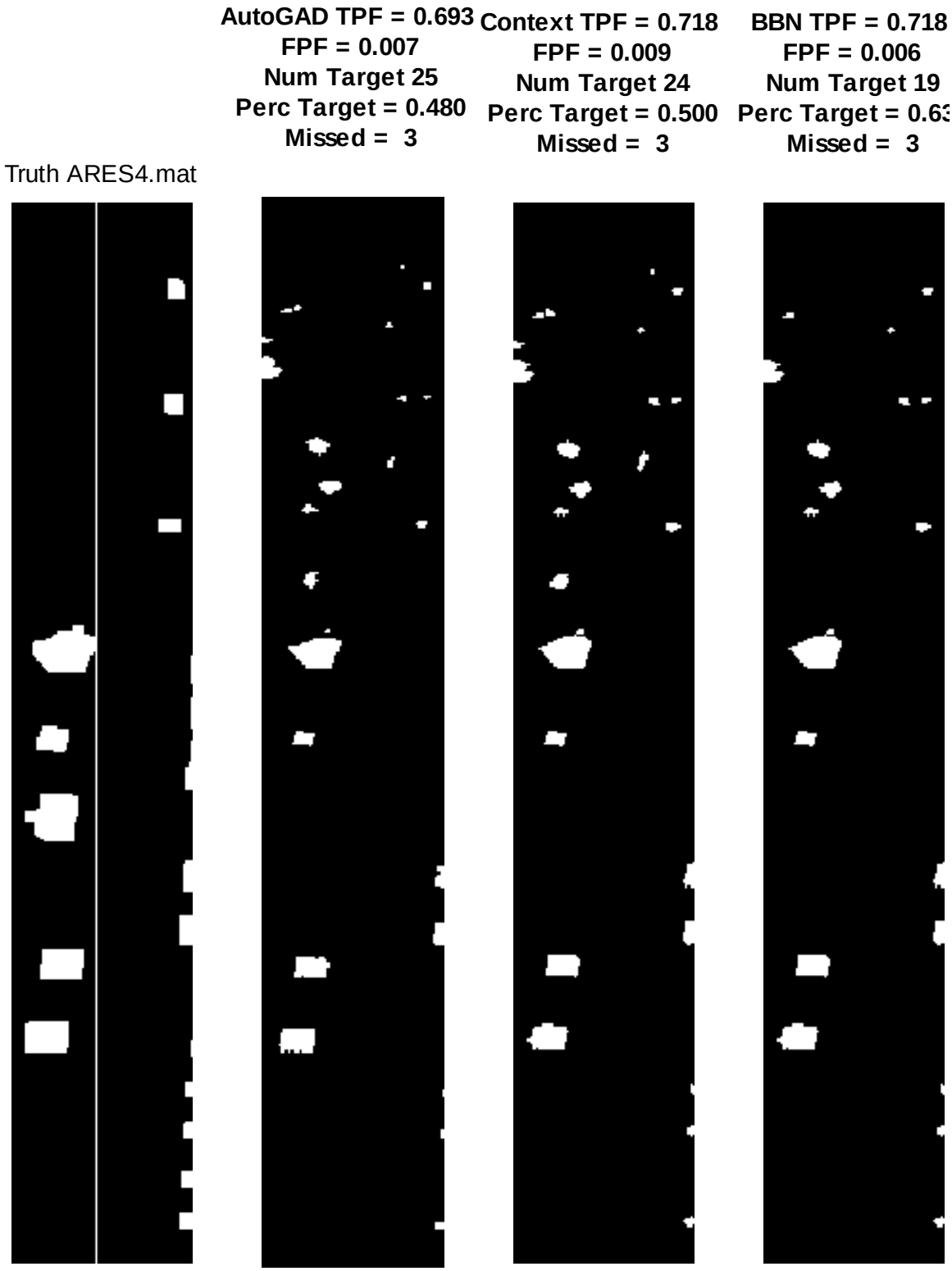


Figure A-14: ARES4 Performance

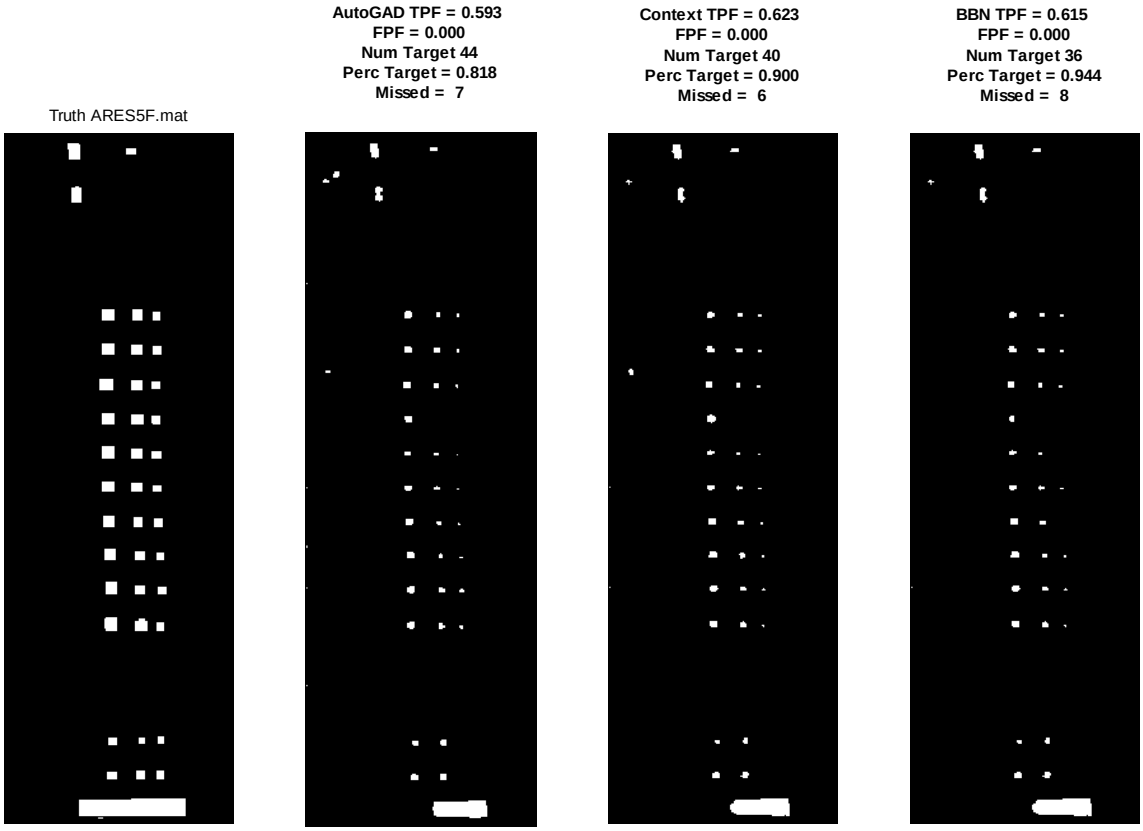


Figure A-15: ARES5F Performance

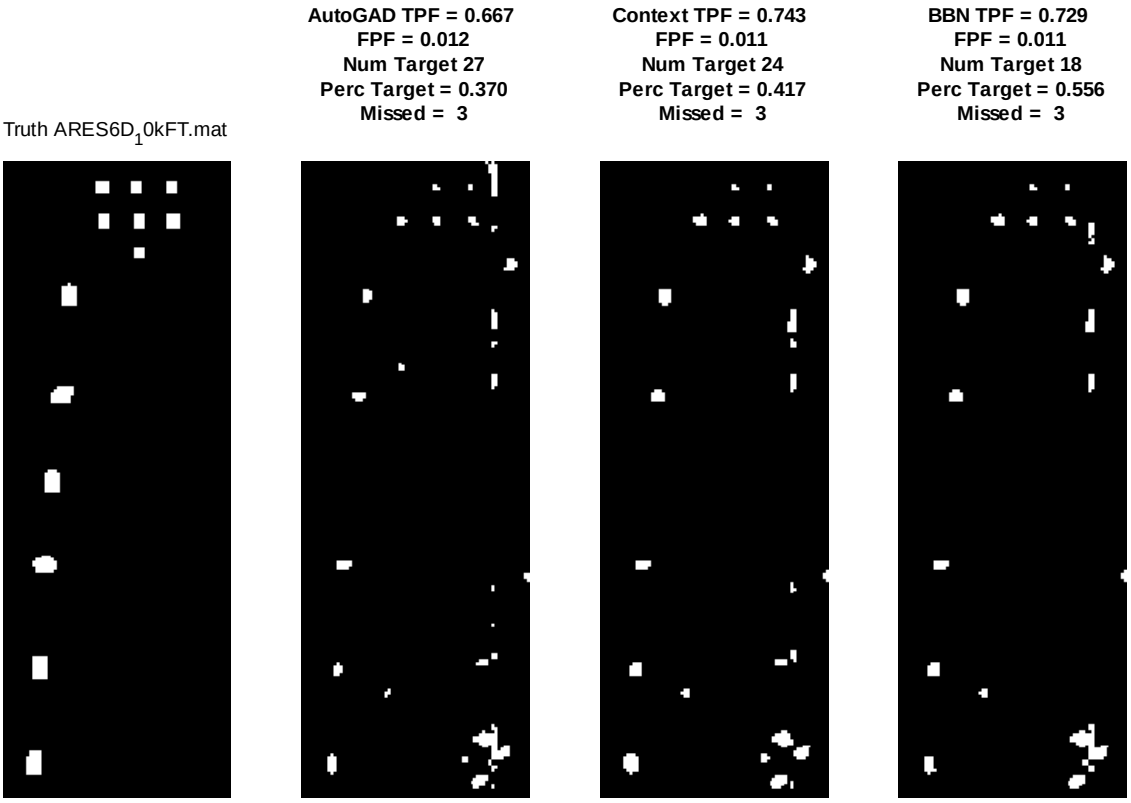


Figure A-16: ARES6D Performance

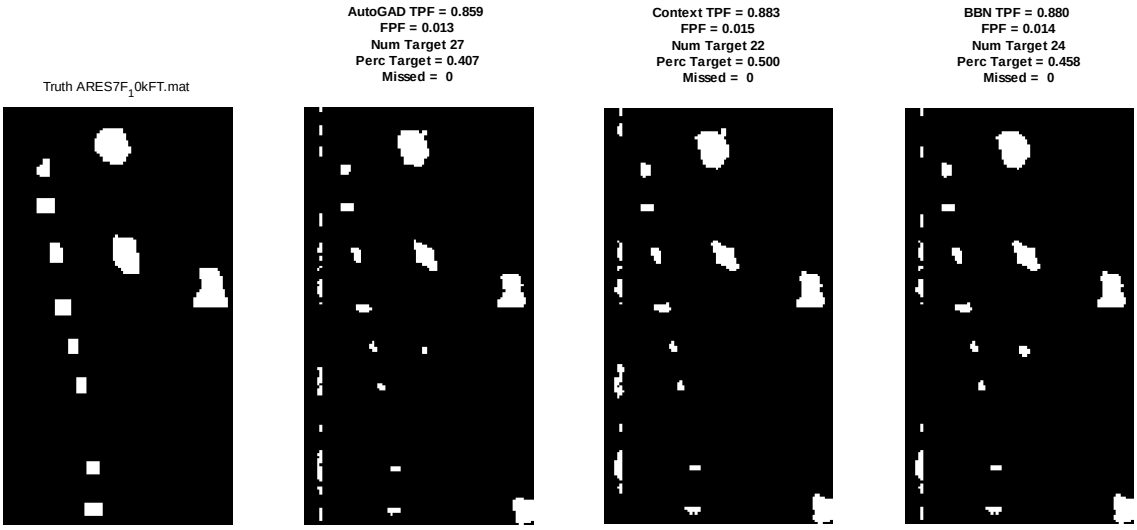


Figure A-17: ARES7F Performance

Appendix B. Blue Dart

Context in Hyperspectral Image Anomaly Detection:

Size and Shape Do Matter

The escalation of remote sensing technology in recent decades has greatly increased the demand for imagery and geospatial intelligence analysts. This demand has outstripped the ability of the Intelligence Community's current infrastructure to support the growing need for human-performed analysis. This critical issue has been identified at all levels within the Intelligence Community at large: "We're going to find ourselves in the not too distant future swimming in sensors and drowning in data" (Lt. Gen. David A. Deptula, Air Force deputy chief of staff for intelligence, surveillance and reconnaissance).

New techniques are being developed to increase amount of intelligence a single, human analyst can process. One such technique is the use of anomaly detection algorithms to preprocess images collected from remote sensors, specifically images collected by hyperspectral sensors. Hyperspectral images are much like images taken with a standard, low-end digital camera. This type of digital camera takes images by recording the amount of red, green, and blue (RGB) light reflected from the