Anomaly detection and localization with state-of-the-art deep learning models to support quality inspection in car manufacturing

Engineering Reports

WILEY

Engineering Reports Open Access

RESEARCH ARTICLE OPEN ACCESS

Anomaly Detection and Localization With State-of-the-Art Deep Learning Models to Support Quality Inspection in Car Manufacturing

Francesco Manigrasso | Davide Calandra | Lia Morra | Fabrizio Lamberti

Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy

Correspondence: Fabrizio Lamberti (fabrizio.lamberti@polito.it)

Received: 18 April 2025 | **Revised:** 24 January 2026 | **Accepted:** 4 February 2026

Keywords: anomaly detection | automated inspection | automotive | deep learning | defects | localization | manufacturing | production line | quality control | unsupervised learning

ABSTRACT

Maintaining high quality in automotive manufacturing is essential, as even small defects can lead to safety issues, costly recalls, and increased operational costs. Manual inspection is often unreliable in fast-paced production, limited by human error and poor scalability. Advanced imaging and deep learning-based Anomaly Detection and Localization (ADL) offer effective alternatives, but their use in industry is challenged by factors like complex geometries, inconsistent lighting, and environmental noise. This work presents an ADL framework for inspecting sealant application in car underbodies that combines a video acquisition system with four state-of-the-art deep learning models. To overcome the lack of annotated data, a synthetic defect generation module is introduced, creating realistic anomalies that improve model evaluation while reducing annotation effort. The framework was tested on both synthetic and real-world data, achieving high localization performance (AUROC up to 99.7%, F1-score of 43.4%) with inference times ranging from 0.08 to 3.33 s depending on model complexity. These results highlight the trade-offs between speed and accuracy, and confirm the potential of ADL models for real-time quality control in industrial automotive settings.

1 | Introduction

Ensuring high product quality is crucial in modern industrial manufacturing. In the automotive sector, even minor defects can lead to costly recalls, safety risks, and increased operational expenses [1–5]. Traditional manual inspection methods are vulnerable to human error and production variabilities. As a result, they are increasingly inadequate in high-throughput environments [6, 7].

Advanced imaging technologies integrated with Artificial Intelligence (AI) have emerged as effective tools for identifying and localizing defects. They enable targeted interventions and help reduce downtimes [8–17]. Deep learning models, in particular, have enabled the development of highly accurate Anomaly Detection and Localization (ADL) techniques.

This work addresses the above need by focusing on quality control of a sealant application process in automotive manufacturing. Sealants play a crucial role in ensuring car durability by preventing water ingress and corrosion in key joints and seams. The application process poses challenges due to the intricate geometry of car underbodies and the difficulty in detecting subtle defects, such as gaps, bubbles, or misalignments.

An ADL framework is presented that combines a video acquisition system with sophisticated deep learning algorithms, such as *PatchCore* [18], *MemSeg* [19, 20], *EfficientAD* [21, 22],

and *SimpleNet* [23, 24]. A comparative analysis is provided, highlighting trade-offs in terms of accuracy, computational efficiency, and robustness to environmental variability. Benchmarking state-of-the-art algorithms with real datasets provides valuable insights into their practical applicability to the automotive sector.

Finally, an ad-hoc strategy is introduced to address imbalanced datasets in industrial settings. The proposed method generates plausible synthetic anomalies, enabling a more rigorous evaluation of model generalization. It tests performance on a synthetic test set containing previously unseen defects. This strategy reduces reliance on extensive annotated datasets, which are often difficult and costly to produce in real-world scenarios.

In summary, the present work provides three contributions:

1. *Real-world application of ADL techniques in the automotive context*: It investigates the application of advanced ADL techniques in real-world automotive settings, particularly regarding the use of sealants. Unlike other works that typically assess techniques in controlled environments, it is assessed within an industrial setting, characterized by varying lighting conditions, ambient noise, and complex structural features.
2. *Assessment of state-of-the-art deep learning models*: It introduces a benchmark that includes multiple state-of-the-art deep learning models, providing a comprehensive evaluation of their performance and inherent limitations within the context of car sealant application tasks.
3. *Synthetic Defect Generation Method*: It introduces a method for generating synthetic defects to address the challenge of imbalanced datasets for the tackled process. This method produces realistic defect simulations, enabling a more thorough evaluation of the models' ability to detect novel anomalies while reducing dependence on labor-intensive and expensive data annotation processes.

These contributions advance industrial ADL by improving existing models and introducing solutions tailored to the specific challenges of the automotive industry. The remaining sections provide an in-depth presentation of the methodology, experimental configuration, obtained results, and their significance for the industrial domain.

2 | Related Work

In industrial environments, particularly manufacturing, ADL and, more generally, anomaly detection are crucial for promptly identifying defects, thereby maintaining product quality and cutting down on operational expenses. Over the years, a variety of methods have been formulated to tackle this need, commonly categorized as supervised or unsupervised.

Supervised methods require extensive labeled datasets and perform well in detecting known defect types but struggle with unseen anomalies [6, 25, 26]. Unsupervised models can identify novel defects without requiring large annotated datasets. This makes them valuable when collecting defective underbodies is costly and time-consuming. However, they often generate false positives, particularly in highly variable production scenarios such as automotive manufacturing [18, 27–29]. Robust solutions are needed to handle unpredictable, real-world conditions effectively, balancing the strengths and limitations of the two approaches [30, 31].

In light of these two paradigms, numerous ADL approaches have been developed for industrial inspection. The following overview presents some of the most representative models, highlighting their typical use cases, advantages, and limitations.

Supervised methods have achieved notable success in industrial applications, where ample labeled data is available. Models like Faster R-CNN [32], YOLO [33], and RetinaNet [34] are particularly effective in environments like automotive manufacturing, where visible surface defects critically impact product quality. These models offer high accuracy by leveraging large, well-annotated datasets. For instance, Faster R-CNN, and YOLO are extensively utilized for detecting surface-level defects, enabling rapid and precise identification of visual imperfections [34].

However, reliance on substantial labeled datasets represents a significant challenge in industrial contexts with rare, costly-to-label anomalies. Additionally, the performance of these models diminishes when dealing with subtle anomalies or limited datasets, especially in complex setups involving multiple viewpoints [34]. Advanced approaches like Modulated Intensity Decoding (MID) [35] enhance defect detection on reflective surfaces by minimizing reflections through encoded fringe patterns. This improves accuracy when integrated with models like Faster R-CNN, YOLO, and DETR [36]. Despite its robustness, MID is computationally intensive and demands extensive labeled data, limiting its scalability for real-time applications.

Multi-view inspection systems [34] enhance detection accuracy by capturing defects from multiple viewpoints. While this approach improves precision, it increases computational costs and system complexity.

In real-time scenarios, lightweight models like YOLO deliver faster performance but sacrifice precision, especially for subtle anomalies. Besides advanced, computationally intensive methods, simpler solutions employing low-cost hardware are also valuable in specific industrial scenarios [16].

To mitigate computational requirements, models like TinyDefectRNet [37] provide improved efficiency while maintaining reasonable accuracy, making them suitable for industries with limited processing resources. Nonetheless, they are less effective at detecting complex defects compared to architectures like RetinaNet or DETR [36].

Template matching approaches, exemplified by RDNN [38], combine object detection with predefined templates to reliably detect defects. This dependence, however, limits flexibility and makes adaptation to novel anomalies difficult. In contrast, models like Faster R-CNN and RetinaNet offer greater adaptability to new defect types, enhancing versatility in dynamic industrial environments.

Few-shot learning methods, such as CAVGA [39], address the scarcity of labeled data by training anomaly detectors with small datasets. While these methods reduce data dependency, they often struggle to generalize to unseen anomaly types. Similarly, iDAAM [40] employs dense modules and attention mechanisms to improve defect classification. However, it may still struggle with defect classes not present in training data.

Unsupervised methods, which do not require labeled datasets, are increasingly preferred in industrial settings where labeled data is scarce. These methods learn normal data patterns and identify deviations as potential anomalies. This makes them ideal for settings with rare or evolving anomalies. However, their effectiveness depends strongly on the industrial context.

A prominent framework in this category is the Teacher-Student model [27], where a student network is trained to replicate the outputs of a teacher network using anomaly-free data. Anomalies are detected when the student's predictions significantly deviate from the teacher's, providing an efficient unsupervised detection mechanism. However, the model may struggle with subtle anomalies or shifting data distributions, requiring periodic retraining.

PatchCore [41], in turn, offers a scalable solution by constructing a core-set of normal patterns and detecting deviations. This method excels at identifying subtle changes, making it suitable for applications where labeling is impractical. While PatchCore handles high-dimensional data efficiently, it may underperform in tasks requiring detailed feature representations, as it emphasizes broader deviations.

MemSeg [19] enhances anomaly localization by using memory modules, especially effective in imbalanced datasets. It provides high precision for small anomaly detection. However, its computational cost may hinder real-time applicability.

Conversely, models like SimpleNet [23] and EfficientAD [21] balance accuracy and efficiency. SimpleNet suits real-time industrial use, learning normal patterns with minimal computational overhead. Although it performs well in many use cases, it may struggle with complex or subtle anomalies. Similarly, EfficientAD employs attention to focus on relevant data regions, optimizing resource use while maintaining solid performance. Still, like SimpleNet, it may not fully capture intricate anomalies, and may need to be complemented by more advanced models in complex tasks.

A common challenge across unsupervised methods is generalizing to unseen anomalies. Template-based supervised models like RDNN [38] perform reliably for known defects but fail to generalize. In contrast, approaches like PatchCore and MemSeg better accommodate new anomaly types without retraining or template updates, providing more flexibility in dynamic industrial settings.

3 | Production Line Analysis

A proper knowledge of the production line considered in this work is crucial for the effective deployment of deep learning models for defect detection in car underbodies during the sealing process. The industrial environment presents substantial challenges, including variable lighting, product diversity, and process inconsistencies. These challenges were addressed through a detailed examination of the environment, which informed the optimization of model performance.

In automotive manufacturing, sealant application is an essential process that safeguards the vehicle's structural stability and shields it from environmental impacts. The examined production line includes three main stages: manual sealing, automated sealing, and quality inspection and correction. In the initial *manual sealing cell*, skilled operators manually apply the sealant to specific joints and seams of the car body. It is used in areas where complex geometries limit automation. Operators adjust their technique based on each car model's geometry, ensuring precision and flexibility. The second stage involves the *automated sealing cell*, where robotic systems apply the sealant to the remaining sections of the car body, specifically the underbody. Automation enhances efficiency and reduces human-induced variability. The robotic application ensures uniform sealant distribution, which improves consistency and throughput. After automated sealing, the car body moves to the *quality inspection & correction cell*. Here, trained personnel inspect the sealant application through visual checks and sensor-based measurements. Defects such as gaps, bubbles, or incomplete coverage are manually detected and rectified. This step is essential for maintaining sealant effectiveness and vehicle durability. An *intermediate sealing cell* exists between the manual and automated stages. It gathers environmental parameters (e.g., temperature and humidity) and vehicle-specific features (e.g., underbody dimensions and material attributes). This data supports adjustment of the sealing process, helping ensure adhesion consistency and product reliability during automation.

To support ADL framework development and evaluation, a video acquisition system was deployed within the inspection cell. Several camera viewpoints were evaluated to identify the one capturing the widest range of sealant patterns applied by the robots.

The most effective view, referred to as "lower," was obtained by placing the camera on the ground, aligned with the conveyor and pointing upward, capturing a full underbody view. Despite occasional occlusions from vehicle components and a fixed viewpoint, capturing frames at predefined intervals allowed effective analysis of the sealant application with a relatively simple setup.

4 | Data Acquisition and Preparation for Robust Anomaly Detection

This section outlines the critical steps for acquiring and processing data. These steps support the training of unsupervised deep learning models for ADL. The procedure starts with video-based data collection. It then continues with systematic underbody extraction and dataset preparation for model training and evaluation. Emphasis is placed on ensuring data diversity and integrity. A dedicated section then introduces the types of sealant defects considered in this study. Given the scarcity of defective underbodies, synthetic data augmentation is employed to investigate model robustness. Pre-processing techniques, such as binary masking, are applied to focus on areas of interest and minimize background noise, further improving frameworks' accuracy. A design of the

entire procedure, from data acquisition to model training and evaluation, is illustrated in Figure 1.

Within this pipeline, ROI-based binary masking, the subsequent size-aware analysis of missed defects, and statistically grounded model comparison (detailed in Sections 7 and 8) are conceived as distinctive methodological components of the proposed framework, tailored to the requirements of industrial sealant inspection.

Unlike several prior industrial ADL studies that rely on controlled laboratory settings or multi-camera acquisition setups, the data considered in this work are collected directly on a single-camera underbody sealant line in a real automotive plant. The resulting dataset combines scarce real defects with a systematically generated set of synthetic anomalies, enabling a more realistic assessment of model behavior under practical constraints.

4.1 | Data Collection and Car Underbody Extraction

Video recordings of car underbodies were acquired following the automated application of sealant in the intermediate sealing cell. Subsequently, a systematic and automated underbody extraction process was developed and implemented. This approach suits real-time data acquisition systems. Prompt and efficient extraction of informative underbodies is essential for accurate analysis and continuous monitoring. The Template Matching technique from the OpenCV library [42, 43] was utilized. This technique entails sliding a predefined template image across the target image to identify regions that exhibit high similarity based on specific similarity metrics. Three distinct templates were employed to extract three different viewpoints: "lower front", "lower center", and "lower back" (Figure 2). The multi-template approach addresses the dynamic movement of the car underbody. As the vehicle moves, different sections are exposed at different time intervals. Consequently, the first template effectively captures the frontal section of the underbody at its peak visibility, followed sequentially by the central and rear sections.

By means of the OpenCV's cv.matchTemplate() function, a sliding-window operation is performed across underbodies, producing a similarity map based on normalized cross-correlation (CCOEFF_NORMED). cv.minMaxLoc() is then used to localize the Region of Interest (ROI) by identifying the coordinates with the highest similarity scores. A user-defined threshold is applied to retain only potentially relevant underbodies. Despite possible challenges such as variations in scale, rotation, or lighting, the generally consistent visual appearance of car underbodies ensured reliable extraction.

4.2 | Sealant Defect Types

Automated sealant application commonly encounters defects such as the *absence of sealant*, where robotic applicators fail to

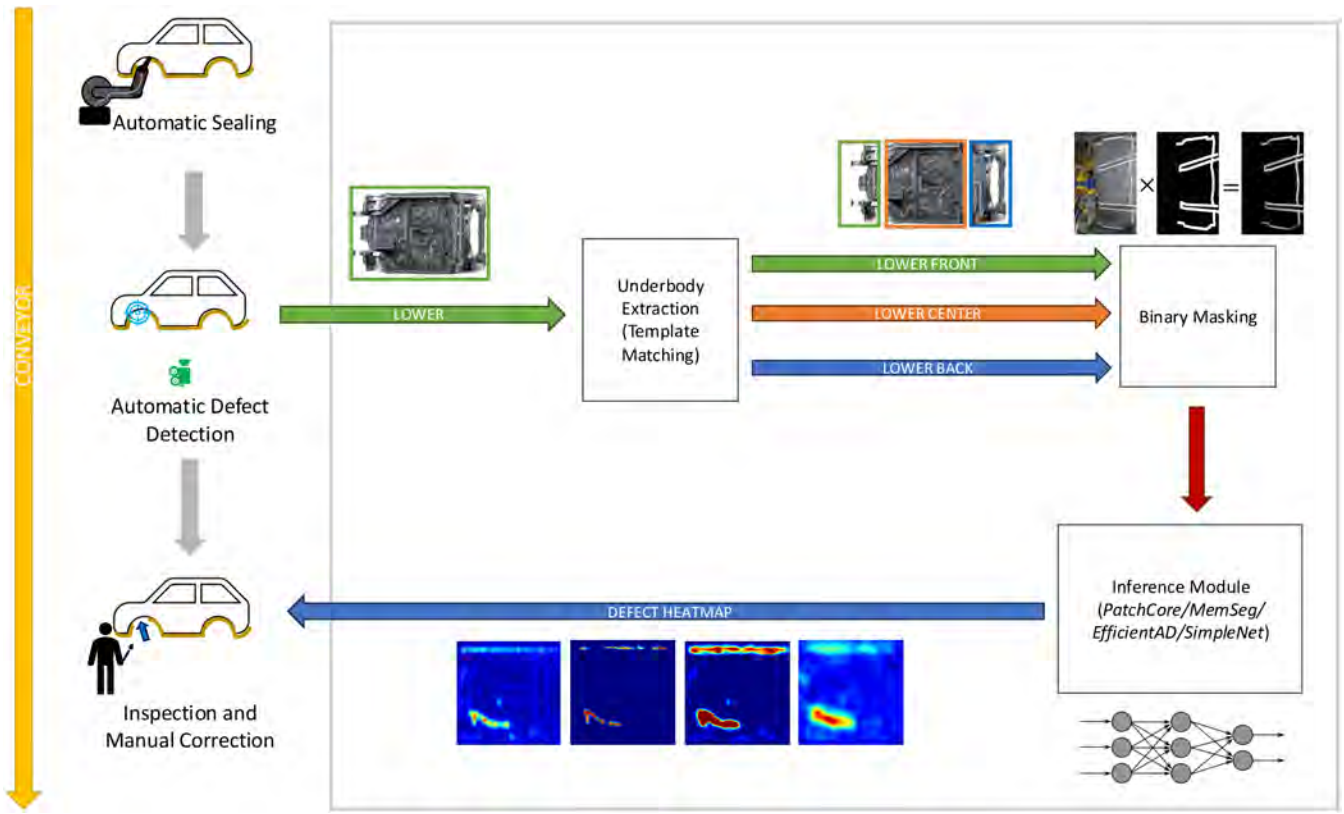


FIGURE 1 | Overview of the proposed Anomaly Detection and Localization (ADL) pipeline for sealant inspection on car underbodies during the automatic sealing process. The figure illustrates the main processing stages, including video acquisition in the inspection cell, template-based underbody extraction, and inference producing pixel-wise anomaly heatmaps used to support operator inspection and manual correction.

deposit sealant in designated areas, leaving sections vulnerable to water ingress and corrosion. *Sealant misalignment* occurs due to displacements between the car underbody and the robotic nozzle, resulting in irregular application that weakens the protective barrier and increases the risk of leaks. *Sealant bifurcation* and *narrowing* arise from partial nozzle blockages, causing uneven splitting of the sealant stream and reduced flow rates, respectively, which lead to incomplete or insufficient sealant layers. Additionally, the presence of *sealant holes*, caused by interruptions in flow, creates gaps in the protective layer, facilitating moisture, and air entry that accelerate corrosion. The different types of defects are visually illustrated in Figure 3.

