DeSTSeg: Segmentation Guided Denoising Student-Teacher for Anomaly Detection

Xuan Zhang[1], Shiyu Li[2], Xi Li[2], Ping Huang[2], Jiulong Shan[2], Ting Chen[1]
[1]Tsinghua University [2]Apple
x-zhang18@mails.tsinghua.edu.cn, {shiyu_li, weston_li, huang_ping, jlshan}@apple.com,
tingchen@tsinghua.edu.cn

Abstract

Visual anomaly detection, an important problem in computer vision, is usually formulated as a one-class classification and segmentation task. The student-teacher (S-T) framework has proved to be effective in solving this challenge. However, previous works based on S-T only empirically applied constraints on normal data and fused multi-level information. In this study, we propose an improved model called DeSTSeg, which integrates a pre-trained teacher network, a denoising student encoder-decoder, and a segmentation network into one framework. First, to strengthen the constraints on anomalous data, we introduce a denoising procedure that allows the student network to learn more robust representations. From synthetically corrupted normal images, we train the student network to match the teacher network feature of the same images without corruption. Second, to fuse the multi-level S-T features adaptively, we train a segmentation network with rich supervision from synthetic anomaly masks, achieving a substantial performance improvement. Experiments on the industrial inspection benchmark dataset demonstrate that our method achieves state-of-the-art performance, 98.6% on image-level AUC, 75.8% on pixel-level average precision, and 76.4% on instance-level average precision.

1. Introduction

Visual anomaly detection (AD) with localization is an essential task in many computer vision applications such as industrial inspection [24, 36], medical disease screening [27, 32], and video surveillance [18, 20]. The objective of these tasks is to identify both corrupted images and anomalous pixels in corrupted images. As anomalous samples occur rarely, and the number of anomaly types is enormous, it is unlikely to acquire enough anomalous samples with all possible anomaly types for training. Therefore, AD tasks were usually formulated as a one-class classification and segmentation, using only normal data for model training.

The student-teacher (S-T) framework, known as knowledge distillation, has proven effective in AD [3, 9, 26, 31, 33]. In this framework, a teacher network is pre-trained on a large-scale dataset, such as ImageNet [10], and a student network is trained to mimic the feature representations of the teacher network on an AD dataset with normal samples only. The primary hypothesis is that the student network will generate different feature representations from the teacher network on anomalous samples that have never been encountered in training. Consequently, anomalous pixels and images can be recognized in the inference phase. Notably, [26, 31] applied knowledge distillation at various levels of the feature pyramid so that discrepancies from multiple layers were aggregated and demonstrated good performance. However, there is no guarantee that the features of anomalous samples are always different between S-T networks because there is no constraint from anomalous samples during the training. Even with anomalies, the student network may be over-generalized [22] and output similar feature representations as those by the teacher network. Furthermore, aggregating discrepancies from multi-level in an empirical way, such as sum or product, could be suboptimal. For instance, in the MVTec AD dataset under the same context of [31], we observe that for the category of *transistor*, employing the representation from the last layer, with 88.4% on pixel-level AUC, outperforms that from the multi-level features, with 81.9% on pixel-level AUC.

To address the problem mentioned above, we propose **DeSTSeg**, illustrated in Fig. 1, which consists of a denoising student network, a teacher network, and a segmentation network. We introduce random synthetic anomalies into the normal images and then use these corrupted images[1] for training. The denoising student network takes a corrupted image as input, whereas the teacher network takes the original clean image as input. During training,

[1]All samples shown in this paper are licensed under the CC BY-NC-SA 4.0.

