Automated Visual Anomaly Detection for Proximity Operations in the Space Domain

by

Selina Leveugle

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Aerospace Science and Engineering
University of Toronto

Abstract

Automated Visual Anomaly Detection for
Proximity Operations in the Space Domain

Selina Leveugle
Master of Applied Science
Graduate Department of Aerospace Science and Engineering
University of Toronto
2024

The demand for autonomy on the Canadarm3 presents several new challenges, including the need for the arm to use its inspection cameras to perform autonomous anomaly detection, that is, to identify hazards within its operating environment. In this thesis, we introduce the ALLO dataset, a novel resource for developing and testing anomaly detection algorithms in the space domain. The ALLO dataset is used to evaluate the performance of state-of-the-art anomaly detection algorithms, demonstrating how current methods struggle to generalize to the complex lighting and scenery of space. We then present MRAD, a novel, shallow anomaly detection algorithm designed specifically for space applications. By leveraging the known pose of the Canadarm3 inspection camera, MRAD reformulates the anomaly detection problem and outperforms existing methods. Given the low tolerance for risk in space operations, this research provides essential tools and a potential solution for visual anomaly detection in lunar orbit.

Acknowledgements

This thesis would not have been possible without the support and encouragement of my mentors, colleagues, and friends. It has been my great pleasure to work on this research and I would like to extend my heartfelt thanks to everyone who has supported me in my graduate studies. First and foremost, I would like to thank my supervisor, Professor Jonathan Kelly, for the guidance, support, and encouragement he gave me throughout my thesis. I am truly grateful to him for offering me the opportunity to work on this project, and for creating an incredible lab to work in. His mentorship is what allowed me to grow academically and succeed in my work.

I'd also like to thank the wonderful team that worked alongside me on this research. From TRAIL Lab, my gratitude goes to Chang Won Lee and Professor Steven Waslander for their partnership, support, and feedback throughout this project. From MDA, I am thankful to Chris Langley, Svetlana Stolpner, and Paul Grouchy for sharing their expertise and guidance. The success of this research would not have been possible without the collaboration and assistance provided by this team.

Furthermore, I would like to thank my lab-mates at STARS lab and at the Robotics Institute for welcoming me to Toronto and providing a kind and supportive community. The camaraderie and friendships I have found in and out of the lab have made every research problem a little bit easier to tackle. Lastly, I am deeply grateful to my friends and family outside of Toronto. Thank you to my parents for always encouraging me to try new things and to challenge myself; and to Alex, Kyle, and Chris for always believing in me.

This thesis is the culmination of the support, guidance, encouragement, and belief of everyone who has helped me over the past two years. I am profoundly grateful to everyone whose contributions made this achievement possible.

Contents

List of Tables

List of Figures

List of Abbreviations

AP Average Precision

AUROC Area Under the Receiver Operating Characteristic

BRDF Bidirectional Reflectance Distribution Function

CAD Computer Aided Design

CNN Convolutional Neural Network

CSA Canadian Space Agency

ISS International Space Station

MDA Macdonald, Dettwiler & Associates

MRAD Model Reference Anomaly Detection

NRHO Near-Rectilinear Halo Orbit

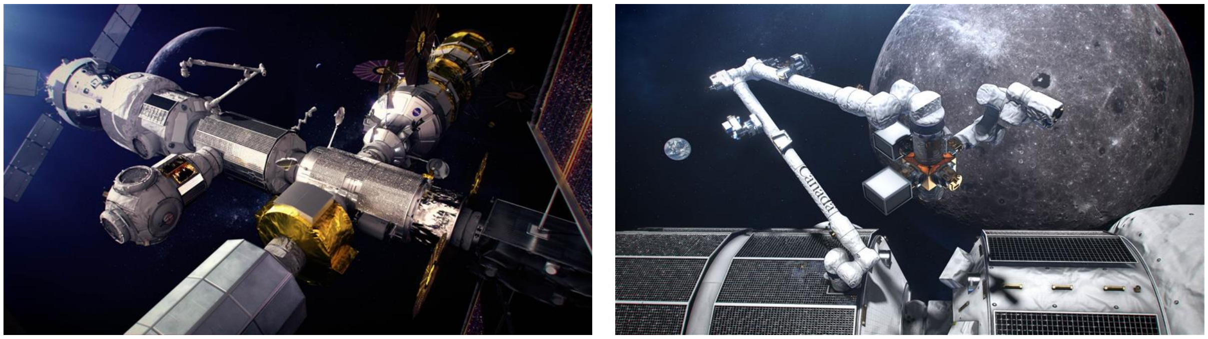
RXD Reed-Xiaoli Detector

Chapter 1

Introduction

The Artemis program is a NASA-led lunar exploration program that aims to extend human space exploration beyond low-earth orbit. The primary goal of the Artemis program is to establish a human presence on the Moon by setting up a permanent base, facilitating future missions to Mars. A key element of the Artemis program is the Lunar Gateway space station. The Gateway will be the first space station in lunar orbit and will serve as a testing ground for the technologies and procedures needed for extended deep-space missions [9].

Canada's primary contribution to the Lunar Gateway is the Canadarm3, a smart robotic arm jointly developed by MDA Space Ltd. and the Canadian Space Agency (CSA). This robotic system will perform multiple tasks at the Gateway, including the maintenance and inspection of the station as well as the capture of visiting vehicles [1]. Conceptual renderings of the Lunar Gateway and Canadarm3 are shown in Figure 1.1.



(a) The Lunar Gateway

(b) The Canadarm3

Figure 1.1: Artist conception of the Gateway and the Canadarm3 from [1]

1.1 Motivation

The Gateway's unique operational requirements present new challenges for its multinational partners. Unlike the International Space Station (ISS), the Gateway will primarily be uncrewed, and both its distance from Earth and its orbit hinder the constant communication that would be needed for direct human control. Therefore, the Gateway must be designed for long-term autonomous operation [10]. This autonomy is especially crucial for the Canadarm3. During autonomous operation, collision avoidance is paramount as a collision could catastrophically damage the arm or the station itself. Accordingly, the arm's vision system must autonomously detect potential hazards that could arise throughout the Gateway's mission [11]. Possible hazards include loose tools or debris, for example the tool bag that was lost by astronauts on an ISS spacewalk in November 2023. This tool bag is shown floating away from the ISS in Figure 1.2.

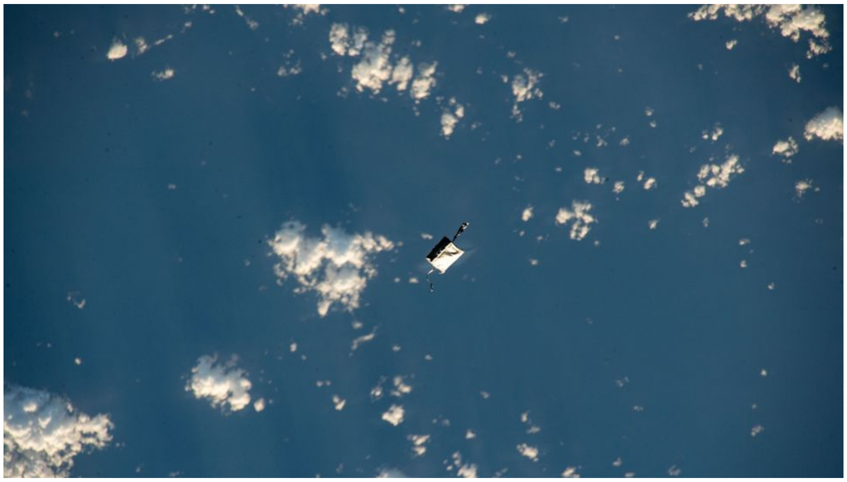


Figure 1.2: Photo taken from the ISS of a tool bag lost during a space walk [2].

In the context of autonomous operation onboard the Lunar Gateway, a *hazard scenario* is defined as any operational scenario that differs from predefined expectations. During operation, the Canadarm3 will use its inspection cameras to navigate around the station; images taken by these cameras will be analyzed to search for hazard scenarios. The problem of identifying data that deviate from the set of expected inputs is called anomaly detection. Algorithms that perform anomaly detection aim to identify objects, features, or pixels in an image that are statistically different from an established baseline [12]. To the best of our knowledge, no anomaly detection algorithm or dataset exists that addresses the problem of vision-based anomaly detection for orbital spacecraft operations. The Canadarm2, currently deployed aboard the ISS, has no autonomous anomaly detection capabilities and relies instead on a human operator to ensure safe conditions before every operation. Anomaly detection is challenging in the space domain in part because of the complex lighting conditions that result from the black background of space coupled with harsh direct sunlight. These conditions make it difficult for an anomaly detection algorithm to properly localize pixels and decide if those pixels are anomalous or non-anomalous.

Current methods of visual anomaly detection are designed for much more simple environments than those the Gateway will encounter in lunar orbit. Existing algorithms are not designed to handle the complex and diverse images taken in the space domain, and have not been tested in this domain due to a lack of applicable data. Given the severe consequences of a collision involving the Canadarm3, an accurate and reliable vision-based anomaly detection algorithm is required for autonomous operation. As no algorithm or data yet exists for the task of anomaly detection in space, both must be developed to ensure the safe operation of the Canadarm3.

1.2 Contributions

We present the task of anomaly detection for a space station in lunar orbit. Many future space exploration missions will require a high level of autonomy and will need to be able to identify conditions in their work environment that deviate from expectations. Given the widespread use of monocular cameras for space applications, visual anomaly detection has the potential to ensure the safe autonomous operation of space robotics. However, the difficult lighting conditions and complexities of the environment must be addressed before such an algorithm can be deployed on an autonomous space station.

As a first step towards solving the problem of anomaly detection for the Canadarm3, we introduce the Anomaly Localization in Lunar Orbit (ALLO) dataset, a novel dataset for anomaly detection on robotic manipulators in space. Since the Lunar Gateway is currently under development and does not yet physically exist, we develop a synthetic dataset replicating the images expected to be captured by the cameras on the Canadarm3. We then use the ALLO dataset to establish a benchmark of several state-of-the-art anomaly detection algorithms, evaluating their performance in the space domain. We show that even with dataset-specific tuning, state-of-the-art algorithms have difficulty generalizing to scenes that are more complex than those they were designed for. We demonstrate that the overlap between non-anomalous and anomalous pixels in the space domain is a significant hurdle that existing anomaly detection algorithms are ill-equipped to handle.

To address the anomaly detection research gap, we introduce Model Reference Anomaly Detection (MRAD), a novel solution for anomaly detection in space that exploits the known model of the space station. MRAD uses the known pose of the Canadarm3's camera to generate a reference image that represents what the arm's camera is expected to capture. By using a specific reference image, MRAD is able to approach each image individually instead of relying on a generic, dataset-specific, non-anomalous distribution. We test MRAD on the ALLO dataset and show that it has superior performance compared to current state-of-the-art anomaly detection algorithms, for the space domain. Finally, we evaluate how the various features in the tested images affect MRAD's final performance.

The main contributions of this thesis are as follows.

- We present a new, relevant task for anomaly detection that goes beyond conventional Earth-bound applications.

- We introduce the ALLO dataset, a novel, open-source dataset for visual anomaly detection during autonomous operation in lunar orbit. The dataset comprises 44,663 anomaly-free images and 27,432 anomalous images with pixel-level ground-truth maps.

- We evaluate several state-of-the-art anomaly detection algorithms on the ALLO dataset, and discuss how existing algorithms are insufficient for anomaly detection in space.

- We introduce a unique solution for anomaly detection called MRAD that exploits the known model prior and does not rely on any deep learning.

- We test MRAD on the ALLO dataset and examine the factors that affect its performance.

Chapter 2

Background

This chapter provides a review of fundamental mathematical concepts needed to understand the contents of the thesis. First, we provide an overview of image rendering and outline how images can be synthetically generated by a computer. Next, we review some of the principles of anomaly detection; we describe how anomalies are usually defined and then identified, and how the performance of anomaly detection algorithms is evaluated. Finally, we provide background on image-based camera pose alignment, outlining the process to align or register one image to another.

2.1 Model-Based Image Rendering

An increased demand for autonomy in space exploration has driven the development of vision-based navigation algorithms. Synthetic image datasets, generated using computer rendering software, are frequently used to test these algorithms. Critically, synthetic datasets are required because existing, real-world images may be unavailable or may fail to meet the specific requirements of the target mission [13]. In turn, rendering programs such as Blender [14] and Unreal Engine 5 [15] have been applied to create space image datasets as these programs use path tracing to create an extensive range of realistic scenarios. We use Blender's path tracing capabilities to create the novel, photorealistic space image dataset presented in Chapter 4.

Path tracing is a computer graphics algorithm used to render images by solving Kijiya's rendering equation [16],

$$L_o(p, \omega_\mathbf{o}) = L_e(p, \omega_\mathbf{o}) + \int_S f_r(p, \omega_\mathbf{i}, \omega_\mathbf{o}) L(p', \omega_\mathbf{i}) G(p, p') V(p, p') d\omega_i, \tag{2.1}$$

where $L_o(p, \omega_\mathbf{o})$ is the outgoing radiance along direction $\omega_\mathbf{o}$ at point p. The term $L_e(p, \omega_\mathbf{o})$

refers to the emitted radiance from the surface along direction $\omega_\mathbf{o}$. The illumination hemisphere S is the unit hemisphere of all possible incoming light directions surrounding the surface normal at p, $f_r(p, \omega_\mathbf{i}$, while $\omega_\mathbf{o})$ is the bidirectional reflectance distribution function (BRDF) that defines how light is reflected from a surface. The term $L(p', \omega_\mathbf{i})$ is the incoming radiance at point p' along direction $\omega_\mathbf{i}$, and $G(p, p')$ describes the geometric relationship between p and p'.

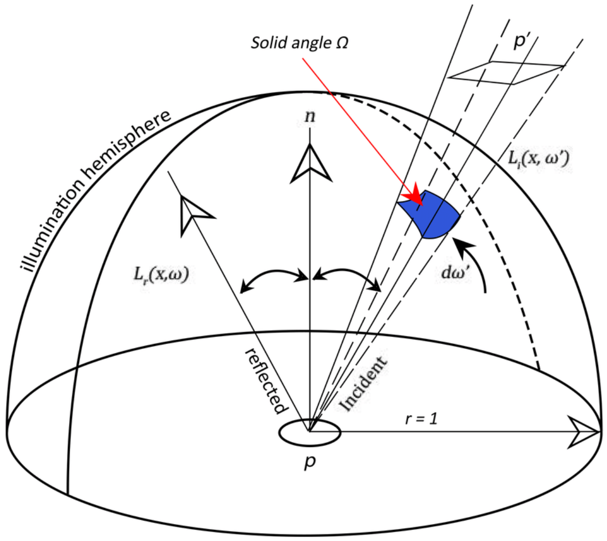


Figure 2.1: Geometric representation of rendering equation from [3], showing the illumination hemisphere around point p.

Kijiya's rendering equation calculates the amount of light emitted from a point along a specific viewing direction based on the incoming light and the BRDF [3]. Path tracing solves this equation through a combination of Monte Carlo integration and ray tracing. By itself, ray tracing calculates the value of a pixel by sending a ray outwards from the camera and tracing the ray as it bounces through the scene until it reaches a light source. Path tracing extends this idea by simultaneously sending out thousands of rays from the camera and tracking them as they bounce through the scene. Monte Carlo integration is then used to achieve a representative yet computationally-feasible output. Path tracing effectively models the physical behaviour of light, allowing for the creation of photorealistic computer-generated images.

2.2 Anomaly Detection

Anomaly detection is the problem of determining whether or not a test sample falls within an established data distribution. An anomaly detection algorithm aims to identify abnormal samples, referred to as anomalies, that have substantial variations from the expected distribution [17]. The research presented in this thesis examines how the problem of anomaly detection can be solved within the context of space exploration. We evaluate the performance of vision-based anomaly detection algorithms with the goal of developing an approach capable of identifying anomalies in proximity to the Canadarm3. The effectiveness of these algorithms is based on their ability to classify images and to localize anomalous pixels.

2.2.1 Principles of Anomaly Detection

Anomaly detection is a binary classification problem that aims to distinguish between anomalous and non-anomalous classes. The non-anomalous, or normal, class comprises regular data points that meet the expectations of a given task. The anomalous class consists of points that deviate from the normal class, indicating possible defects or unexpected occurrences. In multi-class classification, knowledge of multiple classes is used in classifying data during inference. For example, out-of-distribution (OOD) detection seeks to detect data points that are not represented in any of the labelled classes encountered during the training phase. Abnormal data points are those that do not overlap with any of the labelled classes seen during training [18]. However, in anomaly detection, only information about the non-anomalous class is known. By using knowledge of the normal data, a probabilistic distribution of the non-anomalous class can be defined, and any sample significantly different from this distribution is classified as an anomaly.

The distinction between the non-anomalous and anomalous classes can be challenging to establish for a variety of reasons. When defining the non-anomalous data distribution, it is assumed that a process can be understood from the observable data and verified with additional data. Anomalous points are subsequently defined as deviations from the observed process [17]. Therefore, an anomaly can be described as an irregular, unexpected, or unpredictable instance that deviates from an established pattern [19]. Anomalies can be caused by many different systematic variations, presenting the two key challenges for anomaly detection algorithms [20]:

- Anomaly uncertainty: it is usually not known in advance what an anomaly might look like and there may be multiple types of anomalies.

- Anomaly scarcity: since anomalies are very rare and diverse it is difficult to correctly identify all of them (thus high false alarm rates are common).

Two types of anomalies are defined in the visual anomaly detection literature: structural anomalies and logical anomalies [4]. Structural anomalies are locally confined structures or features in an image that are not present in the anomaly-free data distribution. Examples of structural anomalies include an erroneous colour or an unexpected texture in the image. Logical anomalies violate the logical constraints of the scene, such as a missing object or an extra object. Examples of logical and structural anomalies are shown in Figure 2.2. Many existing anomaly detection algorithms focus on identifying structural anomalies by examining image features, with little attention given to detecting logical anomalies.

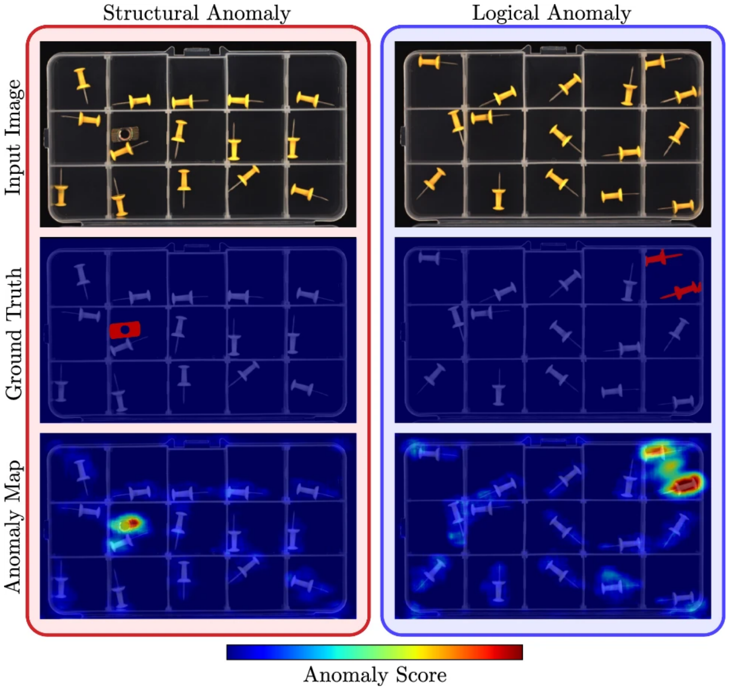


Figure 2.2: Example of structural and logical anomalies from [4].

The primary challenge in anomaly detection is that an algorithm does not inherently know what it is looking for. Instead, the algorithm searches for samples that differ from what it has previously seen. Various approaches to anomaly detection exist, but most algorithms can be broken down according to the following three design choices [12]:

1. The definition of the normal class, including any structural assumptions about the background.

2. The anomaly score metric such as a distance measurement or a bespoke loss function.

3. The decision method such as a statistical model or an empirical threshold.

An anomaly detection algorithm may or may not use learning. Deep learning-based methods extract features using convolutional neural networks (CNNs). These networks typically learn which features are representative and useful for analysis by pre-training on large datasets such as ImageNet [21]. Learning is often performed in conjunction with a custom loss function that aims to minimize the anomaly score on anomaly-free data. In contrast, shallow methods, also known as traditional methods, do not use deep learning [5]. Shallow methods directly use image pixel data or transform the image to a subspace. These methods then use a distance function as the anomaly score and make classification decisions based on an empirical threshold or statistical model [19].

Both deep and shallow anomaly detection algorithms can be divided into three categories: classification, probabilistic, and reconstruction, with classification and probabilistic approaches sometimes grouped together [5]. Classification models discriminatively establish a boundary between non-anomalous and anomalous classes, corresponding to the desired density level of the non-anomalous data distribution. Probabilistic methods, also known as density estimation methods, estimate the probability distribution over the non-anomalous data. In contrast, reconstruction algorithms focus on accurately reconstructing anomaly-free data, identifying anomalies by detecting incorrect reconstructions. A visualization of the differences between the three approaches is shown in Figure 2.3.