4.3 | Synthetic Defect Generation

Due to the scarcity of real defective underbodies, synthetic anomalies were created. Advanced image manipulation techniques were used to mimic real-world defects.

The GIMP open-source image editing software [44] and its Resynthesizer plugin were used. The plugin employs texture synthesis techniques, such as inpainting and resynthesis, which enable the reconstruction of missing or altered sections of an image by extrapolating from the surrounding content. These techniques were leveraged to simulate realistic defects. For instance, the

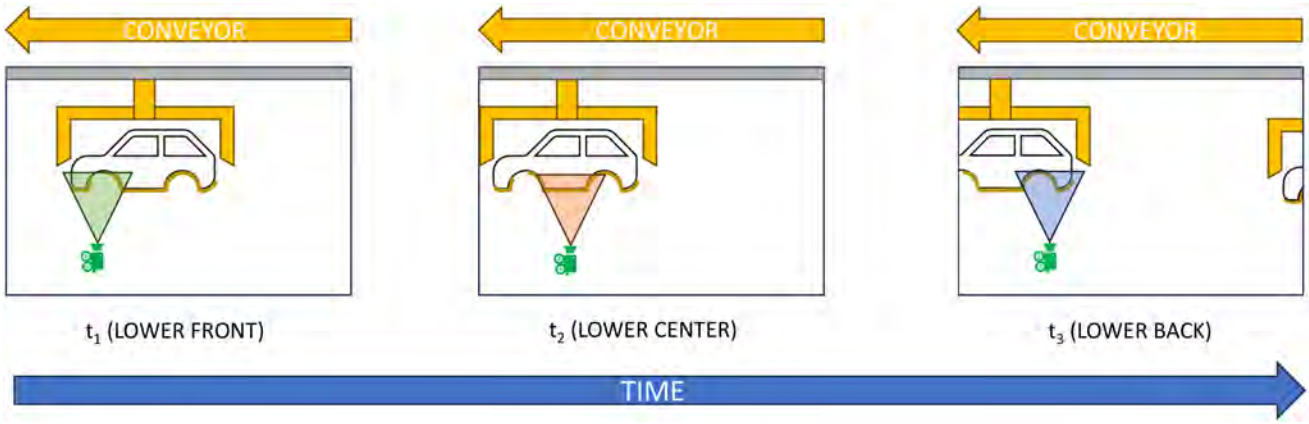


FIGURE 2 | Template matching strategy used to extract three underbody viewpoints (lower front, lower center, and lower back) at different time instants while the vehicle moves on the conveyor. Each template is matched on the incoming frame stream to localize the corresponding Region of Interest (ROI) associated with the sealant trajectories.

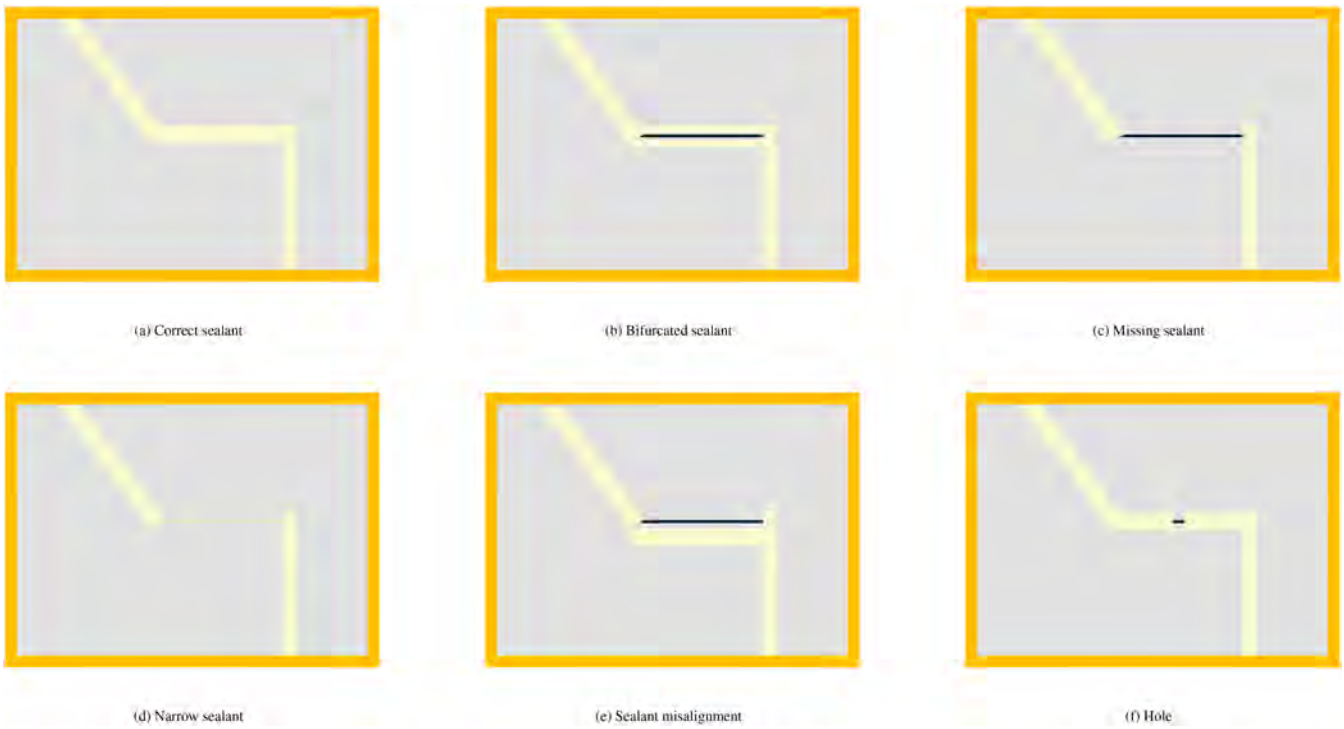


FIGURE 3 | Visual representations of different sealant defect types used for illustration purposes: (a) correctly applied sealant, and (b–f) representative defect types observed in the production line, including bifurcation, missing sealant, narrowing, misalignment, and hole.

absence of sealant was obtained by selectively removing portions of the sealant and filling the gaps with textures that blend seamlessly with the original image. As a result, synthetic defective underbodies were created, each associated with corresponding Gound Truth (GT) annotations. It is worth noting that these image-level operations do not incorporate physical process modeling or photometric variability, which bounds the realism of the generated defects.

The dataset was assembled by extracting 345 "normal" and 15 "defected" unique high-quality car underbodies, divided into training, validation, and test sets. For each underbody in the training set, 5 video frames (same underbody at different time points) were randomly selected; only one video frame each was selected for validation and testing underbodies. Synthetic defective underbodies were generated by modifying video frames originally classified as "normal," evenly distributed between the validation and test subsets to ensure a balanced representation of anomalies.

To ensure exhaustive coverage of the underbody, each specimen was systematically partitioned into predefined regions designated for defect generation. Specifically, the "lower front" region was subdivided into four sections, the "lower center" region into eight sections, and the "lower back" region into five sections. For each region, a series of anomalies were synthesized and incorporated to produce defective underbodies specific to that region. This region-centric methodology enables the systematic simulation of defects across diverse underbody sections, ensuring comprehensive coverage and enhancing the dataset's capability to represent a broad spectrum of defect scenarios.

Table 1 reports the distribution of synthetically generated underbodies across each designated viewpoint. For every viewpoint, a variable number of synthetic defects were synthesized. Table 2 provides a detailed overview of the final dataset distribution, encompassing the allocation of both synthetic and authentic underbodies across the training, validation, and test subsets. Examples of generated defects are illustrated in Figure 4.

While this approach supported systematic evaluation under controlled conditions, it relies on 2D texture–based manipulations (inpainting and resynthesis) that do not model the underlying deposition physics (e.g., sealant rheology, flow, nozzle dynamics, and wetting) nor the scene photometry (variable illumination, specular highlights, and shadows).

Consequently, certain defect classes, such as fine scratches, low-contrast discontinuities, or geometry-dependent irregularities near welds, may be underrepresented or less realistic in the synthetic set. These root causes and potential mitigations (e.g., physics-aware simulation and photometric rendering, targeted acquisition, and active learning) will be discussed in Section 8.4.

In contrast to anomaly simulation mechanisms embedded within specific architectures, such as the internal synthetic pattern generation used in MemSeg, the proposed pipeline operates directly at input-image level, is model-agnostic, and is explicitly structured around geometry-aware regions of interest. This design allows the four benchmark ADL models to be evaluated on a consistent mixture of real and synthetic defects tailored to the operational constraints of the single-camera industrial scenario considered here.

TABLE 1 | Distribution of synthetically generated defects across the three viewpoints (lower front, lower center, lower back). Numbers in parentheses indicate the count of single anomalies injected within each region-specific generation slot.

Defect	Lower front	Lower center	Lower back
Bifurcated sealant	8 (2)	8 (1)	5 (1)
Missing sealant	24 (6)	32 (4)	30 (6)
Narrow sealant	4 (1)	8 (1)	5 (1)
Sealant misalignment	12 (3)	8 (1)	15 (3)
Hole	8 (2)	8 (1)	5 (1)
Total	56	64	60

TABLE 2 | Dataset distribution by viewpoint and split (train, validation, test).

Viewpoint	Split	Normal	Real def.	Synthetic def.
Lower front	Train	69 (345)	0 (0)	0 (0)
	Validation	23 (23)	0 (0)	28 (28)
	Test	23 (23)	3 (3)	28 (28)
Lower center	Train	67 (335)	0 (0)	0 (0)
	Validation	23 (23)	0 (0)	32 (32)
	Test	24 (24)	6 (6)	32 (32)
Lower back	Train	70 (350)	0 (0)	0 (0)
	Validation	23 (23)	0 (0)	30 (30)
	Test	23 (23)	6 (6)	30 (30)

Note: Values report the number of unique underbodies; values in parentheses report the total number of extracted video frames for those underbodies. Real defects are included only in the test split, while synthetic defects are evenly distributed between validation and test.

4.4 | Binary Masking

Detecting anomalies in the sealant applied to car underbodies poses significant challenges due to the inherent variability in both the sealant materials and the assembly line processes. Underbodies exhibit a range of features, including perforations, manually applied sealant patterns, unique structural characteristics, and substantial background noise.

To address these challenges, a binary mask is generated based on car underbody templates to isolate ROIs. This methodology leverages the same template-matching technique employed in the car underbody extraction process. The binary mask effectively concentrates the analysis on the sealant regions while excluding irrelevant background details. To accommodate potential

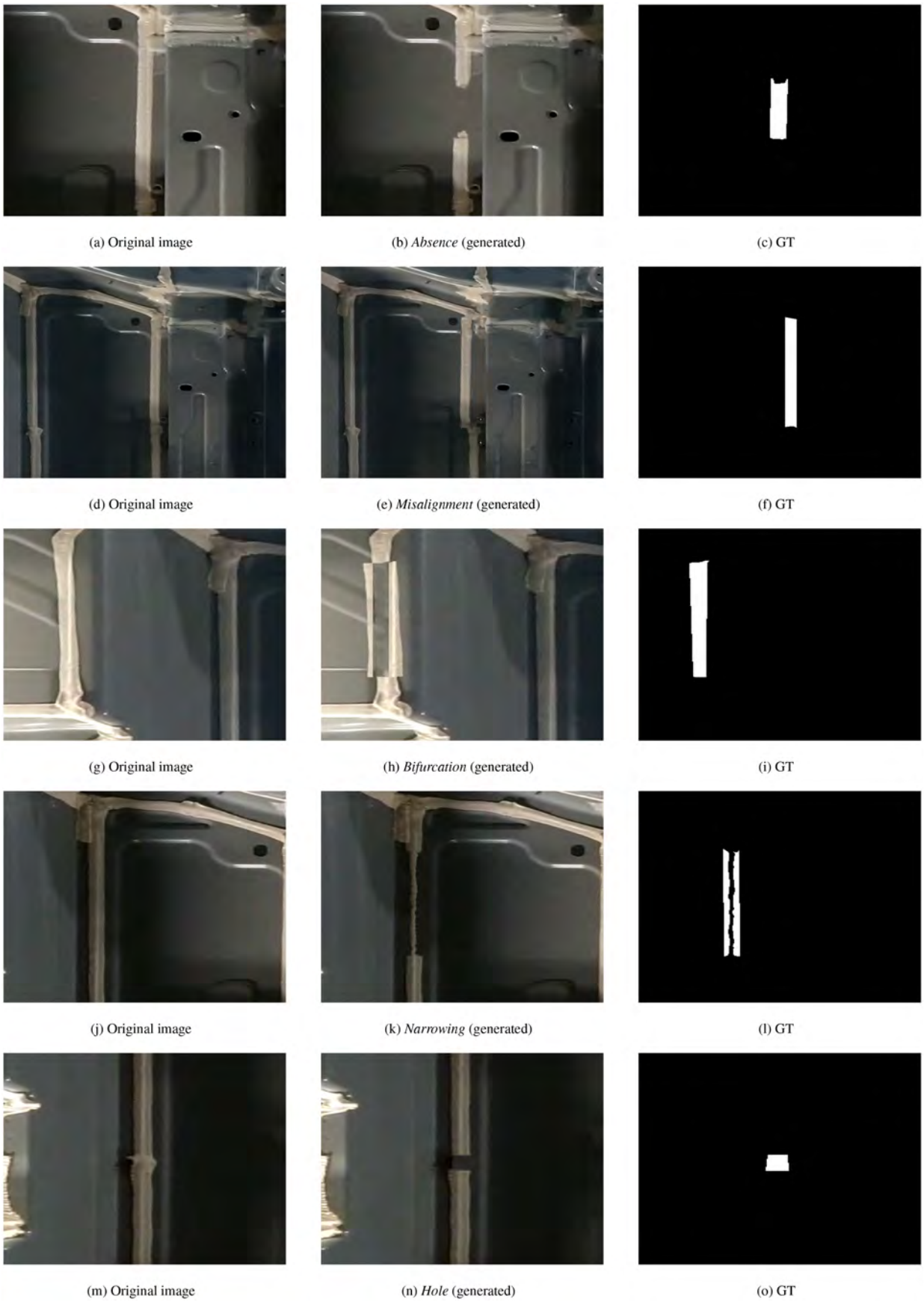


FIGURE 4 | Examples of synthetic defect generation. Each triplet shows (left-to-right): The original normal frame, the corresponding synthetically generated defect, and the associated Ground-Truth (GT) binary mask. Rows illustrate different defect types (missing sealant, misalignment, bifurcation, narrowing, and hole).

misalignments, a margin is incorporated into the mask design, ensuring that critical image regions are not accidentally omitted. The binary mask is applied during both training and inference. It filters out extraneous information and improves data consistency. As illustrated later, a series of experiments were conducted to evaluate the efficacy of this preprocessing step by comparing the performance of models trained and evaluated on masked images with those trained and evaluated on unprocessed images.

5 | ADL Models for Industrial Applications

This section reviews state-of-the-art deep learning models developed for ADL based on unsupervised or semi-supervised learning for scenarios with limited access to anomalous data. These approaches allow models to learn from nominal conditions using a small set of video frames.

PatchCore [18], MemSeg [19], EfficientAD [21], and SimpleNet [23] were considered in this study. Each model is designed to address ADL challenges in variable environments with low defect prevalence, using different strategies to optimize detection and localization.

5.1 | PatchCore

PatchCore is a method designed for ADL in images. It employs patch-level features derived from a pre-trained encoder and uses these features to identify anomalies. The procedure consists of several pivotal stages, each enhancing the efficacy and robustness of the process. The overall functioning is depicted in Figure 5.

The PatchCore workflow consists of the following steps.

1. *Extraction of patch-level features*. Feature maps are extracted from a network such as ResNet. These feature maps, denoted as $\phi_{i,j}$, are obtained from intermediate layers to achieve a balance between abstraction and localization. Patches of dimension $p \times p$ are then sampled from these feature maps. Each patch centered at a spatial location (h, w) is represented by the feature vector $\phi_{i,j}(h, w) \in \mathbb{R}^c$, where c denotes the number of channels in the feature map.

2. *Aggregation of local features*. To account for spatial variations and enhance robustness, PatchCore aggregates features from local patch neighborhoods. A local aggregation function, such as adaptive average pooling, is applied over the neighborhood $N_p(h, w)$:

$$\phi_{i,j}\left(N_p(h,w)\right) = f_{agg}\left(\left\{\phi_{i,j}(a,b) \mid (a,b) \in N_p(h,w)\right\}\right). \quad (1)$$

3. *Coreset reduction*. To reduce memory and ensure efficient inference, a representative subset of the patch features (coreset) is selected. This is done via a minimax facility location algorithm:

$$m_i = \arg \max_{m \in M - M_C} \min_{n \in M_C} \| \psi(m) - \psi(n) \|_2, \quad (2)$$

where M is the full set of patch features, M_C the current coreset, and ψ a random linear projection function.