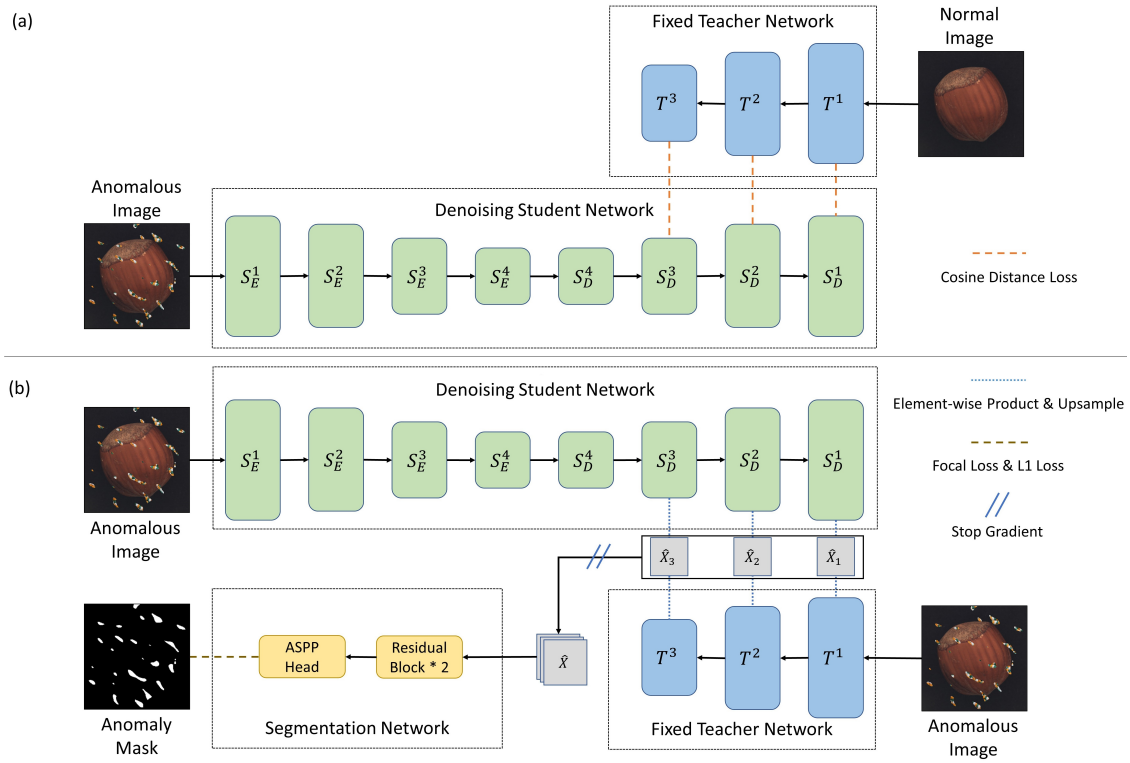

Figure 1. Overview of DeSTSeg. Synthetic anomalous images are generated and used during training. In the first step (a), the student network with synthetic input is trained to generate similar feature representations as the teacher network from the clean image. In the second step (b), the element-wise product of the student and teacher networks' normalized outputs are concatenated and utilized to train the segmentation network. The segmentation output is the predicted anomaly score map.

the feature discrepancy between the two networks is minimized. In other words, the student network is trained to perform denoising in the feature space. Given anomalous images as input to both networks, the teacher network encodes anomalies naturally into features, while the trained denoising student network filters anomalies out of feature space. Therefore, the two networks are reinforced to generate distinct features from anomalous inputs. For the architecture of the denoising student network, we decided to use an encoder-decoder network for better feature denoising instead of adopting an identical architecture as the teacher network. In addition, instead of using empirical aggregation, we append a segmentation network to fuse the multi-level feature discrepancies in a trainable manner, using the generated binary anomaly mask as the supervision signal.

We evaluate our method on a benchmark dataset for surface anomaly detection and localization, MVTec AD [2]. Extensive experimental results show that our method outperforms the state-of-the-art methods on image-level, pixel-level, and instance-level anomaly detection tasks. We also conduct ablation studies to validate the effectiveness of our proposed components.

Our main contributions are summarized as follows. (1) We propose a denoising student encoder-decoder, which is trained to explicitly generate different feature representations from the teacher with anomalous inputs. (2) We employ a segmentation network to adaptively fuse the multi-level feature similarities to replace the empirical inference approach. (3) We conduct extensive experiments on the benchmark dataset to demonstrate the effectiveness of our method for various tasks.

2. Related Works

Anomaly detection and localization have been studied from numerous perspectives. In **image reconstruction**, researchers used autoencoder [4], variational autoencoder [1, 30] or generative adversarial network [21, 27, 28] to train an image reconstruction model on normal data. The presumption is that anomalous images cannot be reconstructed effectively since they are not seen during training, so the difference between the input and reconstructed images can be used as pixel-level anomaly scores. However, anomaly regions still have a chance to be accurately reconstructed due to the over-generalization issue [22]. Another perspective is the **parametric density estimation**, which

assumes that the extracted features of normal data obey a certain distribution, such as a multivariate Gaussian distribution [8, 15, 16, 23], and uses the normal dataset to estimate the parameters. Then, the outlier data are recognized as anomalous data by inference. Since the assumption of Gaussian distribution is too strict, some recent works borrow ideas from normalizing flow, by projecting an arbitrary distribution to a Gaussian distribution to approximate the density of any distributions [13, 35]. Besides, the **memory-based** approaches [7,19,24,34] build a memory bank of normal data in training. During inference, given a query item, the model selects the nearest item in the memory bank and uses the similarity between the query item and the nearest item to compute the anomaly score.

Knowledge distillation. Knowledge distillation is based on a pretrained teacher network and a trainable student network. As the student network is trained on an anomaly-free dataset, its feature representation of anomalies is expected to be distinct from that of the teacher network. Numerous solutions have been presented in the past to improve discrimination against various types of anomalies. For example, [3] used ensemble learning to train multiple student networks and exploited the irregularity of their feature representations to recognize the anomaly. [31], and [26] adopted multi-level feature representation alignment to capture both low-level and high-level anomalies. [9], and [33] designed decoder architectures for the student network to avoid the shortcomings of identical architecture and the same dataflow between S-T networks. These works focus on improving the similarity of S-T representations on normal inputs, whereas our work additionally attempts to differentiate their representations on anomalous input.

