CSAA

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

Few-shot anomaly detection with adaptive feature transformation and descriptor construction

Zhengnan HU [a], **Xiangrui ZENG** [a,*], **Yiqun LI** [a], **Zhouping YIN** [a], **Erli MENG** [b], **Leyan ZHU** [c], **Xianghao KONG** [c]

[a] *School of Mechanical Science & Engineering, Huazhong University of Science and Technology, Hubei, Wuhan 430074, China*
[b] *Xiaomi Corporation, Beijing 100085, China*
[c] *Institute of Artificial Intelligence, Beihang University, Beijing 100191, China*

Received 29 February 2024; revised 7 June 2024; accepted 11 June 2024

KEYWORDS

Industrial applications;
Anomaly detection;
Learning algorithms;
Feature extraction;
Feature selection

Abstract Anomaly Detection (AD) has been extensively adopted in industrial settings to facilitate quality control of products. It is critical to industrial production, especially to areas such as aircraft manufacturing, which require strict part qualification rates. Although being more efficient and practical, few-shot AD has not been well explored. The existing AD methods only extract features in a single frequency while defects exist in multiple frequency domains. Moreover, current methods have not fully leveraged the few-shot support samples to extract input-related normal patterns. To address these issues, we propose an industrial few-shot AD method, Feature Extender for Anomaly Detection (FEAD), which extracts normal patterns in multiple frequency domains from few-shot samples under the guidance of the input sample. Firstly, to achieve better coverage of normal patterns in the input sample, we introduce a Sample-Conditioned Transformation Module (SCTM), which transforms support features under the guidance of the input sample to obtain extra normal patterns. Secondly, to effectively distinguish and localize anomaly patterns in multiple frequency domains, we devise an Adaptive Descriptor Construction Module (ADCM) to build and select pattern descriptors in a series of frequencies adaptively. Finally, an auxiliary task for SCTM is designed to ensure the diversity of transformations and include more normal patterns into support features. Extensive experiments on two widely used industrial AD datasets (MVTec-AD and VisA) demonstrate the effectiveness of the proposed FEAD.

1. Introduction

Anomaly Detection (AD) aims to distinguish abnormal patterns and samples that deviate significantly from the distribution that most data follow Chandola et al.[1] It has been widely used in industrial settings to detect defective products,

[*] Corresponding author.
E-mail address: zeng@hust.edu.cn (X. ZENG).

Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.cja.2024.06.007
1000-9361 © 2024 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics.

Please cite this article in press as: HU Z et al. Few-shot anomaly detection with adaptive feature transformation and descriptor construction, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.06.007

promoting the implementation of quality systems. Unmanned Aerial Vehicle (UAV) manufacturing is a major application scenario for AD, where the AD algorithms are used for detecting the defects in landing gears,[2] gas turbines,[3,4] and so on. Due to the high requirements for precision and integrity of components in the UAV production process, it poses challenges to the AD methods. In industrial applications,[5–7] defects vary in size, shape, and other attributes, presenting significant challenges to AD methods to precisely localize anomalies. Furthermore, current research primarily focuses on full-data-driven self-supervised methods,[8–13] which require a significant amount of normal samples to construct memory banks or train encoder models. The paradigm of few-shot AD is shown in Fig. 1. According to Fig. 1, few-shot AD requires only a few normal samples for each category to locate defects on input images, thus being more efficient and practical. However, compared to full-data AD, only a few works[14–17] focus on the few-shot setting, leaving it largely unexplored.

Existing AD methods can be broadly categorized into three main branches: reconstruction-based methods[9,12,13,18] distribution-based methods[8,10], and feature-matching-based methods.[11,19–22] Reconstruction-based methods and distribution-based methods may suffer from over-generalization problems in few-shot settings, modeling anomalies as normal areas, for there are not enough samples to train the model to capture the distribution discrepancy between normal data and defective data. Thanks to the strong representation capability of pre-trained backbones,[23–25] feature-matching-based methods have shown promising performance recently. As Fig. 2(a) demonstrates, the feature-matching-based methods extract the features from the normal samples (i.e., without anomalies 🟧) and treat them as support features ○. During inference, the features of the input samples □ are matched with the support features ○, and the unmatched parts ▨ are determined as anomalies. However, most feature-matching-based AD methods simply extract regional features in a single frequency to build a memory bank and conduct feature comparisons between the test sample and the memory bank, neglecting the fact that defects exist in various frequencies. Moreover, these methods do not fully exploit the normal patterns contained in few-shot samples, increasing the possibility of misidentifying normal areas as defective ones.

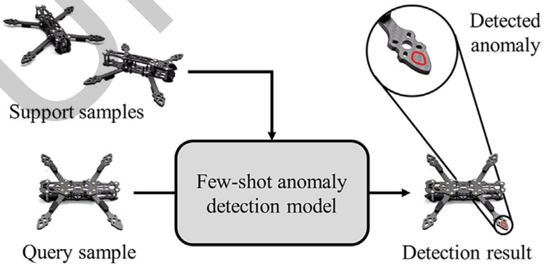
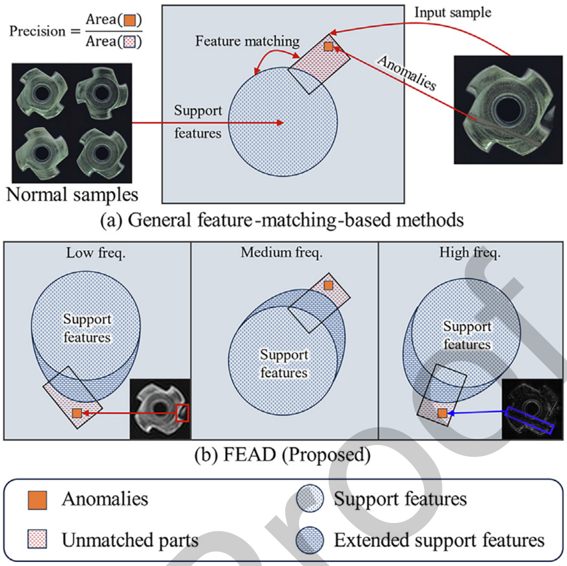


Fig. 1 Illustration of few-shot AD for unmanned aerial vehicle frameworks. Given a few normal samples as support samples, the model can detect anomalies from the query sample.



(a) General feature-matching-based methods

(b) FEAD (Proposed)

Fig. 2 Comparison of existing feature-matching-based AD methods and FEAD. The background stands for normal patterns, while the orange squares indicate abnormal patterns. An anomalous input sample □ contains both normal ▢ and abnormal 🟧 patterns. Existing feature-matching-based AD methods (a) perform the matching with only the extracted support features, while FEAD (b) extends support features to cover more normal patterns in the test sample (○ and ●) and detects anomalies in multiple frequencies. By extending the support features, the matching precision (Area(🟧)/Area(▨)) is significantly improved.

To address the aforementioned problems, we propose a Feature Extender for Anomaly Detection (FEAD), which extracts normal patterns from few-shot samples and extend them to cover a larger feature space, as illustrated in Fig. 2 (b). By extending support features, the matching precision (Area(🟧)/Area(▨)) is improved. Firstly, we devise a Sample-Conditioned Transformation Module (SCTM), which transforms support features under the guidance of the input sample to extract extra normal patterns from support features. These additional patterns, which correspond to the extended support features ●, ensure better coverage of normal patterns in the input sample. Secondly, we design an Adaptive Descriptor Construction Module (ADCM) based on the observation that the anomalies distribute in various frequencies, and should be detected separately, as shown in Fig. 3. In this figure, (a) is a normal sample of gear. It can be seen that its high-frequency feature mainly contains textures and contours, while the low-frequency features primarily represent its shape. (b) is an anomalous sample with both a scratch (blue box) and a bent (red box) on it. In its high-frequency features, the scratch can be identified, but the features are noisy and the bent area is vague. In contrast, the scratch in the low-frequency features cannot be distinguished, while the bent area is regular and easy to distinguish. As a result, features of multiple frequencies play an important role in anomaly detection. We build and select pattern descriptors in various frequencies adaptively, which correspond to the support features in low, medium and high frequencies in Fig. 2 (b). The pattern descriptors are then used to distinguish and localize anomaly patterns. Finally, we introduce an auxiliary task for SCTM to ensure the diversity of transformations and include more normal patterns into support features. FEAD fully exploits few-shot samples to construct multi-frequency pattern descriptors, achieving state-of-

Please cite this article in press as: HU Z et al. Few-shot anomaly detection with adaptive feature transformation and descriptor construction, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.06.007

the-art performance on MVTec-AD[26] and VisA[27] datasets and is readily applicable to industrial scenarios. Our main contributions can be summarized as follows:

(1) We introduce a few-shot industrial AD method FEAD, which extracts input-conditioned multi-frequency patterns from normal samples.
(2) We propose SCTM to transform support features and ensure better coverage of normal patterns, and ADCM to adaptively construct multi-frequency pattern descriptors.
(3) Extensive experiments on two prevalent industrial AD datasets (MVTec-AD[26] and VisA[27]) demonstrate the superiority of FEAD over previous state-of-the-art methods.

2. Related work

2.1. Anomaly detection

Industrial visual AD is attracting an increasing amount of attention in the computer vision community. It aims to identify industrial products with defects and locate the anomalies in the image. Mainstream AD methods can be broadly categorized into three types: feature-matching-based methods,[19,20,22] reconstruction-based methods[9,12,13,18,22,28–32], and distribution-based methods.[8,10,33–38]

Feature-matching-based methods exploit features from pretrained backbones to distinguish anomalies in images. SPADE[19] utilizes memory banks with multi-resolution hierarchical features for kNN-based AD on both pixel level and image level. Patch SVDD[22] extends a deep-learning variant of Support Vector Data Description (SVDD)[39,40] to a patch-based method using self-supervised learning. Li et al.[20] train a CNN feature extractor by distinguishing between normal samples and samples augmented by CutPaste policy. PatchCore[21] builds a memory bank with maximally representative normal patch features and applies a greedy coreset sampling process to reduce the time and memory needed for inference. InReaCh[11] extracts high-confidence nominal patches from training data by associating them into channels, only considering channels with high spans and low spread as nominal. Feature-matching-based methods achieve AD using only the features of normal samples as references. However, such a paradigm requires a sufficient number of normal samples to provide enough cues for distinguishing abnormal patterns from normal ones. Our few-shot method adopts this feature-matching paradigm and solves the problem of insufficient reference samples through SCTM.

