Damaged/missing part detection from images and 3D models in articulated objects

Carlos del Campo Olano

Master of Science

Data Science

School of Informatics

University of Edinburgh

2024

Abstract

Manufacturing, especially in articulated objects, results in complex quality assurance processes. In light of this, devising anomaly detection is quite a difficult task, especially because of the complexity of joints and movable parts. Traditional methods usually fail to accurately differentiate normal joint movements from actual anomalies. This dissertation deals with developing techniques for anomaly detection in articulated objects.

The three primary contributions of this work are: the establishment of an anomaly detection-oriented articulated object dataset that covers a wide range of joint positions and types of anomalies, and an advanced, expressive multi-view representation method that captures the dynamic movement of articulations while expressing a rich visual dataset suitable for training machine learning models. In this regard, the Correspondence Matching Transformer model is examined and a heuristic-based view selection processing is introduced to optimize the selection of multi-view images whereby the computational costs and the scalability are improved.

Experimental results show that, while the CMT model provides the best overall detection accuracy, the enhancement with the heuristic-based method improves its detection in anomalous cases, adding difficulties to the correct classification of natural joint movements.The findings suggest that the choice between using the heuristic-enhanced CMT or the vanilla CMT model should be guided by the specific application requirements—whether minimizing false negatives or false positives is more critical.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Carlos del Campo Olano*)

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Hakan Bilen, for his invaluable guidance, support, and encouragement throughout this research. His expertise and insights were instrumental in shaping the direction of this dissertation, and I am sincerely thankful for the countless hours he dedicated to reviewing my work and providing constructive feedback.

I would also like to extend my heartfelt thanks to my parents Enrique, Esther and my brother Pablo for their unwavering support and belief in me throughout this journey. Their encouragement has been a constant source of motivation.

A special thank you goes to my girlfriend, Ligia, whose love, patience, and understanding have been my greatest support during the ups and downs of this project.

I would also like to acknowledge my flatmates for their friendship and the positive atmosphere they created, which helped me stay focused and balanced during the challenging phases of this dissertation.

Table of Contents

Chapter 1

Introduction

In recent years manufacturing companies have invested high amounts of money into their quality assurance process mechanism, to ensure the products reaching customers satisfy their standards for customer satisfaction [11]. Traditional quality inspection mechanisms rely on visual inspection performed by humans. The main drawbacks of this approach are that the process is highly time-consuming and error-prone. The advancement in computer vision and anomaly detection creates new possibilities for enhancing both the efficiency and precision of manufacturing quality controls.

The detection of defects is especially difficult in articulated objects, whose joints have some degree of movement. Traditional methods, which depend on human inspection or simplistic anomaly detection models, often fall short, as the firsts produce errors on smaller objects or objects with smaller parts and it is extremely expensive. Meanwhile, the second approach lacks the ability to differentiate between the normal movement of a joint and an anomaly. Therefore, there is a pressing need for sophisticated approaches that can accurately capture and analyze the intricate patterns of normal and anomalous behaviours in these objects.

This process would impact the accuracy of detecting errors in production, as the human eye is error-prone and traditional systems rely on 2D images. Integrating 3D baseline imaging with automated systems can refine the detection process and reduce the likelihood of overlooking subtle anomalies produced by defects, without introducing costs, because a 3D reference model is commonly used when producing a good. In addition, earlier detection of anomalies can reduce the shipment and return costs associated with defective products. Moreover, the process would be more scalable, as the number of produced goods grows the system would be able to scale accordingly. Finally, if a client claims he received a defective product, this can be checked with a

picture of the object. This would further enhance the company's customer relations.

However, currently, state-of-the-art anomaly detection systems focus on non-articulated objects. Usually, this is not realistic, as many of the produced goods have movable parts, like the wheels of a chair, the handle in a kettle and the screen on a laptop. This limitation is significant for the adoption of these techniques in many industries, where considering possible normal movements of a product is essential and not labelling normal joint movement as anomalous becomes crucial. This is especially difficult as objects even within the same object category may have different joint ranges or tolerances and even different numbers of joints. Furthermore, many manufacturing companies produce a set of objects of different types, difficulting even more task with different sizes, shapes and forms. Deecke et al. [9] employ a GAN network to learn representations of normality and then contrast a query image. Nevertheless, this methodology must capture a highly complex distribution that includes all possible joint configurations in the latent space for articulated objects. Furthermore, SimpleNet [27], a simple network designed for image anomaly detection and localization uses a pre-trained feature extractor, a feature adaptor to reduce domain bias, and a discriminator trained on synthetic anomaly features generated by adding Gaussian noise to normal features. This approach relies heavily on pre-trained feature extractors and simple feature adaptors. These components are designed to process images with relatively static and consistent shapes, significantly differing from articulated objects. This MSc project dissertation aims to broaden the scope of current anomaly detection systems, making them robust for anomaly detection in settings with articulated objects.

Another challenge in this domain is finding or creating a suitable dataset to train a deep-learning model. A common practice in the literature for anomaly detection is using a well-known dataset like MNIST [8] [9] [28] and establishing a subset of the classes as the abnormal instances. Nevertheless, this does not consider the articulated nature of the objects. Another approach is using real-world anomalous instances, however, no information on normal ranges for the joints is available. Thus, it is impossible to differentiate between normal and abnormal range movement.

1.1 Objectives

Aiming to address the challenges in detecting anomalies in articulated objects this dissertation establishes some goals. These objectives will guide the research and development of advanced techniques and models. The primary objectives of this

dissertation are as follows:

- **A novel dataset:** Create a dataset designed for anomaly detection in articulated objects, including different joint positions and different anomaly types. This dataset is required to train machine learning models aiming to differentiate between normal and abnormal images in articulated objects, as there is no other anomaly detection dataset with focus on articulated objects.
- **Novel Multi-View Representation:** Develop a multi-view representation adapted to the joint movements of articulated objects. The multi-view representation includes articulated objects represented visually so that humans and machine learning models can identify the normal range of movement from the joint.
- **Evaluation of the CMT Model:** Evaluate the Correspondence Matching Transformer (CMT) [1] model. This evaluation assesses the model's ability to handle complex and high-dimensional data associated with articulated objects and identifies if the model can differentiate between normal joint movement and anomalies.
- **Heuristic-Based View Selection:** Implement a heuristic-based view selection method to adapt the CMT model to handle an increased number of multi-view images efficiently. The heuristic involves calculating a similarity score between the query image and the multi-views, selecting the top-k most similar images. This approach addresses the computational challenges associated with processing large sets of multi-view images for attention mechanisms, enhancing the model's scalability and performance.

In pursuit of these objectives, the dissertation will employ a structured methodology that includes data collection, model development, and evaluation phases. The creation of a novel dataset will serve as the foundation, supporting the development of advanced multi-view representations and the adaptation of the CMT model by incorporating the view selection heuristic. By addressing both data and computational challenges, this research aims to contribute to the field of anomaly detection in articulated objects significantly.

1.2 Structure of the Dissertation

The remainder of this dissertation is structured as follows:

- **Chapter 2: Background:** Provides a detailed review of existing literature on anomaly detection. It sets the foundation for understanding the current state of the field and the gaps that this research aims to address.
- **Chapter 3: Methodology:** Describes the process of dataset creation, data generation, and the development of the novel multi-view representation. It also covers the implementation of the Correspondence Matching Transformer (CMT) model and the heuristic-based view selection method.
- **Chapter 4: Experiments:** Presents the experimental setup, the conducted experiments, and the results. This chapter includes a thorough analysis of the model's performance using metrics such as Accuracy and AUC, providing insights into the effectiveness of the proposed approaches.
- **Chapter 5: Conclusions and Future Work**: Summarizes the outcomes of the dissertation, discusses the implications of the results, and suggests directions for future research.

Chapter 2

Background

2.1 Anomaly Detection

Anomaly detection is defined as finding data points, events or observations that substantially differ from the dataset's expected behaviour. Sometimes this term is also referred to as outlier detection. Anomalies can indicate critical situations, such as errors, defects, or security breaches. The process involves analyzing patterns in the data and identifying instances that do not conform to expected norms. Formally, anomaly detection is defined as follows:

Given a dataset D where most data points conform to a defined notion of normality, anomaly detection is the task of identifying all $x \exists D$ that do not comply with this norm. The anomalies represent patterns in data that do not conform to expected behaviour, flagged as outliers or exceptions depending on the context. Different types of anomalies have been identified in the literature.

- **Point Anomalies**: A point anomaly refers to an individual data point that is significantly distant from the majority of data in a dataset. Formally, given a dataset $D \subset R^n$, a point $x \in D$ is considered a point anomaly if it deviates substantially from the other points in D with respect to a chosen metric or distance function. This type of anomaly is the simplest and most common across various applications, including fraud detection in financial transactions and anomaly detection in environmental sensor data.

