

Article

Enhancing PatchCore with Dynamic Scaling and Vision–Language Models for Explainable Industrial Defect Inspection

Oğuz Ergin , Emre Güçlü * , İlhan Aydın  and Erhan Akın 

Department of Computer Engineering, Fırat University, Elazığ 23119, Türkiye; edu@oguzergin.com (O.E.); iaydin@firat.edu.tr (İ.A.); eakin@firat.edu.tr (E.A.)

* Correspondence: egucclu@firat.edu.tr

Abstract

Although unsupervised anomaly detection has shown promising performance in industrial visual inspection, many architectures still struggle with variable input resolutions, and anomaly scores are often difficult for end users to interpret. This study proposes a multi-stage hybrid workflow for pixel-level defect localization and structured reporting in bolt head images. The proposed SA-PatchCore framework customizes PatchCore by extracting multi-scale representations from a frozen deep feature extractor and supporting resolution-adaptive anomaly-map reconstruction through dynamic feature-map sizing. After anomaly detection, a Qwen3-VL-32B-based reporting module, adapted with GRPO, uses both the original image and the anomaly overlay as visual evidence. It generates structured JSON outputs containing defect presence, a 3×3 location label, and a concise textual description. On the industrial bolt dataset, SA-PatchCore achieved 98.69% pixel-level AUROC, 29.73% Pixel-AP, and 39.24% oracle Pixel-F1max. Compared with PatchCore, PaDiM, DRaEM, and CS-Flow, the method delivered strong results, especially in Pixel-AUROC. In the reporting stage, defect presence/absence accuracy improved from 76.63% to 96.41%, while defective-sample recall increased from 74.23% to 96.14% over the baseline Qwen3-VL-32B. Exact location match rose from 28.15% to 53.78%, and mean partial location score improved from 32.29% to 65.34%. Overall, the framework combines accurate anomaly localization with structured reporting, improving interpretability and usability.

Keywords: industrial quality control; unsupervised anomaly detection; defect localization; anomaly map; vision–language model; GRPO



Academic Editors: Iwona Paprocka, Cezary Grabowik and Jozef Husar

Received: 12 June 2026

Revised: 4 July 2026

Accepted: 9 July 2026

Published: 15 July 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Bolts are critical fastening components in many sectors, including automotive, machinery manufacturing, and energy, because they directly affect load transfer and assembly reliability. During production, defects such as folds and cracks may occur on bolt heads. These defects can reduce part safety and assembly quality, make proper tool contact difficult during tightening and loosening operations, and lead to significant quality losses if defective parts continue through the production line [1]. Therefore, increasing quality requirements and the need for repeatable inspection have created a growing demand for visual inspection systems that are less dependent on operator judgment [2].

In traditional quality control, products are generally evaluated through operator observation, periodic checks, and optical inspection methods [3]. However, under mass production conditions, these approaches remain limited due to high labor requirements, operator-dependent evaluation, and the difficulty of consistently detecting small surface

defects [4]. For this reason, camera-based automated visual inspection and deep learning-based defect detection methods have become important components of quality assurance in manufacturing industries [5].

Nevertheless, collecting balanced and labeled datasets is challenging for rare and diverse production defects such as bolt head defects. In addition, generating pixel-level masks for training is often costly and time-consuming. Therefore, unsupervised industrial visual anomaly detection approaches, which learn the normal appearance from non-defective samples and identify deviations as anomalies, have gained importance [6–8]. However, reliable detection of subtle defects in small regions such as bolt heads remains challenging in practice, since performance is strongly affected by design choices such as input resolution, multi-scale representation, and anomaly map rescaling.

In addition to these challenges, high-performance anomaly detection models in the literature show two main limitations when they are applied to specific industrial inspection problems. First, fine scratches and small structural defects may be weakened when high-resolution images are processed at lower resolutions, making input resolution and multi-scale representation important design factors [9,10]. This limitation has motivated recent VLM-based explainable anomaly detection studies [11,12]. When the output is limited to a pixel-wise anomaly map or a statistical anomaly score, field operators may have difficulty interpreting the visual evidence behind the decision, which also reduces industrial traceability [12,13].

In this study, SA-PatchCore [14], a customized PatchCore-based approach, is presented for defective/non-defective classification and pixel-level defect localization in bolt head images. The anomaly detection stage is further supported by a VLM-based reporting layer that uses the anomaly map as visual evidence.

To make the novelty of the proposed framework explicit with respect to standard PatchCore, the previously introduced SA-PatchCore concept, and existing VLM-based reporting models, the main contributions of this study are summarized as follows:

- Compared with standard PatchCore, an improved SA-PatchCore architecture is proposed for bolt-head defect localization. The architecture strengthens the original PatchCore-based pipeline by using a frozen WideResNet-101-2 backbone, Layer-2/Layer-3 multi-scale feature fusion, higher-resolution ROI processing, neighborhood-aware feature aggregation, and coreset-based memory construction to improve the localization of subtle defects on small metallic regions. Dynamic anomaly-map reshaping is additionally used to support resolution-adaptive map reconstruction and stable inference under different input sizes.
- Compared with the previously introduced SA-PatchCore concept, this study extends the approach into a more complete industrial inspection framework. In addition to the architectural configuration, the revised framework includes systematic ablation analysis, robustness evaluation under industrial variations, runtime feasibility analysis, uncertainty estimation, and qualitative visualization of well-localized and challenging cases.
- In the VLM pipeline, a structured reporting framework is developed to convert anomaly-detection outputs into interpretable inspection reports. The framework uses the original bolt image together with the anomaly overlay as visual evidence and adapts Qwen3-VL-32B with GRPO to generate JSON-formatted outputs containing defect presence, a 3×3 regional location label, and a concise textual explanation. This enables the anomaly map to be used not only as a heatmap, but also as machine-readable and operator-readable decision evidence.
- The proposed hybrid inspection workflow is experimentally validated on an industrial bolt-head dataset and additionally evaluated against representative anomaly detection

baselines. The results demonstrate the localization performance of the improved SA-PatchCore configuration and show that the GRPO-adapted VLM substantially improves structured defect reporting compared with baseline VLM prompting and LoRA-SFT adaptation.

2. Related Works

2.1. Unsupervised Industrial Anomaly Detection

In industrial visual inspection, unsupervised anomaly detection approaches that learn the normal appearance only from non-defective samples have become increasingly important due to the rarity of defective samples and the high cost of pixel-level annotation [6]. Recent studies in this field show that factors such as the detection of small defects, data imbalance, and annotation cost directly affect model performance [15,16]. In real industrial environments, learning from limited labeled data becomes critical because defective samples occur rarely and annotation is often costly [17].

In the literature, unsupervised anomaly detection methods are generally examined under two main categories: reconstruction-based approaches and representation- or feature-based approaches. Reconstruction-based methods use reconstruction error as an anomaly signal, whereas representation-based methods generate anomaly maps by comparing local features extracted from pre-trained backbones with a memory bank or coreset. These two directions are commonly represented by methods such as DRÆM [7] and PatchCore [8].

Recent studies building on PatchCore have focused on issues that are important for transferring anomaly detection solutions to production lines. These include SoftPatch [18], which improves robustness against noisy training data, methods that incorporate positional information [19], and models targeting millisecond-level latency [20]. In addition, studies focusing on small and detailed regions such as screw or fastener heads emphasize the difficulty of this problem through supervised architectures for real-time defect segmentation, detection, or classification [21,22].

Despite these developments, reliable detection of subtle defects in small and detailed regions such as bolt heads still remains a critical challenge within the PatchCore-based framework. In particular, strengthening multi-scale representation and effectively using higher input resolution continue to be important design issues [8,18–20]. This study addresses this gap by combining Layer-2 and Layer-3 intermediate features, using a 384×384 input resolution, and dynamically determining the feature map size from the spatial dimensions of the embedding to improve inference stability under variable input sizes.

In addition to improving architectural representations, deriving the final decision from the anomaly map has also become an active research topic in the literature. Existing unsupervised anomaly detection approaches typically use the maximum pixel score for image-level decision-making. Recent studies such as StructCore [23] have emphasized the limitations of this single maximum score and proposed scoring strategies that evaluate the spatial structure of the anomaly map using statistical metrics such as Mahalanobis distance.

When the methods in the literature are examined, it can be seen that unsupervised industrial anomaly detection approaches have evolved based on different assumptions and strengths. Some methods rely on reconstruction error, while others evaluate local features extracted from pre-trained backbones through memory banks, statistical distributions, or teacher–student structures. For small, reflective, and detailed surfaces such as bolt heads, a general image-level decision alone is not sufficient; high resolution, strong local representation capability, spatial consistency, and interpretable reporting outputs become important together. In this context, the general characteristics of prominent methods in the literature are summarized in Table 1.

Table 1. Representative Unsupervised Industrial Anomaly Detection Methods and Their Relevance to Bolt-Head Inspection.

Model	Strengths & Limitations
PaDiM (2020) [24]	Models pre-trained CNN patch features with position-dependent Gaussian distributions. It is a strong baseline for fixed-pose inspection, but alignment and pose changes may affect small-defect detection.
CutPaste (2021) [25]	Learns self-supervised representations using synthetic cut-and-paste anomalies. It does not require defective training data, but synthetic anomalies may differ from real metallic micro-defects.
DRÆM (2021) [7]	Combines reconstruction and discriminative learning using synthetic anomalies. It is effective for surface defects, but its success depends on the realism of generated anomaly patterns.
CFLOW-AD (2022) [26]	Uses normalizing flows to model multi-scale pre-trained features. It supports fast inference, but flow training can be sensitive to noisy normal data.
PatchCore (2022) [8]	Builds a coreset memory bank from nominal patch features and scores test patches by nearest-neighbor distance. It is strong for local defects, but memory and search costs increase with resolution.
SoftPatch (2022) [18]	Extends PatchCore by filtering noisy patches before memory-bank construction. It improves robustness to contaminated training data, but adds extra processing complexity.
PNI (2023) [19]	Uses positional and neighborhood-conditioned feature distributions to refine anomaly maps. It is useful for well-aligned repetitive parts, but depends on stable positional priors.
ReConPatch (2024) [27]	Adapts pre-trained patch features to the target domain using modulation and contrastive learning. It is promising for subtle defects, but requires an additional adaptation stage.
EfficientAD (2024) [20]	Combines a lightweight extractor, student-teacher prediction and a global autoencoder for low-latency detection. It is suitable for fast inspection, but resolution-capacity balance remains critical.

Although such approaches improve image-level performance, their final decision mechanisms still rely on statistical thresholding. Therefore, they do not provide operators with directly interpretable information about the defect type or location. In the proposed method, the anomaly map is not treated only as a statistical score or heatmap output. Instead, it is used as visual evidence in the VLM-based structured reporting stage. In this way, the anomaly detection output is converted into a human-readable and machine-readable report format that includes defect presence, a 3×3 location label, and a brief description.

Standard benchmark datasets also play an important role in evaluating industrial anomaly detection methods. MVTec AD [28] has been widely used as a reference dataset because it provides high-resolution industrial images with pixel-level ground-truth masks

across multiple object and texture categories. More recent datasets such as VisA [29] further extend this evaluation setting by providing larger and more diverse industrial examples with both image-level and pixel-level annotations. However, these public datasets do not fully represent the specific visual characteristics of bolt-head inspection, such as small metallic regions, reflective surfaces, manufacturer stamps, and subtle structural defects. Therefore, in addition to referring to public benchmarks, this study evaluates the proposed method on a task-specific industrial bolt-head dataset collected under controlled imaging conditions.

2.2. Feature Matching and Multi-Resolution Anomaly Localization

Feature matching-based anomaly localization methods compare the representation of a test image with feature representations extracted from non-defective reference images. Unlike reconstruction-based methods, these approaches do not necessarily reconstruct the input image; instead, they estimate abnormal regions by measuring local feature dissimilarity between test and normal samples. Multi-resolution representations are particularly useful in this context because they combine fine spatial details with more semantic feature information.

In this direction, methods that transfer the idea of matching test samples with similar normal samples directly to localization have attracted attention. For example, SPADE [30] generates an anomaly map by matching a test image with a small set of normal images and comparing their representations through multi-resolution feature pyramids. In this way, the method aims to achieve pixel-level localization with very low training cost, or even without requiring additional training.

2.3. Vision–Language Models (VLMs) and Explainability

In industrial applications, simply classifying a product as defective or non-defective is often not sufficient. Reporting the visual evidence behind the decision in an understandable way is critical for traceability and operator confidence. Therefore, VLM-based approaches have recently attracted attention, particularly those that generate short descriptions and summaries from visual evidence such as anomaly maps [11]. Methods that jointly address defect detection and description generation also highlight the importance of reporting for improving interpretability [12].

In recent years, general-purpose pre-trained models, especially CLIP-based approaches, have been increasingly used for anomaly detection and segmentation without requiring labeled data from the target category. WinCLIP [31] aims to improve anomaly classification and segmentation performance by jointly using CLIP’s visual and textual representations. AnomalyCLIP [32] was developed to address the limitation that CLIP representations tend to focus more on object semantics and may not sufficiently distinguish defect patterns. This method learns object-independent normal and abnormal prompts, providing a more generalizable zero-shot anomaly detection approach across different domains. Such approaches are important because they help reduce the challenges of data collection and annotation in industrial settings.

In zero-shot industrial anomaly detection, prompt design has become increasingly important for guiding VLMs to recognize complex defect types more effectively. In this direction, the hybrid explainable guidance enhancement approach proposed in [13] aims to improve both explainability and localization performance in VLM-based methods. AnomalyLVM [33], one of the recent zero-shot studies in the literature, combines frozen SAM2, DINO-X, and CLIP models for anomaly detection and pixel-level mask generation. However, although powerful vision–language models such as AnomalyLVM mainly focus on

generating pixel-level defect maps, they do not directly provide the structured natural-language defect reports that operators often need in real field conditions.

In recent years, general-purpose pre-trained vision–language models have increasingly been used in industrial anomaly detection for both zero-shot/few-shot detection and explainable reporting. Studies in this field can generally be grouped into two main directions. In the first direction, CLIP-based methods focus on anomaly classification and pixel-level localization by using visual and textual representations. In the second direction, MLLM-/VLM-based assistant models target higher-level outputs, such as explaining defects, interpreting visual evidence, and providing structured reports to users. Prominent VLM-based approaches in the literature are summarized in Table 2.

Table 2. VLM-Based Approaches for Industrial Anomaly Detection and Structured Reporting.

Method Type	Models	Brief Assessment
VLM-based methods for industrial anomaly detection	WinCLIP [31], PromptAD [34], AdaCLIP [35], AnomalyCLIP [32], FiLo [36], AnomalyGPT [11], AnomalyLVM [33]	These methods support anomaly detection through vision–language alignment, prompt learning, or MLLM-based reasoning. However, even when they generate descriptions or reports, most do not specifically produce 3×3 location labels and JSON reports from anomaly overlay evidence.
General-purpose VLMs	Qwen3-VL [37], LLaVA-OneVision [38], InternVL3 [39], DeepSeek-VL2 [40]	These models provide strong base architectures for visual understanding, multi-image evaluation, and structured output generation. Qwen3-VL is suitable as the reporting model in this study because of its multi-image input processing, long-context capability, spatial reasoning ability, and JSON-like output generation.

