



Benchmarking of Anomaly Detection Methods for Industry 4.0: Evaluation, Ranking, and Practical Recommendations

Aurélie Cools ^{1,*}, Mohammed Amin Belarbi ² and Sidi Ahmed Mahmoudi ¹

- ¹ Department of Computer Science, Software and Artificial Intelligence, Faculty of Engineering
- (Polytechnic Faculty), University of Mons (UMons), 7000 Mons, Belgium; sidi.mahmoudi@umons.ac.be
- ² Amintechs, 7000 Mons, Belgium
- * Correspondence: aurelie.cools@umons.ac.be

Abstract: Quality control and predictive maintenance are two essential pillars of Industry 4.0, aiming to optimize production, reduce operational costs, and enhance system reliability. Real-time visual inspection ensures early detection of manufacturing defects, assembly errors, or texture inconsistencies, preventing defective products from reaching customers. Predictive maintenance leverages sensor data by analyzing vibrations, temperature, and pressure signals to anticipate failures and avoid production downtime. Image-based quality control has become critical in industries such as automotive, electronics, aerospace, and food processing, where visual appearance is a key quality indicator. Although advances in deep learning and computer vision have significantly improved anomaly detection, industrial deployments remain challenged by the scarcity of labeled anomalies and the variability of defects. These issues increasingly lead to the adoption of unsupervised methods and generative approaches, which, despite their effectiveness, introduce substantial computational complexity. We conduct a unified comparison of ten anomaly detection methods, categorizing them according to their reliance on synthetic anomaly generation and their detection strategy, either reconstruction-based or feature-based. All models are trained exclusively on normal data to mirror realistic industrial conditions. Our evaluation framework combines performance metrics such as recall, precision, and their harmonic mean, emphasizing the need to minimize false negatives that could lead to critical production failures. In addition, we assess environmental impact and hardware complexity to better guide method selection. Practical recommendations are provided to balance robustness, operational feasibility, and sustainability in industrial applications.



Academic Editor: Moulay A. Akhloufi

Received: 31 March 2025 Revised: 29 April 2025 Accepted: 3 May 2025 Published: 13 May 2025

Citation: Cools, A.; Belarbi, M.A.; Mahmoudi, S.A. Benchmarking of Anomaly Detection Methods for Industry 4.0: Evaluation, Ranking, and Practical Recommendations. *Big Data Cogn. Comput.* 2025, *9*, 128. https:// doi.org/10.3390/bdcc9050128

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/ licenses/by/4.0/). Keywords: anomaly detection; computer vision; Industry 4.0

1. Introduction

Industry 4.0 relies on advanced automation, artificial intelligence, and the Internet of Things (IoT) to optimize production and industrial system maintenance. In this context, anomaly detection plays a crucial role in identifying manufacturing defects and malfunctions before they compromise product quality or operational continuity. Here, anomaly detection is understood not merely as a classification task, but rather as the identification of rare and critical deviations from expected patterns, emphasizing the detection of subtle irregularities rather than broad categorical assignments. As highlighted by Lee et al. [1], Cyber-Physical Systems (CPS) are central to this transformation, enabling real-time interaction between digital models and physical production systems. Smart manufacturing frameworks, as outlined by Lu and Weng [2], incorporate intelligent algorithms to monitor, analyze, and improve quality processes.

Image-based quality control is a key component of this digital evolution. By detecting and eliminating defective products before they reach the market, it helps companies avoid reputational risks associated with non-compliant products. A missed defect can lead to costly recalls, loss of customer trust, and potential regulatory penalties. Thanks to advances in computer vision and deep learning, visual inspection can now be automated with greater accuracy and speed. Unlike manual inspections, which are costly and prone to human error, AI-based systems detect subtle defects and provide full traceability in production. This approach is widely used in sectors such as electronics, automotive, and aerospace, where strict quality standards apply.

However, industrial anomaly detection poses several challenges. A major limitation is the rarity of defects, which creates severe class imbalances in datasets. Supervised deep learning methods become ineffective when abnormal examples are scarce, making unsupervised and self-supervised methods more suitable in industrial settings. As described by Ruff et al. [3], a wide spectrum of anomaly detection strategies exists—ranging from shallow statistical methods to deep neural networks—each offering different trade-offs between interpretability and performance.

Moreover, variability in imaging conditions (e.g., lighting, viewing angles, and production parameters) affects the appearance of normal samples, further complicating defect detection. In addition, data confidentiality and infrastructure constraints in industry often require local, on-premises processing, which limits access to powerful computing resources. This necessitates the use of lightweight and efficient models to ensure real-time inference without compromising production throughput. As Sultani et al. [4] pointed out in a different context, anomaly detection methods must remain reliable even in environments where anomalous events are rare and diverse.

Beyond manufacturing, similar computational and detection challenges are encountered in other critical sectors such as healthcare. In both industrial and medical applications, the constraints on energy consumption, latency, and data privacy demand the development of efficient and robust anomaly detection methods. In these contexts, minimizing false negatives—i.e., the ability to detect all critical anomalies—is particularly crucial to ensure safety, prevent severe failures, and protect human lives.

This study focuses specifically on industrial quality control through visual inspection, particularly in the context of manufacturing processes involving objects or materials. The evaluated methods are tested on representative data drawn from the MVTec AD dataset [5], which includes components commonly found in mechanical and materialbased production, such as metal parts, screws, bottles, or fabrics. In practical applications, this often translates to detecting misplaced screws on printed circuit boards (PCBs), incorrectly aligned electronic components, or surface defects such as scratches or contaminants that can jeopardize the functionality or reliability of manufactured products. This selection reflects practical use cases typical of Industry 4.0 environments, where precision in defect detection and computational efficiency are critical.

We conduct a comparative analysis of ten anomaly detection methods tailored for image-based quality control. The evaluation focuses on their ability to detect complex anomalies, their robustness to data variations, and their computational efficiency. The selected methods encompass a diverse set of state-of-the-art approaches, including reconstruction-based models, feature extraction methods, and techniques with or without synthetic anomaly generation. The selection was motivated by their recognition in recent literature, the availability of open-source implementations, and their relevance to industrial applications. All methods were evaluated on a dataset representative of manufacturing environments, ensuring that their performance reflects real-world industrial conditions. It is also important to note that all selected methods are designed to be trained exclusively on normal (defect-free) data, aligning with the industrial reality where anomalous examples are rare, unlabeled, or difficult to collect. The primary objective is to identify approaches that maximize detection performance while minimizing false positives, thereby avoiding unnecessary rejection of compliant products and reducing operational costs.

