# Learning Transferable Representations for Image Anomaly Localization Using Dense Pretraining

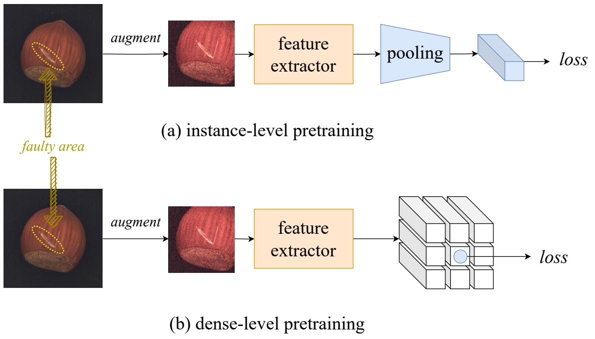
Haitian He[1] Sarah Erfani[1] Mingming Gong[1] Qiuhong Ke[2]
[1]The University of Melbourne [2]Monash University
{haitianh@student., sarah.erfani@, mingming.gong@}unimelb.edu.au qiuhong.ke@monash.edu

## Abstract

*Image anomaly localization (IAL) is widely applied in fault detection and industrial inspection domains to discover anomalous patterns in images at the pixel level. The unique challenge of this task is the lack of comprehensive anomaly samples for model training. The state-of-the-art methods train end-to-end models that leverage outlier exposure to simulate pseudo anomalies, but they show poor transferability to new datasets due to the inherent bias to the synthesized outliers during training. Recently, two-stage instance-level self-supervised learning (SSL) has shown potential in learning generic representations for IAL. However, we hypothesize that dense-level SSL is more compatible as IAL requires pixel-level prediction. In this paper, we bridge these gaps by proposing a two-stage, dense pretraining model tailored for the IAL task. More specifically, our model utilizes dual positive-pair selection criteria and dual feature scales to learn more effective representations. Through extensive experiments, we show that our learned representations achieve significantly better anomaly localization performance among two-stage models, while requiring almost half the convergence time. Moreover, our learned representations have better transferability to unseen datasets. Code is available at https://github.com/terrlo/DS2.*

## 1. Introduction

Image anomaly localization (IAL) has received much research interest in recent years owing to its wide application in industrial inspection and fault detection domains [2, 15]. The main objective of IAL is to discover regional anomalous patterns, but it poses unique challenges as regional anomalies are often tiny and subtle, and can take on different forms, rendering it impossible to build an exhaustive anomaly training set presenting all possible anomalous patterns [22]. Therefore, many early works formulated the task in an unsupervised setting [14, 19, 28], where the model is trained on normal images only. However, as an unsupervised setting usually trains a generative model, the learnt boundary between normal and abnormal embeddings is not discriminative enough [9].



(a) instance-level pretraining

(b) dense-level pretraining

Figure 1. An illustration of the difference between instance-level pretraining and dense-level pretraining. In instance-level pretraining, the visual features of an image are pooled in the spatial dimension before loss calculation, whereas in dense pretraining, the spatial information is preserved. We hypothesize that dense pretraining is more compatible with the IAL task as it requires pixel-level discrimination.

In the last few years, self-supervised learning (SSL) has shown to be effective in learning discriminative representations in tasks with no access to a large quantity of labeled data [5, 11, 32]. SSL achieves this goal by designing a pretraining task which exploits the intrinsic structures from the unlabeled data. The benefit of learning discriminative representations with SSL makes it a suitable method for the IAL task. Training an end-to-end SSL model for IAL is a prevalent choice [19, 24, 33]. A well-designed end-to-end model can achieve great localization performance on the target dataset as the feature extractor and the classifier are updated synchronously with the same training loss. Nonetheless, as its classifier is biased towards the target dataset, an end-to-end model needs to train a separate model for each new dataset. In addition, these end-to-end models often utilize outlier exposure, where they create anomalous training samples by first inspecting the common patterns of real anomalies in the test set, and then designing a simulation process that edits normal images into pseudo-anomalous ones that reflect the common anomaly patterns. However, the simulated anomaly distribution can be incongruous with real anomalies from new datasets, thereby further hampering their transferability.

Recently, two-stage SSL models show potential in learning generic representations for the IAL task. For example, Sohn *et al.* [27] designed a two-stage framework, where in the first stage, SSL is used to learn image representations, and in the second stage, a simple density estimator is fitted using the representations from the first stage, which functions as a one-class classifier. The framework proves to achieve good performance on the IAL task. Following their two-stage framework, Li *et al.* [18] introduced CutPaste, which relies on outlier exposure to simulate anomalies and then trains a classifier to distinguish normal images from pseudo-anomalous ones. However, the pretraining tasks in both works are instance-level-based, which concentrate on learning high-level semantic information from the entire image. As a result, these models are unable to provide adequate pixel-wise discriminative insight. To compensate for this, these models need to be trained for a long period to learn competent representations for the IAL task. We argue that IAL is fundamentally a dense prediction task because it requires pixel-wise anomaly prediction, so it interests us to formulate IAL as a dense pretraining problem. An illustration of the difference between instance-level and dense-level pretraining is shown in Figure 1.

To this end, we introduce a new two-stage, dense pretraining model without outlier exposure, called **D**ual **S**cale **D**ual **S**imilarty (DS2). Our model is motivated by a dense model PixPro [32], which was originally designed for object detection and semantic segmentation. We tailor it to learn representations for the IAL task. As the localized anomalies in an image are often tiny and subtle, the IAL task requires the model to heed nuances in the image by learning high-quality representations. Therefore, our DS2 first tightens the positive-pair selection criterion to reduce the amount of noisy positive pairs learnt by the model. Next, as this tightened positive-pair criterion can result in information loss during pretraining, we devise additional feature-wise top-1 similar pair as the auxiliary positive-pair criterion to compensate for the information loss. Finally, as localized anomalies can appear in various sizes, we tap into the inherent advantage of dense pretraining by adopting multi-scale features from the feature extractor to guide the model to learn multi-scale representations. Figure 2 depicts an overview of our dense pretraining model DS2.

