# VBAD: Industrially Scalable Anomaly Detection

Anonymous WACV Algorithms Track submission

Paper ID 2107

## Abstract

*Anomaly detection is crucial in various industrial applications such as manufacturing where identification of defects or abnormal patterns is critical. While deep neural network (DNN) models have significantly advanced anomaly detection performance, they often accomplish it with high computational costs and memory issues. This makes them less practical for real-time large-scale industrial applications. In this paper, we propose a novel approach that addresses these challenges by offering a scalable and efficient alternative to DNN-based models. Our method eliminates the dependency on deep neural network (DNNs) training, utilizing a binning technique that significantly reduces computational complexity and time while maintaining competitive performance. We validate our approach on the semiconductor photo mask dataset, MVTec-AD, and VisA datasets. Photo mask dataset is highly challenging that requires precise anomaly detection across millions of high-resolution image patches which is nearly 40 GB. The experimental results on three challenging datasets demonstrate that our method outperforms existing methods in terms of computational efficiency while maintaining excellent anomaly detection performance, particularly when applied to industrial-scale data. Our approach offers substantial advantages in applications where real-time processing and scalability are critical, making it an ideal candidate for deployment in industrial settings.*

## 1. Introduction

Anomaly detection is crucial in wide range of applications, from industrial inspection [4] to medical imaging [25, 29] and autonomous driving [13, 16], where identifying defects or abnormality is important. Recent advancements in deep neural networks (DNNs) [3, 24] have led to significant improvements in the accuracy of anomaly detection models. These models leverage deep feature representations learned from large-scale datasets and demonstrate notable performance. However, this increase in accuracy often comes at the expense of substantially higher computational costs. The reliance on deep neural network learning results in significantly longer training and inference times, posing challenges for real-time applications and large-scale industrial environments where both speed and scalability are essential.

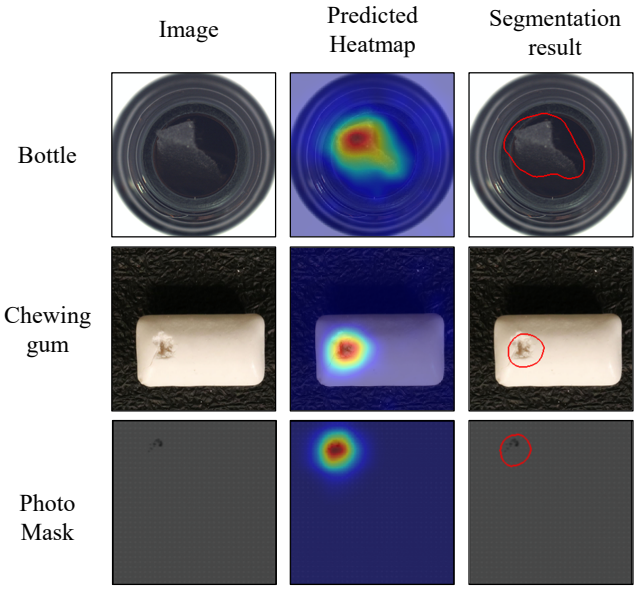


Figure 1. Anomaly detection results generated by our approach on various test images from the MVTec-AD [5], VisA [36] and semiconductor photo mask image dataset. Each row represents a different object category. The first column shows anomalous images whereas the second and third columns show the heatmaps and final anomalous region predictions of our approach respectively.

While methods such as PaDiM [9] and PatchCore [19] have been developed to reduce the training time by utilizing pre-trained models and memory banks [26], they still face substantial challenges in terms of computational efficiency. These methods tend to have high time complexity and large memory usage particularly when dealing with industrial-scale datasets during training or testing. As the dataset size increases, the efficiency of the models drops exponentially,

particularly on resource-constrained hardware common in industrial settings.

In this paper, we introduce a novel anomaly detection approach designed to address the mentioned challenges by providing high performance with efficiency without the need for deep neural network training [7]. Our approach, named Vector Binning Anomaly Detection (VBAD), is specifically optimized for low time complexity, allowing it to handle industrial-scale datasets efficiently while maintaining high accuracy. By eliminating the dependency on DNNs training and optimizing the algorithmic structure [23], our approach reduces the computational burden, enabling its deployment in real-time industrial applications without the risk of memory overflow or performance degradation.

To validate our approach, we conducted extensive experiments on widely popular MVTec-AD and VisA datasets. Moreover, we also tested our approach on a proprietary large-scale Semiconductor Photo Mask dataset. The semiconductor mask dataset presents unique challenges due to its extremely high resolution and the need for precise anomaly detection across millions of small, detailed patches. Our method not only demonstrated competitive performance in the standard benchmarks but also showcased significant advantages in computational efficiency and scalability when applied to the semiconductor mask dataset. Despite the large scale of the data, our method maintained high performance while operating faster than other leading models, particularly in scenarios with massive image counts. The primary contributions of this paper are summarized as follows:

- We propose a DNN-training-free novel anomaly detection approach, which achieves high accuracy with significantly reduced computational complexity and time.
- We provide extensive analysis of the scalability and industrial-scale applicability of the anomaly detection approaches. Our approach is suitable for real-time industrial applications.
- Our method is validated on MVTec-AD, VisA, and semiconductor photo mask datasets, demonstrating its effectiveness in anomaly detection.

## 2. Related Works

Visual anomaly detection is a critical area of research with various applications, such as medical imaging, autonomous driving, and industrial inspection [12, 18]. This section outlines some key methodologies and approaches.

