Texture-AD: An Unsupervised Anomaly Detection Dataset and Benchmark with Product-Variant Shifts for Textured Surfaces

Bohan Wang, Tianwu Lei, Silin Chen, Shurong Cao and Ningmu Zou*
Nanjing University
Suzhou, China
{bohanwang,tianwulei,silin.chen,221900433}@smail.nju.edu.cn *Corresponding Author Email: nzou@nju.edu.cn: nzou@nju.edu.cn

Abstract

Anomaly detection is a crucial process in industrial manufacturing and has made significant advancements recently. However, there is a large variance between the data used for development and the data collected in production environments. Conventional benchmarks typically evaluate algorithms in overly ideal conditions where the training and testing data share identical specifications, leading to overestimated performance that fails to hold up in real-world applications. To address this gap, we present Texture-AD, a comprehensive benchmark designed to systematically evaluate model robustness under realistic domain shifts. This dataset includes images of 15 types of cloth, 14 types of semiconductor wafers, and 10 types of metal plates acquired under different optical schemes. The key design innovation is the significant difference between the training set and the test set. The training set only contains normal samples from a certain subclass, while the test set introduces samples from previously unseen subclasses, along with colors, textures, and controllable variations in illumination. This setup replicates the practical challenge of detecting defects on product types or under imaging conditions not observed during training. Specifically, to adapt to diverse products in automated pipelines, we present a new evaluation method and results of baseline algorithms. The experimental results show that Texture-AD is a difficult challenge for state-of-the-art algorithms. To our knowledge, Texture-AD is the first dataset to be devoted to evaluating industrial defect detection algorithms in the real world. The dataset is available at https://huggingface.co/datasets/texture-ad/Texture-AD-Benchmark.

Keywords: Anomaly Detection, Benchmark Dataset, Texture Defects, Industrial Inspection

1. Introduction

Industrial inspection algorithms are typically developed and tested using collected data before deployment, for use in automated quality control equipment on production lines. In recent years, a variety of detection methods have been developed for detecting an anomalous image region in image data through contemporary machine learning approaches. These methodologies have demonstrated promising results on established datasets. Present evaluation strategies typically entail integrating flawless production data of a single object category during the training stage and evaluating performance using data containing anomalies.

The acquisition of flawless production data has become more accessible when contrasted with defective data. However, a production line is often required to deal with various specifications of similar products, such as gray cloth, red cloth, mesh cloth, different types of wafers, as well as black brushed metal plates, gold frosted metal plates, etc. While these different specifications share certain common features, they also present significant differences. Additionally, minor fluctuations in external conditions, such as lighting environment and camera settings, result in a data distribution after deployment that is unlikely to align with the data collected during the training phase. This situation places increased requirements on the robustness of the algorithms.

Humans have the natural ability to visually discern the similarities and differences in images and to detect defects and irregularities within them. Currently, there are many commonly used datasets for anomaly detection, which vary greatly in the scenes and scale they contain. For example, datasets related to cloth texture [19, 29] generally have a good amount of data, but they differ significantly from actual production scenarios. In addition, as chips become an increasingly important field of research worldwide, wafer defect detection has become an essential part of the pro-

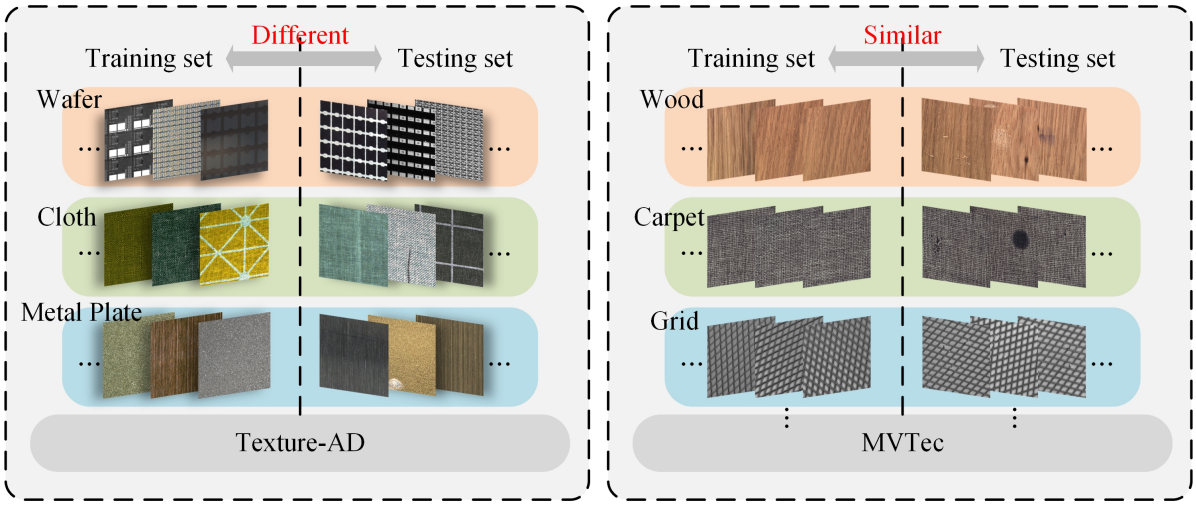
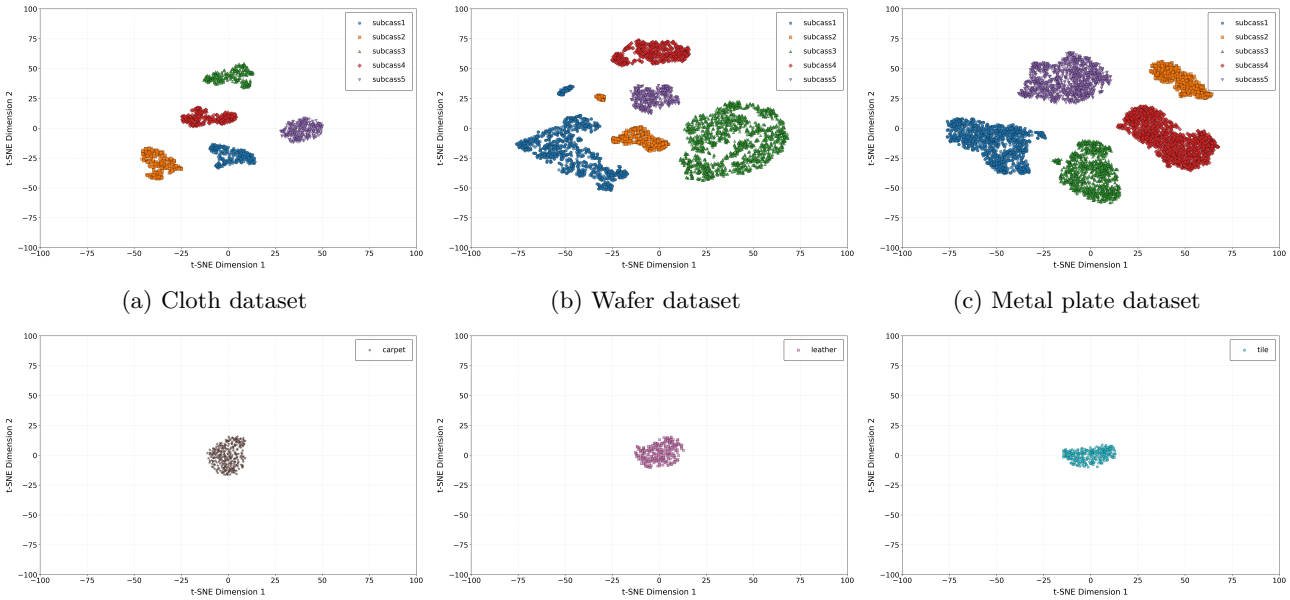