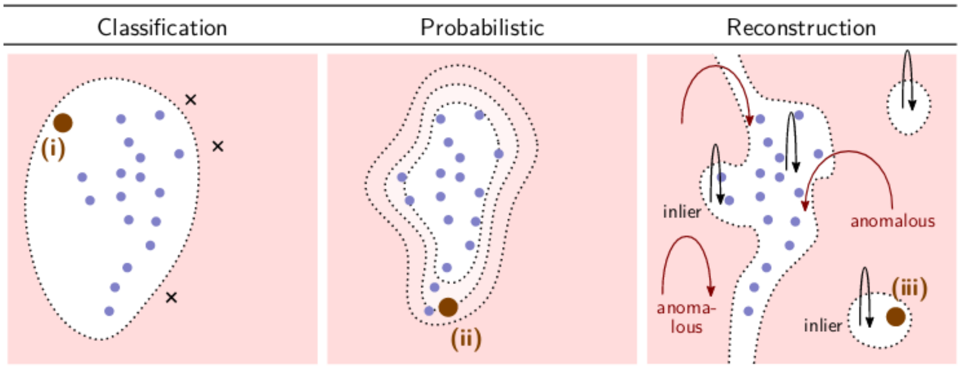


Figure 2.3: Classification, probabilistic, and reconstruction decision functions from [5]. The white corresponds to the non-anomalous data and the red region to the anomalous data, x points are anomalies. Classification methods use a discriminative boundary, probabilistic methods model a density, and reconstruction methods model some underlying geometric structure of the data. Query points (i)-(iii) are non-anomalous data points.

2.2.2 Metrics

A metric is a quantifiable measure used to assess performance on a task, reflecting how well (or poorly) the result meets specified objectives. Several metrics exist for classification problems, but in anomaly detection, the focus is specifically on the binary classification between non-anomalous and anomalous classes. Metrics can be applied at either the image level or the pixel level. An image is classified as anomalous if it contains a sufficient number of anomalous pixels, with the threshold for abnormality varying by method and application. Image-level metrics evaluate whether the entire image is correctly classified as normal or anomalous. In contrast, pixel-level metrics assess whether individual pixels are correctly labelled, thus evaluating both image classification and anomaly localization. Selecting an appropriate metric for anomaly detection can be difficult, particularly when the normal class significantly outnumbers the anomalous samples.

In this research, two binary classification metrics are used. First, the area under receiver operating characteristic (AUROC) score is applied at both the image and pixel levels. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for different classification thresholds. It illustrates the trade-off between sensitivity (TPR) and specificity (FPR) of a binary classifier. The AUROC score quantifies this curve, where a score of 0.5 indicates random classification (50/50 chance of getting the classification right) and a score of 1.0 represents a perfect classifier. Sensitivity, and thus AUROC, is valuable in anomaly detection as it measures how well the model identifies true positives. This metric helps evaluate the model's effectiveness in detecting positive (anomalous) samples.

The second anomaly detection metric used in this research is the average precision (AP) score, also known as the area under precision recall (AUPR). The PR curve defines the relationship between the precision of a binary classifier against the recall at various classification thresholds. The AP score is the area under this curve where the closer the score is to 1.0 the better the classifier is. The precision of the model is the number of true positive predictions, divided by the total number of positive predictions.

In the context of anomaly detection, the interpretation of both AUROC and AP depends on the dataset and use case where they are applied. AUROC is more sensitive to class imbalance than AP because AUROC is dependent on the true negative rate while PR looks at the classifier's performance on the positive class only. Since each metric's output can vary with class imbalance, their interpretation depends on the application and dataset balance. The use and interpretation of these metrics in this thesis are discussed in Section 5.1.2.

2.3 Image Pose Alignment

In the later chapters of this thesis, we perform anomaly detection by directly comparing two similar images. We search for anomalies by generating a reference image that shows what an anomaly-free image looks like and identify anomalies by finding differences between the query image and the reference image. For accurate detection of anomalies, both images must be captured from approximately the same camera pose. To ensure proper alignment of the images, a camera pose estimation step, also referred to as an image alignment step, is conducted before the anomaly detection analysis.

Image alignment, also known as image registration, involves finding a camera pose transformation to map one image to another, ensuring their spatial correspondences match [22]. The goal of image alignment is to adjust the camera pose so that the image aligns as closely as possible with another image at every pixel location. The anomaly detection algorithm presented in Chapter 6 assumes that image alignment has been completed beforehand and that both images were taken from approximately the same camera pose. In this section, we describe the perspective-n-point (PnP) problem, and how it can be solved to align to images.

The PnP problem involves determining the pose of a camera from a set of 2D point projections [23]. This problem is fundamental for camera calibration and multiple methods exist to solve it. Solutions to PnP utilize 3D-2D correspondences, where each correspondence consists of a point's known 3D position relative to the camera and its pixel location in the image. Using these correspondences, a set of linear equations can be formed using the camera matrix form of perspective projection:

$$x_i = \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}}, \tag{2.2}$$

$$y_i = \frac{p_{10}X_i + p_{21}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Z_i + p_{23}}, \tag{2.3}$$

where (x_i, y_i) are the 2D pixel locations and (X_i, Y_i, Z_i) are the corresponding known 3D position for correspondence i, and P is the camera matrix. The intrinsic camera matrix, K, is assumed to be known,

$$P = K[R|t] = \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \end{bmatrix}. \tag{2.4}$$

When at least three correspondences are available, the camera pose can be found by solving the equations above using, for example, the direct linear transform [24]. The PnP problem can be extended to image alignment by matching 2D points from one image to another using a feature matcher. Once 3D-2D correspondences are established between two images, a non-linear least squares optimization can be employed to determine the pose transformation. This optimization minimizes the reprojection error between matched points,

$$\mathbf{e_i} = \tilde{\mathbf{x}}_\mathbf{i} - \mathbf{x_i}, \tag{2.5}$$

where $\tilde{\mathbf{x}}_\mathbf{i}$ are the coordinates of point i in the matched image, and $\mathbf{x_i}$ are the projected coordinates of the known 3D points in the original image, determined using the calculated pose. By solving this optimization and finding the camera pose transformation between two images, the pose of one image can be aligned with the pose of the other.

Chapter 3

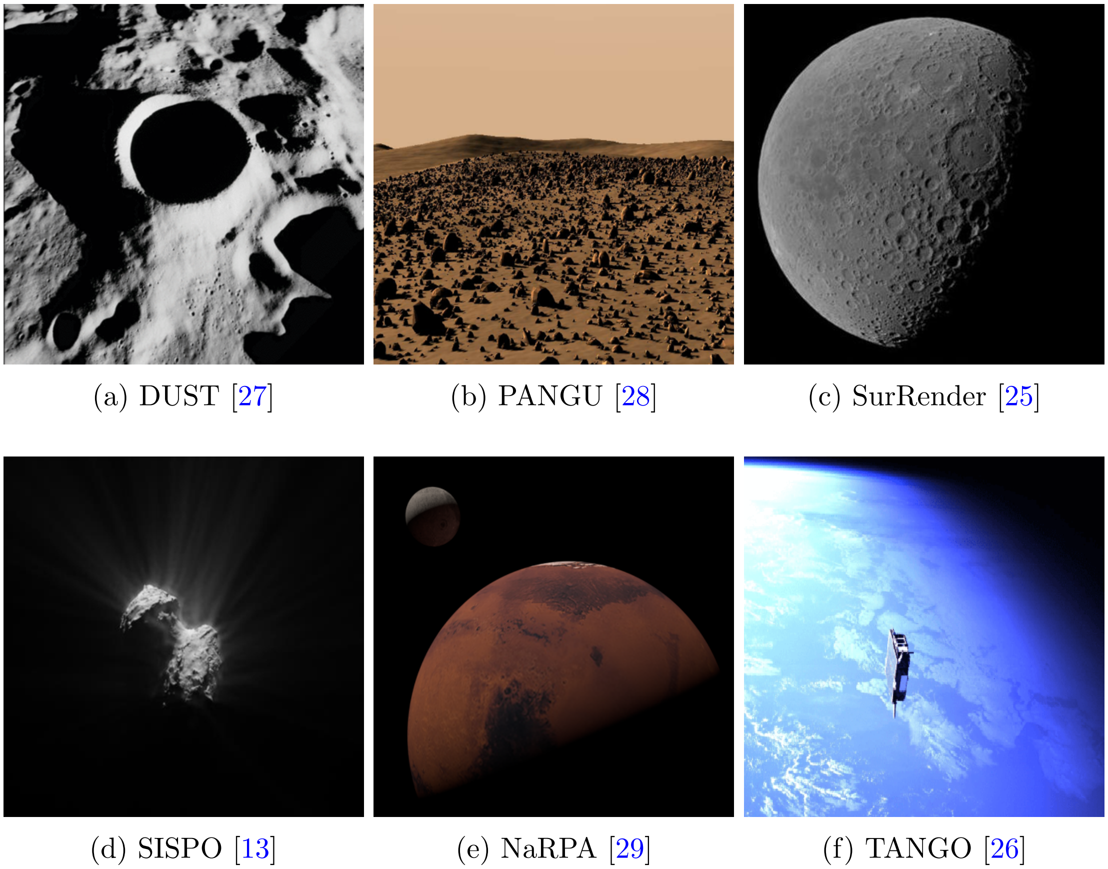
Related Work

This thesis introduces a novel dataset and solution for anomaly detection in challenging space environments. In this chapter, we first provide an overview of existing vision-based space exploration datasets. We detail how these datasets were acquired and what task or mission they were created for. Next, we present an overview of current approaches to anomaly detection, reviewing both learned and traditional anomaly detection algorithms. Finally, we describe existing datasets that were specifically designed for, or can be used for, anomaly detection.

3.1 Datasets for Vision-Based Space Exploration

Vision-based algorithms are increasingly relied upon in space exploration missions, including planetary landings and on-orbit operations, due to their effectiveness in enabling autonomous navigation. Developing and validating these algorithms requires accurate and representative images of the anticipated deployment scenarios. However, data from past missions is usually not suitable because the available images may not cover the same domain, lack ground truth references, or fail to match the specific sensor requirements [25] [26]. Additionally, acquiring new images is often very difficult (e.g., Earth) or impossible in many cases (e.g., from space). As a result, computer-generated images are frequently used to prototype and develop vision-based algorithms, as they allow for the creation of a wide range of possible scenarios. These artificially-rendered images provide accurate details and ground truth reference information, which are essential for development and evaluation.

Computer-generated images are created using rendering engines that perform path tracing calculations to accurately simulate light and shadows. We explain how rendering engines create simulated images in Section 2.1. Multiple image rendering programs have



(a) DUST [27]

(b) PANGU [28]

(c) SurRender [25]

(d) SISPO [13]

(e) NaRPA [29]

(f) TANGO [26]

Figure 3.1: Images from existing space datasets.

been developed to generate synthetic datasets for past and future space exploration missions. These programs use rendering software such as Blender [14] and Unreal Engine 5 [15] to accurately replicate the scenes they expect to encounter during their respective missions. Sample images from some existing rendering programs or datasets are shown in Figure 3.1.

NASA developed the DLES Unreal Simulation Tool (DUST) [27] to generate scenes of the lunar south pole in support of the Artemis missions. DUST uses lunar terrain information from the Digital Lunar Exploration Sites (DLES) dataset and Unreal Engine 5 for rendering. Additional tools were incorporated into DUST simulations to deal with lunar rock placement and manipulation, celestial body positioning, and the visualization of specific lighting conditions. Similarly, the ESA developed Planet and Asteroid Natural Scene Generation Utility (PANGU) [28] to create images of planetary surfaces. Unlike DUST which focused on supporting the development of lunar surface missions, PANGU focused on generating images geared towards supporting the development of planetary landers. PANGU generates images of a celestial body's surface as seen from a spacecraft descending towards the body; its main focus is on accurately generating craters such as those found on the Moon and Mars.

Airbus developed the image rendering software SurRender [25] to create images for vision-based algorithms used during on-orbit missions. This software package is designed for applications such as orbital maneuvering during space debris removal and satellite servicing tasks. SurRender uses its own language derived from C to handle modules within the simulation, including the projection model, the geometric and object material properties models, and the light reflectance functions. The rendering program Space Imaging Simulator for Proximity Operations (SISPO) is designed specifically to replicate images of asteroids to support the development of algorithms for fly-by missions [13]. SISPO leverages the Blender rendering engine to calculate the illumination of the simulated scene. Specifically, the software aims to accurately simulate gas and dust clouds around asteroids.

Some rendering programs incorporate additional sensor data with the rendered images to extend possible applications. The rendering software Navigation and Rendering Pipeline for Astronautics (NaRPA) was developed to create imagery within the context of spacecraft navigation around celestial bodies other than Earth [29]. NaRPA focuses on incorporating additional data such as LIDAR point cloud data and stereo depth data into the generated dataset. An atmospheric modelling procedure is also applied to simulate how a planet's atmosphere scatters sunlight. The TANGO dataset [26] used a CAD model of the TANGO spacecraft from the 2010 PRISMA satellite project for autonomous formation flying to create a synthetic pose estimation dataset. This dataset consists of synthetically-rendered images of the spacecraft along with the ground truth poses.

Current methods of generating image data for vision-based algorithms in the space domain are fully bespoke and mission-specific, making the images less relevant (or, often, unusable) for other applications. In addition, the rendering programs themselves are usually not open source and hence cannot be adapted to other use cases. Therefore, a new rendering process is required to create a dataset for the task of worksite surveying by the Canadarm3.

One significant concern that arises with using rendered images in the development and testing of vision-based algorithms is the simulation-to-real world (Sim2Real) gap. This is the gap between the appearance of the generated, synthetic images and the true appearance in the target domain. The gap must be overcome for an algorithm to perform well in the target domain. One approach, suggested in [30], is to replicate some of the synthetic images through terrestrial experiments and use this limited real data to demonstrate the algorithm's performance in the target domain.

3.2 Anomaly Detection

Anomaly detection, whose fundamental principles are described in Section 2.2.1, involves identifying data points that deviate from an expected distribution. In this section, we review existing approaches to anomaly detection and available datasets. Both traditional and deep learning methods for anomaly detection aim to distinguish between anomalous and anomaly-free data by analyzing only the latter. However, they differ in the way that they handle anomaly-free images. Traditional methods use a cost function to optimize a decision boundary around the non-anomalous images or generate a simplified model of the expected data. Deep learning methods leverage neural networks to learn and manipulate the distribution of non-anomalous images.

3.2.1 Traditional Methods for Anomaly Detection

We define traditional methods of anomaly detection as those that do not use neural networks or any form of deep learning. Most traditional methods fall into one of two categories [5]:

1. Density estimation: These methods use principles of density estimation or direct boundary estimation to define anomaly-free data. Kernel-based One-Class Support Vector Machine (OC-SVM), Support Vector Data Descriptor (SVDD) and nearest-neighbour methods fall into this category.

2. Reconstruction: These methods aim to produce a simplified model of the data and use reconstruction error as a measure of abnormality. Examples include Principal Component Analysis (PCA) and kernel-PCA methods.

Kernel-based OC-SVM and SVDD are methods of one-class classification that learn a decision boundary around the anomaly-free data. This boundary may correspond to a desired density level of non-anomalous data or to a minimized number of false alarms (false positives) and missed anomalies (false negatives). Both methods use a kernel function to map features to a higher-dimensional kernel feature space, addressing the multi-modal, non-linear, and non-convex properties of anomaly detection datasets. The origin in the kernel space is then taken as the sole point in the anomalous class and serves as the starting point for defining the boundary between normal and anomalous classes. OC-SVM optimizes the decision boundary by maximizing the margin separating the mapped vectors from the origin, fitting a hyperplane that is as close as possible to the data points while separating them from the origin [31] [32]. In contrast, SVDD minimizes a hypersphere to enclose the anomaly-free data, ensuring that the sphere's volume is as small as possible [33].

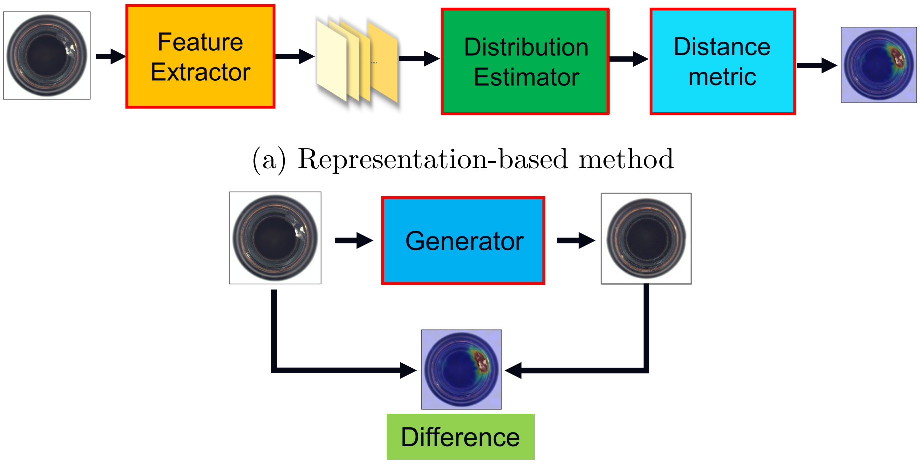
Density estimation is a technique used to estimate the probability distribution function of a random variable. Parametric density estimators capture data behaviour by either fitting a multivariate Gaussian distribution or evaluating the Mahalanobis distance between a test point and the mean of the training data. More complex distributions can be modelled using non-parametric density estimators, such as kernel density estimators (KDE), histogram estimators, and Gaussian mixture models. However, while these non-parametric estimators perform well for low-dimensional problems, they struggle with higher dimensions because the required sample size increases with the feature space dimension [5]. Overall, density estimation methods are limited by the need for a large number of training samples to obtain a reasonable probability density distribution and the fact that real image data rarely follows all the assumptions of a simple parametric distribution [34].

A specific application of density estimation anomaly detection is in hyperspectral imagery. In hyperspectral anomaly detection, an algorithm searches for anomalies in aerial images under the assumptions of a relatively homogeneous background. The Reed-Xiaoli detector (RXD) [35] is a widely used method for unsupervised hyperspectral anomaly detection. RXD uses a multivariate Gaussian distribution as a probabilistic distribution model to represent the non-anomalous data and identifies anomalies as probabilistic outliers. Since the assumption that the background data will follow a multivariate Gaussian distribution does not always hold, for example, if the background is more complex than just homogeneous, multiple extensions of RXD exist. In [36] the pixels in a test image are separated into a target a set and a background set; the probability of a pixel belonging to either set is used to calculate the pixel's anomaly score. The random selection-based anomaly detector (RSAD) [37] iteratively redefines the background set to establish more accurate background statistics.

Reconstruction-based, traditional anomaly detection algorithms primarily use principal component analysis (PCA). The goal of PCA anomaly detection is to generate a simplified model of the non-anomalous distribution by first transforming the data to a feature space and computing the PCA mapping of the data in the feature space [33]. PCA finds the orthonormal subspace that best captures the variance of the non-anomalous data; these vectors form the basis vectors of the mapped data and are used to map test images. The reconstruction error between the original data point and the mapped point is used as an anomaly score. The fundamental idea of this approach is that only anomaly-free images will be properly reconstructed and thus anomalies can identified as points with large a reconstruction error [32].

3.2.2 Deep Anomaly Detection

In recent years, many anomaly detection algorithms have used deep learning techniques since neural networks are well suited to address the challenges surrounding anomaly ambiguity [19]. The majority of learning-based methods are unsupervised [38] [39] [40] due to the absence of representative anomalous data for supervision, although self-supervised [41] [42] methods also exist. Unsupervised methods learn only from anomaly-free images during training; at inference, an image is classified as anomalous if it deviates from the learned non-anomalous distribution. In contrast, self-supervised methods synthetically create anomalous images during training by adding random noise or patches to anomaly-free images. Overall, learned anomaly detection algorithms fall into two categories: representation-based methods and reconstruction-based methods. The methodology for both types is illustrated in Figure 3.2.



(a) Representation-based method

(b) Reconstruction-based method

Figure 3.2: General methodology of reconstruction-based and representation-based anomaly detection from [6].

Representation-Based Methods

Representation-based methods utilize embeddings from pre-trained feature extractors, such as ResNet [43] or ViT [44], and integrate them with an outlier detection framework [45]. As seen in Figure 3.2a, these methods first use a feature extractor to obtain distinct features from the normal images. The distribution of these features is then modelled using techniques like normalizing flow or probabilistic models. Anomalies are detected as features in a query image that are sufficiently different from the learned distribution of normal features [6].