We summarize the contributions of our work as follows:

- We propose a dense pretraining model DS2 and compare it with the SOTA instance pretraining models for the IAL task under the two-stage framework. We find that DS2 achieves better localization performance (at least 1.6% improvement in AUC compared to the SOTA instance pretraining baseline [27], and 3.9% improvement in AUC compared to CutPaste(3-way) [18] under the "one-for-all" setting) while substantially reduces the pretraining time.
- Compared with the SOTA self-supervised end-to-end models, our pretrained DS2 has stronger transferability to new datasets when given no fine-tuning. This indicates our model is a suitable choice as an off-the-shelf baseline model for rapid IAL task on new datasets.
- Our experiments demonstrate that DS2 has other merits such as stable performance, high robustness towards small training dataset size and small batch size.

## 2. Related Work

We first briefly review some previous works on unsupervised IAL task in Section 2.1, and then discuss two different approaches for self-supervised IAL task—end-to-end and two-stage—in Section 2.2.

### 2.1. Unsupervised Image Anomaly Localization

With the introduction of industrial defect datasets such as MVTec AD [2], the Image Anomaly Localization (IAL) task has attracted increasing research interest. Many early works resort to unsupervised setting for this task [3, 7, 8, 14, 17, 23, 26, 28, 30, 31, 34, 35].

Among them, one popular method is to build a reconstruction model with an auto-encoder (AE) [7, 14, 17, 26, 28, 31, 34, 35]. As the AE is trained to reconstruct normal images only, it is assumed that abnormal images will be poorly reconstructed during inference. However, this assumption does not always hold true as the downsampling rate in AE influences its reconstruction ability for both normal and abnormal patterns. Knowledge distillation is another common method [3, 8, 23] where a compact student network distills representations from a teacher network pretrained on some other vision task. During training, the student only learns representations of normal images from the teacher, so during inference, the student and the teacher are expected to produce significantly different representations for anomalous regions.

### 2.2. Self-supervised Image Anomaly Localization

**End-to-end Framework.** Applying self-supervised learning for image anomaly localization has established itself as a new research topic over recent years [18, 19, 24, 27, 33]. Among them, training an end-to-end self-supervised model with outlier exposure is a prevalent choice in the IAL research field [19, 24, 33], where the classifier is co-trained with the feature extractor and preserved for anomaly scoring during inference. The task is to learn which regions (pixels) in the image are contaminated by the simulated anomalies. Liznerski *et al.* [19] added confetti noise to the normal images and trained a hypersphere classifier to localize the anomalies. Zavrtanik *et al.* [33] turned to Perlin noise images [21] for anomaly simulation and trained an AE to discriminate normal and abnormal regions in an image. Schlüter et al [24] argued that discontinuity in the simulated anomalous images will cause overfitting issue. To circumvent it, they used Poisson image editing [20] to seamlessly blend an anomalous patch with a normal image such that the created anomalous image looks smooth.

**Two-stage Framework.** Different from end-to-end models, Sohn *et al.* [27] established the two-stage framework for self-supervised IAL task. In the first stage, a self-supervised pretraining task is conducted to train a model that produces visual representations for the input image; then in the second stage, the representations of normal patterns are fitted by a density estimator, which is then evaluated against each test image to determine its anomaly score. In their work, they applied instance-level pretraining tasks such as rotation prediction [10], vanilla contrastive learning [5], and their novelly proposed distribution-augmented contrastive learning [27]. Following Sohn *et al.* [27], Li *et al.* [18] came up with a new pretraining task named CutPaste, which utilizes the outlier exposure strategy, where it cuts out a portion of the image and pastes it back to some random location in the same image. The pretraining goal is to train a feature extractor and a classifier that effectively differentiate between normal and pseudo-anomalous images. After pretraining, the classifier is discarded and the representations from the feature extractor are fitted by a density estimator. Their pretraining task is categorized as instance-level as the input to the classifier is the average-pooled feature embedding of the input image.

Our proposed DS2 follows the two-stage framework. However, we argue that instance pretraining is sub-optimal for the dense-prediction IAL task, whereas dense pretraining can be more effective given the nature of the problem. Additionally, our DS2 does not rely on outlier exposure.

## 3. Dual Scale Dual Similarity (DS2)

In this section, we formally introduce our DS2, a two-stage transferable dense representation learning model for IAL. Given that PixPro [32] achieves good performance on semantic segmentation and object detection, we adopt it as the baseline pretraining model and tailor it to learn representations for the IAL task. Next, we explain how PixPro works, and then how we tailor it to our problem. Our final dense pretraining model DS2 is illustrated in Figure 2.

For an input image **x**, the model first applies two separate data augmentation processes $t, t' \sim \mathcal{T}$, where $\mathcal{T}$ is a series of data augmentation methods such as horizontal flip, Gaussian blurring, *etc.*, and obtains two views $t(\mathbf{x})$ and $t'(\mathbf{x})$. The first view $t(\mathbf{x})$ passes through the online branch, which consists of a backbone network $f$, a projection head network $g$, and a Pixel-to-Propagation Module (PPM). The second view $t'(\mathbf{x})$ passes through the momentum branch, consisting of a backbone network $f'$ and a projection head $g'$, of which the weights are updated by moving average [12]. The backbones $f$ and $f'$ are ResNet-18 [13] in our study, and the projection heads $g$ and $g'$ are two $1 \times 1$ convolutional layers with batch normalization and ReLU layer in-between. The PPM module is a self-attention module that adds spatial smoothness to the learnt representations.

