UNSUPERVISED ANOMALY DETECTION FOR INDUSTRIAL VISUAL INSPECTION: IMPLEMENTATION AND EVALUATION

Nabayan Saha
4th Year Undergraduate
Indian Institute of Technology Kharagpur,
Kharagpur, India
nabayansaha@kgpian.iitkgp.ac.in

ABSTRACT

Industrial inspection systems increasingly rely on unsupervised visual anomaly detection to identify defects in manufacturing components, where annotated anomalous data is scarce or unavailable. In such contexts, synthesizing anomalies becomes critical to training robust detectors. In this paper, we implement and study two recent frameworks—GLASS (Global and Local Anomaly co-Synthesis Strategy) and SimpleNet—that approach anomaly synthesis and detection from different perspectives. GLASS leverages both feature-space and image-space synthesis guided by gradient ascent, while SimpleNet adopts Gaussian-based feature perturbations to generate synthetic anomalies efficiently. We reimplement both methods from scratch, adapt them to the MVTec AD [3] dataset, and validate their performance through a unified training pipeline. We evaluate the models using standard metrics such as image-level and pixel-level AUROC, and visualize their detection capabilities on texture and object categories. Our experiments confirm the effectiveness of gradient-guided synthesis in capturing weak anomalies and highlight key trade-offs between architectural complexity and inference efficiency. We also discuss implementation challenges.

***K*eywords** Visual Anomaly Detection · Feature Space Perturbation · Gradient Ascent · Unsupervised Learning

1 Introduction

Visual anomaly detection has emerged as a critical component in industrial inspection systems, enabling manufacturers to automatically identify defects such as scratches, dents, and structural deformations across a wide range of components. These systems are essential for ensuring product quality, minimizing production downtime, and improving operational safety. However, supervised approaches to anomaly detection are often impractical in industrial contexts due to the high cost and scarcity of annotated defect samples. This has prompted a shift toward unsupervised methods that learn from only normal data.

The MVTec AD dataset[3] has become a standard benchmark in this domain, providing a diverse collection of industrial objects and textures with annotated anomalies. Unsupervised models trained on this dataset are expected to generalize across subtle and rare defects—especially in scenarios where anomaly appearance varies significantly across categories.

Among the recent advancements in this field, two notable methods stand out: SimpleNet[1] and GLASS [2]. SimpleNet proposes a lightweight, efficient architecture that generates synthetic anomalies in the feature space using Gaussian noise, and trains a simple discriminator to distinguish them from normal features. Its minimal design and high inference speed make it particularly attractive for real-time industrial deployment.

On the other hand, GLASS [2] (Global and Local Anomaly co-Synthesis Strategy) introduces a more elaborate pipeline that synthesizes anomalies both globally (in feature space) and locally (in image space), using gradient ascent and texture fusion techniques. This dual-synthesis approach allows GLASS [2] to generate both subtle, near-distribution anomalies and prominent defects, significantly improving performance in weak defect detection tasks.

In this work, we reimplement both SimpleNet [1] and GLASS [2] from scratch, apply them to the MVTec AD [3] dataset, and critically evaluate their performance. Our goal is to understand the practical trade-offs between simplicity and performance, verify the reproducibility of their reported results, and explore the effectiveness of anomaly synthesis strategies in industrial inspection. We also discuss challenges faced during implementation.

1.1 Types of Anomalies in Industrial Images

In industrial visual inspection tasks, anomalies typically manifest as defects that deviate from the learned pattern of normal components. The MVTec AD [3] dataset, a standard benchmark for such tasks, includes a wide range of texture-based and object-based categories, each with distinct types of anomalies. These defects can be broadly categorized into the following types:

1. **Surface Defects:**
   - Scratches (e.g., in capsule, screw, or transistor)
   - Cracks (e.g., in tiles or wood)
   - Cuts or tears (e.g., leather)
   - Contaminations (e.g., in hazelnuts or pills)
2. **Structural Anomalies:**
   - Deformations (e.g., bent parts in metal nuts or pills)
   - Missing components (e.g., missing pins in a transistor or holes in a zipper)
3. **Texture Irregularities:**
   - Pattern misalignments, irregular spacing, or blurred textures (e.g., in grid, carpet, or tile)
   - Often harder to detect due to subtle deviations from the normal appearance

These anomalies vary in scale, intensity, and position, and can often be weak, meaning they are low in contrast or occupy only a small region of the image.

1.2 Challenges in Industrial Anomaly Detection

Detecting anomalies in industrial settings is inherently difficult due to the scarcity of defective samples and the subtle nature of many real-world defects. Most systems are trained only on normal data, making it hard to generalize to unknown or weak anomalies that resemble normal variations. Additionally, industrial images often contain high intra-class variability—especially in textured surfaces like wood or fabric—further complicating the distinction between normal and abnormal. Precise localization of defects is often required, yet pixel-level annotations are rarely available. Moreover, real-time constraints and limited hardware resources in deployment environments make it challenging to use large, complex models. Finally, deep learning models used in this context often lack interpretability, which limits their adoption in critical applications where explainability and reliability are essential.

1.3 Objective for Anomaly Detection

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ be a set of training images sampled from the normal (defect-free) distribution $P_{\text{normal}}(x)$, where $x_i \in \mathbb{R}^{H \times W \times C}$ denotes an image with height H, width W, and C color channels. The goal of unsupervised anomaly detection is to learn a function $f : \mathbb{R}^{H \times W \times C} \to \mathbb{R}^{H \times W}$, such that for any test image x, the output $f(x)$ provides a pixel-level anomaly score map, where higher values indicate a higher likelihood of being anomalous.