4. *Anomaly detection and localization*. At test time, the anomaly score s^* for a patch m_{test} in image x_{test} is computed as:

$$s^* = \max_{m_{test} \in P(x_{test})} \max_{m \in M_C} \| m_{test} - m \|_2, \quad (3)$$

where $P(x_{test})$ denotes all patches from the test image. The score is then scaled to emphasize rare matches:

$$s = \left(1 - \frac{\exp \| m^*_{test} - m^* \|_2}{\sum_{m \in Nb(m^*)} \exp \| m^*_{test} - m \|_2}\right) \cdot s^*, \quad (4)$$

where $Nb(m^*)$ denotes the b nearest neighbors of m^* in the coreset. This score is used to build an anomaly heatmap over the image.

5. *Inference phase*. The inference phase applies the detection and localization procedure described above to each test image. The same equations and scoring scheme are used to produce anomaly heatmaps for unseen data, supporting both identification and localization of anomalies.

5.2 | MemSeg

MemSeg is a semi-supervised approach specifically designed for surface defect detection in images. It integrates multiple

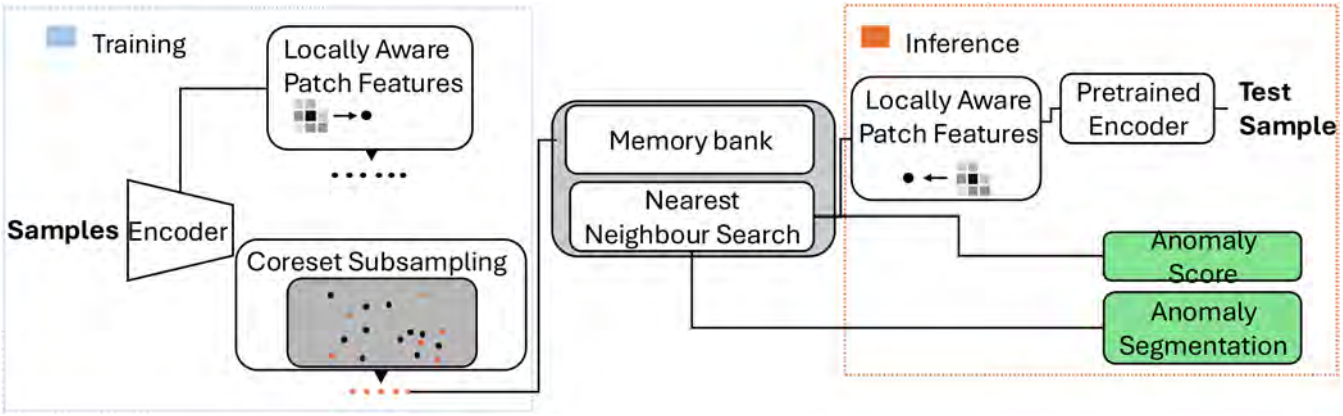


FIGURE 5 | PatchCore architecture [18]. It essentially relies on the extraction and local aggregation of patch-level features, facilitated through the construction of a memory bank that consolidates local feature representations derived from the intermediate layers of a pre-trained network.

components that collectively support anomaly detection, generation, and localization. The overall workflow is illustrated in Figure 6.

MemSeg comprises the following key components:

1. *Anomaly Simulation Strategy*. Anomalies are simulated during training to improve robustness. This is achieved by generating a mask using Perlin noise and combining it with a binarized version of the input image. The result defines the region where artificial noise is injected. The final simulated anomaly image I_s is computed as:

$$I_s = I_b + I_n. \tag{5}$$

This strategy creates realistic-looking defects that support better model generalization.

2. *Memory Module and Spatial Attention*. A memory module stores representative patterns from non-defective data and enables comparison with test samples using L2 distance:

$$D_i = \| F_i - M_j \|_2 . \tag{6}$$

Spatial attention maps are then generated from these distances to highlight abnormal areas:

$$A_i = \frac{1}{C}\sum_{c=1}^{C} D_{i,c}. \tag{7}$$

3. *Multi-Scale Feature Fusion*. Features from the encoder and memory are fused using convolution, coordinate attention, and upsampling. The final fused feature map is:

$$F_{fused} = \mathrm{Conv}\big(\mathrm{UpSample}\big(F_{aligned}\big)\big). \tag{8}$$

This enhances the model's ability to capture defects at various spatial resolutions.

4. *Loss Function*. MemSeg optimizes a combined loss:

$$L_{total} = \lambda_1 L_{L1} + \lambda_2 L_{focal}, \tag{9}$$

where L1 loss ensures pixel-level accuracy and Focal Loss addresses class imbalance between normal and anomalous regions.

5. *Inference Phase*. The inference process follows the same architecture. Features extracted via a ResNet18 encoder are compared against the memory bank using L2 distance. Spatial attention maps are generated from these differences across multiple scales. The attention is used to build a pixel-wise anomaly heatmap that highlights regions deviating from learned normal patterns. This multi-scale and memory-based analysis allows for accurate and interpretable anomaly localization.

5.3 | EfficientAD

EfficientAD is a framework designed for real-time visual anomaly detection, integrating an efficient patch descriptor network with a student–teacher model. The architecture supports both structural and logical anomaly detection with high computational efficiency. An overview is shown in Figure 7. The following sections detail the main components and mechanisms that constitute the EfficientAD framework.

1. *Efficient Patch Descriptors*. EfficientAD employs the Patch Description Network (PDN), a compact convolutional network that extracts features from 33×33 patches in a single forward pass. The PDN is trained by distillation from a deeper pre-trained classifier, minimizing the mean squared error between its output and the original network's features.

2. *Student–Teacher Learning*. Both the teacher and student use the PDN. The teacher is trained on normal images, while the student is trained to replicate the teacher's outputs. The discrepancy between the two identifies anomalies. The student is guided by the hard feature loss:

$$L_{hard} = \mathrm{mean}\big(D_{c,w,h} \geq d_{hard}\big), \tag{10}$$

where $D_{c,w,h}$ is the squared difference and d_{hard} is a quantile-based threshold. A regularization term using ImageNet-like data prevents the student from overfitting to anomalies.

3. *Logical Anomaly Detection*. A separate autoencoder learns to reproduce the teacher's output, capturing logical relationships in the data. The student is also trained to predict this reconstruction, thus learning systematic patterns. Logical

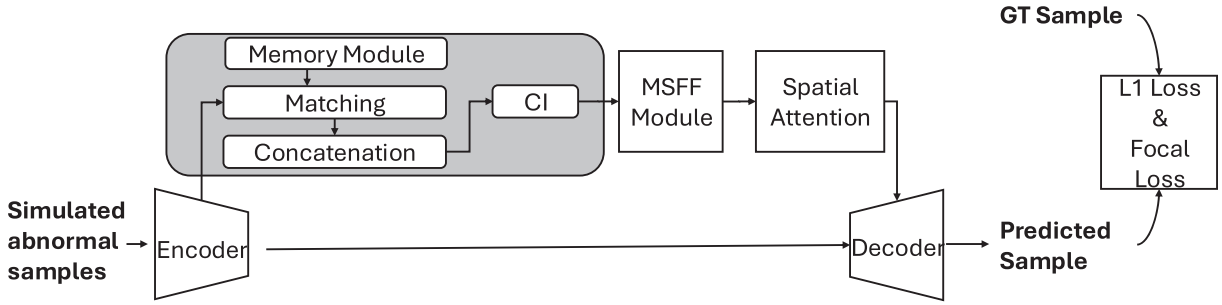


FIGURE 6 | MemSeg architecture [19]. Simulated abnormal underbodies are fed into an encoder. The encoded features pass through a matching stage and combine with the memory module's output, which stores representative features. A Contextual Information Multi-Scale Feature Fusion (CI MSFF) module and spatial attention mechanism enhance the features. The decoder then produces a predicted sample. During training, predictions are compared to GT underbodies, and errors are minimized using L1 and Focal Loss functions for accurate anomaly detection and segmentation.

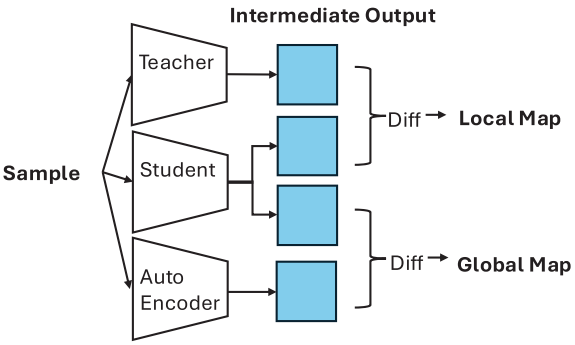


FIGURE 7 | EfficientAD architecture [21]. The model employs a teacher-student framework where the teacher generates a global map of the data distribution, while the student refines its learning through local maps. This differential approach enhances the detection of intricate patterns in high-dimensional data efficiently.

anomalies are then identified as discrepancies between the student and autoencoder outputs.

4. *Fusion of Anomaly Maps*. EfficientAD combines local (student–teacher) and global (autoencoder–student) anomaly maps. To align the two, quantile-based normalization is applied: scaling factors are computed on validation data, followed by a linear transformation to align the score ranges. The final anomaly map is then obtained by averaging the normalized local and global maps.
5. *Inference Phase*. During inference, both teacher and student extract patch features from test images. The difference at each patch yields an anomaly score. A heatmap is generated by aggregating these scores spatially. The final output is the normalized average of structural and logical anomaly maps, providing a comprehensive defect visualization.

5.4 | SimpleNet

SimpleNet is a network developed for detecting and localizing anomalies in images, featuring a straightforward and efficient architecture. It is composed of four main modules, illustrated in Figure 8. The architecture of SimpleNet consists of four primary components, detailed below. A separate section outlines the inference procedure during test time.

1. *Feature Extractor*. A pre-trained WideResNet50 is used to extract local feature representations $\phi_{l,i}$ from input images $x_i \in \mathbb{R}^{H\times W\times 3}$, using selected levels l to balance abstraction and localization. Features are downscaled using adaptive average pooling and concatenated into a unified representation o_i.
2. *Feature Adapter*. To reduce domain bias, the extracted features $o_{i,h,w}$ are projected into a new feature space via a fully connected layer without bias, producing adapted features $q_{i,h,w} = G_\theta\left(o_{i,h,w}\right)$. This adaptation better aligns the representation with the anomaly detection task.
3. *Anomaly Feature Generator*. During training, anomalous features are synthetically created by adding Gaussian noise $\epsilon \sim \mathcal{N}\left(\mu, \sigma^2\right)$ to the adapted features:

$$q^-_{i,h,w} = q_{i,h,w} + \epsilon. \tag{11}$$

This enables the network to learn from both normal and perturbed (anomalous) examples.

4. *Anomaly Discriminator*. A two-layer MLP acts as a binary classifier, trained to distinguish between $q_{i,h,w}$ (normal) and $q^-_{i,h,w}$ (anomalous). The truncated l_1 loss function used is:

$$l_{i,h,w} = \max\left(0, t^+_h - D_\psi\left(q_{i,h,w}\right)\right) + \max\left(0, -t^-_h + D_\psi\left(q^-_{i,h,w}\right)\right), \tag{12}$$

where D_ψ is the discriminator, and t^+_h, t^-_h are truncation thresholds guiding training toward informative features.

5. *Inference Phase*. At test time, the Anomaly Feature Generator is removed. An input image is passed through the Feature Extractor and Adapter to produce $q_{i,h,w}$, which the discriminator evaluates to generate an anomaly map $S_{AL}\left(x_i\right)$. The maximum score in the map can be used as a global anomaly indicator.

6 | Experimental Setup

This section delineates the experimental framework and the hyperparameter optimization procedures employed for training the four ADL models considered. Each model's hyperparameters were meticulously tuned to enhance performance based on the specific application requirements.

For *PatchCore*, ResNext101 was selected as backbone network, with feature extraction executed at the *layer2* and *layer3* stages. To manage computational overhead, the memory bank was subsampled to 5% of its original size. Both the patch size and the number of nearest neighbors were set to 3, and the embedding dimensions for both pretraining and targeting were configured to 1024 each to ensure robust feature representation. Input images were uniformly resized to 848×480 pixels to standardize the resolution. The experiments were conducted on three subsets of the dataset: lower back, lower front, and lower center. Each combination of backbone network and dataset subset was systematically evaluated to ensure consistency and reproducibility.

MemSeg was configured to utilize input images resized to 256×256 pixels. Data preprocessing incorporated a structural grid size of 8. The transparency range for anomaly detection was defined between 0.8 and 1.0. Texture anomalies were synthesized with a probability of 0.9 using Perlin noise. Perlin noise parameters were set to a scale of 5, a minimum scale of 3, and a noise threshold of 0.5. A memory bank with a capacity of 30 samples was maintained to facilitate efficient memory management. The model employed a ResNet18 architecture as the feature extractor without any pre-training. The loss function comprised a weighted combination of Focal Loss and L1 Loss, with weighting coefficients of 0.7 and 0.3, respectively. The Focal Loss utilized a gamma parameter of 4, without any alpha weighting. The model was trained with the AdamW optimizer for 15,000 iterations with a batch size of 8, learning rate of 0.003 and a weight decay of 5×10^{-4}. A cosine warmup scheduler was employed, progressively

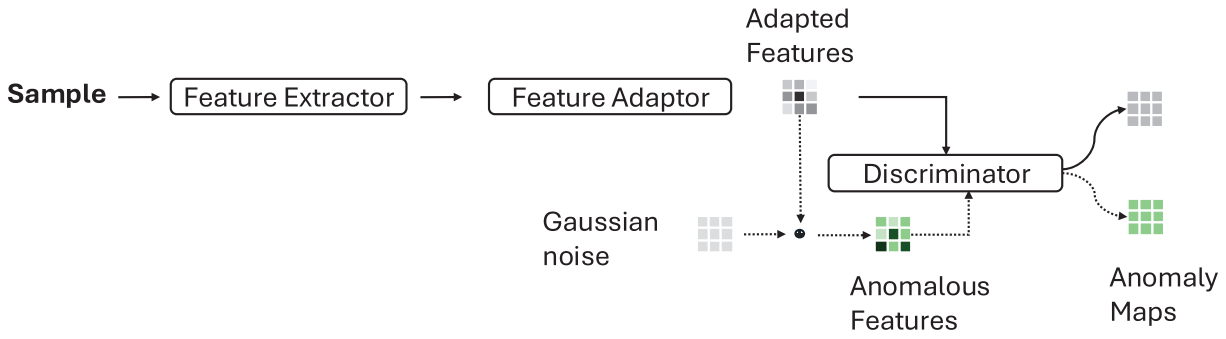


FIGURE 8 | SimpleNet architecture [23]. It consists of a feature extractor and a feature adapter that processes input data to produce adapted features. A discriminator distinguishes between normal and anomalous features, generating anomaly maps that highlight areas of concern. This architecture effectively captures subtle anomalies in the data while ensuring computational simplicity.

adjusting the learning rate with a minimum threshold of 1×10^{-4} and a warmup ratio of 0.2.

In the implementation of *EfficientAD*, the model was trained with a batch size of 1 and an initial learning rate of 0.0001. A learning rate decay strategy was used, reducing the learning rate to 10% of its initial value after completing 95% of the total training iterations, which amounted to 70,000 iterations. The training process utilized both PDN-Medium (EfficientAD-M) and PDN-Small (EfficientAD-S) configurations for feature extraction, with a channel size set to 384. Input images were resized to 256×256 pixels, and the teacher input size was configured to 512.

SimpleNet was implemented using the WideResNet50 architecture as its backbone, with additional configurations allowing for flexibility in layer selection and embedding dimensions. Specifically, the model utilized layers *layer2* and *layer3* within its feature adapter module. Both the pre-trained and target embedding dimensions were set to 1536, ensuring a robust feature representation. A patch size of 3 was employed to capture fine-grained details in the input data. Training was conducted over 40 meta epochs and 4 GAN epochs to balance learning efficiency and model performance. The batch size was maintained at 8, and images were resized and cropped to 322×322 pixels to standardize input dimensions. The embedding size was configured to 256, providing a compact yet expressive latent space. The anomaly generator incorporated a standard deviation parameter, σ, set to 0.015, introducing controlled noise to enhance the model's ability to detect anomalies. The discriminator network was designed with a hidden size of 1024 and comprised 2 layers, facilitating effective discrimination between real and generated samples. A margin of 0.5 was applied to the discriminator's loss function to promote robust training dynamics. Additionally, a pre-projection setting of 1 was utilized to prepare the input features before they were fed into subsequent layers.

The experiments were executed on a workstation equipped with an AMD Ryzen 77,700 processor, 32GB of DDR5 RAM, and an NVIDIA RTX3090 GPU. The models were validated using a dataset comprising synthetically generated anomalies in sealant application and were further evaluated on real-world anomaly underbodies provided by plant operators during inspection processes, thereby assessing the models' adaptability and robustness in identifying anomalies in sealant application across various regions of the car underbody.

Each model in this study was configured with a distinct input resolution, reflecting recommended practices or common usage in prior work. For instance, PatchCore employed a higher resolution (848×480) to capture detailed patch-based features, while MemSeg and EfficientAD used moderate sizes (256×256) to balance accuracy and computational load. SimpleNet retained the resolution of its pretrained backbone (322×322) to ensure consistent feature representations.

For reproducibility, Table 3 summarizes, for each evaluated model, the key training settings (input resolution, backbone, training regime, and method-specific hyperparameters) together with the hardware used for all experiments.