Assuming the online branch's output feature map is $\Omega_1 \in \mathbb{R}^{h \times w \times d_\omega}$ and the momentum branch's is $\Omega_2 \in \mathbb{R}^{h \times w \times d_\omega}$, the PixPro loss is formulated as

$$\mathcal{L}_{PixPro} = - \sum_{y_i \in \Omega_1, z_j \in \Omega_2} \mathbb{1}_{[(i,j) \in p^+]} cos(y_i, z_j), \qquad (1)$$

where $y_i \in \mathbb{R}^{d_\omega}$ is a feature vector from $\Omega_1$ and $z_j \in \mathbb{R}^{d_\omega}$ is a feature vector from $\Omega_2$. The notation $\mathbb{1}_{[(i,j) \in p^+]}$ is evaluated to 1 if dense feature vectors $y_i$ and $z_j$ are a positive pair, and to 0, otherwise. The feature vectors $y_i$ and $z_j$ are considered as a positive pair if their geometric distance (Euclidean distance in the input image space) is smaller than the positive-pair distance threshold $\delta$.

As the PixPro model was not initially proposed for our IAL task, in Sections 3.1–3.3, we introduce our strategies to tailor it towards our task. We first choose highly selective positive pairs to reduce the amount of noisy information. Then, we adopt an additional feature-wise top-1 similar pair to complement this highly selective positive-pair criterion. Finally, we utilize multi-scale feature maps to learn representations of different scales.

### 3.1. Highly Selective Positive Pair

The PixPro model was initially proposed for downstream tasks such as semantic segmentation and object detection. In those tasks, a semantic region or an object usually spans a non-trivial proportion of an image, and the model does not need to differentiate finer discrepancies within a semantic region or an object, allowing positive pairs to be established from a long distance. A semantic segmentation example is illustrated in Figure 3 (left): most of the semantic regions have a non-trivial size, allowing a distant pair like regions *A* and *B* to be regarded as a positive pair. On the contrary, for IAL task, the model needs to differentiate different parts of an object to better localize anomalies. For example, Figure 3 (right) shows a screw with localized defect in the *head* (encircled by a red ellipse). The region *A* and *B* represent screw *thread* and screw *head*, respectively. In order to better localize the anomaly in screw *head*, the IAL model should learn separate normal representations for screw *thread* and

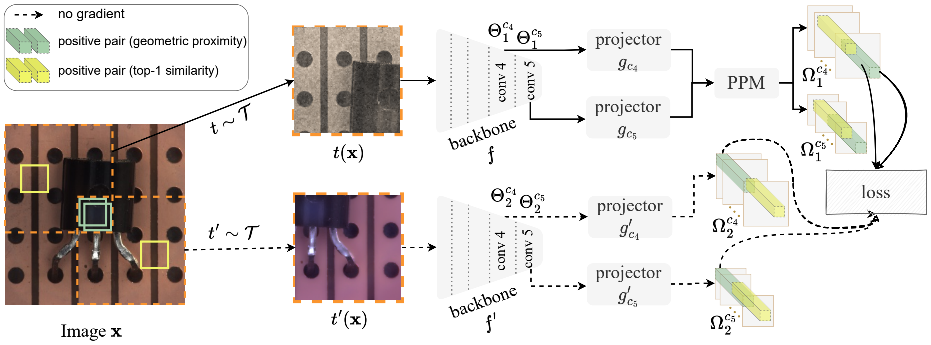


Figure 2. An overview of our dense pretraining model DS2. It contains an online branch (top) and a momentum branch (bottom). Two positive-pair selection criteria are adopted: the geometrically proximate pair (green pair) and the feature-wise similar pair (yellow pair). Also, our model utilizes two scales of feature maps, as indicated by $c_4$ and $c_5$ in the figure.

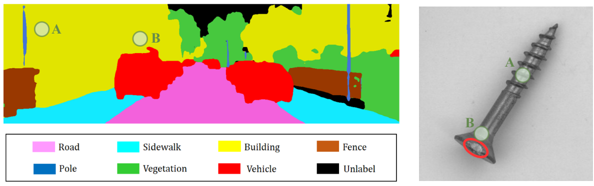


Figure 3. Semantic segmentation (left) [16] vs. anomaly localization (right).

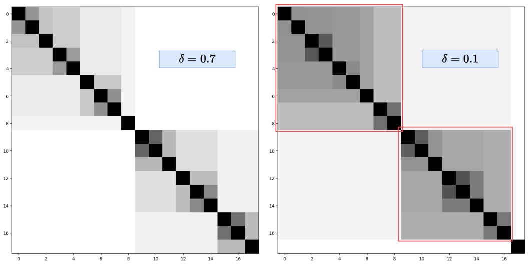


Figure 4. iVAT visualization of normal and faulty screw *head*'s embedding from backbone $f$ trained with $\delta = 0.7$ and $\delta = 0.1$. Darker square indicates higher cluster tendency.

screw *head*. Therefore, point $A$ and $B$ should not be considered as a positive pair in the IAL task. To validate our hypothesis, we train the model with different positive-pair distance thresholds ($\delta = 0.7$ and $0.1$), and then find one normal-screw image and one faulty-*head*-screw image. For both images, we crop nine patches around the screw *head* and embed them with the trained backbones. To see how separated the embeddings of normal and faulty screw *head*s are, we visualize them using iVAT [29], which reveals cluster tendency among data. Darker squares in iVAT indicate higher cluster tendency. As shown in Figure 4, with tighter threshold ($\delta = 0.1$), we can more clearly observe two clusters (highlighted by red squares), meaning the normal and faulty embeddings are better separated. Given these observations, we reckon our dense pretraining model needs to adopt highly selective positive pairs ($\delta = 0.1$) for the IAL task. In this way, we reduce the amount of noisy positive pairs fed to the pretraining model.