### 2.1. Reconstruction-Based Methods

Reconstruction-based approaches [17, 32, 33] typically involve training deep models, such as autoencoders [5, 6, 34, 35], to reconstruct input images. These models are expected to accurately reconstruct normal data while failing to do so for anomalous data. The difference between an original and its corresponding reconstructed image serves as the basis for detecting anomalies. This method, however, can be limited by the quality of the reconstruction [2], particularly when subtle differences are challenging to capture. Such approaches are fundamentally different from our approach as we aim to avoid training a deep neural network and utilize only a pre-trained model without training or fine-tuning.

### 2.2. Embedding-Based Methods

These methods leverage pre-trained neural networks to extract and compress features into a compact space [9, 15]. Anomalous data, which do not conform to the learned normal patterns, stand out as outliers in this feature space. Techniques such as memory banks [14, 19, 28] store representative normal features, which are then compared to new data to identify anomalies through metric learning. One-class classification methods further refine [1] this by establishing explicit boundaries around normal data clusters, such as hyperplanes [22] or hyperspheres [20]. Like our approach, some of these methods also aim to reduce computational complexity by avoiding the training of deep neural networks.

## 3. Method

In this section, we introduce our anomaly detection approach, which is designed to efficiently detect anomalies while maintaining low time complexity. The proposed method consists of three key stages: feature extraction, transition matrix calculation, and distance matrix calculation. These stages work together to produce a final anomaly map that highlights the anomalous regions within an image.

### 3.1. Feature Extraction

Given an input image $I$ of size $H \times W$, we first extract feature embeddings using a pre-trained convolutional neural network $\Phi$ [1, 8, 10]. Each image passes through $\Phi$, and we collect feature maps from a specific intermediate layer.

These feature maps are referred to as embedding vectors $E$, where each vector corresponds to a patch of the input image. Thus, we represent the embedding vector at location $(X_i, Y_j)$ as $E_{i,j} \in \mathbb{R}^C$, where $i, j$ index the spatial locations within the feature map.

To enhance the time efficiency of our anomaly detection method, we apply a binning technique to the embedding vectors. Binning helps in reducing the dimensionality of the feature space and allows for more efficient computation of transition matrices.

We apply binning to group the continuous values in each channel into discrete bins. Let $b_k$ denote the bin value for

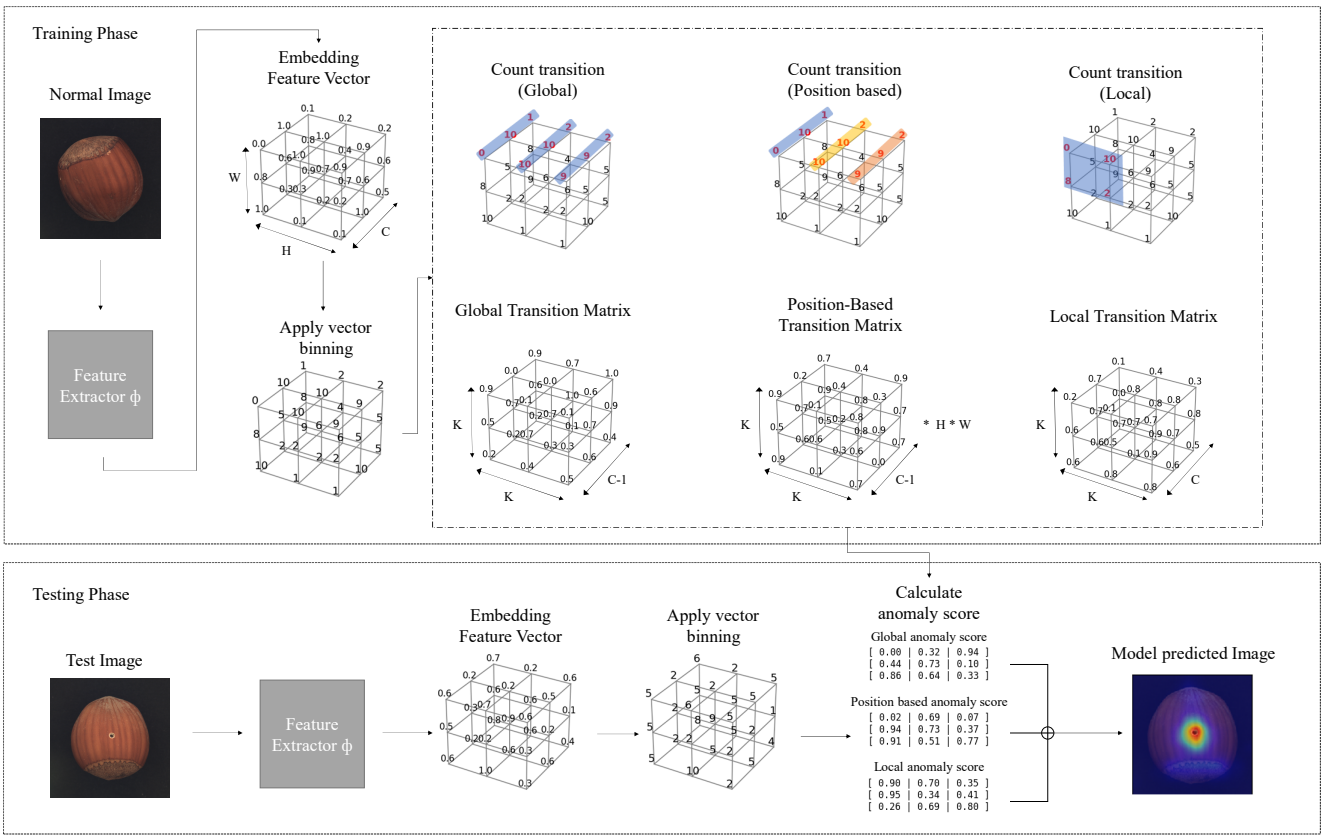


Figure 2. The architecture of the proposed Vector Binning Anomaly Detection (VBAD) method is illustrated. The process begins with extracting the embedding vector of input images, which are passed through a pre-trained convolutional neural network ($\Phi$). These vectors are utilized to compute the transition matrix during the training phase after applying vector binning. In the testing phase, these transition matrices are used to calculate distance matrices, which calculate the final anomaly score map. This anomaly map visually highlights regions in the image that deviate from normal patterns, allowing for precise localization of defects or anomalies. The proposed method emphasizes efficiency in computation while maintaining high accuracy, making it suitable for large-scale industrial applications.