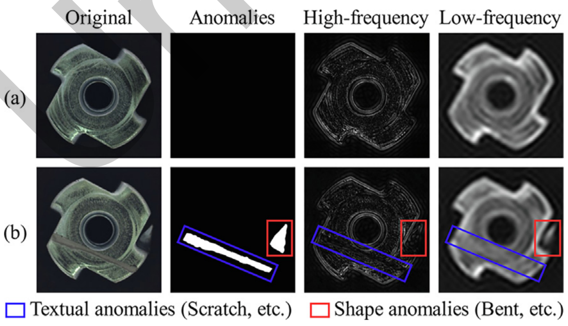

Original
Anomalies
High-frequency
Low-frequency
(a)
(b)
□ Textual anomalies (Scratch, etc.)
□ Shape anomalies (Bent, etc.)

Fig. 3 Illustration of the relation between anomalies and multi-frequency features. (a) shows a metal nut without anomalies, while (b) showns a defective one.

Reconstruction-based methods identify defects with generative models which reconstruct normal samples well but have a high reconstruction error when reconstructing anomaly samples. An et al.[41] utilize the reconstruction probability from variational autoencoders[42] as a measure for image-level AD. Zhou et al.[43] devise a Robust Deep Autoencoder (RDA) by splitting the input data into a clean part and a noisy part, and use the RDA to discriminate abnormal samples. ALAD[44] applies bi-directional GANs to extract adversarially learned features and then uses reconstruction errors to perform AD. AnoGAN[45] leverages convolutional GAN to learn a manifold of normal anatomical variability and an anomaly scoring scheme based on the mapping from image space to a latent space. f-AnoGAN[46] proposes a fast mapping method to map new data to GAN's latent space and detects anomaly through a combination of discriminator feature residual error and feature reconstruction error. GANomaly[28] utilizes a conditional GAN that jointly learns the generation of high-dimensional image space and the inference of latent space. MemAE[29] enhances the autoencoder with a memory updated during training to ensure that the autoencoder does not reconstruct anomalies. Ye et al.[32] propose to erase selected attributes from the training data and force the model to reconstruct the original image in order to let the model learn the semantic information. RIAD[47] takes AD as a self-supervised reconstruction-by-inpainting problem. It randomly removes regions on images and then reconstructs the image. Metaformer[31] achieves high model adaptation capability through meta-learned parameters, and leveraged instance-aware attention to emphasize the reconstruction gap of foreground regions. SSM[30] augments each training sample into triplets by random masking and proposes an iterative mask refinement policy to locate anomalies during inference. Lu et al.,[18] Zhang et al.,[48] and Shin et al.[49] leverage diffusion models to perform AD. THFR[9] designs bottleneck compression to filter out anomaly features while preserving normal features and proposes template-guided compensation, which leverages template embedding features to restore the distorted features. OmniAL[13] proposes a unified CNN framework for AD on multiple classes. FOD[12] designs a two-branch structure to explicitly establish intra- and inter-image correlations, and then fuse the features from two branches to localize anomalies. Although the idea of reconstruction-based methods is easy to understand, the training of reconstruction models usually relies on artificially generated anomaly samples, whose appearance significantly differs from real defects, limiting the effectiveness of reconstruction.

Distribution-based methods model the distribution of normal data to detect anomalies. DSEBMs[36] use deep structured neural networks to predict an energy function for AD. DAGMM[38] jointly optimizes an autoencoder and a Gaussian Mixture Model (GMM), balancing autoencoding reconstruction, density estimation of latent representation, and regularization. Salehi et al.[35] devise a multi-resolution knowledge distillation mechanism and localize anomalies using the discrepancy between the teacher and student networks' intermediate activation values given an input sample. PaDiM[33] leverages multivariate Gaussian distributions to attain a probabilistic representation of the normal classes. Zheng et al.[37] learn a dense and compact distribution of normal samples through a coarse-to-fine alignment process. CFLOW-AD[34] designs a conditional normalizing flow framework that explicitly estimates the likelihood of encoded features by multi-scale generative decoders. PyramidFlow[10] proposes a latent template-based defect contrastive localization paradigm to reduce intra-class variance, and leverages pyramid-like normalizing flows to help generalization in multi-scale fusing and volume normalization. PNI[8] utilizes the conditional probability given neighborhood features to model the distribution of normal samples. Distribution-based methods are well interpretable mathematically. However, since distribution-based methods usually adopt statistical models such as normalized flow to fit the probability distribution of training data, they usually introduce greater computational overhead and also require more data for training. In contrast, the few-shot method we propose does not need heavy training and only requires a few normal samples to achieve AD.

2.2. Few-shot anomaly detection

Full-data AD methods exhibit high data dependence, which limits their application in the industry. Consequently, few-shot AD has developed to meet the demands of manufacturing applications. TDG[17] proposes a hierarchical generative model that captures the multi-scale patch distribution of normal samples. It also applies multiple image transformations and optimizes discriminators to distinguish between real and fake patches, as well as between different transformations applied to the patches. The anomaly score is computed by aggregating patch-based votes of the correct transformations. It can be taken as a reconstruction-based method. DifferNet[16] exploits the descriptiveness of features extracted by CNNs to estimate their density using normalizing flows, which possess the ability to well estimate distributions from a few support samples. It is a distribution-based method. RegAD[14] learns a unified model among multiple categories in a meta-learning style and detects anomalies by comparing the registered features of the input image and its corresponding support images. It can also be categorized as a distribution-based method. WinCLIP[15] equips the pre-trained CLIP[50] model with a compositional ensemble on state words and prompt templates and efficiently extracts and aggregates multi-level features that are well aligned with texts. It is a feature-matching-based method which fully leverages CLIP[50] features.

Despite significant effort devoted to few-shot AD, existing methods pay little attention to localizing abnormal patterns in multiple frequency domains. Moreover, the information carried by support samples has not been fully exploited yet. By introducing ADCM and SCTM, FEAD thoroughly leverages the information in support samples by conditioned feature transformation and adaptively locates anomalies in multiple frequencies.

3. Methodology

3.1. Problem definition

Our work focuses on the few-shot industrial AD problem. Formally, given an image $\boldsymbol{I}$, the goal of industrial AD is to predict whether $\boldsymbol{I}$ is anomalous. This task can be divided into two levels: image level and pixel level. We treat image-level AD as a binary classification problem. For each image $\boldsymbol{I}$, the algorithm predicts a score $s^{\text{img}} \in [0, 1]$, which indicates the possibility of $\boldsymbol{I}$ being an anomalous image. At the pixel level, an anomaly map $\boldsymbol{S}^{H\times W} \in [0, 1]$ is predicted for $\boldsymbol{I}$, where H and W are the height and width of $\boldsymbol{I}$, respectively. Following conventional settings of few-shot learning, for the $k-$shot setting, a subset of the training set containing k images is used to train the model. All training images are anomaly-free.

3.2. Overview

Our method aims to estimate a normal sample distribution as precisely as possible using limited samples. In order to achieve this goal, we introduce the Sample-Conditioned Transformation Module (SCTM) to expand the distribution of normal samples. Additionally, we construct normal pattern descriptors in various frequency spaces through a novel Adaptive Descriptor Construction Module (ADCM) and filter the defective areas with the built descriptors.

The overall framework of our method is shown in Fig. 4. We utilize a shared feature extractor to extract the features $\boldsymbol{F}^{\text{S}}$ from support images and $\boldsymbol{F}^{\text{Q}}$ from the query images. For each query image, SCTM collaborates both $\boldsymbol{F}^{\text{S}}$ and $\boldsymbol{F}^{\text{Q}}$ and generates a transformation parameter $\boldsymbol{\phi}$, which is applied to $\boldsymbol{F}^{\text{S}}$ to form a normal pattern pool. The normal patterns are then fed into ADCM to construct normal pattern descriptors $\boldsymbol{D}$. Finally, we filter $\boldsymbol{F}^{\text{Q}}$ with $\boldsymbol{D}$ and produce image-level and pixel-level anomaly localization results.

3.3. Sample-conditioned transformation

In few-shot settings, only several anomaly-free samples of each category are available for the AD model. Consequently, it is crucial for few-shot AD methods to fully exploit relevant information in the few-shot samples. A straightforward way is to perform augmentations like affine transformations on support samples to obtain more diverse features and cover more normal patterns. Nevertheless, the feature space is so large that random augmentation fails to cover it due to enormous computation and memory costs.

Essentially, only the patterns that appear in the input sample are necessary for AD, while other patterns in the feature space are almost irrelevant. Inspired by Jaderberg et al.,[51] we propose a Sample-Conditioned Transformation Module (SCTM) to extend the support features under the guidance of the input feature in order to better cover the normal patterns in the input sample. SCTM effectively reduces the redundancy compared to random augmentation policy and significantly enhances AD performance with little extra computational cost.