scene. Hyperspectral sensors, on the other hand, record upwards of one-hundred different wavelengths of light. These wavelengths span from ultraviolet light, through the visible spectrum, and all the way to the short-wave infrared spectrum. The information contained within these spectral bands can be used to identify materials that are indistinguishable using normal black and white or RGB images. For example, a camouflage tarp covering a

tank may appear to be the same shade of green as its surrounding area to a normal RGB sensor, while the same tarp, seen with a hyperspectral sensor will contrast sharply as a different material. This allows for detection and defeat of adversaries who attempt to hide targets or activities through camouflage and/or concealment methods.

Researchers at AFIT have already developed algorithms that can process HSI quickly, single-out and return specific, possible target pixels to an analyst for closer examination. However, these algorithms operate in a pixel-by-pixel manner, and like most state-of-the-art algorithms, they completely disregard the spatial context of the discovered anomalies by concentrating too much on the statistical aspect of the spectral bands. Herein lies the major flaw: these algorithms process an image pixel by pixel, focusing on whether or not that particular pixel sticks out as odd from the rest of the image. By doing this, the algorithms may inappropriately identify regions of the image as target pixels that, in reality, could not possibly be targets. These techniques can be improved by reintroducing spatial context information into the algorithm. Our study proposes using variables such as size and shape, in addition to the hyperspectral signature of a region of pixels to determine the anomalous regions within an image.

Much like a human analyst will take into account the entire context of an image to determine which regions to investigate closer, our algorithm takes into account the size and shape of a region and identifies those regions that stand out from the rest of the image. By using the spatial context within the algorithm, we can increase the number of regions that are reported to the analyst as truly anomalous, while controlling or decreasing the number of regions that are incorrectly reported.

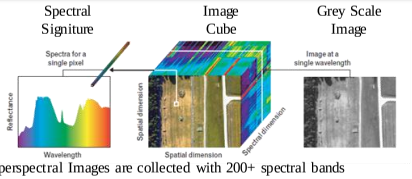
By reporting regions that have a higher likelihood of being a target, the analyst can focus their attention to truly anomalous targets. This will allow for more throughputs per analyst, which will significantly increase the consumption and decimation of critical intelligence information. Without such techniques, algorithms will routinely return possible targets to analysts for examination that are in fact, false targets, thus forcing the analyst to examine benign portions of an image and thereby slowing down the intelligence cycle. Our new techniques will help prevent the imagery and geospatial intelligence analysis processes from drowning in data.

