An Attention-Enhanced Student–Teacher Framework for Structural and Logical Anomaly Detection in Industrial Settings

by

Hafiz A. Amjad

A thesis submitted to the
School of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Masters of Science in **Computer Science**

Faculty of Science
University of Ontario Institute of Technology (Ontario Tech University)
Oshawa, Ontario, Canada
August, 2025

Thesis Examination Information

Submitted by: Hafiz Arslan Amjad

Master of Science in Computer Science

Thesis Title: An Attention-Enhanced Student–Teacher Framework for Structural and Logical Anomaly Detection in Industrial Settings

An oral defense of this thesis took place on August 14, 2025 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee:	Dr. Alvaro Quevedo
Research Supervisor:	Dr. Faisal Qureshi
Examining Committee Member:	Dr. Bill Kapralos
Thesis Examiner:	Dr. Patrick Hung

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

In this work we introduce AeCSAD, a student–teacher framework designed to detect two types of anomalies in images of industrial components: structural anomalies (physical defects) and logical anomalies (incorrect relationships between components). Unlike prior methods, AeCSAD extends the component segmentation–based logical anomaly detection scheme (c. 2024) with self-attention mechanisms, enabling more effective relational modeling. We demonstrate consistent improvements on the MVTec LOCO Anomaly Detection benchmark. Specifically, AeCSAD employs a global student network with self-attention for reasoning across distant components, complemented by a local student network for fine-grained analysis. Additionally, a patch histogram module measures the frequency distribution of components, allowing the system to detect irregularities in their occurrence. During inference, anomaly scores from the histogram module and the fused local–global networks are combined to produce the final anomaly score. Experiments show that AeCSAD achieves superior average AUROC performance on both structural and logical anomaly detection tasks compared to prior approaches.

Keywords: industrial visual inspection; logical anomaly detection ; self-attention mechanism; student-teacher architect;

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I authorize the Ontario Tech University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize the Ontario Tech University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Hafiz A. Amjad

Statement of Contributions

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

Dedication

dedicate this work to the guidance of my supervisor and the inspiration of my family.

Acknowledgements

Before all else, I am profoundly grateful to my God for choosing me and blessing me with health, skills, knowledge, time, resources, supportive people, guidance, and unwavering faith. I am deeply conscious that no effort bears fruit without His will and permission, and that every step taken toward completion was made possible only through His favor. Without any one of these elements, the fulfillment of this work would not have been within my reach.

With deep conviction, I acknowledge that the successful completion of my research would have been unattainable without the mindful and overwatching personality of my supervisor, Dr. Faisal Qureshi. His continuous and much-needed patience, days and nights of sharp experience, deep and insightful discussions, thoughtful and wise feedback, and wholehearted care have guided me through every stage of this journey. He has been more than a mentor, a fatherly figure whose advises helped me made every step meaningful in this work. Every insight I have gained and every gap I was able to bridge in my knowledge and understanding, is the result of his mentorship.

I am deeply thankful to my mother, whose foresight and wisdom taught me what truly matters in life and inspired me to pursue this research with purpose. I am equally grateful to my father, whose boundless support and quiet strength helped me endure the many subtle challenges I encountered along the way.

I thank my brothers and sisters, whose encouraging words in my efforts continuously lifted my spirits. Their support helped my mind to have quiet reassurance that comes from being surrounded by sincere hearts.

Lastly, I am grateful to my childhood friend, Mashhood Butt, for sharing the burdens of this research journey and offering calm and comforting words when they were most needed. By being through thick and thin, he is my friend indeed as he has always been with me whenever I am in need.

Contents

List of Tables

List of Figures

List of Acronyms

AD Anomaly Detection

AST Asymmetric Student Teacher

CNN Convolutional Neural Network

CounTR Counting Transformer

CSAD Component Segmentation Anomaly Detection

LGST Local-Global Student-Teacher

RAM++ Recognize Anything Model++

SAM Segment Anything Model

VAD Visual Anomaly Detection

YOLO You Only Look Once

Chapter 1

Introduction

The modern world is sustained by industries that drive economic growth, technological advancement, and societal development. Among these, manufacturing industries form a backbone, enabling the production of goods, infrastructure, and essential technologies [1]. As industrial systems become increasingly complex and interconnected, ensuring their efficiency has become a central concern [2]. Even when these industrial systems are designed with precision, real-world operations often face unpredictable influences [3]. Small shifts in material properties, mistakes in how components are put together, missing pieces, unexpected objects entering the system, or gradual damage over time can all disturb normal behavior. These unexpected changes, referred to as anomalies, can seriously affect the performance, safety, and quality of industrial processes. Given the consequences of undetected anomalies, the ability to identify them has become essential for maintaining operational integrity. This critical need has given rise to the field of Anomaly Detection (AD), which focuses on recognizing patterns of behavior that deviate from expected norms [4].

1.1 Industrial Settings for Anomaly Detection

In many industrial settings, the ability to detect anomalies through images is crucial not only for identifying visible defects, such as surface damage, but also for uncovering logical inconsistencies, such as misplaced objects [5]. In this thesis, we define industrial settings as manufacturing and assembly environments characterized by controlled imaging conditions. These environments typically include automated or semi-automated inspection systems that analyze images of object assemblies or parts with well-defined structural, spatial, or relational layouts. Defects in such settings are rare but diverse, and the systems are expected to maintain low false alarm rates while detecting both physical defects and semantic inconsistencies. Within this context, our work focuses on enhancing the performance of visual inspection systems in these settings through Visual Anomaly Detection (VAD) [6] which refers to analyzing images or videos to detect instances where the observed visual characteristics diverge from the learned normal cases [4]. Figure 1.1 presents a high-level overview of a VAD.

The process begins with the extraction of representative features from input and test image that capture the visual characteristics of the scene. During training, only normal samples are used to model the expected appearance and behavior of components. At test time, newly observed images are compared with the learned normal patterns using various anomaly scoring techniques such as L2 distance (measuring pixel-wise or feature-wise deviations), Cosine similarity (assessing angular differences in feature space), or reconstruction error (evaluating the discrepancy between the original image and its reconstructed version. The resulting anomaly map and score are then processed through decision heuristics to determine whether the observed deviation constitutes, e.g., physical damage or semantically inconsistent elements. As illustrated in the overview, the final stage of the detection process involves interpreting the nature of the anomaly, which requires a clear distinction between fundamentally different types of anomaly. Recognizing these anomalies is critical, as the nature of an anomaly often determines the most effec-

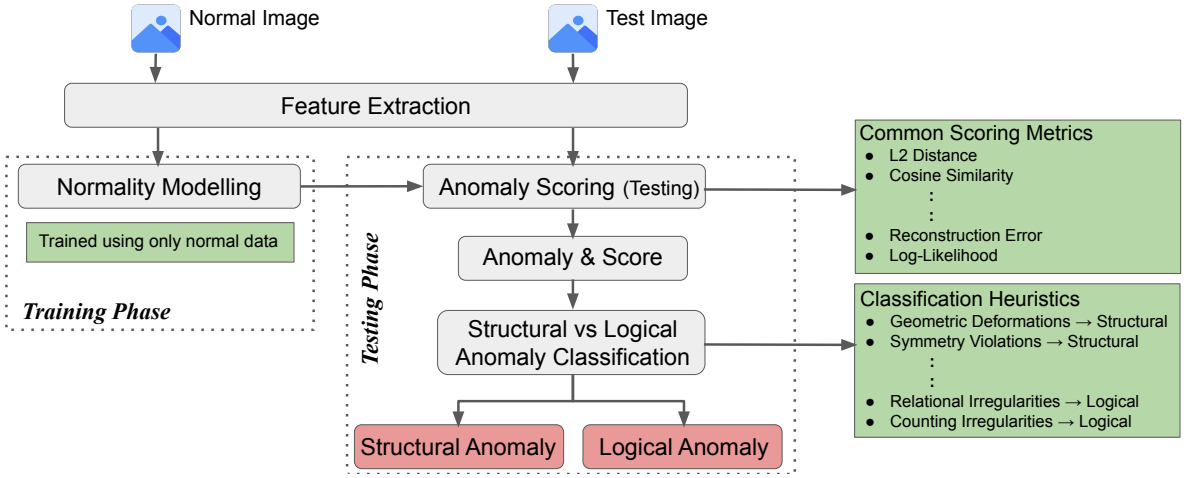


Figure 1.1: High-level overview of a visual anomaly detection pipeline. The system extracts features from input images and uses normal training data to learn typical patterns. At test time, anomaly scores and maps are produced and classified as either structural or logical anomalies using heuristic-based decision logic.

tive strategies for its detection. Consequently, two important subcategories of anomalies arise, namely structural anomalies and logical anomalies.

1.2 Structural and Logical Anomalies

Structural anomalies are linked to geometrical properties. The geometrical properties describe the shapes, proportions, and structural patterns that objects maintain under normal conditions. This type of anomaly typically manifest as visible distortions, irregular shapes, abnormal textures, or misalignments within an object's structure [6]. An example of structural anomaly is shown in Figure 1.2. The detection of structural anomalies focuses on identifying these disruptions by modeling and analyzing the normal geometric consistencies found in non-defective samples [7].

On the other hand, logical anomalies arise when the relationships, arrangements, or sequences among multiple visual elements violate the expected logical pattern. In this context, logical refers to the correctness of how multiple objects or components of objects are connected, combined, or ordered according to predefined rules [5]. Unlike structural

Figure 1.2: Examples from the MVTec LOCO AD dataset [5] showing normal, structural, and logical states in two categories: pushpins and breakfast box. The top row illustrates a correct pushpin layout, a structurally damaged pushpin, and a logical error with two pushpins in one slot. The bottom row shows a complete breakfast box, a damaged mandarin, and a missing granola section despite other components being present.

anomalies, which typically affect the physical form of individual components, logical anomalies concern inconsistencies between visual objects that may appear visually correct in isolation but are incorrect when considered together. A small subset of examples of logical anomaly is shown in Figure 1.2. Detection of logical anomalies therefore focuses on identifying such inconsistencies in rules and contextual meanings within visual data by flagging instances where these relationships are disrupted even when individual parts may not exhibit obvious physical defects [7].

1.3 Problem

The early development of industrial anomaly detection methods focused primarily on the identification of structural anomalies [8, 9] using Convolutional Neural Network (CNN) [10] based architectures. CNNs are a class of deep learning models designed to extract local spatial patterns from images using learnable convolutional filters. CNNs are well suited for detecting structural anomalies since CNN constructs receptive field

over images that spans neighboring pixels, enabling them to capture local spatial patterns in an image. A receptive field refers to the specific region of an image that a CNN "sees" during processing and enables it to recognize localized structures such as edges, corners, textures, and simple part arrangements. As long as the essential information required to recognize the inconsistencies is isolated, non-relational, and contained within these local neighborhoods, CNN based architectures are effective [11]. However, logical inconsistencies are relational and often exist in distant but connected regions of an image causing CNN based architectures to fail to detect logical anomalies.

1.3.1 CNN Architectures as Structural Anomaly Detectors

Visual data inherently contains spatially structured information, where the arrangement, relative positions, and textures convey essential details about object conditions and patterns [12] and effective anomaly detection therefore demands models capable of capturing and interpreting these patterns [7]. For this purpose, Convolutional Neural Networks (CNNs) architectures offered the ability to learn hierarchical spatial representations from visual data [12] since their ability to model spatial information is suited for recognizing structural variations [13]. The systematic benchmarking and comparison of methods for the detection of industrial structural anomalies is performed using a widely used MVTec Anomaly Detection (MVTec AD) [6] [14, 15, 16], offering a diverse collection of industrial objects and defect types. The approaches developed using the architectural characteristics of CNNs reported state-of-the-art performance, frequently achieving scores exceeding 99% [14, 17, 8]. The remarkable performance in structural anomaly detection can be largely attributed to the alignment between features of structural anomalies and the inductive biases inherent in CNN-based architectures [18, 19]. The inductive bias in CNNs stems due to their receptive fields, which focus on neighboring pixel regions. This characteristic of CNNs helps to develop localized structural understanding of objects in images. However, when accurate interpretation depends on relationships

between spatially distant components, such as determining whether two separate objects or components of objects are correctly assembled to reflect logical consistency, standard CNN architectures encounter fundamental limitations. The inductive bias of the models caused by the receptive field prevented them from effectively capturing long-range dependencies [12] which becomes a critical weakness when addressing logical anomaly detection tasks [5, 20].

1.4 Initial Efforts in Logical Anomaly Detection

We explored several preliminary approaches to address the challenge of detecting both structural and logical anomalies in industrial settings. These efforts focused on three main directions: generative models (diffusion-based), object detection methods, and transformer-based relational reasoning. Each strategy offered certain advantages, but also revealed critical limitations that made them unsuitable for unified, unsupervised anomaly detection.

1.4.1 Diffusion-Based Generative Models

DiffusionAD [21], a norm-guided one-step denoising framework, demonstrated strong performance in reconstructing images corrupted with structural defects. Structural anomalies often manifest as localized pixel-level deviations, which the model learns to correct during reconstruction. Anomalies can then be detected by comparing the reconstructed image with the input. However, this method was less effective for logical anomalies, which involve violations of expected object configurations or relationships. For instance, a missing granola bar in an otherwise complete breakfast box may not produce a noticeable difference in pixel values, leading to near-identical reconstructions and a failure to detect the anomaly. Addressing such cases would require the introduction of an additional diffusion-based module specifically trained to focus on identifying logical inconsistencies

by learning typical object arrangements or relational patterns. However, even such a module would still lack the ability to explicitly count objects or verify set-level relationships (e.g., one-to-one correspondence between slots and items). Capturing this level of reasoning would necessitate yet another module dedicated to object counting or spatial consistency enforcement. This layering of specialized modules would substantially increase architectural complexity. Combined with the high computational demands of diffusion-based models, these limitations render the approach unsuitable for scalable and practical anomaly detection in industrial settings.

1.4.2 Object Detection based Models

Object detection models such as You Only Look Once (YOLO) [22] are effective for identifying and localizing spatially distinct items, offering a potential route for detecting logical anomalies that involve object duplication or omission. However, their utility diminishes at the component level, where objects may be densely packed, visually similar, or lack clear boundaries. Additionally, logical anomalies frequently arise not from object presence alone, but from violations of spatial arrangements or co-occurrence rules. For example, two pushpins occupying the same slot, or components of a meal box arranged incorrectly. Such relational inconsistencies fall outside the representational scope of typical object detectors. Furthermore, YOLO and similar models require supervised training with bounding box annotations for each object class. This transforms the problem from unsupervised anomaly detection to supervised localization, imposing a significant labeling burden and reducing adaptability in scenarios where the types of anomalies are not known in advance.

1.4.3 Transformer-Based Relational Models

To address the need for global reasoning over object relationships, transformer-based models like CounTR [23] were considered. Originally designed for crowd counting,

Counting Transformer (CounTR) employs self-attention mechanisms to model spatial dependencies and global context, making it a potential candidate for relational anomaly detection. In theory, such models could learn typical configurations and flag violations involving missing, repeated, or misaligned components.

However, transformers generally require extensive training data to learn reliable attention patterns and to generalize effectively. This requirement presents a major challenge in industrial anomaly detection settings, where datasets are limited and logical inconsistencies are often highly varied and unlabeled. Even with architectural flexibility, applying transformer-based counting models would necessitate auxiliary modules for semantic verification or rule-based reasoning, adding further complexity to the pipeline since counting alone does not guarantee semantic correctness.

1.4.4 Component Segmentation Anomaly Detection

A recent unsupervised Component Segmentation Anomaly Detection (CSAD) [24] framework, has combined segmentation with Recognize Anything Model++ (RAM++) [25], GroundingDINO [26] and Segment Anything Model (SAM) [27] to address both structural and logical anomaly detection. By Using RAM++, GroundingDINO and SAM to generate pseudo-labels, CSAD constructs component-aware segmentations without supervision, enabling explicit reasoning over component counts using the patch histogram module. These modules directly capture logical inconsistencies arising from incorrect quantities. In parallel, the global student network in CSAD's Local-Global Student-Teacher (LGST) module employs an CNN based autoencoder [28] that implicitly captures image-wide dependencies. This integration of explicit (histogram-based) and implicit (CNN based autoencoder) represents a meaningful step in logical anomaly detection. However, a critical limitation persists. CSAD has not taken explicit advantage of self-attention architectures [29, 30], which are inherently well suited to model arbitrary global relationships between components of objects residing in different regions of an image.

1.5 Attention Enhanced CSAD

Motivated by the limited receptive field of CNN based autoencoder that is represented as global student network in CSAD, we implemented self-attention in global student network and propose Attention Enhanced CSAD (AeCSAD). AeCSAD explicitly models the relational structure among segmented components, enabling reason over object or component level contextual dependencies. Unlike convolutional operations, which are inherently local and translation-invariant, self-attention mechanisms treat every position as a potential source of information, allowing for feature interactions that reflect relationships rather than just local appearance. Moreover, self-attention aligns well with the notion of global context modeling, which lies at the heart of CSAD's student-teacher paradigm. Since the teacher network in CSAD defines the normal features, the student must learn not just to replicate features, but to understand how features relate to one another across space. The addition of self-attention enhances this relational sensitivity, enabling the student to fail more meaningfully in the presence of global anomalies. This self-attention based extension thus forms the principal architectural contribution of this thesis. Figure 1.3 presents a high-level overview of AeCSAD.

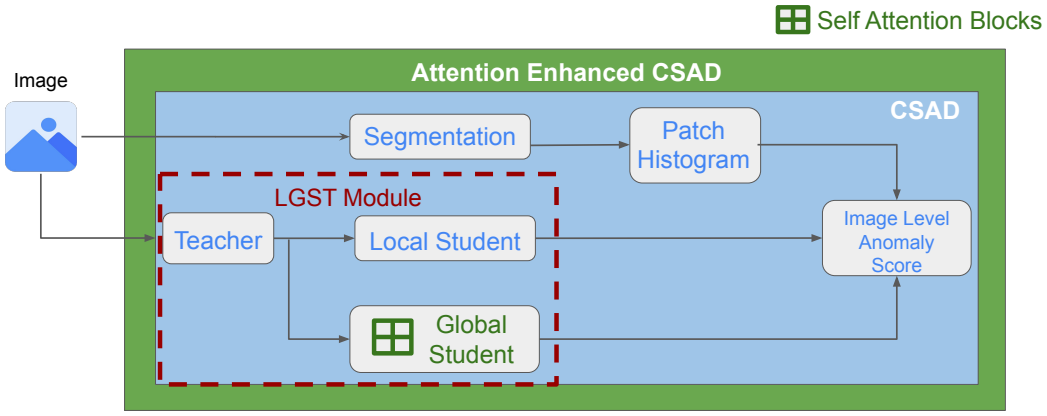


Figure 1.3: High-level overview of the proposed Attention Enhanced CSAD (AeCSAD) framework extending CSAD (in blue). A normal image is processed by a teacher network and compared with local and global student, where the global student incorporates self-attention (green) to capture long-range dependencies. Semantic segmentation and patch-wise histograms are used to compute the image-level anomaly score.

1.6 Research Objectives

Across our intially explored effort, a consistent limitation emerged where structural anomalies could be captured using local visual cues, while logical anomalies required reasoning about object arrangements, roles, and relational consistency. The methods either incurred high computational costs, relied heavily on supervised data, required large amount of training data, or lacked explicit semantic modeling capabilities. Observing these limitations, we aim to achieve the following objectives in our work.

1. **To develop a unified, lightweight anomaly detection framework** capable of effectively handling both structural and logical anomalies in an unsupervised setting.

2. **To design a modular architecture** where local and relational reasoning components operate independently.

3. **To ensure data efficiency** by building a model that performs well with limited training samples.

4. **To bridge the gap between pixel-level and semantic-level understanding**, integrating both local detail sensitivity and global context modeling within a single coherent framework.

1.7 Contributions

This work proposes Attention Enhanced CSAD (AeCSAD), an anomaly detection framework that integrates self-attention into the global student network of CSAD. Self-attention offers a mechanism for modeling global interactions between distant parts of an image, allowing explicit logical reasoning. The primary contributions of this work are as follows.

- We introduce AeCSAD, a self-attention enhanced anomaly detection framework capable of identifying both logical and structural anomalies in a semi-supervised setting.

- We restructure the global student student of CSAD [24] by incorporating self-attention blocks, enabling explicit relational reasoning across distant regions of an image.

- We provide empirical evidence showing that AeCSAD outperforms state-of-the-art baselines across logical and structural anomalies on the MVTec LOCO AD [5] benchmark.

Chapter 2

Literature Review

This chapter surveys the evolution of visual anomaly detection methods, beginning with classical statistical techniques and progressing toward their adaptation into modern convolutional neural network (CNN) based frameworks. We then examine the transition toward logic-aware approaches, driven by structural modifications and redefinition of feature representations. The chapter concludes by presenting recent advances in dual-branch student–teacher architectures, which serve as integrated solutions to address both structural and logical anomalies. Through this review, we position our approach not merely as a continuation of prior methods, but as a synthesis of statistical principles, architectural innovations, and semantic-level reasoning.

With the review of the landscape of visual anomaly detection in the context of industrial manufacturing. Our objective is to trace how foundational methodologies, from classical statistical modeling to modern deep learning architectures, have shaped the field's practices and informed the development of systems capable of detecting both structural and logical anomalies. By analyzing model architectures, assumptions about feature representations and normality modeling, and domain-specific adaptations in prior work, we aim to clarify the methodological boundaries and practical trade-offs among major categories of visual anomaly detection methods.

2.0.1 Overview

As an overview of our literature, we begin discussing anomaly detection using statistical frameworks that attempted to model the distribution of normal data in either raw pixel space or a low-dimensional and transformed space called embedding. These methods, ranging from principal component analysis [31] and Gaussian mixture models [32] to Mahalanobis distance [33] and clustering [34], established a principle that is still a core to anomaly detection today, namely that if one can accurately represent what is normal then deviations from that representation can signal anomalies.

We then move our discussion to a pivotal shift occurred with the advent of deep Convolutional Neural Networks (CNNs), which enabled automatic extraction of hierarchical visual features from raw input images. CNN-based methods improved anomaly localization under variations in illumination, texture, and background clutter. These approaches often maintain statistical roots by applying k-Nearest Neighbors [35] or Mahalanobis distance [36, 37] over CNN feature maps. Others employ student–teacher consistency frameworks, in which a learnable student model is trained to replicate a frozen teacher's representations on normal data, with deviations interpreted as anomalies.

We will also see early methods that were primarily designed for structural anomaly detection. We will learn that, while such methods have demonstrated strong performance for detecting structural anomalies, they fall short in handling logical anomalies. Instead, they require reasoning over part relationships and context[5]. As a result, there has been a growing research focus on models capable of integrating global dependencies, compositional semantics, and relational structure.

