Dual-Student Knowledge Distillation Networks for Unsupervised Anomaly Detection

Liyi Yao
University of Southern California
Los Angeles, California, USA
liyiyao@usc.edu

Abstract

Due to the data imbalance and the diversity of defects, knowledge distillation-based student-teacher networks (S-T) are introduced in unsupervised anomaly detection. This paradigm assumes that the teacher and student networks will exhibit evident discrepancy in feature representation only when processing anomalous images. However, the stability of vanilla S-T networks is inconsistent. While employing identical structures to construct teacher and student networks may result in similar representations of anomalies, using different structures can increase the likelihood of divergent performance on normal data. To address this problem, we propose a novel dual-student knowledge distillation (DSKD) architecture. Different from other S-T networks, we use two student networks a single pre-trained teacher network, where the students have the same scale but inverted structures. This framework can enhance the distillation effect to improve the consistency in recognition of normal data, and simultaneously introduce diversity for anomaly representation. To explore high-dimensional semantic information for capturing anomaly clues, we employ two strategies. First, a pyramid matching mode is used to perform knowledge distillation on multi-scale feature maps in the intermediate layers of networks. Second, an interaction is facilitated between the two student networks through a deep feature embedding module, which is inspired by real-world group discussions. In terms of classification, we obtain pixel-wise anomaly segmentation maps by measuring the discrepancy between the output feature maps of the teacher and student networks, from which an anomaly score is computed for sample-wise determination. We evaluate DSKD on three benchmark datasets and probe the effects of internal modules through ablation experiments. The results demonstrate that DSKD can achieve exceptional performance on the lightweight models like ResNet18 and effectively improve the vanilla S-T networks.

1. Introduction

In industrial manufacturing, automated **Anomaly detection** (AD) usually plays a significant role in quality control, which refers to recognize the defects and faults based on vision technology [24]. Restrictively speaking, anomaly detection determines whether there are anomalies at the image level. In terms of pixels, the goal is to judge whether a pixel falls into the anomalous area, which is also known as **Anomaly Localization** (AL) [27]. Fig. 1 shows pixel-wise examples of AL. To remove ambiguity, we explicitly point out the image-level anomaly detection and the pixel-level anomaly localization in the Sec. 4. And for the sake of simplicity, in other sections, we refer to AD and AL collectively as "anomaly detection".

In general, there are two main challenges for AD. First, defects are usually rare in the normal manufacturing process. It is difficult to acquire as many anomalous samples as anomaly-free samples, which results in data imbalance. And since most current vision recognition algorithms are data-drive, this drawback can damage their performance [27]. On the other hand, the anomalous patterns are diverse and datasets for training can not contain all types of anomalies. Defects probably exist in the texture of the surface or in the inner structure, and their forms can be stains, wear, scratches, the absence of some parts, or unknown. Therefore, without anomaly reference, supervised learning can hardly achieve accurate detection, while unsupervised learning methods can be a better choice.

To tackle these problems, some traditional vision technologies are applied in AD for feature extraction and classification, like one-class support vector machine (ocsvm) [23]. In terms of pattern recognition, deep learning-based approaches show great success in AD. And these approaches can be broadly classified into two categories, **reconstruction-based** models [1, 5, 19, 18, 22, 21] and **deep feature embedding-based** models [6, 7, 17, 8, 3, 28]. The former learn how to reconstruct data in defect-free distribution, and capture the discrepancy between the source

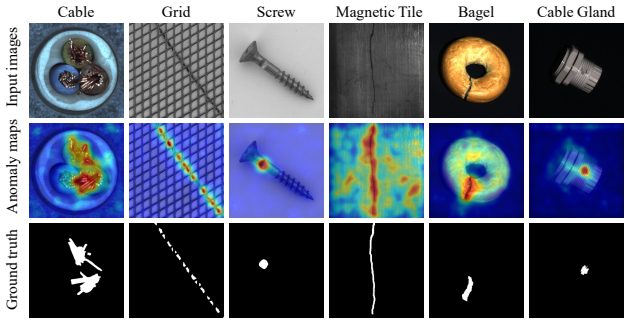

Figure 1. Examples of anomaly localization or segmentation. From the first to the third line, anomaly samples, anomaly maps generated by our proposed model, and ground truth are shown, respectively.

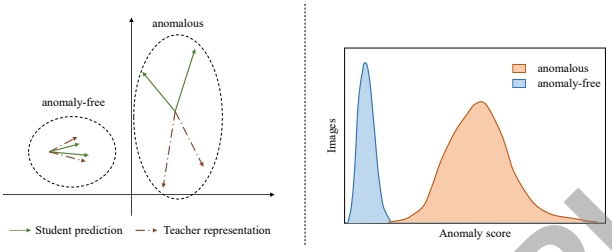

Figure 2. Principles of knowledge distillation in anomaly detection. In the left figure, the student and the teacher have similar representations of anomaly-free patterns but differ significantly in anomalies. Given this, we can calculate the anomaly scores in the right figure.

and reconstructed data. In concrete, if input images are non-anomalous, the trained model can generate images in similar data distribution. Otherwise, the generated images show evident difference from the inputs [22]. With respect to the feature embedding-based methods, they leverage the extracted feature information to estimate the anomalous patterns, usually in the latent space. For unsupervised learning, the feature extraction process is usually based on pre-trained models [27].

Currently, many studies use the **knowledge distillation** (KD) paradigm [12] for anomaly detection, which falls into the feature embedding-based category [3, 20, 28, 9, 8]. KD is first proposed to transfer knowledge from a pre-trained large, complex model (teacher) to smaller, simpler models (student) for model compression [12]. A random initialized model is like a naive child, *i.e.* student, who needs to learn and understand the world, and a pre-trained model is an expert, *i.e.* teacher, with rich knowledge. The new model can learn from the output of the pre-trained model rather than ground truth labels. Then, parameter-wise information and knowledge stored in the teacher network can be transferred to the student, which is called **distillation**. Since both the teacher and the student are involved in this process, it is also known as **student-teacher (S-T)** networks.

In anomaly detection, the teacher network is pre-trained on some large-scale datasets, like ImageNet, and its parameters are frozen in the training process. In the meanwhile, the student is only fed with the anomaly-free data and the distilled knowledge, *i.e.* feature maps, transferred from the teacher. In other word, features extracted by the teacher are embedded into the student to help the student learn how to represent features of normal data. Therefore, the teacher and the trained student can represent anomaly-free images in the same way, but show discrepancy in the representation of anomalies, as shown in Fig. 2. By analogy, if a student only learn math from a well-educated teacher, they are likely to give different answers to some geological questions. Then we can measure the distance between the output feature maps of the teacher and the student to estimate the anomalous patterns by calculating the anomaly score, where a higher score means a higher probability of anomalies. If the anomaly score is higher than the threshold, it can be considered as anomaly as exhibited in (1). In our work, we perform normalization on the anomaly score and set the threshold to 0.5.

$$A(score) = \begin{cases} True, & score \geqslant threshold, \\ False, & score < threshold. \end{cases} \quad (1)$$

To exploit the such representation discrepancy between the teacher and the student, some studies *et al.* [20] use asymmetric S-T networks, where the teacher network usually has a larger and more complex backbone like the classic knowledge distillation paradigm. However, this architecture can also expose the representation discrepancy in anomaly-free data sometimes, which damages the detection accuracy. In contrast, some other works [28] use a symmetric structure, where both teacher and student are based on the same backbones. The limitation of such architecture is that the teacher and the student probably perform similarly on anomalous data, which also makes negative effects as well. An ideal architecture should highlight differences in anomaly representation while maintaining representative similarity for anomaly-free data.