the $k$-th bin. The binning process for each element $E_{i,j}$ of the embedding vector in channel $c$ is defined as:

$$B_{i,j} \mapsto b_k \quad \text{if} \quad b_k \leq E_{i,j} < b_{k+1} \tag{1}$$

$B_{i,j}$ is discrete vector as location $(X_i, Y_j)$. This process maps each element of the embedding vector to a discrete bin, reducing the continuous feature space to a finite set of states. The number of bins $K$ is a hyperparameter that controls the granularity of the binning process.

After binning, each embedding vector $E_{i,j}$ is represented by a discrete vector $B_{i,j} \in \{b_1, b_2, \ldots, b_K\} \in \mathbb{R}^C$. Next, we calculate the transition counts $N_{b_1,b_2}^{(c,c+1)}$ used in the transition matrices. The count $N_{b_1,b_2}^{(c,c+1)}$ represents the number of transitions observed from $b_1$ to $b_2$ in channel $c$ to in channel $c+1$, across all spatial locations $(i, j)$:

$$N_{b_1,b_2,i,j}^{(c,c+1)} = \sum_{m=1}^{M} \delta((B_{m,c,i,j} = b_1) \wedge (B_{m,c+1,i,j} = b_2)) \tag{2}$$

Here, $\delta(\cdot)$ is the Kronecker delta function, which equals 1 when the condition is true and 0 otherwise, and M is the number of images in the dataset. The summation is performed over all spatial positions $(i, j)$ in the feature map. The count $N_{b_1,b_2,i,j}^{(c,c+1)}$ is then used to compute the transition matrices.

### 3.2. Transition Matrix Calculation

To effectively model the transitions between different states in the feature embedding space, we calculate three types of transition matrices: global transition matrices, position-based transition matrices, and local transition matrices. These matrices capture both inter-channel dependencies and spatial relationships within the feature maps, which

are crucial for accurately identifying anomalies.

#### 3.2.1 Global Transition Matrix

The global transition matrix captures the transitions between states across different channels over all spatial positions. The transition probability $P$ from from bin $b_1$ in channel $c$ to bin $b_2$ in channel $c+1$, aggregated over all spatial positions, is given by:

$$P^{glb}_{c,b1,b2} = \frac{\sum_{x=i}^{X}\sum_{y=j}^{Y} N^{(c,c+1)}_{b_1,b_2,i,j}}{\sum_{x=i}^{X}\sum_{y=j}^{Y}\sum_{p=b_1}^{b_K}\sum_{q=b_1}^{b_K} N^{(c,c+1)}_{p,q,i,j}} \tag{3}$$

$$P^{glb}_{c} = \begin{bmatrix} P_{c,b_1,b_1} & P_{c,b_1,b_2} & \cdots & P_{c,b_1,b_k} \\ P_{c,b_2,b_1} & P_{c,b_2,b_2} & \cdots & P_{c,b_2,b_k} \\ \vdots & \vdots & \ddots & \vdots \\ P_{c,b_k,b_1} & P_{c,b_k,b_2} & \cdots & P_{c,b_k,b_k} \end{bmatrix} \tag{4}$$

Here, $P^{glb}_{c,b1,b2}$ represents the probability of transitions from bin $b_1$ in channel $c$ to bin $b_2$ in channel $c+1$ across all positions $i, j$. The summation over all $i, j$ ensures that the transition matrix accounts for global patterns across the entire feature map. Finally, $P^{glb}_{c}$ is calculated as Global Transition Matrix in channel $c$. So total global transition matrix is $P^{glb} \in \mathbb{R}^{(C-1)\times K\times K}$.

#### 3.2.2 Position-Based Transition Matrix

The position-based transition matrix refines the global transition matrix by considering transitions at specific spatial positions. At a specific position $(i, j)$, the transition probability from state $i$ in channel $c$ to state $j$ in channel $c+1$ is defined as:

$$P^{pos}_{c,b1,b2,i,j} = \frac{N^{(c,c+1)}_{b_1,b_2,i,j}}{\sum_{p=b_1}^{b_K}\sum_{q=b_1}^{b_K} N^{(c,c+1)}_{p,q,i,j}} \tag{5}$$

$$P^{pos}_{c,i,j} = \begin{bmatrix} P^{pos}_{c,b_1,b_1,i,j} & P^{pos}_{c,b_1,b_2,i,j} & \cdots & P^{pos}_{c,b_1,b_k,i,j} \\ P^{pos}_{c,b_2,b_1,i,j} & P^{pos}_{c,b_2,b_2,i,j} & \cdots & P^{pos}_{c,b_2,b_k,i,j} \\ \vdots & \vdots & \ddots & \vdots \\ P^{pos}_{c,b_k,b_1,i,j} & P^{pos}_{c,b_k,b_2,i,j} & \cdots & P^{pos}_{c,b_k,b_k,i,j} \end{bmatrix} \tag{6}$$

$$P^{pos}_{c} = \begin{bmatrix} P^{pos}_{c,1,1} & P^{pos}_{c,1,2} & \cdots & P^{pos}_{c,1,Y} \\ P^{pos}_{c,2,1} & P^{pos}_{c,2,2} & \cdots & P^{pos}_{c,2,Y} \\ \vdots & \vdots & \ddots & \vdots \\ P^{pos}_{c,X,1} & P^{pos}_{c,X,2} & \cdots & P^{pos}_{c,X,Y} \end{bmatrix} \tag{7}$$

