JUSTC

Article

http://justc.ustc.edu.cn

Received: January 01, 2023; Accepted: March 22, 2023

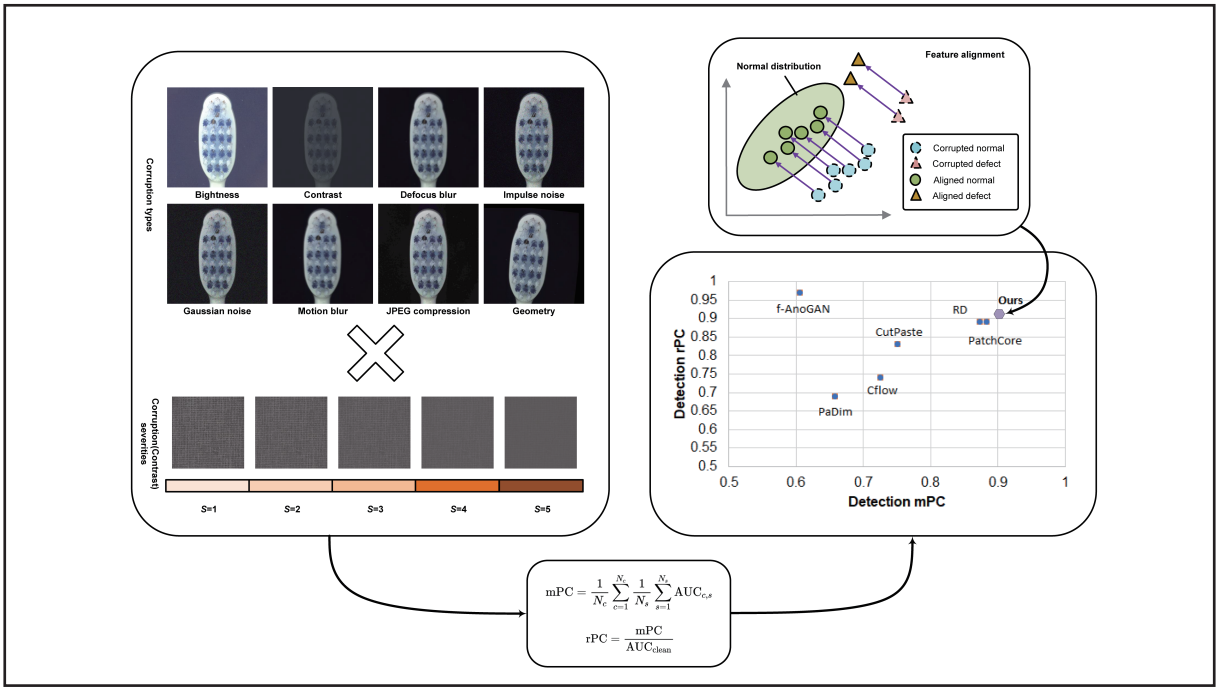
Robustness benchmark for unsupervised anomaly detection models

Pei Wang[1], Wei Zhai[1], and Yang Cao[1,2] ✉

[1]*Department of Automation, University of Science and Technology of China, Hefei 230027, China;*
[2]*Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China*

✉Correspondence: Yang Cao, E-mail: forrest@ustc.edu.cn

Graphical abstract



Benchmarking robustness in unsupervised anomaly detection: dataset, metrics, comparative analysis of existing methods, and enhancing with feature aligning.

Public summary

- We construct a robustness benchmark for unsupervised anomaly detection methods, including a dataset with eight corruption types and five severity levels and metrics to assess robustness.
- We evaluate the accuracy and robustness of mainstream unsupervised anomaly detection methods and find that representation similarity-based and knowledge distillation-based approaches are the best paradigms in terms of performance and robustness.
- The different components of the two best-performed methods are studied for ablation, thus helping to understand the impact of different factors on robustness.
- We propose a feature alignment module to rectify the corrupted features. Combining the proposed module with PatchCore yields a model with both robustness while maintaining high performance.

Citation: Wang P, Zhai W, Cao Y. Robustness benchmark for unsupervised anomaly detection models. *JUSTC*, **2024**, 54(1): 0103. DOI: 10.52396/JUSTC-2022-0165

JUSTC

Article

http://justc.ustc.edu.cn

Received: January 01, 2023; Accepted: March 22, 2023

Robustness benchmark for unsupervised anomaly detection models

Pei Wang[1], Wei Zhai[1], and Yang Cao[1,2] ✉

[1]*Department of Automation, University of Science and Technology of China, Hefei 230027, China;*
[2]*Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China*

✉Correspondence: Yang Cao, E-mail: forrest@ustc.edu.cn

Cite This: *JUSTC*, **2024**, 54(1): 0103 (10pp)

Read Online

Abstract: Due to the complexity and diversity of production environments, it is essential to understand the robustness of unsupervised anomaly detection models to common corruptions. To explore this issue systematically, we propose a dataset named MVTec-C to evaluate the robustness of unsupervised anomaly detection models. Based on this dataset, we explore the robustness of approaches in five paradigms, namely, reconstruction-based, representation similarity-based, normalizing flow-based, self-supervised representation learning-based, and knowledge distillation-based paradigms. Furthermore, we explore the impact of different modules within two optimal methods on robustness and accuracy. This includes the multi-scale features, the neighborhood size, and the sampling ratio in the PatchCore method, as well as the multi-scale features, the MMF module, the OCE module, and the multi-scale distillation in the Reverse Distillation method. Finally, we propose a feature alignment module (FAM) to reduce the feature drift caused by corruptions and combine PatchCore and the FAM to obtain a model with both high performance and high accuracy. We hope this work will serve as an evaluation method and provide experience in building robust anomaly detection models in the future.

Keywords: robustness benchmark; anomaly detection; unsupervised learning; automated optical inspection

CLC number: TP391 **Document code:** A

1 Introduction

With the increasing demands on product quality, it is significant to detect surface defects in products during production. In scenarios such as industrial defect detection, there is a high demand for unsupervised anomaly detection models due to the lack of defect samples. Additionally, since the models need to handle image degradation interference during testing, the robustness of the models needs to be evaluated. Many unsupervised anomaly detection methods have been proposed; these methods can be divided into five categories: reconstruction-based[1–3], representation similarity-based[4–6], normalizing flow-based[7, 8], self-supervised representation learning-based[9], and knowledge distillation-based[10–12].

The existing methods can achieve high accuracy on the MVTec[13] dataset that is a relatively simple dataset. However, the robustness of these models is unknown. Due to the temporal and spatial differences between the training and deployment phases, there is no guarantee that the images processed in actual production will have the same quality as the training images and that unpredictable degradation of image quality may occur. For example, devices that run for long periods may wear out, causing the relative position of the camera to the product to deviate from the preset value, which can lead to defocus blur and geometric shifts in the images. Therefore, there is an urgent need for a means of evaluating the robustness of models to ensure production safety. Past studies on the robustness of CNN models have involved image classification[14], object detection[15], semantic segmentation[16], human pose estimation[17], and other domains. Since defect samples are unavailable to unsupervised anomaly detection models during training, specialized research on the robustness of unsupervised anomaly detection models is necessary.