Anomaly simulation. Although there is no anomalous data for training in the context of one-class classification AD, the pseudo-anomalous data could be simulated so that an AD model can be trained in a supervised way. Classical anomaly simulation strategies, such as rotation [12] and cutout [11], do not perform well in detecting fine-grained anomalous patterns [16]. A simple yet effective strategy is called CutPaste [16] that randomly selects a rectangular region inside the original image and then copies and pastes the content to a different location within the image. Another strategy proposed in [36] and also adopted in [37] used two-dimensional Perlin noise to simulate a more realistic anomalous image. With the simulated anomalous images and corresponding ground truth masks, [29, 36, 37] localized anomalies with segmentation networks. In our system, we adopt the ideas of [36] for anomaly simulation and segmentation.

3. Method

The proposed DeSTSeg consists of three main components: a pre-trained teacher network, a denoising student network, and a segmentation network.

As illustrated in Fig. 1, synthetic anomalies are introduced into normal training images, and the model is trained in two steps. In the first step, the simulated anomalous image is utilized as the student network input, whereas the original clean image is the input to the teacher network. The weights of the teacher network are fixed, but the student network for denoising is trainable. In the second step, the student model is fixed as well. Both the student and teacher networks take the synthetic anomaly image as their input to optimize parameters in the segmentation network to localize the anomalous regions. For inference, pixel-level anomaly maps are generated in an end-to-end mode, and the corresponding image-level anomaly scores can be computed via post-processing.

3.1. Synthetic Anomaly Generation

The training of our model relies on synthetic anomalous images which are generated using the same algorithm proposed in [36]. Random two-dimensional Perlin noise is generated and binarized by a preset threshold to obtain an anomaly mask M. An anomalous image I_a is generated by replacing the mask region with a linear combination of an anomaly-free image I_n and an arbitrary image from external data source A, with an opacity factor β randomly chosen between $[0.15, 1]$.

$$I_a = \beta(M \odot A) + (1-\beta)(M \odot I_n) + (1-M) \odot I_n \quad (1)$$

$\odot$ means the element-wise multiplication operation. The anomaly generation is performed online during training. By using this algorithm, three benefits are introduced. First, compared to painting a rectangle anomaly mask [16], the anomaly mask generated by random Perlin noise is more irregular and similar to actual anomalous shapes. Second, the image used as anomaly content A could be arbitrarily chosen without elaborate selection [36]. Third, the introduction of opacity factor β can be regarded as a data augmentation [38] to effectively increase the diversity of the training set.

3.2. Denoising Student-Teacher Network

In previous multi-level knowledge distillation approaches [26, 31], the input of the student network (normal image) is identical to that of the teacher network, as is the architecture of the student network. However, our proposed denoising student network and the teacher network take paired anomalous and normal images as input, with the denoising student network having a distinct encoder-decoder architecture. In the following two paragraphs, we will examine the motivation for this design.

First, as mentioned in Sec. 1, an optimization target should be established to encourage the student network

to generate anomaly-specific features that differ from the teacher's. We further endow a more straightforward target to the student network: to build normal feature representations on anomalous regions supervised by the teacher network. As the teacher network has been pre-trained on a large dataset, it can generate discriminative feature representations in both normal and anomalous regions. Therefore, the denoising student network will generate different feature representations from those by the teacher network during inference. Besides, as mentioned in Sec. 2, the memory-based approaches look for the most similar normal item in the memory bank to the query item and use their similarity for inference. Similarly, we optimize the denoising student network to reconstruct the normal features.

Second, given the feature reconstruction task, we conclude that the student network should not copy the architecture of the teacher network. Considering the process of reconstructing the feature of an early layer, it is well known that the lower layers of CNN capture local information, such as texture and color. In contrast, the upper layers of CNN express global semantic information [9]. Recalling that our denoising student network should reconstruct the feature of the corresponding normal image from the teacher network, such a task relies on global semantic information of the image and could not be done perfectly with only a few lower layers. We notice that the proposed task design resembles image denoising, with the exception that we wish to denoise the image in the feature space. The encoder-decoder architecture is widely used for image denoising. Therefore, we adopted it as the denoising student network's architecture. There is an alternative way to use the teacher network as an encoder and reverse the student network as the decoder [9, 33]; however, our preliminary experimental results show that a complete encoder-decoder student network performs better. One possible explanation is that the pre-trained teacher network is usually trained on ImageNet with classification tasks; thus, the encoded features in the last layers lack sufficient information to reconstruct the feature representations at all levels.

Following [31], the teacher network is an ImageNet pre-trained ResNet18 [14] with the final block removed (i.e., conv5_x). The output feature maps are extracted from the three remaining blocks, i.e., conv2_x, conv3_x, and conv4_x denoted as T^1, T^2, and T^3, respectively. Regarding the denoising student network, the encoder is a randomly initialized ResNet18 with all blocks, named S_E^1, S_E^2, S_E^3, and S_E^4, respectively. The decoder is a reversed ResNet18 (by replacing all downsampling with bilinear upsampling) with four residual blocks, named S_D^4, S_D^3, S_D^2, and S_D^1, respectively.