For this reason, we proposed a **dual-student knowledge distillation paradigm (DSKD)** for AD, as shown in Fig. 3. In DSKD, we add an extra student network and there are totally three players involved, a teacher T and two students S_e and S_d. T and S_e are based on ResNet18 [11], while the backbone of S_d is completely reversed from S_e. In this way, the symmetric combination of T and S_e helps produce similar results with normal samples and the inverted structure can strengthen the representation discrepancy on anomalous samples. We regard the methods of Salehi *et al.* [20] and Wang *et al.* [28] as baselines for the following experimental study.

In DSKD, T has been pre-trained on ImageNet-1K, and

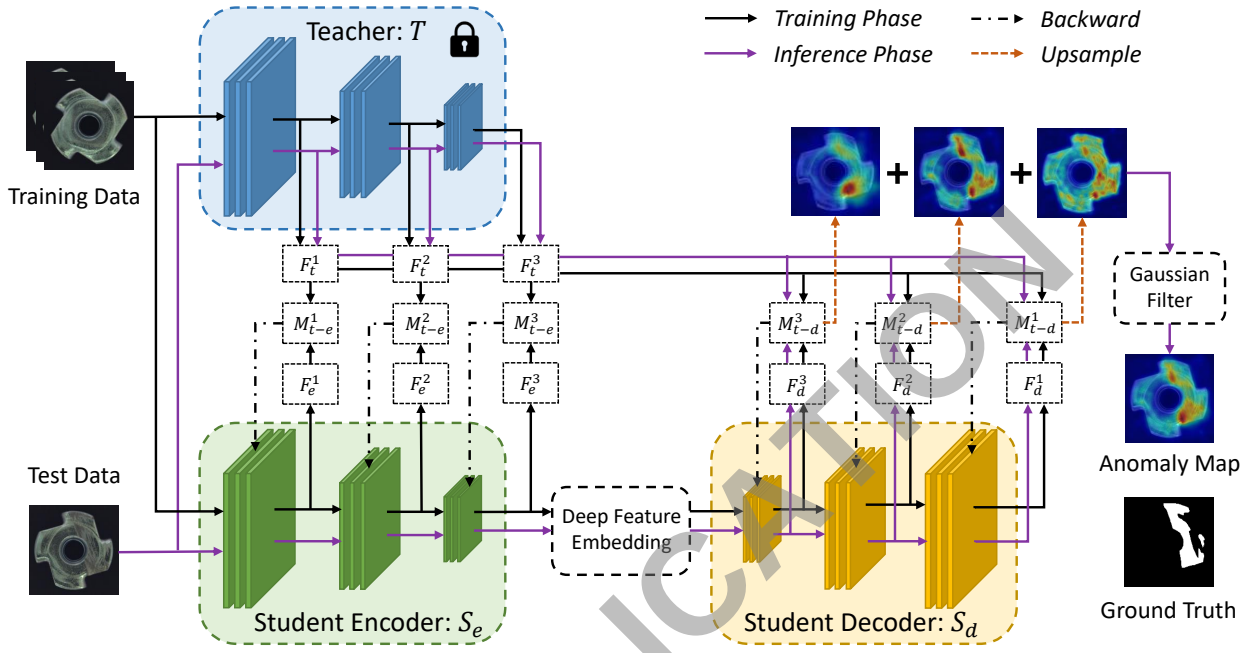

Figure 3. The framework of the proposed dual-student knowledge distillation (DSKD) model.

its parameters are frozen. The first student S_e works as an encoder to extract features from the input. And the second student S_d is a decoder, whose function is to decode information from the high-dimensional features that extracted by S_e. The distillation processes between two students and the teacher is performed separately and independently. Since feature maps of different layers can focus on different types of features in hierarchical networks [20], DSKD fuses multi-scale intermediate feature maps to explore the high-dimensional semantic information to enhance distillation effects rather than simply measuring the final outputs. The multi-scale feature fusion exists not only in the distillation between the teacher and student networks, but also in the collaboration between student networks. S_e and S_d are connected by a bottleneck block, which fuses the intermediate feature maps of S_e through several convolutional blocks and embeds them to S_d in format of low-dimensional vectors. This is inspired by real-world education scenarios, where collaboration among students can aid in a better understanding of the knowledge imparted by the teacher. Regarding the reasoning phase, *i.e.* anomaly inference, data processed by S_e and S_d sequentially is matched with the outputs of T to obtain anomaly maps. The technical details are discussed in Sec. 3. And in Sec. 4, we discuss the experiments and analyze the results.

Our contributions can be summarized as follows:

1. We propose a novel dual-student knowledge distillation paradigm consisting of a teacher network and two student networks for unsupervised anomaly detection, where the networks have the same scale but the two students have totally inverted structures This method can effectively improve the distillation effects by enhancing the diversity of anomalous feature representations and reduce the discrepancy on anomaly-free data.

2. We design the multi-scale feature fusion block to explore high-dimensional semantic information, which matches intermediate feature maps between the teacher and student networks based on a feature pyramid instead of only using the final outputs. Intermediate anomaly maps are upsampled to the same size and summed together for anomaly inference.

3. The two student networks are connected by a bottleneck design, where the deep features of the first student are embedded. This deep feature embedding module uses convolutional blocks to downsample multi-scale feature maps and do channel-wise feature fusion. The collaboration of two student networks can also benefit the detection results.

4. We performed experiments on three benchmark datasets, MVTec AD [2], MVTec 3D-AD [4], and Magnetic Tile Defect [14]. Our proposed method outperforms baselines and achieve good results. And the ablation experiments prove the effectiveness of the dual-student architecture and its interior modules.

Algorithm 1 DSKD Training Procedure

Input: Training dataset $\mathcal{I}^t = \{I_1^t, I_2^t, ..., I_m^t\}$, parameters θ_t of T, epoch number n;
Output: Parameters θ_e of S_e, parameter θ_d of S_d;

randomly initialize parameters of θ_e and θ_d;
$T \leftarrow T$.load_weights(θ_t);
for $i \leftarrow 1$ to n **do**
  for $j \leftarrow 1$ to m **do**
    $S_e \leftarrow S_e$.load_weights(θ_e);
    $S_d \leftarrow S_d$.load_weights(θ_d);
    $\hat{F}_t \leftarrow \ell_2$_normalization($T(I_j^t)$);
    $\hat{F}_e \leftarrow \ell_2$_normalization($S_e(I_j^t)$);
    $\hat{F}_d \leftarrow \ell_2$_normalization($S_d(F_{emb})$);
    for $k \leftarrow 1$ to length(T) **do**
      Obtain anomaly map M_{t-e}^k between T and S_e based on Eq. (6);
      Obtain anomaly map M_{t-d}^k between T and S_d based on Eq. (6);
    end for
    $\ell_e \leftarrow$ mean(M_{t-e});
    $\ell_d \leftarrow$ mean(M_{t-d});
    Update parameters θ_e;
    Update parameters θ_d;
  end for
end for

2. Related Works

Anomaly detection is first considered a kind of one class classification (OCC) task where some traditional machine learning and vision methods are employed. Due to the diversity of anomalous features, data-driven deep learning technologies become more popular, which can be categorized into reconstructed-based methods and deep feature embedding-based methods.