The remainder of this paper is organized as follows. Section 2 presents the categorization framework used to classify the anomaly detection methods. Section 3 details the experimental protocol, including the dataset, training procedures, and evaluation metrics. Section 4 discusses the comparative results across different object types and analyzes the environmental and computational impacts of the methods. Finally, Section 5 provides a general discussion of the findings, and Section 6 concludes the paper with practical recommendations and perspectives for future work.

2. Categorization of Anomaly Detection Methods

We operate under the assumption that no abnormal samples are accessible during training. This constraint forces models to rely solely on normal samples, significantly influencing the choice of approaches and their ability to generalize to previously unseen defects.

To better understand the differences between various anomaly detection approaches, we propose a categorization based on two main axes. The first axis distinguishes methods based on the use of synthetic anomalies: some approaches generate artificial anomalies to support training, while others rely exclusively on normal samples. The second axis focuses on the detection strategy: either based on image reconstruction or on the extraction of discriminative features. This dual categorization provides a structured framework to evaluate the strengths and limitations of each method with respect to the specific constraints of industrial quality control.

The following sections present this categorization in detail and introduce the ten evaluated methods, structured according to the two axes defined above.

2.1. Synthetic vs. Real Anomaly Generation

One of the main challenges in anomaly detection is the scarcity or even the absence of abnormal samples during training. Given the rarity or unavailability of genuine anomalies, some methods address this limitation by generating synthetic anomalies to enrich the training process. However, creating realistic and diverse anomalies remains a significant challenge, as industrial defects can vary widely, from microcracks to texture or color variations and assembly defects. This diversity makes it difficult to produce synthetic anomalies that accurately reflect the defects encountered in production. Moreover, overly simplistic anomaly generation techniques often fail to capture the full variability of realworld anomalies, which can reduce the effectiveness of these methods.

To address this issue, different anomaly generation approaches have been developed. Some methods overlay artificial textures onto the original image using randomly generated masks to simulate surface defects. Others adopt a copy-paste approach, where a region of the image is duplicated and repositioned, creating structural inconsistencies. These techniques help diversify synthetic anomalies and improve models' generalization capabilities, though the quality of the generated anomalies directly impacts real-world detection performance.

Some methods require artificial anomaly generation for training, while others rely solely on normal samples. Methods generating synthetic anomalies are particularly useful when real defects are rare or difficult to collect, but their effectiveness depends on the quality and representativeness of the synthetic data. In contrast, methods trained exclusively on normal data exploit statistical deviations or latent space representations to detect anomalies without requiring abnormal samples. These approaches are often preferred in environments where generating realistic anomalies is impractical or costly.

2.2. Reconstruction-Based vs. Feature-Based Approaches

Anomaly detection methods also differ in how they identify abnormal patterns. Some rely on image reconstruction and analyze discrepancies between the original and reconstructed versions, while others directly extract discriminative features to distinguish normal and abnormal samples.

Reconstruction-based approaches rely on the observation that models trained only on normal images reconstruct them accurately, but tend to fail in reconstructing images containing anomalies. The difference between the input and its reconstruction indicates the presence of a defect. This strategy is particularly effective for detecting subtle anomalies that are difficult to characterize with fixed descriptors. However, it can struggle when certain anomalies are reconstructed too faithfully, making them harder to detect.

Feature-based approaches, by contrast, learn feature representations where anomalies naturally deviate from normal samples. These methods generally provide faster inference times by avoiding the need for full image reconstruction. However, they may be sensitive to natural variations in the data that do not necessarily correspond to true anomalies. The choice between these two families of approaches depends both on the types of anomalies to detect and on the computational constraints of the target application.

2.3. Overview of Anomaly Detection Methods

This section briefly presents the ten studied methods, categorized according to the previously defined framework, and illustrated in Figure 1.



Figure 1. Categorization of anomaly detection methods according to anomaly generation and detection strategy.

5 of 21

2.3.1. Anomaly Generation and Feature-Based Approaches

GLASS [6] introduces synthetic anomalies by applying two complementary strategies: local transformations mix a texture image with the original image using a random mask to simulate surface defects, while global transformations inject Gaussian noise directly into the feature space to disrupt latent representations. A WideResNet50 [7] architecture is used to extract features from intermediate layers (Layer 2 and Layer 3), and an adapter network refines these features to enhance alignment between normal and abnormal samples. Finally, a discriminator network assigns an anomaly score that quantifies how much an image deviates from normality. In the original implementation, the test dataset was mistakenly used during model selection, introducing data leakage and bias. This practice can artificially inflate performance metrics by allowing indirect adaptation to test data. To correct this issue, our evaluation restricts model selection strictly to the training set to ensure a rigorous and unbiased assessment.

CutPaste [8] simulates synthetic anomalies by cutting a patch from an image and pasting it randomly elsewhere within the same image. The model is trained to classify whether an image has undergone such transformation. Several variants exist, such as CutPaste Scar, which generates linear cuts resembling scratches or cracks, and CutPaste 3-Way, combining multiple transformations to enhance anomaly diversity. In addition to image classification, CutPaste incorporates a Gaussian Density Estimator (GDE) that models the distribution of extracted features to better distinguish normal from abnormal images. The GDE estimates the likelihood of new feature vectors belonging to the normal distribution, strengthening anomaly detection alongside the CutPaste transformations. Since the official CutPaste code was unavailable, an unofficial reproduction was used, with validation indicating similar or improved performance relative to the original work.

2.3.2. Approaches Without Anomaly Generation and Feature-Based Detection

PatchCore [9] extracts local features from normal images using a WideResNet50 backbone and stores a compact memory bank by selecting a representative subset of features (e.g., 10%). During inference, a test sample is compared to the nearest memory features, and an anomaly score is computed based on the minimum distance. This strategy minimizes memory consumption and accelerates inference while preserving detection accuracy.

PatchSVDD [10] segments images into patches and embeds them into a compact feature space through self-supervised contrastive learning. A Support Vector Data Description (SVDD) model then encloses normal patch embeddings within a minimum-volume hypersphere. Anomalies are detected as patches that lie outside this hypersphere. Additionally, a pretext task requiring the prediction of the relative positions of patch pairs enhances the quality and structure of the learned features, improving anomaly localization.

MSPBA [11] extends PatchSVDD by applying multi-scale analysis. It constructs K-Means clusters across patch features at three different resolutions rather than using a single hypersphere. This approach captures variations across different spatial scales and improves detection robustness in the presence of complex textures and defect patterns, as typically encountered in industrial scenarios.