As shown in Table 2, CLIP-based approaches mainly focus on anomaly classification and localization map generation, whereas VLM-based approaches are more suitable for explaining the decision and reporting it to users. However, many existing methods either generate only anomaly maps or, even when they provide natural-language explanations, do not directly transform pixel-level evidence into a structured report format. In this study, the map generated by the anomaly detection model is not used only for visualization; it is also provided to the model as visual evidence during the VLM-based reporting stage. The proposed framework jointly uses the original bolt image and the anomaly overlay panel to produce a structured JSON output consisting of defect presence, a 3×3 location label, and a brief description. For this purpose, the Qwen3-VL-32B model is trained through GRPO-based task-specific adaptation, providing more stable defect decisions and location reporting compared with baseline prompt-based use. In this way, the anomaly map does not remain merely a heatmap output; instead, it is transformed into interpretable and machine-readable decision evidence within the VLM-supported reporting process.

3. Materials and Methods

3.1. Image Acquisition Steps

The bolt head images used in this study were obtained using a custom image acquisition setup designed to standardize imaging conditions and make part positioning as repeatable as possible. The setup is composed of a Daheng industrial camera (Daheng Imaging, Beijing, China), a homogeneous LED lighting unit, and a rotating platform driven by a DC motor. The camera was fixed above the system to capture the bolt head region from a top-view perspective, while the bolt samples were positioned within a predefined field of view on the rotating platform. With this configuration, the camera angle, lighting conditions, and part position were kept as consistent as possible during image acquisition.

The homogeneous LED lighting used in the imaging setup was selected to reduce harsh shadows and glare effects that may occur on metallic surfaces. On reflective surfaces such as bolt heads, uncontrolled illumination may cause the defect region to appear distorted or may create false glare regions on defect-free surfaces. Therefore, the camera and light source were positioned to provide balanced illumination over the bolt head surface.

In addition, the light intensity of each color channel in the four-channel RGBW (Red, Green, Blue, White) LED module can be independently adjusted through potentiometers connected to the system. In this way, different lighting conditions were tested by considering the wavelength-dependent sensitivity characteristics of the camera and the reflective structure of the bolt surface, and suitable illumination settings for imaging were obtained. The diffuser layer integrated into the light source also contributed to reducing direct reflections on the bolt surface.

The RGBW LED module, diffuser, operating views under different lighting channels, and potentiometer connections used in the system are shown in Figures 1a, 1b, 1c and 1d, respectively. With this configuration, the model was intended to focus on structural defects on the bolt head surface rather than illumination-related irregularities.

A rotating platform was used to bring bolt samples into the camera field of view in a controlled and repeatable manner. The platform was driven by a DC motor controlled through a simple Arduino-based circuit. A potentiometer was added to the circuit to adjust the motor speed, allowing the rotation speed of the platform to be kept at a suitable level during image acquisition. This speed control mechanism prevented the bolts from passing through the camera field of view too quickly, reduced motion blur, and helped capture each sample under similar viewing conditions.

The images were collected under a fixed camera position, constant illumination, and controlled platform geometry. This reduced variations in viewing angle and scale among the samples in the dataset, resulting in a more stable preprocessing and anomaly detection process. The general structure of the image acquisition setup and an example top-view field of view captured by the industrial camera are shown in Figures 2a and 2b, respectively. The external appearance of the rotating inspection platform used in the experiments is shown separately in Figure 3. The hardware components used in the study are listed in Table 3.

Table 3. Devices Used in the Study.

Component	Technical Specifications
Camera Model	Daheng Image MER2-240-159U3C Industrial Camera
Lens	50 mm industrial lens with a long focal length

Table 3. *Cont.*

Component	Technical Specifications
Lighting Unit	4-chip RGBW LED (Red, Green, Blue, White) and 24V driver circuit
Optical Isolation	Light diffuser that prevents specular reflections
Movement Mechanism	DC motor-driven rotating platform used to carry bolt samples through the camera field of view
Motor Control Unit	Arduino-based motor control circuit
Speed Adjustment	Potentiometer-based manual speed adjustment for controlling the rotation speed of the platform

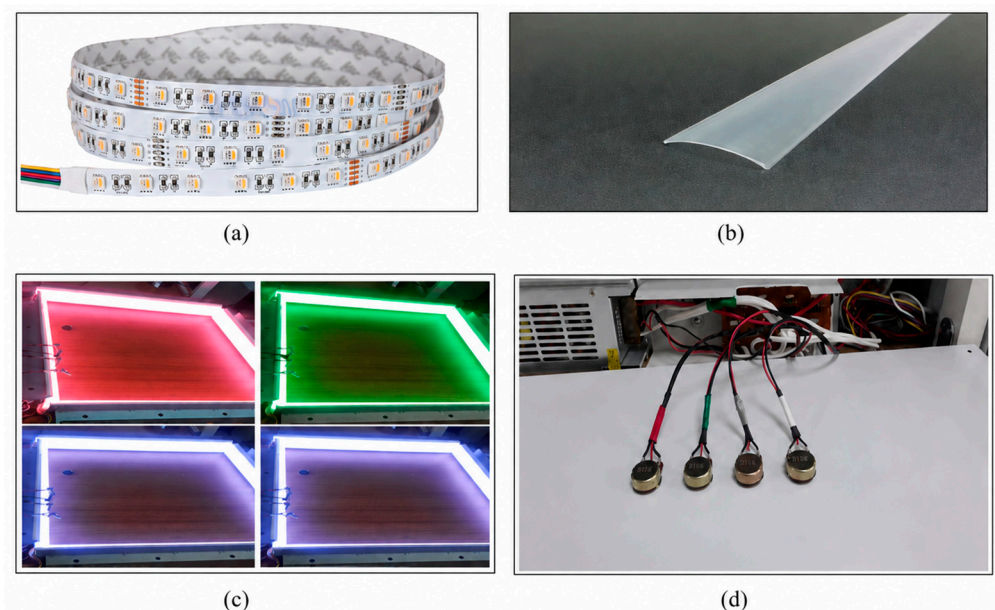


Figure 1. Illumination system components and control structure: (a) 4-chip RGBW LED module; (b) light diffuser used to reduce reflections on metallic surfaces; (c) operating views of the designed illumination system under different light channels, and (d) potentiometers used to adjust the intensity of individual RGBW channels.

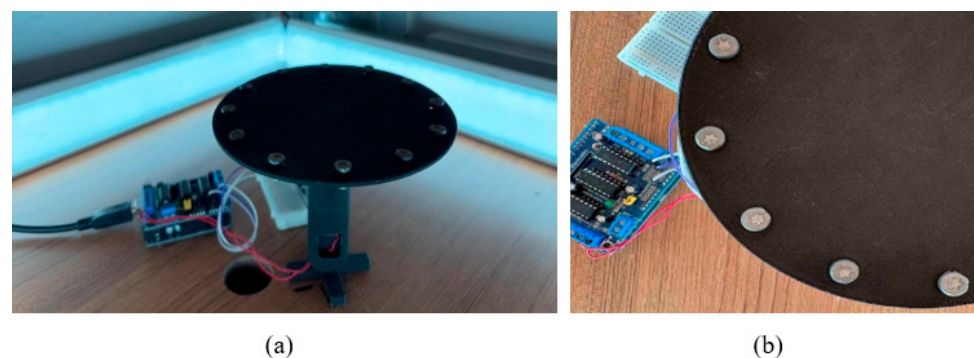


Figure 2. Image acquisition geometry used for bolt-head inspection: (a) general view of the controlled image acquisition setup; (b) sample top-view camera scene showing bolt samples on the rotating platform.

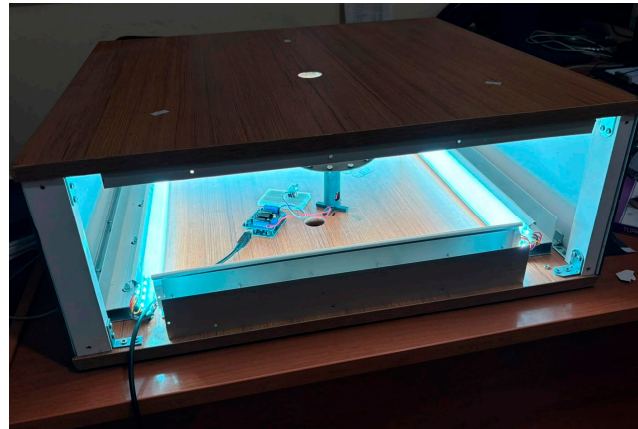


Figure 3. External view of the rotating inspection platform used for bolt-head image acquisition.

3.2. Image Preprocessing and Standardization

To reduce the effect of background, position, and scale variations on the model, all images were first subjected to a preprocessing stage. The dataset used in this study consists of ROI (Region of Interest) images extracted from bolt head regions. The bolt head region was segmented using the alpha channel when available; otherwise, it was separated using a binary mask $M(x,y) \in \{0,1\}$ obtained by simple thresholding. Based on this mask, the rectangular region enclosing the bolt head was defined by the bounding box of the mask. These operations are formulated in Equation (1).

$$\begin{aligned} x_{\min} &= \min\{x \mid \exists y, M(x,y) = 1\} \\ x_{\max} &= \max\{x \mid \exists y, M(x,y) = 1\} \\ y_{\max} &= \max\{y \mid \exists x, M(x,y) = 1\} \\ y_{\min} &= \min\{y \mid \exists x, M(x,y) = 1\} \end{aligned} \quad (1)$$

The resulting $x_{\min}$, $x_{\max}$, $y_{\min}$ and $y_{\max}$ coordinates define the smallest bounding box enclosing the bolt head and allow the ROI to be standardized with reduced background influence. This approach aims to reduce the effect of small positional shifts and background textures on the extracted features, enabling the model to focus mainly on the surface structure of the bolt head.

In addition, this standardized ROI region was used as a common reference area for the 3×3 regional location labels employed in the subsequent VLM-based reporting stage. Therefore, location information was evaluated not with respect to the raw camera image, but according to a normalized inspection area centered on the bolt head. Example defective and non-defective bolt samples from the dataset are shown in Figure 4.



Figure 4. Examples of Defective and Non-Defective Bolts.

3.3. Unsupervised Anomaly Detection Model and Training Architecture

In the proposed SA-PatchCore configuration, a pre-trained WideResNet-101-2 [41] backbone is used as a frozen feature extractor. To better preserve fine-grained defect cues, the Layer-2 and Layer-3 feature maps are extracted through forward hooks and fused within a multi-scale representation.

In the WideResNet-101-2 backbone, the Layer-2 output is represented as $F_2 \in \mathbb{R}^{B \times C_2 \times H_f \times W_f}$, while the Layer-3 output is represented as $F_3 \in \mathbb{R}^{B \times C_3 \times (H_f/2) \times (W_f/2)}$. Here, H_f and W_f denote the spatial dimensions of the extracted intermediate feature maps. For an input size of $H_{in} = 384$, the Layer-3 output is upsampled to the spatial resolution of Layer-2 using bilinear interpolation. The two feature maps are then concatenated along the channel dimension to obtain F_{cat} , as given in Equation (2). This strategy combines the higher spatial resolution of Layer-2 with the stronger semantic representation of Layer-3, aiming to capture subtle defects more effectively.

$$F_{cat} = [F_2; F_3] \in \mathbb{R}^{B \times (C_2 + C_3) \times H_f \times W_f} \quad (2)$$

The proposed method is based on storing local feature vectors extracted from non-defective training samples in a memory bank and querying this memory bank through nearest-neighbor similarity during inference. In this context, patch-based features extracted from the backbone network approximately represent the distribution of the non-defective class in the feature space. For test samples, the degree of deviation from this distribution is interpreted as a pixel-level anomaly score.

Unlike conventional PatchCore implementations, the resolution parameters were optimized in this study to improve pixel-level localization performance. While the original approach commonly uses an input size of 224×224 pixels, the proposed approach first resizes the input images to 416×416 pixels and then applies a 384×384 center crop to increase the level of detail in the feature maps. This higher-resolution configuration improves the detectability of fine scratches and small defects. In addition to the transition to higher input resolution, the Gaussian filter used in the final processing of the anomaly maps was fixed at $\sigma = 1.0$ to smooth patch-level noise.

To preserve spatial neighborhood information, 3×3 average pooling was applied to the fused F_{cat} representation with stride = 1 and padding = 1. The use of padding = 1 allows each spatial location to incorporate information from its neighboring regions without changing the spatial size of the feature map. In this way, the memory bank stores representations computed over a small local neighborhood rather than only isolated feature vectors. This design aims to obtain a more stable localization map without requiring an additional attention mechanism.

In the original PatchCore architecture, feature map dimensions are hard-coded as fixed values, which may cause system errors when different input resolutions are used. In this study, a dynamic feature map processing strategy was developed to support variable input sizes. Instead of relying on a fixed spatial-size assumption, the feature map dimensions are automatically determined from the height and width of the extracted feature tensor. During inference, the anomaly map is reshaped according to these dynamically determined dimensions without requiring any fixed-size assumption. These operations are formulated in Equations (3) and (4). This approach enables the model to operate at different input resolutions without requiring additional changes in the implementation. The proposed modifications have different roles in the final architecture. The higher input resolution is mainly intended to preserve fine defect cues before feature extraction. Layer-2/Layer-3 fusion is used to balance spatial detail and semantic representation, while 3×3 average pooling and neighborhood-aware k-NN scoring are used to stabilize local anomaly responses. Dynamic feature-map sizing does not directly

change the learned feature representation; instead, it supports resolution-adaptive anomaly-map reconstruction and prevents fixed-size assumptions during inference.

$$(H_f, W_f) = \text{shape}(E)[-2:] \quad (3)$$

$$A_{raw} = \text{reshape}\left(\left\{s\left(p_{i,j}^{test}\right)\right\}_{i,j=1}^{H_f, W_f}\right) \in R^{H_f \times W_f} \quad (4)$$

The raw embedding pool M collected from all training images contains a large number of vectors depending on the number of training samples and the size of the extracted feature map. Since this pool is too large to be processed directly by the k-Center Greedy algorithm, a two-stage reduction procedure is applied. In the first stage, when $|M| > M_{max} = 100,000$, a subset $M' \subseteq M$ is generated by parameter-free random sampling. In the second stage, sparse random projection is applied to M' to preserve the distances between vectors while reducing the computational dimensionality. This projection is used only for distance calculations in the k-Center Greedy algorithm. The final coreset vectors are selected from the original feature space.