Figure 1: Difference between existing evaluation methods and actual situation.



(a) Cloth dataset (b) Wafer dataset (c) Metal plate dataset

(d) Carpet (e) Leather (f) Tile

Figure 2: t-SNE feature visualization maps of Texture-AD and carpet, leather, and tile in MVTec.

cess. Therefore, the demand for wafer defect detection datasets [33] in industrial inspection is also growing, yet there are very few open-source wafer defect detection datasets available. Moreover, there are more datasets related to metal defects in industrial production [2, 30, 42, 20, 11, 41], but they generally include material types and apply to a more limited range of scenarios. There are also datasets related to crack defects [13, 35], such as cracks in bridge surfaces and concrete floors.

So far, modern machine learning systems have encountered considerable challenges in addressing related issues, mainly because the existing datasets are not particularly well-suited to real-world scenarios. Currently, the evaluation of anomaly detection algorithms often relies on datasets such as MVTec [4], where the features of flawless and defective items exhibit a high degree of consistency, resulting in higher performance metrics than those observed in actual deployment. Therefore, this paper proposes the Texture-AD dataset [31], which clearly demonstrates the differences between Texture-AD and the MVTec dataset in Table

Table 1: Evaluation protocol difference between Texture-AD and MVTec.

Category	Train		Test	
	Images	Category Labels	Images	Category Labels
MVTec	O	O	O	O
Ours	O	O	O	X

1. As shown in Figure 1, the training data provided by the MVTec dataset and the test data completely belong to the same product, making it impossible to evaluate the algorithms under development correctly. Therefore, in Texture-AD, we provide a variety of specifications of three products as the training set, and at the same time, provide the same type of products with different specifications from the training set as the test set, which can evaluate the performance of the algorithm based on the consideration of algorithm robustness and generalization ability. Furthermore, as shown in the Figure2, using the feature visualization technology based on t-SNE, a comparative analysis was conducted on the deep feature distributions of Texture-AD and three representative texture subclasses (carpet, leather, and tile) in the MVTec dataset. The results indicated that within a certain category of Texture-AD, different subclasses exhibited significant intra-class discreteness in the feature space. Additionally, the feature clusters corresponding to different categories were clearly separated, demonstrating high intra-class diversity and significant inter-class distinction. In contrast, the feature distributions of similar samples in MVTec were more compact, with relatively smaller inter-class distances. This contrast highlights the core advantage of Texture-AD in simulating real industrial inspection scenarios, as this dataset constructs a more diverse and complex feature space by covering various product specifications, changing optical conditions, and natural defects. Therefore, Texture-AD more effectively evaluated the robustness, generalization ability, and feature separation performance of unsupervised anomaly detection algorithms in an open set environment. Through this approach provides a benchmark testing platform that is closer to the actual production environment and better supports the development of algorithms intended for practical deployment. The training set of this dataset includes 15 subclasses of cloth images, 14 subclasses of wafer images, and 10 subclasses of metal plate images. All cloth images come from the same type of cloth, wafer images come from 14 different subclasses of wafers, and metal plate images come from metal plates with 5 different colors of brushed and matte surfaces, photographed under similar lighting conditions. The test set includes defective cloth images, wafer images, and metal plate images photographed from the actual production process, which show slight differences in camera settings, lighting conditions, and the design of cloth, wafers, and metal plates compared to the training set.

The contributions of our paper can be summarized into three main aspects:

- We present a novel and comprehensive dataset for unsupervised anomaly detection in industrial quality inspection. It simulates real-world industrial inspection scenarios, and it has a sufficient number of data samples and data scale, including 43120 high-resolution images collected in various optical environments from 39 different subclasses under three major categories, which contain a variety of different types of defects.
- We conduct a comprehensive evaluation of current state-of-the-art methods for unsupervised anomaly detection, assessing their segmentation and classification performance on the anomalous images during the development process.
- We provide a well-designed evaluation protocol to compare the performance of unsupervised anomaly detection algorithms in actual development environments.

2. Related Work

Computer vision equipment for detecting surface defects has largely replaced manual inspections across industries like 3C electronics, automotive, machinery, semiconductors, chemicals, and so on. Traditional methods use standard image processing and classifiers with handcrafted features, while effective imaging schemes ensure clear defect visibility under uniform lighting. Recently, deep learning has become prevalent for defect detection.

DAGM2007 dataset [19] is artificially generated but resembles real-world problems. Six categories referred to as the development dataset should be used for algorithm development. The remaining four categories (referred to as the competition dataset) can be used to evaluate performance. AITEX dataset [29] is an image dataset focused on the textile industry, designed to

Figure 3: Example images of all fifteen different colors and textures of cloth from the Texture-AD dataset. For each category, the first row displays anomaly-free examples. The second row shows examples of the anomaly.

support research and application of machine learning and computer vision technology in the field of textile quality inspection. However, the aforementioned two datasets have issues with unclear defect labeling and a rather singular background type and defect type, which cannot fully simulate the complex detection scenarios in actual industrial environments.

The WM-811K dataset [33] is a dataset specifically for semiconductor wafer map defect type identification, with images in the dataset mainly coming from actual production environments of wafer maps, obtained through electrical testing, and used to describe the state of wafer defects. However, the WM-811K represents without texture details and pattern information.