To be specific, SCTM first predicts n rotation-scale pairs $\boldsymbol{\phi} = \{(\theta_i, \gamma_i)|1 \leqslant i \leqslant n\}$, which are then used to transform support features $\boldsymbol{F}^{\text{S}}$. The detailed process of transformation parameters prediction is demonstrated in Fig. 5. Support images' features $\boldsymbol{F}^{\text{S}}$ and query image's features $\boldsymbol{F}^{\text{Q}}$ are respectively fed into a localization network f_{loc} which consists of 2 blocks of 3×3 convolution with max pooling operation and

ReLU activation to extract query localization features $\boldsymbol{F}^{\mathrm{Q}}_{\mathrm{loc}}$ and support localization features $\boldsymbol{F}^{\mathrm{S}}_{\mathrm{loc}}$:

$$\boldsymbol{F}^{\mathrm{Q}}_{\mathrm{loc}} = f_{\mathrm{loc}}\left(\boldsymbol{F}^{\mathrm{Q}}\right) \tag{1}$$

$$\boldsymbol{F}^{S}_{loc} = f_{loc}\left(\boldsymbol{F}^{S}\right) \tag{2}$$

Support localization features $\boldsymbol{F}^{\mathrm{S}}_{\mathrm{loc}}$ are then averaged across support samples to get $\overline{\boldsymbol{F}}^{\mathrm{S}}_{\mathrm{loc}}$, which are concatenated with $\boldsymbol{F}^{\mathrm{Q}}_{\mathrm{loc}}$ along the channel dimension to obtain final localization features $\boldsymbol{F}_{loc}$:

$$\boldsymbol{F}_{\mathrm{loc}} = \mathrm{concat}\left[\boldsymbol{F}^{\mathrm{Q}}_{\mathrm{loc}}; \overline{\boldsymbol{F}}^{\mathrm{S}}_{\mathrm{loc}}\right] \tag{3}$$

Next, the localization features $\boldsymbol{F}_{\mathrm{loc}}$ are fed into a 2-layer MLP with ReLU activation to obtain n transformation parameter pairs $\boldsymbol{\phi}' = \left\{\left(\theta'_i, \gamma'_i\right) | 1 \leqslant i \leqslant n\right\}$, which are then transformed into rotation-scale pairs $\boldsymbol{\phi}$ as follows:

$$\boldsymbol{\phi} = \left\{\left(\pi \tanh(\theta'), 1 + \gamma'_i\right) | \left(\theta'_i, \gamma'_i\right) \in \phi'\right\} \tag{4}$$

Finally, the rotation-scale pairs are used to construct the normal pattern pool by transforming support features $\boldsymbol{F}^{\mathrm{S}}$. The normal pattern pool contains patterns from both original support features and transformed support features.

3.4. Adaptive descriptor construction

In industrial scenarios, the appearance of anomalies varies a lot. Most existing methods detect anomalies based on a single extracted feature. However, many defects are visible only for a range of frequencies, and not all features are relevant for the identification of defects. Using the single feature implicitly mixes features of different frequencies, which is detrimental to anomaly localization.

Empirically, small convolution kernels are more sensitive to high-frequency features, such as edges, due to their smaller receptive fields. On the contrary, large kernels are more sensitive to low-frequency features because of their larger receptive fields. In order to perceive features of different frequencies, we propose an Adaptive Descriptor Construction Module (ADCM) to construct convolution kernels of different sizes as normal pattern descriptors. We build descriptors of three sizes (1, 3, and 5) to capture high-frequency, medium-frequency, and low-frequency features respectively. Considering memory efficiency, we use learnable filters to downsample large descriptors.

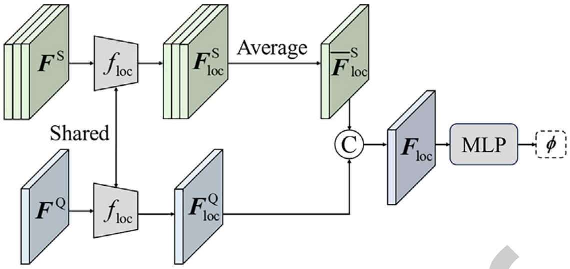

$\boldsymbol{F}^{\mathrm{S}}$
f_{loc}
$\boldsymbol{F}^{\mathrm{S}}_{\mathrm{loc}}$
Average
$\overline{\boldsymbol{F}}^{\mathrm{S}}_{\mathrm{loc}}$
Shared
C
$\boldsymbol{F}_{\mathrm{loc}}$
MLP
$\boldsymbol{\phi}$
$\boldsymbol{F}^{\mathrm{Q}}$
f_{loc}
$\boldsymbol{F}^{\mathrm{Q}}_{\mathrm{loc}}$

Fig. 5 Illustration of SCTM. It first fed support features and query features into a localization network f_{loc} to obtain corresponding localization features. Support localization features are then averaged across support samples and concatenated with query localization features to generate final localization features, which are passed through an MLP to predict transformation parameters.

The pipeline of ADCM is illustrated in Fig. 6. For normal patterns $\boldsymbol{P} \in \mathbf{R}^{B \times C \times h \times w}$ from the normal pattern pool, we adopt a sliding window of size s to process $\boldsymbol{P}$ in the spatial dimension, and features $\boldsymbol{D}$ can be obtained, where $\boldsymbol{D} \in \mathbf{R}^{B \times C \times h' \times w' \times s \times s}$, $h' = h - s + 1$, and $w' = w - s + 1$. By rearranging the dimensions of $\boldsymbol{D}$, we obtain $B \times h' \times w'$ descriptors with size s and C channels.

When constructing large descriptors, there are overlap between adjacent descriptors, which may lead to a heavy memory burden. Additionally, there're backgrounds and noisy parts in the image, which we do not concern and therefore can be removed. To achieve this, we introduce a learnable filter $\boldsymbol{M}$ to downsample the constructed descriptors. As shown in Fig. 7, $\boldsymbol{M} \in \mathbf{R}^{h' \times w'}$ is utilized to filter the constructed $\boldsymbol{D}$. For each positive position on $\boldsymbol{M}$, the descriptor of the corresponding position is retained, and vice versa. In practice, we instan-

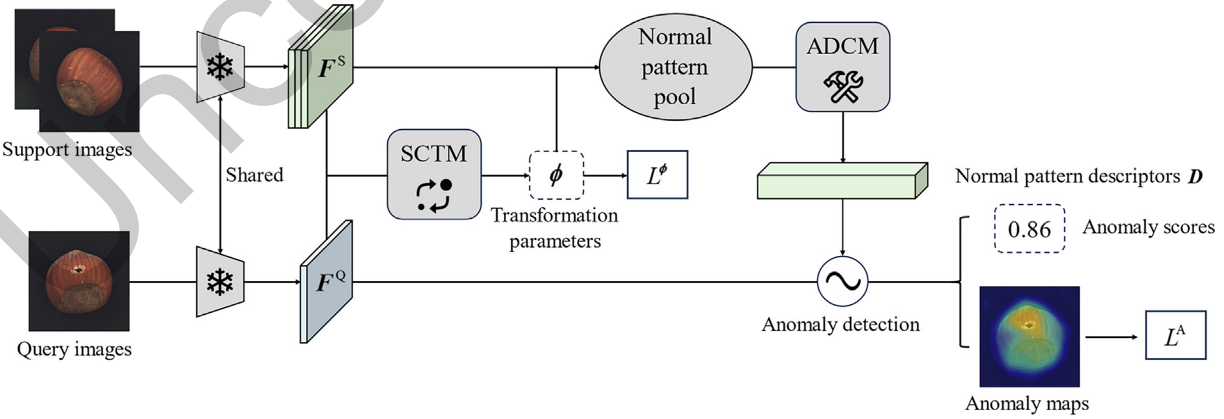


Fig. 4 Illustration of our overall framework. A shared feature extractor is adopted to extract features from support images and query images. The features are then fed into SCTM to generate transformation parameters, which is used to transform support features into normal pattern pool. The ADCM build normal pattern descriptors with normal pattern pool, and the descriptors are used to detect anomalies from the query features.

Please cite this article in press as: HU Z et al. Few-shot anomaly detection with adaptive feature transformation and descriptor construction, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.06.007

tiate $\boldsymbol{M}$ as a float tensor, and during the downsampling process, we convert $\boldsymbol{M}$ into a boolean tensor in the following way:

$$\boldsymbol{M}^{\text{bool}} = \text{bool}(\text{ReLU}(\boldsymbol{M})) \tag{5}$$

The $\boldsymbol{M}^{\text{bool}}$ is shared among all instances of each mini-batch and only the descriptors corresponding to the true values are kept. Considering that $\boldsymbol{M}$ is not supervised explicitly, we devise a pattern that serves as a good initialization for $\boldsymbol{M}$. We sample descriptors with a stride of s along each spatial dimension, which is equivalent to no overlap between two adjacent descriptors. Since features at the edges are only contained in a few descriptors, we sample with a stride of 1 at the edges. All positive positions are initialized to 1, while others are -1.

3.5. Anomaly localization

In an image containing anomalies, the distribution of anomaly-free parts is similar to that of normal samples, so it can be matched with a normal pattern descriptor. For parts containing anomalies, the features are out of distribution, so they should not match well with any normal pattern descriptor. We design a matching-based AD method based on this assumption.

We utilize Euclidean distance to match the normal pattern descriptors and the features of the query sample. Formally, as shown in Algorithm 1, given normal pattern descriptors $\boldsymbol{D} \in \mathbf{R}^{N\times C\times s\times s}$ and query features $\boldsymbol{F}^{\text{Q}} \in \mathbf{R}^{B\times C\times H\times W}$, we calculate a distance matrix $\boldsymbol{M}^{D} \in \mathbf{R}^{BHW\times N}$, which indicates the distances of all $<$descriptor, query feature patch$>$ pairs. For each patch, we take the minimum distance as the patch-level anomaly score. The patch-level anomaly maps are upsampled to the original image size with bilinear interpolation and Gaussian blurred as the pixel-level anomaly maps.