The training stage does not involve gradient-based optimization. Instead, a representative coreset is constructed from the feature vectors collected in a single forward pass using the k-Center Greedy algorithm. To balance the increased number of patches and memory usage caused by the higher input resolution, the coreset sampling ratio was set to 10%. During inference, each patch embedding vector p^{test} obtained from the test image is queried against the final coreset MC using the k-NN algorithm. To improve stability, the nearest five neighbors were first identified ($k = 5$), and the anomaly score between p^{test} and M_C was calculated as the average Euclidean distance of the nearest three neighbors, as given in Equation (5). These patch-level scores were then placed at their corresponding spatial positions through the reshaping operation described in Equation (4), forming the final pixel-level anomaly map. The architectural structure of the proposed method is shown in Figure 5.

$$s(p^{test}) = \frac{1}{3} \sum_{r=1}^3 \|p^{test} - m_r\|_2, m_r \in N_5(p^{test}, M_C) \quad (5)$$

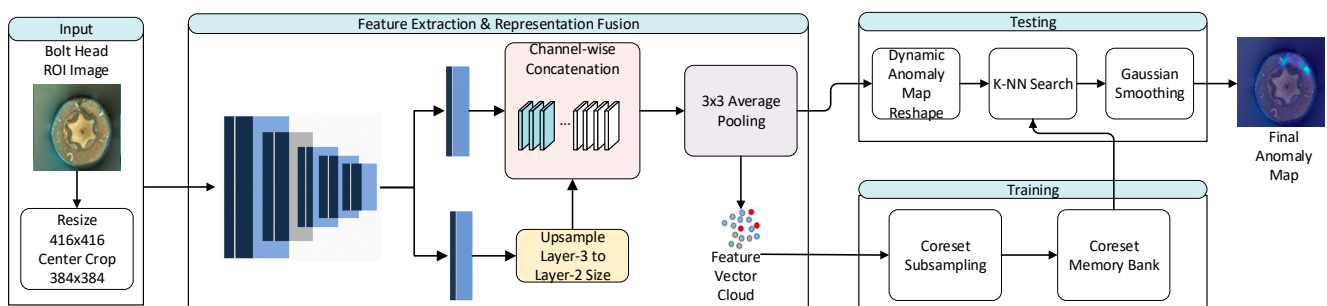


Figure 5. Proposed SA-PatchCore Architecture.

Figure 5 should be read as the complete flow of the proposed SA-PatchCore inference pipeline. The main elements to observe are the higher-resolution ROI processing, Layer-2/Layer-3 feature fusion, neighborhood-aware feature aggregation, coreset memory-bank construction, and dynamic anomaly-map reshaping. These components explain how the model preserves fine defect cues during feature extraction and converts patch-level k-NN distances into a final pixel-level anomaly map.

3.4. Inference, Anomaly Map and Visualization

The Pixel-AUROC and Pixel-AP metrics used in the quantitative evaluation were calculated directly from raw pixel-level anomaly scores and are therefore threshold-independent. For the threshold-dependent F1 analysis, the thresholds generated from the Precision–Recall

curve were scanned using the annotated test masks, and the maximum attainable F1 value was reported as Pixel-F1max. Since this threshold is selected using the test-set ground-truth masks, Pixel-F1max should be interpreted as an oracle upper-bound localization score rather than as a fixed-threshold deployment metric.

To estimate the uncertainty of the reported localization metrics, an image-level bootstrap analysis was additionally performed on the fixed test set. In each bootstrap iteration, test images were sampled with replacement, and Pixel-AUROC, Pixel-AP, and Pixel-F1max were recomputed. The analysis was repeated 1000 times, and the 95% confidence intervals were obtained from the 2.5th and 97.5th percentiles of the bootstrap distribution. For computational efficiency, pixel-level metrics were estimated using a fixed pixel subsampling strategy in each bootstrap iteration. The main result tables report the full-test point estimates, while the bootstrap analysis is used to assess the uncertainty of these estimates.

In contrast to the quantitative evaluation, an additional processing pipeline was applied when visualizing the anomaly maps for end users. This step was used to reduce contrast differences between samples and calibrate the visual scale. For this purpose, lower and upper clipping limits (A_{lw} , A_{hi}) were determined based only on the percentiles obtained from the non-defective training distribution. These limits were selected as the 40th percentile ($q_{low} = 0.40$) and the maximum value ($q_{high} = 1.00$) of the pixel scores in the training set, respectively. The min-max normalization process, applied only for visual comparability, is given in Equation (6), and values outside the limits were clipped to 0 and 1, respectively:

$$\tilde{A}(x, y) = \begin{cases} 0, & A(x, y) < A_{lw} \\ 1, & A(x, y) > A_{hi} \\ \frac{A(x, y) - A_{lw}}{A_{hi} - A_{lw}}, & \text{otherwise} \end{cases} \quad (6)$$

The normalized anomaly map was converted into a heatmap using the JET colormap and then blended with the original bolt image to generate overlay visualizations. This overlay generation process is formulated in Equation (7).

$$O(x, y) = \frac{H(\tilde{A}(x, y)) + I(x, y)}{\max(H(\tilde{A}) + I)} \quad (7)$$

3.5. Natural Language Reporting and Hybrid Localization with VLM

To make the anomaly detection output more interpretable, a VLM-based reporting stage was added after SA-PatchCore. The aim of this stage was not only to decide whether a defect was present, but also to generate a short structured report. Therefore, the VLM was designed to produce three outputs: defect presence, a 3×3 regional location label, and a concise explanation. The original bolt image and the anomaly-map overlay were provided together as visual evidence.

In this study, the Qwen/Qwen3-VL-32B-Instruct [37] model was used for VLM-based reporting. The model was loaded through the Unsloth (version v0.1.41-beta) [42] FastVision-Model framework and operated with 4-bit quantization [43] to improve memory efficiency. The maximum context length was set to 16,384, allowing the model to process both the visual input and the structured prompt text together.

Instead of directly training the full model, a parameter-efficient fine-tuning strategy was adopted. In this scope, a LoRA-based PEFT configuration was applied, and the visual layers, language layers, attention modules, and MLP modules were included in training. The LoRA rank was set to $r = 32$, the LoRA scaling coefficient was set to $\alpha = 32$, the dropout

value was set to 0, and the bias setting was set to None. Unsloth gradient checkpointing was enabled to reduce memory usage. With this configuration, only 283.84 million of the model's approximately 33.64 billion parameters were made trainable, meaning that approximately 0.84% of the total parameters were updated. In this way, the visual reporting capability of the large-scale VLM was preserved, while the model was adapted to produce defect decision and location outputs specific to industrial defect data.

Two different data sources were used during the data preparation stage. The first source was the training folder containing non-defective bolt head images. This folder included a total of 524 non-defective images, which were loaded in ImageFolder format. The second source consisted of the outputs obtained from the SA-PatchCore-based anomaly detection system. This source included the original defective images, anomaly overlay maps, and the corresponding CSV file containing the location information for each sample. A total of 647 defective samples were obtained from the anomaly detection results. The original images and overlay images were matched using common keys derived from their filenames, so that the original image, evidence map, and target location label were stored together for each defective sample.

The location labels were obtained from the 9-element binary grid information in the corresponding CSV file. This grid divides the normalized bolt head ROI into a 3×3 spatial region. The labels used in this study were top-left, top, top-right, left, right, bottom-left, bottom, and bottom-right. When multiple active locations were detected, the labels were concatenated with commas and stored in a multi-location format. For example, if both the top and right regions were active in the target grid, the target output was defined as top, right. The 3×3 grid used here was not applied to the full camera image, but to the normalized bolt head ROI obtained during preprocessing. Therefore, the location label should not be interpreted as an absolute three-dimensional point on the physical bolt or as a mechanical coordinate on the production line. Instead, the labels indicate the approximate regional position of the defect within the bolt head image. This approach makes the location information in the VLM output more consistent and comparable across images.

The 3×3 location labels were derived from the manually annotated defect regions by mapping their spatial positions onto the normalized bolt-head ROI grid. If a defect region activated more than one grid cell, the corresponding region labels were concatenated using commas. For boundary cases, the grid assignment was determined according to the dominant overlap of the annotated defect region within the ROI grid.

A single composite image was provided to the VLM. This composite image consisted of two panels. The left panel contained the original bolt head image, while the right panel served as the evidence panel. For defective samples, the evidence panel could include the anomaly overlay map, whereas it was provided as a blank black panel in the alternative setting. Each panel was resized and padded to 512×512 using the letterbox method, and an 8-pixel gap was left between the two panels. Thus, the final composite image was generated with a size of 1032×512 . During letterboxing, the aspect ratio of the image was preserved, and the remaining empty regions were filled with black. This design allowed the model to evaluate both the actual defect traces in the original image and the spatial evidence obtained from the anomaly detection model within the same visual context.

An important design choice was also introduced for data augmentation and to reduce shortcut learning. For non-defective samples, the right panel was left blank, and the target output was defined as: defect = "no", location_3 $\times$ 3 = "none". For defective samples, two different versions were generated. In the first version, the anomaly overlay map was placed in the right panel. In the second version, the same defective sample was regenerated with a blank right panel. This strategy was used to prevent the model from relying on shortcut-based decisions such as "if the right

panel is blank, the sample is non-defective” or “if the right panel is colored, the sample is defective.” This condition was also explicitly stated in the prompt. The model was instructed to make the decision primarily based on the original image in the left panel and to use the right panel only as supporting spatial evidence.

To prevent data leakage, the train–test split was performed at the base-image level before generating different evidence-panel versions. Therefore, overlay-supported and blank-evidence versions of the same defective image were never shared between the training and test sets. Non-defective images were divided using an 80/20 train–test ratio, while defective images were divided using a 50/50 train–test ratio. This split was selected to preserve a sufficient number of defective samples for VLM evaluation, since the reporting task requires not only a defect presence/absence decision but also a 3×3 regional location output.

After the base-image split, different evidence-panel versions were generated. For non-defective samples, the right evidence panel was left blank, and the target output was defined as defect = “no” and location_3 $\times$ 3 = “none”. For defective samples, two input versions were generated: one with the anomaly overlay in the right panel and one with a blank evidence panel. This design was used to evaluate whether the model could make defect decisions both with and without explicit anomaly-map evidence, while also reducing shortcut learning based only on the presence or absence of a colored overlay panel.

The resulting dataset composition used for VLM-based reporting is summarized in Table 4.

Table 4. Dataset composition used for VLM-based reporting.

Subset	Sample Type	Split Ratio	Independent Images	Evidence Condition	Input Records
Training	Non-defective	80%	419	Blank evidence panel	419
Training	Defective	50%	323	Overlay + blank evidence panels	646
Test	Non-defective	20%	105	Blank evidence panel	105
Test	Defective	50%	324	Overlay + blank evidence panels	648
Total train	-	-	742	-	1065
Total test	-	-	429	-	753

It should be noted that the VLM reporting test set is not class-balanced. At the independent-image level, the test set contains 105 non-defective images and 324 defective images. At the input-record level, this imbalance becomes more pronounced because each defective test image is evaluated under two evidence conditions, namely with overlay evidence and with a blank evidence panel. As a result, the final VLM test set consists of 105 non-defective input records and 648 defective input records.

This design was intentionally used to evaluate the reporting model under both evidence-supported and non-evidence-supported defective cases. However, this class distribution may affect aggregate metrics such as overall defect yes/no accuracy. Therefore, overall accuracy is not interpreted alone in this study. Instead, class-specific metrics, including normal sample accuracy, defect recall, false-positive rate, and false-negative rate, are reported separately. In addition, location metrics are reported both over all input records and over defective records only, since non-defective samples have the target location label none.

In the prompt design, the model was instructed to generate a strict JSON output instead of unstructured free text. The model was expected to return only three fields: defect, location_3 $\times$ 3, and explanation. The defect field indicates the defect status as yes or no. The location_3 $\times$ 3 field specifies the location of the defect on the 3×3 grid. If no defect is present, this field is expected to be none; if a defect is present but its location cannot be determined, it is expected to be unknown. If multiple defect regions are present, the locations are separated by commas. The explanation field is used to provide a brief sentence explaining the decision. The prompt explicitly instructed the

model not to provide additional explanations, not to generate pixel coordinates, and to return only a valid JSON output.

GRPO-based [44] reward functions were used to guide the model toward the desired output format and correct decision-making behavior. Two main reward functions were defined during training. The first was the `formatting_reward` function, which evaluates the structural validity of the output. This function rewards outputs that can be parsed as JSON, contain the three expected keys, include valid values in the defect and location 3×3 fields, and provide a non-empty explanation field. In addition, an extra reward was assigned when the location field was none for non-defective samples and a value other than none for defective samples. Outputs that violated the JSON format or contained unnecessary repetitions were penalized.

The second function was the `correctness_reward` function, which measures decision accuracy. This function compares the defect presence/absence decision generated by the model with the target value. A high positive reward was assigned when the defect decision was correct, while incorrect decisions were penalized. For location prediction, both exact and partial matches were considered. If the location generated by the model exactly matched the target location, a full reward was assigned. In multi-location cases, if the model output contained only a subset of the target locations, a partial reward was applied. For example, if the target location was “top, right” and the model generated only top, the output was not treated as completely incorrect; instead, it received a partial score based on the overlap ratio. This design provides a more flexible evaluation, since anomaly regions in industrial defects may extend across more than one grid cell. Outputs that could not be parsed as JSON were penalized with a negative reward. The location matching score is formulated in Equation (8).

$$S_{\text{loc}} = \begin{cases} 1, & P = T \\ \frac{|P \cap T|}{|T|}, & P \cap T \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

Here, P denotes the set of locations predicted by the model, while T denotes the target location set. With this definition, exact matching is measured for single-location labels, while partial overlap is also taken into account for multi-location labels.

GRPO training was performed using GRPOTrainer from the TRL library. During training, a learning rate of 5×10^{-5} , the AdamW 8-bit optimizer, $\text{beta1} = 0.9$, $\text{beta2} = 0.99$, a weight decay of 0.05 and a cosine learning rate schedule were used. The warmup ratio was set to 0.1. The batch size per device was set to 2, the number of gradient accumulation steps was set to 2, and the number of completions generated per prompt was set to 4. Thus, an effective batch size of 4 was used. The maximum prompt length was set to 1024, the maximum completion length to 512, the total number of training steps to 200, and the checkpoint saving interval to 50 steps. The gradient norm limit was set to 1.0. For the GRPO loss, the `dr_grpo` loss type, sequence-level importance sampling, and the training setting without masking truncated completions were used. During training, the formatting reward and correctness reward were optimized together.

In the final inference stage, the selected VLM checkpoint at step 150 was loaded and evaluated on the entire test set. During inference, the model was run deterministically, sampling was disabled, and `do_sample` was set to False. For each generated output, the target defect status, target location, predicted defect status, predicted location, JSON validity, exact location match, partial location score, false-positive case, and false-negative case were recorded for reporting and analysis. In addition, visual reports showing the composite image together with the model output were generated for each sample. The final structured VLM report can be expressed as shown in Equation (9).

$$R = f_{\theta}(I_c) \rightarrow \{c, l, e\} \quad (9)$$

Here, R denotes the final report generated by the model, f_θ denotes the Qwen3-VL model adapted with GRPO, I_c denotes the composite visual input, c denotes the defect presence decision, l denotes the 3×3 location label, and e denotes the brief explanation. With this structure, the system is not limited to producing only a numerical anomaly score. It also provides an interpretable industrial inspection report that includes the presence, location, and brief explanation of the defect. In summary, GRPO was used to align the VLM with the structured inspection task rather than to improve generic visual recognition alone. The prompt-only VLM can generate JSON-like outputs, but it is not explicitly optimized to make consistent defect decisions or to assign valid 3×3 regional labels from anomaly-map evidence. GRPO addresses this limitation by rewarding three desired behaviors: parseable structured output, correct defect presence/absence prediction, and location labels that exactly or partially match the target grid region. Therefore, after GRPO adaptation, the model is expected to use the original image and anomaly overlay more consistently, reduce missed defective samples, and generate more reliable structured reports.