Probabilistic anomaly detection algorithms use non-parametric methods to fit the extracted features to a statistical distribution. PaDiM [38] employs a locally constrained bag-of-features approach using a pre-trained CNN to concatenate activation vectors [45]. The algorithm then performs dimensionality reduction through random selection and fits a multivariate Gaussian distribution to each extracted vector. Anomalies are identified by thresholding the Mahalanobis distance between the features in a test patch and the corresponding Gaussian distribution. DFM [46] also follows a bag-of-features approach but performs dimensionality reduction using PCA instead of random selection. The distribution of non-anomalous features is established by fitting a Gaussian mixture model to a subset of extracted features, and anomalies are identified by calculating the log-likelihood that a feature falls within the fitted distribution. Alternatively, both PatchCore [45] and CFA [40] use memory banks to store information representative of the normal class. PatchCore extracts and stores patch-level intermediate features using a network pre-trained on ImageNet. CFA performs transfer learning on the target dataset to learn features with a high density around the features of a pre-trained CNN. Both methods score anomalies using a distance metric between a test feature and features in the memory bank.

Some representation methods utilize normalizing flows to transform normal features into a specified, tractable distribution [47] [48]. Normalizing flows are generative neural networks that learn a mapping transformation between two probability distributions in a bijective, fully invertible manner, preserving dimensionality [49]. Anomaly detectors use normalizing flows to estimate the density of anomaly-free features. For instance, C-Flow [50] employs a discriminatively trained feature extractor to map image patches into feature vectors. These features are then mapped to a multivariate Gaussian distribution using a general conditional normalizing flow framework. The anomaly score is calculated based on the Mahalanobis distance between a test vector and the corresponding patch distribution. FastFlow [51] extends normalizing flows to two dimensions using fully convolutional networks. Anomalies are classified by evaluating the likelihood that features falls within the distribution mapped by the normalizing flow. U-Flow [44] adopts a U-shaped architecture, where the encoder is a feature extractor and the decoder is a normalizing flow. The anomaly score is determined by both the feature likelihood and the number of false alarms computed using an a-contrario framework.

Reconstruction-Based Methods

Reconstruction-based anomaly detection methods use generative models to reconstruct a query image and identify anomalies based on the reconstruction error between the original and reconstructed images. These methods operate under the assumption that anomalous features will be poorly reconstructed since they were not encountered during training [6]. Various types image generators can be employed, including generative adversarial networks (GANs) [52][53], auto-encoders [54][55], and student-teacher models [56][39].

Student-teacher methods use transfer learning between a teacher model and a student model to learn how to reconstruct anomaly-free images, ensuring that the student network cannot effectively reconstruct or generate anomalous images. The architectures of the teacher and student networks vary by method. For example, STFPM [56] uses the same architecture for both networks, while Reverse Distillation [39] employs a student network with the reverse architecture of the teacher.

Reconstruction-based methods that use GANs adversarially train an auto-encoder such that only non-anomalous images can be accurately reconstructed. In [57] a reconstruction network is optimized to reconstruct images from the normal class, while a discriminator network learns to classify these reconstructed images. The reconstruction network progressively incorporates discriminative features to fool the discriminator. Thus, the discriminator learns to differentiate between non-anomalous and incorrectly reconstructed images, while the reconstruction network focuses solely on normal images. Ganomaly [52] extends this approach by including an additional encoder sub-network to map the reconstructed image to its latent space. Adversarial training occurs during both the reconstruction and latent space mapping stages. The latent space representation of the reconstructed image is then classified as anomalous using a discriminator network.

Some methods introduce anomalies during the training process to improve performance. Both DRAEM [41] and DSR [58] enhance the robustness of the reconstruction process by training on synthetically generated anomalous data. Synthetic anomalies are created by adding random Perlin noise to normal images; DRAEM incorporates these anomalies into the image space, while DSR introduces them into the image feature space. The approach of using synthetic anomalies is further developed in [59] with the introduction of Collaborative Discrepancy Optimization (CDO). CDO uses margin and overlap optimization modules during training to establish distinct normal and anomalous distributions. These modules aim to help the model learn features that maximize the margin (the average distance between distributions) and minimize the overlap (the percentage of shared features between distributions).

Example architectures of both traditional and deep learning anomaly detection methods are shown in Figure 3.3. Classification, probabilistic, and reconstruction examples are shown for both traditional and deep methods highlighting how neural networks are used to handle data distributions.

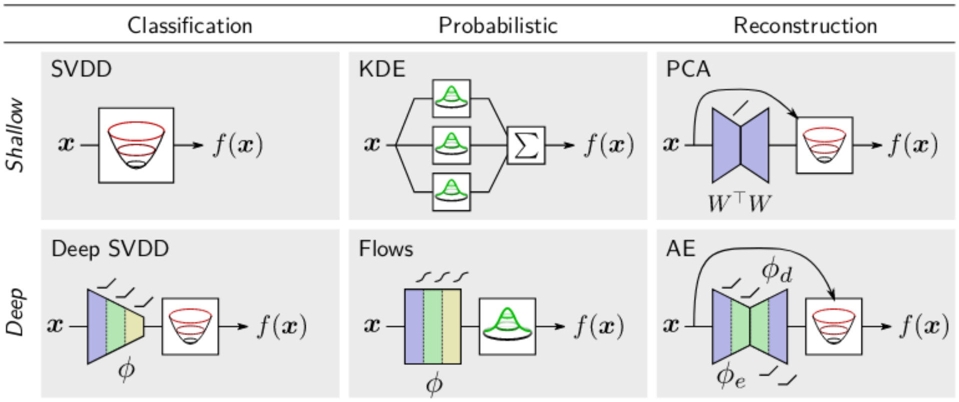


Figure 3.3: Examples of algorithm design architectures for both shallow and deep methods, from [5].

Zero-Shot Anomaly Detection The previously described methods require training for each subject of interest, which can be time-consuming and demands substantial amounts of labelled data. These obstacles have led to the development of zero-shot anomaly detection, a recent approach that aims to perform anomaly detection without task-specific training. Instead, decisions are made based solely on the test image. WinCLIP [60], for example, leverages the pre-trained vision-language model CLIP [61] to execute zero-shot anomaly detection. In the absence of training, the algorithm lacks direct context for distinguishing between non-anomalous and anomalous, so it relies on user-provided text prompts to clarify the distinction between the two classes. These prompts enable the vision-language model to extract relevant information from the image. Although zero-shot anomaly detection does not yet match the performance of fully trained algorithms on benchmark datasets, its independence from training makes it an emerging area of interest.

3.2.3 Anomaly Detection Datasets

Few anomaly detection datasets currently exist, and due to the specific nature of the anomaly detection problem, each dataset is tailored to a particular application. The NanoTWICE dataset [62] comprises grayscale images of nanofibrous materials captured by a scanning electron microscope. NanoTWICE includes 40 anomaly-free images and 5 anomalous images that contain defects like dust flecks. Another dataset for anomaly

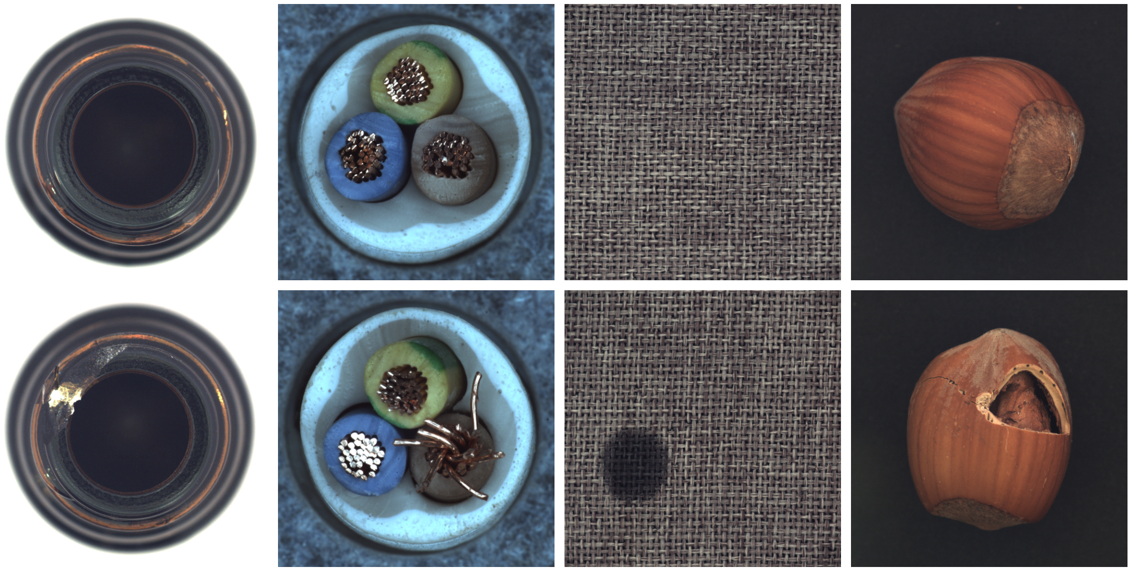


Figure 3.4: Images from MVTec Dataset [7]. Normal images are shown in the top row and corresponding anomalous images in the bottom row.

detection across multiple types of textures was introduced in [63]. Each class in this dataset has 1,000 normal images and 150 anomalous images, and anomalies are colourful ellipses that were synthetically added to the images. Both datasets emphasize the textures within the images and offer limited variety in terms of image content.

The current state-of-the-art dataset for evaluating and benchmarking anomaly detection algorithms is the MVTec 2D AD dataset [7]. This dataset is designed for industrial quality control and includes images of both objects and texture patches. The MVTec dataset contains 3,629 anomaly-free training images and 1,725 test images, which include both normal and anomalous examples. Additionally, all anomalous images have a pixel-level ground truth segmentation map, enabling algorithms to learn to segment anomalies. Each category contains multiple different types of anomalies, all of which are structural. The large number and variety of images, along with the pixel-level ground truth, make this dataset useful for anomaly segmentation in industrial inspection. However, the MVTec dataset assumes that all images are captured from approximately the same point of view and under consistent diffuse lighting conditions. These assumptions can be seen in the examples images from the MVTec dataset shown in Figure 3.4.

An alternative approach to developing an anomaly detection dataset involves adapting existing classification datasets to this task. Large classification datasets such as ImageNet [21] or CIFAR10 [64] can be reformulated for anomaly detection by relabelling an arbitrary set of classes as anomalous [7]. In this approach, the relabelled dataset is used to train an anomaly detection algorithm, where the remaining classes serve as non-anomalous training data. An algorithm's performance is then evaluated based on its ability to identify the outlier classes as anomalous. While using existing classification datasets provides access to a large amount of data, the diversity of the data presents some issues. The wide variety of classes in classification datasets results in significant differences between the features of each class. Consequently, training and testing on these relabelled classes may make it easier for an algorithm to detect anomalies, given the inherent disparity between the anomalous classes and the training data. This raises concerns about how well an algorithm trained on such data can generalize to new data where anomalies are more similar to the anomaly-free distribution.

Chapter 4

The ALLO Dataset

Due to the lack of applicable data, we created the ALLO dataset to evaluate the capabilities of anomaly detection algorithms in the space domain. This dataset was synthetically rendered and specifically designed for developing vision-based anomaly detection algorithms for proximity operations of robots in lunar orbit. This chapter describes the creation of the ALLO dataset, including the modelling of the station's orbit and the simulation of sunlight. We then discuss how the dataset was validated to verify that the synthetically generated images were representative of the real images that will be taken by the Canadarm3. Finally, we explain how the dataset was extended to render an anomaly-free reference image for every query image. These reference images are used by the anomaly detection algorithm presented in Chapter 6, where anomalies are identified by comparing the query image to the reference image and detecting differences between the two.

4.1 Dataset Description

The images captured by Canadarm3 are expected to be taken under a wide range of lighting conditions due to the combination of space's black background, the intense direct sunlight, and inspection lights from the arm. These significant lighting variations are expected to pose a major challenge to the anomaly detection process, making it crucial to replicate these conditions as accurately as possible.

Blender was chosen to create the ALLO dataset because of its advanced rendering capabilities and its Cycles render engine. Using Blender, the operation of the Gateway around the moon was simulated, with the Earth and sun in the background. NASA's Blender model of the ISS [8], shown in Figure 4.1, was used instead of the Gateway, as an accurate and photo-realistic Gateway model is not yet available. The Gateway is expected to resemble the ISS, so the textures and structure of the ISS model are representative of what the Canadarm3's cameras are likely to capture. The intensity of the sun's light in the Blender model was estimated to ensure the resulting images resembled those taken by cameras on the Artemis 1 mission [65].

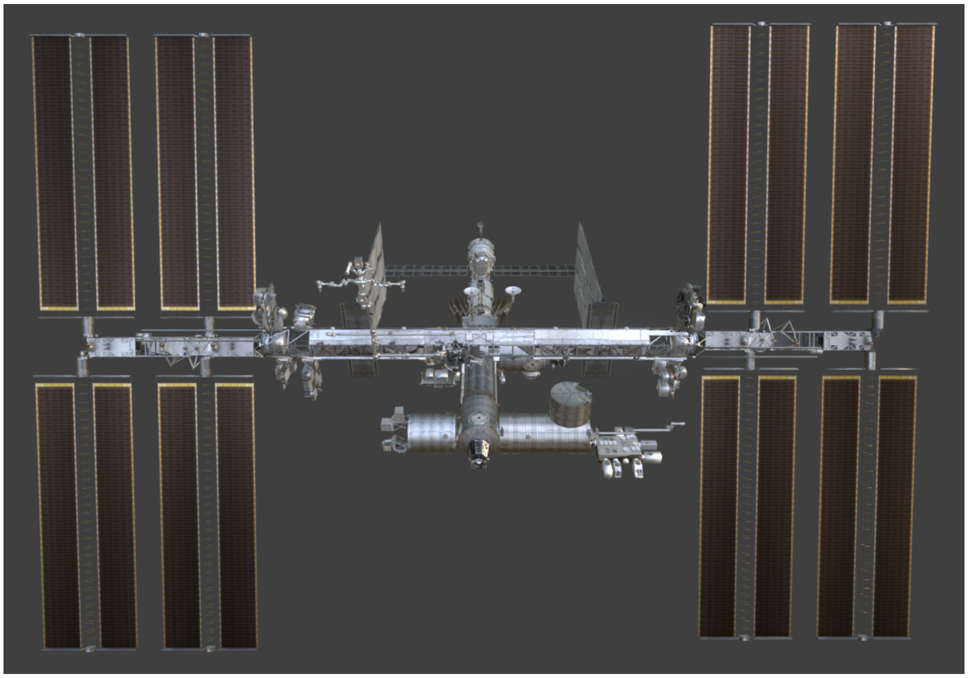


Figure 4.1: Blender model of the ISS [8].

The Gateway will orbit the moon in a 9:2 lunar synodic near-rectilinear halo orbit (NRHO) [66]. This orbit resembles an elongated rectangle, with the moon positioned near the top, and the Gateway completing nine orbits around the moon for every two orbits of the moon around the Earth. The station's NRHO was approximated in the Blender model as an ellipse, with dimensions corresponding to the perilune and apolune of the orbit. The positions of the Earth and sun relative to the moon were then calculated with the Skyfield library [67] using ephemeris data from the year 2030. The positions of the moon, Earth, and sun relative to the station were then simulated over 365 days, replicating the lighting and background conditions that the arm's cameras will experience during operation.

The cameras in the Blender model were positioned around the station to simulate the key locations of the arm-mounted cameras. Fifty camera poses were manually specified as starting positions for placing cameras in the Blender scene. During the rendering process, random Gaussian noise with a standard deviation of 1m was added to each camera's pose to introduce variety to the views. This approach, combined with multiple camera positions and the added noise, ensured that all regions of the station were covered

Camera Parameter	**Value**
Camera Type	Perspective
Focal Length	25 mm
Horizontal Sensor Size	36 mm
Vertical Sensor Size	24 mm

Table 4.1: Blender camera parameters used to render the ALLO dataset.

and that the views varied. Thus, the ALLO dataset replicates the wide range of images expected to be captured by the Canadarm3. The station model and some of the reference camera positions are shown in Figure 4.2 and the Blender camera parameters are listed in Table 4.1.

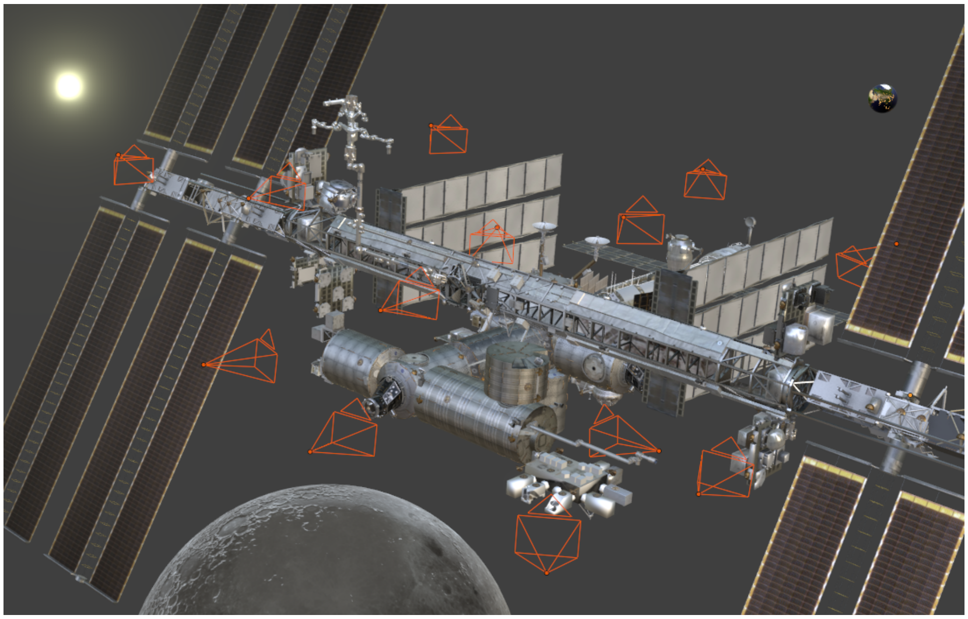
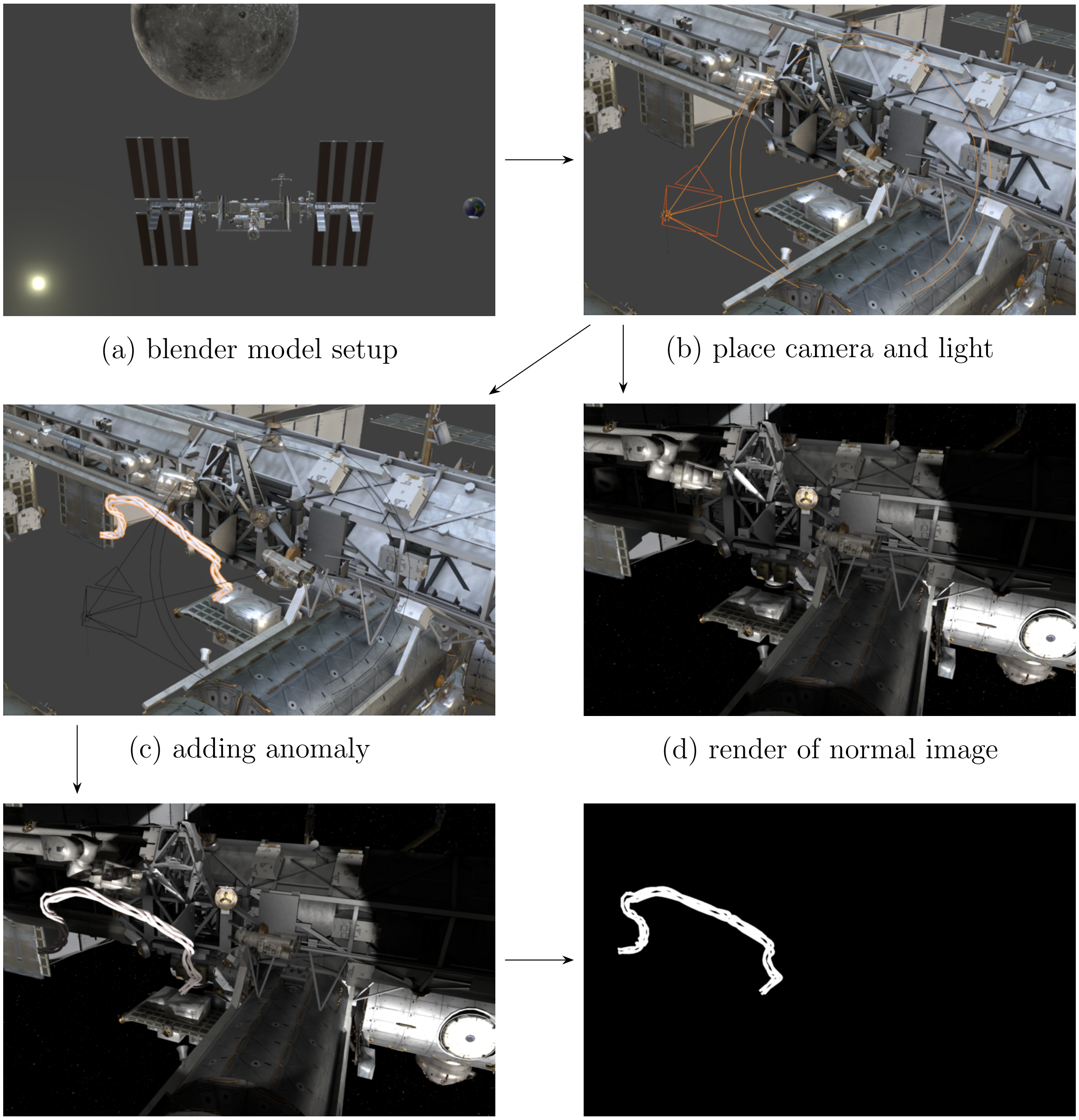


Figure 4.2: Blender model with example camera positions (highlighted in orange) around station.

