Franklin Open 15 (2026) 100543

Contents lists available at ScienceDirect

Franklin Open

journal homepage: www.elsevier.com/locate/fraope

Adaptive anomaly detection via dual autoencoders with automated hyperparameter and batch optimization☆

Attapon Pillai [a], Nur Rusyidah Azri [b], Putsadee Pornphol [a], Saratha Sathasivam [b,*], Akarachai Inthanil [a]

[a] Department of Digital Technology, Faculty of Science and Technology, Phuket Rajabhat University, Ratsada, Muang District, 83000, Phuket, Thailand
[b] School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM Penang, Pulau Pinang, Malaysia

ARTICLE INFO

Keywords:
Autoencoder
Anomaly detection
Dual encoder
Explainable artificial intelligence
Hyperparameter optimization

ABSTRACT

Anomaly detection models based on autoencoders often suffer from unstable training, inconsistent reproducibility, and sensitivity to manually selected hyperparameters. These challenges limit their reliability across domains with varying data scales and feature distributions. This paper proposes an adaptive anomaly detection framework that combines a dual autoencoder architecture with a fully automated training-stability pipeline. The dual encoders capture global structural patterns and fine-grained local variations, enabling robust detection of subtle and heterogeneous anomalies. Unlike existing automated approaches which focus primarily on architecture search, our method targets the overlooked problem of training reproducibility and dataset-adaptive optimization. The proposed framework integrates (i) dynamic seed initialization for bias-resistant reproducibility, (ii) adaptive batch size estimation through a statistically derived logarithmic scaling function, (iii) iterative epoch optimization with multi-attempt checkpointing, and (iv) automated dropout tuning via Keras Tuner. To improve decision reliability, we further incorporate Monte Carlo dropout to estimate predictive uncertainty and reduce false-positive classifications. Extensive experiments on medical imaging, industrial inspection, 2D/3D textures, and surveillance datasets demonstrate consistent improvements over 11 state-of-the-art anomaly detection methods, yielding AUROC gains of up to 12.4% and significantly higher recall–precision stability. Ablation studies confirm that each optimization component contributes meaningfully to performance and reproducibility.

1. Introduction

Deep learning is inspired by the structure and function of the human brain. It uses artificial neural networks with multiple layers to learn and extract patterns from data. With minimal feature engineering, deep learning can automatically develop hierarchical representations and handle complex data [1]. This capability has gained extensive adoption for anomaly detection tasks, where it identifies patterns that deviate significantly from expected behavior in healthcare [2], cybersecurity [3], and industrial inspection [4]. Among deep learning approaches, autoencoders are widely recognized for anomaly detection due to their ability to learn efficient representations of normal data [5]. They encode the input data into a lower-dimensional latent space and reconstruct it, minimizing the errors for normal data. However, autoencoders remain highly sensitive to training randomness, hyperparameter choices, and dataset characteristics [6]. Even small variations in random seed, batch size, or training duration can lead to divergent model behavior [7], undermining reproducibility and limiting their reliability in real-world deployments. Recent studies have shown that performance fluctuations can occur solely due to seed selection [8], while static hyperparameters frequently fail to generalize across datasets with different scales, textures, or feature distributions [9].

To mitigate these challenges, several automated deep learning frameworks have emerged, including EvoAAE [10], MODEO-CNN [11], AD-NEv [12], and AD-NEv++ [13]. These systems primarily focus on automating architecture search or neural evolution, enabling models to adapt their structural design to the dataset at hand. However, they do not address the equally critical issue of training stability arising from fixed seeds, fixed batch sizes, manually tuned dropout, or non-adaptive epoch schedules. They do not incorporate dataset-aware seed generation, dynamic batching, uncertainty estimation, or multi-attempt training stabilization, leaving a substantial gap between architecture automation and reproducible anomaly detection. These

☆ This research is supported by Universiti Sains Malaysia and Phuket Rajabhat University under the Visiting Scholar Program.
* Corresponding author.
E-mail address: saratha@usm.my (S. Sathasivam).

https://doi.org/10.1016/j.fraope.2026.100543
Received 21 May 2025; Received in revised form 11 December 2025; Accepted 22 February 2026
Available online 23 February 2026
2773-1863/© 2026 The Authors. Published by Elsevier Inc. on behalf of The Franklin Institute. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

instability issues are further compounded by the representational limitations of single-encoder autoencoders, which often fail to capture anomalies that manifest at different spatial scales [14]. Small defects, subtle texture deviations, or faint pathological structures may be lost when encoded using only global features. Dual encoder architectures provide a natural solution by combining complementary receptive fields [15]: an image-level encoder captures coarse structural context, while a pixel-level encoder focuses on fine-grained local variations. Prior work has demonstrated the promise of multi-scale representations, yet such architectures remain sensitive to training hyperparameters, and their potential has not been fully realized due to a lack of adaptive optimization.

Motivated by these limitations, this study proposes an adaptive, uncertainty-aware anomaly detection framework that integrates a dual autoencoder architecture with a comprehensive training-stability pipeline. The framework incorporates dynamic seed initialization to ensure bias-resistant reproducibility, adaptive batch size estimation using a statistically derived logarithmic model, and an iterative epoch optimization strategy with multi-attempt checkpointing to prevent overfitting or premature convergence. Automated dropout tuning using Keras Tuner enables dataset-specific regularization without manual effort, while Monte Carlo (MC) dropout provides predictive uncertainty estimation to reduce false positives and improve decision reliability. Together, these components form a lightweight yet powerful optimization layer applicable to a wide range of anomaly detection tasks.

The main contributions of this work are as follows, (i) Dual encoder architecture for multi-scale anomaly detection, integrating global and local feature pathways to enhance sensitivity to subtle or heterogeneous anomalies. (ii) Adaptive training-stability framework, including dynamic seed initialization, statistically driven batch-size estimation, automated dropout tuning, and iterative epoch optimization. (iii) Uncertainty-aware decision mechanism, employing Monte Carlo dropout to quantify predictive uncertainty and reduce false-positive predictions. (iv) Extensive cross-domain evaluation, benchmarking the method in diverse medical, industrial, 2D/3D, and surveillance datasets.

The remainder of this paper is organized as follows. Section 2 reviews the literature on reproducibility challenges, optimization strategies, and anomaly detection architectures. Section 3 presents the proposed model and the adaptive optimization pipeline. Section 4 describes the experimental setup. Section 5 reports results, comparative studies, ablation analysis, uncertainty evaluation, and interpretation findings. Section 6 concludes the work and outlines future research directions.

2. Related works

2.1. Autoencoder-based anomaly detection

Autoencoder-based anomaly detection has become a dominant approach in high-dimensional domains due to its ability to learn reconstruction-based representations of normal data without explicitly modeling abnormal samples. Autoencoders compress the input data into a latent space and attempt to reconstruct it. The deviations between the input and the reconstruction are used as anomaly indicators [16]. Convolutional autoencoders (CAEs) [17] introduce convolutional layers to better exploit spatial locality. By replacing fully connected layers with convolutional filters, CAEs learned translation-invariant features and demonstrated strong performance in image-based anomaly detection. Then, to incorporate probabilistic modeling, variational autoencoders (VAEs) [18] were proposed, allowing the latent space to follow a continuous distribution. This generative formulation improved latent regularization and allowed models to represent uncertainty. However, VAEs often struggled with blurry reconstructions that reduced anomaly sensitivity. In parallel, denoising autoencoders (DAEs) [19] were introduced to improve the robustness to noise and perturbations. DAEs intentionally corrupt the input and train the network to recover clean signals, strengthening feature extraction, and reducing overfitting. More recently, memory-augmented autoencoders (MemAEs) [20] incorporated an external memory module that stores prototypical patterns of normal data. During reconstruction, the model retrieves relevant memory items, preventing the network from inadvertently reconstructing anomalies. This design significantly improved anomaly localization, but added complexity and remained sensitive to hyperparameter settings.

Despite the steady evolution of autoencoder architectures, their improvements remain largely architectural. These variants demonstrate a clear progression toward richer representations and better reconstruction fidelity, yet continue to exhibit instability arising from training randomness, sensitivity to static hyperparameter choices, and limited adaptability across heterogeneous datasets. Such instability is problematic in anomaly detection, where small fluctuations in reconstruction behavior can significantly alter anomaly scores. At the same time, conventional single-encoder autoencoders face fundamental representational constraints. Their reliance on a single feature-extraction pathway limits their ability to capture anomalies that manifest at different spatial scales. Fine-grained defects, subtle texture irregularities, or faint pathological signals can be lost or smoothed out during encoding [21], while large-scale contextual deviations may not be adequately disentangled when the encoder is shallow or overly localized. These limitations reduce anomaly sensitivity and contribute to inconsistent performance across domains with varying resolutions or structural complexity. Recent studies have emphasized that single-scale representations restrict an autoencoder's ability to detect heterogeneous anomalies [22], motivating the development of multi-scale and dual branch architectures.

2.2. Multi-scale and dual autoencoder architectures

Efforts to improve anomaly detection sensitivity have increasingly focused on capturing information on multiple spatial scales. Early approaches such as PaDiM [23], demonstrated the benefit of using pretrained CNN encoders and extracting intermediate feature maps at different resolutions. By modeling the distribution of these multi-scale features for normal data, PaDiM was able to detect abnormalities even when they occurred in subtle textural regions. This idea was further refined by PatchCore [14], which introduced a memory bank of patch-level embeddings and reduced redundancy through greedy coreset sampling. PatchCore showed that fine-grained patch representations—combined with global backbone features—could significantly improve anomaly localization while remaining computationally tractable. Next, based on the momentum of feature-pyramid approaches, FastFlow [24] adopted a normalizing-flow backbone to streamline the extraction of multi-scale statistical representations. By designing a flow-based model that jointly accounts for local textures and higher-level semantics, FastFlow improved the inference speed and anomaly segmentation performance. Together, they illustrate a clear progression from handcrafted multi-scale feature extraction toward more principled, distribution-aware representations.

Parallel to these developments, self-supervised and multi-branch reconstruction models have gained traction. CutPaste [21] introduced synthetic augmentation to force the encoder to learn discriminative patch-level representations, thus improving the model's ability to capture small defects without requiring labeled anomalies. Inspired by this, multi-branch generative methods such as CFLOW [25] extended the idea by integrating coarse and fine-grained flows into a unified network. By aligning region-level embeddings with global contextual cues, CFLOW demonstrated stronger anomaly localization than its single-branch counterparts. In addition, multi-stream CNN architectures [26] explored explicit separation of global and local receptive fields, allowing models to process large-scale structural deviations and small-scale texture irregularities simultaneously. Dual encoder architectures extend

this idea by explicitly separating global and local feature extraction. Image-level encoders capture coarse semantic structure, while pixel-level or patch-level encoders model fine-grained textural cues [27]. Recent studies in medical imaging [28] and defect detection [29] demonstrate that dual branches improve the robustness to subtle anomalies. However, these architectures remain highly sensitive to training stability. Their two pathways amplify the discrepancies caused by seed randomness, static batch sizes, and premature convergence. Minor changes in initialization can cause misalignment between latent spaces, reducing the effectiveness of the shared decoder [6]. The potential of dual encoders is therefore limited by the absence of adaptive optimization strategies capable of stabilizing multi-scale learning.

2.3. Automated and evolutionary autoencoder optimization

Over the past few years, researchers have increasingly turned to automated and evolutionary strategies to reduce the reliance on manual model design in anomaly detection. One of the earliest efforts in this direction is EvoAAE [10], which applies evolutionary principles—mutation, crossover, and selection to iteratively refine the structure of an autoencoder. By treating the architecture as a population that evolves over generations, EvoAAE automatically searches for configurations that better capture normal data distributions. This approach alleviates the burden of manual tuning, yet its focus remains strictly on architectural parameters, leaving the training dynamics unchanged. Then, MODEO-CNN [11] introduces a multi-objective evolutionary framework designed to simultaneously optimize two often competing requirements, which are reconstruction accuracy and computational efficiency. Instead of evolving a single objective, MODEO-CNN explores Pareto-optimal trade-offs, enabling the discovery of architectures that achieve strong performance under resource constraints. However, MODEO-CNN still assumes that training conditions such as seed, batch size, and epoch schedule—remain fixed, and thus it does not address the instability inherent in reconstruction-based anomaly detection.

A more advanced line of work is represented by AD-NEv [12] and its improved variant AD-NEv++, [13]. These frameworks employ neuro-evolution to evolve both the topology and hyperparameters of autoencoders. Through iterative evolutionary cycles, the system can discover increasingly complex architectures that adapt to various anomaly detection tasks. ADNEv++ further improves search efficiency and structural diversity, achieving superior performance across image datasets. Despite these advancements, both of them still treat training randomness as external to the optimization loop, relying on fixed seeds and standard training configurations that may introduce reproducibility issues. Across all these frameworks, the evolutionary and automated approaches excel at discovering architectures, but do not stabilize the training process itself. They operate under the assumption that architecture is the primary driver of anomaly detection performance, while overlooking the substantial impact of training randomness, dataset-sensitive hyperparameters, and multi-scale representation alignment—factors that are particularly critical for autoencoder-based detectors.

2.4. Training stability and reproducibility in deep learning

Ensuring reproducibility and consistency in deep learning models allows experiments to produce identical results under the same conditions for fair performance evaluation and comparison. However, randomness in weight initialization, data shuffling, and other training processes can lead to variations in results. Dutta et al. analyzed 114 machine learning projects and found that 461 tests failed due to the absence of seed settings, making it impossible to replicate the results consistently [30]. A common solution is to fix random seeds to ensure that identical operations yield the same results across runs. However, over-reliance on fixed seed can introduce bias by favoring certain weight initializations, creating an illusion of stability. Studies indicate that random seed selection significantly influences performance, with segmentation tasks showing up to 76% fluctuation depending on the random seed [8]. The nnU-Net framework demonstrated that relying on a single seed can produce misleading conclusions about model stability. To mitigate this, multiseed sampling and α-trimming statistics [31] have been proposed, but they increase computational cost and are less practical for large-scale experiments. Moreover, many researchers arbitrarily set seed values without justification, and Bethard reported that more than 50% of machine learning studies misused random seeds [32]. This reliance on fixed seeds obscures true model performance and undermines practical applications.

Batch size is another influential factor affecting gradient variance, convergence speed, and representation smoothness [9]. Optimal batch sizes vary substantially between datasets, especially in anomaly detection, where texture complexity and imbalance differ widely. Using static batch sizes—common in many anomaly detectors—limits adaptability and can cause unstable learning or oversmoothing of fine-grained features. The batch size is often fixed and inherited from previous work, without dataset-specific tuning or theoretical guidance. For instance, CutPaste [21] and PaDiM [23] commonly use fixed batch sizes of 32 or 64 in multiple datasets. These values are chosen empirically and are not adjusted to reflect the scale or noise level of the dataset. Flow-based models such as CFLOW-AD [25] adopt a batch size of 32 during encoder training and a separate fiber batch size of 64 for decoder-side operations. Although effective, these choices are hardcoded and static. PatchCore, as implemented in Anomalib [33], typically operates with a batch size of 16 or 32, with no adaptive mechanism to match dataset variability. Similarly, the GL-CAE model for video anomaly detection [34] uses a fixed mini-batch strategy and does not explore adjustments between sequences of different lengths.