Formally, the objective is to learn parameters θ such that:

$$f_\theta(x) \approx \begin{cases} 0 & \text{if } x \sim P_{\text{normal}} \\ > 0 & \text{if } x \sim P_{\text{anomalous}} \end{cases}$$

In models like *GLASS* and *SimpleNet [1]*, this is achieved by constructing a latent feature extractor $E_\varphi(x)$ and a feature adaptor A_ϕ, where the adapted feature map is denoted by:

$$u(x) = A_\phi(E_\varphi(x))$$

An anomaly discriminator $D_\psi : \mathbb{R}^d \to [0, 1]$ is then trained to assign a score to each feature vector $u(x)_{h,w}$, where d is the feature dimension. The training process involves minimizing a loss function $\mathcal{L}$, such as a combination of binary cross-entropy or focal loss, using synthetic anomalies (e.g., $u' = u + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$) as negative examples:

$$\min_{\phi,\psi} \; \mathcal{L}(D_\psi(u), y = 0) + \mathcal{L}(D_\psi(u + \epsilon), y = 1)$$

During inference, the image-level anomaly score $S_{\text{image}}(x)$ is computed as the maximum of the pixel-level scores:

$$S_{\text{image}}(x) = \max_{h,w} D_\psi(u(x)_{h,w})$$

And the pixel-level anomaly map $S_{\text{pixel}}(x)$ is given directly by the discriminator output:

$$S_{\text{pixel}}(x) = D_\psi(u(x))$$

2 Literature Review

Anomaly detection in industrial images is crucial for automated manufacturing, quality control, and safety. It is challenging due to the rarity and variety of defects and the typical availability of only defect-free data for training, limiting supervised methods. To address this, unsupervised and self-supervised approaches have been developed to detect deviations from normal patterns.

These methods fall into three main categories: reconstruction-based (relying on normal data reconstruction), embedding-based (modeling feature distributions from pretrained networks), and synthesis-based (generating synthetic anomalies to guide learning). Each has trade-offs in accuracy, efficiency, and handling subtle defects.

Reconstruction-based approaches[10]. rely on the assumption that models trained to reconstruct only normal samples will fail to reconstruct anomalies. Techniques like autoencoders, variational autoencoders (VAEs), and generative adversarial networks (GANs) are commonly used in this space. Anomalies are detected based on the difference between input images and their reconstructions. However, these models often suffer from generalization issues, where the network learns to reconstruct even anomalous regions, leading to false negatives. Methods like AE-SSIM and RIAD fall into this category.

Embedding-based approaches[10] extract high-level features from pretrained convolutional neural networks and compare them against learned distributions of normal features. Popular models like PatchCore, PaDiM, and CS-Flow rely on nearest-neighbor search, multivariate Gaussian modeling, or normalizing flows to represent the normal feature space. These methods achieve strong performance but often suffer from high memory requirements or inference-time complexity, especially when dealing with large datasets or high-resolution images.

Synthesis-based approaches[10] attempt to mitigate the limitations of the above methods by artificially generating anomalies. These synthetic anomalies are used to train discriminative models to better separate normal and anomalous patterns. For example, CutPaste and DRÆM apply patch-based or texture-based perturbations directly in the image space. However, such synthetic data may lack realism or fail to cover subtle defect patterns.

[1]

Type	Reconstruction-based		Synthesizing-based		Embedding-based			
Model	AF-SSIM	RIAD	DR-EM	CutPaste	CS-Flow	PaDiM	RevDist	PatchCore
Bottle	87.61.1	84.2/96.5	97.0/95.5	93.9/98.3	100/-	99.5/99.1	89.9/89.0	98.7/99.1
Grid	94/84.9	99.6/98.8	99.9/99.7	100/97.5	99.0/-	96.7/97.3	100/99.3	98.2/98.7
Leather	72.8/-	100/99.4	100/98.6	100/99.5	100/-	100/99.2	100/93.4	100/99.3
Tile	59/17.5	98.7/89.1	99.9/99.2	94.6/90.5	100/-	98.1/94.1	99.5/95.6	98.7/95.6
Wood	73.6/13	93.0/85.8	99.1/96.4	99.1/95.5	100/-	99.2/94.9	99.2/95.3	99.2/95.6
Avg. Text.	78/56.7	95.1/93.9	99.1/97.9	97.5/96.3	99.8/-	95.5/96.9	99.5/97.7	99.0/97.5
Bottle	93/83.4	99.9/98.4	99.2/99.1	98.2/97.6	99.8/-	99.1/98.3	100/98.7	100/98.6
Cable	82/47.8	81.9/84.2	91.8/94.7	81.2/90.0	99.1/-	97.1/96.7	95.0/97.4	99.5/98.4
Capsule	94/86.0	88.4/92.8	98.5/94.3	98.2/97.4	97.1/-	87.5/98.5	96.5/98.7	98.1/98.8
Hazelnut	97/91.6	83.3/96.1	100/99.7	98.3/97.3	99.6/-	94.4/98.2	99.9/98.9	100/98.7
Metal Nut	89/60.3	88.5/92.5	98.7/99.5	99.9/93.1	99.1/-	97.2/97.2	100/97.3	100/98.4
Pill	91/83.0	83.8/95.7	98.9/97.6	94.9/95.7	98.6/-	90.1/95.7	96.6/98.2	96.6/97.4
Screw	69/88.7	84.5/98.8	92.9/97.6	88.7/96.7	97.6/-	97.5/98.5	97.1/99.6	98.1/99.4
Toothbrush	92/78.4	100/98.9	100/98.1	99.4/98.1	91.9/-	100/98.8	99.5/99.1	100/98.7
Transistor	90/72.5	90.6/87.7	95.1/90.9	96.1/91.0	99.3/-	93.4/97.5	96.7/92.5	100/96.3
Zipper	88/66.5	98.1/97.8	100/98.8	99.9/99.3	99.7/-	98.6/98.5	98.5/98.2	99.4/98.8
Avg. Obj.	91/75.8	89.9/94.3	97.4/97.0	95.5/95.8	98.2/-	96.0/97.8	98.0/97.9	99.2/98.4
Average	87/69.4	91.7/94.2	98.0/97.3	96.1/96.0	98.7/-	95.8/97.5	98.5/97.8	99.1/98.1

3 Motivation for Choosing Methods:

3.1 GLASS

GLASS [2] (Global and Local Anomaly co-Synthesis Strategy) was chosen for its innovative dual-level anomaly synthesis mechanism, which addresses the shortcomings of previous anomaly generation strategies. It operates both in the feature space (GAS) and image space (LAS), allowing it to simulate a wide and realistic variety of anomalies—especially weak or subtle defects, which are often missed by conventional methods.