2.1. Classical Methods

Classical AD methods generally process vectors in a high-dimensional space. The one-class support vector machine (OCSVM) [23] maximized the distance between the hyperplane and the origin in the feature space and estimated the probability density area for one-class classification. Support vector data description (SVDD) [26] uses a hypersphere instead of a hyperplane, which yields better results in defects detection. Deep-SVDD [13] maps data in a smaller hypersphere by deep networks. And Patch-SVDD [29] implements one-class detection at the pixel level. However, these classical methods usually have weak performance in generalization and are inefficient on large-scale datasets.

2.2. Reconstruction-based Methods

These methods are inspired by the idea that models trained with anomaly-free data can only reconstruct both anomalous and normal data in anomaly-free distribution. Therefore, the divergence between the input and output data distributions can be used to identify anomalous patterns [24]. Vanilla auto-encoder (Vanilla AE) [1] is applied in anomaly segmentation for brain images. And Bergmann *et al.* [5] use the structural similarity (SSIM) metric as the optimization objective function.

Additionally, GAN-based models also have good abilities in feature reconstruction. Schlegl *et al.* [22] propose Anomaly GAN (AnoGAN), which uses the Wasserstein distance to measure the discrepancy between normal and anomalous data. And fast AnoGAN (f-AnoGAN) [21] optimizes the architecture by using a trained decoder as the generator and requires less computation resources.

Unlike other generative models, normalizing flows (NF) [16] has advantages in density estimation. Rudolph *et al.* [18] propose an NF-based model, DifferNet, for anomaly detection. To localize anomalous regions, fully convolutional cross-scale normalizing (CS-Flow) [19] retains the spatial arrangement mode in the latent space and can process multi-scale feature maps.

2.3. Deep Feature Embedding-based Methods

Since errors in reconstruction can result in mistakes in detection [24], some studies identify anomalies with extracted features mapped to low-dimension representation, known as **embedding** [10]. And the representative discrepancy can be measured in the embedding space rather than reconstruction. Considering that pre-trained models retain high performance of capturing deep semantic information in downstream tasks, feature extractors are generally pre-trained on large-scale datasets like ImageNet. Cohen *et al.* [6] make improvements on deep pre-trained k-nearest neighbor methods via a feature pyramid. Defard *et al.* [7] propose a patch distribution modeling (PaDiM) framework that implements patch embedding with a pre-trained CNN model and multivariate Gaussian distributions. Inspired by this patch-level feature processing method, Roth *et al.* [17] present the PatchCore model, which deposits extracted features in a compact memory bank and uses the nearest neighbor search algorithm to simplify the computation in the inference period.

In terms of knowledge distillation for AD, the teacher is supposed to impart only knowledge about anomaly-free data to the student so that they can perform differently on anomalous data. Bergmann *et al.* [28] introduce KD in anomaly detection, where the teacher network is pre-trained with its parameters frozen, and the student only learns the non-anomalous distribution. S-T networks can directly capture defective information at the pixel level by comparing

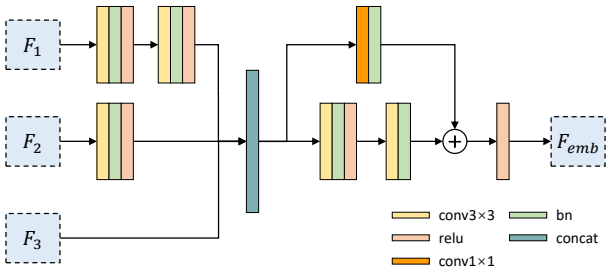

Figure 4. Deep feature embedding process. Feature maps from different layers are resized to the same scale and then are downsampled by convolutional modules. The embedding carries rich semantic information from different intermediate layers.

the output feature maps of the teacher and the student networks. To explore high-dimensional information of deep layers, Salehi *et al.* [20] propose a multi-resolution KD (MKD) model, which leverages the features in intermediate layers. And MKD employs an asymmetric architecture On the contrary, Wang *et al.* [28] fused the outputs of intermediate layers into a feature pyramid within symmetric S-T networks. Furthermore, some studies use feature reconstruction strategies to improve the quality of distillation. Dehaene *et al.* proposed a feature-augmented Variational Auto-Encoder (FAVAE) [8] framework consisting of a feature extractor and a VAE module. The extracted features can be embedded in the decoder as a S-T network. Based on auto-encoder, a reverse distillation [9] approach uses an encoder as the teacher and a decoder as the student to leverage the teacher's embedding information. In our dual-student S-T networks, there are three players employed instead of only one pair of teacher and student networks. Two inverted student networks learn from the same teacher in the encoding and decoding aspects, respectively. In this way, it can amplify the discrepancy between the representation of anomalous features of the teacher and the student and alleviate that of anomaly-free features, as discussed in Sec. 1.

3. Proposed Method

In this section, we present the our model in detail and discuss its technical implementation. Fig. 3 exhibits the overview of our proposed framework. For better expressions, the training dataset and test dataset are respectively marked as $\mathcal{I}^t = \{I_1^t, I_2^t, ..., I_m^t\}$ and $\mathcal{I}^a = \{I_1^a, I_2^a, ..., I_m^a\}$. $\mathcal{I}^t$ only consists of anomaly-free samples, while $\mathcal{I}^a$ has anomaly-free and anomalous samples. And the mth image is $I_m \in \mathbb{R}^{w \times h \times c}$, where w and h represent the width and height of the image, and c denote the number of channels.

3.1. Dual-Student Knowledge Distillation

The classical KD method employs one pair of S-T networks for knowledge transfer and model size compress. Given a pre-trained teacher network T and a target student network S, the objective can be shown as:

$$\arg\min_{\theta} \ell_{KD} = \mathbf{D}(T_{\theta^*}(x), S_\theta(x)) + \lambda \mathbf{D}(y, S_\theta(x)), \quad (2)$$

where θ^* and θ denote the parameters of T and S, y is the ground truth, and $\mathbf{D}$ represents a certain distance metric. Kullback-Leible (KL) divergence is used to measure the distance, which is proven to be equivalent to the Euclidean distance, *i.e.* ℓ_2-distance, in the optimization process according to the mild assumption [12].