Inspired by PatchCore,[21] we decide the image-level anomaly score by the most anomalous patch, and we reweight this score to reduce the impact of chance. We regard the patch with the highest anomaly score as the most anomalous one, and find the descriptor corresponding to the score as a reference:

$$\text{patch}^{\text{A}}, \text{desc}^{\text{ref}} = \arg\max \boldsymbol{M}^{\text{D}} \tag{6}$$

We then take k descriptors $\boldsymbol{D}^{\text{sup}} = \{\text{desc}_1, \text{desc}_2, ..., \text{desc}_k\}$ that are most similar to desc^{ref}, and use the weighted average of their Euclidean distances with patch^{A} as the weight of the image-level score:

$$s^{\text{img}} = \left(1 - \boldsymbol{S}^{\text{ref}}\left(\boldsymbol{M}^{\text{D}}_{\text{patch}^{\text{A}},\boldsymbol{D}^{\text{sup}}}\right)\right)\boldsymbol{M}^{\text{D}}_{\text{patch}^{\text{A}},\text{desc}^{\text{ref}}} \tag{7}$$

where $\boldsymbol{S}^{\text{ref}}$ indicates softmax and take the result corresponding to desc^{ref}. We set $k = 9$ for descriptors of size 1 and $k = 3$ for larger descriptors.

When descriptors of multiple scales are used at the same time, we use the result of the smallest descriptor at the image level, and weight the results of each group of descriptors at the pixel level as the final result.

3.6. Training scheme

During training, the query images are generated from support images. Following DREAM,[52] we generate just-out-of-distribution images with a simulated anomaly generation process. We generate a noise image with a Perlin noise generator[53] and binarize it with a random threshold, forming an anomaly map $\boldsymbol{M}^{\text{A}}$. An anomalous texture image $\boldsymbol{T}$ is sampled from a texture dataset that is unrelated to the query images, and blended with the original image $\boldsymbol{I}$, which can be denoted as:

$$\boldsymbol{I}^{\text{A}} = \bar{\boldsymbol{M}}^{\text{A}} \odot \boldsymbol{I} + (1-\beta)\left(\boldsymbol{M}^{\text{A}} \odot \boldsymbol{I}\right) + \beta\left(\boldsymbol{M}^{\text{A}} \odot \boldsymbol{T}\right) \tag{8}$$

where $\odot$ is the element-wise multiplication operator and β indicates the random opacity parameter.

We adopt Binary Cross Entropy (BCE) loss to supervise the predicted anomaly maps $\widehat{\boldsymbol{M}^{\text{A}}}$:

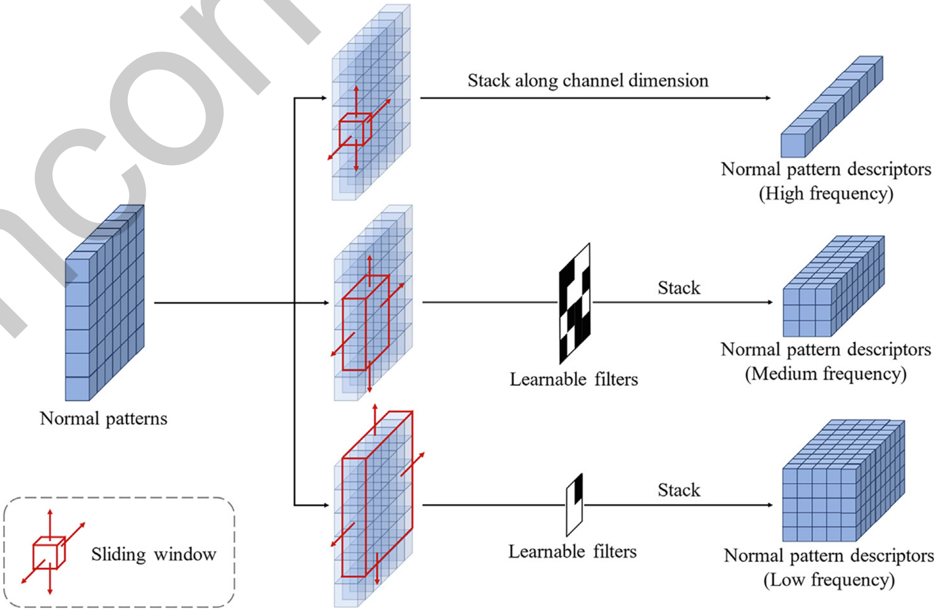


Fig. 6 Illustration of ADCM. The normal patterns are processed with sliding windows of 3 different sizes, generating multi-frequency normal pattern descriptors. Large descriptors are downsampled with learnable filters for reasonable memory consumption.

Please cite this article in press as: HU Z et al. Few-shot anomaly detection with adaptive feature transformation and descriptor construction, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.06.007

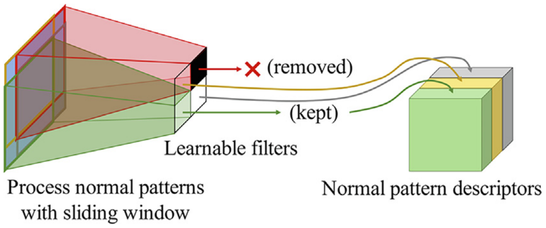

(removed)
(kept)
Learnable filters
Process normal patterns with sliding window
Normal pattern descriptors

Fig. 7 Illustration of the descriptor downsampling process. The position on the learnable filter has a one-to-one correspondence with the descriptor. The learnable filters are trained to preserve the focused parts and remove the not concerned pats. For the position of the positive response, the corresponding descriptor is retained, and the descriptor of the negative response position is removed.

$$L^{\mathrm{A}} = \mathrm{BCE}\left(\boldsymbol{M}^{\mathrm{A}}, \widehat{\boldsymbol{M}^{\mathrm{A}}}\right) \tag{9}$$

Auxiliary losses are also used to supervise the parameters generated by SCTM. In order to make SCTM generate diverse rotation parameters, we supervise Θ with a max–min strategy. Given the Θ parameters are sorted in ascending order, and we optimize the minimum difference between two adjacent parameters:

$$L^{\Theta} = \left(-\min_{k=1}^{n-1}|\Theta_{k+1} - \Theta_k|\right) \cdot E(n > 1) \tag{10}$$

where E stands for indicator function.

Considering that in industrial scenarios, the scales of samples from the same category don't vary seriously, we constrain the scale factors to distribute within a certain threshold:

$$L^{\gamma} = |\gamma| \cdot E(|\gamma| > \mathrm{thresh}) \tag{11}$$

where thresh is the constraint for scale factors. The overall loss function of our model is given as:

$$L = L^{\mathrm{A}} + \lambda^{\Theta}L^{\Theta} + \lambda^{\gamma}L^{\gamma} \tag{12}$$

4. Experiments

4.1. Datasets and evaluation protocols

We conduct our experiments on two most widely used real-world datasets for industrial anomaly localization, including MVTec-AD[26] and VisA[27]. Both datasets contain various categories of objects, high-resolution images, and corresponding pixel-level annotations for anomalies.

MVTec-AD focuses on AD in industrial scenarios. It contains 15 categories with 5354 images in total, 1725 of which are defective and the remaining ones are defect-free. The dataset includes 10 object categories and 5 texture categories, all of which contain multiple types of anomalies.

VisA is an emerging benchmark for industrial AD. The dataset contains 10,821 images with 9621 normal and 1200 anomalous samples. The samples spans 12 objects across 3 domains, including roughly aligned objects, multiple instances, and complex structures in objects.

Evaluation Metrics. Following previous works,[15,27,33] we utilize Area Under the Receiver Operating Characteristic curve (AUROC) and Area Under the Precision-Recall curve (AUPR) as evaluation metrics for image level. Additionally, we employ pixel-wise AUROC to evaluate the performance of our model at pixel level.

4.2. Implementation details

We adopt Wide ResNet-50[25] pretrained on ImageNet-1K[54] for the feature extractor. The input images are resized to 368 × 368 and we use only layer 2 and layer 3 features of the feature extractor as input to the subsequent modules. The number of rotation-scale pairs n in SCTM is set to 12. For the sake of the robustness of the filtering process, all features are softened with an average pooling layer of kernel size 3 and stride 1. The 1 × 1 convolution kernels constructed from the support features are not subsampled, and the 3 × 3 and 5 × 5 kernels are subsampled at a ratio of approximately 20% and 3% respectively. During training, we utilize AdamW[55] optimizer for training 20 epochs with an initial learning rate of 1×10^{-4}. The learning rate is decreased by 10 times at the 10th and 15th epochs. During the evaluation process, we apply Gaussian blur with a kernel size of 49 and sigma of 6 to the anomaly maps. We set the scale factor constraint $\mathrm{thresh} = 0.15$ and the auxiliary loss weights $\lambda^{\Theta} = \lambda^{K} = 1$. We normalize the anomaly scores and anomaly maps with min–max normalization.

4.3. Comparison with state-of-the-art methods

We conduct experiments on MVTec-AD and VisA benchmarks and compare them with the results of previous state-of-the-art methods. Some of the results are reproduced on anomalib,[56] since no reported performances are available for the corresponding methods.[14,57–61] We evaluate our model under the settings of 1-shot, 2-shot, and 4-shot. The pixel-level and image-level results are shown in Table 1 and Table 2 respectively. It can be found that our method achieves the best performance on both datasets at almost all pixel-level metrics, which demonstrates the effectiveness of our approach. At the image level, our method also outperforms all other methods on most of the metrics. It is worth mentioning that despite WinCLIP+[15] using a larger backbone (ViT-Base[23]) and introducing additional language information, our method still achieves comparable accuracy on the MVTec-AD dataset.

We also provide the categorial AUROC performance on MVTec-AD and VisA. All performances are tested under the 4-shot setting. The results are shown in Table 3 and Table 4. According to the tables, our method reaches the top-2 performance among the listed methods in almost all categories. In particular, our method works well for small objects (e.g. capsules, pills, screws, etc.) and objects with complex structures (e.g. PCBs).

We compare our method with full-shot methods,[8,34,57–63] and the results are shown in Table 5. Our method achieves comparable performance and even outperforms some of the full-shot method with minimal training data and trainable parameters, which demonstrates the effectiveness of our method.