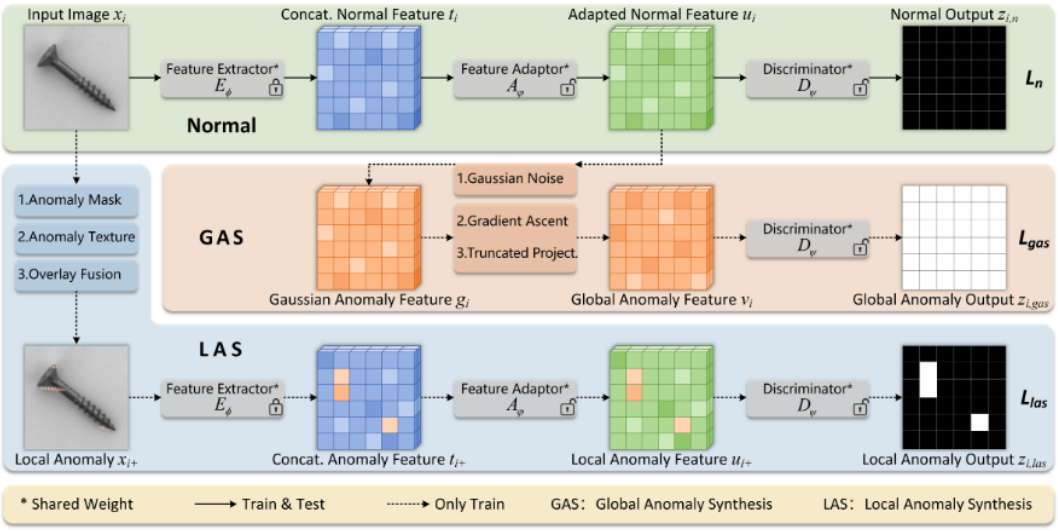

Figure 1: Schematic of the GLASS

- **Pros:**
  - **High Coverage of Anomalies:** Combines global (feature-level) and local (image-level) synthesis to simulate both near-in-distribution and far-from-distribution anomalies.
  - **Controllable Synthesis:** Uses gradient ascent with truncated projection to guide the direction of synthetic anomalies, reducing overlap with normal features.
  - **Strong Performance on Weak Defects:** Particularly effective in detecting subtle anomalies due to the GAS strategy targeting regions near the normal boundary.
  - **State-of-the-Art Accuracy:** Achieves 99.9% image-level AUROC and 99.3% pixel-level AUROC on the MVTec AD [3] dataset, outperforming most existing methods.
- **Cons:**
  - **Training Complexity:** The multi-branch design with separate modules (GAS, LAS, normal path) increases training time and memory requirements.
  - **Distribution Hypothesis Sensitivity:** Requires choosing between manifold and hypersphere hypotheses, which may involve category-specific tuning.
  - **Relatively Slower Inference:** Due to model complexity and branching, it's slower than simpler models like SimpleNet [1] in production environments.

3.2 SimpleNet

SimpleNet [1] was selected for its efficiency, simplicity, and practical deployment readiness, making it well-suited for real-time anomaly detection in industrial scenarios. Instead of generating synthetic anomalies in image space, it synthesizes anomalous features by adding Gaussian noise in feature space, making it computationally lightweight and fast. It employs a shallow discriminator and a single FC-layer adaptor, resulting in very high throughput and easy training.

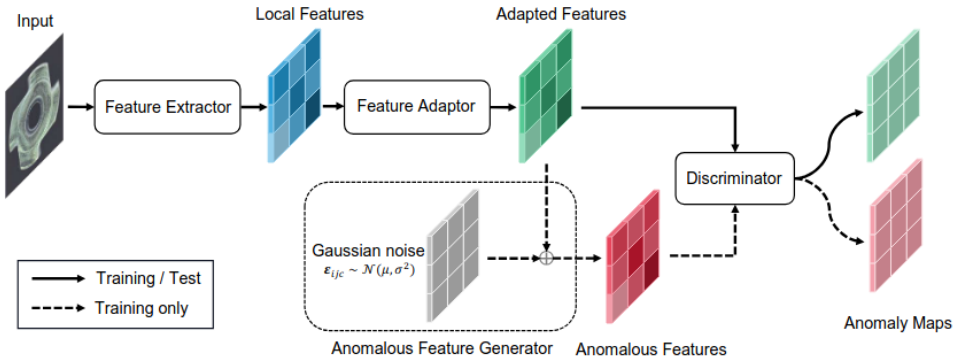

Figure 2: Schematic of the SimpleNet

- **Pros:**
  - **Simple and Lightweight Architecture:** Only four main components—Feature Extractor, Feature Adaptor, Anomaly Generator (training only), and Discriminator.
  - **Strong Accuracy:** Still delivers 99.6% AUROC on MVTec AD [3], placing it among the top-performing models.
  - **Compact and Interpretable Feature Space:** Adapted features become more separable and compact, simplifying anomaly detection.
  - **No Image-level Synthesis Required:** Avoids challenges related to generating visually plausible anomalies.
- **Cons:**
  - **Limited Anomaly Diversity:** Since it only adds isotropic Gaussian noise in feature space, it may not simulate complex or localized anomalies well.
  - **No Localization-Specific Module:** Unlike GLASS [2], which has explicit localization capability via segmentation, SimpleNet [1] relies on the discriminator's response.
  - **Feature Quality Dependence:** Performance depends heavily on the quality of pre-trained features and the effectiveness of the simple adaptor.