A dataset [2] collected six typical surface defects of hot-rolled steel strips. This surface defect dataset faces two major challenges: large differences in appearance among defects within the same category, and similarities between defects of different categories, with defect images affected by lighting and material changes. The NEU-surface-defect-database [30] has six typical surface defects of hot-rolled steel strips, namely rolling scale, patches, cracks, pitted surfaces, inclusions, and scratches. The improved X-SDD dataset [42] includes: seven typical types of hot-rolled steel strip defect images. Due to the imbalance of sample quantity in X-SDD, it provides conditions for researchers to solve the problem of sample imbalance. The SD-saliency-900 dataset [20] includes three types of steel strip surface defects (inclusions, patches, and scratches), including steel surface defect detection images and corresponding pixel-level binary masks. RSDDS-113 dataset [11], with samples taken from the actual industrial production line of a section steel factory, collects 20 track sections with defect information. Each pair of images in this dataset consists of a left camera image and the corresponding depth image; the dataset has a high degree of annotation credibility, but the number of data sam-

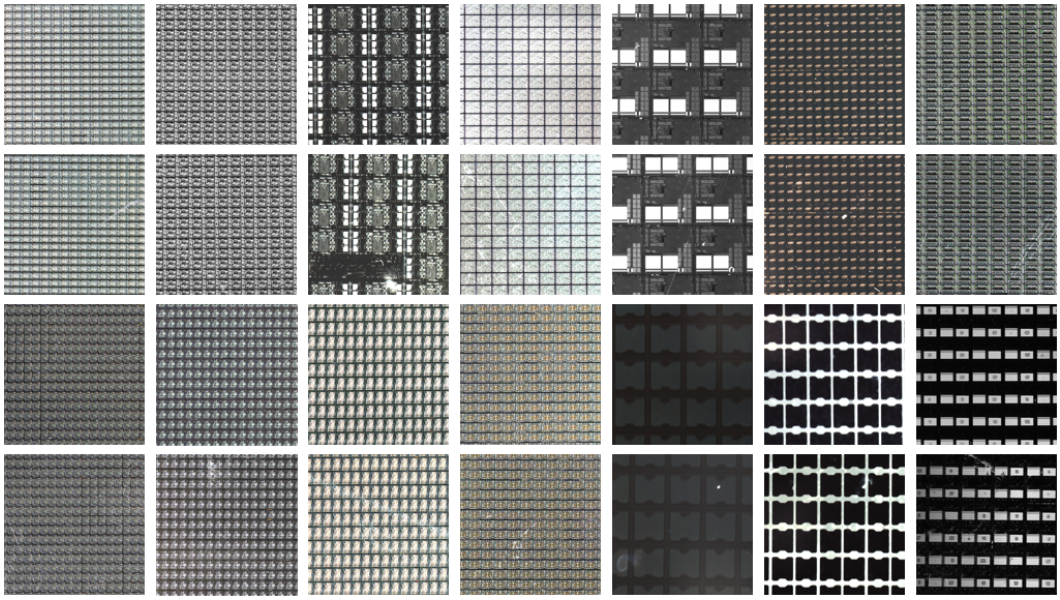


Figure 4: Example images of all ten different colors and textures of metal plates from the Texture-AD dataset. For each category, the first row displays anomaly-free examples. The second row shows examples of the anomaly.

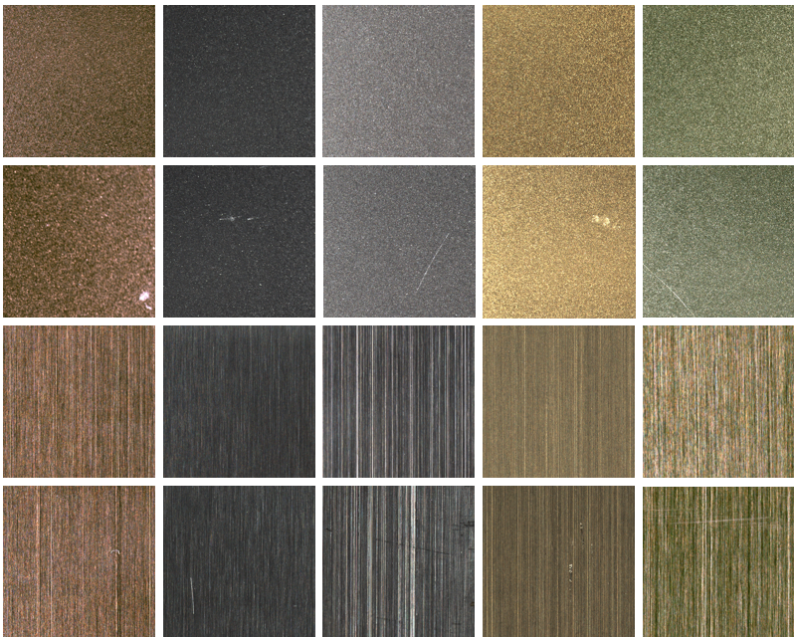


Figure 5: Example images of all fourteen different colors and textures of wafers from the Texture-AD dataset. For each category, the first row displays anomaly-free examples. The second row shows examples of the anomaly.

ples is fewer. The Rail-5k dataset [41] is used for the task of steel rail surface defect detection. The dataset can be used for two settings, the first is a supervised setting trained with marked images, the fine-grained nature of defect categories, and the long-tail distribution make it difficult for visual algorithms to solve. The second is a semi-supervised learning setting promoted by unmarked images, including possible image damage and domain shift with marked images. The dataset can support both supervised and semi-supervised learning settings. In actual production, there may be unknown types of defects, making it difficult for the aforemen-

tioned traditional datasets based on known defect patterns to cope. In addition, it is difficult to obtain a large number of defect samples in the aforementioned datasets, leading to the problem of small sample sizes when training deep learning models.

The Concrete Crack Images for Classification dataset [13] is created specifically for the task of concrete crack classification. This dataset typically contains tens of thousands of images of concrete surfaces, showing cracks of different types and severities. The Crack-Detection dataset [35] is designed specifically for crack detection tasks, containing images for training and evaluating crack identification algorithms. These images usually come from various material surfaces, especially concrete and other construction engineering materials, because cracks in these materials may lead to structural problems. The images in the aforementioned datasets have issues with varying quality, including resolution, lighting conditions, angles, and background complexity, which may affect the performance of crack detection algorithms in the deployment process.

MVTec [4] contains images of anomalous samples with various defects, manually generated. This is a popular dataset for unsupervised anomaly detection that simulates real-world industrial inspection scenarios. The dataset provides the possibility of evaluating unsupervised anomaly detection methods for various textures and object classes with different types of anomalies. Since it provides pixel-level precise ground truth labels for the abnormal areas in the images, it is possible to evaluate anomaly detection methods for image-level classification and pixel-level segmentation.

In industrial settings, the prevalence of normal samples over defective ones creates a dataset imbalance, affecting model training and generalization. Acquiring a significant number of defective samples is costly and time-consuming, especially for rare defects. Current datasets may not cover all defect types, limiting the model's ability to identify unusual defects. The complexity of industrial products' appearance and potential labeling inconsistencies adds to the challenge of defect detection. Moreover, the need for real-time responses in industry is often not met by existing datasets, leading to models that may not perform well in new environments.