We minimize the cosine distance between features from T^k and S_D^k, $k = 1, 2, 3$. Denoting $F_{T_k} \in \mathcal{R}^{C_k \times H_k \times W_k}$ the feature representation from layer T^k, and $F_{S_k} \in \mathcal{R}^{C_k \times H_k \times W_k}$ the feature representation from layer S_D^k, the cosine distances can be computed through Eq. (2) and Eq. (3). i and j stand for the spatial coordinate on the feature map. In particular, $i = 1...H_k$ and $j = 1...W_k$. The loss is the sum of distances across three different feature levels as shown in Eq. (4).

$$X_k(i,j) = \frac{F_{T_k}(i,j) \odot F_{S_k}(i,j)}{||F_{T_k}(i,j)||_2 ||F_{S_k}(i,j)||_2} \tag{2}$$

$$D_k(i,j) = 1 - \sum_{c=1}^{C_k} X_k(i,j)_c \tag{3}$$

$$L_{cos} = \sum_{k=1}^{3} \left(\frac{1}{H_k W_k} \sum_{i,j=1}^{H_k, W_k} D_k(i,j) \right) \tag{4}$$

3.3. Segmentation Network

In [26,31], the cosine distances from multi-level features are summed up directly to represent the anomaly score of each pixel. However, the results can be suboptimal if discriminations of all level features are not equally accurate. To address this issue, we add a segmentation network to guide the feature fusion with additional supervision signals.

We freeze the weights of both the student and teacher networks to train the segmentation network. The synthetic anomalous image is utilized as the input for both S-T networks, and the corresponding binary anomaly mask is the ground truth. The similarities of the feature maps (T^1, S_D^1), (T^2, S_D^2), (T^3, S_D^3) are calculated by Eq. (2) and upsampled to the same size as X_1, which is 1/4 of the input size. The upsampled features, denoted as $\hat{X}_1$, $\hat{X}_2$, and $\hat{X}_3$, are then concatenated as $\hat{X}$, which is fed into the segmentation network. We also investigate alternative ways to compute the input of the segmentation network in Sec. 4.4. The segmentation network contains two residual blocks and one Atrous Spatial Pyramid Pooling (ASPP) module [5]. There is no upsampling or downsampling; thus, the output size equals the size of X_1. Although this may lead to resolution loss to some extent, it reduces the memory cost for training and inference, which is crucial in practice.

The segmentation training is optimized by employing the focal loss [17] and the L1 loss. In the training set, the majority of pixels are normal and easily recognized as background. Only a small portion of the image consists of anomalous pixels that must be segmented. Therefore, the focal loss can help the model to focus on the minority category and difficult samples. In addition, the L1 loss is employed to improve the sparsity of the output so that the segmentation mask's boundaries are more distinct. To compute the loss, we downsample the ground truth anomaly mask to a size equal to 1/4 of the input size, which matches the output (H_1, W_1). Mathematically, we denote the output

Figure 2. The binary ground truth is downsampled with different implementations. (a) The grid image with a crack anomaly. (b) Downsample with bilinear interpolation, then floor all values between $(0, 1)$ to zero [13, 24]. The mask has almost vanished. (c) Downsample with bilinear interpolation, then ceil all values between $(0, 1)$ to one [36]. The mask is thicker than expected. (d) Downsample with the nearest interpolation, interrupting the original contiguous region. (e) Our proposed approach, downsample with bilinear interpolation and round values by threshold=0.5, The original contiguous region is not interrupted.

probability map as $\hat{Y}$ and the downsampled anomaly mask as M, and the focal loss is computed using Eq. (5) where $p_{ij} = M_{ij}\hat{Y}_{ij} + (1 - M_{ij})(1 - \hat{Y}_{ij})$ and γ is the focusing parameter. The L1 loss is computed by Eq. (6), and the segmentation loss is computed by Eq. (7).

$$L_{focal} = -\frac{1}{H_1 W_1} \sum_{i,j=1}^{H_1,W_1} (1 - p_{ij})^{\gamma} \log(p_{ij}) \tag{5}$$

$$L_{l1} = \frac{1}{H_1 W_1} \sum_{i,j=1}^{H_1,W_1} |M_{ij} - \hat{Y}_{ij}| \tag{6}$$

$$L_{seg} = L_{focal} + L_{l1} \tag{7}$$

3.4. Inference

In the inference stage, the test image is fed into both the teacher and student networks. The segmentation prediction is finally upsampled to the input size and taken as the anomaly score map. It is expected that anomalous pixels in the input image will have greater values in the output. To calculate the image-level anomaly score, we use the average of the top T values from the anomaly score map, where T is a tuning hyperparameter.

4. Experiments

4.1. Dataset

We evaluate our method using the MVTec AD [2] dataset, which is one of the most widely used benchmarks for anomaly detection and localization. The dataset comprises 15 categories, including 10 objects and 5 textures. For each category, there are hundreds of normal images for training and a mixture of anomalous and normal images for evaluation. The image sizes range from 700×700 to 1024×1024 pixels. For evaluation purposes, pixel-level binary annotations are provided for anomalous images in the test set. In addition, the Describable Textures Dataset (DTD) [6] is used as the anomaly source image A in Eq. (1). [36] showed that other datasets such as ImageNet can achieve comparable performance but DTD is much smaller and easy to use.