To address the aforementioned needs, we propose a framework that includes a robust evaluation dataset, evaluation metrics, and a method to improve the robustness of the model. It is worth noting that the dataset and evaluation metrics are also applicable for evaluating the robustness of supervised models. Additionally, based on this framework, we evaluate and analyze the robustness of mainstream unsupervised defect detection models and obtain conclusions about the factors affecting robustness.

Inspired by Dan et al.[14], we propose a dataset named MVTec-C with different types of corruptions to investigate the robustness of unsupervised anomaly detection methods systematically (see Fig. 1). We select the corruption types considering the uncertainties as much as possible in the production scenario, including Gaussian noise, Poisson noise, motion blur, defocus blur, light, contrast, JPEG compression, and geometry, with each corruption corresponding to five severity levels. We consider robustness to mean how well the performance on the original data can be maintained on corrupted data, so we define the relative corruption performance to measure the robustness of the model.

To understand the robustness of existing unsupervised

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

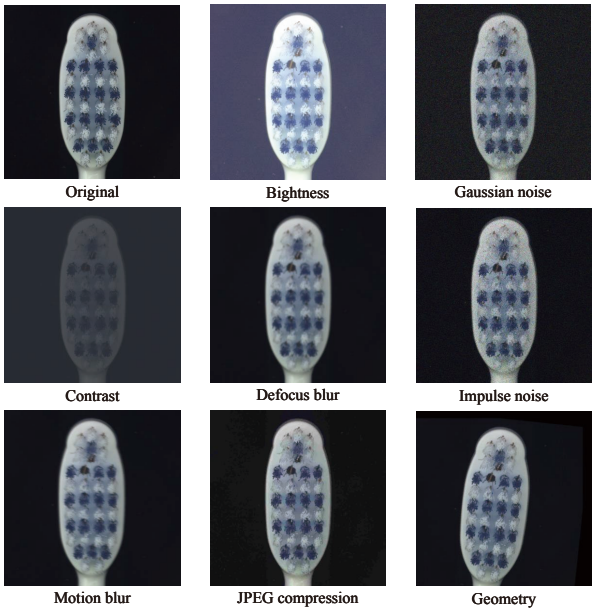


Fig. 1. Samples with different corruption types. The first image is the original image in MVTec.

anomaly detection models, we select one or two representative methods for evaluation from each of the five classes of anomaly detection methods. We find that the representation similarity-based and knowledge distillation-based methods achieve the best balance between performance and robustness, while the reconstruction-based method shows good robustness at the worst performance. Since PatchCore and Reverse Distillation both have high performance and high robustness, studying of their robustness is valuable for practical applications. Therefore, we investigate the effects of different factors on the robustness of these two methods through exhaustive ablation experiments. We draw several conclusions based on the experimental results: (ⅰ) High-level features can help construct robust feature representations. (ⅱ) Multi-scale features are beneficial for improving performance and robustness. (ⅲ) Neighborhood aggregation can not improve the generalization capability of features. (ⅳ) The memory bank can maintain good robustness even with a small sampling ratio, so it is suitable for modeling a normal sample distribution. (ⅴ) Information bottlenecks make the features more compact and thus improve the robustness of reverse distillation. (ⅵ) Multi-scale distillation can consider different defect sizes and hence can improve robustness.

To construct a robust anomaly detection model, we focus on representation similarity-based methods (which perform best considering robustness and performance simultaneously) and attempt to optimize the feature representation. Defects cause a drift in the features we rely on to discriminate defects. However, corruptions introduce an additional drift that leads to a decrease in the accuracy of detecting defects. Considering the different characteristics of corruptions and defects, we hypothesize that corruptions lead to globally consistent drift, while defects lead to local drift in features. Based on this hypothesis, we propose a feature alignment module (FAM) to reduce the globally consistent drift while preserving the local drift caused by defects, thus obtaining a feature representation that is robust to corruptions. We apply the FAM to PatchCore and significantly improve its robustness while maintaining high performance. Overall, our contributions are as follows:

(Ⅰ) We construct a robust benchmark for unsupervised anomaly detection methods, including a dataset with eight corruption types, five severity levels, and metrics to assess robustness.

(Ⅱ) We evaluate the performance and robustness of mainstream unsupervised anomaly detection methods and find that representation similarity-based and knowledge distillation-based approaches are the best paradigms in terms of performance and robustness.

(Ⅲ) The different components of the two best-performing methods are studied for ablation, thus helping to understand the impact of different factors on robustness.

(Ⅳ) We propose a feature alignment module to rectify the corrupted features. Combining the proposed module with PatchCore yields a model with robustness while maintaining high performance.

2 Related work

2.1 Unsupervised anomaly detection

In recent years, researchers' interest in anomaly detection, especially unsupervised anomaly detection, has increased rapidly, and many datasets and unsupervised anomaly detection algorithms have been proposed. The most frequently used dataset is the MVTec[13] that contains five texture and ten object classes, totaling 5354 images.

To identify defects from images, unsupervised anomaly detection algorithms should produce either the image-level anomaly score required for anomaly detection or the pixel-level anomaly map as required for the anomaly segmentation or both. The mainstream unsupervised detection algorithms can be classified into five categories: reconstruction-based, representation similarity-based, self-supervised representation learning-based, normalizing flow-based, and knowledge distillation-based.

Reconstruction. Reconstruction-based approaches typically use generative adversarial networks, autoencoders, and variational autoencoders to reconstruct the input image under the assumption that a model trained on normal samples can correctly reconstruct only normal regions but not defective regions. AnoGAN[1] used generative adversarial networks to learn the distribution of normal images, and the training phase trains a generator of normal image patches. In the test phase, for each patch in the test image, the hidden vector is iteratively adjusted using the discriminant score and the intermediate features of the discriminator as a guide. Finally, the residual map of the test image and the output image of the generator are combined with the residual loss of the discriminator's intermediate features to identify defects. The f-AnoGAN[18] used an autoencoder to generate the hidden vector, avoiding the problem of time-consuming iterative optimization in AnoGAN. Ganomaly[2] detected defects based on an encoder-decoder-encoder structure by comparing the encoding of the first encoder with the encoding reconstructed by the second

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

encoder. AE-SSIM[3] introduced the structural similarity index measure (SSIM) as the reconstruction loss of the autoencoder. To address the problem that defects are unexpectedly well reconstructed in the autoencoder-based approaches, RIAD[19] proposed training an the autoencoder by recovering images that are partially erased.

Representation similarity. In representation similarity-based methods, the normal feature distribution is first explicitly modeled during training. K nearest neighbors or Mahalanobis distance is used to calculate the similarity between test features and normal features during testing. Samples with low similarity are considered defective. GaussianAD[20] proposed the use of a multivariate Gaussian model to describe the distribution of pre-trained features of normal samples. SPADE[4] saved patch-level features and image-level features extracted from ResNet; K nearest neighbors searched from image-level normal features are used to compute anomaly scores; and patch features of these selected images are further used to compute patch-level anomaly maps. PaDim[5] estimated a multivariate Gaussian model at each patch position and uses a random dimension selection strategy to reduce the size of the patch features. PatchCore[6] employed a position-independent feature memory bank and uses a greedy selection strategy to reduce the size of the memory bank.

