Template-guided Hierarchical Feature Restoration for Anomaly Detection

Hewei Guo[1*‡] Liping Ren[2*‡] Jingjing Fu[3†] Yuwang Wang[2†] Zhizheng Zhang[3]
Cuiling Lan[3] Haoqian Wang[2] Xinwen Hou[1]
[1]Institute of Automation, Chinese Academy of Sciences
[2]Tsinghua University [3]Microsoft Research Asia
{guohewei2020, xinwen.hou}@ia.ac.cn
{rlp20@mails., wang-yuwang@mail., wanghaoqian@}tsinghua.edu.cn
{jifu, zhizzhang, culan}@microsoft.com

Abstract

Targeting for detecting anomalies of various sizes for complicated normal patterns, we propose a Template-guided Hierarchical Feature Restoration method, which introduces two key techniques, bottleneck compression and template-guided compensation, for anomaly-free feature restoration. Specially, our framework compresses hierarchical features of an image by bottleneck structure to preserve the most crucial features shared among normal samples. We design template-guided compensation to restore the distorted features towards anomaly-free features. Particularly, we choose the most similar normal sample as the template, and leverage hierarchical features from the template to compensate the distorted features. The bottleneck could partially filter out anomaly features, while the compensation further converts the reminding anomaly features towards normal with template guidance. Finally, anomalies are detected in terms of the cosine distance between the pre-trained features of an inference image and the corresponding restored anomaly-free features. Experimental results demonstrate the effectiveness of our approach, which achieves the state-of-the-art performance on the MVTec LOCO AD dataset.

1. Introduction

Anomaly detection is typically treated as an out-of-distribution detection problem, which learns the distribution from normal samples during training and detects outliers as anomalies during inference. Existing anomaly detection methods have achieved promising results on the MVTec AD benchmark [4], which is mainly composed of images with simple normal patterns, i.e. the material surface is uniform, and background is clean, and only one object exists. However, in many practical applications, normal samples are composed of multiple objects placed by rule, such as PCBs with multiple electronic components, printed surfaces with diverse contents and products with parts assembly. Figure 1 illustrates several examples with complicated structures and predefined normal layout patterns. For example, a normal breakfast box contains oatmeal, nuts, oranges, and peach, which are carefully arranged following a certain rule. Its complicated normal pattern creates difficulties for the existing methods to formulate its normal distribution, and consequently leads to degradation on detection accuracy. More-

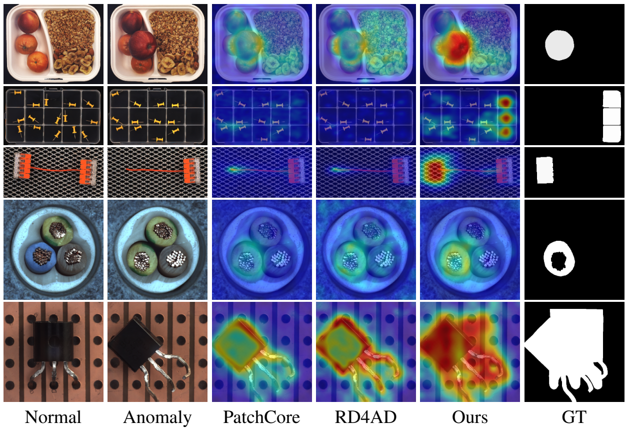


Figure 1: Anomaly detection on different categories. From top to bottom: breakfast box, pushpins, splicing connectors in MVTec LOCO AD [3], and cable, transistor in MVTec AD [4]. From left to right: normal image, anomaly image, anomaly maps of PatchCore [28], RD4AD [11] and our method, and ground truth. The red color corresponds to high anomaly score, whereas the blue represents low anomaly score. Best view in color.

[*] Equal contribution.
[†] Corresponding author.
[‡] Work done during an internship at Microsoft Research Asia.

over, the accurate anomaly localization is critical in practical industrial scenarios for evaluating the impact of defects and identifying their underlying causes. Therefore, to enhance the usability of anomaly detection methods in various practical scenarios, it is crucial to resolve the challenges associated with complex normal patterns and the requirements for accurate localization.

Substantial anomaly detection methods can be categorized as the embedding-based paradigm, as shown in Figure 2 (a). These methods identify anomalies based on the patch-level dissimilarity between the input embedding feature extracted from a pre-trained network and the normal feature distribution, through such as kNN [9, 25, 28], Mahalanobis distance [10, 26] or normalizing flows [30, 14, 44]. These methods rely on patch-wise feature comparison and prefer to detect local anomalies rather than global anomalies. Once the normal patterns become complicated, the corresponding normal distribution becomes challenging to be formulated, which in turn harms the detection performance.

From the aspect of reconstruction quality divergence, numerous reconstruction-based methods [13, 39, 16] are proposed to expect perfect reconstruction on normal regions and poor reconstruction on anomaly regions, as shown in Figure 2 (b), where the samples could be images or features. The reconstruction-based paradigm is highly adaptable to normal samples with complicated distributions, making it suitable for anomaly detection with rich diversity. However, they suffer from a significant drawback that in certain cases they may fail to detect defects, where the anomaly regions are reconstructed with similar quality as the normal regions. Meanwhile, reconstruction-based methods often incorporate autoencoder (AE) that involves several downsampling operations. Image details are susceptible to being lost due to feature compression operations, which leads to blurry outputs with large reconstruction error even for normal samples [16], and inevitably causes inaccurate identification of defect locations.