7 | Results

This section presents the experimental evaluation of the four considered ADL models (PatchCore, MemSeg, EfficientAD, and SimpleNet) based on quantitative metrics, inference time, and qualitative analysis.

Quantitative metrics include AUROC (Area Under the Receiver Operating Characteristic), AU-PR (Area Under the Precision–Recall curve), Precision, Recall, F1-Score, and IoU (Intersection over Union), assessed at both image and pixel levels. Specifically, for a binary problem (anomalous vs. normal), let $Y \subseteq \Omega$ be the set of truly anomalous pixels (GT) and let $\hat{Y}_\tau \subseteq \Omega$ be the set of pixels predicted as anomalous from the thresholded anomaly map $S : \Omega \to \mathbb{R}$ that is $\hat{Y}_\tau = \{x \in \Omega | S(x) \geq \tau\}$. The operating threshold is chosen, for each model, as the value that maximizes IoU:

$$\tau^\star = \arg\max_\tau \ \mathrm{IoU}\left(Y, \hat{Y}_\tau\right).$$

Under this choice, IoU is computed as:

$$\mathrm{IoU}\left(Y, \hat{Y}_{\tau^\star}\right) = \frac{|\, Y \cap \hat{Y}_{\tau^\star} \,|}{|\, Y \cup \hat{Y}_{\tau^\star} \,|} = \frac{TP}{TP + FP + FN},$$

where *TP*, *FP*, and *FN* are pixel-level counts.

The error mix is quantified using the *false-positive share*:

$$\mathrm{FP\_share} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{FN}}, \tag{13}$$

TABLE 3 | Summary of key training settings and hardware for each evaluated model.

Model	Input size	Backbone/encoder	Training regime	Optimizer/LR	Key method-specific settings
PatchCore	848×480	ResNeXt101 (features from layer2, layer3)	No gradient training; memory bank built from training set	—	Coreset subsampling 5%; patch size 3; k-NN = 3; embed dim 1024/1024
MemSeg	256×256	ResNet18 (no pretraining)	15,000 iterations; batch size 8	AdamW; LR 0.003; wd 5×10^{-4}	Memory bank size 30; grid size 8; Perlin-based anomaly synthesis; loss 0.7 Focal +0.3 L1
EfficientAD-S	256×256 (teacher input 512)	PDN-Small (student–teacher + AE)	70,000 iterations; batch size 1	LR 1e-4; decay to 10% at 95% iterations	Structural + logical anomaly maps; quantile-based normalization and fusion
EfficientAD-M	256×256 (teacher input 512)	PDN-Medium (student–teacher + AE)	70,000 iterations; batch size 1	LR 1e-4; decay to 10% at 95% iterations	Same as EfficientAD-S; channel size 384
SimpleNet	322×322	WideResNet50 (features from layer2, layer3)	40 meta epochs +4 GAN epochs; batch size 8	As in reference implementation	Patch size 3; embed dim 1536/1536; anomaly noise $\sigma = 0.015$; discriminator 2 layers, hidden 1024; margin 0.5

Note: All experiments were run on a workstation with AMD Ryzen 7 7700 CPU, 32GB DDR5 RAM, and NVIDIA RTX 3090 GPU. Inference times are measured with batch size equal to 1.

Computed at the chosen evaluation granularity. For size-aware FN analysis, for each test image i with height H_i and width W_i, let $M_i \in \{0, 1, \dots\}^{H_i \times W_i}$ denote the GT mask, where any non-zero value indicates a defect pixel. We measure defect size via the image-normalized mask area

$$\text{area\_ratio}(i) = \frac{1}{H_i W_i} \sum_{p \in \Omega_i} \mathbf{1}\big[M_i(p) > 0\big], \quad \Omega_i = \{1, \dots, H_i\} \times \{1, \dots, W_i\}. \tag{14}$$

To make size bins comparable within each camera view v, we compute view-specific tertiles on the test-set distribution $A_v = \{\text{area\_ratio}(i) : i \in v\}$; letting $Q_q(\cdot)$ denote the empirical q-quantile, we set

$$b_1 = Q_{0.333}(A_v), \qquad b_2 = Q_{0.667}(A_v). \tag{15}$$

Each false negative is assigned to exactly one bin based on the GT mask size of its image:

$$\textit{small} \text{ if area\_ratio} \le b_1, \quad \textit{medium} \text{ if } b_1 < \text{area\_ratio} \le b_2, \quad \textit{large} \text{ if area\_ratio} > b_2.$$

This percentile-based binning is distribution-aware (no arbitrary absolute thresholds), yields approximately balanced groups per view, and is fixed across models to enable fair comparisons; when multiple annotations exist for the same sample, we retain the one with the largest mask coverage.

7.1 | Image-Level Metrics

Table 4 shows that masking (†) systematically boosts performance: every masked variant outperforms its un-masked counterpart across all five metrics. Conversely, EfficientAD-S† attains the highest recall (96.3% ± 3.8). However, this comes at the cost of precision, resulting in an F1-Score comparable to that of EfficientAD-M†. Paired t-tests with Benjamini–Hochberg FDR control confirm the superiority of PatchCore† over all the other models in every anatomical section and masking condition ($p_{\text{BH}} < 10^{-6}$). After correction, the performance gap between EfficientAD-M† and SimpleNet† is not significant ($p_{\text{BH}} > 0.05$). This indicates comparable effectiveness despite their different precision–recall trade-offs. In this subsection, FP_share is computed at image level, that is, using image-wise TP/FP/FN counts obtained from the anomalous vs. normal decision for each test frame.

Globally, masking reduces false negatives by −33.9% and false positives by −34.6%. By size class, the decrease is most marked on large (−47.1%), followed by medium (−33.3%), and then small (−25.0%). In terms of error composition, the share of FN on large drops from 27.4% to 22.0%; that on medium remains approximately stable (∼ 34%), while small become relatively more prevalent (38.7% → 43.9%). At the model level, the EfficientAD-M† and EfficientAD-S† variants show the largest reductions in total errors (−78.1% and − 76.3%, *respectively*), driven by strong drops in FPs (−87.0% and − 81.3%) and FNs (−55.6% and − 50.0%); PatchCore† improves by −34.8% (FP −78.6%, FN −15.6%) and MemSeg† by −22.7% (FP −23.1%, FN −22.2%). In contrast, SimpleNet† reduces FN to zero (−100.0%) but increases FP (+24.1%), with an increase in overall error (+11.7%), highlighting a more recall-oriented operating point. On real defective samples (Table 5), masking consistently improves image-level performance across methods. In particular, PatchCore† shows a marked gain over PatchCore (AUROC 97.1 ± 0.1 vs. 86.6 ± 0.7;

TABLE 4 | Comparison of the models on the custom dataset using image-level metrics, averaged across the three viewpoints ("lower back," "lower front," "lower center").

Model	AUROC	AU-PR	Precision	Recall	F1-Score
PatchCore	84.6 ± 7.2 (86.8 ± 0.9)	87.0 ± 7.1 (88.1 ± 0.7)	83.6 ± 8.3 (82.4 ± 1.5)	92.1 ± 2.6 (92.9 ± 1.4)	87.4 ± 4.0 (87.2 ± 0.4)
PatchCore†	**95.6 ± 0.6** (96.9 ± 0.1)	**96.9 ± 0.7** (97.9 ± 0.1)	93.6 ± 4.4 (92.4 ± 0.8)	94.6 ± 5.9 (94.8 ± 0.6)	**93.8 ± 1.8** (93.5 ± 0.2)
MemSeg	92.7 ± 1.8 (95.3 ± 1.6)	95.2 ± 1.6 (96.2 ± 1.6)	88.1 ± 3.6 (92.0 ± 3.1)	91.2 ± 4.2 (93.0 ± 5.3)	89.5 ± 1.4 (92.4 ± 2.4)
MemSeg†	90.1 ± 4.7 (92.7 ± 4.0)	93.7 ± 3.1 (94.4 ± 3.7)	87.5 ± 8.3 (90.9 ± 5.0)	91.2 ± 4.3 (88.8 ± 8.2)	89.0 ± 4.6 (89.5 ± 3.8)
EfficientAD-S	70.7 ± 9.9 (76.5 ± 6.3)	77.7 ± 8.8 (78.4 ± 4.3)	68.9 ± 9.9 (73.6 ± 6.4)	94.4 ± 6.4 (91.7 ± 5.7)	78.9 ± 3.9 (81.5 ± 5.3)
EfficientAD-S†	93.2 ± 2.3 (95.8 ± 3.2)	95.2 ± 2.0 (96.7 ± 2.8)	87.3 ± 3.1 (93.7 ± 7.0)	**96.3 ± 3.8** (96.5 ± 4.0)	91.5 ± 1.3 (94.9 ± 4.1)
EfficientAD-M	75.7 ± 7.5 (79.8 ± 6.7)	80.7 ± 8.3 (80.8 ± 7.1)	72.9 ± 7.6 (74.7 ± 5.8)	90.6 ± 7.1 (91.8 ± 7.3)	80.3 ± 2.8 (82.2 ± 4.7)
EfficientAD-M†	94.0 ± 2.0 (96.3 ± 2.2)	95.6 ± 1.7 (97.5 ± 1.5)	89.7 ± 3.3 (95.3 ± 4.8)	95.6 ± 3.5 (95.6 ± 4.3)	92.5 ± 1.8 (95.3 ± 2.3)
SimpleNet	81.5 ± 6.6 (85.3 ± 4.4)	86.2 ± 7.7 (85.8 ± 5.9)	79.4 ± 10.1 (80.6 ± 4.9)	89.4 ± 8.3 (89.6 ± 4.7)	83.3 ± 3.3 (84.7 ± 2.8)
SimpleNet†	94.2 ± 2.5 (96.6 ± 1.7)	96.5 ± 1.8 (97.6 ± 1.0)	**93.7 ± 4.0** (89.9 ± 3.4)	90.9 ± 4.1 (97.0 ± 3.6)	92.1 ± 2.0 (93.2 ± 2.4)

Note: Models marked with † employ ROI-based binary masking (Section 4). For each metric, values outside parentheses refer to the test set, while values in parentheses refer to the validation set. Results are reported as mean ± standard deviation over three independent runs. The best results are highlighted in bold.

TABLE 5 | Comparison of the evaluated models on real defective samples using image-level metrics, averaged across the three viewpoints (lower back, lower front, lower center).

Model	AUROC	AU-PR	Precision	Recall	F1-Score
PatchCore	86.6 ± 0.7	87.7 ± 0.8	82.0 ± 1.1	92.9 ± 1.7	87.1 ± 0.8
PatchCore†	97.1 ± 0.1	**98.1 ± 0.1**	93.4 ± 1.1	94.4 ± 1.1	93.8 ± 0.8
MemSeg	96.3 ± 8.8	94.0 ± 13.6	**97.6 ± 5.8**	91.7 ± 20.4	93.2 ± 13.3
MemSeg†	98.3 ± 2.2	94.5 ± 7.0	88.9 ± 13.5	98.5 ± 5.0	92.8 ± 7.6
EfficientAD-S	87.3 ± 6.6	61.1 ± 21.5	61.8 ± 10.7	84.8 ± 15.7	70.6 ± 10.3
EfficientAD-S†	98.1 ± 2.0	93.2 ± 6.8	85.6 ± 14.5	98.5 ± 5.0	91.0 ± 8.9
EfficientAD-M	88.4 ± 7.1	64.1 ± 22.3	66.1 ± 15.4	80.3 ± 16.0	70.6 ± 10.3
EfficientAD-M†	97.8 ± 2.3	91.5 ± 6.9	80.5 ± 12.9	**100.0 ± 0.0**	88.7 ± 7.6
SimpleNet	95.0 ± 5.7	80.2 ± 21.5	82.9 ± 19.7	90.7 ± 12.1	84.8 ± 12.8
SimpleNet†	**99.5 ± 0.8**	98.0 ± 3.4	95.2 ± 7.1	**100.0 ± 0.0**	**97.4 ± 3.8**

Note: Models marked with † employ ROI-based binary masking (Section 4). Results are reported as Mean ± Standard Deviation over three independent runs. The best results are highlighted in bold.

AU-PR 98.1 ± 0.1 vs. 87.7 ± 0.8), confirming the benefit of ROI filtering in the presence of structural clutter. Among all models, SimpleNet† achieves the highest AUROC (99.5 ± 0.8) and F1-Score (97.4 ± 3.8), while EfficientAD-M† and SimpleNet† reach perfect recall (100.0 ± 0.0), consistent with a recall-oriented operating point on real defects.

7.2 | Pixel-Level Metrics

As reported in Table 6, the two *PatchCore* variants still deliver the strongest pixel-wise detection performance: PatchCore achieves the highest AUROC (99.7%) and AUPRO (98.6%), while the masked version (PatchCore†) attains the best AU-PR (34.2%) and the highest recall (58.6%). In contrast, *MemSeg* provides the most accurate spatial localization of anomalies, yielding the top values for IoU (27.8%), precision (40.5%), and the overall best F1-Score (43.4%).

Consistent trends are observed on real defective samples (Table 7). In particular, PatchCore† achieves the highest AUROC (99.9 ± 0.0) and AUPRO (99.4 ± 0.0), confirming its strong pixel-level separability. Conversely, MemSeg† provides the best localization quality, reaching the top AU-PR (42.4 ± 6.5), IoU

TABLE 6 | Comparison of the models on the custom dataset using pixel-level metrics, averaged across the three viewpoints ("lower back," "lower front," "lower center").

Model	AUROC	AU-PR	AUPRO	IoU	Precision	Recall	F1-Score
PatchCore	**99.7 ± 0.1** (99.8 ± 0.0)	22.2 ± 7.4 (20.3 ± 0.1)	**98.6 ± 0.1** (97.8 ± 0.1)	20.3 ± 5.2 (18.1 ± 0.1)	24.9 ± 7.1 (25.1 ± 0.6)	53.4 ± 11.0 (40.6 ± 3.0)	33.5 ± 7.4 (30.3 ± 0.1)
PatchCore†	99.2 ± 1.0 (99.9 ± 0.0)	**34.2 ± 7.2** (28.8 ± 0.0)	98.5 ± 1.4 (99.4 ± 0.0)	26.3 ± 5.7 (22.8 ± 0.1)	32.5 ± 8.0 (29.6 ± 0.5)	**58.6 ± 6.7** (51.9 ± 1.1)	41.3 ± 6.9 (37.1 ± 0.1)
MemSeg	94.1 ± 3.7 (95.6 ± 4.2)	33.8 ± 5.0 (24.5 ± 7.3)	89.8 ± 4.9 (91.1 ± 5.5)	**27.8 ± 3.6** (20.8 ± 4.4)	**40.5 ± 4.9** (32.0 ± 7.3)	47.0 ± 5.1 (37.5 ± 6.7)	**43.4 ± 4.5** (34.3 ± 6.2)
MemSeg†	93.3 ± 4.4 (93.9 ± 4.6)	34.1 ± 4.8 (25.7 ± 8.7)	87.6 ± 5.2 (89.6 ± 6.3)	26.4 ± 3.1 (21.4 ± 5.4)	37.7 ± 5.0 (30.4 ± 7.4)	47.1 ± 6.0 (42.5 ± 9.7)	41.6 ± 3.9 (34.9 ± 7.5)
EfficientAD-S	99.4 ± 0.4 (99.5 ± 0.5)	8.6 ± 1.7 (5.3 ± 4.9)	93.6 ± 3.2 (94.6 ± 3.1)	9.9 ± 1.1 (5.7 ± 4.0)	12.9 ± 2.8 (9.2 ± 7.8)	32.9 ± 8.4 (26.3 ± 13.5)	17.9 ± 1.9 (10.5 ± 7.2)
EfficientAD-S†	**99.7 ± 0.1** (99.9 ± 0.0)	14.2 ± 4.7 (8.2 ± 3.9)	97.9 ± 1.2 (98.6 ± 1.2)	13.4 ± 3.7 (9.5 ± 4.1)	15.6 ± 4.9 (11.8 ± 5.8)	50.4 ± 5.1 (46.4 ± 18.6)	23.4 ± 5.6 (17.1 ± 6.9)
EfficientAD-M	99.4 ± 0.8 (99.5 ± 0.4)	8.1 ± 1.4 (4.8 ± 4.2)	94.2 ± 3.1 (94.8 ± 3.2)	9.5 ± 1.4 (5.3 ± 3.0)	12.2 ± 1.7 (10.6 ± 10.6)	31.3 ± 7.6 (22.7 ± 11.0)	17.3 ± 2.3 (9.9 ± 5.4)
EfficientAD-M†	99.4 ± 0.5 (99.9 ± 0.0)	12.1 ± 2.9 (7.1 ± 3.4)	98.0 ± 1.0 (98.3 ± 1.3)	11.8 ± 2.6 (7.8 ± 2.8)	13.9 ± 3.1 (10.1 ± 5.3)	46.4 ± 12.2 (38.1 ± 13.7)	21.0 ± 4.1 (14.3 ± 4.8)
SimpleNet	94.4 ± 8.6 (97.7 ± 2.3)	7.0 ± 2.8 (5.3 ± 4.0)	88.3 ± 5.2 (90.6 ± 6.5)	0.0 ± 0.0 (0.0 ± 0.0)	10.0 ± 4.0 (9.8 ± 7.0)	35.3 ± 10.2 (25.1 ± 12.5)	15.2 ± 5.2 (11.0 ± 6.8)
SimpleNet†	97.8 ± 2.7 (99.7 ± 0.2)	9.7 ± 2.4 (6.7 ± 2.5)	96.1 ± 1.6 (97.5 ± 1.7)	0.0 ± 0.0 (0.0 ± 0.0)	12.5 ± 3.1 (11.2 ± 5.5)	38.5 ± 10.5 (26.0 ± 10.1)	18.4 ± 3.9 (14.1 ± 4.8)

Note: Models marked with † employ ROI-based binary masking (Section 4). Values outside parentheses refer to the test set; values in parentheses refer to the validation set. Results are Mean ± Standard Deviation over three independent runs. The best results are highlighted in bold.