Self-supervised representation learning. Several methods[21–23] are dedicated to learning discriminative representations through proxy tasks such as predicting the geometric transformation of the image or contrast learning[24]. However, these methods are limited to learning high-level semantic information. CutPaste[9] constructed anomalies by randomly cropping and pasting image blocks and demonstrates that the network's perception of such anomalies can be generalized to actual defects.

Normalizing flow. The normalizing flow-based approach uses normalizing flows to transform the normal feature distribution into a Gaussian distribution. The network outputs a probability indicating whether the input is within the normal distribution, and a low probability is expected when the input is defective. DifferNet[7] applied a normalizing flow to image-level pre-trained features to obtain the image-level anomaly score. Cflow[8] formed a conditional normalizing flow by incorporating positional encoding in the flow, then the anomaly map is obtained by sliding the conditional normalizing flow over patches.

Knowledge distillation. In knowledge distillation-based approaches, the teacher model and the student model are expected to produce features with differences in the defective images[10–12, 25] that use a set of teacher–student model pairs with different perceptual fields to compute residual maps at multiple scales. Salehi et al.[11] proposed that the distillation of features from an expert network at various layers is preferable to that of a single layer. In these approaches, the teacher and student models use similar structures, leading to a convergence of the teacher and student models' representations of defects. Deng et al.[12] proposed a reverse distillation paradigm to solve this problem.

2.2 Evaluating the robustness of anomaly detection models

In recent years, many works have focused on evaluating the robustness of anomaly-based intrusion detection systems. In intrusion detection scenarios, attacks are applied to anomaly detection models, and the changes in model performance are calculated as robustness criteria. Researchers are dedicated to designing better attacks to evaluate model weaknesses. Goodge et al.[26] designed different types of attacks to evaluate model robustness and improve the robustness to adversarial attacks by optimizing the latent representation. Schneider et al.[27] proposed using different feature robustness metrics, and Han et al.[28] proposed using gray/black-box traffic-space adversarial attacks to evaluate model robustness. To better determine whether the attacked samples are anomalies, Gómez et al.[29] proposed using multiple supporting models and found that the 1D CNN is more robust than LSTM according to their evaluation method. The impact of data poisoning on anomaly detection systems has also been studied[30–32].

However, these studies on the robustness of anomaly detection have focused mainly on intrusion detection scenarios and are difficult to apply to other scenarios for the following reasons. First, most of these works have evaluated the robustness against adversarial attacks or data pollution, but the robustness against common corruption is also important, especially in industrial defect detection scenarios. Second, anomaly detection models in intrusion detection scenarios mainly address one-dimensional time-series data, while image anomaly detection models process multi-dimensional image data, and robust models need to adapt to more complex environmental changes. To the best of our knowledge, research on the robustness of image-based anomaly detection in scenarios such as industrial defect detection is lacking.

2.3 Benchmarking robustness to common corruptions

Many methods have been proposed for studying the robustness of CNNs against common image corruptions[14–17, 33, 34]. To investigate the robustness of different classification networks against common image corruptions and perturbations, Dan et al.[14] proposed two datasets, ImageNet-C and ImageNet-P. Michaelis et al.[15] provide a benchmark for evaluating the performance of a target detection model in the face of image corruptions. Similarly, Kamann et al.[16] accessed the robustness of the semantic segmentation model to real-world image corruptions. Altindis et al.[33] evaluated instance segmentation models with real-world image corruptions and out-of-domain images. Wang et al.[17] constructed three robustness benchmarks to study the drawbacks of human pose estimation models.

3 Methods

3.1 Robustness benchmark

We construct the robust benchmark dataset MVTec-C by adding eight corruption types with five severity levels to each image in the MVTec test set. The corruption types include Gaussian noise, Poisson noise, motion blur, defocus blur, light, contrast, JPEG compression, and geometry. Fig. 1

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

shows some samples from the MVTec-C.

The performance of an anomaly detection model is usually measured using the area under the receiver operator characteristic (AUC). Specifically, image-level AUC is used for image-level anomaly detection, while the performance of anomaly segmentation is determined by the pixel-level AUC. Referring to Michaelis et al.[15], the performance on corrupted data is measured by the mean performance under corruption (mPC):

$$\mathrm{mPC} = \frac{1}{N_c}\sum_{c=1}^{N_c}\frac{1}{N_s}\sum_{s=1}^{N_s}\mathrm{AUC}_{c,s}. \tag{1}$$

Here, N_c denotes the total number of different corruption types included in the test, and N_s represents the number of severity levels for each type of corruption. In this context, c indexes the specific type of corruption, and s indicates the severity level of the corruption. The robustness of a model is defined as its ability to maintain original performance levels under various corruption scenarios. To quantify this, we introduce the relative performance under corruption (rPC), defined as:

$$\mathrm{rPC} = \frac{\mathrm{mPC}}{\mathrm{AUC}_{\mathrm{clean}}}. \tag{2}$$

In the above equation, $\mathrm{AUC}_{\mathrm{clean}}$ refers to the model's AUC performance on clean, uncorrupted data. Therefore, rPC provides a comparative measure of the model's robustness by evaluating its performance under corrupted conditions relative to its performance on clean data.

3.2 Evaluation of representative methods

We evaluate several of the most representative contemporary unsupervised anomaly detection methods using our benchmark and discuss the advantages and disadvantages of these methods. Then, we select two methods that perform best in terms of both performance and robustness combined and investigate the impact of different components of these methods on the models' robustness.

3.2.1 Comparison between different methods

The f-AnoGAN[18] used an encoder to map the image patches into the hidden space and then uses the WGAN to recover the image patches. The output image of the generator and the intermediate features of the discriminator are used to compute the anomaly score. PaDim[5] used features from multiple semantic layers in the ImageNet pre-trained model, described using a multivariate Gaussian model at each patch position, and used the Mahalanobis distance to compute the anomaly score at each location to generate the anomaly map. PatchCore[6], based on ImageNet pre-trained features, used a greedy strategy to keep the maximal representative set of normal patch features and uses kNN to compute the anomaly score for each test patch. CutPaste[9], a data augmentation method is used in CutPaste to learn self-supervised features, a multivariate Gaussian model is used to model the distribution of global features, and the image-level anomaly score is computed using the Mahalanobis distance. Cflow[8] uses conditional normalizing flow to explicitly estimate the anomaly score at each patch location. Reverse Distillation[12] uses teacher–student models to learn the distribution of normal features at multiple scales. The feature maps of the teacher and student models are compared to obtain the anomaly map.