To address the challenges of the complicated normal data and the growing demand on accurate defect localization, we propose Template-guided Hierarchical Feature Restoration (THFR) network for anomaly detection, which restores anomaly-free features. The basic idea of our method is shown in Figure 2 (c), the image features are first compressed with bottleneck structure to filter out anomaly features at different levels, and then the distorted features are compensated with template embedding features, which is retrieved from template bank through image-level nearest neighbor search. The bottleneck structure in our approach is designed to retain the essential features that are common to normal samples. To achieve this, we propose global bottleneck and local bottleneck that respectively preserve normal semantic features and normal detailed features. We establish a template bank using multi-level embedding features of normal samples, and retrieve the most similar sample as the template based on the cosine similarity with the input feature. Additionally, we introduce a template-guided compensation module that leverages relation represetation between input features and template features to restore anomaly-free features.

Thanks to the template guidance and hierarchical compensation design, our method achieves state-of-the-art performance on MVTec LOCO AD [3], a dataset containing normal patterns of rich diversity, and also achieves competitive performance on MVTec AD [4], which is commonly used as anomaly detection benchmark.

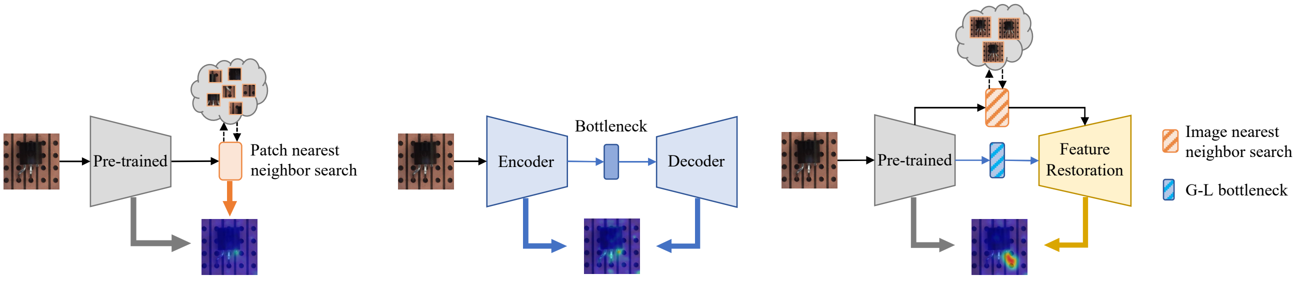
In summary, our main contributions are threefold:

- We propose a new framework to tackle the problem of anomaly detection upon data of rich diversity, in which the anomaly-free features with complicated distribution can be restored from the anomaly features with the guidance of template features.
- We design bottleneck compression to retain normal features while partially filtering out anomaly features, and propose template-guided compensation that leverage template embedding features to restore the distorted features towards normal.
- Extensive experiments on the standard benchmarks demonstrate the outstanding performance of the proposed method, especially for localization.

2. Related work

Unsupervised anomaly detection methods are trained on normal samples, while testing on both normal and anomaly samples. Classical anomaly detection methods [36, 38, 32, 42] treat the task as one class classification and focus on defining a compact closed one-class distribution.

Embedding-based methods Embedding-based methods use deep neural networks pre-trained on a large dataset to extract features from an image for anomaly detection and localization. Schirrmeister *et al.* [34] show that large natural-image datasets such as ImageNet [12] can extract more powerful features than a small application specific dataset. SPADE [9] uses a pre-trained network with multi-scale pyramid pooling and segments the anomalies region via among input image features and kNN normal features. PatchCore [28] also uses nearest neighbor to detect anomaly and meanwhile leverages greedy coreset subsampling to lighten memory bank. Different from methods using memory bank, PaDiM [10] abandons slow kNN algorithm and uses Mahalanobis distance metric as an anomaly score. There are follow-up methods [50, 17] to continue to improve the effectiveness of normal features based on Mahalanobis distance. DifferNet [30], CFLOW [14] and Fast-



(a) Embedding-based paradigm [28] (b) Reconstruction-based paradigm [16] (c) Restoration-based THFR method (ours)

Figure 2: Different paradigms for anomaly detection. (a) Embedding-based methods detect anomaly based on patch-level pre-trained feature similarity. (b) Reconstruction-based methods detect anomaly relying on the assumption that anomalies cannot be reconstructed with good quality. (c) THFR method detects anomaly by comparison between the original and restored anomaly-free features. It utilizes a combination of global and local bottlenecks to filter out anomaly features, known as the G-L bottleneck, and restores distorted features with template features retrieved by image-level nearest neighbor search.

Flow [44] utilize normalizing flows to explicitly approximate the data density, and detect anomaly samples based on their assigned likelihood. It is challenging for embedding-based methods to model the normal distribution when the patterns are complicated.

Reconstruction-based methods Reconstruction-based methods are proposed based on an assumption that a model trained on normal data only, cannot represent or reconstruct the anomalies accurately [51, 6]. They typically reconstruct samples from the manifold of the training data, using generative adversarial network (GAN) [20], autoencoder (AE) [29], or variational autoencoder (VAE) [19]. DAGAN [37] trains autoencoder with adversarial losses as the anomaly score of the image. RIAD [47], InTra [23], and SCADN [41] design inpainting frameworks and train models on masked normal data to recover the unseen regions using context for anomaly detection. Deng *et al.* [11] propose a feature reconstruction network, in which the decoder reconstructs multi-level pre-trained features. Ristea *et al.* [27] integrate the reconstruction-based functionality into a generic self-supervised block to improve the anomaly detection performance.