4 Impact on Research World

The integration and comparative evaluation of GLASS [2] and SimpleNet [1] in this work offer meaningful contributions to the ongoing evolution of industrial anomaly detection.

- **Bridging Performance and Practicality:** By presenting GLASS [2]—an advanced synthesis-based method with state-of-the-art accuracy—and contrasting it with SimpleNet [1]—a lightweight, deployment-ready model—this research highlights the trade-offs between model complexity, detection strength, and real-world applicability. This dual perspective encourages future research to not only focus on AUROC metrics but also on speed, robustness, and interpretability, which are essential for industrial adoption.
- **Advancing Anomaly Synthesis Techniques:** GLASS [2] introduces a novel gradient-guided synthesis strategy that pushes the frontier of controllable and diverse anomaly generation, especially for weak defects. This has inspired deeper exploration into feature-level perturbation frameworks and distribution-aware synthesis, opening new directions for designing training paradigms without needing real defect data.
- **Promoting Minimalist Architectures:** SimpleNet [1] sets a benchmark for how effective anomaly detection can be achieved with minimal architecture, challenging the norm of over-complicated designs. Its success emphasizes that domain adaptation and feature space manipulation can rival or even surpass heavier statistical and generative models, thereby encouraging a shift towards computationally sustainable anomaly detection.
- **Dataset Benchmarking and Evaluation Standards:** By evaluating in MVTec AD [3] dataset, this work contributes to the standardization of comprehensive benchmarking practices, including both pixel-level and image-level performance, as well as real-time inference analysis. This supports the community's efforts in developing more generalizable and robust detection frameworks.

5 Implementation and Methodology Details:

5.1 Dataset:

The MVTec Anomaly Detection (MVTec AD [3]) dataset is a widely used benchmark for evaluating unsupervised anomaly detection methods, particularly in the context of industrial visual inspection. It is specifically designed to reflect real-world manufacturing scenarios, where defective samples are scarce, and models are typically trained only on normal (defect-free) images.

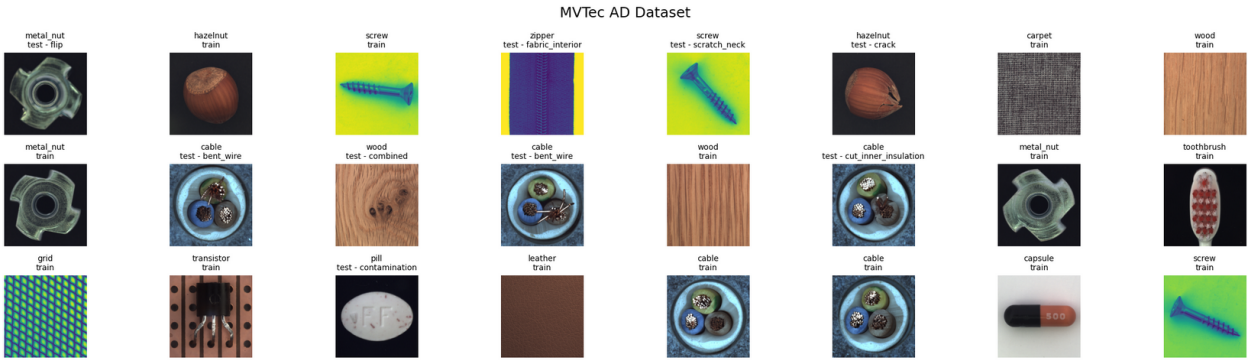

Figure 3: MVTec AD [3] DataSet

The dataset consists of 15 distinct categories, divided into 5 texture classes (e.g., carpet, grid, leather, tile, wood) and 10 object classes (e.g., bottle, cable, capsule, hazelnut, metal nut, pill, screw, toothbrush, transistor, zipper). Each category includes:

- A training set comprised exclusively of normal images, used to learn the distribution of defect-free data.
- A test set containing both normal and anomalous images, with pixel-level ground-truth masks provided for anomalous samples, enabling both image-level and pixel-level evaluation.

5.2 Experimental Setup:

All experiments were conducted using the Kaggle Notebook environment, equipped with two NVIDIA T4 GPUs (16 GB each). This setup provided sufficient computational capacity for training both lightweight and complex models while maintaining reproducibility and accessibility.

The training was carried out on the MVTec AD [3] dataset, following the standard protocol of using only normal images for training and evaluating on a mixed test set containing both normal and anomalous samples. Each image was resized and center-cropped to a resolution of 256×256 to ensure consistency and compatibility across different models.

For both GLASS [2] and SimpleNet [1], we used the WideResNet50 backbone pretrained on ImageNet, and extracted features from intermediate layers (typically levels 2 and 3). The Adam optimizer was used with separate learning rates for the feature adapter and discriminator modules:

- Feature adaptor learning rate: 0.0001
- Discriminator learning rate: 0.0002

Training Details:

- **Batch size:** 8
- **Number of epochs:** 100 (for each class)
- **Noise settings:** Gaussian noise with mean = 0 and standard deviation = 0.015 was used for synthetic anomaly generation in the feature space.
- **Image synthesis:** LAS applied random Perlin masks with DTD textures and blending coefficients sampled from a truncated normal distribution.

6 Details about GLASS:

The **GLASS [2] (Global and Local Anomaly co-Synthesis Strategy)** framework enhances anomaly detection by synthesizing diverse anomalies at both feature and image levels. It consists of two branches:

- **Global Anomaly Synthesis (GAS)**: operates in feature space to generate weak, near-in-distribution anomalies.
- **Local Anomaly Synthesis (LAS)**: operates in image space to generate strong, far-from-distribution anomalies.