3.2.2 In-depth study of specific methods

To better investigate robustness, it is essential not only to evaluate the mainstream methods' performance in robustness but also to understand the factors that affect robustness and how they influence it. Ablation experiments can be conducted to investigate these factors and achieve this goal. However, the significant differences among the different methods make it difficult to study these factors in a unified framework. Our benchmark testing revealed that PatchCore by Roth et al.[6] and Reverse Distillation by Deng et al.[12] outperform other methods in both accuracy and robustness and thus warrant further research. Therefore, we conducted more thorough research on these two methods separately to explore the factors that affect robustness.

PatchCore. The PatchCore method workflow is shown in Fig. 2. First, local patch features are extracted and aggregated into a memory bank. The mid-level patch features are utilized to minimize bias toward ImageNet classes, while a feature aggregation over a local neighborhood ensures retention of sufficient spatial context. Second, the coreset-reduction method is introduced to reduce redundancy in the extracted patch-level memory bank and significantly reduce the storage memory and inference time. This is achieved through greedy coreset subsampling for nominal feature banks. Finally, the nearest neighbors of the test features are retrieved from the memory bank and compared with the test patch features to compute anomaly scores. We explore the robustness of the model from the following aspects:

- We explore the impact of features of different scales as patch representations.
- Different neighborhood sizes in neighborhood aggregation are compared.
- We compare the results when sampling with different sampling ratios to obtain the memory bank.

Reverse Distillation. Fig. 3 shows the flow of Reverse

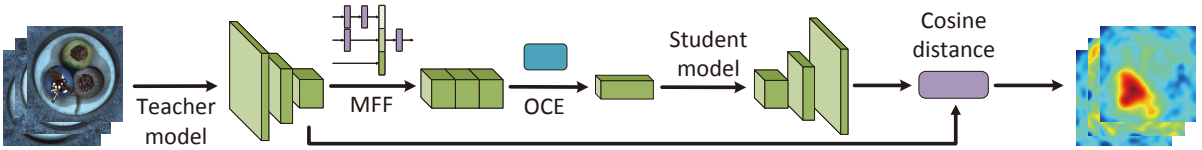


Fig. 2. The pipeline of PatchCore[6]. Neighborhood aggregation enhances the patch representations extracted from a pre-trained model. Then the greedy sampling constructs a compact yet maximal representative memory bank. Finally, each patch is scored by calculating the distance between the patch representation and its nearest neighbor in the memory bank.

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

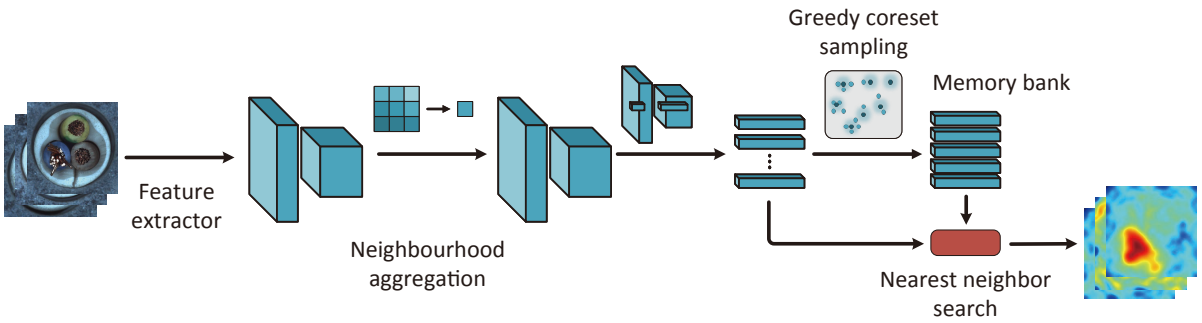


Fig. 3. The pipeline of Reverse Distillation[12]. First, the teacher model acts as a feature extractor and outputs multi-scale features. Secondly, multi-scale features are merged and compressed into low-dimensional embeddings by the multi-scale feature fusion (MFF) and the one-class embedding (OCE) module. Third, the student model tries to reconstruct the multi-scale features output by the teacher model. Finally, the cosine distance is used to compare the outputs of the student and teacher models, resulting in an anomaly map.

Distillation. First, a pre-trained deep neural network is used as a teacher model to extract the feature representations of the original input data. Second, multi-scale feature fusion (MFF) and a method called "one-class embedding" (OCE) are used to transform the extracted features into low-dimensional, compact representations. Third, a student model is used as a decoder to reconstruct the teacher model's representations at different scales from the low-dimensional feature embeddings. Fourth, the output of the student model is compared with the output of the teacher model, and the anomaly score and anomaly localization information are obtained by calculating the reconstruction error. We explore several aspects of the factors affecting the robustness of Reverse Distillation:

- Different scales of features are aggregated and compared in the MFF module.
- We compare the base model, model with OCE, and models with both MFF and OCE.
- Features at different scales are distilled and compared.

3.3 Robust anomaly detection model

To construct a robust anomaly detection model, we try to optimize the feature representation of the representation similarity-based approach, which yields the best performance. Corruptions cause features to deviate from the original distribution, thus making the use of the original discriminant function to distinguish defective samples from normal samples less effective. We assume that the feature drift caused by corruption is globally consistent, while the feature drift caused by defects is local. This assumption is motivated by the fact that, unlike natural scenes, in fields such as industrial defect detection and medical imaging, it is often necessary to analyze and process images of small regions to detect defects or lesions. It is reasonable to assume that the corruption-induced feature drift is globally consistent in such small regions. In contrast, defects or lesions typically results in mutations in a smaller local area of the image, which introduces uneven feature drift. Therefore, the average drift of all patch features in an image can be used to estimate the former but not the latter. By minimizing the average drift, we can compensate significantly for corruption-induced drift. Although, this approach will alter defect features, it is not enough to compensate for the defect drift. Thus, a feature alignment module can be designed to recover uncorrupted features by minimizing the average drift. We illustrate the concept in Fig. 4. To compute the average drift, we need to obtain the features before degradation. Since obtaining such features is not feasible, we approximate uncorrupted features by selecting K nearest uncorrupted neighbors from the memory bank.

Let the corruption transformation be G, the inverse transformation be H, the normal patch feature be $\boldsymbol{x}$, and the degraded patch feature be $\hat{\boldsymbol{x}}$. Then, we have $\boldsymbol{x} = H(G(\boldsymbol{x})) = H(\hat{\boldsymbol{x}})$. If the inverse transformation H can be estimated, the feature $\boldsymbol{x}$ can be returned. We introduce the FAM in the test phase as an estimate of the inverse transformation H; then, the original feature can be estimated as $\boldsymbol{x}^* = \mathrm{FAM}_M(\hat{\boldsymbol{x}})$, where M represents for the parameters of the FAM. We optimize the FAM to minimize the distance between patch features and