2.3.3. Approaches with Anomaly Generation and Reconstruction-Based Detection

DRAEM [12] combines an autoencoder trained for normal image reconstruction with a segmentation network trained to detect discrepancies. Synthetic anomalies are introduced during training by applying external textures and color perturbations using random masks. The autoencoder reconstructs the unaltered image, while the segmentation module learns to identify regions that differ between the original and reconstructed images, producing detailed anomaly maps.

DiffusionAD [13] relies on a probabilistic diffusion model, where Gaussian noise is progressively added to normal images during training, and the model learns to reverse this process to restore the original content. Synthetic anomalies are injected during training through random masking and external texture overlays. At inference, the model reconstructs a defect-free version of the input image, and a segmentation network identifies discrepancies, highlighting anomalies even in cases where reconstruction alone would be insufficient.

2.3.4. Approaches Without Anomaly Generation and Reconstruction-Based Detection

RIAD [14] addresses anomaly detection through inpainting-based reconstruction. During training, random masks at three scales are applied to normal images, and the model learns to predict and restore the missing regions using a UNet-based [15] encoder-decoder architecture. A similarity loss ensures that reconstructed regions preserve fine structural details. At inference, anomalies are identified by analyzing pixel-wise differences between the input and its reconstructed version.

DDAD [16] employs a denoising diffusion model, guiding the reconstruction process by conditioning it on the original input image. Anomalies are detected by calculating discrepancies both at the pixel level and within an adapted feature space extracted by a pretrained network. An unsupervised domain adaptation step refines the feature extractor to better match the characteristics of industrial images generated during training, improving detection precision on unseen data.

Finally, Dinomaly [17] adopts a hybrid approach based on feature consistency. The model compares latent representations extracted by an encoder (based on DinoV2 [18]) and a decoder. For normal images, feature transformations are expected to remain consistent between the two stages. Discrepancies between encoder and decoder features serve as anomaly indicators, without requiring pixel-level reconstruction. This mechanism reduces the risk of anomalies being reconstructed too faithfully and improves detection robustness in complex settings.

2.4. Overview of Anomaly Detection Methods

While recent anomaly detection methods have achieved impressive performance improvements—sometimes reaching near-saturation on standard benchmarks—it is important to note that these performances are increasingly difficult to surpass. In particular, most existing works focus primarily on maximizing global metrics such as the AUROC. However, in industrial quality control applications, the primary concern is to minimize missed anomalies (false negatives), as undetected defects can have severe operational and financial consequences. Few methods explicitly address this need, suggesting that a shift toward evaluation strategies prioritizing anomaly recall over purely global optimization is necessary for realistic deployments.

Furthermore, two important methodological aspects deserve particular attention. First, methods that generate synthetic anomalies during training can enrich the diversity of defect types encountered and potentially improve generalization. However, they risk introducing unrealistic artifacts that do not accurately represent real-world defects, potentially biasing the model toward detecting "artificial" anomalies. On the other hand, methods that avoid synthetic anomalies better align with industrial constraints, where genuine anomalies are rare and diverse, but may suffer from limited exposure to defect variability during training.

Second, reconstruction-based approaches offer the advantage of detecting subtle, localized deviations from normality, making them effective for complex or small-scale defects. Nevertheless, these methods can inadvertently reconstruct anomalies too faithfully, reducing the difference between normal and abnormal images and complicating detection. Feature-based approaches circumvent this risk by learning directly discriminative representations but may be more sensitive to natural variations within the normal class. The choice between these strategies must therefore be guided by the specific operational priorities, such as defect criticality, data variability, and computational constraints.

3. Methodology

3.1. Materials

All models were trained under identical hardware conditions to ensure a fair and consistent performance comparison. Experiments were conducted on an HP OMEN computer equipped with an Intel Core i9-11900K processor (11th Gen, 3.50 GHz, 16 threads), an NVIDIA RTX 3090 GPU with 24 GB of VRAM, 64 GB of RAM, and 64 GB of swap memory.

3.2. Data and Parameters

The evaluation of the ten anomaly detection methods was performed using the MVTec Anomaly Detection dataset [5], a widely recognized benchmark for industrial anomaly detection. It comprises 15 categories, including 10 object classes (e.g., bottle, cable, capsule, metal) and 5 texture classes (e.g., wood, fabric, leather).

The training set contains only normal samples, while the test set includes both normal and anomalous images, as illustrated in Figure 2. However, the test set is imbalanced, with normal images representing only 25% to 30% of the samples depending on the class. To ensure a more balanced and fair evaluation, we restructured the test set by transferring a portion of normal images from the training set to the test set.



Figure 2. Illustration of the MVTec dataset.

Although this adjustment reduced the amount of training data, it enabled a more representative evaluation by restoring a better balance between normal and anomalous samples. Importantly, this rebalancing does not introduce bias: all normal images adhere to the same quality standards and are visually and functionally interchangeable. Thus, the selection of normal images for reallocation has no significant impact on the results.

All models were trained exclusively on normal images, following an unsupervised approach designed to learn the distribution of normal data and detect deviations. For methods requiring synthetic anomaly generation, the transformations described in their original publications were faithfully applied.

Additionally, we observed that some methods originally used test images during training, particularly for model selection based on test performance. To ensure proper experimental rigor, we corrected these practices by excluding all test images from training

and validation phases. Final model evaluation was conducted exclusively on unseen test data.

Regarding hyperparameters, we adhered as closely as possible to the configurations recommended in the original publications to maintain consistency. However, in cases where the memory requirements exceeded the capabilities of the RTX 3090 GPU (24 GB of VRAM), we adapted the settings by first reducing the batch size. If necessary, we further decreased the input image resolution, maintaining a minimum of 256×256 pixels to preserve evaluation fidelity.

3.3. Metrics Evaluation

To comprehensively assess the anomaly detection methods, we categorize the evaluation metrics into three groups: Performance Metrics, Environmental Metrics, and Hardware Complexity Metrics.

3.3.1. Performance Metrics

These metrics measure the accuracy and robustness of the models based on standard classification and segmentation criteria.

- Area Under the ROC Curve (AUC Image): This metric measures the model's ability to distinguish between normal and anomalous images, corresponding to the area under the ROC (Receiver Operating Characteristic) curve, which plots the true positive rate (TPR) against the false positive rate (FPR). A value closer to 1.0 indicates better performance [19].
- Recall: Recall evaluates the proportion of correctly detected anomalies among all actual anomalies, defined as:

$$\operatorname{Recall} = \frac{TP}{TP + FN} \tag{1}$$

where *TP* denotes true positives and *FN* denotes false negatives. A high recall reflects effective anomaly detection but may also increase false positives [20].

