JTE HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

An Evaluation of Diffusion-Based Anomaly Detection in Metal Can Products

Xuan-Vy Huynh[1], Quoc-Danh Pham[1], Viet-Nhat Pham[2], Duy-Vuong Tran[1], Manh-Hung Nguyen[1*]

[1]*Ho Chi Minh City University of Technology and Education, Vietnam*
[2]*Bosch Global Software Technologies Company Limited, Vietnam*

[*]Corresponding author. *Email: hungnm@hcmute.edu.vn*

ARTICLE INFO

Received: 12/02/2025
Revised: 06/06/2025
Accepted: 17/06/2025
Published: 28/11/2025

KEYWORDS

Anomaly detection;
Unsupervised learning;
Diffusion model;
Quality assurance;
Computer vision.

ABSTRACT

Deep learning has been considered a successful solution to process images and complex data. Hence, it is expected to be a standard solution for quality assurance in manufacturing. A standard deep learning-based method tries to reconstruct a normal image, and the difference between a testing image and its corresponding reconstructed image serves as an anomaly map. While a reconstruction model had been proposed, diffusion methods had been considered as SoTA solutions for image generation. However, these methods are tested on well-prepared datasets collected by costly devices in less noisy conditions. In an industrial environment, the image may be collected using economic hardware with serious noise. Motivated by the observation, this work tries to evaluate how diffusion-based anomaly detections work in a practice environment. We first collected a dataset by ourselves and tested it on various well-known anomaly detection methods. The hardware includes a rotating disk and a camera to capture sample data from various angles, while the software serves as an anomaly detection system for test samples. Also, we focus on well-known diffusion models to address whether this method works in high-variance environments. Comprehensive experiments on three can-datasets had been implemented, and the result shows that at the image level, the diffusion method works robustly without error.

Doi: https://doi.org/10.54644/jte.2025.1831

1. Introduction

Stable and consistent operations are essential in manufacturing, making an efficient quality control (QC) system indispensable. Regularly, QC is crucial for timely defect detection and maintaining production efficiency. Additionally, it plays a critical role in determining product acceptance or rejection, thereby ensuring high output quality and prolonging the lifespan of production lines.

Conventional QC methods require a lot of human resources, while deep learning models could speed up the process efficiently. Hence, deep learning has been considered as a possible solution in QC. Early deep-learning models used supervised learning to predict error categories. While achieving very high accuracy, a solid limitation of supervised learning lies in its dependence on predefined defect types, which humans typically specify. In a supervised learning pipeline, employees identify potential defect types (e.g., deformation defect), and engineers collect corresponding data to train a classifier. However, during real-world operation, if a previously unseen defect (e.g., paint defect) occurs—one that the employee did not anticipate—the pre-trained supervised model will likely fail to detect it, because it lacks training data for that category. This dependency fundamentally limits the adaptability of supervised approaches. Since identifying all possible defects during dataset preparation is often impractical, supervised models may struggle in dynamic production environments where new or rare defects can arise.

In manufacturing environments, the defect categories are only available after a long time in the operation phase. Hence, we may not have the information to implement the data acquisition and labeling in the early operation phase. Motivated by the observation, anomaly detection methods have been introduced to address the challenge. These methods learn from normal data, and if the testing sample

differs from the training samples, it could be considered an anomaly sample. This approach does not require the error category from early states and can fulfill the dynamic requirement in an industrial environment.

Regularly, there are two well-known approaches for anomaly detection. The prototype-based approach focuses on learning the characteristics of normal data and using these as a reference for comparison. A sample is considered an anomaly when it significantly deviates from the learned distribution of normal data. PaDiM [1], PatchCore [2], and CS-Flow [3] are techniques that belong to the prototype-based approach. PaDiM [1] models a multivariate Gaussian distribution from local features extracted from deep learning layers, enabling precise anomaly detection for complex data. PatchCore [2] applies Core-Set techniques to minimize stored data size while effectively detecting small anomalous regions, while maintaining performance. CS-Flow [3] utilizes Normalized Flow [4] to integrate multi-level feature information, delivering high-performance anomaly detection by leveraging multi-resolution insights.

The second approach is reconstruction-based solutions. They apply generative models to learn and reconstruct normal data from a given input. The difference between the original and reconstructed images helps indicate the degree of abnormality of the sample. Generative models such as Variational Autoencoders (VAEs) [5], Generative Adversarial Networks (GANs) [6], and Diffusion Models (DMs) [6] are possible solutions for this task. VAEs learn the latent space from normal data and reconstruct the data, with the difference between the original and reconstructed data helping to detect anomalies. GANs use competition between the generator and discriminator to detect anomalies, while DMs, through injection and denoising, learn to recover data from noise, which helps detect anomalies based on poor denoising for unfamiliar samples. DMs provide higher stability than VAEs and GANs, although they require longer training times.

While these methods report promising results on academic datasets [7], the performance in practical industrial environments remains an open question. Usually, data acquisition systems frequently rely on low-cost commercial hardware and are subject to noise, such as uneven lighting or lens distortions. For example, as shown in Figure 1, metal cans in industrial production frequently reflect light from illumination systems, and images may become misaligned due to camera shifts during operation. The differences between the two samples are evident in the background, lighting, and pose. Figure 1.b was captured under optimal conditions, featuring a uniform background, evenly distributed lighting that reduces reflections, and a frontal shooting angle to emphasize the object clearly. In contrast, Figure 1.a illustrates the challenges faced in real-world conditions, such as a noisy, non-uniform background, strong reflections from metallic surfaces, and a tilted shooting angle, which create imbalances and complicate recognition or analysis. This observation raises a research question: “Could state-of-the-art anomaly detection models designed and validated under ideal conditions be robust enough in real-world industrial environments where environmental factors may affect image quality?”.