- **Contextual Anomalies**: Contextual anomalies are data points that are anomalous only within a specific context or condition. Often they are referred to as conditional anomalies. These are formally defined based on the surrounding data

in a contextual space C, which could be temporal, spatial, or defined by other environmental conditions. A data point x is considered a contextual anomaly in D if it behaves as an outlier within a subset $D_C \subset D$ where D_C is the set of data points in a specific context $c \in C$.

- **Collective Anomalies**: Collective anomalies refer to a collection of related data points in D that are anomalous when considered together, although the individual points may not be outliers by themselves. These are formally recognized when a sequence or a group of data points in D deviates significantly from the entire dataset's expected pattern or sequence. Collective anomalies are particularly relevant in domains like signal processing or motion tracking, where a sequence of measurements or events may indicate a malfunction or another significant anomaly that is not discernible at the individual level.

If the anomaly detection is performed in imaging or videographic data then it is referred to as visual anomaly detection.

2.2 Anomaly detection literature review

Historically, anomaly detection relied on statistical methods and shallow learning models, which were proficient at handling low-dimensional datasets typically found in early anomaly detection tasks. Despite these techniques working well in some situations, they frequently struggled to handle complicated data structures and high-dimensional data, such as graphs, sequences, and images. Examples of these methods are principal component analysis (PCA) [5] [37] [17] and random projection [23] [29] [33].

With the advent of deep learning, the ability to process high-dimensional and complex data types has significantly improved. Deep learning methods for anomaly detection, can be classified into three categories: Deep Feature Extraction, Learning Feature Representations of Normality and End-to-End Anomaly Score Learning.

2.2.1 Deep Feature Extraction

This approach leverages deep neural networks to transform raw data into a set of features that can be used by traditional anomaly detection techniques, as deep learning models have demonstrated better capability than other dimension reduction methods such as PCA[30]. This method follows the assumption that the extracted feature representations

preserve the discriminative information that helps separate anomalies from normal instances and helps extract semantic-rich features and non-linear feature relations [2] [14]. One of the most prominent approaches is using pre-trained models to extract low-dimensional features. Zhou et al.[46] focus on detecting anomalies in video surveillance. To do so they use RankSVM [4] to compress a sequence of frames into a single static image and then leverage the ImageNet dataset [35] to train a model that aims to capture the features of a given image. Finally, the extracted features are used by a LSTM model that makes the final predictions. Moreover, Liang et al. [24] leverage transfer learning techniques to extract a weight vector for each feature in the source dataset and apply it to preprocess the target dataset. Another approach is to specifically train a deep learning model to detect anomalous data. Xu et al.[43] introduce a model that combines convolutional neural networks (CNNs) for spatial feature extraction and recurrent neural networks (RNNs) for temporal feature analysis to enhance anomaly detection in videos. Furthermore, Erfani et al.[12] propose a hybrid model where an unsupervised Deep Belief Network (DBN) is trained to extract generic underlying features, and a one-class SVM is trained from the features learned by the DBN. This hybrid model not only improves detection performance but also significantly reduces computational costs. Moreover, Yu et al.[45] introduce NetWalk, a dynamic network embedding approach for anomaly detection in evolving networks. NetWalk learns representations that are dynamically updated, employing clique embedding and a deep autoencoder for effective feature learning. The method achieves real-time anomaly detection with constant memory usage, proving its efficiency and flexibility in handling various types of networks. However, the main drawback of the deep feature extraction models is that the disjointed nature of feature extraction and anomaly scoring often leads to suboptimal anomaly scores.

2.2.2 Learning Feature Representations of Normality

Techniques under this category aim to model what normal data should look like and in that way identify deviations. Autoencoders are a popular choice in this framework [15] [19] [39] [18], where the model learns to compress and decompress data and anomalies are detected based on reconstruction errors. These methods can perform poorly if the training data does not accurately represent the full distribution of normal behaviour. The main hypothesis behind this method is that the model would reconstruct normal instances with low error. Meanwhile, anomalies will result in higher reconstruction

errors as they vary significantly from the training data. This approach has been effective in various domains, including image and video anomaly detection.

2.2.3 End-to-End Anomaly Score Learning

The most direct approach involves training a model to classify data points as normal or anomalous in a single step [38]. Deep neural networks are trained on labelled data to differentiate between the normal and abnormal directly. Nevertheless, these methods often need a customized loss function difficulting the optimization of this approach. Additionally, manually identifying and annotating anomalous data results in immense financial costs.

A notable example of this direct approach is the Deep Weakly-supervised Anomaly Detection framework proposed by Pang et al. [31]. This method introduces a Pairwise Relation Prediction Network (PReNet) that learns pairwise relation features and anomaly scores by predicting the relationship between any two randomly sampled training instances. The pairwise relations can be anomaly-anomaly, anomaly-unlabeled, or unlabeled-unlabeled. This innovative approach unifies relation prediction and anomaly scoring, enabling the model to assign higher anomaly scores to pairs that contain anomalies compared to pairs that do not. PReNet leverages the fact that unlabeled data is mostly normal, allowing it to learn a wide variety of normal and abnormal pairwise patterns. This results in improved detection of both seen and unseen anomalies. The pairwise relation approach also significantly augments the training anomaly data, enhancing the model's robustness and generalization capabilities. A notable drawback of PReNet is its sensitivity to the contamination rate in the unlabeled data. While the model is designed to be robust to small amounts of anomaly contamination, its performance can degrade if the proportion of anomalous data in the unlabeled set is too high. This sensitivity necessitates careful preprocessing and filtering of the data to ensure the effectiveness of the model, adding another layer of complexity to its application.

Similarly, the Self-trained Deep Ordinal Regression for End-to-End Video Anomaly Detection framework proposed by Pang et al. [32] applies self-trained deep ordinal regression to video anomaly detection, overcoming two key limitations of existing methods: reliance on manually labelled normal training data and sub-optimal feature learning. By formulating a surrogate two-class ordinal regression task, the approach develops an end-to-end trainable video anomaly detection method that enables joint

representation learning and anomaly scoring without the need for manually labeled normal and abnormal data. The method starts with a pre-trained model, such as ResNet-50, on relevant auxiliary labelled data. Initial pseudo-labels of normality and abnormality are generated using generic anomaly detectors. These pseudo labels are used to create a self-training loop, where the model iteratively refines its anomaly scores by learning from the initial pseudo normal and anomalous frames. This iterative process allows the model to improve its accuracy by leveraging the ordinal dependence in the supervision information. This approach entails similar drawbacks as PReNet.

2.2.4 AD Image Benchmarks

One of the problems of developing an anomaly detection model is the absence of large datasets for this purpose. A wide range of works in the literature have used existing classification datasets, such as MNIST [8] and CIFAR [22], arbitrarily selected a subset of classes and treat them as anomalous classes, training the model on the rest of the classes only [7] [34]. In contrast, a different approach is using a dataset containing real-world anomalous instances. These exist containing irregularly shaped objects [36], objects with different defects such as scratches, dents or contamination [3] and defects in the materials [6]. Additionally, [1] introduces a benchmark for the specific AD task, this project aims to solve, composed of different 2D images of chairs annotated as anomalous or not and linked to the baseline 3D model. The generated anomalies include 5 different anomaly types: positional anomalies, rotational anomalies, broken or damaged parts, generated by boolean subtraction, component swapping and missing components. However, this dataset does not include the possibility of moving articulated objects nor any information about possible articulations, limiting the use of this dataset to rigid bodies.

2.3 Articulated Objects Representation

Articulated Objects are complex structures where multiple parts are interconnected and can move relative to each other through joints. This category includes human bodies, robotic arms, doors, drawers, laptops and other mechanical devices. Modelling these objects and their behaviour is a significant challenge in fields like robotics and computer vision, as it involves representing their geometry and the possible movement and interaction of the parts.

One main approach to representing articulated objects is to model these objects through skeleton-based approaches. Under this approach, objects are modelled as a hierarchy of fixed parts (bones) and joints. This method works best when the dynamics of a specific object are known beforehand allowing accurately representing their structure. However, when the joint structures are extremely complex or unknown this representation becomes impractical. [42] introduces a novel dataset encompassing 2347 articulated objects following this methodology. Each object in the dataset is represented in the Universal Robot Description Format (URDF), a widely used XML format for representing robots' kinematic and dynamic properties. Furthermore, [16] and [44] represent articulated objects as a mesh file and include annotations on each of their shapes. For each shape of the dataset, a pair of parts is labelled as *moving part* and *reference part*. Each part is then taken as a mobility unit, annotated with the corresponding motion parameters. These parameters consist of four elements, the type of transformation, the location and orientation of the transformation axis, and the extent or range of the motion. The main disadvantage of the use of the mesh file is the lack of texture, which effectively limits its use in real-world scenarios.