### 3.2. Additional Top-1 Similar Pair

One issue with using highly selective positive pairs is that it inhibits the formation of a positive pair when two regions in an image are geometrically distant but share similar semantic information. One example is illustrated in Figure 2: the two yellow-box regions contain similar patterns—part of a circuit board with a vertical groove in the middle. Nonetheless, they do not qualify for the highly selective positive pair criterion due to their large geometric distance. To address this problem, we propose to utilize the information from the feature maps of the backbone networks $f$ and $f'$. Specifically, we denote the feature map delivered by online backbone $f$ as $\Theta_1 \equiv f(t(\mathbf{x})) \in \mathbb{R}^{h \times w \times d_\theta}$ and the feature map by momentum backbone $f'$ as $\Theta_2 \equiv f'(t'(\mathbf{x})) \in \mathbb{R}^{h \times w \times d_\theta}$. A feature vector from $\Theta_1$ is denoted as $p_i \in \mathbb{R}^{d_\theta}$ and a feature vector from $\Theta_2$ is denoted as $p_j \in \mathbb{R}^{d_\theta}$. We hypothesize that if the two regions contain similar semantic information, the backbone networks would encode them to similar feature vectors. As such, we aim to find the vector pair $(p_i^*, p_j^*)$ that has the highest similarity among all vector pairs $\{p_i, p_j \mid i, j \in h \times w\}$ and treat this pair as the additional positive pair for the loss calculation. Mathematically, we extend the definition of positive pair to

$$p_{DS2}^{+} = \{(i,j) \mid \text{dist}(i,j) < 0.1 \text{ or } (i,j) := \arg\max_{i,j}(sim(p_i, p_j))\}. \quad (2)$$

### 3.3. Multi-scale Pretraining

Anomalous patterns in images come at different sizes, and some previous works [14, 23, 26] showed effectiveness of using multi-scale features to address this challenge.

**Algorithm 1** DS2 pretraining procedure (one iteration)

```
Inputs: input image $\mathbf{x}$; data augmentations $t$, $t' \sim \mathcal{T}$; backbone networks $f$, $f'$; projection heads for conv4 $g_{c_4}$, $g'_{c_4}$; projection heads for conv5 $g_{c_5}$, $g'_{c_5}$; pixle-to-propogation module $\mathcal{PPM}$.    ▷ apostrophe $'$ denotes momentum branch
Output: final training loss $\mathcal{L}_{DS2}$.
$\Theta_1^{c4}, \Theta_1^{c5} \leftarrow f(t(\mathbf{x}))$; $\Theta_2^{c4}, \Theta_2^{c5} \leftarrow f'(t'(\mathbf{x}))$
For each scale $s \in \{c_4, c_5\}$, create positive-pairs set $p^+_{DS2}$, which records the index pairs $(i,j)$ satisfying the condition $\{(i,j) \mid \text{dist}(i,j) < 0.1 \text{ or } (i,j) := \arg\max_{i,j}(sim(p_i,p_j)),\ \text{where } p_i \in \Theta_1^s, p_j \in \Theta_2^s\}$
$\Omega_1^{c4} \leftarrow \mathcal{PPM}(g_{c_4}(\Theta_1^{c4}))$; $\Omega_1^{c5} \leftarrow \mathcal{PPM}(g_{c_5}(\Theta_1^{c5}))$
$\Omega_2^{c4} \leftarrow g'_{c_4}(\Theta_2^{c4})$; $\Omega_2^{c5} \leftarrow g'_{c_5}(\Theta_2^{c5})$
$\mathcal{L}_{DS2} = -\frac{1}{2} \sum_{s \in \{c_4, c_5\}} \sum_{y_i \in \Omega_1^s, z_j \in \Omega_2^s} \mathbb{1}_{[(i,j) \in p^+_{DS2}]} cos(y_i, z_j)$
```

Compared with instance representation learning, dense representation learning is naturally compatible with leveraging multi-scale features. As such, we exploit the feature maps from the last two convolutional blocks (*i.e.,* conv4 and conv5) of the backbone network. The conv4's feature map has larger spatial resolution so that each feature vector captures fine-grained information, whereas conv5's feature map is smaller and each feature vector encodes higher-level information.

We attach separate projectors to process the feature maps from conv4 and conv5 as they have different spatial resolutions. In the online branch, once the two feature maps, $\Theta_1^{c4}$ and $\Theta_1^{c5}$, are processed by the projectors, they are further processed by the same PPM module and generate two output feature maps $\Omega_1^{c4}$ and $\Omega_1^{c5}$. The multi-scale pretraining extends the loss function Eq. (1) into

$$\mathcal{L}_{DS2} = -\frac{1}{2} \sum_{s \in \{c_4, c_5\}} \sum_{y_i \in \Omega_1^s, z_j \in \Omega_2^s} \mathbb{1}_{[(i,j) \in p^+_{DS2}]} cos(y_i, z_j), \tag{3}$$

where $s$ denotes the feature map scale and we use two scales in our model.

The pretraining procedure of our DS2 model in one iteration is summarized in Algorithm 1.