(a) (b)

Figure 1. *Images in the datasets: (a) self-collected images; (b) images in the MVtec dataset*

To address the question, this paper tries to collect data from economic hardware and evaluate the abnormal detection results using SoTA methods. We mainly focus on DDAD [7] because the method has been considered a successful solution in many datasets [8]. Also, with standard datasets (MVtec) used in academia benchmarking, diffusion-based models have led the overall performance rankings [9] it is a well-known solution based on the diffusion method for anomaly detection and has been considered a successful solution in many datasets. The major works are listed below:

- We build up simple hardware for data acquisition. It includes a rotating disk and a RealSense camera.
- We construct and standardize three datasets for three categories of can. Each category has only one video of the normal can and many videos of deflected cans. The normal video is used to extract standard samples, whereas videos of deflected cans are used for validation and testing.

We applied SoTA anomaly detection methods on the dataset and evaluated the performance by three metrics, including Image-based Area Under the Receiver Operating Characteristic Curve (AUROC), Pixel-based AUROC, and PRO (Per-Region Overlap). Among them, Image-based AUROC has a higher ability to detect faulty products, whereas others focus on localization errors. Hence, a good performance on Image-based AUROC implies that the model is robust in challenging environments.

2. Related works

2.1. Prototype-based approach

Anomaly detection involves extracting key features from data, comparing the features of new samples with those of normal data, and identifying mismatched samples as anomalies. One effective method is PaDiM [1], which models the distribution of image patches using a multivariate Gaussian. PaDiM enhances anomaly detection and localization performance while maintaining low time and space complexity, making it suitable for practical applications. However, it may encounter challenges when anomalies have complex structures or the training data is not sufficiently representative.

Another advanced technique is PatchCore [2], which builds on previous methods by maximizing the representativeness of regular patch features. It employs coreset sampling techniques to reduce computational complexity. PatchCore [2] has demonstrated strong performance on benchmark datasets like MVtec [10] and Magnetic Tile Defects (MTD) [11]. Its ability to operate effectively with limited training data makes it suitable for real-world applications. However, its performance may decline in large-scale deployments where highly diverse distributions and computational demands increase.

Additionally, density estimation methods have been applied to anomaly detection. The CS-Flow [3] approach simultaneously processes multi-scale feature maps to retain global and local contexts, enabling accurate anomaly detection and localization. By leveraging a fully convolutional architecture with cross-scale connections, CS-Flow [3] achieves flexible and precise modeling of feature distributions. This method has shown superior performance on datasets such as MVtec [10] and MTD [11], effectively handling diverse distributions while preserving positional information in the latent space. However, its higher computational complexity, particularly when processing multi-scale feature maps, may hinder efficiency in real-time systems.

2.2. Reconstruction-based methods

Anomaly detection using image reconstruction relies on training a model on normal data to identify deviations in new images. The model learns the distribution of normal images in the training process, then the model can reconstruct a normal image in a testing phase. However, this method struggles to reconstruct anomalous regions that deviate from the learned distribution. The error is calculated as the pixel-wise difference between the original and reconstructed images. Regions with high reconstruction errors indicate the presence of anomalies, enabling precise detection and localization. This approach is practical and highly interpretable. Also, it is isolating and highlighting anomalies based on deviations from standard patterns.

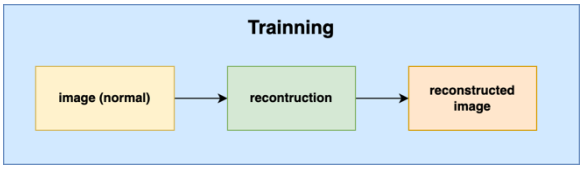

Figure 2. *The training process for reconstruction-based methods*

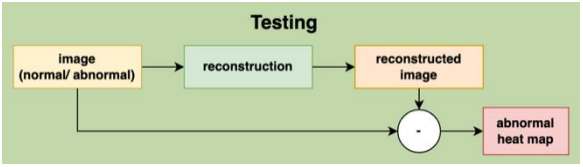

Figure 3. *The testing process for reconstruction-based methods*

JTE HCMUTE
ISSN: 2615-9740
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Figure 2 and Figure 3 illustrate the training and testing processes. Figure 2 demonstrates the training phase, where normal images are used to train a reconstruction module. In Figure 3, an input image (whether normal or anomalous) is fed into the reconstruction module. The original and reconstructed images are then compared to identify discrepancies. These differences are visualized in a heat map, which indicates anomalous regions within the image. This method not only effectively detects anomalies but also precisely pinpoints their locations.