To create a diverse and comprehensive dataset, certain scene parameters were altered during rendering to prevent overfitting and improve the generalizability of models trained on the dataset. For both the normal and anomalous image sets, the sunlight illumination was varied so that each view was rendered under three different intensity levels. During the creation of anomalous images, the anomaly's depth relative to the camera, its scale, and its colour were randomly adjusted. The ALLO dataset was rendered according to the steps below and an example process is shown in Figure 4.3.

1. The positions of the moon, Earth, and sun were placed relative to the station



(a) blender model setup

(b) place camera and light

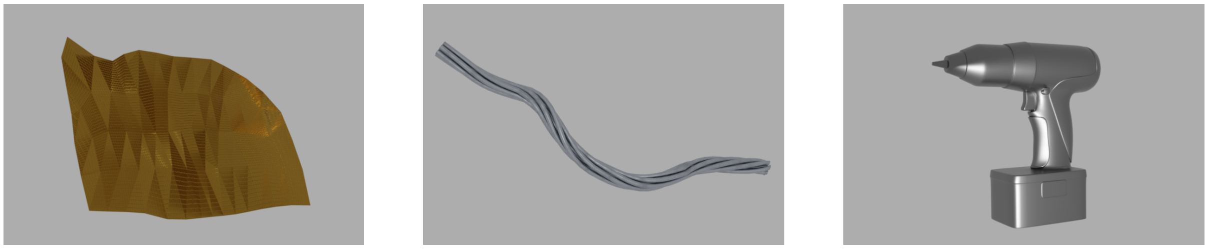
(c) adding anomaly

(d) render of normal image

(e) render of anomalous image

(f) segmentation mask

Figure 4.3: Rendering process of the ALLO dataset.



(a) Thermal blanket (b) Cable (c) Drill

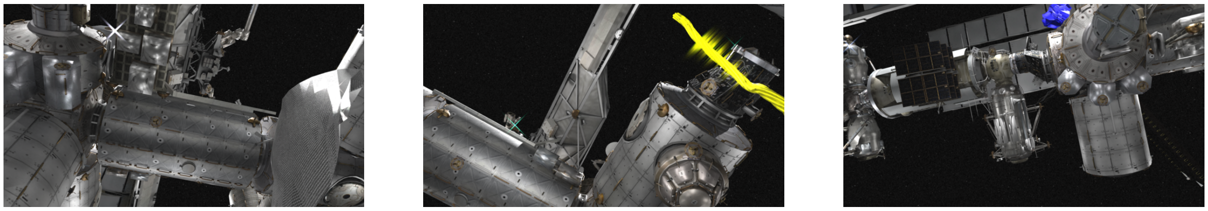
Figure 4.4: Blender models of anomalies used in the ALLO Dataset.

according to the ephemeris and orbital data of that day.

2. A camera and spotlight were placed around the station by selecting one of the pre-specified locations and slightly perturbing the pose.
3. If specified, an anomaly was placed inside the camera's frustum and a segmentation mask was rendered.
4. The test image was rendered using Cycles, and both noise and glare were added in post-processing.
5. Steps 3-5 were repeated with a different sun strength, anomaly position, scale, or colour.
6. Steps 2-5 were repeated with a different camera position.

The training set, consisting of only non-anomalous images, was created using 40 of the 50 possible camera positions. The test set, which included both normal and anomalous images was rendered using the 10 remaining, unseen camera positions. During the rendering of anomalous images, models of thermal blankets, cables, and maintenance tools were used as anomalies; some of these models are shown in Figure 4.4. Each anomalous image contained only one anomaly. This approach was taken because the performance of anomaly detection algorithms on an image with one anomaly is typically reflective of their performance on an image with multiple anomalies.

Given that existing anomaly detection methods rely on feature extraction, it is hypothesized that the colour of an anomaly will heavily impact an algorithm's performance. To test this hypothesis the ALLO test set was divided into two subsets based on the colour of the anomaly in the test images. One subset contained images with 'default' coloured anomalies, such as those resembling thermal blankets and drills, which are typically metallic and gray, as shown Figure 4.4. These anomalies are more similar to the



(a) Default blanket (right) (b) Yellow cable(top right) (c) Blue blanket (top)

Figure 4.5: Anomalous images from the ALLO dataset with colourful anomalies.

Dataset Set	**Training**	**Test**	**Total**
Cameras	1-40	41-50	50
Normal Images	43,799	864	44,663
Anomalous Images	0	13,716	13,716

Table 4.2: Images in training and testing sets of the ALLO dataset.

station in material and appearance. The second subset included images with anomalies rendered in bright colours like red, blue, and yellow, which do not closely resemble the station. This variation allows us to determine how much algorithms depend on the visual features of anomalies. Example images with anomalies of various colours are shown in Figure 4.5.

All images were rendered to a resolution of 1,920 x 1,080 pixels. The breakdown of images into the training and testing set (both default and colourful anomalies) is shown in Table 4.2. Sample images from the ALLO dataset are shown in Figures 4.6 and 4.7. These images demonstrate how the scenes expected to be encountered by the Canadarm3 can be quite crowded, may contain both illuminated and shadowed structures, and have large black portions of space.

4.1.1 Dataset Validation

In this section, we demonstrate that the synthetically generated images accurately represent the scenes expected to be encountered by the Canadarm3. We achieved this by using real images taken by the ISS and replicating them using the Blender model. Two pairs of replicas images are shown in Figure 4.8. As seen in these images, both the lighting and textures of the scenes are precisely captured by the Blender model. The reflection of sunlight on the station and the stark shadows are also accurately rendered. Differences between the real and simulated images arise due to slight discrepancies in the CAD model, such as the absence of some wires on the station and colour differences between the real station and the ISS model.

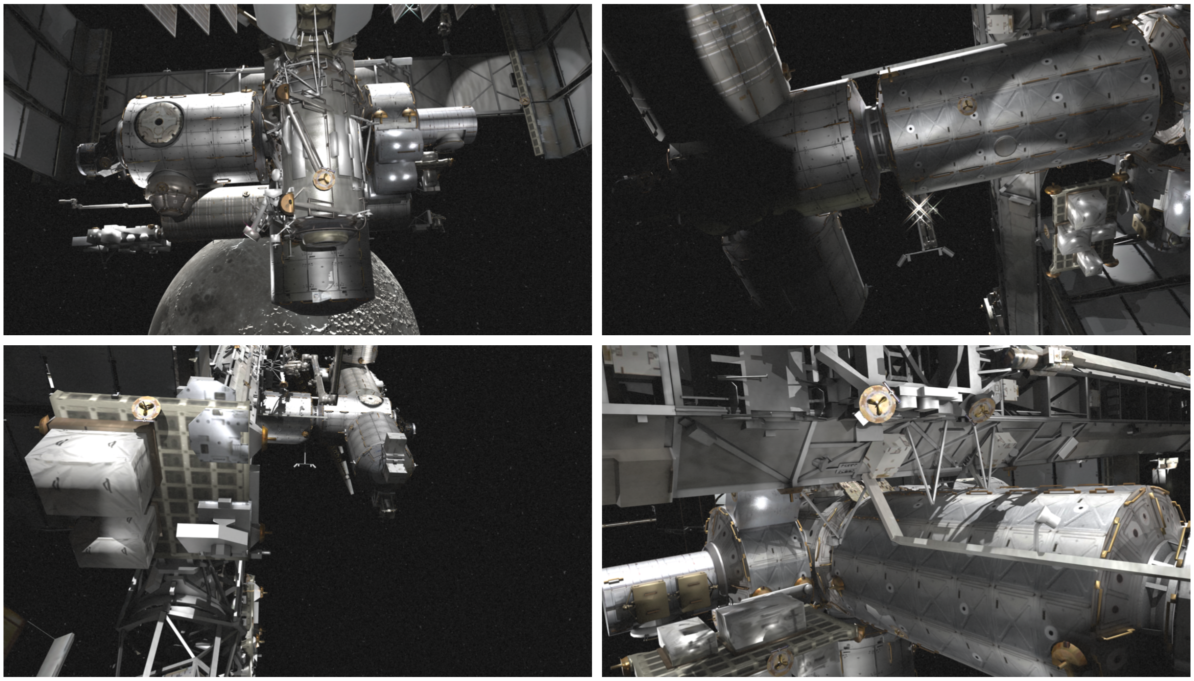


Figure 4.6: Sample non-anomalous images from the ALLO dataset.

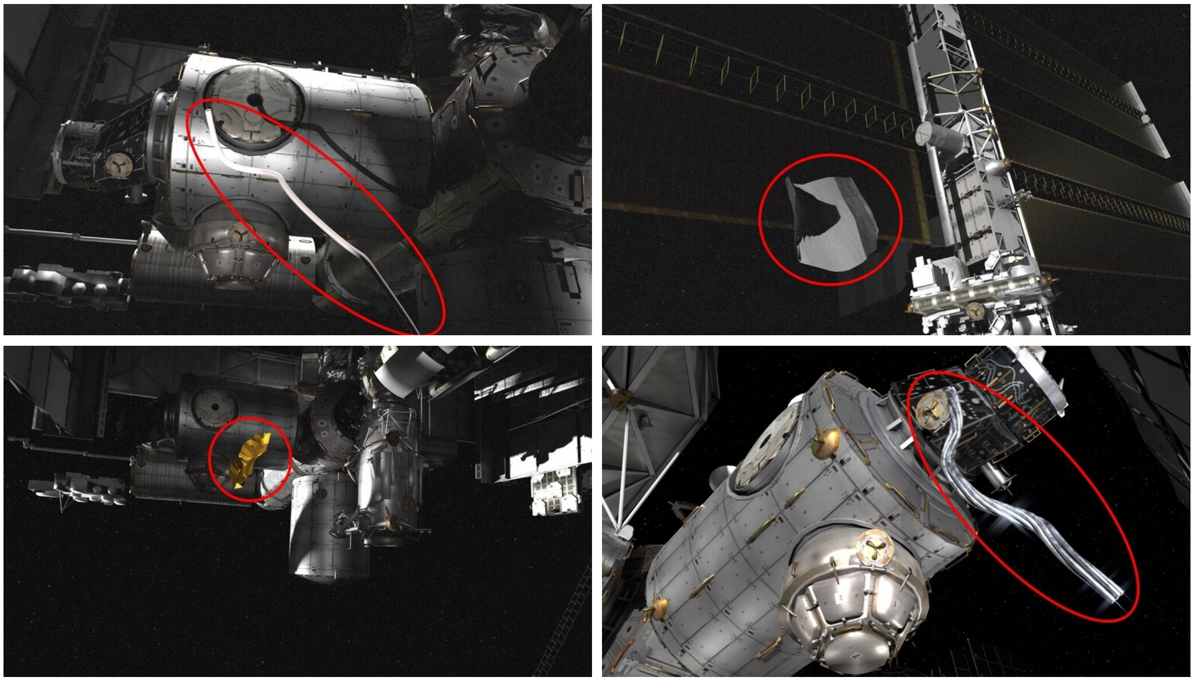


Figure 4.7: Sample anomalous images from the ALLO dataset; anomalies are circled in red.

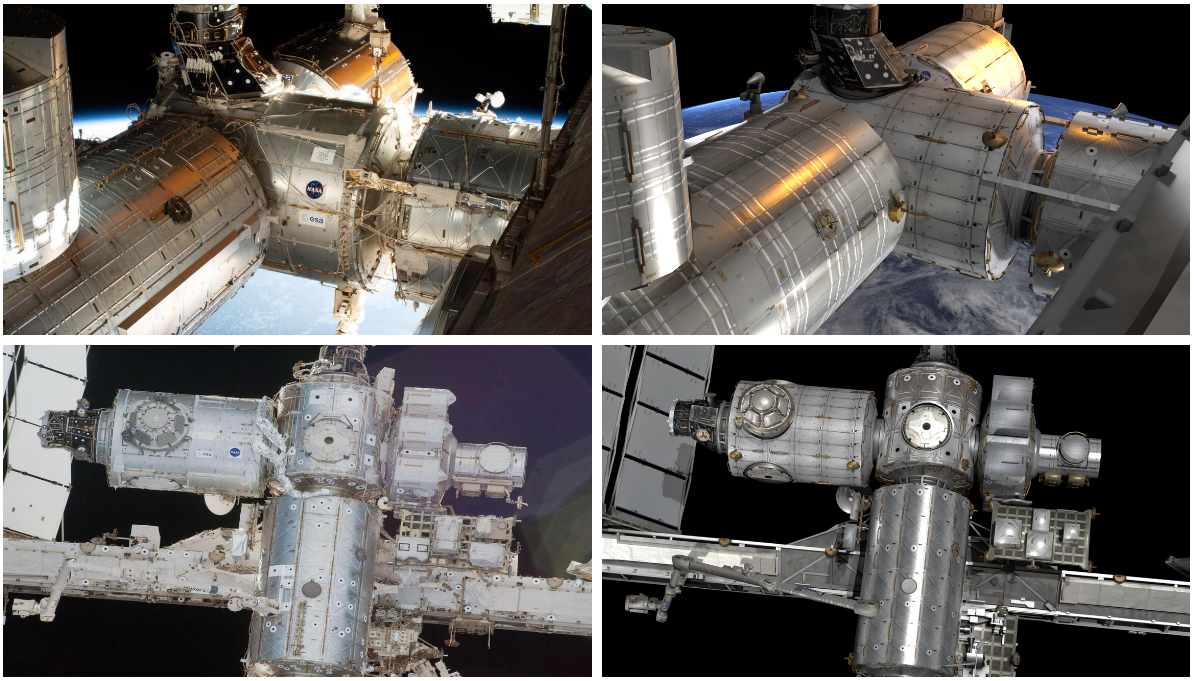
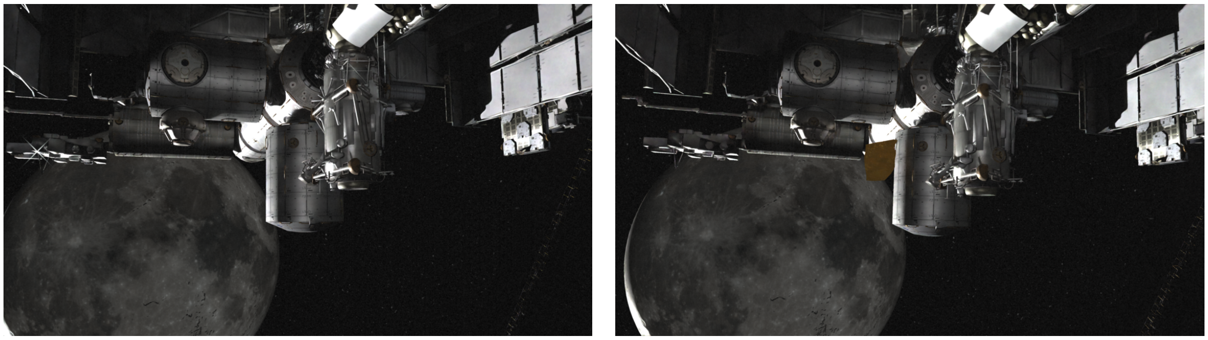


Figure 4.8: Images taken by the ISS are shown in the first column and their replications created using the blender model are shown in the second column.

Seven replica images were compared to real ISS images using the Structural Similarity Index Measure (SSIM) [68]. The average SSIM score was 0.32, indicating a resemblance between the synthetic and real ISS images. The lower score is mainly due to luminescence differences caused by the station's colour; however, the contrast and structure are well-represented.

4.2 Dataset Extension

The ALLO dataset as described above was designed and created for use on existing anomaly detection algorithms, such as those evaluated in Chapter 5. These algorithms are learned methods that train on normal images and test on a mix of normal and anomalous images. However, the anomaly detection algorithm developed in this thesis requires an additional image: the reference image. In Chapter 6, an anomaly detection algorithm is introduced that searches for anomalies by comparing a test image to an anomaly-free reference image. The reference image represents the expected view that the arm should capture during operation. The principle behind using a reference image is that any differences between the query image and the reference image may indicate the presence of an anomaly. In this section, we describe how the ALLO dataset was extended to render a reference image for every query image, both normal and anomalous.



(a) Reference image (b) Anomalous query image

Figure 4.9: Reference image and corresponding query anomalous images from extension of ALLO dataset.

To extend the ALLO dataset, a function was added to the rendering script outlined in the previous section so that for each query image generated (either normal or anomalous), a reference image was also generated. Referring to the step-by-step outline in Section 4.1, the reference image was rendered between steps 4 and 5. The camera's pose was slightly varied during the rendering of the reference image to simulate the measurement error between the camera's theoretical and actual positions, based on the estimated pose error provided by MDA's engineers. No noise was added to the reference image during post-processing. These variations mimic potential false positives that the anomaly detection algorithm must be able to handle. Consequently, the reference image shows almost the same scene but from a slightly different point of view and without noise.

An example reference image is shown in Figure 4.9 alongside the corresponding anomalous image. In the query image, Figure 4.9b, the anomaly is a gold-coloured blanket located at the center of the image. However, there are additional differences between the reference and query images beyond this anomaly. The difference in camera pose, for instance, causes part of the sun's reflection off of the moon's surface to be visible in the lower-left corner of the query image, but not in the reference image. The station components also do not completely overlap, as seen by the differing number of panels visible on the right side of each image. All of these variations could be mistakenly flagged as anomalies if an anomaly detection algorithm was to rely solely on pixel differences between the two images. The reference image offers useful data for anomaly detection that is specific to this application, as it provides the expected view of the camera. However, to utilize this image an anomaly detection algorithm is needed that can differentiate between the above false positives and real anomalies.

Chapter 5

Benchmark Experiments

In this chapter, we use the ALLO dataset to evaluate existing anomaly detection algorithms from Intel's Anomalib repository [69]. As described in Section 3.2, these algorithms are optimized to perform well on the MVTec 2D dataset. We benchmark their performance on the ALLO dataset to assess their ability to generalize to the space domain. We determine which type of algorithm is best suited for anomaly detection in space and identify areas where these algorithms fail. Through our benchmark experiments, we establish the research gap that needs to be addressed to develop a viable anomaly detection algorithm for the Canadarm3.

5.1 Methodology

In this section, we outline the benchmarking process for algorithms from Anomalib on the ALLO dataset. We review the selected algorithms, and detail how they were trained and tested, including any modifications that were made to improve their performance on the ALLO dataset. Finally, we describe the pixel-level and image-level metrics used for our evaluation.

5.1.1 Experimental Setup

Intel's Anomalib repository [69] was used to implement and test existing anomaly detection algorithms on the ALLO dataset. Anomalib was selected because it includes multiple state-of-the-art anomaly detection algorithms, allowing us to comparatively test and tune various anomaly detection methods on our data. The code in the Anomalib repository was modified to train on the ALLO dataset. The following seven algorithms were evaluated: STFPM [56], CFA [40], Reverse Distillation [39], DRAEM [41], FastFlow [51], U-Flow [44], and DSR [58]. Algorithms that require processing the entire dataset at once, including PaDiM [38] and PatchCore [45], were not evaluated since the computational requirements to simultaneously store all pixel values from the entire ALLO dataset was deemed unreasonable. While both PaDiM and PatchCore perform very well on the MVTec dataset, their reliance on an analysis of every pixel in the entire dataset at once limits their applications beyond industrial inspection.

All benchmarked algorithms are state-of-the-art deep learning-based methods. FastFlow, UFlow, and CFA are representation-based methods; FastFlow and UFlow use normalizing flows, while CFA employs a memory bank to learn the distribution of non-anomalous features. STFPM, Reverse Distillation, DSR, and DRAEM are reconstruction-based methods. Specifically, STFPM and Reverse Distillation are student-teacher methods, and DSR and DRAEM are semi-supervised methods that introduce artificial anomalies during training. Additional details on the structure and methodology of each algorithm are described in Section 3.2.2. The evaluated algorithms were trained on non-anomalous data and then tested on the default colour test set described in Section 4.1. During training, 10% of the test set was set aside to be used for validation to enhance model generalization. The pixel AP score of the validation was monitored for early stopping and training was halted if the score did not improve over ten consecutive epochs.