4.2. Evaluation Metrics

Image-level evaluation. Following the previous work in anomaly detection work, AUC (i.e., area under the ROC curve) is utilized to evaluate image-level anomaly detection.

Pixel-level evaluation. AUC is also selected to evaluate the pixel-level result. Additionally, we report average precision (AP) since it is a more appropriate metric for heavily imbalanced classes [25].

Instance-level evaluation. In real-world applications, such as industrial defect inspection and medical imaging lesion detection, users are more concerned with whether the model can fully or partially localize an instance than with each individual pixel. In [3], per-region-overlap (PRO) is proposed, which equally weights the connected components of different sizes in the ground truth. It computes the overlap between prediction and ground truth within a user-specified false positive rate (30%). However, because instance recall is essential in practice, we propose to use instance average precision (IAP) as a more straightforward metric. Formally, we define an anomaly instance as a maximally connected ground truth region. Given a prediction map, an anomalous instance is considered detected if and only if more than 50% of the region pixels are predicted as positive. Under different thresholds, a list of pixel-level precision and instance-level recall rate points can be drawn as a curve. The average precision of this curve is calculated as IAP. For those applications requiring an extremely high recall, the precision at $recall = k\%$ is also computed and denoted as IAP@k. In our experiments, we evaluate our model under a high-stakes scenario by setting $k = 90$.

Ground truth downsampling method. We notice that the prior implementations of pixel-level evaluation are poorly aligned. Most of the works downsampled the ground

US [3]	STPM [31]	CutPaste [16]	DRAEM [36]	DSR [37]	PatchCore [24]	Ours
87.7	95.1	95.2	98.0	98.2	98.5	**98.6**±0.4

Table 1. Image-level anomaly detection AUC (%) on MVTec AD dataset. Results are averaged over all categories.

	US [3]	STPM [31]	CutPaste [16]	DRAEM [36]	DSR [37]	PatchCore [24]	Ours
bottle	97.8 / 74.2	98.8 / 80.6	97.6 / -	**99.3** / 89.8	- / **91.5**	98.9 / 80.1	99.2±0.2 / 90.3±1.8
cable	91.9 / 48.2	94.8 / 58.0	90.0 / -	95.4 / 62.6	- / **70.4**	**98.8** / 70.0	97.3±0.4 / 60.4±2.3
capsule	96.8 / 25.9	98.2 / 35.9	97.4 / -	94.1 / 43.5	- / 53.3	**99.1** / 48.1	**99.1**±0.0 / **56.3**±1.1
carpet	93.5 / 52.2	**99.1** / 65.3	98.3 / -	96.2 / 64.4	- / **78.2**	**99.1** / 66.7	96.1±2.2 / 72.8±5.8
grid	89.9 / 10.1	99.1 / 45.4	97.5 / -	**99.5** / 56.8	- / **68.0**	98.9 / 41.0	99.1±0.1 / 61.5±1.6
hazelnut	98.2 / 57.8	98.9 / 60.3	97.3 / -	99.5 / 88.1	- / 87.3	99.0 / 61.5	**99.6**±0.2 / **88.4**±2.2
leather	97.8 / 40.9	99.2 / 42.9	99.5 / -	98.9 / 69.9	- / 62.5	99.4 / 51.0	**99.7**±0.0 / **75.6**±1.2
metal_nut	97.2 / 83.5	97.2 / 79.3	93.1 / -	98.7 / 91.7	- / 67.5	**98.8** / 88.8	98.6±0.4 / **93.5**±1.1
pill	96.5 / 62.0	94.7 / 63.3	95.7 / -	97.6 / 46.1	- / 65.7	98.2 / 78.7	**98.7**±0.4 / **83.1**±4.2
screw	97.4 / 7.8	98.6 / 26.9	96.7 / -	**99.7** / **71.5**	- / 52.5	99.5 / 41.4	98.5±0.3 / 58.7±3.7
tile	92.5 / 65.3	96.6 / 61.7	90.5 / -	**99.5** / **96.9**	- / 93.9	96.6 / 59.3	98.0±0.7 / 90.0±2.5
toothbrush	97.9 / 37.7	98.9 / 48.8	98.1 / -	98.1 / 54.7	- / 74.2	98.9 / 51.6	**99.3**±0.1 / **75.2**±1.8
transistor	73.7 / 27.1	81.9 / 44.4	93.0 / -	90.0 / 51.7	- / 41.1	**96.2** / 63.2	89.1±3.4 / **64.8**±4.0
wood	92.1 / 53.3	95.2 / 47.0	95.5 / -	97.0 / 80.5	- / 68.4	95.1 / 52.3	**97.7**±0.3 / **81.9**±1.2
zipper	95.6 / 36.1	98.0 / 54.9	99.3 / -	98.6 / 72.3	- / 78.5	99.0 / 64.0	**99.1**±0.5 / **85.2**±3.3
average	93.9 / 45.5	96.6 / 54.3	96.0 / -	97.5 / 69.3	- / 70.2	**98.4** / 61.2	97.9±0.3 / **75.8**±0.8