In this case, $P^{pos}_{c,b1,b2,i,j}$ represents the probability of position-based transitions from bin $b_1$ in channel $c$ to bin $b_2$ in channel $c+1$ at poss $(i, j)$. Unlike the global matrix, this transition matrix is normalized at each spatial position $(i, j)$, allowing the model to detect anomalies that manifest as spatially localized deviations. Finally, $P^{pos}_{c,i,j}$ is calculated as Position-based Transition Matrix in channel $c$ at positions $(i, j)$. So total position-based transition matrix is $P^{pos} \in \mathbb{R}^{(C-1)\times K\times K\times H\times W}$.

#### 3.2.3 Local Transition Matrix

The local transition matrix focuses on transitions within the same channel but across neighboring spatial positions. At a given position $(i, j)$ in channel $c$, the matrix models transition between the position at $(i, j)$ and the neighboring positions within a tolerance $\pm t$. Let $\Delta i$ and $\Delta j$ denote the relative offsets from the central position $(i, j)$ within the tolerance $t$. So, to calculate the local transition matrix, we count discrete vector $B_{i,j}$ as a different method.

$$N^{c}_{b_1,b_2,i,j} = \sum_{m=1}^{M}\sum_{\Delta i=-t}^{t}\sum_{\Delta j=-t}^{t} \delta((B_{m,c,i,j} = b_1) \wedge (B_{m,c,i+\Delta i,j+\Delta j} = b_2)) \tag{8}$$

$$P^{loc}_{c,b1,b2} = \frac{\sum_{x=i}^{X}\sum_{y=j}^{Y} N^{c}_{b_1,b_2,i,j}}{\sum_{x=i}^{X}\sum_{y=j}^{Y}\sum_{p=b_1}^{b_K}\sum_{q=b_1}^{b_K} N^{c}_{p,q,i,j}} \tag{9}$$

$$P^{loc}_{c} = \begin{bmatrix} P^{loc}_{c,b_1,b_1} & P^{loc}_{c,b_1,b_2} & \cdots & P^{loc}_{c,b_1,b_k} \\ P^{loc}_{c,b_2,b_1} & P^{loc}_{c,b_2,b_2} & \cdots & P^{loc}_{c,b_2,b_k} \\ \vdots & \vdots & \ddots & \vdots \\ P^{loc}_{c,b_k,b_1} & P^{loc}_{c,b_k,b_2} & \cdots & P^{loc}_{c,b_k,b_k} \end{bmatrix} \tag{10}$$

Here, $P^{loc}_{c,b1,b2}$ represents the probability of transitions from bin $b_1$ in channel $c$ to bin $b_2$ in channel $c+1$ across all positions $i, j$. The summation over all $i, j$ ensures that the transition matrix accounts for global patterns across the entire feature map. Finally, $P^{loc}_{c}$ is calculated as Local Transition Matrix in channel $c$. So local global transition matrix is $P^{loc} \in \mathbb{R}^{C\times K\times K}$.

By structuring the transition matrices in this way, we can effectively model both global and local dependencies, providing a robust mechanism for detecting anomalies in images.

### 3.3. Distance Matrix Calculation

Once the transition matrices are established, the next crucial step is calculating the distance matrix, which quantifies deviations from the normal patterns captured by the transition matrices. These distance matrices are essential for generating the final anomaly score map.

Given an input image, the embedding vector $E_{i,j}$ is first discretized into $B_{i,j}$ using the predefined binning process. The $B_{i,j}$ corresponding to each embedding vector is then used to compute three types of distance matrices: Global, Position-Based, and Local. These matrices account for transitions across different channels (global transitions), specific spatial positions (position-based transitions), and neighboring spatial positions within the same channel (local transitions).

The global transition matrix captures transitions between states across different channels over all spatial positions. The global distance matrix $D^{glb}(i,j)$ is calculated by computing the log-probability of these transitions, defined as:

$$D^{glb}_{i,j} = \sum_{c=1}^{C-1} |\log(P^{glb}_{c}(B_{c,i,j}, B_{c+1,i,j}))| \tag{11}$$

where $P^{glb}_{c}(B_{c,i,j}, B_{c+1,i,j})$ is the transition probability calculated by using discrete vector at $c$-th channel and $(i,j)$ positions. And by summation all channel's $P^{glb}_{c}(B_{c,i,j}, B_{c+1,i,j})$, calculate global distance value at position $(i,j)$, denote as $D^{glb}_{i,j}$.

The position-based transition matrix refines the global transition matrix by focusing on transitions at specific spatial positions. The position-based distance matrix $D^{pos}_{i,j}$ is computed as:

$$D^{pos}_{i,j} = \sum_{c=1}^{C-1} |\log(P^{pos}_{c,i,j}(B_{c,i,j}, B_{c+1,i,j}))| \tag{12}$$

where $P^{pos}_{c,i,j}(B_{c,i,j}, B_{c+1,i,j})$ is the transition probability calculated by using discrete vector at $c$-th channel and $(i,j)$ positions. And by summation all channel's $P^{pos}_{c,i,j}(B_{c,i,j}, B_{c+1,i,j})$, calculate position-based distance value at position $(i,j)$, denote as $D^{pos}_{i,j}$.