Among the leading approaches, Score-based PR [12] employs a diffusion process to normal images by calculating the gradient of the log-likelihood (score), thereby projecting noisy samples back onto the data manifold to reconstruct normal images. However, this method involves a complex tuning process and performs poorly when handling intricate structures or shape anomalies. DRAEM [13] learns a joint representation of normal and anomalous samples while defining decision boundaries in the feature space. This method combines original and reconstructed images in the feature space rather than solely relying on the divergence in the pixel domain. The approach helps avoid overfitting caused by synthetic anomalies. While outperforming many advanced anomaly detection and localization methods, the method complex post-processing steps. Last but not least, DDAD [7] introduce a conditional denoising diffusion model for anomaly detection. This method controls the diffusion process using the target image to ensure precise reconstruction of standard features. Additionally, it incorporates domain adaptation by fine-tuning the feature extractor based on examples generated during the diffusion process, improving generalization and adaptability.

3. Methodology

3.1. Hardware to collect the dataset

Figure 4 depicts the hardware setup for data collection. This hardware includes a rotating disk powered by an electric motor and a gearbox, which maintain a stable rotation speed of 2 RPM. The system is housed in a sealed enclosure to eliminate environmental interference and features uniform lighting alongside a RealSense camera for capturing detailed surface images of the products. The camera is securely mounted and synchronized with a central computer for efficient data management and storage, ensuring precision and consistency throughout the data collection process.

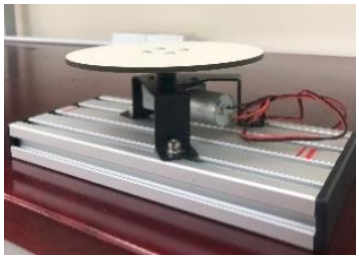

Figure 4. *Rotating disk in the data collection hardware system.*

3.2. Diffusion model for anomaly detection.

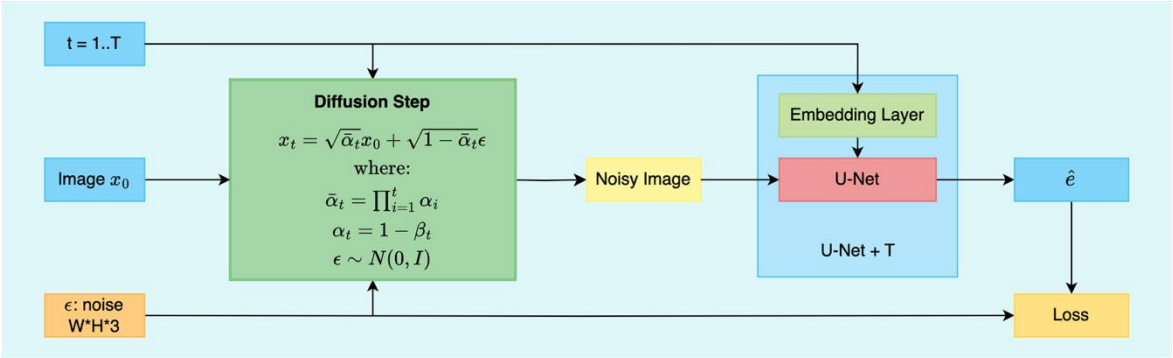

Figure 5. *Architecture of the Diffusion Model*

This paper uses the diffusion model [14] for image reconstruction. As described in Figure 5, the diffusion model's architecture consists of two main steps. In the first step, Gaussian noise is added to the image over T time steps using a pre-defined formula. This step generates a noisy image which fed

to the Unet-liked denoised model. In the second step, the U-Net model predicts the added noise given the noised image and the step t. The model is trained by optimizing the Mean Squared Error (MSE) loss function to estimate the noise as closely as possible to the added noise. Detail of two steps are as below:

Forward Diffusion Process

During the forward process, Gaussian noise is gradually added to the image over T time steps. At each step t, the noisy image x_t is computed using Equation (1):

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon \tag{1}$$

Where α_t controls the noise level (typically decreasing over time), and $\epsilon \sim \mathcal{N}(0, I)$ represents Gaussian noise.

After T steps, the image becomes completely noisy, losing all recognizable structure from the original image x_0. Based on the additive property, combining multiple Gaussian distributions still results in a Gaussian distribution. Therefore, the state x_T can be directly computed from the original image x_0 using Equation (2):

$$x_T = \sqrt{\overline{\alpha_T}} x_0 + \sqrt{1-\overline{\alpha_T}}\epsilon \tag{2}$$

In which: $\overline{\alpha_T} = \prod_{t=1}^{T} \alpha_t$: represents the accumulated noise up to step T

The U-Net model training process

The U-Net model training process is carried out in three main steps. First, a clean image x_0 is randomly selected from the training dataset, and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ is added to create the noisy image x_t. The noisy image at step t is calculated according to the forward diffusion formula (2). Next, the noisy image x_t and the time step information t are input into the U-Net model to predict the noise $\widehat{\epsilon_\theta}(x_t, t)$. The error between the actual noise ϵ and the predicted noise is computed using the Mean Squared Error (MSE) loss function as Equation (4):

$$\mathcal{L}(\theta) = |\epsilon - \widehat{\epsilon_\theta}(x_t, t)|^2 \tag{3}$$

In which $\widehat{\epsilon_\theta}(x_t, t)$ is the noise predicted by U-Net with parameter θ, and E denotes the expectation over the training data, random noise, and time steps. Finally, the loss value $\mathcal{L}(\theta)$ is used to update the parameter θ of the U-Net model through the gradient descent algorithm.