4. Experimental Setup

4.1. Bolt-Head Dataset, Annotation and Evaluation Protocol

The bolt-head dataset used in this study was constructed from ROI images extracted from the head regions of bolt samples. The final curated dataset was organized into non-defective and defective groups. One ROI image per selected bolt-head sample was used in the final dataset. The defective samples mainly exhibited two dominant visual patterns: surface cracks on the bolt head and deformation around the drive-slot/recess region. In the experimental protocol, these patterns were evaluated together under the defective category, since the aim of the study was binary anomaly localization rather than supervised defect-type classification.

The SA-PatchCore anomaly detection model was trained using only non-defective ROI images ($N = 524$). This follows the standard unsupervised industrial anomaly detection setting, where the model learns the normal appearance distribution and treats deviations from this distribution as anomalies. For evaluation, a test set consisting of 100 non-defective ROI images and 100 defective ROI images was used. The defective test images were manually annotated at the defect-region level and converted into binary ground-truth masks for pixel-level evaluation. The split was made reproducible using a fixed seed value (seed = 42), and all compared methods were evaluated using the same data split and the same evaluation protocol.

The general composition of the bolt-head dataset is summarized in Table 5.

The defect-region annotations used for pixel-level evaluation were prepared manually using LabelMe [45] (version 5.9.1). The annotation process focused on visible defect traces on the bolt-head ROI images, and only the regions labeled as defective were converted into binary ground-truth masks for evaluation. After annotation, the ROI images and corresponding masks were visually checked for consistency to ensure that the marked regions corresponded to visible defect areas and that non-defective visual elements were not included in the defect masks.

Table 5. Composition of the bolt-head ROI dataset used for anomaly detection.

Dataset Part	Class	Annotation Type	Usage	Number of Images
Training set	Non-defective	Image-level normal label	Memory-bank construction	524
Test set	Non-defective	Image-level normal label	False-positive and image-level evaluation	100
Test set	Defective	Manual defect-region mask	Pixel-level localization evaluation	100
Total				724

A single finalized annotation set was used in the experiments. Therefore, a formal inter-annotator agreement analysis was not performed. The annotations were used as reference masks for pixel-level evaluation and also served as the basis for deriving the regional 3×3 location labels used in the VLM reporting stage.

To characterize the scale of the annotated defects, the main annotation statistics computed from the manually annotated defective test images are summarized in Table 6.

Table 6. Summary of defect annotation statistics for the defective test images.

Property	Value
Annotated defective images	100
Total defect regions	185
Images with multiple defect regions	70
Annotation label	defected
ROI image resolution	512×512
Mean defect-region area	$1270.70 \pm 694.28 \text{ px}^2$
Mean defect-region area ratio	$0.485 \pm 0.265\%$ of ROI
Defect-region area ratio range	0.060–1.578% of ROI
Mean defect bounding-box size	$43.46 \times 47.31 \text{ px}$
Mean per-image union defect area	$0.897 \pm 0.336\%$ of ROI

These statistics show that the annotated defect regions occupy only a small portion of the bolt-head ROI. The mean per-region defect area was 0.485% of the ROI, while the mean per-image union defect area was 0.897% of the ROI. Therefore, the pixel-level evaluation represents a highly imbalanced localization problem, where the number of normal pixels is substantially larger than the number of defective pixels.

For reproducibility, the anomaly detection, robustness, and VLM reporting experiments were kept as separate evaluation settings. The anomaly detection and baseline comparison experiments used the fixed SA-PatchCore split summarized in Table 5, and all compared anomaly detection methods were evaluated on the same test images and masks. The robustness analysis used the same trained SA-PatchCore model without retraining and applied only test-time perturbations to the defective masked test images. For the VLM reporting module, the train/test split was performed at the base-image level before generating blank-evidence and anomaly-overlay evidence versions, preventing different evidence versions of the same original image from appearing in both training and testing.

4.2. Public Benchmark Dataset: MVTec AD

To evaluate the generalizability of the proposed SA-PatchCore configuration beyond specific bolt defects and its adaptability to standard industrial inspection scenarios reported in the literature, the widely accepted MVTec AD dataset was used as a benchmark. The MVTec AD dataset consists of high-resolution industrial images and includes 15 categories in total: 5 texture categories (carpet, grid, leather, tile, and wood) and 10 object categories (bottle, cable, capsule, hazelnut, metal_nut, pill, screw, toothbrush, transistor, and zipper). As in the bolt dataset experiments, the overall performance of the model was evaluated using pixel-level defect localization metrics, namely Pixel-AUROC, Pixel-AP, and Pixel-F1max.

4.3. Proposed SA-PatchCore Configuration

The proposed SA-PatchCore configuration was implemented using a frozen WideResNet-101-2 backbone. The anomaly map was generated by applying the k-NN algorithm to the embedding pool extracted from the training set. To better represent the distribution of non-defective samples in the feature space, the coreset sampling ratio was set to 10%. To preserve the sharpness of the anomaly map, a Gaussian filter

with $\sigma = 1$ was applied during the post-processing stage. To support variable input sizes, the reshaping step of the anomaly map was computed dynamically according to the input resolution.

4.4. Baseline Comparison Protocol

To ensure a fair comparison, ROI standardization was applied to all methods evaluated on the bolt dataset. As described in the Materials and Methods section, a mask-based bounding box approach was used to reduce background effects, and the resulting standardized ROI images were resized to 384×384 pixels. For the MVTec AD dataset used in the literature comparison, this dataset-specific masking procedure was not applied; instead, the default preprocessing protocol of the dataset was followed.

Since the compared methods belong to different architectural families, such as representation-based memory- or nearest-neighbor-based methods and reconstruction-based encoder–decoder architectures, conducting a controlled comparison using a single common backbone for all methods is not practically feasible. Therefore, the training and testing procedures of the reference methods, including DRÆM, PatchCore, PaDiM and related methods, were carried out using the widely used open-source Anomalib [46] library (version 2.1.0). Each method was trained and tested on our dataset using the default architectural configurations and training hyperparameters provided by the library. Only the data split protocol and the 384×384 input resolution were standardized, allowing the methods to be compared fairly under their own default settings.

It should be noted that the compared methods belong to different architectural families and do not share the same set of tunable parameters. For example, memory-bank-based, reconstruction-based, flow-based, and teacher–student-based methods expose different resolution, feature extraction, training, and post-processing parameters. Therefore, applying an identical hyperparameter optimization grid to all methods is not technically meaningful. In this study, the comparison was designed as a controlled reference evaluation rather than an exhaustive hyperparameter search. The same dataset split, ROI standardization procedure, input resolution, and evaluation metrics were used across all methods, while the baseline models were kept at their widely used Anomalib default configurations.

4.5. Metrics and Thresholding

Model performance was evaluated using Pixel-AUROC, Pixel-AP, and Pixel-F1max. Pixel-AUROC and Pixel-AP were computed in a threshold-independent manner using the raw pixel-level anomaly scores. Pixel-F1max was obtained by scanning the thresholds derived from the Precision–Recall curve and selecting the threshold that maximized F1 on the annotated test masks. Therefore, Pixel-F1max is reported only as an oracle upper-bound estimate of threshold-dependent localization performance. It was not used as a validation-selected fixed-threshold test metric. For this reason, the interpretation of the results primarily relies on the threshold-independent Pixel-AUROC and Pixel-AP metrics, while Pixel-F1max is used as a supplementary measure of the best attainable segmentation overlap under the evaluated score distribution. The scikit-learn [47] library was used for these metric computations.

4.6. Computational Resources

The SA-PatchCore anomaly detection pipeline was run on a local workstation (CPU: 11th Gen Intel® Core™ i5-11400H @ 2.70 GHz (Intel Corporation, Santa Clara, CA, USA), GPU: NVIDIA GeForce RTX 3050 Laptop GPU (NVIDIA Corporation, Santa Clara, CA, USA), RAM: 16 GB). VLM-based reporting and the comparison experiments conducted using Anomalib were performed on Google Colab with an NVIDIA A100 GPU (NVIDIA A100-SXM4-80GB VRAM; CUDA: v12.8; PyTorch: v2.9.0 + cu128). The preprocessing

and evaluation protocols were kept fixed across all experiments, while the execution environments differed only in terms of computational resources. Therefore, latency and runtime measurements should be considered environment-dependent.

5. Results

5.1. Pixel-Level Localization Performance (Bolt Dataset)

The performance of the developed SA-PatchCore anomaly detection workflow was reported using quantitative metrics in terms of pixel-level defect localization and was also evaluated through anomaly maps and overlay visualizations.

All images were loaded at 416×416 pixels and processed using a 384×384 center crop. The bolt head region was separated using a mask-based bounding box to reduce background influence. The proposed SA-PatchCore framework was implemented on a frozen WideResNet-101-2 backbone with Layer-2/Layer-3 multi-scale feature fusion. The embedding pool was extracted in a single pass, and the coreset was selected using k-Center Greedy. During inference, an anomaly map was generated using the k-NN algorithm. For visualization, overlay images were created by applying min–max normalization and the JET colormap. The saturation thresholds used for visualization were determined only from the anomaly score distribution obtained from non-defective training images. Quantitative metrics, however, were calculated directly from the raw pixel-level anomaly scores without applying any visual normalization.

A comparison was conducted with widely used anomaly detection methods on the same dataset. The comparison included the original SA-PatchCore, PatchCore, PaDiM, DRaEM, and CS-Flow [48]. The proposed SA-PatchCore configuration was evaluated under the same protocol. All methods were implemented using the Anomalib library and evaluated with the same data split (seed = 42) and the same preprocessing pipeline. Pixel-AUROC and Pixel-AP were computed using the raw anomaly scores. Pixel-F1max was reported as the maximum F1 value obtained by scanning the thresholds derived from the precision–recall curve on the defective test subset with ground-truth masks. The performance metrics are reported in Table 7. To provide a broader comparison with recent industrial anomaly detection approaches, EfficientAD and SuperSimpleNet [49] were also evaluated under the same dataset split, ROI preprocessing pipeline, input resolution, and pixel-level evaluation protocol.

Table 7. Pixel-level performance (masked defective test subset).

Model	Pixel-AUROC	Pixel-AP (PR-AUC)	Pixel-F1max (Oracle)
Proposed SA-PatchCore	98.69%	29.73%	39.24%
Original SA-PatchCore	97.79%	15.65%	25.85%
PatchCore	97.10%	31.42%	40.08%
PaDiM	91.93%	12.56%	22.14%
DRaEM	91.14%	14.03%	20.50%
EfficientAD	81.64%	0.01%	0.03%
CS-Flow	78.27%	2.33%	4.82%
SuperSimpleNet	50.90%	0.00%	0.027%

The term “Proposed SA-PatchCore” refers to the customized framework introduced in this study. The term “Original SA-PatchCore configuration” refers to the earlier SA-PatchCore setting evaluated without the complete set of proposed architectural and inference-stage modifications. All values in the table were computed using the same test split, the same ground-truth masks, and raw anomaly scores. A more detailed qualitative analysis of the anomaly-map failure modes is provided in Section 5.2.

To further assess the reliability of the main pixel-level localization results, image-level bootstrap resampling was applied to the fixed test set. The resulting 95% confidence intervals for the main configurations are reported in Table 8.

Table 8. Bootstrap uncertainty estimates for the main pixel-level localization metrics.

Model	Pixel-AUROC Mean, 95% CI (%)	Pixel-AP Mean, 95% CI (%)	Pixel-F1max Mean, 95% CI (%)
Original SA-PatchCore	97.75 [97.11, 98.28]	16.87 [12.12, 22.79]	27.28 [21.65, 32.61]
PatchCore	97.85 [97.28, 98.42]	22.71 [16.75, 30.39]	29.35 [23.60, 35.36]
Proposed SA-PatchCore	98.70 [98.38, 98.99]	30.15 [24.64, 36.52]	39.52 [35.19, 44.21]

The bootstrap results support the improvement of the proposed SA-PatchCore configuration over the original SA-PatchCore configuration, particularly in Pixel-AP and Pixel-F1max. The confidence intervals also show that the proposed configuration provides higher point estimates than PatchCore for the reported pixel-level metrics. However, since some confidence intervals overlap, especially for PR-based metrics, small numerical differences between the proposed method and PatchCore should be interpreted cautiously rather than as definitive statistical superiority.

Although the Pixel-AUROC values are high, the relatively lower Pixel-AP and Pixel-F1max values can be attributed to the fact that defect pixels occupy only a small portion of the image area and that PR-based metrics are more sensitive to class imbalance. This observation is also supported by the annotation statistics reported in Table 6, where the mean individual defect-region area corresponds to only 0.485% of the ROI, and the mean per-image union defect area corresponds to 0.897% of the ROI. Therefore, Pixel-AP and Pixel-F1max were reported in addition to Pixel-AUROC to provide a more complete evaluation of localization performance.

It should be noted that the reported Pixel-F1max values are not validation-selected fixed-threshold F1 scores. They represent the maximum F1 values obtained by scanning thresholds on the annotated test masks. Therefore, they are useful for analyzing the best attainable overlap between predicted anomaly regions and ground-truth defect masks, but they may provide an optimistic estimate compared with a threshold fixed on an independent validation set. For this reason, Pixel-AUROC and Pixel-AP are emphasized as the primary threshold-independent localization metrics, while Pixel-F1max is reported as a supplementary oracle upper-bound measure.

The proposed SA-PatchCore configuration achieved the highest Pixel-AUROC among the evaluated methods. Compared with the baseline SA-PatchCore configuration, the proposed version increased Pixel-AP from 15.65% to 29.73% and Pixel-F1max from 25.85% to 39.24%. This improvement can be attributed to the combined effect of higher-resolution ROI processing, Layer-2/Layer-3 multi-scale feature fusion, neighborhood-aware feature aggregation, and dynamic anomaly-map reshaping, which help preserve subtle defect responses on small metallic regions. To ensure that this improvement was not caused by an evaluation mismatch, all methods were evaluated using the same ROI standardization pipeline, the same train–test split, the same ground-truth masks, and raw anomaly scores without applying visualization normalization.

However, the results also show that the proposed configuration does not dominate all PR-based metrics. PatchCore achieved slightly higher Pixel-AP and Pixel-F1max values than the proposed SA-PatchCore. This indicates that the proposed configuration improves the global ranking of anomalous pixels, as reflected by Pixel-AUROC, while PatchCore still provides stronger precision–recall behavior at the evaluated threshold range. Therefore, the proposed model should be interpreted as improving localization sensitivity and AUROC performance, rather than uniformly outperforming all baselines across every metric.