The local transition matrix captures transitions within the same channel but across neighboring spatial positions. The local distance matrix $D^{loc}_{i,j}$ is defined by the log-probability of these local transitions:

$$D^{loc}_{i,j} = \sum_{c=1}^{C} \sum_{\Delta i=-t}^{t} \sum_{\Delta j=-t}^{t} |\log(P^{loc}_{c}(B_{c,i,j}, B_{c,i+\Delta i,j+\Delta j}))| \tag{13}$$

where $P^{loc}_{c}(B_{c,i,j}, B_{c,i+\Delta i,j+\Delta j})$ is the transition probability calculated by using discrete vector at $c$-th channel and $(i,j)$ positions. And by summation all channel's $P^{loc}_{c}(B_{c,i,j}, B_{c,i+\Delta i,j+\Delta j})$, calculate global distance value at position $(i,j)$, denote as $D^{loc}_{i,j}$.

The overall distance matrix $D_{i,j}$ for each pixel is obtained by aggregating the contributions from all three distance matrices:

$$D_{i,j} = D^{glb}_{i,j} + D^{pos}_{i,j} + D^{loc}_{i,j} \tag{14}$$

This aggregation ensures that the method captures both global and local deviations effectively. The final anomaly score map is generated by summing the distance matrices across all channels and spatial positions, with higher scores indicating stronger deviations from normal patterns.

By integrating information from the global, position-based, and local transitions, the proposed method provides a comprehensive mechanism for detecting anomalies, accounting for both broad and localized deviations within the image data.

# 4. Experiments

## 4.1. Experimental Setup

All experiments were conducted on a machine equipped with an Intel Core i9-13900KS CPU, 128GB of RAM, and an NVIDIA GeForce RTX 4090 GPU. For the computational efficiency, each compared model is trained and tested on this setup to ensure fairness.

**Datasets:** For our experiments, we used three datasets including private industrial dataset of semiconductor images, MVTec [5] and VisA [36]. Each of these datasets used is explained next. Semiconductor Photo Mask dataset is a large-scale privately owned dataset by *Omitted Due to Blind Review*. The images in this dataset have a resolution of 172,000 $\times$ 249,000 pixels with a precision of 0.8837 pixels per micrometer. When these images are divided into smaller patches of 200 $\times$ 200 pixels, a total of 1,070,700 images are generated. This dataset is currently not available publicly, however, we are hoping to release it publicly in the future. We have provided a sample from the dataset as Supplementary. MVTec dataset consists of 15 categories, comprising 3,629 training images and 1,725 test images. The training dataset contains only normal images, while the test dataset includes both normal and anomalous images. Additionally, the test dataset provides ground truth masks for anomaly segmentation evaluation. VisA dataset consists of 12 categories, comprising 9,621 training images and 1,200 test images. The training dataset contains only normal images, while the test dataset includes both normal and anomalous images.

**Augmentations:** We applied a few standard augmentation techniques to enhance the training images. These include rotation, gamma correction, Gaussian noise, and ISO noise. Each image was first resized to 256 $\times$ 256 pixels and then center-cropped to 224$\times$224 pixels for training. This preprocessing step ensured that the input images were standardized across all experiments.

**Feature Extraction:** We employed the Wide-Resenet-101 [30] model pre-trained on ImageNet [21] as a feature extractor. This model was selected for its popularity in capturing

| Class | | Reconstruction based | | | | Feature embedding based | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | AE-SSIM | RIAD | DRAEM | CutPaste | SimpleNet | CFA | PaDiM | PatchCore | Ours |
| DNN training | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| Object class | Bottle | 93 | 99.9 | 99.2 | 98.2 | **100** | **100** | 99.1 | **100** | **100** |
| | Cable | 82 | 81.9 | 91.8 | 81.2 | **99.9** | 99.8 | 97.1 | 99.5 | 93.4 |
| | Capsule | 94 | 88.4 | **98.5** | 98.2 | 97.7 | 97.3 | 87.5 | 98.1 | 92.9 |
| | Hazelnut | 97 | 83.3 | **100** | 98.3 | **100** | 100 | 99.4 | **100** | 95.4 |
| | Metal Nut | 89 | 88.5 | 98.7 | 99.9 | **100** | **100** | 96.2 | **100** | **100** |
| | Pill | 91 | 83.8 | 98.9 | 94.9 | **99** | 97.9 | 90.1 | 96.6 | 96.9 |
| | Screw | 96 | 84.5 | 93.9 | 88.7 | **98.2** | 97.3 | 97.5 | 98.1 | 86.7 |
| | Toothbrush | 92 | **100** | **100** | 99.4 | 99.7 | **100** | **100** | **100** | 96.1 |
| | Transistor | 90 | 90.9 | 93.1 | 96.1 | **100** | **100** | 94.4 | **100** | 99.5 |
| | Zipper | 88 | 98.1 | **100** | 99.9 | 99.9 | 99.6 | 98.6 | 99.4 | 97 |
| Texture class | Carpet | 87 | 84.2 | 97 | 93.9 | 99.7 | 97.3 | **99.8** | 98.7 | 99.1 |
| | Grid | 94 | 99.6 | 99.9 | **100** | 99.7 | 99.2 | 96.7 | 98.2 | 97.8 |
| | Leather | 78 | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| | Tile | 59 | 98.7 | 99.6 | 94.6 | **99.8** | 99.4 | 98.1 | 98.7 | 99.1 |
| | Wood | 73 | 93 | 99.1 | 99.1 | **100** | 99.7 | 99.2 | 99.2 | 98.7 |
| Time | | - | - | - | - | 7353.09 | 2501.97 | 1170.76 | 240.47 | **189.03** |