We will shift our discussion towards an end, where we will learn about recent works have extended structural anomalies focused frameworks to also focus on logical anomalies. One direction modifies the student–teacher paradigm, introducing bottlenecks, memory modules, or multi-branch architectures to learn local and global anomaly signals jointly [38]. The other leverages segmentation-based logic modeling, employing unsupervised

clustering or few-shot part segmentation to reason over object composition [39]. Such methods often incorporate memory banks or statistical priors to track part distributions and spatial relationships, enabling more principled reasoning about object logic.

2.1 Statistical Approaches

At the fundamental level, statistical methods aim to characterize the behavior of data using probability theory. In the context of anomaly detection, this typically involves estimating the probability distribution of "normal" data and assessing whether new samples deviate from this distribution. Applied to visual domains, these methods must operate in high-dimensional spaces where raw images consist of numerous pixel values, and the underlying patterns of normality can be highly complex. To address this, statistical approaches rely on feature extraction to transform raw visual data into compact, informative representations that emphasize structural regularities. These features may range from simple pixel intensities to more abstract embeddings derived from dimensionality reduction techniques[40, 41, 42].

2.1.1 Statistical Assumptions for Anomaly Detection

At their core, statistical approaches are typically based on the following assumptions:

1. **Normality is learnable from the data**: That is, if provided enough examples of normal behavior, a statistical model can learn the parameters or structure of this behavior.

2. **Anomalies deviate significantly**: Anomalous samples lie far from the learned model in terms of distance, density, or reconstruction error.

The assumptions then guide the statistical anomaly detection models to formulate normality in terms of data distributions, distances, variances, and correlations. The methods

therefore, then model normality by representing the high-dimensional visual input in a lower-dimensional manifold that maintains important correlations. After that, these compressed representations form the basis for detecting and measure deviations. The practical implementations of these assumptions vary widely. Some models define boundaries in feature space (e.g., One-Class SVM [43]), others estimate density (e.g., Gaussian Mixture Models [32]), and still others measure how well new samples can be reconstructed from normal data (e.g., PCA [31]). These statistical models offer different levels of utility for these anomaly types:

- **Structural Anomalies**: Statistical methods like PCA can be highly effective here because the anomalies manifest as measurable deviations in brightness, edge density, spatial structure, or texture regularity. In particular, metrics like Mahalanobis distance or control charts can detect when these visual patterns deviate from expected norms.

- **Logical Anomalies**: Traditional methods can still be used here if the features capture semantic information, such as object co-occurrence, shape consistency, or spatial arrangement. For example, applying a One-Class SVM to high-level embeddings from a pretrained CNN can detect if a certain composition is highly unlikely, even if each individual object appears normal.

2.1.2 Limitations of Statistical Approaches

Classical statistical approaches for anomaly detection faced notable limitations that constrained their scalability in real-world visual inspection tasks. The rigid nature of handcrafted feature definitions, the need for manual tuning of parameters, and the dependence on domain-specific knowledge reduced their adaptability across varying visual domains. Moreover, these models were particularly sensitive to common environmental variations in industrial imaging such as lighting, noise, and reflection.This need catalyzed a method-

ological shift toward deep learning-based approaches, particularly those grounded in Convolutional Neural Networks (CNNs)[44, 45].

2.1.3 Summary of Statistical Based Approaches

While structural anomalies can often be addressed through statistical analysis of raw or low-level features, logical anomalies require feature representations that capture semantic and relational dependencies capabilities that are often enabled by hybrid models. Moreover, statistical methods provide critical safeguards in deployment scenarios; when deep models exhibit uncertainty or encounter out-of-distribution inputs, statistical detectors serve as lightweight yet reliable anomaly indicators[46, 47, 48].

2.2 CNN-Based Statistical Modeling and Limitations

Convolutional Neural Networks (CNNs) [49] are architectures that can learn hierarchical visual features from raw image data, thereby replacing hand-crafted representations with adaptable feature hierarchies. These models retained many statistical intuitions such as modeling distributions, quantifying deviations, or scoring reconstruction errors but embedded them within a learned feature space. A common design paradigm in CNN based anomaly detection is the *unsupervised* [50] setting, in which models are trained exclusively on normal samples, and anomalies are identified at inference time based on deviation from learned representations. Within this framework, several methods extract convolutional features from pretrained backbones and model their distributions using multivariate Gaussians. Anomaly scores are then computed using the Mahalanobis distance [51]. This strategy assumes that, under normal conditions, the distribution of deep features is approximately Gaussian per location, and that significant deviations from this distribution indicate anomalous content[6, 7, 4]. The upcoming subsections would review methods such as *SPADE* [52], *PaDiM* [53], and *RegAD* [54] embodying this approach.

2.2.1 SPADE

Semantic Pyramid Anomaly Detection (SPADE) [52] presents unsupervised visual anomaly detection by reframing the task as a problem of correspondence and alignment failure. Rather than modeling statistical distributions directly or learning reconstruction mappings, SPADE considers pixel level detail to detect anomalies by measuring how well a test image can be matched to a set of normal training images. The architecture of SPADE is entirely inference-based and relies on pretrained deep convolutional features, extracted from a WideResNet-50 model trained on ImageNet dataset[55, 56] where SPADE utilizes frozen features from different layers of WideResNet-50, capturing a hierarchy of information. These features are assembled into a *pyramid* representation, which preserves both high-resolution detail (via early layers) and abstract semantic information (via deeper layers).

Once the pyramid features are extracted, SPADE performs two stages of anomaly scoring. First, it computes an image-level anomaly score by measuring the distance between the global representation of the test image and its k-nearest neighbors (kNN) [57] from the training set to filter out globally consistent images. For localization, SPADE compares each patch (or pixel-wise feature vector) in the test image to all patches in the top-K nearest training images. It calculates the L2 distance [58] between a patch and its closest match, then averages the distances across the pyramid to obtain a dense anomaly map. Patches with high average distances are flagged as anomalous.

Statistically, SPADE embodies core ideas that bridge classical and deep anomaly detection. It assumes that patches from normal images lie in dense regions of feature space, while anomalies fall in sparse or isolated areas, similar to density estimation models. However, it does not explicitly model the distribution of normal data. Instead, it implicitly estimates the local feature density through kNN retrieval, which avoids imposing strong parametric assumptions (such as Gaussianity). This design makes SPADE robust to multimodal data and adaptable to various appearance patterns without retraining.

SPADE only requires a gallery of normal images and a pretrained CNN backbone. This simplicity makes SPADE efficient to deploy. Additionally, its use of multi-resolution features allows it to detect subtle deviations across multiple spatial scales. However, SPADE also presents certain limitations. Since it relies entirely on appearance similarity and spatial alignment, it is primarily sensitive to anomalies that affect visual consistency such as defects in texture, color, or structure [52] and it may underperform in detecting logical anomalies[5, 59]. Moreover, the computational cost of comparing every test patch to a large memory bank during inference can become substantial for large and high-resolution datasets.

2.2.2 PaDiM

Patch Distribution Modeling (PaDIM) [16] also introduces a statistical formulation but grounded in parametric modeling of local convolutional features. At its core, PaDiM constructs per-patch statistical distribution by extracting hierarchical feature embeddings from a pretrained ResNet [60] backbone. The method selects features from early, mid, and last convolutional layers of ResNet and concatenates them to form spatial descriptors for each patch. These descriptors encapsulate both low-level texture and high-level semantic cues. This multi-layer fusion mirrors the representational goals of feature pyramids, but PaDiM diverges from SPADE in its downstream use. Rather than retrieving neighbors, PaDiM learns the distributional properties of normal patches.

In training, PaDiM estimates a separate multivariate Gaussian distribution for each spatial location (i.e., patch position) across all training samples. Specifically, for every fixed spatial index (i, j) in the feature map, the method computes a mean vector μ_{ij} and covariance matrix Σ_{ij}, encapsulating the variability of feature embeddings at that location under normal conditions. These parameters are stored and reused during inference. At test time, anomaly scoring is performed by computing the Mahalanobis distance [33] between the test patch's embedding at location (i, j) and a correspond-

ing Gaussian model. Notably, this scoring mechanism eliminates the need to store or search over training embeddings during inference, yielding significant gains in memory and speed.

PaDiM models anomalies as statistical outliers by learning the probability density of each patch location in feature space, forming a probabilistic profile of normal visual patterns. Unlike retrieval-based approaches that approximate density via sample proximity, PaDiM's generative modeling allows for analytic scoring which is efficient when the feature distribution is unimodal and well-separated. Moreover, the Mahalanobis distance naturally incorporates both the mean and covariance structure, adapting sensitivity based on learned variability i.e patches with stable appearance are penalized more heavily for deviations, while those with high intra-class variance tolerate greater fluctuations.

However, PaDiM's assumptions come with trade-offs. Its modeling assumes per-location Gaussianity, which may be limiting in settings where feature distributions are multimodal or spatially entangled. Additionally, its scoring remains appearance-centric i.e it detects whether something "looks unusual" at a given location, but not whether the overall composition or object logic is violated.

2.2.3 RegAD

Registration-Based Few-Shot Anomaly Detection (RegAD) [54] introduces a registration-based formulation for few-shot anomaly detection(FSAD) [61], designed to operate across novel categories without retraining. The core of RegAD lies in the use of a Siamese network [62] that learns to align features from image pairs of the same class. Each branch of the Siamese network encodes the input image using a convolutional backbone without global pooling, preserving fine-grained spatial resolution. To bring the feature maps of the two inputs into correspondence, RegAD embeds Spatial Transformer Networks (STNs [63]) at multiple stages of the encoder. These modules learn geometric transformations that maximize local similarity in the resulting feature space, effectively

"registering" the query image to a support exemplar.

During training, the model is presented with only normal image pairs from a variety of object categories. The registration process is optimized to align spatial features via cosine similarity, allowing the model to learn category-agnostic structural correspondences. In doing so, RegAD internalizes a general notion of spatial regularity that allows it to later compare new test images to a small set of normal exemplars drawn from previously unseen classes.

Once registration is performed, the anomaly detection phase begins. For a given test image and its registered support set, RegAD computes per-location feature distributions using the registered feature maps of the normal samples. At each spatial index, a multivariate Gaussian distribution is estimated over the registered support features. The Mahalanobis distance [33] is then used to score how far each location in the test image deviates from these local distributions. This produces a dense anomaly map that reflects deviations from expected registered structure.

The distinction of RegAD from earlier methods such as PaDiM and SPADE is its ability to generalize across categories. While SPADE uses retrieval from a fixed training set and PaDiM learns distributions fixed to object categories, RegAD dynamically builds feature distributions conditioned on a few support images. This few-shot paradigm removes the need for a separate model per category and enables practical deployment in industrial settings where labeled data is scarce.

Statistically, RegAD remains grounded in classical anomaly detection principles. Its use of Mahalanobis distance and per-patch Gaussian modeling matches prior approaches. RegAD excels in detecting spatial anomalies across diverse object types, especially when appearance deviations are subtle but misaligned. However, it does not explicitly model the logical composition of scenes. This is a consequence of RegAD's design philosophy where it enforces spatial alignment, but not semantic reasoning. A comparative summary of key architecutural elements of methods discussed in this section is given in Table 2.1.

Table 2.1: Comparative summary of SPADE, PaDiM, and RegAD across core architectural components and anomaly detection performance dimensions.

Method	**Feature Modeling**	**Scoring Mechanism**	**Spatial & Logical Sensitivity**	**Generalization Ability**
SPADE [52]	Uses pretrained CNN features aggregated into multi-scale pyramids. Anomaly scoring relies on direct deep feature similarity.	Employs pixel-wise kNN distance over nearest-neighbor features. Scoring is non-parametric and appearance-driven.	Captures spatial layout well via pyramid fusion but lacks semantic awareness. Sensitive to visual anomalies but not logical inconsistencies.	Limited. Depends on fixed training data and lacks adaptability to novel categories.
PaDiM [16]	Models patch-level multivariate Gaussians from intermediate ResNet features to represent local normality.	Computes Mahalanobis distance from learned patch-wise feature distributions. Scoring is fully parametric.	Maintains spatial resolution using per-location statistics. Assumes unimodal appearance and ignores inter-object semantics.	Low generalization. Designed for closed-set evaluation with no support for unseen categories or few-shot adaptation.
RegAD [54]	Learns support-conditioned representations using Siamese encoders and spatial transformer networks (STNs).	Uses Mahalanobis scoring in aligned feature space dynamically estimated from support data.	Preserves structural alignment and detects spatial misconfigurations. Lacks semantic logic modeling but excels in geometric consistency.	High. Adapts effectively across categories with minimal supervision using a few-shot paradigm.

2.3 Non-Parametric kNN-Based Feature Models

Another prominent family of industrial anomaly detection methods leverages deep features from pretrained networks, applying non-parametric anomaly scoring via k-nearest neighbor (kNN) algorithms. These approaches rely on distance based comparisons in feature space representing a low-cost strategy for anomaly detection. Recent advances have significantly diversified and enhanced this framework across four critical dimensions: spatial awareness, feature refinement, training-data assumptions, and computational scalability.

2.3.1 PatchCore

PatchCore [14] exemplifies a non-parametric, training-free approach to anomaly detection built entirely on nearest-neighbor search over deep features extracted from pretrained networks. Rather than fitting parametric models or learning reconstruction mappings, PatchCore relies on the idea that anomalies can be detected by measuring distance to previously observed normal examples. Specifically, it implements a patch-level kNN retrieval strategy in deep feature space, optimized for both memory and computational efficiency.

PatchCore begins by extracting intermediate convolutional feature maps from a pretrained WideResNet50 [55]. These feature maps are flattened into a set of patch-wise embeddings, each corresponding to a localized spatial region of the input image that allows the method to localize small or spatially concentrated anomalies. To reduce the size of the reference memory bank without sacrificing coverage of the normal feature space, PatchCore applies coreset subsampling via greedy k-center clustering. This ensures that the retained features span the diversity of normal patches while enabling fast retrieval. Anomaly detection is performed at inference time by comparing each patch in the test image to its closest match in the memory bank using L2 distance. These patch-level

scores are then aggregated to form image-level anomaly scores and spatial heatmaps.

PatchCore's kNN-driven logic is built around the assumption that normal patches form dense clusters in feature space, while anomalous patches will lie at greater distances. However, its focus is not on explicitly modeling distributions or densities; rather, it uses kNN purely as a distance-based decision rule. While PatchCore is effective at detecting surface-level defects and irregular textures since its scoring is grounded in local visual similarity rather than semantic context or logical relationships between parts. Moreover, although the coreset improves inference speed, it introduces a sensitivity to the coverage quality of the retained memory causing trade-off between memory size and detection accuracy. Nevertheless, PatchCore demonstrates how pretrained CNN features, when paired with simple non-parametric retrieval mechanisms, can yield robust performance without the need for additional learning stages.

2.3.2 Position-aware Neighborhood Information

Position-aware Neighborhood Information(PNI) [64] introduces an enhancement to the non-parametric, kNN-based anomaly detection paradigm by embedding spatial priors into the inference process. Building upon the core logic of methods like PatchCore, PNI extends this framework by recognizing that the likelihood of an anomaly is not determined solely by visual similarity, but also by where and how features appear within the spatial structure of an image. The model begins by extracting mid-level patch embeddings from a pretrained convolutional backbone ResNet18 [60], with each embedding corresponding to a spatial location in the image. These features are stored in a memory bank of normal data, constructed in the same way as PatchCore. During inference, each test-time patch is compared to the memory bank, and its nearest neighbor is retrieved based on L2 distance[58]. However, unlike standard kNN scoring, PNI does not treat all deviations as equal. To refine this scoring mechanism, PNI introduces two contextual priors:

(1) Positional Priors: The likelihood of certain visual features varies across spatial

positions. PNI learns a prior map over the training data that models the typical spatial positions of patch types. At test time, the anomaly score is adjusted based on how expected a patch's location is relative to this learned prior.

(2) Neighborhood Consistency Priors: Local regions in normal images tend to exhibit spatially coherent features. To encode this, PNI measures the similarity of a patch not just to its own nearest neighbor, but in the context of its surrounding neighbors. If a patch's anomaly score diverges significantly from its neighborhood, it may be suppressed or amplified accordingly. This local smoothing enhances structural consistency in the anomaly heatmaps and reduces false positives from isolated outliers.

Through its priors, PNI introduces rudimentary awareness of capturing structure of object structure—capturing. However, the model still operates within a fundamentally appearance-driven framework. It improves spatial sensitivity, but does not reason over logical composition. PNI thus offers a bridge between basic retrieval-based scoring and richer spatial understanding, but stops short of true logic-aware anomaly detection. Nonetheless, PNI's contribution shows that incorporation of reasoning about context into scoring using heuristics can elevate the expressiveness of kNN detection without introducing full model retraining. These ideas extend the design space of non-parametric methods toward structured understanding, and lay conceptual groundwork for hybrid approaches that combine local matching with global reasoning.

2.3.3 Feature-Space Refinement in kNN-Based Method

ReConPatch [65] addresses a key limitation in non-parametric kNN-based anomaly detection, which is the reliance on raw pretrained features from networks trained for object classification (e.g., ImageNet). While these features are effective at capturing general visual patterns, they are not optimized for detecting subtle structural deviations or contextual inconsistencies commonly encountered in industrial anomaly detection. ReConPatch proposes a solution that neither discards the efficiency of the kNN pipeline nor retrains

the backbone. Instead, it introduces a lightweight contrastive learning [66] module to reshape the feature space, making it more sensitive to anomaly-relevant distinctions. The architecture follows the conventional kNN-based detection framework adopted by PNI and PatchCore.

However, ReConPatch introduces a critical modification in the form of a projection head trained using a patch-wise contrastive loss [67]. This head is shallow (typically one or two fully connected layers) and is trained to cluster visually and semantically similar patches, extracted often from nearby spatial regions or data augmentations and repelling dissimilar patches. The training is performed only on normal data and is self-supervised [68], requiring no anomaly labels. Crucially, it does not involve fine-tuning the backbone, preserving the lightweight, inference-focused nature of the overall system. This contrastive objective reshapes the geometry of the embedding space. It ensures that feature similarity reflects more meaningful local and contextual relationships, allowing the kNN scoring mechanism to better distinguish between subtle variations of normal and abnormal patterns.

As a result, ReConPatch significantly improves anomaly localization accuracy, especially in scenarios where raw pretrained features produce misleading neighbor matches due to texture or shape confusion. Yet, ReConPatch, like its predecessors, operates within the visual similarity paradigm. Its scoring also remains grounded in appearance-based deviation and does not explicitly model object composition, spatial configuration, or logical structure. Nevertheless, ReConPatch success reinforces the idea that representational quality and not just architecture plays a critical role in anomaly detection performance, even in training-free or few-shot regimes.

2.3.4 Unsupervised kNN Filtering of Channels

Inter-Realization Channels (InReaCh) [69] is a fully unsupervised anomaly detection framework that adapts the kNN-based detection paradigm to operate effectively even in

the presence of substantial noise or contamination in the training set. Unlike previous kNN methods that assume anomaly-free training data, InReaCh offers a mechanism to filter and retain only highly confident nominal features before constructing the memory bank for test-time comparison. By focusing on robust feature realizations across multiple augmentations, InReaCh enhances reliability and reduces the impact of corrupted training data.

Each image is treated as a separate realization of a shared underlying data distribution. From this perspective, nominal patches should associate reliably across multiple images. To operationalize this, InReaCh introduces inter-realization channels that consists of sequences of spatially aligned and mutually nearest patches, constructed by comparing seed patches from a subset of images against patches from the remaining dataset. Only those channels that show both high span (i.e., presence across many images) and low spread (i.e., internal similarity) are retained as representations of nominal behavior. The final nominal memory bank model is built from these filtered channels. At test time, anomaly scores are assigned by computing the L2 distance [58] between each test patch and its nearest neighbor in this memory. Crucially, this step remains non-parametric and does not involve any training or parameter fitting.

InReaCh is notably resilient to noisy training data, using a span-and-spread trimming strategy to exclude anomalies even when up to 40% of the training set is corrupted. This design choice alleviates a major bottleneck of earlier approaches like PatchCore, which may inadvertently retain outlier patches during coreset subsampling. In contrast, InReaCh sacrifices some recall for high precision in modeling nominality, ensuring that the anomaly scoring remains grounded in consistently repeating, high-confidence visual patterns. However, InReaCh inherits the same limitation as PatchCore and PNI in terms of detecting logical inconsistencies as its deign focus was on resilience to noise and not relational and compositional design. A summarized comapartive analysis of this section is also given in Table 2.2.

Table 2.2: Comparative Structural and Logical Analysis of kNN-Based CNN Anomaly Detectors

Method	**Context Modeling**	**Feature Refinement**	**Memory Strategy**	**Structural Awareness**	**Logical Awareness**
PatchCore [14]	Operates independently per patch; lacks spatial or relational modeling	Uses raw CNN embeddings without additional refinement	Greedy coreset to reduce memory; no contextual indexing	Limited to local patch anomalies; no inter-patch structure captured	No awareness of object roles, semantics, or contextual coherence
PNI [64]	Models spatial and neighborhood dependencies for context-aware scoring	Uses raw features; no contrastive or supervised refinement	Retains full feature memory; uses spatial priors for scoring	Encodes local structure through spatial proximity	Does not explicitly consider object semantics or relational roles
ReConPatch [65]	Incorporates global similarity maps; captures contextual resemblance	Contrastive embedding refines feature space toward anomaly separation	Projects features to low dimension for efficient memory	Captures similarity patterns; not geometric or part-based structure	Lacks object-level reasoning or scene-consistency constraints
InReaCh [69]	Learns inter-image structural cues via cross-sample relational matching	No feature learning beyond pretrained encoder	Retains full feature bank; no compression	Aware of relational structure across images; supports topology-based cues	Partially sensitive to logical layout, but lacks semantic object understanding

2.4 Student-Teacher Methods in Anomaly Detection

Student-teacher (ST) paradigm [70] introduce a predictive learning setup, where a student network is trained to imitate a fixed teacher's feature representation on anomaly-free data. The underlying assumption is intuitive where the student encounters anomalous content at test time, it will fail to replicate the teacher's features, leading to detectable divergence. This setup blends transfer learning [71], self-supervised [72] regression, and out-of-distribution generalization into a unified framework which we will discuss in section.

2.4.1 Uninformed Students

The Uninformed Students [70] model formalizes a foundational hypothesis that when a student model is trained exclusively on normal data to replicate the output of a fixed teacher network, its response to anomalous inputs will exhibit measurable deviations. These deviations, quantified as reconstruction errors in the feature space, form the basis for anomaly scoring. The method employs a pretrained teacher network, typically a ResNet [60] or WideResNet [55], to extract multiscale feature representations from the input image. The teacher is not fine-tuned on the anomaly detection dataset and instead provides frozen, general-purpose semantic embeddings. One or more student networks are then trained in a regression setting [73] to reproduce the teacher's intermediate feature maps. Training is performed on anomaly-free data using a mean squared error (MSE) loss [74] function, with the objective of minimizing the discrepancy between the student and teacher features at corresponding spatial locations.