Please cite this article in press as: HU Z et al. Few-shot anomaly detection with adaptive feature transformation and descriptor construction, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.06.007

Algorithm 1. Distance matrix calculation.

```
Require:
  Normal pattern descriptors $\boldsymbol{D} \in \mathbf{R}^{N\times C\times s\times s}$ and query features $\boldsymbol{F}^{\mathrm{Q}} \in \mathbf{R}^{B\times C\times H\times W}$
Ensure:
  Distance matrix between descriptors $\boldsymbol{D}$ and features $\boldsymbol{F}^{\mathrm{Q}}$, denoted as $\boldsymbol{M}^{\mathrm{D}} \in \mathbf{R}^{BHW\times N}$
1. Let sim $= \boldsymbol{F}^{\mathrm{Q}} \otimes \boldsymbol{D}$, where $\otimes$ means operating 2D convolution and interpolating to original shape
2. Reshape sim from $B\times N\times H\times W$ to $BHW\times N$
3. $\boldsymbol{F}^{\mathrm{Q,norm}} = \left(\boldsymbol{F}^{\mathrm{Q}} \odot \boldsymbol{F}^{\mathrm{Q}}\right) \otimes \boldsymbol{O}^{1\times C\times s\times s}$, where $\odot$ is element-wise multiplication and $\boldsymbol{O}$ means 1-filled tensor
4. Reshape $\boldsymbol{F}^{\mathrm{Q,norm}}$ from $B\times 1\times H\times W$ to $BHW\times 1$
5. $\boldsymbol{D}^{\mathrm{norm}} = \boldsymbol{D} \odot \boldsymbol{D}$ reshapes from $N\times C\times s\times s$ to $s^2C\times N$ and sums along the first dimension
6. $\boldsymbol{M}^{\mathrm{D}} = \sqrt{\boldsymbol{F}^{\boldsymbol{Q,norm}} + \boldsymbol{D}^{\boldsymbol{norm}} - 2\mathrm{sim}}/s$
7. return $\boldsymbol{M}^{\mathrm{D}}$
```

4.4. Ablation study

We conduct an ablation study on the MVTec-AD dataset to evaluate our proposed modules. All experiments are conducted under the 4-shot setting.

Component Analysis. In order to evaluate the effect of each proposed module, we conduct an ablation study on each module. For the design of the baseline, we only use the untransformed support features to construct the normal pattern pool, and only use feature patches of size 3 to construct normal pattern descriptors. As shown in Table 6, by adding ADCM, the image-level and pixel-level AUROCs are improved by 5.4 and 0.3 respectively, proving the effectiveness of multi-frequency features. After adding SCTM, the two metrics further improve by 3.4 and 0.3 respectively. Considering that the pixel-level AUROC of the baseline is already high, the overall performance improvement of 0.6 is reasonable. We further conduct a more detailed ablation study on ADCM, as shown in Table 7. The baseline is a single-frequency model without learnable filters and multi-frequency pattern construction. According to the table, the model with multi-frequency descriptors performs significantly better than the single-frequency model on all metrics. Further introducing the learnable filters reduces the model's memory usage to nearly half of the original model while the performance is almost unchanged.

Ablation on Frequencies. In order to verify the effect of multi-frequency normal pattern descriptors, we conduct an ablation study on the frequencies constructed by ADCM. We construct descriptors with only the size of 5 as the baseline, and add descriptors with sizes of 3 and 1 step by step. According to Table 8, by adding normal pattern descriptors of medium frequency and high frequency, the image-level AUROC increases by 5.6 and 11.5 respectively, which is a huge improve-

Table 1 Comparison of pixel-level AD performance on MVTec-AD and VisA. † indicates the method uses a larger feature extractor and introduces external language information, ‡ indicates that the performance is evaluated based on the implementation of Anomalib[56] and official codes due to the lack of available performance results, and bold indicates the best performance among the methods with the same setting.

Dataset	Method	Publication	AUROC		
			1-shot	2-shot	4-shot
MVTec-AD	SPADE[19]	ArXiv 2020	91.2	92.0	92.7
	PaDiM[33]	ICPR 2021	89.3	91.3	92.6
	RegAD[14]	ECCV 2022	–	94.6	95.8
	PatchCore[21]	CVPR 2022	92.0	93.3	94.3
	DDAD[57]	ArXiv 2023	–	85.3‡	88.6‡
	DiffusionAD[58]	ArXiv 2023	79.5‡	82.3‡	85.6‡
	SimpleNet[59]	CVPR 2023	91.5‡	95.6‡	96.1‡
	DeSTSeg[60]	CVPR 2023	91.4‡	92.0‡	93.1‡
	EfficientAD[61]	WACV 2024	83.5‡	86.8‡	87.1‡
	WinCLIP+[†,15]	CVPR 2023	**95.2**	96.0	96.2
	FEAD	Proposed	94.7	**96.1**	**96.6**
Dataset	Method	Publication	AUROC		
			1-shot	2-shot	4-shot
VisA	SPADE[19]	ArXiv 2020	95.6	96.2	96.6
	PaDiM[33]	ICPR 2021	89.9	92.0	93.2
	RegAD[14]	ECCV 2022	70.9‡	73.8‡	77.1‡
	PatchCore[21]	CVPR 2022	95.4	96.1	96.8
	DDAD[57]	ArXiv 2023	–	88.5‡	90.1‡
	DiffusionAD[58]	ArXiv 2023	77.2‡	79.1‡	82.9‡
	SimpleNet[59]	CVPR 2023	92.6‡	93.9‡	92.0‡
	DeSTSeg[60]	CVPR 2023	88.4‡	87.6‡	91.1‡
	EfficientAD[61]	WACV 2024	84.2‡	87.0‡	88.8‡
	WinCLIP+[†,15]	CVPR 2023	96.4	96.8	97.2
	FEAD	Proposed	**96.9**	**97.4**	**97.6**

Table 2 Comparison of image-level AD performance on MVTec-AD and VisA. † indicates the method uses a larger feature extractor and introduces external language information, ‡ indicates that the performance is evaluated based on the implementation of Anomalib[56] and official codes due to the lack of available performance results, and bold indicates the best performance among the methods with the same setting.

Dataset	Method	Publication	1-shot		2-shot		4-shot	
			AUPR	AUROC	AUPR	AUROC	AUPR	AUROC
MVTec-AD	SPADE[19]	ArXiv 2020	90.6	81.0	91.7	82.9	92.5	84.8
	TDG[17]	ICCV 2021	–	–	–	71.2	–	72.7
	PaDiM[33]	ICPR 2021	88.1	76.6	89.3	78.9	90.5	80.4
	DifferNet[16]	WACV 2021	–	–	–	80.6	–	81.3
	RegAD[14]	ECCV 2022	–	–	–	85.7	–	88.2
	PatchCore[21]	CVPR 2022	92.2	83.4	93.8	86.3	94.5	88.8
	DDAD[57]	ArXiv 2023	–	–	82.9‡	81.7‡	85.2‡	85.9‡
	DiffusionAD[58]	ArXiv 2023	77.8‡	74.1‡	81.1‡	79.3‡	83.7‡	81.5‡
	SimpleNet[59]	CVPR 2023	78.5‡	84.1‡	85.4‡	89.6‡	87.5‡	92.3‡
	DeSTSeg[60]	CVPR 2023	74.5‡	76.1‡	76.7‡	80.3‡	78.1‡	79.0‡
	EfficientAD[61]	WACV 2024	84.4‡	68.5‡	84.2‡	68.9‡	83.9‡	72.1‡
	WinCLIP+[†,15]	CVPR 2023	**96.5**	**93.1**	97.0	**94.4**	97.3	**95.2**
	FEAD	Ours	**96.5**	91.6	**97.2**	92.9	**97.9**	94.7
Dataset	Method	Publication	1-shot		2-shot		4-shot	
			AUPR	AUROC	AUPR	AUROC	AUPR	AUROC
VisA	SPADE[19]	ArXiv 2020	82.0	79.5	82.3	80.7	83.4	81.7
	PaDiM[33]	ICPR 2021	68.3	62.8	71.6	67.4	75.6	72.8
	RegAD[14]	ECCV 2022	–	93.5‡	–	94.9‡	–	96.1‡
	PatchCore[21]	CVPR 2022	82.8	79.9	84.8	81.6	87.5	85.3
	DDAD[57]	ArXiv 2023	–	–	83.9‡	84.2‡	84.9‡	84.6‡
	DiffusionAD[58]	ArXiv 2023	80.2‡	73.4‡	82.5‡	78.1‡	85.4‡	80.6‡
	SimpleNet[59]	CVPR 2023	75.2‡	76.5‡	79.9‡	80.1‡	78.9‡	83.0‡
	DeSTSeg[60]	CVPR 2023	62.9‡	63.1‡	58.6‡	59.7‡	66.9‡	61.0‡
	EfficientAD[61]	WACV 2024	71.9‡	64.8‡	71.2‡	63.3‡	75.3‡	69.7
	WinCLIP+[†,15]	CVPR 2023	85.1	83.8	85.8	84.6	**88.8**	87.3
	FEAD	Ours	**86.5**	**84.7**	**87.5**	**86.2**	88.5	**88.1**

Table 3 Categorial AUROC performance comparison on MVTec-AD. † indicates the method uses a larger feature extractor and introduces external language information, bold indicates the best performance among the methods with the same setting, and underline indicates the second best performance.