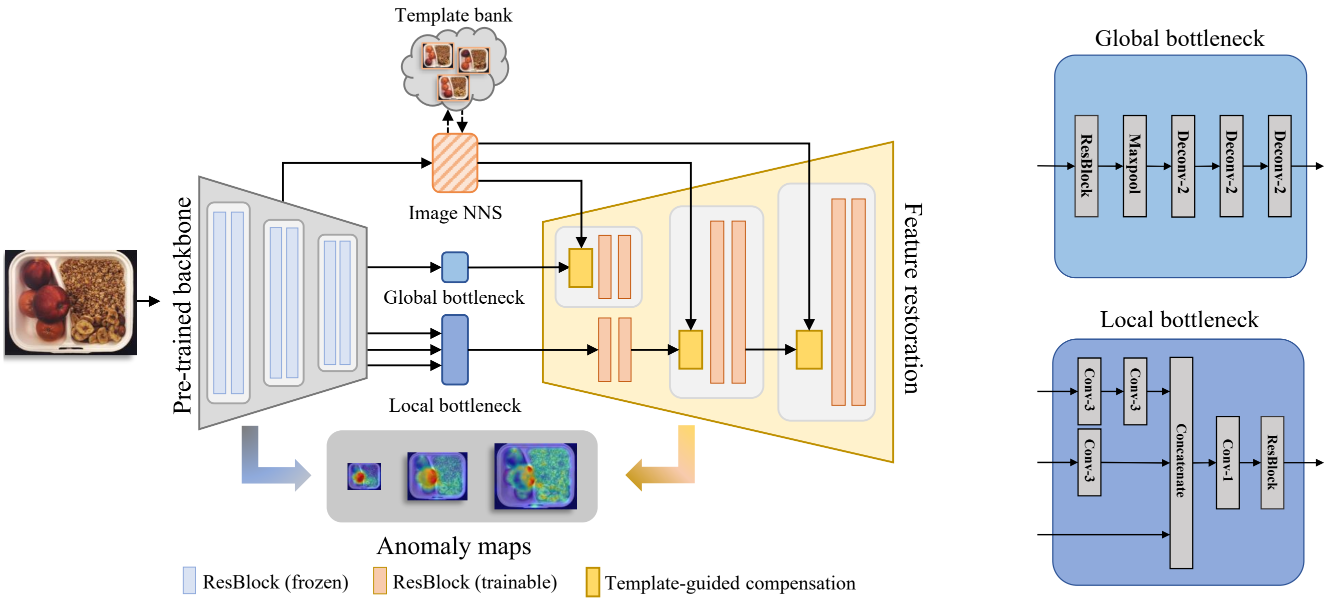
Nevertheless, due to the powerful representation capability of CNNs, considerable anomalies are reconstructed as well as the normal samples, leading to hypothesis failure. Several memory augmented networks [13, 16, 22] are proposed to enlarge the reconstruction quality gap between normal and anomaly regions. However, these learnable memory based methods focus on patch-wise normal feature representation and thus fail to tackle layout anomalies. Meanwhile, the down-sampling operations are commonly used for crucial feature extraction in reconstruction-based methods, causing the loss of detailed information, which harms the pixel-level anomaly localization accuracy. Several works [2, 16] adopt skip-connection to assist the reconstruction of details, but anomaly information is also passed to the decoder, resulting in defect leakage.

Our method aims to restore anomaly-free features from anomaly features instead of recognizing the reconstruction quality gap. For more effective restoration, we filter anomalies via bottleneck and recover the normal pattern and details through template-guided compensation. Note that our method uses the image-level pre-trained features of normal samples as templates to provide both global normal pattern priors and detailed normal feature priors for restoration.

Distillation-based methods have recently been successfully applied to anomaly detection, where anomalies are identified based on feature differences between a teacher network pre-trained on a large dataset and a student network trained only on normal samples using knowledge distillation [7, 5, 33, 3, 11, 40, 31]. Additionally, some researchers have attempted to convert unsupervised anomaly detection into a supervised learning task by augmenting normal samples with pseudo-anomalies [24, 21, 46, 35]. However, these approaches are prone to bias towards pseudo outliers and fail to detect a large variety of anomaly types. To improve the embedding feature quality, Zou *et al.* [52] propose SPD which leverage pseudo anomalies as negative sample for contrastive self-supervised pre-training instead of supervised pre-trained with classification.

3. Method

The overview architecture of our proposed method is shown in Figure 3. Our method first obtains the multi-level embedding features of a given image using a backbone pre-trained on ImageNet [12] as feature extractor. In the training stage, these embedding features are treated as the targets of feature restoration. In order to filter out the anomalies of various sizes, the embedding features are compressed by



(a) Template-guided Hierarchical Feature Restoration framework

(b) G-L bottleneck

Figure 3: Overview architecture of proposed Template-guided Hierarchical Feature Restoration (THFR) framework. (a) THFR consists of one pre-trained backbone encoder, bottleneck, and a restoration decoder. Given a test image, THFR uses ImageNet pre-trained backbone encoder to extract features for restoration target. THFR compresses input features through bottleneck, and the compressed features are compensated by template features retrieved from the template bank using image-level nearest neighbor search (image NNS). Finally, anomaly detection is performed in terms of the cosine distance between restored anomaly-free features and input features. (b) Detailed network design of global bottleneck and local bottleneck.

global bottleneck (GBN) and local bottleneck (LBN).

The compression along spatial and channel dimensions inevitably causes information loss on features. To recover the normal pattern with compensation for the lost details, we establish a template bank to record image-level normal embedding features and retrieve the most similar normal feature from the template bank through an image-level nearest neighbor search module called image NNS, and treat the retrieved result as template to guide feature restoration.

3.1. Bottleneck

We design global bottleneck and local bottleneck that hierarchically compress the embedding features to retain the crucial normal features at multiple levels, and thereby filter out anomaly features caused by anomalies of various sizes. The previous reconstruction-based works [13, 16, 11] use a fixed-scale loose bottleneck, which works on small-scale anomalies, but underperforms when anomalies are of large sizes. In our network, uncompressed template embedding features could provide rich hierarchical normal features while compensating the compressed features, so we could employ the bottleneck of different tightness to effectively filter anomaly features and obtain compact normal representation.