3. Dataset

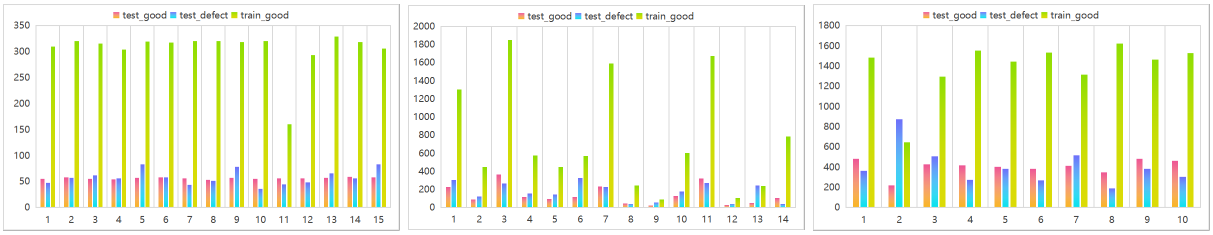
The anomaly detection dataset we propose includes 15 subclasses of cloth, covering a variety of colors, materials, and texture defects, 14 different subclasses of wafers, and 10 subclasses of metal plates, including 5 colors each with brushed and matte finishes, totaling 10 subclasses of textures. Examples and comparisons of normal images and representative defect images from the three aforementioned categories are shown in Figure 3, Figure 4, and Figure 5. The defects in our dataset are imperfections that occur in actual production environments, making it extremely valuable for the study of industrial quality inspection algorithms. Cloth defects include pencil marks, cuts, marker stains, water stains, black and white dots, threads, inconsistent sewing distances, and color differences caused by dyeing. Wafer and metal plate defects include scratches, stains, and inherent manufacturing defects, all of which naturally occur in the production process. As shown in Figure 6, our dataset contains a total of 43120 images, with 28973 images used for training and validation, and 14147 images for testing. The training set includes only defect-free images. The test set contains two types of images: images with various types of defects and defect-free images. Figure 7 shows the percentage of the image area occupied by the anomalous regions.

Specific to the division of the dataset, we provide good production images from multiple subclasses for each category as the training set, allowing the model to learn the characteristics and differences of each subclass. At the same time, we also provide defect images and good production images from the same category for the test set to evaluate the model's recognition ability when facing actual defects. The number of samples for each category and the specific allocation of subclasses are detailed in the appendix for reference.

3.1. Data Generation

All images were captured using a high-resolution industrial camera (MV-CS200-10 GC) at a resolution of 5472×3648 pixels, in conjunction with two light sources. The optical scheme was altered by adjusting the position and brightness of the light sources. We programmatically controlled the brightness and incidence angle of two light sources to systematically simulate non-uniform illumination conditions commonly encountered in production environments. Specifically, the brightness of each light source was varied linearly between 70% and 130% of its rated intensity, while the incidence angle was adjusted within $\pm 15^{\circ}$ relative to the sample surface normal. This controlled variation generated a diverse range of lighting conditions, encompassing differences in brightness, reflections, and shadow patterns.

To better align with the defects produced in the industrial manufacturing process, we created some artificial defects on the cloth, while the wafers and metal plates exhibited naturally occurring defects. In the ex-



(a) Cloth dataset (b) Wafer dataset (c) Metal plate dataset

Figure 6: Data Statistics (a) The cloth dataset consists of 6283 images, with 4569 images in the training set and 1714 images in the test set. (b) The wafer dataset consists of 14861 images, with 10525 images in the training set and 4336 images in the test set. (c) The metal plate dataset consists of 21976 images, with 13879 images in the training set and 8097 images in the test set.

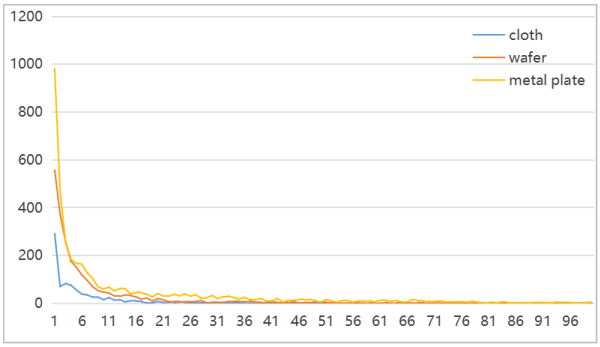


Figure 7: Statistics of the percentage of the image area occupied by the anomaly regions.

perimental design, our work employed a physical simulation method to construct a dataset of cloth defects. Specifically, stains were created on the cloth surface using artificial contaminants (such as pencil graphite and lime water suspension), and controlled mechanical damage (such as sharp instrument scratching) was used to simulate scratches. As a result, synthetic defect samples with clear annotations were generated. This method has a certain apparent similarity in visual features, such as macroscopic morphology, contrast, and texture fracture, to the defects produced during real production due to contamination and mechanical scratching, providing the model with repeatable and diverse training samples. The introduction of synthetic data enhances the ability of the model to recognize the basic visual patterns of defects and improves its robustness in detecting different types of defects. By combining such physically synthesized data with real samples for mixed training, it is possible to effectively enrich the diversity of the features learned by the model, providing it with more comprehensive representation information of defects, and thereby enhancing its generalization performance and practicality in complex real scenarios.

3.2. Data Labeling

Our image acquisition and defect annotation process is depicted in Figure 8. The defects in our dataset were manually annotated using the LabelMe annotation tool. To ensure high performance and strong generalization capability of deeplearning-based industrial defect detection models, our annotation pipeline rigorously adheres to a systematic data annotation protocol. This protocol emphasizes the principles of annotation consistency, geometric accuracy, and stringent quality control, with clearly defined standards for defect definition and visual representation. All annotations are required to ensure that bounding boxes or segmentation masks precisely align with the actual edges of defects. Unified handling rules have been established for challenging scenarios such as small, blurry, and densely clustered defects. The annotation task is carried out jointly by three engineers with extensive expertise in defect detection labeling. Initially, the three annotators independently label the same set of data, after which the union of their annotations is taken. Subsequently, state-of-the-art industrial defect detection algorithms are employed for auxiliary identification. Based on the algorithmic outputs, a second round of manual annotation is performed, followed by a two-level review process and regular calibration meetings. This workflow forms a closed-loop quality assurance system aimed at minimizing noise and ambiguity from the data source. The outcome is a set of high-quality, consistent benchmark ground-truth annotations that provide a reliable foundation for model training and underpin the robustness and reliability of downstream algorithms.