# 4. Experiment

To validate the performance of our dense pretraining model DS2, we first compare it with SOTA self-supervised IAL models that follow the two-stage framework. In this scenario, our baselines are: (1) the best-performing instance pretraining models reported in [27], including RotNet(MLP head), SimCLR [5], and DistAug; (2) CutPaste(3-way) [18], which uses outlier exposure for data augmentation. Next, we compare DS2 with end-to-end self-supervised IAL models to evaluate their transferability on new datasets given no fine-tuning or retraining. In this scenario, we adopt FCDD [19] and DRÆM [33] as our baselines.

We discuss the experiment setup in Section 4.1, compare DS2 with two-stage baselines in Section 4.2, and compare DS2 with end-to-end baselines in Section 4.3. Finally, we offer some deeper insights of the model through ablation studies in Section 4.4.

## 4.1. Experiment Setup

Here we provide a high-level description of the evaluation datasets and the evaluation protocols used in our experiments. More details on the implementation of DS2 and the usage of the evaluation datasets can be found in the supplementary material.

**Datasets.** When comparing DS2 with other two-stage models, we adopt the widely-used MVTec AD [2] as the benchmark dataset. We pretrain one model using all the training images regardless of their categories. When comparing transferability of DS2 with end-to-end models, we evaluate them on three datasets: MVTec LOCO [1], KSDD2 [4], and MTD [15]. All of them are proposed as image anomaly localization benchmarks with pixel-accurate ground-truth masks.

**Evaluation.** Following [18, 27], we use a simple generative classifier, which is a Gaussian Density Estimator (GDE) [25] in our case, to evaluate the learnt representations. We follow the baselines by resizing each image into 256 × 256 and cropping patches of 32 × 32 with a stride of 4, resulting in 57 × 57 patches per image, each with an anomaly score. For pixel-level localization, we upsample the patch scores into 256 × 256 with a Gaussian kernel. The localization performance is measured with Area Under the Receiver Operating Characteristic Curve (AUC) score, as this is the commonly used metric in the baselines.

## 4.2. Comparison with Two-stage Models

**Overall Quantitative Results.** Here we compare the quantitative results of our DS2 to the two-stage baselines. Following [18, 27], we run our DS2 model five times with different seeds and report the mean and standard deviation scores. We also include PixPro [32] as a baseline here for the purpose of validating our model adaptation method's efficacy. The results are reported in Table 1.

When we apply PixPro, with backbone architecture switched to ResNet-18 to match all the baseline models, its performance (88.9 AUC) is not optimal. By incorporating all the adaption methods introduced in Sections 3.1–3.3, our DS2 model achieves 94.6 AUC in the overall localization score. This demonstrates the efficacy of our adaptation strategies. Our DS2 also significantly outperforms all the instance pretraining baselines reported in [27]: SimCLR (85.6 AUC) and DistAug (90.4 AUC), and RotNet(MLP head) (93.0 AUC). This finding demonstrates that dense pretraining learns better-quality visual representations for the IAL task than instance pretraining does.

| Gran | OE | Model | object | texture | **all** |
|---|---|---|---|---|---|
| Ins. | ✗ | RotNet(MLP head) [27] | 96.4±0.4 | 86.3±2.0 | 93.0±0.9 |
| Ins. | ✗ | SimCLR [27] | 91.7±1.0 | 73.4±1.8 | 85.6±1.3 |
| Ins. | ✗ | DistAug [27] | 94.4±0.5 | 82.5±1.5 | 90.4±0.8 |
| Ins. | ✓ | CutPaste(3-way) (**category-wise**, reported in [18]) | 95.8±0.1 | **96.3**±0.1 | **96.0**±0.1 |
| Ins. | ✓ | CutPaste(3-way) (**one-for-all**, our re-implementation) | 93.5±0.4 | 85.1±0.7 | 90.7±0.5 |
| Dns. | ✗ | PixPro [32] | 92.5±0.3 | 81.7±0.4 | 88.9±0.2 |
| Dns. | ✗ | DS2(ours) (**one-for-all**) | **96.5**±0.1 | 90.8±0.4 | 94.6±0.1 |

Table 1. The localization performance of different two-stage pretraining models on MVTec AD. The best localization results are bold-faced. (**Gran**: Granularity; **Ins.**: Instance Pretraining; **Dns.**: Dense Pretraining; **OE**: Outlier Exposure)

Meanwhile, CutPaste(3-way) [18] outperforms our DS2 according to their reported results (96.0 AUC). However, we need to note that the CutPaste paper pretrains a separate model for each MVTec category ("category-wise"), while our method pretrains one holistic model for all categories ("one-for-all"). For a fair comparison, we re-implemented the CutPaste(3-way) following the exact design choices as reported in the original paper (refer to our supplementary material), and pretrain CutPaste(3-way) under the "one-for-all" setting with five random seeds, and report the mean and standard deviation scores. The high-level results are shown in Table 1, and the category-level results are in Table 2.

The results show that, under the same "one-for-all" pretraining setting, our DS2 outperforms CutPaste(3-way) in most of the categories. In terms of the overall performance, CutPaste(3-way) can only achieve 90.7 AUC, which is much lower than DS2 (94.6 AUC). Besides, the localization performance of CutPaste(3-way) has a large variance among different categories. Particularly, it performs poorly on the *grid* category (69.0 AUC). As CutPaste resorts to outlier exposure, this could indicate that its performance is impacted by the distribution gap between simulated and real anomalies. In contrast, our DS2 does not make assumptions of real anomalies, therefore achieving a balanced outcome among all the categories, with the majority of categories having localization scores above 90.0 AUC, and all the categories scoring above 80.0 AUC.