Unlike other S-T network-based approaches, we add a new student to enhance inconsistencies in the feature representation of out-of-distribution data. As shown in Fig. 3, DSKD has three parts, teacher T, students S_e and S_d. Their feature maps can be represented as $F_t^k(I_m), F_e^k(I_m), F_d^k(I_m) \in \mathbb{R}^{w_k \times h_k \times c_k}$, where t, e, d, respectively, denote T, S_e, S_d, and k is the index of the intermediate layer. We hypothesize that the student S_e can progressively acquire knowledge related to anomaly-free features by learning from T, in turn, help S_d enhance its understanding of the same knowledge transferred from T. Because in a hierarchical network different intermediate layers probably focus on different semantic information [20], we integrate the intermediate feature maps like a feature pyramid rather than only using the last layer's outputs. Specifically, in our work, we select the first three blocks in ResNet18, *i.e.* conv2_x, conv3_x, conv4_x [11], so the total number of intermediate layers $K = 3$. The vectors on each feature map are localized by the indices i and j, and noted as $F_t^k(I_m)_{ij}, F_e^k(I_m)_{ij}, F_d^k(I_m)_{ij} \in \mathbb{R}^{c_k}$. These vectors are normalized by ℓ_2-distance as follows:

$$\hat{F}_t^k(I_m)_{ij} = \frac{F_t^k(I_m)_{ij}}{||F_t^k(I_m)_{ij}||_{\ell_2}^2}, \quad \hat{F}_e^k(I_m)_{ij} = \frac{F_e^k(I_m)_{ij}}{||F_e^k(I_m)_{ij}||_{\ell_2}^2},$$
$$\hat{F}_d^k(I_m)_{ij} = \frac{F_d^k(I_m)_{ij}}{||F_d^k(I_m)_{ij}||_{\ell_2}^2}. \quad (3)$$

T is pre-trained on a large-scale dataset, ImageNet-1K, which is regarded as prior knowledge. In the training phase, T's parameters are frozen and T is used to extract features from data, providing reference for students. S_e has the same network as T, learning from T in the distillation process. S_d is reversed from S_e with a completely transposed backbone, where downsampling operations are replaced by upsampling operations. The two students networks are connected by a deep feature embedding block, which is discussed in Sec. 3.2. S_d decodes the embedded data flow and

upsamples the low-dimensional vectors to the same size as the input images. Regarding the knowledge distillation, we measure the pixel-wise distance of each intermediate feature map between the teacher and student networks and obtain the multi-scale anomaly maps. To match the resolutions, the distillation is performed in intermediate layers with the same sizes, as exhibited in Fig. 3. T transfers the knowledge to S_e and S_d separately, and S_d also receives the knowledge from S_e. Algo. 1 describes the training procedure.

To evaluate the discrepancy between the teacher and the students, DSKD uses a combined loss function. The first part is ℓ_2-distance, minimizing the Euclidean distance. $t-e$ and $t-d$ separately denote the distillation between T and S_e, and between T and S_d, as shown in Eq. (4).

$$\begin{aligned}\ell_2^{t-e}(I_m)_{ij} &= \frac{1}{2}||\hat{F}_t^k(I_m)_{ij} - \hat{F}_e^k(I_m)_{ij}||_{\ell_2}^2, \\ \ell_2^{t-d}(I_m)_{ij} &= \frac{1}{2}||\hat{F}_t^k(I_m)_{ij} - \hat{F}_d^k(I_m)_{ij}||_{\ell_2}^2,\end{aligned} \tag{4}$$

The other part is based on cosine similarity, which measures the directional distance between two vectors. To minimize the loss function for optimization, this part ℓ_{cos} is written as follows:

$$\begin{aligned}\ell_{cos}^{t-e}(I_m)_{ij} &= 1 - \frac{(\hat{F}_t^k(I_m)_{ij})^T \cdot \hat{F}_e^k(I_m)_{ij}}{||\hat{F}_t^k(I_m)_{ij}||\ ||\hat{F}_e^k(I_m)_{ij}||}, \\ \ell_{cos}^{t-d}(I_m)_{ij} &= 1 - \frac{(\hat{F}_t^k(I_m)_{ij})^T \cdot \hat{F}_d^k(I_m)_{ij}}{||\hat{F}_t^k(I_m)_{ij}||\ ||\hat{F}_d^k(I_m)_{ij}||}.\end{aligned} \tag{5}$$

The value of (i, j) pixel on the anomaly map of the kth layer can be obtained by Eq. (6), where λ is a coefficient. To integrate anomaly maps in different sizes, we sum up pixels of all anomaly maps, and take the mean values as the total loss as described in Eq. 7 where ℓ_e denotes the loss of S_e, and ℓ_d denotes that of S_d.

$$\begin{aligned}M_{t-e}^k(I_m)_{ij} &= \lambda\ell_2^{t-e}(I_m)_{ij} + \ell_{cos}^{t-e}(I_m)_{ij}, \\ M_{t-d}^k(I_m)_{ij} &= \lambda\ell_2^{t-d}(I_m)_{ij} + \ell_{cos}^{t-d}(I_m)_{ij},\end{aligned} \tag{6}$$

$$\begin{aligned}\ell_e(I_m) &= \frac{1}{K}\sum_{k=1}^{K}[\frac{1}{h_k w_k}\sum_{i=1}^{h_k}\sum_{j=1}^{w_k} M_{t-e}^k(I_m)_{ij}], \\ \ell_d(I_m) &= \frac{1}{K}\sum_{k=1}^{K}[\frac{1}{h_k w_k}\sum_{i=1}^{h_k}\sum_{j=1}^{w_k} M_{t-d}^k(I_m)_{ij}].\end{aligned} \tag{7}$$

Algorithm 2 DSKD Anomaly Inference

Input: Test image I^a, parameters θ_t of T, parameters θ_e of S_e, parameters θ_d of S_d;
Output: Anomaly maps $\widetilde{M}$, detection result A

$M \leftarrow$ zeros(I^a.height, I^a.width);
$T \leftarrow T$.load_weights(θ_t);
$S_e \leftarrow S_e$.load_weights(θ_e);
$S_d \leftarrow S_d$.load_weights(θ_d);
$\hat{F}_t \leftarrow \ell_2$_normalization($T(I_a)$);
$\hat{F}_e \leftarrow \ell_2$_normalization($S_e(I_a)$);
$F_{emb} \leftarrow$ feature_embedding($\hat{F}_e$);
$\hat{F}_d \leftarrow \ell_2$_Normalization($S_d(F_{emb})$);
for $k \leftarrow 1$ to length(T) **do**
    Obtain anomaly map M_{t-d}^k between T and S_d based on Eq. (6);
    $M \leftarrow M +$ upsample(M_{t-d}^k);
end for
Denoise following Eq. (9): $\widetilde{M} = G_\sigma(M,\ \sigma = 4.0)$;
Obtain the anomaly score: $score = \max(\widetilde{M})$;
$threshold \leftarrow 0.5$;
if $score \geqslant threshold$ **then**
    $A \leftarrow$ True;
else
    $A \leftarrow$ False
end if

3.2. Deep Feature Embedding

DSKD uses the deep feature embedding (DFE) module as a bottleneck to connect of S_e and S_d and activate their collaboration. S_e learns to encode anomaly-free data from T and transfers one-way data flow to S_d. Due to the resolution mismatch, source images are not available to S_d. So the input of S_d is supposed to preserve feature information of anomaly-free data as much as possible. Referring to [9], we conduct feature fusion on intermediate outputs of S_e and embed the features into a low-dimensional space, *i.e.* embeddings. The DFE block has two benefits. First, as described above, DFE can carry rich semantic information which can help S_d comprehend the knowledge of anomaly-free distributions in the training phase. Second, transferring embeddings instead of original feature maps can enhance the representative diversity on anomaly data in the inference phase.

Fig. 4 describes the DFE process in detail. Based on ResNet18, there are three intermediate feature maps involved in the fusion. F_1, F_2, and F_3 represent the feature maps of the three layers with increasing depth, respectively. F_1 and F_2 are downsampled twice and once by a convolutional blocks with a kernel size of 3×3 and a stride of 2. After be resizing to the same size, the three feature maps are concatenated together. And then the embedded feature

map F_{emb} can be obtained through a residual block [11], in which a convolutional block with a 1×1 kernel is used in the shortcut connection to avoid degradation problem.