Subsequently, these images were cropped to the appropriate output size. All images have a resolution of 1024×1024 pixels. The training set images were ob-

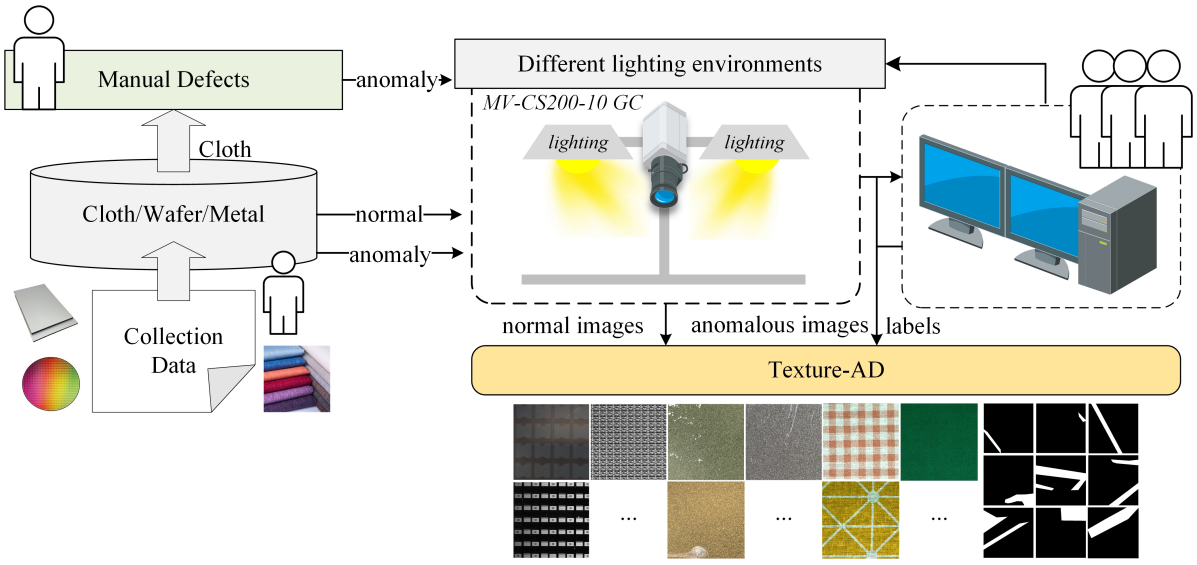


Figure 8: Image acquisition and defect annotation processes. The Texture-AD images were captured using a high-resolution industrial camera (MV-CS200-10 GC). The optical scheme was altered by adjusting the position of the light source and the brightness of the two light sources. The cloth images include both artificial and natural defects, while the wafer and metal plate images consist solely of natural defects. The defect annotation work for the images was performed using LabelMe.

tained under relatively stable lighting conditions. However, for the test set, we intentionally varied the optical scheme to simulate the imaging discrepancies between the algorithm training phase and actual deployment. We provided pixel-level ground truth annotations for each defective image area.

4. Anomaly Detection Methods

The current research trend in anomaly detection is primarily focused on unsupervised anomaly detection. This trend has emerged due to the fact that obtaining anomalous samples requires a significant investment of human and financial resources. In this research context, training data contains only normal samples, while test data includes both normal and anomalous samples. Industrial image anomaly detection is a specific branch within the field of anomaly detection, and we mainly evaluate and compare it using the following three research directions.

4.1. Synthesis-based Anomaly Detection

Some supervised learning methods use a limited number of anomaly samples to synthesize more anomaly samples to enhance training effectiveness. For example, A basic architecture that integrates CycleGAN [8] with ResNet/U-Net as the generator is used to transfer defects from one image to another [25]. SDGAN [17] achieved better results than CycleGAN by improving the style transfer network. DRAEM [39] first restores the normal image with pseudo-anomaly interference to obtain feature representation and then uses a discriminator network to distinguish anomalies, demonstrating excellent performance. Although this field has made certain research progress, it still has a huge development space compared to other fields with clear research directions.

4.2. Reconstruction-based Anomaly Detection

These methods are based on the assumption that a reconstruction model trained only on normal samples can successfully reconstruct images in normal areas [6, 7, 40, 26, 37] but fail in abnormal areas. Early attempts included autoencoders(AE) [6, 9], variational autoencoders(VAE) [40, 15] and generative adversarial networks(GAN) [26, 1, 22, 38]. However, these methods may cause the model to learn certain tricks, leading to the effective recovery of anomalies as well. To address this issue, researchers have adopted various strategies, such as introducing guidance information (structure [43] or semantics [28, 34]), memory mechanisms [12, 14, 21], iterative mechanisms [10], image masking strategies [36], and pseudo-anomaly [9, 23] PyramidFlow [16], based on the transformer and further design, set a new record on MVTec.

4.3. Feature-Embedding Based Methods

Feature embedding methods are committed to distinguishing normal and abnormal samples at the feature representation level. Uniformed Students [5] pioneered the use of discriminative latent embeddings for anomaly detection. This model is simple and effective, significantly outperforming other benchmark methods. STPM [32] and MKD [27] utilize multi-scale features on different network layers for feature distillation, although there are differences in their methods. In addition, SimpleNet [18] has achieved satisfactory results by introducing noise into the feature embedding to simulate negative samples.

5. Benchmark

5.1. Baseline Methods

5.1.1 SimpleNet

SimpleNet [18] proposed a simple and easy-to-apply network for detecting and localizing anomalies in images. We evaluated using the publicly available SimpleNet implementation on PyTorch. The backbone network used Wide Resnet50 as the backbone network, setting the feature dimension of the feature extractor to 1536 to accommodate 329×329 sized input images. The anomaly feature generator added isotropic Gaussian noise $N(0, \sigma^2)$, where σ defaults to 0.015. The subsequent discriminator includes a linear layer, a batch normalization layer, a leaky ReLU with a slope of 0.2, and a linear layer. The Adam optimizer was used, with learning rates of 0.0001 and 0.0002 set for the feature adapter and discriminator, respectively, and a weight decay of 0.00001. Each dataset was trained for 160 epochs with a batch size of 8.

5.1.2 PyramidFlow

PyramidFlow [16] proposed a new anomaly localization method, which is based on the defect contrastive localization paradigm using a pyramid of normalization flows for multi-scale fusion and volume normalization to achieve high-resolution defect localization. We used a fixed pyramid layer number $L = 8$, image resolution of 256×256, and channel number $C = 24$, and varied the stacked layer number D to explore the trends in memory usage and model parameterization. During training, sample mean normalization was used, and the running mean was updated with a momentum of 0.1. At test time, volume normalization was based on the running mean.