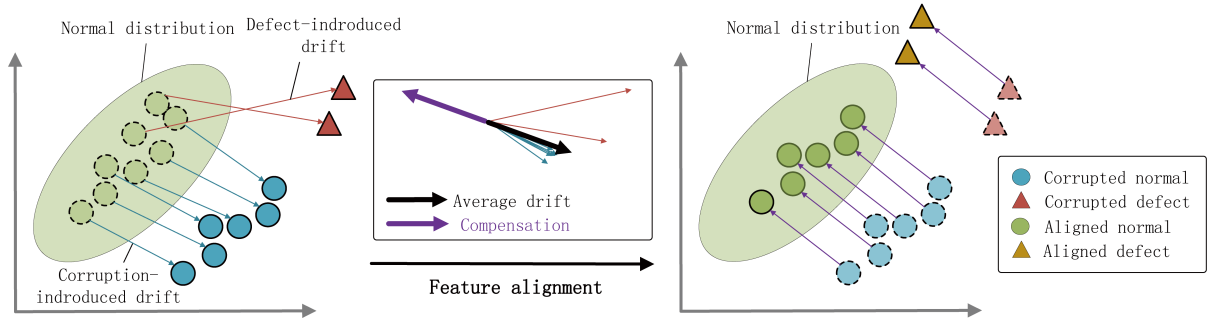


Fig. 4. Corruptions cause a globally consistent drift, while defects cause a local drift from the normal distribution. The average shift of all image patch features tends to align more closely with the drift caused by corruption than that caused by defects. Consequently, minimizing the average shift can effectively reduce the drift induced by corruption by aligning the corrupted features with the normal ones. Although this approach can alter the defect features, it remains insufficient to compensate for the drift caused by defects.

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

their reference features, as follows :

$$M^* = \arg\min_M \left\| \mathrm{FAM}_M(\hat{\boldsymbol{X}}) - \boldsymbol{X}_{\mathrm{ref}} \right\|, \quad (3)$$

where $\boldsymbol{X}$ refers to all the patch features in a single image, and $\boldsymbol{X}_{\mathrm{ref}}$ represents the reference features of the test features. Each test feature corresponds to k reference features selected from the memory bank with kNN. The Euclidean distance between the features transformed by the FAM and their nearest neighbor features in the memory bank is computed to obtain the anomaly score.

The parameter M of the FAM consists of the coefficient vector $\boldsymbol{W}$ and the bias vector $\boldsymbol{B}$, defined as $\mathrm{FAM}(\boldsymbol{x}) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{B}$. It is designed to fit simple transformations, thus avoiding learning semantic information. In addition, the number of iterations n of the optimization process needs to be limited to prevent overfitting. In summary, our FAM module is designed with parameters $\boldsymbol{W}$ and $\boldsymbol{B}$ that need to be optimized online for each test sample during the testing phase, and two other hyperparameters, including the number of reference features k and the number of iterations n.

4 Experiments

4.1 Implementation details

For robustness evaluation, we use publicly available implementations of these methods. Except for f-AnoGAN, WideResnet50 is selected as the backbone network by default. The input image size is 256×256. For PaDim, 550 dimensions are randomly selected from features generated by the backbone. For PatchCore, the memory bank constructed at a sampling ratio of 0.01, and the neighborhood size was 3. For CutPaste, we used a 3-way augmentation method with a two-layer MLP network for the projection head. For Cflow, three pooling layers and eight coupling layers are used. For Reverse Distillation, when fusing features of different scales, the number of output channels of OCE is not mentioned in the original paper, and we set it to a fixed value of 2048. For f-AnoGAN, the dimensionality of the hidden vector is set to 100. When combining the proposed FAM with PatchCore, the FAM transforms the extracted features and calculates the anomaly score by performing the kNN on the memory bank. The initial parameters of the FAM are $\boldsymbol{W} = \boldsymbol{1}$ and $\boldsymbol{B} = \boldsymbol{0}$. Thus it is equivalent to the original PatchCore before FAM is optimized. We use Adam optimizer to adjust the parameters of FAM and set the learning rate to 0.01. All experiments are performed on four NVIDIA GeForce RTX 3090 GPUs.

4.2 Robustness study

4.2.1 Comparison of different methods

The comparison of the results of different methods is shown in Table 1. Based on the experimental results, we make the following analysis: (i) f-AnoGAN[18] has the lowest performance but the highest robustness. We speculate that this is not because the decoder reconstructs the degraded image well but because, in the reconstruction-based paradigm, degradation interferes little with the identification of "easy" defects. (ii) Although both PaDim[5] and PatchCore[6] are methods based on representation similarity, their robustness differs significantly. It is hypothesized that the distribution of the location-dependent patch features of the training samples does not generalize well to the new data due to the lack of location diversity of the images in MVTec. In contrast, a uniform location-independent memory bank is more generalizable. Also, according to Ref. [8], the multivariate Gaussian model does not fit the complex distribution well. (iii) CutPaste[9] also adopts a multivariate Gaussian model, but its image-level robustness greatly exceeds PaDim. This indicates that the features obtained by CutPaste through self-supervised learning are more focused on semantic information and, therefore, less subject to interference from non-semantic degradation. (iv) As one of the highest-performance methods on clean data, Cflow[8] does not perform well in terms of robustness. We speculate that it is because the conditional normalizing flow is sensitive to feature degradation. (v) Reverse Distillation[12]

Table 1. Comparison of the results of the different methods. "clean" refers to the accuracy (AUC) on MVTec. The image-level and pixel-level metrics are displayed for each method. For some methods (f-AnoGAN, CutPaste), we do not find implementations provided by the authors, and the publicly available implementations we used only generate image-level results.

Method	f-AnoGAN[18]	PaDim[5]	Cutpaste[9]	Cflow[8]	Reverse Distillation[12]	PatchCore[6]
clean	0.623/–	0.955/0.973	0.903/–	0.979/0.976	0.980/0.978	**0.990/0.980**
Gaussian noise	0.636/–	0.870/0.913	0.717/–	0.654/0.733	**0.905**/0.912	0.902/**0.936**
Shot noise	0.629/–	0.857/0.915	0.715/–	0.667/0.762	0.897/0.917	**0.903/0.940**
Defocus blur	0.625/–	0.941/0.964	0.836/–	0.815/0.911	**0.971/0.972**	0.960/0.966
Motion blur	0.619/–	0.513/0.686	0.854/–	0.855/0.943	**0.966/0.969**	0.954/0.966
Brightness	0.542/–	0.518/0.685	0.790/–	0.775/0.845	0.947/**0.946**	**0.955**/0.942
Contrast	0.597/–	0.553/0.680	0.636/–	0.635/0.732	**0.859/0.899**	0.845/0.877
JPEG compression	0.623/–	0.518/0.692	0.764/–	0.741/0.821	**0.898/0.925**	0.853/0.914
Geometry	0.567/–	0.492/0.636	0.685/–	0.660/0.895	0.544/0.786	**0.690/0.899**
mPC	0.605/–	0.658/0.771	0.750/–	0.725/0.830	0.873/0.916	**0.883/0.930**
rPC	**0.971**/–	0.689/0.793	0.830/–	0.741/0.851	0.891/0.937	0.892/**0.950**