To complement the quantitative results, additional qualitative examples of the proposed SA-PatchCore outputs are shown in Figure 6. The selected examples include three well-localized cases and three challenging cases. This visualization helps demonstrate

both the strengths and limitations of the proposed method. In the well-localized cases, the anomaly responses are concentrated around the manually annotated defect regions. In the challenging cases, the predicted response may become spatially diffuse or partially shifted from the annotated region, which is consistent with the lower PR-based localization metrics despite the high Pixel-AUROC values.

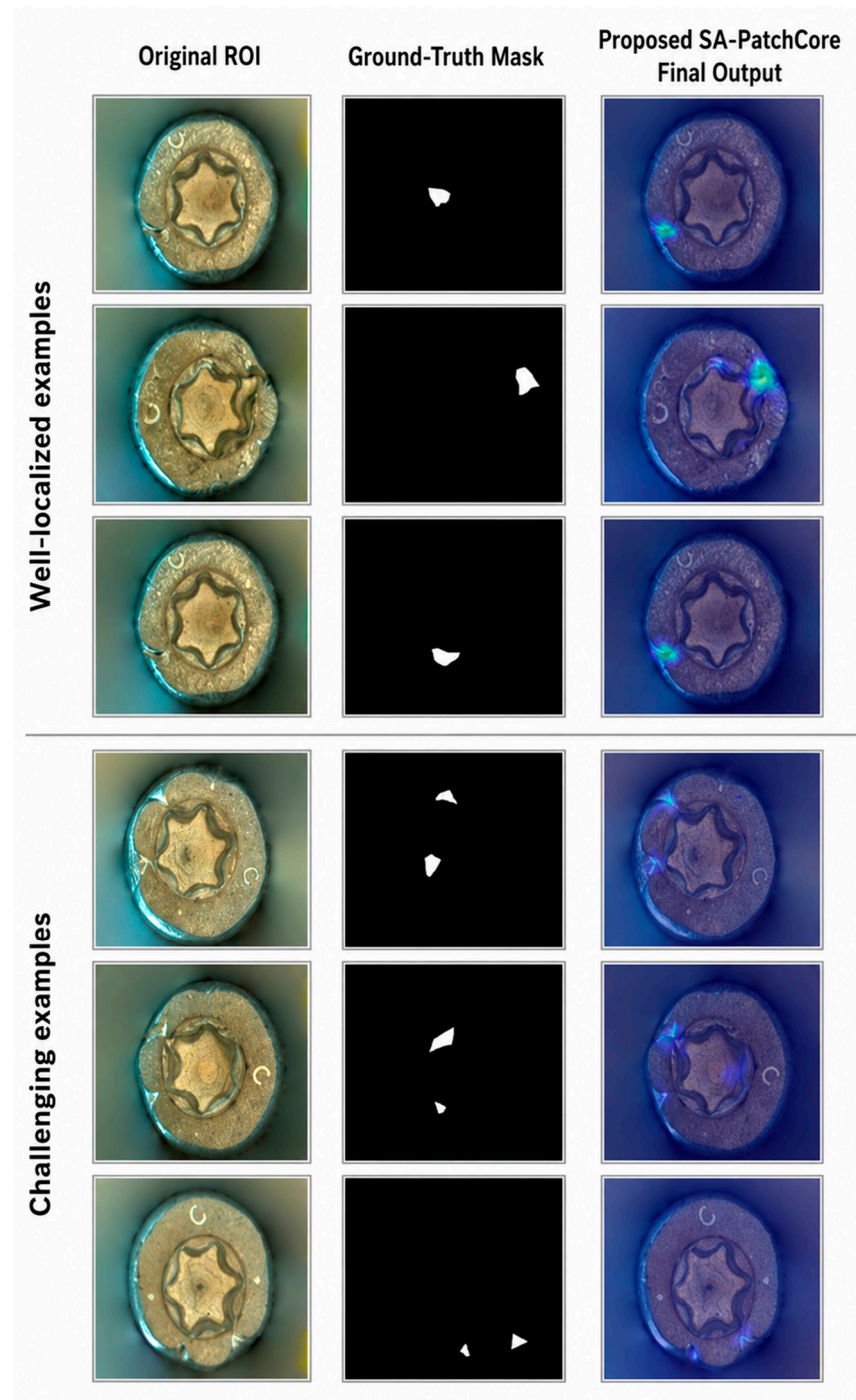


Figure 6. Qualitative examples of the proposed SA-PatchCore predictions on defective bolt-head samples. The first three rows show well-localized examples, while the last three rows show challenging cases. Each row includes the original ROI image, the manually annotated ground-truth mask, and the final anomaly-map overlay produced by the proposed SA-PatchCore model. The challenging examples illustrate cases where the anomaly response becomes spatially diffuse or only partially overlaps with the annotated defect region.

5.2. Confusion Matrix & Failure Analysis

To assess the reliability of the model for industrial quality control applications, the image-level and pixel-level prediction results were analyzed using a confusion matrix on a test set of 200 samples. The resulting confusion matrix is shown in Figure 7.

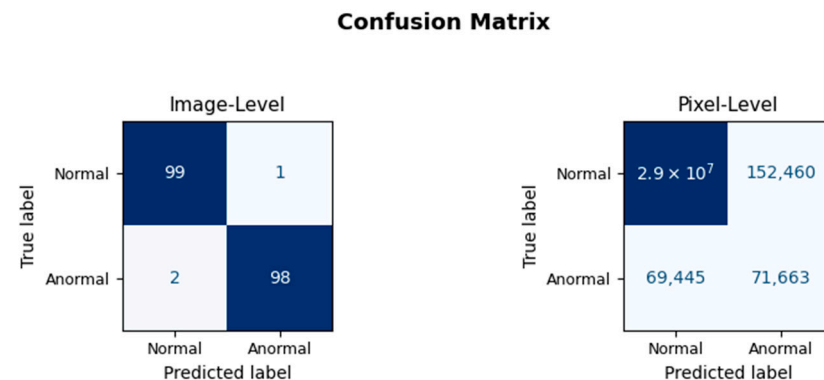


Figure 7. Confusion matrix.

When the image-level classification results were examined, the proposed SA-PatchCore model successfully detected 98 of the 100 defective samples (TP = 98) and missed only 2 defective samples (FN = 2). This result indicates that the model can detect defective samples with high sensitivity and provides strong defect detection capability for quality control scenarios. An examination of the misclassified samples showed that manufacturer stamps on the bolt surfaces and asymmetric circular die marks affected the model predictions.

In addition to the confusion matrix, the anomaly maps generated by SA-PatchCore were examined qualitatively to better understand the gap between the high Pixel-AUROC and the relatively lower Pixel-AP and Pixel-F1max values. Pixel-AUROC evaluates whether defective pixels tend to receive higher anomaly scores than non-defective pixels over all possible thresholds. Therefore, it can remain high even when the final high-score region is spatially diffuse or does not exactly match the manually annotated defect boundary. In contrast, Pixel-AP and Pixel-F1max are more sensitive to the precision of the localized region, false-positive responses around the defect, and the overlap between the predicted high-score area and the ground-truth mask. Table 9 summarizes the main failure modes observed in the anomaly-map outputs.

These observations indicate that the proposed SA-PatchCore configuration is effective at assigning higher anomaly scores to defective regions, as reflected by the high Pixel-AUROC. However, the anomaly maps do not always produce compact segmentation masks that perfectly match the manually annotated defect boundaries. Therefore, the lower Pixel-AP and Pixel-F1max values mainly reflect the difficulty of converting continuous anomaly-score maps into precise binary defect masks for very small and low-contrast bolt-head defects.

At the pixel level, the model correctly identified 71,663 anomalous pixels, while 152,460 normal pixels were incorrectly marked as anomalous. In addition, 69,445 anomalous pixels were missed, whereas approximately 29 million normal pixels were correctly classified. The pixel-level Recall value (50.79%) being higher than the Precision value (31.97%) indicates that the model tends to mark defect regions more comprehensively. In other words, it produces high anomaly scores that slightly extend beyond the actual defect boundaries. Although this tendency reduces pixel-level precision, it contributes to the spatial localization of defect regions. Since the Accuracy value (99.25%) may be misleading on its own due to class imbalance, Precision, Recall, and F1 should be evaluated together when interpreting the actual pixel-level performance. The relatively large number of false-positive pixels can be explained by the sensitivity of the proposed configuration to fine local texture variations. The higher 384×384 input resolution helps preserve subtle defect cues, but it also increases the visibility of small metallic texture changes,

manufacturer marks, slot edges, and reflection-related intensity variations. As a result, some normal surface details may receive elevated anomaly scores. In addition, the threshold used for Pixel-F1max is selected to maximize overlap on the annotated test masks and does not represent a fixed deployment threshold. This can lead to over-detection when continuous anomaly maps are converted into binary predictions. From an industrial perspective, this behavior means that the system may trigger more candidate alarms or require additional operator review if used with an overly sensitive threshold. Therefore, practical deployment should include threshold calibration according to the acceptable false-alarm rate, illumination control, and possible post-processing to suppress isolated or texture-related responses. Representative comparison results obtained from the test dataset are shown in Figure 8.

Table 9. Qualitative failure analysis of SA-PatchCore anomaly maps.

Failure Mode	Observation	Effect on Metrics
Very small defect regions	Some defects occupy only a very small part of the bolt-head ROI.	Pixel-AUROC may remain high, but Pixel-AP and Pixel-F1max decrease because a small number of false-positive pixels strongly affects precision.
Diffuse anomaly responses	The anomaly map sometimes highlights the neighborhood of the defect rather than only the annotated defect area.	The model ranks the defective region correctly, but the predicted region becomes larger than the ground-truth mask.
Boundary mismatch	The high-score region may be shifted slightly from the manually annotated defect boundary.	Pixel-level overlap decreases, reducing Pixel-F1max despite visually reasonable localization.
Low-contrast defect traces	Fine cracks or weak surface deformations may produce lower contrast anomaly responses.	The defect may be detected as an elevated anomaly region, but only part of the mask exceeds the effective threshold.
Responses on structural details	Manufacturer marks, slot edges, or reflective metallic texture may receive elevated anomaly scores.	These responses increase false-positive pixels and reduce PR-based metrics.

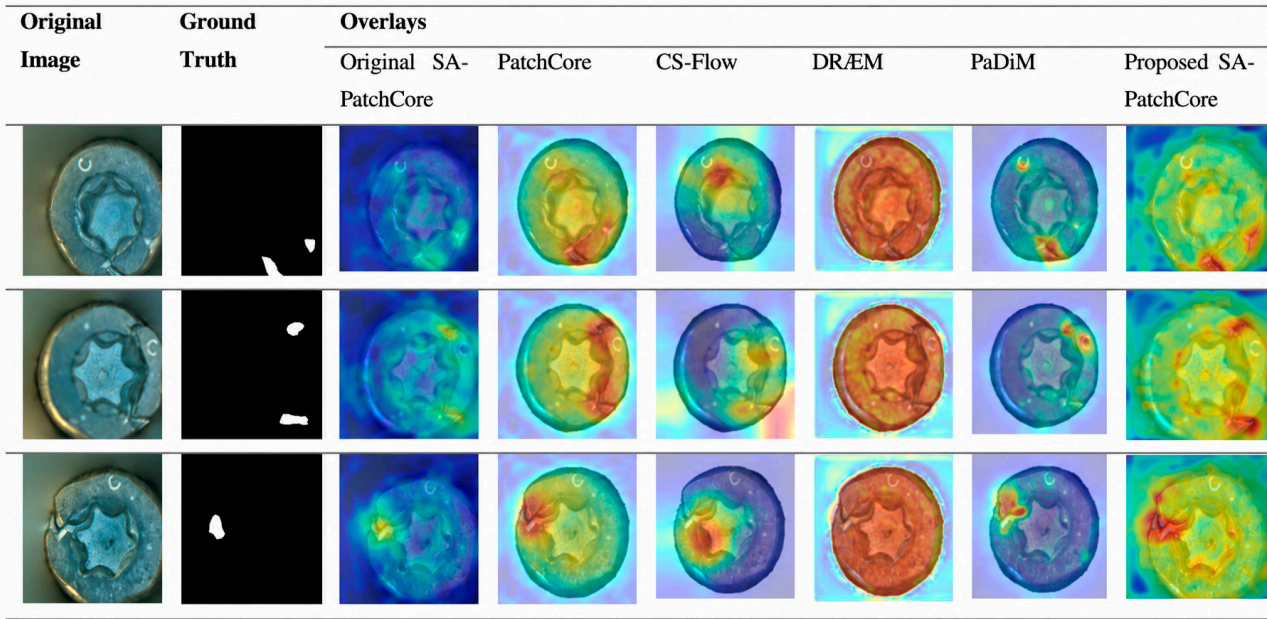


Figure 8. Model Prediction Examples. In the overlay maps, warmer colors indicate higher anomaly scores, whereas cooler/darker colors indicate lower anomaly scores. In the ground-truth masks, white regions indicate manually annotated defective areas.

5.3. Ablation Studies

A series of ablation experiments were conducted to isolate and examine the effects of the main design choices in the proposed SA-PatchCore architecture, namely, input resolution, feature extraction layer selection, and the neighborhood parameter, on the model's localization performance. In these experiments, all other hyperparameters were kept fixed, while only one independent variable was varied at a time. The results were reported using the bolt dataset.

5.3.1. The Effect of Input Resolution

Original PatchCore and many similar anomaly detection models commonly use a default input resolution of 224×224 pixels. However, since detecting defects such as hairline scratches on the bolt head surface requires the preservation of fine details, the proposed SA-PatchCore architecture was evaluated at higher input resolutions. As shown in Table 10, increasing the resolution from 224×224 to 384×384 improved Pixel-AUROC from 97.99% to 98.69%, Image-AUROC from 99.24% to 99.49%, and the Image-F1 score from 96.48% to 98.49%. Although the 320×320 resolution also produced better results than 224×224 , the highest performance was achieved at 384×384 . This indicates that aggressive downsampling at lower input resolutions may weaken the signals associated with subtle defects. Therefore, 384×384 was selected as the most suitable input resolution among the evaluated settings.

Table 10. Resolution Ablation.

Resolution	Pixel-AUROC	Image-AUROC	Image-F1
224×224 (Standard)	97.99%	99.24%	96.48%
320×320	98.63%	99.45%	97.98%
256×256	98.31%	99.12%	97%
384×384 (Recommended)	98.69%	99.49%	98.49%

In addition to the resolution, feature-layer, and neighborhood-parameter analyses, the effect of the coreset sampling ratio was also evaluated to further investigate the contribution of the memory-bank construction strategy.

5.3.2. The Effect of Feature Layers

The effect of different intermediate layer combinations on the proposed SA-PatchCore architecture was examined, and the results are presented in Table 11. In this context, four configurations were compared: Layer-3 only, Layer-2 only, Layer-2 + Layer-3, and Layer-1 + Layer-2.

Table 11. Ablation of Layer Combinations.