Contextual Detection of Anomalies in Hyperspectral Images

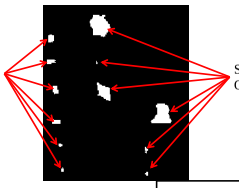
Capt Adam Messer
Department of Operational Sciences (ENS)
ADVISOR
Dr. Kenneth Bauer
READER
Maj Mark Friend

Research Objectives:

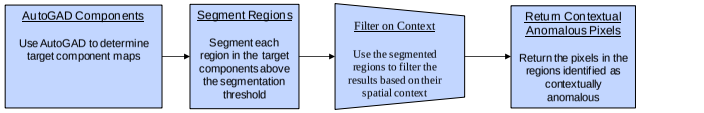
- Integrate the spatial context (size and shape) of anomalies to increase detection performance
- Specifically, create a post processor to AutoGAD output that combines the statistical and spatial information



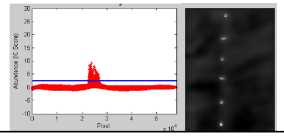
- Hyperspectral Images are collected with 200+ spectral bands
- Spectral signatures can identify targets with a much lower spatial resolution



METHODOLOGY



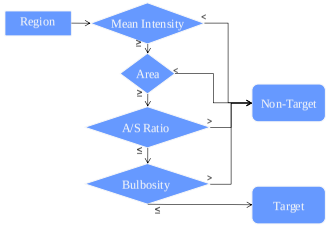
Statistical View- Each pixel identified independently as anomalous



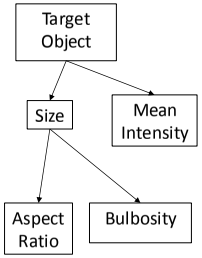
Spatial View Regions of anomalous pixels become evident

Two filtering frameworks proposed

Contextual Thresholding Decision Tree



Bayesian Belief Network



After segmentation regions are filtered for context and declared as anomalies

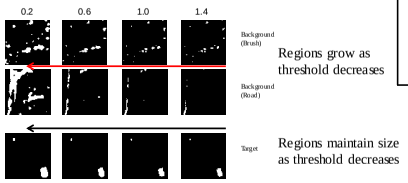
$$P(\text{Target} \mid \text{Context}) = \frac{p(T, A, AS, B, MI)}{\sum_T p(T, A, AS, B, MI)} = \frac{P(A, As, B \mid T) * P(MI \mid T) * P(T)}{\sum_T P(A, As, B \mid T) * P(MI \mid T) * P(T)}$$

Contextual Values

$$\text{Mean Intensity} = \frac{\sum_{region} \text{Pixel Scores}}{\text{Num Pixels}}$$

$$\text{Aspect Ratio} = \frac{\text{Major Axis}}{\text{Minor Axis}}$$

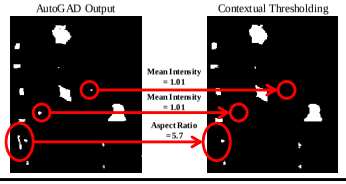
$$\text{Bulbosity} = \frac{\text{Major Axis} * \text{Minor Axis}}{\text{Area}}$$



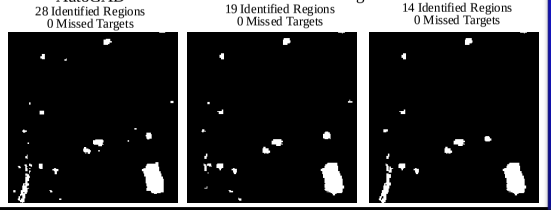
Suggests target regions retain a high mean intensity at lower thresholds!

Anomalies are declared using the region's context information versus pixels

AutoGAD components are segmented at a lower percentage of the original threshold. This allows for the inclusion of larger regions



Results



Using Context significantly decreases the number of false positive regions!
This indicates size and shape do matter!