3.3. Anomaly Inference

In the anomaly inference phase, an input image is judged as anomaly or not according to the anomaly score. Because anomalous patterns at the pixel level mean that defects and faults exist in this image, the detection results at the image level can be obtained from the localization results.

In DSKD, we evaluate the representation discrepancy between T and S_d to score each pixel for anomaly localization, as shown in Fig. 3. Anomaly maps of all deep layers are upsampled to the same size as inputs firstly and then summed up to a map as follows:

$$M(I_m) = \sum_{k=1}^{K} Upsample(M_{t-d}^{k}(I_m)), \tag{8}$$

where K is the total number of intermediate layers involved. For denoising, $M(I_m)$ is processed by a Gaussian filter with parameter $\sigma = 4$ as follows:

$$\widetilde{M}(I_m) = G_\sigma(M(I_m)),\ \sigma = 4. \tag{9}$$

The value of point (i, j) in the final anomaly map $\widetilde{M}(I_m)$ corresponds to the anomaly score on a pixel in an image. And we take the maximum value in $\widetilde{M}(I_m)$ as the anomaly score for the detection result, as shown in Eq. (10). And the whole process of anomaly inference is exhibited in Algo. 2

$$score = max(\widetilde{M}(I_m)). \tag{10}$$

4. Experimental results

In this section, we discuss the experiments and the results. The proposed DSKD model is compared with other methods. In the ablation experiment, we analyze the function of each module in DSKD.

4.1. Experiment Settings

4.1.1 Datasets

We performed experiments on three datasets, MVTec AD [2], MVTec 3D-AD [4], Magnetic Tile Defects [14]. **MVTec AD** is a benchmark dataset for unsupervised anomaly detection. It consists of over 5000 high-resolution industrial images, which are divided into 15 categories. Each category has about 240 anomaly-free data for training and about 100 images for test. And the pixel-level ground truth data can be used to evaluate the results of anomaly segmentation. And **MVTec 3D-AD** is a comprehensive 3D dataset for AD and AL, which contains more than 4,000 high-resolution industrial images in 10 categories. Each 3D image has two formats, XYZ for 3D data and RGB for 2D data. In this paper, we only use the RGB images to test our methods. **MT Defects** is a dataset for surface defect detection of magnetic tiles, which contains five categories of defects.

4.1.2 Implement and Environment Details

All of the networks, including T, S_e and S_d are based on ResNet18. The input images are firstly resized, 256×256 for MVTec AD and MVTec 3D-AD, and 128×128 for MT Defects. The resized images are then normalized by ℓ_2 function. In the training phase, we use the adaptive moment estimation algorithm (Adam) as the optimizer, with $\beta = (0.5, 0.999)$. The learning rate is 0.001, and the model is trained in 200 epochs. As for the loss function, λ is set to 0.1. The experiments on three datasets are compared with other approaches. The ablation experiment is used to prove the effectiveness of each module in the dual-student architecture.

4.1.3 Evaluation Criteria

We employ three commonly used evaluation criteria in this paper, including the area under the receiver operating characteristic curve (AUROC) at the image level and the pixel level, and the normalized area under the per-region overlap curve (PRO). AUROC indicates the probability that the predicted value of a positive sample is larger than that of a negative sample in the random selection. And PRO evaluates the segmentation results by measuring regions of each size with the same weight and the threshold of false positive should be below 0.3. AUROC is possibly impacted by the size of anomaly regions, while PRO is not. The image-wise AUROC evaluates the performance in image classification (*i.e.* AD), and pixel-wise AUROC and PRO are used for anomaly segmentation (*i.e.* AL), where higher scores mean better performance. As for model complexity, we compare DSKD with other methods in inference time (inf. time), floating point operations (FLOPs), the scale of parameters, and memory sizes.

4.2. Results and Discussion

4.2.1 MVTec AD

We performed experiments on the three datasets and the results are compared with other approaches. Table 1 shows the image-level detection results in MVTec AD. Our method performs best in many categories, including carpet, grid, leather, wood, bottle, cable, and hazelnut. And our method produces the highest mean score at 98.5%, exceeding our baseline methods, STPM [28] and MKD [20]

Table 1. Image-level AUROC(100%) Results on MVTec AD

Method/Category	Patch SVDD[29]	AE-SSIM[5]	AnoGAN[22]	Spade[6]	Padim[7]	CutPaste[15]	MB-PFM[27]	DSN[25]	MKD[20]	STPM[28]	FAVAE[8]	Ours
Carpet	92.9	67.0	49.0	92.8	99.8	93.9	**100**	96.8	79.3	98.9	67.1	**100**
Grid	94.6	69.0	51.0	47.3	96.7	**100**	98.0	95.6	78.0	**100**	97.0	**100**
Leather	90.9	46.0	52.0	95.4	**100**	**100**	**100**	91.8	95.1	**100**	67.5	**100**
Tile	97.8	52.0	51.0	96.5	98.1	94.6	**99.6**	96.4	91.6	95.5	80.5	98.4
Wood	96.5	83.0	68.0	95.8	99.2	99.1	99.5	98.3	94.3	99.2	94.8	**99.6**
Bottle	98.6	88.0	69.0	97.2	99.9	98.2	**100**	**100**	99.4	**100**	99.9	**100**
Cable	90.3	61.0	53.0	84.8	92.7	81.2	98.8	98.3	89.2	92.3	95.0	**99.2**
Capsule	76.7	61.0	58.0	91.0	91.3	**98.2**	94.5	91.6	80.5	88.0	80.4	93.1
Hazelnut	92.0	54.0	50.0	88.1	92.0	98.3	**100**	99.4	98.4	**100**	99.3	**100**
Metal nut	94.0	54.0	50.0	71.0	98.7	99.9	100	97.7	73.6	**100**	85.2	99.7
Pill	86.1	60.0	68.0	80.1	93.3	94.9	**96.5**	89.5	82.7	93.8	82.1	96.2
Screw	81.3	51.0	35.0	66.7	85.8	88.7	91.8	**98.1**	83.3	88.2	83.7	97.1
Toothbrush	**100**	74.0	57.0	88.9	96.1	99.4	88.6	**100**	92.2	87.8	95.8	98.9
Transistor	91.5	52.0	67.0	90.3	97.4	96.1	**97.8**	91.3	85.6	93.7	97.2	97.5
Zipper	97.9	80.0	59.0	96.6	90.3	**99.9**	97.4	96.1	93.2	93.6	93.2	97.4
Mean	92.1	63.0	55.0	85.5	95.5	96.1	97.5	96.1	87.8	95.4	87.9	**98.5**