Table 1. Comparison of Image ROCAUC scores across all classes of the MVTec-AD dataset. For each class, the best-performing method is shown in bold . The time taken for training and testing all classes of the MVTec-AD dataset is shown in Time (seconds). The method in green is the fastest to train and test whereas red is the second fastest. It is well-established that reconstruction-based models require significantly longer training and testing times compared to feature embedding-based models. Therefore, we did not include these models in our evaluation.

rich feature representations, which are critical for effective anomaly detection. The feature embeddings extracted from this network are used for calculating the transition matrices proposed in our anomaly detection approach.

### 4.2. Evaluation on Semiconductor Photo Mask Dataset

We evaluated the performance of our proposed method on the Semiconductor Photo Mask dataset, which presents unique challenges due to its extremely high resolution and intricate patterns. Given the massive scale and high precision required for anomaly detection in this dataset, traditional methods, while accurate, tend to suffer from high computational complexity and longer processing times. Our method, while potentially yielding slightly lower accuracy in some cases, performs better with large-scale datasets due to its lower time complexity and scalability. Tab. 3 shows the image-level and pixel-level ROCAUC scores for various methods applied to the semiconductor mask dataset. Despite the scale of the dataset, our method achieves competitive performance while offering significant computational advantages.

### 4.3. Evaluation on MVTec and VisA Datasets

Tab. 1 presents the ROCAUC performance comparison of various anomaly detection methods on MVTec-AD [5] dataset.

| | DRAEM | SimpleNet | CFA | PatchCore | Ours |
|---|---|---|---|---|---|
| DNN | ✓ | ✓ | ✓ | ✗ | ✗ |
| Candle | 94.4 | **98.7** | 97.2 | **98.7** | 97.8 |
| Capsules | 76.3 | **89.9** | 85.2 | 68.8 | 66.5 |
| Cashew | 90.7 | 97.5 | 96.2 | **97.7** | 91.9 |
| Gum | 94.2 | **99.8** | 98.9 | 99.1 | 99.1 |
| Fryum | 97.4 | **98.1** | 94.3 | 91.6 | 88.8 |
| Macaroni1 | 95 | **99.4** | 91.6 | 90.1 | 86.7 |
| Macaroni2 | **96.2** | 82.4 | 80.3 | 63.4 | 70.4 |
| PCB1 | 54.8 | **99** | 98 | 96 | 92.8 |
| PCB2 | 77.8 | **99.1** | 98.8 | 95.1 | 87.4 |
| PCB3 | 94.5 | **98.5** | 97.7 | 93 | 87.9 |
| PCB4 | 93.4 | 99.6 | 99.8 | 99.5 | 99.0 |
| Pipe Fryum | 99.4 | **99.7** | 99.4 | 99 | 95.3 |
| Time | - | 11065.93 | 3215.92 | 891.07 | **263.49** |

Table 2. Comparison of Image ROCAUC scores across all classes of the VisA [36] dataset. DNN represents whether methods require DNN training. For each class, the best-performing method is shown in bold. The time taken for training and testing all classes of the VisA dataset is shown in Time (seconds). The method in green is the fastest to train and test whereas red is the second fastest. It is well-established that reconstruction-based models require significantly longer training and testing times compared to feature embedding-based models. Therefore, we did not include these models in our evaluation.

**1. Texture Classes:** In texture-based categories such as Carpet and Tile, VBAD performs comparably to the SOTA methods like SimpleNet [15] and PatchCore [19]. This suggests that our method can effectively handle complex tex-

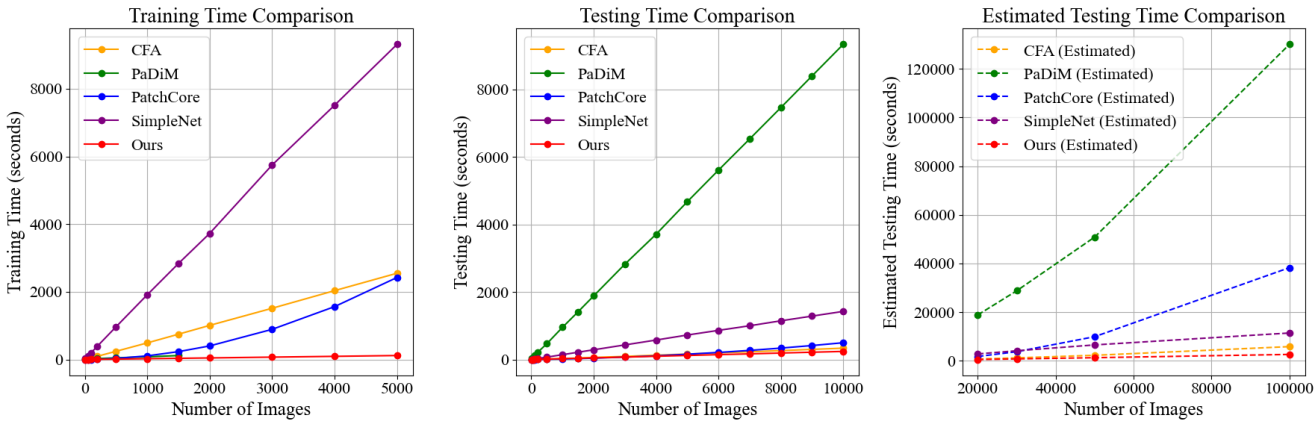


Figure 3. A comparative analysis of training and testing times for different anomaly detection methods (CFA, PaDiM, PatchCore, SimpleNet, and Ours) as a function of the number of images using augmented MVTed-AD [5] dataset. The left graph displays the training time required by each method, showing that Ours achieves significantly faster training compared to deep learning-based methods like SimpleNet [15] which shows a steep increase in time as the dataset grows. The center graph compares the testing time, with Ours demonstrating a linear growth in time, highlighting its efficiency and scalability in contrast to the exponential growth seen in PatchCore [19] and PaDiM [9]. The right graph provides an estimation of testing times when applied to a large-scale dataset, emphasizing Ours scalability. This analysis underscores Ours suitability for real-time industrial applications where large volumes of data must be processed efficiently.

