Enhancing Satellite Trail Detection in Night Sky Imagery with Automatic Salience Thresholding

Nikolaus Kollo

Submitted in partial fulfilment
of the requirements for the degree of
Master of Science in Applied Science

at

Saint Mary's University
Halifax, Nova Scotia
Canada

Approved by:	Dr. Jiju Poovvancheri Supervisor
	Dr. Yasushi Akiyama Supervisor
	Dr. Marcin Sawicki Examiner
	Dr. Jason Rhinelander External Examiner

Date: September 1, 2023

Abstract

Enhancing Satellite Trail Detection in Night Sky Imagery with Automatic Salience Thresholding

By Nikolaus Kollo

This study proposes a novel automatic thresholding method called Automatic Salience Thresholding (AST) for creating binary masks for detecting satellite streaks in night sky imagery. The approach utilizes a combination of Gaussian filtering, a salience-based thresholding technique, shape-based morphological filtering and line detection using Probabilistic Hough Transformations to identify the satellite trail in the image. We evaluated our method on diverse datasets of night sky images containing satellite trails in varying lighting conditions. The results show that AST outperforms the compared methods when tested with several performance metrics. The proposed AST method was also used to generate annotated binary masks for Hubble Space Telescope (HST) image data with promising results.

September 1, 2023.

Acknowledgments

I want to express my heartfelt gratitude to my supervisors, Jiju Poovvancheri and Yasushi Akiyama, for their unwavering guidance, invaluable insights, and continuous encouragement. Your mentorship has been instrumental in shaping my research and academic growth. I would extend my deepest thanks to Marcin Sawicki and Jason Rhinelander for providing me with their time and expertise.

To my beloved wife, Christine, I am deeply grateful for your unwavering support and understanding throughout this challenging academic journey. Your love and encouragement have been my constant source of strength and mean more to me than words can ever express.

I extend my sincere appreciation to the Canadian Forces and the Royal Canadian Air Force for providing me with the opportunity to pursue this academic endeavour. The support I have received from my administrators and commanding officers has allowed me to take on this rewarding challenge.

I want to thank my parents (Steve & Maria) and my in-laws (Gary & Candy) for their belief in me and continued support in all my endeavours. Your love and encouragement have been my motivation to strive for excellence.

Lastly, I am thankful to all my friends, colleagues and supporters (Porter, Mehfuz, Prachi, Cassie, Steven, Alex, Serena, Ellie and Chloe) who have been part of this journey, offering encouragement, assistance, and camaraderie.

This work is dedicated to my children, Austin & Olivia.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Satellite trail detection is an important task in the fields of astronomy, satellite tracking, and remote sensing, facilitating the identification and characterization of satellites orbiting our planet. Binary threshold masks are an intermediate step in the detection process and are essential for successful detection. However, this process poses numerous challenges, including varying lighting conditions, noise, and uneven backgrounds, which can adversely affect the performance of existing thresholding techniques. This thesis presents a comprehensive study on automatic thresholding for satellite trail detection, focusing on developing a novel algorithm called Automatic Salience Thresholding (AST). This research aims to address the limitations of current methods and provide a robust solution that results in the accurate and efficient detection of satellite trails in diverse image scenarios. This chapter motivates the detection task, states the key contributions of the study, and outlines the remaining chapters in the thesis.

1.1 Lights in the Sky

The explosion of growth in the commercial space sector has caused the night sky to transform as more satellites accumulate in Earth's orbit. The rapid increase coincides with progress in mankind's ability to transact and communicate globally instantaneously. This progress, however, does not come without a price, and its effects are being felt by those who study distant stars to understand our universe. The clusters of artificial satellites that circle our planet work in communication networks called mega-constellations [1].

Satellites are strategically placed in Low Earth Orbit (LEO) owing to their short orbital periods and low transmission latency, enabling efficient data transmission and response times. However, the close proximity of LEO satellites has an unintended consequence during twilight hours when the sun's rays reflect off the satellite's body and solar arrays. This phenomenon causes the satellites to become as visible as the stars in the sky. In long-exposure imaging, these fast-moving faint objects create extended trails that mar the images and lead to the loss of valuable observational data in affected regions. Attempts have been made to reduce the impact of Starlink satellite trails by installing sun visors and using lower albedo construction materials, but neither approach has successfully eliminated the problem [2].

Satellites are not a new issue for those who perform night sky imaging, and contaminated exposures are usually left out of the final image stack. Manual image masking for individual exposures can be labour-intensive at scale, and automatic interventions are usually preferred. The historic growth of satellites/active payloads (see Figure 1.1 [3]) suggests that the preference for automation will likely continue. It's not only the increase in the number of satellites that causes concern but also that larger and, therefore, more visible satellites are being deployed in LEO. Figure 1.2 [3] shows a sharp increase in 2017 that coincides with the commercialization of the space launch industry. This will mean that more exposures will be excluded from the final image stack, resulting in an overall loss of observational efficiency.

While wide-field and long medium-field images would only notice a small increase of activity during twilight, ultra-wide-field imaging will likely suffer [1]. A recent study exploring the impact of satellite trails on the Zwicky Transient Facility (ZTF) found that the number of affected images increased from 0.5% to 18% between late 2019 and 2021 [5]. The negative effects are not limited to only ground-based telescopes; the Hubble Space Telescope (HST) has also been impacted. A study that quantified the fraction of HST exposures containing at least one satellite trail over twenty years

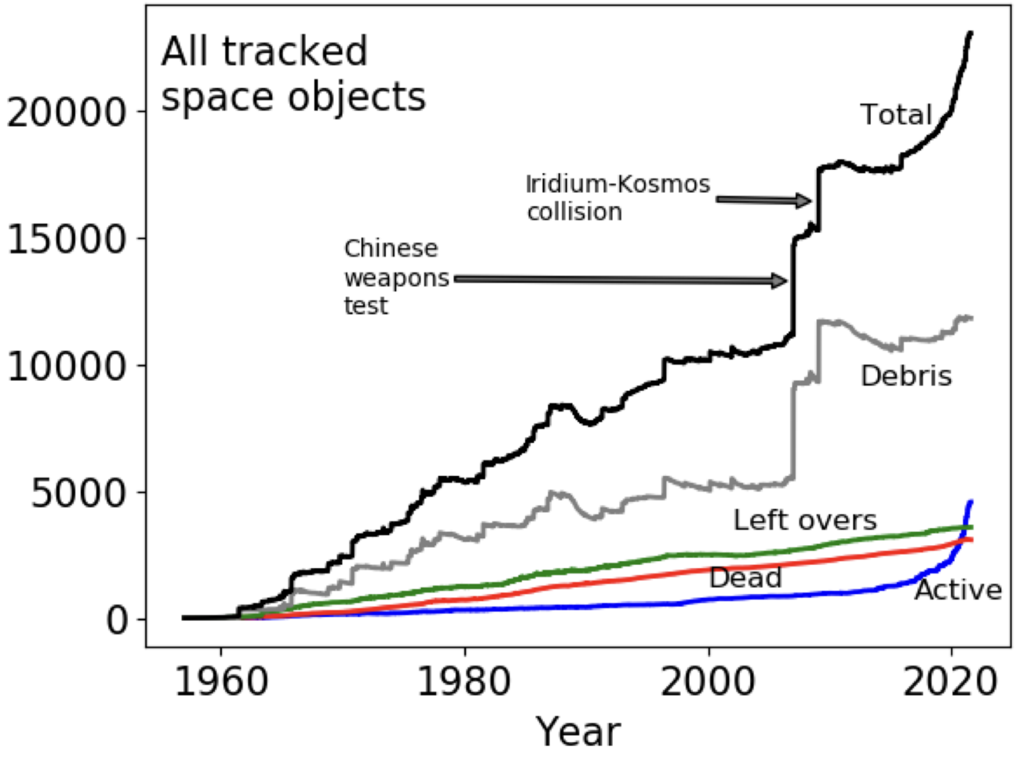


Figure 1.1: The rapidly growing number of active satellites in orbit over the last 60 years. Reproduced with permission from Springer Nature [4].

found that the occurrence has more than doubled from 2% to 5% [6].

The impact on Space-borne telescopes is not only limited to their observational efficiency. Satellites acquire their positions and orientations without prior knowledge of position by solving the "lost in space" problem. The problem is split into two parts: acquisition and tracking. During acquisition, the satellite utilizes the stars to determine its orientation by identifying star patterns and inferring attitude information. After identifying a star pattern, the star tracker switches to a recursive tracking mode that maintains a frame of reference based on the visible stars in the field of view. Star identification algorithms use pattern matching or angular distance calculations to determine the stars in the field of view. Both algorithms generally become less accurate

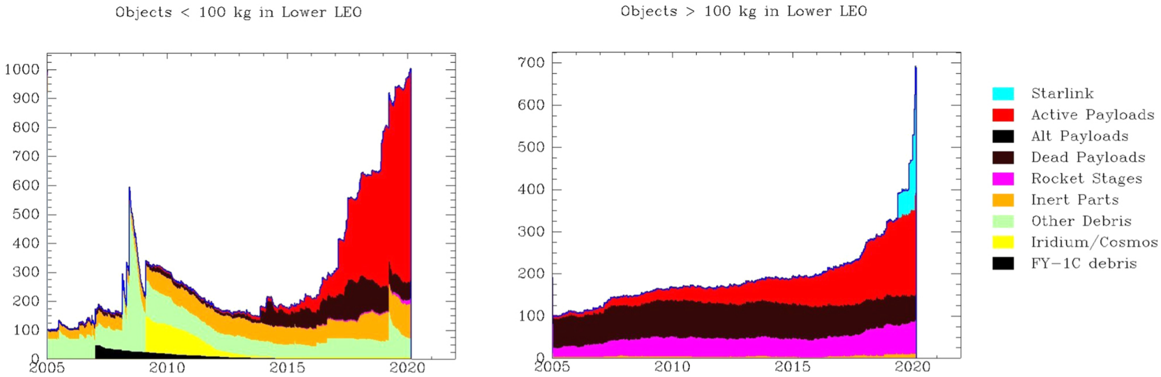


Figure 1.2: Objects in LEO classified by size showing the explosive growth of both large and small satellites [3].

when the image contains false or missing stars [7].

Ultra-wide-field ground-based imaging is the most vulnerable to satellite trail contamination. Modelling of the impact of the 12,000 proposed Starlink satellites to the Vera C. Rubin Observatory in Chile found that 30% to 40% of the images at twilight would contain at least one satellite trail [2]. A similar study confirmed these results and added 17 proposed mega-constellations totalling 26 thousand modelled satellites [1]. The study found that the larger Starlink satellites that occupied LEO's lower region were the greatest contributing factor for satellite trails in ultra-wide-field imaging.

SpaceX can deploy 60 Starlink satellites in a single launch and will conduct 32 launches in 2022. A major milestone was achieved in May 2023, when the four thousandth Starlink satellite was launched into orbit. At the time of writing, Starlink has requested regulatory approval from the Federal Communications Commission (FCC) to launch nearly thirty thousand satellites for the Starlink mega-constellation [5].

Smaller satellites like the ultra-compact low-cost CubeSats have also contributed to the number of satellites orbiting in LEO. Figure 1.1 shows a growing number of rocket stages, dead payloads, debris, and inert parts that increase the risk of collision. These defunct objects renew early fears of the formation of impassable debris belts created by chain reaction collisions of artificial satellites orbiting the Earth [8]. The growing number of satellites and the catastrophic scenarios like the chain reactions of the Kessler syndrome provide a clear motivation for satellite trail detection for future skies.

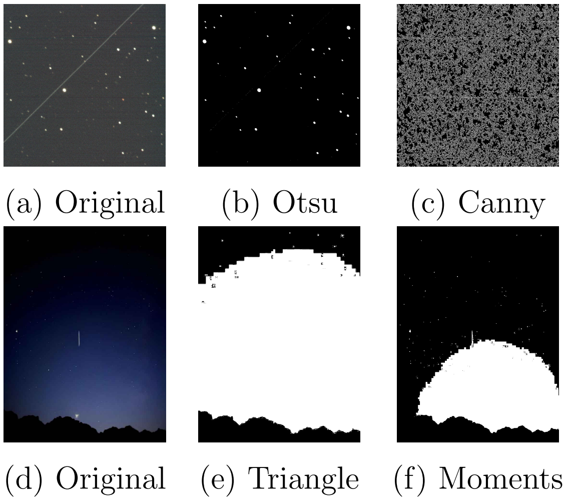


Figure 1.3: A comparison of automatic threshold mask generation shows data loss during binary thresholding. Figures (a) and (d) are ground-based images from the NASA Streak Watcher dataset [9]. Figures (b) and (c) show examples of over-thresholding and under-thresholding, respectively. The uneven background in Figure (d) results in over-thresholding in Figures (e) and (f).

1.2 Space Domain Surveillence

Satellites do not report their positions in real-time to a central organization in a standardized and universal manner. Satellites' positions are typically tracked and monitored by ground-based tracking systems, such as the U.S. Space Surveillance Network (SSN), which utilize radar, optical telescopes, and other tracking technologies to determine their positions, trajectories, and orbits, including satellites, space debris, and other man-made objects [10]. The project aims to obtain the most up-to-date information on all the objects orbiting Earth to achieve space situational awareness (SSA). This ground-based sky surveillance is performed to avoid collisions and monitor suspicious activity. With the rapid growth of orbital objects to track, having the most up-to-date information about space activity is becoming more essential daily.

Much of the difficulty of detecting satellite trails is associated with the low signal-to-noise ratio common in night sky imaging. The image exposure responsible for creating the trail also results in additional noise that can cause false or missed detections. The amount of generated noise can vary from one exposure to another depending on environmental factors such as temperature and light conditions. Some of these challenges can be overcome with a great deal of pre-processing that comes at the expense of speed.

It is often necessary to binarize the input to reduce the computational complexity before detection. Using a fixed intensity value for the binary threshold would not tolerate much input variability. Automatic global binary thresholding algorithms use calculated metrics to determine a single threshold value for the entire image. An advantage of automatic global thresholding methods is that they are computationally efficient and do not require kernel tuning. A disadvantage of automatic global thresholding is that noise and uneven backgrounds can influence the calculated metrics, as seen in Figures 1.3(b,c,e,f). In these scenarios, the satellite trail is destroyed before the detection stage in the pipeline.

Over the years, satellites were primarily viewed as contaminants, and their detection only gained significance with the commercialization of space launches. Consequently, there has been a scarcity of publicly available satellite trail data suitable for training machine-learning models. As a solution, researchers turned to synthetic data generation methods to augment their training datasets. While recent times have seen a rise in available data, a considerable challenge remains in the form of unannotated datasets, necessitating a substantial effort to complete the pixel-level annotation process effectively.

1.3 Contributions

In this study, we present several significant contributions to the field of satellite trail detection and image processing:

1. We propose a novel automatic global thresholding algorithm designed specifically for efficiently binarizing night sky images with sparse bi-modal histograms. By mitigating the influence of image noise and uneven backgrounds, our method enhances the accuracy and robustness of satellite trail detection.

2. A novel detection pipeline is introduced, capable of detecting satellite trails in RGB images without relying on prior scene information. The pipeline's flexibility allows for the processing of various image sizes and fields of view, while a shape filter effectively removes stars and other irrelevant objects, enabling detection in images with arbitrary sizes.

3. To ensure an accurate evaluation, we curate a ground truth dataset by collaborating with citizen scientists through a dedicated website [11]. This annotated dataset serves as a reliable reference for assessing the performance of our generated detection masks.

4. To facilitate further research and the development of machine-learning models for satellite trail detection, we make our annotated dataset publicly available. By providing access to real-world satellite trail images, we aim to accelerate advancements in automatic detection methods.

These contributions collectively advance the state-of-the-art in satellite trail detection and hold the potential to enhance space situational awareness, contributing to the safety and sustainability of space operations.

1.4 Outline

The study is divided into five parts, and relevant sections, tables, and figures can be directly referenced from the table of contents found on page iii. of this study. The first chapter provides a description of the problem and the challenges that are frequently encountered with satellite trail detection. The section also provides a clear motivation as to the importance of this type of research to ground and space astronomy, as well as SSA.

The second chapter provides the reader with information on the various methods used for satellite trail detection in recent literature. This study relied on the presented works to help synthesize a pipeline structure that could achieve our detection objectives.

The third chapter describes the operation of the proposed Automatic Salience Thresholding algorithm and detection pipeline in detail. Information on the shape filter and line detection stages will also be described in detail.

The fourth chapter will contain information about the datasets, the citizen science website, and the experiments performed to evaluate our proposed methods.

The fifth chapter will provide a discussion and conclusion for the study and outline possible future studies that this work could benefit.

Chapter 2

Literature Review

The surveilling of artificial satellites is a critical aspect of achieving a high level of SSA and ensuring the safety and security of objects in space. The SSN is vital in tracking, identifying, and cataloging all artificial objects in space, utilizing ground-based optical and radar imaging systems. However, one of the significant challenges in satellite surveillance lies in detecting faint satellite trails in astronomical images with low signal-to-noise ratios. To address this challenge, researchers have utilized various detection methods reviewed in this chapter. The methods that will be discussed include streak filter/kernel detection, Radon transform detection, clustering detection, Hough line detection, and machine learning detection. Each approach has its strengths and limitations, making it imperative to conduct a review and evaluation for optimal performance.

This chapter reviews the literature and existing research on satellite trail detection methods, star removal techniques, and automatic thresholding algorithms. By analyzing the strengths and limitations of each approach, the groundwork is laid for developing an efficient and accurate automatic global thresholding algorithm for satellite trail detection. The insights gained from this review will pave the way for the integration of the proposed algorithm into a state-of-the-art detection pipeline, ultimately contributing to the advancement of space surveillance and SSA efforts.

2.1 The Surveilling of Artificial Satellites

The SSN describes four categories of space object identification (SOI): wideband, narrowband, photometric, and optical imaging. Wideband SOI provides a detailed radar picture of the satellite. Narrow-band SOI provides a two-dimensional depiction of the radar energy charted on a graph as amplitude versus time. Photometric SOI is the analysis of the intensity, luminance, and illuminance. Finally, optical imaging SOI refers to object identification obtained using optical telescopes. These space sensors collate position information into a single domain to achieve space situational awareness. High-power sensors like the Air Force Space Surveillance System (AFSSS), also called the Fence, use three transmitters and six receivers to illuminate objects that pass through the observation area and then detect them using a triangulation of the reflected signal. The system covers 10 percent of the Earth's circumference and extends 15,000 miles into space [10]. The two disadvantages of a phased array radar are its high cost and complex maintenance. Smaller, less powerful arrays have been constructed using a single Yagi-Uda antenna to detect, track and catalogue objects in LEO. The array has a limited range compared to the AFSSS, but these smaller arrays can be built for a fraction of the cost [12].