AFIT
AIR FORCE INSTITUTE OF TECHNOLOGY

Appendix D. MATLAB Code

```
%Select Mode of Algorithm
% test=0; validation = 0; %Training;
Test=1; Validation = 0; %Testing;
% Test=1; Validation = 1; %Validation;

%Select Folder Containing Image and Truth Cubes
pn='J:\Sensor Fusion Lab\DataSets\HSI Images\';

%Training Images
im_names={'ARES1F.mat';'ARES2F.mat';'ARES3D.mat';'ARES5.mat'; };
truth_names={'ARES1F_mask.mat';'ARES2F_mask.mat';'ARES3D_mask.mat';...
    'ARES5_mask.mat'; };

%Test or Validation Images
if test==1
    im_names_val={'ARES1F.mat';'ARES2F.mat';'ARES3D.mat';...
        'ARES5.mat';'ARES2D.mat';'ARES3F.mat';'ARES4F.mat';};
    truth_names_val={'ARES1F_mask.mat';'ARES2F_mask.mat';...
        'ARES3D_mask.mat';'ARES5_mask.mat';'ARES2D_mask.mat';...
        'ARES3F_mask.mat';'ARES4F_mask.mat';};
else
    im_names_val={'ARES1C.mat';'ARES1D.mat';'ARES2C.mat';...

'ARES4.mat';'ARES5F.mat';'ARES6D_10kFT.mat';'ARES7F_10kFT.mat';};
    truth_names_val={'ARES1C_mask.mat';'ARES1D_mask.mat';...
        'ARES2C_mask.mat';'ARES4_mask.mat';'ARES5F_mask.mat';...
        'ARES6D_10kFT_mask.mat';'ARES7F_10kFT_mask.mat';};
end

%Thresholds for Contextual Thresholding
thres_map_threshold     = 0.6;
MeanIntensity_threshold = 1.1;
aspectratio_threshold   = 4;
Area_threshold_low      = 0;
bulbosity_threshold     = 3.5;
MaxIntensity_threshold  = 2;
bnt_thres               = 0.6;

cd(pn);
bulbosity=[];
Hits=[];
NHits=[];
positive_fractions = [];
performance=[];

if validation==0
     num_ims=size(im_names);
     all_targets_truth=[];
     all_targets=[];
else
     num_ims=size(im_names_val);
end
```

```
%Run the algorithm on the desired set of images
%Collects segmentation data if in training state
for im_count=1:num_ims

    missed_targets_bnt=0;
    missed_targets_context=0;
    missed_targets_autogad=0;
    context_hits=0;
    bnt_hits=0;
    autogad_hits=0;


    if validation==0
        fn=char(im_names(im_count));
        fn2=char(truth_names(im_count));
    else
        fn=char(im_names_val(im_count));
        fn2=char(truth_names_val(im_count));
    end

    index = strfind(fn,'.');Name = sscanf(fn(1:index-1),'%c');
    temp_HSI = load(strcat(pn,fn));im_cube = double(temp_HSI.(Name));

    index = strfind(fn2,'.');Name = sscanf(fn2(1:index-1),'%c');
    truthimage = load(strcat(pn,fn2));truth =
double(truthimage.(Name));
    %----------------------Run AutoGAD--------------------------------
----
    [m,n,dims]=size(im_cube);
    num_pixels=m*n;
    data_matrix = reshape(im_cube,[num_pixels,dims]);
    dim_adjustment=0;

    good_bands = [10:97,115:132,158:200];
    %----------Keep bands that are not atmospheric absorption bands----
----
    % data_matrix_new=double(data_matrix(:,good_bands));
    data_matrix = data_matrix(:,good_bands);
    dims=size(data_matrix,2);
    % clear data_matrix;
    %-----------------------------------------------------------------
----

    %------------------------------Perform PCA------------------------
----
    [Ac,Lc,TotVarCompC,YscorC]=Center_and_PCA_optimized(data_matrix);
    Lplot=diag(Lc);
    %checks for eigenvalues 10^-4 and smaller and moves the endpoint of
the
    %eigenvalue curve to the point where eigenvalues are greater than
10^-4
    %so that the MDSL method in the next section is not biased by
    %pathological cases where the endpoint of the log scale eigenvalue
```

```
    %curve has extremely small endpoints and grossly alters the
theoretical
    %shape of the curve that should arise for eigenvalues of covariance
    %matrices of spectral data that follow the LMM
    while Lplot(dims)<=10^-4;
        dims=dims-1;
    end
    L=log10(Lplot);
    clear data_matrix;
    %---------------------Dimensionality Assessment--------------------
----
    %slope of line connecting endpoints of scree plot of eigenvalues
    m_slope = (L(1)- L(dims))/(1-dims);

    %calculate Euclidean distances from scree plot curve to line
connecting
    %endpoints
    dummy = ones(dims,1);
    x_int = (L - L(1)*dummy + m_slope*dummy + (1:dims)'./m_slope)./...
        (m_slope + 1/m_slope) ;
    y_int = L(1)*dummy + m_slope.*(x_int - dummy);
    Eqdist = sqrt( ( (1:dims)' - x_int).^2 + (L - y_int).^2)' ;
    clear x_int y_int dummy m_slope
    %find the point on the log scale eigencurve curve with the largest
    %distance from the line connecting the endpoints
    [max_Eqdist, index_dim] = max(Eqdist);
    clear Eqdist
    reduced_dim = index_dim;
    k=reduced_dim-1;
    k=k+dim_adjustment;
    percent_var=TotVarCompC(k,1);
    Y=YscorC(:,1:k);
    clear YscorC;


    % [ROC_data, RX] = Standard_RX_functional(im_cube,truth,k,35,1);



    %   User Input
    funct=2;%objective function in ICA to use.  Options [1=tanh,
2=pow3]
    orthog=1;%find ICs in parallel (symm) or one by one (delf).
    %Options [symm=1, defl=2]
    dim_adjustment=0;%how much to adjust max distance log scale secant
line
    %(MDLS) dimensionality decision
    max_score_thresh=10;%threshold above which decision is made to
        %declare target
    bin_width_SNR=.05;%bin width when using zero-detection histogram
    %method to determine breakpoint between background and potential
    %targets for calculating potential target SNR (PT SNR)

    bin_width_ident=.05;%bin width when using zero-detection histogram
    %method to determine breakpoint between background and targets for
```

```
    %identifying target pixels from selected target signals
    threshold_both_sides=0;%1=identifiy outliers on both sides of IC
signal,
    %0=identify ouliers on side with highest magnitude scores only
    clean_sig=1;%0 = no signal smoothing, 1 = signal smoothing prior to
    %target identification
    smooth_iter_high=100;%number of iterations to complete for