• F1-Score: The F1-Score provides a harmonic mean of precision and recall, offering a balanced view of detection performance, particularly for imbalanced datasets. It is defined as:

$$F1-Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
(2)

Precision measures the proportion of true anomalies among detected anomalies, and recall measures the proportion of detected anomalies among all actual anomalies [21].

 Average Precision (AP): Average Precision evaluates the model's average precision across all possible thresholds, computed as:

$$AP = \sum_{n} (R_n - R_{n-1})P_n \tag{3}$$

where P_n and R_n are the precision and recall at threshold n. This metric is widely used in object detection and anomaly detection tasks [22].

3.3.2. Environmental Metrics

Environmental metrics assess the ecological footprint associated with model training and inference. Specifically, we monitor:

 Energy Consumption and CO₂ Emissions: Energy usage is recorded throughout the training phase based on GPU power draw and total runtime. CO₂ emissions are estimated using a standard emission factor of 0.475 kg CO₂/kWh, providing an approximate measure of the environmental impact. Reducing energy and emissions is critical for sustainable deployment at industrial scales.

3.3.3. Hardware Complexity Metrics

Hardware metrics quantify the computational demands of the models, focusing on the following aspects:

- Inference Time: The time required for a model to process a single image or batch. Low inference times are crucial for real-time applications in industrial environments.
- GPU Memory Usage: The amount of VRAM consumed during training and inference. Models with high memory usage may face deployment constraints, particularly on resource-limited hardware or edge devices.
- Giga Multiply-Accumulate Operations (GMAC): GMAC quantifies the number of billions of operations (multiplications and additions) required during a single forward pass through the network. Higher GMAC values imply greater computational demands, which can impact execution time, energy efficiency, and hardware requirements. Thus, reducing computational complexity while maintaining detection performance remains a key optimization goal.

4. Results

This section presents the evaluation results of the ten anomaly detection methods under different conditions. First, the global performance is analyzed, followed by a detailed examination across three specific categories: fixed objects, rotating objects, and textures. Finally, the environmental impact and hardware complexity associated with each method are discussed.

4.1. Global Comparison of Anomaly Detection Methods

Table 1 summarizes the global performance of all evaluated methods across four key metrics: AUC, Recall, F1-Score, and Average Precision (AP). The ranking is based on the average results across the 15 classes of the MVTec AD dataset. A detailed breakdown of the results per class is provided in Appendix A.

Table 1. Global ranking of anomaly detection methods based on AUC, Recall, F1-Score, and Average Precision metrics (averaged over 15 classes).

Method	AU	JC	Rec	call	F1-S	core	AP I	nage
	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Dinomaly	99.61	1	97.56	1	98.27	1	99.65	1
PatchCore	98.89	2	96.84	2	97.10	2	99.08	2
GLASS	97.65	3	92.95	4	93.90	3	97.83	3
DDAD	96.79	4	91.84	5	89.74	6	96.79	4
DRAEM	95.88	5	94.00	3	93.63	4	95.59	6
DiffusionAD	95.29	6	90.71	6	92.16	5	95.99	5
MSPBA	92.45	7	86.20	7	87.00	7	93.24	7
CutPaste	91.72	8	83.64	8	86.83	8	92.64	8
RIAD	81.71	9	75.96	9	77.47	9	82.07	9
PatchSVDD	77.98	10	76.68	10	75.87	10	78.11	10

The results indicate that Dinomaly outperforms all other evaluated methods across every metric, demonstrating excellent robustness and detection accuracy. PatchCore follows closely, offering competitive performance with lower computational demands. GLASS achieves strong precision but a slightly lower recall, suggesting a tendency to miss certain anomalies. Conversely, methods such as RIAD and PatchSVDD exhibit significantly lower performance, making them less suitable for demanding industrial quality control tasks.

4.2. Performance on Fixed Objects

Table 2 presents the performance of the anomaly detection methods specifically on fixed objects, evaluated using the same four metrics: AUC, Recall, F1-Score, and Average Precision (AP).

Method	AU	JC	Rec	all	F1-S	core	AP I	mage
	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Dinomaly	99.46	1	97.25	1	97.72	1	99.52	1
PatchCore	98.48	2	95.34	3	95.97	2	98.69	2
DDAD	96.68	3	94.73	2	86.30	6	97.06	4
GLASS	96.35	4	90.17	5	90.99	3	96.63	3
DiffusionAD	93.14	5	86.68	6	88.42	7	94.32	7
MSPBA	93.13	6	85.79	7	88.14	5	94.26	6
DRAEM	92.54	7	90.65	4	90.89	4	91.77	5
CutPaste	89.15	8	79.25	8	83.45	8	90.05	8
RIAD	83.17	9	79.93	9	79.83	9	83.77	9
PatchSVDD	83.59	10	78.66	10	79.84	10	83.49	10

Table 2. Performance of anomaly detection methods on fixed objects.

Consistent with the global results, Dinomaly outperforms all other methods for fixed objects across all evaluated metrics. PatchCore again ranks second, confirming its robustness on static components. DDAD shows strong AUC and Recall values but a significant drop in F1-Score, indicating a lower precision. GLASS remains competitive for distinguishing anomalies but continues to exhibit slightly lower recall compared to Dinomaly and PatchCore. Lower-tier methods, such as RIAD and PatchSVDD, continue to show limited performance, suggesting poor generalization on fixed object categories.

4.3. Performance on Rotating Objects

Table 3 summarizes the performance of the evaluated methods on rotating objects, according to AUC, Recall, F1-Score, and Average Precision (AP).

Method	AU	JC	Re	call	F1-S	core	AP I	mage
	Value	Rank	Value	Rank	Value	Rank	Value	Rank
PatchCore	99.49	1	98.52	1	98.21	1	99.49	2
Dinomaly	99.38	2	96.64	2	98.09	2	99.50	1
DRAEM	99.05	3	95.56	3	95.73	3	99.10	3
GLASS	97.18	4	92.20	4	93.30	4	97.44	4
DDAD	93.96	5	84.33	8	88.34	6	94.52	5
DiffusionAD	92.93	6	86.56	5	89.55	5	93.65	6
MSPBA	90.37	7	86.49	6	85.32	7	90.97	8
CutPaste	88.86	8	78.34	9	83.41	8	91.21	7
RIAD	73.93	9	62.63	10	68.49	10	74.43	9
PatchSVDD	63.02	10	85.60	7	74.19	9	62.75	10

Table 3. Performance of anomaly detection methods on rotating objects.