The Ground-Based Electro-Optical Deep Space Surveillance (GEODSS) System utilizes a global network of optical detection systems integral to the SSN. To maintain the stars' positions within the field of view, the telescopes move in synchrony with the motion of the stars across the sky. This enables the GEODSS cameras to capture electronic snapshots of the field of view swiftly. The images are subsequently stacked and processed so that the movements of manmade space objects become visible streaks across the image plane. The streaks are measured for calculating object positions, such as satellites positioned between 5,000 to 35,000 kilometres in orbit. This information is transmitted from the sites to the Joint Space Operations Center, facilitating the constant update of the orbiting objects list. The GEODSS system can remarkably track objects as small as a basketball, even at distances exceeding 32,000 kilometres in space [10]. The following sections will review the literature related to the stages of satellite trail detection identified in Figure 2.1. The remaining sections

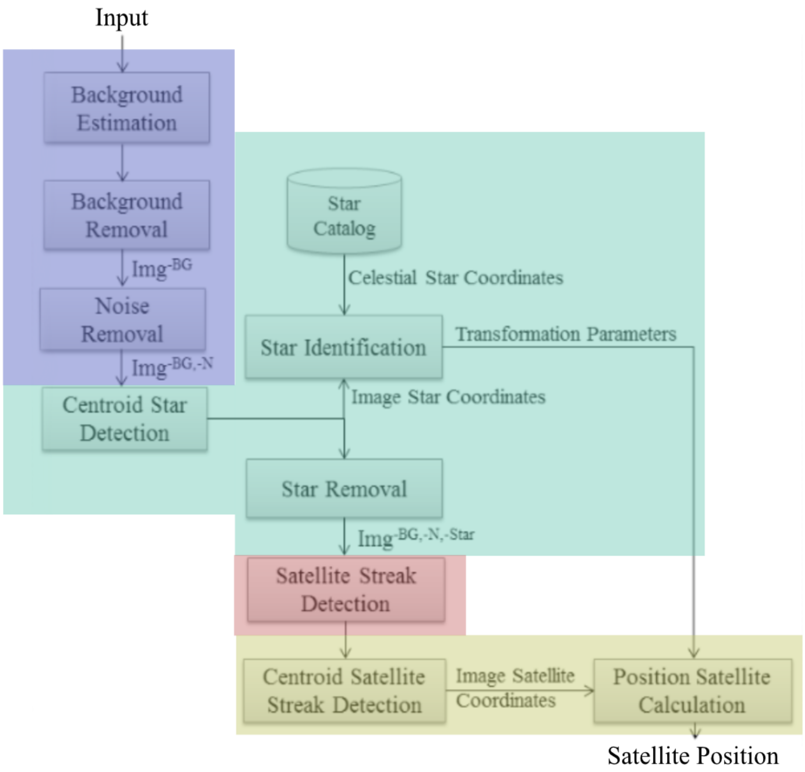


Figure 2.1: This system diagram illustrates the design and implementation of an optical astronomical satellite tracking system. The system comprises several key stages, including processing (blue region), star removal (green region), trail detection (red region), satellite identification and cataloguing (yellow region) [13].

in this chapter will discuss the stages of the satellite detection process, excluding the satellite identification/cataloguing stage (yellow), as it is not a core process for extracting the visual signal of a satellite trail. To highlight the input preconditions required by each stage, the processing stages will be discussed in reverse order, starting from the detection stage (red).

2.2 Satellite Trail Detection Stage

This section covers the different types of satellite trail detection algorithms used in recent literature. The information from this section was important for selecting the most effective detection method for the proposed pipeline in Section 3.3. Given the sharp increase of artificial objects in Earth's orbit, detecting satellite trails in astronomical images has become an important part of achieving a high level of SSA. The image-processing pipeline shown in Figure 2.1 provides a high-level overview of how this is accomplished in practice. In the blue area, the input undergoes a series of processing steps to manipulate the image before detection. The green area contains steps to remove stars and other artifacts to improve detection outcomes further. The red region is where the spatial position of the satellite trail is detected and is critical to all preceding and forthcoming steps. The yellow area contains steps for identifying, cross-referencing, and cataloguing detected satellites. To best understand the algorithms that support detection, it is fundamental that the different types of detection are well understood.

2.2.1 Streak Filter/Kernel Detection

A Defence Research and Development Canada (DRDC) study [14] explains detecting satellite trails using a double-gated filter. Every detected star is tested using the filter seen in Figure 2.2, and objects that spill into the outer regions of the kernel are flagged as possible satellite trails. The study concedes that very bright stars can deceive the filter; the detection method remains sensitive to dim, faint streaks without identifying the stars in the frame. While certain circumstances can cause false positives, forgoing the star identification step eliminates a substantial amount of pre-processing overhead.

An optical detection system was proposed by Samadzadegan et al.[13] that is configured similarly to GEODSS. The system was designed and implemented to track, detect and catalogue satellites using an image-processing pipeline that repackages the double-gated filter to detect stars and reject trails. A functional diagram of the

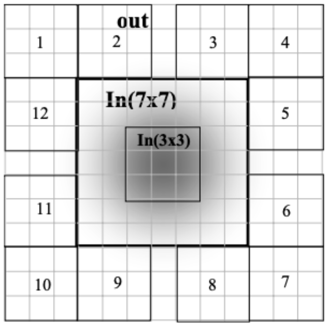


Figure 2.2: The double-gated filter used for detecting expected satellite streaks in astronomical images. The filter is designed to identify potential satellite trails by analyzing the intensity levels of detected stars in the image. Objects that extend beyond the outer regions of the filter are flagged as possible streaks caused by satellites transiting through the field of view [14].

pipeline proposed in the study is shown in Figure 2.1. The diagram shows how the stages of the detection process interact and represent computational overhead for satellite trail detection. In the star identification stage, the visible stars are analyzed and identified to establish a reference grid before removing them to improve satellite trail detection. The detection step abandons the double-gated filter as a satellite trail detection method and uses a Fast Fourier Transform and a Radon Transform to detect the satellite's orientation.

2.2.2 Radon Transform Detection

A 2018 study by Nir et al.[15] used a similar method to Samadzadegan et al.[13] for detecting satellite trails in astronomical images. The authors used a filter with a dynamic width that can adjust to the image point spread function (PSF) for improved dim star detection. The Radon Transform [16] is again used for line detection, this time on a variance map calculated in an earlier processing step. The method is a mathematical technique to detect lines or line-like structures in an image. It represents an image in a transformed domain called the Radon space (see Figure 2.3). The transformation only provides a set of points in the Radon space, each corresponding to a line passing through the detected areas in the input image. The Radon Transform

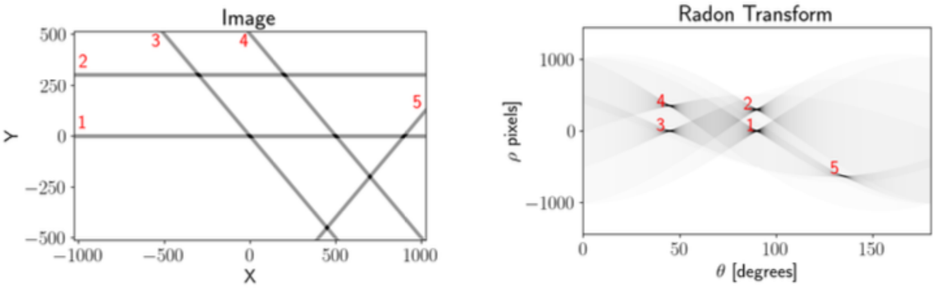


Figure 2.3: The left figure illustrates the Radon Transform's outcomes applied to the input image for line detection. The detections are represented in the Radon space shown on the right, each point corresponds to a detected line in the original image, characterized by its polar coordinates (p, θ) [17].

cannot precisely identify the full extent of long and curved lines, as it represents them using points in the Radon space. The study also experimented with a fast Radon transform detection method that improves detection from $O(N^3)$ to $O(N^2 * P)$ for an $N \times N$ image, where P is the number of projection angles. The algorithm was tested using real and simulated images and is capable of short and long-trail detection with a low false alarm rate given a 2048 × 2048 input.

A 2022 Space Telescope Science Institute report presented a Mean Radon transform-based detection method for identifying satellite trails in Advanced Camera for Surveys (ACS) Wide Field Channel (WFC) data from the Hubble Space Telescope [17]. The method increases the time complexity of the original Radon transform algorithm in exchange for increased dim trail detection sensitivity and robustness against bright sources. The study identified that the method was prone to false detections in approximately 10% of images. Images with densely populated star fields or when trails appear in the image corners were the cases most prone to false detections [17].

2.2.3 Clustering Detection

A real-time satellite trail detection system was proposed in 2017 that utilizes a lightweight image processing pipeline with a clustering trail detection stage [18]. Notably, the processing pipeline does not utilize star identification, favouring scale-invariant feature transform (SIFT) [19]. The study compares k_means [20], particle swarm optimization [21], genetic algorithm [22], and Gaussian mixture model to determine the optimal algorithm for satellite trail detection.

The genetic algorithm detection method had the highest accuracy, and all detection methods could detect the satellite trails in the test set. A genetic algorithm (GA) comprises multiple stages; identifying unknown variables is initially determined based on the specific data and problem. Subsequently, these variables are encoded and represented as chromosomes in a suitable format. A fitness function is then defined for the chromosomes, considering the cost function. The initial population is randomly selected, and the fitness value for each chromosome is computed. Subsequently, subsequent steps are carried out sequentially, following the process outlined in Figure 2.4. A particularly interesting feature of the pipeline described is that image segmentation occurs after removing the visible stars. The reason for this was likely to take advantage of SIFT for star feature extraction rather than relying on a fixed detection kernel.

2.2.4 Hough Line Detection

The Hough transform [23] stands as a fundamental technique within the realms of computer vision and image processing, excelling at the detection of shapes like lines and circles. In this transformative approach, the shapes present in the input image are transposed into curves within a parameter space, as vividly illustrated in Figure 2.5. This unique perspective enables the identification of shapes, even when they're fragmented, distorted, or concealed. The classic formulation of the Hough transform involves discretizing the parameter space. Traditionally, its time complexity is estimated as $O(N^2M)$, wherein N signifies the number of data points or image pixels, and M denotes the count of bins or cells within the parameter space.

In contrast, the Probabilistic Hough Transform leverages a more strategic sampling approach, which proves particularly advantageous. Due to the markedly smaller number of sampled points than the overall data points, this variant's time complexity becomes linear—amounting to $O(N)$. This scalability grants the Probabilistic Hough

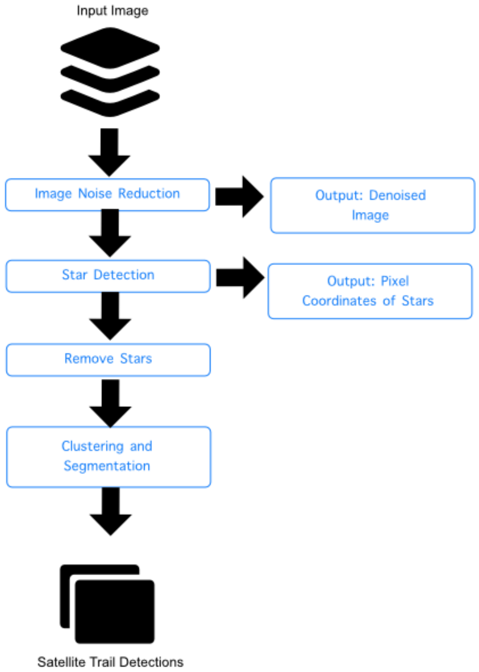

Input Image
Image Noise Reduction
Output: Denoised Image
Star Detection
Output: Pixel Coordinates of Stars
Remove Stars
Clustering and Segmentation
Satellite Trail Detections

Figure 2.4: A detection pipeline that uses image processing techniques to detect and remove stars so that clustering algorithms can detect satellite trails in the denoised and filtered images [18].

Transform a commendable performance boost, rendering it especially well-suited for scenarios where efficiency is paramount. Furthermore, the Probabilistic Hough Transform maintains the distinctive ability to decipher shapes amidst partial, distorted, or obscured information, enhancing its utility across various image analysis tasks.

The SatDet satellite detection software suite was introduced in 2016 as a part of the ACSTools package [25]. The detection software demonstrated good performance in detecting satellite trails in HST ACS/WCF image data. The method was tested on the Hubble Frontier Fields (HFF) data, and a manual review showed a low false-positive rate. The SatDet software does not use a filter for detecting stars or any star identification techniques. Instead, it uses the Canny edge detection algorithm

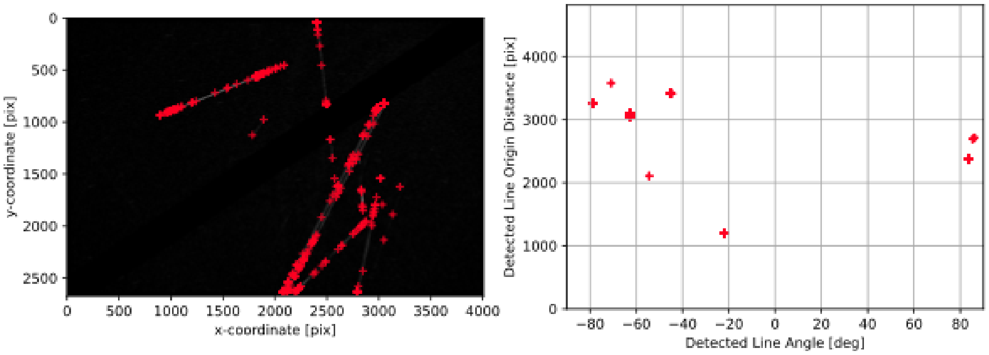


Figure 2.5: The left figure displays the results of satellite trail detection using the Probabilistic Hough transform. The algorithm localizes the satellite trails in the input image, represented as red streaks against a dark background. In the right figure, the detections from the Probabilistic Hough transform are visualized in Hough line space (r, θ) [24].

to segment the image and a size filter to remove regions smaller than 75 pixels. The probabilistic Hough transform detector then processes the remaining objects.

A 2019 study by Privett et al.[26] investigated the practicality of readily deployable equipment that can autonomously detect satellite trails in wide-angle optical systems. The authors use a static global thresholding algorithm with the classic Hough transform for satellite trail detection. The method is computationally lightweight but sensitive to noise and image artifacts. Danarianto et al.[27] also proposed a similar idea, showing that the method can be used on amateur and professional-grade telescopes.

In 2022, a study by Rood et al.[24] sought to decrease the processing speed and increase the detection performance by making the imaging device stationary rather than tracking the sky. This reduces the over-exposure of the visible stars while also detecting geosynchronous satellite orbits. Individual exposures are aligned and stacked before detection. The completed coadd is then windsorized, and the standard deviation and mean of the image are calculated to determine a threshold value of the mean plus three times the standard deviation. The stars are removed by subtracting a mask that accumulates point sources that maintain a fixed relative pattern. The identified limitations of the method include dim stars evading the star filter and satellite trails being mistakenly filtered out [24].

2.2.5 Machine Learning Detection Methods

A Hough transform-based detection method was used by Rachith et al.[28] to train a Lookup-based Convolutional Neural Network (LCNN) to detect satellite trails with a subset of VLT Survey Telescope (VST) images. The pre-trained network weights were originally used to detect lines in urban environments and utilize a trainable Hough transform block. The network was trained using 244 mosaics corresponding to 7168 individual images, which allowed the network to learn the features for detecting trails in VST images. No image augmentation was described for the training images in the study, which may have contributed to the final performance outcomes. The authors also created a classic Hough transform trail detection pipeline with similar characteristics to the SatDet method, which was used to compare the performance of the LCNN. The described detection results for the LCNN were reported as disappointing, according to the qualitative analysis conducted by the authors. It was found that while the model can detect satellite trails in many of the input images, it also generates many false positives at the edges of the image. The explanation for the excess false positive detections was related to the original algorithm being optimized to detect many lines in a given image rather than just a few. The classical detection pipeline, on the other hand, demonstrated comparable results to many of the aforementioned studies. The limitations observed by the authors for the classical pipeline were a failure to detect shorter trails due to a hardcoded binary threshold value and size filter.

Paillassa et al.[29] proposed using machine learning to detect satellite trails and other contaminations in astronomical images in a 2020 study. The study introduces two convolutional neural network (CNN) classifiers designed for detecting contaminants in astronomical images. The first classifier, MaxiMask, performs semantic segmentation and generates bad pixel maps for different types of contaminants, including cosmic rays, hot and bad pixels, persistence effects, satellite or plane trails, residual fringe patterns, nebulous features, saturated pixels, diffraction spikes, and tracking errors. The second classifier, MaxiTrack, assesses the probability of tracking errors affecting the entire images and mosaics. The researchers collected training and testing data from real observations and augmented them with image simulations to develop and evaluate the performance of both classifiers. The study demonstrates that MaxiMask identifies satellite trails and cosmic rays well in ZTF and HST data. The authors attribute the performance outcomes to the availability of several optical instruments in the private COSMIC DANCE archive [30]. The authors added synthetic contaminations for the training data to overcome the enormous annotation burden. The training set used also does not include images from space-borne telescopes, and the study identifies that HST data pushes the detection limits of both classifiers.

2.3 Object Filtering for Improved Trail Detection

This section covers the different types of star and small object filtering used in recent literature in the context of satellite trail detection. The information from this section was important for selecting the most effective star removal method for the proposed pipeline in Section 3.2. The green area in Figure 2.1 contains steps for removing stars and other small objects from the image before the satellite trail detection stage. The following section will discuss the methodologies used in the aforementioned studies for star object detection/removal and discuss the strengths and weaknesses of each.

2.3.1 Filter-based Star Detection

Detecting stars in images using a kernel/filter is a common technique in astronomical image processing. The process involves convolving the image with a specific kernel or filter to enhance the star-like features and suppress the background noise. The kernel typically has a shape that matches the expected profile of a star. Often, a star is likely to have a Gaussian distribution due to the effects of PSF, which increases the apparent size of the point source. By convolving the image with this kernel, the intensity of the star-like features is enhanced while the background noise is smoothed out. The convolved image is then thresholded to identify regions that exceed a certain intensity threshold, indicating the presence of stars. Additional post-processing steps may be applied to refine the detected star positions and remove false detections. This approach allows astronomers to automatically detect and locate stars in images, enabling various astronomical analyses and measurements. The 2013 Iranian study by Samadzadegan et al.[13] adopted this optimization to reduce the processing time of their proposed detection pipeline. The 2009 study by Lavesque [14] also suggests that brighter stars and stars with diffraction spikes could be misidentified and could ultimately cause errors in the star removal and satellite detection stages.

2.3.2 Star Identification

Another method for removing stars from astronomical images is to utilize star identification algorithms commonly used in satellite navigation to solve the lost space problem. The process extracts a subset of the brightest visible stars and uses the pattern recognition pyramid algorithm due to its speed and robustness to false stars. A star catalogue is used to identify stars by matching scale-invariant partial or complete patterns and using an error threshold to calculate the level of uncertainty about the prediction. The method is considered fast and efficient compared to other pattern and angular methods but is generally used in conjunction with space-borne images [31].

Eliminating stars and other diminutive entities from an image results in a reduction of data points taken into account during the satellite trail detection stage. Nonetheless, identifying visible stars is optional for satellite trail detection, as showcased by a recent satellite trail detection pipeline [24] that showed that detecting and removing stars is satisfactory for satellite trail detection. The feature extraction phase within star identification pipelines exhibits numerous commonalities with filter-based and size-based star detection techniques.

2.3.3 Satellite Detection without Star Identification

Recent satellite trail detection pipelines [24, 25, 27] have moved away from star identification as a method for removing stars from image backgrounds in favour of image processing techniques. Stars are point source objects but are often subject to the effects of PSF due to imperfections in the optics or atmospheric conditions. This results in point source objects appearing larger and dimmer, producing a distinct grouping of pixels that can be detected through various methods, such as image analysis algorithms or by known properties of the imaging system [32].

Analyzing the size of detected objects in the image makes distinguishing stars from other larger objects or artifacts possible. Once the stars are identified based on size, they can be selectively removed or masked in the image to mitigate their impact on subsequent detection tasks. The primary limitation of using size thresholds for removing stars from astronomical images is that bright stars and diffraction spikes may evade detection. Other issues concerning this filtering technique relate to requiring prior knowledge of specific input image sizes to set the size threshold appropriately.

2.4 Automatic Thresholding for Image Binarization

This section covers the different automatic thresholding methods used in recent literature that can be used to reduce the complexity of the input. The blue area in Figure 2.1 contains steps for removing the noise and the background from the image before the star removal process. This step allows the star identification step to act on binary information, which is ideal for determining the centroid of a star so that it can be removed. The information from this section was important for selecting the most effective preprocessing method for the proposed pipeline in Section 3.1.