iterative
    %smoothing of low SNR object
    smooth_iter_low=20;%20;%number of iterations to complete for
iterative
    %smoothing of high SNR object
    low_SNR=5;%Threshold decision for choosing smooth_iter_low or
    %smooth_iter_high
    window_size=3;%image window size for smoothing
    iteration_coeff = 10;

    PT_SNR_thresh=2;%2;%threshold above which decision is made to
declare
    % target
    req_corr = 0.98514236; %Threshold correlation required for bands to
be
    % clustered together
    Kurtosis_thresh=9;%threshold above which decision is made to
declare
    % target
    target_fraction_thresh = 0.0269; %The maximum fraction of the image
    %expected to contain target pixels.
    Left_Kurt_Thresh=9;%If left side kurtosis is less than threshold
    %program will not perform thresholding on both sides for that map

    %Thresholds for AutoGAD v2 using ICA replacement
    %Kurtosis_thresh=3;
    %target_fraction_thresh = 0.2;


    good_bands = [10:97,115:132,158:200];

    AGAD_meth = 1;

    ICA_improv  = 0;
    ICA_gains = [0.001, 0.01, 0.1];
    ICA_gains2 = [0.1, 0.0001];

    if AGAD_meth == 1

        PT_SNR_thresh=2;%2;%threshold above which decision is made
        %to declare target

        [k,icasig, ICsig_gray, PT_SNR_line,PT_SNR,target_sig,...
            tgt_sig_map,   target_sig_clean_left,target_sig_clean,...
            linetrh_ident,linetrh_ident_left,rind,Lplot,...
            APER, TPF, FPF, Perc_tgt, target_pic_color] =...

AutoGAD_v1_function(im_cube,truth,good_bands,funct,orthog,...
```

```
dim_adjustment,max_score_thresh,bin_width_SNR,PT_SNR_thresh,...
            bin_width_ident,threshold_both_sides,clean_sig,...
            smooth_iter_high,smooth_iter_low,low_SNR,window_size,...
            ICA_improv,ICA_gains );

        wave = 149;
    %     figure
    % semilogy(Lplot(1:wave), '.-');
    % title({'Plot of Eigenvalue vs. PC Component',...
    %        sprintf('Dimensionality = %i',k)},'fontweight','b');


    elseif AGAD_meth == 2



        [k,icasig, ICsig,ICsig_gray, PT_SNR_line,PT_SNR,target_sig,...
            tgt_sig_map,  target_sig_clean,...
            linetrh_ident,linetrh_ident_left,rind,...
            APER, TPF, FPF, Perc_tgt, target_pic_color] =...

AutoGAD_v2_function(im_cube,truth,good_bands,funct,orthog,...
            max_score_thresh,PT_SNR_thresh,iteration_coeff,...
            clean_sig,window_size,...
            req_corr,Kurtosis_thresh,...
            target_fraction_thresh, Left_Kurt_Thresh,...
            threshold_both_sides,ICA_improv,ICA_gains2);


    end


    outliers_index=find(target_pic_color>=1);
    background_index=find(target_pic_color==0);

    num_TP = length(find(truth(outliers_index) == 1));
    num_FP = length(find(truth(outliers_index) == 0));
    num_TN = length(find(truth(background_index) == 0));
    num_FN = length(find(truth(background_index) == 1));

    %True Positive Fractions
    TPF = num_TP/(num_TP + num_FN);

    %False Positve Fractions
    FPF = num_FP/(num_FP + num_TN);

    %-----------Normalize the components to the zero-bin threshold-----
----
    %-----------Gather Context Data or make decision on regions--------
----
    [PT_SNRtemp,thresh_pt_ident] = find_PTsnr(target_sig_clean,...
        bin_width_ident,0);

    target_sig_norm=[];
```

```
    target_map_norm=[];

    context_targets_map=zeros(m,n);
    bnt_context_targets_map=zeros(m,n);
    bnt_context_targets_lvls=zeros(m,n);
    for i=1:size(target_sig_clean,2)
        %Normalize the component
        target_sig_norm(:,i)=target_sig_clean(:,i) ./
thresh_pt_ident(i);
        target_sig_map_norm =reshape(target_sig_norm(:,i),m,n);

        %Create map for segmentation
        thres_map = target_sig_map_norm > thres_map_threshold;

        %Gather context info
        context_info = regionprops(bwlabel(thres_map),...
            target_sig_map_norm,'all');
        truth_context = regionprops(bwlabel(thres_map),...
            double(truth==1),'all');

        context_targets = [];
        bnt_context_targets = [];
        for i=1:size(context_info)
            context_info(i).Bulbosity =
(context_info(i).MajorAxisLength...
                *
context_info(i).MinorAxisLength)/context_info(i).Area;
            context_info(i).Hit = (truth_context(i).MeanIntensity > 0);
            context_info(i).im_name=fn;
            if context_info(i).Hit == 1
                Hits = [Hits;context_info(i).MeanIntensity];
            else
                NHits = [NHits; context_info(i).MeanIntensity];
            end

            %Contextual Thresholding Filter
            if (context_info(i).MeanIntensity >
MeanIntensity_threshold...
                    | context_info(i).MaxIntensity > ...
                    MaxIntensity_threshold)& ...
                    context_info(i).MajorAxisLength/...
                    context_info(i).MinorAxisLength <...
                    aspectratio_threshold &...
                    context_info(i).Area >...
                    Area_threshold_low
                    context_info(i).Bulbosity > bulbosity_threshold;
                   context_targets = ...
                       [context_targets;context_info(i).PixelList];
            end

            %Prep data for BBN
            if validation==1
                evidence=cell(1,5);
                if context_info(i).MajorAxisLength/...
                        context_info(i).MinorAxisLength...
```

```
        <= aspectratio_threshold_0
    evidence(2)= {1};
elseif context_info(i).MajorAxisLength/...
        context_info(i).MinorAxisLength...
        <= aspectratio_threshold_1
    evidence(2)= {2};
elseif context_info(i).MajorAxisLength/...
        context_info(i).MinorAxisLength <=...
        aspectratio_threshold_2