6.1 Notations

Let the training set contain only normal images:

$$X_{\text{train}} = \{x_i \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^{N}$$

A pre-trained backbone ϕ extracts features from multiple layers. These are adapted by A_φ:

$$u_i = A_\varphi(E_\phi(x_i)), \quad u_i \in \mathbb{R}^{H'\times W'\times C}$$

where E_ϕ denotes the multilevel feature extractor.

6.2 Global Anomaly Synthesis (GAS)

Step 1: Gaussian Perturbation

$$g_{h,w} = u_{h,w} + \varepsilon_{h,w}, \quad \varepsilon_{h,w} \sim \mathcal{N}(0, \sigma^2 I)$$

Step 2: Gradient Ascent

$$\tilde{g}_{h,w} = g_{h,w} + \eta \cdot \frac{\nabla_{g_{h,w}} \mathcal{L}_{\text{GAS}}}{\|\nabla_{g_{h,w}} \mathcal{L}_{\text{GAS}}\|}$$

Step 3: Truncated Projection

$$\tilde{\varepsilon}_{h,w} = \tilde{g}_{h,w} - u_{h,w}$$

$$\hat{\varepsilon}_{h,w} = \begin{cases} r_1 \cdot \frac{\tilde{\varepsilon}_{h,w}}{\|\tilde{\varepsilon}_{h,w}\|}, & \text{if } \|\tilde{\varepsilon}_{h,w}\| < r_1 \\ r_2 \cdot \frac{\tilde{\varepsilon}_{h,w}}{\|\tilde{\varepsilon}_{h,w}\|}, & \text{if } \|\tilde{\varepsilon}_{h,w}\| > r_2 \\ \tilde{\varepsilon}_{h,w}, & \text{otherwise} \end{cases}$$

$$v_{h,w} = u_{h,w} + \hat{\varepsilon}_{h,w}$$

6.3 Local Anomaly Synthesis (LAS)

Step 1: Mask Generation

Let m_1, m_2 be Perlin noise masks and m_f be the foreground mask:

$$m = \begin{cases} (m_1 \wedge m_2) \wedge m_f, & 0 \le p_m \le \alpha \\ (m_1 \vee m_2) \wedge m_f, & \alpha < p_m \le 2\alpha \\ m_1 \wedge m_f, & 2\alpha < p_m \le 1 \end{cases} \quad p_m \sim \mathcal{U}(0,1),\ \alpha = \frac{1}{3}$$

Step 2: Texture Synthesis

Select a random texture x_i' and apply random augmentations $\mathcal{T}$:

$$x_i'' = \mathcal{T}(x_i')$$

Step 3: Overlay Fusion

$$x_i^+ = x_i \odot \bar{m} + (1-\beta) x_i'' \odot m + \beta x_i \odot m, \quad \beta \sim \mathcal{N}(0.5, 0.12)$$

6.4 Discriminator and Loss Functions

A shared discriminator D_ψ is trained using the following losses:

Normal Branch

$$\mathcal{L}_n = \sum_{x_i} \text{BCE}(D_\psi(u_i), 0)$$

GAS Branch

$$\mathcal{L}_{\text{GAS}} = \sum_{x_i} \text{BCE}(D_\psi(v_i), 1)$$

LAS Branch

$$\mathcal{L}_{\text{LAS}} = \text{Focal}(D_\psi(u_i^+), m_i)$$

Total Loss

$$\mathcal{L} = \mathcal{L}_n + \mathcal{L}_{\text{GAS}} + \text{OHEM}(\mathcal{L}_{\text{LAS}})$$

6.5 Inference and Scoring

During testing, only the normal branch is used:

$$S_{AL}(x_i) = \text{Smooth}(\text{Upsample}(D_\psi(u_i))) \quad \text{(Localization Score)}$$
$$S_{AD}(x_i) = \max_{h,w} D_\psi(u_{i,h,w}) \quad \text{(Detection Score)}$$

Algorithm 1 GAS under Manifold Hypothesis

Input: normal feature map u_i, number of batch n_{batch}, number of iteration n_{step}, interval of projection n_{proj}
Output: global anomaly feature map v_i
for batch = 1 **to** n_{batch} **do**
  Initialize u_i by E_ϕ and A_φ
  Gaussian noise. Add ε_i to $u_i \to g_i$
  for step = 1 **to** n_{step} **do**
    Gradient ascent.
    (a) Calculate the loss L_{gas} of GAS branch by g_i
    (b) Update g_i according to Eq. 1 with no grad
    if step is a multiple of n_{proj} **then**
      Truncated projection.
      (c) Get gradient ascent distance $\tilde{\varepsilon}_i = g_i - u_i$
      (d) Constrain the range by Eq. 2 to get truncated distance $\tilde{\varepsilon}_i \to \hat{\varepsilon}_i$
      (e) Get GAS feature $v_i = u_i + \hat{\varepsilon}_i$
    end if
  end for
end for
return v_i

7 Details about SimpleNet:

SimpleNet [1] is a lightweight and high-performance architecture for anomaly detection and localization. It combines pretrained feature extraction, domain adaptation, and feature-space anomaly synthesis to form an efficient pipeline. The model consists of the following components:

- A pretrained **Feature Extractor** F_ϕ
- A simple **Feature Adaptor** G_θ
- A **Feature-Space Anomaly Generator** (training only)
- A shallow **Discriminator** D_ψ