The number of training epochs also influences the model performance. An epoch refers to a complete pass through the training data to update model weights to minimize errors. Insufficient epochs prevent the model from capturing meaningful patterns, while excessive epochs cause it to memorize noise [35]. Recent studies observe rapid error reduction in early epochs before stabilizing [36], suggesting the existence of an optimal training range. However, this pattern can shift due to epoch-wise double descent, where overfitting initially degrades performance, but further training may later improve generalization [37]. Next, the i-Epoch strategy discards weak candidates early based on a fixed epoch threshold [38], but performance depends on the chosen limit. Setting this threshold too low can eliminate promising candidates, while too high can increase training time without added benefit. Adaptive early stopping addresses this by halting training based on validation loss trends rather than fixed counts [39]. However, most implementations rely on static patience values that require tuning per dataset, limiting generalization. This makes it difficult to establish a universal training epoch strategy that effectively balances computational cost and convergence stability.

2.5. Adaptive optimization and hyperparameter tuning

Automated hyperparameter optimization is driven by the need to reduce manual tuning and improve model generalization. Early approaches relied on simple grid or random search, but these methods quickly became impractical as modern models grew deeper and more sensitive to hyperparameters. This motivated the development of more sophisticated optimization strategies built around Bayesian principles and evolutionary exploration. Methods such as Hyperband [40] and BOHB [41] introduced adaptive resource allocation, allowing poorly performing configurations to be discarded early, while promising ones received more computational budget. These approaches significantly reduced the cost of tuning by balancing exploration with efficiency. More recently, frameworks such as Keras Tuner [42] have made automated search accessible through Bayesian optimization, Hyperband,

and evolutionary strategies packaged in user-friendly interfaces. These tools iteratively refine hyperparameter choices such as learning-rate schedules, dropout rates, batch sizes, and activation functions based on performance signals gathered during training. Next, adaptive learning strategies such as curriculum learning [43] demonstrated that models learn more efficiently when exposed to gradually increasing difficulty. This idea was later extended by dynamic augmentation techniques such as AutoAugment [44], which adaptively select augmentation policies to expose models to various variations of the input distribution. Similarly, adaptive regularization strategies [45] modify dropout rates, weight decay, or loss penalties in response to evolving training dynamics, improving network robustness and preventing overfitting.

These automated methods and adaptive mechanisms have proven to be highly effective in supervised tasks such as image classification [46] and natural language processing [47]. The availability of labeled data allows the model to receive continuous feedback on its learning trajectory. However, these methods have seen limited adoption in autoencoder-based anomaly detection. Reconstruction-based models behave differently from supervised networks. Most automated tuning frameworks evaluate a single training run per configuration, assuming that model behavior is stable across attempts. This assumption does not apply in anomaly detection, where reconstruction instability and dataset-specific variance make single-run evaluations unreliable. Moreover, autoencoder-based anomaly detection relies exclusively on modeling the distribution of normal data, without labeled anomalies to guide or correct the learning process. As a result, adaptive strategies must be designed carefully to avoid distorting the normal manifold, oversmoothing fine-grained details, or biasing the latent space toward suboptimal representations.

Recently, researchers have attempted to introduce adaptive elements into unsupervised anomaly detection through learning-rate scheduling [35], progressive resizing [48], and domain adaptation [49]. Learning-rate schedules help stabilize the optimization process, but they typically operate independently of other training parameters. Progressive resizing gradually increases input resolution during training, improving efficiency, but offers no protection against instability caused by seed randomness or static batch sizes. Domain adaptation methods address distribution shifts between training and testing sets, yet they rely on assumptions that do not hold for many reconstruction tasks, where anomalies are rare or structurally diverse. Together, these methods offer promising ingredients, but remain fragmented. Few (if any) existing approaches integrate the three key components required for stable unsupervised anomaly detection: (i) seed adaptivity to ensure reproducibility without biasing weight initialization, (ii) batch-size adaptivity to maintain stable gradient dynamics across datasets with varying complexity, and (iii) epoch adaptivity to prevent both premature convergence and late-stage memorization.

2.6. Interpretability in anomaly detection

Interpretability has become an indispensable requirement in domains where decisions must be transparent, verifiable, and operationally defensible. This is especially true in medical imaging, industrial inspection, and safety-critical systems, where practitioners must understand why a model raises an alert before taking action. Traditional post-hoc explanation methods, including LIME [50], SHAP [51], and integrated gradients [52], attempt to provide insights by approximating the contribution of each input feature to the output of a model. Although effective for supervised classifiers, these methods exhibit fundamental limitations when applied to unsupervised anomaly detection due to their reliance on surrogate explanations that do not reflect reconstruction dynamics or latent-space behavior. Therefore, the anomaly detection community increasingly turns to reconstruction-aware interpretability tools. Reconstruction error maps [53] visualize the pixel-wise differences between the input and its reconstruction, revealing local regions that deviate from the normal patterns learned. Gradient-based saliency methods [54] highlight the spatial areas that most influence the reconstruction objective, while feature-level or attention-based mechanisms [55] provide cues about which regions the model prioritizes during encoding and decoding. These approaches are more faithful to the underlying reconstruction mechanism and have demonstrated improved localization of anomalous areas in industrial and medical datasets.

However, these reconstruction-driven visualizations remain deterministic. They describe what the model reconstructs, but not how confident it is in its reconstruction. In practice, anomaly detection systems often encounter borderline cases—regions with subtle artifacts, ambiguous textures, or low contrast, where deterministic heatmaps fail to distinguish between genuine anomalies and reconstruction uncertainty. Models may also produce low-error reconstructions for abnormal regions if the learned representations are too smooth or biased by training instability. Without a measure of uncertainty, practitioners cannot determine whether a highlighted region reflects a true anomaly or merely a low-confidence prediction. Therefore, uncertainty estimation serves as an important complement to interpretability. Monte Carlo dropout [56] offers a simple yet powerful mechanism to approximate predictive uncertainty by enabling dropout during inference and sampling multiple stochastic reconstructions. The variability in these reconstructions provides an uncertainty score that naturally complements the reconstruction error. However, MC dropout has rarely been integrated into autoencoder-based anomaly detection pipelines. Existing works tend to focus on reconstruction error or post-hoc explanations, leaving a gap in frameworks that jointly provide reconstruction-aware, feature-level, and uncertainty-informed interpretability.

2.7. Summary of gaps and positioning of this work

The reviewed literature reveals several limitations that motivate the need for a more integrated anomaly detection framework. First, single-encoder autoencoders struggle to capture anomalies that manifest on different spatial scales. Second, automated evolutionary frameworks reduce manual configuration but largely overlook the fundamental issue of training reproducibility. Thus, architectural automation alone is insufficient to guarantee model stability. Third, existing adaptive learning strategies address only isolated components of the optimization pipeline. These methods rarely offer coordinated adaptation of seed initialization, batch-size scaling, and epoch optimization, and none incorporate multi-attempt stabilization or dataset-driven learning dynamics. As a result, current adaptive techniques lack the comprehensive control needed to stabilize multi-encoder architectures across heterogeneous datasets. Finally, although interpretability and uncertainty estimation are increasingly recognized as essential components of trustworthy anomaly detection, they remain largely absent from mainstream frameworks. Table 1 provides a consolidated comparison of representative methods in the reviewed categories.

To address these limitations, the proposed framework unifies multiple strategies into a single integrated system. It combines a dual encoder architecture for multi-scale representation with a comprehensive adaptive optimization pipeline that includes dynamic seed initialization, statistically derived batch-size scaling, iterative epoch optimization with multi-attempt checkpointing, automated dropout tuning via Keras Tuner, and Monte Carlo dropout for predictive uncertainty estimation. Together, these components offer a lightweight and robust alternative to architecture-search-based approaches, enabling reproducible, multi-scale, and uncertainty-aware anomaly detection across diverse domains.

3. Methodology

3.1. Overview of the proposed framework

The proposed anomaly detection framework integrates a dual autoencoder architecture with an adaptive optimization pipeline to

Table 1
Summary of related anomaly detection methods and limitations.

Category	Representative Methods	Mechanism	Limitations
Single-encoder autoencoders	CAE, VAE, DAE, MemAE	Learn compressed latent representation and reconstruct input; anomaly score is the reconstruction error	Limited to single-scale features; miss fine-grained anomalies; highly sensitive to seed, batch size, training randomness
Multi-scale methods	PaDiM, PatchCore, FastFlow	Extract multi-scale features from pretrained backbones; compare embeddings with normal distribution	Rely on pretrained CNNs; not reconstruction-based; limited interpretability; computationally heavy
Dual autoencoder architectures	CutPaste, CFLOW, Dual AE variants	Combine global and local feature pathways for multi-scale anomaly representation	Very sensitive to hyperparameters; unstable without adaptive seeds, batch size, or epoch tuning
Automated architecture search	EvoAAE, MODEO-CNN, AD-NEv, AD-NEv++	Use neuro-evolution/genetic algorithms to evolve network structure	Do not address training stability; rely on fixed seeds, static batch size, conventional early stopping; no uncertainty estimation; computationally expensive
Hyperparameter search	Hyperband, BOHB, Keras Tuner	HUse Bayesian or evolutionary search for hyperparameters	Optimizes single runs only; does not stabilize multi-encoder architectures; no seed/batch/epoch adaptation
Interpretability methods	LIME, SHAP, Grad-CAM, MC Dropout	Provide post-hoc explanations or uncertainty estimates	Post-hoc, not integrated; may produce misleading explanations; limited use for reconstruction models

enhance feature representation, training stability, and decision reliability. The architecture employs two complementary encoders that operate on the same input image. The first encoder captures the global image structure, modeling large-scale spatial patterns and overall shape characteristics. The second encoder focuses on local features, emphasizing fine-grained textures and small deviations that may correspond to early or subtle anomalies. This dual pathway design enables the model to learn multi-scale descriptions of normality that cannot be achieved through a single encoder. Then, the latent representations produced by both encoders are concatenated and projected into a unified space before being passed to a shared decoder. The decoder reconstructs the input image using this fused latent representation, thereby integrating global context with local detail within a single reconstruction process. This fusion mechanism strengthens the model's ability to capture heterogeneous anomalies and contributes to a more stable reconstruction in a variety of input types.

To address the sensitivity of unsupervised training in initialization and hyperparameter choices, the architecture is paired with an adaptive optimization pipeline. The training procedure incorporates dynamic seed initialization, dataset-aware batch size scaling, automated dropout tuning, and attempt-level epoch selection. These components are designed to mitigate the variance that arises from fixed seeds, heuristic batch parameters, and rigid training schedules. During inference, the framework incorporates uncertainty estimation via Monte Carlo dropout. Multiple stochastic forward passes are made through the decoder to quantify the variability in reconstruction. This uncertainty-aware mechanism improves anomaly scoring by distinguishing confidently reconstructed normal samples from unstable or ambiguous reconstructions often associated with anomalies. Fig. 1 provides an overview of the complete workflow of the proposed framework.

3.2. Dual autoencoder architecture

Let $I \in \mathbb{R}^{A\times B}$ denote an input image of spatial size $A \times B$, and let

$$x = \mathrm{vec}(I) \in \mathbb{R}^d, \qquad d = A \cdot B, \tag{1}$$

be its vectorized representation. The proposed dual autoencoder maps x into two complementary latent embeddings: a global (image-level) embedding Z_i and a local (pixel-level) embedding Z_p. These embeddings are then concatenated and passed through a shared decoder to obtain a reconstruction $\hat{x}$, which is reshaped back to image form $\hat{I} \in \mathbb{R}^{A\times B}$.

3.2.1. Global image encoder

The global encoder f_i is designed to capture high-level structural and contextual information from the entire image. It is parameterized by θ_i and defined as

$$Z_i = f_i(I, \theta_i) = f_i\big(\mathrm{vec}(I), \theta_i\big) = f_i(x, \theta_i) \in \mathbb{R}^{d_i}, \tag{2}$$

where d_i is the dimensionality of the global embedding.

In the implementation, f_i is realized as a stack of fully-connected layers with leaky-ReLU activations and dropout regularization, applied to the flattened input x. Let

$$h_i^{(0)} = x, \tag{3}$$

and for $\ell = 1, \ldots, L_i$ define

$$h_i^{(\ell)} = \sigma\Big(W_i^{(\ell)} h_i^{(\ell-1)} + b_i^{(\ell)}\Big), \tag{4}$$

where $W_i^{(\ell)}$ and $b_i^{(\ell)}$ are the weight matrix and bias vector of the ℓ-th layer, and $\sigma(\cdot)$ denotes the element wise leaky-ReLU nonlinearity. During training, dropout is applied to each hidden representation $h_i^{(\ell)}$,

$$\tilde{h}_i^{(\ell)} = m_i^{(\ell)} \odot h_i^{(\ell)}, \qquad m_i^{(\ell)} \sim \mathrm{Bernoulli}(1 - p_i^{(\ell)}), \tag{5}$$

where $p_i^{(\ell)}$ is the layer-specific dropout rate and $\odot$ denotes element-wise multiplication. For clarity of notation, we keep $h_i^{(\ell)}$ to denote the effective hidden state after dropout.

The final global embedding is obtained as

$$Z_i = h_i^{(L_i)} \in \mathbb{R}^{d_i}. \tag{6}$$

In the experiments, we use $L_i = 4$ hidden transformations with decreasing widths

$$d = AB \;\to\; 2048 \;\to\; 1024 \;\to\; 512 \;\to\; 256 = d_i, \tag{7}$$

which allows the encoder to aggregate global structure while compressing the input into a compact image-level representation.

3.2.2. Local pixel encoder

The local encoder f_p is introduced to preserve fine-grained variations that may be suppressed by global representations. It shares the same input I but is parameterized independently by θ_p.

$$Z_p = f_p(I, \theta_p) = f_p\big(\mathrm{vec}(I), \theta_p\big) = f_p(x, \theta_p) \in \mathbb{R}^{d_p} \tag{8}$$

Analogous to the global encoder, f_p is implemented as a multi-layer perceptron with leaky-ReLU activations and dropout. Let

$$h_p^{(0)} = x, \tag{9}$$

and for $\ell = 1, \ldots, L_p$ define

$$h_p^{(\ell)} = \sigma\Big(W_p^{(\ell)} h_p^{(\ell-1)} + b_p^{(\ell)}\Big), \tag{10}$$

with dropout

$$\tilde{h}_p^{(\ell)} = m_p^{(\ell)} \odot h_p^{(\ell)}, \qquad m_p^{(\ell)} \sim \mathrm{Bernoulli}(1 - p_p^{(\ell)}). \tag{11}$$

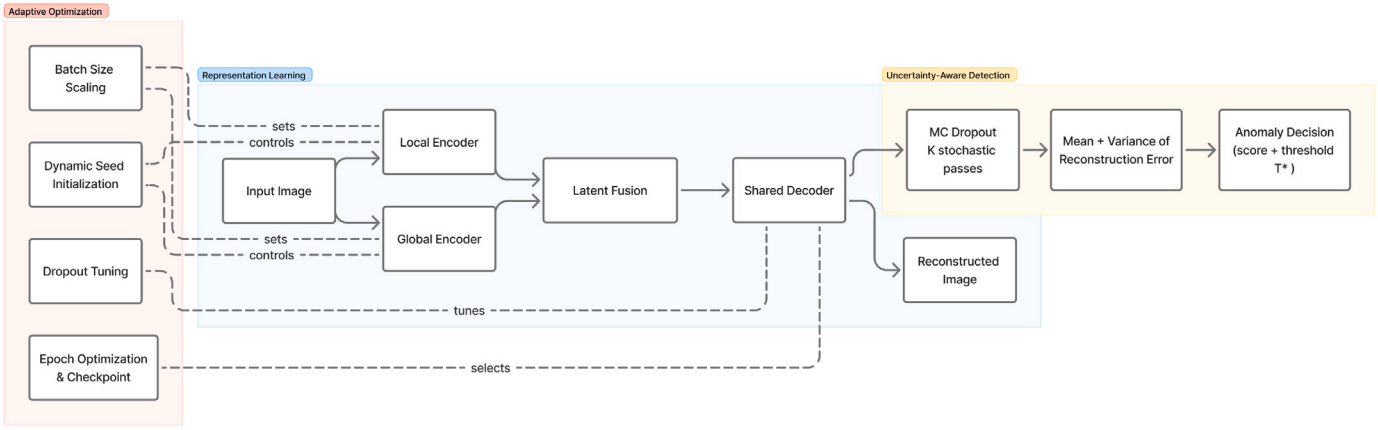

Fig. 1. Overview of the proposed anomaly detection framework.