5.1.3 Mean-Shift

Mean-Shift [24] introduced a novel self-supervised representation learning method to improve anomaly detection. It pointed out that traditional contrastive learning methods are not suitable for pre-trained features. Hence, they proposed the Mean-Shifted Contrastive Loss. In the experiments targeting ResNet152, we fine-tuned the last two blocks of a ResNet152 model pre-trained on the ImageNet dataset and added an ℓ_2 normalization layer, a process that lasted for 10 training epochs. For the experiments with ResNet18, we fine-tuned the entire backbone of a ResNet18 model pre-trained on ImageNet and similarly added an ℓ_2 normalization layer, a process that included 20 training epochs. In both cases, we minimized the Mean-Shifted Contrastive loss function with a temperature parameter τ set to 0.25. We used the Stochastic Gradient Descent (SGD) optimizer with a weight decay of 5×10^{-5}, and without momentum. We set the size of each mini-batch to 64.

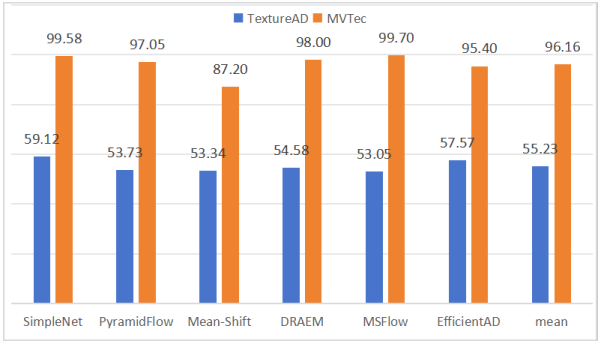


Figure 9: The comparison of the average Image-AUROC obtained by various algorithms on TextureAD and MVTec.

5.1.4 DRAEM

In addition to reconstruction methods, DRAEM [39] primarily regards surface anomaly detection as a discriminative problem and proposes a Discriminatively Trained Reconstruction Anomaly Embedding Model (DRAEM). This method learns the joint representation of anomalous images and their anomaly-free reconstructions while learning the decision boundary between normal and anomalous examples. The method can directly localize anomalies without the need for additional complex post-processing of the network output, and can be trained using simple and universal anomaly simulation. In our experiments, the network was trained for 700 epochs. The learning rate was set to 10^{-4}, and it was multiplied by 0.1 after 400 and 600

Table 2: Comparison of state-of-the-art works on the cloth of Texture-AD. Image-AUROC (top row) and Pixel-AUROC (bottom row) are displayed in each entry.

Category	subclass1	subclass2	subclass3	subclass4	subclass5	Average
SimpleNet	65.08	59.26	58.83	70.40	68.47	64.41
	58.30	51.52	63.48	70.68	54.47	59.69
PyramidFlow	57.88	63.18	60.74	59.39	49.72	58.18
	68.00	57.06	60.74	57.26	34.84	55.58
Mean-Shift	66.22	33.66	66.21	65.69	39.54	54.26
	-	-	-	-	-	-
DRAEM	57.58	50.21	55.44	58.01	55.95	55.44
	60.99	65.36	56.91	53.45	77.03	62.75
MSFlow	50.00	54.01	50.00	50.00	50.14	50.83
	56.11	63.14	51.66	47.44	42.23	52.12
EfficientAD	65.65	76.98	55.69	42.38	72.20	62.58
	62.76	58.92	47.08	38.75	61.77	53.86

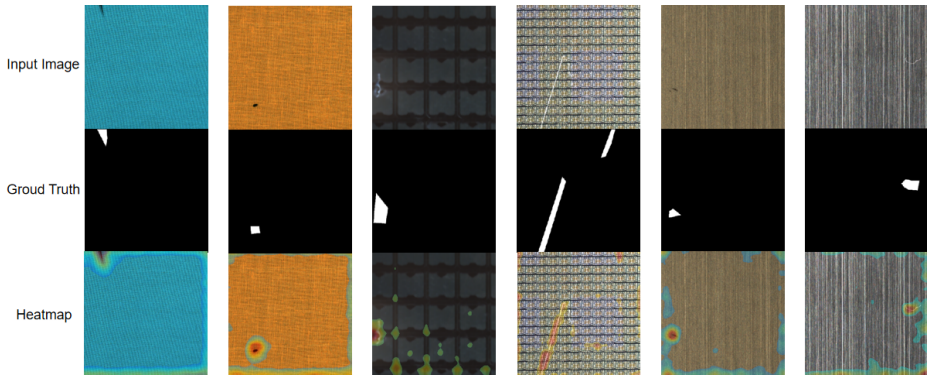


Figure 10: Visualization of SimpleNet results. It presents the anomaly segmentation results for three categories of materials in Texture-AD: cloth, wafer, and metal plate. The top row demonstrates the original image, the middle row shows pixel defect region annotation, and the bottom row is the heatmap of SimpleNet.

epochs. Image rotation from -45 to 45 degrees was used as a data augmentation method.

5.1.5 MSFlow

MSFlow [44] proposed a multi-scale flow-based framework for unsupervised anomaly detection, which utilizes normalization flows to handle features at different scales to adapt to anomalies of various sizes. During the experimental process, we used Wide ResNet50 and ResNet18 as feature extractors. The training was conducted with a batch size of 16. The optimizer used was Adam with an initial learning rate of 10^{-4}, and the learning rate was reduced at 70% and 90% of the training progress.

5.1.6 EfficientAD

EfficientAD [3] proposed a lightweight feature extractor that processes images with millisecond-level latency on modern GPUs, using a student-teacher approach to detect anomalous features and effectively detect logical anomalies. In the experiments, we set the hard feature loss mining factor (phard) to 0.999, meaning that on average, 10% of the values in each dimension are used for backpropagation. The Adam optimizer was used with an initial learning rate of 10^{-4} and a weight decay of 10^{-5}. During training, if the number of iterations exceeded 66500, the learning rate was reduced to 10^{-5}.

5.2. Evaluation Method

To ensure comparability with existing research and alignment with real-world application scenarios, this section elaborates on the adopted evaluation protocol, data preprocessing pipeline, performance metrics, and specific implementation details.

Table 3: Comparison of state-of-the-art works on the wafer of Texture-AD. Image-AUROC (top row) and Pixel-AUROC (bottom row) are displayed in each entry.