TABLE 7 | Comparison of the evaluated models on real defective samples using pixel-level metrics, averaged across the three viewpoints (lower back, lower front, lower center).

Model	AUROC	AU-PR	AUPRO	IoU	Precision	Recall	F1-Score
PatchCore	99.8 ± 0.0	20.5 ± 0.2	97.9 ± 0.0	18.1 ± 0.3	25.4 ± 1.6	39.7 ± 4.2	30.4 ± 0.4
PatchCore†	**99.9 ± 0.0**	28.8 ± 0.0	**99.4 ± 0.0**	22.8 ± 0.1	29.8 ± 0.1	51.8 ± 0.9	37.1 ± 0.1
MemSeg	94.7 ± 4.7	36.7 ± 11.2	85.3 ± 9.3	29.2 ± 9.0	39.9 ± 12.1	51.7 ± 9.6	44.6 ± 11.7
MemSeg†	95.3 ± 4.4	**42.4 ± 6.5**	86.4 ± 10.1	**31.3 ± 4.0**	**42.2 ± 6.4**	56.2 ± 9.3	**47.6 ± 4.7**
EfficientAD-S	99.6 ± 0.2	12.5 ± 1.1	96.8 ± 1.5	12.7 ± 1.5	15.5 ± 1.8	41.7 ± 6.2	22.5 ± 2.3
EfficientAD-S†	99.7 ± 0.2	19.1 ± 5.0	94.1 ± 9.0	17.1 ± 4.5	19.8 ± 5.3	**56.3 ± 11.8**	28.9 ± 6.8
EfficientAD-M	99.7 ± 0.1	12.0 ± 3.0	98.3 ± 0.8	12.5 ± 2.7	15.7 ± 4.2	40.0 ± 7.0	22.1 ± 4.2
EfficientAD-M†	99.3 ± 0.7	15.6 ± 5.3	93.6 ± 8.5	14.3 ± 4.4	17.2 ± 4.3	48.0 ± 18.4	24.7 ± 6.7
SimpleNet	99.6 ± 0.1	9.8 ± 1.9	96.6 ± 1.6	0.1 ± 0.0	13.4 ± 2.2	38.5 ± 16.3	19.3 ± 3.4
SimpleNet†	99.3 ± 0.6	11.3 ± 2.2	91.1 ± 11.2	0.1 ± 0.0	13.2 ± 3.2	42.5 ± 11.6	19.7 ± 4.1

Note: Models marked with † employ ROI-based binary masking (Section 4). Results are reported as Mean ± Standard Deviation over three independent runs. The best results are highlighted in bold.

(31.3 ± 4.0), precision (42.2 ± 6.4), and F1-Score (47.6 ± 4.7). EfficientAD-S† attains the highest recall on real samples (56.3 ± 11.8), at the cost of lower precision and IoU, indicating a more recall-oriented operating point.

In summary, PatchCore is the most reliable choice when the primary goal is to detect whether a pixel is anomalous at all. Meanwhile, MemSeg excels when fine-grained segmentation quality is required. In this subsection, FP_share is computed at pixel level that is, using pixel-wise TP/FP/FN counts derived from the thresholded anomaly map and the GT mask.

7.3 | Inference Time

Despite the superior accuracy of PatchCore and MemSeg in pixel-level metrics, EfficientAD-S and EfficientAD-M distinguish themselves with significantly faster inference times, as shown in

TABLE 8 | Average per-image inference time (s/image) for each model, measured on a single GPU with batch size equal to 1.

Model	Inference time (s/image)
PatchCore	3.33
MemSeg	0.17
EfficientAD-S	0.09
EfficientAD-M	0.10
SimpleNet	0.26

Table 8. In particular, EfficientAD-S achieves the fastest inference time at 0.09 s per image, followed closely by EfficientAD-M at 0.10 s per image. In contrast, PatchCore is considerably slower, with an inference time of 3.33 s per image. Despite its high accuracy, this may limit its practicality for real-time applications. MemSeg maintains a moderate inference time of 0.17 s per image, balancing speed and accuracy effectively. The application of mask filtering (†) significantly enhances PatchCore and MemSeg's pixel-level performance. AUROC increases by 1.4% and 0.5%, respectively.

7.4 | Qualitative Results

The qualitative analysis, depicted in Figures 9–11, complements the quantitative results by providing visual insights into the models' anomaly detection capabilities.

PatchCore and MemSeg produce more precise and clearly defined segmentations of defect regions with minimal noise. This makes them well-suited for tasks requiring fine-grained detection. In contrast, EfficientAD-M and SimpleNet generate noisier outputs, particularly when identifying finer details. This highlights limitations in their defect localization capabilities within complex environments. For instance, Figure 9b shows MemSeg's precise detection of intricate defects, whereas Figure 9c illustrates EfficientAD-M's struggle with subtle anomalies, resulting in scattered noise around defect boundaries.

Finally, Figures 12–14 illustrate representative misclassified samples for each underbody region, showing the original input image, the corresponding GT mask, and the anomaly heat-map generated by every model side-by-side.

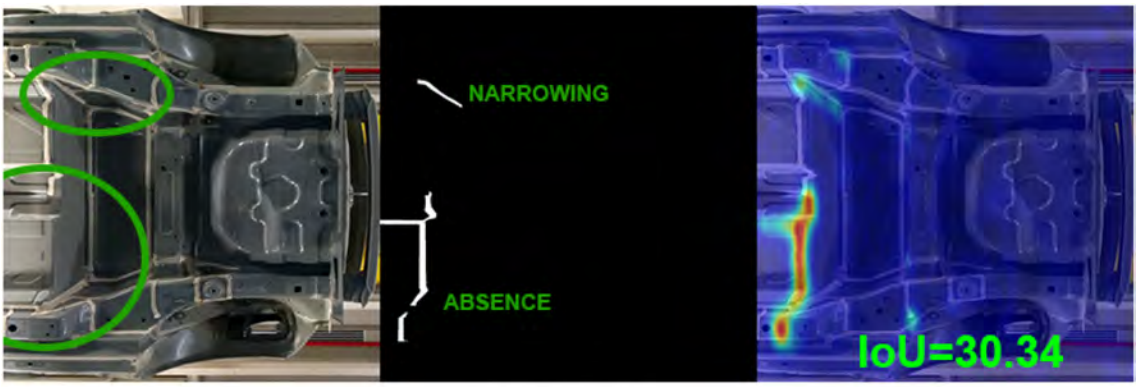
8 | Discussion

This section explains the experimental results and discusses deployment, real-world integration, current limitations, and future work.

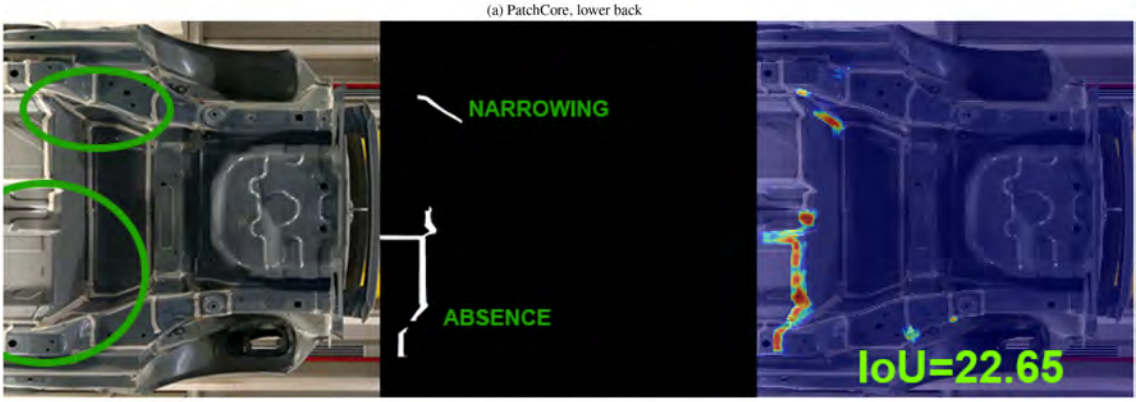
8.1 | Model Performance and Interpretation

The qualitative results reveal distinct differences in how models handle the spatial characteristics of anomalies.

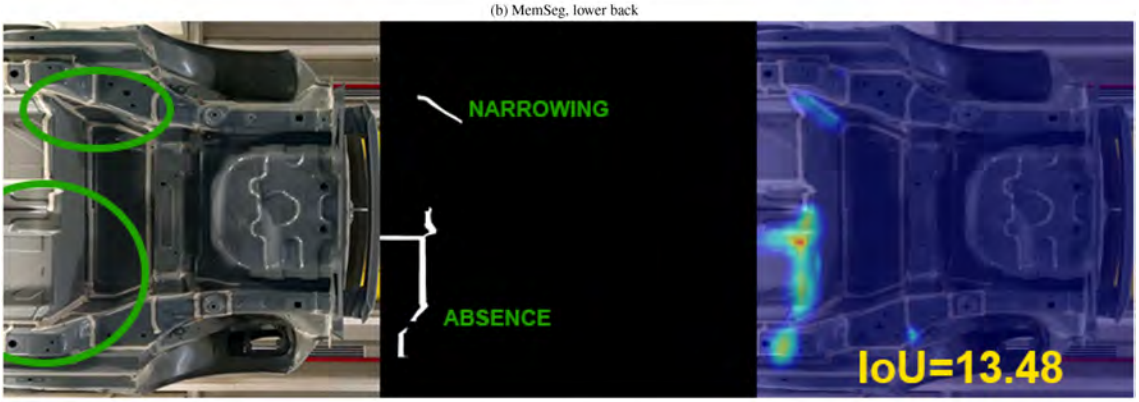
- *Model segmentation behavior*. PatchCore and MemSeg exhibit coherent and localized segmentation behavior, producing dense and well-contoured anomaly maps. This makes them particularly suitable for applications requiring accurate spatial resolution, such as manual inspection or robotic correction of fine-grained defects. Both models clearly benefit from mask filtering: removing irrelevant regions before inference reduces noise and yields more focused, interpretable predictions.
- *Challenges in detection precision*. Conversely, EfficientAD-M and SimpleNet tend to produce noisier heatmaps, often struggling to isolate defect boundaries in the presence of subtle or low-contrast anomalies. These limitations may stem from architectural differences or a reduced ability to incorporate contextual information. In some cases, the models flag large, imprecise regions, which may trigger unnecessary interventions or lead to missed corrections. In production settings requiring fine spatial precision, such behavior may reduce trust in the system, highlighting the need for architectures tailored to localization rather than global classification.
- *Common causes of misclassification*. Many missed detections are attributable to recurring factors such as small defect size or low color contrast. Most errors originate from a few tens of pixels, whose signal is attenuated during patch-based processing. In addition, defect colors often fall within the typical RGB variation of painted metal, resulting in feature distances below the detection threshold. Consequently, the model may fail to distinguish the defect from the surrounding material. Illumination changes can also introduce distributional shifts, further degrading performance and potentially necessitating model tuning or retraining.
- *Influence of structural noise*. Structural features such as ribs, welds, and specular reflections introduce high intra-class variance, forcing models to raise detection thresholds and suppressing subtle anomalies. For instance, in the lower part of Figure 12, PatchCore and MemSeg are misled by spurious reflections, missing the sealing error. Likewise, in the lower center of Figure 13, a small sealing error yields uniform heatmaps across all models, as feature descriptors vary more within the background than between the background and the defect. In addition to the quantitative results, Figures 12–14 allow for a more detailed analysis of misclassifications. These examples highlight the main difficulties encountered by models in different underbody regions. Small defects or defects with low chromatic contrast are often attenuated in patch-based feature descriptors, while structural reflections and illumination variations can generate false positives, causing models to mistake structural elements for real anomalies. In particular, PatchCore and MemSeg tend to produce consistent segmentations but can be fooled by structural noise, while EfficientAD and SimpleNet, still maintaining good image-level performance, generate noisier and less precise heatmaps in areas of detail. Integrating these qualitative observations with quantitative metrics provides a more complete view of the strengths and limitations of different approaches, clarifying the trade-offs between overall accuracy and fine-grained localization capabilities.
- *Practical implications of FP/FN balancing*. Error-balancing analysis (FP_share derived from Precision/Recall) clarifies the operational behavior: most methods are FP-dominated



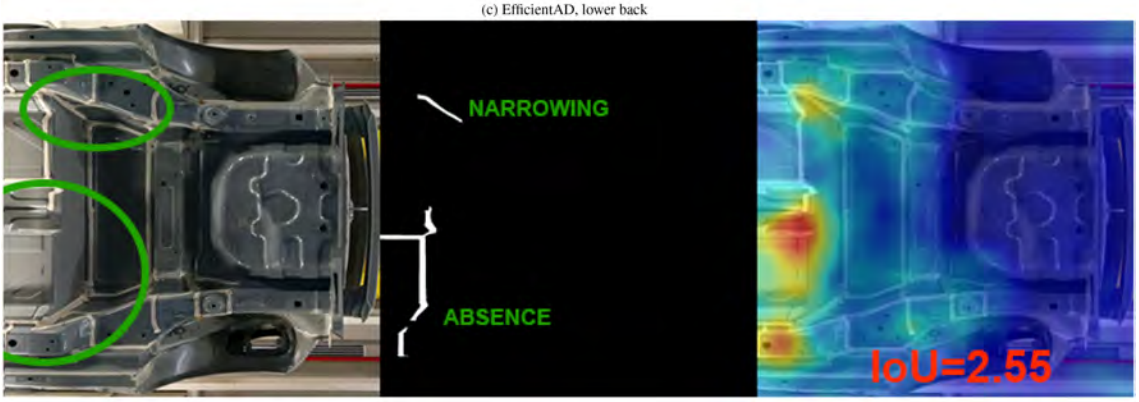
(a) PatchCore, lower back



(b) MemSeg, lower back

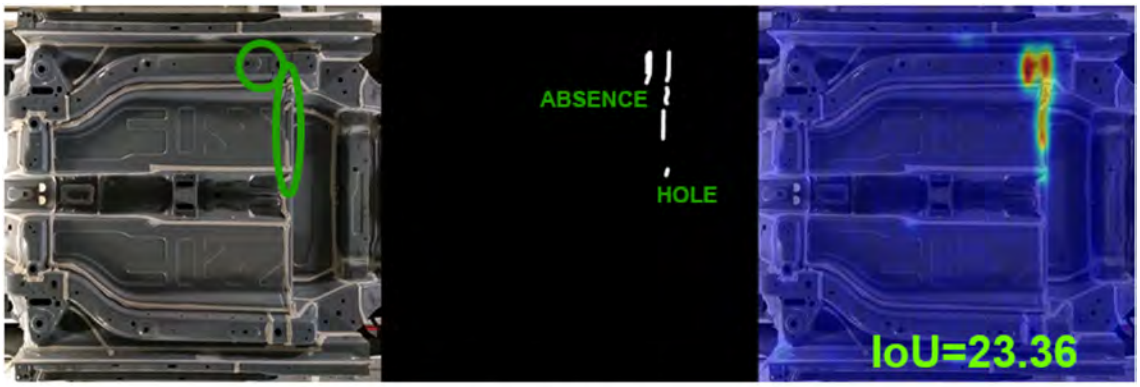


(c) EfficientAD, lower back

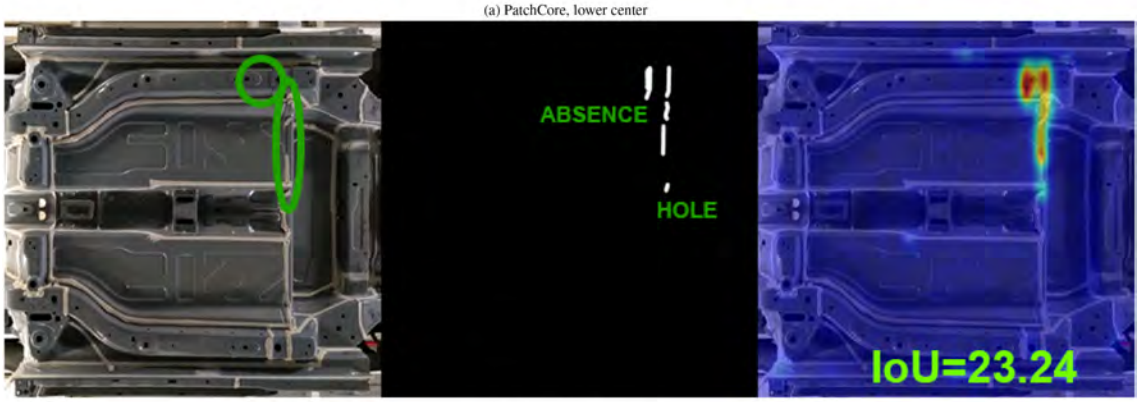


(d) SimpleNet, lower back

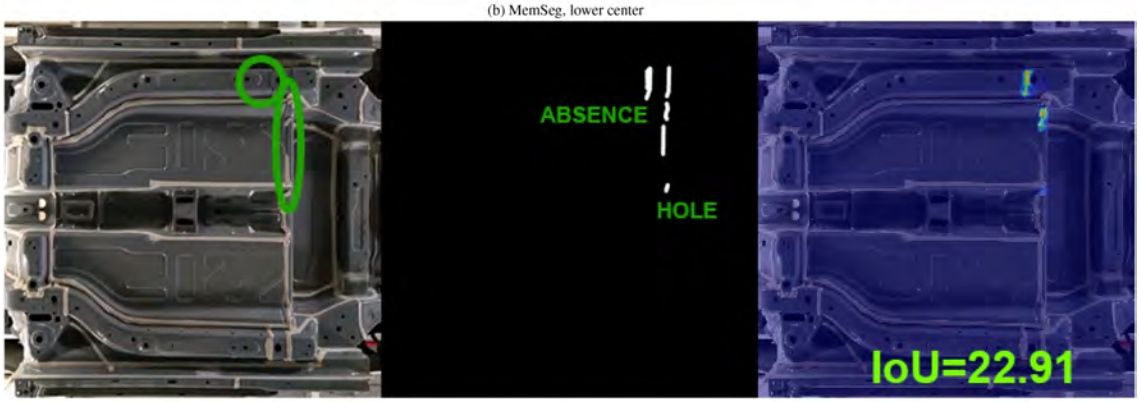
FIGURE 9 | Qualitative results for the lower back viewpoint. For each model, the panels show, from left to right, the original input image, the Ground-Truth (GT) mask, and the predicted anomaly heatmap. Defects in the input are marked with green indicators. When reported, IoU values refer to the thresholded heatmap at the operating point defined in Section 7.