Automatic thresholding methods can be divided into local and global thresholding, where each has distinct advantages and disadvantages depending on the input. Local thresholding involves selecting localized threshold values based on neighbourhoods of pixels. The effectiveness of local thresholding methods for dim object detection can vary depending on noise and object size. For example, suppose the objects of interest are small and surrounded by much noise. In that case, local thresholding methods may not be effective because the local neighbourhood around the object may also contain noise. This will result in a more restrictive threshold value for the region. These methods are more computationally complex than global methods and require selecting an appropriate kernel size for the best performance. Global binary thresholding involves selecting a single threshold value applied to the entire image. All pixels in the image with an intensity value above the threshold are assigned to the foreground, while the rest are assigned to the background. Global thresholding is a simple and computationally efficient technique that works well when the foreground and background pixels have distinct intensity values (bimodal) and the lighting conditions are relatively uniform across the image.

The selection criteria for determining the threshold value can be found in various ways. The classical types of automatic global thresholding algorithms can be divided into three classes: histogram-based, entropy-based and edge detection methods.

2.4.1 Histogram Thresholding

The histogram-based algorithms are a family of image thresholding methods that automatically use the image's intensity histogram to determine the threshold value. One of the most commonly used histogram-based thresholding methods is Otsu's method [33], also known as the maximum variance thresholding method. This method calculates the threshold value that maximizes the variance between the two classes of pixels, which is equivalent to minimizing the intra-class variance.

Another common histogram-based thresholding method is the Triangle algorithm [34], which assumes that the image's histogram is unimodal and determines the threshold value by constructing a triangle using the peak modal. A line segment is drawn from the peak of the histogram to the last non-zero bin in the histogram. Because the triangle method assumes a uni-modal histogram, it is designed to choose a threshold value near the base of the primary histogram mound.

Some other histogram-based thresholding methods included in this study are the Moments method [35], the Mean method [36], and the Minimum Error thresholding method [37], which determine the threshold value based on different statistical measures of the image histogram. The advantage of histogram-based automatic thresholding algorithms is that they are simple and computationally efficient.

2.4.2 Entropy Thresholding

Entropy-based thresholding methods are a category of image segmentation techniques that leverage the concept of entropy to determine the most suitable threshold value for binarizing an image. Entropy is a measure of the information content or uncertainty in an image, and it quantifies the level of disorder or randomness in the pixel intensities. In entropy-based thresholding, the image's histogram estimates the probability distribution of pixel intensities. The entropy of this distribution is then computed as a measure of the information content in the image.

The objective of entropy-based thresholding is to find a threshold value that maximizes the entropy of the foreground and background regions, thus creating a clear separation between them. A high entropy value indicates a higher level of information or diversity in the pixel intensities, generally corresponding to regions with mixed foreground and background elements. Conversely, low entropy suggests a relatively uniform distribution of pixel intensities, typically corresponding to either foreground or background regions. Unlike histogram-based thresholding methods, which often rely on specific statistical properties of the histogram (e.g., variance or bimodality), entropy-based thresholding approaches do not make any assumptions about the distribution of pixel intensities.

One popular entropy-based thresholding method is the Max-Entropy thresholding algorithm [38], also known as Kapur's method. This method calculates the entropy of the foreground and background pixel intensities, and the threshold value is chosen to maximize the sum of the two entropies. Entropy-based thresholding methods have the advantage of being able to handle images with uneven lighting, as they can adjust the threshold value to different illumination conditions. However, they can be computationally expensive, especially for large images. They can also be sensitive to the choice of the entropy measure, and different entropy measures may work better for different types of images than others.

2.4.3 Thresholding with Edge Detection

Edge detection is a technique to identify boundaries between objects in an image. Edge detection methods can also be used for image thresholding by identifying the edges of objects and then applying a threshold to the image to separate the foreground and background. One common method for edge detection is the Canny edge detection algorithm [39]. This works by computing the gradient magnitude of the image and then applying non-maximum suppression to thin the edges to single-pixel widths. The remaining edges are then thresholded to obtain a binary image. However, edge detection may not be effective for images where the objects do not have well-defined edges or boundaries. Edge detection-based thresholding methods can be sensitive to noise, as the noise can cause false edges to be detected. Pre-processing techniques, such as morphology filtering, may be necessary before performing edge detection. The HST image processing suite ACSTools [40] uses a combination of pre-processing techniques and Canny edge detection with Hough transformations [23] in the SatDet [25] tool for detecting satellite trails.

Chapter 3

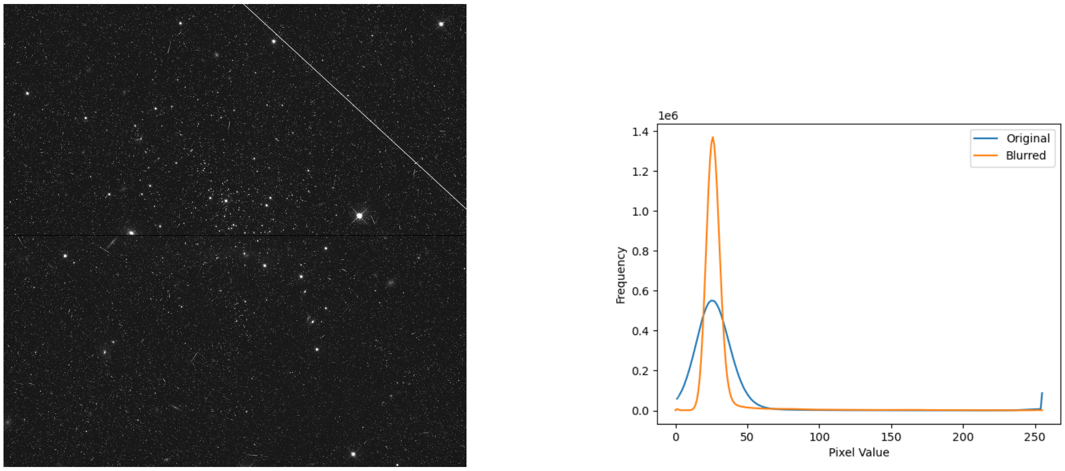
The Automatic Salience Thresholding Pipeline

In Chapter 2, the common functions of satellite trail detection pipelines were identified and how each stage contributes to the overall task. While successes have been achieved by Paillassa et al.[29] using synthetic images to train the convolutional neural networks MaxiTrack and MaxiMask, the literature showed that satellite trail image detection research overwhelmingly favoured image processing techniques. Because of a lack of heterogeneity in the available data, the most straightforward approach was developing a classic detection pipeline using image processing techniques to accomplish the detection task. The studies reviewed revealed that trail detection unfolds in sequential stages, and the effectiveness of each subsequent stage critically hinges on the outcomes of its preceding stages. This chapter explores utilizing the lessons learned from the reviewed literature to construct a pipeline that can perform efficient automated detections on various optical systems. A consolidated detection pipeline will be proposed, and each stage will be described in the order in which it executes.

The stages in the pipeline shown in Figure 2.1 have been grouped together into coloured blocks that are further described in Section 2.2. The groupings provide useful context for the combined efforts that the individual stages are working to achieve. The first three stages are grouped together to represent their combined effort toward noise suppression and background/foreground segmentation. This task is divided into several steps in the optical system proposed by Samadzadegan et al.[13] that include background and noise removal. The study uses a computational method to remove instrument artifacts that could improve the overall image quality. This technique utilizes an image acquired by a sensor with the same temperature and exposure time but without a signal to generate a dark frame [14]. The dark frame is then subtracted from the input to reduce image noise and improve signal fidelity. When considered in an automated application, the computational method becomes an appropriate methodology for image noise suppression.

Farzaneh et al.[18] eliminated the background estimation and removal steps entirely and used a Gaussian filter [41] to reduce image noise before detection. In contrast, Samadzadegan et al. use a statistical measure to identify and remove spike noise by eliminating pixels with intensity values greater than three times the standard deviation of a normal distribution [13]. The additional computation for calculating the input mean and standard deviation is required to reduce the number of false star detections in the star identification steps. A 2019 study [42] comparing star tracker performance shows that only as few as thirty false stars result in 25% fewer successful detections in state-of-the-art methods.

Danarianto et al.[27] use a median convolution kernel to reduce the impact of noise on detection. The authors replaced the background estimation and removal steps with a predefined global thresholding stage with a static threshold value. The pipeline relied heavily on reconstructive techniques before the detection stage, increasing computational overhead. To achieve a fast, flexible automatic detection pipeline that can be used on arbitrary optical equipment, an automatic global thresholding algorithm that is not influenced by noise and other interference should be used. The detection speed and accuracy can be used as important evaluation metrics to achieve detection outcomes in real time. To produce real-time performance, the entire detection process must produce a result in less time than the imaging exposure time.



(a) An image taken by the Hubble Space Telescope containing a satellite trail located in the upper right quadrant.

(b) The sparse bimodal image histogram for Figure 3.1(a) and the resulting histogram after applying a Gaussian function.

Figure 3.1: The effect of a Gaussian function being applied to an image with a class imbalanced sparse bimodal histogram.

3.1 Automatic Salience Thresholding

Automatic salience thresholding (AST) is a novel histogram-based algorithm optimized for analyzing sparse class-imbalanced bimodal images, including those commonly encountered in the analysis of night sky imagery. An example of an image with a bimodal histogram can be seen in Figure 3.1 Specifically, AST is designed to identify a global threshold value that is not biased by class imbalance and is permissive to dim foreground objects while still being resilient to noise. This is achieved by interpolating between two histogram bins by locating two salient features in the image histogram using pixel intensity probabilities. Like the Triangle method [34], AST selects the threshold value relative to the peak intensity. Unlike the Triangle method, AST does not assume an unimodal histogram, which makes it more flexible and versatile when considering the balance between sensitivity and selectivity for the subsequent detection task.

AST leverages a common feature found in many astronomical and night sky images.

The sparse bimodal class-imbalanced image histograms refer to a specific type of histogram distribution commonly encountered in outdoor nighttime images. In such histograms, the pixel intensity values typically exhibit two distinct peaks, hence the term "bimodal." These peaks represent two major groups of pixel intensities, often corresponding to the foreground (object of interest) and background regions of the image.

The term "sparse" indicates that these peaks contain significantly different pixel intensities, leading to a large gap between the two modes. The class imbalance implies that one class dominates the other regarding the number of pixels it contains. For instance, in the case of satellite trail detection, most pixels may belong to the background (sky) class, while a smaller number of pixels represent the satellite trails and stars (foreground).

Sparse bimodal class-imbalanced image histograms pose unique challenges in image segmentation and thresholding algorithms. Traditional thresholding techniques that assume balanced histograms may not perform optimally in these scenarios, as they may be biased towards the dominant class and fail to identify the minority class accurately. Consequently, there is a need for specialized and adaptive thresholding methods that can effectively handle such imbalances and accurately separate the foreground from the background.

The AST algorithm accepts a grayscale input and consists of five steps: pre-processing, histogram generation, peak finding, foreground detection, and interpolation. In the pre-processing step, a Gaussian blur is applied to the grayscale input image using a 3x3 kernel to remove high-frequency noise. The distribution of pixel intensities of the image histogram changes due to a narrowing of the background mode and a spreading of the foreground mode that can be seen in Figure 3.1. A histogram of normalized intensity values is then generated in the second step. The bin with the peak intensity value is identified as the first salient feature in the third step. The fourth step involves finding the foreground class, achieved by identifying the histogram intensity bin containing a low pixel probability. If no appropriate point is found, the last histogram bin is used as the second salient position. Finally, the fifth step calculates the threshold value by determining the bins between the two salient features and interpolating a new position sensitive and robust to image noise.

3.1.1 Preprocessing and Normalization

Applying the Gaussian function [43] to the grayscale input image before histogram generation is a common preprocessing step in image thresholding methods for reducing image noise. In addition to noise suppression, the 3x3 Gaussian kernel will cause the background mode to narrow and the foreground mode to spread, as seen in Figure 3.1(b). As a result, some of the histogram bins that didn't contain pixels will now have a small number associated with them. The Gaussian function removes small, high-frequency noise, resulting in a smoother, more continuous histogram. This improves the accuracy of the threshold determination, particularly for dim object detection and detecting satellite trails in night sky imagery where low contrast and high noise levels can make accurate detection challenging. The method for calculating the accuracy evaluation metric is explained in detail in Section 4.3.

A normalized histogram is used to represent the distribution of data in a way that accounts for differences in the total number of data points between different data sets. Normalizing the histogram allows the AST algorithm to utilize proportions rather than explicit intensities to determine the threshold value.

3.1.2 Peak Determination and Foreground Segmentation

The histogram's bin containing the largest proportion pixel value determines the first salient point. In equation 3.1, the *argmax* function is used to determine the maximum value in the normalized histogram, where h_i represents the frequency (or count) of pixels with intensity i in the image histogram. T is a normalization factor. It represents the total number of pixels in the image. Dividing h_i by T gives us the proportion of a pixel having the intensity i in the image. Applying the *argmax* to the largest i produces the *peak* bin value.

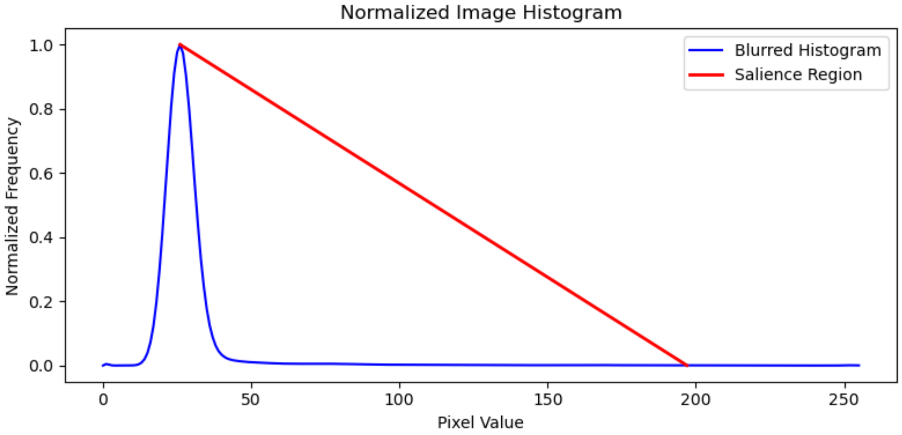


Figure 3.2: The first stage of the AST algorithm is determining the salience region with the maximum pixel probability value and a low probability pixel value as its endpoints.

$$peak = \arg\max_{i=0}^{255} \frac{h_i}{T} \tag{3.1}$$

$$term = \min\left\{ i \mid i > \arg\max_{i=0}^{255} \frac{h_i}{T}, h_i > 0, \frac{h_i}{T} < 0.001 \right\} \tag{3.2}$$

The terminating bin is found by iteratively checking the remaining bins for a non-zero, low-probability pixel value. The Gaussian function caused the foreground histogram pixels to spread into the neighbouring bins, which still contain enough pixels to be differentiable from the image noise in the salience region in Figure 3.2. Eighteen images from the NASA Streak Watcher dataset [9] were used to test the change of accuracy when the term value was varied. More information about the dataset can be found in Section 4.1.1. A starting value of 0.003 was found by comparing the proportion of pixels that moved to histogram bins that formally contained zero pixels before applying a Gaussian blur using the test images. Increasing this value causes the algorithm to become more restrictive and results in fewer detections while reducing it has the opposite effect. Figure 3.3 demonstrates the *term* value of 0.001 maximizes accuracy by comparing mean accuracy scores for different values of *term*. If no bin

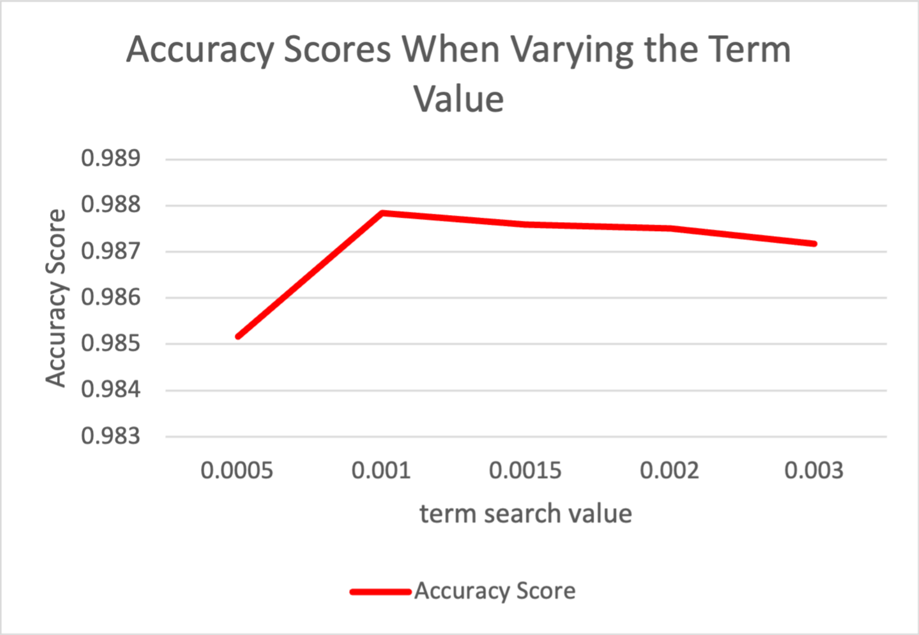


Figure 3.3: The detection accuracy scores from a basic detection pipeline were used to determine the optimal term value for the AST algorithm.

meets this criterion, the bin value 255 is selected as the second salient feature. This condition was included as a catch-all condition but was not triggered while testing the test images, further supporting the choice of the selected *term* value. The *term* value can be calculated using Equation 3.2 where the first value for i that is greater than the *peak*, has a pixel count h_i greater than 0 and has a normalized h_i value less than 0.001.

3.1.3 Threshold Interpolation

The interpolation step allows the algorithm to position the global threshold toward the foreground within the salience region. Equation 3.3 shows the interpolation step, where the final threshold value is determined using the bin positions of the identified salient histogram features *pk* and *term*. The number of bins between the two features is then interpolated by a value of 0.9 to select a threshold value from the sparse

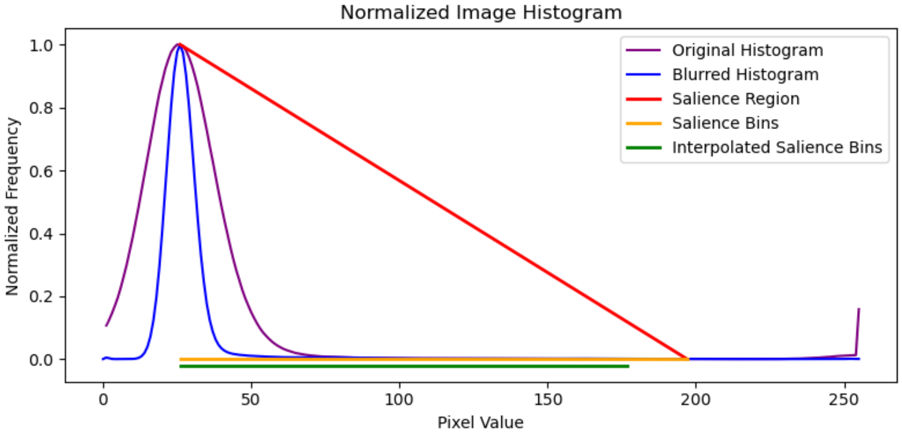


Figure 3.4: The salience region (red) determined the salience bin (yellow), which was interpolated (green) to obtain a foreground-biased global threshold value.

region. The value is rounded to the nearest integer and is represented by the right side endpoint in Figure 3.4

The same eighteen images used to find the optimal *term* value were again used to experiment to find the interpolation factor that will result in the best detection outcomes. This was done using the Mean Square Error (MSE) rather than the Accuracy metric as it provides a more sensitive dissimilarity measure. The method for computing the MSE evaluation metric is described in detail in Section 4.3. The interpolation factor values between 0.75 and 1.0 were calculated on the test set, and the result that produced the lowest MSE score was used to determine the final threshold value. This value chosen successfully biases the threshold toward the foreground and reduces false positives caused by image noise. The average MSE results for the group of images were plotted while varying the interpolation value, and the results can be seen in Figure 3.5.