3.3. The Reconstruction Process based on conditional image (DDAD methods)

In the image denoising problem, the objective is to transform a noisy image x_0 into a reconstructed image x_t that closely resembles the target image y. The proposed method accomplishes this by conditioning the score function based on a posterior score function $\nabla_{x_t} \log p_\theta(x_t \mid y)$. However, directly computing this posterior score is challenging due to the difference in the signal-to-noise ratio between x_t and y.

To address this issue, the method assumes that when x_0 is close to y, adding a similar noise to y will create a noisy image $y_t \sim x_t$. The amount of noise $\hat{\epsilon}$ is estimated by Equation (4):

$$\hat{\epsilon} = \epsilon_\theta^{(t)}(x_t) - w\sqrt{1-\alpha_t}(y_t - x_t) \tag{4}$$

Where w is the parameter adjusting the degree of conditioning influence, and $\epsilon_\theta^{(t)}(x_t)$ is the amount of noise in x_t predicted by the model. The less noisy image x_{t-1} is calculated through the denoising process as Equation (5):

$$x_{t-1} = \sqrt{\alpha_{t-1}}\hat{f}\theta^{(t)}(x_t) + \sqrt{1-\alpha t - 1 - \sigma_t^2}\hat{\epsilon} + \sigma_t\epsilon_t \tag{5}$$

The process of reconstructing the image from noisy images is detailed in Algorithm 1.

JTE HCMUTE
ISSN: 2615-9740
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Algorithm 1 Reconstruction Process

1. $x_{T'} \leftarrow \sqrt{\alpha_{T'}}x + \sqrt{1-\alpha_{T'}}\epsilon_t$
2. **For all** $t = T', \dots, 1$ **do**
3. $\quad y_t \leftarrow \sqrt{\alpha_t}y + \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t)$
4. $\quad \hat{\epsilon} \leftarrow \epsilon_\theta^{(t)}(x_t) - w\sqrt{1-\alpha_t}(y_t - x_t)$
5. $\quad \widehat{f_\theta^{(t)}}(x_t) \leftarrow \left(x_t - \sqrt{1-\alpha_t}\hat{\epsilon}\right)/\sqrt{\alpha_t}$
6. $\quad x_{t-1} \leftarrow \sqrt{\alpha_{t-1}}\widehat{f_\theta^{(t)}}(x_t) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\hat{\epsilon} + \sigma_t\epsilon_t$
7. **end for**
8. **return** x_0

Initially, the noisy image $x_{T'}$ is created by adding noise to x_0 through Equation (2). Then, the algorithm proceeds to gradually denoise x_t to resemble the target image y. In each iteration:

1. A noisy version y_t is synthesized by adding noise similar x_t to the target image y using the expression $y_t \leftarrow \sqrt{\alpha_t}y + \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t)$.

2. The amount of noise $\hat{\epsilon}$ is calculated from the noise predicted by the model and the deviation between y_t and x_t.

3. The less noisy image is predicted based on $\hat{\epsilon}$ as $\widehat{f_\theta^{(t)}}(x_t) \leftarrow \left(x_t - \sqrt{1-\alpha_t}\hat{\epsilon}\right)/\sqrt{\alpha_t}$. To estimate $\hat{\epsilon}$, we use the w parameter to adjust the influence of conditioning while σ_t controlling the randomness of the sampling process.

4. A new image x_{t-1} is generated by combining three factors: the predicted original image, the estimated noise, and random noise controlled by the parameter σ_t.

3.4. Anomaly map generation

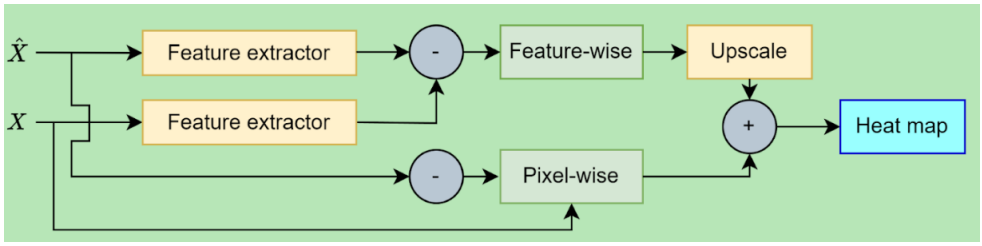

Figure 6. *The estimation process of an anomaly map*

Error! Reference source not found. presents the process to estimate an anomaly map. Denote X and $\hat{X}$ are the testing image and the reconstructed image; the anomaly map is combined by two components as Equation (6). Here, the first component is the pixel-based anomaly map (Mp), represented by equation (7). The second component is the feature-based anomaly map (Mf), represented by equation (8).

$$M = M_p + M_f \tag{6}$$

$$M_p = \|X - \hat{X}\| \tag{7}$$

$$M_f = \|\Phi(X) - \Phi(\hat{X})\| \tag{8}$$

In Equation (8), $\Phi(.)$ is a extractor from a pre-trained backbone. Usually, feature are extracted at diffirent scale; feature hence the M_f should be estimated at multible scale. Denote $\Phi_j(.)$ is the extrector at the j^{th} scale, the anomaly map is rewrote as Equation (9).