| Metric | PaDiM | PatchCore | SimpleNet | CFA | Ours |
|---|---|---|---|---|---|
| Image ROCAUC | 96.4 | 97.3 | **98.4** | 97.2 | 96.9 |
| Pixel ROCAUC | 97.9 | 98.3 | **98.4** | 98.1 | **98.4** |
| Testing time | 9337.51 | 497.71 | 1426.5 | 337.27 | **240.87** |

Table 3. Comparison of Image ROCAUC, Pixel ROCAUC, and Testing time (seconds) for 10,000 images on the Semiconductor Photo Mask Dataset. Our proposed method achieves the best performance in terms of testing time while maintaining competitive ROCAUC scores for both image and pixel levels.

tures, likely due to its ability to capture subtle transitions between feature maps.

**2. Object Classes:** VBAD achieves perfect scores in several object-based categories, such as Bottle and Metal Nut, performing on par with SOTA methods like PatchCore [19] and SimpleNet [15]. This indicates that our method excels at detecting anomalies in well-defined, structured objects. However, in more challenging object classes like Screw and Capsule, VBAD falls behind top performers, suggesting room for improvement when handling finer details in object shapes or textures.

Similar trends are observed in Tab. 2 where a comparison of our approach with existing SOTA methods is provided on VisA dataset. Overall, VBAD performs on par with existing approaches such as PaDiM [9] and CFA [14], but does not always achieve the highest scores. However, the trade-off lies in computational efficiency, VBAD providing significant advantages over the compared methods (highlighted as green while the second fastest method is highlighted as red). More on computational efficiency is discussed in the subsequent section.

### 4.4. Speed and Computational Efficiency

In real-world industrial applications, the time efficiency of anomaly detection methods is as critical as their accuracy. As shown in Fig. 3, we compared the training and testing times of several anomaly detection methods, including our proposed method, across various dataset sizes. To carry out these experiments, we artificially generated over 10,000 images by randomly augmenting the original images to observe the impact of large-scale datasets on the compared methods.

This study focuses solely on feature embedding-based methods, as reconstruction-based methods typically incur high computational costs during training and testing. Reconstruction-based methods such as DRAEM [31] and RIAD [32] require substantial computational resources, as they rely on training deep neural networks and performing reconstruction tasks for anomaly detection. In contrast, our method eliminates the need for training a neural network, enabling faster anomaly detection with fewer resources.

The feature-embedding-based methods such as SimpleNet [15], PaDiM [9], and CFA [14], while generally less computationally expensive than reconstruction-based methods, still require either complex feature extraction steps or pre-trained models with memory banks and struggle to scale on industrial scale data. Compared to these, our method achieves similar or better accuracy, while exhibiting superior computational efficiency. As highlighted in Fig. 3, our method showcases a nominal linear increase in testing time as the dataset size increases, whereas methods like Patch-

Core experience exponential growth in computational time, particularly when the dataset exceeds 5,000 images.

Additionally, our method's testing time scales efficiently with larger datasets, making it suitable for real-time anomaly detection tasks. Fig. 3 further illustrates the estimated result when dataset size increases significantly over 10,000 images, our approach maintains computational efficiency by several orders of magnitude. The results are generated by using polynomial method, and this results indicate that while other methods may achieve slightly higher accuracy, the trade-off between accuracy and computational efficiency positions our method as a highly practical solution for large-scale anomaly detection in industrial settings.

From these results, it is evident that while other methods may offer slightly higher accuracy in certain scenarios, our approach provides a significant advantage in terms of computational efficiency. This makes it particularly well-suited for applications where enormous datasets, like those in semiconductor manufacturing, are common. The ability to maintain high performance while reducing processing time can lead to more practical and scalable deployment in industrial environments.

| Backbone Architecture | Image / Pixel ROCAUC |
|---|---|
| ResNet18 | 93.4 / 95.8 |
| ResNet50 | 95.0 / 96.3 |
| ResNeXt50_32x4d | 95.1 / 96.6 |
| ResNeXt101_32x8d | 96.4 / 96.6 |
| Wide_ResNet50 | 96.0 / 96.7 |
| Wide_ResNet101 | **96.8 / 97.0** |

Table 4. Performance comparison of different backbone architectures tested on the MVTec-AD dataset. The results show the Image ROCAUC and Pixel ROCAUC scores for each architecture.

## 5. Additional Analysis

**On using different backbone architectures:** Results in Tab. 4 show that deeper networks like ResNeXt101 $32 \times 8d$ [27] and wider networks like Wide ResNet 101 outperform smaller architectures like ResNet 18 [11]. The Wide ResNet [30] models, especially Wide ResNet 101, provide the best balance of capacity and feature extraction capabilities for anomaly detection tasks.