Layers	Pixel-AUROC	Image-AUROC	Image-F1
Layer-3 Only	98.02%	99.17%	98%
Layer-2 + Layer-3	98.69%	99.49%	98.49%
Layer-2 Only	98.90%	98.86%	96.52%
Layer-1 + Layer-2	98.61%	98.27%	95%

As shown in Table 11, the Layer-2 + Layer-3 combination provided the most balanced and overall best results in terms of Pixel-AUROC (98.69%), Image-AUROC (99.49%), and Image-F1 (98.49%). Although using only Layer-2 achieved the highest Pixel-AUROC value (98.90%), the lower Image-AUROC (98.86%) and Image-F1 (96.52%) scores indicate that this configuration is more limited in image-level discrimination.

Similarly, although the Layer-1 + Layer-2 combination preserved shallow-layer details, its overall performance remained below that of the Layer-2 + Layer-3 configuration. These findings show that Layer-2 + Layer-3 fusion is the most suitable feature-layer combination for reliably distinguishing bolt head defects at both the pixel and image levels. When the ablation results are interpreted together, the largest measurable gain relative to the standard PatchCore-style input setting is associated with increasing the input resolution from 224×224 to 384×384 . This change improved Pixel-AUROC from 97.99% to 98.69% and Image-F1 from 96.48% to 98.49%, indicating that preserving fine spatial details is important for subtle bolt-head defects. The feature-layer ablation shows a different but complementary effect. Although Layer-2 alone produced the highest Pixel-AUROC, it reduced image-level discrimination performance. In contrast, the Layer-2 + Layer-3 combination provided the best balance between pixel-level localization and image-level defect recognition. Therefore, the final configuration should be interpreted as the interaction of two design choices: higher input resolution preserves small defect patterns, while Layer-2/Layer-3 fusion combines local spatial detail with more semantic information to obtain stable performance across both pixel-level and image-level metrics.

5.3.3. Effect of k-NN Neighborhood Parameter

The effect of the number of neighbors k used in the k-NN algorithm for anomaly scoring in the embedding space was examined over the range $k \in \{1, 3, 5, 7, 9, 11, 15\}$. The results of these configurations are presented in Table 12.

Table 12. The effect of the k-NN neighbor count on model performance.

k	Pixel-AUROC	Image-AUROC	Image-F1
$k = 1$	98.5%	99.53%	98.49%
$k = 3$	98.69%	99.49%	98.49%
$k = 5$	98.75%	99.48%	98.49%
$k = 7$	98.78%	99.43%	98.49%
$k = 9$	98.78%	99.42%	98.49%
$k = 11$	98.78%	99.34%	98%
$k = 15$	98.77%	99.19%	98%

The ablation results should be interpreted as a sequential component analysis rather than as a complete factorial ablation study. The proposed SA-PatchCore configuration includes several coupled design choices, including higher-resolution ROI processing, Layer-2/Layer-3 feature fusion, local average pooling, neighborhood-aware k-NN scoring, Gaussian smoothing, and dynamic anomaly-map reshaping. Since these components interact with each other, the present ablation does not fully isolate all possible interaction effects among them. Instead, it provides an indicative analysis of how the main implementation and inference-stage modifications affect localization performance under the adopted evaluation protocol.

The results show that the model is generally robust to the k parameter. Pixel-AUROC varied between 98.50% and 98.78%, while Image-AUROC ranged from 99.19% to 99.53%. The lowest value, $k = 1$, produced the highest Image-AUROC score (99.53%), whereas the highest Pixel-AUROC value (98.78%) was obtained for $k = 7$ and $k = 9$. In contrast, the Image-F1 score remained largely stable at 98.49% over the range of $k = 1$ to $k = 9$.

These findings indicate that increasing the number of neighbors does not substantially change the overall model performance, but it introduces small trade-offs between pixel-level and image-level metrics. While a limited improvement was observed in Pixel-AUROC as k increased, Image-AUROC showed a slight decreasing trend. Since Image-F1 remained

stable across a wide range of k values, $k = 5$ was selected in the final model configuration as a balanced choice between high pixel-level localization performance and strong image-level discrimination performance.

5.3.4. Effect of Coreset Sampling Ratio

To further analyze the effect of the coreset memory strategy, the coreset sampling ratio was varied while keeping the remaining SA-PatchCore settings fixed. In this experiment, Layer-2/Layer-3 feature fusion, 384×384 input resolution, k -NN mean scoring with five neighbors, the same Gaussian smoothing parameter, and the same train–test split were used. The results are summarized in Table 13.

Table 13. Effect of coreset sampling ratio on SA-PatchCore performance.

Coreset Sampling Ratio	Image-AUROC (%)	Image-AP (%)	Image-F1 (%)	Pixel-AUROC (%)	Pixel-AP (%)	Pixel-F1max (%)
1%	98.84	99.23	97.49	98.43	26.49	36.59
5%	99.40	99.53	98.49	98.57	29.10	38.42
10%	99.27	99.48	98.49	98.69	29.73	39.24
20%	99.32	99.51	98.49	98.85	31.27	40.59

The results show that increasing the coreset sampling ratio generally improves pixel-level localization performance. In particular, Pixel-AP increased from 26.49% at a 1% coreset ratio to 31.27% at a 20% coreset ratio, while Pixel-F1max increased from 36.59% to 40.59%. This indicates that a larger coreset can preserve more representative normal feature vectors and improve the separation of defective pixel responses. However, increasing the coreset ratio also increases the memory-bank size and nearest-neighbor search cost. Therefore, the 10% coreset ratio used in the main configuration provides a practical balance between localization performance and computational cost.

5.4. Robustness Analysis Under Industrial Variations

To evaluate the robustness of the proposed SA-PatchCore model under realistic industrial variations, an additional stress-test analysis was conducted without retraining the model. The trained SA-PatchCore configuration was evaluated on perturbed versions of the masked defective test images. The perturbations were designed to simulate common production-line variations, including illumination changes, small positional shifts, image noise, and resolution changes. Pixel-level metrics were computed using the same ground-truth masks and raw anomaly scores as in the main evaluation. Table 14 summarizes the robustness results under these test-time variations.

Table 14. Robustness analysis of the proposed SA-PatchCore model under industrial variations.

Test Condition	Pixel-AUROC (%)	Pixel-AP (%)	Pixel-F1max (%)
Original test images	98.69	29.73	39.24
Illumination decrease (−20%)	98.71	30.46	39.97
Illumination increase (+20%)	98.47	27.81	38.55
Positional shift (+8 px)	98.69	29.83	38.92
Gaussian noise ($\sigma = 10$)	71.51	1.26	4.27
Resolution variation (320×320)	98.72	29.39	39.26

The robustness analysis shows how the model behaves when the test images deviate from the original acquisition conditions. The results indicate that the proposed SA-PatchCore configuration remains stable under moderate illumination changes, small

positional shifts, and resolution variation. This stability is consistent with the ROI standardization and resizing steps used before feature extraction. However, Gaussian noise caused a substantial decrease in both Pixel-AUROC and PR-based metrics, indicating that strong image noise can disrupt the local texture responses used by the feature extractor. These findings suggest that the proposed model is robust to moderate geometric, illumination, and resolution variations, while controlled image acquisition and noise reduction remain important for reliable industrial deployment.

5.5. Generalization Performance on the MVTec AD Dataset

To evaluate the general applicability of the proposed SA-PatchCore model in industrial quality control applications, it was tested on all 15 categories of the MVTec AD dataset. Table 15 presents a comparison of the pixel-level AUROC scores obtained by the proposed model with baseline methods in the literature.

Table 15. Class-Based Performance Comparison on the MVTec AD Dataset (Pixel-AUROC)

Model	PatchCore [8]	PaDiM [24]	DRÆM [7]	CS-Flow [48]	Proposed SA-PatchCore
Avg	98.1	97.5	97.3	94.6	98.2
Bottle	98.6	98.3	99.1	88.5	98.6
Cable	98.5	96.7	94.7	96.1	98.5
Capsule	98.9	98.5	94.3	98.1	99
Carpet	99.1	99.1	95.5	95.2	99
Grid	98.7	97.3	99.7	94.4	98.7
Hazeln.	98.7	98.2	99.7	96.1	98.9
Leather	99.3	99.2	98.6	89.3	99.4
M. Nut	98.4	97.2	99.5	96.0	98.6
Pill	97.6	95.7	97.6	95.8	98.3
Screw	99.4	98.5	97.6	98.3	99.3
Tile	95.9	94.1	99.2	90.7	96.6
Toothb.	98.7	98.8	98.1	97.4	99
Trans.	96.4	97.5	90.9	96.3	94.8
Wood	95.1	94.9	96.4	91.2	95.5
Zipper	98.9	98.5	98.8	95.8	98.8

Note: Since the CS-Flow paper does not provide pixel-level AUROC data, the outputs of the CS-Flow method are taken from Liao et al. [9].

The results on MVTec AD show that the proposed SA-PatchCore approach provides balanced and competitive performance across different categories. According to Table 15, the proposed method achieved an average Pixel-AUROC of 98.2%, which is numerically close to the PatchCore result of 98.1%. Since repeated-run experiments, confidence intervals, or bootstrap-based uncertainty estimates were not computed for the MVTec AD evaluation in this study, this small absolute difference should not be interpreted as a statistically significant improvement. Instead, the MVTec AD results indicate that the proposed configuration achieves performance comparable to strong industrial anomaly detection baselines while maintaining competitive pixel-level discrimination across diverse object and texture categories. In particular, the higher input resolution (384×384) and Layer-2/Layer-3 multi-scale feature fusion may help obtain clearer localization responses for small defects such as scratches and cracks. In this context, the proposed SA-PatchCore approach achieved Pixel-AUROC values of 99.3% for the screw category, 98.6% for the metal_nut category, and 98.8% for the zipper category, producing strong and competitive results in metallic hardware-like categories. When these three categories are considered together, the average Pixel-AUROC is approximately 98.9%, indicating that the proposed configuration remains effective on structural surfaces that require fine defect localization. Figure 9 presents a qualitative comparison of anomaly localization outputs on selected samples from different

categories of the MVTec AD dataset. This qualitative comparison should be interpreted as visual support for the category-level quantitative results. The figure shows whether the high anomaly-score regions produced by the proposed method are spatially consistent with the ground-truth masks across different object types. In particular, the examples illustrate that the proposed configuration can produce competitive localization responses on structural and metallic categories, while also showing that the compactness of the highlighted regions may vary across categories.

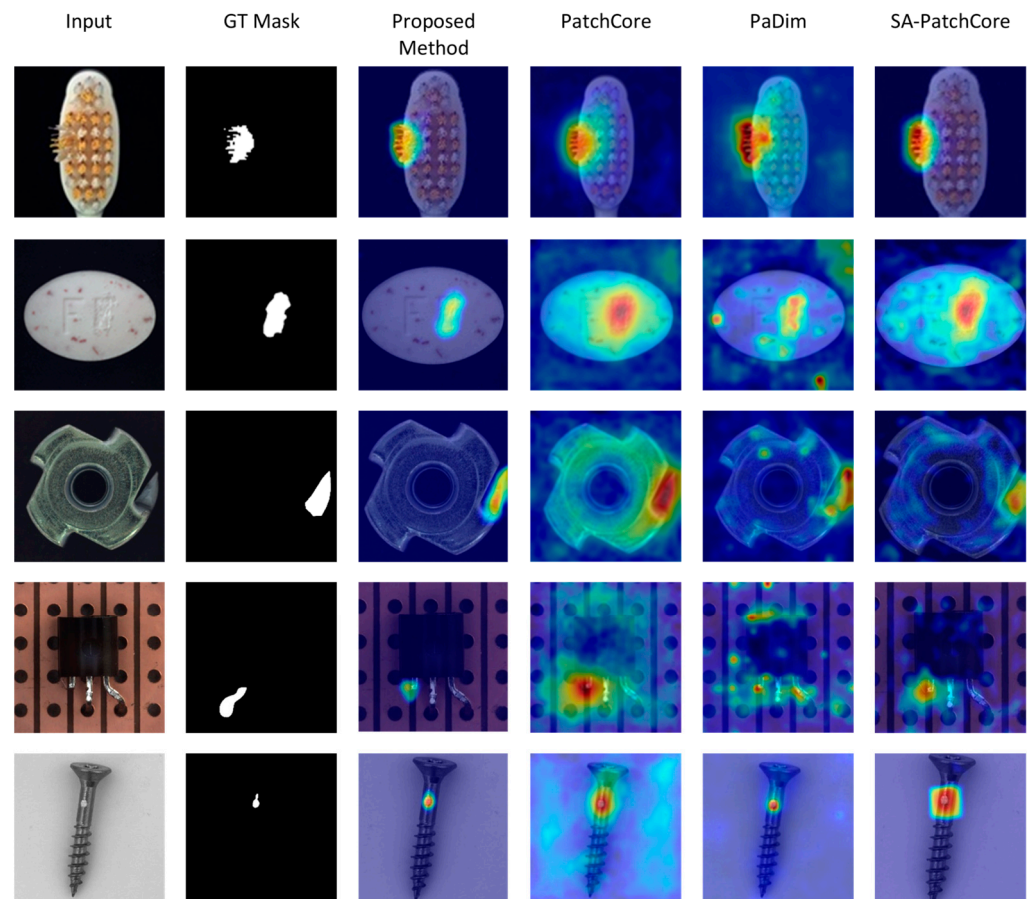


Figure 9. Qualitative comparison of anomaly localization outputs on selected samples from the MVTec AD dataset. From top to bottom, the rows correspond to the capsule, pill, metal nut, transistor, and screw categories, respectively. The columns show the input image, the ground-truth anomaly mask, and the anomaly localization maps generated by the compared methods. Highlighted regions indicate areas assigned high anomaly scores by each method. Warmer colors in the anomaly maps indicate higher anomaly scores, whereas cooler/darker colors indicate lower anomaly scores.

5.6. Natural Language Reporting and Hybrid Localization Performance Using VLM

In addition to pixel-level anomaly detection metrics, the output consistency, defect decision performance, and localization performance of the proposed VLM-based reporting framework were also evaluated. In this evaluation, three VLM settings were considered to separate the effects of the backbone model, supervised LoRA-based adaptation, and reward-based GRPO optimization. The first setting used the original Qwen3-VL-32B model with only the structured prompt and without task-specific training. The second setting used the same Qwen3-VL-32B backbone with LoRA-based supervised fine-tuning. The third setting used the GRPO-adapted final VLM model. This comparison was designed to evaluate whether the performance improvement comes from task-specific supervised adaptation alone or from the proposed reward-based GRPO optimization strategy. In this

way, the effect of the proposed GRPO-based adaptation on the output format, as well as its contribution to defect presence/absence decision-making and 3×3 location label generation, was examined.

The evaluation was performed using the structured JSON outputs generated for each test input. For each output, JSON validity, defect presence/absence decision, true-negative performance on non-defective samples, recall on defective samples, false-positive rate, false-negative rate and 3×3 location label accuracy were calculated. The test results were reported over 753 input records; however, this number was considered as the total number of evaluation inputs generated under different visual evidence conditions, not as the number of independent physical images. This distinction is particularly important because defective samples were evaluated under both overlay-supported and non-overlay conditions.