Table 2. Pixel-level AUROC(100%)/PRO(100%) Results in MVTec AD

Method/Category	Patch SVDD[29]	AE-SSIM[5]	AnoGAN[22]	Spade[6]	Padim[7]	CutPaste[15]	MB-PFM[27]	DSN[25]	MKD[20]	STPM[28]	FAVAE[8]	Ours
Carpet	92.6/–	87.0/64.7	54.0/20.4	97.5/94.7	99.1/96.2	98.3/–	**99.2**/96.9	99.1/–	95.6/–	98.8/95.8	96.0/–	**99.2/97.7**
Grid	96.2/–	94.0/84.9	58.0/22.6	93.7/86.7	97.3/94.6	97.5/–	98.8/96.0	98.1/–	91.8/–	99.0/96.6	99.3/–	**99.4/97.7**
Leather	97.4/–	78.0/56.1	64.0/37.8	97.6/97.2	99.2/97.8	**99.5**/–	99.4/98.8	99.2/–	98.1/–	99.3/98.0	98.1/–	99.4/**99.1**
Tile	91.4/–	59.0/17.5	50.0/17.7	87.4/75.9	94.1/86.0	90.5/–	96.2/88.7	90.9/–	82.8/–	**97.4/92.1**	71.4/–	94.5/88.2
Wood	90.8/–	73.0/60.5	62.0/38.6	88.5/87.4	94.9/91.1	95.5/–	95.6/92.6	94.1/–	84.8/–	**97.2/93.6**	89.9/–	94.3/91.6
Bottle	98.1/–	93.0/83.4	86.0/62.0	98.4/95.5	98.3/94.8	97.6/–	98.4/95.4	96.4/–	96.3/–	**98.8**/95.1	96.3/–	98.7/**96.4**
Cable	96.8/–	82.0/47.8	78.0/38.3	97.2/90.9	96.7/88.8	90.0/–	96.7/**94.2**	97.1/–	82.4/–	95.5/87.7	96.9/–	**97.5**/92.5
Capsule	95.8/–	94.0/86.0	84.0/30.6	**99.0**/93.7	98.5/93.5	97.4/–	98.3/91.7	98.3/–	95.9/–	98.3/92.2	97.6/–	98.7/**94.4**
Hazelnut	97.5/–	97.0/91.6	87.0/69.8	**99.1**/95.4	98.2/92.6	97.3/–	**99.1/96.7**	98.8/–	94.6/–	98.5/94.3	98.7/–	98.8/94.6
Metal nut	98.0/–	89.0/60.3	76.0/32.0	98.1/94.4	97.2/85.6	93.1/–	97.2/**94.6**	**98.3**/–	86.4/–	97.6/94.5	96.6/–	96.7/89.5
Pill	95.1/–	91.0/83.0	87.0/77.6	96.5/94.6	95.7/92.7	95.7/–	97.2/96.1	96.7/–	89.6/–	**97.8/96.5**	95.3/–	97.1/94.7
Screw	95.7/–	92.0/88.7	80.0/46.6	98.9/**96.0**	98.5/94.4	96.7/–	98.7/93.4	99.3/–	96.0/–	98.3/93.0	99.3/–	**99.4**/94.2
Toothbrush	98.1/–	96.0/78.4	90.0/74.9	97.9/93.5	98.8/93.1	98.1/–	98.6/90.7	98.6/–	96.1/–	98.9/92.2	98.7/–	**99.1/94.5**
Transistor	97.0/–	90.0/72.5	80.0/54.9	94.1/**87.4**	97.5/84.5	93.0/–	87.8/74.9	87.0/–	76.5/–	82.5/69.5	**98.4**/–	91.5/80.2
Zipper	95.1/–	88.0/66.5	78.0/46.7	96.5/92.6	98.5/**95.9**	**99.3**/–	98.2/94.8	98.2/–	93.9/–	98.5/95.2	96.8/–	97.5/93.5
Mean	95.7/–	87.0/69.4	74.3/44.3	96.5/91.7	**97.5**/92.1	96.0/–	97.3/93.0	96.7/–	90.7/–	97.0/92.1	95.3/–	**97.6/93.5**

by 2.9% and 10.5%, respectively. Regarding anomaly localization, results of the pixel-level AUROC and PRO are shown in Table 2. The proposed DSKD is superior to other methods, with 97.6% AUROC and 93.5% PRO scores. In addition, Padim [7] and MB-PFM [27] are also competitive, whose AUROC scores are 97.5% and 97.3%, and PRO scores are 92.1% and 93.0%, respectively.

The qualitative localization results are visualized in Fig.5. DSKD can accurately recognize defective patterns and achieve segmentation results that are very close to the ground truth. However, the limitation is that our method identifies surrounding anomaly-free pixels as anomalies, especially for the toothbrush, tile and wood.

4.2.2 MVTec 3D-AD

As for MVTec 3D-AD dataset, we conduct experiments on RGB images. The image-level detection results are shown in Table 3. Our method outperforms others a lot. But the detection AUROC score is only 86.4%, and DSKD cannot precisely recognize abnormal and normal data in some categories, especially cookies and potato. However, DSKD performs excellently in anomaly localization, as shown in Table 4, where the pixel-level AUROC and PRO are 98.6% and 95.2%. One possible reason is that the background of an RGB image is usually a uniform color, such as black, which occupies a large percentage of the image and is easy to recognize. As a result, the monolithic background can raise the localization score of the entire image. And Fig. 6 presents the visualized results of pixel-wise localization. Defective patterns can be accurately recognized, but some neighbor defect-free pixels are sometimes classified as anomalies, *e.g.* the carrot and bagel. And in the cookie sample, the central anomaly is ignored.

Table 3. Image-level AUROC(100%) Results in MVTec 3D-AD (RGB)

Method/Category	Padim[7]	PatchCore[17]	CS-flow[19]	STPM[28]	Ours
Bagel	**97.5**	87.6	94.1	93.0	96.1
Cable Gland	77.5	88.0	93.0	84.7	**93.0**
Carrot	69.8	79.1	82.7	89.0	**90.0**
Cookie	58.2	68.2	**79.5**	57.5	65.4
Dowel	95.9	91.2	**99.0**	94.7	97.6
Foam	66.3	70.1	**88.6**	76.6	84.8
Peach	85.8	69.5	73.1	71.0	**88.0**
Potato	53.5	61.8	47.1	59.8	**65.5**
Rope	83.2	84.1	**98.6**	96.5	98.0
Tire	76.0	70.2	74.5	70.1	**86.2**
Mean	76.4	77.0	83.0	79.3	**86.4**

4.2.3 MT Defects

As shown in Table 5, the proposed model is dominant in AD with an AUROC score at 96.2%. But the average results of pixel-level localization are 83.3% AUROC and 70.3% PRO. DSN [25] achieves the best performance in AL, whose pixel-level AUROC score is at 98.0%, while our DSKD exceeds DSN by 7.1% in image-level detection.