A distinguishing feature of this framework is the use of a student ensemble to estimate predictive variance [75]. Rather than relying on a single deterministic prediction, the approach trains multiple students independently and computes both the mean feature reconstruction error and the variance across student outputs. During inference, the

combined error and variance are aggregated into a dense anomaly map, highlighting spatial regions likely to contain defects[76, 77, 75, 78]. The Uninformed Students model was designed to balance semantic abstraction and localization fidelity. The feature maps are are incorporated from different depths of the teacher network, capturing both fine-grained texture anomalies and broader structural inconsistencies. This multi-scale feature regression enhances the model's capacity to detect diverse anomaly types without relying on pixel-level supervision or handcrafted features. Importantly, the method does not require labeled anomalies or segmentation masks, making it highly suitable for industrial settings where annotated data are scarce.

2.4.2 Reverse Distillation and Embedding Bottlenecks

While Uninformed Students [70] framed anomaly detection as a feature prediction task using homogeneous encoder-based student-teacher networks, the Reverse Distillation strategy [79] proposed a structural rethinking. This approach introduces architectural heterogeneity and a reversed flow of information between teacher and student, aiming to increase the representation of anomalous regions by breaking architectural and functional symmetry. [70].

Conventional ST methods Students [70] often employ similar network structures and share the same data input pipeline for both teacher and student. This results in overlapping inductive biases and insufficient variance in response to anomalies. In contrast, Reverse Distillation addresses this by pairing a frozen encoder-based teacher with a decoder-based student, and by reversing the information flow. Rather than passing raw input to both networks, the teacher first extracts high-level feature embeddings which are then compressed into a low-dimensional one-class bottleneck embedding (OCBE).

The student decoder receives this compact representation and attempts to reconstruct the teacher's features. This architectural asymmetry serves multiple purposes. First, it introduces filter diversity, eliminating redundancy due to similar receptive field

activations between teacher and student. Second, the low-dimensional OCBE acts as a semantic bottleneck that encourages the student to retain only anomaly-free patterns, as anomalous activations are treated as perturbations and filtered out during embedding. Third, the student's reconstruction target spans multiple feature layers from the teacher, enabling the detection of both local and global anomalies.

The anomaly score is computed by evaluating the cosine similarity [80] between the teacher's and student's multiscale features. Cosine similarity measures the angle between two vectors, indicating how aligned their directions are, regardless of their magnitude. Both, the memory efficiency and inference speed was improved, as only the compact embeddings and decoder weights require storage. In addition, the integration of multi-scale feature fusion (MFF), which aggregates semantic information from different network depths to enhance spatial and contextual representation, and the one-class embedding (OCE) module, which constrains feature learning to focus exclusively on normal class characteristics by projecting them into a compact, discriminative embedding space, further enhances the model's ability to isolate and distinguish normal patterns from anomalies.

However, the ability of Reverse Distillation in handling logical anomalies remains limited. The student is trained to reconstruct multiscale features from a low-dimensional one-class embedding (OCBE), which filters out anomalies through semantic compression. This approach effectively suppresses visual outliers and improves localization structure-level irregularities. However, the model's decision process remains grounded in fidelity of low-dimensional features reconstruction, without modeling relationships between object parts or enforcing semantic consistency across spatial regions. As such, Reverse Distillation underperformed in scenarios where anomalies are defined by relational inconsistency, such as parts being semantically valid but positioned in implausible arrangements. These limitations suggest Reverse Distillation does not explicitly account for the structured reasoning that is required to detect logical anomalies [81, 82].

2.4.3 Memory-Guided Distillation and Normality Forgetting

Building upon the architectural asymmetry introduced in Reverse Distillation, Memory-Guided Knowledge Distillation (MemKD) [83] addresses the limitation normality forgetting, which refers to the student's tendency to lose fidelity to normal feature patterns over time. It introduces an adaptive memory mechanism that explicitly preserves and reinforces the representation of previously encountered normal patterns. Instead of relying solely on instantaneous reconstruction loss [80], it incorporates stored feature priors to stabilize the student's feature responses over time. The memory module is integrated with the reconstruction objective, enabling the model to more effectively differentiate transient noise from consistent anomaly signals.

MemKD further acknowledges that student networks may either overfit to normality, leading to reduced sensitivity to true anomalies or become overly reactive to benign variations. To address this, the model employs a normality recall (NR) memory module that recalibrates the student's intermediate representations based on previously stored high-fidelity features from normal data. This is implemented using a key-value memory structure, where key entries assign relevance weights for retrieval, and value entries store prototypical normal features that are fused with current student representations to reinforce normality alignment.

In contrast to earlier methods such as Reverse Distillation [13], MemKD maintains a traditional encoder-encoder ST pairing but integrates a memory-mediated feedback loop. This augmentation allows the student to adaptively align its features with stored nominal references at each stage of inference. A second core component is the normality embedding learning (NEL) strategy. During training, a set of exemplar images is used to construct a normality embedding bank from the teacher network, which guides the memory module to generalize nominal patterns. This refinement evolves the student-teacher paradigm from passive reconstruction based discrepancy detection toward a more proactive modulation of the student's latent space using long-term normality priors.

Although MemKD enhances feature-level robustness by recalibrating the student's representations with recalled normal priors, its efficacy for logical anomaly detection presents both strengths and caveats. By embedding high-fidelity normal patterns into the student's feature space, the model gains increased resilience against noise and benign variations that typically mislead standard reconstruction-based detectors. However, the method's reliance on memorized exemplars may limit its flexibility when faced with logical anomalies that involve relational violations or unseen spatial configurations. Since the recall mechanism reinforces consistency with previously learned features, anomalies that manifest as novel part-to-part dependencies or semantic dislocations may not be fully captured. Thus, while MemKD offers strong generalization in representing known normality, its structural bias toward memorized appearances could underrepresent the anomaly space associated with higher-order logical inconsistencies [84, 85].

2.4.4 Projection-Based Suppression of Anomalous Signals

Reverse Distillation++ [86] revisits the reverse distillation (RD) framework and introduces an enhanced variant that addresses key limitations in previous student–teacher architectures. RD++ identifies and corrects two critical shortcomings in RD that includes the insufficient suppression of anomalous signals during training and the absence of explicit compactness constraints in the student's projected feature space. These limitations reduced anomaly localization precision and compromised feature reconstruction.

The RD++ architecture retains the reverse flow principle of RD but introduces lightweight projection layers after each intermediate block of the teacher network. These layers map teacher features into compact representations before passing them to the one-class bottleneck embedding (OCBE) module and eventually the student. To improve anomaly suppression, the training process incorporates pseudo-anomalous signals generated by simplex noise injection, which is a biologically plausible perturbation model that simulates realistic deviations from nominal distributions.

The optimization objective is formulated as a multi-task learning loss. It combines four components: (i) the standard cosine distillation loss between teacher and student; (ii) a self-supervised optimal transport loss (SSOT) to encourage compactness among projected normal features; (iii) a reconstruction loss ensuring the student can recover normal structure from noisy pseudo-anomalies; and (iv) a contrastive loss that maximizes the margin between normal and perturbed feature embeddings. Together, these terms guide the student to form a tightly clustered and robust representation of normality.

Despite its improved capacity to suppress anomalous signals and enforce compact feature representations, RD++ remains fundamentally limited in its ability to handle logical anomalies. The projection-based suppression mechanism is highly effective in eliminating low-level reconstruction noise and enhancing spatial precision in detecting structural deviations. However, its detection logic is grounded in localized feature discrepancy, without modeling higher-order semantic relationships or contextual dependencies among parts. As a result, RD++ may fail to recognize anomalies that involve plausible visual elements in scenarios where anomaly cues emerge from violations of spatial logic, or relational consistency—factors that fall outside the inductive scope of feature wise similarity metrics. [81, 82].

2.4.5 Aligning Global and Local Context Using Multi-Heads

The GAP framework [87] introduces a dual-branch architecture that enhances anomaly localization by integrating both global and local contextual information. Unlike conventional student–teacher approaches that rely primarily on reconstruction or regression in feature space, GAP adopts a discriminative matching strategy. Anomaly detection is performed by comparing the features of each image patch against those of its surrounding spatial context, allowing the model to identify inconsistencies in relational structure. This design addresses a key limitation of patch-level models, which often fail to detect relational anomalies.

The model used two parallel subnetworks consisting of a Local-Net that embeds the selected patch and a Global-Net that infers the surrounding region via masked convolutions to avoid information leakage. To evaluate consistency between these two representations, the authors introduce a dual-head detection module, composed of an Inconsistency Anomaly Detection (IAD) head and a Distortion Anomaly Detection (DAD) head. The IAD head captures semantic inconsistency between the patch and its neighborhood, while the DAD head detects subtle structural distortions within the patch itself. The two outputs are combined through a learnable fusion mechanism to produce the final anomaly score.

GAP does not explicitly rely on a teacher-student regression loss or a distillation objective; instead, it operates as a self-supervised relational matcher, making use of partial convolutions, negative patch synthesis, and multi-head contrastive learning. The training involves cross-entropy and contrastive losses to discriminate normal patches from perturbed or spatially inconsistent ones. Unlike memory-based methods, this architecture is compact and efficient, requiring no feature banks or dense storage. GAP represents an important conceptual link between ST-based feature regression and logic-aware anomaly segmentation. Its dual-head modeling of spatial coherence and structural integrity directly motivates designs such as CSAD's local-global dual student framework, which similarly decomposes anomaly cues into region-specific and context-driven components.

GAP is designed to identify mismatches between a patch and its surrounding visual context, thereby improving detection of misplaced but visually plausible regions. This offers an advantage over traditional patch-based detectors, which often miss such relational cues. However, the framework does not incorporate explicit modeling of object dependencies, symbolic rules, or scene-level logic. As such, GAP's capacity for logical anomaly detection is confined to local-global spatial misalignments, without extending to abstract or relational violations that require structured reasoning [88, 89]. An architectural Summary of Student–Teacher Based Methods is given in Table 2.3

Table 2.3: Architectural Summary of Student–Teacher Based Anomaly Detection Methods

Method	**Architecture Symmetry**	**Bottleneck / Projection**	**Memory or Prior Recall**	**Multi-Scale Features**	**Contextual Awareness**
Uninformed Students [70]	Symmetric (Encoder–Encoder)	None	No memory	Multi-layer fusion	Absent
RD [70]	Asymmetric (Encoder–Decoder)	Bottleneck (OCE layer)	No memory	Hierarchical feature use	Absent
MemKD [83]	Symmetric (Encoder–Encoder)	None	Normality memory	Multi-layer fusion	Memory-based context
RD++ [86]	Asymmetric with projection heads	Bottleneck + projection	No prior memory	Hierarchical feature maps	SSOT-based spatial context
GAP [87]	Dual-branch (Patch–Context)	None	No memory	Multi-level representation	Relational matching

2.5 Logical AD Methods

The emergence of logical anomaly detection (LAD) represents a evolution in the visual anomaly detection literature, aimed at identifying failures through violations of object-level structure, semantics, and relational integrity. In recent years, several works have proposed fundamentally different mechanisms to model such logical constraints, leading to a fragmented landscape of LAD architectures with varying degrees of semantic fidelity, supervision requirements, and computational cost. In this section, we explore methods that represent strategies for embedding logical constraints into the anomaly detection pipeline.

2.5.1 Component-Aware Anomaly Detection

Component-aware Anomaly Detection (ComAD) [59] introduces a logic-centric framework that leverages unsupervised segmentation and statistical reasoning to detect anomalies arising from component-level violation. Unlike prior methods that rely on appearance-based scoring over patches or features, ComAD conceptualizes an image as a composite of semantic components, each of which may carry distinct logical roles.

To model this, ComAD constructs a pipeline that begins with unsupervised object-part segmentation using features from a pretrained DINO-ViT backbone [90]. These features are clustered via KMeans [91], with resulting segments refined using Conditional Random Fields (CRFs) [92] to enhance spatial coherence and eliminate noise. This segmentation process yields a set of pseudo-part regions, treated as proxy components of the object. From this point, ComAD departs from standard anomaly detection practices by computing metrological statistics over these components, including their area, color histogram, centroid location, and frequency across training samples. Anomalies are then detected by measuring deviations from this learned distribution using k-Nearest Neighbor (kNN) searches and outlier scoring in the derived component feature space. Notably, the

method does not assume access to part-level ground truth or pixel-wise annotations, enabling it to remain label-free while still modeling logical structure. It also avoids the need for part segmentation masks or object detection heads, relying entirely on the emergent properties of self-supervised ViT features [90].

A core contribution of ComAD is its introduction of adjustable anomaly importance, allowing users to assign semantic weights to components depending on their functional importance. This is critical in industrial applications where certain parts (e.g., safety-critical connectors) demand higher scrutiny than cosmetic ones. Furthermore, ComAD offers high interpretability by generating anomaly maps that are traceable to individual pseudo-components, making inspection outcomes both explainable and actionable. However, several trade-offs are present. First, the model's effectiveness is tightly coupled to the quality of unsupervised segmentation, which may vary significantly across domains. In cluttered scenes or deformable objects, pseudo-components may be spatially fragmented or inconsistently clustered. Second, ComAD relies on aggregated component-level statistics, and while this captures part-based deviations, it may lack relational expressiveness i.e., it does not model part-to-part dependencies or geometric constraints directly. Moreover, the pipeline introduces computational overhead during segmentation and statistical analysis, especially in high-resolution settings.

2.5.2 Part Segmentation-Based Anomaly Detection

Part Segmentation-based Anomaly Detection (PSAD) [93] advances logical anomaly detection by explicitly modeling the compositional structure of industrial products through part-level semantic segmentation. Unlike ComAD, which relies on unsupervised component clustering, PSAD introduces a few-shot supervised segmentation strategy [94, 95] to generate high-fidelity component maps using only a limited number of labeled images. This segmentation step enables the system to reason about logical constraints within an image in a way that is consistent with manufacturer-defined logic.

At the core of PSAD is a segmentation pipeline jointly trained on visual and positional features, supervised by cross-entropy and Dice losses [96, 97] on labeled samples, and regularized with entropy and histogram-matching losses [98, 99] on unlabeled data. These losses ensure consistency in segmenting components across similar samples, even when labeled data is scarce. The segmentation output is then used to construct three memory banks, each capturing a different dimension of logic-aware representation: (1) a class histogram memory to model expected part distributions, (2) a component composition memory to encode typical inter-part relationships, and (3) a patch-level feature memory to capture fine-grained local texture details. During inference, anomaly scores from each memory bank are computed via nearest-neighbor searches, then adaptively normalized using statistics from the training set to enable multi-source score fusion. This adaptive scaling ensures robustness to differences in score distributions and prevents the dominance of any single memory bank. Notably, PSAD demonstrates that segmentation-aware logic reasoning is not only effective for logical anomalies but also improves structural anomaly detection—achieving state-of-the-art AUROC on both LA and SA categories in MVTec LOCO AD.

One of PSAD's primary innovations lies in its balance between supervision and generalization, as it uses only five labeled images per product type to drastically reduce annotation overhead while outperforming prior unsupervised LAD models. It also addresses key weaknesses in unsupervised segmentation-based LAD methods (e.g., ComAD), which suffer from ambiguous part boundaries and weak consistency. However, PSAD's reliance on segmentation labels—even in small quantities may still limit its scalability to unseen classes or diverse real-world deployments. The method also inherits the computational costs of dense segmentation and multi-memory inference. In the broader LAD landscape, PSAD establishes a principled pipeline that formalizes logic modeling through segmentation priors. While it is the most label-efficient segmentation-driven model to date, it contrasts with CSAD, which eliminates segmentation entirely by distilling logical

structure directly through dual-branch student-teacher training. This divergence highlights CSAD's contribution: the first LAD model to embed logical consistency within the training dynamics rather than as a separate component recognition step.

2.5.3 Logic-Aware Detection at Industrial Speeds

EfficientAD [8] proposes a novel direction in logic-aware anomaly detection by combining the conceptual richness of student–teacher frameworks with autoencoder-driven logic modeling, all under an aggressively optimized computational budget. Unlike PSAD and ComAD, which emphasize explicit segmentation or logic priors, EfficientAD operates without segmentation supervision, delivering high logical anomaly detection accuracy at millisecond-scale latencies. This is achieved through a synergy of architectural simplification, selective feature learning, and strategic fusion of anomaly maps.

At its core, EfficientAD introduces a lightweight Patch Description Network (PDN) [8] for both teacher and student, distilled from WideResNet-101 [55], yet capable of generating patch-level descriptors in less than 1 ms. This enables the system to produce a dense feature grid at a throughput exceeding 600 images per second. The student model is trained to mimic the teacher on normal images but is constrained using a hard feature loss and pretraining penalties to avoid overfitting. These losses focus the student on harder-to-learn regions and suppress its tendency to generalize outside the normal data manifold.

To bridge the gap between spatially local and globally structured anomalies, EfficientAD augments the student-teacher setup with an autoencoder that attempts to reconstruct the teacher's feature space. Logical anomalies—such as misplaced components or extra parts—cause the autoencoder to fail, especially in terms of feature fidelity. A second student head is then trained to predict the autoencoder's reconstruction. Discrepancies between the autoencoder and this predictive head form the global anomaly map, capturing logic-level violations (e.g., wrong arrangement or missing elements), while the

traditional student-teacher difference provides local anomaly maps for structural issues. A key design contribution is the calibrated fusion of these anomaly maps. By computing quantile-based normalization [100, 101] on validation data, EfficientAD ensures that both maps contribute meaningfully to the final anomaly score. This enables it to avoid the common trade-off between logic accuracy and localization sharpness.

However, EfficientAD's global logic modeling remains approximate: it does not use component-level representations or segmentation masks, which limits its ability to explain what is wrong in a human-readable way. Additionally, although it avoids part supervision, it does require training time to converge the joint autoencoder–student loss, unlike inference-only approaches. Compared to PSAD or ComAD, EfficientAD provides a speed-accuracy compromise, targeting fast, deployable inspection pipelines with robust logic reasoning, albeit with reduced interpretability. In relation to CSAD, EfficientAD's two-headed architecture and implicit logic modeling serve as a conceptual precursor. CSAD, however, builds upon this by embedding logical structure directly into the dual-branch distillation process (LGST), removing the need for auxiliary decoders and improving alignment between visual and logical anomalies at training time. Table 2.4 provides key comparison of logical anomaly detection methods that are discussed.

2.6 Summary of Literature Review Discussion

Having systematically reviewed the existing literature, it becomes evident that each class of methods was introduced to address specific limitations observed in preceding approaches. This progression reflects a gradual shift in the operational definition of anomalies, moving from low-level structural deviations to high-level logical inconsistencies. The following synthesis summarizes how these methodological advances unfolded and how the resulting insights have informed the design decisions in our proposed **AeC-SAD** method.

Table 2.4: Comparison of Logical Anomaly Detection Methods by Supervision, Segmentation Use, Logic Modeling, and Efficiency

Method	**Supervision**	**Part Segmentation**	**Logic Modeling**	**Explanation Quality**	**Runtime**
ComAD [59]	Fully unsupervised	Clustered parts	Region-level statistical metrology	High (segment-based traceability)	Moderate (post-hoc clustering)
PSAD [93]	Few-shot labels (5 per type)	Supervised part maps	Multi-memory (histogram, texture, composition)	High (part-aware attention)	Low (segmentation + 3 memory banks)
EfficientAD [8]	Trained without labels	Not used	Feature discrepancy via autoencoder	Moderate (score fusion, implicit logic)	High (1–4 ms per image)
CSAD [24]	Fully unsupervised	Not used	Dual-branch distillation (local–global semantics)	High (LGST-based attribution)	High (lightweight inference)

From Mahalanobis to kNN-Based Methods Mahalanobis-based approaches form an early class of methods for visual anomaly detection, where deep features extracted from convolutional networks are modeled as samples from multivariate Gaussian distributions. These methods rely on global covariance estimation and assume unimodal feature distributions, which limits their ability to capture spatially localized anomalies or variations across object categories. To overcome these limitations, non-parametric distance-based approaches, particularly k-Nearest Neighbor (kNN) scoring in deep embedding space, were introduced. Unlike Mahalanobis models that impose global parametric assumptions, kNN-based methods operate directly on exemplar sets and preserve local neighborhood structures, making them well-suited for detecting localized or irregular defects. This transition marked a shift toward retrieval-based detection paradigms that offered improved robustness to structural variability and category-level diversity.

From kNN-Based Methods to Student–Teacher Frameworks Although kNN-based methods effectively preserve local feature structure and avoid strong distributional assumptions, these methods introduced increased memory requirements, reduced scalability, and limited context awareness along with the lack of an explicit training objective to model normality. These methods operate purely during inference and do not capture how anomalous patterns diverge from normal representations during training. To address this limitation, student–teacher frameworks were introduced. In this formulation, a student network is trained to replicate features generated by a fixed teacher network using only normal data, and anomalies are detected as deviations in the student's predicted representations. This approach introduces two key contributions to anomaly detection: first, it enables learning-based feature consistency as a supervisory signal, and second, it allows the integration of architectural inductive biases, such as projection layers, memory components, or asymmetry between teacher and student networks, which can enhance sensitivity to semantic or logically inconsistent patterns.

From Student–Teacher Models to Logical Anomaly Detection Although student–teacher frameworks introduced learning-based representations and enabled partial semantic abstraction, they remained largely limited to detecting structural anomalies such as texture defects, surface inconsistencies, and local geometric disruptions. As the field expanded to applications requiring higher-level reasoning, such as object assembly validation, part completeness checking, and semantic configuration verification, it became necessary to address logical inconsistencies that go beyond local structural deviations. Logical Anomaly Detection methods were proposed to meet this requirement by modeling compositional structure, spatial dependencies among parts, and object-level functionality. The discussed approaches were broadly categorized into three directions: (1) unsupervised logic-based reasoning using statistical constraints, (2) few-shot semantic part segmentation to capture part–whole relationships, and (3) efficient, part-agnostic logic approximation techniques. Each of these strategies offered a different trade-off in terms of supervision level, interpretability, and computational efficiency.

2.6.1 Literature-Inspired Contribution in AeCSAD

Despite these methodological advances, most prior approaches including statistical models, retrieval-based techniques, and logic-aware frameworks share a fundamental limitation. They rely primarily on localized feature representations and lack explicit mechanisms to model long-range dependencies or global semantic relationships across the image. In contrast, self-attention mechanisms, particularly those employed in Transformer-based architectures, provide a principled means of encoding multi-scale contextual relationships. When integrated into anomaly detection frameworks, self-attention enhances the model's ability to reason not only about local inconsistencies but also about global coherence and inter-part dependencies. Based on these observations, we incorporate a self-attention block into the global student network within the CSAD architecture.