Moreover, PARIS, is presented in [26], which uses a part-level reconstruction technique to analyze the motion of articulated objects. PARIS uses Neural Implicit Representations to separately model the static and moving parts of an object. The main disadvantage of this approach is the lack of stability, which leads to inconsistent results.

The Ditto model [20] creates digital twins of articulated objects through interactive perception. Ditto makes use of implicit neural representations to jointly estimate part-level geometry and articulation models from point cloud data captured before and after an object interaction. Despite showing some generalization, this model is not designed to handle unknown objects. This results in a limited applicability for objects not used for training the model.

A different approach is the use of Neural Implicit Representations (NIRs). These are continuous, differentiable functions parameterized by neural networks that can model objects' geometry and motion. NIRs' main strength is that they allow for flexible and accurate modelling of complex shapes and articulations without the need for predefined structures or extensive training data. Weng et al. [41] introduce a novel method in this category, building upon Ditto and PARIS. The method consists of two stages. First, the object's shape is reconstructed using a Signed Distance Function (SDF) representation, and then the articulation model is estimated by identifying the joints and segmenting the parts. Therefore, the model can generalize to objects with multiple moving parts

and not rely on prior knowledge of the object. Nevertheless, the camera parameters are required to be known for this method to work.

Chapter 3

Methodology

3.1 Dataset

A crucial aspect of developing a robust anomaly detection system is the creation of a comprehensive and representative dataset. This section details the processes of creating the dataset used in this study, including the source of the objects, the types of anomalies introduced, the simulation of object movements, and the multi-view generation of images.

3.1.1 Source of Objects

The foundation of the dataset is built upon the PartNet-Mobility Dataset [42], a well-established and extensively used resource in the field of computer vision and robotics. PartNet-Mobility provides a rich collection of 3D models with detailed part annotations, hierarchical structures, and photorealistic textures, making it an ideal foundation for creating a diverse set of articulated objects for anomaly detection.

PartNet-Mobility is notable for its extensive variety of articulated objects, including furniture, appliances and other everyday items. Each object in the dataset is segmented into functional parts, which are annotated with semantic labels. This segmentation facilitates a deeper understanding of the object's structure and functionality, allowing for more precise manipulation and analysis.

The hierarchical structure of PartNet-Mobility objects captures the relationships between different parts, such as parent-child relationships and kinematic constraints. This inherent hierarchy is crucial for accurately simulating real-world scenarios where parts move relative to one another. For instance, the dataset includes detailed information

about how doors open, drawers slide and wheels rotate, providing a realistic basis for generating articulated movements.

The PartNet-Mobility Dataset originally consists of 2,347 articulated objects. However, some objects contain errors that make them unusable for the purposes of this study. These errors are mainly caused due to missing files, especially in the files required to load the textures. Consequently, the dataset was carefully examined, and unusable objects were filtered out to ensure the quality and reliability of the remaining data. The remaining dataset includes objects of different types with various joint types such as revolute joints, prismatic joints, and fixed joints, which provide a diverse set of articulation mechanisms for realistic simulation and anomaly detection.

The objects in the dataset are represented as .urdf (Unified Robot Description Format) files. URDF files provide a standardized way to describe the physical properties, visual representation, and joint configurations of the objects. This format is particularly useful for simulating and manipulating articulated objects in robotics and computer vision research.

To manipulate the movement of the joints and create realistic simulations, the SAPIEN framework was used, as it seamlessly integrates with the .urdf file format. SAPIEN is a powerful physics simulation tool that allows for the precise control of joint movements and interactions within a simulated environment. By using SAPIEN, various joint positions and configurations can be accurately simulated. However, as SAPIEN is a physics simulation software it tries to avoid anomalous behaviour, difficulting the anomaly generation process as discussed in Section 3.1.2.1.

3.1.2 Data Generation Process

To capture a comprehensive set of normal and abnormal views, N cameras are strategically positioned around the azimuth plane of the object at a fixed distance and elevation. This setup ensures that the object is viewed from various angles, providing a rich dataset of visual information.

For each joint position, a subset of cameras *query_cameras* of size X is randomly selected from the N_n available cameras. The size X is randomly assigned between 0 and 5, ensuring variability and diversity in the captured views. This approach simulates different viewing conditions and perspectives, enhancing the dataset's robustness. Finally, a normal and an abnormal picture is taken using each camera in the *query_cameras* subset.

To simplify the simulation process and maintain control over the experimental variables, each object is assigned a single joint movement per simulation. This means that during the simulation, only one joint per object is moved. This approach ensures that the generated anomalies are clear and isolated, allowing for a focused analysis of the effects of individual joint movements. Additionally, this avoids exponentially increasing the number of generated multi-views as discussed in Section 3.1.3.

Statistic	**Count**
Total Images	134,236
Anomalies	62,811
Normal Conditions	71,425
Total Usable Objects	1,663
Object Categories	42

Table 3.1: Summary of Dataset Statistics

The final dataset comprises 134.236 images, with 62.811 of them depicting anomalies and 71.425 depicting normal conditions. These images are generated from 1663 usable objects, after filtering out unusable ones and taking into account the available storage space. This objects are categorized into 42 different object categories such as chairs, USBs or boxes. A summary of these statistics can be found in Table 3.1. While it is possible to expand the dataset with additional storage, the current size is sufficient for the purposes of this study.

3.1.2.1 Anomaly Generation Process

The anomaly generation process is a critical aspect of creating a dataset that effectively challenges and trains the anomaly detection model. This process involves the systematic introduction of various types of anomalies into the dataset, ensuring that each anomaly is realistic, detectable, and representative of potential real-world defects. The following sections describe the steps and considerations involved in generating these anomalies.

To cover a wide range of potential defects, six types of anomalies were identified and introduced into the dataset, with visual examples provided in Figure 3.1:

- **Rotational Anomalies:** Parts are rotated incorrectly, simulating an error in the assembly process. Since SAPIEN avoids these anomalous rotations, the .urdf file is modified by adding or modifying the random roll-pitch-yaw (RPY) value. If the

part has no RPY value the initial RPY value is considered $init = (0,0,0)$. Then a random rotation is selected between a maximum rotation angle in degrees, then converted to radians and added to a random axis. The value is set to 20 degrees. This effectively introduces both a rotation and a translation to the object part. However, due to limitations in SAPIEN this translation can not be corrected by applying the inverse translation. Despite these limitations, this type of anomaly has proven to introduce visible anomalies that can be realistic.

- **Translational Anomalies**: Parts of the object are displaced from their original positions, representing misalignments that can occur during manufacturing or use. To do so the .urdf file is modified in a similar way to the rotational anomalies. Nevertheless, the modified field is the origin field, indicating the origin with respect to its parent part. A maximum distance is established and a random value between its negative and positive value is chosen. This value is then added to a random axis. The value of the maximum distance was set to 0.5.

- **Removed Parts:** Parts are missing from the object, mimicking situations where components are omitted or lost. The process of removing a part involves selecting one of the leaf nodes from the .urdf file and removing it along with all joints connected to this part. The selection of a leaf node is done, as selecting a non-leaf node will result in a completely unrealistic anomaly by missing some critical elements in the object. Additionally, it would cause the SAPIEN engine to fail, as it would effectively create two disjoint objects.

- **Out-of-Range Joint Movements:** Joints are moved beyond their designed range, testing the model's ability to detect excessive or unsafe movements. This type of anomaly is restricted to object whose join has a limited range. To perform this anomaly in revolute joints, the minimum angle needs to be converted to a positive angle. After that, the difference between both the minimum and the maximum angle, $diff$, is extracted. A random value in the increase of the maximum limit is selected between $\frac{1}{8}\pi$ and $diff - \frac{1}{8}\pi$. Formally this can be represented as: $x \sim \text{Uniform}(\frac{1}{8}\pi, diff - \frac{1}{8}\pi)$. On the other hand for prismatic joints, the process also involves identifying the difference in the limits and the lower bound is selected to be the maximum between 30% of the difference and a minimum threshold of 0.4, to ensure visibility. Similarly, the upper bound is also dependent on the maximum over 70% of the difference and the minimum threshold. Finally,

a random value between the computed bounds is added to the upper limit of the joint.

- **Out-of-Axis Joint Movements:** Joints are moved out of their intended axes, representing misconfigured or damaged joints. This is performed by randomly selecting one of the rotating joints of the object and generating a random rotation around a perpendicular axis to the joint's axis, to ensure an anomaly is created. The angle for the rotation θ is selected using $\theta = \text{Uniform}(0.3, 2\pi)$.