Table 2. Pixel-level anomaly localization AUC / AP (%) on MVTec AD dataset.

truth to 256×256 for faster computation, but some performed an extra 224×224 center crop [7, 8, 24]. In addition, the downsampling implementations are not standardized either [13,24,36], resulting in the varying ground truth and unfair evaluation. In some cases, the downsampling introduces severe distortion, as illustrated in Fig. 2. In our work, we use bilinear interpolation to downsample the binary mask to 256×256 and then round the result with a threshold of 0.5. This implementation can preserve the continuity of the original ground truth mask without over or under-estimating.

4.3. Results

In order to make fair comparisons with other works, we re-evaluated the official pre-trained models of [31], [24], and [36] using our proposed evaluation introduced in Sec. 4.2. For methods without open-source code, we use the results mentioned in the original papers. Unavailable results are denoted with '-'. We repeat the experiments of our method 5 times with different random seeds to report the standard deviation.

Image-level anomaly detection. We report the AUC for the image-level anomaly detection task in Tab. 1. The performance of our method reaches state-of-the-art on average. Category-specific results are shown in the supplementary material.

Pixel-level anomaly localization. We report the AUC and AP values for the pixel-level anomaly localization task in Tab. 2. On average, our method outperforms state-of-the-art by 5.6% on AP and achieves AUC scores comparable to PatchCore [24]. Our method reaches the highest or near-highest score in the majority of categories, indicating that our approach generalizes well over a wide range of industrial application scenes.

Instance-level anomaly detection. The IAP and IAP@90 of the instance-level anomaly detection are reported in Tab. 3. Our method achieves the state of the art for both metrics. On average, our approach reaches an IAP@90 of 57.8%, which indicates that when 90% of anomaly instances are detected, the pixel-level precision is 57.8%, or equally, the pixel-level false positive rate is 42.2%. As some categories (e.g., carpet, pill) contain hard samples close to the decision boundary, their standard deviations of IAP@90 are relatively high. In practice, these metrics can be used to determine whether the performance is acceptable for an application.

Category-specific analysis. For the category cable, memory-based approaches [24,37] have better performance than ours since the normal pixels have larger intra-class distances than categories with periodic textures. For the categories grid, screw, and tile, the anomalies are relatively small or thin. Therefore, methods with higher resolution

	STPM [31]	DRAEM [36]	PatchCore [24]	Ours
bottle	83.2 / 73.3	90.3 / **84.8**	81.8 / 70.1	**90.5**±1.7 / 82.5±4.1
cable	54.9 / 17.2	47.0 / 10.8	**69.2** / **50.6**	51.1±2.5 / 26.7±3.7
capsule	37.2 / 17.9	**50.7** / 21.4	44.2 / 26.9	49.4±1.5 / **27.3**±3.3
carpet	68.4 / 52.2	76.8 / 32.3	64.4 / 43.7	**84.5**±4.9 / **58.6**±17.1
grid	45.7 / 21.0	55.5 / 42.3	39.1 / 15.6	**61.6**±1.8 / **47.4**±2.9
hazelnut	64.8 / 56.2	**95.7** / **89.0**	63.8 / 52.5	87.7±1.8 / 77.6±3.4
leather	46.2 / 24.9	**78.6** / 55.0	50.1 / 30.1	77.5±1.8 / **65.3**±3.9
metal_nut	83.4 / 81.7	92.6 / 83.9	90.1 / 84.6	**93.6**±1.3 / **86.5**±2.7
pill	72.0 / 45.5	46.9 / 41.5	82.7 / **63.5**	**84.8**±3.8 / 61.1±12.4
screw	24.4 / 4.2	**68.8** / **33.0**	38.4 / 16.3	53.6±3.6 / 8.6±2.3
tile	62.9 / 55.3	**98.9** / **98.2**	60.0 / 52.1	94.7±1.8 / 86.5±3.6
toothbrush	41.9 / 23.4	44.7 / 21.5	40.4 / 22.1	**59.8**±2.9 / **32.1**±5.1
transistor	53.4 / 8.5	59.3 / 22.8	69.9 / 36.8	**78.3**±2.5 / **49.6**±8.4
wood	56.0 / 35.4	**88.4** / 72.6	59.7 / 35.6	87.8±2.8 / **76.4**±3.4
zipper	59.1 / 46.6	78.7 / 67.0	66.0 / 52.4	**90.6**±2.3 / **80.3**±4.9
average	56.9 / 37.5	71.5 / 51.7	61.3 / 43.5	**76.4**±1.0 / **57.8**±1.8

Table 3. Instance-level anomaly detection IAP / IAP@90 (%) on MVTec AD dataset.