Table 2.5: Summary Comparison of Anomaly Detection Methods Across Methodological Categories

Method	**Category**	**Supervision**	**Logic Awareness**	**Segmentation Use**	**Self-Attention**
PatchCore [14]	kNN-Based	Unsupervised	None	Not used	No
PNI [64]	kNN-Based	Unsupervised	Spatial context cues	Not used	No
ReConPatch [65]	kNN-Based	Unsupervised (contrastive)	Contextual matching	Not used	No
InReaCh [69]	kNN-Based	Unsupervised	Cross-instance structure	Not used	No
Uninformed Students [70]	ST	Unsupervised	Prediction error (implicit)	Not used	No
Reverse Distillation [79]	ST	Unsupervised	Bottleneck-based abstraction	Not used	No
MemKD [83]	ST	Unsupervised	Memory-enhanced logic	Not used	No
RD++ [86]	ST	Unsupervised	Pseudo-anomaly logic	Not used	No
GAP [87]	ST	Unsupervised	Patch–context matching	Not used	No
ComAD [59]	Logical AD	Unsupervised	Statistical component reasoning	Part maps (unsup.)	ViT features
PSAD [93]	Logical AD	Few-shot (5 labels)	Explicit logic via parts	Few-shot segmentation	Transformer encoder
EfficientAD [8]	Logical AD	Unsupervised	Approx. via discrepancy	Not used	No
CSAD [24]	Unified (CSAD)	Unsupervised	Local–global distillation	Not used	No

Chapter 3

Technical Preliminaries

This chapter provides the conceptual background of AeCSAD framework based upon CSAD [24] modules that are necessary to understand the design choices and architectural elements that serve as key foundations for our proposed AeCSAD framework. We begin by introducing the usage of Recognize Anything Model (RAM++) [25] and move to the discussion of the Segment Anything Model (SAM) [27] and GroundingDINO [26], the two open-world segmentation frameworks that enable region proposal and component segmentation through visual and prompt-based mechanisms. We then describe the pseudo-labeling and Patch Histogram [39] strategy used for compositional reasoning, followed by a discussion of Local–Global Student–Teacher (LGST) learning module wherein we have made our contribution in this thesis.

3.1 Recognize Anything Model++

Recognize Anything Model++ (RAM++) [25] is an image tagging model trained for open-set scenarios where object categories are not fixed or known in advance. Given an image, RAM++ outputs a diverse set of tags that include possible object names, contextual nouns, and background-related words. RAM++ is applied to a single normal training image to facilitate the generation of candidate list of tags that serve as open-

vocabulary descriptors of the visible components in the scene.

Table 3.1: Examples of auto-generated tags from RAM++ [25] and the filtered tags retained for semantic reasoning in selected MVTec LOCO AD [5] categories. Adapted from [24].

LOCO Category	**Auto-generated Tags via RAM++**	**Semantic Tags Used for Segmentation**
Breakfast Box	almond, banana, cereal, granola, tray, food, container, fill, nut, oatmeal	almond, banana, cereal, nut, oatmeal, tray
Juice Bottle	apple juice, lemonade, bottle, jug, glass jar, milk, syrup, honey, orange juice, sauce	apple juice, lemonade, bottle, jug, glass jar, milk, syrup, honey
Splicing Connectors	cable, plug, socket, wire, electric outlet, pole, connector, attach, connect	cable, plug, socket, wire, connector

One of the key challenges in utilizing open-set models like RAM++ for component-level analysis lies in the noisy and heterogeneous nature of the tags they generate. These tags often include abstract terms, verbs, or irrelevant contextual elements that are unsuitable for structured reasoning. To address this, CSAD introduces a filtering step [24] that retains only semantically meaningful object-level nouns. Table 3.1 provides illustrative examples of RAM++ generated tags and their manually filtered counterparts across several categories from the MVTec LOCO AD [5] dataset. As shown, filtering eliminates non-object terms such as packaging elements or actions, resulting in a more focused tag

set that aligns with the goals of compositional reasoning.

3.2 Segmentation using SAM and GroundingDINO

The Segment Anything Model (SAM) [27] is a general-purpose image segmentation that is trained on a large-scale dataset comprising over one billion masks from millions of images. SAM exhibits strong generalization capabilities across diverse visual domains without requiring retraining or task-specific fine-tuning. A key characteristic of SAM is its promptable segmentation behavior, in which the model responds to simple user-provided prompts such as points or bounding boxes, and generates segmentation masks around the indicated regions without relying on fixed category labels. This design allows it to isolate object-like areas, making it effective in environments where explicit semantic annotations are unavailable.

SAM operates in a zero-shot setting, where no examples of the target classes are seen during training, which allows it to handle novel object types that it has not explicitly encountered before. Moreover, it is class-agnostic by design, meaning it focuses on segmenting coherent regions without needing to understand or predict their semantic categories [102, 103]. While SAM provides strong capabilities for spatial segmentation through visual prompts, it lacks the ability to distinguish or locate components based on semantic descriptions alone [104, 105]. To overcome this limitation, CSAD incorporates both SAM and GroundingDINO [26], leveraging their complementary strengths where SAM is used for visual region proposals, while GroundingDINO enables text-conditioned component identification.

GroundingDINO is a vision-language model designed for open-set object detection, enabling it to detect and localize objects in an image based on free-form text prompts. Unlike traditional closed-set detectors that require predefined object categories, GroundingDINO can respond to arbitrary descriptions, making it suitable for scenarios where

object types are undefined or vary widely across settings. Its strength lies in aligning visual content with natural language through training on large-scale datasets that pair regions with textual annotations. This alignment enables the model to semantically interpret image regions and locate relevant objects. In combination with SAM, Grounding DINO can guide segmentation using language prompts, producing bounding boxes for described objects. This fusion enables the segmentation of semantically meaningful objects or their components based on language input.

3.2.1 Limitations of SAM and GroundingDINO

Segment Anything Model (SAM) and GroundingDINO are trained on large-scale datasets featuring naturalistic imagery and broad semantic categories, enabling them to generalize across diverse visual domains [106, 26]. However, the architectural design and training objectives of these models are not well aligned with the specific requirements of logical anomaly detection in industrial settings [107].

SAM generates segmentation masks in response to spatial prompts, including points and bounding boxes, and is inherently class-agnostic. While this design supports generalization across object types, it also limits the model's ability to distinguish between semantically or functionally different components that share similar textures or shapes [106, 107].

GroundingDINO, while leveraging text-conditioned detection, presents complementary challenges. The effectiveness of its predictions depends on the quality and specificity of textual prompts. However, industrial parts rarely possess universally recognized or uniquely descriptive language, which limits the model's ability to locate or disambiguate components. Moreover, its detection capacity is shaped by priors inherited from its training on natural-image datasets, which are not representative of industrial part distributions [106, 107, 26].

Neither model natively supports compositional reasoning or relational understand-

ing. Their outputs are confined to instance-level predictions without validating global coherence or enforcing logical constraints across multiple parts. Consequently, when applied directly to scenes with complex part relationships, their segmentations tend to be inconsistent or ambiguous. *Due to these limitations, CSAD employs SAM and GroundingDINO not as inference-time modules, but as supervisory tools to guide the generation of pseudo-labels during training.*

3.3 Component Clustering and Pseudo-Labeling

Semantic pseudo-labeling serves as a bridge between automated mask generation and component level segmentation in industrial anomaly detection. In CSAD [24], mask proposals from SAM and GroundingDINO are refined to approximate individual components. These component masks are then clustered using MeanShift, which groups visually similar regions without requiring a predefined number of clusters. Each cluster is treated as a pseudo-label representing a component category. These pseudo labeled masks provide supervisory signals for training a semantic segmentation model, specifically DeepLabV3+ [108] to make it capable of decomposing objects into semantically meaningful parts. This enables detecting discrepancies in component count, spatial layout, or logical consistency central to the detection of logical anomalies.

3.3.1 Component Level Segmentation and Patch Histogram

The concept of compositional anomaly detection can be supported by integrating component level segmentation with *statistical modeling of spatial configurations.* One such approach uses Patch Histogram [39], which offers an explicit mechanism to capture regularities in component arrangement [39]. This strategy divides a segmented image into spatial bins and tracks the frequency of pseudo labeled components across regions. The resulting histograms reflect the compositional structure of normal samples, enabling the

identification of deviations such as missing, misplaced, or redundant parts [39].

Although segmentation-based reasoning provides interpretable and localized representations of object structure, it is inherently limited by the discretization imposed through pseudo-labels and spatial binning [24]. The Patch Histogram, for instance, operates under the assumption that logical anomalies manifest as explicit deviations in component count or location. However, not all anomalies conform to this pattern. Certain inconsistencies may involve subtle visual cues, ambiguous spatial relationships, or violations of overall composition of scene that cannot be captured through histogram statistics alone [107, 14]. These challenges highlight the need for a complementary mechanism that models visual representations in order to detecting deviations in both local and global structure.

3.4 Student-Teacher Paradigm as Detection Core

To complement segmentation and Patch Histogam based compositional reasoning, CSAD also incorporates a feature-based detection module using a student–teacher [70] framework. CSAD implements a pretrained teacher network that encodes normal features using normal training images, while two student networks, referred to as the (1) local student and the (2) global student, are trained to approximate these features. Discrepancies between the teacher and student outputs during inference indicate potential anomalies [70, 109, 110].

CSAD implements local and global students primarily different in their spatial receptive fields, which allows the system to capture anomalies at multiple levels. The local student is sensitive to fine-grained, localized deviations. In contrast, the global student integrates broader spatial context and is capable of detecting logical anomalies. By combining these complementary perspectives in the Local- Global Student-Teacher (LGST) module, CSAD enables detection of both spatially local and distributed anomalies. However, this division of focus is achieved implicitly through differences in receptive field size,

rather than through explicit reasoning over component relationships or compositional dependencies.

3.5 Enhancing CSAD With Self-Attention

Self-attention is a mechanism in modern deep learning, particularly in transformer-based models that plays a crucial role in capturing global dependencies across input data. Unlike convolutional operations, which are limited to fixed receptive fields and primarily capture local patterns, self-attention enables each input element to dynamically attend to all other elements in the sequence, allowing the model to reason over long-range relationships [29]. Given an input sequence $X \in \mathbb{R}^{n \times d}$, where n denotes the number of tokens (e.g., image patches or spatial positions) and d is the feature dimension, self-attention first projects the input into three learned spaces:

$$Q = XW^Q, \quad K = XW^K and \quad V = XW^V.$$

Here, $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$ are learnable weight matrices corresponding to the query, key, and value projections. The attention scores are then computed using the scaled dot-product:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V.$$

The scaling factor $\sqrt{d_k}$ prevents the dot products from becoming excessively large, which could otherwise destabilize training when passed through the softmax function. This mechanism produces a new representation for each token that encodes its relationship with every other token, allowing the network to incorporate both local and global context. In practice, multiple attention heads are used in parallel, enabling the model to capture diverse aspects of the input features.

In the context of this thesis, self-attention is employed to address a critical limita-

tion of conventional CNN-based anomaly detection methods: their inability to explicitly model dependencies between spatially distant regions. CNNs, despite their success in detecting structural anomalies, rely on local receptive fields and are limited in their ability to identify logical inconsistencies that depend on relationships across separate image components. Hence, our proposed AeCSAD, integrates self-attention to explicitly model relational dependencies among segmented components. By enabling global context reasoning, AeCSAD captures relationships among components allowing the model to also consider consider semantic understanding along with local appearance.

Chapter 4

Methodology

Our proposed method, Attention-Enhanced CSAD (AeCSAD), extends the CSAD framework [24], which is organized around three core modules. The first module performs semi-supervised component-level segmentation, while the second captures component frequencies for counting, as well as their spatial layout to identify structural regularities. The third module called the Local-Global Student-Teacher (LGST) module, includes two student networks namely a local student and a global student. These networks are trained to learn features extracted from a pretrained teacher network to help structural and logical consistency. In this work, we preserve this modular design and introduce a self-attention mechanism in the global student. This enhancement improves the model's ability to capture long-range relational dependencies between components, thereby strengthening its effectiveness in detecting logical anomalies.

4.1 Overview of the Proposed Framework

The overall pipeline starts by detecting distinct components within the images and generating corresponding semantic segmentation masks using open-vocabulary vision models, namely RAM++ [25], GroudningDINO [26] and SAM [27]. These masks are passed through a multistage process designed to group visually similar components and assign

pseudo-labels. It begins with component wise feature extraction, where each component is cropped and resized using the diagonal of its minimum bounding rectangle, then augmented through multiple rotations and passed through a pre-trained CNN [55] to obtain features. A rotation invariant stage in this process averages features across spatial and rotational dimensions to form compact representations that are robust to orientation.

Next, component clustering is performed using MeanShift [111] algorithm, which identifies clusters without requiring the number of clusters in advance. To ensure the reliability of the semantic pseudo-label maps, HDBSCAN [112], a hierarchical density-based extension of DBSCAN, is applied specifically on the class histograms of the pseudo-labeled maps rather than on component features. The purpose of this clustering is to distinguish between reliable and unreliable masks based on the consistency of their component distributions. As a result, the dataset is partitioned into two subsets: a labeled set $\mathcal{D}_L = \{X_1^l, \ldots, X_{N_l}^l\}$ with corresponding high-confidence pseudo-label maps $\{Y_1^l, \ldots, Y_{N_l}^l\}$, and an unlabeled set $\mathcal{D}_U = \{X_1^u, \ldots, X_{N_u}^u\}$ containing images with less consistent or uncertain masks.

The pseudo labeled masks form the basis for training a segmentation network, specifically DeepLabV3+ [108], in a semi-supervised fashion, where high-confidence pseudo labeled masks $\mathcal{D}_L$ are used as supervision signals through supervised objectives and low-confidence pseudo labeled masks $\mathcal{D}_U$ contribute only in unsupervised learning objectives to enhance component level understanding. The trained DeepLabV3+ can then generate final segmented masks that serve as input to a Patch Histogram module.

The Patch Histogram module generates a grid-based representation and captures compositional anomalies by encoding the frequency of part occurrences and outputs an anomaly score. In parallel, the Local-Global Student-Teacher (LGST) module is also trained. The LGST module consists of the student network, global student network, and pre-trained teacher network. LGST module compares feature maps from the local and global student with those of a teacher's feature maps, enabling the detection of

structural and global anomalies and generating an anomaly score. The anomaly score of the LGST module is combined with the anomaly score generated by Patch Histogram module through a normalization based fusion strategy to produce the final image-level anomaly score.

4.2 Semantic Pseudo-Label Generation

An effective component-level anomaly detection relies heavily on accurate segmentation masks that provides individual object parts. However, obtaining such masks through manual annotation is labor-intensive. We adopt the approach proposed in CSAD, which leverages an unsupervised method to automatically generate semantic pseudo-labels. This enables the training of a segmentation network without the need for human-annotated ground truth, making the process more scalable and efficient. The first stage of the pseudo label generation pipeline involves leveraging general purpose and open vocabulary vision models to produce component masks. This process comprises two substeps: (i) automatic textual tag prediction using RAM++[25], and (ii) mask generation using promptable segmentation models GroundingDino [26] and SAM [27]. These models are only used during the offline phase and are discarded during inference.

4.2.1 Generating Textual Tags Using RAM++

As discussed in Section 3.1, the Recognize Anything Model++ (RAM++) is an image tagging model that when applied to normal images, generates a diverse set of tags identifying visible components in the scene. The generated tag set is manually filtered to remove irrelevant or ambiguous terms and to retain background and meaningful object-related nouns. To derive semantic component labels from noisy visual tags, we follow the RAM++ filtering process proposed in CSAD [24]. The RAM++ conceptual pipeline process to generate textual tags for an image is shown in Figure 4.1.

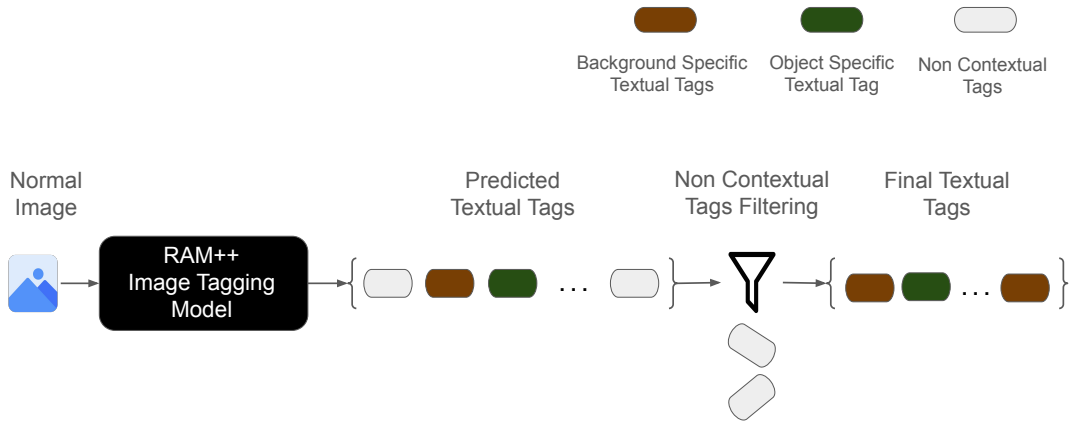


Figure 4.1: Conceptual illustration of textual tags generation using RAM++.

4.2.2 Generating Segmentation Masks for Object Components

The background and object specific textual tags that were derived from RAM++ are now passed to GroundingDINO [26], an open-vocabulary object detector that localizes relevant regions in the form of bounding boxes discussed in Section 3.2. A conceptual illustration of GroundingDINO pipeline is shown in Figure 4.2. Object and background prompts with image are processed to generate corresponding bounding boxes, which are subsequently normalized and rescaled to align with objects in the image. These bounding boxes are then used to guide the first run of the **Segment Anything Model (SAM)** [27], discussed in Section 3.2, which produces segmentation masks.

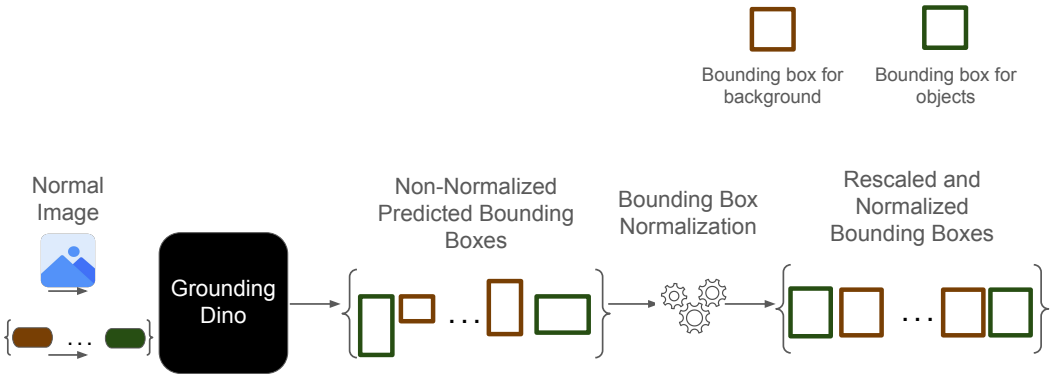


Figure 4.2: Conceptual illustration of background and object bounding box generation using GroundingDINO.

SAM takes the bounding boxes and produces corresponding object and background segmentation masks for a given image. Depending upon the prompt and visual context, these masks may capture entire objects, or their individual components resulting in semantic-guided masks as $\mathcal{M}_{\mathrm{G}}$. Among this set, masks that contain background regions, are explicitly identified and separated from at this stage using background specific textual tags obtained from RAM++. Subsequently, SAM is executed a second time in automatic segmentation mode i.e independent of any textual input or bounding boxes. This run is specifically intended to move beyond high-level object segmentation and toward capturing finer-grained structural details i.e, the individual components that make up an object. In its default mode, SAM generates a dense set of candidate masks by analyzing local visual features throughout the image. The output of this stage is a dense set of masks, denoted as $\mathcal{M}_{\mathrm{S}}$, which captures candidate component regions. Due to the generic nature of foundation models and the absence of explicit supervision, the initial candidate mask pool $\mathcal{M}_{\mathrm{S}}$ contains significant redundancy and noise, including background segments, fragmented parts, and over-merged components. To address this, we follow CSAD's two two-stage filtering strategy.

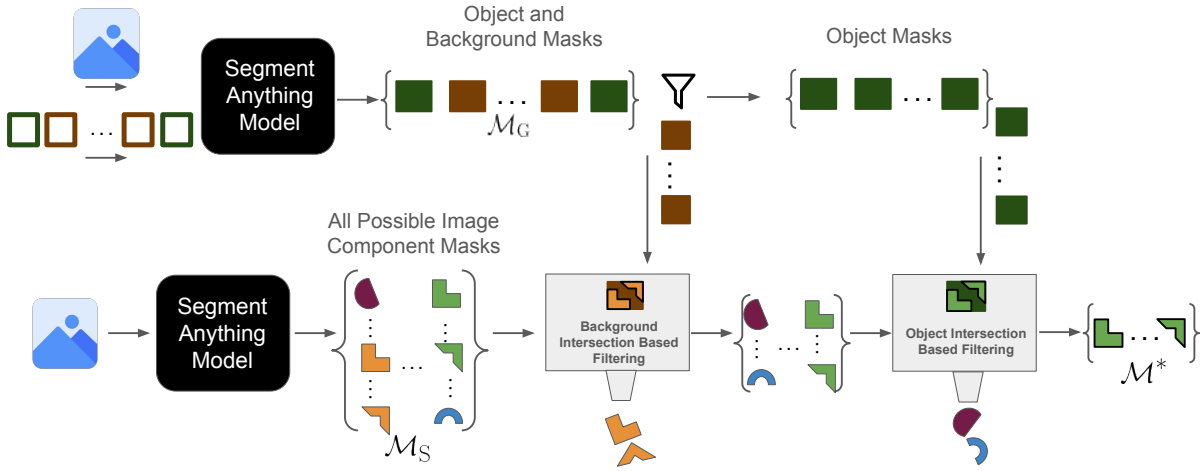


Figure 4.3: Component-level mask generation using SAM. Stage one uses bounding boxes and prompts to produce object and background masks, forming the set $\mathcal{M}_{\mathrm{G}}$ after removing background. Stage two runs SAM in automatic mode to generate candidate masks $\mathcal{M}_{\mathrm{S}}$, which are refined through (i) background intersection filtering and (ii) object alignment with $\mathcal{M}_{\mathrm{G}}$, yielding the final mask set $\mathcal{M}^*$.