**Performance Reliability and Pretraining Time.** In terms of performance reliability, we find that DS2 offers steadier performance, as its performance's standard deviations are always smaller than those instance pretraining models from [27] and our re-implemented CutPaste(3-way).

| Category | CutPaste(3-way) (one-for-all) | DS2 (one-for-all) |
|---|---|---|
| carpet | **92.7**±1.9 | 92.3±1.1 |
| grid | 69.0±0.8 | **84.5**±0.8 |
| leather | 91.6±2.0 | **98.9**±0.1 |
| tile | 85.6±1.5 | **88.1**±0.7 |
| wood | 86.8±1.1 | **89.9**±0.2 |
| *texture* | 85.1±0.7 | **90.8**±0.4 |
| bottle | 96.6±0.2 | **97.6**±0.1 |
| cable | 88.0±2.8 | **96.4**±0.2 |
| capsule | 94.8±0.5 | **96.6**±0.2 |
| hazelnut | **97.7**±0.1 | 97.6±0.1 |
| metal_nut | 89.2±2.2 | **96.6**±0.2 |
| pill | **94.6**±0.9 | 92.2±0.3 |
| screw | 96.3±0.1 | **97.1**±0.1 |
| toothbrush | 92.4±0.6 | **97.1**±0.2 |
| transistor | 94.0±0.5 | **97.0**±0.1 |
| zipper | 91.3±0.9 | **96.6**±0.1 |
| *object* | 93.5±0.4 | **96.5**±0.1 |
| *overall* | 90.7±0.5 | **94.6**±0.1 |

Table 2. A category-level localization performance comparison between CutPaste(3-way) and our DS2 under the "one-for-all" setting. The best localization results are bold-faced.

Regarding pretraining time, although we pretrain DS2 for 400 epochs, its performance usually converges around epoch 200, whereas the baselines from [27] require 2048 epochs to pretrain. As for CutPaste(3-way), when using two A100-40GB GPUs, it takes around 32.5 hours to pretrain the "one-for-all" model, while our DS2 only takes about 9 hours under the same setting. Additionally, since CutPaste runs a fixed number of steps (256) per epoch instead of running through all the training images in each epoch, when pretrained in the "category-wise" setting, it requires on average the same amount of pretraining time as in "one-for-all" setting for each of the 15 MVTec categories, which is a huge demand on computational resources.

Given these findings, it can be observed DS2 model is the best option when performance reliability and computational resources are the primary concerns.

**Qualitative Results of DS2.** Here we provide some qualitative results of successful and failed localization cases by our DS2 model in Figure 5. As can be seen, DS2 is good at discerning irregular patterns in objects, such as the bent lead in the transistor and the broken teeth in the zipper, *etc*. It is also capable of discovering alien patterns in texture images, such as the thread in the grid. Nevertheless, the learnt representations are also susceptible to perturbations in the image. For example, the tile and pill images have many

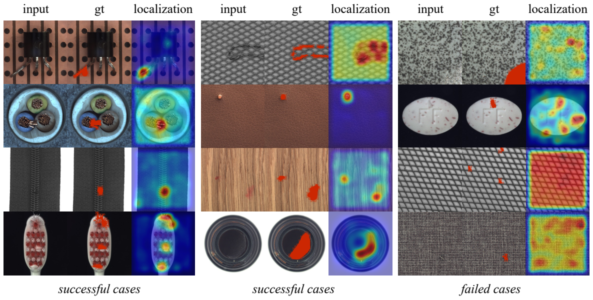


Figure 5. Localization visualization of DS2. The first two columns show successful cases and the last column shows failed cases of our model. The *gt* stands for ground truth, where anomalous parts are highlighted by a red mask.

random noises, and our model mistakes them as anomalies. Besides, it has difficulty discovering subtle deformation in texture images, such as the bent grid and the tiny x-shape metal in the carpet. More qualitative results can be found in the supplementary file.

### 4.3. Transferability of DS2 over End-to-end Models

To prove that our DS2 has stronger transferability to new datasets without any fine-tuning than end-to-end models do, we compare DS2 with the FCDD [19] and DRÆM [33], the localization scores of which on the MVTec dataset are 92.0 AUC and 97.3 AUC, respectively. We train them from scratch in the "one-for-all" setting, using their default design choices. We do not include NSA [24] in this study because NSA trains an individual model for each MVTec category with very different hyperparameter settings based on authors' visual inspection, so it is nearly impossible to train an "one-for-all" NSA model without compromising the fairness. We directly apply the trained DS2, FCDD and DRÆM on three new datasets (MVTec LOCO [1], KSDD2 [4], MTD [15]) without any fine-tuning. As the results in Table 3 show, DS2 outperforms FCDD and DRÆM on all the tested datasets in this scenario. Their performance gap is large on the MVTec LOCO dataset, which contains assorted defect types. On KSDD2 and MTD, where the anomaly types are limited, DS2 still has an edge over them. The visualization examples shown in Figure 6 demonstrate that, when coupled with a new classifier, the pretrained representations of DS2 are more capable of localizing anomalies on a new dataset.