- **Rotating Non-Rotating Joints:** Joints that should not rotate are rotated, challenging the model to identify inappropriate movements. A random joint out of the non-rotating joints in an object is selected. Additionally, a random axis is chosen. The applied rotation in radians θ to the selected axis is given by: $\theta = \text{Uniform}\left(\frac{1}{32}\pi, \frac{1}{4}\pi\right)$. This formula ensures the anomaly is not extremely abrupt, fitting closer to real-world scenarios. Finally, the rotation vector is converted to a quaternion to fit the SAPIEN pose format.

(a) Translational Anomaly (b) Removed Parts (c) Out-of-Range Joint Movement

(d) Out-of-Axis Joint Movement (e) Rotating Non-Rotating Joints (f) Rotational Anomaly

Figure 3.1: Examples of different anomaly types in articulated objects.

An anomaly type is randomly selected from the predefined set of anomalies for every camera in *query_cameras*. This randomness ensures a diverse and comprehensive dataset, covering all possible defect scenarios. The chosen anomaly is then applied to the object and a picture is taken.

After applying the anomaly, it is crucial to verify that the defect is visible and significant enough to be detected by the model. This is done by calculating the masks of the normal object and the anomalous object. The Intersection over Union (IoU) metric is used to measure the overlap between these masks. The IoU is computed as follows:

$$IoU = \frac{A \cap B}{A \cup B} \tag{3.1}$$

where A is the anomaly image and B is the corresponding normal image.

If the IoU is greater than a predefined threshold, it indicates that the anomaly is not sufficiently visible. In such cases, the anomaly is discarded. This ensures that only meaningful and detectable anomalies are included in the dataset. For the creation of this dataset, the threshold of 0.98 was used, as some objects were composed of extremely small parts, causing the IOU values to be greater. Nevertheless, after visual inspection of the anomalies, the threshold has proven enough for the anomalies to be visible.

Contrary to [1] I argue that anomalies resulting in two disjoint bodies can represent realistic scenarios in manufacturing when one object part is not correctly assembled and falls from the main body. Thus, this anomalies are not removed as part of the quality control mechanism.

To ensure the robustness of the dataset, the process attempts to generate a successful anomaly up to 10 times. If a visible anomaly is not achieved within these attempts, the process moves on to the next object or anomaly type. This iterative approach guarantees that each anomaly introduced into the dataset is both realistic and detectable, while still producing anomalies.

Moreover, the bounding boxes of the generated anomalies, along with the camera parameters and IoU scores of the generated anomalies, are saved into a JSON file. Although only the bounding boxes are used during the training process, the saved IoU scores and camera parameters may be beneficial for future research.

3.1.3 Multi-view rendering

The process of rendering multi-views is similar to the normal data generation but the number of cameras N_m involves significantly fewer cameras, this means $N_m << N_n$.

This is done to simulate a continuous space in the query images, but a discrete space for the multi-views, as the query images can be taken in any position. Images are captured across different time steps where the joint is gradually moved at a constant velocity. This continues until the joint reaches its maximum movement or completes a full cycle in the case of infinite movements, such as rotations.

The velocity differs depending on the type of joint. For rotating joints, the angular velocity ω is set to 0.2 radians per time step. In contrast, the velocity for prismatic joints is adjusted based on the total range of the joint, to fit the movement into 20 total time steps. However, if the joint limits are too small to be divided into 20 steps that SAPIEN can accurately simulate, a single time step is used to accommodate the entire range of motion.

At each timestep, images are captured from the selected subset of cameras, providing a dynamic view of the object as the joint moves. This method ensures that the dataset includes a sequence of images that depict the gradual movement of the joint, which is essential for understanding the effects of the movement over time.

Nevertheless, SAPIEN objects possess photo-realistic textures. This is not realistic in many industrial setups, where companies may have a textureless 3D render. Therefore the segmentation image is taken, which is a feature provided by SAPIEN engine, and the photo-realistic texture is lost in this process. Furthermore, the picture is converted to a grey-scale image to closer fit industry setups.

3.1.4 Visibility and Depth Information

For both normal data and multi-views, additional processing is performed due to the training requirements of the model. The process starts with obtaining the captured image from SAPIEN's camera. The 3D points are in OpenGL space and must be transformed into world coordinates using the camera's model matrix. These world coordinates are flattened into a 2D array for easier processing.

A KD-Tree is built from the flattened world coordinates to facilitate efficient nearest-neighbor searches. For each 3D point of interest, the KD-Tree is queried to compute the nearest distance to the camera points. A visibility threshold is applied to determine if each point is visible from the camera's perspective, which is set to 0.1.

The world coordinates are then downsampled for efficiency. The 3D coordinates are projected into 2D image coordinates using the camera's intrinsic K and extrinsic RT parameters, and the resulting 2D points are adjusted for the image resolution.

Finally, the 2D points and their visibility flags are saved into a .npy file. The 3D world points are also saved, providing comprehensive data that can be used for further analysis and future research. This data is crucial for the 2D-3D correspondence training in the model. On the other hand, the camera parameters are saved into a JSON file.

3.2 Correspondence Matching Transformer

Based on [1] and its promising results for the Correspondence Matching Transformer (CMT), this architecture is used to detect anomalies in articulated objects, as the adapted multi-view mechanism seamlessly integrates into the CMT architecture. This section will describe the CMT architecture and its main components.

The CMT employs the ResNet18 feature pyramid network (ResNet18-FPN) [25] as its feature encoder. This encoder, represented as $\phi : \mathbb{R}^{3\times H\times W} \rightarrow \mathbb{R}^{d\times \frac{H}{8}\times \frac{W}{8}}$, reduces the input size by a factor of 8 through the network (with $h = \frac{H}{8}$ and $w = \frac{W}{8}$). Following the extraction of features from the query image q and each reference view v, the resulting feature maps $\phi(q)$ and $\phi(v)$ are reshaped into $d \times n^q$ dimensional matrices, denoted as f^q and f^v, respectively, where $n^q = h \times w$. Each column in f^q and f^v represents a d-dimensional local feature. The notation $f[.j]$ indicates the j-th local feature or patch encoding, with each encoding approximately corresponding to a local patch in the input image due to the convolutional encoder's locality.

The multi-view representation provides a straightforward and efficient model design through a shared feature encoder, but it can obscure 3D information, making it challenging to relate local features across different views accurately. The multi-view images are enhanced with 3D information to address this issue. For each patch encoding $f^v[.j]$, the corresponding image patch in v is identified and the 3D position of the corresponding patch $x_j \in \mathbb{R}^3$ in world coordinates is computed using known camera intrinsic and extrinsic parameters. After this, Fourier encoding is applied to obtain a higher-dimensional vector for each x_j. This is then additionally processed through a multi-layer perceptron (MLP) block to get a d-dimensional 3DPE. Formally,this mapping is denoted by $\gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^d$.

Unlike the 2D standard positional encoding used in transformer models [10], 3DPE encodes 3D object geometry in world space. For each f^v including n^q patch encodings, a corresponding $d \times n^q$ dimensional matrix p^v is computed. Next, f^v and p^v over N views are gathered and each set is concatenated along their second dimensions, resulting in $F^v \in \mathbb{R}^{d\times n^v}$ and $P^v \in \mathbb{R}^{d\times n^v}$ respectively, where $n^v = N \times n^q$. Augmenting F^v with

P^v creates a novel hybrid 2D-3D representation. This is caused by incorporating explicit 3D information into the 2D multi-view images.

3.2.1 Correspondence-Guided Attention (CGA)

The Correspondence-Guided Attention (CGA) network, denoted as φ, efficiently computes the correlations across two modalities to predict the anomaly label. The CGA consists of B consecutive transformer blocks, each indexed by subscript b.