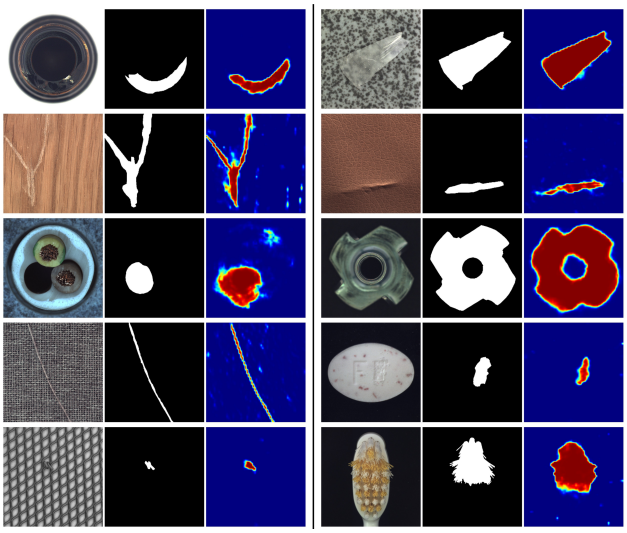

Figure 3. Visualization examples of our method. For each example, left: input image; middle: ground truth; right: prediction map.

predictions, such as [36, 37], can achieve higher performance, but require more memory and computation. For the remaining categories, our method achieves comparable or higher performance than the compared methods.

Visualization examples. Several visualization examples of our method from various categories are presented in Fig. 3. Our method can precisely localize the anomaly regions. More examples are shown in the supplementary material.

Analysis of failure cases. We analyze some failure cases illustrated in Fig. 4. On the one hand, several ambiguous

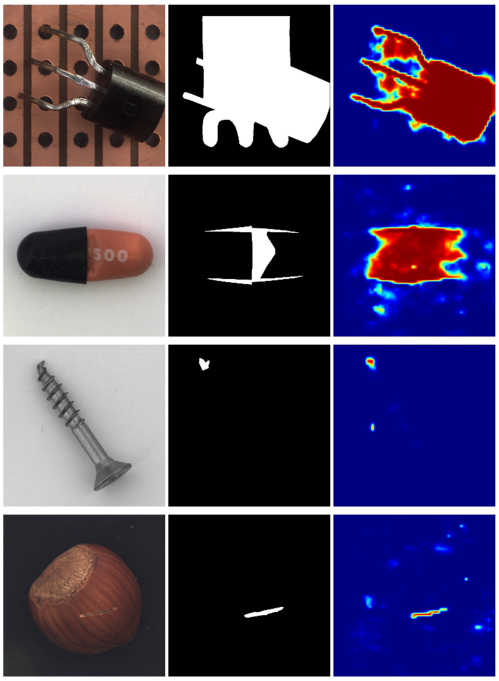

Figure 4. Failure cases of our method. The examples are chosen from transistor, capsule, screw, and hazelnut (from top to bottom). For each example, left: input image; middle: ground truth; right: prediction map.

ground truths are responsible for a number of failure occurrences. In a transistor case from the first row, the ground truth highlights both the original and misplaced location, while the prediction mask only covers the misplaced loca-

tion. For a capsule case shown in the second tow, the ground truth contains most of the distorted parts, whereas the prediction mask covers the entire capsule. In these cases, we would argue that our predictions are still useful.

On the other hand, some failure cases, such as those shown in the third and fourth rows, result from noisy backgrounds. Tiny fibers and stains are highlighted due to the susceptibility of our model. We leave it to future work to investigate whether these anomalies are acceptable in order to draw more accurate conclusions.

4.4. Ablation Studies

Network architecture. In Tab. 4, we evaluate the effectiveness of our three designs: replacing the training inputs of the vanilla student network with synthetic anomalies to enable a denoising procedure (**den**), applying encoder-decoder architecture to the student network(**ed**), and appending the segmentation network (**seg**) to replace the empirical feature fusion strategy, i.e., a product of cosine distances [31]. (a) Comparing experiments 1 and 2, it can be found that only changing the student network's input to anomalous images undermines performance. However, experiment 5 shows improvement when **ed** is added, indicating that the **den** can be boosted by adopting **ed** architecture. (b) The comparisons of experiments 1 with 4, 2 with 6, 3 with 7, and 5 with 8, showcase that the segmentation network can significantly improve the performances of all three metrics. (c) Comparing experiments 4 and 8, it can be found that the combination of **den** and **ed** provides more useful features for the segmentation network than a vanilla S-T network does. The best result is achieved by combining all three main designs.

Segmentation loss. In Tab. 5, we examine the effectiveness of the L1-loss in the segmentation loss (Eq. (7)). It can be observed that the L1-loss improves performance.

Segmentation network input. As mentioned in Sec. 3.3, the input of the segmentation network is the element-wise product between the normalized feature maps of S-T networks as defined by Eq. (2). To prove the rationality of this setting, we build two distinct feature combinations as input. The first is to directly concatenate the feature maps of S-T networks F_{S_k} and F_{T_k} as the input of the segmentation network, which preserves the information of the S-T networks more effectively. The second is to compute the cosine distance of the S-T networks' feature maps using Eq. (3), which utilizes more prior information when we train the student network by optimizing the cosine distance. We show the results in Tab. 6. Both approaches result in suboptimal performance, indicating that $\hat{X}$ is a suitable choice as the input to balance the information and prior.