$$M_f = \sum_{j=1}^{J} \left\| \Phi_j(X) - \Phi_j(\hat{X}) \right\| \tag{9}$$

In these equations, the feature extractor is a critical factor to estimate M. The extractor is fined-tune by the loss function in Equation (10). Here, the term L_1 imply that X and $\hat{X}$ have similar feature given by a fine-tune feature extractor $\Phi_j(.)$; and the term L_2 imply the fine-tune feature extractor and the pre-trained feature are similar. The term λ is used to control the important of L_2. To model these objective functions, the function $\cos(\Phi_j(.);,\Phi_j(.);)$ has been used. The $\cos(\Phi_j(.);,\Phi_j(.);)$ function measures the similarity between two feature vectors; therefore, the value 1- $\cos(\Phi_j(.);,\Phi_j(.);)$ describes the loss function. The j values represent different scale in feature extractor. Specifically, three resolution levels are selected: 1, 2, and 3, corresponding to the levels of the feature extractor model.

$$L_M = L_1 + \lambda L_2 \tag{10}$$

$$= \sum_{j=1}^{J} \left(1 - \cos\left(\Phi_j(X), \Phi_j(\hat{X})\right)\right) + \lambda \sum_{j=1}^{J} \left(1 - \cos\left(\Phi_j(X), \overline{\Phi_j}(X)\right)\right) + \lambda \sum_{j=1}^{J} \left(1 - \cos\left(\Phi_j(\hat{X}), \overline{\Phi_j}(\hat{X})\right)\right)$$

4. Experiment

4.1. Data collection process and experimental setting.

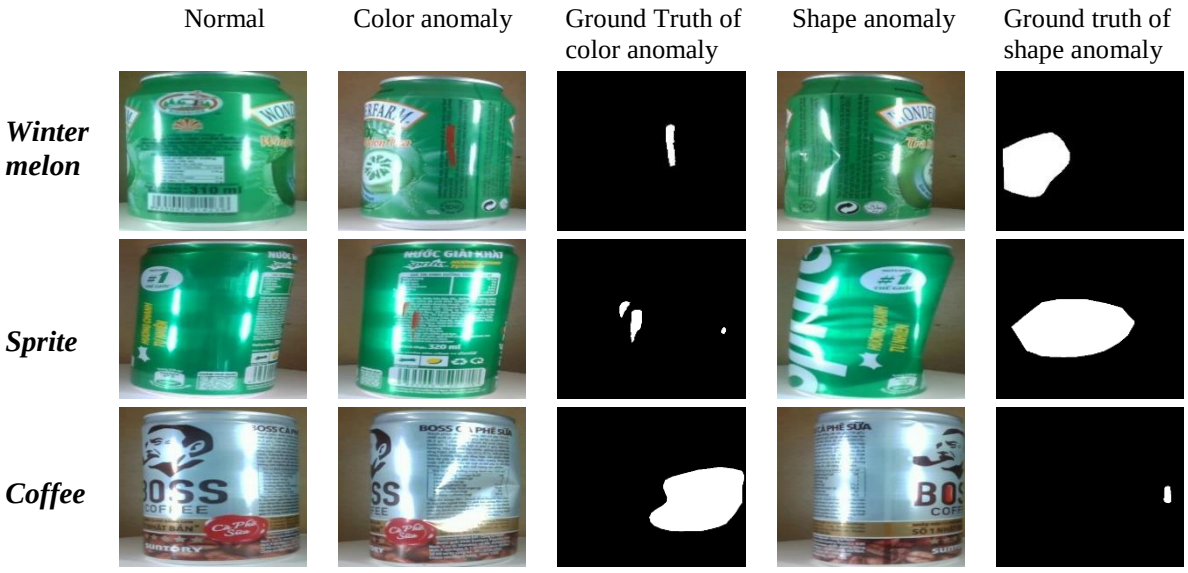

Figure 7. *Example images from the collected dataset and Ground Truth.*

The hardware described in Section 3.1 is employed to collect training samples across three categories of cans. In our experiments, 45 videos were recorded, with 15 videos for each can category. Each category includes five videos captured from various camera angles and under different lighting conditions, reflecting three states: normal, shape anomaly, and color anomaly. The dataset comprises data from 9 cans, evenly distributed across three can categories and two defect categories. After collection, the videos were processed and converted into images, with one image extracted per second. The training dataset consists of 300 normal images, while the testing dataset includes 150 abnormal images for each defect type, along with a selection of normal samples captured under challenging conditions. Examples of can categories and defect categories are shown in Figure 7. When one product has multiple anomalies, it is easier to detect than a product with a single anomaly. Hence, in our paper, we only focus on single anomaly cases. In addition, evaluation metrics and experimental settings are represented in Appendix A1 and A2, respectively.

4.2. Quantitative Results

To evaluate the effectiveness of DDAD on the collected dataset, we compared it with several state-of-the-art methods, including EfficientAD [15], PaDiM [1], PatchCore [2], CS-Flow [3], DRAEM [13], and Score-based PR [12]. EfficientAD, PaDiM, PatchCore, and CS-Flow are prototype-based approaches, while DRAEM, Score-based PR, and DDAD are reconstruction-based methods.