7.1 Feature Extraction and Adaptation

Given an input image $x_i \in \mathbb{R}^{H \times W \times 3}$, features are extracted using a pretrained backbone:

$$o_i = F_\phi(x_i)$$

The feature map $o_i \in \mathbb{R}^{H' \times W' \times C}$ is then adapted to the target domain:

$$q_i = G_\theta(o_i)$$

The feature adaptor G_θ is implemented as a simple linear layer without bias:

$$q_{i,h,w} = W \cdot o_{i,h,w}$$

7.2 Anomalous Feature Generation

To simulate anomalies during training, Gaussian noise is added to the adapted features:

$$q^-_{i,h,w} = q_{i,h,w} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

7.3 Discriminator and Loss Function

The discriminator D_ψ is trained to distinguish normal features from anomalous ones. It outputs a scalar score per spatial location:

$$s_{i,h,w} = D_\psi(q_{i,h,w}) \quad \text{and} \quad s^-_{i,h,w} = D_\psi(q^-_{i,h,w})$$

SimpleNet [1] uses a **truncated L1 loss** defined as:

$$\ell_{i,h,w} = \max(0, t^+ - s_{i,h,w}) + \max(0, -t^- + s^-_{i,h,w})$$

where $t^+ = 0.5$ and $t^- = -0.5$ are truncation thresholds.

The total training loss is averaged across spatial locations and samples:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{H' \cdot W'} \sum_{h,w} \ell_{i,h,w}$$

7.4 Inference and Scoring

During inference, the anomaly generator is discarded. The final anomaly score at each location is:

$$s_{i,h,w} = -D_\psi(q_{i,h,w})$$

The full-resolution anomaly map is:

$$S_{\text{AL}}(x_i) = \text{Smooth}(\text{Upsample}(\{s_{i,h,w}\}))$$

The image-level anomaly score is:

$$S_{\text{AD}}(x_i) = \max_{h,w} s_{i,h,w}$$

7.5 Pseudocode: Training Loop

Algorithm 2 SimpleNet [1] Training (Pytorch-like)

Initialize F_ϕ (pretrained), G_θ, D_ψ
for each batch x in training data **do**
  $o \leftarrow F_\phi(x)$ ▷ Extract normal features
  $q \leftarrow G_\theta(o)$ ▷ Adapt features
  $q^- \leftarrow q + \mathcal{N}(0, \sigma^2)$ ▷ Generate anomalies
  $s \leftarrow D_\psi(q)$
  $s^- \leftarrow D_\psi(q^-)$
  Compute loss:

$$\ell = \max(0, t^+ - s) + \max(0, -t^- + s^-)$$

  loss = mean(ℓ)
  Backpropagate and update G_θ, D_ψ
end for

8 Evaluation Metrics and Model Results

8.1 Evaluation Metrics

To evaluate the performance of our anomaly detection models, we employ the **Area Under the Receiver Operating Characteristic Curve (AUROC)** as the primary metric. This metric is widely used in binary classification tasks and is particularly suitable for imbalanced settings such as industrial anomaly detection.

Mathematical Definition of AUROC

Let the anomaly detection model assign a score $s(x_i) \in \mathbb{R}$ to each input image x_i. Let $y_i \in \{0, 1\}$ denote the ground truth label, where $y_i = 1$ indicates an anomaly and $y_i = 0$ indicates a normal sample.

Define the set of all anomalous and normal samples as:

$$\mathcal{P} = \{(x_i, s_i) \mid y_i = 1\}, \quad \mathcal{N} = \{(x_j, s_j) \mid y_j = 0\}$$

The AUROC is defined as the probability that a randomly chosen anomalous sample receives a higher score than a randomly chosen normal sample:

$$\text{AUROC} = \mathbb{P}(s_i > s_j \mid x_i \in \mathcal{P},\ x_j \in \mathcal{N})$$

In practice, this is estimated as:

$$\text{AUROC} = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{(x_i,s_i)\in\mathcal{P}} \sum_{(x_j,s_j)\in\mathcal{N}} \delta(s_i, s_j)$$

where

$$\delta(s_i, s_j) = \begin{cases} 1, & \text{if } s_i > s_j \\ 0.5, & \text{if } s_i = s_j \\ 0, & \text{if } s_i < s_j \end{cases}$$

AUROC values range from 0 to 1, where 0.5 indicates random performance and 1.0 represents perfect separation of anomalies from normal data.

Image-Level vs Pixel-Level AUROC

We report two types of AUROC scores:

- **Image-level AUROC (I-AUROC):** Measures whether an entire image contains any anomaly.
- **Pixel-level AUROC (P-AUROC):** Measures how accurately the model can localize anomalous pixels using a pixel-wise anomaly map.

Both metrics are computed across all categories in the MVTec AD [3] dataset, and the average is reported to assess overall performance.

8.2 Model Results

We evaluate both **GLASS** and **SimpleNet [1]** on the MVTec AD [3] benchmark. GLASS achieves superior pixel-level localization due to its LAS branch, while SimpleNet [1] performs competitively with significantly faster inference. Detailed results are presented in Table 1 and 2, including AUROC scores per category and averaged over all textures and objects.

Figure 4: Defect Identified by **GLASS** and the Ground Truth images for bottle and leather

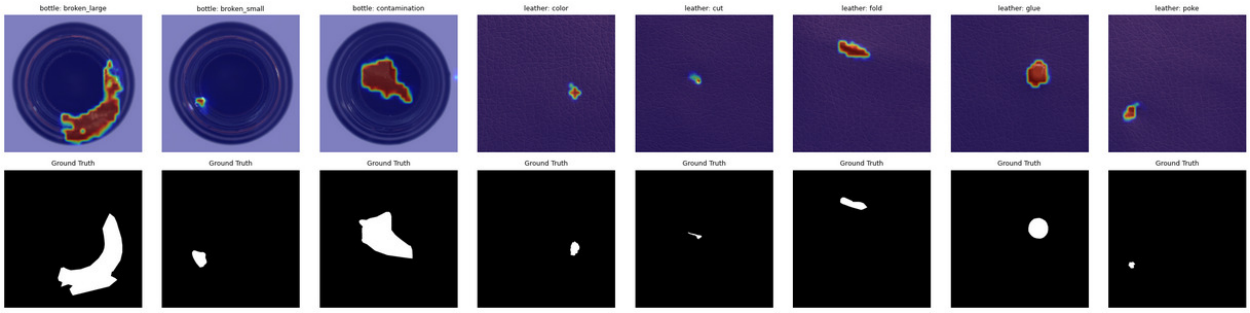

Figure 5: Defect Identified by **GLASS** and the Ground Truth images for carpet and tile