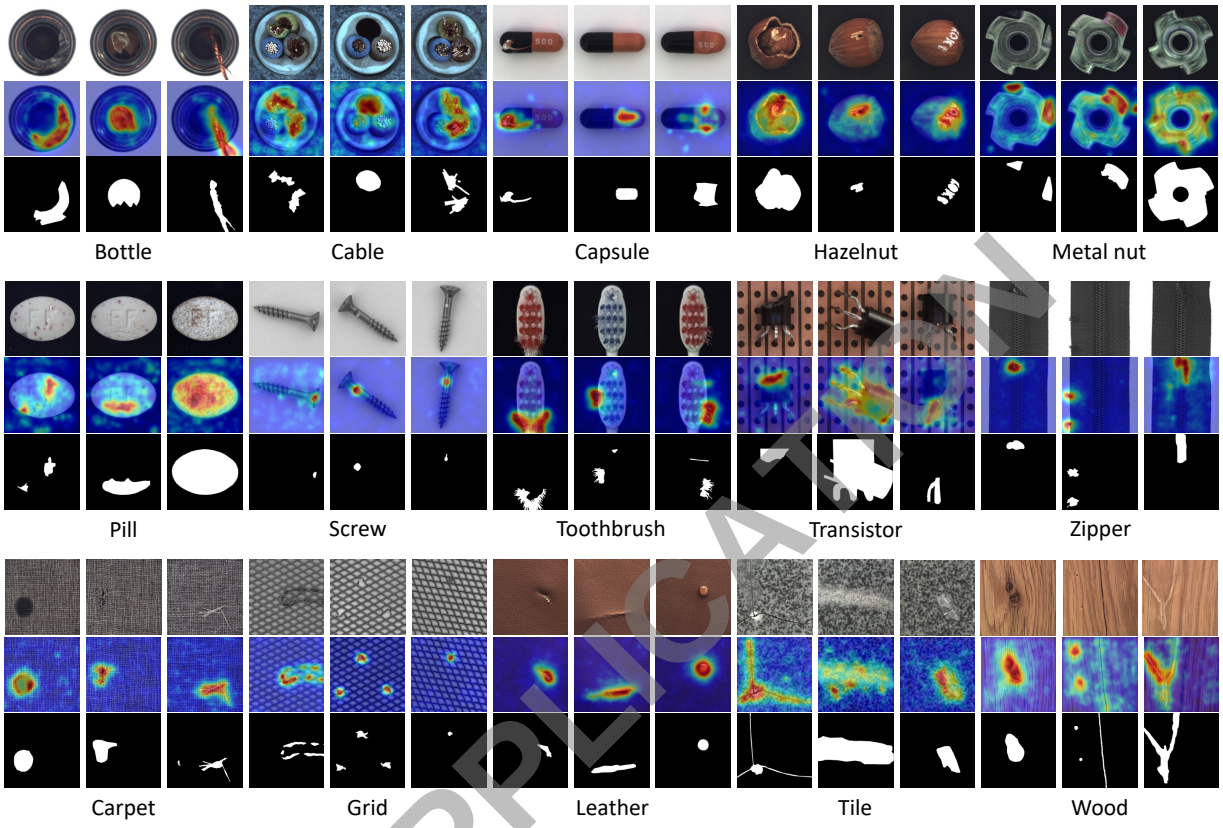

Figure 5. Visualization of anomaly localization results on 15 categories in MVTec AD. The first, second and third rows respectively represent the original defective images, anomaly maps and ground truth data

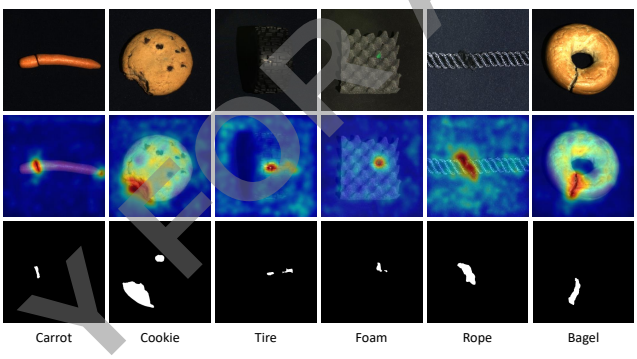

Figure 6. Visualization of anomaly localization performance in MVTec 3D-AD.

Table 4. Pixel-level Segmentation Results in MVTec 3D-AD (RGB)

Category	AUROC(100%)	PRO(100%)
Bagel	99.0	95.0
Cable Gland	99.1	98.8
Carrot	99.2	98.5
Cookie	97.9	89.0
Dowel	99.3	97.6
Foam	95.5	85.8
Peach	99.1	97.9
Potato	99.0	97.6
Rope	99.1	94.7
Tire	99.1	97.4
Mean	98.6	95.2

Fig.7 shows the visualization results of AL. There are five defective categories of magnetic tiles including fray, break, crack, uneven, and blowhole. Compared with the ground truth data, our model can recognize most regions with anomalies, which contributes the high accuracy of image-level detection. Nonetheless, not all defective pixels can be detected with a high anomaly score. For example, in the fray category as shown in Fig.7, the anomaly region can be highlighted, but many inner pixels are not marked with bright colors, which means that these pixels do not have high enough anomaly scores to be recognized as anomalies. Our model is sensitive to the edge of an anomaly region, whereas inside this area, the data distribution is uniform, which is more like an anomaly-free region. Therefore, many pixels cannot be correctly classified, and DSKD gets low accuracy in anomaly localization.

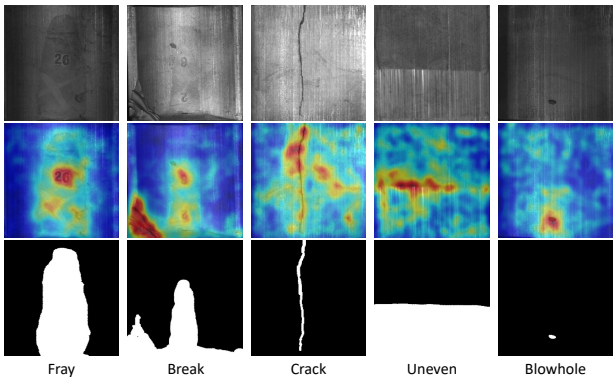

Figure 7. Visualization of anomaly localization performance on MT Defect.

Table 5. Anomaly Detection Results on MT Defects

Method	Image-level AUROC(100%)	Pixel-level AUROC(100%)	PRO(100%)
AE-SSIM[5]	60.1	78.3	-
RIAD[30]	75.2	82.7	-
FAVAE[8]	83.4	86.2	-
DSN[25]	89.1	**98.0**	-
Ours	**96.2**	83.3	**70.3**

Table 6. Comparison for the Results of Model Complexity

Method	Padim[7]	MKD[20]	Ours
Inf. time(s)	0.890	0.034	0.217
FLOPs(G)	22.85	20.96	6.17
Parameters(M)	68.9	15.0	39.3
Memory(MB)	3800	4	105

4.2.4 Model Complexity

The efficiency of anomaly inference plays a significant role in industrial inspection. Unlike the training process, anomaly scores in the inference phase are calculated by the CPU, which is an Intel i7, @2.3GHz in our experiments. As shown in Table 6, Padim [7] costs most resources in computing and memory. In spite of the high FLOPs at 20.96GB, MKD [20] achieves the best efficiency with 0.034s for inference and takes the smallest scale of parameters and memory. But MKD cannot perform well in AD and AL, as shown in Table 1 and 2. Our method has the fewest FLOPs at 6.17G. And compared with the other methods, DSKD has lower complexity in inference time and memory spaces.

4.3. Ablation Experiment

4.3.1 Influence of Dual-Student Model Architecture

To analyze the effectiveness of the proposed method based on dual-student networks, we conduct experiment on MVTec AD and compare the results with other three models with different structures, as shown in Fig.8. The dual-student, teacher-encoder, teacher-decoder and encoder-decoder structures are respectively marked as DS, T-E, T-D and E-D. Both of the image-level and pixel-level detection results are shown in Table 7. And the results for each category are shown in Fig.9. The DS model is much superior to others. The second best is T-D, with AUROC at the image level and pixel level, and PRO scores at 97.8%, 97.1% and 93.1%, respectively. Consequently, the discrepancy in the representation of anomalous features between T and S_d is the greatest, while the smallest is between S_e and S_d. Besides, this experiment can also demonstrate that interaction between S_e and S_d can effectively strengthen the ability to detect the representative discrepancy and improve the performance.