Category	PaDiM[33]		PatchCore[21]		WinCLIP+[†,15]		FEAD (Proposed)	
	Image	Pixel	Image	Pixel	Image	Pixel	Image	Pixel
Bottle	98.8	97.1	99.2	98.2	99.3	97.8	**99.8**	**98.5**
Cable	70.0	92.1	91.0	**97.5**	90.9	94.9	**94.4**	96.5
Capsule	65.2	96.2	72.8	96.8	82.3	96.2	**93.0**	**98.4**
Carpet	97.9	98.4	96.6	98.6	**100.0**	**99.3**	97.4	99.0
Grid	68.1	77.0	67.7	69.4	**99.6**	**98.0**	90.6	94.6
Hazelnut	91.9	97.2	93.2	97.6	98.4	**98.8**	**99.8**	98.1
Leather	98.5	98.8	97.9	99.1	**100.0**	**99.3**	99.7	99.2
Metal Nut	60.7	82.7	77.7	95.9	**99.5**	92.9	96.9	**96.4**
Pill	54.9	88.9	82.9	94.8	**92.8**	97.1	90.1	**97.2**
Screw	50.0	90.8	49.0	91.3	**87.9**	96.0	73.0	**96.7**
Tile	93.1	88.9	98.5	94.6	**99.9**	**96.6**	99.8	96.0
Toothbrush	89.2	98.4	85.9	98.4	**96.7**	98.4	91.7	**99.2**
Transistor	82.4	**94.0**	90.0	90.7	85.7	88.5	**98.1**	87.7
Wood	97.0	92.2	98.3	93.5	**99.8**	**95.4**	99.1	93.0
Zipper	88.3	96.1	94.0	98.1	94.5	94.2	**97.6**	**98.9**
Mean	80.4	92.6	88.8	94.3	**95.2**	96.2	94.7	**96.6**

Table 4 Categorial AUROC performance comparison on VisA. † indicates the method uses a larger feature extractor and introduces external language information, bold indicates the best performance among the methods with the same setting, and underline indicates the second best performance.

Category	PaDiM[33]		PatchCore[21]		WinCLIP+[†,15]		FEAD (Proposed)	
	Image	Pixel	Image	Pixel	Image	Pixel	Image	Pixel
Candle	77.5	95.4	87.8	97.9	**95.1**	97.8	90.0	**99.3**
Capsules	52.7	79.1	63.4	94.8	**86.8**	97.1	82.2	**98.1**
Cashew	77.7	97.2	93.0	98.3	**95.2**	**98.7**	91.4	97.6
Chewing Gum	83.5	94.4	98.3	96.8	97.7	98.5	**99.1**	**99.3**
Fryum	71.2	95.0	88.6	94.2	90.8	**97.1**	**98.2**	92.9
Macaroni1	65.9	93.5	82.9	97.0	**85.2**	97.0	74.3	**97.4**
Macaroni2	55.0	90.2	61.7	93.9	**70.9**	**97.3**	65.9	95.7
PCB1	82.6	93.2	84.7	98.1	88.3	98.1	**88.4**	**99.4**
PCB2	73.5	93.7	84.3	96.6	67.5	94.6	**87.9**	**96.9**
PCB3	65.9	95.7	**87.0**	**97.4**	83.3	95.8	86.3	97.1
PCB4	85.4	92.1	95.6	97.0	87.6	96.1	**96.8**	**98.3**
Pipe Fryum	82.9	98.5	96.4	**99.1**	**98.5**	98.7	96.9	98.9
Mean	72.8	93.2	85.3	96.8	87.3	97.2	**88.1**	**97.6**

Table 5 Comparison with full-shot methods on AUROC metric. In the table, PDN means the Patch Description Network proposed by EfficientAD[61]. ‡ indicates that the performance is evaluated based on the implementation of Anomalib[56] and official codes due to the lack of available performance results.

Method	Shot	Publication	Backbone	Trainable parameter (M)	MVTec-AD		VisA	
					Image	Pixel	Image	Pixel
FEAD	1	Proposed	WideResNet-50	0.6	91.6	94.7	84.7	96.9
	2				92.9	96.1	86.2	97.4
	4				94.7	96.6	88.1	97.6
SimpleNet[59]	Full	CVPR 2023	WideResNet-50	3.9	99.6	98.1	–	–
EfficientAD[61]	Full	WACV 2024	PDN	5.4	98.8	96.8	97.5	97.6‡
DeSTSeg[60]	Full	CVPR 2023	ResNet-18	32.4	98.6	97.9	–	–
DDAD[57]	Full	ArXiv 2023	Unet	32.	99.8	98.1	98.9	97.6
PNI[8]	Full	ICCV 2023	WideResNet-101	126.0	99.6	99.0	–	–
MSFlow[62]	Full	TNNLS 2024	WideResNet-50	144.1	99.7	98.8	95.2	97.8
DiffusionAD[58]	Full	ArXiv 2023	Unet	160.0	99.7	98.7	98.8	98.9
Cflow[34]	Full	WACV 2022	WideResNet-50	169.7	98.3	98.6	91.5	98.0‡
CS-Flow[63]	Full	WACV 2022	EfficientNet-B5	292.8	98.7	92.1	85.0‡	92.6‡

Table 6 Ablation study of our proposed modules. The experiments are conducted on MVTec-AD under the 4-shot setting. We report the image-level and pixel-level AUROC metrics.

Setting	Image	Pixel
Baseline	85.9	96.0
+ADCM	91.3 (**+5.4**)	96.3 (**+0.3**)
+SCTM	94.7 (**+3.4**)	96.6 (**+0.3**)

ment. The pixel-level AUROC also increases by 0.8 and 0.7, demonstrating the effectiveness of multi-frequency descriptors.

Ablation on Descriptor Downsample Rate. We further evaluate the impact of descriptor downsampling on the model performance. We only construct the medium frequency descriptors and downsample them with M initialized with different strides $s \in \{1, \ldots, 6\}$. According to Fig. 8(a), the pixel-level AUROC is almost unaffected by downsampling, while the image-level AUROC does not decline significantly until the downsampling rate reaches 0.2. This demonstrates that our downsampling strategy is able to reduce memory consumption while keeping the performance loss with an acceptable range.

Ablation on Number of Reference Descriptors. We look into the impact of reference descriptor numbers on the image-level AUROC. We set the number of reference descriptors $k \in \{1, 2, 3, 6, 9\}$ for the three groups of descriptors respectively and the results are shown in Fig. 8 (b). It can be found that setting only $k = 1$ can make the models perform

Table 7 Ablation study on ADCM. The experiments are conducted on MVTec-AD under the 4-shot setting.

Setting	Image AUPR	Image AUROC	Pixel AUROC	Memory Usage
Single Frequency	94.8	89.4	95.0	2.7 GiB
+ Multi-frequency	97.7 (+**2.9**)	94.9 (+**5.5**)	96.8 (+**1.8**)	5.3 GiB
+ Filter	97.9 (+**3.1**)	94.7 (+**5.3**)	96.6 (+**1.6**)	2.8 GiB

Table 8 Ablation study on frequencies of normal pattern descriptors. The experiments are conducted on MVTec-AD under the 4-shot setting. We report the image-level and pixel-level AUROC metrics.

Setting	Image	Pixel
5 × 5	77.6	95.1
+3 × 3	83.2 (+**5.6**)	95.9 (+**0.8**)
+1 × 1	94.7 (+**11.5**)	96.6 (+**0.7**)

considerably well, and the models achieve best image-level AUROC when $k = 9$. Considering that the performance improvement of the medium and low frequency descriptors from $k = 3$ to $k = 9$ is not much, we set $k = 3$ for them to balance the computational complexity.

Ablation on Learning Rate. We study the impact of the learning rate of the training process on the model performance. We set the learning rate to $\mathrm{lr} = \left\{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\right\}$ and keep the other settings unchanged. As shown in Fig. 8(c), The performance changes slightly with the learning rate, demonstrating the stability of our method.

4.5. Qualitative analysis

The visualization of results on MVTec-AD and VisA are shown in Fig. 9. In addition to our method, we also provide the results of PaDiM[33] and PatchCore[21] for reference. All results are obtained under the 4-shot setting. It can be found that compared to other methods, the response of the heatmap of our method is more concentrated around the ground truth. More specifically, PaDiM tends to give a wide range of responses, and the locations of high response areas are not accurate enough, while our method responds with finer granularity and more accurate locations. In addition, compared with PatchCore, our method has fewer irrelevant responses within the background and is more accurate.

We also provide visualization results of the ablation experiments on frequency, as shown in Fig. 10. It can be found that by using multiple frequencies, the model perceives anomalies more accurately and is more sensitive to textural and small anomalies.

5. Conclusions

In this work, we introduce a few-shot AD method FEAD, which extracts multi-frequency normal patterns from few-shot samples under the guidance of the input sample. We propose SCTM, which transforms support features to ensure better coverage of normal patterns, and ADCM, which adaptively constructs multi-frequency pattern descriptors with minimum redundancy. Experiments on two widely-used industrial AD datasets (MVTec-AD and VisA) have validated the effectiveness of the proposed FEAD. Compared to existing few-shot AD methods that only extract features in a single frequency without further transformation, FEAD can achieve a better coverage of normal patterns in the matching process, reducing the ambiguity in detecting anomalies. We hope that FEAD can facilitate the implementation of quality control systems in industrial manufacturing processes.

Limitations and Future Work. The proposed FEAD can be used to localize anomalies in the UAV manufacturing process. However, to the best of our knowledge, there has been no publicly available AD dataset for UAV production. Consequently, we validate the effectiveness of FEAD on other publicly available industrial AD datasets instead. We leave the validation of

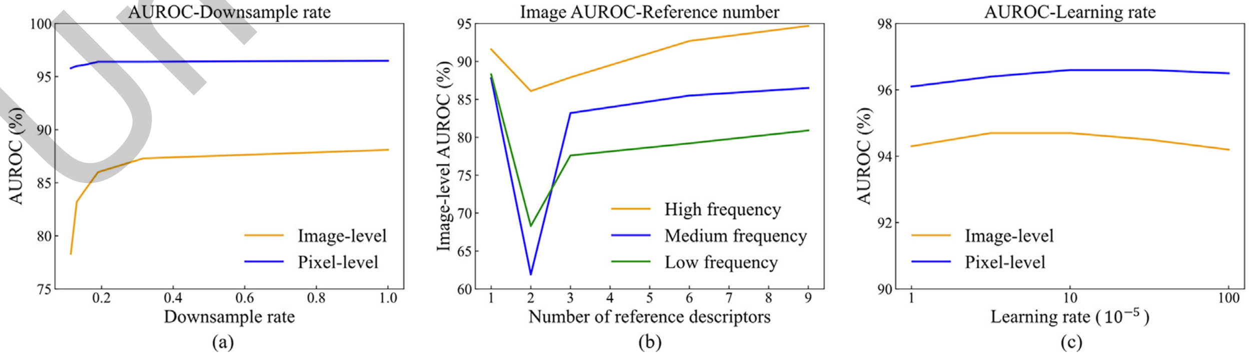


Fig. 8 Line Charts of the ablation study results. (a) Model performance as a function of descriptor downsampling rate. (b) Image-level AUROC as a function of reference descriptors numbers. (c) Model performance as a function of learning rate. Better view in color.