The AST value is calculated using Equation 3.4, which is the sum of the *peak* and *interp* values and is used to segment the image by pixel intensity. A binary image thresholding that uses the calculated threshold value creates an AST binary mask used in subsequent pipeline stages.

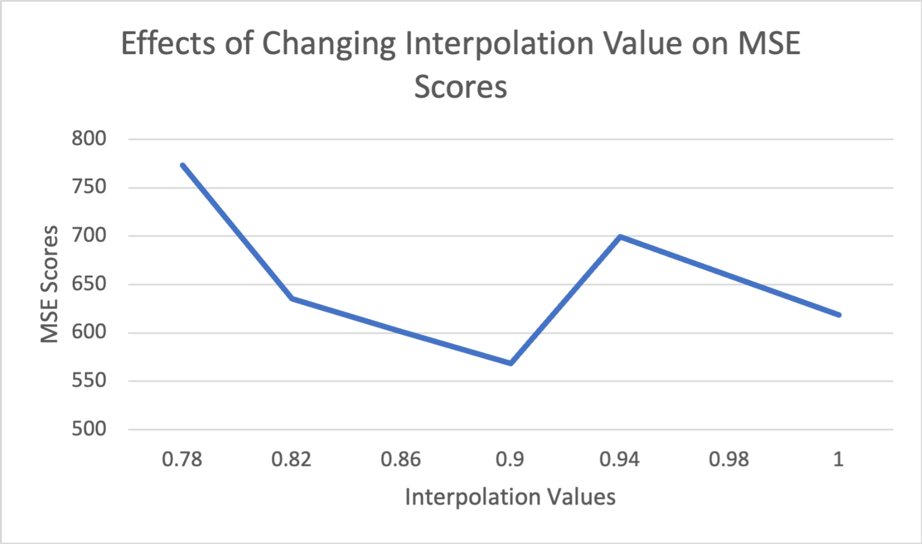


Figure 3.5: The detection MSE scores from a basic detection pipeline were used to determine the optimal interpolation value for the AST algorithm.

$$interp = \lfloor 0.9 \times \lceil term - peak \rceil \rceil \tag{3.3}$$

$$AST = peak + interp \tag{3.4}$$

3.2 Morphology Filtering

The morphology filtering stage is a critical component in the trail detection pipeline, aimed at reducing spurious detections that could lead to false positives. Its primary goal is to remove smaller pixel groups representing stars, noise, or other non-satellite trail objects from the binary mask generated in the previous stage. In Chapter 2, various methods for removing stars from the background were explored, including size thresholds and star identification. However, the proposed approach adopts a connectivity-based method using connected-component analysis (CCA) [44] with pixel neighbourhood labelling to group objects in the binary image generated by AST. The CCA examines each pixel in the image and labels the foreground pixels belonging to the same connected region with the same label. Pixels with a value of 0 (background) are ignored. As CCA processes each pixel, it identifies groups of connected foreground pixels and assigns them the same label. This process continues until all foreground pixels in the image have been labelled. The output of CCA is a labelled binary image, where each connected component is assigned a unique label. These labels can then be measured, and filtering logic can be applied based on a computed metric for each region.

Table 3.1: Binary Mask Similarity Scores Compared to Image Ground Truth

Threshold Type	Mean IoU	Mean Accuracy	Mean MSE
AST	0.226559	0.955596	1077.99

Table 3.1 contains the mean baseline detection scores of the AST pipeline using the test images described in this chapter without any morphology filtering. The AST masks were passed directly to a Hough line detector, forming a two-stage pipeline. The performance metrics were calculated by comparing each pixel from the pipeline output to the hand-annotated ground truth mask. The baseline scores show a high mean accuracy score likely biased by the substantial class imbalance between the foreground and background. To provide a more sensitive measure, the MSE and the Intersection over Union (IoU) metric were used to form a baseline for the pipeline performance. The IoU is a metric that measures the overlap between two regions, often used to assess the similarity between a detected object and its corresponding ground truth. The method for computing the IoU evaluation metric is detailed in Section 4.3.

3.2.1 Neighbourhood Connectivity

The connectivity of pixels determines which neighbouring pixels are considered part of the same component. Figure 3.6 shows the 4-connectivity or 8-connectivity schemes

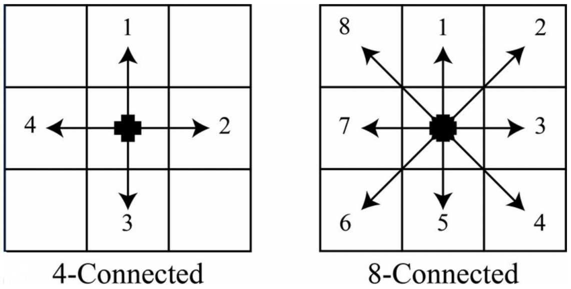


Figure 3.6: When using a connected component analysis, each pixel is labelled based on the pixels in its neighbourhood. The two region selection schemes that use a 4 and 8 Connected Component Analysis are shown above.

that are typically used. In a 4-connectivity, two pixels are considered connected if they share a common edge, while in an 8-connectivity, they are connected if they share a common edge or a corner. To determine which connectivity type would best suit the pipeline, a simple size filtering metric is used to test the effect that changing the connectivity has on detection outcomes. Using the SatDet [25] size filter settings as a starting point, a filter size of 75 pixels was selected as a baseline value. Table 3.2 shows the results of varying the neighbourhood connectivity values and applying a size filter to remove objects smaller than 75 pixels. By comparing the results to Table 3.1 it can be seen that the accuracy score alone is not sensitive enough to identify any performance differences between the pipeline configurations. The IoU and MSE provide a more sensitive measure of how well the output detections match the ground truth. This comparison shows that better detection outcomes are achievable when using a CCA with an 8-connectivity scheme.

Table 3.2: Evaluation Metrics and Time for Different Connection Types

Connection Type	Mean IoU	Mean Accuracy	Mean MSE	Mean Filter Time
8-Connected	0.279715	0.959356	848.2581	0.680974 sec
4-Connected	0.22876	0.959222	879.8409	0.763812 sec

The 8-Connected method is sensitive to diagonal pixels, which is ideal for satellite trail detection. This is reflected in the improvements in the mean IoU and MSE scores

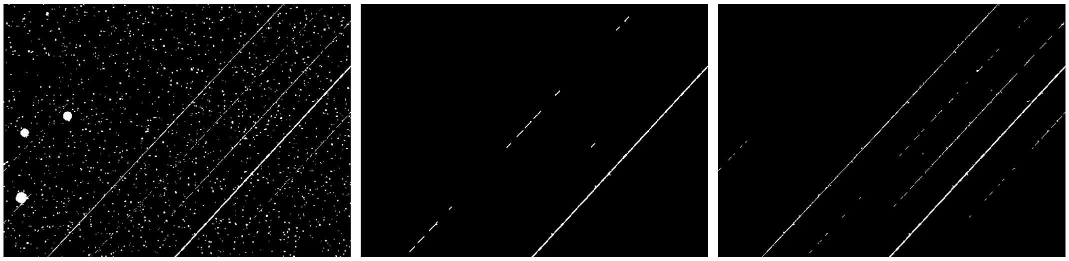


Figure 3.7: A qualitative comparison of morphology masks with AST masks shows improved size filtering using an 8-connected scheme. Left: The AST mask. Middle: 4-Connected CCA detection mask. Right: 8-Connected CCA detection mask.

compared to the baseline. The mean processing time for the size filter was calculated, and the 8-Connected CCA resulted in a quicker runtime than the 4-Connected variant. The larger processing time was attributed to more regions being detected during the 4-Connected type, resulting in additional object area calculations compared to the 8-Connected method.

The binary threshold/morphology mask comparison in Figure 3.7 shows that using a 4-connectivity metric for region extraction can inadvertently reduce the length of the detected trails in cases where the trails traverse in a diagonal orientation relative to the frame. A combination of fragmenting the satellite trail and a size filter causes this loss of fidelity, which occurs less when using the 8-connected labelling scheme. Based on these results, the 8-connected scheme was incorporated into the morphology filter of the proposed pipeline.

3.2.2 Size Filtering

Taking inspiration from Farzaneh et al.[18], the proposed pipeline also rejects star identification for satellite trail detection in favour of direct detection. For the morphology filter to be most effective, the regions that do not match the characteristics of a satellite trail are not added to the morphology mask. This can be achieved by directly targeting the shape characteristics of satellite trails and rejecting non-trail-like objects.

The ideal filtering strategy will be achieved by selecting the most suitable metric that preserves the full satellite trail in the morphology mask. The first approach would first filter out regions by size, rejecting small objects while permitting larger regions in the morphology mask. Choosing an appropriate size threshold for an input with an unknown field of view, resolution or image size would result in a loss of sensitivity in certain cases. Varying the minimum allowable area for each region shows that a value of 83 pixels delivers the best results in Table 3.3. The value represents the maximum and minimum values for the average IoU and MSE for the eighteen images in the test group.

Table 3.3: Area Threshold Pipeline Detection Performance Results

Area	Mean IoU	Mean Accuracy	Mean MSE	Mean Filter Time
50 pixels	0.278645	0.958859	876.4641	0.719936
70 pixels	0.279001	0.959162	859.5817	0.694171
75 pixels	0.279715	0.959356	848.2581	0.691148
80 pixels	0.27971	0.959474	840.2291	0.625231
82 pixels	0.28182	0.959507	839.1506	0.686727
83 pixels	0.282397	0.959534	837.3539	0.689233
84 pixels	0.281662	0.959532	838.1433	0.691334
85 pixels	0.281338	0.959528	838.4116	0.690035
90 pixels	0.281245	0.959591	835.8018	0.684056
100 pixels	0.280817	0.959768	824.9602	0.680968

The perimeter region property is also a useful metric for filtering small objects from the threshold masks. A comparison of average detection scores in Table 3.4 demonstrates the best perimeter threshold value that was identified for the test group images showed agreement with the value used in the SatDet [25] software and result in an 8% improvement over filtering by the area over the test group. The mean accuracy and MSE scores both improved using a perimeter-based filtering method.

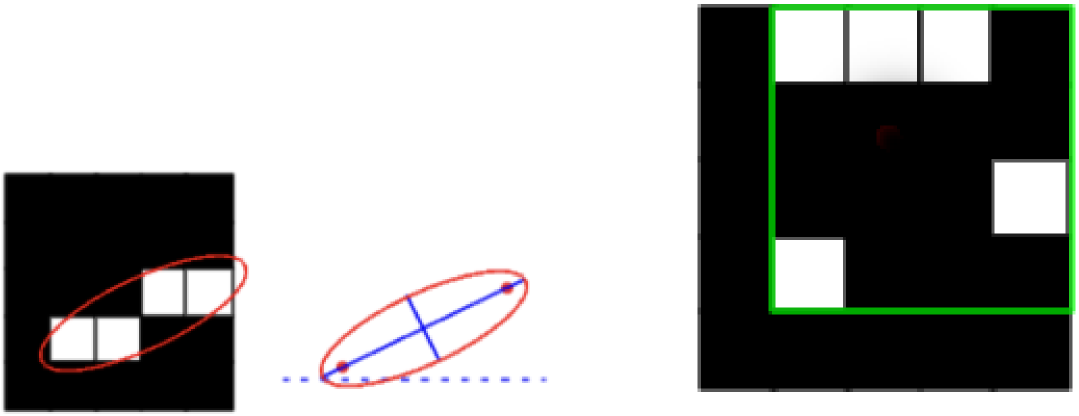
The analysis shows that a perimeter-based method can better remove stars and other small objects than an area-based size filter on the test data. This metric is still spatially linked to the input image size and must be optimized for the expected input size and field of view. For this reason, a shape-based method would likely produce better detection outcomes. The object measurement must first be normalized or described as a ratio to make the input size an arbitrary factor.

Table 3.4: Perimeter Threshold Pipeline Detection Performance Results

Perimeter (pixels)	Mean IoU	Mean Accuracy	Mean MSE	Mean Filter Time
65	0.290341	0.96057	766.4925	0.671831
70	0.290146	0.96063	762.1503	0.676214
75	0.292541	0.960728	756.8078	0.688515
80	0.290191	0.960758	757.8619	0.677749
82	0.291131	0.960786	756.3889	0.686727
83	0.290591	0.960788	755.3749	0.682464
84	0.290359	0.960776	756.5375	0.677319
85	0.291946	0.960783	755.9893	0.6819
86	0.291946	0.960783	755.9893	0.680468
87	0.29134	0.960779	756.6499	0.688297
90	0.290738	0.960843	751.6883	0.670056
95	0.290307	0.960868	750.7544	0.671594

3.2.3 Shape Filtering

The morphology filter employed in the proposed satellite trail detection pipeline takes advantage of the eccentricity and extent properties derived from the CCA of the binary image produced by the AST algorithm. Instead of relying on specific hard-coded size thresholds, the filter calculates the eccentricity and/or extent of each labelled region obtained from CCA, offering a relative measure of elongation. The eccentricity of a region is computed by fitting an ellipse with the same second moments as the shape to the region and then determining the distance ratio between the ellipse's two foci to the length of its major axis. This yields a value between 0 and 1, where an eccentricity of 1 represents an infinitely elongated shape, while values closer to 0 indicate more circular or compact shapes. Figure 3.8a shows how the eccentricity property provides



(a) The shape is fitted with an ellipse and the major and minor axis measured to produce a measure of elongation.

(b) The ratio of the area of the connected region and its bounding box are used to produce a measure of compactness called the extent.

Figure 3.8: A visualization of the extent and eccentricity shape metrics.

a measure of elongation useful for detecting satellite trails.

The extent property is calculated by dividing the region's area by the area of its bounding box. Figure 3.8b shows a visual example of how the extent property provides a measure of compactness. The property calculates the ratio of how much of the bounding box is occupied by the region. A higher extent value indicates that the region occupies a larger portion of its bounding box, indicating a more compact and filled shape, while a lower extent value suggests a more elongated or sparse shape. This property is useful for filtering hot columns and pixels caused by cosmic rays, which can often produce false negatives in other detectors [25].

By utilizing the eccentricity filter, regions with high eccentricity values (closer to 1) are considered, indicating elongated and linear structures that align with the characteristics of satellite trails. As a result, smaller detected areas, often representing stars and noise, are accurately removed from further consideration during the detection phase. Table 3.5 demonstrates the effectiveness of this approach. When focusing solely on elongated regions, the detection pipeline eliminates spurious detections, significantly reducing the false positives caused by irrelevant compact objects. This shape-based filtering ensures that the pipeline prioritizes regions with trail-like characteristics, enhancing the accuracy and reliability of satellite trail detection while efficiently discarding non-relevant regions. By avoiding rigid size thresholds and instead capitalizing on eccentricity as a shape-based metric, the morphology filter provides a more robust and adaptable solution, making it well-suited for satellite trail detection in diverse and challenging image scenarios.

Table 3.5: Eccentricity Threshold Pipeline Detection Performance Results

Eccentricity	Mean IoU	Mean Accuracy	Mean MSE	Mean Filter Time
0.90	0.3161	0.962044	644.6783	1.314329
0.95	0.331936	0.962964	583.2133	1.248986
0.97	0.333166	0.963278	566.378	1.230459
0.98	0.340622	0.963547	547.8724	1.198448
0.99	0.323458	0.963602	552.2453	1.172987

The eccentricity property produced the lowest average MSE score and the highest IoU scores when comparing the results in Tables 3.3, 3.4 and 3.5. Figure 3.9 shows that many small objects populate the morphology masks. Although the Mean IoU and MSE scores improved while using the eccentricity shape metric, the mean filter processing time increased by nearly 100%. Ideally, the proposed pipeline should improve detection outcomes without substantially increasing processing time, and perhaps the extent property could produce better results.

Using the extent property to filter the connected regions produced performance on par with the perimeter metric results in Table 3.4. The extent results in Table 3.6 show that the property produced a less accurate outcome on average but achieved similar MSE scores to eccentricity. Figure 3.10 shows that while the extent property achieved lower scores than the eccentricity property, the extent property removed more small objects, which resulted in a faster processing time. The mean filter time was not nearly as large as the mean filter time achieved by the eccentricity property.

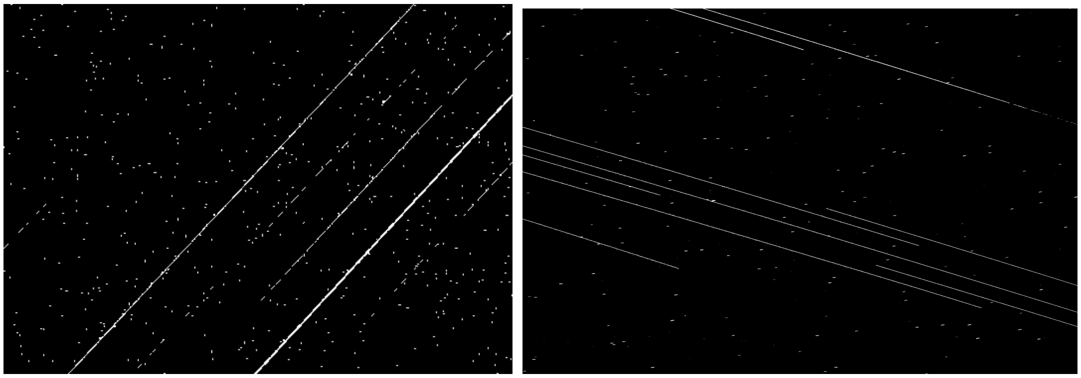


Figure 3.9: Morphology filter masks using the eccentricity value of 0.98 as a shape threshold value.

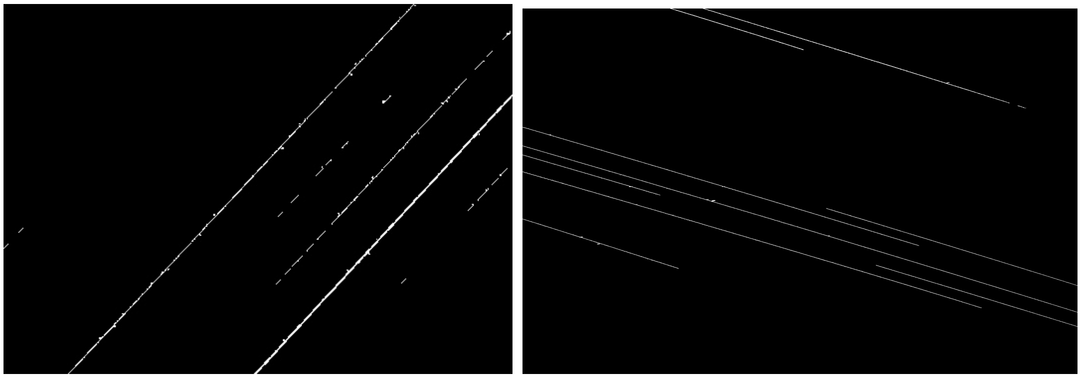


Figure 3.10: Morphology filter masks generated with the optimal extent value of 0.24.

This suggests that perhaps no single metric will effectively meet the performance criteria being sought. Combining these two properties in a single morphology filter was explored to reduce the processing time of the eccentricity. When combining the eccentricity and extent metrics, the extent value needed to be increased to produce productive results and resulted in improved mean IoU and MSE scores, which can be seen in Table 3.7.