The resulting pixel-level embedding is

$$Z_p = h_p^{(L_p)} \in \mathbb{R}^{d_p}. \tag{12}$$

In practice, we match the global encoder depth and widths,

$$d = AB \;\to\; 2048 \;\to\; 1024 \;\to\; 512 \;\to\; 256 = d_p, \tag{13}$$

but allow the dropout rates $p_p^{(\ell)}$ to be tuned independently via Keras Tuner. This design encourages the *pixel* encoder to be more sensitive to subtle, low-contrast patterns, leveraging different regularization configurations even when the architectural widths are identical.

3.2.3. Latent fusion and shared decoder

The outputs of the two encoders are concatenated to form a unified multi-scale latent representation.

$$Z = [Z_i \parallel Z_p] \in \mathbb{R}^{d_z}, \qquad d_z = d_i + d_p. \tag{14}$$

With $d_i = d_p = 256$ in our implementation, we obtain $d_z = 512$. Then, the shared decoder g_ϕ maps Z back to the image space. Let

$$h_d^{(0)} = Z, \tag{15}$$

and for $\ell = 1, \ldots, L_d$ define

$$h_d^{(\ell)} = \sigma\left(W_d^{(\ell)} h_d^{(\ell-1)} + b_d^{(\ell)}\right), \tag{16}$$

again with dropout masks $m_d^{(\ell)}$ applied during training

$$\tilde{h}_d^{(\ell)} = m_d^{(\ell)} \odot h_d^{(\ell)}, \qquad m_d^{(\ell)} \sim \text{Bernoulli}(1 - p_d^{(\ell)}). \tag{17}$$

The final reconstruction in vectorized form is obtained via a sigmoid output layer.

$$\hat{x} = \sigma_{\text{sig}}\left(W_d^{(L_d+1)} h_d^{(L_d)} + b_d^{(L_d+1)}\right) \in \mathbb{R}^d, \tag{18}$$

and reshaped back to the image format,

$$\hat{I} = \text{reshape}(\hat{x}) \in \mathbb{R}^{A \times B}. \tag{19}$$

In the experiments, we use

$$d_z = 512 \;\to\; 512 \;\to\; 1024 \;\to\; 2048 \;\to\; d = AB, \tag{20}$$

which directly mirrors the encoder widths and facilitates symmetric reconstruction.

3.2.4. Training objective

The dual autoencoder is trained on normal samples using a reconstruction-based objective. Given a dataset $\{I^{(n)}\}_{n=1}^N$ of normal images and their vectorizations $x^{(n)} = \text{vec}(I^{(n)})$, the reconstruction for each sample is

$$\hat{x}^{(n)} = g_\phi\left([f_i(x^{(n)}, \theta_i); f_p(x^{(n)}, \theta_p)]\right). \tag{21}$$

Table 2
Layer-wise configuration of the global image encoder f_i and the local pixel encoder f_p. Dropout rates r_ℓ are selected automatically by the adaptive optimization pipeline.

Block	Layer type	Output shape	Activation	Dropout
Input	–	$(A, B, 1)$	–	–
1	Flatten	$A \cdot B$	–	–
2	Dense	2048	leaky-ReLU	[0, 0.5]
3	Dense	1024	leaky-ReLU	[0, 0.5]
4	Dense	512	leaky-ReLU	[0, 0.5]
5	Dense	256	leaky-ReLU	0

We employ the mean squared logarithmic error (MSLE) loss,

$$\ell_{\text{MSLE}}(x, \hat{x}) = \frac{1}{d} \sum_{j=1}^{d} \Big(\log(1 + x_j) - \log(1 + \hat{x}_j)\Big)^2, \tag{22}$$

and minimize the empirical risk

$$\mathcal{L}(\theta_i, \theta_p, \phi) = \frac{1}{N} \sum_{n=1}^{N} \ell_{\text{MSLE}}\big(x^{(n)}, \hat{x}^{(n)}\big). \tag{23}$$

Optimization is carried out using the Adam optimizer with an adaptive training strategy described in Section 3.3.

3.2.5. Architectural details

Both the global image encoder f_i and the local pixel encoder f_p share the same architectural structure. The two sub-networks are architecturally identical multilayer perceptrons (MLPs) operating on different views of the same input. This design choice ensures that the dual encoder framework isolates the effect of feature perspective (global vs. local) without introducing confounding changes due to differing network depth, width, or parameter count. The complete layer configuration is summarized in Table 2.

The shared decoder receives the concatenated latent representation and maps it back to the original image domain through a sequence of dense layers with progressively increasing dimensionality, as shown in Table 3.

3.3. Adaptive training stability pipeline

Deep learning-based anomaly detection is highly sensitive to stochastic variations arising from weight initialization, data shuffle, hyperparameter selection, and convergence dynamics. Variations in random seeds, batch sizes, early stopping behavior, and dropout rates directly influence the resulting anomaly scores, often leading to training instability, reduced reproducibility, and inconsistent detection performance across datasets of differing size or complexity. To mitigate these issues, the proposed framework integrates an Adaptive Training Stability

Table 3
Layer-wise configuration of the shared decoder g_ϕ. Dropout rates r_ℓ are tuned automatically as part of the adaptive optimization pipeline.

Block	Layer type	Output shape	Activation	Dropout
Input	–	512	–	–
1	Dense	512	leaky-ReLU	[0, 0.5]
2	Dense	1024	leaky-ReLU	[0, 0.5]
3	Dense	2048	leaky-ReLU	[0, 0.5]
4	Dense	$A \cdot B$	sigmoid	0
5	Reshape	$(A, B, 1)$	–	–

Pipeline (ATSP), a unified optimization process that regulates randomness, normalizes learning dynamics, enforces stable convergence, and quantifies predictive uncertainty.

3.3.1. Dynamic seed initialization

Random seed selection strongly influences the trajectory of optimization in latent-space formation and reconstruction stability. Fixed seeds guarantee reproducibility, but introduce initialization bias, whereas fully random seeds may lead to irreproducible models. ATSP resolves this by assigning dataset-dependent yet deterministic seeds.

$$\text{seed} = (|N| \bmod 200) + k, \tag{24}$$

where N is the dataset size and $k \in \{0, 1, 2\}$ corresponds to the seeds used for Python's `random` module, NumPy, and TensorFlow, respectively. This method ensures that the randomness across different libraries remains consistent, but does not perfectly mirror each other, reducing unintended dependencies in the stochastic processes of model training. Due to the inclusion of k, the final seed ranges from 0 to 201.

The choice of 200 as the modulus value was determined through empirical evaluation to balance uniqueness and computational efficiency. Smaller modulus values led to increased seed collisions when different datasets received the same seed, while excessively large modulus values did not provide additional benefits and unnecessarily complicated seed distribution. This dynamic seed initialization strategy aligns with the principles of pseudo-random number generation, where the quality and period of the generated sequence depend on the chosen parameters [57]. By deriving seeds from dataset sizes, we introduce controlled randomness that enhances the robustness and reproducibility of deep learning experiments.

3.3.2. Derived adaptive batch size scaling

The batch size is determined by a predefined empirical equation based on the size of the dataset N. The batch size selection process was formulated by empirical analysis in 30 datasets from various domains, including BMAD [58], UCSD [59], MVTec AD [53], MedMNIST [60], and others. The initial experiments showed that no single batch size was optimal for all datasets.

An iterative batch selection process (Algorithm 1) was carried out during the empirical analysis phase. The process started with an initial batch size $B_1 = N/2$. However, if $B_1 > 2000$, the batch size was recursively halved until a feasible value was reached,

$$B_k = \frac{B_{k-1}}{2}, \quad \text{where } B_k \leq 2000.$$

The AUROC value was calculated for each batch size, and a recursive refinement process was applied, iteratively halving and evaluating the batch size until the convergence criteria were met.

Example 1 (*Batch Size Calculation for Bracket Brown from the MPDD Dataset*)**.** The initial batch size for Bracket Brown is set to 131, as the dataset contains 262 samples. Following Algorithm 1, it identifies 98 as the optimal value with the highest AUROC of 97.62%, as shown in Table 4.

Algorithm 1 Batch size selection for optimal AUROC

Require: Dataset size N, AUROC evaluation function $f(B)$, stopping threshold $\epsilon = 10$.
Ensure: Optimal batch size B^*.
Initialize $B_1 \leftarrow \frac{N}{2}$.
Compute $A_1 \leftarrow f(B_1)$.
Set $i \leftarrow 1$, $B_i \leftarrow B_1$, $B_{best} \leftarrow 0$.
while $B_{best} = 0$ **do**
  Set $B_{i+1} \leftarrow \frac{B_i}{2}$.
  Compute $A_{i+1} \leftarrow f(B_{i+1})$.
  if $A_{i+1} \leq A_i$ **then**
    $B_{best} \leftarrow B_i$, $A_{best} \leftarrow A_i$
    $B_{close1} \leftarrow B_{i+1}$, $B_{close2} \leftarrow B_{i+1}$.
  end if
  $i \leftarrow i + 1$.
end while
$B_{alt} \leftarrow \arg\max_{B \in \{B_{close1}, B_{close2}\}} f(B)$.
while $|B_{alt} - B_{best}| \geq \epsilon$ **do**
  Compute $B_{mid} \leftarrow \frac{B_{best} + B_{alt}}{2}$, $A_{mid} \leftarrow f(B_{mid})$.
  if $A_{mid} > A_{best}$ **then**
    $B_{alt} \leftarrow B_{best}$, $A_{alt} \leftarrow A_{best}$
    $B_{best} \leftarrow B_{mid}$, $A_{best} \leftarrow A_{mid}$.
  else
    $B_{alt} \leftarrow B_{mid}$, $A_{alt} \leftarrow A_{mid}$.
  end if
  if $|A_{alt} - A_{best}| < 1$ **then**
    Break
  end if
end while
return $B^* = B_{best}$.

Table 4
Batch size selection process for Bracket Brown.

Batch size	AUROC value	Calculation for the next batch size
131	86.76	$131/2 = 65.5 \approx 65$
65	91.94	$65/2 = 32.5 \approx 32$
32	86.64	$(65 + 131)/2 = 98$
98	**97.62**	$(98 + 65)/2 = 81.5 \approx 81$
81	97.62	end

Table 5
Batch size selection process for RSNA Pneumonia.

Batch size	AUROC value	Calculation for the next batch size
13,342	–	$13342/2 = 6671$
6,671	–	$6671/2 = 3335.5 \approx 3335$
3,335	–	$3335/2 = 1667.5 \approx 1667$
1667	60.71	$1667/2 = 833.5 \approx 833$
833	69.62	$833/2 = 416.5 \approx 416$
416	85.21	$416/2 = 208$
208	74.10	$(208 + 416)/2 = 312$
312	91.54	$(312 + 416)/2 = 364$
364	**96.95**	$(364 + 338)/2 = 338$
338	92.54	$(364 + 338)/2 = 351$
351	95.14	$(364 + 351)/2 = 357.5 \approx 357$
357	96.06	end

Example 2 (*Batch Size Calculation for RSNA Pneumonia from the BMAD Dataset*)**.** The initial batch size for RSNA is set to 13,342, as the dataset contains 26,684 samples. Since $B_1 > 2000$, it was recursively halved until an evaluable value was reached. Algorithm 1 identified 364 as the optimal batch size with the highest AUROC of 96.95%, as shown in Table 5.

The empirical trend visualized in Fig. 2 reveals two distinct scaling behaviors in the relationship between dataset size N and the batch size that produces the highest AUROC. Smaller datasets exhibit a curved growth pattern dominated by logarithmic sensitivity, while larger datasets follow a flatter and gradually decreasing trajectory,

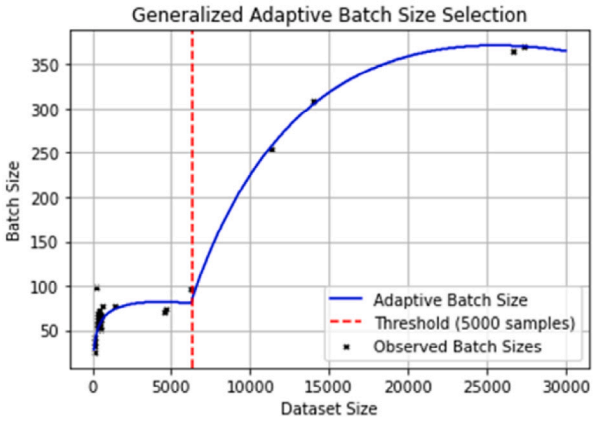

Fig. 2. Optimal AUROC-driven batch size for 30 datasets with fitted piecewise scaling.

where AUROC gains saturate as batch size increases. The boundary at $N = 6300$ marks a clear regime transition, justifying a piecewise batch-size formulation instead of a single global rule. Based on the curve fitting performed on the 30-dataset aggregation, the adaptive batch size estimate given by Eq. (25).

$$\text{Batch} = \begin{cases} -0.0045N + 19.3299 \log N - 59.7382, & N \le 6300, \\ 444.1692 \log N - 0.0174N - 3691.6901, & N > 6300 \end{cases} \tag{25}$$

The fitted analysis reveals two different batch-size scaling behaviors. For datasets with $N \le 6300$ (first regime), the equation mixes a small negative linear term with a logarithmic growth term. This allows the batch size to increase gradually as N grows, which helps stabilize training by lowering the gradient noise when data is limited. For datasets with $N > 6300$ (second regime), the trend changes, as larger batches no longer improve AUROC as much. The equation reflects this by applying a stronger negative linear term, which restricts excessive batch growth and favors moderate batches for large datasets.

3.3.3. Iterative epoch optimization with multi-attempt convergence control

Epoch selection is a dominant factor in reconstruction quality. The proposed framework incorporates an epoch optimization strategy that integrates early stopping with a model checkpoint mechanism to dynamically adjust the number of epochs based on the model's learning progress. Early stopping prevents excessive training by stopping the process when further iterations do not significantly improve performance [61], while the checkpoint mechanism ensures that the best-performing model is retained.

Step 1 — Multi-attempt training. The model is trained for up to ten attempts. Each attempt begins with a reshuffled dataset but uses the same deterministic seeds, enabling diverse convergence trajectories while preserving reproducibility.

Step 2 — Detection metric monitoring. During training, a stability-oriented detection metric is computed by evaluating the sum of TPR (true positive rate) and TNR (true negative rate) at each iteration,

$$M_t = \text{TPR}_t + \text{TNR}_t \tag{26}$$

where M_t represents the detection metric at iteration t.

Step 3 — Exponential smoothing. An exponential moving average (EMA) technique is applied to the detection metric to stabilize the decision-making process,

$$S_t = \alpha M_t + (1 - \alpha) S_{t-1} \tag{27}$$

where S_t is the smoothed detection metric at iteration t, and α is the smoothing factor controlling the influence of recent values. This approach prevents small, temporary fluctuations in the detection metric from prematurely triggering early stopping.

Step 4 — Adaptive early stopping. If the metric in Eq. (26) does not improve for two consecutive iterations,

$$M_t \le M_{t-1} \quad \text{and} \quad M_{t-1} \ge M_{t-2}$$

early stopping is triggered, ensuring that the model does not stagnate in local optima.