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

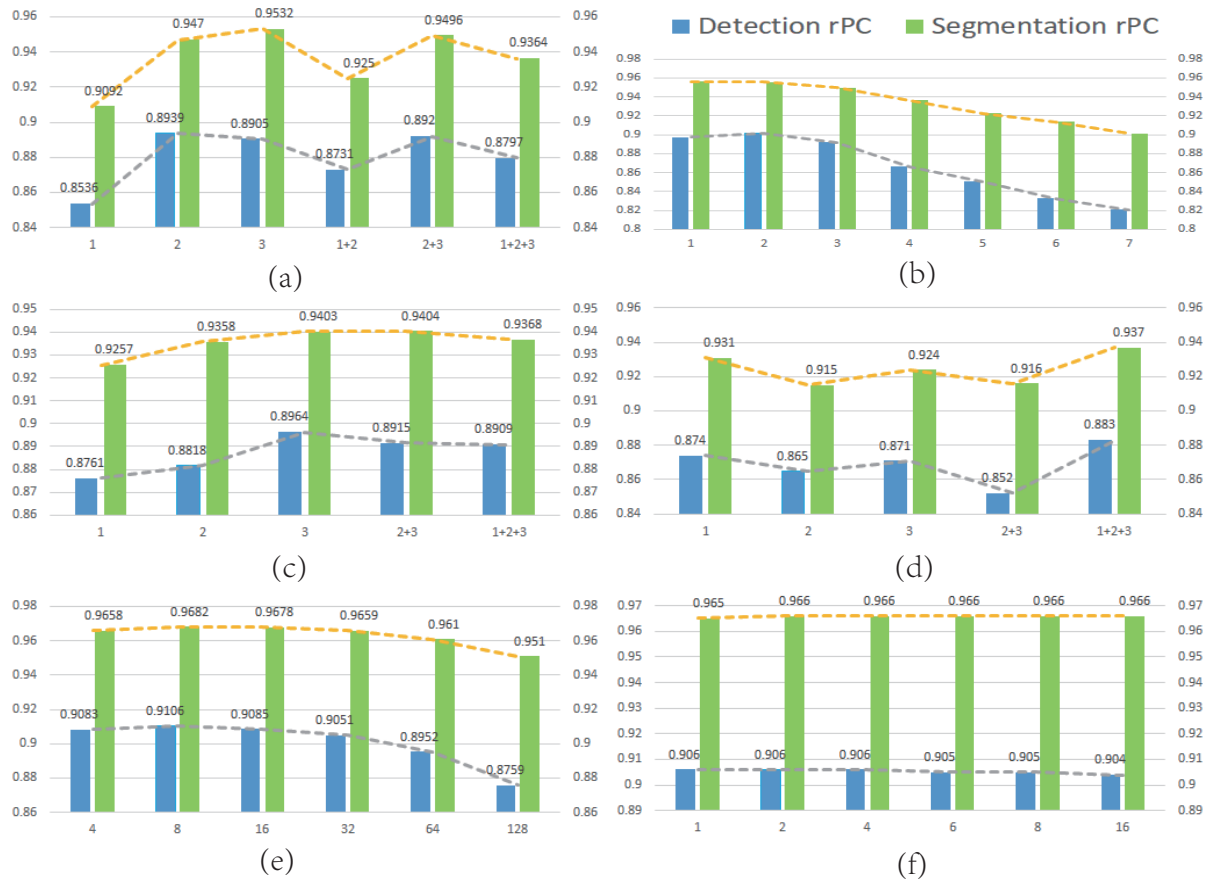


Fig. 5. Analysis experiments. (a) The effect of multi-scale features in PatchCore on robustness. "1", "2", and "3" represent the features from layer1, layer2, and layer3 in ResNet, respectively. (b) The effect of neighborhood size on the robustness of the domain aggregation module in PatchCore. (c) The effect of multi-scale features on robustness in Reverse Distillation. (d) Effect of multi-scale distillation in Reverse Distillation on robustness. (e) Effect of the number of FAM iterations. (f) Effect of the number of FAM reference features.

and PatchCore achieve the best balance between accuracy and robustness. To understand the robustness of these two methods in more detail, we perform more in-depth experiments and analyses below.

4.2.2 In-depth exploration of PatchCore

We analyze the results of multi-scale features, sampling ratio, and neighborhood size, respectively.

Multi-scale features. The results of using features of different scales are shown in Fig. 5a. Layer2 and layer3 features have significantly better robustness than layer1. It is reasonable to assume that features from higher semantic levels are less disturbed by the degradation of image quality and thus have better robustness. As for detection rPC, layer2 is slightly worse than layer3 because layer2 is better at detecting defects, which in turn improves its robustness.

Sampling ratio. The effect of the sampling ratio is shown in Table 2. The robustness increases with the sampling ratio, indicating that increasing the number of sampled features helps to generalize the memory bank to degenerate features. Even if only 0.01 of normal features are sampled, the robustness is only slightly lower than when 0.4 of normal features are sampled, indicating that the memory bank obtained by greedy sampling is a good representation of the original normal sample and thus a robust way to model the original normal distribution.

Table 2. Impact of samplings ratio in PatchCore.

Sampling ratio	0.01	0.04	0.1	0.4
Detection rPC	0.892	0.898	0.9	0.904
Segmentation rPC	0.95	0.951	0.952	0.955

Neighborhood size. As shown in Fig. 5b, when the neighborhood size exceeds 2, the robustness gradually decreases as the neighborhood size increases. Though neighborhood aggregation improves the performance[6], it does not help improve robustness. A reasonable speculation is that aggregating more local information does not alleviate the feature shift caused by corruptions. At the same time, too large neighborhood size leads to patch features being damaged, which is detrimental to anomaly detection.

4.2.3 In-depth exploration of Reverse Distillation

The results of multi-scale features, OCE and MFF module, and multi-scale distillation are analyzed.

Multi-scale features. As shown in Fig. 5c, using higher semantic features leads to better robustness, similar to PatchCore[6]. This is because the higher-level features are less

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

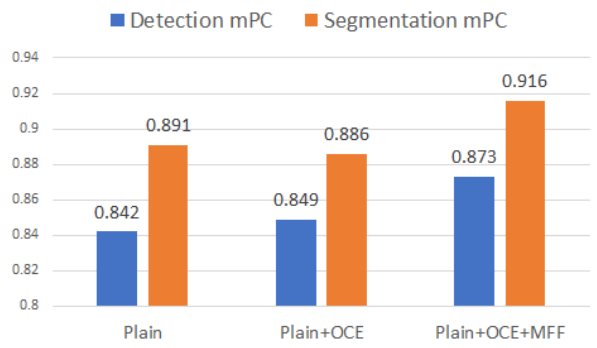


Fig. 6. Impact of Reverse Distillation OCE module and MFF module on robustness. "Plain" represents the base model.

disturbed by corruptions and closer to the original feature distribution, thus maintaining better robustness, which also explains the better performance of using features from layer3 than multi-scale features.

OCE and MFF module. As shown in Fig. 6, the OCE module improves the robustness, but the MFF module does not contribute to the robustness. Because OCE provides an information "bottleneck" so that only the critical information needed to reconstruct the teacher's network response is retained, thus filtering out non-critical low-level information about degradation. Additionally, the robustness is slightly reduced by introducing the MFF since MFF introduces low-level features that degrade more heavily.