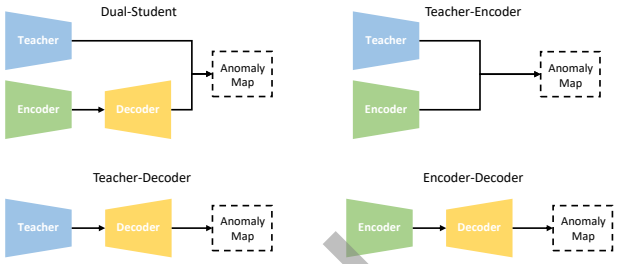

Figure 8. Structures in ablation experiments. Both deep feature embedding and multi-scale feature fusion blocks are used in this experiment.

Table 7. Ablation Experiments on Model Architecture on MVTec AD

Structure	Image-level AUROC(100%)	Pixel-level AUROC(100%)	PRO(100%)
DS	**98.5**	**97.6**	**93.5**
T-E	95.3	96.1	89.4
T-D	97.8	97.1	93.1
E-D	93.2	94.6	87.7

4.3.2 Influence of Deep Feature Embedding

To study the function of the DFE module, we remove the multiscale feature map fusion step between S_e and S_d, and the output of S_e is directly downsampled and then fed to S_d. As shown in Table 8, the scores of the three metrics, image-level AUROC, pixel-level AUROC and PRO, of the model with DFE module are much superior to those of the model without DFE module by 22.6%, 7.9% and 17.0%. Therefore, the embedding of S_e carries a lot of semantic information that can help S_d understand the representation of feature-free. And the detection model can eventually achieve better performance.

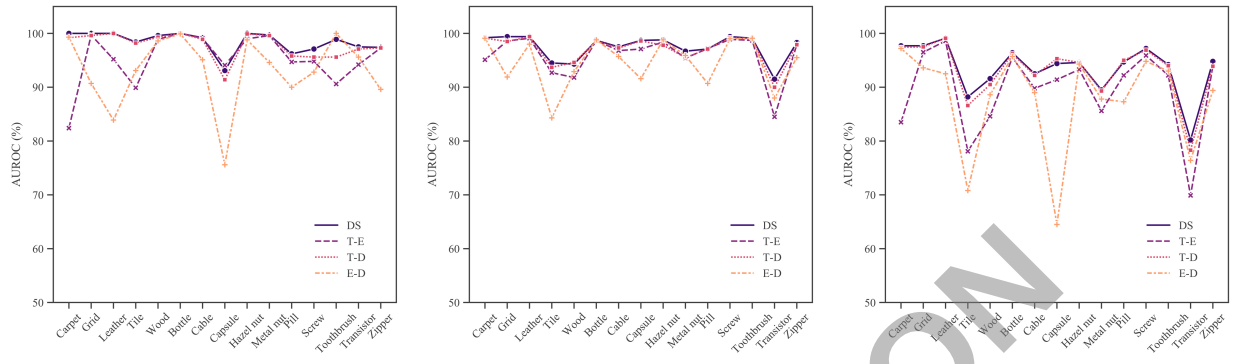

Figure 9. Results for each category in MVTec AD in ablation experiments.

Table 8. Ablation Experiments on Deep Feature Embedding

DFE	Image-level AUROC(100%)	Pixel-level AUROC(100%)	PRO(100%)
✗	75.6	89.6	76.5
✓	**98.5**	**97.6**	**93.5**

Table 9. Ablation Experiments on Multiple Feature Maps Fusion

Used Map	Image-level AUROC(100%)	Pixel-level AUROC(100%)	PRO(100%)
M_3	92.7	94.2	88.7
M_2	96.5	96.4	91.0
M_1	96.0	95.9	88.0
M_{1-3}	**98.5**	**97.5**	**93.5**

4.3.3 Influence of Multi-scale Feature Fusion

In the anomaly inference phase, DSKD uses feature maps of three different sizes and obtains the anomaly map by multi-scale fusion, as discussed in Sec 3. To investigate the influence of this operation, we compared the results of models that use anomaly maps of different sizes, as shown in Table 9. The M_1, M_2 and M_3 correspond to M^1_{t-d}, M^2_{t-d} and M^3_{t-d} in Fig. 3, respectively. And M_{1-3} means that the all of the anomaly maps are used. M_2 performs best in three single-layer anomaly maps, with scores of image-level AUROC, pixel-level AUROC and PRO at 96.5%, 96.4% and 91.0%. But the results of M_{1-3} achieve the highest scores, which prove the advantage of multi-scale anomaly maps fusion.

5. Conclusion

In this paper, we proposed a dual-student KD-based model, DSKD, for unsupervised anomaly detection in industrial inspection. DSKD consists of a teacher network and two inverted student networks. The experimental results show that our method achieves great performance with low complexity and a small model size. And the dual-student architecture and its internal modules are proven to be effective in AD and AL.

However, DSKD is not sensitive to the inner area of anomalies, especially for large-area defects. On the other hand, if the target region is small, the neighbor pixels are likely to be recognized as anomalies as well. To mitigate these problems, feature representation methods with higher robustness are supposed to be explored. And we may improve DSKD by optimizing objective functions, adjusting backbones or employing data augmentation, etc. Furthermore, we only study the performance of DSKD in 2D images, which can be extended to 3D images in the future, which can better show the structure and details of real-world objects. 3D anomaly detection will be a new research trend.

References

[1] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4*, pages 161–169. Springer, 2019.

[2] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021.

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

[4] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. *arXiv preprint arXiv:2112.09045*, 2021.

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

[6] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences, 2021.

[7] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization, 2020.

[8] David Dehaene and Pierre Eline. Anomaly localization by modeling perceptual features, 2020.

[9] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding, 2022.

[10] Eric Golinko and Xingquan Zhu. Generalized feature embedding for supervised, unsupervised, and online learning tasks. *Information Systems Frontiers*, 21:125–142, 2019.

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

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] Chuanfei Hu, Kai Chen, and Hang Shao. A semantic-enhanced method based on deep svdd for pixel-wise anomaly detection. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021.

[14] Yibin Huang, Congying Qiu, and Kui Yuan. Surface defect saliency of magnetic tile. *The Visual Computer*, 36:85–96, 2020.

[15] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9664–9674, 2021.

[16] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.

[17] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection, 2022.

[18] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows, 2020.

[19] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022.

[20] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad Hossein Rohban, and Hamid R. Rabiee. Multiresolution knowledge distillation for anomaly detection, 2020.

[21] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Margarethe Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019.

[22] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, 2017.

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

[24] Xian Tao, Xinyi Gong, Xin Zhang, Shaohua Yan, and Chandranath Adak. Deep learning for unsupervised anomaly localization in industrial images: A survey. *IEEE Transactions on Instrumentation and Measurement*, 2022.

[25] Xian Tao, Dapeng Zhang, Wenzhi Ma, Zhanxin Hou, Zhenfeng Lu, and Chandranath Adak. Unsupervised anomaly detection for surface defects with dual-siamese network. *IEEE Transactions on Industrial Informatics*, 18(11):7707–7717, 2022.

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

[27] Qian Wan, Liang Gao, Xinyu Li, and Long Wen. Unsupervised image anomaly detection and segmentation based on pretrained feature mapping. *IEEE Transactions on Industrial Informatics*, 19(3):2330–2339, 2023.

[28] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for anomaly detection, 2021.

[29] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian conference on computer vision*, 2020.

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