PatchCore achieves the best performance on rotating objects, slightly outperforming Dinomaly in terms of recall and F1-Score. Both methods maintain excellent results across all evaluated metrics, confirming their robustness to rotational transformations. DRAEM and GLASS also deliver strong performance, particularly in AUC and AP. In contrast, methods

11 of 21

such as RIAD and PatchSVDD exhibit significant drops in AUC and F1-Score, highlighting their vulnerability to rotational variations.

4.4. Performance on Textures

Table 4 presents the evaluation results of the anomaly detection methods specifically on texture categories.

Method	AU	JC	Ree	call	F1-S	core	AP I	mage
	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Dinomaly	99.94	1	98.56	2	99.15	1	99.94	1
GLASS	99.74	2	97.30	4	98.32	3	99.75	2
DiffusionAD	99.70	3	98.85	1	98.98	2	99.74	3
PatchCore	99.09	4	97.93	3	98.02	4	99.38	4
DRAEM	98.65	5	97.76	5	96.20	5	98.84	5
DDAD	98.65	6	92.31	7	95.41	6	98.82	6
MSPBA	92.74	7	86.58	8	86.42	8	93.18	7
CutPaste	97.04	8	92.95	6	93.62	7	97.13	8
RIAD	84.33	9	78.39	9	79.56	9	84.27	9
PatchSVDD	79.11	10	68.56 10		71.32	10	79.78	10

Table 4. Performance of anomaly detection methods on textures.

Dinomaly achieves the highest performance on texture data, leading in AUC, F1-Score, and AP Image, while closely following DiffusionAD in recall. GLASS and PatchCore also exhibit strong results, with GLASS being particularly competitive in precision-oriented metrics. In contrast, RIAD and PatchSVDD consistently remain the least effective methods across all evaluated metrics.

4.5. Environmental Impact

Table 5 presents the environmental impact of each anomaly detection method. The metrics include total CO₂ emissions (in kilograms), total energy consumption (in kilojoules), and average GPU power consumption (in watts) during training.

Table 5. Environmental impact of anomaly detection methods based on CO_2 emissions, total energy consumption, and GPU power usage.

Method	CO ₂ Emissions (kg)	Total Energy (kJ)	GPU Power (W)
DDAD	0.822	6400.66	326.81
RIAD	0.066	501.61	304.32
Dinomaly	0.121	917.66	282.01
PatchCore	0.030	233.46	71.21
PatchSVDD	0.206	1563.66	60.81
MSPBA	0.503	3811.89	128.07
GLASS	0.280	2378.25	195.53
CutPaste	0.010	276.30	149.73
DiffusionAD	2.190	16,772.40	335.51
DRAEM	0.354	2682.33	287.28

The results show that feature extraction–based methods, such as PatchCore and Cut-Paste, are the most energy-efficient, with the lowest CO₂ emissions and energy consumption. In contrast, models incorporating generative reconstruction processes, such as DiffusionAD and DRAEM, exhibit significantly higher environmental costs, driven by their increased computational complexity and GPU usage during training. Another important observation is that models relying on synthetic anomaly generation (e.g., DiffusionAD, GLASS) tend to consume more energy than models trained without it. Although synthetic anomalies can improve accuracy in some contexts, they considerably raise the computational load and thus the environmental impact. For example, DiffusionAD produces 2.19 kg of CO_2 emissions—equivalent to the carbon footprint of a 15–20 km car journey using a conventional vehicle or the electricity consumption of a European household over approximately 12–15 h. While these values may seem modest in isolation, they scale significantly when applied across large datasets or repeated training cycles in industrial settings.

Overall, feature extraction–based methods without synthetic anomaly generation emerge as the most sustainable and energy-efficient choice for industrial deployment.

4.6. Hardware Complexity and Computational Performance

Table 6 details the hardware requirements and computational complexity for each anomaly detection method. The metrics include training time (in seconds), inference time per image (in seconds), average GPU utilization during training (in %), model size (in megabytes), and GMAC (Giga Multiply-Accumulate Operations).

Method	Training Time (s)	Inference Time (s)	GPU Utilization (%)	Model Size (MB)	GMAC
DDAD	19,067	0.1271	96.83	32.95	69.87
RIAD	992	0.4372	43.14	28.80	77.20
Dinomaly	2441	0.0296	45.85	148.00	136.64
PatchCore	135	0.7760	11.27	68.88	9.24
PatchSVDD	22,791	1.0065	7.96	0.45	0.12
MSPBA	28,627	2.1366	17.27	46.62	8.32
GLASS	10,912	0.3474	20.78	70.46	7.615
CutPaste	250	0.0634	20.17	11.77	2.38
DiffusionAD	49,482	0.0974	97.28	32.95	69.87
DRAEM	9137	0.3279	53.76	97.42	198.44

Table 6. Hardware complexity and computational performance of anomaly detection methods.

Feature extraction–based methods, such as PatchCore and CutPaste, show the best computational efficiency, with low training and inference times, minimal GPU utilization, and low GMAC values. In contrast, generative reconstruction–based methods, such as DiffusionAD, DDAD, and DRAEM, present significantly higher training costs and computational demands.

In several cases, the batch size had to be reduced to fit within available GPU memory, particularly for methods exceeding 90% GPU utilization, which typically require at least 24 GB of VRAM. While such methods remain feasible for deployment on high-performance industrial hardware, they are generally unsuitable for lightweight edge devices where resource constraints are stricter.

Dinomaly offers a strong compromise between speed, resource usage, and detection performance, making it particularly suitable for real-time deployment on mid-range industrial hardware.

Ultimately, model selection must balance detection performance with hardware constraints and energy considerations to ensure efficient and sustainable deployment in Industry 4.0 environments.

4.7. Visual Results

Figure 3 provides qualitative examples of the anomaly detection results on the *bottle* class from the MVTec AD dataset. For each example, the original input image, the ground truth mask, and the predictions from various methods are shown.

Image				\bigcirc	
Mask	•	,			1
DiffusionAD	\$	Contraction of the			A. C.
Dinomaly		Ĺ			
DDAD	•				
DRAEM	*	1	Ó	٥	Õ
Glass	0	1	្	Ó	
MSPBA	•				
PatchSVDD	0	1	Ô		0
PatchCore	•	1	\mathbf{O}	۲	0
RIAD	<		•	•	

Figure 3. Visual examples of anomaly detection results on the *bottle* class.