Each transformer block begins by concatenating the feature matrices F^v and P^v along their first dimension. The resulting $2d \times n_v$ dimensional matrix is then reduced to a $d \times n^v$ dimensional matrix $\bar{F}^v$ through a linear projection layer $\alpha^{(b)} : \mathbb{R}^{2d} \rightarrow \mathbb{R}^d$:

$$\bar{F}^v_{(b)} \leftarrow \alpha_{(b)}(\begin{bmatrix} F^v \\ P^v \end{bmatrix}) \tag{3.2}$$

Next, a self-attention operation (SA) is applied to the query features $f^q_{(b)}$, where $f^q_{(1)} = f_q$:

$$\bar{f}^q_{(b)} \leftarrow \mathrm{SA}(f^q_{(b)}) \tag{3.3}$$

The query $Q_{(b)} \in \mathbb{R}^{d \times n_q}$ and key-value matrices $K_{(b)} \in \mathbb{R}^{d \times n^v}$ and $V_{(b)} \in \mathbb{R}^{d \times n^v}$ are then computed by applying linear projections W^Q, W^K, and $W^V \in \mathbb{R}^{d \times d}$ respectively:

$$Q_{(b)} \leftarrow W^Q \bar{f}^q_{(b)} \tag{3.4}$$

$$K_{(b)} \leftarrow W^K \bar{F}^v_{(b)} \tag{3.5}$$

$$V_{(b)} \leftarrow W^V \bar{F}^v_{(b)} \tag{3.6}$$

The outputs $O_{(b)}$ are then passed to the top-k sparse cross-attention (TKCA) module:

$$O_{(b)} \leftarrow \mathrm{TKCA}(Q_{(b)}, K_{(b)}, V_{(b)}, M) \tag{3.7}$$

where

$$\mathrm{TKCA}(Q,K,V,M) = \mathrm{softmax}\left(T^M_k(\frac{QK^T}{\sqrt{d}})\right)V \tag{3.8}$$

and

$$\mathrm{T}_k^M(A)_{ij} = \begin{cases} A_{ij}, & \text{if } M_{ij} \in \mathrm{top}_k(M[i.]) \\ -\infty, & \text{otherwise} \end{cases} \tag{3.9}$$

The T_k^M operation selects the k most similar features from the multi-view representation for the i-th query feature. To compute M, an auxiliary function $\beta : \mathbb{R}^d \to \mathbb{R}^d$ is used, instantiated as a four-layer MLP followed by channel-wise normalization. This projects f^q and each view in F^v to a view-agnostic feature space:

$$M = \beta(f^q)^T \beta(F^v) \in \mathbb{R}^{n^q \times n^v} \tag{3.10}$$

The top-k sparse attention mechanism in TKCA improves efficiency by focusing only on the most relevant features. After the cross-correlation, standard residual addition, normalization, and feedforward (FFN) layers are applied to obtain f^q as input to the next block $b+1$:

$$O_{(b)} \leftarrow \mathrm{Norm}(O_{(b)} + Q_{(b)}) \tag{3.11}$$

$$O_{(b)} \leftarrow \mathrm{Norm}(\mathrm{FFN}(O_{(b)}) + O_{(b)}) \tag{3.12}$$

$$f^q_{(b+1)} \leftarrow O_{(b)} \tag{3.13}$$

Multiple heads are used in the attention mechanism, with the outputs from multi-head attention concatenated and passed through a linear projection to derive the final attention results. Throughout the transformer blocks, the output state of the [tok] token develops a consolidated representation enriched by learned shape-image correlations, which is then used as input to the classification head.

3.3 View-Agnostic Local Feature Alignment

In addition to the CMT, [1] also introduces an auxiliary task, as image-level supervision alone is not enough to capture meaningful correlations between the multi-views V and query q. This auxiliary task proved helpful in detecting anomalies. The primary objective of VLFA is to densely align corresponding parts between query images and related views. This is achieved by mapping the local features f^q and f^v to a view-agnostic space via the mapping function β, ensuring that local features corresponding

to the same object part are mapped to similar points irrespective of the viewpoint from which the image is captured.

Given that the viewpoints of the query image q are unknown, ground-truth correspondences between the query and reference views cannot be directly obtained through inverse rendering. To address this, a self-labelling strategy is employed to generate pseudo-correspondences. This strategy involves finding the most similar local feature in the reference view for each local feature in the query at each training step, after mapping their features to the view-invariant space and normalising them. Formally this is denoted as:

$$\hat{c}_i = \arg\max_j \left(\left(\frac{\beta(f^q[.i])}{\|\beta(f^q[.i])\|}\right)^T \cdot \frac{\beta(f^v[.j])}{\|\beta(f^v[.j])\|} \right) \tag{3.14}$$

The pseudo-label for each z^q is stored in a lookup table $\hat{P}(q,v,i)$. Another lookup table $\hat{P}(q,v,i)$ stores the remaining set of reference view and index values that do not correspond to the pseudo-label. Moreover, the lookup tables $\hat{P}$ and $\hat{N}$ are further used as positive and negative correspondences respectively to minimize a contrastive loss over each query-view pair. This is formalized by Equation 3.15, where where τ is a temperature parameter and $z^v_+ = z^v_{\hat{P}(q,v,i)\cdot}$.

$$\ell_{va}(q,v) = \sum_{i=1}^{n_q} -\log \left(\frac{\exp(z^q_i \cdot z^v_+/\tau)}{\exp(z^q_i \cdot z^v_+/\tau) + \sum_{j\in\hat{N}(q,v,i)} \exp(z^q_i \cdot z^v_j/\tau)} \right) \tag{3.15}$$

However, generating pseudo-labels for all query features across all views is computationally expensive. Therefore the pseudo-labels are only computed for randomly sampled views each training session. Learning correspondences through self-learning alone can be noisy, especially in the presence of the domain gap between query and reference views. Hence, the known viewpoints of the multi-view images are exploited by densely aligning their local features in each view pair v, v' after computing the ground truth dense correspondences between them and discarding those occluded in one of the views.

By aligning different views using their ground truth labels, a more accurate correspondence learning between query images and views is enabled, as the parameters of the projection β are shared across the two domains. Two lookup tables $P(v,v')$ and $N(v,v')$ are created to store the positive and negative correspondences between two views. Following the creation of the tables they are randomly subsampled. After mapping them to the view-invariant space and normalizing them, the loss function described in Equation 3.16 is computed and minimized for the pairs in the lookup tables. Where

$L_{va}(\hat{P},\hat{N})$ where and $L_{va}(P,N)$ are the alignment loss functions over query-view pairs and view-view pairs respectively, and a is a loss balancing weight set to 0.5.

$$L_{bce}(D) + aL_{va}(\hat{P},\hat{N}) + (1-a)L_{va}(P,N) \tag{3.16}$$

3.4 Heuristic-based multi-view selection

The CGA has been proven to learn how to effectively use multi-views, prioritizing those that provide more information about the potential anomaly [1]. It achieves this through a sparse cross-attention mechanism. Despite being more computationally efficient than the standard cross-attention mechanism, it still introduces significant time and space complexity. Hence, training the model with all multi-views generated by the process described in Section 3.1.3, becomes impossible.

The time complexity for the standard cross-attention mechanisms is $O(n^q \cdot n^v \cdot d)$, where n^q is the query sequence length and n^v is the length of the key-value sequence [13]. For the sparse attention mechanism, the time complexity is reduced, because the attention weights are calculated over the top k keys and then the weighted sum of the corresponding values is computed, resulting in $O(n^q \cdot k \cdot d)$ time. As the number of multi-view images grows this becomes intractable.

In a top-k sparse cross-attention mechanism, the space complexity is significantly reduced compared to the standard cross-attention mechanism by only considering the top-k keys for each query. The input embeddings for the queries and keys require $O(n^q \cdot d)$ and $O(n^v \cdot d)$ space, respectively. In a typical cross-attention mechanism, the attention score matrix has a size of $O(n^q \cdot n^v)$. However, in the top-k sparse cross-attention, each query only stores the top-k attention scores, resulting in a space complexity of $O(n^q \cdot k)$ for the attention scores. Additionally, storing the indices of the top-k keys for each query also requires $O(n \cdot k)$ space. The attention weights, which are derived from the top-k scores, similarly require $O(n^q \cdot k)$ space. Moreover, The output embeddings, computed as a weighted sum of the top-k value embeddings for each query, need $O(n^q \cdot d)$ space. Finally, combining these components, the total space complexity for the top-k sparse cross-attention mechanism is $O((n^q + n^v) \cdot d + n^q \cdot k)$. This demonstrates a substantial reduction in space requirements, particularly when k is much smaller than m. However, this space complexity is still largely dependent on m and therefore dependent on the multi-views, as described in Section 3.2, $n^v = N \times n^q$. Resulting in extremely large memory requirements when the number of multi-view

images is greater than 20. Furthermore, if applied to all available multi-view images, with V denoting the total number of multi-view images, the space complexity would be $O((n^q + V \cdot n^q) \cdot d + n^q \cdot k)$. This exceeds the available computing power and the memory specifications of the available GPUs. Moreover, for this study the joint movement is limited to one joint, in an industrial setting the number of movable joints could be larger, increasing the number of required multi-views V.

Hence, a method to discriminate between the views using heuristics is introduced to substitute the random sampling which was originally used when $N < V$, where N is the number of selected multi-views and V is the total number of multi-views. Multi-views and query images have one main difference: the query image is rendered with texture, and the multi-views are rendered textureless and in grey scale. Because of this, a preprocessing step is applied to convert both images to the same space, first by removing any potential backgrounds and then applying a mask for every non-zero pixel and creating a grey-scale copy of the query image. This image is then compared to every multi-view using the Structural Similarity Index (SSIM).