For a given image, its multi-level pyramid features $I_k \in \mathbb{R}^{H_k \times W_k \times C_k}$ are extracted by pre-trained ResNet [15] and fed into bottleneck structure for hierarchical compression, where $H_k \times W_k$ denotes the spatial dimension, C_k is the number of channels and k indicates the layer index with maximum value equal to K. The input feature maps are distinct between global bottleneck and local bottleneck. Specially, the high-level feature I_K is fed to global bottleneck, while $[I_1, \cdots, I_K]$ for local bottleneck.

Global bottleneck Considerable large-size anomalies result from missing or misplaced normal components [3], which appear to be globally abnormal but locally normal, creating difficulties for existing anomaly detection methods. We propose global bottleneck to preserve the most representative normal semantic features and filter out the anomaly semantic features. As shown in Figure 3 (b), global bottleneck reforms the deep-level feature I_K by a ResBlock and then compress the features by global maxpooling, producing a feature vector $Z_{GBN} \in \mathbb{R}^{1\times1\times2C_K}$. Global maxpooling extremely compresses the feature spatially to eliminate the anomaly features. It could alleviate the attention on object misalignment and instead focuses on global pattern representation. Finally, we use 2×2 deconvolution [48] with stride 2 to upsample the compressed feature Z_{GBN} to a 8×8 spatial feature map for succeeding feature restoration.

Local bottleneck Besides the compression on semantic features, we introduce a trainable local bottleneck to compress the multi-level features to preserve the local normal features and filter out the local anomaly features. The design of the local bottleneck follows the work [11], as shown in Figure 3 (b). To align multi-level features, the shallow features are downsampled through 3×3 convolution layers with stride of 2. We concatenate multi-level representations and fuse them by a ResBlock to obtain compressed feature $Z_{LBN} \in \mathbb{R}^{\frac{H_K}{2}\times\frac{W_K}{2}\times 2C_K}$, which is fed into detail compensation module for fine-grained feature restoration.

3.2. Template-guided compensation

Although the essential normal features are kept during compression, the information loss on normal features is unavoidable and needs to be compensated. We leverage multi-level normal embedding features as template to guide feature restoration. Since there is no one-to-one correspondence between the positional information of input and template features, we leverage the relationship between the input and template features to guide the restoration process. We fuse the input and template by stacking these two embeddings and the relation feature, and then passing them through multiple 1×1 convolution layers. The fused feature is fed to a series of ResBlocks for further restoration.

Template bank We construct template bank using image-level pre-trained features extracted from N normal samples in the all train set as $\mathcal{B} = \{B_1^1, \cdots, B_k^i, \cdots, B_K^N\}$, where B_k^i represents the k-th layer template feature of the i-th normal sample in template bank. During inference, We could reduce template bank using coreset subsampling method [1, 8] to reduce inference time and memory usage.

Image-level nearest neighbor search Given an input sample, we take the deepest pre-trained K-th layer feature I_K as the query to retrieve its correlated template. Image NNS obtains the template with index t by randomly selecting from the template candidates which are n most similar templates to increase the robustness during the training process. The template selection process could be formulated as follows:

$$t = \text{random}(\underset{\mathcal{S}\subset\{1,\cdots,N\},|\mathcal{S}|=n}{\arg\min} \sum\nolimits_{i\in\mathcal{S}} d(I_K, B_K^i)), \quad (1)$$

where $d(\cdot)$ denotes the image-level distance between input feature query I_K and template feature key B_K^i by flattening them to vectors to compute cosine distance. S is a subset of $\{1, \cdots, N\}$ denotes the indexes of n template candidates, which are the top-n nearest neighbors of the input feature I_K. Note that, the n set to 1 during inference. Compared with the traditional point-by-point search strategy, image NNS not only ensures that the reference is completely normal but also improves search efficiency. The multi-level embedding features of the template sample are used as references in the corresponding feature level, which are denoted as T for simplicity.

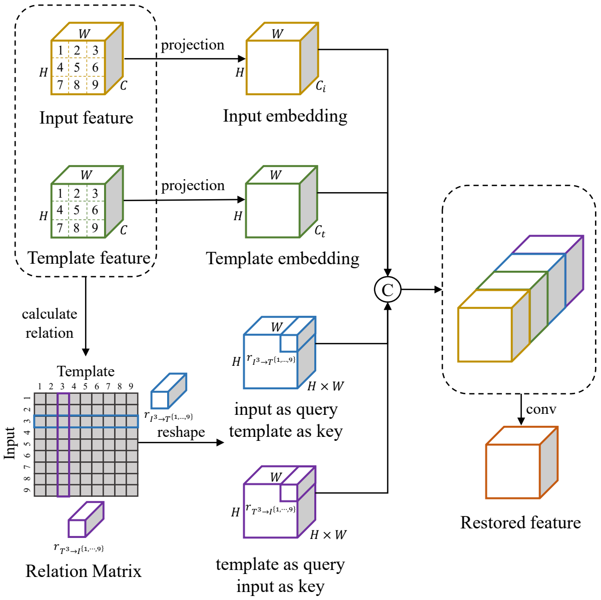


Figure 4: Template-guided compensation process. We leverage normal template feature to guide anomaly-free feature restoration with the relation representation between the input feature and the template feature.