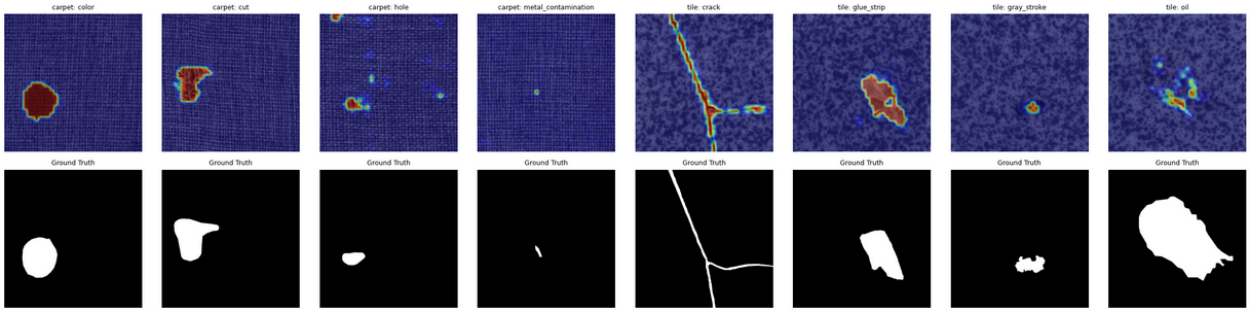

Table 1: Performance of **GLASS** on the MVTec AD [3] dataset. We report image-level (I-AUROC) and pixel-level (P-AUROC) scores in percentage for each category.

Category	**I-AUROC (%)**	**P-AUROC (%)**
Bottle	99.60	83.76
Cable	87.95	71.55
Capsule	94.34	63.60
Carpet	93.24	93.51
Grid	97.24	88.02
Hazelnut	94.61	73.80
Leather	99.80	97.50
Metal Nut	99.56	70.58
Pill	91.49	62.11
Screw	86.62	68.41
Tile	100.00	87.81
Toothbrush	87.50	90.57
Transistor	86.50	44.69
Wood	98.51	83.10
Zipper	99.89	81.96
Average	**94.91**	**77.89**

GLASS shows strong performance on the MVTec AD [3] dataset with an average I-AUROC of 94.91% and P-AUROC of 77.89%. It excels in categories like Tile, Leather, and Zipper, achieving near-perfect image-level accuracy. However, pixel-level performance is lower in complex categories like Transistor and Pill, indicating challenges in precise anomaly localization.

Figure 6: Defect Identified by **SimpleNet** and the Ground Truth images for leather

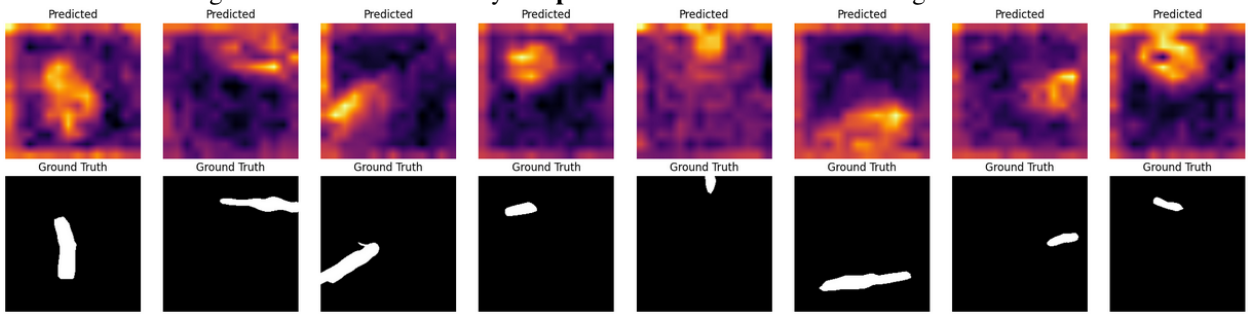

Table 2: Performance of **SimpleNet [1]** on the MVTec AD [3] dataset. We report image-level (I-AUROC) and pixel-level (P-AUROC) scores in percentage for each category.

Category	**I-AUROC (%)**	**P-AUROC (%)**
Bottle	98.49	82.75
Cable	88.08	79.62
Capsule	90.03	95.23
Carpet	93.86	75.40
Grid	92.94	87.87
Hazelnut	60.04	68.99
Leather	84.14	87.19
Metal Nut	71.36	89.96
Pill	71.09	86.19
Screw	69.34	76.93
Tile	78.86	67.79
Toothbrush	80.83	85.73
Transistor	77.08	79.08
Wood	92.98	76.82
Zipper	98.53	92.17
Average	**83.66**	**81.99**

SimpleNet shows strong performance on the MVTec AD [3] dataset with an average I-AUROC of 83.66% and P-AUROC of 81.99%. It performs particularly well in categories like Bottle, Capsule, and Zipper, achieving high image- and pixel-level accuracy. However, the model exhibits lower image-level scores in challenging categories such as Hazelnut, Pill, and Screw, suggesting limitations in detecting subtle global anomalies. Overall, SimpleNet demonstrates competitive performance, especially in localizing defects at the pixel level.