One may question that the new classifier gives DS2 a boost as it is exposed to the normal training images of the new datasets whereas end-to-end models are not. To ascertain if the superior performance of DS2 is wholly owing to the new classifier, we adapt FCDD and DRÆM to the two-stage pipeline: we take the representation from an intermediate layer and feed it into the same classifier architecture used by DS2. For FCDD, it is the output before the last convolutional block and, for DRÆM, it is the bottleneck layer of its discriminative sub-network. The adapted models are named FCDD(two-stage) and DRÆM(two-stage), respectively, and their results are reported in Table 3. The two-stage design improves their performance on the MVTec LOCO dataset by a large degree: the FCDD(two-stage) improves by 27.3 AUC and the DRÆM(two-stage) improves by 19.6 AUC. This shows that the two-stage framework has an advantage over end-to-end models in terms of transferability to new datasets. Nonetheless, the adapted models do not enjoy similar improvement on the other two datasets—MTD and KSDD2. For these two datasets, only FCDD(two-stage) improves by 5.1 AUC on MTD and, in all other cases, their performance degrades. A common trait of MTD and KSDD2 is that they contain only grayscale images. As FCDD and DRÆM's anomaly simulation process involves color overlay, they assume real anomalies deviate from normality in color to some degree. This assumption works well with colorful images (*e.g.*, MVTec LOCO), but is less agreeable with grayscale images. We conjecture that, although their end-to-end co-trained classifier can salvage this unfit assumption, in the two-stage adaptation, the new classifier's performance hinges on how discriminative the intermediate feature representations are between normal and abnormal patterns. As such, due to the unfit presumption of real anomalies, their performance drops on these grayscale datasets. In contrast, our DS2 integrates no prior assumption of real anomalies during pretraining so its representations are more transferable to novel anomaly patterns.

Given the generalizability of DS2, we suggest that, in the case where training resources or training time is limited, DS2 can be used as an off-the-shelf baseline model for rapid anomaly localization tasks on new datasets.

| Dataset | Ctg. | FCDD | FCDD<br>(two-stage) | DRÆM | DRÆM<br>(two-stage) | DS2 |
|---|---|---|---|---|---|---|
| LOCO [1] | bb | 34.0 | 80.8 | 61.0 | **87.7** | 79.5 |
| | jb | 80.6 | **97.2** | 68.4 | 92.4 | 96.0 |
| | pu | 73.6 | **89.4** | 78.8 | 85.8 | 88.7 |
| | sb | 45.3 | **88.3** | 81.2 | 73.3 | 87.6 |
| | sc | 79.1 | 93.6 | 29.1 | 77.4 | **95.3** |
| | *overall* | 62.5 | **89.8** | 63.7 | 83.3 | 89.4 |
| KSDD2 [4] | | 76.2 | 68.6 | 78.0 | 38.1 | **81.3** |
| MTD [15] | | 52.3 | 57.4 | 54.9 | 52.8 | **60.7** |

Table 3. The transferability of DS2 and end-to-end models on new datasets without fine-tuning. The best results are bold-faced. (**Ctg.**: Category; **bb**: breakfast box; **jb**: juice bottle; **pu**: pushpins; **sb**: screw bag; **sc**: splicing connectors)

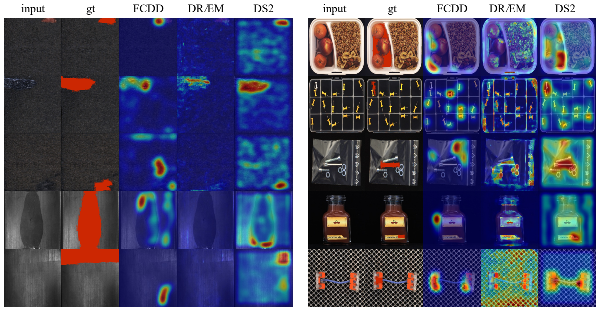


Figure 6. Localization visualization of FCDD, DRÆM and DS2 on new datasets without fine-tuning. The *gt* stands for ground truth, where anomalous parts are highlighted by a red mask.

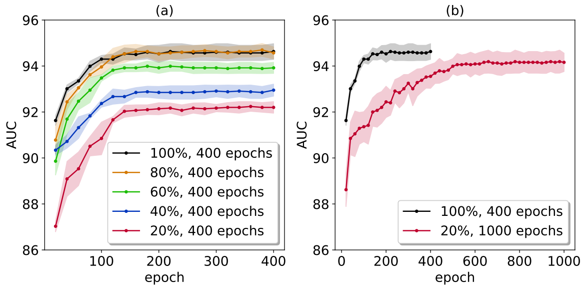


Figure 7. The localization performance of DS2 with various dataset sizes for pretraining.

### 4.4. Ablation Studies

We conduct various ablation studies to test DS2's robustness under different settings. All ablation studies are conducted with three different seeds, and we report the mean and standard deviation values. The detailed results for ablation studies (b)-(e) can be found in the supplementary file.

**(a) Robustness to Training Dataset Size.** We pretrain DS2 with different percentages of training images from each class of MVTec AD. In the first experiment, we pretrain the model for 400 epochs using 20%, 40%, 60%, 80%, and 100% of training images, respectively. The results are shown in Figure 7(a). We observe that the localization score decreases incrementally as we reduce the training dataset size. When we use only 20% of training images, the localization performance is around 92.2 AUC, which is 2.4 AUC lower than full-dataset pretraining (94.6 AUC), but still higher than three of our full-dataset-trained baselines: SimCLR (85.6 AUC), DistAug (90.4 AUC), and CutPaste$_{\text{(3-way, one-for-all)}}$ (90.7 AUC), and not far from RotNet$_{\text{(MLP head)}}$ (93.0 AUC). We then study whether longer pretraining (1,000 epochs) can compensate for the reduced size (20%) of training images, and the result in Figure 7(b) shows that the model can achieve 94.1 AUC with 1000-epoch pretraining, which narrows the performance gap with full-dataset 400-epoch pretraining to 0.5 AUC. This shows that our DS2 model is robust to small training datasets and is befitting in situations where the amount of training samples is limited.

**(b) Incorporating Instance Branch.** We are interested in testing if incorporating an instance branch would bring benefit to our DS2. To this end, we incorporate several instance pretraining models, including BYOL [11], MoCoV2 [6], and the two top-performing models in [27]: RotNet$_{\text{(MLP head)}}$ and DistAug. In terms of the loss coefficient, we use 0.5 for each branch, *i.e.*, loss $= 0.5 \times \text{loss}_{\text{(instance)}} + 0.5 \times \text{loss}_{\text{(dense)}}$. The results show that including an instance branch hampers DS2's performance, which demonstrates that DS2 already learns good representations for the IAL task without incorporating any instance pretraining.