Please cite this article in press as: HU Z et al. Few-shot anomaly detection with adaptive feature transformation and descriptor construction, *Chin J Aeronaut* (2024), https://doi.org/10.1016/j.cja.2024.06.007

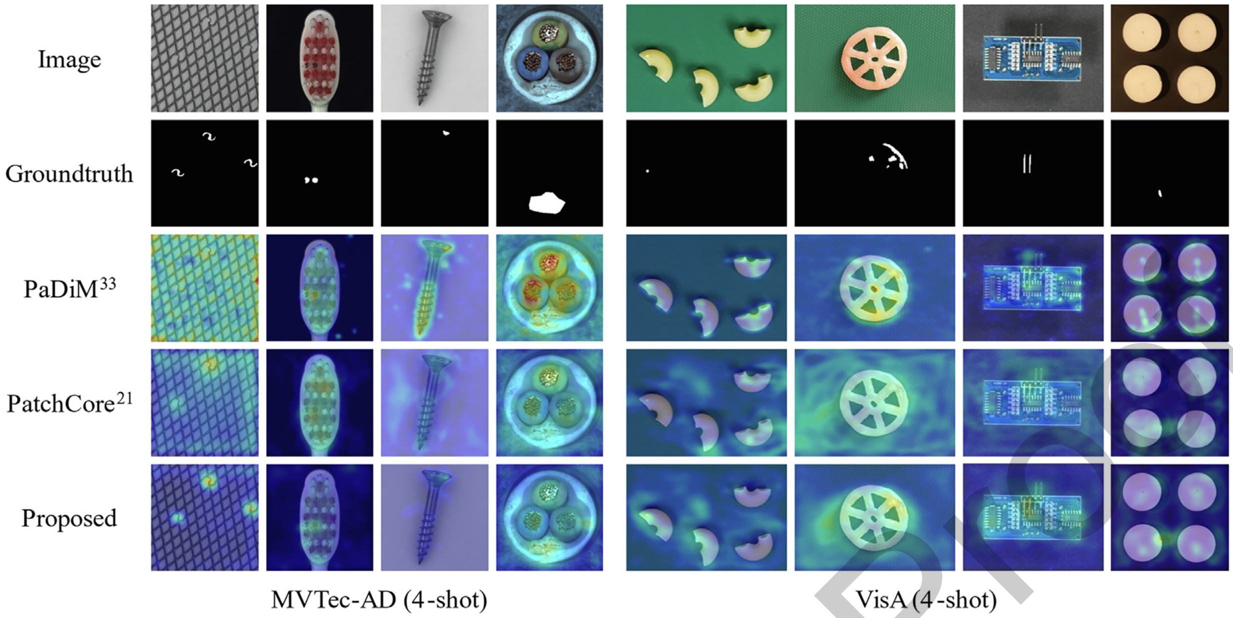


Fig. 9 Visualization of AD results. High-response areas in the heatmap indicate where the anomalies occur. Better view in color and zoom in.

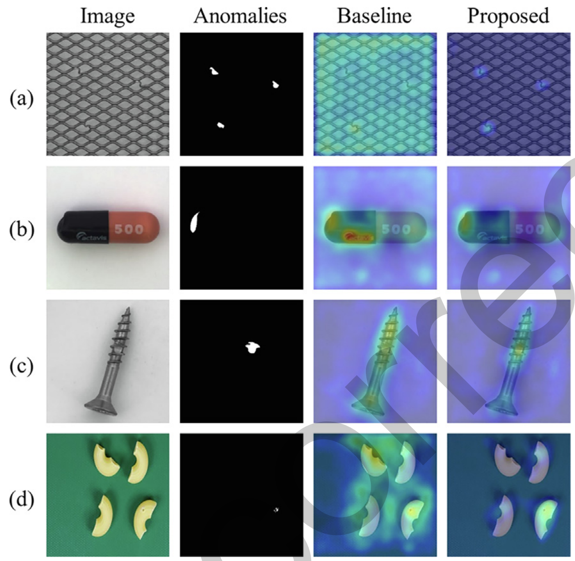


Fig. 10 Visualization of the ablation experiments results of various frequency features. The baseline corresponds to the first row (5×5) in Table 8, and ours corresponds to the third row $(+1 \times 1)$. (a) to (d) show the anomalies on textures, shape anomaly of a single object, textual anomalies of a single object and the case of multiple objects, respectively.

the effectiveness of FEAD on UAV manufacturing AD datasets as a future work. Furthermore, although FEAD localize anomalies in multiple frequencies by constructing normal pattern descriptors in multiple frequency domains, these descriptors work independently in each frequency without interaction with others, which may bring complementary information. A potential future work will be designing a cascaded paradigm in the descriptor matching stage to achieve a coarse-to-fine localization, which utilizes low-frequency descriptors to coarsely localize the anomalies and adopts high-frequency descriptors to finely revise the boundary parts of the anomalies.

CRediT authorship contribution statement

Zhengnan HU: Writing – original draft, Resources, Investigation, Formal analysis, Conceptualization. **Xiangrui ZENG:** Resources, Methodology. **Yiqun LI:** Methodology. **Zhouping YIN:** Supervision, Funding acquisition. **Erli MENG:** Project administration. **Leyan ZHU:** Writing – review & editing, Methodology. **Xianghao KONG:** Writing – original draft, Methodology.

Declaration of competing interest

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

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 52188102).

References

1. Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. *ACM Comput Surv* 2009;**41**(3):1–58.
2. Kähler F, Schmedemann O, Schüppstuhl T. Anomaly detection for industrial surface inspection: application in maintenance of aircraft components. *Procedia CIRP* 2022;**107**:246–51.
3. Fu S, Zhong SS, Lin L, et al. A re-optimized deep auto-encoder for gas turbine unsupervised anomaly detection. *Eng Appl Artif Intel* 2021;**101**:104199.