The first stage, *Background Intersection Based Filtering*, ensures semantic alignment. Each candidate mask in $\mathcal{M}_S$ is compared with the set $\mathcal{M}_G$ produced by GroundingDINO guided SAM. Masks with insufficient spatial overlap relative to any mask in $\mathcal{M}_G$ are discarded. This enforces that retained regions in $\mathcal{M}_S$ correspond to plausible object components. The second stage, *Object Intersection Based Filtering*, addresses the issue of spatial compositionality. SAM's dense proposals may include masks that erroneously merge multiple adjacent components into a single region. To fix it, the spatial redundancy of each candidate mask is evaluated by computing the intersection statistics with all other masks in $\mathcal{M}_S$. Masks that exhibit excessive overlap without contributing distinct coverage are eliminated, while those representing well-separated entities are combined. The result of these two stages is a refined mask set, denoted $\mathcal{M}^*$, characterized by stronger semantic alignment. The complete conceptual process is shown in Figure 4.3 The complete summarized conceptual understanding for component level semantic segmentation mask generation process is shown in Figure 4.4

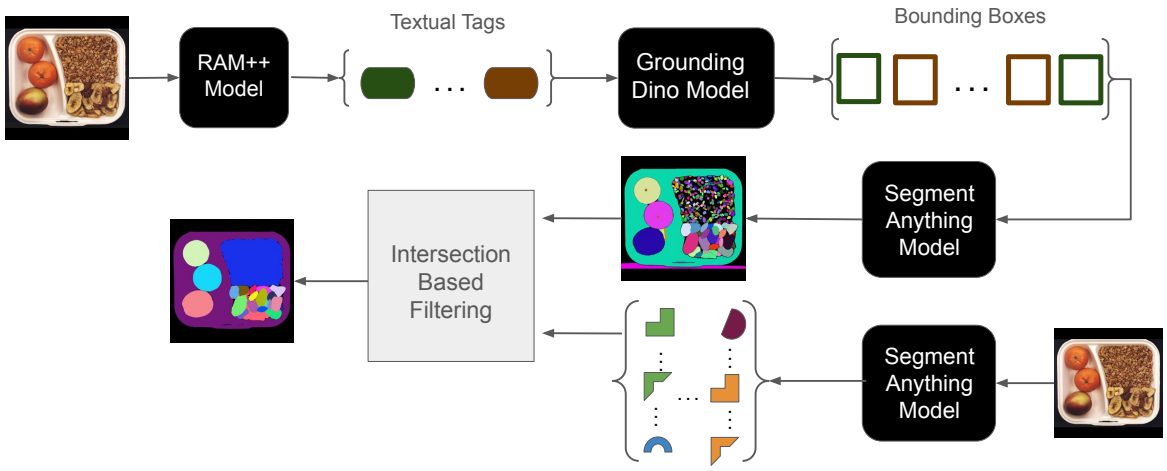


Figure 4.4: Summarized conceptual overview of pseudo label generation pipeline. The input image from the breakfast category of the MVTec LOCO AD dataset is processed by the RAM++ model to extract open-vocabulary textual tags corresponding to both object and background concepts. These are used by GroundingDINO model to generate bounding boxes, which are used to generate segmentation masks using SAM. An intersection-based filtering strategy integrates masks from two independent SAM branches to produce the final semantic pseudo label masks.

To ensure that components with similar shape and texture are assigned the same pseudo label, it is important to minimize the impact of rotational variation during feature extraction. To achieve this, each retained mask $m_i \in \mathcal{M}^*$ is used to extract a square crop from the original image, enclosing the component region. This step results in a set of image patches $\mathcal{P} = \{p_i\}$, where each p_i corresponds to an object component that undergoes through rotations at 0°, 90°, 180°, and 270°, as well as horizontal and vertical flips. Each patch is then resized to 64×64 and passed through a pretrained WideResnet-50 encoder $\phi(\cdot)$, which extracts features from the 4th layer. These features are aggregated to form a fixed-length representation $f_i = \phi(p_i) \in \mathbb{R}^d$. The resulting set of vectors for each image forms a feature matrix $F^{(i)} = [f_1, f_2, \ldots, f_{n^{(i)}}]^T$, where $n^{(i)}$ denotes the number of components in the ith image. This matrix serves as input for unsupervised clustering and subsequent pseudo-label assignment.

4.3 Clustering and Semantic Label Inference

To assign semantic pseudo labels in an unsupervised manner, the framework uses the MeanShift algorithm [111], which avoids predefining the number of clusters. MeanShift identifies clusters of similar components in the visual feature embedding space extracted from component patches, where each cluster center denotes a group of similar parts. Once clustering is complete, each patch p_i is assigned a cluster label $c_i \in \{1, 2, ..., K\}$, where K is the number of clusters. These cluster assignments are then mapped back to the original segmentation masks, producing pseudo-label maps for each training image. To improve reliability of the pseudo labels masks, hierarchical density-based clustering HDBSCAN is applied [112] to image-level class histograms where each histogram summarizes distribution of pseudo label classes within an image. This clustering step discovers the most representative pattern and captures a structure that occurs consistently across multiple training images. Segmentation masks that fall outside this pattern often due to under

or over segmentation artifacts, are treated as unreliable. A conceptual understanding is shown in Figure 4.5

Based on this process, the dataset is partitioned into two subsets: a high-confidence labeled set $\mathcal{D}_L = \{(x_i^{(l)}, y_i^{(l)})\}$, consisting of images with consistent and trusted pseudo-label maps, and an unlabeled set $\mathcal{D}_U = \{x_j^{(u)}\}$, where the masks are considered too noisy for direct supervision. This dual-structured dataset forms the foundation for the semi-supervised training of component level segmentation network. While each pseudo labeled component corresponds to a localized region within the image, training the segmentation network requires dense masks that annotate the entire image with semantic class indices. To achieve this, a complete pseudo-label map is reconstructed for each training image by compositing the retained component masks based on their spatial locations. Specifically, the cluster label c_i assigned to each component that was determined using MeanShift clustering, is used to fill the corresponding region of the image with its class index. This yields a dense semantic label map $Y \in \mathbb{N}^{H \times W}$, where each pixel is annotated with a pseudo-class that represents its associated component.

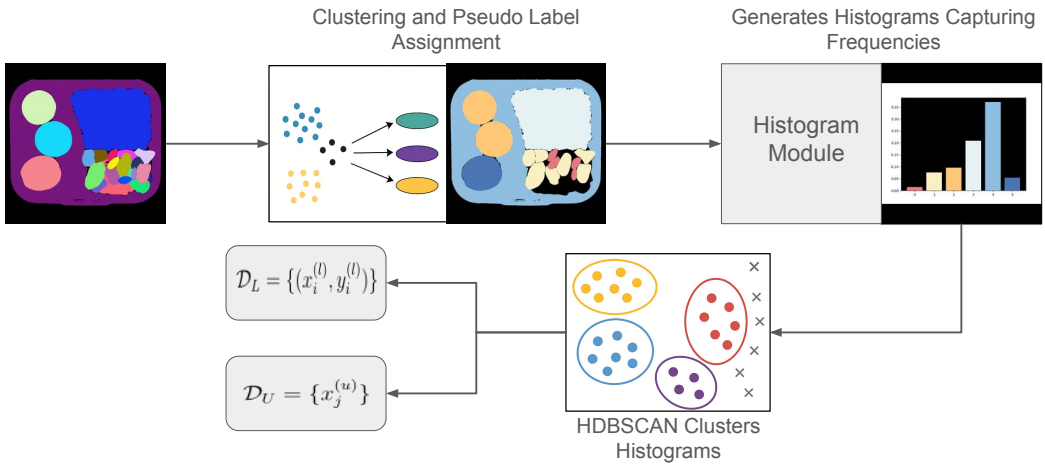


Figure 4.5: Overview of the clustering and histogram pipeline. Pseudo-label masks are clustered into semantic classes, followed by histogram extraction to capture component distributions.

4.4 Component Level Segmentation Network

Component level segmentation network enables the identification of discrete parts and supports downstream reasoning tasks involving the counting of components or detecting the positional misalignments. By following CSAD, we employ domain-adaptable segmentation network DeepLabV3+ [108] trained entirely using semantic pseudo labeled masks derived from unsupervised clustering. DeepLabV3+ employs a decoder architecture built on top of a WideResNet-50 encoder pretrained on ImageNet [60, 56]. The encoder extracts multi-scale features from intermediate layers indexed 1, 2, and 3. These features are fed into the DeepLabV3+ decoder, which employs an atrous spatial pyramid pooling (ASPP) module [108] to aggregate contextual information across multiple receptive fields. The use of ASPP enables the decoder to incorporate broad semantic context while preserving spatial granularity, which is essential for dense component segmentation. These features, denoted as F_T^{seg}, provide rich contextual information. During training, the encoder is kept frozen, and only the decoder is updated, ensuring a balance between semantic expressiveness and computational efficiency.

4.4.1 Training the Component Segmentation Network

With the dataset now partitioned into high-confidence and low-confidence subsets as described in the previous Section 4.3, the component segmentation network is trained in a semi-supervised fashion. The labeled subset $\mathcal{D}_L$ provides direct supervision through standard segmentation losses, while the unlabeled subset $\mathcal{D}_U$ contributes additional training signals to encourage semantic consistency and prediction confidence.

The learning process jointly optimizes segmentation losses computed over $\mathcal{D}_L$, including cross-entropy loss $\mathcal{L}_{\text{CE}}$, Dice loss $\mathcal{L}_{\text{Dice}}$, and focal loss $\mathcal{L}_{\text{Focal}}$, each formally defined in Section 4.4.2. To leverage information from $\mathcal{D}_U$, two auxiliary losses are employed. First, the histogram alignment loss $\mathcal{L}_{\text{Hist}}$ enforces global consistency by aligning predicted

class distributions with those in $\mathcal{D}_L$. Second, the entropy minimization loss $\mathcal{L}_{\text{Entropy}}$ encourages confident predictions by penalizing uncertainty. The training process adopts a staged learning strategy in which supervision is initially restricted to the high-confidence subset $\mathcal{D}_L$. Once the model has established stable representations under trusted guidance, the unlabeled subset $\mathcal{D}_U$ is introduced to incorporate broader variation. This training setup, where strongly supervised losses guide learning on $\mathcal{D}_L$ while auxiliary objectives regularize predictions on $\mathcal{D}_U$, reduces the risk of overfitting to noisy pseudo labels and encourages the model to learn generalizable features and provides better segmentation. The final loss function is a weighted sum of five terms, each addressing a different aspect of segmentation quality.

4.4.2 Loss Functions for Component Segmentation

Each component of the total loss function plays a distinct role in guiding the segmentation network to produce accurate and robust masks, especially in semi-supervised settings:

1. Cross-Entropy Loss

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{c=1}^{C} w_c \, y_{ij}^{(c)} \log \hat{y}_{ij}^{(c)} \tag{4.1}$$

This is the standard pixel-wise classification loss, where $y_{ij}^{(c)} \in \{0, 1\}$ is the one-hot ground truth label for class c at pixel (i, j), and $\hat{y}_{ij}^{(c)}$ is the predicted probability. The class-specific weights w_c address class imbalance. This loss ensures the predicted segmentation matches the labeled annotations.

2. Dice Loss

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i,j}\sum_{c=1}^{C} y_{ij}^{(c)} \hat{y}_{ij}^{(c)}}{\sum_{i,j}\sum_{c=1}^{C}\left(y_{ij}^{(c)} + \hat{y}_{ij}^{(c)}\right) + \epsilon} \tag{4.2}$$

Dice loss measures region overlap between predicted and ground truth masks. It is

especially effective in imbalanced settings where some classes may occupy small spatial areas.

3. Focal Loss

$$\mathcal{L}_{\text{Focal}} = -\sum_{i,j}\sum_{c=1}^{C} \alpha_c \left(1 - \hat{y}_{ij}^{(c)}\right)^{\gamma} y_{ij}^{(c)} \log \hat{y}_{ij}^{(c)} \tag{4.3}$$

Focal loss emphasizes harder, misclassified examples by reducing the relative importance of well-classified ones. The parameter γ adjusts the focusing effect, and α_c balances class contributions.

4. Histogram Matching Loss

$$\mathcal{L}_{\text{Hist}} = \sum_{c=1}^{C} \left(\frac{1}{HW} \sum_{i,j} \hat{y}_{ij}^{(c)} - \frac{1}{HW} \sum_{i,j} y_{ij}^{(c)} \right)^2 \tag{4.4}$$

This loss encourages the predicted class histogram to match the global distribution of the ground truth. It helps maintain component frequency and layout consistency across the image.

5. Entropy Loss

$$\mathcal{L}_{\text{Entropy}} = -\sum_{i,j}\sum_{c=1}^{C} \hat{y}_{ij}^{(c)} \log \hat{y}_{ij}^{(c)} \tag{4.5}$$

Entropy loss encourages confident predictions in the absence of ground truth, making it useful in semi-supervised learning. It penalizes uncertainty and promotes sharper class probabilities.

The final loss function is a weighted sum of five terms, each addressing a different aspect of segmentation quality:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{CE}} + \lambda_2 \mathcal{L}_{\text{Dice}} + \lambda_3 \mathcal{L}_{\text{Focal}} + \lambda_4 \mathcal{L}_{\text{Hist}} + \lambda_5 \mathcal{L}_{\text{Entropy}}, \tag{4.6}$$

where $\lambda_1, \ldots, \lambda_5$ are empirically chosen weights. A typical configuration uses $\lambda_1 = 1.0$,

$\lambda_2 = 0.5$, $\lambda_3 = 0.5$, $\lambda_4 = 0.3$, and $\lambda_5 = 0.1$.

4.4.3 Lightweight Segmentation for Real-Time Inference

One of the design goals of this segmentation network is to operate without relying on heavyweight foundation models such as GroundingDINO [26] and SAM [27] during inference. While those models are valuable in pseudo-label generation, they are computationally prohibitive for real-time or embedded use. During inference, each test image is processed through this DeepLabV3+ decoder to produce a component level semantic segmentation map that encodes the spatial distribution of component classes. This segmentation map is subsequently utilized by the Patch Histogram module to generate histograms as discussed in Section 4.5.

4.5 Patch Histogram Module

The Patch Histogram module operates on semantic segmentation maps generated by component segmentation network discussed in Section 4.4. Its primary function is to capture the spatial frequencies and layout of component classes across the image. This enables the system to identify logical anomalies that manifest as abnormal spatial distributions.

Given a segmentation map $S \in \mathbb{N}^{H \times W}$, where each pixel is assigned a cluster index, the image is conceptually partitioned into a $P \times P$ spatial grid. Each patch in this grid defines a localized region B_{ij} of the image, where $i, j \in \{1, ..., P\}$. Within each bin B_{ij}, a histogram vector $h_{ij} \in \mathbb{N}^C$ is defined, where C denotes the number of pseudo-classes. Each entry h_{ij}^c encodes the number of pixels in region B_{ij} that are assigned to component class c. These histograms serve as structured descriptors of local semantic composition, forming the basis for detecting part-level irregularities in subsequent anomaly reasoning stages.

To ensure comparability across samples and to reduce sensitivity to scale variations,

each local histogram is normalized such that it expresses relative frequency rather than absolute pixel counts. These normalized histograms are then concatenated in raster-scan order to produce a global feature vector $H \in \mathbb{R}^{P^2 \cdot C}$, which encodes the spatial distribution of component classes across the image grid. This grid-aware embedding serves as a soft structural blueprint, capturing the expected layout of object parts.

To statistically model normal structural patterns, histogram vectors H_i are computed for all training samples belonging to the normal class. From these, the empirical mean $\mu \in \mathbb{R}^{P^2 \cdot C}$ and covariance matrix $\Sigma \in \mathbb{R}^{(P^2 \cdot C) \times (P^2 \cdot C)}$ are estimated, defining a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ that characterizes the typical spatial configurations under normal operating conditions. During inference, a test sample yields a histogram vector H^{test}, which is evaluated against the learned distribution using the Mahalanobis distance:

$$A_{\text{hist}}(H^{\text{test}}) = \sqrt{(H^{\text{test}} - \mu)^\top \Sigma^{-1} (H^{\text{test}} - \mu)}. \tag{4.7}$$

This anomaly score A_{hist} reflects deviations in component arrangement, accounting for both inter-class and inter-bin correlations. Higher values indicate stronger departures from the normal layout, thus signaling potential logical anomalies.

4.6 Attention Enhanced Autoenconder in AeCSAD

We propose AeCSAD that extends the Local Global Student Teacher (LGST) module of CSAD by introducing a key architectural enhancement in the global student network. While CSAD's LGST module adopts a student-teacher paradigm consisting of a frozen teacher network $\mathcal{T}$ and two student networks $\mathcal{S}_{\text{local}}$ and $\mathcal{S}_{\text{global}}$, AeCSAD retains this structure but modifies the global student. Specifically, the global student, originally implemented as an autoencoder, is enhanced with self-attention blocks metnioned in Section 3.5 and Section 1.5.

This modification enables the model to better capture long-range dependencies and

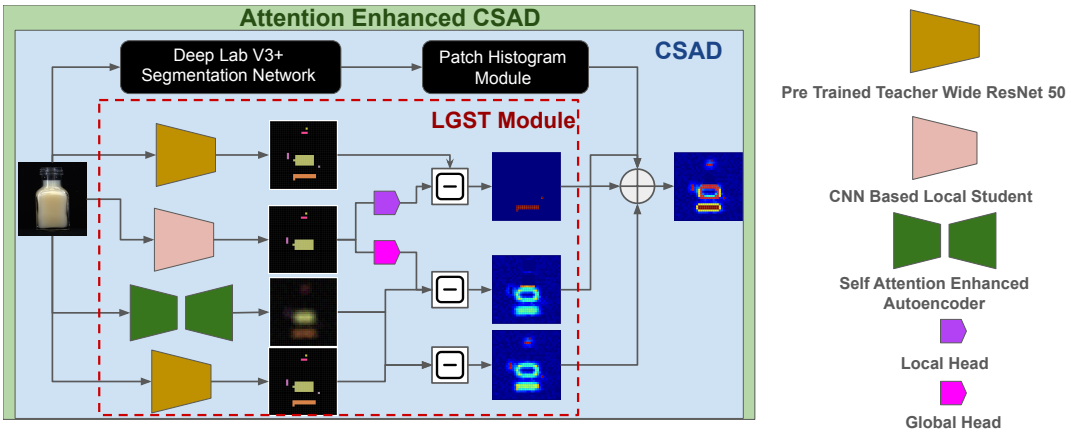


Figure 4.6: Conceptual illustration of anomaly map generation in AeCSAD. LGST module consists of a frozen teacher network $\mathcal{T}_{\text{teacher}}$, based on a pretrained WideResNet-50 encoder; a CNN-based local student $\mathcal{S}_{\text{local}}$; and a global student $\mathcal{S}_{\text{global}}$, implemented as a self-attention enhanced autoencoder. Both student branches aim to replicate the teacher's feature representations, and their deviations are used to produce multi-scale anomaly maps. The segmentation network and Patch Histogram module are used downstream to support semantic-guided anomaly detection.

contextual patterns during reconstruction, thereby improving its ability to model structural normality in complex scenes. All three networks receive the same input image $x \in \mathbb{R}^{H \times W \times 3}$. The students are optimized to minimize their divergence from the teacher under normal training data. During inference, failure of the student to reproduce the teacher's representation implies an anomaly. This framework enables the detection of both localized structural defects and global semantic inconsistencies without requiring annotated anomaly data.

4.6.1 Teacher Network

The teacher network $\mathcal{T}$ is implemented using a WideResNet-50 [55] backbone pretrained on ImageNet [56]. This architecture is selected for its widened residual blocks, which enhance representational capacity while preserving spatial granularity, an essential property for identifying fine-scale industrial components. The teacher extracts feature maps from the second and third convolutional stages, capturing both local structural detail and high-level semantic context. These multi-scale features are spatially aligned, concate-

nated, and projected to a fixed-dimensional representation via a 1×1 convolution. The resulting feature map, denoted as $F_T^{\text{teacher}} \in \mathbb{R}^{B \times C \times H \times W}$, serves as a stable supervisory signal for the local and global student networks.

4.6.2 Local Student Network

The local student $\mathcal{S}_{\text{local}}$ is a shallow convolutional network. Its convolutional layers consists of small kernels (3×3) to preserve local detail. There are no downsampling or pooling operations, which constrains the receptive field to approximately 33×33 pixels in the input space. This ensures that the local student is sensitive to fine-grained structural anomalies. Given an input image, the local student produces a feature map $F_L \in \mathbb{R}^{B \times C \times H \times W}$ that mirrors the shape of the teacher's output F_T^{teacher}, ensuring spatial alignment for element-wise comparison. The features predicted by the local student are compared with the corresponding teacher features at each spatial location to compute the anomaly map A_{local}. This map is calculated as the per-pixel Euclidean distance:

$$A_{\text{local}}(i,j) = \left\| F_T^{\text{teacher}}(i,j) - F_L(i,j) \right\|_2 \tag{4.8}$$

Here $A_{\text{local}}(i,j)$ denotes the local anomaly score at pixel location (i,j). This network provides dense anomaly maps highlighting localized visual deviations such as surface cracks, missing rivets, or misaligned holes.

4.6.3 Global Student Network

The original global student network $\mathcal{S}_{\text{global}}$ in CSAD [24] utilizes a bottleneck architecture that progressively reduces the spatial dimensions of the feature map, ultimately collapsing it to a 1×1 resolution. This design ensures that the receptive field encompasses the entire image, allowing the network to encode high-level contextual and structural information.

To compare with the teacher features, the compressed representation is subsequently

upsampled through a projection layer that restores the spatial dimensions. This results in a dense feature map $F_G \in \mathbb{R}^{B\times C\times H\times W}$, which mirrors the shape of the teacher output F_T^{teacher}, ensuring spatial alignment for element-wise comparison. The features predicted by the global student are compared with the corresponding teacher features at each spatial location to compute the global anomaly map using:

$$A_{\text{global}}(i,j) = \left\| F_T^{\text{teacher}}(i,j) - F_G^{\text{student}}(i,j) \right\|_2 \tag{4.9}$$

Here $A_{\text{global}}(i,j)$ denotes the global anomaly score at pixel location (i,j), reflecting discrepancies in high-level semantic structure between the teacher and student networks. Moreover, the global student features $F_G^{\text{student}} \in \mathbb{R}^{B\times C\times H\times W}$ are also compared with the local student outputs $F_L^{\text{student}} \in \mathbb{R}^{B\times C\times H\times W}$ at each spatial location to compute the global-local anomaly map using:

$$A_{\text{global-local}}(i,j) = \left\| F_G^{\text{student}}(i,j) - F_L^{\text{student}}(i,j) \right\|_2 \tag{4.10}$$

Here $A_{\text{global-local}}(i,j)$ denotes the combined anomaly score that captures inconsistencies between coarse semantic reasoning and local pixel-level reconstructions. This bottlenecked design captures high-level statistics; however, the reliance on CNN encoders and average pooling limits the model's ability to encode long-range dependencies and inter-component semantic structure.