Since the VLM test records are imbalanced toward defective samples, the overall defect yes/no accuracy should not be interpreted as the only performance indicator. For this reason, the evaluation also reports normal sample accuracy, defective-sample recall, false-positive rate, and false-negative rate. Similarly, location performance is reported both over all input records and over defective records only, because non-defective samples have the target location label none.

The GRPO model was trained for 200 steps, and a checkpoint was saved every 50 steps. In the final stage, the checkpoints at steps 50, 100, 150, and 200 were tested, and the checkpoint at step 150, which achieved the best performance, was selected as the final model. The baseline Qwen3-VL-32B model, the LoRA-SFT model and the final VLM model adapted with GRPO were evaluated under the same test protocol. The detailed quantitative results of this comparison are reported in Table 16, while the training-strategy ablation including the LoRA-SFT baseline is summarized in Table 17.

Table 16. Quantitative comparison of baseline Qwen3-VL-32B and the GRPO-adapted final VLM model on the test set.

Metric	Baseline Qwen3-VL-32B	GRPO-Adapted Final VLM
Number of evaluated input records	753	753
Valid JSON output rate	100.00%	100.00%
Defect yes/no accuracy	76.63%	96.41%
Normal sample accuracy	91.43%	98.10%
Defect recall	74.23%	96.14%
Blank evidence accuracy, all records	89.51%	96.04%
Blank defect recall	88.89%	95.37%
Overlay defect recall	59.57%	96.91%
Exact location match, all records	28.15%	53.78%
Mean partial location score, all records	32.29%	65.34%
Exact location match, defect records only	17.90%	46.60%
Mean partial location score, defect records only	22.71%	60.03%
False-positive rate	8.57%	1.90%
False-negative rate	25.77%	3.86%

The training-strategy ablation shows that LoRA-SFT improved several reporting metrics compared with the prompt-only Qwen3-VL-32B baseline, especially for defective-sample recall and location reporting. However, the LoRA-SFT model remained clearly below the GRPO-adapted model in defect-decision consistency. The defect yes/no accuracy increased from 76.63% in the original model to 81.67% with LoRA-SFT, but further increased to 96.41% with GRPO adaptation. Similarly, the defective-sample recall increased from 74.23% to 81.79% with LoRA-SFT and to 96.14% with GRPO. The LoRA-SFT model also produced unparseable defect-field outputs in 4.12% of the test records, whereas the GRPO-

adapted model maintained parseable structured outputs for all test cases. These results indicate that supervised LoRA adaptation alone improves task alignment, but the proposed GRPO optimization provides a stronger contribution to reliable defect decisions, valid structured reporting, and 3×3 location consistency.

Table 17. Training-strategy ablation of the VLM-based reporting module.

Metric	Original Qwen3-VL-32B	LoRA-SFT Qwen3-VL-32B	GRPO-Adapted Final VLM
Number of evaluated input records	753	753	753
Parseable defect-field rate (%)	100.00	95.88	100.00
Defect yes/no accuracy (%)	76.63	81.67	96.41
Normal sample accuracy (%)	91.43	80.95	98.10
Defect recall (%)	74.23	81.79	96.14
False-positive rate (%)	8.57	1.90	1.90
False-negative rate (%)	25.77	16.20	3.86
Exact location match, all records (%)	28.15	48.61	53.78
Mean partial location score, all records (%)	32.29	58.76	65.34
Exact location match, defect records only (%)	17.90	40.59	46.60
Mean partial location score, defect records only (%)	22.71	52.39	60.03

To further clarify the contribution of each component in the VLM-based reporting module, an additional component-level comparison was derived from the evidence-panel evaluation. The comparison separates three settings: image-only prompting, image plus anomaly-map prompting, and image plus anomaly-map prompting with GRPO adaptation. The results are summarized in Table 18.

Table 18. Component-level ablation of the VLM-based reporting module.

Setting	Visual Input	Task-Specific Adaptation	Defect Recall (%)	Main Interpretation
Image-only + prompt	Original image + blank evidence panel	No GRPO	88.89	Baseline model can use visual image cues but remains unstable
Image + anomaly map + prompt	Original image + anomaly overlay	No GRPO	59.57	Overlay alone does not reliably improve the non-adapted VLM
Image + anomaly map + GRPO	Original image + anomaly overlay	GRPO	96.91	GRPO strongly improves use of visual evidence and defect decision

The ablation results show that simply adding the anomaly overlay to the input does not guarantee better reporting performance for the non-adapted VLM. In the baseline Qwen3-VL-32B setting, the overlay-supported recall was lower than the image-only recall, indicating that the model could misinterpret the anomaly overlay without task-specific adaptation. In contrast, the GRPO-adapted model achieved the highest defective-sample recall when both the original image and anomaly overlay were provided. This indicates that GRPO adaptation helps the VLM use the anomaly-map evidence more effectively and generate more consistent defect reports.

In the direct comparison between the prompt-only baseline and the GRPO-adapted final model, both models produced valid JSON outputs for all test inputs, achieving a JSON validity rate of 100%. This indicates that the structured prompt design was able to control the output format to a certain extent. However, a clear difference was observed between the two models in terms of defect decision performance. While the defect presence/absence accuracy of the baseline model was 76.63%, this value increased to 96.41% in the VLM model adapted with GRPO. Similarly, recall for defective samples increased from 74.23% to 96.14%, while the false-negative rate decreased from 25.77% to 3.86%. These results show that task-specific adaptation with GRPO substantially reduced missed defective samples.

The evaluation based on the evidence panel condition provides a more detailed view of the model behavior. Under the blank-evidence condition, the trained model achieved an accuracy of 96.04% across all input records. When only defective samples without overlay support were considered, the defect recall was 95.37%. For defective samples containing an overlay evidence panel, the GRPO-adapted model achieved a recall of 96.91%. The small difference between these two defective test conditions indicates that the GRPO-adapted model did not rely solely on the anomaly overlay map in the right panel when making its decision. In other words, even without overlay support, the model was largely able to maintain its defect decision based on the original image in the left panel. This finding supports the idea that presenting defective samples under both overlay-supported and non-overlay evidence conditions during training reduced the model's overdependence on a single visual cue.

In the location evaluation, the location_3 $\times$ 3 field generated by the model was compared with the reference location label. Location metrics were reported both over all input records and only over defective samples. Since the target location is none for non-defective samples, the evaluation conducted only on defective samples provides a stricter measure of localization performance. When all records were considered, the exact location match increased from 28.15% in the baseline model to 53.78% in the trained model. The mean partial location score also increased from 32.29% to 65.34%. When only defective samples were considered, the exact location match increased from 17.90% to 46.60%, while the mean partial location score increased from 22.71% to 60.03%. These results show that GRPO adaptation substantially improved not only the defect presence/absence decision, but also the generation of regional location labels.

However, localization performance remained lower than defect presence/absence decision performance. In some cases, the model correctly detected the presence of a defect but reported a neighboring region instead of the target 3 $\times$ 3 location label. This issue becomes more apparent when the defect is small, low-contrast, or close to grid boundaries. Therefore, while exact location match provides a stricter measure, the mean partial location score better reflects predictions that are close to or partially overlap with the reference region.

To further evaluate the generated textual reports beyond defect classification and 3 $\times$ 3 localization, an additional report-level analysis was performed on the explanation field of the structured JSON outputs. Since the reporting task requires short and operationally useful explanations rather than open-ended captions, the generated explanations were evaluated using rule-based quality checks. These checks measured whether the explanation field was non-empty, concise, consistent with the predicted defect decision, consistent with the predicted regional location when a location was provided, and free from unsupported pixel-coordinate or production-line claims. Table 19 summarizes the report-level explanation quality criteria used in this analysis.

This analysis complements the defect-decision and localization metrics by evaluating whether the generated report is usable as a structured inspection output. The results showed that the GRPO-adapted model produced valid and concise report fields more consistently than the baseline model. In particular, the GRPO-adapted outputs were less likely to contain unsupported extra statements and were more consistent with the predicted defect decision and location fields. Therefore, the improvement provided by GRPO was not limited to classification and localization metrics; it also improved the practical consistency of the generated inspection reports.

In addition to the quantitative results, the model outputs were also examined qualitatively on a sample basis. For this purpose, the composite input image, target label, model-generated JSON output, JSON validity, defect decision accuracy, location match

status, partial location score, and error types were saved in CSV format for each test input. In addition, visual reports showing the composite input image together with the model prediction were generated for each sample. Table 20 presents examples from overlay-supported defective, blank-evidence defective, and blank-evidence non-defective cases. In these examples, the baseline Qwen3-VL-32B model produced false-negative or false-positive outputs, whereas the GRPO-adapted model corrected the defect decision and generated the expected regional label. These examples illustrate that GRPO adaptation improves defect-decision consistency and structured location reporting under different evidence conditions.

Table 19. Report-level evaluation criteria for VLM-generated explanations.

Criterion	Description
Non-empty explanation	The explanation field must contain a textual reason for the generated decision.
Conciseness	The explanation should be short and inspection-oriented, without unnecessary repetition.
Decision consistency	The explanation should be consistent with the predicted defect status. For example, a defect = yes output should describe a visible abnormality, while a defect = no output should not describe a defect.
Location consistency	When a regional location is predicted, the explanation should not contradict the predicted 3×3 location label.
No unsupported detail	The explanation should not introduce unsupported pixel coordinates, mechanical measurements, or production-line actions that were not requested by the prompt.

Table 20. Qualitative comparison of baseline Qwen3-VL-32B and GRPO-adapted outputs.

Sample	Evidence Condition	Target	Baseline Qwen3-VL-32B Output	Trained Model Output	Interpretation
	Overlay	defect = yes, location = top	defect = no, location = none	defect = yes, location = top	FN corrected + exact location
	Blank	defect = yes, location = right	defect = no, location = none	defect = yes, location = right	No-overlay defect recovered
	Blank/normal	defect = no, location = none	defect = yes, location = center	defect = no, location = none	False positive corrected

5.7. Computational Analysis and Industrial Feasibility

To evaluate the direct runtime performance of the SA-PatchCore-based defect detection module, an inference FPS analysis was conducted on 100 defective test ROI images. The trained model weights were used, and a total of 100 inference operations were performed. The total inference time was 3.0545 s, corresponding to an overall throughput of 32.74 FPS. The average inference latency was 30.54 ms per image, with a median latency of 30.47 ms. The minimum and maximum latencies were 29.74 ms and 32.14 ms, respectively.

These results show that the SA-PatchCore module satisfies the 30 FPS real-time inspection target at the direct model-inference level, independently of camera acquisition or graphical user interface overhead. Therefore, the proposed anomaly detection stage can be considered feasible for online ROI-level inspection in an industrial setting.

To reduce the memory burden of the VLM stage, Qwen3-VL-32B was used with 4-bit quantization and LoRA-based adaptation. However, since it remains a large-scale vision-language model, it is not intended to be executed for every incoming frame in a real-time production-line loop. Instead, the VLM is used as an asynchronous reporting component that can be triggered after the SA-PatchCore module detects a defective or uncertain sample. In this deployment scenario, SA-PatchCore performs the real-time inspection, while the VLM converts selected anomaly-map evidence into structured JSON reports for operator support and traceability.

6. Discussion

The framework proposed in this study addresses both pixel-level defect localization and the need for structured reporting for end users in bolt head images. Therefore, the system should not be considered only as an anomaly detection model, but rather as a multi-stage quality control workflow that integrates anomaly map generation with VLM-based natural language reporting.

When the pixel-level results are examined, the proposed SA-PatchCore configuration achieves strong discrimination performance on the bolt dataset, with a Pixel-AUROC value of 98.69%. This result indicates that the 384×384 input resolution and Layer-2/Layer-3 multi-scale feature fusion are effective for improving the discrimination of subtle defects, while dynamic feature-map sizing mainly supports resolution-adaptive anomaly-map reconstruction and stable inference. However, the Pixel-AP (29.73%) and Pixel-F1max (39.24%) values remained lower than Pixel-AUROC. This can be attributed to the fact that defect pixels occupy only a very small portion of the image area and that PR-based metrics are more sensitive to class imbalance. Therefore, the results should be interpreted not only through AUROC, but also together with AP and F1 values.

In the comparative results, the proposed SA-PatchCore model achieved the highest Pixel-AUROC value. However, PatchCore obtained higher scores for the Pixel-AP and Pixel-F1max metrics. This indicates that the proposed method has strong overall anomaly discrimination capability, while there is still room for improvement in producing more precise pixel-level boundaries. In particular, for small or low-contrast defects, or defects that blend with the surface texture, anomaly scores extending beyond the actual defect boundaries may reduce the Precision value.

A further limitation of this study is that the baseline methods were evaluated using their default Anomalib configurations, whereas the proposed SA-PatchCore includes task-specific architectural and inference-stage design choices. Since the compared methods belong to different architectural families and have different hyperparameter spaces, a fully identical tuning protocol across all methods was not applied. Therefore, the baseline comparison should be interpreted as a controlled reference comparison under a shared dataset, preprocessing, input resolution, and evaluation protocol, rather than as an ex-

haustive hyperparameter-optimized benchmark. In addition, detailed runtime, memory consumption, coreset size, and inference-speed benchmarking across all methods was not the main focus of this study and will be considered in future work.

Another limitation concerns the ablation analysis of the proposed SA-PatchCore configuration. The current ablation study was designed as a sequential component analysis and does not constitute a complete factorial evaluation of all architectural and post-processing choices. In particular, the individual and interaction effects of input resolution, Layer-2/Layer-3 fusion, local pooling, neighborhood-aware scoring, Gaussian smoothing, and dynamic anomaly-map reshaping were not exhaustively evaluated with separate runtime, memory, and inference-speed measurements for each configuration. Therefore, the ablation results should be interpreted as an indicative analysis of the main design choices rather than a complete decomposition of their independent contributions. A more extensive factorial ablation with computational-cost reporting will be considered in future work.

The ablation experiments support the contribution of the main design choices in the proposed architecture to model performance. The 384×384 input resolution enabled small surface defects to be represented more clearly, while Layer-2 + Layer-3 fusion provided a more balanced structure between spatial detail and semantic representation. The close results obtained for different k-NN neighborhood parameters show that the method is not overly dependent on a single parameter value.

The second important aspect of this study is that the anomaly detection outputs are supported by VLM-based structured reporting. Although the baseline Qwen3-VL-32B model was able to preserve the valid JSON format, it remained limited in defect decision-making. The final VLM model adapted with GRPO increased defect presence/absence accuracy from 76.63% to 96.41% and recall for defective samples from 74.23% to 96.14%. The reduction in the false-negative rate from 25.77% to 3.86% is important because it reduces the risk of defective parts being classified as non-defective.

The results obtained according to the evidence panel condition show that the final VLM model does not rely solely on the anomaly overlay map in the right panel. The recall value was 96.91% for defective samples with overlay evidence and 95.37% for defective samples without overlay evidence. These close values indicate that using both overlay-supported and non-overlay training samples reduces the model's overdependence on a single visual cue.

However, localization performance remained a more challenging task than defect presence/absence decision-making. Across all records, the exact location match increased from 28.15% to 53.78%, while the mean partial location score increased from 32.29% to 65.34%. However, in some samples, although the model correctly detected the defect, it reported a neighboring region instead of the target 3×3 location label. This is an expected limitation, especially for small or low-contrast defects, or defects located close to grid boundaries.