The speed boost realized when extent is paired with eccentricity is due to the eccentricity calculation being performed less, as the extent metric overrides the conditional check. This is accomplished by requiring that each detected region conforms to the

Table 3.6: Extent Threshold Pipeline Detection Performance Results

Extent	Mean IoU	Mean Accuracy	Mean MSE	Mean Filter Time
0.15	0.2636	0.838141	608.6356	0.668582
0.2	0.302995	0.900778	587.0016	0.67842
0.22	0.304174	0.900789	583.144	0.679256
0.23	0.304027	0.900782	582.7165	0.677467
0.24	0.306824	0.900864	574.3378	0.679609
0.25	0.305918	0.900807	578.5485	0.681388
0.26	0.308085	0.900809	576.5182	0.675952
0.3	0.303638	0.900544	591.7463	0.692615
0.4	0.301882	0.900459	591.3907	0.698437
0.45	0.298201	0.900243	606.2672	0.723822
0.47	0.297657	0.962623	612.3187	0.734399
0.5	0.294702	0.962302	633.8768	0.796467
0.6	0.272593	0.959041	843.4391	0.844511

minimum extent and the eccentricity values using a logical AND operator. All regions that are not compact and sufficiently elongated are then added to the morphology mask. The inclusion of extent filtering reduces the false positive detections of hot pixels and diffraction spikes in the final detection stage. The filter became much too restrictive when the extent metric threshold was set below 0.67 as detection failed to locate any satellite trails in one of the test images. This value produced the best mean IoU, Accuracy and MSE scores on the test group of eighteen images in quantitative and qualitative comparisons.

Figure 3.11 shows the morphology masks for the combined eccentricity/extent shape filter. When comparing the qualitative results, it can be seen that the combination of eccentricity and extent filtering improves the detection scores and reduces spurious detections. It is not, however, required that the morphology filter remove all small objects as the Probabilistic Hough transform stage can tolerate some image clutter. Both the qualitative and quantitative results support the inclusion of a combined

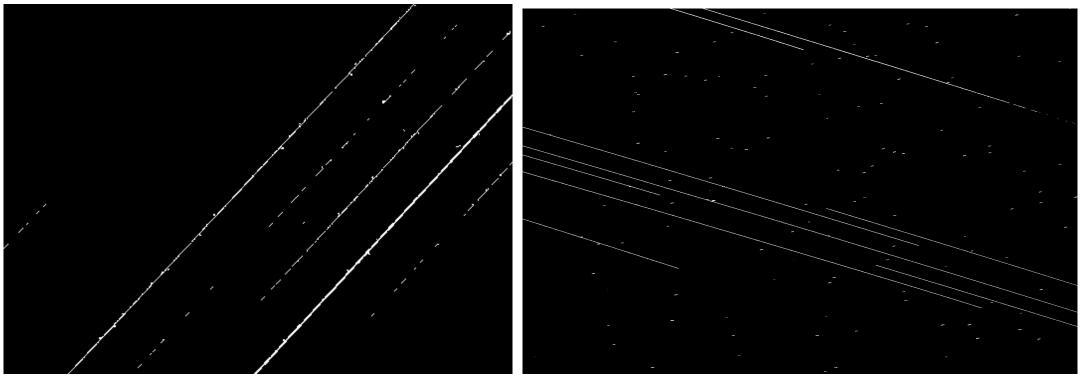


Figure 3.11: Morphology filter masks generated with the optimal eccentricity/extent value of 0.67.

shape filter in the morphology stage of the proposed pipeline.

Table 3.7: Varying Extents with Eccentricity Threshold Pipeline Detection Performance Results

Eccentricity	Mean Extent	Mean IoU	Mean Accuracy	Mean MSE	Mean Time
0.98	0.24	0.306013	0.901066	566.1102	0.689602
0.98	0.5	0.315012	0.901154	546.9486	0.711089
0.98	0.6	0.317101	0.901176	541.1558	0.723023
0.98	0.65	0.315036	0.901165	542.087	0.724902
0.98	0.67	0.353648	0.963623	543.6464	0.727304
0.98	0.68	0.353648	0.963623	543.6464	0.733846
0.98	0.69	0.35341	0.963616	544.1315	0.727129
0.98	0.7	0.353416	0.963618	543.9904	0.732558
0.98	0.75	0.353405	0.963611	544.3667	0.742882
0.98	0.8	0.353368	0.963604	544.7579	0.73615

3.3 Probabilistic Hough Transformation Line Detection

The final stage in our lightweight pipeline is a line detection algorithm that constructs the output detection mask. In Section 2.2.4, it was demonstrated that the Hough transformation method for line detection is widely used to locate satellite trails in astronomical imaging. The method is flexible and can detect lines with gaps and partial occlusion accurately and quickly. The Probabilistic Hough transform method was selected because it directly reveals the feature's location on the image plane. In contrast, other methods like the Radon Transform and the regular Hough transform yield linear features in (r, θ) coordinates or line space. While this representation describes a line on the image plane, it lacks information about the feature's position along that line. In cases where a short feature is detected, an additional iteration is required to extract the image plane coordinates from the (r, θ) representation. Rather than exhaustively searching the parameter space, the probabilistic Hough transform uses random sampling of edge points and iteratively fitting line segments. This significantly reduces the computation time and makes it more efficient, particularly for real-time applications.

The Probabilistic Hough transform algorithm accepts parameters to specify minimum line length and maximum line gap thresholds for line detection. These values were set to be proportional to the input image by passing the value of the longest dimension. Initially, the shortest satellite trail the proposed pipeline could detect is 10%, and the line gap threshold produced the most efficient detections when set to 1%. These values were set as the baseline and remained unchanged throughout the testing conducted in Chapter 3. Experimentation results shown in Figure 3.12 reveal that manually changing the values did not produce any further performance benefits, so the values were returned to their original values and left unchanged. Any minor variation in the results was attributed to the computational optimizations from the probabilistic Hough transform.

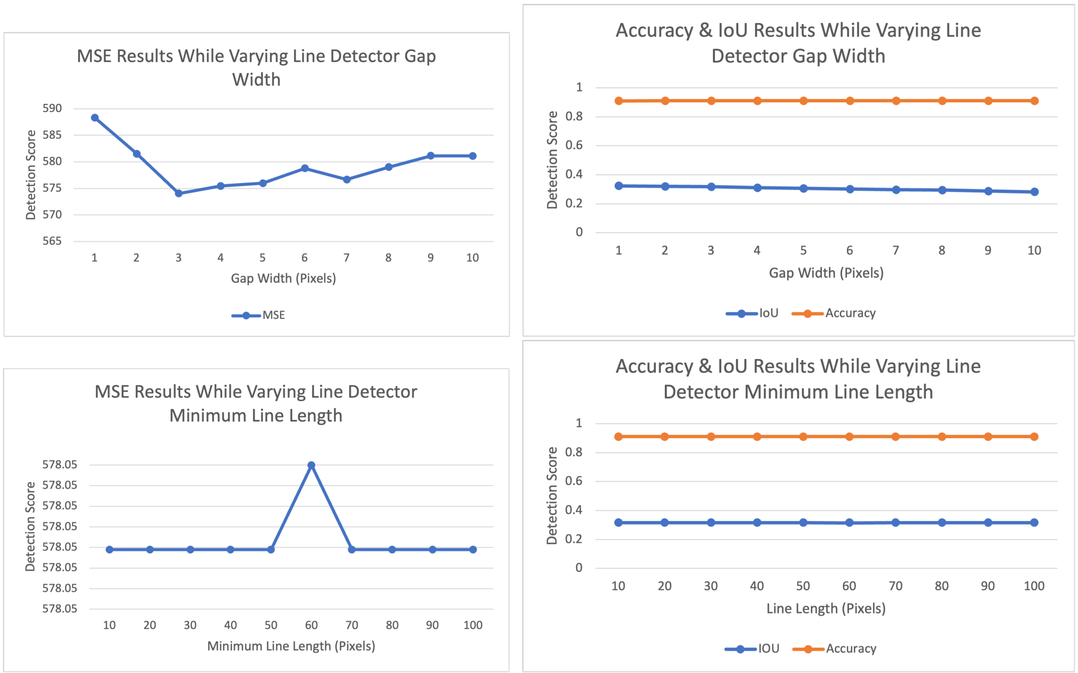


Figure 3.12: The maximum gap width and minimum line length parameters for the probabilistic Hough detector were varied to observe the impact on the IoU, Accuracy & MSE scores. No meaningful impact resulted from varying the line detector parameters in the given ranges because the satellite trails in the test images exceeded the tested thresholds.

Chapter 4

Experiments

Generally, satellite detection pipelines are tuned to suit the specific optical equipment and imaging sensors responsible for generating the input images. This customization poses a challenge when directly comparing different satellite trail detection methods. Poor data heterogeneity adds to the complexity of such comparisons as the purpose-built pipelines are configured to expect an input with multiple known parameters. Consequently, a limited number of public datasets containing images of satellite trails were available for this study.

To overcome this obstacle, an experiment was devised to assess the effectiveness of the proposed automatic thresholding algorithm and detection pipeline in segmenting satellite trails in night sky imagery. The pipeline was modified to allow for different thresholding methods to be used in the preprocessing stage. The morphological filter and detection stage parameters were fixed, and only the automatic thresholding sub-stage could vary. The pipeline structure shown in Figure 4.1 enables the generated output to be scored against the ground truth using the metrics listed in the scoring stage.

The choice of an appropriate threshold is critical as it directly affects the accuracy of the detection results. However, image noise can pose challenges in determining an optimal threshold. The presence of noise can lead to inconsistencies in pixel intensity,

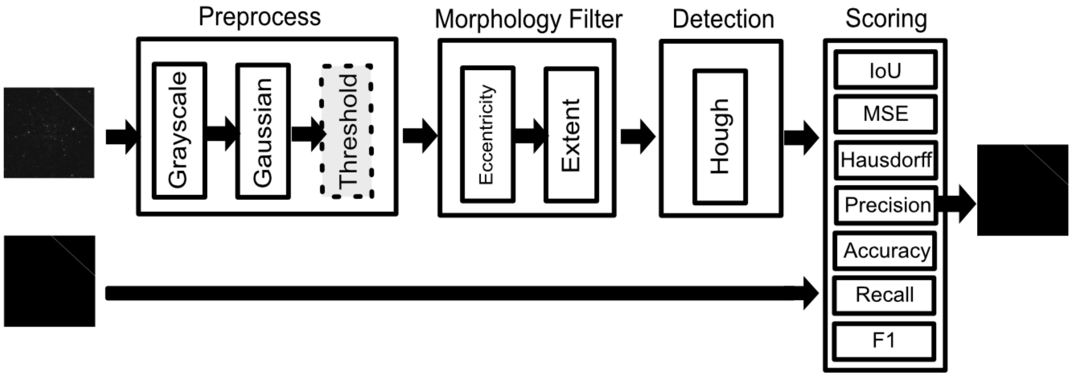


Figure 4.1: The 3 stage detection pipeline with the evaluation metrics stage that was used to evaluate detection performance against a ground truth. The threshold sub-stage allows for the automatic thresholding algorithm to vary, while the other stages/sub-stages are fixed so that the automatic thresholding algorithms can be compared using the generated output detection masks

making it difficult to find a threshold that effectively separates the foreground and background regions. This is seen in Figure 1.3, where uneven lighting and sensor noise caused data loss during thresholding. To understand and quantify if AST can improve detection outcomes, it is necessary to perform a comparison. The Canny edge detection [39] method has been used in the SatDet [25] software for image binarization. The medical imaging community employs various automatic binary image algorithms to perform image segmentation tasks. The Fiji [45] imaging software suite was used to implement many of the selected methods used in this study (Yen [46], Renyi Entropy

[47], MaxEntropy [38], Shanbhag [48], Moments [35], Triangle [34], Otsu [33], Huang [49], Intermodes [50], Li [51], Mean [36]). By substituting these automatic thresholding methods into the detection pipeline and comparing the generated masks to the ground truth, we can evaluate the performance of each thresholding method.

4.1 Evaluation Data

This section will discuss the datasets used to evaluate the satellite trail detection pipeline. Traditionally, optical satellite trail detection systems have commonly utilized astronomy-grade FITS (Flexible Image Transport System) files. However, for practical reasons and broader accessibility of data, the selected datasets in this study deviate from this standard and comprise images in the widely used JPG format. Nevertheless, it is important to highlight that the pipeline can be extended to incorporate FITS files using the Astropy library [52], which provides tools for reading, writing, and manipulating astronomical data in FITS format. By leveraging the Astropy library, researchers can integrate FITS images into the pipeline alongside JPG images using Astropy to extract the image data for trail detection. The consideration of public satellite trail data availability drove the decision to opt for JPG files for this specific evaluation. This approach allows for the utilization of publicly available JPG datasets, further enhancing the accessibility and reproducibility of the study's results.

4.1.1 NASA Streak Watcher Dataset

This study utilized a dataset from the NASA Satellite Streak Watcher citizen science project [9], providing 233 ground-based images of the night sky containing satellite streaks. The dataset consists of images taken by participants from the Internet, showcasing satellite streaks across various locations and timeframes. The project focuses on photographically tracking satellite streaks across the night sky, providing valuable insights into the extent of sky pollution caused by satellites. This dataset is part of a long-term project aimed at monitoring the population growth of satellites and studying their impact on ground-based astronomy.

The images were captured under different lighting conditions, with varying exposures, cameras, and settings. Amateur skywatchers captured the images, so not all images were usable for this study. Some of the images in the dataset contained/used convex camera lenses, urban skylines and human subjects. Any duplicate images were removed and only images that were free of the aforementioned conditions were used in this study. As a result of these constraints, 57 images were selected and hand-annotated ground-truth masks were created by the author to test the performance of the proposed algorithm.

4.1.2 Asteroid Hunters Dataset

The Asteroid Hunters dataset [6] is the largest collection of publicly available images of satellite trails to date. The dataset contains 114,607 images taken by the Hubble Space Telescope (HST) over 19 years and was classified using online crowdsourcing to provide labels that describe the image contents. These images have been processed to combine individual exposures, resulting in composite images without geometric corrections or filling the gap between detectors. The dataset provides classifications for 3,072 images containing satellite trails, carefully reviewed and validated by the authors. It includes observation IDs, instrument information, exposure details, celestial coordinates, and image URLs. The satellite classifications were conducted through a combination of citizen science contributions and machine learning.

The entire dataset was processed using only the AST detection pipeline and was visually sorted into two groups afterwards. The first group (1276 images) contained detection masks containing isolated satellite trails with little to no star/noise residue. The second group (1796) contained the remaining images comprised of noisy/partial/missed detections.

4.2 Data Treatment

The satellite trails in the datasets were hand-annotated to establish a ground truth for evaluating and comparing the performance of the AST method with other thresholding techniques. This process involved manually identifying and marking the satellite trails within each image. An annotation tool, such as a polygon tool, was utilized to outline the paths of the satellite trails accurately. This allowed for precise delineation of the regions corresponding to the trails in the images.

During the hand annotation process, several factors were taken into consideration. One important aspect was the identification of the satellite trail itself. Each image was carefully examined to locate the distinct streak or path created by the satellite's movement across the night sky. This involved differentiating the trail from other image features, such as stars, noise, or asteroids. Additionally, the presence of diffuse lighting from some satellite trails posed a challenge during the annotation process. In these cases, it was not always clear where the satellite trail's edge was. It was crucial to capture the entire satellite trail, so masks were made larger to accommodate both the edge and the residual glow. An example of this can be seen in Figure 4.2, where space was left to either side of the visible trail. By carefully examining the characteristics of the satellite trail and considering the overall lighting conditions, annotators produced a reliable ground truth dataset that accurately captured the presence and location of the satellite trails. Increasing the width of the annotated area to include the diffused light caused by the trail will also help future machine-learning models learn the features necessary for successful detection.

The hand annotation of satellite trails in the datasets serves as a crucial reference for evaluating the performance of the AST method and comparing it with alternative thresholding methods. This ground truth data provides a reliable basis for assessing the accuracy and effectiveness of the thresholding algorithms in detecting and segmenting satellite trails. Selecting and annotating the 57 images from the NASA Streak Watcher dataset required much effort, and the Asteroid Hunters dataset would require considerably more. A citizen science data annotation project [11] was created to help reduce the burden of labelling the Asteroid Hunters dataset. Participants are asked to download polygon labelling software and a dataset segment and are provided instructions. Their task is to carefully examine each image and annotate the satellite trails they encounter. Each participant annotated 10 datapacks, or approximately

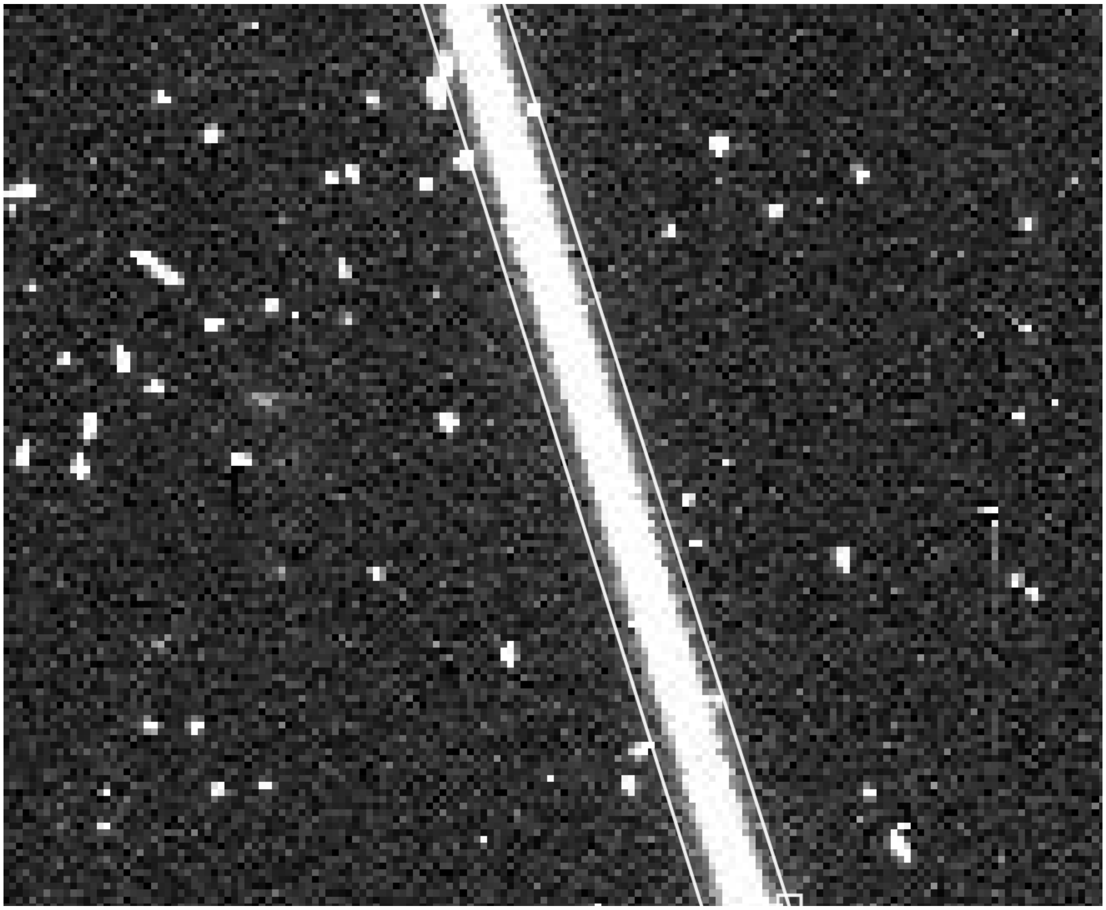


Figure 4.2: A closeup view of a satellite trail from the Asteroid Hunters Dataset that has been annotated as a part of citizen science labelling project that produced ground truth data for this study.

400 images. Once the annotation task was completed, participants uploaded their results, creating a comprehensive annotated image dataset. Engaging citizen scientists in this collaborative effort expanded our study to include a quantitative analysis of space-borne imagery for AST. The data annotation process for our citizen science project involved four main steps.