4.6.4 Limitation of Global Student Network in CSAD

The bottlenecked CNN encoder architecture is a heavily downsampled representation to summarize input features. This is typically achieved through a series of strided convolutions followed by a final projection that collapses spatial dimensions to a single vector. Such compression removes cues related to spatial arrangement, part orientation, and component co-occurrence. It assumes that all relevant information is encoded in feature

content alone, ignoring the role of structure in determining semantic correctness. It fails to model relational semantics, e.g. the left-right symmetry of parts or presence of invalid co-occurrence patterns because spatial alignment is discarded during pooling. In scenarios where anomalies result from altered spatial arrangements, such as repositioned or duplicated components, the compressed global representation often lacks the granularity to distinguish these from valid layouts [113]. This limitation arises because global representations prioritize content-based feature aggregation, often learned through channel mixing, while disregarding spatial dependencies that are critical for interpreting structural coherence [114]. Consequently, models relying on this type of global representation are prone to missing relational anomalies.

4.7 Attention Enhanced Global Student Network

To overcome these limitations, we redesign the global student architecture using stacked self-attention layers that preserves spatial resolution while modeling inter-region dependencies. The proposed AeCSAD integrates self-attention blocks directly into the encoder and decoder pathways of the autoencoder based global student. Unlike convolutional filters that capture only local context, self-attention allows every spatial position to interact with every other, enabling the network to learn dependencies across distant regions. Self-attention serves two main roles: (1) it models long-range interactions between spatial locations, and (2) it dynamically weighs features based on their contextual importance. The resulting output encodes not only visual content but also spatial relationships and part configurations.

4.7.1 Global Student Architecture in AeCSAD

AeCSAD retains the convolutional backbone of the original encoder and interleaves self-attention layers after selected after fourth and fifth convolutional blocks to introduce

global contextual reasoning during mid-to-high level feature extraction. These stages are chosen because they preserve spatial granularity while capturing increasingly abstract representations. At each of these points, the intermediate feature maps are reshaped from (B, C, H, W) into token sequences of shape (B, N, C), where $N = H \times W$, treating each spatial location as an individual token. These sequences are then passed through single-head self-attention blocks, where each token attends to all others using scaled dot-product attention as described in Section 3.5. The attention scores modulate the contribution of each location to the contextualized feature map. The output is then projected back to the original tensor shape and fused with the input via a residual connection.

These lightweight attention modules effectively enhance the network's capacity to capture long-distance dependencies. Importantly, the final representation is not globally pooled. Instead, the full spatial resolution is maintained during decoding. This ensures that structural cues preserved by attention can influence the reconstruction directly, allowing the model to preserve and evaluate spatial correctness. The global anomaly map with teacher network and the global-local map with local student network are computed respectively using 4.11 and 4.12:

$$A_{\text{global}}(i,j) = \left\| F_T^{\text{teacher}}(i,j) - F_G^{\text{attention}}(i,j) \right\|_2 \tag{4.11}$$

$$A_{\text{global-local}}(i,j) = \left\| F_G^{\text{attention}}(i,j) - F_L^{\text{student}}(i,j) \right\|_2 \tag{4.12}$$

The attention-enhanced global student is fully compatible with the original LGST module, therefore, the same loss functions are reused to compare attention-enhanced features and are discussed in Section 4.7.2. During inference, the attention-enhanced global student yields features that encode both semantic identity and spatial configuration, and enables the detection of anomalies such as:

- Mirror-symmetric misplacements;
- Relational duplications (e.g., two identical parts next to each other); and
- Logical contradictions (e.g., two exclusive parts coexisting).

4.7.2 Loss Formulation and Joint Optimization

The LGST module is trained by jointly optimizing local student network and attention enhanced global student network to learn the teacher's feature maps. Let $F_T^{\text{teacher}} \in \mathbb{R}^{B\times C\times H\times W}$ represent the normalized feature maps produced by the teacher, where B is the batch size, C the number of channels, and $H \times W$ the spatial dimensions. The local student produces output $F_L \in \mathbb{R}^{B\times C\times H\times W}$, focusing on short-range spatial features, while the attention-enhanced global student generates $F_G^{\text{attention}} \in \mathbb{R}^{B\times C\times H\times W}$, leveraging self-attention to capture contextual relationships across spatially distant regions. The anomaly maps are computed as per-pixel Euclidean distances between the student predictions and teacher targets. The total training objective consists of three mean squared error (MSE)-based losses computed over all pixels and channels. These are defined as:

$$\mathcal{L}_{\text{local}} = \frac{1}{BCHW}\sum_{i,j}\left\|F_T^{\text{teacher}}(i,j) - F_L(i,j)\right\|_2^2; \tag{4.13}$$

$$\mathcal{L}_{\text{global}} = \frac{1}{BCHW}\sum_{i,j}\left\|F_T^{\text{teacher}}(i,j) - F_G^{\text{attention}}(i,j)\right\|_2^2; \text{ and} \tag{4.14}$$

$$\mathcal{L}_{\text{fusion}} = \frac{1}{BCHW}\sum_{i,j}\left\|F_G^{\text{attention}}(i,j) - F_L(i,j)\right\|_2^2. \tag{4.15}$$

Here $\mathcal{L}_{\text{local}}$, $\mathcal{L}_{\text{global}}$, and $\mathcal{L}_{\text{fusion}}$, respectively, measure the error between the teacher and local student, the teacher and global student, and between the global and local students. All three losses are computed as mean squared error (MSE) over all spatial positions (i, j), channels C, batch size B, and feature height and width $H \times W$. This multi-branch

objective encourages the local student to capture fine-grained spatial details, the global student to model semantic regularities, and their mutual consistency to be preserved through fusion alignment. The final loss aggregates these three terms as a weighted sum and is computed using:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{local}} \cdot \mathcal{L}_{\text{local}} + \lambda_{\text{global}} \cdot \mathcal{L}_{\text{global}} + \lambda_{\text{fusion}} \cdot \mathcal{L}_{\text{fusion}}. \tag{4.16}$$

Here, λ_{local}, λ_{global}, and λ_{fusion} are weighting coefficients that balance the influence of each loss component.

4.8 Anomaly Scoring and Final Decision

At the final stage of the pipeline, multiple signals reflecting different aspects of normality are combined to produce a unified anomaly score, where normality is defined as the presence of consistent geometrical structure and semantically valid arrangements among visual components that reflect both structural integrity and logical coherence. These signals include (i) distributional irregularities captured by the Patch Histogram module and (ii) feature-level errors from the attention enhanced LGST module. These modalities are complementary. Many anomalies manifest in layout rather than content, or vice versa.

To harness this complementarity, the anomaly scores from the two modules are fused together. Direct score fusion can be misleading due to differences in score ranges and statistical distributions across modules. Therefore, we follow CSAD's approach to combine and normalize scores using a trimmed z-normalization strategy. For a given raw score S, the normalized score $\hat{S}$ is computed as:

$$\hat{S} = \frac{S - \mu_S}{\sigma_S} \tag{4.17}$$

Here, μ_S and σ_S represent the trimmed mean and trimmed standard deviation of the

anomaly scores, respectively and are computed for both anomaly scores. These statistics are computed over a held-out validation set, excluding the lowest and highest 20% of values. This percentile trimming mitigates the influence of extreme outliers, promoting more robust and interpretable normalization under varied testing conditions.

The normalization is independently applied to:

- S_{hist}: the Patch Histogram anomaly score; and
- S_{LGST}: the attention enhanced LGST based anomaly score.

The final image-level anomaly score is obtained by a straightforward additive fusion of the normalized scores from the histogram and attention enhanced LGST modules

$$S_{\text{final}} = \hat{S}_{\text{hist}} + \hat{S}_{\text{LGST}}. \tag{4.18}$$

where $\hat{S}_{\text{hist}}$ denotes the normalized structural anomaly score derived from histogram module, and $\hat{S}_{\text{LGST}}$ represents the normalized semantic anomaly score obtained via attention enhanced LGST module.

Chapter 5

Results and Discussion

This chapter outlines the implementation details, training strategy, and hardware configuration used to develop the proposed AeCSAD anomaly detection framework. It also describes the evaluation metrics and presents the results obtained by testing AeCSAD on the MVTec LOCO AD dataset. The chapter concludes with a detailed discussion of the observed results and insights. The experimental setup closely follows the protocol established in the original CSAD framework [24], ensuring a fair and consistent comparison.

5.1 Experimental Setup and Dataset

All experiments were conducted on a workstation equipped with an NVIDIA Tesla V100-SXM2-32GB GPU and an Intel Xeon Silver 4114 CPU. The operating system was Ubuntu 22.04.4 LTS, with CUDA 12.5 and PyTorch 2.4.1. The environment was managed using Conda, and fixed random seeds were applied across NumPy, PyTorch (CPU and GPU), and Python's random module to ensure consistent and reproducible results. The experiments in this study were conducted using the MVTec LOCO Anomlay Detection dataset [5], a benchmark dataset specifically designed to evaluate both structural and logical anomaly detection in industrial settings.

The MVTec LOCO AD dataset [5] consists of high-resolution images across five industrial object categories: breakfast box, juice bottle, pushpins, screw bag, and splicing connector. For each category, it provides normal samples and two distinct types of anomalies: structural and logical. The training set contains only normal data, while the test set includes both normal and anomalous samples.

We selected this dataset as the primary benchmark for our study because it explicitly incorporates logical constraints, such as correct object arrangements, presence or absence of components, and relational consistency, which are rarely addressed in other industrial datasets. This makes it uniquely suited for evaluating methods aimed at detecting not only localized physical defects but also context-aware, relational violations. As our method targets both structural and logical anomaly detection, MVTec LOCO AD offers the most appropriate and challenging benchmark for validating such capabilities.

To ensure consistency across the model pipeline, only minimal preprocessing was performed on the input images. No additional data augmentation techniques were applied beyond standard resizing, except during the pseudo-label generation stage. At this stage, patch-level augmentations such as random rotation and horizontal or vertical flipping were employed to enhance sample diversity and improve the robustness of the clustering process. Furthermore, all input images were normalized using the mean and standard deviation values of the [56] dataset, aligning with the statistics expected by the pretrained WideResNet-50 [55] backbone utilized throughout the model.

5.2 Architecture Design

The segmentation network is built upon WideResNet-50 [55] pretrained on ImageNet [56] based encoder and DeepLabV3+ [108] decoder. We have followed CSAD implementation and extracted features from layers 1, 2, and 3 of the WideResNet-50 encoder. These intermediate feature maps are passed to a DeepLabV3+ decoder, which integrates an

ASPP [108] to expand the receptive field and enhance multi-scale context aggregation. Batch Normalization [115] is employed after each convolutional block to enhance training stability and facilitate faster convergence by mitigating internal covariate shift. As the primary non-linear activation function, the Rectified Linear Unit (ReLU) [116] is used throughout the network to introduce non-linearity and improve model expressiveness. No dropout regularization is applied in the architecture.

The teacher network $\mathcal{T}$ is implemented using a WideResNet-50 [55] backbone pre-trained on the ImageNet dataset [56]. Following the design in CSAD, intermediate features are extracted from the second and third residual blocks (referred to as block2 and block3). Features from these two stages are upsampled to a fixed spatial resolution using bilinear interpolation and then concatenated along the channel dimension. This fused representation is subsequently passed through a fixed 1×1 projection layer to produce a unified feature embedding of size $512 \times 64 \times 64$, which serves as the teacher's output. All parameters in both the encoder and the projection layer are frozen throughout training.

The local student network $\mathcal{S}_{\text{local}}$ is designed as a lightweight CNN. It consists of a series of five convolutional layers interleaved with ReLU activations and two average pooling layers, progressively increasing the channel dimensionality from 3 to 512. The feature maps are then spatially resized using bilinear interpolation to a fixed resolution of 64×64, ensuring compatibility with the teacher's output. The final feature map is passed through two separate 1×1 convolutional heads, producing parallel outputs: one for alignment with the teacher's embedding (used in the student-teacher comparison), and another for reconstruction consistency with the autoencoder. Unlike the teacher, all parameters in the student network are trainable.

The global student network $\mathcal{S}_{\text{global}}$ is implemented as a self-attention enhanced convolutional autoencoder designed to capture long-range dependencies and contextual relations across the input image. To better preserve spatial context and model global interactions, two self-attention blocks are embedded within the encoder at resolutions of

16×16 and 8×8, respectively. The bottleneck representation is then upsampled through a multi-stage decoder, which includes both convolutional and attention-enhanced layers. Specifically, additional self-attention modules are integrated into the decoder at intermediate resolutions to reinforce global consistency during reconstruction. Unlike the teacher, all parameters in the global student are learnable. The autoencoder is trained to reconstruct the teacher's features rather than raw input.

5.3 Training Segmentation Network

The training of segmentation network DeepLabV3+ is conducted in two distinct stages over 120 epochs. To supervise the network, two parallel dataloaders are constructed. One is sourced from $\mathcal{D}_L$ mentioned in Section 4.4.1 and provides dense pseudo-label masks treated as ground truth. The other is built from $\mathcal{D}_U$ mentioned in Section 4.4.1 and contributes unlabeled images that support regularization. During training, the segmentation network receives batched inputs from both sources in each iteration.

During the first 40 epochs, the network is trained exclusively on the labeled dataset $\mathcal{D}_L$. In this phase, we also adopt the Logical Synthetic Anomaly (LSA) augmentation strategy introduced in the original CSAD framework [24]. LSA is selectively applied to those training samples whose pseudo-labels have passed a filtering criterion and are considered to be of high quality. The LSA process involves randomly sampling components from a source image, extracting them using their corresponding segmentation masks, and pasting them onto a different target image after applying random translation and rotation. This synthesis generates new training images with logical inconsistencies, and the associated pseudo-label maps are updated accordingly. These augmented samples are included in the supervised training pipeline to increase semantic diversity and expose the model to abnormal spatial configurations.

The labeled set is used to compute a combination of supervised losses including cross-

entropy, focal loss, and class-balanced Dice loss. These are combined using fixed weights ($\lambda_1 = 0.5$, $\lambda_2 = 10$, $\lambda_3 = 1$ respectively). All input images are resized to 256×256, converted to RGB format, and normalized to have zero mean and unit variance. The corresponding labels are segmentation masks of the same resolution where each pixel index indicates its cluster assignment. The number of semantic classes is variable depending on the category and typically ranges from 5 to 15. To mitigate class imbalance, especially when some components occupy disproportionately large regions, sample reweighting is applied during loss computation. Each training batch includes 16 images, and additional augmentations such as random horizontal and vertical flips, rotations in multiples of 90 degrees, color jittering, and Gaussian noise are applied in real time. After epoch 40, the training set uses unlabeled images. From this point onward, unsupervised losses are introduced: Histogram Matching Loss and Entropy Minimization Loss.

All training is performed with the Adam optimizer using an initial learning rate of 1×10^{-4} and a batch size of 16. A cosine learning rate scheduler is used to gradually reduce the learning rate to 1×10^{-6} by the final epoch. Early stopping was not applied, as the goal was to reach a consistent performance baseline for comparison.

5.4 Training Student-Teacher Networks

The student-teacher framework is trained using a combination of complementary loss functions discussed in Section 4.7.2 that supervise both local and global representation learning. Local supervision encourages the student to capture fine-grained spatial features from small receptive fields. In contrast, global supervision facilitate the detection of high-level logical inconsistencies.

The teacher network is initialized with a pretrained WideResNet-50 and remains frozen throughout the entire training process. It produces high-dimensional features extracted from layers 2 and 3. The teacher's features are first concatenated and re-

sized to a uniform resolution before being projected into a final embedding space of size $512 \times 64 \times 64$. The local student network is trained to replicate the teacher's feature embedding using the original input image, which is resized to 256×256 and normalized using ImageNet statistics. The output of the local student is interpolated to match the teacher's resolution. In contrast, the global student receives a logically augmented variant of the same input and is trained to reconstruct the teacher's embedding through a self-attention-enhanced autoencoder. We follow the training methodology of CSAD and include the Logical Synthetic Anomaly (LSA) augmentation process to simulate semantically invalid compositions. These augmented images are passed through both the teacher and global student networks. All components are optimized using the Adam optimizer with an initial learning rate of 0.0002 and weight decay of 10^{-4}. A step-based learning rate scheduler reduces the learning rate by a factor of 10, Training is conducted using a batch size of 16, with all inputs resized to 256×256 and the shared feature embedding space maintained at $512 \times 64 \times 64$.

5.5 Metrics and Results

To assess the performance of the proposed AeCSAD, we report the Area Under the Receiver Operating Characteristic curve (AUROC) as the primary image-level metric for evaluating anomaly detection across both logical and structural anomaly categories. This metric quantifies the ability to rank normal and anomalous images across a continuum of decision thresholds. Mathematically, AUROC is defined as the probability that a randomly chosen anomalous image receives a higher anomaly score than a randomly chosen normal image:

$$\text{AUROC} = \mathbb{P}(S_{\text{anomaly}} > S_{\text{normal}}), \tag{5.1}$$

where S_{anomaly} and S_{normal} denote the anomaly scores for anomalous and normal images, respectively. A score of 1.0 indicates perfect separation, while a score of 0.5 implies

Table 5.1: Comparison of MVTec LOCO AD performance with state-of-the-art methods, as measured by image AUROC (%). The "AeCSAD" column presents the results of our method. Bold value represents top value across categories or groups. A value underlined represents second to top across categories or groups.

	No Segmentation				**Segmentation**		
Category	SimpleNet	PatchCore	AST	EfficientAD	ComAD	CSAD	AeCSAD (Ours)
Logical Anomalies (LA)							
Breakfast Box	77.1	74.8	80.0	85.5	91.1	**95.8**	95.6
Juice Bottle	87.8	93.9	91.6	**98.4**	95.0	95.4	95.5
Pushpins	69.0	63.6	65.1	97.7	95.7	99.6	**100.0**
Screw Bag	51.6	57.8	80.1	56.7	71.9	99.3	**99.7**
Splicing Connectors	72.0	79.2	81.8	*95.5*	93.3	95.7	**97.0**
Average (Logical)	71.5	73.9	79.7	86.8	89.4	97.16	**97.56**
Structural Anomalies (SA)							
Breakfast Box	80.9	80.1	79.9	**88.4**	81.6	85.9	86.2
Juice Bottle	90.4	98.5	95.5	**99.7**	98.2	97.3	96.7
Pushpins	81.6	87.9	77.8	**96.1**	91.1	93.2	94.0
Screw Bag	83.3	92.0	**95.9**	90.7	88.5	91.8	94.5
Splicing Connectors	82.6	88.0	89.4	**98.5**	94.9	92.4	91.3
Average (Structural)	83.7	89.3	87.7	**94.7**	90.9	92.12	92.54
Total Average	77.6	81.6	83.7	90.7	90.1	94.64	**95.05**

performance equivalent to random guessing. Since our method builds upon CSAD, we inherit its lightweight training and inference pipeline, which involves training local and global student networks using only normal samples. The main architectural change in our approach is the introduction of a self-attention mechanism in the global student to better capture long-range dependencies for logical anomaly detection. While this modification adds moderate computational overhead during both training and inference due to the quadratic complexity of attention operations, the overall framework remains efficient and deployable in real-time settings. Other components, such as the local student and patch histogram module, remain lightweight and do not significantly impact runtime.

Following the evaluation protocol of CSAD, we compute AUROC scores for each of the five categories within logical and structural subsets and report the mean AUROC values for both categories separately. We computed all evaluation metrics using the test split of MVTec LOCO AD dataset. No data augmentation or test-time ensembling is used

during evaluation. These metrics also enable precise comparisons with existing baselines such as SimpleNet [117], PatchCore [14], AST [110], EfficientAD [8], ComAD [59], and the original CSAD [24] and the quantitative results are shown in Table 5.1.

5.6 Discussion

At a high level, AeCSAD achieves the **highest total average AUROC of 95.05%**, outperforming all other segmentation-free and segmentation based baselines. This indicates that the performance gains introduced by the proposed architecture generalize across diverse anomaly types. In the **logical anomaly categories**, the AeCSAD achieves an average AUROC of **97.56%**, exceeding the orginal CSAD (97.16%). In logical anomaly categories, AeCSAD has also outperformed No-Segmentation based methods such as PatchCore (71.5%) SimpleNet (73.9%), AST (79.7%) and EfficientAD (86.8%). The incorporation of self-attention into the global student enables the model to evaluate part-to-part consistency across spatially distant regions, leading to more semantically informed difference maps. In the **structural anomaly category**, the model also achieves good results, with an average AUROC of **92.54%**, again outperforming segmentation based methods ComAD (90.9%) and CSAD (92.12%). The Global Student, while enhanced with self-attention, retains its local sensitivity through early convolutions and refines spatial consistency via attention. To better understand the observed improvements, the following sections analyze the results through multiple lenses: architectural design and category-wise performance.

5.6.1 Interpretation of Logical Anomaly Results

Pushpins In the pushpins category, the model achieves a perfect AUROC of 100.0%, successfully identifying duplicated or irregularly placed components by leveraging relational cues. AeCSAD's self-attention enhanced framework enables holistic reasoning over

the entire image, allowing it to detect layout-dependent anomalies that are difficult for convolution-based methods like SimpleNet (69.0%) or PatchCore (63.6%) to capture. While CSAD (99.6%) perform competitively, AeCSAD matches or exceeds all strategies without relying on additional labels, demonstrating its strength in global spatial understanding.

Screw Bag Similarly, in the screw bag category, the model achieves a near-perfect AUROC of 99.7%, outperforming both ComAD (71.9%) and CSAD (99.3%). The anomalies in this category often involve duplicated or missing screws, where the texture and appearance of individual screws remain locally plausible. The difficulty lies in recognizing when the overall quantity or spatial arrangement deviates from expectations. AeCSAD's attention mechanism allows the model to evaluate layout of object globally, making it highly sensitive to such subtle logical inconsistencies. This advantage is especially clear when compared to segmentation-free methods like EfficientAD (56.7%) or AST (80.1%), which lack mechanisms to reason about part-to-whole relationships.

Splicing Connectors AeCSAD also performs strongly in splicing connectors (97.0%). In splicing connectors, logical defects typically emerge when individual segments are swapped, mirrored, or misaligned during assembly. AeCSAD captures these relational inconsistencies by enabling every image region to attend to all others. This ability proves especially advantageous compared to segmentation-free methods like AST (81.8%) and EfficientAD (95.5%), which often misinterpret plausible textures as valid configurations. It also surpasses ComAD (93.3%) and CSAD (95.7%), demonstrating that attention-based reasoning improves even over strong semi-supervised and unsupervised baselines.

Juice Bottle In the juice bottle category, logical anomalies typically manifest as missing caps, misplaced labels, or labels oriented in reversed directions. These anomalies challenge models to understand the expected spatial and semantic arrangement of visual

components. AeCSAD achieves an impressive AUROC of 95.5%, which is comparable to CSAD (95.4%). This strong performance reflects AeCSAD's ability to reason over long-range spatial dependencies and capture misalignments between distant parts of the object. The inclusion of attention-enhanced global features enables the model to detect when essential components, such as the cap or label, are improperly configured even if the individual parts are present. In contrast, segmentation-free methods such as SimpleNet (87.8%) and PatchCore (93.9%) underperform, primarily because they lack the architectural mechanisms to model part-to-part relationships. These methods rely heavily on local appearance and texture features, which limits their ability to recognize logically inconsistent but visually subtle anomalies.