The SSIM is a perceptual metric, introduced by Wang et al [40] that quantifies the similarity between two images by comparing three main components: luminance, contrast, and structure. Formally for two images x and y, the SSIM index is computed as:

$$\text{SSIM}(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{3.17}$$

where $l(x,y)$, $c(x,y)$, and $s(x,y)$ represent the luminance, contrast, and structure comparisons respectively, and α, β, and γ are parameters that adjust their relative importance. The luminance, contrast, and structure comparisons are defined as represented in Equations 3.18, 3.19 and 3.20 respectively.

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{3.18}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{3.19}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{3.20}$$

The symbols μ_x and μ_y are the mean intensities, σ_x and σ_y are the standard deviations, and σ_{xy} is the covariance of x and y. The constants C_1, C_2, and C_3 are used to stabilize the division with weak denominators.

The multi-view images are then sorted using their SSIM scores and the top R images are retrieved for the model. This approach allows leveraging all multi-view images while focusing only on the most similar ones, effectively performing a task that the TKCA would undertake if sufficient computing power were available, but at a fraction of the cost. More specifically the time and space complexity are $O(V \log V)$ and $O(V)$ respectively. However, it is worth noting that TKCA would likely learn more advanced patterns that the SSIM might not detect.

Based on the results obtained in Section 4.3 a variation of this heuristic was designed to force the inclusion of different angles into the selected multi-views. This heuristic would be referred as Heu-B, for the rest of this work. The proposed change is the selection of the image with the highest score from each angle, starting with the angles with higher scores. If the desired number of multi-views is reached the process is instantly stopped. Otherwise, if all angles are included and some images are missing the rest of the images are chosen following their scores. By including all angles, while leveraging the similarity score, this heuristic should include more 3D information into the correspondence matching and thus, limit the weakness of the heuristic-based CMT, which fails in detecting natural alignments of the joint as discussed in Section 4.3.

Finally, due to the inability to test the Heu-B approach a mixture between random selection and the heuristic-based multi-view selection is implemented. This implementation has a calibration parameter λ which determines the percentage of multi-view images selected using the heuristic. The rest of the images are randomly selected from the subset of unselected images. This will be named as Heu-C in the rest of the work.

Chapter 4

Experiments

4.1 Implementation

For the CMT implementation, [1] was followed. Therefore the encoder ϕ takes a 256×256 image in RGB format, resulting in a $3 \times 256 \times 256$ and returns a $128 \times 32 \times 32$ feature block. Additionally, the CGA is configured with three transformer blocks $(B = 3)$, with each of them applying an 8-headed attention mechanism. The k value for the TCKA is set to 100. Moreover, following [1] the number of selected views was set to 20 as this was the maximum possible for the available computing units. This resulted in a maximised performance. Basic data augmentation is applied to the query images, including random horizontal flips and random cropping of 224×224 regions, followed by resizing the cropped regions back to the original size of 256×256. this is done aiming to prevent overfitting to the training data and enhance the models' generalization capabilities. The model is trained for 20 epochs using 4 Titan RTX GPUs, with a batch size of 8 per GPU. The Adam optimizer [21] is used with a learning rate of $2 \times 10^{(-5)}$.

Using this computing resource the training of the CMT model leveraging the modified heuristic, HEU-B, was not possible, as the cost of the training would have been extremely high. The simulations returned an estimated cost of 1920 GPU hours, which effectively resulted in 20 days of training. Therefore no experimental results of the heuristic modification exist.

The dataset was randomly split, with 80% of the objects used for training and 20% for testing. This split is done by objects, not by query images, to ensure that the model does not learn specific patterns of individual objects, which could negatively impact the assessment of generalization performance.

4.2 Evaluation Metrics

To objectively assess the performance of the proposed Correspondence Matching Transformer (CMT) architecture, two two key evaluation metrics are employed: Area Under the Receiver Operating Characteristic Curve (AUC) and Accuracy. These metrics provide a comprehensive view of the model's ability to detect anomalies in articulated objects.

4.2.1 Accuracy

Accuracy is used to assess the performance of the CMT model in detecting anomalies. It is defined as the ratio of correctly predicted instances, considering both true positives and true negatives to the total number of instances.

Accuracy is calculated using the following formula:

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

Where TP is the number of True Positives, TN is the number of True Negatives, FP is the number of False Positives and FN is the number of False Negatives.

It provides a straightforward measure of the model's overall correctness in its predictions. Nevertheless, it is important to note that in the context of anomaly detection, accuracy alone may not be sufficient to evaluate the model's performance due to the potential class imbalance. Despite this is not the case for this specific use case as discussed in Section 3.1, it is important to combine this metric with other metrics to verify that the model is correctly detecting both classes. Hence, combining AUC with accuracy gives a more nuanced understanding of the model's efficacy.

4.2.2 Area Under the Receiver Operating Characteristic Curve (AUC)

The AUC metric is widely used in classification problems to measure the performance of a model. It represents the degree of separability achieved by the model in distinguishing between classes (in this case, normal and anomalous objects). The Receiver Operating Characteristic (ROC) curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The AUC is the area under this ROC curve. The True Positive Rate is given by the ratio of correctly identified positive instances (anomalies) to the total number of actual positive instances. It is defined as:

$$TPR = \frac{TP}{TP + FN} \tag{4.2}$$

Meanwhile, FPR is as the ratio of incorrectly identified positive instances to the total number of actual negative instances. It is given by:

$$FPR = \frac{FP}{FP + TN} \tag{4.3}$$

Mathematically, the AUC is calculated as follows:

$$AUC = \int_0^1 TPR(x)dx \tag{4.4}$$

Where TPR is the True Positive Rate, also known as sensitivity or recall and x is the False Positive Rate (FPR). A higher AUC value indicates better performance, with a value of 1.0 representing a perfect model and a value of 0.5 indicating a model with no discriminative power, equivalent to random guessing.

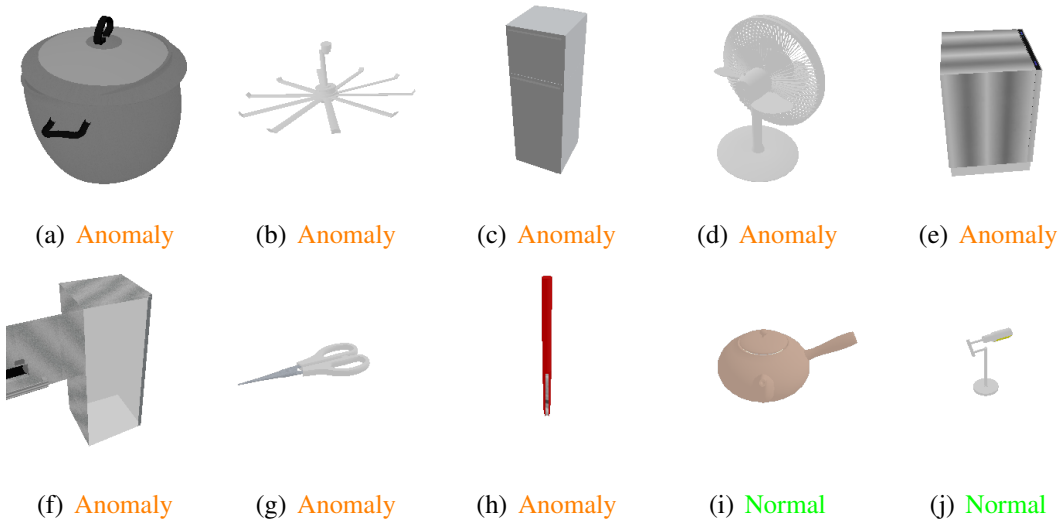
4.3 Results

This section describes the experimental results obtained in this study. As baselines for this work, ResNet18-FPN and VIT [10]. Additionally, Resnet18-FPN with attention mechanism is also used as a baseline following [1]. The obtained results for these baselines along with the CMT-variants tested are presented in Table 4.1.

Model	**AUC**	**Accuracy**
ViT	0.5053	0.5338
ResNet18-FPN	0.6203	0.5840
ResNet18-FPN + Attention	0.7056	0.6276
CMT	0.7056	0.6429
CMT (Heuristic)	0.6603	0.5941
CMT + Heuristic in Test	0.6961	0.6335
CMT + New Heuristic in Test	0.6932	0.6323
CMT + HEU-C $\lambda = 0.5$	0.6837	0.6372

Table 4.1: AUC and Accuracy Scores for the models.

Even though the performance of the VIT was the best along the baselines in [1], in this study this model completely fails to capture any patterns in the data, predicting in every case the image belongs to the normal class. Nevertheless, ResNet18-FPN with attention mechanism achieves a great score considering the complexity of the task serving as a great baseline.