Step 5 — Checkpoint preservation. The final model is selected as the one achieving the lowest validation loss across all attempts,

$$\theta^* = \arg\min_{\theta_t} \mathcal{L}_{\text{val}}(\theta_t). \tag{28}$$

where θ^* represents the best model parameters at iteration t. Instead of keeping all intermediate models, the checkpoint function only saves the model with the lowest validation loss. This ensures that even if training continues beyond the optimal epoch count, the framework retains the most generalizable model.

3.3.4. Monte Carlo dropout for uncertainty estimation

Deterministic autoencoders do not offer a measure of predictive certainty. ATSP enables uncertainty estimation by activating dropout at inference time. Given an input image I, the decoder g_ϕ produces T stochastic reconstructions by sampling different dropout masks.

$$\hat{x}^{(t)} = g_\phi\left([Z_i \parallel Z_p], m_d^{(t)}\right), \quad t = 1, \dots, T \tag{29}$$

where $Z_i = f_i(x, \theta_i)$ and $Z_p = f_p(x, \theta_p)$ are the global and local encoder embeddings of the flattened input $x = \text{vec}(I)$, and $\hat{I}^{(t)} = \text{reshape}(\hat{x}^{(t)})$ is the reconstructed image for sample t.

For each reconstruction, the model computes the L2 reconstruction error,

$$\epsilon^{(t)} = \left\| \hat{I}^{(t)} - I \right\|_2. \tag{30}$$

Uncertainty is then estimated from the mean and variance of these T error samples,

$$\mu_\epsilon = \frac{1}{T} \sum_{t=1}^{T} \epsilon^{(t)}, \tag{31}$$

$$\sigma_\epsilon^2 = \frac{1}{T} \sum_{t=1}^{T} \left(\epsilon^{(t)} - \mu_\epsilon\right)^2. \tag{32}$$

A low value of σ_ϵ^2 suggests stable reconstructions and higher model certainty, while high variance reflects inconsistent reconstructions, indicating lower confidence. In anomaly detection, images containing abnormalities typically yield both high error and high variance, which can support more reliable threshold selection, especially near the decision boundary.

3.4. Interpretability of the reconstruction process

The interpretability module is designed to explain how the model reconstructs normal and anomalous patterns using pixel-level evidence derived from the reconstruction objective itself, without altering the encoder–decoder formulation previously established in Section 3.2.

3.4.1. Pixel-wise reconstruction error map

Given a reconstructed output $\hat{x}$ generated by the shared decoder g_ϕ, the framework computes a spatial reconstruction error map $\mathcal{E}_{\text{recon}} \in \mathbb{R}^{A \times B}$ as

$$\mathcal{E}_{\text{recon}}(u, v) = \left(x(u, v) - \hat{x}(u, v)\right)^2, \tag{33}$$

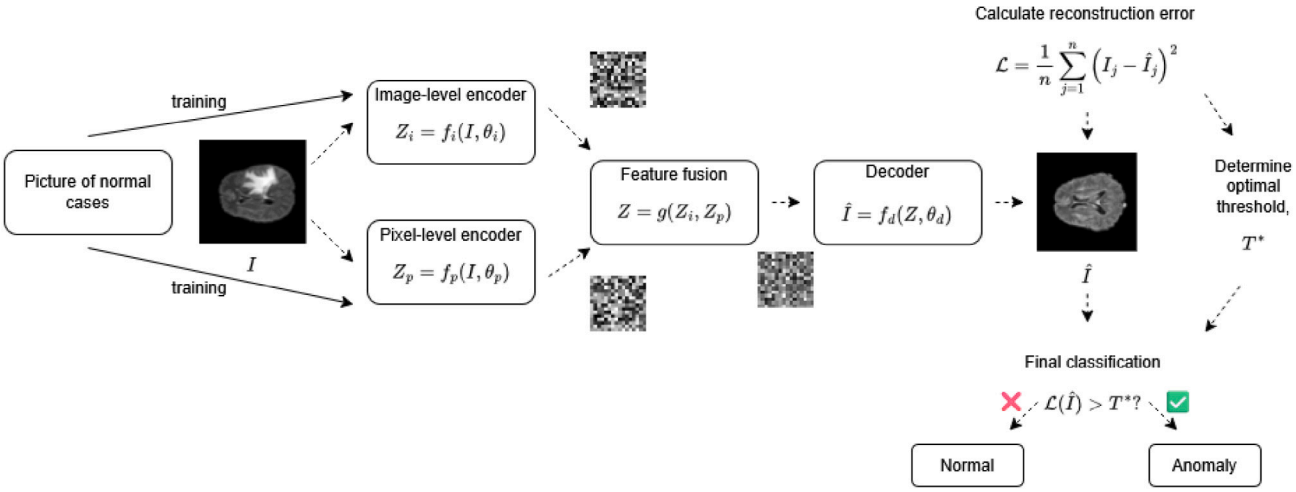

Fig. 3. Example of the anomaly detection process using the proposed dual autoencoder framework. The model is trained exclusively on normal samples. At inference, a test image I is processed by the global encoder $f_i(I, \theta_i)$ and the local encoder $f_p(I, \theta_p)$. The resulting latent codes Z_i and Z_p are concatenated and passed to the shared decoder g_ϕ to obtain the reconstruction $\hat{I}$. A reconstruction-based anomaly score is computed and compared against the ROC-derived threshold T^*; images with scores exceeding T^* are classified as anomalous, and their reconstruction error maps are visualized as heatmaps for interpretability.

where (u, v) indexes the spatial coordinates of the resized input image. The error map is normalized to $[0, 1]$ using min–max scaling

$$\mathcal{E}_{\text{recon}}^{\text{norm}} = \frac{\mathcal{E}_{\text{recon}} - \min(\mathcal{E}_{\text{recon}})}{\max(\mathcal{E}_{\text{recon}}) - \min(\mathcal{E}_{\text{recon}}) + \epsilon}, \tag{34}$$

with $\epsilon \ll 1$ preventing numerical instability. The resulting map is visualized as a heatmap overlay to locate reconstruction deviations associated with anomalies.

3.4.2. Reconstruction objective input sensitivity

To capture which pixels most influence the reconstruction behavior, an absolute gradient (saliency) map is derived from the reconstruction loss $\ell_{\text{MSLE}}(x, \hat{x})$ with respect to the input,

$$G_{\text{sal}}(u, v) = \left| \frac{\partial \ell_{\text{MSLE}}(x, \hat{x})}{\partial x(u, v)} \right|. \tag{35}$$

Since the model is trained using the mean squared logarithmic error, the gradient map reflects the sensitivity of input pixels to reconstruction discrepancies in log space, offering reviewers direct visibility into the reconstruction dynamics optimized by the ATSP.

3.5. Training procedure

This subsection summarizes how these components are integrated into a single end-to-end training routine for each dataset. For a given dataset, all images are first resized to a fixed spatial resolution and vectorized, as described in Section 3.2. The data is then shuffled and partitioned into three disjoint subsets: a training set used to learn the reconstruction function, a validation set used for model and hyperparameter selection, and a held-out test set used exclusively for final evaluation.

To regulate randomness in a reproducible yet dataset-specific manner, ATSP initializes random seeds using the dynamic seed initialization strategy in Section 3.3.1. Then, adaptive batch size is computed using the piecewise scaling rule in Eq. (25) to ensure both convergence stability and hardware feasibility. Given these initial settings, the model is trained using the MSLE reconstruction loss defined in the training objective section and optimized with the Adam optimizer. Hyperparameters such as dropout rates are automatically tuned using Keras Tuner, which searches over a predefined configuration space and retains the candidate model with the lowest validation loss.

Training proceeds in a multi-attempt fashion as described in the iterative epoch optimization subsection. For each attempt, the dataset is reshuffled using the same deterministic seeds. The model is trained with early stopping guided by the smoothed detection metric in Eq. (26), and the model state that reaches the lowest validation loss is stored by checkpointing. After at most ten attempts, the final model parameters θ^* are selected to ensure that the retained model corresponds to the most stable and generalizable reconstruction behavior observed during training.

3.6. Inference process

Once the model has been trained and the parameters θ^* have been selected, anomaly detection is performed in four main stages: reconstruction, error computation, threshold selection, and classification.

3.6.1. Reconstruction and anomaly scoring

A pixel-wise reconstruction error map $\mathcal{E}_{\text{recon}}$ is computed as described in the interpretability subsection and can be visualized as a heatmap for qualitative analysis. For scalar anomaly scoring, the framework uses the L_2 norm of the reconstruction error as Eq. (30), where higher scores indicate larger deviations from the learned notion of normality.

3.6.2. Adaptive threshold selection

The decision threshold is determined using Receiver Operating Characteristic (ROC) analysis on the validation set. For a grid of candidate thresholds T, the true positive rate $\text{TPR}(T)$ and the false positive rate $\text{FPR}(T)$ are computed, and the optimal threshold T^* is obtained by minimizing the Euclidean distance to the ideal classifier point $(0, 1)$,

$$T^* = \arg\min_T \sqrt{\left(1 - \text{TPR}(T)\right)^2 + \text{FPR}(T)^2}. \tag{36}$$

This minimum distance criterion avoids arbitrary threshold choices and promotes balanced sensitivity and specificity between datasets with different class imbalance characteristics.

3.6.3. Classification and visualization

During deployment, each test image is assigned an anomaly label by comparing its anomaly score $s(I)$ with the optimal threshold T^*:

$$\text{label}(I) = \begin{cases} \text{anomalous}, & s(I) > T^*, \\ \text{normal}, & s(I) \le T^*. \end{cases} \tag{37}$$

In parallel, the normalized reconstruction error map is overlaid on the input image to highlight regions that contribute the most to the anomaly decision, providing an interpretable explanation that can be inspected by domain experts. Fig. 3 illustrates the complete inference workflow, from dual-path encoding and latent fusion through reconstruction, anomaly scoring, threshold-based decision, and interpretability overlay.

For applications requiring additional confidence assessment, the Monte Carlo dropout mechanism in the ATSP can be activated at inference time to generate multiple stochastic reconstructions and estimate the variance of the anomaly scores. High reconstruction variance in conjunction with high anomaly scores typically indicates uncertain but potentially critical anomalies, while low-variance scores correspond to confident decisions.

We provide a structured pseudocode representation that summarizes the entire optimization and anomaly decision workflow. The pseudocode serves to clarify the sequence of operations in a form that can be directly implemented in real anomaly detection pipelines.

```
Pseudocode for the Proposed Method

BEGIN
  Initialize seed:
    seed = (dataset_size mod 200) + k
    NumPy seed = seed (k = 0)
    TensorFlow seed = seed + 1 (k = 1)
    Python Random seed = seed + 2 (k = 2)
  Batch size selection:
    IF dataset_size ≤ 6300:
      batch_size = ⌊ -0.0045N + 19.3299 log(N) - 59.7382 ⌋
    ELSE:
      batch_size = ⌊ 444.1692 log(N) - 0.0174N - 3691.6901 ⌋
    batch_size = max(batch_size, 16)
  Shuffle dataset and split into:
    Training set (train_set)
    Validation set (valid_set)
    Test set (test_set)
  SET early_stopping = False
  SET best_validation_loss = ∞
  SET best_detection_metric = -∞
  SET best_threshold = None
  SET best_model = None
  FOR attempt in range(10):
    Shuffle dataset and re-split
    Initialize Keras Tuner:
      Search for best hyperparameters using Keras Tuner
    CALL Train_Dual_Encoder_Model()
    CALL Compute_Training_Loss()
    CALL Compute_Validation_Loss()
    Save best model:
      θ* ← arg min L_val(θ_t)
    Compute detection metric:
      M_t = TPR_t + TNR_t
      S_t = αM_t + (1 − α)S_{t−1}
    IF M_t ≤ M_{t−1} AND M_{t−1} ≥ M_{t−2}:
      SET early_stopping = True
      BREAK
    CALL Determine_Optimal_Threshold()
  FOR each I_test:
    MC dropout for confidence analysis:
      FOR t = 1 to T: sample decoder mask m_d^(t)
      Compute reconstruction Î^(t)
      Estimate mean μ_err and variance σ²_err
    Compute anomaly score:
      Score(I_test) = ||I_test − Dec(Enc(I_test))||²
    Classify:
      IF Score(I_test) > T*: Label as Anomaly
      ELSE: Label as Normal
  CALL Compute_Evaluation_Metrics()
  Generate heatmap for anomaly visualization
END
```

The dropout layers remain active internally during a separate uncertainty sampling stage on validation and test images, producing T stochastic reconstructions $\hat{I}^{(t)}$ to estimate the mean reconstruction error μ_{err} and variance σ^2_{err}. These values are not used to produce visual reconstructions or replace the primary anomaly score but serve to quantify prediction stability. Low variance corresponds to stable (normal) reconstructions, whereas high variance paired with high reconstruction error supports anomaly decision auditing near the threshold.

4. Experimental setup

This section describes the evaluation environment used to assess the proposed anomaly detection framework. All training strategies and architectural formulation follow the design established in Sections 3.2 and 3.3. Here, we focus only on the empirical assessment pipeline, data characteristics, and inference-time configuration.

4.1. Datasets

To examine model behavior across class imbalance profiles and anomaly characteristics, experiments are conducted on datasets representing three major domains: (i) fine-grained medical anomalies, (ii) scene-dependent surveillance deviations, and (iii) localized or distributed structural defects. The datasets used are

1. BraTS2021 [58]: 11,298 multimodal brain magnetic resonance images (MRI) of 1251 patients, standardized at 1 mm^3 voxel resolution and spatially aligned with a common anatomical template for scale-sensitive medical anomaly evaluation.
2. OCT2017 [58,62]: 84,484 retinal Optical Coherence Tomography (OCT) images, of which 27,315 are normal samples used for one-class training.
3. PneumoniaMNIST [60]: 5856 pediatric chest radiographs from MedMNIST, forming a binary benchmark for infected versus normal evaluation.
4. UCSD [59]: 14,000 video frames from surveillance footage containing motion-related anomalies, including skaters and cyclists.
5. MVTec AD [53] (selected objects):
   - Zipper (391 images): Missing or misaligned teeth.
   - Hazelnut (501 images): Surface cracks, mold, or texture corruption.
   - Metal Nut (335 images): Dents, deformations, or surface scratches.
6. Rope: 432 depth-based rope scans of MVTec 3D-AD [63] that contain 3D anomalies such as knots and fraying.

All images are resized to a common spatial resolution of 68×68 and normalized to $[0, 1]$ before processing by the dual encoders, ensuring consistent latent-space fusion and pixel-aligned reconstruction error analysis.

4.2. Evaluation metrics

Reconstruction-based anomaly detection requires performance measures that capture separability, alert correctness, prediction balance, agreement quality, and inference stability. To address these requirements comprehensively, we evaluate the proposed framework using the following metrics, each representing a distinct axis of detection reliability and performance:

- AUROC (Area Under the ROC Curve): Measures the ability of the anomaly score to classify normal and anomalous samples correctly across all thresholds. As it is threshold-agnostic, AUROC is a primary indicator of separability in unsupervised anomaly detection.
- Precision: Quantifies the proportion of detected anomalies that are truly anomalous. High precision reflects trustworthy alerts and reduces false-alarm burden on domain experts.