Exp.	den	ed	seg	img (AUC)	pix (AP)	ins (IAP)
1				94.8	52.9	55.8
2	✓			93.4	49.6	53.9
3		✓		95.4	53.3	57.7
4			✓	97.3	70.1	71.8
5	✓	✓		94.5	54.0	58.5
6	✓		✓	97.3	70.9	72.3
7		✓	✓	97.7	69.7	71.2
8	✓	✓	✓	**98.6**	**75.8**	**76.4**

Table 4. Ablation studies on our main designs: denoising training (den), the encoder-decoder architecture of student network (ed), and segmentation network (seg). AUC, AP, and IAP (%) are used to evaluate image-level, pixel-level, and instance-level detection, respectively. Exp. 1 uses the same architecture of [31], but different training settings to align with Exp. 2∼8.

	img (AUC)	pix (AP)	ins (IAP)
w/o L1 loss	97.9	72.2	74.4
w/ L1 loss	**98.6**	**75.8**	**76.4**

Table 5. Ablation studies on the segmentation loss: AUC, AP, and IAP (%) are used to evaluate image-level, pixel-level, and instance-level detection, respectively.

	img (AUC)	pix (AP)	ins (IAP)
concatenated-ST input	98.0	72.2	72.6
cosine-distance input	98.5	72.0	74.5
DeSTSeg	**98.6**	**75.8**	**76.4**

Table 6. Ablation studies on the input of segmentation network: AUC, AP, and IAP (%) are used to evaluate image-level, pixel-level, and instance-level detection, respectively.

5. Conclusion

We propose the DeSTSeg, a segmentation-guided denoising student-teacher framework for the anomaly detection task. The denoising student-teacher network is adopted to enable the S-T network to generate discriminative features in anomalous regions. The segmentation network is built to fuse the S-T network features adaptively. Experiments on the surface anomaly detection benchmark show that all of our proposed components considerably boost performance. Our results outperform the previous state-of-the-art by 0.1% AUC for image-level anomaly detection, 5.6% AP for pixel-level anomaly localization, and 4.9% IAP for instance-level anomaly detection.

References

[1] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI brainlesion workshop*, pages 161–169. Springer, 2018. 2

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

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4183–4192, 2020. 1, 3, 5, 6

[4] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 372–380. SciTePress, 2019. 2

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 4

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 5

[7] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 3, 6

[8] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 3, 6

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

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[11] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[12] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 3

[13] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 98–107, 2022. 3, 5, 6

[14] 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. 4

[15] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 3

[16] 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. 3, 6

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4

[18] Yusha Liu, Chun-Liang Li, and Barnabás Póczos. Classifier two sample test for video anomaly detections. In *Proceedings of the British Machine Vision Conference*, page 71, 2018. 1

[19] Tiago S Nazare, Rodrigo F de Mello, and Moacir A Ponti. Are pre-trained cnns good feature extractors for anomaly detection in surveillance videos? *arXiv preprint arXiv:1811.08495*, 2018. 3

[20] Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12173–12182, 2020. 1

[21] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2898–2906, 2019. 2

[22] Jonathan Pirnay and Keng Chai. Inpainting transformer for anomaly detection. In *International Conference on Image Analysis and Processing*, pages 394–406. Springer, 2022. 1, 2

[23] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6726–6733. IEEE, 2021. 3

[24] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022. 1, 3, 5, 6, 7

[25] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015. 5

[26] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14902–14912, 2021. 1, 3, 4

[27] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44, 2019. 1, 2

[28] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017. 2

[29] Jouwon Song, Kyeongbo Kong, Ye-In Park, Seong-Gyun Kim, and Suk-Ju Kang. Anoseg: Anomaly segmentation network using self-supervised learning. *arXiv preprint arXiv:2110.03396*, 2021. 3

[30] Aleksei Vasilev, Vladimir Golkov, Marc Meissner, Ilona Lipp, Eleonora Sgarlata, Valentina Tomassini, Derek K Jones, and Daniel Cremers. q-space novelty detection with variational autoencoders. In *Computational Diffusion MRI*, pages 113–124. Springer, 2020. 2

[31] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for anomaly detection. In *Proceedings of the British Machine Vision Conference*, 2021. 1, 3, 4, 6, 7, 8

[32] Haruna Watanabe, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Bone metastatic tumor detection based on anogan using ct images. In *2019 IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech)*, pages 235–236. IEEE, 2019. 1

[33] Shinji Yamada and Kazuhiro Hotta. Reconstruction student with attention for student-teacher pyramid matching. *arXiv preprint arXiv:2111.15376*, 2021. 1, 3, 4

[34] 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. 3

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

[36] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draema discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021. 1, 3, 5, 6, 7

[37] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Dsr–a dual subspace re-projection network for surface anomaly detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 539–554. Springer, 2022. 3, 6, 7

[38] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference for Learning Representations (ICLR)*, 2018. 3