Relation representation We use pairwise relation [49] between input feature and template feature as additional information to restore normal feature. Specifically, we use $r_{I^i\to T^j}$ to represent the relation from the i-th feature point of input feature to the j-th feature point of the template feature. The pairwise relation can be defined as a dot-product affinity in the embedding spaces as:

$$r_{I^i\to T^j} = \theta(I^i)^\top\phi(T^j) \quad (2)$$

where θ and ϕ are two embedding functions implemented by a 1×1 spatial convolutional layer. Similarly, we can get the affinity from j-th feature point of template feature to i-th feature point of input feature as $r_{T^j\to I^i}$. We use the pair ($r_{I^i\to T^j}$,$r_{T^j\to I^i}$) to describe the bi-directional relation between i-th feature point of input feature to j-th feature point of template feature. Then, we represent the relation matrix $\mathbf{R} \in \mathbb{R}^{m\times m}$ among all the nodes, where m is the resolution of the feature. The bi-directional pairwise relation vector of i-th feature point is $\mathbf{r}_i = [r_{I^i \to T^{\{1,\cdots,m\}}}, r_{T^i \to I^{\{1,\cdots,m\}}}]$. For example, as shown in Figure 4, the third row and the third column of the relation matrix, i.e. $\mathbf{r}_3 = [r_{I^3 \to T^{\{1,\cdots,9\}}}, r_{T^3 \to I^{\{1,\cdots,9\}}}]$. In this relation vector, the first half channels indicate the relation that regards the input feature point as query and template feature point as key, while the second half indicates the relation of reverse operation. The bi-directional pairwise relation vectors of all nodes form the relation feature.

Feature compensation We leverage normal template features to guide anomaly-free feature restoration. As shown in Figure 4, the input compressed feature with channel size C is projected to input embedding $E_i \in \mathbb{R}^{H \times W \times C_i}$, and the template feature with channel size C is projected to template embedding $E_t \in \mathbb{R}^{H \times W \times C_t}$. Here, C_i represents the channel size of input embedding, and C_t represents the channel size of template embedding, while H and W represent the height and width of the feature maps, respectively. The values of C_i and C_t can be adjusted with respect to the feature compression method used. For global bottleneck compression, C_i is set to the original channel size C to provide detailed information, while C_t is set to 1 to generate normal layout guidance for high-level semantic feature compensation. For local bottleneck compression, C_i is set to 1 to provide spatial layout guidance, and C_t is set to C to supply detail feature for low-level feature compensation. Then, we concatenate the input embedding E_i, the template embedding E_t and the relation feature, and passing them through multiple 1×1 convolution layers to get the restored feature. In order to explicitly demonstrate the effectiveness of our method, we visualize the restored feature using a visualization network [43] in experiment.

3.3. Loss

We use the cosine distance between the pre-trained feature and restored feature as restoration loss, as it is capable to evaluate the feature differences at multi-levels regardless of feature dimension differences among each level. Mathematically, for input feature I and restored feature $\hat{I}$, we calculate their feature cosine distance along the channel axis as restoration loss:

$$L_k = 1 - \frac{I_k^\top \cdot \hat{I}_k}{\|I_k\| \|\hat{I}_k\|}, \tag{3}$$

where L_k denotes the restoration loss of k-th level.

3.4. Anomaly map

In the testing phase, we detect anomalies in terms of the cosine distance between the pre-trained feature of an inference image and the corresponding restored anomaly-free feature. The restored feature tends to be close to input image feature for the normal regions, and departs on anomalies. Mathematically, we calculate pixel-wise cosine distance to obtain a 2D anomaly map in k-th level:

$$M_k(h, w) = 1 - \frac{(I_k(h, w))^\top \cdot \hat{I}_k(h, w)}{\|I_k(h, w)\| \|\hat{I}_k(h, w)\|}, \tag{4}$$

where (h, w) denotes the position of the feature vector, and $M_k \in \mathbb{R}^{H_k \times W_k}$. The final anomaly map is calculated as:

$$M = \sum_{k=1}^{K} U_k(M_k), \tag{5}$$

where $U_k(\cdot)$ denotes the upsampling operation of k-th anomaly map. In order to remove the noises in the score map, we smooth anomaly map M by a Gaussian filter.

4. Experiments

4.1. Experimental setup

Datasets We evaluate the proposed method on two industrial anomaly detection datasets: MVTec LOCO AD [3] and MVTec AD [4]. MVTec LOCO AD consists of 5 real-world sub-datasets for anomaly detection, including 1772 images for training, 304 for validation and 1568 for testing. The normal samples in MVTec LOCO AD often have complicated patterns, and the anomaly samples appear in form of logical or structural anomalies [3]. MVTec AD dataset consists of 15 real-world sub-datasets, with 5 categories of textures and 10 categories of objects. It contains 3629 training images and 1725 test images. Note that all the training images of the two datasets are normal, while some of test images are normal and others are anomalies.

Implementation details In our experiments, anomaly detection and localization are performed on one category at a time. We resize input images to 256 × 256 for data preprocessing and use pre-trained WideResNet50 [45] on ImageNet as feature extractor, with the deepest feature level K set to 3. To train our THFR network, we utilize Adam optimizer [18] and set the learning rate to 0.005. Each model is trained for 200 epochs with a batch size of 16 on NVIDIA Tesla V100 GPU. We use Gaussian filter with $\sigma = 2$ to smooth anomaly map on MVTec LOCO AD, while $\sigma = 4$ on MVTec AD. For training sample pair selection, we use $n = 1$ for all categories in MVTec LOCO AD and $n = 6$ for all categories in MVTec AD.

Evaluation metrics To fairly compare with other methods, we follow the common evaluation metrics of each dataset. For MVTec AD [4], we take area under the receiver operating characteristic (AUROC) as the image-level and pixel-level evaluation metric, and per-region-overlap (PRO) for pixel-level evaluation metric, which can better evaluate

Table 1: Pixel-level anomaly localization accuracy on MVTec LOCO AD dataset (sPRO) [3]. Remarkably, our approach achieves state-of-the-art performance in four categories and the mean of all categories. Best and second-best scores are bolded and underlined.