Multi-scale distillation. The impact of the distillation of features at different scales on the robustness is shown in Fig. 5d, and it is worth noting that layer1 > layer3 > layer2 in terms of robustness, and it is necessary to analyze the mechanism. Due to its reverse distillation paradigm, the high-resolution features of the student model do not suffer from more degradation interference than the lower-level features. In contrast, the high-resolution features of the teacher model suffer more degradation interference than the low-resolution features. However, the high-resolution features have more detailed information and thus facilitate the segmentation of defects. Therefore, layer1 and layer3 have better robustness than layer2, and aggregating features of all scales can achieve higher robustness.

4.3 Robust anomaly detection method

We combine the proposed FAM with PatchCore[6] to obtain a new model, and the evaluation results are shown in Table 3, where $k = 2$ and $n = 8$. The results show that the FAM improves robustness for most corruptions types. For example, compared to PatchCore, the image-level rPC and pixel-level rPC under the JPEG compression category are improved by 5.9% and 3.9%, respectively. In addition, there is no significant improvement in robustness under the geometry category since the geometric transformation does not result in a globally consistent feature drift, which confirms that FAM mainly serves to reduce the globally consistent feature drift. Although the robustness of the model has been significantly improved, there are still limitations. First, the pixel-level AUC on the clean dataset (MVTec) decreased by 0.4%. This is because, in the absence of defects, the iteration of the FAM causes the defect features to overfit normal features. However, the model still maintains high accuracy, and the robustness has been significantly improved, so this compromise is acceptable. Second, the image-level rPC in the geometric category decreased by 1.1%. This is because the simple structure of FAM cannot capture the complex transformations caused by image rotation. Specifically, CNN lacks rotation invariance. When the image rotates, the image pixels that affect a patch feature and the combination of pixels and convolution kernel parameters change. Hence, the resulting changes in patch features are complex, exceeding the learning ability of FAM's linear structure. Increasing the complexity of FAM and introducing more contextual information for patch features may help solve this problem. Fig. 7 shows the comparisons between PatchCore and our method on some samples.

4.3.1 Effect of the number of reference features

The robustness of different reference feature numbers is compared to Fig. 5f. $k = 2$ is slightly better than $k = 1$, and the robustness does not change significantly when $k > 2$, which indicates that when the sampling ratio of 0.01, the memory bank still maintains good densities to represent the normal distribution. We speculate that a larger number of reference features will help improve the robustness when the sampling ratio is further reduced.

4.3.2 Effect of the number of iterations

As shown in Fig. 5e, when the number of iterations is small, the robustness increases subsequently, but further increasing the number of iterations leads to a decrease in robustness. A plausible reason for this is overfitting. FAM uses a simple model to fit the mapping between degenerate feature distributions and normal feature distributions. It is difficult to recover the original features, so we use the nearest neighbors from the normal memory bank instead. However, there are

Table 3. Comparison of the results of different methods. "clean" refers to the performance on MVTec. The image-level results and pixel-level results are shown for each method. The last column shows the improvement of our method. Blue indicates a decrease, green indicates an improvement of less than 5%, and red indicates a significant improvement of more than 5%.

Method	PatchCore	Ours	Improvement (%)
clean	0.990/0.980	0.991/0.976	+0.1/–0.4
Gaussian noise	0.902/0.936	0.927/0.947	+2.5/+1.1
Shot noise	0.903/0.940	0.927/0.953	+2.4/+1.3
Defocus blur	0.960/0.966	0.968/0.969	+0.8/+0.3
Motion blur	0.954/0.966	0.970/0.969	+1.6/+0.3
Brightness	0.955/0.942	0.968/0.965	+1.3/+2.3
Contrast	0.845/0.877	0.867/0.900	+2.2/+2.3
JPEG	0.853/0.914	0.912/0.953	+5.9/+3.9
Geometry	0.690/0.899	0.679/0.903	−1.1/+0.4
mPC	0.883/0.930	0.902/0.945	+1.9/+1.5
rPC	0.892/0.950	0.911/0.968	+1.9/+1.8

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

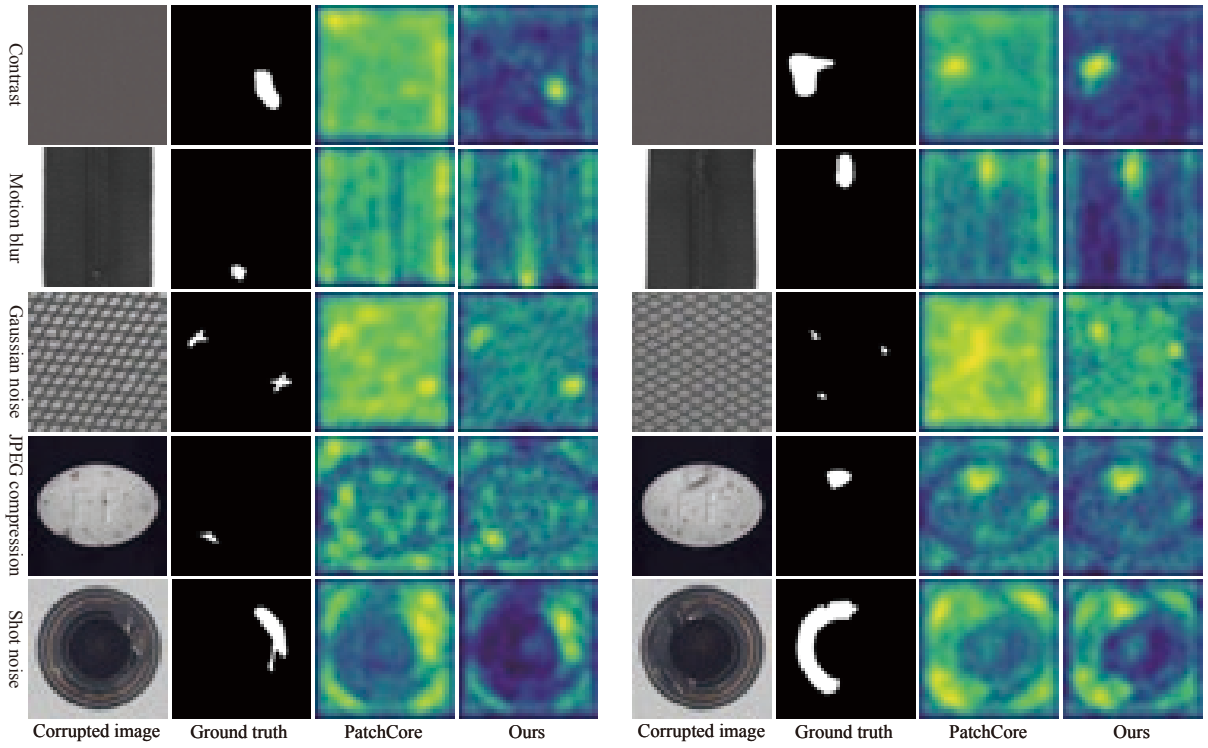


Fig. 7. Qualitative comparisons between PatchCore and our method.

differences between the nearest neighbors and the original features, so too many iterations will damage the feature representation and thus degrade the model's performance. Therefore $n = 8$ may be a good compromise.