Visual inspection reveals that while most methods succeed in identifying anomalous regions, the precision of segmentation varies significantly. Dinomaly and PatchCore produce effective anomaly detection results but often display coarse and poorly delineated predictions. DRAEM, on the other hand, provides sharper segmentation with better localization of defects.

It is important to note that the primary objective of this study is to detect the presence of anomalies rather than to achieve precise pixel-level segmentation. Consequently, these visual examples are presented mainly for illustrative purposes, highlighting qualitative differences between the evaluated methods.

4.8. Summary of Results

The overall evaluation highlights several key findings:

- Dinomaly consistently achieves the best performance across all metrics and object categories, combining high detection rates with moderate computational cost.
- PatchCore offers an excellent balance between detection performance and efficiency, especially suitable for resource-constrained deployments.
- GLASS and DRAEM deliver strong performances, particularly when synthetic anomaly generation is acceptable to enhance training diversity.
- Generative methods such as DiffusionAD and DDAD show acceptable detection rates but incur significant computational and environmental costs.
- PatchSVDD and RIAD consistently demonstrate lower detection performance, limiting their suitability for industrial quality control applications.

These results also emphasize that feature extraction–based methods without anomaly generation tend to be more environmentally sustainable and computationally efficient, making them attractive options for large-scale and real-time industrial deployments.

5. Discussion

The results presented in the previous section reveal clear differences in the behavior, performance, and computational cost of the evaluated anomaly detection methods. This section provides a deeper interpretation of these findings, analyzing the strengths and limitations of each approach in the context of industrial quality control requirements, and proposes recommendations based on specific operational constraints.

5.1. General Observations

Dinomaly and PatchCore consistently achieve the best detection performance without relying on synthetic anomaly generation. This likely stems from their ability to learn the normal distribution directly, without introducing biases caused by artificial defects that may not accurately reflect real-world anomalies.

From an environmental perspective, feature-based methods without synthetic anomaly generation demonstrate significantly lower resource consumption. This efficiency is particularly evident in terms of GPU memory utilization, energy consumption, and CO₂ emissions.

Inference times, reported without specific code optimizations, vary considerably across methods. Consequently, absolute values should be considered indicative rather than definitive.

5.2. Impact of Architecture on GPU Memory Consumption

An important technical observation concerns the substantial GPU memory consumption observed in diffusion-based models, notably DiffusionAD and DDAD. This high resource requirement can be attributed to specific architectural choices:

Storage of intermediate noise states during the forward and reverse diffusion processes.

- Unrolling of sequential denoising operations, each of which must retain activation maps for gradient computation.
- Use of U-Net–based backbones with extensive skip connections, increasing memory usage by storing high-resolution feature maps.

Similarly, methods like DRAEM exhibit elevated memory usage, primarily due to their dual-branch architecture combining an autoencoder for image reconstruction and a segmentation network operating in parallel.

Thus, the combination of sequential operations (in diffusion models) and dual-branch architectures (in DRAEM) inherently leads to higher VRAM usage, making these models less suitable for deployment in constrained environments without further optimization.

5.3. Challenges in Data and Evaluation

A critical observation concerns the presence of subtle or unlabeled defects in some images labeled as "normal" in the MVTec AD dataset. As shown in Figure 4, models such as Dinomaly correctly identify these hidden defects, leading to apparent false positives during evaluation. This highlights the need for more rigorous dataset annotation or robust methods capable of handling label noise.



(a)



(b)





Figure 4. Examples of normal images misclassified as anomalies due to subtle or unlabeled visual artifacts: (a) Bottle. (b) Capsule. (c) Screw. (d) Transistor. Defects are highlighted in red.

Another major limitation observed is the reliance on post-hoc threshold selection based on test set performance. Most methods optimize thresholds after observing the evaluation results, introducing what we term the "detection bias curse". This practice artificially inflates model performance and undermines the true generalization ability. Developing threshold calibration methods independent of test data is essential for real-world deployment.

5.4. Recommendations and Future Directions

Several avenues for improvement emerge:

- Threshold Calibration: Future methods should integrate threshold selection mechanisms independent of the evaluation set to eliminate bias and improve deployment reliability.
- Segmentation Precision: Although detection remains the primary objective, enhancing anomaly localization would greatly benefit quality control applications requiring fine-grained defect analysis.
- Dataset Refinement: Improved annotation quality and the development of new benchmark datasets with fewer labeling inconsistencies are crucial for fairer evaluations.
- Synthetic Anomaly Generation: Current synthetic defect generation techniques often
 produce overly simplistic artifacts. Exploring advanced generative models, such as
 multimodal diffusion architectures or augmented reality-based defect simulation,
 could offer more realistic and diverse training data, thereby improving generalization.

Addressing these challenges is key to advancing the robustness, sustainability, and industrial applicability of anomaly detection systems.

6. Conclusions

This study presented a comprehensive benchmarking of ten anomaly detection methods applied to the MVTec Anomaly Detection dataset, with a focus on their application to quality control in Industry 4.0 environments. The methods were categorized along two axes: the use (or not) of synthetic anomaly generation, and the reliance on reconstruction-based versus feature-based detection strategies.

Dinomaly and PatchCore consistently achieved top performance without the need for synthetic anomaly generation, confirming their robustness and adaptability across diverse settings. PatchCore, in particular, stands out for its computational efficiency, while Dinomaly combines high detection accuracy with low inference latency, making both approaches highly suitable for real-world industrial deployment.

Diffusion-based models, although promising in detection accuracy, remain computationally intensive, particularly in terms of memory consumption. As highlighted in Section 5, their iterative reconstruction processes contribute to high resource demands, posing challenges for deployment in constrained industrial environments.

The comparison of environmental and hardware performance revealed that featurebased methods without synthetic anomaly generation are generally more energy-efficient, offering significant advantages in sustainable industrial applications.

A critical limitation identified is the reliance on post-hoc threshold calibration based on test set performance, which introduces evaluation bias. Future work must focus on developing autonomous, test-independent threshold selection strategies to ensure more realistic and deployable solutions.

Moreover, while anomaly detection performance is high, segmentation precision remains insufficient for applications requiring fine-grained defect localization. Improving segmentation capabilities is a key avenue for enhancing model usability in quality-critical industries.