Method	Breakfast	Screw Bag	Pushpins	Connectors	Juice Bottle	Mean
S-T [5]	49.6	60.2	52.3	69.8	81.1	62.6
DRÆM [46]	49.9	49.0	49.3	67.3	80.0	59.1
CFLOW [14]	48.6	60.2	55.1	71.6	83.6	63.8
RD4AD [11]	42.2	57.4	61.4	71.3	85.1	63.5
PatchCore [28]	51.6	59.8	53.5	75.7	83.2	64.7
GCAD [3]	50.2	55.8	73.9	79.8	**91.0**	70.1
THFR (ours)	**58.3**	**61.5**	**76.3**	**84.8**	89.6	**74.1**

Table 2: Image-level anomaly detection accuracy on MVTec LOCO AD dataset (AUROC) [3]. Remarkably, our approach achieves the state-of-the-art performance on four categories and the mean of all categories. Best and second-best scores are bolded and underlined.

Method	Breakfast	Screw Bag	Pushpins	Connectors	Juice Bottle	Mean
S-T [5]	-	-	-	-	-	77.3
DRÆM [46]	75.7	72.7	76.0	82.5	93.9	80.1
CFLOW [14]	77.4	73.0	76.0	82.6	95.3	80.8
RD4AD [11]	68.7	**74.9**	75.9	84.4	94.8	79.7
PatchCore [28]	77.1	73.3	74.1	86.0	94.6	81.0
GCAD [3]	-	-	-	-	-	83.3
THFR (ours)	**78.0**	73.7	**88.3**	**92.7**	**97.1**	**86.0**

Table 3: Anomaly detection and localization results on MVTec AD dataset [4], including image-level and pixel-level AUROC, and pixel-level PRO. Best and second-best scores are bolded and underlined. * To ensure fair comparison with previous studies, CFLOW [14] is evaluated using input images with resolution of 256x256 pixels.

Method	Image	Pixel	Pixel(PRO)
SPADE [9]	85.5	96.0	91.7
PaDiM [10]	95.3	97.5	92.1
S-T [5]	-	-	91.4
DRÆM [46]	98.0	97.3	-
CFLOW * [14]	96.8	97.9	92.7
RD4AD [11]	98.5	97.8	93.9
PatchCore [28]	99.1	98.1	93.4
THFR (ours)	**99.2**	**98.2**	**95.0**

the performance of small size anomalies through weighting ground-truth regions of different size equally. Following the protocol mentioned in [5], we evaluate the PRO value for a large number of increasing thresholds until an average per-pixel false-positive rate of 30% for the entire dataset is reached. For MVTec LOCO AD [3], we use AUROC to measure image-level performance, and use saturated-per-region-overlaps (sPRO) [3] with per-pixel false-positive rate of 5% to evaluate pixel-level performance.

4.2. Main results

We compare the proposed model with several deep learning-based methods for anomaly detection as baselines, including embedding-based methods [9, 10, 14, 28], reconstruction-based methods [11, 46], distillation-based methods [5, 11, 3], and data-augmented methods [46].

MVTec LOCO AD Anomaly detection results on MVTec LOCO AD [3] are presented in Table 1 and Table 2. In comparison with the SOTA method [3], our approach obtain an absolute AUROC gain of 2.7% for the image-level detection, and an remarkable absolute sPRO gain of 4.0% for the pixel-level localization. Our approach achieves new state-of-the-art performance on both detection and localization results, and surpasses other methods by a large margin. More detailed results will be provided in the supplementary materials.

MVTec AD Quantitative comparison results on MVTec AD [4] between baselines and our approach are summarized in Table 3. For overall categories, our method produces comparable results with the other advanced methods. Our approach excels in achieving high accuracy on both image-level detection and pixel-level localization at the same time. More detailed results will be presented in the supplementary materials.

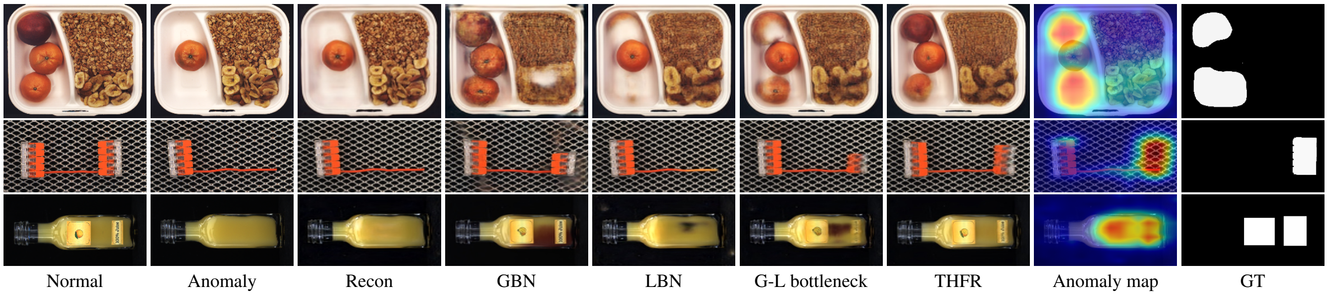


Figure 5: Visualization of restored features from different restoration networks. From left to right: normal image, anomaly image, reconstructed images from the pre-trained embedding features (Recon), restored features of restoration network only with GBN, only with LBN, with G-L bottleneck, and with our template-guided hierarchical feature restoration framework, anomaly map, and ground truth. The visualization results show how restoration networks progressively achieve anomaly-free restoration with G-L bottleneck and hierarchical feature compensation.

4.3. Ablation study

To explicitly demonstrate the effectiveness of our method, we visualize the restored features using a visualization network. We investigate the effectiveness of bottleneck design and template-guided compensation, and study the the influence of subsampling within the template bank.

Visualization of restored features As shown in Figure 5, we visually compared the pre-trained feature with the restored feature acquired from the network utilizing a visualization network [43]. We observe that the restoration networks employing solely local bottleneck (LBN) cannot effectively filter out anomalies that violate complex normal rules, such as missing connector heads and absent labels on the juice bottle. On the other hand, restoration networks with only global bottleneck (GBN) can filter out rule-breaking anomalies, but it encounters challenges in accurately restoring intricate details, such as the recovery of oatmeal and banana slices in the breakfast box. The restoration network with G-L bottleneck also has anomaly feature leakage after bottleneck compression, manifesting as anomalies like the irregular coloration of the juice. In contrast, our THFR network recovers anomaly-free feature through both semantic level correction and detailed feature compensation, resulting in an effective restoration to normal of various types of anomaly areas.