5 Conclusions

In this paper, a comprehensive benchmark is constructed to evaluate the robustness of unsupervised anomaly detection models based on real-world corruptions in production scenarios. Thanks to this benchmark, we find many meaningful conclusions and a way to improve robustness. First, the mainstream unsupervised anomaly detection methods in five different paradigms are evaluated, and the advantages and disadvantages of each paradigm are analyzed. Then, the two approaches that perform best in terms of performance and robustness are investigated through careful ablation experiments, and the effects of different modules on the robustness of these two models are explored. Finally, a module is proposed to eliminate the feature drift caused by corruptions, and its effectiveness in improving robustness is demonstrated experimentally.

Acknowledgements

This work was supported by National Natural Science Foundation of China (62306295).

Conflict of interest

The authors declare that they have no conflict of interest.

Biographies

Pei Wang is currently working at Alibaba Cloud in Hangzhou. He received his master's degree in Control Science and Engineering from the University of Science and Technology of China in 2023. His research interests focus on computer vision.

Yang Cao is currently an Associate Professor in the Automation Department at the University of Science and Technology of China. He received his Ph.D. degree in Pattern Recognition and Intelligent System from the Northeastern University in 2004. His research interests include computer vision and multimedia processing.

References

[1] Schlegl T, Seeböck P, Waldstein S M, et al. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Information Processing in Medical Imaging. Cham: Springer, **2017**: 146–157.

[2] Akcay S, Atapour-Abarghouei A, Breckon T P. GANomaly: Semi-supervised anomaly detection via adversarial training. In: Jawahar C, Li H, Mori G, et al. editors. Computer Vision–ACCV 2018. Cham: Springer, **2018**: 622–637.

[3] Bergmann P, Löwe S, Fauser M, et al. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv: 1807.02011, **2018**.

[4] Cohen N, Hoshen Y. Sub-image anomaly detection with deep pyramid correspondences. arXiv: 2005.02357, **2020**.

[5] Defard T, Setkov A, Loesch A, et al. PaDiM: A patch distribution modeling framework for anomaly detection and localization. In: Pattern recognition. ICPR international workshops and challenges. Cham: Springer, **2021**: 475–489.

[6] Roth K, Pemula L, Zepeda J, et al. Towards total recall in industrial anomaly detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, USA: IEEE, **2022**: 14298–14308.

[7] Rudolph M, Wandt B, Rosenhahn B. Same same but DifferNet: Semi-supervised defect detection with normalizing flows. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). Waikoloa, USA: IEEE, 2021: 1906–1915.

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103

[8] Gudovskiy D, Ishizaka S, Kozuka K. CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Waikoloa, USA: IEEE, **2022**: 1819–1828.

[9] Li C L, Sohn K, Yoon J, et al. CutPaste: self-supervised learning for anomaly detection and localization. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, USA: IEEE, **2021**: 9659–9669.

[10] Bergmann P, Fauser M, Sattlegger D, et al. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, USA: IEEE, **2020**: 4182–4191.

[11] Salehi M, Sadjadi N, Baselizadeh S, et al. Multiresolution knowledge distillation for anomaly detection. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, USA: IEEE, **2021**: 14897–14907.

[12] Deng H, Li X. Anomaly detection via reverse distillation from one-class embedding. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, USA: IEEE, **2022**: 9727–9736.

[13] Bergmann P, Fauser M, Sattlegger D, et al. MVTec AD —a comprehensive real-world dataset for unsupervised anomaly detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, USA: IEEE, **2019**: 9584–9592.

[14] Hendrycks D, Dietterich T. Benchmarking neural network robustness to common corruptions and perturbations. arXiv: 1903.12261, **2019**.

[15] Michaelis C, Mitzkus B, Geirhos R, et al. Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv: 1907.07484, **2019**.

[16] Kamann C, Rother C. Benchmarking the robustness of semantic segmentation models with respect to common corruptions. *International Journal of Computer Vision,* **2021**, *129*: 462–483.

[17] Wang J, Jin S, Liu W, et al. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, USA: IEEE, **2021**: 11850–11859.

[18] Schlegl T, Seeböck P, Waldstein S M, et al. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis,* **2019**, *54*: 30–44.

[19] Zavrtanik V, Kristan M, Skočaj D. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition,* **2021**, *112*: 107706.

[20] Rippel O, Mertens P, Merhof D. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). Milan, Italy: IEEE, **2021**: 6726–6733.

[21] Golan I, El-Yaniv R. Deep anomaly detection using geometric transformations. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. New York: ACM, **2018**: 9781–9791.

[22] Hendrycks D, Mazeika M, Kadavath S, et al. Using self-supervised learning can improve model robustness and uncertainty. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM, **2019**: 15663–15674.

[23] Bergman L, Hoshen Y. Classification-based anomaly detection for general data. arXiv: 2005.02359, **2020**.

[24] Sohn K, Li C L, Yoon J, et al. Learning and evaluating representations for deep one-class classification. arXiv: 2011.02578, **2011**.

[25] Wang G, Han S, Ding E, et al. Student-teacher feature pyramid matching for anomaly detection. arXiv: 2103.04257, **2021**.

[26] Goodge A, Hooi B, Ng S K, et al. Robustness of autoencoders for anomaly detection under adversarial impact. In: IJCAI'20: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. New York: ACM, **2021**: 1244–1250.

[27] Schneider M, Aspinall D, Bastian N D. Evaluating model robustness to adversarial samples in network intrusion detection. In: 2021 IEEE International Conference on Big Data (Big Data). Orlando, USA : IEEE, **2021**: 3343–3352.

[28] Han D, Wang Z, Zhong Y, et al. Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. *IEEE Journal on Selected Areas in Communications,* **2021**, *39* (8): 2632–2647.

[29] Perales Gómez Á L, Maimó L F, Clemente F J G, et al. A methodology for evaluating the robustness of anomaly detectors to adversarial attacks in industrial scenarios. *IEEE Access,* **2022**, *10*: 124582–124594.

[30] Kloft M, Laskov P. Online anomaly detection under adversarial impact. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS). Sardinia, Italy: PMLR, **2010**: 405–412.

[31] Madani P, Vlajic N. Robustness of deep autoencoder in intrusion detection under adversarial contamination. In: Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security. New York: ACM, **2018**: 1–8.

[32] Bovenzi G, Foggia A, Santella S, et al. Data poisoning attacks against autoencoder-based anomaly detection models: A robustness analysis. In: ICC 2022 —IEEE International Conference on Communications. Seoul, Korea: IEEE, **2022**: 5427–5432.

[33] Altindis S F, Dalva Y, Pehlivan H, et al. Benchmarking the robustness of instance segmentation models. arXiv: 2109.01123, **2021**.

[34] Dooley S, Goldstein T, Dickerson J P. Robustness disparities in commercial face detection. arXiv: 2108.12508, **2021**.

DOI: 10.52396/JUSTC-2022-0165
JUSTC, **2024**, 54(1): 0103