To evaluate performance, each algorithm was initially trained with the original hyperparameters as outlined in the respective, publishing papers. Since all the algorithms were designed and tuned for use on the MVTec dataset, we adjusted the hyperparameters to optimze their performance on the ALLO dataset. First, we applied the following data augmentations before training: horizontal and vertical flipping, and random brightness and contrast adjustments. Each augmentation had a 50% probability of being applied. The random brightness limit was set to 0.2 and the contrast limit was set to 0.1, meaning brightness varied between [0.8, 1.2] and contrast between [0.9, 1.1]. Consequently, half of the training images had at least one augmentation, increasing the diversity of the dataset. Varying brightness and contrast is particularly important for the ALLO dataset, given the wide range of lighting conditions present in the images. An algorithm's performance on this dataset depends on its ability to generalize across various lighting scenarios. Subsequently, custom normalization values were used in place of the ImageNet default values. The default normalization values are the mean and standard deviation of the pixels in the ImageNet dataset; these images differ significantly from those in the ALLO dataset, making the default normalization values inaccurate. The custom normalization values used in the benchmark were calculated on the ALLO training set. The algorithm or parameter set with the highest pixel AP score was considered the best-performing.

The impact of dataset-specific tuning and the results for all algorithms, including how the algorithms' performance was influenced by the colour of anomalies, are detailed in Section 5.2.

5.1.2 Experiment Metrics

Three metrics were used to evaluate the performance of anomaly detection algorithms on the ALLO dataset. Image AUROC and pixel AUROC were chosen due to their effectiveness in evaluating binary classifiers, and their widespread use in the anomaly detection literature [44]. While AUROC is generally reliable at the image level, it can sometimes misrepresent an algorithm's performance at the pixel level. The AUROC score is influenced by an algorithm's true negative rate, which means that the pixel AUROC score can be artificially inflated if the anomaly occupies a relatively small area (approximately less than 10% of the image). This is often the case for anomalies in the ALLO dataset; thus, a high pixel AUROC score does not necessarily indicate that an algorithm is good at detecting anomalies, but rather that it is proficient at identifying normal pixels.

We require a pixel-level metric that can accurately evaluate an algorithm's ability to localize anomalous pixels in an image. Average Precision (AP), detailed in Section 2.2.2, measures the area under the precision-recall curve. AP is particularly valuable for imbalanced datasets, where positive instances are significantly outnumbered by negative ones, because it considers both precision and recall. Given the importance of correctly identifying anomalies in space applications, AP was selected as the primary pixel-level metric. AP's proportional relationship to precision makes it effective in evaluating a model's ability to correctly identify anomalous pixels [50][46]. Pixel AUROC was also included for completeness and comparison with the literature.

5.2 Benchmark Results

Results for all algorithms on the default test sets are shown in Table 5.1. Reverse Distillation had the highest pixel AP at 47.6%, followed by CFA and STFPM at 44.7% and 44.6% respectively. Compared to its performance on the MVTec dataset Reverse Distillation's image AUROC score was 43.1% lower and its pixel AUROC was 41.1% lower on the ALLO dataset. UFlow recorded the highest image AUROC score at 80.9%, although this is still 18.0% lower than its performance on the MVTec dataset. Generally, the student-teacher methods (STFPM and Reverse Distillation) performed best, followed by

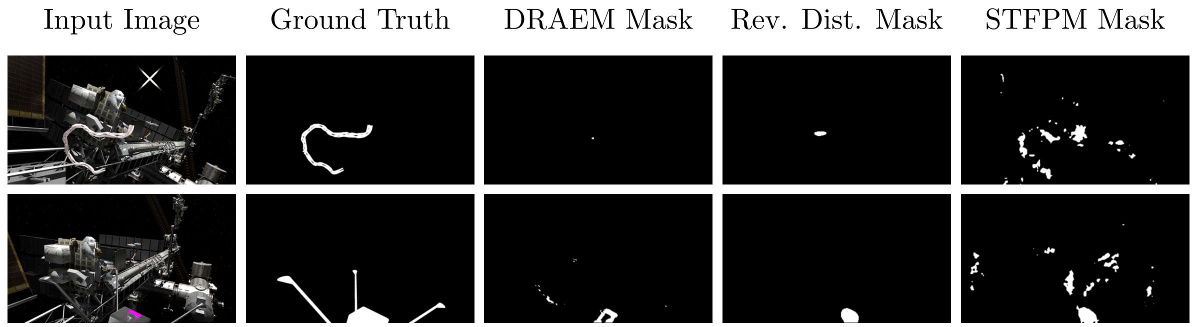


Figure 5.1: Example inference from anomaly detection algorithms.

the normalizing flow methods (FastFlow and UFlow), while the semi-supervised methods (DRAEM and DSR) performed worse. Examples of inference from some of the evaluated methods are displayed in Figure 5.1.

Most algorithms benefited from dataset-specific tuning. Although all algorithms faced challenges in generalizing to the space domain, modifying their hyperparameters, compared to those used for the MVTec dataset, improved the performance of some. Improvements were achieved using the augmentations specified in Section 5.1.1. The effects of data augmentation and custom normalization on the algorithms are shown in Table 5.2. Both CFA and Reverse Distillation, the two best-performing algorithms, performed better with their baseline parameters than with data augmentation, showing a slight decrease in performance with augmentations. In contrast, all other algorithms exhibited marginal to significant performance improvements with augmentations. DSR showed the largest increase in pixel AP with a 10.9% improvement, although it still performed poorly overall. DRAEM had the lowest performance among all algorithms, with a pixel AP of 19.9% even with augmentations. Generally, using custom normalization values instead of the ImageNet default ones had a significant impact on performance. This highlights the substantial differences between the ALLO dataset and commonly-used datasets, making standard ImageNet values unsuitable for this application.

Despite extensive dataset-specific tuning, existing anomaly detection algorithms still struggled in the space domain. These algorithms were all designed with the assumption that the tested images would be very similar in perspective and lighting to the training images. Even with data augmentations and custom normalization values intended to improve performance, the algorithms had difficulty generalizing to the more complex scenes and lighting conditions encountered in space.

Following the hypothesis on anomaly colour presented in Section 4.1, we evaluate how different aspects of an image affect the performance of anomaly detection algorithms in the space domain. We specifically examine how the colour of anomalies impacts the

Type	**Algorithm**	**Image AUROC**	**Pixel AUROC**	**Pixel AP**
Representation	FastFlow [51]	51.0	79.5	20.1
	UFlow [44]	**80.9**	**91.6**	39.2
	CFA [40]	50.9	49.8	44.7
Reconstruction	DRAEM [41]	54.4	69.0	19.9
	DSR [58]	52.2	54.4	29.4
	Rev. Dist. [39]	55.4	56.7	**47.6**
	STFPM [56]	62.8	64.5	44.6

Table 5.1: Performance of state-of-the-art anomaly detection algorithms on the default ALLO dataset. For each metric, the algorithm that performed best is bolded.

Algorithm	**Parameters**	**Image AUROC**	**Pixel AUROC**	**Pixel AP**
FastFlow	Baseline	63.7	81.0	16.2
	Augmentation	**51.0**	**79.5**	**20.1**
UFlow	Baseline	80.4	93.9	38.7
	Augmentation	**80.9**	**91.6**	**39.2**
CFA	Baseline	**50.9**	**49.8**	**44.7**
	Augmentation	51.0	51.3	43.0
DRAEM	Baseline	53.4	65.2	14.0
	Augmentation	**54.4**	**69.0**	**19.9**
DSR	Baseline	51.0	50.6	18.5
	Augmentation	**52.2**	**54.4**	**29.4**
Rev. Dist.	Baseline	**55.4**	**56.7**	**47.6**
	Augmentation	53.7	57.0	46.1
STFPM	Baseline	62.4	61.4	35.1
	Augmentation	**62.8**	**64.5**	**44.6**

Table 5.2: Data augmentation experiment on anomaly detection algorithms. For each algorithm the parameters with the highest pixel AP score are highlighted.

Type	Algorithm	Image AUROC	Pixel AUROC	Pixel AP
Representation	FastFlow [51]	51.1	83.2	25.0
	UFlow [44]	**98.3**	**99.1**	59.1
	CFA [40]	56.1	53.0	48.0
Reconstruction	DRAEM [41]	82.7	88.2	58.7
	DSR [58]	56.5	58.5	36.3
	Rev. Dist. [39]	78.6	61.2	49.0
	STFPM [56]	95.0	92.8	**73.2**

Table 5.3: Performance of state-of-the-art anomaly detection algorithms on the colourful ALLO dataset. For each metric, the algorithm that performed best is bolded.

effectiveness of these algorithms. All algorithms were tested on the coloured anomaly subset of the ALLO dataset, using the best hyperparameters identified in the previous experiment. The results on the colour test set are shown in Table 5.3 and inference examples from both test sets are shown in Figure 5.2. Most algorithms showed significant improvements in performance when evaluated on coloured anomalies, except for Reverse Distillation and CFA, which were the top performers on the default test set. Reverse Distillation and CFA showed only slight improvements of 1.4% and 3.3% respectively. In contrast, DRAEM and STFPM experienced the largest gains, with increases of 38.8% and 28.6%, respectively. STFPM achieved the highest performance on the colour test set, with a pixel AP of 73.2% and an image AUROC of 95.0%, nearly matching its MVTec image AUROC results.

For an algorithm to identify a pixel as anomalous, the pixel must deviate significantly from the learned distribution. This is evident in the inference examples shown in Figure 5.2 where the top three algorithms detect anomalies more accurately when these objects have a colour distinct from that of the station. Anomalies that resembles the station's colour or appearance are not effectively identified because the algorithms cannot accurately separate the anomalies from the station itself.

5.3 Limitations of Existing Methods

As shown in Section 5.2 existing anomaly detection algorithms are ill-suited for the space environment. In this section, we discuss why these state-of-the-art algorithms struggle in the space domain.

The semi-supervised methods, DRAEM and DSR, performed poorly, with pixel AP scores of 19.9% and 29.4% respectively. These methods introduced synthetically-generated

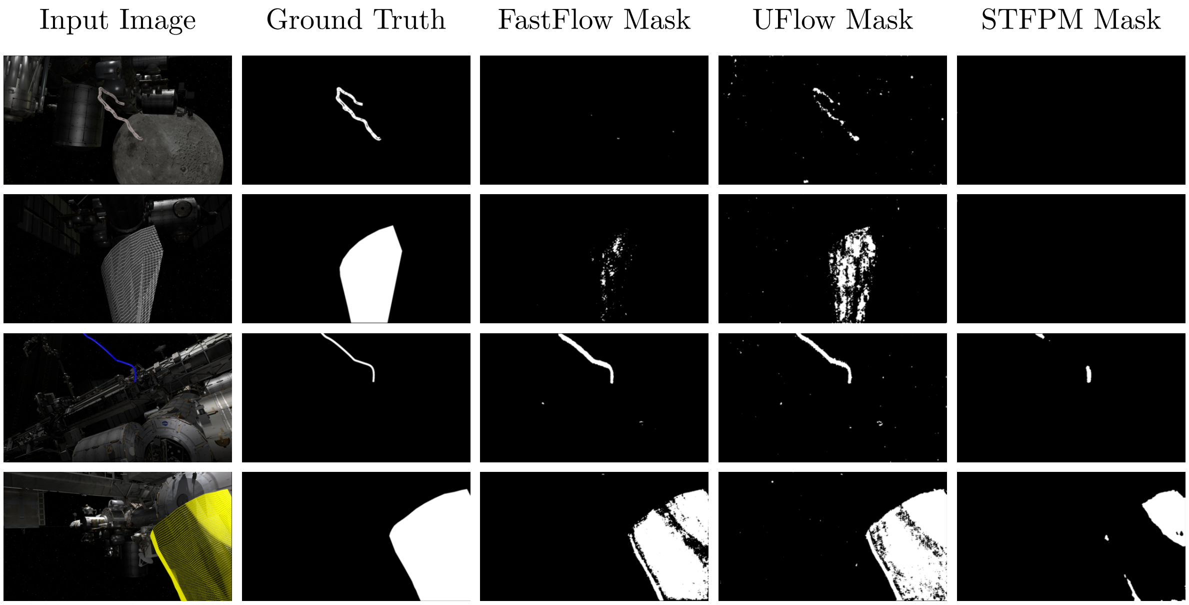


Figure 5.2: Example inference from anomaly detection algorithms on default (top two rows) and colourful anomalies (bottom two rows).

anomalies during training to better define the boundary between normal and anomalous features. However, this approach introduces the risk of overfitting to the synthetic anomalies. When an algorithm overfits to anomalies seen during training, it may fail to generalize to real anomalies during inference [41]. In the space domain, where anomalous and normal features can closely resemble each other, overfitting is already a significant concern. Adding synthetic anomalies during training on the ALLO dataset may distort an algorithm's decision boundary, potentially including some anomalous features within the normal range.

Reconstruction-based anomaly detection algorithms also have difficulty in the space domain. These algorithms aim to learn how to reconstruct images so that only normal features are accurately reconstructed. However, these models can generalize so well that they may successfully reconstruct anomalous features [39]. As a result, when an anomalous feature closely resembles a normal one, it is likely to be reconstructed by the model and incorrectly classified as normal.

Existing anomaly detection algorithms often rely on assumptions about the types of images they will process. One major assumption is that the features in the images are standard and commonly seen, such as those in ImageNet. This assumption does not translate well to the space domain, where features differ significantly from those in the ImageNet dataset. Current methods use network backbones pre-trained on ImageNet in an attempt to leverage common deep representations [45]. However, because the images in the ALLO dataset differ greatly from those in ImageNet, the pre-trained CNN extracts features that are not representative. Without adaptation to the target domain, the inaccurate feature representation from the pre-trained CNN makes it difficult for the anomaly detection algorithm to learn the correct feature distribution. This is demonstrated by the improved performance of nearly all algorithms when using normalization values calculated from the ALLO dataset, rather than the default ImageNet values.

Furthermore, current anomaly detection algorithms assume that features will follow a Gaussian distribution and require a uni-modal distribution of anomaly-free features. This assumption can be problematic for diverse datasets where anomalies might have subtle features close to the normal distribution or when anomaly-free data is multimodal [41]. The normalizing flow method, UFlow, performs well in our benchmark experiments, particularly on the colour test set, because it can create a distribution that allows for a better definition of the anomaly-free distribution. However, UFlow's effectiveness in handling the larger distribution of features in the ALLO is dependent on anomalies being relatively distinct from the learned distribution. As shown by the examples in Figure 5.2 even the best performing algorithms struggled to find anomalies with texture and colour similar to that of the station. Meanwhile, anomalies whose colours were significantly different from the station were identified much more accurately.

Further evidence of current algorithm's dependence on colour can be found by examining the results from STFPM. Although STFPM is classified as a reconstruction method, it relies heavily on feature extraction and matching during reconstruction. STFPM ranked third overall on the default test set and improved by 28.6% on the colour test set, maintaining third place. Among the top three algorithms in both test sets, the top two from the default set (CFA and Reverse Distillation) performed significantly worse on the colour test set. CFA and Reverse Distillation showed minimal improvement on the colour test set, as they do not depend heavily on extracted features. Conversely, the top-performing algorithms on the colour test set were those that relied more on extracted features and, consequently, on anomaly colours.

The most significant limitation of existing anomaly detection algorithms is their assumption about image consistency. These algorithms assume that all images in the training and testing sets have been acquired under the same monochrome lighting conditions and from the same viewpoint, so that content and features appear at approximately the same pixel locations. Consequently, the algorithms calculate a pixel's anomaly score using only the nominal data at that pixel location. This approach can lead to incorrect anomaly scores if there is a misalignment between anomaly-free training images and the testing image [70]. In the space domain, there is an extremely large variety in lighting between images based on the position of celestial bodies. The points-of-view of the images in the ALLO dataset also differ significantly as these images capture the wide range of operations that will be carried out by the Canadarm3. Therefore, two key assumptions made by existing anomaly detection algorithms do not apply to our problem in the space domain.

Another issue with using current anomaly detection methods in the space domain is their focus on structural anomalies rather than logical anomalies. Existing algorithms are designed to detect structural, surface-level anomalies but may not identify objects that are out of place. Since the anomalies anticipated for the Canadarm3 will be primarily logical rather than structural, this reinforces the assertion that current anomaly detection methods are not suitable for the space domain. Based on the non-transferable assumptions of existing methods discussed and the results of the benchmark experiments, it is clear that these algorithms cannot be effectively applied to our problem given the risk of a collision.

Chapter 6

Model-Reference Anomaly Detection

This chapter introduces MRAD (Model Reference Anomaly Detection), a shallow algorithm developed for anomaly detection on a robotic manipulator in lunar orbit. MRAD was developed in response to the limitations of existing methods and utilizes information specific to the Canadarm3's operations to simplify the anomaly detection process. This chapter provides a detailed explanation of the algorithm and discusses the tuning of its parameters for optimal performance on the extended ALLO dataset.

6.1 Algorithm Description

As demonstrated in Chapter 5, existing anomaly detection methods are inadequate for detecting anomalies in lunar orbit, necessitating a new approach. We utilize the fact that for every image captured by the Canadarm's cameras, a corresponding anomaly-free image can be obtained. Since the pose of the arm's camera is always known, a synthetic image representing the expected view can be generated using a CAD model of the station. Once the station is operational, images from past missions can also be used. This image captures what the camera is expected to see and is called the reference image, while the image captured by the arm's camera, which is being evaluated, is called the query image. In this research, the corresponding reference image is generated through a rendering process similar to the one used to create the ALLO dataset. The specific details of how reference images were generated for this study are provided in Section 4.2.

Current methods struggle to accurately define the wide variety of content and complex lighting conditions in images from the space domain. To address this, we simplify the problem by introducing image-specific data rather than relying on a broad distribution of non-anomalous data. By using a predicted, anomaly-free image for each query image, we significantly simplify the anomaly detection process. Instead of searching for deviations from a pre-established normal distribution, MRAD identifies differences between the reference and query images. This approach overcomes the primary challenge that current anomaly detection algorithms face in the space domain: dealing with a widespread normal distribution. We develop a multi-step anomaly detection algorithm that identifies anomalous pixels by first generating a corresponding reference image and then comparing it to the query image.

The domain gap between the reference image and the query image means that differences between them may arise from various factors other than actual anomalies. These differences could be due to errors in camera pose, incorrect lighting in the reference image, missing details in the model, or actual anomalies. For an anomaly detection algorithm to be accurate, it must specifically identify true anomalies rather than all differences between the two images. Therefore, simple image subtraction is inadequate, and a more sophisticated approach is required to ensure that only differences corresponding to actual anomalies are detected.

MRAD assumes that the query and reference images are captured from approximately the same camera pose. Variations in pose between the two images are expected to be small enough that the images are effectively aligned or registered. Pose variation can be minimized using an optimization process, such as feature-based PnP, described in Section 2.3, prior to applying MRAD. Given this assumption of approximate pose alignment, MRAD then focuses on scoring and grouping anomalies. Additionally, MRAD assumes that no more than four separate anomalies are present in the image, due to the way that the query image is divided. However, the maximum number of detectable anomalies can be adjusted by modifying a hyperparameter (see Section 6.1.2).

6.1.1 RXD Anomaly Scores

We begin by developing a method to assign a numerical anomaly score to each pixel in the query image. The Reed-Xiaoli Detector (RXD) [35] [71] is a widely used anomaly detection algorithm, originally designed for unsupervised target detection. As detailed in Section 3.2.1, RXD calculates a pixel's anomaly score by measuring its distance from the background. Although RXD was initially developed for hyperspectral anomaly detection, its mathematical basis allows for adaptation to other use cases, including this one. In this section, we explain the core principles of RXD, how it computes anomaly scores, and the modifications we have made to it.

RXD is based on two primary assumptions: (1) anomalous pixels have values distinct from their surroundings, and (2) anomalies are rare. Under these assumptions, RXD models normal data using a multivariate Gaussian distribution and identifies anomalies as statistical outliers from that distribution. RXD is defined as follows.

Letting H_0 represent the normal data and H_1 denote the anomalous data, the anomaly detection problem can be written as:

$$H_0 : \mathbf{x} = \mathbf{b} \tag{6.1}$$

$$H_1 : \mathbf{x} = \mathbf{s} + \mathbf{b} \tag{6.2}$$

where $\mathbf{x} \in \mathbb{R}^{3\times1}$ is the RGB intensity of a query pixel, $\mathbf{s} \in \mathbb{R}^{3\times1}$ is an anomalous pixel's intensity, and $\mathbf{b} \in \mathbb{R}^{3\times1}$ is the intensity of normal data. RXD assumes that the normal data $\mathbf{b}$ follows a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^{3\times1}$ and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$ such that $\mathbf{b} \sim \mathbf{N}(\mu, \mathbf{\Sigma})$. Using the assumed Gaussian distribution and Equations 6.1 and 6.2, pixels in the normal and anomalous distributions are then defined as:

$$\mathbf{x}|\mathbf{H_0} \sim \mathbf{N}(\mu, \mathbf{\Sigma})$$

and

$$\mathbf{x}|\mathbf{H_1} \sim \mathbf{N}(\mu + \mathbf{s}, \mathbf{\Sigma})$$

Therefore, the probability density functions of the normal and anomalous distributions are:

$$p(\mathbf{x}|H_0) = \frac{1}{(2\pi^{K/2})|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{x}-\mu)} \tag{6.3}$$

and

$$p(\mathbf{x}|H_1) = \frac{1}{(2\pi^{K/2})|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu-\mathbf{s})^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{x}-\mu-\mathbf{s})} \tag{6.4}$$

where K is the number of channels in the query image. The fundamental principle of RXD is the expectation that an anomalous pixel, $\mathbf{x_s}$, will be statistically different from the normal distribution. The probability that said pixel belongs to the normal distribution, $p(\mathbf{x_s}|H_0)$, should be very small. With a fixed normal data distribution, $1/[(2\pi)^{K/2}|\Sigma|^{1/2}]$, the value $(\mathbf{x}-\mu)^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{x}-\mu)$ should be larger for an anomalous pixel than for a normal background pixel. Therefore, RXD calculates the scalar anomaly score of a pixel, $\mathbf{x}$, relative to an image, I, as:

$$\mathrm{RXD}_I(\mathbf{x}) = (\mathbf{x} - \mu)^{\mathbf{T}} \mathbf{\Sigma}^{-\mathbf{1}} (\mathbf{x} - \mu) \tag{6.5}$$

Thresholding the anomaly scores of all pixels in an image is equivalent to thresholding the normal data probability distribution [36].