- TPR (True Positive Rate) / Recall: Captures the proportion of anomalies correctly detected. Recall is essential in safety-critical applications, where missing anomalies typically incur a higher operational cost.
- TNR (True Negative Rate) / Specificity: Assesses the proportion of normal samples correctly identified as normal. High specificity ensures faithful modeling of normality, which dominates real-world distributions.
- F1-score: The harmonic mean of precision and recall with *equal weighting*,

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{38}$$

- F2-score: A recall-emphasized harmonic mean that *penalizes false negatives more strongly*,

$$F2 = 5 \cdot \frac{Precision \cdot Recall}{4 \cdot Precision + Recall}. \tag{39}$$

  The key distinction between F1 and F2 lies in the misclassification cost structure: F1 balances precision and recall equally, whereas F2 assigns a larger weight to recall, reflecting scenarios where failing to detect anomalies is more costly than raising false positives.
- G-Mean: Evaluates balanced detection capability under label imbalance,

$$G\text{-Mean} = \sqrt{Recall \cdot Specificity}. \tag{40}$$

- Matthews Correlation Coefficient (MCC): Measures prediction correlation and quality in imbalanced settings.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{41}$$

- Cohen's Kappa: Quantifies prediction agreement corrected for chance:

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \tag{42}$$

  where P_o is the observed agreement and P_e is the expected agreement.

4.3. Hardware and software environment

All experiments are conducted on a workstation equipped with a GeForce GTX 1660 GPU (6GB VRAM), an Intel i5-14400F CPU, and 32GB DDR5 RAM. The software stack consists of Python 3.10, TensorFlow 2.10, Keras 2.10, and scientific libraries including NumPy, Pandas, and SciPy. The model hyperparameter search is performed using Keras Tuner. Development and execution are performed in Spyder 5.4.3 within the Anaconda 2.6.1 environment. This configuration confirms that the proposed pipeline can be reproduced efficiently without specialized or high-end hardware dependencies.

5. Results and discussion

5.1. Comparative performance evaluation

5.1.1. Model performance

Table 6 demonstrates that the proposed framework achieves consistently high anomaly discrimination across heterogeneous image domains. The two medical imaging benchmarks (BraTS2021 and OCT2017) demonstrate that the proposed model is highly effective in domains where anomalies exhibit a strong spatial or intensity contrast against normal tissue. BraTS2021 records the highest overall AUROC (97.62%), paired with near-perfect precision (99.97%) and an F1-score of 97.63%, indicating that false alarms are extremely minimal while almost all tumor regions are consistently flagged. This can be explained by the well-structured nature of MRI abnormalities, which occupy larger connected regions and produce sustained reconstruction deviations, allowing the shared decoder to generalize normal anatomy while emphasizing out-of-distribution pathology during inference. Similarly, OCT2017 achieves a competitive AUROC of 96.07% with 99.67% precision and 99.08% TNR, confirming that retinal abnormalities (fluid build-ups, drusen, layer distortions) generate reconstruction errors that are spatially localized and semantically distinct. However, its TPR (92.77%) is slightly lower than BraTS2021, suggesting that a small portion of subtle anomalies remains harder to separate due to inherent variability in retinal thickness and illumination profiles. Overall, these results confirm that the model excels when training distributions are coherent and anomaly signals generate pronounced decoder mismatch.

On the other hand, a pediatric chest radiograph dataset (PneumoniaMNIST) produces a lower AUROC (92.61%) compared to MRI and OCT. Although its TPR remains high at 92.91%, the TNR drops to 92.31%, and the FPR increases substantially (7.69%), resulting in an MCC of 84.53% and a Kappa of 84.48%. This aligns with known difficulties in X-ray anomaly detection, where infected lung opacities often manifest as low-contrast texture changes. Since autoencoder-based models rely on learning dominant normal feature manifolds, overlapping pneumonia features within normal anatomical variance can reduce decoder divergence and increase false positives. Nevertheless, the high F2-score (93.38%) shows that the model still preserves good detection sensitivity, making it suitable for screening but limitations of unsupervised reconstruction methods under overlapping abnormal signals.

Among industrial object datasets, Metal Nut exhibits excellent detection performance (AUROC 96.39%, F1-score 98.67%), supported by high recall and reconstruction stability. This suggests that manufacturing surface anomalies (scratches, dents, cracks, incomplete structures) are visually consistent across normal samples. Zipper, Hazelnut, and Rope display varying tradeoff behavior. Zipper shows the highest TPR (97.92%) and strong F1 (97.41%), but lower TNR (88.00%) and a 12.00% FPR, indicating aggressive detection where even small reconstruction inconsistencies raise anomaly flags. This implies that zipper textures, teeth alignment, and occlusion patterns introduce higher background variance, making normal reconstructions less stable. Hazelnut (AUROC 91.74%) represents the lower bound among industrial datasets. Its FPR (9.38%) and reduced G-Mean (91.73%) indicate that the anomaly scores partially overlap with the normal reconstruction variance. The rope dataset reaches 100% TPR, but the TNR is 88.46%, reflecting that the model generalizes rope curvature patterns well, but it struggles to restrict anomaly boundaries due to geometric flexibility in the normal class.

The UCSD pedestrian dataset achieves one of the highest AUROC values (97.56%) with strong TNR (98.30%) and G-Mean (97.55%), indicating that the model reliably learns the regular motion and appearance manifold of normal video frames. Across all datasets, Cohen's Kappa values consistently mirror MCC trends, confirming stable threshold selection and label consistency between predicted anomalies and ground truth partitions. High precision in most datasets suggests that the model produces tight reconstruction manifolds, with anomaly scores driven largely by decoder mismatch rather than noise artifacts. Strong G-Mean values further imply that class imbalance effects are well-controlled by adaptive batch size and dynamic seed initialization in training.

Fig. 4 presents a comprehensive visual analysis of the model on multiple datasets to complement the quantitative benchmarks in Table 6. The ROC topology in Fig. 4(a) confirms that most of the datasets exhibit a steep early ascent toward TPR ≈ 1.0 and FPR ≈ 0. This indicates that the dual encoder latent fusion successfully forms a compact normal manifold that the shared decoder can reconstruct with low residual energy, while anomalous inputs generate an immediate decoder mismatch. Medical and visually coherent datasets such as BraTS2021, OCT2017, UCSD, Metal Nut, and Rope produce curves tightly aggregated near the optimal corner, confirming that anomaly signals have high separability from learned normal features even before

Table 6
Performance evaluation of the proposed anomaly detection model across different datasets (all metrics are expressed in percentage values).

Dataset	AUROC	F1-score	F2-score	TPR	TNR	FPR	Precision	G-Mean	MCC	Cohen's Kappa
BraTS2021	97.62	97.63	96.28	95.40	99.84	0.16	99.97	97.60	88.31	87.66
OCT2017	96.07	96.10	94.07	92.77	99.08	0.92	99.67	95.87	86.68	85.96
PneumoniaMNIST	92.61	94.10	93.38	92.91	92.31	7.69	95.32	92.61	84.53	84.48
UCSD	97.56	94.52	95.89	96.82	98.30	1.70	92.32	97.55	93.36	93.32
Zipper	92.96	97.41	97.71	97.92	88.00	12.00	96.91	92.83	87.24	87.21
Hazelnut	91.74	93.69	93.19	92.86	90.63	9.38	94.55	91.73	82.95	82.93
Metal Nut	96.39	98.67	98.67	98.67	94.12	5.88	98.67	96.37	92.78	92.78
Rope	94.23	94.83	97.86	100.00	88.46	11.54	90.16	94.05	89.31	88.74

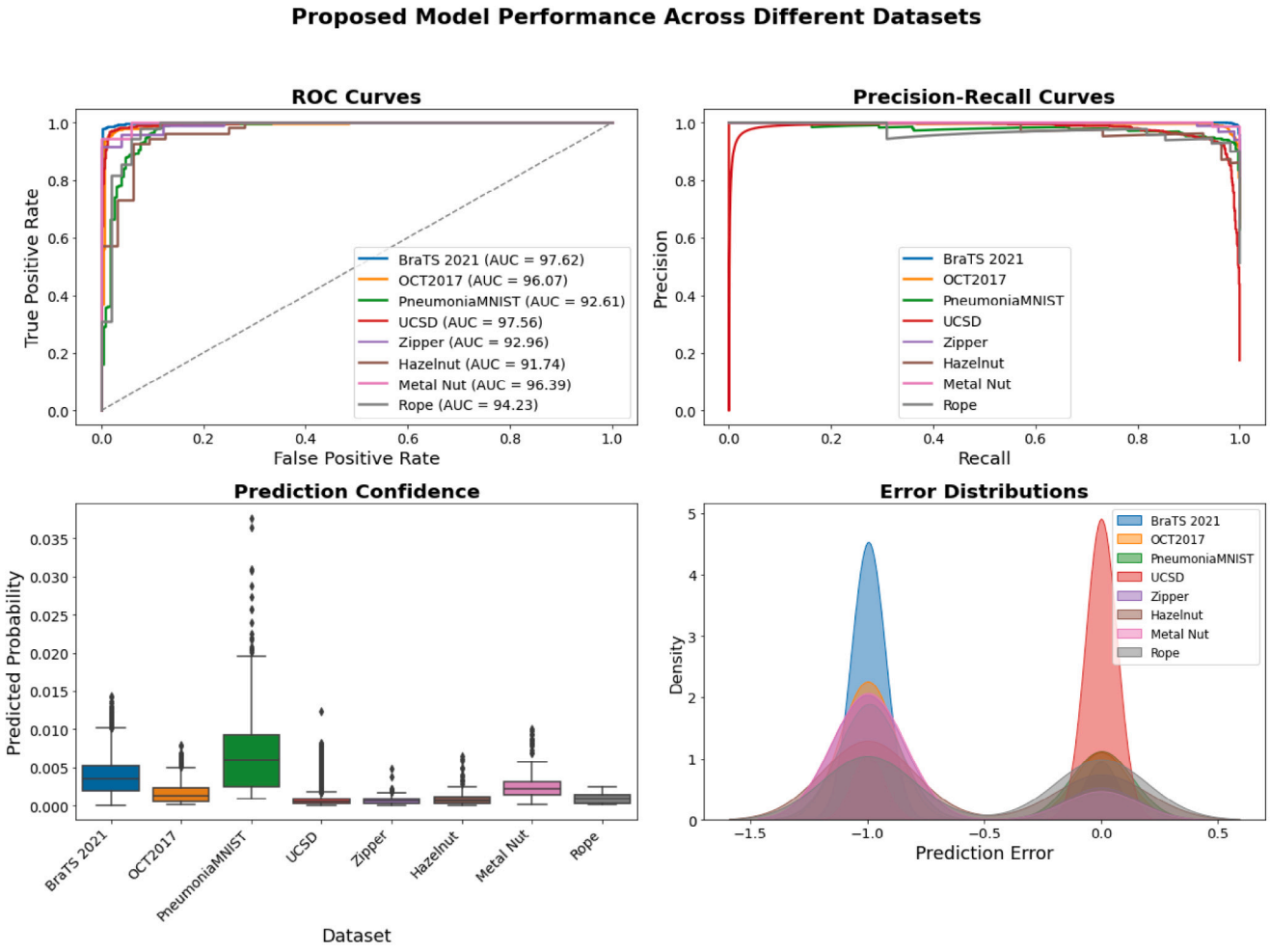

Fig. 4. Proposed model performance across different datasets. (**a**) ROC curves comparing true positive rate versus false positive rate. (**b**) Precision–recall curves showing the trade-off between precision and recall. (**c**) Prediction confidence distributions. (**d**) Error distributions showing the density of prediction errors.

thresholding. In contrast, Hazelnut and PneumoniaMNIST show a slight inward curve in the low FPR region. This does not reflect poor representation learning, but suggests score intersection with high variance normal samples under subtle or low contrast decoder divergence.

The precision–recall behavior in Fig. 4(b) shows that precision remains consistently high across the recall range, indicating stable recall gains without excessive false positive inflation. Additionally, the boxplot of predicted probabilities in Fig. 4(c) illustrates variations in model confidence across datasets. The Rope and Zipper datasets exhibit a lower variance in predicted probabilities, indicating stable predictions with high confidence. In contrast, the PneumoniaMNIST dataset demonstrates a wider distribution, suggesting greater uncertainty in classification due to the complexity and variability of pneumonia-related abnormalities in pediatric chest radiographs.

The error distribution plot in Fig. 4(d) further reinforces the model performance trends. Across all datasets, the model produces a clear and dominant peak at very low error values corresponding to normal samples that the decoder accurately reconstructs. For structured datasets such as BraTS2021, OCT2017, UCSD, and Metal Nut, the separation between the normal peak and anomaly peak is wide, showing that anomalous inputs produce strong and distinct reconstruction deviations. In contrast, some datasets display a heavier overlap between the two peaks. Hazelnut and Zipper show the largest overlap because normal samples themselves contain small natural variations that occasionally produce higher reconstruction errors.

5.1.2. *Baseline methods for comparative study*

To ensure a comprehensive and fair evaluation, we compare the proposed framework with 11 state-of-the-art (SOTA) anomaly detection

Table 7
Summary of the 11 state-of-the-art (SOTA) anomaly detection baselines compared in this study.

Model	Category/Mechanism	Brief description
ADPR [64]	Reconstruction-based CNN	Learns normal appearance via pixel-wise reconstruction; strong on simple textures but sensitive to noise.
CutPaste [21]	Self-supervised augmentation	Creates synthetic defects through patch cut-and-paste; effective on industrial textures but limited in complex medical domains.
PaDiM [23]	Embedding distribution modeling	Uses pre-trained CNN features and models per-pixel multi-variate Gaussian distributions; strong on MVTec but weak in high-variance images.
UTRAD [65]	Transform-based reconstruction	Learns both reconstruction and transformation robustness; good localization but unstable under small defects.
PatchCore [14]	Memory bank nearest-neighbor	Stores sparse high-dimensional normal embeddings; excellent on industrial datasets but memory-heavy and unstable on medical scans.
CFlow [25]	Flow-based model (normalizing flow)	Learns invertible mapping to compute likelihood; fast but suffers from likelihood overestimation and collapse on irregular textures.
CS-Flow [66]	Conditional flow-based model	Extends flow mappings with conditional embeddings; better than CFlow on some textures but highly dataset-dependent.
SimpleNet [67]	Lightweight embedding model	Efficient CNN-based anomaly scoring; good speed but limited feature depth reduces separability in subtle anomalies.
CITN [68]	Transformer-based normal embedding	Uses contextual transformer blocks; strong on structured domains but prone to attention collapse on small defects.
MGAD [69]	Multi-scale generative model	Models anomalies at multiple resolutions; performs well on high-texture datasets but unstable on fine-grained medical images.
MPDE [15]	Multi-scale pixel-distribution estimation	Learns statistical distribution across pixel hierarchies; strong overall but sensitive to contrast variations and modality shifts.

baselines spanning reconstruction, density estimation, self-supervision, flow-based modeling, and transformer-based representations. Table 7 summarizes each method and its core mechanism.

Fig. 5 provides a comprehensive comparison of the proposed framework with the SOTA anomaly detection models in four evaluation metrics: AUROC, F1-score, TPR, and G-Mean. Note that darker heatmap intensities indicate higher performance values. The AUROC heatmap in Fig. 5(a) shows that the proposed model consistently attains high AUROC values across all eight datasets. This stability reflects the effectiveness of the dual-path representation and adaptive training pipeline in maintaining reliable decision boundaries. In contrast, competing methods exhibit pronounced dataset-dependent fluctuations. PatchCore and MPDE remain competitive on BraTS2021 and Rope but deteriorate on Zipper and Hazelnut. CFLOW demonstrates the most severe instability, with AUROC falling below 40% on OCT2017 and approaching zero on Rope. Fig. 5(b) further confirms the stability of the proposed method through consistently high F1-scores, outperforming all baselines across datasets. The high F1-score indicates that the model manages the trade-off between precision and recall effectively, even in domains prone to false positives (e.g., Rope, Zipper) or low-contrast anomalies (e.g., PneumoniaMNIST). Competing models show highly uneven behavior; for example, CFLOW collapses entirely on Rope (0.00%) and yields only moderate performance on OCT2017 and Hazelnut. PatchCore performs well on structured datasets, but degrades on domains where anomaly signals are subtle. The proposed method's ability to maintain F1-scores above 93% in all datasets reflects the advantage of reconstruction-aware scoring combined with adaptive thresholding.