Overall, the proposed approach does not leave the anomaly detection output only as a numerical score or heatmap. Instead, it transforms the output into a more interpretable report that includes defect presence, a regional location label, and a brief explanation. However, VLM-based reporting produces a 3×3 regional location label and does not directly replace a precise pixel-level mask output. In addition, since the use of a large-scale VLM requires substantial computational resources, it appears more suitable to run this module for reporting after anomaly detection, rather than on every frame in real-time production line applications.

7. Conclusions

This study presents a multi-stage industrial anomaly detection workflow for pixel-level defect localization and structured natural language reporting in bolt head images.

The proposed SA-PatchCore approach is built on an unsupervised framework that learns the normal appearance only from non-defective samples, while defective samples are used to evaluate the model's pixel-level localization and decision performance. This structure provides a practical solution for industrial scenarios in which collecting labeled defective data is challenging.

Within the SA-PatchCore framework, Layer-2/Layer-3 multi-scale feature fusion, 384×384 input resolution, 3×3 average pooling, k-Center Greedy-based coresets selection, and dynamic feature-map sizing were used together. On the bolt dataset, the method achieved Pixel-AUROC, Pixel-AP, and oracle Pixel-F1max values of 98.69%, 29.73%, and 39.24%, respectively. The results show that the proposed method provides strong discrimination performance, particularly in terms of Pixel-AUROC. However, the PR-based metrics indicate that there is still room for improvement due to small defect regions and class imbalance. The average Pixel-AUROC value of 98.2% obtained on the MVTec AD dataset further indicates that the proposed configuration provides competitive performance across different industrial categories. For the bolt dataset, the bootstrap confidence intervals further support the improvement of the proposed configuration over the original SA-PatchCore setting. However, small numerical differences between strong baselines should be interpreted cautiously when confidence intervals overlap.

The second contribution of this study is the integration of anomaly detection outputs with VLM-based structured reporting. For this purpose, the Qwen3-VL-32B model was adapted with GRPO for a reporting task that jointly uses the original image and anomaly overlay evidence. The final VLM model generated the defect presence, 3×3 location label, and brief description fields in a valid JSON format. Compared with the baseline Qwen3-VL-32B model, defect presence/absence accuracy increased from 76.63% to 96.41%, while recall for defective samples increased from 74.23% to 96.14%. In addition, exact location match increased from 28.15% to 53.78% across all records, and the mean partial location score increased from 32.29% to 65.34%.

Overall, the results show that the proposed multi-stage framework is not limited to producing only anomaly scores or heatmaps. Instead, it can generate a more interpretable industrial inspection output by presenting the defect decision, regional location information, and a brief explanation together. However, location estimation remains a more challenging task than defect presence/absence decision-making. In future studies, the localization accuracy and production-line applicability of the system can be improved through more detailed location labels, pixel-level mask-supported VLM training, a wider variety of non-defective samples, and more lightweight VLMs.

Author Contributions: Conceptualization, O.E. and E.G.; methodology, O.E.; software, O.E.; validation, O.E., E.G. and İ.A.; formal analysis, O.E. and E.A.; investigation, O.E.; resources, E.G., İ.A. and E.A.; data curation, O.E.; writing—original draft preparation, O.E.; writing—review and editing, E.G., İ.A. and E.A.; visualization, O.E.; supervision, E.G., İ.A. and E.A.; project administration, E.G. and İ.A.; funding acquisition, E.G. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by Firat University with FUBAP-MF.25.53.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to project-related restrictions and because the dataset was generated within the scope of an ongoing industrial inspection research project.

Acknowledgments: This study was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), under Project No. 125E031.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Dubiel, T.; Balawender, T. Discontinuity Defects of the Bolt Head Bearing Surface. *Eng. Fail. Anal.* **2022**, *136*, 106140. [\[CrossRef\]](#)
2. Luo, Q.; Fang, X.; Liu, L.; Yang, C.; Sun, Y. Automated Visual Defect Detection for Flat Steel Surface: A Survey. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 626–644. [\[CrossRef\]](#)
3. Hütten, N.; Gomes, M.; Hölken, F.; Andricevic, K.; Meyes, R.; Meisen, T. Deep Learning for Automated Visual Inspection in Manufacturing and maintenance: A Survey of Open-Access Papers. *Appl. Syst. Innov.* **2024**, *7*, 11. [\[CrossRef\]](#)
4. Ameri, R.; Hsu, C.-C.; Band, S.S. A Systematic Review of Deep Learning Approaches for Surface Defect in Industrial Applications. *Eng. Appl. Artif. Intell.* **2024**, *130*, 107717. [\[CrossRef\]](#)
5. Tang, B.; Chen, L.; Sun, W.; Lin, Z.-K. Review of Surface Defect Detection of Steel Products Based on machine Vision. *IET Image Process.* **2023**, *17*, 303–322. [\[CrossRef\]](#)
6. Li, Z.; Yan, Y.; Wang, X.; Ge, Y.; Meng, L. A Survey of Deep Learning for Industrial Visual Anomaly Detection. *Artif. Intell. Rev.* **2025**, *58*, 279. [\[CrossRef\]](#)
7. Zavrtanik, V.; Kristan, M.; Skočaj, D. DRÆM—A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 8310–8319. [\[CrossRef\]](#)
8. Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; Gehler, P. Towards Total Recall in Industrial Anomaly Detection. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 14298–14308.
9. Liao, L.; Lu, C.; Gao, Y.; Yu, H.; Cai, B. Gcaflow: Multi-Scale Flow-Based Model with Global Context-Aware Channel Attention for Industrial Anomaly Detection. *Sensors* **2025**, *25*, 3205. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Zhang, X.; Xu, M.; Zhou, X. Towards High-Resolution Industrial Image Anomaly Detection. *arXiv* **2025**, arXiv:2508.12931.
11. Gu, Z.; Zhu, B.; Zhu, G.; Chen, Y.; Tang, M.; Wang, J. AnomalyGPT: Detecting Industrial Anomalies Using Large Vision-Language Models. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 1932–1940. [\[CrossRef\]](#)
12. Zhang, Z.; Ruan, J.; Gao, X.; Liu, T.; Fu, Y. EIAD: Explainable Industrial Anomaly Detection Via Multi-Modal Large Language Models. In Proceedings of the 2025 IEEE International Conference on Multimedia and Expo (ICME), Nantes, France, 30 June–4 July 2025. [\[CrossRef\]](#)
13. Cai, W.; Huang, W.; Cao, Y.; Huang, C.; Yuan, F.; Zhang, B.; Wen, J. Towards VLM-Based Hybrid Explainable Prompt Enhancement for Zero-Shot Industrial Anomaly Detection. In Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 16–22 August 2025; pp. 711–719.
14. Ishida, K.; Takena, Y.; Nota, Y.; Mochizuki, R.; Matsumura, I.; Ohashi, G. SA-PatchCore: Anomaly Detection in Dataset with Co-Occurrence Using Self-Attention. *IEEE Access* **2023**, *11*, 3232–3240. [\[CrossRef\]](#)
15. He, Y.; Li, S.; Wen, X.; Xu, J. A Survey on Surface Defect Inspection Based on Generative Models in Manufacturing. *Appl. Sci.* **2024**, *14*, 6774. [\[CrossRef\]](#)
16. Ma, Y.; Yin, J.; Huang, F.; Li, Q. Surface Defect Inspection of Industrial Products with Object Deep Networks: A Systematic Review. *Artif. Intell. Rev.* **2024**, *57*, 333. [\[CrossRef\]](#)
17. Jin, Q.; Chen, L. A Survey of Surface Defect Detection of Industrial Products Based on A Small Number of Labeled Data. *arXiv* **2022**, arXiv:2203.05733. [\[CrossRef\]](#)
18. Jiang, X.; Liu, J.; Wang, J.; Nie, Q.; Wu, K.; Liu, Y.; Wang, C.; Zheng, F. SoftPatch: Unsupervised Anomaly Detection with Noisy Data. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 15433–15445. [\[CrossRef\]](#)
19. Bae, J.; Lee, J.-H.; Kim, S. PNI: Industrial Anomaly Detection Using Position and Neighborhood Information. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 6373–6383.
20. Batzner, K.; Heckler, L.; König, R. EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies. In Proceedings of the 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2024; pp. 127–137.
21. Chen, W.-Y.; Tsao, Y.-R.; Lai, J.-Y.; Hung, C.-J.; Liu, Y.-C.; Liu, C.-Y. Real-Time Instance Segmentation of Metal Screw Defects Based on deep Learning Approach. *Meas. Sci. Rev.* **2022**, *22*, 107–111. [\[CrossRef\]](#)
22. Karbouj, B.; Topalian-Rivas, G.A.; Krüger, J. Comparative Performance Evaluation of One-Stage and Two-Stage Detectors for Screw Head Detection and Classification in disassembly Processes. *Procedia CIRP* **2024**, *122*, 527–532. [\[CrossRef\]](#)

23. Chae, J.; Luo, L.; Liu, Y.; Wang, R.; Yu, D.; Liang, Z.; Yuan, X.; Zhang, D.; Chen, Z.; Qin, P. StructCore: Structure-Aware Image-Level Scoring for Training-Free Unsupervised Anomaly Detection. *arXiv* **2026**, arXiv:2602.17048.
24. Defard, T.; Setkov, A.; Loesch, A.; Audigier, R. PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. In *International Conference on Pattern Recognition*; Springer: Cham, Switzerland, 2020; pp. 475–489. [[CrossRef](#)]
25. Li, C.L.; Sohn, K.; Yoon, J.; Pfister, T. CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. In Proceedings of the 2021 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 9659–9669. [[CrossRef](#)]
26. Gudovskiy, D.; Ishizaka, S.; Kozuka, K. CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows. In Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022; pp. 1819–1828. [[CrossRef](#)]
27. Hyun, J.; Kim, S.; Jeon, G.; Kim, S.H.; Bae, K.; Kang, B.J. ReConPatch: Contrastive Patch Representation Learning for Industrial Anomaly Detection. In Proceedings of the 2024 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2024; pp. 2041–2050. [[CrossRef](#)]
28. Bergmann, P.; Fauser, M.; Sattlegger, D.; Steger, C. MVTEC Ad-A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In Proceedings of the 2019 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9584–9592. [[CrossRef](#)]
29. Zou, Y.; Jeong, J.; Pemula, L.; Zhang, D.; Dabeer, O. Spot-the-Difference Self-Supervised Pre-Training for Anomaly Detection and Segmentation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 392–408. [[CrossRef](#)]
30. Cohen, N.; Hoshen, Y. Sub-Image Anomaly Detection with Deep Pyramid Correspondences. *arXiv* **2020**, arXiv:2005.02357.
31. Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; Dabeer, O. Winclip: Zero-/Few-Shot Anomaly Classification and Segmentation. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 19606–19616.
32. Zhou, Q.; Pang, G.; Tian, Y.; He, S.; Chen, J. AnomalyCLIP: Object-Agnostic Prompt Learning for Zero-Shot Anomaly Detection. *Int. Conf. Learn. Represent.* **2024**, 2024, 49705–49737. [[CrossRef](#)]
33. Zhao, Y.; Meng, M.; Wu, J. AnomalyLVM: Vision-Language Models for Zero-Shot Anomaly Detection. *Expert Syst. Appl.* **2026**, 312, 131392. [[CrossRef](#)]
34. Li, X.; Zhang, Z.; Tan, X.; Chen, C.; Qu, Y.; Xie, Y.; Ma, L. PromptAD: Learning Prompts with Only Normal Samples for Few-Shot Anomaly Detection. In Proceedings of the 2024 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16848–16858. [[CrossRef](#)]
35. Cao, Y.; Zhang, J.; Frittoli, L.; Cheng, Y.; Shen, W.; Boracchi, G. AdaCLIP: Adapting CLIP with Hybrid Learnable Prompts for Zero-Shot Anomaly Detection. In *European Conference on Computer Vision; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2025; pp. 55–72. [[CrossRef](#)]
36. Gu, Z.; Zhu, B.; Zhu, G.; Chen, Y.; Li, H.; Tang, M.; Wang, J. FiLo: Zero-Shot Anomaly Detection by Fine-Grained Description and High-Quality Localization. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, VIC, Australia, 28 October–1 November 2024; pp. 2041–2049. [[CrossRef](#)]
37. Bai, S.; Cai, Y.; Chen, R.; Chen, K.; Chen, X.; Cheng, Z.; Deng, L.; Ding, W.; Gao, C.; Ge, C. Qwen3-VL Technical Report. *arXiv* **2025**, arXiv:2511.21631.
38. Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Zhang, P.; Li, Y.; Liu, Z.; et al. LLaVA-OneVision: Easy Visual Task Transfer. *arXiv* **2024**, arXiv:2408.03326. [[CrossRef](#)]
39. Zhu, J.; Wang, W.; Chen, Z.; Liu, Z.; Ye, S.; Gu, L.; Tian, H.; Duan, Y.; Su, W.; Shao, J.; et al. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. *arXiv* **2025**, arXiv:2504.10479. [[CrossRef](#)]
40. Wu, Z.; Chen, X.; Pan, Z.; Liu, X.; Liu, W.; Dai, D.; Gao, H.; Ma, Y.; Wu, C.; Wang, B.; et al. DeepSeek-VL2: Mixture-of-Experts Vision-Language Models for Advanced Multimodal Understanding. *arXiv* **2024**, arXiv:2412.10302. [[CrossRef](#)]
41. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. In Proceedings of the 2016 British Machine Vision Conference (BMVC), York, UK, 19–22 September 2016. [[CrossRef](#)]
42. Han, D.; Han, M.; Unsloth Team. Unsloth, Version v0.1.41-Beta; Unsloth AI, 2026. Available online: <https://github.com/unslothai/unsloth> (accessed on 21 May 2026).
43. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. Qlora: Efficient Finetuning of Quantized LLMs. *Adv. Neural Inf. Process. Syst.* **2023**, 36, 10088–10115. [[CrossRef](#)]
44. Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.K.; Wu, Y.; et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv* **2024**, arXiv:2402.03300. [[CrossRef](#)]
45. Torralba, A.; Russell, B.C.; Yuen, J. LabelMe: Online Image Annotation and Applications. *Proc. IEEE* **2010**, 98, 1467–1484. [[CrossRef](#)]

46. Akcay, S.; Ameln, D.; Vaidya, A.; Lakshmanan, B.; Ahuja, N.; Genc, U. Anomalib: A Deep Learning Library for Anomaly Detection. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 1706–1710. [[CrossRef](#)]
47. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
48. Rudolph, M.; Wehrbein, T.; Rosenhahn, B.; Wandt, B. Fully Convolutional Cross-Scale-Flows for Image-Based Defect. In Proceedings of the 2022 IEEE/CVF winter conference on applications of computer vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 1088–1097.
49. Rolih, B.; Fučka, M.; Skočaj, D. No Label Left Behind: A Unified Surface Defect Detection Model for All Supervision Regimes. *J. Intell. Manuf.* **2026**, *37*, 3189–3209. [[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.