4. Liu XF, Chen YJ, Xiong LQ, et al. Intelligent fault diagnosis methods toward gas turbine: a review. *Chin J Aeronaut* 2023;**37**(4):93–120.
5. He Y, Heng L, Zhang ZY, et al. Advances and trends on tube bending forming technologies. *Chin J Aeronaut* 2012;**25**(1):1–12.
6. Liu LS, Peng Y, Wang LL, et al. Improving EGT sensing data anomaly detection of aircraft auxiliary power unit. *Chin J Aeronaut* 2020;**33**(2):448–55.
7. Kumar A, Gandhi C, Hesheng T, et al. Adaptive sensitive frequency band selection for VMD to identify defective components of an axial piston pump. *Chin J Aeronaut* 2022;**35**(1):250–65.
8. Bae JH, Lee JH, Kim SY. PNI: Industrial anomaly detection using position and neighborhood information. In: *ICCV 2023: Proceedings of the IEEE/CVF international conference on computer vision*; 2023 Oct 1-6; Paris, France. Piscataway: IEEE Press; 2023. p. 6373–83.
9. Guo HW, Ren LP, Fu JJ, et al. Template-guided hierarchical feature restoration for anomaly detection. In: *ICCV 2023: Proceedings of the IEEE/CVF international conference on computer vision*; 2023 Oct 1-6; Paris, France. Piscataway: IEEE Press; 2023. p. 6447–58.
10. Lei JR, Hu XB, Wang Y, et al. PyramidFlow: High-resolution defect contrastive localization using pyramid normalizing flow. In: *CVPR 2023: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; 2023 Jun 18-22; Vancouver, Canada. Piscataway: IEEE Press; 2023. p. 14143–52.
11. Mcintosh D, Albu A. Inter-realization channels: Unsupervised anomaly detection beyond one-class classification. In: *ICCV 2023: Proceedings of the IEEE/CVF international conference on computer vision*; 2023 Oct 1-6; Paris, France. Piscataway: IEEE Press; 2023. p. 6285–95.
12. Yao XC, Li RQ, Qian ZF, et al. Focus the discrepancy: Intra- and inter-correlation learning for image anomaly detection. In: *ICCV 2023: Proceedings of the IEEE/CVF international conference on computer vision*; Paris, France. Piscataway: IEEE Press; 2023. p. 6803–13.
13. Zhao Y. Omnial: A unified CNN framework for unsupervised anomaly localization. In: *ICCV 2023: Proceedings of the IEEE/CVF international conference on computer vision*; 2023 Oct 1-6; Paris, France. Piscataway: IEEE Press; 2023. p. 3924–33.
14. Huang CQ, Guan HY, Jiang AF, et al. Registration based few-shot anomaly detection. In: *ECCV 2022: European conference on computer vision*; 2022 Oct 23-27; Tel Aviv, Israel. Cham: Springer; 2022. p. 303–19.
15. Jeong JH, Zou Y, Kim TW, et al. WinCLIP: Zero-/few-shot anomaly classification and segmentation. In: *CVPR 2023: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; 2023 Jun 18-22; Vancouver, Canada. Piscataway: IEEE Press; 2023. p. 19606–16.
16. Rudolph M, Wandt B, Rosenhahn B. Same same but differnet: Semi-supervised defect detection with normalizing flows. In: *WACV 2021: Proceedings of the IEEE/CVF winter conference on applications of computer vision*; 2021 Jan 5-9; Virtual Event. Piscataway: IEEE Press; 2021. p. 1907–16.
17. Sheynin S, Benaim S, Wolf L. A hierarchical transformation-discriminating generative model for few shot anomaly detection. In: *ICCV 2021: Proceedings of the IEEE/CVF international conference on computer vision*; 2021 Oct 11-17; Virtual Event. Piscataway: IEEE Press; 2021. p. 8495–504.
18. Lu FB, Yao XF, Fu CW, et al. Removing anomalies as noises for industrial defect localization. In: *ICCV 2023: Proceedings of the IEEE/CVF International Conference on Computer Vision*; 2023 Oct 1-6; Paris, France. Piscataway: IEEE Press; 2023. p. 16166–75.
19. Cohen N, Hoshen Y. Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint: 2005.02357; 2020.
20. Li CL, Sohn KH, Yoon JS, et al. Cutpaste: Self-supervised learning for anomaly detection and localization. In: *CVPR 2021: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; 2021 Jun 19-25; Virtual Event. Piscataway: IEEE Press; 2021. p. 9664–74.
21. Roth K, Pemula L, Zepeda J, et al. Towards total recall in industrial anomaly detection. In: *CVPR 2022: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; 2022 Jun 19-24; New Orleans, USA. Piscataway: IEEE Press; 2022. p. 14318–28.
22. Yi JH, Yoon SR. Patch SVDD: Patch-level SVDD for anomaly detection and segmentation. In: *ACCV 2020: Proceedings of the Asian conference on computer vision*; 2020 Nov 30-Dec 4; Virtual Event. Cham: Springer; 2019. p. 1-16.
23. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint: 2010.11929; 2020.
24. He KM, Zhang XY, Ren SQ, et al. Deep residual learning for image recognition. In: *CVPR 2016: Proceedings of the IEEE conference on computer vision and pattern recognition*; 2016 Jun 26-Jul 1; Las Vegas, USA. Piscataway: IEEE Press; 2016. p. 770–8.
25. Zagoruyko S, Komodakis N. Wide residual networks. arXiv preprint:1605.07146; 2016.
26. Bergmann P, Fauser M, Sattlegger D, et al. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In: *CVPR 2019: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; 2019 Jun 16-20; Long Beach, USA. Piscataway: IEEE Press; 2019. p. 9592–600.
27. Zou Y, Jeong JH, Pemula L, et al. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: *ECCV 2022: European conference on computer vision*; 2022 Oct 23-27; Tel Aviv, Israel. Cham: Springer; 2022. p. 392–408.
28. Akcay S, Atapour-abarghouei A, Breckon T. Ganomaly: Semi-supervised anomaly detection via adversarial training. In: *ACCV 2018: Asian conference on computer vision*; 2018 Dec 2-6; Perth, Australia. Cham: Springer; 2019. p. 622–37.
29. Gong D, Liu LQ, Le V, et al. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: *ICCV 2019: Proceedings of the IEEE/CVF international conference on computer vision*; 2019 Oct 27-Nov 2; Seoul, Korea. Piscataway: IEEE Press; 2019. p. 1705–14.
30. Huang CQ, Xu QW, Wang YF, et al. Self-supervised masking for unsupervised anomaly detection and localization. *IEEE Trans Multimedia* 2022;**45**:4426–38.
31. Wu JC, Chen DJ, Fuh CS, et al. Learning unsupervised metaformer for anomaly detection. In: *ICCV 2021: Proceedings of the IEEE/CVF international conference on computer vision*; 2021 Oct 11-17; Virtual Event. Piscataway: IEEE Press; 2021. p. 4369–78.
32. Ye F, Huang CQ, Cao JK, et al. Attribute restoration framework for anomaly detection. *IEEE Trans Multimedia* 2020;**24**:116–27.
33. Defard T, Setkov A, Loesch A, et al. PaDiM: a patch distribution modeling framework for anomaly detection and localization. In: *ICPR 2021: International Conference on Pattern Recognition*; 2021 Jan 10-15; Virtual Event. Cham: Springer; 2021. p. 475–89.
34. Gudovskiy D, Ishizaka S, Kozuka K. CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: *WACV 2022: Proceedings of the IEEE/CVF winter conference on applications of computer vision*; 2022 Jan 4-8; Waikoloa, USA. Piscataway: IEEE Press; 2022. p. 98–107.
35. Salehi M, Sadjadi N, Baselizadeh S, et al. Multiresolution knowledge distillation for anomaly detection. In: *CVPR 2021:*

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2021 Jun 19-25; Virtual Event. Piscataway: IEEE Press; 2021. p. 14902–12.

36. Zhai SF, Cheng Y, Lu WN, et al. Deep structured energy based models for anomaly detection. In: *ICML 2016: International conference on machine learning*; Jul 19-24; New York, USA. New York: ACM; 2016. p. 1100–9.
37. Zheng Y, Wang X, Deng R, et al. Focus your distribution: Coarse-to-fine non-contrastive learning for anomaly detection and localization. In: *ICME 2022: IEEE international conference on multimedia and expo*; 2022 Jul 18-22; Taipei, Taiwan. Piscataway: IEEE Press; 2022. p. 1–6.
38. Zong B, Song Q, Min M, et al. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In: *ICLR 2018: International conference on learning representations*; 2018 Apr 30-May 3; Vancouver Canada. New York: ACM; 2018.
39. Ruff L, Vandermeulen R, Goernitz N, et al. Deep one-class classification. In: *ICML 2018: International conference on machine learning*; 2018 Jul 10-15; Stockholmsmässan, Sweden. New York: ACM; 2018. p. 4393–402.
40. Tax D, Duin R. Support vector data description. *Mach Learn* 2004;**54**:45–66.
41. An JW, Cho SZ. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE* 2015;**2**(1):1–18.
42. Kingma D, Welling M. Auto-encoding variational bayes. arXiv preprint:1312.6114; 2013.
43. Zhou C, Paffenroth R. Anomaly detection with robust deep autoencoders. In: *KDD 2017: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*; 2017 Aug 13-17; Halifax, Canada. New York: ACM; 2017. p. 665–74.
44. Zenati H, Romain M, Foo C, et al. Adversarially learned anomaly detection. In: *ICDM 2018: IEEE International conference on data mining*; 2018 Nov 17-20; Singapore, Singapore. Piscataway: IEEE Press; 2021. p. 727–36.
45. Schlegl T, Seeböck P, Waldstein S, et al. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: *IPMI 2017: International conference on information processing in medical imaging*; 2017 Jun 25-30; Boone, USA. Cham: Springer. 2017. p. 146–57.
46. Schlegl T, Seeböck P, Waldstein S, et al. F-ANOGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Med Image Anal* 2019;**54**:30–44.
47. Zavrtanik V, Kristan M, Skočaj D. Reconstruction by inpainting for visual anomaly detection. *Pattern Recogn* 2021;**112**:107706.
48. Zhang XY, Li NQ, Li JW, et al. Unsupervised surface anomaly detection with diffusion probabilistic model. In: *ICCV 2023: Proceedings of the IEEE/CVF international conference on computer vision*; 2023 Oct 1-6; Paris, France. Piscataway: IEEE Press; 2023. p. 6782–91.
49. Shin WS, Lee JH, Lee TH, et al. Anomaly detection using score-based perturbation resilience. In: *ICCV 2023: Proceedings of the IEEE/CVF international conference on computer vision*; Paris, France. Piscataway: IEEE Press; 2023. p. 23372–82.
50. Radford A, Kim J, Hallacy C, et al. Learning transferable visual models from natural language supervision. In: *ICML 2021: International conference on machine learning*; 2021 Jul 18-24; Virtual Event. New York: ACM; 2021. p. 8748–63.
51. Jaderberg M, Simonyan K, Zisserman A, et al. Spatial transformer networks. In: *NIPS 2015: Advances in neural information processing systems 28*; 2015 Dec 7-12; Montreal, Canada. New York: ACM; 2015.
52. Zavrtanik V, Kristan M, Skočaj D. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In: *ICCV 2021: Proceedings of the IEEE/CVF international conference on computer vision*; 2021 Oct 11-17; Virtual Event. Piscataway: IEEE Press; 2021. p. 8330–9.
53. Perlin K. An image synthesizer. *ACM Siggraph Computer Graphics* 1985;**19**(3):287–96.
54. Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database. In: *CVPR 2009: IEEE conference on computer vision and pattern recognition*; 2009 Jun 20-25; Miami, USA. Piscataway: IEEE Press; 2009. p. 248–55.
55. Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv preprint:1711.05101; 2017.
56. Akcay S, Ameln D, Vaidya A, et al. Anomalib: A deep learning library for anomaly detection. In: *ICIP 2022: IEEE international conference on image processing;* 2022 Oct 16-19; Bordeaux, France. Piscataway: IEEE Press; 2022. p. 1706–10.
57. Mousakhan A, Brox T, Tayyub J. Anomaly detection with conditioned denoising diffusion models. arXiv preprin:2305.15956; 2023.
58. Zhang H, Wang Z, Wu ZX, et al. Diffusionad: Denoising diffusion for anomaly detection. arXiv preprin:2303.08730; 2023.
59. Liu ZK, Zhou YM, Xu YS, et al. Simplenet: A simple network for image anomaly detection and localization. In: *CVPR 2023: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; 2023 Jun 18-22; Vancouver, Canada. Piscataway: IEEE Press; 2023. p. 20402–11.
60. Zhang X, Li SY, Li X, et al. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In: *CVPR 2023: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; 2023 Jun 18-22; Vancouver, Canada. Piscataway: IEEE Press; 2023. p. 3914–23.
61. Batzner K, Heckler L, König R. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In: *WACV 2024: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*; 2024 Jan 4-8; Waikoloa, USA. Piscataway: IEEE Press; 2024. p. 128–38.
62. Zhou YX, Xu X, Song JK, et al. Msflow: multiscale flow-based framework for unsupervised anomaly detection. *IEEE Trans Neural Networks Learn Syst* 2024.
63. Rudolph M, Wehrbein T, Rosenhahn B, et al. Fully convolutional cross-scale-flows for image-based defect detection. In: *WACV 2022: Proceedings of the IEEE/CVF winter conference on applications of computer vision*; 2022 Jan 4-8; Waikoloa, USA. Piscataway: IEEE Press; 2022. p. 1088–97.