The main limitation of RXD is its assumption that non-anomalous data follows a Gaussian distribution. To address this, various modifications of RXD have been developed, as outlined in Section 3.2.1. Most of these modifications aim to improve the characterization of the normal background distribution. For example, the Probabilistic Anomaly Detector (PAD) proposed in [36] enhances RXD by iteratively defining the background and target distributions. This approach segments the query image into two distributions, with the anomaly score of a pixel calculated by comparing its RXD score relative to the background distribution with its RXD score relative to the target distribution. The PAD-RXD anomaly score of a pixel, $\mathbf{x}$, is:

$$\mathrm{PAD}(\mathbf{x}) = RXD_0(\mathbf{x}) - RXD_1(\mathbf{x}) \tag{6.6}$$

where $RXD_0(\mathbf{x})$ is the distance from the pixel to the normal background distribution, and $RXD_1(\mathbf{x})$ is the distance from the pixel to the target anomalous distribution. This approach takes into account both how close a pixel is to the predicted normal distribution and to the anomalous distribution. However, PAD requires accurate knowledge of both distributions.

We adapt this principle of using both the normal and anomalous distributions to our anomaly detection problem. However, instead of iteratively separating the normal and anomalous data in a single image, we use the reference image pixel values as the normal background distribution. Namely, the image statistics of the normal distribution ($\mu_{\mathbf{0}}$ and $\mathbf{\Sigma_0}$) are calculated using the reference image, and the image statistics of the anomalous distribution ($\mu_{\mathbf{1}}$ and $\mathbf{\Sigma_1}$) are calculated using the query image. The anomaly score of a pixel is then found using Equation 6.7:

$$\mathrm{MRAD}(\mathbf{x}) = RXD_{ref}(\mathbf{x}) - RXD_{que}(\mathbf{x}) \tag{6.7}$$

Given the complexity and variability of the images (e.g., the black background of space versus the station), both the reference and query images are divided into 4x4 grids, resulting in 16 cells. The statistics (mean and covariance) for both the reference and query images are calculated at the grid level rather than across the entire image. This grid-based approach more accurately represents the image content, as using the entire image would be too general to reliably capture the normal distribution. Each pixel's anomaly score is then calculated using Equation 6.7, with RXD's background statistics derived from the corresponding grid cell. MRAD scores range from 0 to 255, producing a greyscale anomaly score image with the same dimensions as the query image.

Next, the anomaly score map is thresholded so that all scores below a minimum MRAD score are set to zero. This is equivalent to thresholding the probability density function of the reference image. Pixels with scores below this threshold are considered normal, as they closely match the reference image. The minimum MRAD score is empirically determined based on the distribution of anomaly scores, as discussed in Section 6.2. While many anomaly detection algorithms rely solely on score thresholding to distinguish between normal and anomalous pixels, this approach is insufficient for our application. Because several normal pixels may still show high anomaly scores, additional analysis is required to reduce the number of false positives identified by the algorithm.

A significant source of high anomaly scores is amplified noise from the black background of space. When sections of either the reference or query image are completely black, small background variations can be amplified during anomaly score calculation, leading to noisy outputs, as shown in Figure 6.1. This noise, which has a broad range of scores, cannot be effectively addressed by thresholding or blurring alone. To tackle this issue, the physical distribution of anomaly scores within each grid cell is examined before incorporating them into the score map. Initially, a preliminary check identifies cases where an anomaly might cover the entire grid, indicated by high anomaly scores across the grid. If over 90% of the scores in a grid exceed the minimum MRAD score, it is assumed that a large anomaly is present, eliminating the need for noise removal. If there is a range of scores, a kernel-based noise removal method is then applied. A kernel, whose dimensions are 10% of the grid, is defined. The kernel is moved over the grid in a sliding window and the standard deviation of the pixels within each window is calculated. If the standard deviation between all of the windows does not vary by more than an experimentally-set limit (around 3-5%), the entire grid is declared to be noise and all scores are changed to zero. This approach is based on the principle that amplified noise, caused by small variations in the black reference and query images, should be evenly and randomly distributed across the grid cell. Any anomaly present in the grid cell would distort this distribution. After removing the amplified noise, a Gaussian blur is applied to smooth the anomaly score image. The smoothing operation filters local variations in the anomaly scores to effectively join anomalous regions together. This step reinforces the continuity of anomalous pixels, ensuring that anomalous pixels can be clustered in the next step.

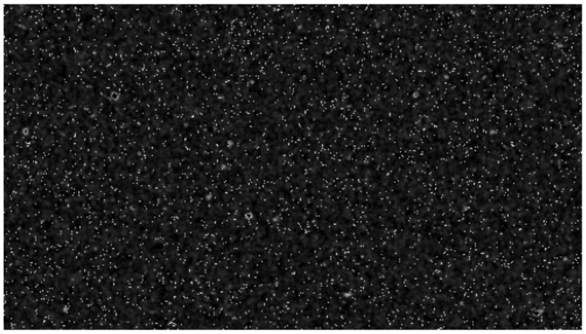


Figure 6.1: Amplified noise caused by the anomaly score calculation in Eq. 6.7.

6.1.2 Region Growing

Next, region growing is applied to group anomalous pixels into clusters representing anomalous objects. This step aims to aggregate anomalous pixels based on their location and anomaly score. MRAD's region growing step uses three parameters: the starting point, the footprint, and the tolerance. The starting point, or seed, is where the algorithm begins to search for pixels with similar values. The footprint is the 2D search area centered at a pixel, within which the algorithm looks for pixels that fall within the tolerance range. The footprint can be square, diamond, octagonal, etc.; MRAD uses a square footprint. The tolerance value defines the range of pixel values considered similar to the seed pixel. Pixels within this tolerance range are grouped with the seed pixel. Both the footprint size and the tolerance are determined empirically. The region growing step produces a binary anomaly mask.

The image is evaluated in quadrants to account for anomalies spanning multiple score grids. A region growing seed is calculated as the centroid of all nonzero pixels in each respective quadrant. Using multiple seeds accommodates the possibility of several anomalies (up to four when the image is divided into quadrants) and ensures a more accurate seed location, as the centroid of an anomaly remains unaffected by noise elsewhere in the image. Next, a search area and tolerance are defined. From the start point, any pixel within the search area and within the specified tolerance is added to the region. Region growing continues for all grouped pixels until no more are found. A larger search area and tolerance result in more pixels being grouped as anomalies, but increasing these parameters also raises the risk of false positives.

The principle behind this step is that anomalies typically consist of pixels with similar values (anomaly scores) located close to each other. Region growing aggregates these anomalous pixels while excluding those whose position or value is too distant from the identified cluster. Once the binary anomaly map is generated, the final step is to count the number of anomalous pixels and classify the image as either normal or anomalous. An image is classified as anomalous if it contains a minimum number of anomalous pixels.

The MRAD algorithm described above is outlined in Algorithm 1, and is demonstrated step-by-step in Figure 6.2. In the query image (Figure 6.2a), an anomalous blanket is visible on the left side. The anomaly score image (Figure 6.2c), shows how MRAD effectively captures the blanket as an anomaly. However, amplified noise appears in the bottom left of the score image, where both the query and reference images are very dark. This noise is removed, and the image is blurred, resulting in the refined score image in Figure 6.2d. Additionally, a false positive region with higher scores appears near the top of the image due to a pose misalignment between the query and reference images, causing part of the station to be shifted in the reference image. Region growing is then applied, producing the anomaly map in Figure 6.2e. This map accurately encompasses the anomalous blanket while deeming the false positive at the top of the image too small to be labelled as an anomaly.

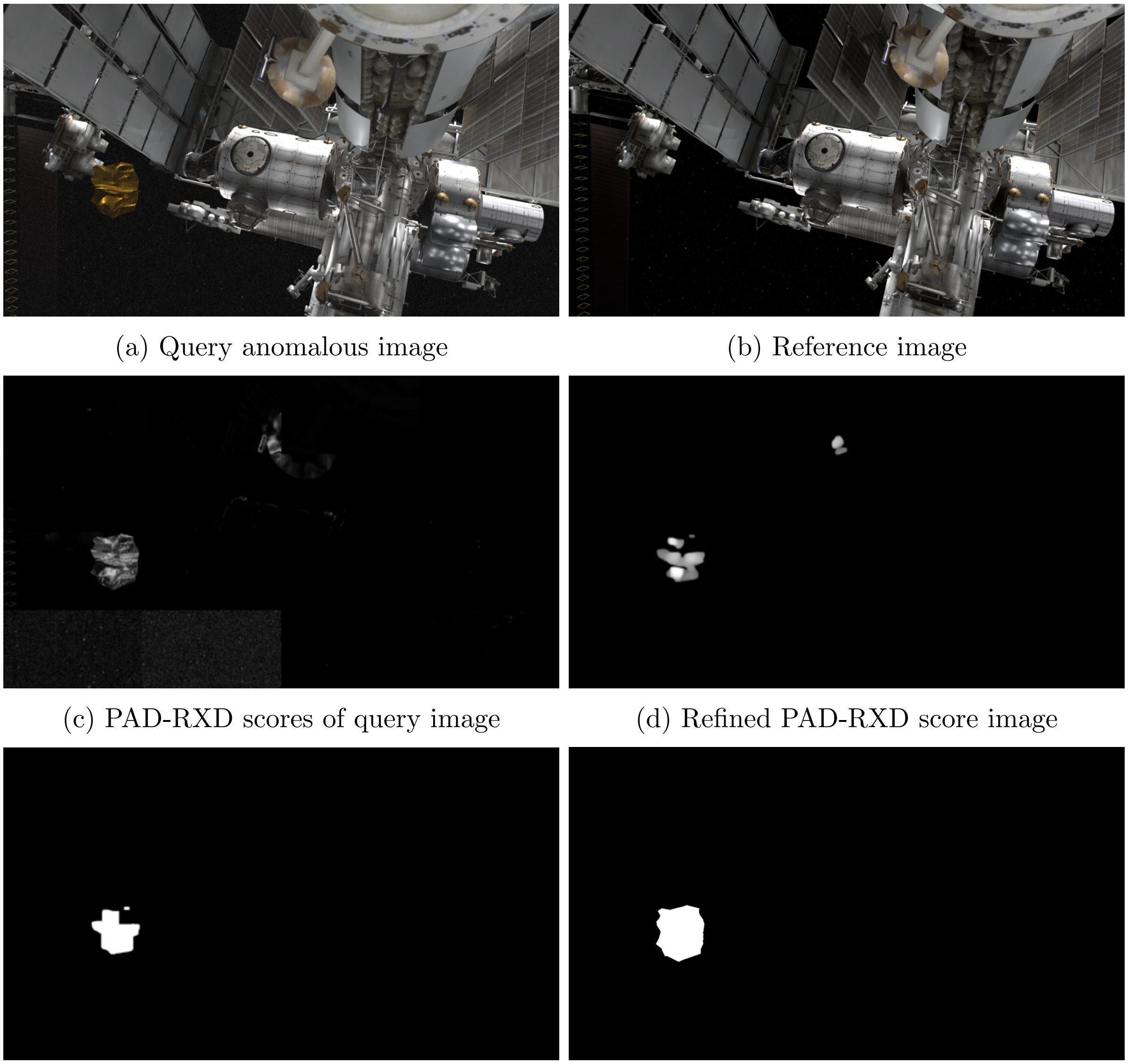
6.2 Algorithm Tuning

MRAD's anomaly detection performance is affected by several of its parameters. In this section we describe which parameters were tuned and what metrics were selected for optimization.

MRAD was optimized on the extended ALLO dataset to maximize its pixel AP score, the primary metric used in the benchmark analysis conducted in Chapter 5. Pixel AP measures the algorithm's ability to identify true positives, which is crucial in anomaly detection. A grid search optimization was conducted to find the parameter values that maximized pixel AP. The following four parameters were evaluated and optimized:

1. The minimum PAD-RXD value such that any pixel below said value was automatically considered normal.
2. The region growing footprint size.
3. The region growing tolerance.
4. The final classification threshold.

A lower minimum PAD-RXD value increases the number of pixels considered in the second half of the algorithm, making anomaly detection easier but also raising the like-



(a) Query anomalous image

(b) Reference image

(c) PAD-RXD scores of query image

(d) Refined PAD-RXD score image

(e) Anomaly map

(f) Query Image Mask

Figure 6.2: Step-by-step example of the MRAD algorithm.

Algorithm 1 Model-Prediction Anomaly Detection

```
for grid cell in image do
    for pixel in grid cell do
        $AnomalyScore = MRAD(pixel, grid)$
        if $pixel < MinPixel$ then
            $AnomalyScore = 0$
        end if
    end for
    $gridSD = standard\ deviation(grid, kernel)$
    if $range(gridSD) < 3\%$ then                          ▷ 3% is a tuning parameter
        grid cell $= 0$
    end if
end for
$AnomalyScoreImage = Reconstruct(gridcells)$
Apply Gaussian Blur to $AnomalyScoreImage$                  ▷ Blur is a tuning parameter
for pixel in $AnomalyScoreImage$ do
    if $pixel < MinPixel$ then
        $anomalyscore = 0$
    end if
end for
for SearchArea in $AnomalyScoreImage$ do
    $StartPoint = centroid\ of\ nonzero\ pixels$
    $StartPoint = True$
    for pixel in SearchArea if $pixel = True$ do
        if pixel $in$ footprint $AND < abs(tolerance)$ then
            $pixel = True$
        else $pixel = False$
        end if
    end for
end for
AnomalyMap = region grown search areas
$ScoreCount = sum(AnomalyMap)$
if $ScoreCount > anomaly\ threshold$ then
    $Image = Anomalous$
else $Image = Normal$
end if
```

lihood of false alarms. The shape and size of the region growing process determine how anomalous pixels are grouped. A larger region growing size means the flooding algorithm searches further from the start pixel, making it more likely to encompass anomalies fully. However, this also increases the potential for incorrectly enlarging false alarms. Setting a higher region growing tolerance allows MRAD to group pixels with scores farther from the start pixel, which is helpful when anomalies have a wide range of scores due to complex textures or lighting. However, a higher tolerance can also cause false positives to be grouped with actual anomalies, distorting the results. The classification threshold determines how large a group of pixels must be to be considered an anomaly. A lower threshold allows for the identification of smaller anomalies but can also increase false alarms. The key principle in optimizing these parameters is that anomalies tend to appear in larger, more structured regions than false alarms caused by pose or lighting differences.

Each possible combination of these parameters, within a reasonable range, was tested on the extended ALLO dataset, and the pixel AP was calculated. The parameter combination that produced the highest pixel AP was selected as the final configuration for the MRAD algorithm. The optimal parameters are listed in Table 6.1, resulting in a pixel AP of 60.1%. The minimum required anomaly score for a pixel to be considered for region growing is 60 out of 255. This value is relatively high, considering it is just the first step in the MRAD algorithm. Many anomaly detection algorithms have a higher threshold for anomalies, but those algorithms rely solely on anomaly score thresholding. As outlined in Chapter 3 current algorithms operate in simpler environments and are not faced with anomalies that closely resemble normal images. Therefore, simple thresholding suffices for these algorithms. The fact that the first step in the MRAD algorithm still requires a relatively high threshold to filter out false positives before further analysis underscores the complexity of this anomaly detection application.

The region growing step of the MRAD algorithm is optimized to have a square footprint of 200 × 200 pixels and a tolerance of 50 pixels. A 200 × 200 square footprint represents nearly 2% of the query image (all images are $1,920 \times 1,080$ pixels), meaning that the region growing step searches for pixels in close proximity to one another. However, the relatively large tolerance reflects the understanding that while anomalous pixels are closely located, they can have a wide range of anomaly scores due to the scene's complexity. These optimized values align with the MRAD algorithm's principle that anomalies are assumed to be closely grouped pixels with relatively high but varied anomaly scores.

The classification threshold is optimized to 2,000 pixels, which is slightly less than 0.1% of a 1,080p image. Therefore, anomalies must occupy at least 0.1% of the image

Parameter	**Optimal Value**
Minimum pixel	60
Region growing size	200
Region growing tolerance	50
Classification Threshold	2,000

Table 6.1: MRAD algorithm parameters optimized for the ALLO dataset.

to be detected by the MRAD algorithm. This small percentage reflects the algorithm's ability to detect smaller anomalies. However, even with noise removal and region growing steps, very small false positive pixels can be identified and must be dismissed by MRAD. Further analysis of these tuning parameters and the performance of MRAD with its optimized parameters is discussed in the following chapter.

Chapter 7

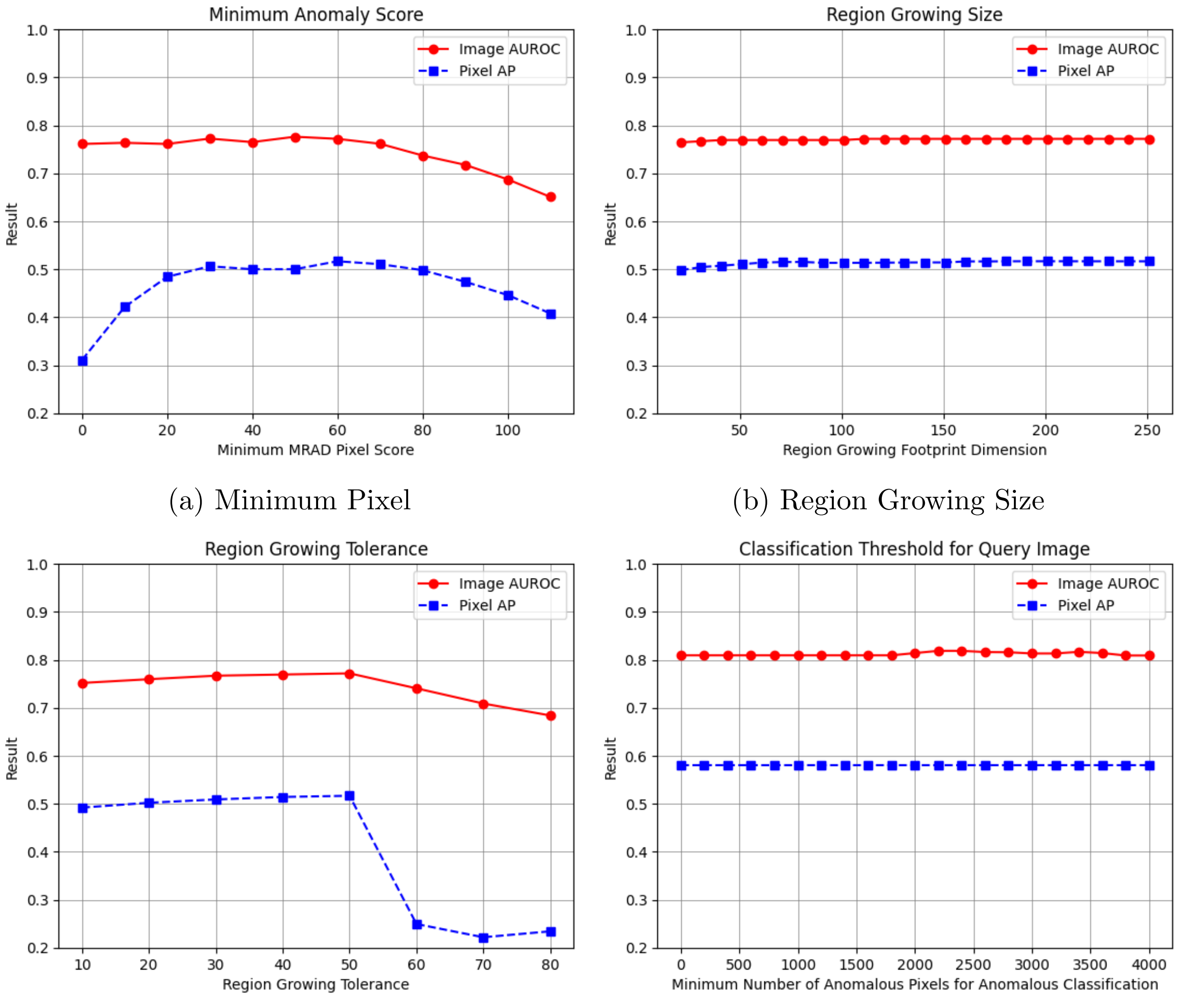
MRAD Experiments

In this chapter, we evaluate the effectiveness of the MRAD algorithm on the ALLO dataset. We first analyze the impact of key parameters on the algorithm's overall results and explore how these parameters shape MRAD's anomaly detection process. MRAD is then applied to the extended ALLO dataset, and its performance is compared with the Anomalib benchmark discussed in Chapter 5. Additionally, we investigate how different factors in the query image, such as anomaly colour, size, and lighting variations, influence MRAD's detection capabilities. Finally, we review instances where MRAD either fails to detect an anomaly or generates a false alarm.