    evidence(2)= {3};
else evidence(2)= {4};;
end

if (context_info(i).MajorAxisLength *...
        context_info(i).MinorAxisLength)/...
        context_info(i).Area <= bulbosity_threshold_0
    evidence(3)= {1};
elseif (context_info(i).MajorAxisLength *...
        context_info(i).MinorAxisLength)/...
        context_info(i).Area <= bulbosity_threshold_1
    evidence(3)= {2};
elseif (context_info(i).MajorAxisLength *...
        context_info(i).MinorAxisLength)/...
        context_info(i).Area <= bulbosity_threshold_2
    evidence(3)= {3};
else evidence(3)= {4};
end

if context_info(i).Area <= Area_threshold_low_0
    evidence(4)= {1};
elseif context_info(i).Area <= Area_threshold_low_1
    evidence(4)= {2};
elseif context_info(i).Area <= Area_threshold_low_2
    evidence(4)= {3};
else evidence(4)= {4};
end

if  context_info(i).MeanIntensity <=...
        MeanIntensity_threshold_0
    evidence(5)= {1};
elseif  context_info(i).MeanIntensity <=...
        MeanIntensity_threshold_1
    evidence(5)= {2};
elseif  context_info(i).MeanIntensity <=...
        MeanIntensity_threshold_2
    evidence(5)= {3};
else evidence(5)= {4};
end

[engine, loglik] = enter_evidence(engine, evidence);
marg = marginal_nodes(engine, tar);
for j=1:size(context_info(i).PixelList,1)
    bnt_context_targets_lvls(...
        context_info(i).PixelList(j,2),...
        context_info(i).PixelList(j,1))=marg.T(2);
```

```
                end
                if marg.T(2)>bnt_thres
                    bnt_context_targets = [bnt_context_targets;...
                        context_info(i).PixelList];
                end
                context_info(i).BNT_prob=marg.T(2);
            end
        end
        if validation ==0
            all_targets_truth= [all_targets_truth;truth_context;];
            all_targets=[all_targets;context_info];
        end

        %----------------------Make Context Decision-------------------
----
        for i=1:size(context_targets)
            context_targets_map(context_targets(i,2),...
                context_targets(i,1))=1;
        end
        for i=1:size(bnt_context_targets)
            bnt_context_targets_map(bnt_context_targets(i,2),...
                bnt_context_targets(i,1))=1;

        end
    end

    %Calculate Contextual Thresholding Pixel  Performance

    outliers_index=find(context_targets_map==1);
    background_index=find(context_targets_map~=1);

    num_TP = length(find(truth(outliers_index) == 1));
    num_FP = length(find(truth(outliers_index) == 0));
    num_TN = length(find(truth(background_index) == 0));
    num_FN = length(find(truth(background_index) == 1));

    %True Positive Fractions
    TPF2 = num_TP/(num_TP + num_FN);

    %False Positve Fractions
    FPF2 = num_FP/(num_FP + num_TN);

    %Calculate BBN Pixel Performane
    outliers_index=find(bnt_context_targets_map==1);
    background_index=find(bnt_context_targets_map~=1);

    num_TP = length(find(truth(outliers_index) == 1));
    num_FP = length(find(truth(outliers_index) == 0));
    num_TN = length(find(truth(background_index) == 0));
    num_FN = length(find(truth(background_index) == 1));

    %True Positive Fractions
    TPF3 = num_TP/(num_TP + num_FN);

    %False Positve Fractions
```

```
    FPF3 = num_FP/(num_FP + num_TN);

    %Calculate Region Performance of the algorithms
    truth_context_complete_bnt = regionprops(...
        bwlabel(bnt_context_targets_map),double(truth==1),'all');
    for j=1:size(truth_context_complete_bnt,1)
        truth_context_complete_bnt(j).hit=...
            truth_context_complete_bnt(j).MeanIntensity>0;
        bnt_hits=bnt_hits+truth_context_complete_bnt(j).hit;
    end
    truth_context_complete_context = regionprops(...
        bwlabel(context_targets_map),double(truth==1),'all');
    for j=1:size(truth_context_complete_context,1)
        truth_context_complete_context(j).hit=...
            truth_context_complete_context(j).MeanIntensity>0;

context_hits=context_hits+truth_context_complete_context(j).hit;
    end
    truth_autogad = regionprops(bwlabel(target_pic_color),...
        double(truth==1),'all');
    for j=1:size(truth_autogad,1)
        truth_autogad(j).hit=truth_autogad(j).MeanIntensity>0;
        autogad_hits=autogad_hits+truth_autogad(j).hit;
    end
    target_region=regionprops(double(truth==1),'all');
    missed_targets_bnt_region=regionprops(bwlabel(truth==1),...
        double(truth==1)-bnt_context_targets_map,'all');
    missed_targets_context_region=regionprops(bwlabel(truth==1),...
        double(truth==1)-context_targets_map,'all');
    missed_targets_autogad_region=regionprops(bwlabel(truth==1),...
    double(truth==1)-double(target_pic_color/4),'all');

    for j=1:size(missed_targets_bnt_region,1)
        missed_targets_bnt=missed_targets_bnt+(...
            missed_targets_bnt_region(j).MeanIntensity==1);
        missed_targets_context=missed_targets_context+(...
            missed_targets_context_region(j).MeanIntensity==1);
        missed_targets_autogad=missed_targets_autogad+(...
            missed_targets_autogad_region(j).MeanIntensity==1);
    end

    if validation==1
        performance(im_count,:)=[TPF,FPF,size(truth_autogad,1),...

missed_targets_autogad,autogad_hits/size(truth_autogad,1),...
            TPF2,FPF2,size(truth_context_complete_context,1),...
            missed_targets_context,context_hits/size(...
            truth_context_complete_context,1),...
            TPF3,FPF3,size(truth_context_complete_bnt,1),...
            missed_targets_bnt,bnt_hits/size(...
            truth_context_complete_bnt,1)];
    end


end
```

```
Attempt to execute SCRIPT test as a function:
I:\My Documents\Thesis\Data\context\test.m

Error in ==> AutoGAD_context at 15
if test==1
```