**On varying input sizes:** In this experiment, images were resized to three different resolutions ($128 \times 128$, $256 \times 256$, and $512 \times 512$), and corresponding embeddings were obtained using center-cropped images at $112 \times 112$, $224 \times 224$, and $448 \times 448$ resolutions. The results are reported for both Image and Pixel ROCAUC metrics, providing insights into the model's performance at different input scales. The model achieves its highest Image ROCAUC (96.8) with $256 \times 256$ images and the best Pixel ROCAUC (97.4) with $512 \times 512$ images, indicating that higher resolutions capture finer anomalies. Due to space limitation, we provide extended analysis of these experiments in the Supplementary

| Category | Binning Configuration | |
|---|---|---|
| | Image ROCAUC | Pixel ROCAUC |
| 0/0.6/0.1 | 96.7 | **97** |
| 0/0.8/0.1 | 96.4 | **97** |
| 0/1.1/0.2 | **96.8** | 96.8 |
| 0/1.1/0.05 | 96.2 | **97** |
| 0/1.1/0.1 | **96.8** | **97** |
| 0/1.1/0.15 | **96.8** | 96.9 |
| 0/1.6/0.1 | 96.6 | **97** |
| 0/2.1/0.1 | 96.5 | **97** |

Table 5. Impact of different binning configurations on Image ROCAUC and Pixel ROCAUC. Each configuration is denoted by the range (start/end/interval) used for binning.

**Exploring Binning Configurations:** In this analysis, we examine how different binning configurations impact the performance of the anomaly detection model in terms of Image ROCAUC and Pixel ROCAUC. Tab. 5 presents results on several binning configurations where the first number represents the start of the binning range, the second number represents the maximum (excluded from binning), and the third number indicates the interval used to create the bins. Binning with a moderate range and interval (e.g., '0/1.1/0.1' or '0/1.1/0.2') provides a good balance, capturing key features effectively for both image-level and pixel-level anomaly detection. Configurations with overly small or large ranges or intervals show a slight decline in performance, as they either struggle to capture sufficient details or introduce too much noise. These results highlight the importance of selecting appropriate binning parameters to maximize anomaly detection performance.

## 6. Conclusion

In this paper, we presented Vector Binning Anomaly Detection (VBAD), a novel approach designed to address the challenges of scalability and computational efficiency in anomaly detection, particularly for large-scale datasets. By eliminating the dependency on deep neural networks (DNNs) and leveraging a binning technique, our method significantly reduces the computational overhead while maintaining competitive accuracy. Through extensive experiments on the MVTec-AD [5], VisA [36], and semiconductor photo mask datasets, we demonstrated that our approach performs exceptionally well, especially in scenarios where large datasets and high-resolution images are involved.

## References

[1] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. Image anomaly detection and localization with position and neighborhood information. *arXiv preprint arXiv:2211.12634*, 2(5):6, 2022. 2

[2] Christoph Baur, Stefan Denner, Benedikt Wiestler, Nassir Navab, and Shadi Albarqouni. Autoencoders for unsupervised anomaly segmentation in brain mr images: a comparative study. *Medical Image Analysis*, 69:101952, 2021. 2

[3] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*, 2020. 1

[4] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4):947–969, 2022. 1

[5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 1, 2, 5, 6, 7, 8

[6] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018. 2

[7] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019. 2

[8] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 2

[9] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 1, 2, 7

[10] Denis Gudovskiy, Shun Ishizaka, and Kazuki 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*, pages 98–107, 2022. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[12] Zhanghexuan Ji, Yan Shen, Chunwei Ma, and Mingchen Gao. Scribble-based hierarchical weakly supervised learning for brain tumor segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*, pages 175–183. Springer, 2019. 2

[13] Okan Kopuklu, Jiapeng Zheng, Hang Xu, and Gerhard Rigoll. Driver anomaly detection: A dataset and contrastive learning approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 91–100, 2021. 1

[14] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access*, 10:78446–78454, 2022. 2, 7

[15] 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*, pages 20402–20411, 2023. 2, 6, 7

[16] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013. 1

[17] Arian Mousakhan, Thomas Brox, and Jawad Tayyub. Anomaly detection with conditioned denoising diffusion models. *arXiv preprint arXiv:2305.15956*, 2023. 2

[18] Eduardo HP Pooch, Pedro Ballester, and Rodrigo C Barros. Can we trust deep learning based diagnosis? the impact of domain shift in chest radiograph classification. In *Thoracic Image Analysis: Second International Workshop, TIA 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 2*, pages 74–83. Springer, 2020. 2

[19] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2022. 1, 2, 6, 7

[20] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018. 2

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 5

[22] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999. 2

[23] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010. 2

[24] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *European Conference on Computer Vision*, pages 485–503. Springer, 2020. 1

[25] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017. 1

[26] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 1

[27] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 8

[28] Peng Xing and Zechao Li. Visual anomaly detection via partition memory bank module and error estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(8):3596–3607, 2023. 2

[29] Ke Yan, Jinzheng Cai, Adam P Harrison, Dakai Jin, Jing Xiao, and Le Lu. Universal lesion detection by learning from multiple heterogeneously labeled datasets. *arXiv preprint arXiv:2005.13753*, 2020. 1

[30] Sergey Zagoruyko. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5, 8

[31] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8330–8339, 2021. 7

[32] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112:107706, 2021. 2, 7

[33] Hui Zhang, Zheng Wang, Zuxuan Wu, and Yu-Gang Jiang. Diffusionad: Norm-guided one-step denoising diffusion for anomaly detection. *arXiv preprint arXiv:2303.08730*, 2023. 2

[34] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017. 2

[35] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018. 2

[36] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022. 1, 5, 6, 8