As highlighted in **Error! Reference source not found.**, the DDAD method surpasses all other existing approaches, excelling over reconstruction-based methods as Score-based PR [12] and prototype-based methods as PatchCore [2]. DDAD achieves a perfect Image AUROC score 1.0 across all datasets (Winter Melon, Sprite, Coffee), significantly outperforming competing methods. For example, compared to PatchCore [2], DDAD demonstrates an improvement ranging from 7.6% to 7.9%. Even Score-based PR [12], which achieves Image AUROC scores of 0.968, 1.0, and 0.971 across the three product categories, lags behind DDAD in the first two categories. These results underscore DDAD's exceptional image-level anomaly detection performance, delivering consistent results across diverse datasets.

DDAD maintains its superiority at the pixel level, achieving the highest Pixel AUROC scores of 0.918, 0.936, and 0.968, outperforming both PatchCore [2] and Score-based PR [12]. This demonstrates DDAD's capacity to detect fine-grained or challenging anomalies. DDAD's exceptional performance on the Per-Region Overlap (PRO) metric is particularly notable and critical for assessing anomaly localization. DDAD achieves substantial improvements in PRO, with increases ranging from 75.5% to 122.7% compared to other methods, highlighting its ability to localize anomalies effectively and handle complex reconstruction scenarios.

Table 1. *Detailed comparison of classification and anomaly localization performance of different methods on our benchmark dataset.*

Model	**Winter Melon**			**Sprite**			**Coffee**		
	Image AUROC	Pixel AUROC	PRO	Image AUROC	Pixel AUROC	PRO	Image AUROC	Pixel AUROC	PRO
EfficientAD	0.814	0.64	0.212	0.928	0.757	0.662	0.894	0.727	0.574
PaDiM	0.904	0.738	0.296	0.964	0.758	0.667	0.94	0.775	0.528
PatchCore	0.924	0.846	0.375	0.983	0.857	0.575	0.981	0.94	0.521
CS-Flow	0.606	0.689	0.123	0.851	0.859	0.391	0.931	0.855	0.146
DRAEM	0.835	0.638	0.161	0.983	0.525	0.617	0.905	0.632	0.526
Score-base PR	0.968	0.896	0.435	1.0	0.903	0.502	0. 971	0.952	0.456
DDAD	**1.0**	**0.918**	**0.835**	**1.0**	**0.936**	**0.853**	**1.0**	**0.968**	**0.914**

Additionally, DDAD demonstrates remarkable performance across all data categories. While Score-based PR achieves strong results in specific areas, it falls short compared to DDAD, especially in PRO and Pixel AUROC metrics. DDAD exhibits exceptional performance in the Coffee category, further validating its capability to address real-world challenges requiring high precision in anomaly detection and reconstruction. Overall, DDAD emerges as a robust and effective solution, excelling in research settings and practical applications demanding accurate anomaly detection and reconstruction.

Leveraging a diffusion model, DDAD excels in detecting and localizing anomalies and reconstructing anomalies through a generative framework. The method achieves significant improvements, including a minimum increase of 2.8% in anomaly detection compared to normal samples and a 2.2% improvement over other reconstruction-based methods, as demonstrated by pixel-level evaluations.

JTE HCMUTE
ISSN: 2615-9740
JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

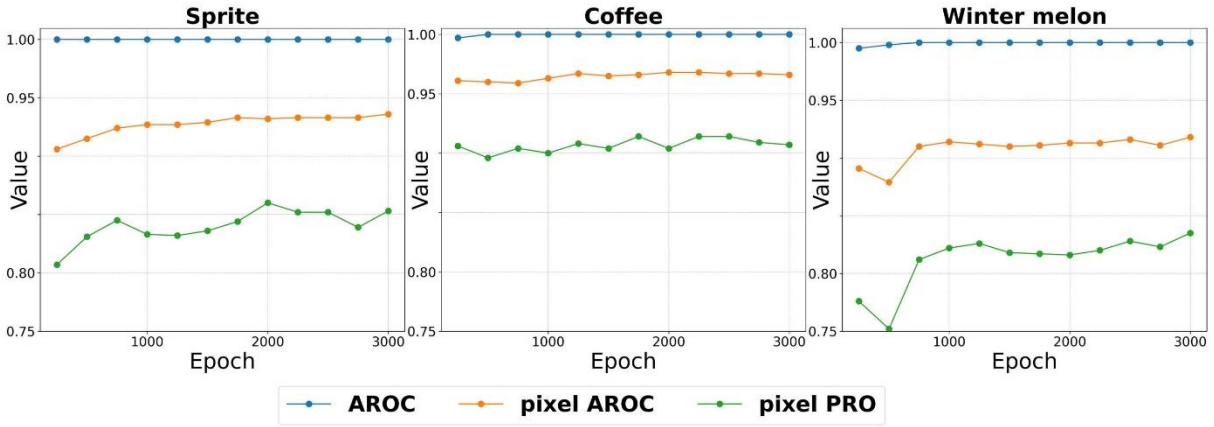

Figure 8. *Performances on the testing set during the training process by DDAD*

Figure 8 illustrates the performance during the training process of the DDAD. The experiment is carried out on the Sprite, Coffee, and Winter Melon datasets across multiple epochs. The Image AUROC metric quickly achieves a perfect score of 100% and remains stable throughout the process. This highlights exceptional image-level classification performance. In addition, Pixel AUROC and Pixel PRO show steady improvements over the epochs, indicating enhanced pixel-level performance. However, signs of overfitting emerge as Pixel PRO begins to plateau. Based on Figure 8, the training process could be shortened by reducing the number of epochs by two-thirds. This property significantly reduces training time while maintaining high performance, particularly for image-level predictions.