Step 1: Downloading the Data Pack

The participants downloaded the "datapackDownloader" script, which provided access to one of the 43 Hubble Space Telescope image data packs. Each data pack contained a unique set of images that required annotations. After obtaining the script, volunteers extracted the contents of the downloaded zip file to a folder they chose on their computer. To ensure efficient collaboration and avoid duplicate efforts, volunteers were encouraged to check the comments section below the download link to see if the data pack they intended to work on had already been annotated by other participants.

Step 2: Running the Annotation Script

Once the data pack was successfully extracted, volunteers extracted the files to a location on their computer. To initiate the data annotation procedure, they opened the terminal or command line within the script folder and executed the command:

```
python3 getDB.py sets/hubbleSatellites_X.csv
```

Here, X represented a number between 1 and 43, corresponding to the specific data pack they were annotating. It is important to note that the data pack size was approximately 250 MB and contained approximately 43 images.

Step 3: Utilizing Labelme for Data Annotation

To facilitate the data annotation process, volunteers were recommended to download the latest release of “labelme” [53], a user-friendly image annotation tool which can be seen in Figure 4.3. Once installed, volunteers launched the application and navigated to the ”HSTData” folder, which was located within the script directory. This folder contained the Hubble Space Telescope images that needed annotations. Using “labelme”, volunteers could easily mark regions of interest directly on the HST images to annotate satellite trails. Upon completing the annotation for a given image, volunteers were encouraged to save their annotations to the ”annotations” folder inside the ”HSTData” directory.

Step 4: Collaborative Comments and Data Submission

To foster collaboration and coordination among volunteers, a comments section was available on the website. Volunteers were encouraged to leave comments indicating

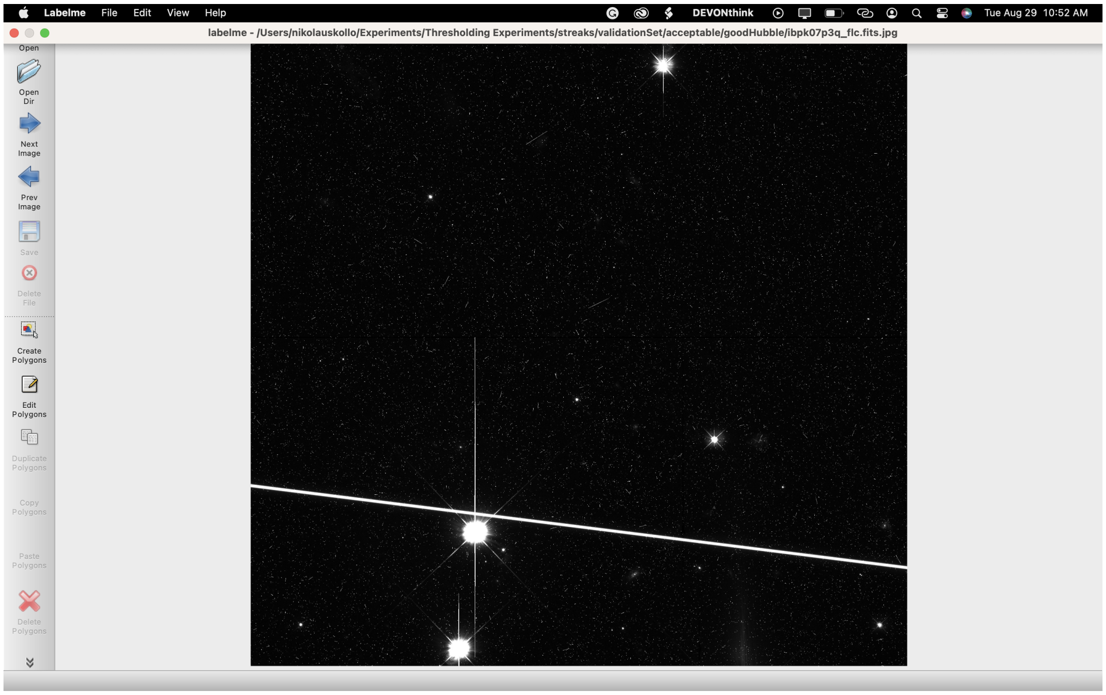


Figure 4.3: The labelMe image annotation software being used to create pixel-level annotations of satellite trails in HST images.

which data pack they were working on and sharing progress updates. By doing so, participants could easily identify data packs already annotated by others and avoid duplication. Once the data annotation task for a specific data pack was completed, volunteers were requested to zip the ”annotations” folder containing the annotated images and submit it through the designated submission process.

Through these four steps, the citizen science website enabled volunteers to contribute to the data annotation process effectively, ensuring the successful implementation of the satellite trail detection pipeline.

4.3 Evaluation Metrics

By systematically varying the automatic thresholding algorithm of the pipeline, the experiment aimed to determine the extent to which the proposed automatic thresholding algorithm enhances the pipeline's performance. This variation in the thresholding stage allowed for a comprehensive assessment of the algorithm's contribution to accurately segmenting satellite trails in the captured night sky imagery. Through this approach, the experiment aimed to provide insights into the efficacy of the automatic thresholding algorithm within the context of the specific detection pipeline, shedding light on its ability to improve the overall performance of satellite trail segmentation.

This section will describe the process of comparing the proposed method to other automatic global thresholding algorithms. Descriptions of the satellite trail datasets and details about the data annotation methods used for the study will also be provided. Information about the metrics used to compare the output detection masks and the comparison results will also be discussed in detail.

To quantitatively assess the performance of our proposed method, we compared the outputs of the detection pipeline to the ground truth for our data. In addition to the following metrics, we use the processing time, precision, and recall to compare AST to the other automatic binary thresholding methods.

Accuracy: The accuracy measures the proportion of pixels correctly classified as foreground or background in both the ground truth and test images. It is calculated using Equation 4.1 where TP (True Positive) is the number of pixels that are correctly classified as foreground in both the ground truth and the test image, TN (True Negative) is the number of pixels that are correctly classified as background in both images, FP (False Positive) is the number of pixels that are classified as foreground in the test image but not in the ground truth, and FN (False Negative) is the number of pixels that are classified as background in the test image but not in the ground truth. In scenarios where the number of background pixels greatly outweighs the number of object pixels, a high accuracy score can be achieved by simply classifying all pixels as background. This is because the classifier will be correct for the majority class (background) but fail to identify the minority class (object of interest) correctly. As a result, the accuracy score may appear deceptively high, even though the pipeline fails to detect the satellite trails accurately.

The accuracy score, by itself, cannot reliably assess the performance of the pipeline when class imbalance is present. Instead, it is crucial to consider additional evaluation metrics that take into account false positives and false negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Area Under the Receiver Operating Characteristic (ROC-AUC): The ROC-AUC is a metric commonly used to evaluate the performance of binary classifiers, including those used for comparing binary masks. The ROC curve is a graphical representation of the classifier's performance at various threshold levels, plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The ROC-AUC quantifies the classifier's ability to distinguish between positive and negative instances. It ranges from 0 to 1, where a higher AUC value indicates better discrimination and performance. An AUC of 0.5 signifies random guessing, while an AUC closer to 1 indicates better classifier performance. The ROC-AUC is particularly useful when dealing with class imbalance, as it takes into account true positive rates and false positive rates, providing a more reliable evaluation of the classifier's ability to handle unbalanced classes.

The ROC-AUC can be calculated using various methods, and one commonly used approach is the trapezoidal rule to approximate the area under the ROC curve. Given the True Positive Rate (TPR) and False Positive Rate (FPR) at different threshold levels, the ROC-AUC was computed using Equation 4.2 where the TPR(FPR) represents the True Positive Rate at a specific False Positive Rate, and the integral approximates the area under the ROC curve.

$$ROCAUC = \int_0^1 TPR(FPR), dFPR \tag{4.2}$$

Mean Square Error (MSE): This measures the average squared difference between the pixel values in the ground truth image and the test image. It is calculated using 4.3 where I is the ground truth image, J is the test image, N is the total number of pixels, and I_i and J_i are the pixel values at location i in the two images. By squaring the pixel-wise differences and taking the average, MSE amplifies the effect of larger differences between the two images. This is particularly useful when dealing with binary masks, as it emphasizes areas of significant disagreement, making it sensitive to both false positives and false negatives.

A lower MSE value indicates a closer match between the ground truth and the test image, implying a higher level of accuracy in detecting the satellite trails. Conversely, a higher MSE value suggests greater dissimilarity and lower accuracy in the test image's identification of satellite trails.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (I_i - J_i)^2 \tag{4.3}$$

Intersection over Union (IoU): The Intersection over Union (IoU), also known as the Jaccard Index, is a metric that quantifies the similarity between the foreground regions in the ground truth image and the test image. It measures the overlapping area between the two regions relative to their total combined area.

The IoU is calculated using Equation 4.4 where TP (True Positive) represents the number of pixels that are correctly classified as foreground in both the ground truth and the test image. FP (False Positive) is the number of pixels that are classified as foreground in the test image but not in the ground truth, and FN (False Negative) is the number of pixels that are classified as background in the test image but not in the ground truth.

IoU values range from 0 to 1, where 0 indicates no overlap (complete dissimilarity) between the two regions, and 1 represents a perfect match (complete similarity). Higher IoU scores indicate a better alignment between the predicted and ground truth regions, reflecting a higher level of accuracy in detecting the satellite trails. Conversely, lower IoU values suggest less overlap and lower accuracy in the test image's identification of satellite trails.

$$IoU = \frac{TP}{TP + FP + FN} \tag{4.4}$$

Hausdorff Distance: The Hausdorff Distance is a metric that quantifies the maximum distance between the closest points of two sets. In the context of image segmentation, it measures the boundary dissimilarity between the ground truth boundary and the predicted mask's boundary. The calculation of Hausdorff Distance is based on Equation 4.5, where A and B represent two sets, and $d(a, b)$ denotes the Euclidean distance between points a and b. The Hausdorff Distance considers the greatest lower bound (infimum) and the least upper bound (supremum) to ensure a robust measurement that accounts for the extreme distances between the two sets.

In the context of image segmentation evaluation, a lower Hausdorff Distance indicates a closer alignment between the boundaries of the predicted mask and the ground truth. A smaller Hausdorff Distance value signifies a higher level of accuracy in identifying the satellite trails' boundaries. The Hausdorff Distance metric is particularly useful for evaluating segmentation models when dealing with binary masks and unbalanced classes. It considers the maximum boundary discrepancy between the ground truth and predicted masks, providing a comprehensive evaluation of how well the pipeline captures the satellite trails' intricate boundaries. It is a powerful tool for assessing segmentation performance, especially when precise boundary localization is essential for accurate analysis and detection tasks.

$$H(A, B) = \max\left(\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b)\right) \tag{4.5}$$

Precision: Precision is a metric that assesses the accuracy of positive predictions made by the model. It quantifies the proportion of true positive predictions out of all positive predictions, indicating how well the model correctly identifies positive instances. Precision is calculated using Equation 4.6, where TP represents the number of pixels correctly classified as foreground in both the ground truth and the test image, and FP represents the number of pixels classified as foreground in the test image but not in the ground truth.

In the context of image segmentation evaluation with binary masks, a higher Precision score indicates a higher accuracy in identifying the satellite trails or positive instances. It reflects the pipeline's ability to minimize false positives, which are instances falsely classified as positive (foreground) but are actually negative (background). In tasks where false positives have significant consequences or implications, a high Precision score is important to ensure that the identified positive instances are more likely to be true positives. By considering the correct positive predictions relative to the total predicted positive instances, Precision provides a focused assessment of the pipeline's ability to detect the satellite trails accurately.

$$Precision = \frac{TP}{TP + FP} \tag{4.6}$$

Recall: Recall, also known as sensitivity or true positive rate, measures the ability of the model to identify all positive instances correctly. It quantifies the proportion of true positive predictions from all actual positive instances in the ground truth. The recall is calculated using Equation 4.7 where TP is the number of pixels correctly classified as foreground in both the ground truth and the test image, and FN is the number of pixels classified as background in the test image but not in the ground truth. The recall is valuable in scenarios where missing positive instances is critical, as it provides an indication of the model's ability to capture all relevant positive examples.

$$Recall = \frac{TP}{TP + FN} \tag{4.7}$$

F1 Score: The F1 score measures the accuracy of the pipeline that balances both precision and recall, providing a single metric that combines the two. The F1 score considers false positives and false negatives and is calculated using Equation 4.8. The F1 score ranges from 0 to 1, where a higher value indicates better performance in terms of both precision and recall. It is commonly used as a single evaluation metric when precision and recall are equally important or when there is a need to balance the trade-off between the two. The F1 score is particularly useful when a class imbalance exists in the dataset, as it considers both false positives and false negatives in its calculation.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.8}$$

4.4 Experiment Results

A total of thirteen existing automatic thresholding algorithms were compared using the NASA Streak Watcher data. The quantitative results can be found in Tables 4.1, 4.2, and 4.3. The scores for each metric are shown in descending order, and our proposed method is highlighted for convenience. In addition to the previously described metrics, box plots and other charts have been included to provide the reader with additional context for the calculated scores. The bar chart in Figure 4.7a presents the total average pipeline processing time for each thresholding method. The pie chart in Figure 4.7 visualizes the detected satellite trails based on a qualitative review. Figure 4.11 presents several HST images and the detection masks produced by the proposed pipeline with AST. The odd columns contain original HST images, while the even columns contain the corresponding detection mask.

4.4.1 Entropy-Based Results

Entropy-based thresholding methods, including Yen [46], Renyi Entropy [47], Max-Entropy [38], and Shanbhag [48], leverage entropy measures to determine the optimal threshold.

Yen's Thresholding: Yen's method maximizes the sum of the entropies of the foreground and background regions, offering a balanced threshold selection. However, it demonstrated lower accuracy, IoU, and F1 scores, indicating challenges in effectively

Table 4.1: Evaluation Metrics for Different Entropy-Based Methods

Method	IOU	Precision	Recall	Accuracy	F1	ROC AUC
MaxEntropy	0.129	0.143	0.431	0.620	0.225	0.615
RenyiEntropy	0.128	0.142	0.433	0.603	0.214	0.622
Yen	0.119	0.130	0.435	0.673	0.200	0.601
Shanbhag	0.032	0.047	0.173	0.327	0.074	0.528

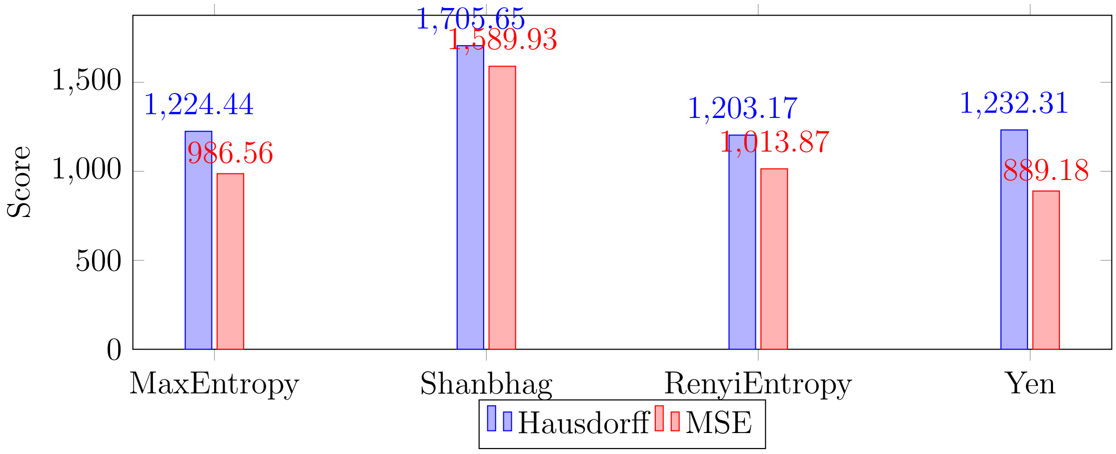


Figure 4.4: Hausdorff and MSE Scores for Entropy Thresholding Methods

separating foreground and background regions in complex images.

Renyi Entropy Thresholding: Renyi Entropy minimizes the Renyi entropy to determine the optimal threshold, providing a flexible approach with control over global and local thresholds. While this method showcased moderate performance, it faced limitations in accurately identifying satellite trails in certain images, resulting in lower IoU and F1 scores.

MaxEntropy Thresholding: MaxEntropy, also known as the maximum entropy method, maximizes the entropy between foreground and background, ensuring maximum information gain. However, it exhibited lower accuracy, IoU, and F1 scores, indicating challenges in effectively capturing the satellite trails' boundaries.

Shanbhag Thresholding: Shanbhag's method minimizes the weighted variances of the foreground and background regions based on histogram entropy. This approach aims to achieve a threshold that optimizes the separation of the two regions. While it performed reasonably well, Shanbhag's method faced challenges in accurately segmenting complex image scenarios, resulting in lower accuracy and IoU scores.

4.4.2 Histogram-Based Results

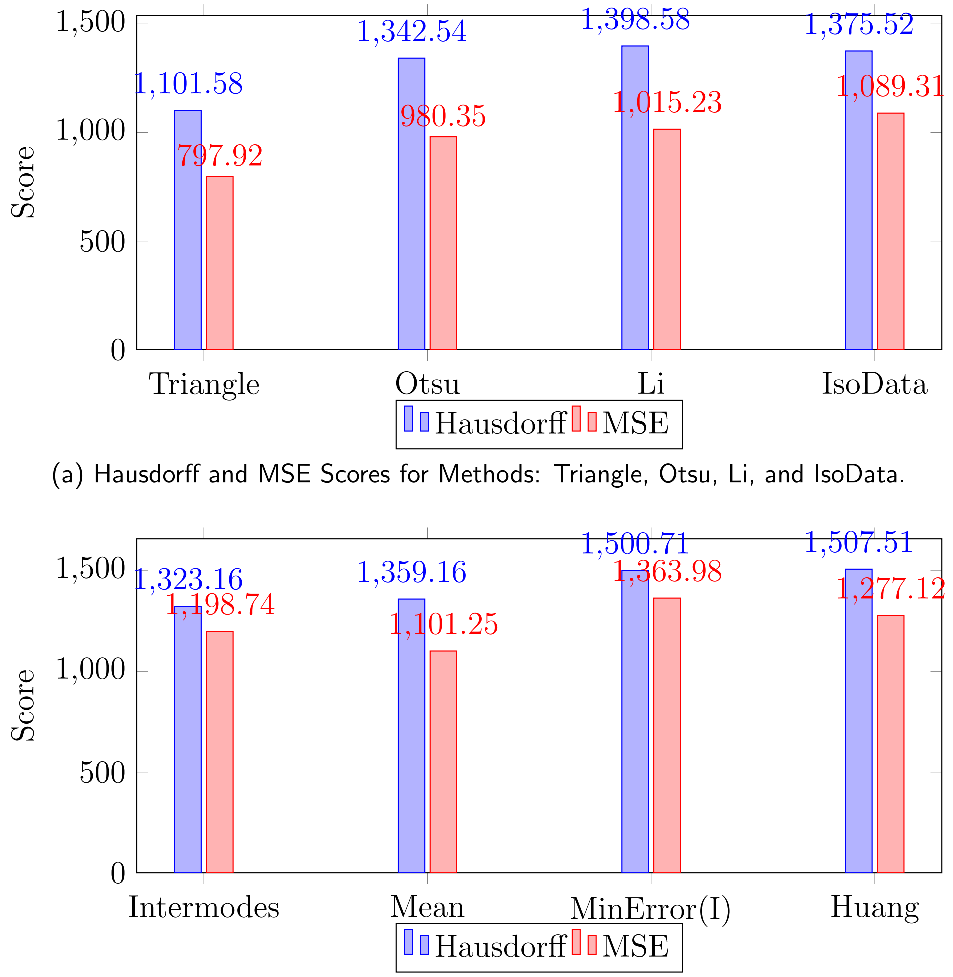
Histogram-based thresholding methods were evaluated in this study to determine the optimal threshold for satellite trail detection automatically. Among the techniques tested were Moments, Triangle, Intermodes, Otsu, Li, Huang, Mean, MinError, and IsoData.