The consistently high TPR indicates that the model reliably identifies anomalies with low reconstruction energy or small-scale deviations. The TPR heatmap in Fig. 5(c) highlights the strong anomaly sensitivity of the proposed model, with TPR values consistently exceeding 92% across all datasets, and achieving 100% on Rope. This is particularly important for safety scenarios where missed anomalies are costly. Competing models show much weaker and less consistent anomaly sensitivity. ADPR and CFLOW perform reasonably well on some datasets but collapse on others. Several methods exhibit extreme sensitivity variance (e.g., CITN with TPR as low as 2.08% on Zipper). Fig. 5(d) shows a similar trend in G-Mean, which evaluates the balanced contribution of sensitivity and specificity. The proposed model again achieves the strongest and most balanced performance, maintaining G-Mean above 91% in all datasets. This metric is relevant in anomaly detection because anomalies are typically rare. The large performance gaps between the proposed approach and baselines (e.g., PaDiM, UTRAD, MGAD) in OCT2017, PneumoniaMNIST, and Metal Nut show that existing SOTA models struggle to maintain high sensitivity without sacrificing specificity.

Many existing methods exhibit limited reproducibility and require multiple runs to achieve stable results due to the reliance on stochastic training procedures. Among the 11 SOTA models, only PaDiM, PatchCore, and MPDE consistently produce the same results over multiple runs. PatchCore performs well in industrial anomaly detection due to its ability to extract localized structural defects. However, it struggles with medical datasets where anomalies lack distinct structural deformations and often present as subtle intensity variations. Similarly, hybrid generative models such as CFLOW and CS-Flow demonstrate inconsistent results due to their dependence on latent space distributions. These methods work well when anomalies have well-defined distributions in latent space but struggle when anomalies overlap with normal variations, as seen in PneumoniaMNIST and UCSD. Threshold-dependent models like PaDiM and UTRAD demonstrate strong performance in structured datasets where anomalies are clearly distinct but require careful hyperparameter tuning to achieve optimal results. This sensitivity makes them highly dataset-dependent, as small variations in normal samples can cause overfitting or under-detection. MPDE and MGAD achieve competitive performance in selective cases, but fail to maintain robustness when applied to broader domains.

Fig. 6 presents a bubble graph that jointly visualizes AUROC (horizontal axis), F1-score (vertical axis), and TPR (bubble size) for all SOTA baseline models and the proposed framework across eight datasets. This tri-dimensional comparison highlights overall detection quality, decision consistency, and anomaly sensitivity in a single view. The proposed model occupies the upper-right region of the plot, forming a tight cluster of large bubbles. This indicates that it simultaneously achieves high AUROC, high F1-score, and high TPR across all datasets. The consistently large bubble sizes further confirm strong anomaly sensitivity, meaning that the model rarely misses true anomalies. This clustering pattern also reflects the robustness across domain.

In contrast, most SOTA baselines demonstrate wide dispersion throughout the plot. Several of them lie in the mid-range AUROC band with noticeably smaller bubble sizes, indicating weaker recall and inconsistent anomaly separability. If the AUROC and F1-score drop sharply, the corresponding bubble will appear as an isolated point in the lower-left region. These scattered low-performing points reflect the sensitivity of existing methods to dataset complexity, texture flexibility, or the presentation of subtle anomalies. Traditional industrial anomaly detectors such as PatchCore, MPDE, and UTRAD cluster in the upper-mid region, but still exhibit irregular bubble sizes, showing that they

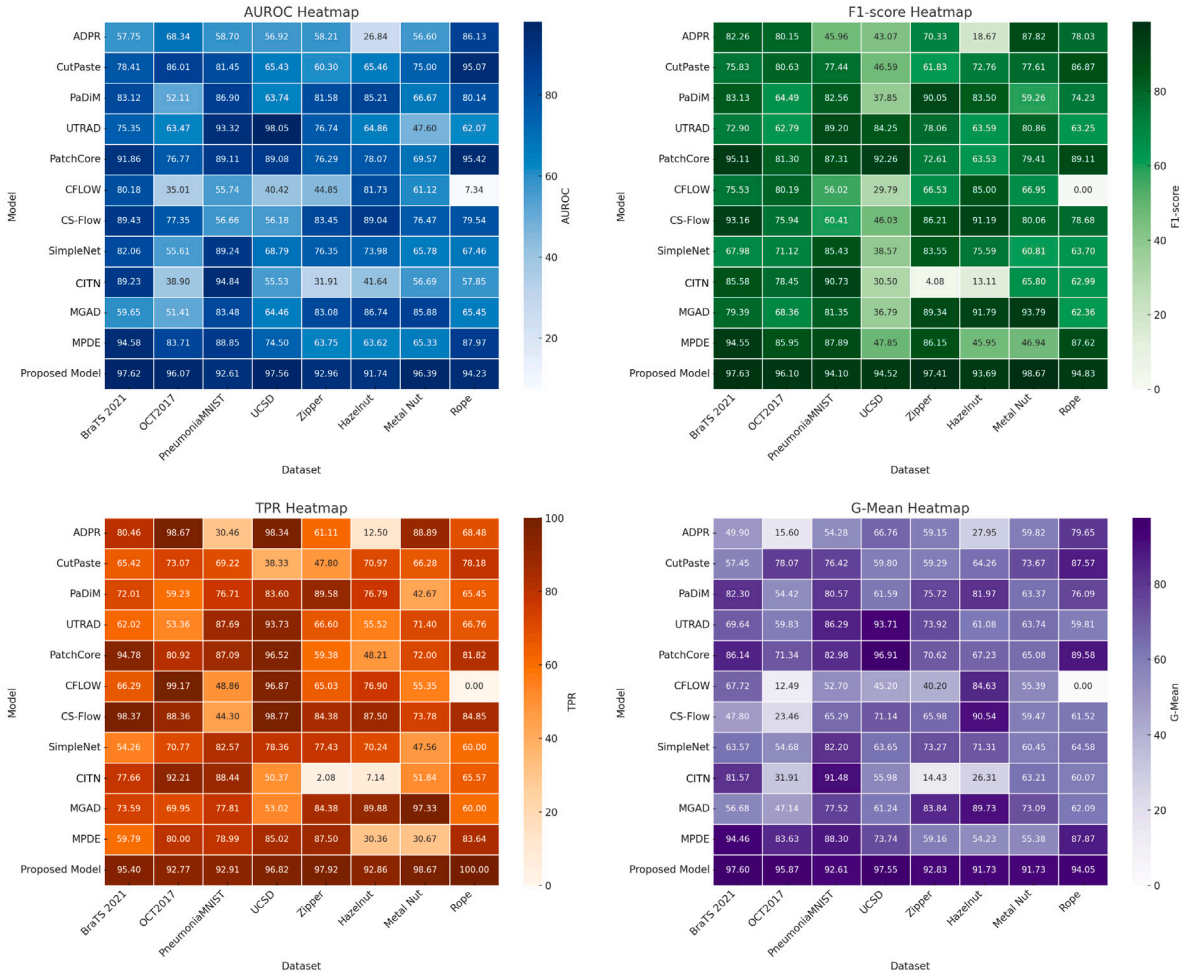

Fig. 5. Performance evaluation heatmaps of the proposed model across different datasets. (**a**) AUROC heatmap visualizing the model's ability to distinguish between normal and anomalous samples. (**b**) F1-score heatmap showing the balance between precision and recall. (**c**) TPR heatmap illustrating the model's true positive rate performance. (**d**) G-Mean heatmap measuring the geometric mean of sensitivity and specificity.

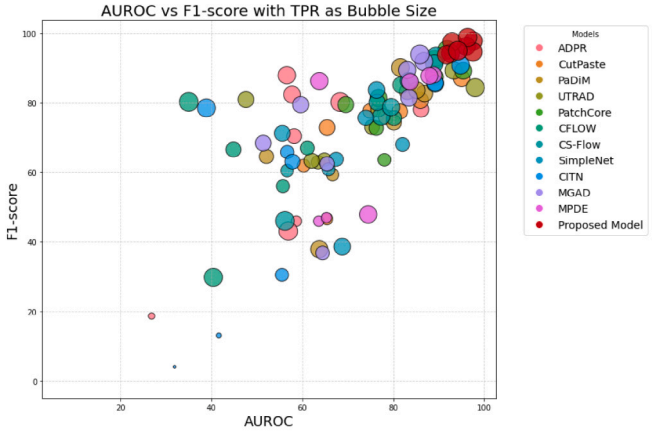

Fig. 6. Bubble graph of AUROC vs. F1-score across models and datasets with TPR represented as bubble size.

struggle to balance precision and recall consistently across domains. Although several of these models reach competitive AUROC values on selected datasets, the variability in bubble size reveals recall instability and susceptibility to false negatives. Overall, the bubble chart demonstrates that the proposed dual autoencoder framework not only achieves the highest separability and decision quality, but also provides the most stable recall pattern across diverse anomaly categories.

5.2. *Justification for the dual encoder design*

To justify the architectural design of the proposed framework, we performed a controlled comparison against two simplified variants: (i) an image-encoder-only model that captures global structural information and (ii) a pixel-encoder-only model that focuses on localized texture patterns. The full performance of the proposed dual autoencoder model has been reported in Table 6, but its results are reproduced in Table 8 for completeness, allowing for a direct side-by-side comparison with the single-encoder baselines across all eight datasets.

Table 8 shows that the image-only and pixel-only encoders only succeed on certain datasets but fail to generalize between domains. The image encoder performs well when anomalies reflect large structural deviations (e.g., BraTS2021 and Metal Nut), but it often misses subtle or fine-grained defects, leading to major drops on Hazelnut and UCSD. In contrast, the pixel encoder is highly sensitive to local texture changes and achieves strong results on PneumoniaMNIST, Metal Nut, and Rope, but its lack of global context causes frequent over-activation on datasets with natural background variability, such as OCT2017 and Hazelnut.

The proposed dual autoencoder combines these complementary strengths by jointly capturing global structure and local detail. As a result, it delivers the most consistent performance in all eight datasets,

Table 8
Comparison of single-encoder variants and the proposed dual-encoder model across eight datasets (all metrics are expressed in percentage values).

Model variant	BraTS2021	OCT2017	PneumoniaMNIST	UCSD	Zipper	Hazelnut	Metal nut	Rope
AUROC								
Image encoder only	97.52	91.68	92.43	72.35	86.79	66.74	**96.39**	94.39
Pixel encoder only	96.09	86.67	**93.40**	74.41	88.35	71.88	95.06	**95.19**
Dual encoder (Proposed)	**97.62**	**96.07**	92.61	**97.56**	**92.96**	**91.74**	**96.39**	94.23
F1-score								
Image encoder only	97.46	93.72	94.44	46.94	92.47	72.38	**98.67**	94.55
Pixel encoder only	96.00	90.64	**95.36**	47.18	94.18	65.12	97.30	**95.65**
Dual encoder (Proposed)	**97.63**	**96.10**	94.10	**94.52**	**97.41**	**93.69**	**98.67**	94.83
TPR								
Image encoder only	95.04	91.85	94.68	75.94	89.58	67.86	**98.67**	94.55
Pixel encoder only	92.33	89.38	**96.20**	87.87	92.71	50.00	96.00	**100.00**
Dual encoder (Proposed)	**95.40**	**92.77**	92.91	**96.82**	**97.92**	**92.86**	**98.67**	**100.00**

achieving the highest AUROC and F1-scores in the majority cases. This cross-domain stability indicates that both global and local representations are essential for reliable anomaly detection and that neither single pathway is sufficient on its own. Therefore, dual encoder fusion provides a balanced and robust feature representation that effectively generalized across diverse anomaly types.

5.3. Ablation study on adaptive training components

Following standard practice in anomaly detection research, ablation experiments are conducted on a single representative dataset, BraTS2021. This dataset provides stable reconstruction behavior, clear class separation, and balanced structural complexity, making it suitable for isolating the contribution of each adaptive module. For each ablation variant, one adaptive component is disabled and replaced with a non-adaptive fixed alternative. The following design choices are aligned with common practice in anomaly detection and ensure that comparisons remain controlled, reproducible, and scientifically fair.

No dynamic seed. When dynamic seed initialization is removed, a single fixed seed is required to avoid uncontrolled randomness. We used seed = 42, a widely adopted default in deep learning libraries and prior literature [70]. This ensures a stable starting point for optimization while removing the dataset-dependent seed adaptation of the full model.

No batch size estimation. Replacing the batch size estimator with a fixed value requires a batch size that is realistic for high-resolution medical images and commonly used in prior work. We select batch size = 128, which fits comfortably in GPU memory for BraTS2021 [15].

No epoch optimization. Validation-based adaptive epoch selection is disabled and the model is trained for a fixed 500 epochs. This value corresponds to the typical convergence region observed in the full model and prevents unfair under- or over-training.

No dropout tuning. Dropout tuning is removed and replaced with a fixed dropout rate of 0.2, a balanced regularization level commonly used in autoencoder architectures [18].

No adaptive thresholding. Adaptive ROC-based thresholding is replaced with a simple quantile-based rule. The anomaly threshold is set to the 95th percentile of the reconstruction error distribution on normal validation samples. This non-adaptive threshold is consistent with established unsupervised anomaly detection practices [71].

Table 9 summarizes the contribution of each adaptive training component on BraTS2021. The results make it clear that not all components contribute equally. Removing the dynamic seed initialization reduces AUROC by 3.39%, indicating the model becomes less stable over runs. Although the performance is still acceptable, this drop shows that dynamic seeding helps produce more consistent and reproducible training.

Table 9
Ablation study on BraTS2021: effect of removing each adaptive component (all metrics are expressed in percentage values).

Ablated component	AUROC	F1	G-Mean
Full model (baseline)	97.62	97.63	97.60
No dynamic seed	94.23	94.02	94.07
No batch size estimation	96.65	96.66	96.60
No epoch optimization	97.62	97.63	97.60
No dropout tuning	53.95	71.63	53.33
No adaptive thresholding	84.09	93.45	83.34

Batch size estimation produces only a minor reduction in AUROC (−0.97%), suggesting that heuristic batch selection improves gradient smoothness but is not strictly necessary for BraTS2021. The model therefore remains stable even when the batch size is fixed. Similarly, removing epoch optimization results in almost no change. The validation-driven training loop already prevents overfitting, so this component is less important for this dataset.

The most impactful component is dropout tuning. Removing dropout selection causes AUROC to collapse from 97.62% to 53.95% and reduces G-Mean to 53.33%. This reflects extreme overfitting and a loss of generalizable feature structure, especially because BraTS2021 contains high-dimensional anatomical patterns. An appropriately tuned dropout rate is essential to prevent the encoders from learning overly specific patterns that do not generalize.

Adaptive thresholding also plays an important role. Without it, AUROC falls to 84.09% and G-Mean to 83.34%, although the F1-score stays relatively high. This mismatch shows that using a fixed threshold does not align well with the reconstruction-error distribution, reducing the separation between normal and anomalous samples. Adaptive threshold selection therefore improves calibration and leads to more reliable decisions.

5.4. Hyperparameter sensitivity analysis

Although the ablation study isolates the contribution of each adaptive training component, it does not reveal how sensitive the model is to continuous variations in individual hyperparameters. We therefore perform a hyperparameter sensitivity analysis in which each parameter is systematically varied while the full adaptive pipeline is unchanged. This analysis evaluates the robustness, stability range, and degradation patterns induced by extreme configurations. Following common practice, two representative datasets are selected: (i) BraTS2021, which exhibits high-contrast and structurally consistent anomalies. (ii)Hazelnut, which contains small, low-contrast industrial defects. These two datasets span opposite ends of the anomaly spectrum, enabling us to assess sensitivity under stable medical and challenging industrial conditions.