**(c) Impact of Color Augmentation Choice.** With random resized crop and horizontal flip as the default geometric augmentation, we test with different combinations of color augmentation methods. The results show that color jitter is essential for good performance, and in general, using fewer color augmentation gives better result.

**(d) Impact of Batch Size.** We choose batch sizes from the set $\{32, 64, 128, 256, 512\}$. The results show that DS2's performance is stable across various batch sizes. When the batch size is as small as 32, its overall score (94.2 AUC) drops only by 0.4 AUC compared to the optimal value.

**(e) Impact of Feature Map Scales.** We examine if DS2 learns better representations with more scales of feature maps. To this end, we pretrain the model with ResNet's conv5's feature map only (c5), ResNet's conv4-5's feature maps (c4c5), and ResNet's conv3-5's feature maps (c3c4c5). The results show that using c4c5 feature maps gives the best result. The additional conv3 feature map does not bring extra performance gain. Therefore, we adopt the c4c5 in our DS2 model.

## 5. Conclusion

We present DS2, a two-stage dense representation learning model for the IAL task. Our studies demonstrate that its learned representations are compatible with the dense IAL task and show strong transferability to new datasets. When compared to instance pretraining, DS2 can learn better representations for IAL task with much faster convergence speed; when compared to end-to-end models, DS2 has stronger transferability towards new datasets without fine-tuning. Additionally, DS2 provides other benefits such as steadier performance and high robustness towards small training sets and small batch sizes. In future works, we aim to further improve the model's discriminative ability on subtle defects by incorporating new self-supervised tasks.

**Acknowledgments.** This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200.

## References

[1] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4):947–969, 2022. 5, 7

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, pages 9592–9600, 2019. 1, 2, 5

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *CVPR*, pages 4183–4192, 2020. 2

[4] Jakob Božič, Domen Tabernik, and Danijel Skočaj. Mixed supervision for surface-defect detection: From weakly to fully supervised learning. *Computers in Industry*, 129:103459, 2021. 5, 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 1, 3, 5

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 8

[7] David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. In *ICLR*, 2020. 2

[8] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *CVPR*, pages 9737–9746, 2022. 2

[9] Choubo Ding, Guansong Pang, and Chunhua Shen. Catching both gray and black swans: Open-set supervised anomaly detection. In *CVPR*, pages 7388–7398, 2022. 1

[10] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. *Advances in neural information processing systems*, 31, 2018. 3

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 1, 8

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[14] Jinlei Hou, Yingying Zhang, Qiaoyong Zhong, Di Xie, Shiliang Pu, and Hong Zhou. Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection. In *ICCV*, pages 8791–8800, 2021. 1, 2, 4

[15] Yibin Huang, Congying Qiu, and Kui Yuan. Surface defect saliency of magnetic tile. *The Visual Computer*, 36:85–96, 2020. 1, 5, 7

[16] Jongmin Jeong, Tae Sung Yoon, and Jin Bae Park. Towards a meaningful 3d map using a 3d lidar and a camera. *Sensors*, 18(8):2571, 2018. 4

[17] Gukyeong Kwon, Mohit Prabhushankar, Dogancan Temel, and Ghassan AlRegib. Backpropagated gradient representations for anomaly detection. In *ECCV*, pages 206–226. Springer, 2020. 2

[18] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *CVPR*, pages 9664–9674, 2021. 2, 3, 5, 6

[19] Philipp Liznerski, Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus-Robert Müller. Explainable deep one-class classification. In *ICLR*, 2021. 1, 2, 3, 5, 7

[20] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318, 2003. 3

[21] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985. 3

[22] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021. 1

[23] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *CVPR*, pages 14902–14912, 2021. 2, 4

[24] Hannah M Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In *ECCV*, pages 474–489. Springer, 2022. 1, 2, 3, 7

[25] Simon J Sheather. Density estimation. *Statistical science*, pages 588–597, 2004. 5

[26] Yong Shi, Jie Yang, and Zhiquan Qi. Unsupervised anomaly segmentation via deep feature reconstruction. *Neurocomputing*, 424:9–22, 2021. 2, 4

[27] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. In *ICLR*, 2021. 2, 3, 5, 6, 8

[28] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *ECCV*, pages 485–503. Springer, 2020. 1, 2

[29] Liang Wang, Uyen TV Nguyen, James C Bezdek, Christopher A Leckie, and Kotagiri Ramamohanarao. ivat and avat: enhanced visual analysis for cluster tendency assessment. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 16–27. Springer, 2010. 4

[30] Shenzhi Wang, Liwei Wu, Lei Cui, and Yujun Shen. Glancing at the patch: Anomaly localization with global and local feature comparison. In *CVPR*, pages 254–263, 2021. 2

[31] Jhih-Ciang Wu, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Learning unsupervised metaformer for anomaly detection. In *ICCV*, pages 4369–4378, October 2021. 2

[32] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, pages 16684–16693, 2021. 1, 2, 3, 5, 6

[33] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *ICCV*, pages 8330–8339, 2021. 1, 2, 3, 5, 7

[34] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112:107706, 2021. 2

[35] Kang Zhou, Yuting Xiao, Jianlong Yang, Jun Cheng, Wen Liu, Weixin Luo, Zaiwang Gu, Jiang Liu, and Shenghua Gao. Encoding structure-texture relation with p-net for anomaly detection in retinal images. In *ECCV*, pages 360–377. Springer, 2020. 2