Category	subclass1	subclass2	subclass3	subclass4	Average
SimpleNet	52.11	59.66	53.66	50.68	54.03
	57.18	66.16	57.58	53.40	58.58
PyramidFlow	55.54	43.35	52.76	46.36	49.50
	51.23	39.47	51.52	44.63	46.71
Mean-Shift	52.83	53.29	55.44	48.28	52.47
	-	-	-	-	-
DRAEM	55.69	57.09	59.22	52.46	56.12
	44.91	34.10	35.01	43.59	39.40
MSFlow	51.19	49.78	53.64	50.00	51.15
	44.91	34.10	35.01	43.59	39.40
EfficientAD	50.28	42.25	50.23	45.51	47.07
	55.76	33.98	51.53	40.02	45.32

Table 4: Comparison of state-of-the-art works on the metal plate of Texture-AD. Image-AUROC (top row) and Pixel-AUROC (bottom row) are displayed in each entry.

Category	subclass1	subclass2	subclass3	Average
SimpleNet	59.07	59.87	57.83	58.92
	62.27	58.33	58.97	59.86
PyramidFlow	52.87	48.74	58.92	53.51
	53.42	48.86	57.67	53.31
Mean-Shift	44.34	47.39	45.04	53.29
	-	-	-	-
DRAEM	52.07	56.32	51.48	45.59
	58.41	51.53	57.31	55.75
MSFlow	62.90	53.54	59.78	58.74
	65.37	57.34	60.37	61.02
EfficientAD	65.27	55.46	68.73	63.30
	59.69	51.04	54.91	55.21

5.2.1 Training and Testing Protocol

As shown in Table 1, the information available during the training phase of this benchmark is consistent with the MVTec AD dataset, i.e., only normal (defect-free) samples are used. However, to simulate the generalization requirement of models for unknown product types in real industrial scenarios, a key distinction is that the use of sub-category labels is prohibited during the testing process. This means that the model must be able to correctly identify anomalies without prior knowledge of which specific sub-category (e.g., cloth, wafer, or metal plate) the test sample belongs to. This setting significantly increases the challenge of the task and aims to evaluate the algorithm's performance under more realistic and open conditions that are closer to actual production.

5.2.2 Data Augmentation

Given that deep learning-based evaluation methods typically require training on large-scale datasets to fully converge, we adopted a systematic data augmentation strategy to expand the training set, preventing overfitting and enhancing model robustness. This strategy applies to both texture and object categories. The specific procedure is as follows: First, all images are resized to fit the model's input dimensions. Then, apart from basic mirror flipping, we do not introduce other complex spatial or color transformations to ensure the authenticity of the augmented data and avoid introducing artifacts similar to real defects. Through this process, we augment the training samples for each category to 10000 images, providing sufficient and diverse normal samples for deep models to learn from.

5.2.3 Evaluation Metric

To ensure fair comparison with mainstream research [4, 38, 5], we adopt the Area Under the Receiver Operating Characteristic Curve (AUROC) as the core evaluation metric. This metric measures the model's overall ability to distinguish between positive and negative samples at different discrimination thresholds, is insensitive to class imbalance, and is thus well-suited for anomaly detection tasks.

Specifically, the evaluation is conducted at two levels:

- Image-Level Anomaly Detection (Image-Level AUROC): Denoted as I-AUROC. This metric evaluates the model's ability to classify an entire image as "normal" or "anomalous". It provides a direct basis for the model's performance in rapid quality screening and sorting on the production line.

- Pixel-Wise Anomaly Localization (Pixel-Wise AUROC): Denoted as P-AUROC. This metric evaluates the model's ability to precisely localize anomalous regions at the pixel level. It offers valuable spatial information for subsequent repair or process analysis and is a key indicator of the algorithm's localization accuracy.

5.3. Result

As shown in Table 2, Table 3, and Table 4, we present the evaluation results of anomaly image classification and anomaly region segmentation for all methods and dataset categories, reflecting the challenges

that current algorithms still face in cross-category generalization and adaptation to complex real-world scenarios, respectively. No method performs consistently well across all texture categories. In the cloth category, SimpleNet outperforms the other methods. This is attributed to its sensitive modeling ability for texture local features. But in the wafer category, DRAEM performs better than SimpleNet, indicating that the method based on joint learning of reconstruction and discrimination has advantages in images of semiconductor wafers with relatively regular structures and variable defect shapes. In the metal plate category, EfficientAD leads in second place by 4.38% in I-AUROC, demonstrating its lightweight architecture's ability to maintain detection speed while being robust to metal surface reflections and complex optical changes. Further analysis of the performance of each subclass reveals that certain defect types and material variations pose significant challenges to the algorithm. For example, in the cloth subclass 5, the Pixel-AUROC of most methods significantly decreases, due to the small defect area and high similarity to background texture in this type of cloth. In the wafer subclass 4, the performance of all methods is generally low, indicating that subtle point-like or contamination defects in a uniform background remain difficult to detect. In the metal plate subclass 3, EfficientAD has a higher Pixel-AUROC, reflecting its strong ability to locate scratches and blemishes on the metal surface. Compared with the MVTec dataset, the average performance of all methods on Texture-AD has significantly decreased, revealing the limitations of existing algorithms in dealing with multiple product specifications, cross-category generalization, and complex imaging conditions in real industrial environments. For instance, intentional differences in lighting and product models between training and testing sets cause algorithms to overly rely on the distribution of training data to easily experience performance degradation. In conclusion, Texture-AD not only assesses the overall detection capability of the algorithm but also reveals its weaknesses through fine-grained category and defect analysis, such as insufficient sensitivity to small defects and limited adaptability to material and lighting changes. These findings suggest that future research can focus on improving the stability of the model in cross-category generalization, small defect detection, and complex imaging conditions, promoting the development of anomaly detection algorithms towards more practical and robust directions. As shown in Figure 9, when applying our dataset Texture-AD for evaluation alongside the MVTec dataset, it was found that the evaluation results of our dataset are generally lower, which can expose the problem domains where the algorithm fails, facilitating targeted optimization of the algorithm's weak points in subsequent improvements. Here are the evaluation results of each method. Some examples of performance were provided in Figure 10. All experimental results are the mean of 3 replicates.

5.4. Conclusion

We introduce the Texture-AD Anomaly Detection Benchmark, a novel dataset for unsupervised anomaly detection that mimics real-world industrial detection scenarios. The dataset provides a way to evaluate unsupervised anomaly detection methods in realistic algorithm development scenarios. Since pixel-accurate ground truth labels of anomaly regions in images are provided, both image-level classification and pixel-level segmentation anomaly detection methods can be evaluated. Several state-of-the-art methods are evaluated on this dataset. The evaluation provided a benchmark for showing how different algorithms perform in real-world application scenarios and indicating that there is still much room for improvement. We hope that the proposed dataset will stimulate the development of new unsupervised anomaly detection methods.

Acknowledgement

This study was funded by the National Natural Science Foundation of China (grant number 62341408).