9 Future Work

While this study provides valuable insights into the strengths and limitations of GLASS and SimpleNet [1] for unsupervised industrial anomaly detection, several promising directions remain open for further exploration.

One key area for future work is the development of hybrid models that combine the strengths of both approaches. For instance, integrating the dual-level synthesis capability of GLASS with the lightweight architecture of SimpleNet [1] could yield a model that balances accuracy and efficiency, making it both effective in defect detection and suitable for deployment in resource-constrained environments.

Another important direction is the automation of hyperparameter tuning and category-specific adaptation. As observed during our implementation, models like GLASS can be sensitive to choices such as the synthesis bounds or projection constraints, which may require manual tuning for different defect types. Future research could incorporate meta-learning or adaptive controllers to tune these parameters dynamically based on the input distribution or intermediate model feedback.

There is also significant scope for improving feature-space anomaly synthesis. While current approaches often rely on isotropic Gaussian noise, more expressive perturbation strategies—such as directional noise based on learned uncertainty or adversarial-style training—could better mimic real-world anomalies without requiring visual-level synthesis.

Moreover, expanding the scope of anomaly detection frameworks to include temporal information (e.g., in industrial video streams) or multi-modal inputs (such as thermal images, depth maps, or sensor data) could substantially improve robustness in production environments.

Finally, the reproducibility challenge encountered in our study suggests a broader need for standardized benchmarks, transparent implementation repositories, and detailed reporting of architectural and training details. Community-driven efforts in this direction would significantly improve the reliability and comparability of future research in this domain.

In summary, advancing industrial anomaly detection will require progress not only in model architecture and training techniques, but also in usability, interpretability, and adaptability to real-world constraints—areas where both academia and industry must continue to collaborate.

10 Conclusion

This work presents a comparative study of two recent unsupervised anomaly detection methods—GLASS and SimpleNet [1] tailored for industrial visual inspection. GLASS employs a sophisticated dual-synthesis mechanism, generating anomalies in both feature and image spaces, which allows it to effectively detect subtle and localized defects. SimpleNet [1], on the other hand, adopts a minimalistic and efficient architecture that perturbs feature space using Gaussian noise, making it highly suitable for real-time applications where computational resources are limited.

Through careful reimplementation and evaluation on the MVTec AD [3] dataset, we demonstrated the practical strengths and trade-offs of each approach. GLASS excels in localization accuracy and diversity of defect modeling but demands higher training time and complexity. SimpleNet [1] offers fast, interpretable inference with competitive detection performance, though it struggles with capturing more intricate or localized anomalies.

While our experimental results were largely consistent with the core findings of the original papers, we were not able to fully reproduce their reported performance. This discrepancy can be attributed to several factors, including limited access to precise implementation details, hidden hyperparameter tuning, variation in backbone initialization, and infrastructure constraints such as batch size and memory availability. These challenges underline the importance of clearer benchmarking standards and reproducibility practices in anomaly detection research.

Overall, our study emphasizes that the design of industrial anomaly detection systems must balance accuracy with efficiency and interpretability. Future work can build on this foundation by combining the controllability of GLASS with the speed of SimpleNet [1], developing more generalizable synthesis techniques, and exploring architectures that adapt dynamically to different types of defects without relying on manual tuning.

References

[1] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. SimpleNet: A Simple Network for Image Anomaly Detection and Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. https://arxiv.org/abs/2303.15140

[2] Qiyu Chen, Huiyuan Luo, Chengkan Lv, and Zhengtao Zhang. A Unified Anomaly Synthesis Strategy with Gradient Ascent for Industrial Anomaly Detection and Localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. https://arxiv.org/abs/2407.09359

[3] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD [3] — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019. https://www.mvtec.com/company/research/datasets/mvtec-ad

[4] Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep Industrial Image Anomaly Detection: A Survey. *Machine Intelligence Research*, 21(1):104–135, 2024. https://link.springer.com/article/10.1007/s11633-023-1459-z

[5] M-3LAB. Awesome Industrial Anomaly Detection. GitHub repository, 2023. https://github.com/M-3LAB/awesome-industrial-anomaly-detection

[6] Qiyu Chen, Huiyuan Luo, Chengkan Lv, and Zhengtao Zhang. A Unified Anomaly Synthesis Strategy with Gradient Ascent for Industrial Anomaly Detection and Localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. https://arxiv.org/abs/2407.09359

[7] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. *arXiv preprint arXiv:2011.08785*, 2020. https://arxiv.org/abs/2011.08785

[8] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. *arXiv preprint arXiv:2104.04015*, 2021. https://arxiv.org/abs/2104.04015

[9] João Santos, Triet Tran, and Oliver Rippel. Optimizing PatchCore for Few/Many-Shot Anomaly Detection. *arXiv preprint arXiv:2307.10792*, 2023. https://arxiv.org/abs/2307.10792

[10] Long Wen, Yang Zhang, Wentao Hu, and Xinyu Li. The Survey of Industrial Anomaly Detection for Industry 5.0. *International Journal of Computer Integrated Manufacturing*, 37(9):1059–1080, 2024. https://doi.org/10.1080/0951192X.2024.2397821

[11] Dinh-Cuong Hoang, Phan Xuan Tan, Anh-Nhat Nguyen, Minh-Khanh Pham, Ta Huu Anh Duong, Tuan-Minh Huynh, Son-Anh Bui, Duc-Manh Nguyen, Quang-Huy Ha, and Viet-Anh Trinh. Unsupervised industrial anomaly detection using paired well-lit and low-light images. *Results in Engineering*, 25:104309, 2025. https://doi.org/10.1016/j.rineng.2025.104309

[12] Aimira Baitieva, David Hurych, Victor Besnier, and Olivier Bernard. Supervised Anomaly Detection for Complex Industrial Images. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17754–17762, 2024. https://arxiv.org/abs/2405.04953