In addition to evaluating the accuracy of anomaly detection, we also compare the execution time among these methods. All experiments were conducted using the following hardware configuration: NVIDIA GeForce RTX 2080 Ti 12GB, Intel Core i7-10700 CPU, 16GB RAM, and patch size = 4.

Table 2. *Evaluation of model complexity and performance.*

Method	EfficientAD	PaDiM	PatchCore	CS-Flow	DRAEM	Score_base PR	DDAD
Number of Parameters [x10^6]	8.1	2.8	24.4	63.5	97.4	17.84	32.95
Execution time (ms)	112	96	140	176	220	120	316

The results from Table 2 show that the DDAD model is not memory-intensive. Specifically, the model contains 32,95 million parameters, which is approximately half the size of CS-flow and one-third the size of DRAEM. However, due to the iterative nature of the decoding process, the method requires a longer inference time up to 316ms for a patch containing 4 samples. This limitation can be solved in practical deployments using patch-based processing. In such cases, images from the camera can be temporarily stored in a buffer and grouped into patches. These patches can then be processed in parallel to fully leverage the computational power of the GPU.

4.3. Ablation study

In Section 3.4, the anomaly map is based on two components, including a pixel-based anomaly map (M_p) and a feature-based anomaly map (M_f). This section aims to evaluate the effect of the map on detection results. Figure 9 represents Image-level AUROC, pixel-level AUROC, and PRO with only the pixel-based anomaly map, only the feature-based anomaly map, and both. The results demonstrate that relying only on pixel-wise differences is insufficient for effective anomaly detection. Specifically, the performance metrics based on pixel differences are approximately 50%, which indicates that the detection is nearly equivalent to a random guess. This phenomenon is because pixels are highly sensitive to illumination changes and common environmental issues when data is collected using low-quality hardware.

In contrast, feature maps are more stable and robust than raw pixels, significantly improving detection accuracy. For instance, the pixel-level AUROC improved from 51.7% to 85%, while image-level accuracy increased from 45.7% to 99.7%. Combining both pixel-wise and feature-wise information further enhances performance across all evaluation metrics. However, the improvement is more noticeable when measured using pixel-level AUROC or PRO. This is because pixel-AUROC and PRO are more stringent metrics, requiring precise localization at the pixel level. In contrast, image-level AUC considers only the overall prediction for the entire image and is less sensitive to pixel-wise accuracy.

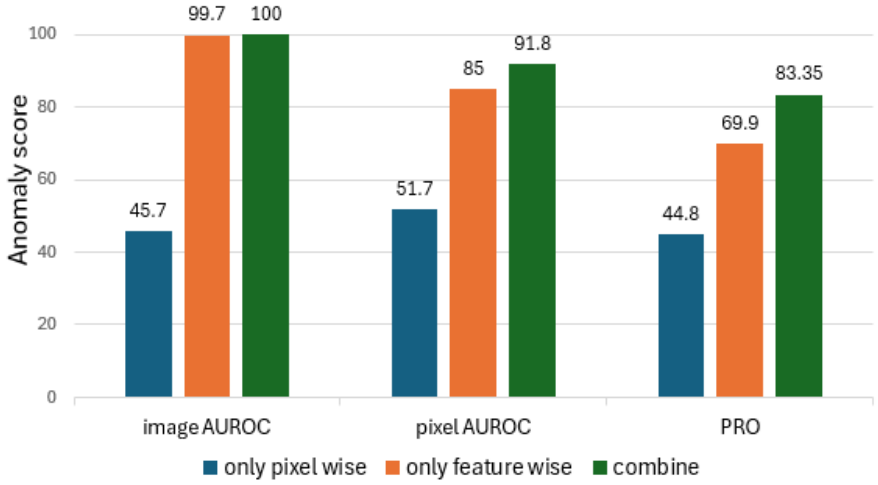

Figure 9. *Ablation study to detect anomaly based on Image AUROC, Pixel AUROC, and PRO*

5. Conclusion

This paper presents a performance evaluation of SoTA anomaly detection solutions on high-noise datasets collected by economic hardware. On the hardware side, the system features a rotating disk and a camera setup to capture sample data from multiple angles. The DDAD anomaly detection method is employed on the software side to identify defects across three product categories. Experimental results demonstrate the system's ability to detect subtle defects that are challenging for human inspection. Especially, on the image level, all anomaly products could be detected without error. On the pixel level, the localization result is higher than other recent methods. This result points out that DDAD is suitable for working in an industrial environment when defect categories are unknown and the data acquisition system is economic.

Acknowledgments

This work is part of project **SV2025-58**, funded in 2025 by the Ho Chi Minh City University of Technology and Education, Vietnam.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are provided in the 'Support Information' file, publicly accessible at the following Google Drive link:
https://drive.google.com/drive/folders/12lFnQ8cK06x8ZO7J8R4XZl0I5VbU1C2_?usp=drive_link

REFERENCES

[1] T. Defard, A. Setkov, A. Loesch, and R. Audigier, “PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization,” Nov. 17, 2020, *arXiv*: arXiv:2011.08785. doi: 10.48550/arXiv.2011.08785.

[2] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, “Towards Total Recall in Industrial Anomaly Detection,” May 05, 2022, *arXiv*: arXiv:2106.08265. doi: 10.48550/arXiv.2106.08265.