Published with MATLAB® 7.10

BIBLIOGRAPHY

Ashton, E. A. (1998). Detection of subpixel anomalies in multispectral infrared imagery using an adaptive bayesian classifier. *Geoscience and Remote Sensing, IEEE Transactions on, 36*(2), 506-517.

Bauer, K. W. (2010). *Course notes, OPER 685, applied mulivariate analysis I*

Bauer, K. W. (2011). *Course notes, OPER 785, applied multivariate statistics II*. Air Force Institute of Technology: doi:Jan 2011

Bieniecki, W., & Grabowski, S. (2004). Nearest neighbor classifiers for color image segmentation. Paper presented at the *Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2004. Proceedings of the International Conference,* 209-212.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput.Surv., 41*(3), 1-58. doi:10.1145/1541880.1541882

Davis, M. (2009). *Using multiple robust parameter design techniques to improve hyperspectral anomaly detection algorithm performance.* (Thesis, Air Force Institute of Technology).

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York, NY: John Wiley & Sons, Inc.

Johnson, R. J. (2008). *Improved feature extraction, feature selection, and identification techniques that create a fast unsupervised hyperspectral target detection algorithm.* (Thesis, Air Force Institute of Technology).

Landgrebe, D. A. (2003). *Signal theory methods in multispectral remote sensing*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Manolakis, D. (2003). Hyperspectral image processing for automatic target detection applications. *The Lincoln Laboratory Journal, 14*(1), 79.

Mathworks, I. (2010). *Measure properties of image regions.* Retrieved 01/24, 2011, from http://www.mathworks.com/help/toolbox/images/ref/regionprops.html

Miller, M. K. (2009). *Exploitation of intra-spectral band correlation for rapid feature selection, and target identification in hyperspectral imagery.* (Thesis, Air Force Institute of Technology).

Murphy, K. (2007). *Bayes net toolbox for matlab*

Shaw, G., & Manolakis, D. (2002). Signal processing for hyperspectral image exploitation. *Signal Processing Magazine, IEEE, 19*(1), 12-16.

Song, X., Wu, M., Jermaine, C., & Ranka, S. (2007). Conditional anomaly detection. *Knowledge and Data Engineering, IEEE Transactions on, 19*(5), 631-645.

Stein, D. W. J., Beaven, S. G., Hoff, L. E., Winter, E. M., Schaum, A. P., & Stocker, A. D. (2002). Anomaly detection from hyperspectral imagery. *Signal Processing Magazine, IEEE, 19*(1), 58-69.

Sun, P., & Chawla, S. (2004). On local spatial outliers. Paper presented at the *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining,* 209-216.

Xiang Wang, & Davidson, I. (2009). Discovering contexts and contextual outliers using random walks in graphs. Paper presented at the *Data Mining, 2009. ICDM '09. Ninth IEEE International Conference on,* 1034-1039.

Vita

Adam Messer entered the Air Force upon graduation from the United States Air Force Academy with a Bachelor of Science Degree in Operations Research. His first assignment was to the Force Management Division at Headquarters Air Force at the Pentagon. There he specialized in Enlisted Force Management analysis. After that assignment he started his studies at AFIT.

Upon graduation from AFIT he will proceed to Headquarters United States Air Forces Europe to work as an analyst on the staff.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
24-03-2011	**Master's Thesis**	Sep 2010 - Mar 2011

4. TITLE AND SUBTITLE
Contextual Detection of Anomalies in Hyperspectral Images

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Messer, Adam, J., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Street, Building 642
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-OR-MS-ENS-11-15

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NASIC/DAIA
Attn: John Jacobson
4180 Watson Way
Wright-Patterson AFB OH 4533-5648

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The majority of anomaly detectors in Hyperspectral Imaging use only the statistical aspects of the spectral readings in the image. These detectors fail to use the spatial context that is contained in the images. The use of this information can yield detectors that out perform their spatially myopic counterparts. To demonstrate this, we will use an independent component analysis based detector, autonomous global anomaly detector (AutoGAD), developed at AFIT augmented to account for the spatial context of the detected anomalies. Through the use of segmentation algorithms, the anomalies identified are formed into regions. The size and shape of these regions are then used to decide if the region is anomalous or not. A Bayesian Belief Network structure is used to update a posterior probability of the region being anomalous.

15. SUBJECT TERMS
Hyperspectral Imagery (HSI), Unsupervised Target Detection, Target Identification, Contextual Anomaly Detection

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Kenneth W. Bauer (ENS)
U	U	U	UU	90	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636, ext 4631; e-mail: Kenneth.Bauer@afit.edu