Impact of bottleneck Table 4 shows anomaly detection performance of local bottleneck and global bottleneck on MVTec LOCO AD dataset. The dataset has a rich diversity of data and anomalies, and the combination of local bottleneck and global bottleneck yields substantial enhancements in detection and localization results. Figure 6 illustrates that the global bottleneck primarily concentrates on filtering large-size anomalies, while the local bottleneck excels in filtering small-size anomalies.

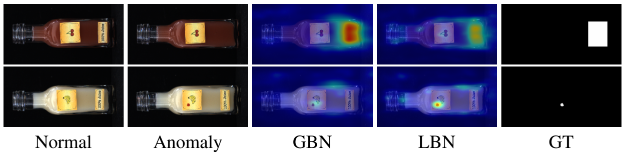


Figure 6: Visualization of anomaly detection results with different bottlenecks.

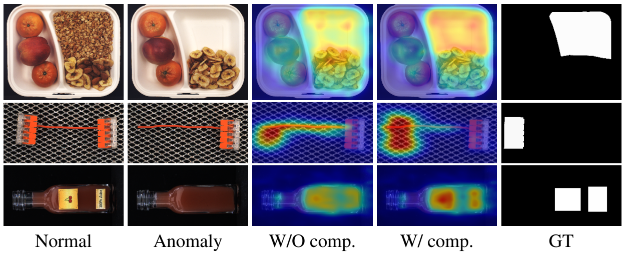


Figure 7: Visualization of anomaly detection results with or without template-guided compensation. More precise localization results are obtained with template-guided compensation.

G-L bottleneck vs. different defect size To access the ability of different bottleneck designs to detect anomalies of different sizes, we categorize defect images of LOCO [3] into 6 groups based on their defect sizes, and compare the performance of the restoration networks with LBN only, GBN only, and a combination of the two on these 6 groups in terms of anomaly localization. As shown in Figure 8, the accuracy of anomaly localization varies with the size of defects. LBN performs well on small defects, while GBN exhibits superior performance on large defects. The best performance is achieved by using a combination of LBN and GBN across all groups.

Impact of template-guided compensation The effectiveness of our compensation design, with or without com-

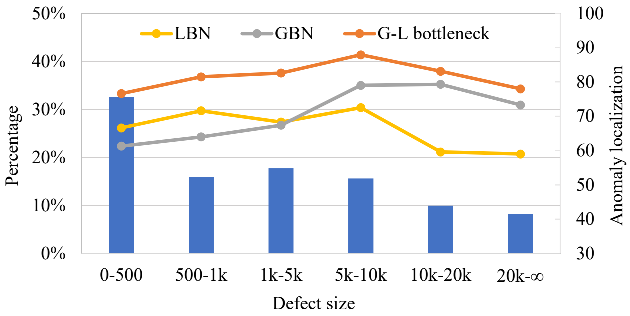


Figure 8: Performance comparison of the restoration networks with distinct bottleneck settings on varying defect sizes in terms of anomaly localization. We divide anomaly samples into 6 groups based on defect sizes, and show the percentage of each group with a bar chart.

pensation,is shown in Table 5 and Figure 7. Our compensation design improves detection result by 2.8% and localization result by 2.9% over the baseline on MVTec LOCO AD, and improves detection result by 0.6%, localization result by 0.4% and localization (PRO) result by 1.1% over the baseline on MVTec AD. By incorporating compensation design, it is possible to enhance the precision of localization and achieve more accurate coverage of abnormal areas. We also investigated the impact of relation design in our compensation module. The relation design improves detection result by 6.3% and localization result by 5.4% over the baseline on MVTec LOCO AD, and improves detection result by 0.5%, localization result by 0.1% and localization (PRO) result by 0.4% over the baseline on MVTec AD.

Impact of template bank subsampling The inference time and memory usage are highly relevant to the size of the template bank. We adopt coreset subsampling [1, 8] to find a subset that best describes the template bank during the testing phase. In Table 6, we compare performance of template banks with different subsampling ratios, and observe that subsampling has little impact on performance of MVTec AD. Although subsampling slightly reduces the mean detection accuracy of MVTec LOCO AD, its detection accuracy surpasses peers by a remarkable margin.

4.4. Complexity analysis

We compare inference speed and memory usage with different methods in Table 7. The inference time is measured with Intel Xeon E5-2698v4. Because the backbones are identical (i.e. WideResNet50 [45]), we only report complexity on the additional models. We use image NNS and template bank subsampling to improve searching efficiency and reduce memory cost. Compared with peer works, our method with tempalte bank subsampling to 10% obtain competitive performance on complexity and accuracy.

Table 4: The impact of bottleneck design on MVTec LOCO AD Dataset. Best scores are bolded.

Metrics\Bottleneck	GBN	LBN	G-L bottleneck
Image-level	79.7	80.3	**86.0**
Pixel-level	55.6	66.8	**74.1**

Table 5: Ablation study on template compensation and relation representation. Best scores are bolded.

Dataset		LOCO [3]		AD [4]		
w/ temp.	w/ rela.	Image	Pixel	Image	Pixel	Pixel(PRO)
		83.2	71.2	98.5	97.8	93.9
✓		79.7	68.7	98.6	98.1	94.6
✓	✓	**86.0**	**74.1**	**99.2**	**98.2**	**95.0**

Table 6: Anomaly detection performance with template bank subsampling at different ratios.