Table 4.2: Evaluation Metrics for Histogram Thresholding Methods

Method	IOU	Precision	Recall	Accuracy	F1	ROC AUC
Triangle	0.182	0.214	0.454	0.725	0.290	0.652
Otsu	0.109	0.142	0.343	0.637	0.200	0.618
Li	0.107	0.140	0.317	0.637	0.194	0.625
IsoData	0.098	0.123	0.315	0.602	0.177	0.615
Intermodes	0.073	0.080	0.307	0.516	0.127	0.566
Mean	0.067	0.116	0.201	0.686	0.147	0.583
MinError(I)	0.050	0.087	0.149	0.530	0.110	0.545
Huang	0.043	0.076	0.149	0.548	0.096	0.561

Moments Thresholding: Moments thresholding method relies on statistical moments of the image histogram to calculate the threshold. It aims to minimize spatial variance within segmented regions. While Moments achieved moderate accuracy and IoU scores, it struggled to perform optimally in complex image scenarios, particularly those affected by noise.

Triangle Thresholding: The Triangle method determines the threshold by maximizing the perpendicular distance from the peak histogram value to the last non-



(a) Hausdorff and MSE Scores for Methods: Triangle, Otsu, Li, and IsoData.

(b) Hausdorff and MSE Scores for Methods: Intermodes, Mean, MinError(I), and Huang.

Figure 4.5: Comparison of Hausdorff and MSE Scores for Intermodes, Mean, MinError and Huang.

zero histogram bin, forming a triangle. It achieved high IoU scores but exhibited lower accuracy due to its sensitivity to noise, impacting its performance in certain images.

Intermodes Thresholding: The Intermodes method identifies the threshold as the value between the modes of the image histogram. This method effectively segments foreground and background regions with distinct intensity peaks. However, it faced challenges in dealing with non-uniform illumination and images with multiple modes, leading to lower IoU and accuracy scores in such cases.

Otsu Thresholding: Otsu's method, known as maximum between-class variance, maximizes the between-class variance to determine the threshold. It is well-suited for bimodal and multi-modal images with distinct intensity peaks. While Otsu achieved moderate accuracy and IoU scores, its effectiveness heavily relies on the presence of distinct histogram peaks, which may limit its performance in certain scenarios.

Li Thresholding: Li's method calculates the threshold by considering the histogram's slope, representing the transition between foreground and background intensities. It adapts to different image conditions and effectively handles images with non-uniform illumination. However, it exhibited lower precision and recall, indicating challenges in minimizing false positives and false negatives.

Huang Thresholding: Huang's method calculates the threshold by minimizing the sum of variances between the thresholded regions and the overall image. It demonstrated robustness against uneven illumination and noise. However, it suffered from lower accuracy and IoU scores, affecting its overall performance.

Mean Thresholding: Mean thresholding computes the threshold as the average of the image's minimum and maximum intensity values. While straightforward, this method may not be effective for images with uneven illumination or complex intensity distributions. It faced challenges in achieving high accuracy and IoU scores.

MinError Thresholding: MinError aims to minimize the classification error between the thresholded and original images using statistical measures such as the Bhattacharyya distance. Although it performed reasonably well, it lacked sensitivity and misidentified many foreground pixels as background pixels, leading to challenges in minimizing false positives.

IsoData Thresholding: IsoData calculates the threshold by iteratively estimating the threshold value until the segmented regions stabilize. It adapts to changing image conditions and uneven illumination. However, it faced limitations in accurately segmenting complex image scenarios and exhibited lower IoU and accuracy scores.

4.4.3 AST & Canny Edge Detection Results

Canny edge detection [39] is primarily used for detecting edges in images rather than image binarization. However, it is possible to utilize the Canny edge detection algorithm to generate a binary mask. After obtaining the edge map from the edge detection step, it can be further processed to generate a binary image. A common approach is to apply a threshold to the edge map, converting all edge pixels above the threshold to white and the rest to black. The Canny method achieved the highest IoU and ROC-AUC scores of the methods being compared to AST. IoU score, it faced challenges in achieving high accuracy due to its susceptibility to noise and complex image scenarios. Additionally, the processing time for Canny edge detection, was faster than the histogram and entropy-based methods.

Table 4.3: Evaluation Metrics for Canny and AST Methods

Method	IOU	Precision	Recall	Accuracy	F1	ROC AUC
AST	0.385	0.465	0.616	0.795	0.530	0.779
Canny	0.256	0.410	0.394	0.689	0.402	0.735

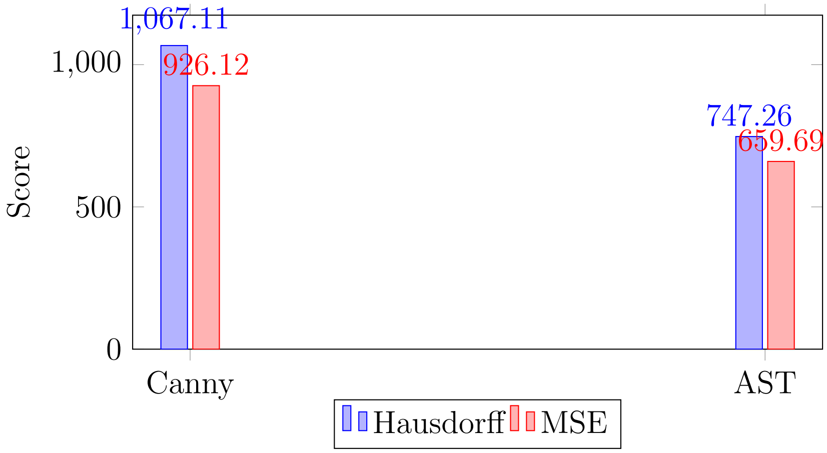
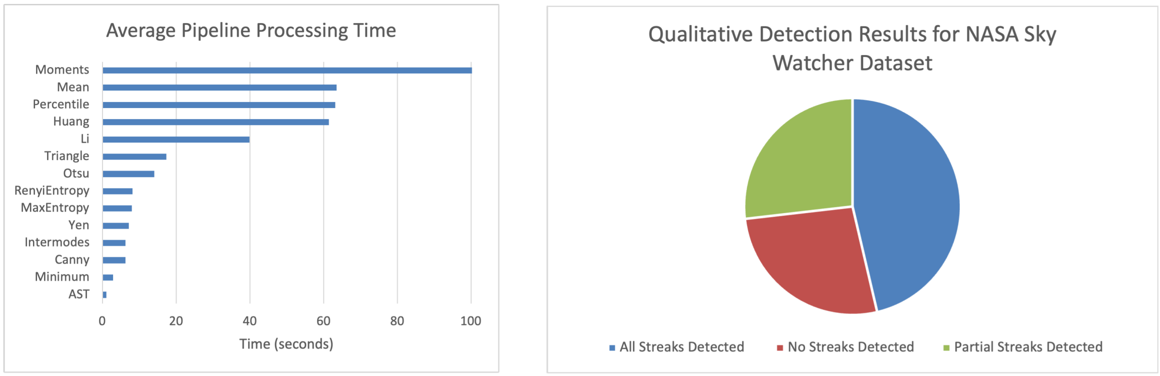


Figure 4.6: Hausdorff and MSE Scores for Canny and AST Methods



(a) Mean Total Pipeline Processing Times for Detection Mask Generation

(b) Qualitative detection results that show the proportion of complete trail detections, compared to partial and failed detection results.

Figure 4.7: A Comparison of Mean Pipeline Processing Times (a) and Qualitative Detection Results (b) for the NASA Sky Watcher Dataset

The results for each thresholding type are combined into a table in Tables 4.4 and 4.5 where AST is highlighted for convenience.

4.4.4 Overall Performance Comparison

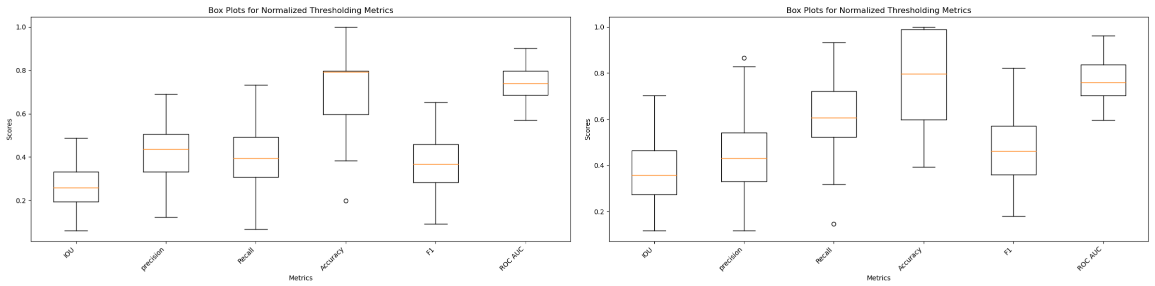
This section analyzes the performance of four key methods used for satellite trail detection: AST, Canny, Triangle, and MaxEntropy. In this section, we aim to gain deeper insights into the effectiveness of these methods. The top-performing methods were used for a Monte Carlo performance test, where five random images were sampled from the NASA Streak Watcher dataset 100 times. This process generates a second round of performance metrics for each method, as shown in Figures 4.8 and 4.9. The mean average scores for the thresholding methods can be seen in Table 4.6.

4.4.5 Qualitative Results for the Asteroid Hunters Dataset

A qualitative analysis was performed using the Asteroid Hunters dataset to test the performance of the AST detection pipeline. A sample of images from the analysis can be seen in Figure 4.11. The top three rows show examples of images where full

Table 4.4: Combined performance metrics table for detection results on the NASA Streak Watcher Dataset.

Method	IOU	Precision	Recall	Accuracy	F1	ROC AUC
AST	0.3849	0.4652	0.6160	0.7947	0.6160	0.7789
Canny	0.2557	0.4097	0.3939	0.6893	0.3939	0.7346
Huang	0.0431	0.0763	0.1488	0.5475	0.1488	0.5606
Intermodes	0.0735	0.0799	0.3069	0.5164	0.3069	0.5665
IsoData	0.0978	0.1228	0.3148	0.6016	0.3148	0.6148
Li	0.1070	0.1404	0.3173	0.6367	0.3173	0.6251
MaxEntropy	0.1288	0.1428	0.4306	0.6203	0.4306	0.6147
Mean	0.0670	0.1159	0.2009	0.6860	0.2009	0.5829
MinError(I)	0.0496	0.0869	0.1492	0.5303	0.1492	0.5452
Otsu	0.1091	0.1420	0.3431	0.6372	0.3431	0.6183
RenyiEntropy	0.1280	0.1418	0.4331	0.6029	0.4331	0.6217
Shanbhag	0.0316	0.0473	0.1727	0.3266	0.1727	0.5275
Triangle	0.1821	0.2141	0.4540	0.7247	0.4540	0.6521
Yen	0.1194	0.1301	0.4345	0.6728	0.4345	0.6006



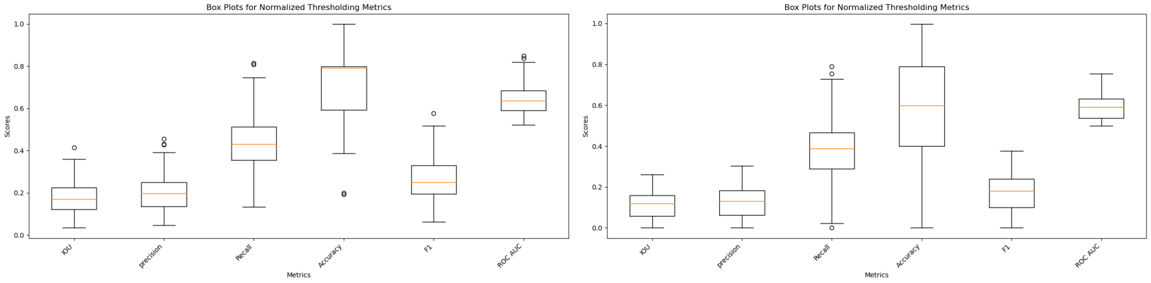
(a) Boxplots for the Canny detection masks

(b) Boxplots for the AST detection masks

Figure 4.8: The boxplots for Canny and AST results for the NASA Streak Watcher dataset when compared to the hand-annotated ground truth.

Table 4.5: Combined performance metrics table for detection results on the NASA Streak Watcher Dataset - Continued.

Method	MSE	Hausdorff	Morph Filter Time
AST	659.6938	747.2596	0.9763
Canny	926.1197	1067.1047	6.1189
Huang	1277.1212	1507.5085	61.0165
Intermodes	1198.7416	1323.1594	6.3944
IsoData	1089.3049	1375.5149	22.9175
Li	1015.2250	1398.5775	39.5106
MaxEntropy	986.5552	1224.4359	7.9260
Mean	1101.2507	1359.1609	63.7269
MinError(I)	1363.9829	1500.7077	41.1090
Otsu	980.3519	1342.5372	14.4877
RenyiEntropy	1013.8749	1203.1673	8.4766
Shanbhag	1589.9256	1705.6501	9.2487
Triangle	797.9232	1101.5745	17.2928
Yen	889.1839	1232.3058	7.0894



(a) Boxplots for the Triangle detection masks

(b) Boxplots for the MaxEntropy detection masks

Figure 4.9: The boxplots for Triangle and MaxEntropy results for the NASA Streak Watcher dataset when compared to the hand-annotated ground truth

Table 4.6: Performance Metrics for Different Methods

Method	IOU	Precision	Recall	Accuracy	F1	ROC AUC	Morph Time (s)
AST	0.403	0.487	0.650	0.829	0.465	0.763	0.984
Canny	0.267	0.419	0.414	0.711	0.365	0.737	5.544
Triangle	0.182	0.212	0.457	0.728	0.263	0.644	15.084
MaxEntropy	0.130	0.146	0.426	0.623	0.175	0.593	7.866

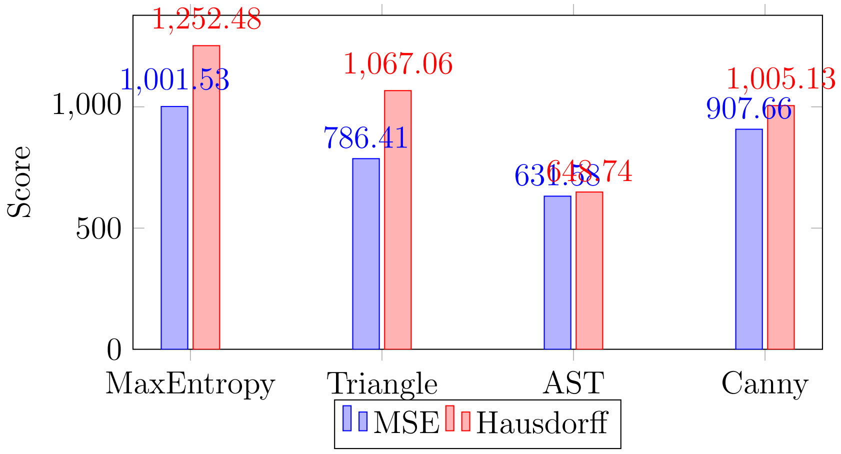


Figure 4.10: MSE and Hausdorff Scores for the top 4 automatic threshold methods for the Monte Carlo experiments.

trail detection was achieved. Based on a qualitative review of the 3073 images in the dataset, the AST detection pipeline was able to detect satellite trails in 1275 images fully. The detection mask had to be free of small objects and capture more than half the trail length to be considered a full detection. The remaining images contained partial or failed detections as shown in row D of Figure 4.11.

4.5 Discussion

Comparing the results, we find that AST consistently outperformed the other automatic thresholding methods. The AST algorithm achieved the highest Mean IoU,

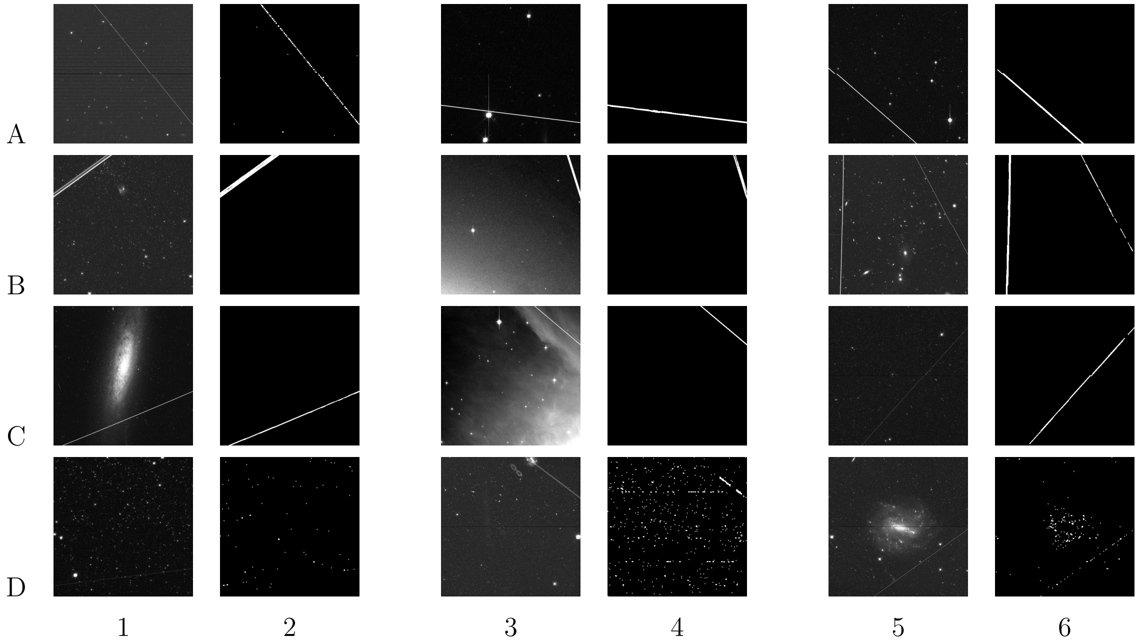


Figure 4.11: A qualitative comparison of automatic threshold masks from the Asteroid Hunters dataset. Row D shows rejected images and masks, and Rows A, B, and C show successfully masked images. Columns 1, 3, and 5 contain original HST data, and columns 2, 4, and 6 contain the AST detection masks.

indicating better overlap between the detection masks and the ground truth. The Triangle and Canny methods achieved high IoU scores but lower accuracy scores because of greater susceptibility to noise, which results in greater processing time, which can be seen in Figure 4.7a. The Mean Accuracy score demonstrated AST's ability to classify pixels as foreground or background correctly. Additionally, AST yielded the lowest Mean MSE and Mean Hausdorff Distance, indicating closer alignment with the ground truth and more accurate boundary detection.

The AST algorithm showed superior performance in terms of Mean Precision, Mean Recall, and Mean F1 score. These metrics provide insights into the algorithm's ability to accurately identify positive instances (satellite trails) while minimizing false positives and negatives. AST achieved high precision, indicating a low rate of false positives and moderate recall, suggesting effective identification of positive instances.

Table 4.6 shows that AST achieved an ROC-AUC score of 0.763, which indicates good performance on this binary classification task. The proposed method outperformed the other methods in the Monte Carlo tests in all measured categories. Comparing AST to the other automatic thresholding methods, it is evident that AST offers distinct advantages. While the entropy-based methods (Yen, Renyi Entropy, MaxEntropy, Shanbhag) and histogram-based methods (Moments, Triangle, Intermodes, Otsu, Li, Huang, Mean, MinError) utilize different strategies to determine the optimal threshold, they often struggle with varying lighting conditions, noise, and complex image scenarios (see Figure 1.3. AST, on the other hand, leverages adaptive techniques to dynamically determine the threshold, leading to improved performance and robustness across different image characteristics.