Ongoing research directions include the lightweight optimization of diffusion-based models, autonomous threshold calibration without test data, and improving the realism of synthetic anomalies. In particular, leveraging advanced generative architectures or augmented reality simulation could provide more diverse and representative training sce-

narios, ultimately strengthening the robustness and generalizability of industrial anomaly detection systems.

Author Contributions: Conceptualization, A.C.; methodology, A.C.; validation, A.C.; formal analysis, A.C.; investigation, A.C.; resources, A.C.; data curation, A.C.; writing—original draft preparation, A.C.; writing—review and editing, S.A.M. and M.A.B.; visualization, A.C.; supervision, S.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset used in this study is the publicly available MVTec Anomaly Detection dataset (MVTec AD), which can be accessed at: https://www.mvtec.com/company/research/datasets/mvtec-ad (accessed on 2 May 2025). All code implementations referenced in this work are based on publicly available repositories. The authors did not release any additional proprietary code.

Conflicts of Interest: Author Mohammed Amin Belarbi was employed by the company Amintechs. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

GDE	Gaussian Density Estimator
SVDD	Support Vector Data Description
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
TPR	True positive rate
FPR	False positive rate
GMAC	Giga Multiply-Accumulate Operations
AP	Average precision
GPU	Graphics Processing Unit

Appendix A. Detailed Results

 Table A1. Synthesis of Overall AUC Performance per Object Class.

Methods	Bottle	Cable	Hazelnut	Pill	Screw	Capsule	Carpet	Grid	Leather	Metal Nut	Tile	Toothbrush	Transistor	Wood	Zipper	Average
Dinomaly	100	99.82	100	98.19	98.14	98.46	99.96	100	99.99	100	99.94	99.89	99.94	99.81	99.95	99.61
PatchCore	100	99.35	100	95.92	98.68	96.80	99.55	98.00	100	99.80	98.60	97.56	100	99.31	99.73	98.89
GLASS	100	96.15	100	87.60	91.87	96.18	99.04	100	100	99.66	99.99	96.67	98.19	99.69	99.68	97.65
DDAD	100	95.46	87.73	86.78	95.75	97.42	99.02	100	100	98.40	99.97	99.78	99.50	94.25	97.80	96.79
DRAEM	98.90	93.20	100	73.11	97.73	89.02	95.58	100	99.98	99.42	99.91	99.78	93.81	97.78	99.96	95.88
DiffusionAD	99.10	94.16	99.94	80.10	78.94	89.08	99.43	100	100	99.92	99.43	100	89.56	99.64	100	95.29
MSPBA	99.95	86.62	98.39	87.84	76.10	89.43	88.61	97.11	95.17	96.61	88.73	92.44	97.12	94.06	98.50	92.45
Cutpaste	99.82	86.91	96.98	81.78	73.43	68.72	92.82	98.15	99.73	96.16	95.39	94.11	93.37	99.11	99.36	91.72
RIAD	98.92	61.02	81.06	69.67	68.32	79.59	69.26	98.21	97.84	72.41	80.40	89.00	88.87	75.92	95.14	81.71
PatchSVDD	97.67	82.58	67.92	75.10	32.82	74.92	59.40	70.82	94.32	88.32	88.45	87.78	90.19	82.58	76.88	77.98

Table A2. Synthesis of Average Recall Performance per Object Class.

Methods	Bottle	Cable	Hazelnut	Pill	Screw	Capsule	Carpet	Grid	Leather	Metal Nut	Tile	Toothbrush	Transistor	Wood	Zipper	Average
Dinomaly	100	100	100	92.91	89.92	94.50	98.88	100	98.91	100	100	96.67	97.50	95.00	99.16	97.56
PatchCore	100	93.48	100	91.49	96.64	89.91	96.63	98.25	100	98.92	96.43	93.33	100	98.33	99.16	96.84
GLASS	100	90.22	100	87.94	79.83	88.07	91.01	100	100	96.77	98.81	76.67	92.50	96.67	95.80	92.95
DDAD	100	90.22	71.43	95.04	89.08	86.24	89.89	100	100	92.47	100	96.67	100	71.67	94.96	91.84
DRAEM	98.41	80.43	100	93.62	89.92	86.24	89.89	100	98.91	96.77	100	96.67	80.00	100	99.16	94.00
DiffusionAD	96.83	90.22	98.57	47.52	62.18	87.16	96.63	100	100	98.92	97.62	100	85.00	100	100	90.71
MSPBA	98.41	73.91	98.57	83.69	70.59	77.06	85.39	96.49	89.13	90.32	78.57	83.33	87.50	83.33	96.64	86.20
Cutpaste	95.24	75.00	97.14	58.16	52.94	60.55	79.78	94.74	100	84.95	95.24	86.67	82.50	95.00	96.64	83.64
RIAD	96.83	60.87	58.57	74.47	60.50	71.56	53.93	92.98	94.57	68.82	73.81	80.00	80.00	76.67	95.80	75.96
PatchSVDD	93.65	84.78	80.00	80.14	98.32	68.81	32.58	61.40	92.39	78.49	71.43	66.67	70.00	85.00	86.55	76.68

Methods	Bottle	Cable	Hazelnut	Pill	Screw	Capsule	Carpet	Grid	Leather	Metal Nut	Tile	Toothbrush	Transistor	Wood	Zipper	Average
Dinomaly	100	99.46	100	93.91	94.27	94.50	99.44	100	99.45	100	99.41	98.31	98.73	97.44	99.16	98.27
PatchCore	100	95.56	100	90.53	96.23	92.45	97.73	98.25	100	98.40	98.18	94.92	100	95.93	98.33	97.10
GLASS	100	90.22	100	80.26	82.61	89.72	94.74	100	100	97.30	99.40	86.79	92.50	97.48	97.44	93.90
DDAD	100	91.71	80.00	31.88	89.45	90.82	94.12	100	100	95.56	99.41	98.31	96.39	83.50	94.96	89.74
DRAEM	98.41	86.55	100	82.50	91.45	83.19	90.40	100	99.45	95.74	98.82	98.31	87.67	92.31	99.58	93.63
DiffusionAD	97.60	89.25	98.57	64.11	71.15	81.90	97.73	100	100	98.92	98.80	100	86.08	98.36	100	92.16
MSPBA	99.20	79.53	93.88	82.23	70.29	80.77	83.52	91.67	87.70	91.80	81.48	89.29	90.91	87.72	95.04	87.00
Cutpaste	97.56	79.31	93.79	68.62	66.67	64.39	84.52	97.30	99.46	89.77	89.39	89.66	88.00	97.44	96.64	86.83
RIAD	94.57	61.54	71.93	68.40	65.45	74.29	61.54	94.64	94.05	68.09	73.37	84.21	84.21	74.19	91.57	77.47
PatchSVDD	95.93	80.00	72.73	73.86	67.83	71.09	44.62	66.04	87.18	82.02	78.43	80.00	81.16	80.31	76.87	75.87

 Table A3. Synthesis of Average F1-Score Performance per Object Class.