Dataset\Ratio		10%	25%	50%	100%
LOCO [3]	Image	85.6	**86.0**	**86.0**	**86.0**
	Pixel	73.7	74.0	74.0	**74.1**
AD [4]	Image	99.1	**99.2**	**99.2**	**99.2**
	Pixel	98.1	**98.2**	**98.2**	**98.2**
	Pixel(PRO)	**95.0**	**95.0**	**95.0**	**95.0**

Table 7: The comparison of pre-trained based approaches in terms of inference time (s), memory usage (MB), and performance (image/pixel/pixel-PRO) on MVTec AD [4].

Method	Inf. time	Memory	Performance
CFLOW [14]	0.178	947	(96.8/97.9/92.7)
RD4AD [11]	0.079	352	(98.5/97.8/93.9)
PatchCore(100%) [28]	0.149	1015	(99.1/98.0/93.3)
PatchCore(10%) [28]	0.113	102	(99.0/98.1/93.5)
Ours(100%)	0.130	1130	(**99.2/98.2/95.0**)
Ours(10%)	0.099	448	(99.1/98.1/**95.0**)

5. Conclusion

In the practical industrial scenarios, it is challenging to identify anomalies from the complicated normal patterns, and meanwhile precisely localize the anomalous regions of various sizes. We propose a novel template-guided hierarchical feature restoration network for anomaly detection. The proposed method recovers anomaly-free features from anomaly features by using bottleneck to filter anomaly features, and compensating compressed features with template retrieved from template bank. The hierarchical design benefits detection upon anomalies of various sizes. Experimental results demonstrate the effectiveness of our approach, especially on localization. Our method achieves state-of-the-art performance on MVTec LOCO AD dataset.

References

[1] Pankaj Agarwal, Sariel Har, Peled Kasturi, and R Varadarajan. Geometric approximation via coresets. *Combinatorial and Computational Geometry*, 52, 11 2004. 5, 9

[2] Samet Akçay, Amir Atapour-Abarghouei, and Toby P. Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019. 3

[3] 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, 2, 3, 4, 6, 7, 8, 9

[4] 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 (CVPR)*, June 2019. 1, 2, 6, 7, 9

[5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3, 7

[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. 3

[7] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5008–5017, June 2021. 3

[8] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algorithms*, 6(4), sep 2010. 5, 9

[9] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 2, 7

[10] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: A patch distribution modeling framework for anomaly detection and localization. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 475–489, Cham, 2021. Springer International Publishing. 2, 7

[11] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9737–9746, June 2022. 1, 3, 4, 5, 7, 9

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 3

[13] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 3, 4

[14] 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 (WACV)*, pages 98–107, January 2022. 2, 7, 9

[15] 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 (CVPR)*, June 2016. 4

[16] Jinlei Hou, Yingying Zhang, Qiaoyong Zhong, Di Xie, Shiliang Pu, and Hong Zhou. Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8791–8800, October 2021. 2, 3, 4

[17] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 303–319, Cham, 2022. Springer Nature Switzerland. 2

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[20] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19(143-155):18, 1989. 3

[21] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9664–9674, June 2021. 3

[22] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[23] Jonathan Pirnay and Keng Chai. Inpainting transformer for anomaly detection. In Stan Sclaroff, Cosimo Distante, Marco Leo, Giovanni M. Farinella, and Federico Tombari, editors, *Image Analysis and Processing – ICIAP 2022*, pages 394–406, Cham, 2022. Springer International Publishing. 3

[24] Masoud Pourreza, Bahram Mohammadi, Mostafa Khaki, Samir Bouindour, Hichem Snoussi, and Mohammad Sabokrou. G2d: Generate to detect anomaly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2003–2012, January 2021. 3

[25] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2806–2814, June 2021. 2

[26] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6726–6733, 2021. 2

[27] Nicolae-Cătălin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B. Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13576–13586, June 2022. 3

[28] 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 (CVPR)*, pages 14318–14328, June 2022. 1, 2, 3, 7, 9

[29] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Structuring autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 3

[30] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1907–1916, January 2021. 2

[31] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric student-teacher networks for industrial anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2592–2602, January 2023. 3

[32] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR, 10–15 Jul 2018. 2

[33] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H. Rohban, and Hamid R. Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14902–14912, June 2021. 3

[34] Robin Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21038–21049. Curran Associates, Inc., 2020. 2

[35] Hannah M. Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 474–489, Cham, 2022. Springer Nature Switzerland. 3

[36] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13(7):1443–1471, 07 2001. 2

[37] Ta-Wei Tang, Wei-Han Kuo, Jauh-Hsiang Lan, Chien-Fang Ding, Hakiem Hsu, and Hong-Tsu Young. Anomaly detection neural network with dual auto-encoders gan and its industrial inspection applications. *Sensors*, 20(12), 2020. 3

[38] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004. 2

[39] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 485–503, Cham, 2020. Springer International Publishing. 2

[40] Shinji Yamada, Satoshi Kamiya, and Kazuhiro Hotta. Reconstructed student-teacher and discriminative networks for anomaly detection. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2725–2732, 2022. 3

[41] Xudong Yan, Huaidong Zhang, Xuemiao Xu, Xiaowei Hu, and Pheng-Ann Heng. Learning semantic context from normal samples for unsupervised anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3110–3118, May 2021. 3

[42] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020. 2

[43] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 6, 8

[44] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021. 2, 3

[45] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. 6, 9

[46] 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 (ICCV)*, pages 8330–8339, October 2021. 3, 7

[47] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112:107706, 2021. 3

[48] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, 2010. 4

[49] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Xin Jin, and Zhibo Chen. Relation-aware global attention for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5

[50] Ye Zheng, Xiang Wang, Rui Deng, Tianpeng Bao, Rui Zhao, and Liwei Wu. Focus your distribution: Coarse-to-fine non-contrastive learning for anomaly detection and localization. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2022. 2

[51] 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. 3

[52] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 392–408, Cham, 2022. Springer Nature Switzerland. 3