References

[1] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training, 2018. 8

[2] Y. Bao, K. Song, J. Liu, Y. Wang, Y. Yan, H. Yu, and X. Li. Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. IEEE Transactions on Instrumentation and Measurement, 70:1–11, 2021. 2, 4

[3] K. Batzner, L. Heckler, and R. König. Efficientad: Accurate visual anomaly detection at millisecond-level latencies, 2024. 10

[4] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9584–9592, 2019. 2, 6, 11

[5] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2020. 9, 11

[6] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. CoRR, abs/1807.02011, 2018. 8

[7] L. Chen, Z. You, N. Zhang, J. Xi, and X. Le. Utrad: Anomaly detection and localization with u-transformer. Neural Networks, 147:53–62, 2022. 8

[8] C. Chu, A. Zhmoginov, and M. Sandler. Cyclegan, a master of steganography. CoRR, abs/1712.02950, 2017. 8

[9] A.-S. Collin and C. De Vleeschouwer. Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise. In 2020 25th International Conference on Pattern Recognition (ICPR), pages 7915–7922, 2021. 8

[10] D. Dehaene, O. Frigo, S. Combrexelle, and P. Eline. Iterative energy-based projection on a normal data manifold for anomaly localization, 2020. 8

[11] X. Feng, X. wen Gao, and L. Luo. X-sdd: A new benchmark for hot rolled steel strip surface defects detection. Symmetry, 13:706, 2021. 2, 4

[12] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, 2019. 8

[13] L. Guo, R. Li, B. Jiang, and X. Shen. Automatic crack distress classification from concrete surface images using a novel deep-width network architecture. Neurocomputing, 397:383–392, 2020. 2, 6

[14] J. Hou, Y. Zhang, Q. Zhong, D. Xie, S. Pu, and H. Zhou. Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection, 2021. 8

[15] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022. 8

[16] J. Lei, X. Hu, Y. Wang, and D. Liu. Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow, 2023. 8, 9

[17] Y. Liu, G. Wu, and Z. Lv. Sdgan: A novel spatial deformable generative adversarial network for low-dose ct image reconstruction. Displays, 78:102405, 2023. 8

[18] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang. Simplenet: A simple network for image anomaly detection and localization, 2023. 9

[19] D. Ninja. Visualization tools for industrial optical inspection dataset. https://datasetninja.com/industrial-optical-inspection, aug 2024. visited on 2024-08-14. 1, 3

[20] M. Niu, K. Song, L. Huang, Q. Wang, Y. Yan, and Q. Meng. Unsupervised saliency detection of rail surface defects using stereoscopic images. IEEE Transactions on Industrial Informatics, 17(3):2271–2281, 2021. 2, 4

[21] H. Park, J. Noh, and B. Ham. Learning memory-guided normality for anomaly detection, 2020. 8

[22] P. Perera, R. Nallapati, and B. Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2893–2901, 2019. 8

[23] M. Pourreza, B. Mohammadi, M. Khaki, S. Bouindour, H. Snoussi, and M. Sabokrou. G2d: Generate to detect anomaly, 2020. 8

[24] T. Reiss and Y. Hoshen. Mean-shifted contrastive loss for anomaly detection, 2022. 9

[25] O. Rippel, M. Müller, and D. Merhof. Gan-based defect synthesis for anomaly detection in fabrics. In 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), volume 1, pages 534–540, 2020. 8

[26] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. CoRR, abs/1802.09088, 2018. 8

[27] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee. Multiresolution knowledge distillation for anomaly detection, 2020. 9

[28] Y. Shi, J. Yang, and Z. Qi. Unsupervised anomaly segmentation via deep feature reconstruction. Neurocomputing, 424:9–22, 2021. 8

[29] J. Silvestre-Blanes, T. Albero-Albero, I. Miralles, R. Pérez-Llorens, and J. Moreno. A public fabric database for defect detection methods and results. Autex Research Journal, 19, 06 2019. 1, 3

[30] G. Song, K. Song, and Y. Yan. Saliency detection for strip steel surface defects using multiple constraints and improved texture features. Optics and Lasers in Engineering, 128:106000, 2020. 2, 4

[31] Texture-ad. Texture-ad-benchmark. https://huggingface.co/datasets/texture-ad/Texture-AD-Benchmark, 2024. Accessed: 2024-08-15. 2

[32] G. Wang, S. Han, E. Ding, and D. Huang. Student-teacher feature pyramid matching for anomaly detection, 2021. 9

[33] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. IEEE Transactions on Semiconductor Manufacturing, 28(1):1–12, 2015. 2, 4

[34] Y. Xia, Y. Zhang, F. Liu, W. Shen, and A. Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation, 2020. 8

[35] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui, and X. Chen. Automatic bridge crack detection using a convolutional neural network. Applied Sciences, 9:2867, 07 2019. 2, 6

[36] X. Yan, H. Zhang, X. Xu, X. Hu, and P.-A. Heng. Learning semantic context from normal samples for unsupervised anomaly detection. In AAAI Conference on Artificial Intelligence, 2021. 8

[37] Z. You, K. Yang, W. Luo, L. Cui, Y. Zheng, and X. Le. Adtr: Anomaly detection transformer with feature reconstruction, 2022. 8

[38] M. Z. Zaheer, J. ha Lee, M. Astrid, and S.-I. Lee. Old is gold: Redefining the adversarially learned one-class classifier training paradigm, 2020. 8, 11

[39] V. Zavrtanik, M. Kristan, and D. Skocaj. Dræm - A discriminatively trained reconstruction embedding

for surface anomaly detection. CoRR, abs/2108.07610, 2021. 8, 9

[40] K. Zhang, B. Wang, and C. J. Kuo. Pedenet: Image anomaly localization via patch embedding and density estimation. CoRR, abs/2110.15525, 2021. 8

[41] Z. Zhang, S. Yu, S. Yang, Y. Zhou, and B. Zhao. Rail-5k: a real-world dataset for rail surface defects detection. CoRR, abs/2106.14366, 2021. 2, 5

[42] X. Zhao, J. Shi, Q. Yin, Z. Dong, Y. Zhang, L. Kang, Q. Yu, C. Chen, J. Li, X. Liu, and K. Zhang. Controllable synthesis of high-quality two-dimensional tellurium by a facile chemical vapor transport strategy. iScience, 25(1):103594, 2022. 2, 4

[43] K. Zhou, Y. Xiao, J. Yang, J. Cheng, W. Liu, W. Luo, Z. Gu, J. Liu, and S. Gao. Encoding structure-texture relation with p-net for anomaly detection in retinal images. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, Computer Vision – ECCV 2020, pages 360–377, Cham, 2020. Springer International Publishing. 8

[44] Y. Zhou, X. Xu, J. Song, F. Shen, and H. T. Shen. Msflow: Multi-scale flow-based framework for unsupervised anomaly detection, 2023. 10