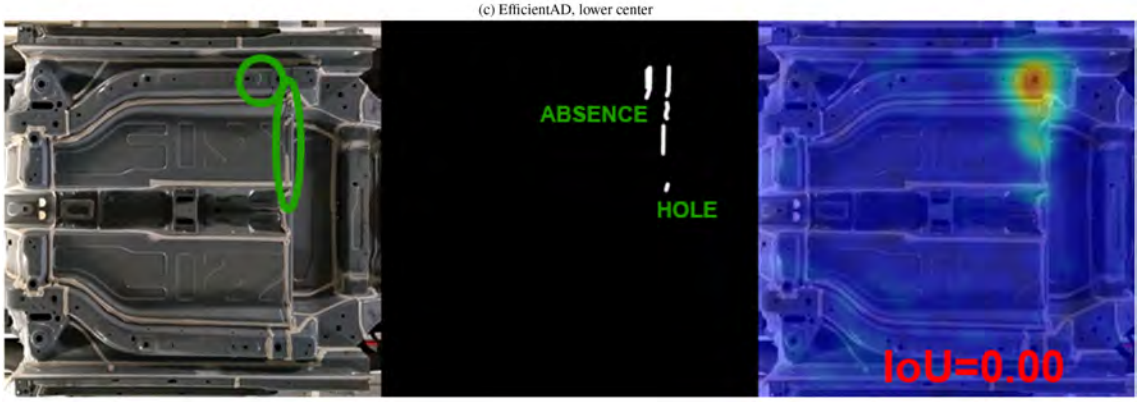
(a) PatchCore, lower center



(b) MemSeg, lower center



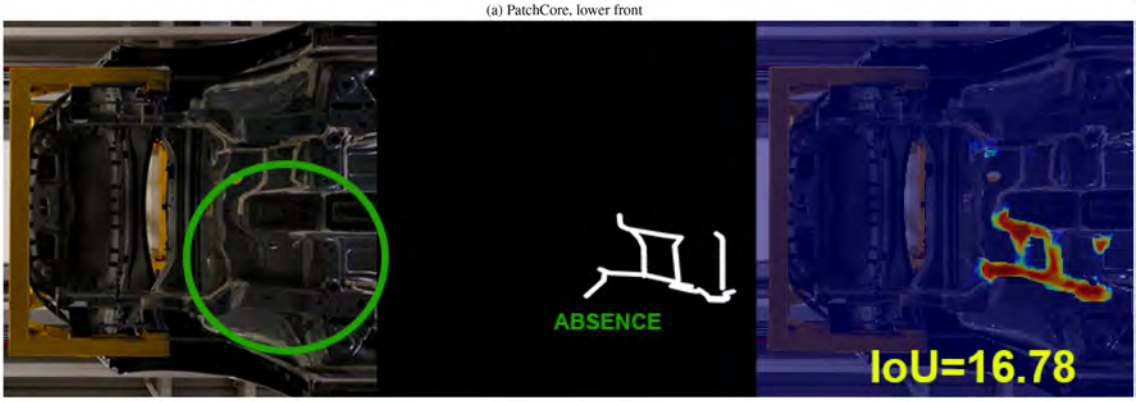
(c) EfficientAD, lower center



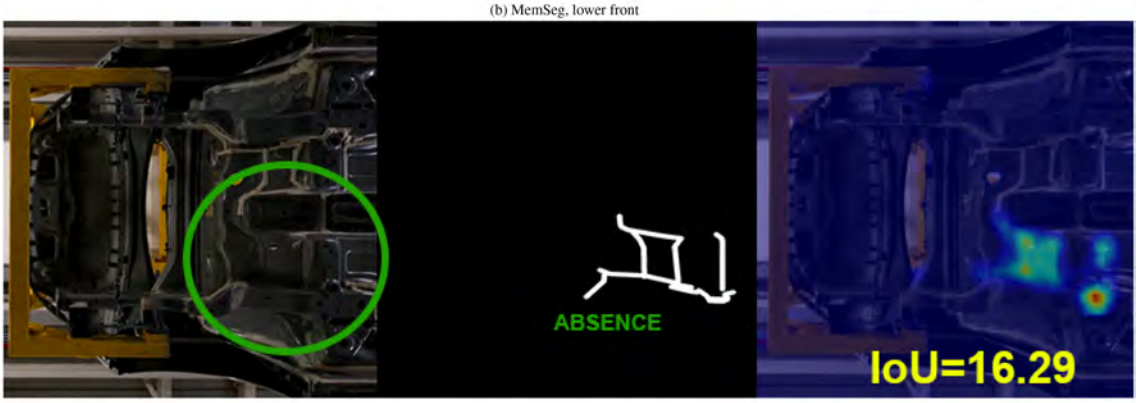
(d) SimpleNet, lower center

FIGURE 10 | Qualitative results for the lower center viewpoint. For each model, the panels show, from left to right, the original input image, the Ground-Truth (GT) mask, and the predicted anomaly heatmap. Defects in the input are marked with green indicators. When reported, IoU values refer to the thresholded heatmap at the operating point defined in Section 7.

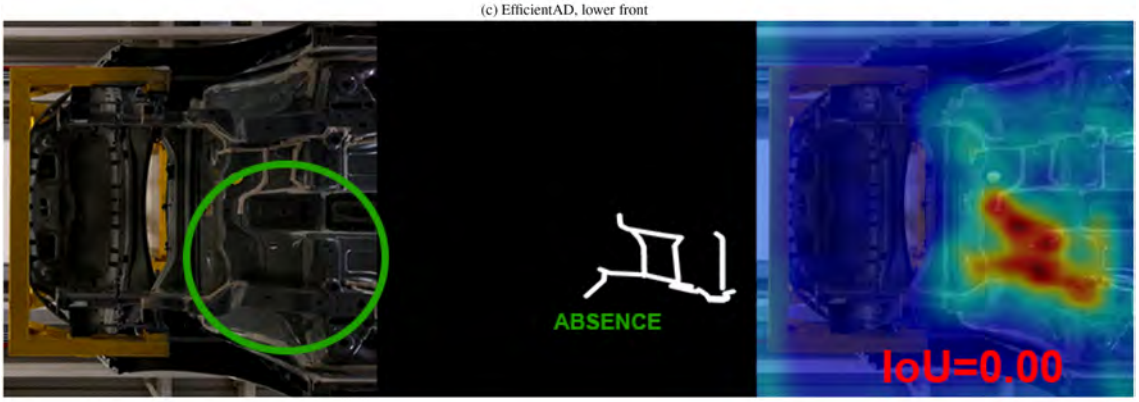
(a) PatchCore, lower front



(b) MemSeg, lower front



(c) EfficientAD, lower front



(d) SimpleNet, lower front

FIGURE 11 | Qualitative results for the lower front viewpoint. For each model, the panels show, from left to right, the original input image, the Ground-Truth (GT) mask, and the predicted anomaly heatmap. Defects in the input are marked with green indicators. When reported, IoU values refer to the thresholded heatmap at the operating point defined in Section 7.

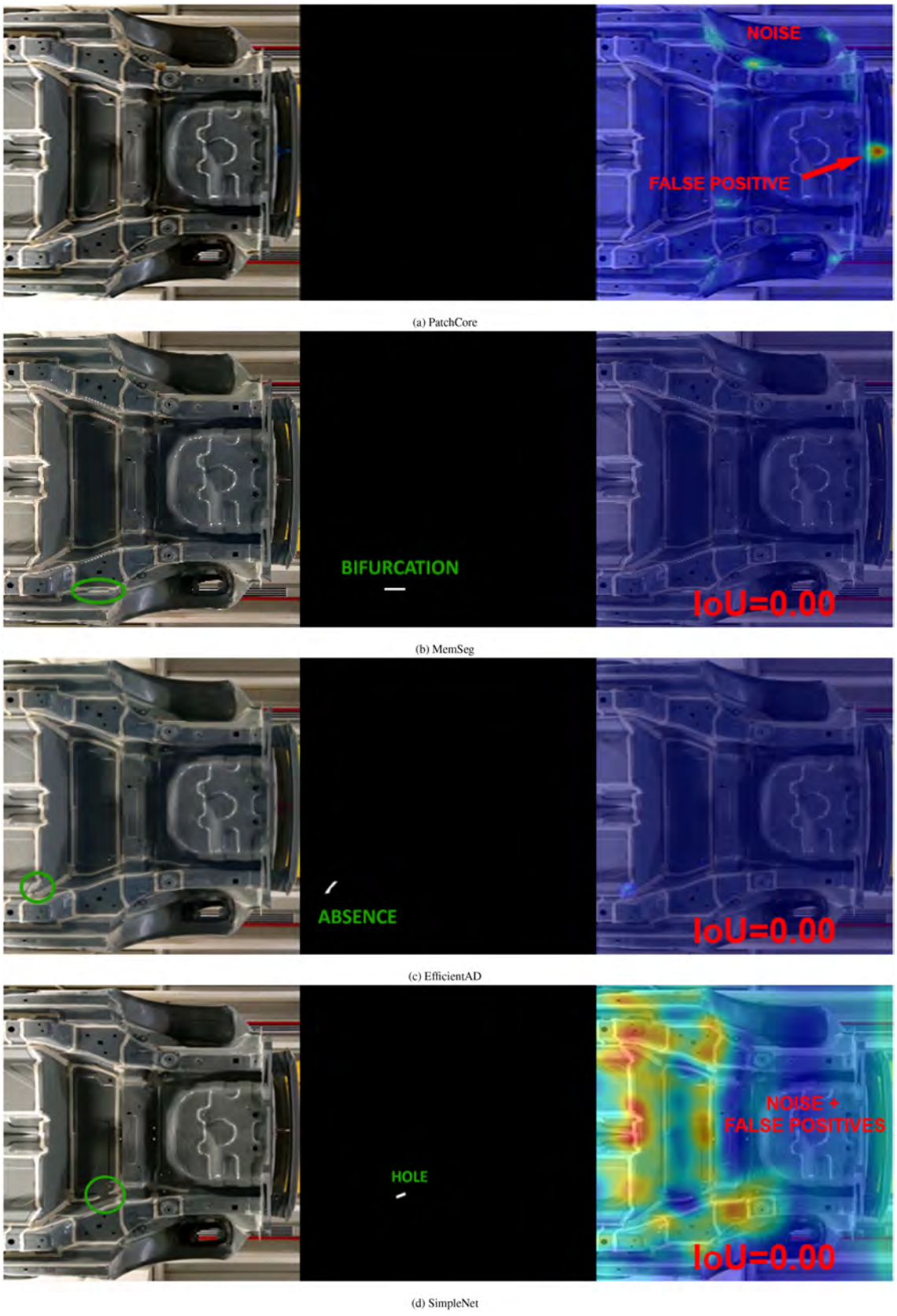


FIGURE 12 | Representative misclassified samples for the lower back viewpoint. For each model, the panels show, from left to right, the original input image, the Ground-Truth (GT) mask, and the predicted anomaly heatmap. Green markers highlight the suspected defect location in the input. For normal samples incorrectly flagged as anomalous, short notes indicate the main source of false positives (e.g., reflections or structural edges).

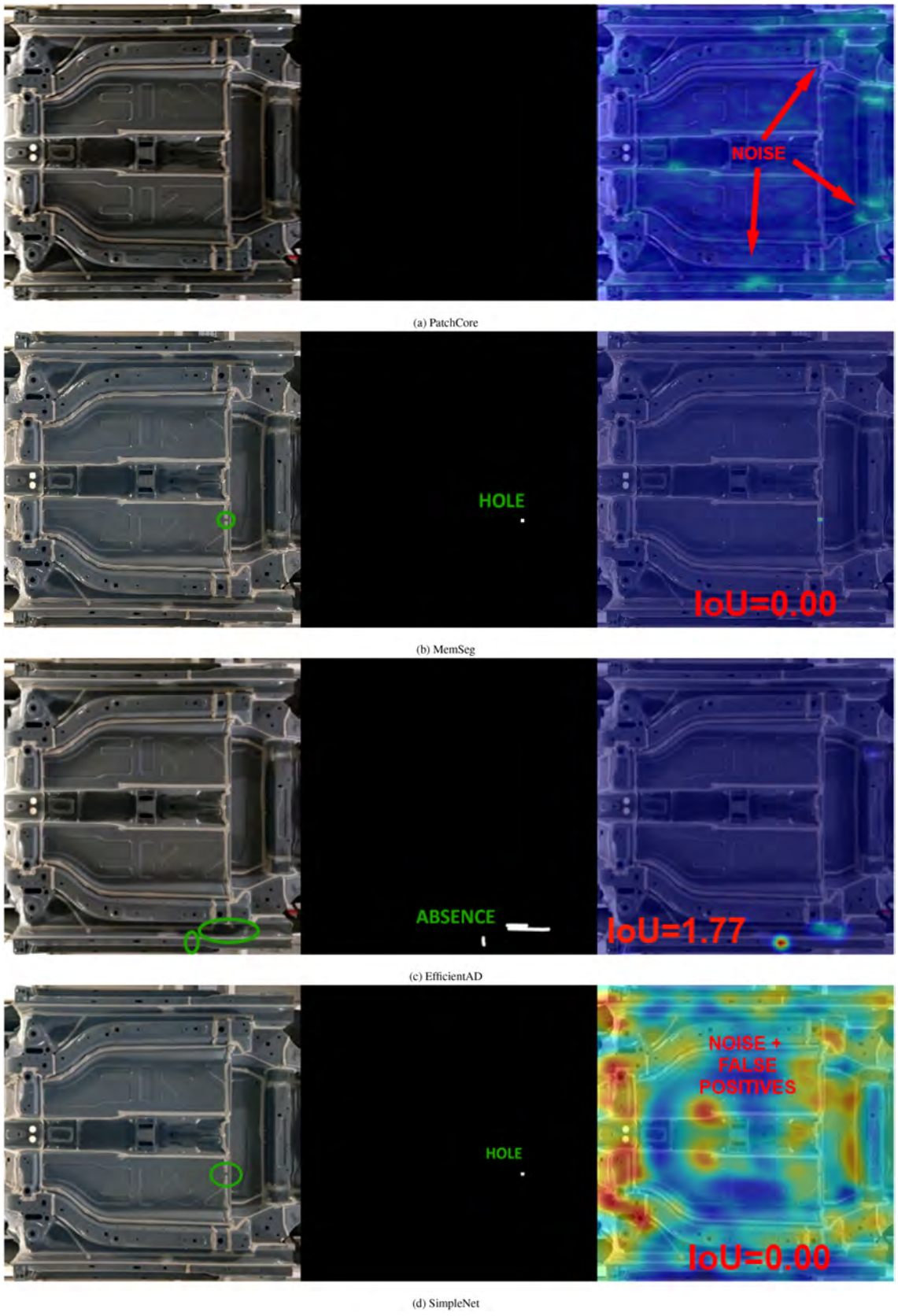


FIGURE 13 | Representative misclassified samples for the lower center viewpoint. For each model, the panels show, from left to right, the original input image, the Ground-Truth (GT) mask, and the predicted anomaly heatmap. Green markers highlight the suspected defect location in the input. For normal samples incorrectly flagged as anomalous, short notes indicate the main source of false positives (e.g., reflections or structural edges).