7.1 Parameter Evaluation

In this section, we examine how the different parameters of the MRAD algorithm, described in the previous chapter, influence its performance. We vary each of the four parameters listed in Section 6.2 individually, while keeping the remaining three constant as indicated in Table 6.1. The effects of varying each parameter on pixel AP and image AUROC are presented in Figure 7.1.

The first MRAD parameter we evaluate is the minimum pixel value, which significantly influences MRAD's performance. Higher minimum pixel values lead to lower pixel AP and image AUROC scores, as more pixels are incorrectly discarded as normal. Many of these discarded pixels are actually anomalous but have lower anomaly scores, highlighting how closely anomalous pixels can resemble normal ones. This demonstrates how setting a classification threshold based solely on anomaly score, as many existing methods do, can decrease performance. However, setting the minimum pixel value too low can also reduce performance by incorrectly resulting in the labelling of more normal pixels as anomalous.



(a) Minimum Pixel

(b) Region Growing Size

(c) Region Growing Tolerance

(d) Classification Threshold

Figure 7.1: Effect of various parameters on the performance of the MRAD algorithm.

The region growing size parameter has a minimal impact on either metric. Expanding the search area for anomalous pixels during the region growing step does not significantly affect MRAD's performance. This is consistent with the fact that the anomalies in our dataset and application are objects rather than random pixels, meaning that anomalous pixels are typically clustered together. The MRAD anomaly score effectively identifies these pixels, making it unnecessary to search far beyond an identified pixel to locate others. Utilizing a smaller footprint during region growing also reduces the computational complexity of the algorithm, as fewer pixels need to be evaluated.

Unlike the region growing size, the region growing tolerance parameter significantly impacts MRAD's performance. Smaller tolerance values result in a lower pixel AP score, indicating that some anomalous pixels have been missed. Although increasing the tolerance value improves the pixel AP, the image AUROC remains relatively unchanged. The reason for the result is because an image is classified as anomalous as long as the tolerance is low enough that a sufficient number of anomalous pixels are detected. As the tolerance increases, more pixels are grouped into the anomaly. However, if the tolerance is set too high, the entire image can be mistakenly classified as normal due to excessive pixel grouping. This can happen either because the image is homogeneous with no anomalies, or because the tolerance is excessively high, causing all pixels to be grouped together. Therefore, optimizing the tolerance parameter for maximum pixel AP is crucial to ensure the tolerance is high enough to capture anomalies, but not so high that it causes incorrect grouping of the entire image.

The classification threshold has a minimal effect on the image AUROC and does not impact the pixel-level metrics, as expected. Increasing the classification threshold slightly improves the image AUROC by correctly classifying more images as anomalous. However, this improvement is constrained because higher thresholds cause MRAD to mistake smaller anomalies as false negatives and incorrectly label the image as non-anomalous. The classification threshold essentially sets the minimum size of an anomaly that MRAD can detect. For example, a threshold of 2,000 pixels in a 1080p image requires the anomaly to occupy at least 0.01% of the image. Although this is a small percentage, larger anomalies may still be mistaken as false negatives if fewer than 2,000 anomalous pixels are detected. The impact of anomaly size on overall performance is evaluated in Section 7.2.2.

Algorithm	**Image AUROC**	**Pixel AP**
FastFlow	51.0	20.1
UFlow	**80.9**	39.2
CFA	50.9	44.7
DRAEM	54.4	19.9
DSR	52.2	29.4
Rev. Dist.	55.4	47.6
STFPM	62.8	44.6
MRAD (ours)	79.0	**60.1**

Table 7.1: Anomalib algorithms and MRAD results on the ALLO dataset default coloured anomalies.

7.2 Experimental Results

Using the optimal parameters identified in Section 6.2 the MRAD algorithm is applied to the extended ALLO dataset. The calculated image AUROC, pixel AUROC, and pixel AP are presented in Table 7.1 alongside the results of Reverse Distillation which performed best in the benchmark conducted in Chapter 5. The image AUROC is 79.0%, and the pixel AUROC is 68.3%, which are 23.6% and 11.6% higher, respectively, than those achieved by Reverse Distillation, the best-performing algorithm in the benchmark. Regarding image classification performance, 75.6% of anomalous images are correctly classified as anomalous, while 82.4% of normal images are correctly classified as non-anomalous. The pixel AP is 60.1%, representing a 12.5% improvement over the benchmark.

Sample results showing the anomalous query image, the pixel-level ground truth segmentation mask, and the calculated anomaly map are provided in Figure 7.2. These results demonstrate that MRAD can identify anomalies of various shapes and sizes. MRAD's use of a reference image provides uniquely relevant information for the query image, eliminating the need for a general non-anomalous distribution that might fail to accurately encompass only normal features. However, MRAD may not be able to detect anomalies that are very small, blend in with other objects, or are poorly illuminated. Several examples of failure cases and false alarms are discussed in Section 7.3.

Anomalies that are well-illuminated or of a brighter colour are more easily identified than those hidden by shadows or similar in colour to the station. The most significant factor impacting the algorithm's performance is the colour of the anomaly, as evaluated in Section 7.2.1. The size of the anomalies also influences detection; larger anomalies are more likely to be detected during the region-growing stage, while smaller anomalies

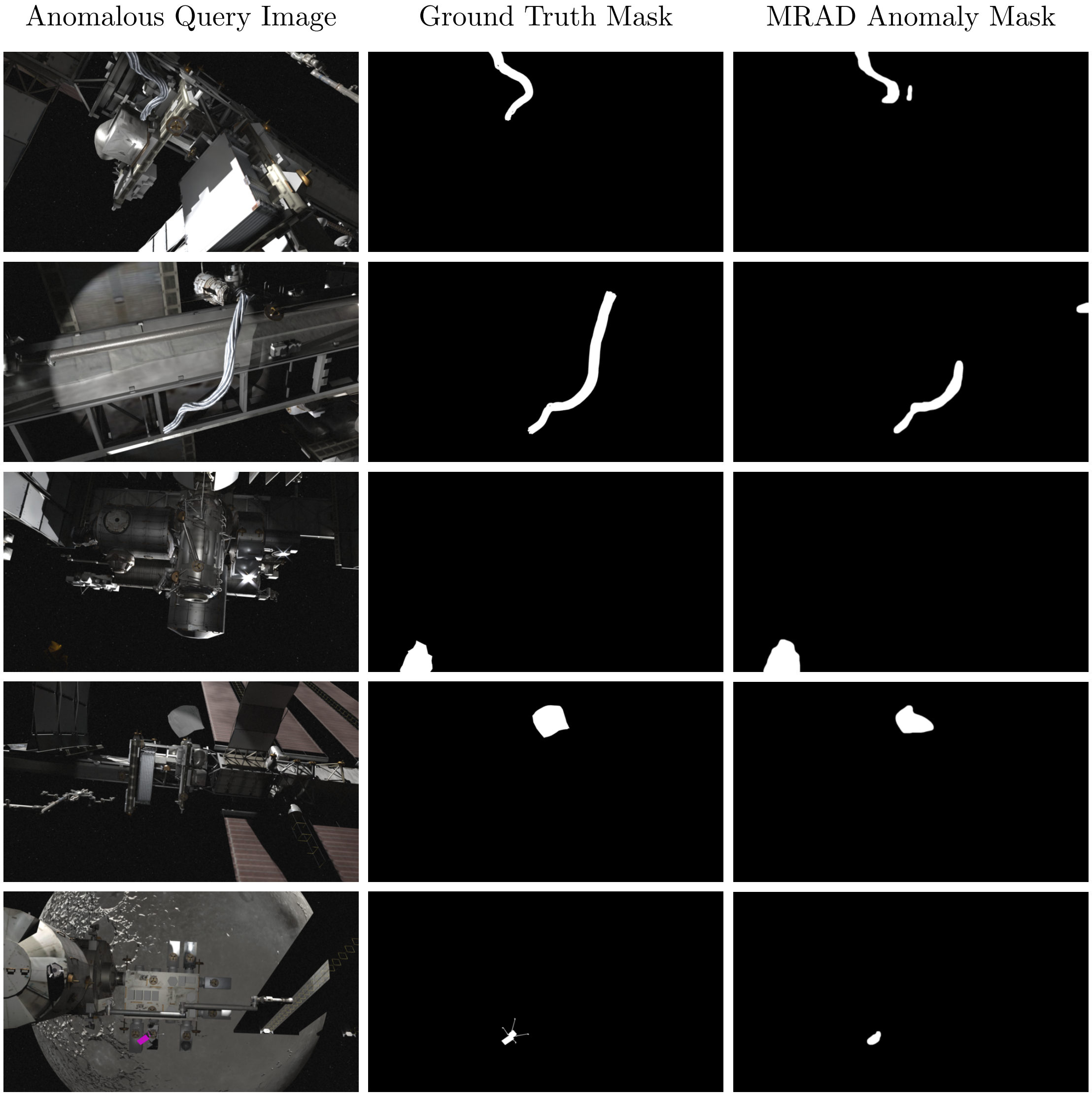


Figure 7.2: Inference examples from MRAD algorithm.

are only identified if they had a higher anomaly score. The impact of anomaly size on MRAD's performance is discussed in Section 7.2.2.

7.2.1 The Effect of Colour and Lighting

We evaluate the factors in the query and reference images that impact MRAD's performance. First, we examine how the colour of anomalies affects the algorithm's performance. Next, we study how differences in lighting between the reference and query images influence MRAD's ability to detect anomalies. For each experiment, 1,000 random images from the extended ALLO dataset are used, with their query images varied as needed.

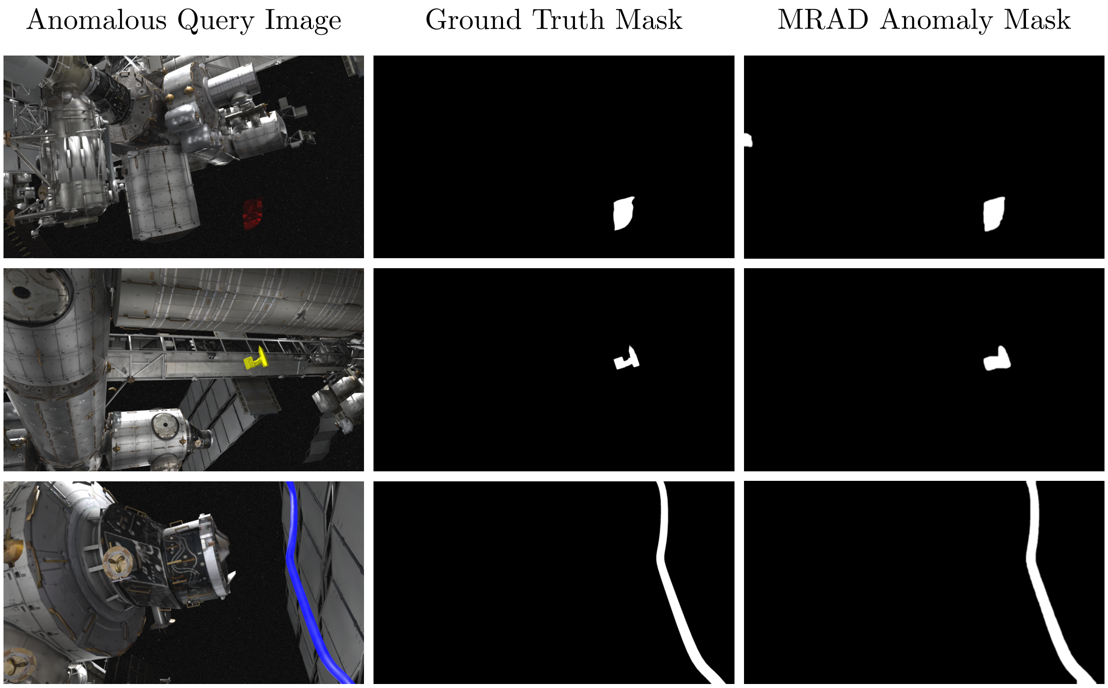
Metric	Colourful Anomalies	Lighting Differences
Image AUROC	84.7	75.7
Pixel AUROC	94.8	67.7
Pixel AP	86.2	58.0

Table 7.2: MRAD results on colourful anomalies and on query image with varied lighting.

The MRAD algorithm is initially tested on a subset of the extended ALLO dataset where all anomalies had brighter colours. In this test, the anomalous images contain red, blue, or yellow anomalies. The resulting image AUROC, pixel AUROC, and pixel AP are shown in the middle column of Table 7.2. These results indicate that the colour of an anomaly significantly impacts the algorithm's performance. All metrics improved: the image AUROC increasing by 5.7%, the pixel AUROC by 26.5%, and the pixel AP by 26.1%. Similar to the learned algorithm evaluated in Chapter 5, MRAD is more effective at detecting anomalies that differ in appearance from the background.

Sample results of the MRAD algorithm's output when tested on colourful anomalies are shown in Figure 7.3. In cases where the anomaly is significantly different in colour, the algorithm clearly identifies nearly all of the anomaly's pixels. When the anomaly closely resembles the station, the boundary of the anomaly may appear slightly blurred or bleed outwards, as seen in the cable example in the first row of Figure 7.2. However, for colourful anomalies, the detected boundary is much clearer. Additionally, MRAD can detect anomalies that are poorly illuminated or small in size, provided they are brightly coloured. This is illustrated by the red blanket and yellow drill in Figure 7.3. Therefore, it is clear that similar to existing methods, MRAD performs well when dealing with anomalies that look very different from the station. Where MRAD is able to exceed existing methods is in the more nuanced cases where the anomaly resembles the station more closely or is ill-illuminated.

We next evaluate how differences in lighting between the query and reference images affect MRAD's performance to understand how the algorithm responds to larger variations between the images. We create a smaller test set with 1000 images from the extended ALLO dataset where the lighting conditions in the query image are randomly altered. The strength and position of either the sunlight or the camera's spotlight are randomly decreased when generating the query image. Thus, the query image has weaker light and slightly different shadows, while default colour anomalies are used as in the original experiment. This analysis helps assess MRAD's robustness to lighting variations, which are likely to occur due to the Sim2Real domain gap. The calculated image AUROC, pixel AUROC, and pixel AP are presented in the right column of Table



Anomalous Query Image Ground Truth Mask MRAD Anomaly Mask

Figure 7.3: Example inference from MRAD algorithm on colourful anomalies.

7.2.

There is a slight decrease in all three metrics, with the image AUROC decreasing by 3.30%, the pixel AUROC by 0.60%, and the pixel AP by 2.10%. This indicates that MRAD can handle differences in lighting between the reference and query images with minimal performance loss. Since a pixel's anomaly score is based on its distance to a reference grid, the average content of the grid determines the anomaly score even if lighting conditions differ. Additionally, because the region growing step groups pixels with similar scores that are close to one another, variations in lighting that distort anomaly scores are less likely to significantly affect performance. However, larger lighting variations can result in false positives, such as those caused by intense sunlight or bright reflections from the station, as further discussed in Section 7.3.

7.2.2 The Effect of Anomaly Size

We examine how the size of an anomaly affects the performance of the MRAD algorithm. The anomalous test set is divided into categories based on the percentage of the image occupied by the anomaly. Images where the anomaly occupies 9% or less are categorized into individual percent ranges (1%, 2%, 3%, ..., 8%). All images with anomalies occupying more than 9% of the image are grouped together. Since most images have anomalies occupying less than 9% of the image, this categorization ensures that each group contains

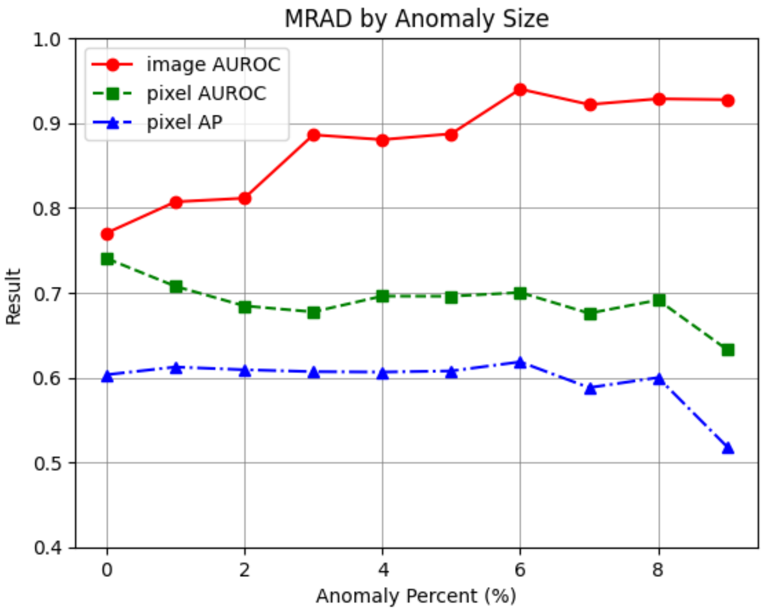


Figure 7.4: MRAD metrics by anomaly size.

at least 20 images.

The effect of the anomaly size is evaluated using anomalies with default colours, and the results of this experiment are presented in Figure 7.4. Both the pixel AUROC and pixel AP showed minimal changes compared to the image AUROC. MRAD's ability to detect anomalous pixels is not highly dependent on anomaly size, as the algorithm relies on anomaly scores and the proximity of pixels to other anomalous pixels. The decline in pixel performance for larger anomalies occurs in cases where the algorithm struggles to detect a large thermal blanket that matches the station's colour. This anomaly is difficult for MRAD to identify because its texture and colour blend into the station, resulting in lower anomaly scores. The image AUROC, however, is significantly affected by anomaly size, as image classification depends on the number of detected anomalous pixels. If only a limited percentage of the pixels in an anomaly are detected, that anomaly will still be identified if the number of detected pixels is above the required threshold. Therefore, not all pixels in a larger anomaly need to be detected for the image to correctly classified as anomalous making larger anomalies more likely to be identified.