Breakfast Box In the breakfast box category, the model scores 95.6%, closely trailing CSAD's 95.8%, yet still outperforming segmentation-free baselines by a significant margin; with the best among them, EfficientAD, achieving 85.5%, and others like AST and PatchCore trailing further behind. This narrow gap between AeCSAD and CSAD suggests that anomalies in this category may rely less on long-range contextual dependencies and more on localized visual cues. Many of the logical defects in breakfast box involve subtle shifts in position, missing flaps, or partially obscured elements, all of which are spatially confined and maintain plausible textures. Such anomalies can be effectively captured using convolutional operations with moderate receptive fields, which explains why CSAD remains competitive in this case.

5.6.2 Interpretation of Structural Anomaly Results

Breakfast Box In the breakfast box category under structural anomalies, AeCSAD achieves an AUROC of 86.2%, slightly outperforming CSAD (85.9%) and strongly outperforming ComAD (81.6%), while falling just short of the segmentation-free EfficientAD, which leads with 88.4%. The narrow margins across these models suggest that struc-

tural defects in this category are relatively localized or visually prominent, reducing the reliance on long-range relational reasoning. Common anomalies in this category include broken edges of the break fast box, damaged fruits, which are often texture-level issues occurring in small regions. These can be effectively captured by both convolutional models and segmentation-free approaches that are sensitive to local visual irregularities. Although AeCSAD is designed to enhance spatial reasoning, its ability to compete with EfficientAD and exceed other baselines reflects its balanced sensitivity to both global structure and fine-grained texture cues.

Juice Bottle In the juice bottle category, AeCSAD attains an AUROC of 96.7%, trailing behind unsupervised baselines including CSAD (97.3%), ComAD (98.2%), and segmentation-free methods like EfficientAD (99.7%) and PatchCore (98.5%). The high scores across all methods suggest that structural anomalies in this category are visually salient and can be detected with relatively low reliance on global context. Typical structural defects in juice bottle include damaged labels, decloration in juices. These features create clear local deviations in shape or texture and present cues that are easily captured by both convolutional models and segmentation-free approaches.

Pushpins In the pushpins category, AeCSAD reaches an AUROC of 94.0%, outperforming all unsupervised baselines except EfficientAD (96.1%). It also improves over CSAD (93.2%) and ComAD (91.1%). Structural anomalies in pushpins often include bent or broken pinheads, missing tips, or abnormal deformations, many of which involve fine-grained shape disruptions rather than large-scale layout changes. Such defects pose challenges for methods that rely solely on high-level semantic cues, and instead favor approaches capable of integrating both local detail and relational context. AeCSAD's strong performance here reflects its hybrid capability where it leverages self-attention to assess coherence across the image while maintaining sensitivity to local deformations through convolutional features. EfficientAD marginally surpasses AeCSAD, likely due to

its specialized design for efficient texture-level anomaly detection.

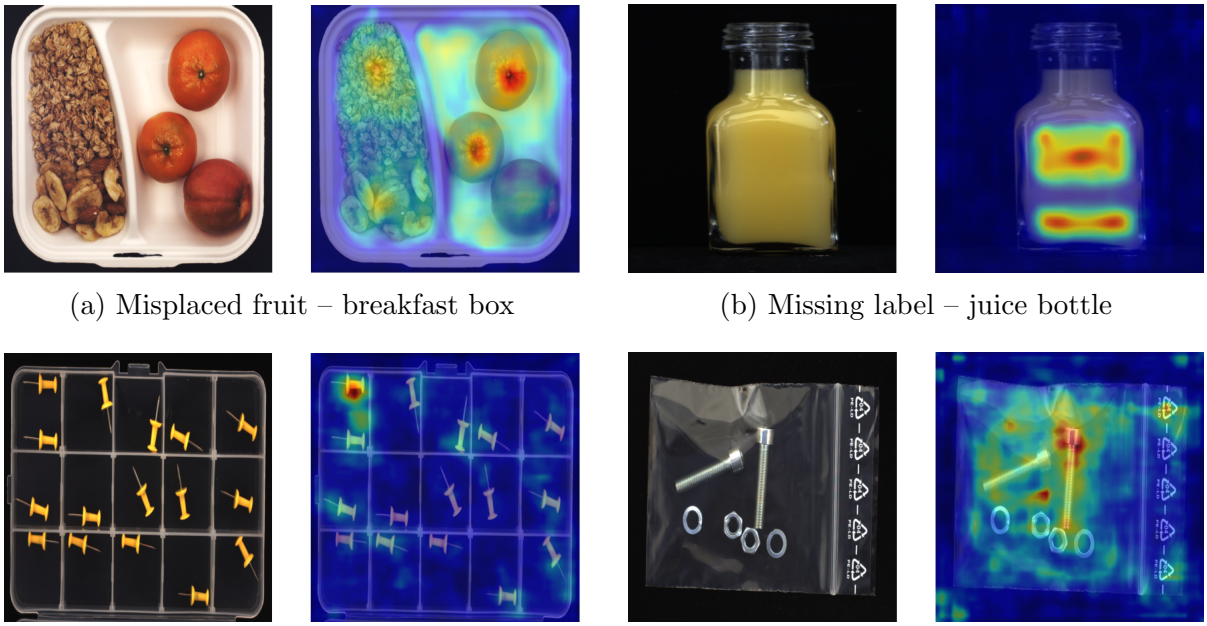
Screw Bag In the screw bag category, AeCSAD delivers a high AUROC of 94.5%, outperforming all unsupervised baselines including CSAD (91.8%) and ComAD (88.5%), while coming close to the best-performing method, AST (95.9%), highlighting its strength in detecting subtle yet structured defects that are visually complex. Structural anomalies in this category typically involve deformed screw shapes, inconsistencies in packaging seams, misalignments, or torn regions in the plastic bag. These distortions often vary in scale, location, and appearance, making this a particularly challenging setting that benefits from both local detail capture and contextual reasoning. AeCSAD's performance gain over CSAD suggests that the added global attention helps in assessing the overall coherence of the bag's visual layout by modeling long-range dependencies. While Asymmetric Student Teacher (AST) outperforms AeCSAD, the gap is marginal, illustrating that attention-based reasoning enhances structural anomaly localization in cluttered and spatially settings like screw bag, even with lightweight architectural modifications.

Splicing Connectors In the splicing connectors category, AeCSAD achieves an AUROC of 91.3%, placing it behind CSAD (92.4%) and further behind top-performing methods like ComAD (94.9%) and EfficientAD (98.5%). Although AeCSAD maintains a strong score, this result suggests that its architectural strengths may be less directly leveraged in this particular category. The structural anomalies in splicing connectors generally involve surface-level damage, broken connector edges, or scratches. Such defects are often confined to specific regions and exhibit strong local visual signals. Such anomalies are well suited to models like EfficientAD or ComAD that focus heavily on high-resolution, texture-based cues without requiring broad spatial reasoning. The dip in AeCSAD's performance relative to CSAD might be attributed to the trade-off introduced by self-attention which may not always prioritize local high-frequency deformations with the same granularity as texture-oriented baselines.

5.6.3 Qualitative Results

The qualitative results on some of the MVTec LOCO AD images shown in Figures 5.1 and 5.2 illustrate AeCSAD's ability to detect and localize both logical and structural anomalies. Each example shows the original image on the left and the corresponding anomaly heatmap generated by the attention-enhanced LGST module on the right. Figure 5.1 showcases logical inconsistencies that involve semantic violations in object relationships or expected configurations. These include misplaced components, missing parts, duplicated items, and improper arrangements. It shows the types of deviations that CNN-based detectors typically struggle with due to their limited receptive fields.

In contrast, AeCSAD correctly highlights such anomalies across diverse categories, benefiting from its self-attention-enhanced global student branch. For example, in the juice bottle (Figure 5.1b), the missing label is distinctly localized despite having no obvious low-level texture difference. Similarly, AeCSAD accurately flags the duplicated large screws in the screw bag (Figure 5.1d), demonstrating sensitivity to logical violations that arise from improper part counts or layout inconsistencies. Figure 5.2 displays structural anomalies, which involve localized visual defects such as breakage, contamination, or physical deformation. The local student network, operating over pixel-level feature reconstructions, effectively captures such fine-grained abnormalities. The attention maps reflect spatial deviations precisely around the defective regions. For instance, the system highlights the broken pushpin in the center compartment (Figure 5.2c) and the torn region in the screw bag (Figure 5.2d), emphasizing AeCSAD's ability to localize and interpret structural damage with high spatial fidelity. These visualizations demonstrate not only high detection accuracy but also model interpretability. The heatmaps provide spatial evidence of decision-making and reflect the complementary strengths of the LGST and histogram-based scoring modules. Logical anomalies often lead to broad or relational attention patterns, while structural anomalies produce sharply localized regions, reinforcing the dual-branch design of AeCSAD.



(a) Misplaced fruit – breakfast box

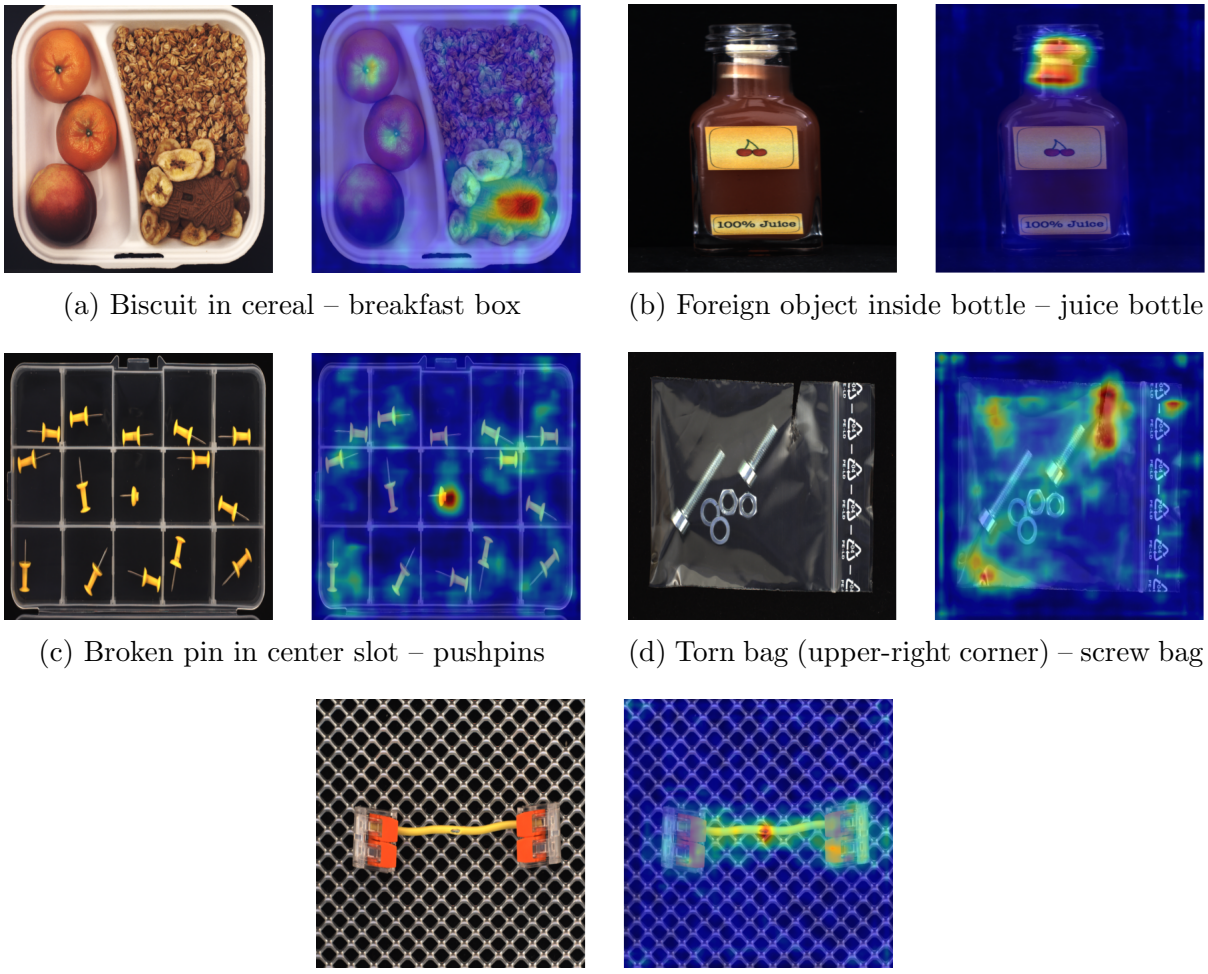
(b) Missing label – juice bottle

(c) Two pins in one compartment – pushpins

(d) Duplicate large screws – screw bag

(e) Miswired cable – splicing connectors

Figure 5.1: **Logical anomalies** detected by AeCSAD. Each pair shows the original image (left) and the corresponding attention-based anomaly map (right). These anomalies involve semantic inconsistencies such as missing components, relational duplications, or mirrored misplacements. AeCSAD effectively highlights these layout violations across multiple object categories in the MVTec LOCO AD dataset.



(a) Biscuit in cereal – breakfast box

(b) Foreign object inside bottle – juice bottle

(c) Broken pin in center slot – pushpins

(d) Torn bag (upper-right corner) – screw bag

(e) Scratched wire in center – splicing connectors

Figure 5.2: **Structural anomalies** detected by AeCSAD. Each pair shows the original image (left) and the corresponding attention-based anomaly map (right). These anomalies reflect physical or geometric defects such as object breakage, incorrect geometry, inclusion of unexpected items, or localized surface damage. AeCSAD accurately localizes these anomalies across diverse object types in the MVTec LOCO AD dataset using spatial attention mechanisms.

5.6.4 Architectural Analysis of Logical Anomaly Results

The improvements observed in the proposed method can also be understood through examination of a self-attention enhanced autoencoder. This change directly addresses the core limitation of the original CSAD architecture where it lacks the strong ability to reason over long-range spatial relationships.

The baseline CSAD framework utilizes a convolutional autoencoder to learn global consistency patterns from the teacher's latent representations. Although, this design enables reconstruction-based anomaly detection, it fundamentally inherits the local inductive bias of convolutional operations. By introducing a self-attention mechanism, the encoder is able to model relationships between spatially distant features, allowing the model to form a globally coherent latent representation. Moreover, the attention layers of the decoder reinforce semantic consistency during reconstruction, ensuring that the generated output conforms not just to local texture expectations but to the holistic structural layout learned from normal examples. This attention-based enhancement enables the global student to reason over what features should be present, where they should be located, and whether their spatial relationships are consistent with learned patterns of normalcy. As a result, the difference map between the teacher and student becomes more semantically informed and highlights logical inconsistencies.

5.6.5 Architectural Analysis of Structural Anomaly Results

The proposed method also achieves strong performance in detecting structural anomalies, despite introducing global reasoning mechanisms that could, in theory, de-emphasize local detail. Understanding this result requires a careful examination of how anomaly maps are derived from Teacher network $\mathcal{T}_{\text{teacher}}$, Local Student network $\mathcal{S}_{\text{local}}$, and Global Student network $\mathcal{S}_{\text{global}}$. The anomaly detection pipeline in the proposed framework computes two parallel forms of discrepancy with respect to the teacher output:

- The **Local Student** attempts to approximate the teacher's representation using shallow convolutional layers. Its architectural simplicity biases it toward learning low-level texture and edge patterns.

- The **Global Student**, redesigned with attention modules, reconstructs the teacher features in a way that captures semantic relationships and broader contextual patterns.

For structural anomalies, the Local Student plays a pivotal role. Structural defects often produce abrupt, local changes in visual appearance. As the Local Student has a limited receptive field and no mechanism to smooth over inconsistent regions with global context, it tends to produce high activation in areas with localized visual defects. This makes it highly sensitive to structural variations. On the other hand, the Global Student, even though attention-enhanced, continues to be trained to reconstruct teacher features. Attention mechanisms do not eliminate the model's ability to preserve local detail, rather, they enhance it by injecting context selectively. As a result, the Global Student still detects structural inconsistencies, but in a way that is spatially coherent and semantically aware.

5.6.6 Summary of Discussion

The presented results offer category-wise improvements along with implications for the broader field of unsupervised anomaly detection. Chief among these is the recognition that anomaly detection, especially in complex real-world environments, demands a capacity for both local perceptual sensitivity and global semantic reasoning. Traditional convolutional approaches are inherently limited by their inductive bias toward locality. While this is advantageous for identifying anomalies that manifest as sharp or localized defects, it leaves models vulnerable in scenarios where the anomaly is not defined by local pixel deviation but by higher-order inconsistencies in object structure, component

count, or spatial layout. These are precisely the types of anomalies found in logical categories, where the notion of normality depends on spatial relationships rather than texture anomalies.

The performance gains observed in this work across both logical and structural anomaly categories indicate that architectures which combine shallow, texture-aware modules with context-aware reasoning mechanisms can achieve robust generalization across fundamentally different types of anomalies. This suggests a broader paradigm for future models. Anomaly detection systems should be equipped with mechanisms that operate not only at multiple scales but also at multiple semantic levels. These levels range from texture and material composition to part–whole relations. Furthermore, the use of attention-based reconstruction as a student mechanism introduces an important shift in unsupervised settings. It enables anomaly detection models to learn not just "what is present" but also "what is expected" in relation to other components. In sum, the findings support a generalizable claim that **robust unsupervised anomaly detection requires architectural designs which explicitly model both appearance and arrangement**. The success of such hybrid models opens new avenues not only for industrial inspection but also for domains where data labels are sparse and anomalies are defined more by semantic logic than by pixel-level aberration.

Chapter 6

Conclusion

This thesis introduced AeCSAD, an anomaly detection framework that builds upon the CSAD [24] framework but introduces reasoning-level improvement that yields measurable performance gains. Our contribution is the architectural modification of the autoencoder-based Global Student network in the Local-Global Student-Teacher (LGST) module of CSAD. Whereas the original CSAD model employed a bottlenecked convolutional autoencoder for global scoring, the AeCSAD proposed in this thesis extends it with stacked self-attention blocks. The self-attention mechanism enables the network to model contextual interactions between components, a capacity that proves especially valuable in logical anomaly categories where part identities and arrangements must conform to implicit assembly logic.

The proposed AeCSAD retains modularity by adhering to CSAD's design principle of independently normalized scoring branches and late fusion. Each scoring component, including the Patch Histogram, Local Student, and Attention-Enhanced Global Student, produces interpretable outputs that enable clearer understanding and visual traceability of anomaly decisions. This contribution advances the field of visual anomaly detection by demonstrating that logical reasoning, compositional regularity, and attention-based feature learning can be effectively combined to detect errors that go beyond surface

texture. AeCSAD represents a step toward making unsupervised industrial inspection systems more reliable, interpretable, and adaptable to complex real-world scenarios.

6.0.1 Key Findings

The development and evaluation of the proposed anomaly detection framework yielded several key findings that contribute both practical and conceptual insights into visual anomaly detection in industrial contexts. First, the integration of independently trained scoring branches, namely the Patch Histogram, Local Student, and Global Student, demonstrated a robust and modular architectural design. Each branch contributed complementary anomaly cues: the Patch Histogram captured deviations in component-level spatial distributions, the Local Student focused on fine-grained appearance irregularities, and the Global Student modeled high-level contextual inconsistencies. The subsequent fusion of these signals enabled reliable decision-making while maintaining interpretability and modular traceability. This result underscores the effectiveness of decoupling anomaly detection into specialized processing pathways, as opposed to relying on a single end-to-end model to learn all patterns jointly.

Second, the integration of self-attention mechanisms into the autoencoder-based global student network resulted in improved performance on the logical anomaly subset of the dataset. The self-attention layers proved particularly effective in capturing long-range contextual dependencies, enabling the network to reason about object relationships and semantic consistency beyond localized pixel-level cues. This architectural enhancement strengthened the model's capacity to detect subtle yet semantically significant anomalies, especially those that do not present clear low-level visual deviations. These findings highlight a broader design principle in industrial anomaly detection: effective systems are not solely those that classify anomalies, but those that are structured to reason in a modular, interpretable manner that aligns with the compositional and semantic structure of real-world manufacturing environments.

6.1 Limitations

Despite the strengths of the proposed framework, several limitations remain that highlight areas for further improvement and refinement. These limitations are not only technical but also conceptual, reflecting the boundaries of current approaches to unsupervised anomaly detection in real-world industrial scenarios. First, the system's performance depends heavily on the quality of semantic pseudo-labels generated in the early stages. Although foundation models such as SAM and GroundingDINO offer impressive zero-shot capabilities, their behavior in industrial contexts is often brittle. Visual clutter, occlusion, poor lighting, and domain shift can lead to segmentation masks that are either fragmented or semantically incorrect. These inaccuracies propagate into the clustering and training stages, introducing noise into the segmentation network and ultimately affecting anomaly scoring. While filtering strategies help reduce the impact of poor masks, the system remains vulnerable to errors introduced during pseudo-label generation.

Second, the Patch Histogram module, though effective in modeling quantity-based or compositional anomalies, is sensitive to spatial bin granularity. Coarse binning may obscure meaningful deviations in layout, while overly fine binning can lead to sparsity and unstable statistical representations. There is also no built-in mechanism to account for rotational or positional invariance, which may lead to false positives in categories where object arrangement is naturally variable. The histogram-based approach relies on the assumption that correct part configurations are spatially consistent across training examples that may not hold in complex assemblies or multi-view inspection setups. Third, while self-attention improves the global student's ability to reason over part relationships, its effectiveness is contingent on exposure to sufficient variation during training. Infrequent logical configurations or rare component combinations may be incorrectly flagged as anomalous due to lack of representation. This reflects a broader tension between learning context-dependent norms and generalizing to exceptions. Additionally, attention-based models, while interpretable to some extent, are still susceptible to dis-

tributed or diffuse attention that is difficult to localize or explain definitively. Fourth, the scoring fusion mechanism, though robust, uses fixed weights across categories and does not dynamically adapt to category-specific anomaly structures. In categories dominated by texture defects, local scoring may be sufficient, while in logic-heavy categories, the global or histogram branches carry more importance. A learned or adaptive fusion mechanism might yield better category-specific performance, but was not explored in this work due to supervision constraints.

Fifth, the model is trained entirely on RGB (colored) images, and we explicitly assume that input images are colored during both training and inference. Many of the anomalies, particularly logical inconsistencies, depend on subtle visual cues such as color, shading, and texture contrast. These cues play a critical role in enabling the segmentation network and student–teacher modules to recognize abnormal patterns. If these cues were removed, as in grayscale images, the model's ability to detect anomalies would be significantly impaired. Therefore, grayscale imaging is not supported in the current framework and would require retraining or a redesign of feature extractors that are sensitive to intensity-based representations.

Finally, the system lacks an explicit mechanism for uncertainty estimation or self-assessment. It does not provide a confidence measure or flag instances where the output may be unreliable. While anomaly scores offer a proxy for deviation severity, they do not guarantee that the system can detect its own failure modes. Incorporating model uncertainty, calibration techniques, or abstention mechanisms remains an important direction for future work.