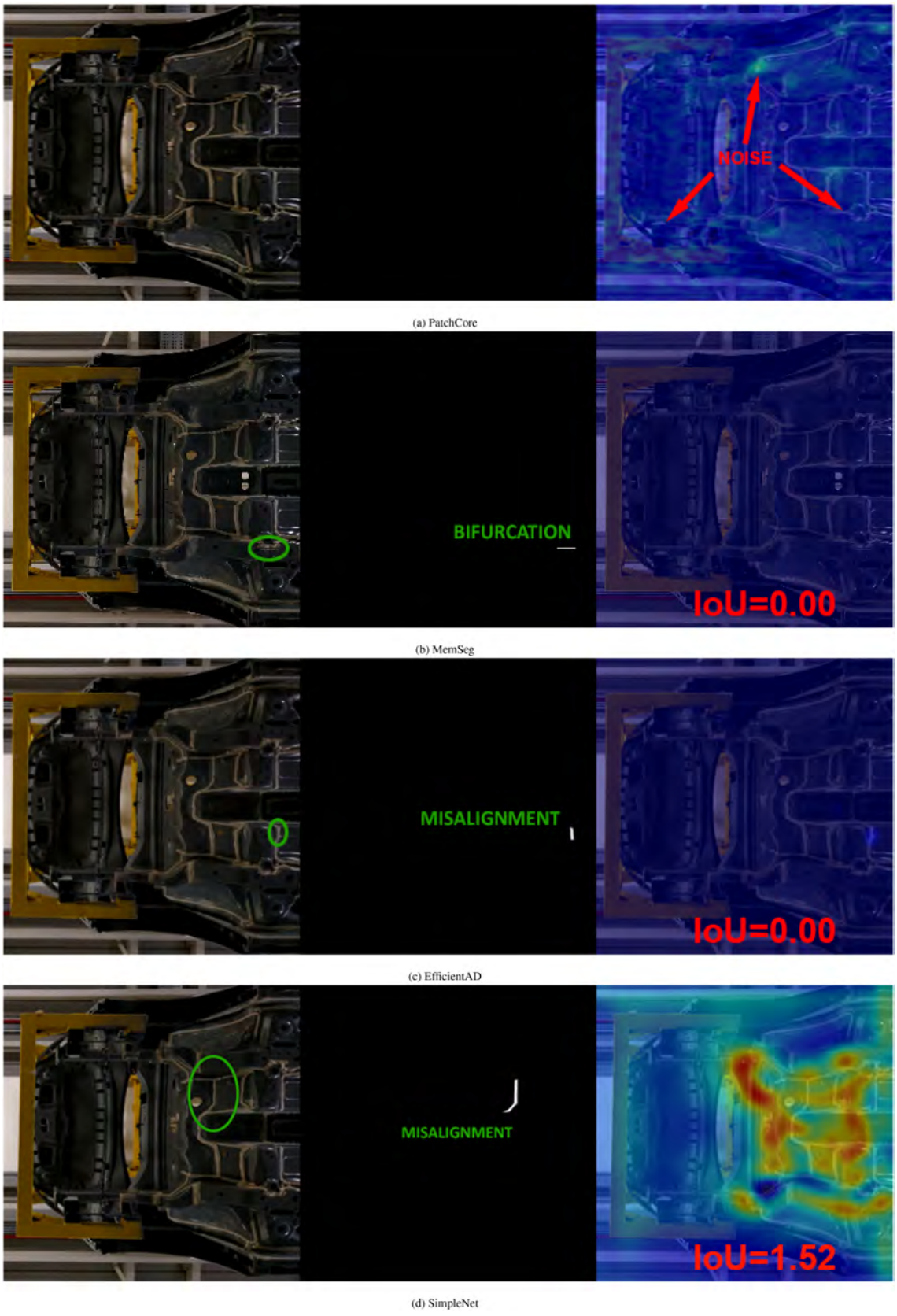


FIGURE 14 | Representative misclassified samples for the lower front viewpoint. For each model, the panels show, from left to right, the original input image, the Ground-Truth (GT) mask, and the predicted anomaly heatmap. Green markers highlight the suspected defect location in the input. For normal samples incorrectly flagged as anomalous, short notes indicate the main source of false positives (e.g., reflections or structural edges).

at the pixel level (consistent with lower pixel-wise precisions at high coverage), while MemSeg shows the lightest FP load; at the image level, PatchCore† is nearly balanced, EfficientAD is more aggressive (more FP), and SimpleNet† is more conservative (more FN). These results suggest practical guidelines for usage: PatchCore† when a reliable "defect yes/no" decision on the image is needed; MemSeg when fine-grained segmentation is the priority; EfficientAD when latency is the main constraint and a higher number of false alarms is acceptable; SimpleNet† when precision needs to be preserved while accepting lower recall. In addition, the size-aware analysis of errors across defect-area bins indicates that larger anomalies benefit more from masking than small, low-contrast ones, further motivating future work on explicit, size-dependent threshold tuning in collaboration with domain experts.

- *Sensor and lighting configurations*. While the current study focuses on a single, fixed RGB viewpoint to meet the constraints of the inspected sealing cell, different sensor and lighting configurations can further improve coverage and robustness in production. First, a multi-view RGB setup (e.g., two to four synchronized viewpoints) can mitigate viewpoint-dependent occlusions and appearance variations, particularly in the lateral and frontal sealant paths, which are partially unobservable from the current camera positioning. In this case, anomaly detection can be performed per view and combined via late fusion, for example, by aggregating image-level scores (maximum/average) and merging the anomaly maps pixel-by-pixel after geometric alignment. Second, RGB-D detection can provide complementary geometric clues that are difficult to infer from RGB alone, such as thickness-related irregularities or geometry-dependent defects near structural elements. Finally, designing a good lighting system can improve results: improving uniformity and reducing specular highlights through controlled lighting, stable mounting, and consistent exposure settings can reduce photometric domain shift and stabilize anomaly maps over time. These options introduce additional integration costs (calibration, synchronization, maintainability) and latency constraints; however, they provide a clear roadmap for extending the current baseline when full underbody coverage and greater photometric robustness are required.

To summarize the findings across all evaluation dimensions, the comparative analysis reveals nuanced trade-offs between detection accuracy, localization capability, and computational efficiency. Each evaluated model exhibits distinct strengths: while EfficientAD-M and SimpleNet offer faster inference and solid image-level performance, MemSeg and PatchCore excel in pixel-wise localization, making them more suitable for tasks requiring precise spatial guidance.

8.2 | Deployment Considerations and Real-World Integration

This section outlines the practical requirements and trade-offs involved in deploying ADL systems within real-world manufacturing environments, particularly in the context of the MANAGE 5.0 project.

- *Trade-offs between accuracy and localization*. While EfficientAD-M achieves the highest image-level AUROC and AU-PR, it slightly compromises on pixel-level IoU compared to MemSeg. This indicates that EfficientAD-M is better suited for applications prioritizing overall anomaly detection accuracy, whereas MemSeg is preferable for tasks requiring precise localization. These findings reveal important trade-offs between accuracy and computational efficiency. EfficientAD-M offers a compelling balance between high image-level performance and rapid inference, making it suitable for real-time applications. However, further refinements are needed to reduce output noise for scenarios demanding precise localization. Similarly, SimpleNet demonstrates strong image-level metrics but exhibits lower pixel-level precision, indicating a need for improved segmentation capabilities.

- *Architectural configurations for real-time deployment*. For deployment, ADL systems must be accurate, fast, and compliant with production line timing and structural constraints. In sealant application, there is typically a short but usable interval between the sealing process and the opportunity for inspection or correction. This time window, defined by the car body's transit from the sealing station to the next workstation, allows for near real-time detection. A practical baseline setup involves a fixed RGB camera in the sealing cell, connected to an industrial PC with a high-performance GPU. This configuration supports frame-by-frame inference with minimal latency. To improve coverage, particularly of lateral and frontal regions, a more advanced multi-camera RGB system can be employed. This would capture the underbody from multiple angles, ensuring full visibility of sealant paths and increasing the likelihood of detecting subtle anomalies.

  In a representative deployment, the system could consist of 4 IP-based cameras (e.g., NDI-HX compliant, Full HD, $\geq 80^{\circ}$ FOV, IP54-rated) anchored to the cell frame via adjustable mounts. These cameras would connect to a Gigabit PoE+ switch, which in turn connects to an industrial-grade PC. A practical deployment setup can be supported by a mid-range machine, for instance equipped with an Intel Core i7-12,700 or AMD Ryzen 77,700 CPU, 32 GB of RAM, and an NVIDIA RTX 3070 or RTX 4070 GPU. This hardware configuration offers a cost-effective balance between processing power and scalability, and is well-suited to the performance requirements of near real-time ADL applications. Based on empirical inference benchmarks conducted during the experimental analysis on a similar machine (equipped with an NVIDIA RTX 3090), such a system is expected to achieve inference times in the range of 90–170 milliseconds per frame for the evaluated models. This estimate depends on the specific architecture used (e.g., MemSeg, EfficientAD-M), the image resolution, and the batch size, but remains within the acceptable processing window available between sealing and subsequent inspection stages in a typical automotive production line. From an XR viewpoint, this translates into an end-to-end latency budget (acquisition, inference, rendering on the headset) on the order of a few hundred milliseconds, which comfortably fits within the time window between sealing and manual correction. In the MANAGE 5.0

use case, this is sufficient to keep anomaly overlays synchronized with the physical underbody motion, while preserving the accuracy–latency trade-off required for real-time decision support. Camera positioning and angle are also critical to ensuring full coverage; multi-view capture improves detection robustness for lateral and frontal sealant paths. Standard CAT6 Ethernet cabling provides both power (PoE) and data transmission, simplifying integration into the existing factory infrastructure.

- *Enhancements* via *additional sensing*. A further enhancement involves incorporating additional sensing modalities. For instance, depth cameras can provide 3D geometric information about sealant thickness, one of the most critical quality indicators not detectable with RGB imaging alone. Mounting such sensors on collaborative robots would allow for flexible viewpoint selection and enhanced defect characterization, albeit at the cost of increased system complexity and expense.
- *Timing and process integration*. Once anomalies are detected, effective visualization is essential for human–machine collaboration. Traditional displays (e.g., monitors, tablets) may distract operators from the workspace, introducing delays. A more integrated option is XR (AR/MR) headsets, such as HoloLens 2, which enable operators to see anomaly data overlaid in their field of view. These wearable devices support contextualized, hands-free guidance during inspection and repair [45–47].
- *Operator feedback and XR integration*. Several studies highlight the ergonomic and operational benefits of XR-based inspection interfaces. For example, Seeliger et al. [45] show that AR headsets improve task performance and user experience by delivering contextual visual information. Similarly, Muñoz et al. [47] found that MR interfaces enhance comfort and reduce cognitive load compared to traditional displays. Calandra et al. [46, 48] proposed various AR metaphors (3D arrows, text panels, and video overlays) that can be dynamically adapted based on the defect type. Building on these prior findings, several visual encoding strategies for XR integration are being evaluated within MANAGE 5.0. These include localized highlighting of the detected anomaly directly on the underbody surface, directional indicators designed to guide the operator's attention toward the affected region, and heatmap-style visualizations presented within the XR interface. These metaphors are intended to support rapid and intuitive understanding while preserving unobstructed visibility of the physical scene. In line with these findings, the design of the XR interface in MANAGE 5.0 will explicitly account for human factors such as limiting visual clutter, using interpretable color scales and metaphors for defect severity, and preserving operator trust by ensuring that overlays remain stable, consistent, and clearly distinguishable from background structures. These considerations are essential to avoid overloading the user and to foster effective human–AI collaboration in the production workspace.

The ultimate objective of the MANAGE 5.0 project is to identify the most effective ADL model for XR-based quality inspection. The selected model's outputs will guide the operator with contextualized visual cues. To achieve this, the system must operate in real time, providing timely feedback and enabling intervention during production [45]. Based on the observed trade-offs, MemSeg emerges as the most suitable choice, offering a balance between accurate localization and sufficiently fast inference to support real-time human–machine collaboration.

However, fully realizing this potential requires addressing technical challenges beyond model selection. These include improving spatial alignment between digital and physical environments (e.g., via object tracking or SLAM), developing adaptive visualizations based on defect types, and ensuring robust inference under typical industrial variability such as lighting changes, vibrations, and reflective surfaces. With these capabilities, XR-enhanced ADL systems can become integral components of automotive quality control.

8.3 | Model Generalizability and Transferability

Ensuring long-term model reliability in industrial scenarios requires more than just accuracy on a fixed dataset and instead calls for adaptive strategies that respond to evolving production contexts.

- *Handling domain shifts*. Robust model generalization and transferability in industrial inspection need more than basic components; they require a structured workflow for data collection, annotation, and model adaptation. Pre-trained frameworks excel at the distributions they were trained on, but when real-world conditions change, their internal representations may no longer capture the nuances of the new context. When domain shift occurs, e.g., due to camera angle changes, new parts, line modifications, or lighting variation, the features from pre-trained backbones can become less representative. This often leads to a degradation in performance metrics such as AUROC, IoU, and F1-score, as well as an increase in false positives and false negatives.
- *Adaptation through targeted fine-tuning*. To restore high levels of accuracy, it is therefore essential to carry out a targeted acquisition campaign. This involves collecting new images, both "normal" and defective, in the updated context, annotating them using the same scheme applied during the original training phase, and fine-tuning the existing weights. This helps maintain the capabilities already learned while adapting to the new domain.
- *Minimizing annotation effort*. To contain labeling costs and accelerate production release, transfer learning and domain adaptation techniques can be combined. These approaches reduce the amount of labeled data needed, although they do not completely eliminate the collection of target-specific examples.
- *Continuous training and monitoring*. The integration of a continuous training and evaluation pipeline is key to long-term reliability. This includes automatic monitoring of low-confidence samples during production, dataset versioning based on hardware or software changes, and periodic scheduling of retraining jobs for updated models. In this context, integration with lightweight IoT communication protocols such as MQTT can enable real-time transmission of serialized detection results (e.g., using Google Protobuf) to

external visualization or decision-making modules, enhancing system interoperability.

- *Hardware flexibility and sensor integration*. From a deployment viewpoint, the fixed-camera setup catches large anomalies visible to humans, reducing the need for continuous manual oversight. However, capturing finer or less obvious defects (i.e., those that even trained operators might miss) may require enhanced viewpoint flexibility. This can be achieved either by repositioning static cameras closer to the inspected surfaces or, as previously discussed, mounting additional sensors on a robotic arm to enable on-demand scanning and increased visual detail.
- *Sensing occluded sealants*. Finally, in cases where the sealant is occluded due to subsequent processes (e.g., PVC coating), inspection becomes infeasible for both human and machine vision. Building on the previous discussion of additional sensing modalities, combining RGB and depth data could enable indirect assessment of hidden features. For instance, the mentioned robotic arm could be equipped with an RGB-D camera (e.g., Intel RealSense) to estimate sealant thickness through the coating layer and flag potential anomalies that are currently undetectable.
- *Scaling to other lines and defect types*. Beyond sealant inspection, scaling the framework to different production lines or defect types will require addressing a number of practical constraints. For instance, the production layout must include a dedicated inspection stage with enough physical space for camera positioning and controlled lighting. The workpiece must remain consistently aligned with respect to the viewpoint, and differences in geometries or process cycles across parts need to be accounted for. This implies identifying distinct workpiece types, managing inference-time model selection or configuration, and collecting targeted training data accordingly. Furthermore, defining which defects are within the scope of detection—and excluding irrelevant variations through masking strategies—is essential to avoid false positives and ensure operational reliability in broader industrial contexts.

These additions would extend the applicability of the system to more complex scenarios and help maintain high levels of detection accuracy over time.

8.4 | Limitations and Future Work

The present study focuses on evaluating ADL algorithms in a complex, realistic industrial scenario. However, there are a few limitations that should be addressed in future research to generalize the present results.

- *Limited real-world defect variability*. The most intrinsic constraint is the scarcity of real-world anomaly data, which introduces bias and restricts the ability to accurately predict model performance in production environments. Although synthetic defects are an effective tool for expanding defect class coverage and reducing annotation costs, they do not always reproduce the domain-specific complexity observed online. In this study, artificial anomalies were generated through inpainting and texture resynthesis from defect-free frames, systematically distributed along the sealant trajectories in the underbody. This approach was necessary given the limited availability of real defect samples but has inherent limitations: synthetic examples may fail to capture fine scratches, low-contrast discontinuities, or complex geometries near welds and irregular surfaces, where three-dimensional material variations are significant. As discussed in Section 4.3, these limitations stem from the lack of physical process modeling and photometric variability. This issue was partially mitigated through the generation of synthetic anomalies, validated with domain experts. Future work could address these limitations by combining active learning strategies (i.e., incrementally incorporating newly observed anomalies during deployment) with more realistic simulation and acquisition setups, such as physics-informed rendering, variable illumination conditions, and higher-resolution imaging, to produce more diverse and representative datasets of real defects.
- *Size-dependent threshold tuning and expert-driven calibration*. The size-aware analysis of missed defects highlighted that different anomaly scales (small, medium, large) exhibit distinct error patterns and benefit differently from masking. Turning these qualitative observations into explicit, size-dependent thresholding policies would be valuable for practitioners. However, defining such policies requires a joint calibration phase with domain experts at the target production line, since acceptable FP/FN trade-offs, production rates, and safety margins are plant-specific rather than universal. For this reason, fixed thresholds were intentionally not prescribed in this work, and size-aware tuning, carried out in collaboration with process engineers during pre-deployment integration, can be identified as an explicit avenue for future work.
- *Lack of datasets for supervised learning*. Another data-related limitation is the lack of a representative dataset containing diverse real-world anomalies, which hindered the exploration of supervised learning approaches. These approaches typically offer higher accuracy when well-labeled data is available. Future work should focus on creating a structured, statistically representative dataset to enable comprehensive experiments with supervised models, particularly for sealant defect localization tasks.
- *Constraints in underbody extraction*. From a hardware viewpoint, the underbody extraction process is another practical constraint. A tailored solution would enhance both accuracy and processing speed. Alternatively, utilizing external equipment (e.g., photoelectric sensors) to trigger frame acquisition would also be advantageous, especially in facilities where such equipment is already installed. This is particularly relevant since motion on the conveyor can vary across different passes.
- *Incomplete visual coverage*. While the current camera viewpoint was selected to maximize visibility of sealant patterns, a few lateral and frontal areas of the underbody remained unobservable. Future developments will therefore aim to overcome this limitation by implementing a multi-view version of the system, which would allow monitoring of the entire area where the sealant is applied by industrial robots. To this end, in addition to determining the

minimum number and optimal positioning of the additional viewpoints, the best method for combining the information from the various cameras will also need to be studied. Possible approaches include modifying models to process multiple synchronized underbodies simultaneously, or performing inference separately for each view and combining the outputs intelligently. Suitable fusion strategies will need to be developed, either by adapting models to process multiple synchronized views or by combining separate inferences through late fusion techniques.