(a) Anomaly (b) Anomaly (c) Anomaly (d) Anomaly (e) Anomaly

(f) Anomaly (g) Anomaly (h) Anomaly (i) Normal (j) Normal

Figure 4.1: Query Images where CMT succesfully classifies instances and attention Res-Net fails.

The CMT model achieves a similar AUC score to the ResNet18-FPN with attention mechanisms, however, its slightly better accuracy suggests it captures more complex patterns in the data, perhaps identifying less obvious anomalies. To explore this even further a direct comparison between instances correctly classified by CMT and incorrectly classified by ResNet18-FPN with attention mechanisms is carried over. A sample of these instances is presented in Figure 4.1, which reinforces the hypothesis as the anomalies are subtle in the instances correctly classified by the CMT model and incorrectly classified by the ResNet18-FPN with attention mechanisms. To gain further insights into the performance of the CMT model, an analysis of the bounding boxes was performed. The outputs demonstrated that in many instances the bounding box was inaccurate, as represented in Figure 4.2, where the green box represent the ground truth box and the red box the predicted box. Hence, the model failed in the 2D-3D correspondence task between multi-views and query images in some complex instances.

Thus, the CMT model trained with the proposed heuristic was examined. However, despite drastically improving the bounding box accuracy from 0.2175 to 0.2705 the overall accuracy experienced a drop-off. The model identified anomalous instances better than the base CMT model, improving from 58.78% to 69.66%. However, the performance in normal query images decreased from 69.09% to 50.51%. This may be produced by the model identifying the natural movement of the joint as an anomaly.

Figure 4.3 illustrates the bounding boxes generated for normal instances by the model using the heuristic for both training and testing. The picture exhibits various

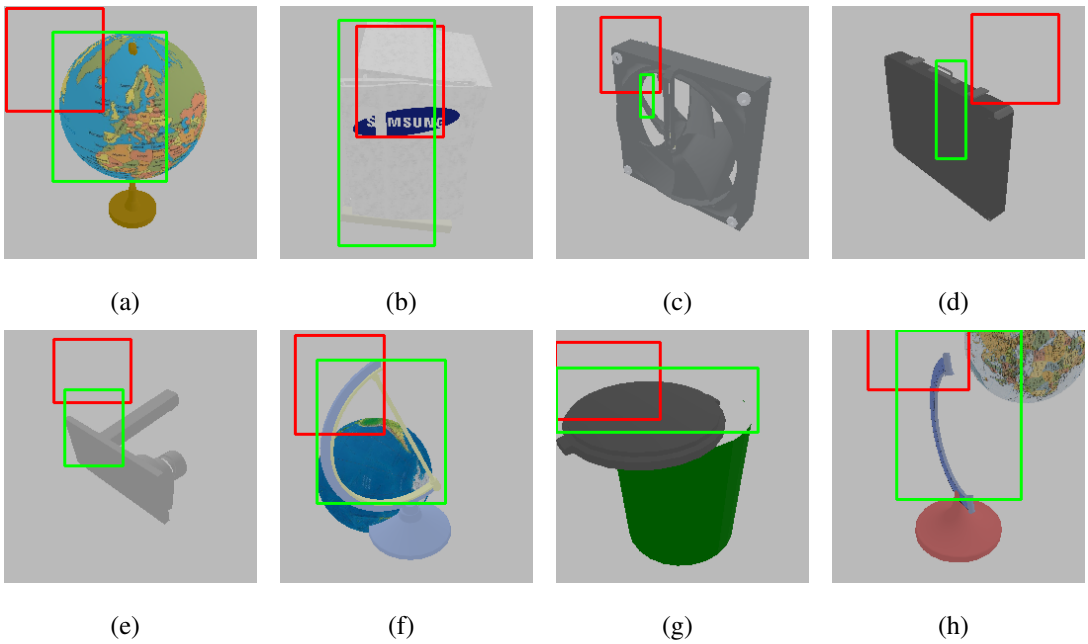


Figure 4.2: Query Images and depicted anomaly boxes.

instances where the detected box aligns with the natural movement of the joint. A possible explanation for this is the model losing 3D information using the heuristic, as the heuristic would select the closest images and therefore the closest camera angles to the query image effectively eliminating distant angles and missing crucial 3D information.

Aiming to explore if a hybrid between the more robust 3D correspondence learned by the original model and the heuristics capabilities in improving the anomaly detection can further enhance the models' performance, an experiment using the heuristic only during testing is performed. Despite this showing a slight improvement in the accuracy of detecting the abnormal instances from 58.78% to 60.46% there is a similar loss in performance for the normal ones. Additionally, the bounding box accuracy is similar between the CMT using heuristics in inference time and the CMT without heuristics. Hence, there is no clear improvement in using the heuristic only in inference time.

With the same goal, HEU-C was examined. The results proved that the performance with a randomly selected subset of multi-views dramatically improves performance on the normal images as the performance on these images increased up to 0.7837 accuracy. Nevertheless, a trade-off between correct classification of normal and abnormal instances is clear, with the later experimenting a fall in the accuracy up to 0.4691. The bounding box accuracy of the model is as expected between the CMTs and the CMTs with a heuristic. The improved bounding box accuracy suggests that despite the accuracy being lower for the abnormal images, the model is learning better to detect

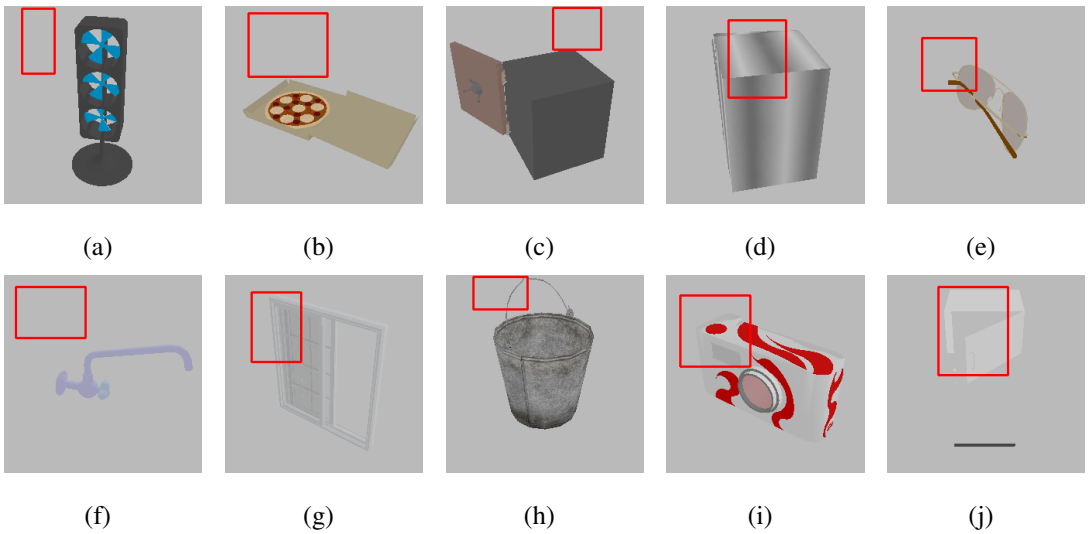


Figure 4.3: Predicted Bounding Boxes using CMT with heuristics

anomalies. More exploration into the λ parameter would be helpful. However, for this work, the computing resources were limited and an extensive hyperparameter search is not possible, as each run requires around 350 GPU hours.

Due to the dataset's complexity, as it uses an extensive number of categories, a study on the accuracy of the CMT model, the CMT model incorporating the designed heuristic and the ResNet18-FPN with attention mechanisms is performed. The results of this study are exhibited in Table 4.2. The 'Chair' category showed high accuracy across all models, with CMT achieving 82.19%, ResNet18-FPN with attention mechanisms at 83.05%, and CMT with heuristics at 84.76%. This suggests that all models are well-tuned for detecting chairs, with CMT with heuristics slightly outperforming the others. This aligns with the finding of [1], with the high accuracy suggesting that anomaly detection in chairs is easier than in smaller objects where the anomalies might be difficult to detect even for the human eye. In the 'Luggage' category, ResNet18-FPN outperformed both CMT and CMT with heuristics with a notable accuracy of 71.86%, compared to CMT achieving 54.49% and CMT with heuristic 55.09%. This highlights ResNet18-FPN robustness in this particular category and its strength as a baseline for this specific task. Luggage has a high variation in the possible anomalies if the anomaly occurs in the main body it can easily be identified. Nevertheless, when the anomaly occurs on the wheels these anomalies produce a challenge for the models. Furthermore, a significant performance gap was observed in the Keyboard category, where CMT and CMT with heuristics had accuracies of 83.84% and 82.56% respectively, while Resnet18-FPN with SA achieved 75.52%. This suggests that variations of the CMT

model are better at handling extremely small pieces and their corresponding anomalies, similar to the keys on a keyboard. This hypothesis is strengthened by the difference in performance exhibited in the Remote category, where CMT led with 82.60% accuracy and CMT with heuristics followed closely with 80.92%. Meanwhile, ResNet18-FPN with self-attention accomplished 76.88% accuracy.