[3] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, “Fully Convolutional Cross-Scale-Flows for Image-based Defect Detection,” Oct. 06, 2021, *arXiv*: arXiv:2110.02855. doi: 10.48550/arXiv.2110.02855.

[4] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker, “Normalizing Flows: An Introduction and Review of Current Methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3964–3979, Nov. 2021, doi: 10.1109/TPAMI.2020.2992934.

[5] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 10, 2022, *arXiv*: arXiv:1312.6114. doi: 10.48550/arXiv.1312.6114.
[6] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," Jun. 10, 2014, *arXiv*: arXiv:1406.2661. doi: 10.48550/arXiv.1406.2661.
[7] A. Mousakhan, T. Brox, and J. Tayyub, "Anomaly Detection with Conditioned Denoising Diffusion Models," vol. 15297, 2025, pp. 181–195. doi: 10.1007/978-3-031-85181-0_12.
[8] P. Dhariwal and A. Nichol, "Diffusion Models Beat GANs on Image Synthesis," Jun. 01, 2021, *arXiv*: arXiv:2105.05233. doi: 10.48550/arXiv.2105.05233.
[9] "Papers with Code - MVTec AD Benchmark (Anomaly Detection)." Accessed: Jun. 02, 2025. [Online]. Available: https://paperswithcode.com/sota/anomaly-detection-on-mvtec-ad
[10] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 9584–9592. doi: 10.1109/CVPR.2019.00982.
[11] Y. Huang, C. Qiu, Y. Guo, X. Wang, and K. Yuan, "Surface Defect Saliency of Magnetic Tile," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, Aug. 2018, pp. 612–617. doi: 10.1109/COASE.2018.8560423.
[12] W. Shin, J. Lee, T. Lee, S. Lee, and J. P. Yun, "Anomaly Detection using Score-based Perturbation Resilience," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 23315–23325. doi: 10.1109/ICCV51070.2023.02136.
[13] V. Zavrtanik, M. Kristan, and D. Skočaj, "DRAEM -- A discriminatively trained reconstruction embedding for surface anomaly detection," Sep. 27, 2021, *arXiv*: arXiv:2108.07610. doi: 10.48550/arXiv.2108.07610.
[14] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," Dec. 16, 2020, *arXiv*: arXiv:2006.11239. doi: 10.48550/arXiv.2006.11239.
[15] K. Batzner, L. Heckler, and R. König, "EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies," Feb. 08, 2024, *arXiv*: arXiv:2303.14535. doi: 10.48550/arXiv.2303.14535.

Xuan-Vy Huynh completed his high school education at Thuc Hanh High School in Ho Chi Minh City, Vietnam, from 2020 to 2022, where he built a solid academic foundation and developed essential skills for further studies. In 2022, he began his undergraduate studies in Embedded Systems and the Internet of Things (IoT) at Ho Chi Minh City University of Technology and Education, actively engaging in a dynamic and innovative academic environment. He is currently an undergraduate student, actively exploring and conducting research in the fields of deep learning and image processing.
Email: 22139079@student.hcmute.edu.vn. ORCID: https://orcid.org/0009-0005-0671-7277

Quoc-Danh Pham completed his high school education at Phu My 1 High School in Gialai, Vietnam, from 2019 to 2021, where he built a solid academic foundation and developed essential skills for further studies. In 2021, he began his undergraduate studies in Embedded Systems and the Internet of Things (IoT) at Ho Chi Minh City University of Technology and Education, actively engaging in a dynamic and innovative academic environment. He is currently an undergraduate student, actively exploring and conducting research in the fields of deep learning and image processing.
Email: 21139073@student.hcmute.edu.vn. ORCID: https://orcid.org/0009-0008-1182-5020

Viet-Nhat Pham completed his high school education at Pham Van Dong High School, Quang Ngai, Vietnam, in 2021. Since then, he has been pursuing his undergraduate degree in Embedded Systems and the Internet of Things (IoT) at Ho Chi Minh City University of Technology and Education. His academic interests lie in deep learning and image processing, where he is actively engaging in research and project development.
Email: hnm8hc@bosch.com. ORCID: https://orcid.org/0009-0002-5866-4785

Duy-Vuong Tran completed his high school education at Nguyen Dieu High School in Binh Dinh Province, Vietnam. He later pursued studies in the field of Internet of Things (IoT) but has not yet completed his undergraduate degree.
His research interests are in deep learning and image processing. In addition, he is passionate about digital transformation in enterprises, exploring how emerging technologies such as AI, and cloud computing can help optimize business operations and decision-making processes.
Email: 22139078@student.hcmute.edu.vn. ORCID: https://orcid.org/0009-0004-8003-0871

Manh-Hung Nguyen received a B.S. and M.S. in electrical engineering from the National University of Technology and Education, Ho Chi Minh City, Vietnam, in 2009 and 2011, respectively, and a Ph.D. degree in electrical engineering from the National Kaohsiung University of Applied Sciences, Taiwan, in 2016. He is currently an Assistant Professor in the Faculty of Electrical Electronic Engineering, University of Technology and Education, Vietnam. His research interests are in deep learning and image processing.
Email: hungnm@hcmute.edu.vn. ORCID: https://orcid.org/0000-0003-3869-4610