The seed sensitivity results in Fig. 7 show that the model performance is generally stable across different random initializations, but

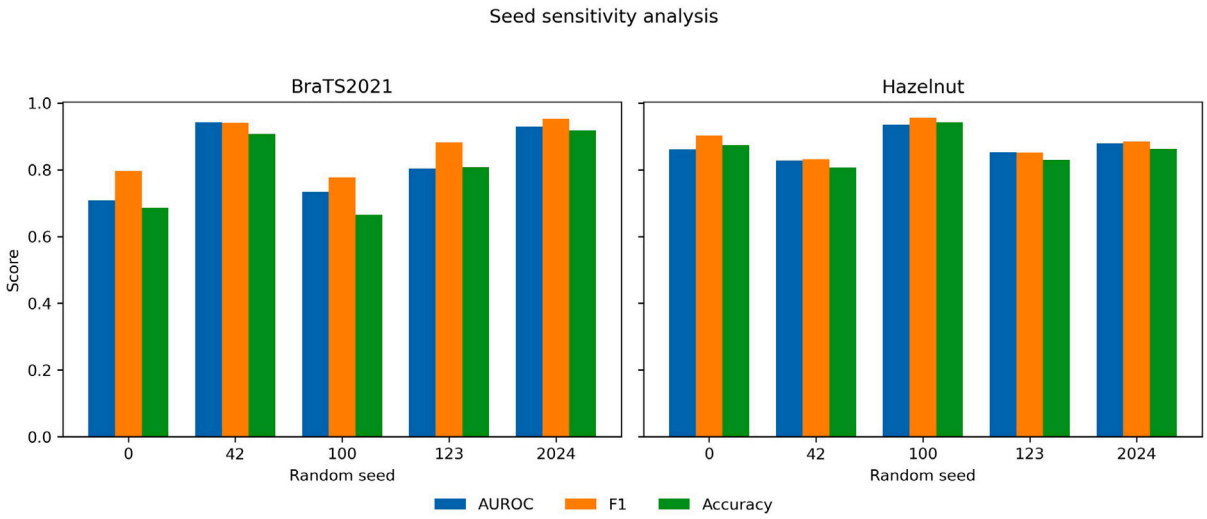

Fig. 7. Seed sensitivity analysis on BraTS2021 and Hazelnut.

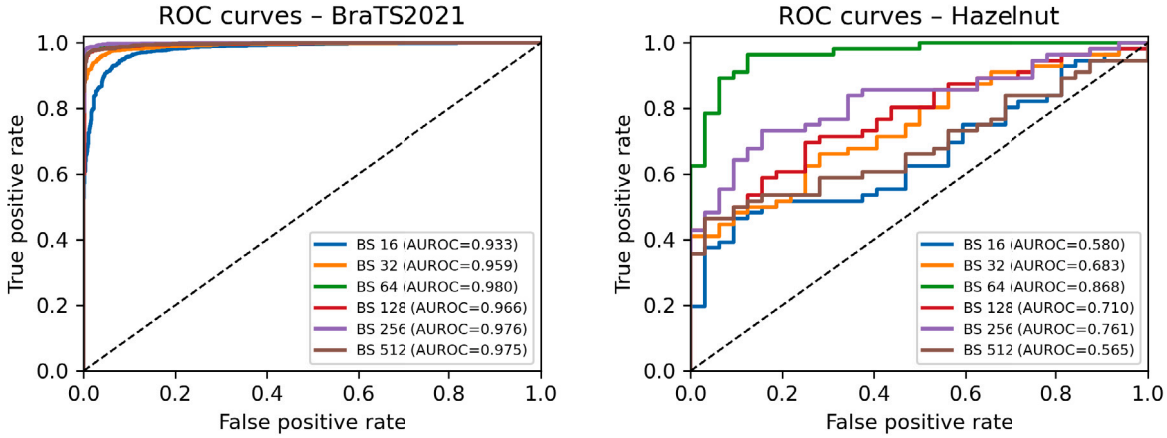

Fig. 8. Effect of batch size on AUROC.

the variation depends on the dataset. For BraTS2021, performance fluctuates between seeds, where seeds 0 and 100 yield lower scores, and seed 42 provides the best results. This suggests that the initial seed affects how well the model learns the complex structural patterns in brain MRI data. In contrast, Hazelnut shows only minor differences between seeds. Its simpler and more homogeneous textures reduce the impact of initialization, leading to more consistent convergence. Overall, the results confirm that the model is robust to random seeds, but datasets with higher structural variability are more sensitive to initialization. This reinforces the importance of using adaptive seeding strategies within the full framework.

The batch size sensitivity analysis in Fig. 8 shows how different batch configurations influence the discriminative behavior of the model by examining the ROC curves and their values. The results reveal a clear contrast between the two datasets: BraTS2021 is highly stable across batch sizes, while Hazelnut is sensitive and exhibits greater fluctuation in anomaly separability.

For BraTS2021, all curves cluster tightly near the upper-left corner, and the AUROC values remain uniformly high (0.933–0.980), indicating that different batch configurations do not materially alter the model's ability to separate normal and anomalous MRI. In contrast, Hazelnut exhibits wider variation, with smaller batches (16–32) producing flatter curves and lower AUROC values, while mid-range batches such as 64 and 256 yield stronger separability. Extremely large batches (512) again reduce performance, suggesting oversmoothing of learning updates. Therefore, the results show that the batch size has minimal influence on a structurally rich dataset but plays a substantial role in texture-based datasets, where moderate batch sizes strike the best balance between gradient noise and convergence stability.

The epoch sensitivity line plots in Fig. 9 show different training patterns for the two datasets. In BraTS2021, all metrics remain high throughout the range, reaching above 90% after 200 epochs. After this point, the gains are small, suggesting that the model learns the important tumor structures early and does not benefit much from a longer training. Hazelnut shows a different trend. Its performance varies at low epochs but improves steadily as training continues, with all metrics rising up to 1000 epochs. This indicates that the texture-based nature of the dataset requires longer training to avoid underfitting. The contrast between the two datasets highlights a key property of the framework: medical images with strong structural cues allow faster convergence, while fine-grained industrial textures require more extended optimization to achieve stable performance. Overall, the results show that BraTS2021 is less sensitive to epoch choice, whereas Hazelnut performs better with longer training.

The dropout sensitivity heatmaps in Fig. 10 show that both datasets benefit from moderate regularization with different degrees of sensitivity. For BraTS2021, performance is the best at a dropout rate of 0.1, where AUROC, F1, Accuracy, and TNR all exceed 96%. Performance

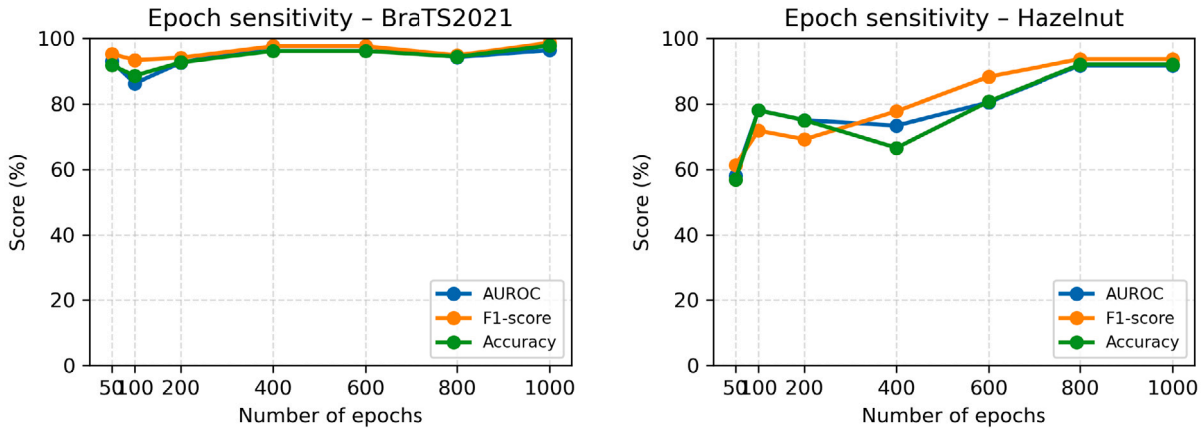

Fig. 9. Epoch sensitivity analysis showing AUROC, F1-score, and Accuracy across different training durations for BraTS2021 and Hazelnut.

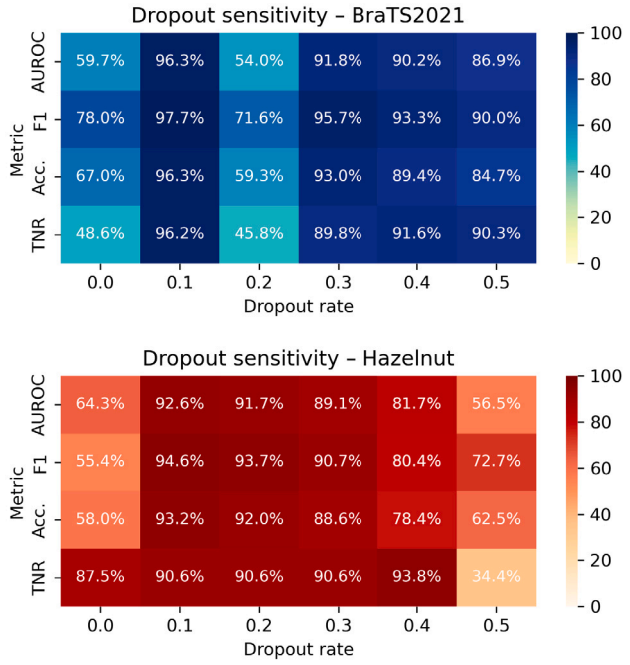

Fig. 10. Dropout sensitivity for BraTS2021 and Hazelnut, showing AUROC, F1-score, Accuracy, and TNR for dropout rates of 0.0–0.5.

drops sharply at 0.0 and 0.2, with AUROC and TNR falling to about 55%–60%. At higher dropout levels (0.3–0.5), the model becomes more stable again and most performance metrics improve. On the other hand, Hazelnut shows a similar performance peak in the 0.1–0.3 range but deteriorates more noticeably at the extreme settings. A dropout rate of 0.0 yields only moderate scores, and a rate of 0.5 results in a pronounced decline across key metrics, particularly TNR. This demonstrates that Hazelnut is more sensitive to deviations from the optimal regularization level and requires tighter control of the dropout parameter.

5.5. Uncertainty estimation via Monte Carlo dropout

Monte Carlo (MC) dropout is applied during inference to quantify predictive uncertainty and evaluate how it supports reliable anomaly detection. The model generates T stochastic reconstructions under active dropout sampling. This enables the estimation of uncertainty in the learned representation through the variance of pixel-wise reconstruction errors and provides additional evidence to distinguish ambiguous normal samples from true anomalies. Across all evaluated datasets, anomalous regions consistently show a higher reconstruction-error variance than normal regions. This pattern aligns with the idea that the model becomes less certain in areas where the latent representation deviates from the training data. On the other hand, normal samples produce reconstructions that are close to each other with only minimal variation across repeated passes.

Table 10
Summary of calibration and false-positive behavior for the deterministic and MC Dropout variants (averaged across datasets).

Metric	Deterministic	MC Dropout
FPR	7.10%	6.16%
ECE	0.17	0.14
Variance (Normal)	Low	Low
Variance (Anomaly)	–	High

We observe that many false positives from the deterministic model come from samples with slightly high reconstruction error but very low reconstruction variance. MC dropout helps reclassify these cases as confident normal samples, allowing the combined score (reconstruction error plus uncertainty weighting) to reduce false positives. We then analyzed calibration metrics such as the expected calibration error (ECE) and the confidence distribution between samples to evaluate the reliability of the uncertainty estimates.

Table 10 summarizes the effect of MC dropout on false positives and calibration behavior. On average, the deterministic model achieves a false positive rate of 7.10% on the datasets, while the MC Dropout variant further reduces this rate and improves calibration, as reflected by lower ECE and a clearer separation in variance between normal and anomalous samples. Deterministic reconstructions tend to be overconfident and give very low variance even for normal samples that are unusual or difficult to reconstruct. MC dropout adds a small amount of randomness during inference, which helps spread the confidence scores and reduces this overconfidence. The anomalies form a clear group with high uncertainty, while normal samples remain grouped in low-variance regions. This separation makes the scores easier to interpret and leads to more stable thresholding.

5.6. Anomaly localization & visual explanation

Fig. 11 presents qualitative localization results in four representative datasets. In BraTS2021, the reconstruction error highlights tumor regions with high spatial precision, capturing both the core lesion and its irregular boundaries. For Hazelnut, the heatmaps emphasize

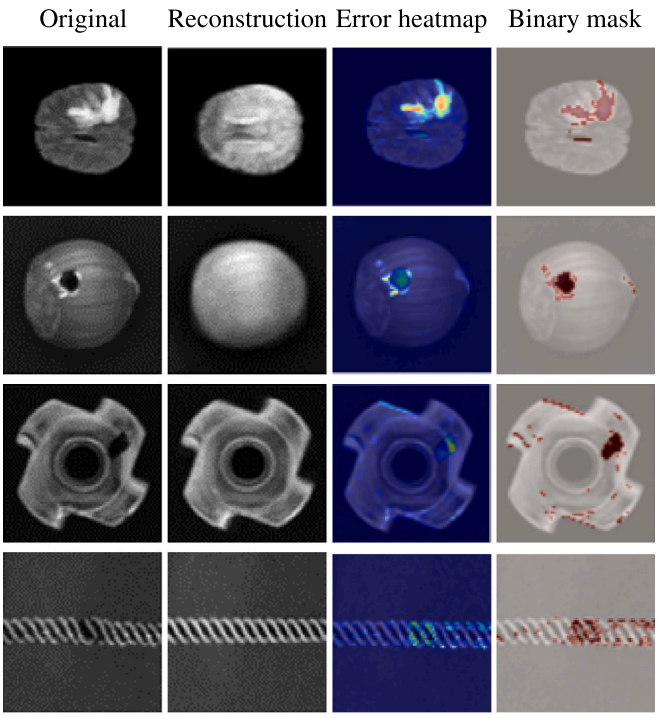

Fig. 11. Anomaly localization across four representative datasets. The proposed framework consistently localizes structural deviations across distinct anomaly types, including medical lesions (BraTS2021), texture irregularities (Hazelnut), mechanical defects (Metal Nut), and pattern disruptions (Rope).

subtle texture inconsistencies that are easily overlooked in the raw image. Metal Nut defects produce sharp localized activations along distorted edges and surface irregularities. Rope samples, which exhibit periodic structural patterns, generate elongated heat responses corresponding to frayed or misaligned strands. Across all cases, the binary anomaly masks remain compact and structurally coherent, confirming that the model consistently localizes true abnormal regions without introducing spurious activations. These observations confirm that the dual autoencoder framework can localize anomalies in various types of datasets.

5.7. Error analysis

5.7.1. False positives case study

The primary source of false positives arises from high-variance normal frames or texture regions that inflate the reconstruction error without representing the true pathology or defect. PneumoniaMNIST contributes the highest false positive proportion (7.69% FPR) among medical datasets, driven by diffuse lung opacity patterns that mimic infection-like noise, but belong to healthy anatomical variation. These samples generate pixel errors that are widely distributed across the image, causing threshold breaches due to overlapping texture statistics. Zipper and Rope show elevated FPR (12.00% and 11.54%), dominated by non-rigid deformation in the normal class. Sharply bent rope segments falsely detected as cracks, and minor zipper teeth misalignment (still acceptable) misinterpreted as irregular patterns. In UCSD, false positive cases arise from high-motion frames affected by motion blur or flickering illumination. These frames generate temporally sharp that are spatially noisy and can cross the anomaly threshold even though their latent representations remain within the learned normal manifold.