 Table A4. Synthesis of Average AP Image Performance per Object Class.

Methods	Bottle	Cable	Hazelnut	Pill	Screw	Capsule	Carpet	Grid	Leather	Metal Nut	Tile	Toothbrush	Transistor	Wood	Zipper	Average
Dinomaly	100	99.81	100	98.43	98.49	98.59	99.96	100	99.99	100	99.94	99.89	99.94	99.81	99.95	99.65
PatchCore	100	99.37	100	96.53	98.65	96.96	99.59	98.80	100	99.82	99.18	98.20	100	99.32	99.75	99.08
GLASS	100	96.62	100	88.71	92.64	96.04	99.07	100	100	99.68	99.99	96.92	98.44	99.71	99.70	97.83
DDAD	100	96.48	88.73	87.64	96.16	97.63	99.04	100	100	98.68	99.97	99.78	99.51	95.08	98.39	97.14
DRAEM	98.35	94.04	100	64.53	97.85	90.44	96.48	100	99.98	99.44	99.92	99.79	95.29	97.84	99.96	95.59
DiffusionAD	99.33	94.60	99.94	83.32	81.08	90.61	99.50	100	100	99.92	99.59	100	92.40	99.62	100	95.99
MSPBA	99.95	89.31	98.39	90.10	77.47	89.94	87.45	97.11	95.25	97.06	90.74	94.75	97.34	95.35	98.41	93.24
Cutpaste	99.83	88.58	96.70	83.65	80.27	70.81	92.66	98.74	99.69	96.65	95.32	95.44	92.64	99.25	99.38	92.64

References

- Lee, J.; Bagheri, B.; Kao, H.A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manuf. Lett.* 2015, *3*, 18–23. [CrossRef]
- Lu, Y.; Weng, X. Smart manufacturing systems for Industry 4.0: Conceptual framework, architecture and key technologies. J. Manuf. Syst. 2018, 48, 25–34.
- 3. Ruff, L.; Vandermeulen, R.A.; Görnitz, N.; Deecke, L.; Siddiqui, S.A.; Binder, A.; Müller, K.-R.; Kloft, M. A unifying review of deep and shallow anomaly detection. *Proc. IEEE* 2021, 109, 756–795. [CrossRef]
- 4. Sultani, W.; Chen, C.; Shah, M. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 July 2018; pp. 6479–6488.
- Bergmann, P.; Fauser, M.; Sattlegger, D.; Steger, C. 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), Long Beach, CA, USA, 15–20 June 2019; pp. 9592–9600.
- Chen, Q.; Luo, H.; Lv, C.; Zhang, Z. A Unified Anomaly Synthesis Strategy with Gradient Ascent for Industrial Anomaly Detection and Localization. In *Computer Vision—ECCV* 2024; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2024; Volume 15125. [CrossRef]
- 7. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. arXiv 2016, arXiv:1605.07146.
- Li, C.-L.; Sohn, K.; Yoon, J.; Pfister, T. CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 9659–9669. [CrossRef]
- Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; Gehler, P. Towards Total Recall in Industrial Anomaly Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14318–14328. Available online: https://openaccess.thecvf.com/content/CVPR2022/papers/Roth_Towards_Total_ Recall_in_Industrial_Anomaly_Detection_CVPR_2022_paper.pdf (accessed on 25 February 2025)
- 10. Yi, J.; Yoon, S. Patch SVDD: Patch-Level SVDD for Anomaly Detection and Segmentation. In Proceedings of the Asian Conference on Computer Vision (ACCV), Kyoto, Japan, 30 November–4 December 2020.
- 11. Tsai, C.C.; Wu, T.H.; Lai, S.H. Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022; pp. 3992–4000. Available online: https://openaccess.thecvf.com/content/WACV2022/papers/Tsai_Multi-Scale_ Patch-Based_Representation_Learning_for_Image_Anomaly_Detection_and_Segmentation_WACV_2022_paper.pdf (accessed on 17 January 2025)
- Zavrtanik, V.; Kristan, M.; Skočaj, D. DRAEM—A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 8330–8339. Available online: https://openaccess.thecvf.com/content/ICCV2021/papers/Zavrtanik_ DRAEM_-_A_Discriminatively_Trained_Reconstruction_Embedding_for_Surface_Anomaly_ICCV_2021_paper.pdf (accessed on 15 February 2025)
- 13. Zhang, H.; Wang, Z.; Wu, Z.; Jiang, Y.G. DiffusionAD: Norm-guided One-step Denoising Diffusion for Anomaly Detection. *arXiv* **2023**, arXiv:2303.08730.
- 14. Zavrtanik, V.; Kristan, M.; Skočaj, D. Reconstruction by Inpainting for Visual Anomaly Detection. *Pattern Recognit.* **2021**, *112*, 107706. [CrossRef]
- 15. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015, Proceedings, Part III; Springer Nature: Cham, Switzerland, 2015; pp. 234–241.*
- Mousakhan, A.; Brox, T.; Tayyub, J. Anomaly Detection with Conditioned Denoising Diffusion Models. In *Pattern Recognition*. DAGM GCPR 2024; Cremers, D., Lähner, Z., Moeller, M., Nießner, M., Ommer, B., Triebel, R., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15297. [CrossRef]
- Guo, J.; Lu, S.; Zhang, W.; Chen, F.; Li, H.; Liao, H. Dinomaly: The Less Is More Philosophy in Multi-Class Unsupervised Anomaly Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025.
- Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. DINOv2: Learning Robust Visual Features Without Supervision. Transactions on Machine Learning Research. 2024. Available online: https://openreview.net/forum?id=a68SUt6zFt (accessed on 15 February 2025).
- 19. Davis, J.; Goadrich, M. The relationship between precision-recall and ROC curves. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 233–240.
- 20. Saito, T.; Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* 2015, *10*, e0118432. [CrossRef] [PubMed]

- 21. Van Rijsbergen, C.J. Information Retrieval; Butterworth-Heinemann: Oxford, UK, 1979.
- 22. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.