- *Lack of defect classification*. At the modeling level, extending anomaly detection models to classify the type of anomaly by leveraging object detection methods and incorporating domain-specific knowledge could improve system utility by enabling more advanced diagnostics. These capabilities would be particularly useful for training plant operators to recognize and address specific defect types during production.
- *System feedback and real-time warnings*. In real-time manufacturing contexts, it would also be interesting to integrate anomaly maps into a system that alerts plant operators of potential issues in sealant dispensing robots. By correlating anomalies with specific robots and establishing error frequency thresholds, automatic warnings could be triggered. This would enable rapid intervention and reduce production downtime.
- *Lighting variability and photometric robustness*. Although training and test data were collected under a consistent imaging protocol, future deployments in settings with different lighting will benefit from domain adaptation and illumination-invariant feature learning to mitigate photometric shifts.
- *Scale variability and illumination-invariant representations*. Beyond ROI masking, two complementary directions can further increase robustness in industrial deployments: explicit multi-scale feature aggregation and illumination-invariant representations. First, small, or low-contrast sealant discontinuities are particularly sensitive to the effective receptive field and to downsampling in patch-based pipelines. Future work will therefore explore multi-scale aggregation strategies (e.g., FPN-style fusion or multi-level pooling) to better preserve fine details across viewpoints, including systematic ablations on the number of backbone levels and fusion rules. Second, illumination changes and specularities can induce photometric domain shift, increasing false positives on reflective structures and suppressing subtle defects. To mitigate this, we will consider (i) photometric augmentation during training (brightness/contrast/gamma and color temperature shifts), (ii) lightweight color-constancy or contrast-normalization preprocessing suitable for real-time pipelines, and (iii) representation learning choices that encourage invariance (e.g., self-supervised pretraining with photometric perturbations).
- *Hyperparameter sensitivity and deployment tuning*. Finally, further research should explore the influence of different hyperparameters on performance metrics. Understanding the trade-offs between inference speed and detection accuracy will aid in assessing the practical viability of each model for real-time industrial applications. Automated tuning methods, such as Bayesian optimization or adaptive search, could help identify robust configurations across varying deployment scenarios.
- *Broader applicability to other processes*. While this study focuses on sealant application, the framework's core components (synthetic anomaly generation, ROI masking, and inference-time filtering) can be adapted to a variety of manufacturing processes where quality control relies on identifying rare visual anomalies under real-time constraints. Representative examples include weld seam inspection (detecting porosities or discontinuities), paint quality control (sagging, bubbles, or thickness variations), component assembly (misalignments or missing parts), and plastic molding (flashes or incomplete injections). The same principles also extend to more complex processes such as metal casting, precision machining (e.g., milling or turning, where micro-scratches or dimensional deviations must be detected), and electronics manufacturing, where small soldering defects or missing microcomponents can affect reliability. In all these cases, ROI-based analysis and domain-specific synthetic data generation help to focus detection on functionally relevant areas and improve generalization across process types. However, several domain-specific challenges remain. Variable lighting, reflections, and surface gloss in industrial environments may require periodic model recalibration and adaptation of the acquisition setup. Accuracy can be further enhanced by integrating RGB-D or multispectral imaging to better estimate geometry and differentiate material properties, especially on metallic or textured surfaces. Finally, to ensure that the model's behavior aligns with process requirements, the inspection scope should be clearly defined in advance, specifying target defect types, acceptable size thresholds, and required levels of granularity.

Overall, addressing these limitations through targeted improvements in data acquisition, model design, sensing strategies, and integration mechanisms will be essential to ensure the robustness and scalability of ADL systems in real industrial environments.

9 | Conclusion

This work investigated the effectiveness of state-of-the-art deep learning models for ADL in automotive quality inspection, with a specific focus on sealant application. It evaluated four prominent ADL architectures, namely PatchCore, MemSeg, EfficientAD, and SimpleNet, using a custom dataset that included both synthetic and real-world defects captured through a strategically positioned camera framing car underbodies.

The experimental results demonstrate that MemSeg and PatchCore outperform other models in pixel-level anomaly segmentation, achieving high IoU scores. This outcome highlights the capability of these models to accurately localize fine-grained defects such as sealant misalignment and holes. Conversely, EfficientAD-M and SimpleNet showed superior performance in image-level anomaly detection metrics, including AUROC and AU-PR, while also exhibiting significantly lower inference times. These characteristics make EfficientAD-M and SimpleNet

particularly suitable for real-time applications where rapid decision-making is essential.

The incorporation of synthetic defect generation played a crucial role in mitigating the challenge of data scarcity, enhancing the models' ability to generalize to unseen anomalies. However, reliance on synthetic data introduces certain limitations, as real-world defects may exhibit greater complexity and variability. Despite these constraints, the findings underscore the potential of advanced ADL models to substantially improve quality control processes, thereby reducing the incidence of unresolved defects and enhancing overall production efficiency.

This study was carried out within the framework of the MANAGE 5.0 project, which aims to apply ADL techniques to the production line of a car manufacturing plant to improve quality control and assist operators in correcting potential defects. Future developments will include the integration of the output of an ADL system into an XR platform that will display detected defects in real-time directly on the car underbodies using holographic technologies. This integration will guide operators in their corrective actions, helping to reduce operator workload and the likelihood of missed defects. Ultimately, the system will be deployed on a real automotive production line to evaluate its performance in a practical, real-world scenario.

Taken together, the findings presented in this paper support the practical feasibility of deploying ADL systems for automotive quality inspection, particularly when combined with XR-based operator support. Among the evaluated models, MemSeg emerges as the most suitable candidate for the MANAGE 5.0 use case, providing a strong balance between accurate defect localization and sufficiently fast inference to enable real-time interaction on the production line. From a methodological standpoint, the study also leverages ROI-based masking, size-aware error analysis, and statistically grounded model comparison to systematically characterize model performance under industrial constraints, providing a structured basis for future deployment-oriented adaptations.

To realize this integration in a real-world setting, further work is required to address both technical and non-technical aspects. Key technical priorities include improving spatial alignment between digital content and physical components (e.g., through SLAM or object tracking), refining adaptive visualizations based on defect type, and ensuring robust performance under typical industrial variability such as noise, lighting, and surface reflectance. Additionally, deployment across production lines requires assessing layout constraints (e.g., inspection space, camera placement, lighting), managing workpiece variability, and applying use-case-specific masking strategies to reduce false detections.

At the same time, a critical consideration is deployability. While some open-source ADL models can be embedded directly in commercial XR platforms, others may require additional licenses or face compatibility constraints. Understanding which architectures are legally and technically viable for industrial integration is essential. In parallel, potential collaborations with robotics and IoT teams could further enhance system integration, enabling real-time data exchange, adaptive inspection strategies, and tighter coupling with existing automation infrastructure.

Finally, from a practical standpoint, it is important to assess the economic trade-offs between deployment and maintenance costs, such as hardware, software, and operator training, on the one side, and the potential benefits, including reduced rework, fewer downstream failures, and improved inspection throughput, on the other side. A cost–benefit analysis tailored to the specific manufacturing context will be fundamental to driving adoption and ensuring long-term sustainability.

Author Contributions

Francesco Manigrasso: investigation, methodology, software, writing – original draft. **Davide Calandra:** conceptualization, software, writing – review and editing. **Lia Morra:** methodology, validation, writing – review and editing. **Fabrizio Lamberti:** funding acquisition, project administration, supervision, writing – review and editing.

Acknowledgments

The authors would like to thank Paolo Drago Leon and Andrea Taurino for their contributions to the project. Open access publishing facilitated by Politecnico di Torino, as part of the Wiley - CRUI-CARE agreement.

Funding

This work was funded by the Ministry of Enterprises and Made in Italy (MISE) through the MANAGE 5.0 "MANufacturing Automotive Green Evolution 5.0" project (project code F/310302/01-05/X56).

Conflicts of Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Francesco Manigrasso, Davide Calandra, Lia Morra, and Fabrizio Lamberti report financial support provided by the Ministry of Enterprises and Made in Italy (MISE), which funded the MANAGE 5.0 project (F/310302/01-05/X56).

Data Availability Statement

Research data are not shared due to a non-disclosure agreement (NDA).

References

1. I. Ahmed, M. Ahmad, A. Chehri, and G. Jeon, "A Smart-Anomaly-Detection System for Industrial Machines Based on Feature Autoencoder and Deep Learning," *Micromachines* 14, no. 1 (2023): 1–12.

2. L. Zhou, L. Zhang, and N. Konz, "Computer Vision Techniques in Manufacturing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53, no. 1 (2023): 105–117.

3. L. Armesto, J. Tornero, A. Herraez, and J. Asensio, "Inspection System Based on Artificial Vision for Paint Defects Detection on Cars Bodies," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2011), 1–4.

4. X. Zheng, S. Zheng, Y. Kong, and J. Chen, "Recent Advances in Surface Defect Inspection of Industrial Products Using Deep Learning Techniques," *International Journal of Advanced Manufacturing Technology* 113 (2021): 35–58.

5. A. Roy, "SMART Monitoring and Automated Real-Time Visual Inspection of Sealant Application (SMART-VIStA)," *Manufacturing Letters* 35 (2023): 1134–1145.

6. S. Howard, R. Jang, V. O'Keeffe, et al., "Visual Inspection With Augmented Reality Head-Mounted Display: An Australian Usability Case

Study," *Human Factors and Ergonomics in Manufacturing & Service Industries* 33, no. 3 (2023): 272–296.

7. I. Kuric, J. Klarák, V. Bulej, et al., "Approach to Automated Visual Inspection of Objects Based on Artificial Intelligence," *Applied Sciences* 12, no. 2 (2022): 1–19.

8. Q. Zhou, R. Chen, B. Huang, W. Xu, and Y. Jie, "DeepInspection: Deep Learning Based Hierarchical Network for Specular Surface Inspection," *Measurement* 160 (2020): 1–14.

9. E. N. Malamas, E. G. Petrakis, M. Zervakis, L. Petit, and J. D. Legat, "A Survey on Industrial Vision Systems, Applications and Tools," *Image and Vision Computing* 21, no. 2 (2003): 171–188.

10. F. K. Konstantinidis, S. G. Mouroutsos, and A. Gasteratos, "The Role of Machine Vision in Industry 4.0: An Automotive Manufacturing Perspective," in *Proceeding of IEEE International Conference on Imaging Systems and Techniques (IST)* (IEEE, 2021), 1–6.

11. Q. Zhou, R. Chen, B. Huang, C. Liu, J. Yu, and X. Yu, "An Automatic Surface Defect Inspection System for Automobiles Using Machine Vision Methods," *Sensors* 19, no. 3 (2019): 1–18.

12. P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders," in *Proceedings of 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)* (INSTICC, 2019), 372–380.

13. P. Kamani, E. Noursadeghi, A. Afshar, and F. Towhidkhah, "Automatic Paint Defect Detection and Classification of Car Body," in *Proceedings of 7th Iranian Conference on Machine Vision and Image Processing (MVIP)* (IEEE, 2011), 1–6.

14. D. Kosmopoulos and T. Varvarigou, "Automated Inspection of Gaps on the Automobile Production Line Through Stereo Vision and Specular Reflection," *Computers in Industry* 46, no. 1 (2001): 49–63.

15. J. Molina, J. E. Solanes, L. Arnal, and J. Tornero, "On the Detection of Defects on Specular Car Body Surfaces," *Robotics and Computer-Integrated Manufacturing* 48 (2017): 263–278.

16. R. B. Tudeschini and Á. M. de Souza Soares, "Automatic Inspection System of Adhesive on Vehicle Windshield Using Computational Vision," *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 45, no. 2 (2023): 1–15.

17. F. Chang, M. Liu, M. Dong, and Y. Duan, "A Mobile Vision Inspection System for Tiny Defect Detection on Smooth Car-Body Surfaces Based on Deep Ensemble Learning," *Measurement Science and Technology* 30, no. 12 (2019): 1–9.

18. K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards Total Recall in Industrial Anomaly Detection," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE/CVF, 2022), 14318–14328.

19. M. Yang, P. Wu, J. Liu, and H. Feng, "MemSeg: A Semi-Supervised Method for Image Surface Defect Detection Using Differences and Commonalities," *Engineering Applications of Artificial Intelligence* 119 (2023): 105835, https://doi.org/10.48550/arXiv.2205.00908.

20. TooTouch, "MemSeg: Unofficial Re-Implementation of MemSeg for Anomaly Detection," (2022), https://github.com/TooTouch/MemSeg.

21. K. Batzner, L. Heckler, and R. König, "EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (IEEE/CVF, 2024), https://doi.org/10.48550/arXiv.2303.14535.

22. rximg, "EfficientAD: Unofficial Version of EfficientAD," (2023), https://github.com/rximg/EfficientAD.

23. Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "SimpleNet: A Simple Network for Image Anomaly Detection and Localization," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE/CVF, 2023), 20402–20411.

24. Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "SimpleNet: A Simple Network for Image Anomaly Detection and Localization," (2023), https://github.com/DonaldRR/SimpleNet.

25. M. Mazzetto, M. Teixeira, É. O. Rodrigues, and D. Casanova, "Deep Learning Models for Visual Inspection on Automotive Assembling Line," *arXiv preprint arXiv:2007.01857* (2020), https://doi.org/10.48550/arXiv.2007.01857.

26. C. Ge, J. Wang, J. Wang, Q. Qi, H. Sun, and J. Liao, "Towards Automatic Visual Inspection: A Weakly Supervised Learning Method for Industrial Applicable Object Detection," *Computers in Industry* 121 (2020): 1–11.

27. P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE/CFV, 2020), 4182–4191.

28. S. Mei, H. Yang, and Z. Yin, "An Unsupervised-Learning-Based Approach for Automated Defect Inspection on Textured Surfaces," *IEEE Transactions on Instrumentation and Measurement* 67, no. 6 (2018): 1266–1277.

29. X. Zhu, A. Maki, and L. Hanson, "Unsupervised Domain Adaptive Object Detection for Assembly Quality Inspection," in *Proceedings CIRP Conference on Intelligent Computation in Manufacturing Engineering (ICME)*, vol. 112 (2022), 477–482.

30. F. Caetano, P. Carvalho, and J. Cardoso, "Deep Anomaly Detection for In-Vehicle Monitoring—An Application-Oriented Review," *Applied Sciences* 12, no. 19 (2022): 1–22.

31. Y. Cao, X. Xu, J. Zhang, et al., "A Survey on Visual Anomaly Detection: Challenge, Approach, and Prospect," *arXiv preprint arXiv:2401.16402* (2024), https://doi.org/10.48550/arXiv.2401.16402.

32. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 6 (2017): 1137–1149.

33. J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 779–788.

34. S. B. Block, R. D. da Silva, L. B. Dorini, and R. Minetto, "Inspection of Imprint Defects in Stamped Metal Surfaces Using Deep Learning and Tracking," *IEEE Transactions on Industrial Electronics* 68 (2021): 4498–4507.

35. Y. He, B. Wu, J. Mao, W. Jiang, J. Fu, and S. Hu, "An Effective MID-Based Visual Defect Detection Method for Specular Car Body Surface," *Journal of Manufacturing Systems* 72 (2024): 1–9.

36. X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-To-End Object Detection," *arXiv preprint arXiv:2010.04159* (2020), https://doi.org/10.48550/arXiv.2010.04159.

37. F. Chang, M. Dong, M. Liu, L. Wang, and Y. Duan, "A Lightweight Appearance Quality Assessment System Based on Parallel Deep Learning for Painted Car Body," *IEEE Transactions on Instrumentation and Measurement* 69 (2020): 5298–5307.

38. Z. Zeng, B. Liu, J. Fu, and H. Chao, "Reference-Based Defect Detection Network," *IEEE Transactions on Image Processing* 30 (2021): 6637–6647.

39. S. Venkataramanan, K. Peng, R. V. Singh, and A. Mahalanobis, "Attention Guided Anomaly Detection and Localization in Images," in *Proceedings of 16th European Conference on Computer Vision (ECCV)* (Springer, 2020), 485–503.

40. G. Bhattacharya, B. Mandal, and N. B. Puhan, "Interleaved Deep Artifacts-Aware Attention Mechanism for Concrete Structural Defect

Classification," *IEEE Transactions on Image Processing* 30 (2021): 6957–6969.

41. J. Zipfel, F. Verworner, M. Fischer, U. Wieland, M. Kraus, and P. Zschech, "Anomaly Detection for Industrial Quality Assurance: A Comparative Evaluation of Unsupervised Deep Learning Models," *Computers & Industrial Engineering* 177 (2023): 1–17.

42. G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools* 25 (2000): 120–125.

43. OpenCV, "Template Matching," (2024), https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html.

44. The GIMP Development Team, "GIMP," (2024), https://www.gimp.org.

45. A. Seeliger, L. Cheng, and T. Netland, "Augmented Reality for Industrial Quality Inspection: An Experiment Assessing Task Performance and Human Factors," *Computers in Industry* 151 (2023): 103985.

46. D. Calandra, A. Cannavò, and F. Lamberti, "Improving AR-Powered Remote Assistance: A New Approach Aimed to Foster Operator's Autonomy and Optimize the Use of Skilled Resources," *International Journal of Advanced Manufacturing Technology* 114, no. 9 (2021): 3147–3164.

47. A. Muñoz, X. Mahiques, J. E. Solanes, A. Martí, L. Gracia, and J. Tornero, "Mixed Reality-Based User Interface for Quality Control Inspection of Car Body Surfaces," *Journal of Manufacturing Systems* 53 (2019): 75–92.

48. D. Calandra, A. Cannavò, and F. Lamberti, "Evaluating an Augmented Reality-Based Partially Assisted Approach to Remote Assistance in Heterogeneous Robotic Applications," in *Proceedings of IEEE 7th International Conference on Virtual Reality (ICVR)* (2021), 380–387.