6.2 Future Work

While the proposed framework demonstrates effectiveness in addressing both structural and logical anomaly detection, it also opens several avenues for future research. These

directions aim to improve robustness, generalization, semantic richness, and autonomy in anomaly detection systems operating in complex industrial domains.

6.2.1 Smarter Pseudo-Label Filtering and Selection

The reliance on segmentation masks from foundation models remains a bottleneck in terms of reliability and control. Future work could explore learning-based or confidence-aware filtering mechanisms that assign quality scores to candidate masks before clustering. By introducing pseudo-label selection criteria based on intra-cluster consistency, objectness, or cross-model agreement (e.g., SAM vs. GroundingDINO, the system could better isolate high-confidence training samples and avoid noisy supervision.

6.2.2 Slot Attention or Part Graph Networks

While self-attention improves relational reasoning, it remains limited in terms of explicit representation of parts. Future architectures may benefit from more structured relational encodings, such as slot attention [118] or component-centric graph neural networks (GNNs) [119]. These models would allow parts to be explicitly identified, embedded, and linked through spatial or semantic relationships. Logical inconsistencies could then be detected by reasoning over graph topology or learned relational embeddings.

6.2.3 Category-Agnostic Generalization and Few-Shot Adaptation

While the current model performs well within known categories, generalizing to entirely new object types or categories with few normal examples remains an open challenge. Exploring few-shot adaptation strategies such as feature adaptation via contrastive learning, pseudo-label bootstrapping, or episodic training—could help extend this framework to zero or few-shot anomaly detection without requiring extensive retraining [120, 121].

6.2.4 Self-Supervised Segmentation Pretraining

Although the segmentation network is trained using pseudo-labels, its backbone remains frozen and initialized from ImageNet. Future work could explore self-supervised segmentation pretraining within the industrial domain itself. This might enable the model to better extract domain-relevant part-level features without relying entirely on pretrained visual representations [122, 123].

Bibliography

[1] Mark Muro, Robert Maxim, and Jacob Whiton. "America's advanced industries: New trends". In: *Brookings Institution Report* (2019). URL: https://www.brookings.edu/research/americas-advanced-industries-new-trends.

[2] Edward A Lee. "Cyber physical systems: Design challenges". In: *11th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)* (2015), pp. 363–369.

[3] Hyunwoo Jung, Seunghyun Yoon, et al. "Industrial anomaly detection: Benchmark dataset and method". In: *CVPR* (2021).

[4] Lukas Ruff et al. "A unifying review of deep and shallow anomaly detection". In: *Proceedings of the IEEE* 109.5 (2021), pp. 756–795.

[5] Paul Bergmann, Tim Meinhardt, and Tomas Pfister. "Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization". In: *International Journal of Computer Vision* 130 (2022), pp. 518–536. DOI: 10.1007/s11263-021-01531-0.

[6] Paul Bergmann et al. "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9592–9600. DOI: 10.1109/CVPR.2019.00982.

[7] Guansong Pang et al. "Deep learning for anomaly detection: A review". In: *ACM Computing Surveys (CSUR)* 54.2 (2021), pp. 1–38. DOI: 10.1145/3439950.

[8] Kilian Batzner, Lars Heckler, and Rebecca König. "EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2024, pp. 128–138. DOI: 10.1109/WACV56688.2024.00020.

[9] Zhikang Liu et al. "SimpleNet: A Simple Network for Image Anomaly Detection and Localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 20409–20418. DOI: 10.1109/CVPR52729.2023.01968.

[10] Yann LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[11] Wenjie Luo et al. "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016, pp. 4898–4906.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[13] Hoo-Chang Shin et al. "Convolutional neural networks: an overview and application in radiology". In: *Insights into Imaging* 9.4 (2018), pp. 611–629. DOI: 10.1007/s13244-018-0639-9.

[14] Karsten Roth et al. "Towards Total Recall in Industrial Anomaly Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 14298–14308. DOI: 10.1109/CVPR52688.2022.01392.

[15] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. "CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 98–107. DOI: 10.1109/WACV51458.2022.00188.

[16] Thomas Defard et al. "PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization". In: *Proceedings of the International Conference on Pattern Recognition (ICPR) Workshops*. 2020, pp. 475–489. DOI: 10.1007/978-3-030-68799-1_35.

[17] Jiawei Yu et al. "FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows". In: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*. 2021, pp. 2813–2822. DOI: 10.1145/3459637.3482232.

[18] Yifan Ma et al. "Towards Accurate Unified Anomaly Segmentation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2025, pp. 1234–1243. DOI: 10.1109/WACV56688.2025.00123.

[19] Haiming Yao and Wenyong Yu. "Generalizable Industrial Visual Anomaly Detection with Self-Induction Vision Transformer". In: *arXiv preprint arXiv:2211.12311* (2022).

[20] Xian Tao et al. "Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey". In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–21. DOI: 10.1109/TIM.2022.3192043.

[21] Hui Zhang et al. "DiffusionAD: Norm-Guided One-Step Denoising Diffusion for Anomaly Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025), pp. 1–14. DOI: 10.1109/TPAMI.2025.3570494.

[22] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[23] Haoyue Bai et al. "CounTR: An End-to-End Transformer Approach for Crowd Counting and Density Estimation". In: *Computer Vision – ECCV 2022 Workshops*. Vol. 13666. Lecture Notes in Computer Science. Springer, 2022, pp. 207–222. DOI: 10.1007/978-3-031-25075-0_16.

[24] Yu-Hsuan Hsieh and Shang-Hong Lai. "CSAD: Unsupervised Component Segmentation for Logical Anomaly Detection". In: *Proceedings of the 35th British Machine Vision Conference (BMVC)*. BMVA. 2024.

[25] Youcai Zhang et al. "Recognize Anything: A Strong Image Tagging Model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2024.

[26] Shilong Liu et al. "Marrying DINO with Grounded Pre-Training for Open-Set Object Detection". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer. 2024.

[27] Alexander Kirillov et al. "Segment Anything". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 4015–4026.

[28] Chong Zhou and Randy C Paffenroth. "Anomaly detection with robust deep autoencoders". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017), pp. 665–674.

[29] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[30] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. "Exploring self-attention for image recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10076–10085.

[31] H. Hotelling. "Analysis of a Complex of Statistical Variables into Principal Components". In: *Journal of Educational Psychology* 24.6 (1933), pp. 417–441.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[33] Prasanta Chandra Mahalanobis. "On the generalized distance in statistics". In: *Proceedings of the National Institute of Sciences of India* 2.1 (1936), pp. 49–55.

[34] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data Clustering: A Review". In: *ACM Computing Surveys (CSUR)* 31.3 (1999), pp. 264–323.

[35] Chao Huang, Zhao Kang, and Hong Wu. "A Prototype-Based Neural Network for Image Anomaly Detection and Localization". In: *Neural Processing Letters* 56 (2024), p. 169. DOI: 10.1007/s11063-024-11466-7.

[36] Oliver Rippel, Patrick Mertens, and Dorit Merhof. "Modeling the Distribution of Normal Data in Pre-Trained Deep Features for Anomaly Detection". In: *arXiv preprint arXiv:2005.14140* (2020).

[37] Garnik Vareldzhan, Kirill Yurkov, and Konstantin Ushenin. "Anomaly Detection in Image Datasets Using Convolutional Neural Networks, Center Loss, and Mahalanobis Distance". In: *arXiv preprint arXiv:2104.06193* (2021).

[38] Yi Yang et al. "MST: Multiscale Flow-Based Student–Teacher Network for Unsupervised Anomaly Detection". In: *Electronics* 13.16 (2024), p. 3224. DOI: 10.3390/electronics13163224.

[39] Soopil Kim et al. "Few shot part segmentation reveals compositional logic for industrial anomaly detection". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 6. 2024, pp. 8591–8599.

[40] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[41] Vic Barnett and Toby Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 1994.

[42] Victoria J. Hodge and Jim Austin. "A survey of outlier detection methodologies". In: *Artificial Intelligence Review* 22.2 (2004), pp. 85–126.

[43] Bernhard Schölkopf et al. "Estimating the Support of a High-Dimensional Distribution". In: *Neural Computation* 13.7 (2001), pp. 1443–1471.

[44] Jiaqi Liu et al. "Deep Industrial Image Anomaly Detection: A Survey". In: *Machine Intelligence Research* 21.1 (2024), pp. 104–135.

[45] Gilles Pitard et al. "Robust Anomaly Detection Using Reflectance Transformation Imaging for Surface Quality Inspection". In: *International Conference on Quality Control by Artificial Vision*. Springer. 2019, pp. 1–8.

[46] Matan Haroush et al. "A Statistical Framework for Efficient Out of Distribution Detection in Deep Neural Networks". In: *International Conference on Learning Representations (ICLR)*. 2022.

[47] Ronit Das and Tie Luo. "LightESD: Fully-Automated and Lightweight Anomaly Detection Framework for Edge Computing". In: *Electronics* 12.12 (2023), p. 1234.

[48] Saikiran Bulusu et al. "Anomalous Example Detection in Deep Learning: A Survey". In: *IEEE Access* 8 (2020), pp. 132330–132347.

[49] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[50] Edward W Forgy. "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". In: *Biometrics* 21.3 (1965), pp. 768–769.

[51] Kimin Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018, pp. 7167–7177.

[52] Niv Cohen and Yedid Hoshen. "Sub-Image Anomaly Detection with Deep Pyramid Correspondences". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4813–4822. DOI: 10.1109/CVPR42600.2020.00487.

[53] Thomas Defard et al. "PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization". In: *Proceedings of the International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 123–138.

[54] Chaoqin Huang et al. "Registration Based Few-Shot Anomaly Detection". In: *Computer Vision – ECCV 2022*. Vol. 13684. Lecture Notes in Computer Science. Springer, 2022, pp. 303–319. DOI: 10.1007/978-3-031-20053-3_18.

[55] Sergey Zagoruyko and Nikos Komodakis. "Wide Residual Networks". In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2016, pp. 87.1–87.12. DOI: 10.5244/C.30.87.

[56] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[57] Evelyn Fix and Joseph L. Hodges. "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties". In: *Project 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas* (1951).

[58] Michel Marie Deza and Elena Deza. *Encyclopedia of Distances*. Springer, 2009. ISBN: 978-3-642-00233-5.

[59] Tongkun Liu et al. "Component-aware anomaly detection framework for adjustable and logical industrial visual inspection". In: *Advanced Engineering Informatics* 58 (2023), p. 102161.

[60] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[61] Author Names. "A Survey on Anomaly Detection with Few-Shot Learning". In: *Proceedings of the International Conference on Machine Learning*. Springer, 2024.

[62] Jane Bromley et al. "Signature verification using a "Siamese" time delay neural network". In: *Advances in Neural Information Processing Systems*. Vol. 6. 1994, pp. 737–744.

[63] Max Jaderberg et al. "Spatial Transformer Networks". In: *Advances in Neural Information Processing Systems*. Vol. 28. 2015, pp. 2017–2025.

[64] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. "PNI: Industrial Anomaly Detection using Position and Neighborhood Information". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. 2023, pp. 6373–6383. DOI: 10.1109/ICCV51070.2023.00586. URL: https://openaccess.thecvf.com/content/ICCV2023/html/Bae_PNI_Industrial_Anomaly_Detection_using_Position_and_Neighborhood_Information_ICCV_2023_paper.html.

[65] Jeeho Hyun et al. "ReConPatch: Contrastive Patch Representation Learning for Industrial Anomaly Detection". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024, pp. 2052–2061. DOI: 10.1109/WACV56688.2024.00226.

[66] Pauline De Haan et al. "RegAD: Region-based anomaly detection in images with self-supervised learning". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 1560–1569.

[67] Shuai Lu et al. "PatchCL-AE: Anomaly Detection for Medical Images Using Patch-Wise Contrastive Learning-Based Auto-Encoder". In: *Computerized Medi-*

cal Imaging and Graphics 114 (2024), p. 102366. DOI: 10.1016/j.compmedimag.2024.102366.

[68] Hadi Hojjati, Thi Kieu Khanh Ho, and Narges Armanfard. "Self-supervised anomaly detection in computer vision and beyond: A survey and outlook". In: *Neural Networks* 172 (2024), p. 106106. DOI: 10.1016/j.neunet.2024.106106. URL: https://arxiv.org/abs/2205.05173.

[69] Declan McIntosh and Alexandra Branzan Albu. "Inter-Realization Channels: Unsupervised Anomaly Detection Beyond One-Class Classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 6285–6295. URL: https://openaccess.thecvf.com/content/ICCV2023/html/McIntosh_Inter-Realization_Channels_Unsupervised_Anomaly_Detection_Beyond_One-Class_Classification_ICCV_2023_paper.html.

[70] Paul Bergmann et al. "Uninformed Students: Student–Teacher Anomaly Detection with Discriminative Latent Embeddings". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4183–4192.

[71] Fuzhen Zhuang et al. "A Comprehensive Survey on Transfer Learning". In: *Proceedings of the IEEE* 109.1 (2021), pp. 43–76. DOI: 10.1109/JPROC.2020.3004555. URL: https://arxiv.org/abs/1911.02685.

[72] Chuan Li et al. "CutPaste: Self-Supervised Learning for Anomaly Detection and Localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 13761–13770. DOI: 10.1109/CVPR46437.2021.01356.

[73] Zhi-Hua Zhao et al. "Machine Learning: Algorithms, Real-World Applications and Research Directions". In: *SN Computer Science* 2.3 (2021), pp. 1–21.

[74] Darrell L. Hodson et al. "Mean Squared Error, Deconstructed". In: *AGU Advances* 2.3 (2021), e2021MS002681.

[75] M A Ganaie et al. "Ensemble deep learning: A review". In: *arXiv preprint arXiv:2104.02395* (2021).

[76] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.

[77] Seijin Kobayashi, Pau Vilimelis Aceituno, and Johannes von Oswald. "Disentangling the Predictive Variance of Deep Ensembles through the Neural Tangent Kernel". In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 1–12.

[78] Yansong Liu et al. "Selective ensemble method for anomaly detection based on parallel learning". In: *Scientific Reports* 14.1 (2024), p. 1420.

[79] Hanqiu Deng and Xingyu Li. "Anomaly Detection via Reverse Distillation from One-Class Embedding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 9737–9746. DOI: 10.1109/CVPR52688.2022.00951.

[80] Sanghyun Woo, Dahun Kim, and In So Kweon. "Student-Teacher Feature Pyramid Matching for Unsupervised Anomaly Detection". In: *European Conference on Computer Vision (ECCV)*. Springer, 2021, pp. 481–498.

[81] Zhaoqing Zhang et al. "Contextual Affinity Distillation for Image Anomaly Detection". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024, pp. 4530–4539.

[82] Xinyue Liu et al. "Dual-Modeling Decouple Distillation for Unsupervised Anomaly Detection". In: *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*. ACM. 2024, To appear. DOI: 10.1145/3664647.3681669.

[83] Zhihao Gu et al. "Remembering Normality: Memory-guided Knowledge Distillation for Unsupervised Anomaly Detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 16401–16409. DOI: 10.1109/ICCV51070.2023.01503. URL: https://openaccess.thecvf.com/content/ICCV2023/html/Gu_Remembering_Normality_Memory-guided_Knowledge_Distillation_for_Unsupervised_Anomaly_Detection_ICCV_2023_paper.html.

[84] Donghyeong Kim et al. "Separating Novel Features for Logical Anomaly Detection". In: *arXiv preprint arXiv:2407.17909* (2023).

[85] Ying Zhao et al. "LogicAL: Towards Logical Anomaly Synthesis for Unsupervised Anomaly Localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2024, pp. 4021–4030.

[86] Tran Dinh Tien et al. "Revisiting Reverse Distillation for Anomaly Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 24511–24520. DOI: 10.1109/CVPR52729.2023.02344. URL: https://openaccess.thecvf.com/content/CVPR2023/html/Tien_Revisiting_Reverse_Distillation_for_Anomaly_Detection_CVPR_2023_paper.html.

[87] Shenzhi Wang et al. "Glancing at the Patch: Anomaly Localization With Global and Local Feature Comparison". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 254–263. DOI: 10.1109/CVPR46437.2021.00030. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Glancing_at_the_Patch_Anomaly_Localization_With_Global_and_Local_CVPR_2021_paper.html.

[88] Iñigo Artola et al. "GLAD: A Global-to-Local Anomaly Detector". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 3337–3346.

[89] Rui Liu et al. "Progressive Reconstruction and Feature Fusion for Image Anomaly Detection". In: *Sensors* 23.21 (2023), p. 8750. DOI: 10.3390/s23218750.

[90] Mathilde Caron et al. "Emerging Properties in Self-Supervised Vision Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9650–9660. DOI: 10.1109/ICCV48922.2021.00953. URL: https://openaccess.thecvf.com/content/ICCV2021/html/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.html.

[91] Prasanna K Sahoo, Said Soltani, and Ahmed KC Wong. "A survey of thresholding techniques". In: *Computer Vision, Graphics, and Image Processing* 41.2 (1988), pp. 233–260. DOI: 10.1016/0734-189X(88)90022-9.

[92] Liang-Chieh Chen et al. "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2015. URL: https://arxiv.org/abs/1412.7062.

[93] Xiang Li et al. "PSAD: Few-Shot Industrial Anomaly Detection via Prototype-Enhanced Semantic Alignment". In: *CVPR*. 2024.

[94] Shir Amir et al. "On the Effectiveness of ViT Features as Local Semantic Descriptors". In: *Computer Vision–ECCV 2022 Workshops*. Springer, 2023, pp. 39–55. DOI: 10.1007/978-3-031-25069-9_3. URL: https://link.springer.com/chapter/10.1007/978-3-031-25069-9_3.

[95] Bingfeng Zhang, Jimin Xiao, and Terry Qin. "Self-Guided and Cross-Guided Learning for Few-Shot Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 8312–8321. DOI: 10.1109/CVPR46437.2021.00821. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Zhang_Self-Guided_and_Cross-Guided_Learning_for_Few-Shot_Segmentation_CVPR_2021_paper.html.

[96] Carole H. Sudre et al. "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA/ML-CDS)*. Vol. 10553. Lecture Notes in Computer Science. Springer, 2017, pp. 240–248. DOI: 10.1007/978-3-319-67558-9_28. URL: https://link.springer.com/chapter/10.1007/978-3-319-67558-9_28.

[97] Michael Yeung et al. "Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation". In: *Computers in Biology and Medicine* 140 (2022), p. 105061. DOI: 10.1016/j.compbiomed.2021.105061.

[98] Xiaowei Qian, Wen Lu, and Yifan Zhang. "Histogram Matching-Enhanced Adversarial Learning for Unsupervised Domain Adaptation in Cross-Modality Medical Image Segmentation". In: *Medical Physics* 51.12 (2025), pp. 8865–8881.

[99] Wei Feng et al. "Unsupervised Domain Adaptation for Medical Image Segmentation by Selective Entropy Constraints and Adaptive Semantic Alignment". In: *Medical Image Analysis* 84 (2023), p. 102695.

[100] Yaxing Zhao, Limsoon Wong, and Wilson Wen Bin Goh. "How to do quantile normalization correctly for gene expression data analyses". In: *Scientific Reports* 10.1 (2020), p. 15534.

[101] Daniel Skubleny et al. "Feature-specific quantile normalization and feature-specific mean–variance normalization deliver robust bi-directional classification and feature selection performance between microarray and RNAseq data". In: *BMC Bioinformatics* 25.1 (2024), p. 136.

[102] Ayush Jaiswal et al. "Class-agnostic Object Detection". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 1269–1278.

[103] Jia Guo et al. "Absolute-Unified Multi-Class Anomaly Detection via Class-Agnostic Distribution Alignment". In: *arXiv preprint arXiv:2404.00724* (2024).

[104] Maryam Qamar et al. "Can Segment Anything Model Improve Semantic Segmentation?" In: *International Conference on Big Data and Smart Computing (BigComp)*. 2023.

[105] Weixuan Sun et al. "An Alternative to WSSS? An Empirical Study of the Segment Anything Model (SAM) on Weakly-Supervised Semantic Segmentation Problems". In: *arXiv preprint arXiv:2305.01586* (2023).

[106] Wei Ji et al. "Segment Anything Is Not Always Perfect: An Investigation of SAM on Different Real-world Applications". In: *Machine Intelligence Research* 20.5 (2023), pp. 401–415.

[107] Yun Peng et al. "SAM-LAD: Segment Anything Model Meets Zero-Shot Logic Anomaly Detection". In: *Knowledge-Based Systems* 314 (2025), p. 113176.

[108] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 801–818.

[109] Xuan Zhang et al. "DeSTSeg: Segmentation Guided Denoising Student–Teacher for Anomaly Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, p. 21768.

[110] Marco Rudolph et al. "Asymmetric Student–Teacher Networks for Industrial Anomaly Detection". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 2591–2601.

[111] Yizong Cheng. "Mean Shift, Mode Seeking, and Clustering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.8 (1995), pp. 790–799. DOI: 10.1109/34.400568.

[112] Leland McInnes, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering". In: *Journal of Open Source Software* 2.11 (2017), p. 205. DOI: 10.21105/joss.00205. URL: https://doi.org/10.21105/joss.00205.

[113] Haohan Yang, Jian Fu, and Jiwen Lu. "Rethinking the Role of Spatial Information in Vision Models". In: *arXiv preprint arXiv:1809.02601* (2023).

[114] Bolei Zhou et al. "Interpretable Convolutional Neural Networks with Dual Local and Global Attention". In: *arXiv preprint arXiv:1807.03247* (2018).

[115] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. PMLR, 2015, pp. 448–456. URL: https://arxiv.org/abs/1502.03167.

[116] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML)*. Omnipress, 2010, pp. 807–814. URL: https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf.

[117] Zhikang Liu et al. "SimpleNet: A Simple Network for Image Anomaly Detection and Localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 20402–20411.

[118] F. Locatello et al. "Object-Centric Learning with Slot Attention". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020, pp. 11519–11530. URL: https://proceedings.neurips.cc/paper/2020/hash/8511df98c02ab60aea1b23Abstract.html.

[119] Quanshi Zhang et al. "Growing Interpretable Part Graphs on ConvNets via Multi-Shot Learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. 2017, pp. –. DOI: 10.1609/aaai.v31i1.10924.

[120] Duo Peng et al. "Harnessing Text-to-Image Diffusion Models for Category-Agnostic Pose Estimation". In: *European Conference on Computer Vision (ECCV)*. 2024.

[121] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)* (2017), pp. 1126–1135.

[122] Xiaohang Zhan et al. "Mix-and-Match Tuning for Self-Supervised Semantic Segmentation". In: *arXiv preprint arXiv:1712.00661*. 2017.

[123] Adrian Ziegler and Yuki M. Asano. "Self-Supervised Learning of Object Parts for Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 14502–14511.