7.3 Failure Cases

In this section, we examine cases where the MRAD algorithm fails. First, we look at instances where MRAD returned a false negative, meaning it did not detect enough

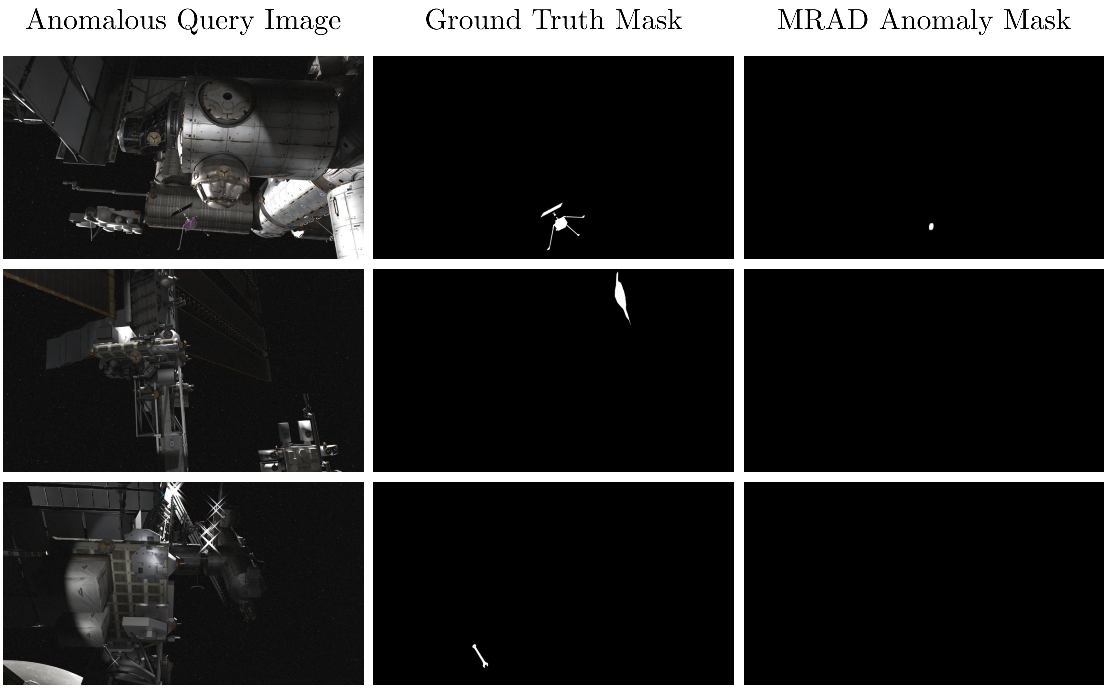


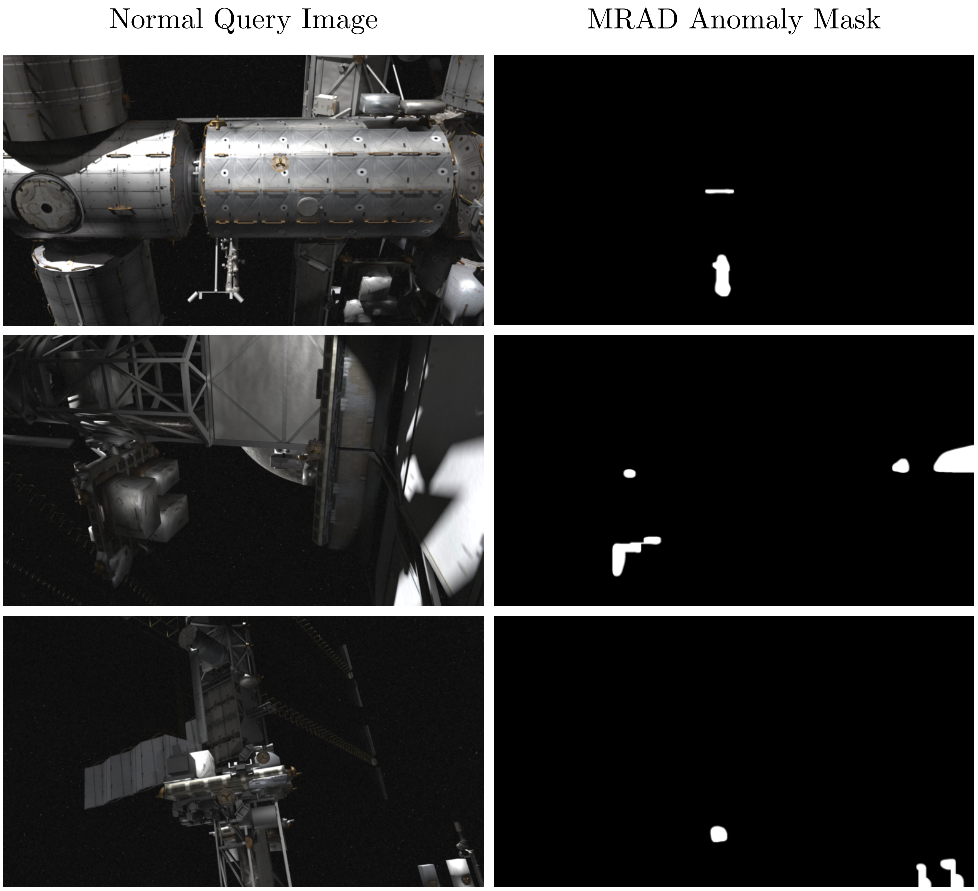
Figure 7.5: Examples of failed anomaly detection from MRAD algorithm.

anomalous pixels to classify the image as anomalous. Specific examples of these failure cases are shown in Figure 7.5.

The first example (first row) of Figure 7.5 shows a scenario where MRAD detected some anomalous pixels but did not find enough to classify the image as anomalous. Some anomalous pixels may have had scores too low to be included in the region-growing step, leading to an incomplete identification of the anomaly. The second example illustrates how MRAD fails to detect anomalous pixels when the anomaly is completely shadowed. In such cases, there is insufficient information for MRAD to identify the anomaly because the algorithm cannot detect the blanket hidden in the shadow. The third example involves a small wrench visible in the query image. However, it is too small to be successfully separated from the station's background. MRAD incorrectly identifies the wrench as part of the station due to its size and similar colour.

These examples of false negatives emphasize how an anomaly must be of a sufficient size and adequately illuminated to be detected. If an anomaly is too small or too dark, MRAD may lack enough information to identify it within a complex image. Additionally, MRAD struggles to detect anomalies that blend well with the station background. When an anomaly's colour and texture closely resemble those of the station, distinguishing the anomaly from the station can be challenging.

Next, we examine cases where the MRAD algorithm returned a false positive, also known as a false alarm. In these cases, the algorithm incorrectly identifies part of the query image as an anomaly. Examples of false alarms are shown in Figure 7.6, where these false alarms correspond to isolated, bright areas of the station. When parts of the query image differ significantly in illumination or location from the reference image, MRAD may identify those pixels as anomalous. False alarms can occur due to pose differences between the query and reference images or lighting variations. For instance, if sunlight or a camera's spotlight reflects differently off a part of the station in the query image, that region may be incorrectly flagged as anomalous. Many false alarms result from a combination of lighting differences and pose misalignment. When slight pose errors cause a bright patch in the query image to appear in a different location in the reference image, this patch is detected as an anomaly. The MRAD algorithm uses noise removal and minimum pixel requirements to address these false positives. However, setting these parameters too aggressively can increase the risk of missing genuine anomalies. Therefore, MRAD's sensitivity to false alarms should be adjusted according to the specific application. For the Canadarm3, a more conservative approach is preferred, focusing on optimizing pixel AP.



Normal Query Image

MRAD Anomaly Mask

Figure 7.6: Examples of false alarms from MRAD algorithm.

Chapter 8

Conclusion and Future Work

The Canadarm3 will be a crucial Canadian contribution to international efforts in expanding human exploration beyond low Earth orbit. The operational conditions of the Lunar Gateway, and consequently the Canadarm3, present several new challenges, including the need for the arm to autonomously detect anomalies in its environment. Given the critical importance of avoiding potentially catastrophic collisions, addressing the problem of anomaly detection is essential to ensuring the safe operation of the Canadarm3. To our knowledge, no existing work has explored the task of anomaly detection in space.

8.1 Discussion

In this thesis, we introduced the problem of anomaly detection in the space domain, specifically for robotic operations of a manipulator in lunar orbit. The primary objective of this research was to develop an anomaly detection algorithm for the Canadarm3. Previous work on anomaly detection has mainly focused on industrial inspection applications, which involves much more simple and consistent data. Furthermore, while many datasets related to space exploration exist, each has been designed for a specific application or mission, and none are tailored for anomaly detection. By evaluating existing methods for anomaly detection, we identified the challenges that must be addressed to develop an effective anomaly detection algorithm for the Canadarm3.

First, we created the ALLO (Anomaly Localization in Lunar Orbit) dataset, a synthetic collection of photorealistic images replicating those expected to be captured by the cameras on Canadarm3. The ALLO dataset was generated in Blender by utilizing Blender's ray tracing capabilities and detailed models of the ISS and moon. To address the Sim2Real domain gap, Gaussian noise was added to the images, along with random variations in camera pose, lighting, and glare. Pixel-level ground truth maps were generated for all anomalous test images.

We used the ALLO dataset to benchmark existing anomaly detection algorithms, assessing how well current approaches generalize to the space domain. Current anomaly detection methods, designed for applications with much narrower normal distributions, were found to struggle significantly in the space domain. Based on these observations and leveraging unique operational information, we developed MRAD (Model Reference Anomaly Detection), a novel anomaly detection algorithm specifically for the Canadarm3. MRAD utilizes the known pose of the camera relative to the station to create a reference image for all query images. Pixel-level anomaly scores are calculated by comparing pixels in the query image to grid-level statistics from the reference image. These scores are then thresholded, noise is removed, and region growing is applied to cluster anomalies into identifiable objects. MRAD relies solely on the reference image, avoiding dependence on a dataset-specific non-anomalous distribution which often fails to capture the detailed information needed for complex applications.

MRAD was extensively tested on the ALLO dataset and compared our established benchmark. The results demonstrated that MRAD outperformed all evaluated state-of-the-art methods across all metrics. Additionally, we assessed the algorithm's sensitivity to various factors within a query image, including lighting variations, anomaly colour, and anomaly size. These experiments highlighted MRAD's ability to effectively handle the diverse lighting conditions and complex imagery typical of the space domain.

8.2 Limitations and Potential Improvements

Finally, we outline the limitations of the MRAD algorithm, propose potential solutions to address these challenges, and suggest directions for future research. The primary limitation of MRAD is its dependence on an accurate reference image. While the reference image eliminates the need for a general dataset-wide normal distribution, it must closely resemble the query image to ensure precise anomaly detection. However, matching the reference image to the query image can be difficult due to the Sim2Real gap. Two potential strategies could be considered to address this issue. First, as the Gateway's development progresses and operational data becomes available, more information could be integrated into the reference image rendering process, either through enhanced simulations or by using images from actual operations. Alternatively, inspiration could be drawn from [36] where the query image was iteratively segmented into background and target distributions. It could be valuable to explore ways of eliminating the need for a reference image by segmenting the query image into a set of background pixels and a set of 'suspected-anomaly' pixels. This approach would simplify the anomaly detection process to a single-image problem, though it may introduce the risk of insufficient information to identify anomalies.

Another limitation of the MRAD algorithm is its binary classification output, which does not include any measure of confidence in its decision. To enhance the algorithm, a future improvement could involve integrating a probabilistic model to quantify the overall confidence in the predictions. This model could utilize both historical data and internal parameters of MRAD. For instance, if an image is classified as anomalous but the anomaly scores and size are small, the probabilistic model could indicate lower confidence in the classification. Additionally, this model could be designed to address false positives by assessing the proximity of detected anomalies to known sun positions. If an anomaly in the query image is detected near the sun's location in the reference image, the model could assign a lower confidence level to this anomaly. Incorporating a probabilistic model would provide more detailed and relevant information, thereby enhancing the operational decision-making process for the Canadarm3.

Another area for future research could involve enhancing the capabilities of existing deep anomaly detection algorithms. While current methods struggle to generalize to the space domain, neural networks, with their ability to extract relevant features, present a promising solution for detecting anomalies that blend into the station's background. As demonstrated in the benchmark in Chapter 5, existing methods benefited from using custom normalization values derived from the ALLO dataset. Extending this approach to extract features specific to the space environment could further improve the performance of learned anomaly detection algorithms. Additionally, learned anomaly detection algorithms could leverage segmentation to isolate anomalies from the background. Since the station will remain a constant throughout operations, exploring the training of a network to segment the station from unknown anomalies could be advantageous.

Anomaly detection is a complex problem, particularly in the space domain, where there are significant variations in lighting and image content. We have demonstrated that current anomaly detection methods are inadequate for the space domain and have introduced a novel algorithm that addresses these limitations. Given the novelty of this application and the importance of accurate results, we hope the presented dataset, benchmark, and algorithm will be valuable for future research and development.

Bibliography

[1] "About Canadarm3." https://www.asc-csa.gc.ca/eng/canadarm3/about.asp, 2020.

[2] "Lost astronaut tool bag from ISS shines in new telescope image (photo)." https://www.space.com/international-space-station-spacewalk-lost-tool-bag-photo.

[3] A. Alhakamy and M. Tuceryan, "Real-time illumination and visual coherence for photorealistic augmented/mixed reality," *ACM Computing Surverys*, vol. 53, pp. 49:1–49:34, May 2020.

[4] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization," *International Journal of Computer Vision*, vol. 130, pp. 947–969, April 2022.

[5] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, pp. 756–795, May 2021.

[6] "Benchmarking unsupervised anomaly detection and localization," *arXiv preprint arXiv:2205.14852*.

[7] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD — a comprehensive real-world dataset for unsupervised anomaly detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9584–9592, 2019.

[8] "International space station 3d model - NASA science." https://science.nasa.gov/resource/international-space-station-3d-model/.

[9] J. C. Crusan, R. M. Smith, D. A. Craig, J. M. Caram, J. Guidi, M. Gates, J. M. Krezel, and N. B. Herrmann, "Deep space gateway concept: Extending human presence into cislunar space," in *2018 IEEE Aerospace Conference*, pp. 1–10, March 2018.

[10] “NASA’s gateway program - NASA.” https://www.nasa.gov/reference/nasas-gateway-program/.

[11] “How often does the international space station dodge space debris? | space.” https://www.space.com/international-space-station-space-dodge-debris-how-often, December 2023.

[12] T. Ehret, A. Davy, J.-M. Morel, and M. Delbracio, “Image anomalies: A review and synthesis of detection methods,” *Journal of Mathematical Imaging and Vision*, vol. 61, pp. 710–743, June 2019.

[13] M. Pajusalu, I. Iakubivskyi, G. J. Schwarzkopf, O. Knuuttila, T. Väisänen, M. Bührer, H. Teras, G. L. Bonhomme, M. F. Palos, J. Praks, and A. Slavinskis, “SISPO: Space imaging simulator for proximity operations,” *PLOS ONE*, vol. 17, p. e0263882, March 2022.

[14] “Blender.” https://www.blender.org/.

[15] “Unreal engine 5.” https://www.unrealengine.com/en-US/unreal-engine-5.

[16] J. T. Kajiya, “The rendering equation,” in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’86, pp. 143–150, Association for Computing Machinery, August 1986.

[17] K. G. Mehrotra, C. K. Mohan, and H. Huang, *Anomaly Detection Principles and Algorithms*. Terrorism, Security, and Computation, Springer International Publishing, 2017.

[18] J. Yang, K. Zhou, Y. Li, and Z. Liu, “Generalized out-of-distribution detection: A survey,” *International Journal of Computer Vision*, June 2024.

[19] T. M. Tran, T. N. Vu, N. D. Vo, T. V. Nguyen, and K. Nguyen, “Anomaly analysis in images and videos: A comprehensive review,” *ACM Computing Surveys*, vol. 55, pp. 148:1–148:37, December 2022.

[20] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel, “Deep learning for anomaly detection: A review,” *ACM Computing Surveys*, vol. 54, pp. 38:1–38:38, March 2022.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc., 2012.

[22] V. Kaynig, B. Fischer, and J. M. Buhmann, "Probabilistic image registration and anomaly detection by nonlinear warping," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[23] R. Szeliski, *Computer Vision: Algorithms and Applications.* Texts in Computer Science, Springer International Publishing, 2022.

[24] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2 ed., 2004.

[25] R. Brochard, J. Lebreton, C. Robin, K. Kanani, G. Jonniaux, A. Masson, N. Despré, and A. Berjaoui, "Scientific image rendering for space scenes with the SurRender software," in *International Astonautial Conference 2018*, (Bremen, Germany), 2018.

[26] M. Bechini, M. Lavagna, and P. Lunghi, "Dataset generation and validation for spacecraft pose estimation via monocular images processing," *Acta Astronautica*, vol. 204, pp. 358–369, March 2023.

[27] L. Bingham, J. Kincaid, B. Weno, N. Davis, E. Paddock, and C. Foreman, "Digital lunar exploration sites unreal simulation tool (DUST)," in *2023 IEEE Aerospace Conference*, pp. 1–12, IEEE, August 2023.

[28] S. Parkes, I. Martin, M. Dunstan, and D. Matthews, "Planet surface simulation with PANGU," in *2nd RPI Space Imaging Workshop.*

[29] R. T. Eapen, R. R. Bhaskara, and M. Majji, "NaRPA: Navigation and rendering pipeline for astronautics," *arXiv preprint arXiv:2211.01566*, 2022.

[30] M. Jawaid, E. Elms, Y. Latif, and T.-J. Chin, "Towards bridging the space domain gap for satellite pose estimation using event sensing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11866–11873, May 2023.

[31] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, July 2001.

[32] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *Journal of Machine Learning Research*, vol. 2, pp. 139–154, 2001.

[33] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognition*, vol. 40, pp. 863–874, March 2007.

[34] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," *arXiv preprint arXiv:2005.02357*, February 2021.

[35] I. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1760–1770, 1990.

[36] L. Gao, Q. Guo, A. J. Plaza, J. Li, and B. Zhang, "Probabilistic anomaly detector for remotely sensed hyperspectral data," *Journal of Applied Remote Sensing*, vol. 8, p. 083538, November 2014.

[37] B. Du and L. Zhang, "Random-selection-based anomaly detector for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, pp. 1578–1589, May 2011.

[38] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV*, pp. 475–489, Springer-Verlag, January 2021.

[39] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9727–9736, IEEE, June 2022.

[40] S. Lee, S. Lee, and B. C. Song, "CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization," *IEEE Access*, vol. 10, pp. 78446–78454, 2022.

[41] V. Zavrtanik, M. Kristan, and D. Skocaj, "DRÆM – a discriminatively trained reconstruction embedding for surface anomaly detection," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8310–8319, October 2021.

[42] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "CutPaste: Self-supervised learning for anomaly detection and localization," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9659–9669, June 2021.

[43] M. Tailanian, P. Musé, and A. Pardo, "A multi-scale a contrario method for unsupervised image anomaly detection," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 179–184, December 2021.

[44] M. Tailanian, A. Pardo, and P. Musé, “U-flow: A u-shaped normalizing flow for anomaly detection with unsupervised threshold,” *Journal of Mathematical Imaging and Vision*, May 2024.

[45] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, “Towards total recall in industrial anomaly detection,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14298–14308, June 2022.

[46] N. A. Ahuja, I. Ndiour, T. Kalyanpur, and O. Tickoo, “Probabilistic modeling of deep features for out-of-distribution and adversarial detection,” *arXiv preprint arXiv:1909.11786*, September 2019.

[47] N. Marchal, C. Moraldo, H. Blum, R. Siegwart, C. Cadena, and A. Gawel, “Learning densities in feature space for reliable segmentation of indoor scenes,” *IEEE Robotics and Automation Letters*, vol. 5, pp. 1032–1038, April 2020.

[48] H. Zhang, Z. Wang, Z. Wu, and Y.-G. Jiang, “DiffusionAD: Norm-guided one-step denoising diffusion for anomaly detection,” *arXiv preprint arXiv:2303.08730*, 2023.

[49] E. D. Cook, M.-A. Lavoie, and S. L. Waslander, “Feature density estimation for out-of-distribution detection via normalizing flows,” *Proceedings of the Conference on Robots and Vision*, February 2024.

[50] D. Gudovskiy, S. Ishizaka, and K. Kozuka, “CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1819–1828, January 2022.

[51] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu, “FastFlow: Unsupervised anomaly detection and localization via 2d normalizing flows,” *arXiv preprint arXiv:2111.07677*.

[52] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “GANomaly: Semi-supervised anomaly detection via adversarial training,” in *Computer Vision – ACCV 2018* (C. V. Jawahar, H. Li, G. Mori, and K. Schindler, eds.), pp. 622–637, Springer International Publishing, 2019.

[53] T. Schlegl, P. Seeböck, S. Waldstein, G. Langs, and U. Schmidt-Erfurth, “f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks,” *Medical Image Analysis*, vol. 54, January 2019.

[54] A.-S. Collin and C. De Vleeschouwer, “Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7915–7922, January 2021.

[55] A. Bauer, S. Nakajima, and K.-R. Müller, “Self-supervised training with autoencoders for visual anomaly detection,” *arXiv preprint arXiv:2206.11723*, January 2024.

[56] G. Wang, S. Han, E. Ding, and D. Huang, “Student-teacher feature pyramid matching for anomaly detection,” in *British Machine Vision Conference*, March 2021.

[57] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, “Adversarially learned one-class classifier for novelty detection,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3379–3388, June 2018.

[58] V. Zavrtanik, M. Kristan, and D. Skočaj, “DSR – a dual subspace re-projection network for surface anomaly detection,” in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), Lecture Notes in Computer Science, pp. 539–554, Springer Nature Switzerland.

[59] Y. Cao, X. Xu, Z. Liu, and W. Shen, “Collaborative discrepancy optimization for reliable image anomaly localization,” *IEEE Transactions on Industrial Informatics*, vol. 19, pp. 10674–10683, November 2023.

[60] J. Jeong, Y. Zou, T. Kim, D. Zhang, A. Ravichandran, and O. Dabeer, “WinCLIP: Zero-/few-shot anomaly classification and segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19606–19616.

[61] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, “Reproducible scaling laws for contrastive language-image learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

[62] D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone, “Defect detection in SEM images of nanofibrous materials,” *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 551–561, April 2017.

[63] M. Wieler, T. Hahn, and F. Hamprecht, “Weakly supervised learning for industrial optical inspection,” in *27th DAGM Symposium*, vol. 6, p. 11, 2007.

[64] "CIFAR-10 and CIFAR-100 datasets." https://www.cs.toronto.edu/~kriz/cifar.html.

[65] "View the best images from NASA's artemis i mission - NASA." https://www.nasa.gov/humans-in-space/view-the-best-images-from-nasas-artemis-i-mission/.

[66] D. E. Lee, "White paper: Gateway destination orbit model: A continuous 15 year NRHO reference trajectory." NTRS Author Affiliations: NASA Johnson Space Center NTRS Report/Patent Number: JSC-E-DAA-TN72594 NTRS Document ID: 20190030294 NTRS Research Center: Johnson Space Center (JSC).

[67] "ASCL.net - skyfield: High precision research-grade positions for planets and earth satellites generator." https://ascl.net/1907.024.

[68] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[69] S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc, "Anomalib: A deep learning library for anomaly detection," in *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 1706–1710, October 2022.

[70] J. Jang, E. Hwang, and S.-H. Park, "N-pad: Neighboring pixel-based industrial anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4365–4374, 2023.

[71] X. Yu, I. Reed, and A. Stocker, "Comparative performance analysis of adaptive multispectral detectors," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2639–2656, August 1993.