The analysis reveals that while all models exhibit strong performance across various categories, certain models excel in specific areas. For instance, ResNet18-FPN with self-attention outperforms in categories like Luggage, whereas CMT and CMT with heuristic show superior performance in categories such as Keyboard and Remote. Therefore, models should be selected depending on the use case and specific objects. In instances where false positives are costly, the CMT model should be employed without heuristics, whereas when the cost of missing an anomaly is high the CMT model with heuristic is the better option.

Table 4.2: Category accuracies for CMT, RESNET, and CMT HEU models

Category	CMT	RESNET	CMT HEU
box	0.495475	0.522624	0.542986
bucket	0.591453	0.576068	0.579487
camera	0.576786	0.602679	0.529464
cart	0.540230	0.540230	0.534483
chair	0.821888	0.830472	0.847639
coffeemachine	0.598214	0.545918	0.602041
dishwasher	0.697947	0.680352	0.674487
dispenser	0.711624	0.650399	0.694765
eyeglasses	0.643098	0.686869	0.634680
fan	0.598256	0.585446	0.592259
faucet	0.600304	0.642857	0.592705
foldingchair	0.531746	0.642857	0.531746
globe	0.667568	0.700000	0.658108
kettle	0.568182	0.660839	0.597902
keyboard	0.838400	0.755200	0.825600
kitchenpot	0.656347	0.634675	0.640867
knife	0.602964	0.620905	0.581903
lamp	0.660985	0.503788	0.668561

Continued on next page

Table 4.2 – *Continued from previous page*

Category	CMT	RESNET	CMT HEU
laptop	0.681592	0.659204	0.681592
lighter	0.740933	0.626943	0.709845
luggage	0.544910	0.718563	0.550898
mouse	0.639024	0.551220	0.643902
oven	0.712195	0.663415	0.746341
pen	0.761246	0.713495	0.757093
phone	0.686888	0.522505	0.549902
pliers	0.710526	0.631579	0.743421
printer	0.683908	0.517241	0.672414
refrigerator	0.609861	0.580922	0.607717
remote	0.826038	0.768799	0.809203
safe	0.633929	0.622024	0.641369
scissors	0.711538	0.733974	0.708333
stapler	0.726804	0.654639	0.737113
suitcase	0.618902	0.567073	0.591463
switch	0.598639	0.564626	0.553288
table	0.570417	0.556255	0.572777
toaster	0.585227	0.568182	0.539773
toilet	0.619850	0.610487	0.595506
trash$_c$*an*	0.651917	0.545723	0.660767
trashcan	0.640728	0.625828	0.604305
usb	0.666667	0.707921	0.665017
washingmachine	0.597403	0.512987	0.551948
window	0.646330	0.731173	0.648236

Chapter 5

Conclusions

This study addressed the challenges associated with anomaly detection in articulated objects. It did so by developing a dataset encompassing a range of anomalies and objects of different categories. Additionally, a novel multi-view rendering is created, which captures the joint movement of the object. Various models were evaluated on this newly created dataset, especially focusing on the performance of the CMT model, which demonstrated promising performance in similar tasks. Furthermore, as the increasing number of multi-view images introduced a computational problem, a heuristic to discriminate between these images was introduced and tested.

The vanilla CMT model yielded the best overall detection accuracy, however, the CMT model using the heuristics resulted in a higher detection rate in anomalous instances and a better box accuracy. Nevertheless, the inclusion of the heuristic limited the 3D information the model received and this made the differentiation between natural joint movement and anomalous instances for the model difficult. Thus, the heuristic is helpful when missing an anomalous instance is costly, and the vanilla CMT model is better suited for instances when labelling a normal instance as anomalous is expensive.

This research contributes significantly to the anomaly detection field. The main contribution is the newly created dataset, which could be used in future research. Furthermore, the insights on the performance of the CMT model demonstrate its ability to generalise across various object types and identify anomalies in articulated objects. Despite the remarkable performance, this model exhibits room for improvement in the multi-view selection process, finding a method that yields a more balanced performance between detecting anomalous instances and correctly labelling normal images. Similarly, exploring how a helper task in detecting the joints of an object affects the model performance could be beneficial, however, this might influence the model to predict as

normal all anomalies occurring in the joint. Additionally, this research focuses on one joint due to the explainability offered by this approach. Future research could focus on how the increased number of joints affects the performance of the CMT model. Moreover, the dataset is limited to the physics simulator used in this study and the types of anomalies, which may not cover all real-world scenarios. Expanding the anomaly types could refine this work and enhance the real-world applications.

Bibliography

[1] Anonymous. Looking 3d: Anomaly detection with 2d-3d alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, number 7563, 2024. Confidential review copy. Do not distribute.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad - a comprehensive real-world dataset for unsupervised anomaly detection. 06 2019.

[4] Hakan Bilen, Basura Fernando, Efstratios Gavves, and Andrea Vedaldi. Action recognition with dynamic image networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2799–2813, dec 2018.

[5] Emmanuel J. Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis?, 2009.

[6] Diego Carrera, Fabio Manganini, Giacomo Boracchi, and Ettore Lanzarone. Defect detection in sem images of nanofibrous materials. *IEEE Transactions on Industrial Informatics*, 13:551–561, 04 2017.

[7] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks, 2019.

[8] Feiyang Chen, Nan Chen, Hanyang Mao, and Hanlin Hu. Assessing four neural networks on handwritten digit recognition dataset (mnist), 2019.

[9] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML*

PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18, pages 3–17. Springer, 2019.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[11] Belmiro P.M. Duarte, Nuno M.C. Oliveira, and Lino O. Santos. Dynamics of quality improvement programs – optimal investment policies. *Computers Industrial Engineering*, 91:215–228, 2016.

[12] Sarah M. Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[13] Mozhdeh Gheini, Xiang Ren, and Jonathan May. Cross-attention is all you need: Adapting pretrained transformers for machine translation, 2021.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[15] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[16] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Trans. Graph.*, 36(6), nov 2017.

[17] Trevor Hastie Hui Zou and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[18] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video, 2019.

[19] Xinwei Jiang, Junbin Gao, Xia Hong, and Zhihua Cai. Gaussian processes autoencoder for dimensionality reduction. pages 62–73, 05 2014.

[20] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction, 2022.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[22] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research).

[23] Ping Li, Trevor J. Hastie, and Kenneth W. Church. Very sparse random projections. KDD '06, page 287–296, New York, NY, USA, 2006. Association for Computing Machinery.

[24] Jia Liang, Huanyi Shui, Rajesh Gupta, Devesh Upadhyay, and Eric Darve. Transfer learning for anomaly detection in rotating machinery using data-driven key order estimation. *TechRxiv*, Feb 2024. Preprint.

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

[26] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-level reconstruction and motion analysis for articulated objects, 2023.

[27] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20402–20411, June 2023.

[28] Manpreet Singh Minhas and John Zelek. Anomaly detection in images, 2019.

[29] Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '18. ACM, July 2018.

[30] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):1–38, March 2021.

[31] Guansong Pang, Chunhua Shen, Huidong Jin, and Anton van den Hengel. Deep weakly-supervised anomaly detection, 2023.

[32] Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection, 2020.

[33] Tomá Pevný. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102:275–304, 2016.

[34] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

[36] Babak Saleh, Ali Farhadi, and Ahmed Elgammal. Object-centric anomaly detection by attribute-based reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[37] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. *Kernel Principal Component Analysis*, volume 1327, pages 583–588. 10 2006.

[38] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos, 2019.

[39] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders, 2017.

[40] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[41] Yijia Weng, Bowen Wen, Jonathan Tremblay, Valts Blukis, Dieter Fox, Leonidas Guibas, and Stan Birchfield. Neural implicit representation for building digital twins of unknown articulated objects, 2024.

[42] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[43] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection, 2015.

[44] Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen, Oliver Van Kaick, Hao Zhang, and Hui Huang. Rpm-net: recurrent prediction of motion and parts from point cloud. *ACM Transactions on Graphics*, 38(6):1–15, November 2019.

[45] Wenchao Yu, Wei Cheng, Charu C. Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. KDD '18, page 2672–2681, New York, NY, USA, 2018. Association for Computing Machinery.

[46] Joey Tianyi Zhou, Jiawei Du, Hongyuan Zhu, Xi Peng, Yong Liu, and Rick Siow Mong Goh. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, 14(10):2537–2550, 2019.