The processing time comparison also revealed that AST offers a competitive advantage. The average pipeline processing time for AST was lower than most other thresholding methods, suggesting its efficiency in real-time applications or processing large datasets. The morphological filtering stage represents most of the processing time during pipeline execution and is related to the number of detected regions during the CCA. The Mean thresholding technique had the fourth-highest mean accuracy score but lacked sensitivity and misidentified many foreground pixels as background pixels. This shows that the accuracy score has been biased by the class imbalance caused by the background.

However, it is important to consider the limitations and challenges encountered during the evaluation. The datasets used in this study, while representative of ground-based and space-borne images, may not capture the full range of image variations encountered in satellite trail detection. Many of the parameter values used in the configuration of the pipeline are experimental in nature and might not be optimal in all cases. The selection and annotation of images also relied on human judgement, introducing a potential for subjectivity.

Chapter 5

General Conclusions

In this thesis, we discuss the challenges in satellite trail detection by focusing on the development and evaluation of the Automatic Thresholding (AST) algorithm and detection pipeline. By addressing the limitations of existing methods, AST offers a promising solution for quickly and accurately segmenting satellite trails in night sky imagery. In this concluding chapter, we summarize the key findings, contributions, and future directions of this research.

5.1 Summary of Findings

Chapter 1 introduced the growing number of satellites in LEO and the importance of SSA for safety and security. The chapter discusses optical satellite detection of the challenges in satellite trails in night sky imagery, including issues related to thresholding, data availability, and comparison of detection pipelines. We identified the need for an automatic thresholding algorithm to improve detection accuracy, adapt to different imaging systems, and enhance processing efficiency. The subsequent chapters discussed the related approaches, identified challenges and demonstrated the performance of the AST algorithm.

Chapter 2 describes the literature that compared and evaluated various satellite trail detection pipelines and automatic thresholding techniques that have been studied recently. The detection steps were broken down into high-level functions, which allowed the reviewed detection pipelines to be better compared. The chapter provides a detailed description of the operation of a satellite trail detection pipeline by comparing the detection stages used in the literature. Special attention was paid to the pre-processing and star removal stages of the reviewed detection pipelines, as these were identified as possible areas that could be improved. Finally, the types of automatic thresholding algorithms were explained, and their strengths and weaknesses were compared.

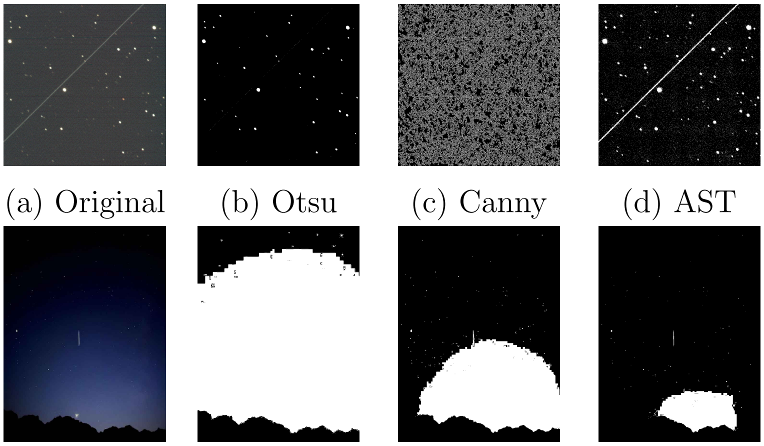
Chapter 3 presented the development and implementation of the AST algorithm. By leveraging adaptive techniques and statistical measures, AST was able to dynamically determine the threshold, which resulted in improved accuracy and reduced false positives and false negatives. The algorithm's flexibility allows for integration into existing detection pipelines by allowing for a greater range of input image sizes with no prior scene knowledge required. The chapter continued by describing the proposed morphology stage that includes two shape-based filters to remove compact objects using extent and eccentricity properties.

Chapter 4 described the experimental evaluation of AST using datasets from the NASA Streak Watcher project and the Asteroid Hunters dataset. The results demonstrated the superior performance of AST compared to other automatic thresholding methods. AST consistently achieved higher accuracy, precision, recall, and F1 scores, indicating its effectiveness in accurately segmenting satellite trails.

5.2 Contributions

This research makes several contributions to the field of satellite trail detection:

1. Development of the AST Algorithm: This study provides insights into the challenges associated with satellite trail detection and proposes AST as a solution. The proposed AST algorithm introduces a novel approach to automatic thresholding, improving the accuracy and efficiency of satellite trail segmentation. By dynamically adjusting the threshold based on the assumption of a sparse bi-



(a) Original (b) Otsu (c) Canny (d) AST

(e) Original (f) Triangle (g) Moments (h) AST

Figure 5.1: A comparison of automatic threshold mask generation using the proposed detection pipeline with different automatic thresholding algorithms used for image binarization. The results for AST in subfigures (d) and (h) show that the proposed algorithm can produce a binary mask without over/under thresholding the input image.

modal histogram, AST outperforms traditional methods and provides reliable detection results by not over/under thresholding the image during binarization (see Figure 5.1.)

2. Morphological Detection Pipeline: To further enhance satellite trail detection, a novel star/small object filter based on extent and eccentricity was proposed to improve efficiency in the detection stage. The shape-based filter results in more robust detection than size-based methods and enables detection pipelines to allow for inputs of arbitrary size. By combining eccentricity and extent properties together, the morphological filter enhances satellite trail detection by reducing detection clutter more effectively than perimeter/area-based methods.

3. Pixel-level Annotated Satellite Trail Data: To evaluate our proposed method, a large amount of annotated data was required. To achieve this, a citizen science website was constructed to facilitate the distributed labelling of the Hubble Asteroid Hunters satellite trail data. The dataset provides ground truth pixel-level annotations for 3073 HST images, which can be used to train future segmen-

tation models. The resulting annotated dataset has been made available for public use to study satellite trail detection further.

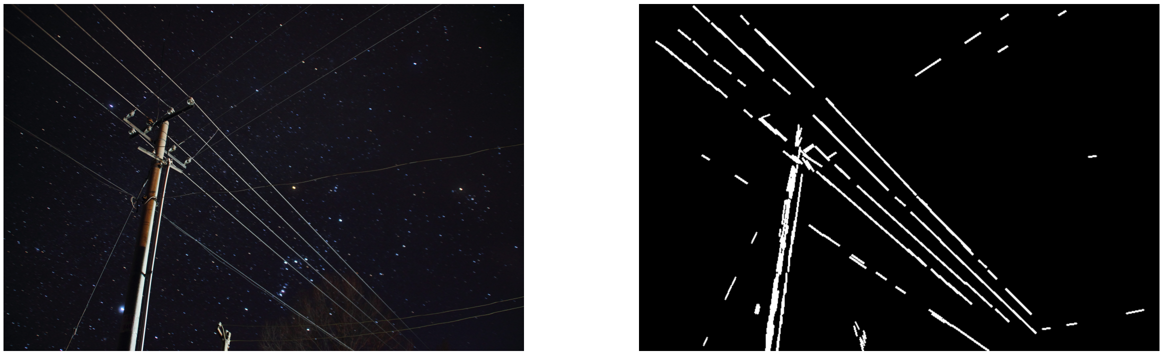
5.3 Future Directions

While this research has achieved significant advancements in satellite trail detection using the AST algorithm and detection pipeline, there are several avenues for future exploration:

1. Pipeline Integration: The AST algorithm can be easily implemented in other satellite detection pipelines and most often be used in cases where manual global thresholding is employed. A FITS interface would be useful for making AST more accessible for use with astronomical imagery. Further testing on other optical imaging platforms would support automatic thresholding in satellite detection tasks. It also strengthens the argument that AST can be used effectively using various imaging configurations, enabling a higher level of SSA in the future.

2. Develop AST Adaptability: Continued research can focus on adapting AST for specific image detection tasks where sparse, unbalanced bimodal histograms can be leveraged. Examples of such applications where AST could prove beneficial include the low-light detection of power lines and underwater cables/pipes (see Figure 5.2). The AST algorithm can be reconfigured to work backward in situations where the background pixels have a higher intensity than those of the detected object. This could allow AST to be utilized in daytime applications like aircraft contrail or horizon detection tasks.

3. Improve Pipeline Capabilities: The proposed detection pipeline shows robust detection results in both ground-based and space-borne telescopes based on the available data. The evaluated datasets also reveal the shortcomings in our proposed detection pipeline that are likely present in the literature discussed in Chapter 2. The proposed morphology filter results in an improvement of the detection efficiency in the Probabilistic Hough detection stage. This perfor-



(a) The input image used to generate the detection mask in the accompanying figure.

(b) Output detection mask from the AST detection pipeline.

Figure 5.2: The AST algorithm was applied to an image of illuminated powerlines at night. The detection pipeline was used without morphological filtering as the extent/eccentricity removes too many foreground objects.

mance optimization comes at the price of missed detections due to the CCA merging intersecting trails. This leads to the satellite trails being grouped as a single object and ultimately removed by the shape filter as an object with insufficient eccentricity. This problem can be solved by performing the CCA on zones or binned regions rather than the entire image. Therefore, the detection pipeline results could be improved if the CCA stage can be more robust to non-parallel/intersecting satellite trails.

4. Machine Learning Models: The stages of the AST algorithm and the proposed detection pipeline are functions that can be taught to a machine learning algorithm. The data collected for this study with the AST and hand-annotated ground truths can be used to train a deep-learning model to generate detection masks in real-world images using data augmentation. The orientation-free images from the Asteroid Hunters dataset would allow for substantial augmentation, which would greatly improve data availability for training satellite trail detection models. An auto-encoder segmentation model could be used to learn the features of satellite trails from the annotated, possibly resulting in improved detection outcomes.

5.4 Conclusion

In conclusion, this thesis has addressed the challenges in satellite trail detection and proposed the AST algorithm as a promising solution. The experimental evaluation has demonstrated the superiority of AST in accurately segmenting satellite trails compared to traditional thresholding methods. The contributions made in this research pave the way for future advancements in satellite trail detection and image segmentation tasks.

By improving accuracy, offering flexibility, and ensuring efficient processing, AST opens up new possibilities in satellite trail detection, benefiting various applications in astronomy, environmental monitoring, and space exploration. With further research and collaboration, the AST algorithm can be refined, integrated into detection pipelines, and extended to other image segmentation tasks, advancing the field and enabling new insights and applications.

Overall, this research serves as a significant contribution to satellite trail detection and sets the stage for further developments in this rapidly evolving field. We hope the findings and insights presented in this thesis will inspire and guide future research in pursuing accurate and efficient satellite trail detection methods.

Bibliography

[1] Hainaut, Olivier R and Williams, Andrew P. Impact of satellite constellations on astronomical observations with ESO telescopes in the visible and infrared domains. *A&A* 2020;636:A121. doi:10.1051/0004-6361/202037501.

[2] Tyson JA, Željko Ivezić, Bradshaw A, et al. Mitigation of LEO Satellite Brightness and Trail Effects on the Rubin Observatory LSST. *The Astronomical Journal* 2020;160:226. ISSN 1538-3881. doi:10.3847/1538-3881/abba3e.

[3] McDowell JC. The low Earth orbit satellite population and impacts of the spacex starlink constellation. *The Astrophysical Journal* 2020;892(2). doi:10.3847/2041-8213/ab8016.

[4] Lawrence A, Rawls ML, Jah M, et al. The case for space environmentalism. *Nature Astronomy* 2022;6(4):428–435. doi:10.1038/s41550-022-01655-6.

[5] Mroz P, Otarola A, Prince TA, et al. Impact of the SpaceX Starlink Satellites on the Zwicky Transient Facility Survey Observations 2022;doi:10.3847/2041-8213/ac470a.

[6] Kruk S, García-Martín P, Popescu M, et al. The impact of satellite trails on Hubble Space Telescope Observations. *Nature Astronomy* 2023;7(3):262–268. doi:10.1038/s41550-023-01903-3.

[7] Rijlaarsdam D, Yous H, Byrne J, et al. A Survey of Lost-in-Space Star Identification Algorithms Since 2009. *Sensors* 2020;20(9). ISSN 1424-8220. doi:10.3390/s20092579.

[8] Kessler DJ and Cour-Palais BG. Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research* 1978;83(A6):2637–2646.

[9] shared with Satellite Streak Watcher on Anecdataorg O. Dataset accessible at https://www.anecdata.org/explore?project_id=687 2023. Accessed on

2023-02-25. Licensed under Creative Commons Attribution 4.0 International License.

[10] Chatters EP, Crothers BJ, Command A, et al. Space Surveillance Network. AU-18 Space Primer 249-258, Air University Press 2009.

[11] Nikolaus Kollo. Satellite Annotation Platform. https://satellites.oaktreedesigns.ca [2023]. Accessed: [1 June 2023].

[12] Kopycinski J, Kuklinski P, Rzesa W, et al. Satellites detection, tracking and cataloguing system. In *2018 22nd International Microwave and Radar Conference (MIKON)* 2018; pages 370–373. doi:10.23919/MIKON.2018.8405229.

[13] Samadzadegan F and Alidoost F. The design and implementation of an optical astronomical satellite tracking system. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2013;XL-1/W3:25–30. doi:10.5194/isprsarchives-XL-1-W3-25-2013.

[14] Levesque M. Automatic Reacquisition of Satellite Positions by Detecting Their Expected Streaks in Astronomical Images. In Ryan S, editor, *Advanced Maui Optical and Space Surveillance Technologies Conference* 2009; page E81.

[15] Nir G, Zackay B, and Ofek EO. Optimal and efficient streak detection in astronomical images. *The Astronomical Journal* 2018;156(5):229. doi:10.3847/1538-3881/aaddff.

[16] Radon J. On the determination of functions from their integral values along certain manifolds. *IEEE Transactions on Medical Imaging* 1917;5(4):170–176.

[17] Stark DV, Grogin N, Ryon J, et al. Improved Identification of Satellite Trails in ACS/WFC Imaging Using a Modified Radon Transform. Instrument Science Report ACS 2022-8, 25 pages 2022.

[18] Farzaneh S, Sharifi MA, and Kosary M. Automatic satellite streaks detection in astronomical images. *Journal of the Earth and Space Physics* 2017;43:473–487.

[19] Lowe DG. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2. Ieee 1999; pages 1150–1157.

[20] Jain AK. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters* 2010;31(8):651–666.

[21] Eberhart RC and Kennedy J. A new optimizer using particle swarm theory. In

Proceedings of the sixth international symposium on micro machine and human science. IEEE 1995; pages 39–43.

[22] Coley DA. *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific Publishing Company 1999.

[23] Hough P. A method and apparatus for analysis of bubble chamber pictures. *International Conference on High-Energy Physics* 1962;4:179–184.

[24] Rood S Wijnen. Automated Satellite Detection and Sky-position Extraction in Astronomical Images. In *Advanced Maui Optical and Space Surveillance Technologies Conference* 2022; .

[25] Borncamp D and Lim PL. Satellite Detection in Advanced Camera for Surveys/Wide Field Channel Images. Instrument Science Report ACS 2016-01, 10 pages 2016.

[26] Privett G, George S, Feline W, et al. AUTOMATED EXTRACTION OF SATELLITE TRAILS FROM WIDE ANGLE CCD IMAGERY. *Revista Mexicana de Astronomía y Astrofísica Serie de Conferencias* 2019;51:150–158. doi: 10.22201/ia.14052059p.2019.51.26.

[27] Danarianto MD, Maharani AM, Falah BM, et al. Prototype of automatic satellite streak detection, identification and initial orbit determination pipeline from optical observation. *International Journal of Remote Sensing and Earth Sciences (IJReSES)* 2019;16(2):142–154.

[28] Rachith K Chang. Detection methods for a statistical analysis of the population of satellites and space debris from astronomical images. In *Advanced Maui Optical and Space Surveillance Technologies Conference* 2022; .

[29] Paillassa M, Bertin E, and Bouy H. MAXI MASK and MAXI TRACK: Two new tools for identifying contaminants in astronomical images using convolutional neural networks. *Astronomy and Astrophysics* 2020;634. ISSN 14320746. doi: 10.1051/0004-6361/201936345.

[30] Bouy H, Bertin E, Moraux E, et al. Dynamical analysis of nearby clusters. Automated astrometry from the ground: precision proper motions over a wide field. *Astronomy & Astrophysics* 2013;554:A101. doi:10.1051/0004-6361/201220748.

[31] Toloei A, Zahednamazi M, Ghasemi R, et al. A comparative analysis of star identification algorithms. *Astrophysics and Space Science* 2020;365. doi:10.1007/s10509-020-03775-9.

[32] Kennamer N, Kirkby D, Ihler A, et al. ContextNet: Deep learning for Star Galaxy Classification. In Dy J and Krause A, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*. PMLR 2018; pages 2582–2590.

[33] Otsu N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 1979;9(1):62–66. doi:10.1109/TSMC.1979.4310076.

[34] Zack GW, Rogers WL, and Latt SA. Automatic measurement of sister chromatid exchange frequency. *Journal of Histochemistry & Cytochemistry* 1977;25(7):741–753.

[35] Tsai Dw. Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *IEEE transactions on image processing* 1985;4(1):1–7.

[36] Glasbey C. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical Models and Image Processing* 1993;55(6):532–537. doi:10.1006/cgip.1993.1040.

[37] Kittler J and Illingworth J. Minimum error thresholding. *Pattern Recognition* 1986;19(1):41–47. doi:10.1016/0031-3203(86)90030-0.

[38] Kapur J, Sahoo P, and Wong A. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing* 1985;29(3):273–285. doi:10.1016/0734-189x(85)90125-2.

[39] Canny J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1986;PAMI-8(6):679–698. doi:10.1109/TPAMI.1986.4767851.

[40] Blakeslee JP, Anderson J, Meurer G, et al. Hubble ACS Tools: A Comprehensive Data Reduction System for ACS Images. *Publications of the Astronomical Society of the Pacific* 2009;121(880):621.

[41] Coll B and Morel JM. A Review of Image Denoising Algorithms, with a New One. *SIAM Journal on Multiscale Modeling and Simulation* 2005;4. doi:10.1137/040616024.

[42] Xu L, Jiang J, and Liu L. RPNet: A representation learning-based star identification algorithm. *IEEE Access* 2019;7:92193–92202.

[43] Abramowitz M and Stegun IA. *Handbook of Mathematical Functions with For-*

mulas, Graphs, and Mathematical Tables. Dover Publications, Inc. 1972. ISBN 978-0486612720.

[44] Rosenfeld A and Pfaltz JL. Connected Component Analysis. *Journal of the Association for Computing Machinery* 1966;13(4):622–626. doi:10.1145/321356.321357.

[45] Schindelin J, Arganda-Carreras I, Frise E, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods* 2012;9(7):676–682.

[46] Yen JC, Chang FJ, and Chang S. A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing* 1995;4(3):370–378. doi:10.1109/83.366472.

[47] Rényi A. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4. University of California Press 1961; pages 547–562.

[48] Shanbhag AG. Utilization of information measure as a means of image thresholding. *Graphical models and image processing* 1994;56(5):414–419.

[49] Huang LK and Wang MJ. Image thresholding by minimizing the measures of fuzziness. *Pattern Recognition* 1995;28(1):41–51. doi:10.1016/0031-3203(94)E0043-K.

[50] Prewitt J and Mendelsohn M. The analysis of cell images. *Annals of the New York Academy of Sciences* 1966;128(3):1035–1053. doi:10.1111/j.1749-6632.1965.tb11715.x.

[51] Li CH and Tam PKS. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognition Letters* 1998;19(8):771–776. doi:10.1016/S0167-8655(98)00057-9.

[52] Price-Whelan AM, Sipőcz B, Günther H, et al. The Astropy Project: Building an open-science project and status of the v2. 0 core package. *The Astronomical Journal* 2018;156(3):123.

[53] Wada K. labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme 2018.