Fig. 12 presents representative false positive cases. The model incorrectly triggers (a) high-motion blurred frames in UCSD, (b) bent rope segments, and (c) specular reflections on zipper teeth. Although these

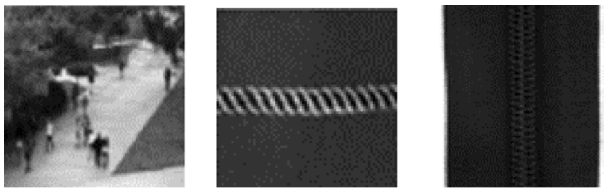

Fig. 12. Representative false positive (FP) examples triggered by reconstruction score magnitude bias.

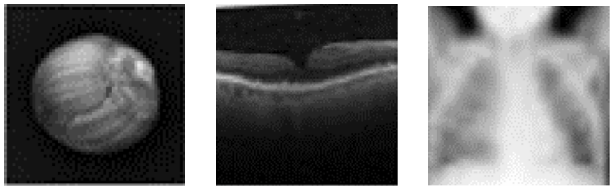

Fig. 13. Representative false negative (FN) examples where anomaly residual magnitude is insufficient to exceed threshold due to small size or low contrast.

inputs reside within the learned normal latent manifold, their reconstruction residuals exceed the threshold, producing high but spatially diffuse error activations.

5.7.2. False negatives case study

False negative misdetections are mainly observed in cases where anomaly regions are small, low-contrast, or semantically consistent with the surrounding context. This causes reconstruction errors to remain below the threshold even though anomalies are present. Hazelnut contains the clearest false negative cluster, where small surface holes or micro-defects occupy minimal pixel areas and form the lower AUROC tail in industrial inspection. In OCT2017, the few false negative cases are linked to extremely subtle retinal layer distortions or faint fluid-like silhouettes. These anomalies generate minimal decoder divergence because normal retinal thickness varies considerably across scans, weakening the reconstruction contrast between healthy and abnormal regions. PneumoniaMNIST shows a small number of false negatives when diffuse opacities mimic normal lung texture, producing low reconstruction divergence because some anomaly patterns lie near the core of the learned normal manifold."

Fig. 13 illustrates characteristic false negatives, where anomalies are not detected due to (a) extremely small surface holes in Hazelnut, (b) faint distortion of retinal layer in OCT2017, and (c) low-contrast opacity granularity in PneumoniaMNIST. These cases yield spatially concentrated errors, but their normalized energy remains below the detection threshold, causing missed detections.

6. Conclusion

This work introduced a dual autoencoder framework integrated with adaptive training components that improve stability, robustness, and interpretability in anomaly detection. The use of dynamic seed initialization, data-driven batch size estimation, epoch optimization, and dropout tuning reduces hyperparameter sensitivity and improves performance consistency across datasets. Reconstruction-error heatmaps, saliency maps, and dual-path feature analysis show that the model captures meaningful structural differences and produces well-localized anomaly maps. Overall, the results demonstrate that the method generalizes well and maintains reliable performance across domains with diverse anomaly characteristics.

Despite its strong performance, the framework still relies on reconstruction-based signals, which may struggle with anomalies that are subtle or spread across the entire image. The training time is also higher because the method uses hyperparameter tuning and repeated validation updates. In addition, although the Monte Carlo dropout improves calibration, it introduces an additional computational cost during inference. Future work will explore faster uncertainty estimation methods, hybrid reconstruction–classification models, and more advanced generative approaches that can capture fine-grained texture anomalies. Extending the framework to multimodal inputs and continuous anomaly scoring, as well as incorporating contrastive learning objectives, may further improve its robustness in real-world applications.

CRediT authorship contribution statement

Attapon Pillai: Writing – review & editing, Validation. **Nur Rusyidah Azri:** Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation. **Putsadee Pornphol:** Writing – review & editing, Validation. **Saratha Sathasivam:** Writing – review & editing, Supervision, Project administration, Methodology, Conceptualization. **Akarachai Inthanil:** Writing – review & editing, Supervision.

Ethical statement

This research did not involve any studies with human participants or animals performed by any of the authors. Ethical approval was therefore not required.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by Universiti Sains Malaysia and Phuket Rajabhat University under the Visiting Scholar Program.

References

[1] G. Shao, X. Chen, X. Zeng, L. Wang, Deep learning hierarchical representation from heterogeneous flow-level communication data, IEEE Trans. Inf. Forensics Secur. 15 (2019) 1525–1540.
[2] A.A. Abdullah, M.M. Hassan, Y.T. Mustafa, A review on bayesian deep learning in healthcare: Applications and challenges, IEEe Access 10 (2022) 36538–36562.
[3] P. Dixit, S. Silakari, Deep learning algorithms for cybersecurity applications: A technological and status review, Comput. Sci. Rev. 39 (2021) 100317.
[4] B. Maschler, M. Weyrich, Deep transfer learning for industrial automation: A review and discussion of new techniques for data-driven machine learning, IEEE Ind. Electron. Mag. 15 (2) (2021) 65–75.
[5] Z. Cheng, S. Wang, P. Zhang, S. Wang, X. Liu, E. Zhu, Improved autoencoder for unsupervised anomaly detection, Int. J. Intell. Syst. 36 (12) (2021) 7103–7125.
[6] H. Stephani, T. Weibel, R. Rösch, A. Moghiseh, Challenges and approaches when realizing online surface inspection systems with deep learning algorithms, Discov. Data 1 (1) (2023) 3.
[7] R. Touati, M. Mignotte, M. Dahmane, Anomaly feature learning for unsupervised change detection in heterogeneous images: A deep sparse residual model, IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 13 (2020) 588–600.
[8] J. Åkesson, J. Töger, E. Heiberg, Random effects during training: Implications for deep learning-based medical image segmentation, Comput. Biol. Med. 180 (2024) 108944.
[9] J.-S. Hwang, S.-S. Lee, J.-W. Gil, C.-K. Lee, Determination of optimal batch size of deep learning models with time series data, Sustainability 16 (14) (2024) 5936.
[10] T. Walczyna, D. Jankowski, Z. Piotrowski, Enhancing anomaly detection through latent space manipulation in autoencoders: A comparative analysis, Appl. Sci. 15 (1) (2024) 286.
[11] K.-D. Lu, J.-C. Huang, G.-Q. Zeng, M.-R. Chen, G.-G. Geng, J. Weng, Multi-objective discrete extremal optimization of variable-length blocks-based CNN by joint NAS and HPO for intrusion detection in IIoT, IEEE Trans. Dependable Secur. Comput. (2025).
[12] M. Pietroń, D. Żurek, K. Faber, R. Corizzo, AD-NEv: A scalable multilevel neuroevolution framework for multivariate anomaly detection, IEEE Trans. Neural Networks Learn. Syst. (2024).
[13] M. Pietroń, D. Żurek, K. Faber, A. Wójcik, R. Corizzo, AD-NEv++-The multi-architecture neuroevolution-based multivariate anomaly detection framework, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2024, pp. 607–610.
[14] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, P. Gehler, Towards total recall in industrial anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14318–14328.
[15] N.R. Azri, S. Sathasivam, M.K.M. Ali, Multi-phase dual-encoder model for anomaly detection in medical imaging, J. Qual. Meas. Anal. 21 (1) (2025) 267–285.
[16] G. Pang, C. Shen, L. Cao, A.V.D. Hengel, Deep learning for anomaly detection: A review, ACM Comput. Surv. 54 (2) (2021) 1–38.
[17] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 52–59.
[18] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.
[19] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 1096–1103.
[20] D. Gong, L. Liu, V. Le, B. Saha, M.R. Mansour, S. Venkatesh, A.v.d. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1705–1714.
[21] C.-L. Li, K. Sohn, J. Yoon, T. Pfister, Cutpaste: Self-supervised learning for anomaly detection and localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9664–9674.
[22] A. Mezina, R. Burget, C.M. Travieso-González, Network anomaly detection with temporal convolutional network and U-net model, IEEE Access 9 (2021) 143608–143622.
[23] T. Defard, A. Setkov, A. Loesch, R. Audigier, Padim: a patch distribution modeling framework for anomaly detection and localization, in: International Conference on Pattern Recognition, 2021, pp. 475–489.
[24] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, L. Wu, Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows, 2021, arXiv preprint arXiv:2111.07677.
[25] D. Gudovskiy, S. Ishizaka, K. Kozuka, Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 98–107.
[26] X. Bian, X. Luo, C. Wang, W. Liu, X. Lin, DDA-net: Unsupervised cross-modality medical image segmentation via dual domain adaptation, Comput. Methods Programs Biomed. 213 (2022) 106531.
[27] X. Cao, H. Yu, K. Yan, R. Cui, J. Guo, X. Li, X. Xing, T. Huang, DEMF-Net: A dual encoder multi-scale feature fusion network for polyp segmentation, Biomed. Signal Process. Control. 96 (2024) 106487.
[28] C. Zhu, R. Zhang, Y. Xiao, B. Zou, X. Chai, Z. Yang, R. Hu, X. Duan, DCFNet: An effective dual-branch cross-attention fusion network for medical image segmentation, Comput. Model. Eng. Sci. (CMES) 140 (1) (2024).
[29] J. Liu, K. Song, M. Feng, Y. Yan, Z. Tu, L. Zhu, Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection, Opt. Lasers Eng. 136 (2021) 106324.
[30] S. Dutta, A. Arunachalam, S. Misailovic, To seed or not to seed? an empirical analysis of usage of seeds for testing in machine learning projects, in: 2022 IEEE Conference on Software Testing, Verification and Validation, ICST, 2022, pp. 151–161.
[31] S. Banerjee, T. Marrinan, R. Cannon, T. Chiang, A.D. Sarwate, Measuring model variability using robust non-parametric testing, 2024, pp. 1–28, arXiv preprint arXiv:2406.08307.
[32] S. Bethard, We need to talk about random seeds, 2022, pp. 1–6, arXiv preprint arXiv:2210.13393.
[33] B. Nadler, et al., Anomalib: A deep learning library for anomaly detection, 2022, https://github.com/openvinotoolkit/anomalib.
[34] F. Sun, J. Zhang, X. Wu, Z. Zheng, X. Yang, Video anomaly detection based on global–local convolutional autoencoder, Electronics 13 (22) (2024) 4415.
[35] S. Afaq, S. Rao, Significance of epochs on training a neural network, Int. J. Sci. Technol. Res 9 (06) (2020) 485–488.
[36] C.I. Shimabukuro, et al., Deep learning applied to stock prices: Epoch adjustment in training an LSTM neural network, Int. J. Bus. Manag. 19 (4) (2024) 1–80.
[37] A. Olmin, F. Lindsten, Towards understanding epoch-wise double descent in two-layer linear neural networks, 2024, pp. 1–48, arXiv preprint arXiv:2407.09845.

[38] R. Egele, F. Mohr, T. Viering, P. Balaprakash, The unreasonable effectiveness of early discarding after one epoch in neural network hyperparameter optimization, Neurocomputing 597 (2024) 127964.
[39] B.A.S. Al-Rimy, F. Saeed, M. Al-Sarem, A.M. Albarrak, S.N. Qasem, An adaptive early stopping technique for densenet169-based knee osteoarthritis detection model, Diagnostics 13 (11) (2023) 1903.
[40] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: A novel bandit-based approach to hyperparameter optimization, J. Mach. Learn. Res. 18 (185) (2018) 1–52.
[41] S. Falkner, A. Klein, F. Hutter, BOHB: Robust and efficient hyperparameter optimization at scale, in: International Conference on Machine Learning, PMLR, 2018, pp. 1437–1446.
[42] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, et al., Keras tuner, 2019.
[43] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 41–48.
[44] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, Autoaugment: Learning augmentation strategies from data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 113–123.
[45] J. Ba, B. Frey, Adaptive dropout for training deep neural networks, Adv. Neural Inf. Process. Syst. 26 (2013).
[46] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
[47] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (140) (2020) 1–67.
[48] J. Howard, S. Gugger, Fastai: a layered API for deep learning, Information 11 (2) (2020) 108.
[49] M. Du, K. Liang, L. Zhang, H. Gao, Y. Liu, Y. Xing, Deep-learning-based metal artefact reduction with unsupervised domain adaptation regularization for practical CT images, IEEE Trans. Med. Imaging 42 (8) (2023) 2133–2145.
[50] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[51] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017).
[52] R. Hesse, S. Schaub-Meyer, S. Roth, Fast axiomatic attribution for neural networks, Adv. Neural Inf. Process. Syst. 34 (2021) 19513–19524.
[53] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, The MVTec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection, Int. J. Comput. Vis. 129 (4) (2021) 1038–1059.
[54] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013, arXiv preprint arXiv:1312.6034.
[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[56] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, PMLR, 2016, pp. 1050–1059.
[57] D.V. Origines, A.M. Sison, R.P. Medina, A novel pseudo-random number generator algorithm based on entropy source epoch timestamp, in: 2019 International Conference on Information and Communications Technology, ICOIACT, 2019, pp. 50–55.
[58] J. Bao, H. Sun, H. Deng, Y. He, Z. Zhang, X. Li, Bmad: Benchmarks for medical anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4042–4053.
[59] W. Li, V. Mahadevan, N. Vasconcelos, Anomaly detection and localization in crowded scenes, IEEE Trans. Pattern Anal. Mach. Intell. 36 (1) (2014) 18–32.
[60] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, B. Ni, Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification, Sci. Data 10 (1) (2023) 41.
[61] A. Taşdelen, Enhancing green computing through energy-aware training: An early stopping perspective, Curr. Trends Comput. 2 (2) (2025) 108–139.
[62] D. Kermany, Labeled optical coherence tomography (oct) and chest x-ray images for classification, 2018.
[63] P. Bergmann, X. Jin, D. Sattlegger, C. Steger, The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization, 2021, pp. 1–12, arXiv preprint arXiv:2112.09045.
[64] P. Mishra, C. Piciarelli, G.L. Foresti, Image anomaly detection by aggregating deep pyramidal representations, in: International Conference on Pattern Recognition, 2021, pp. 705–718.
[65] L. Chen, Z. You, N. Zhang, J. Xi, X. Le, UTRAD: Anomaly detection and localization with U-transformer, Neural Netw. 147 (2022) 53–62.
[66] H. Shi, Y. Zhou, K. Yang, X. Yin, K. Wang, CSFlow: Learning optical flow via cross strip correlation for autonomous driving, in: 2022 IEEE Intelligent Vehicles Symposium, IV, 2022, pp. 1851–1858.
[67] Z. Liu, Y. Zhou, Y. Xu, Z. Wang, Simplenet: A simple network for image anomaly detection and localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 20402–20411.
[68] A. Bhushan, A.K. Singh, V.K. Dwivedi, Anomaly detection model for convolutional image transformation networks, in: 2024 3rd International Conference on Power Electronics and IoT Applications in Renewable Energy and Its Control, PARC, 2024, pp. 129–134.
[69] R. Wei, Z. Li, L. Geng, M. Wuken, Y. Liu, Industrial image anomaly detection based on multi Gaussian discriminant model and robust core set, Meas. Sci. Technol. 35 (11) (2024) 116009.
[70] D. Picard, Torch. manual_seed (3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision, 2021, arXiv preprint arXiv:2109.08203.
[71] N. Pinon, C. Lartizien, OCSVM-guided representation learning for unsupervised anomaly detection, 2025, arXiv preprint arXiv:2507.21164.