

ReConPatch : Contrastive Patch Representation Learning for Industrial Anomaly Detection

Jeeho Hyun Sangyun Kim Giyoung Jeon Seung Hwan Kim
Kyunghoon Bae Byung Jun Kang*

LG AI Research

Abstract

Anomaly detection is crucial to the advanced identification of product defects such as incorrect parts, misaligned components, and damages in industrial manufacturing. Due to the rare observations and unknown types of defects, anomaly detection is considered to be challenging in machine learning. To overcome this difficulty, recent approaches utilize the common visual representations from natural image datasets and distill the relevant features. However, existing approaches still have the discrepancy between the pre-trained feature and the target data, or require the input augmentation which should be carefully designed particularly for the industrial dataset. In this paper, we introduce ReConPatch, which constructs discriminative features for anomaly detection by training a linear modulation attached to a pre-trained model. ReConPatch employs contrastive representation learning to collect and distribute features in a way that produces a target-oriented and easily separable representation. To address the absence of labeled pairs for the contrastive learning, we utilize two similarity measures, pairwise and contextual similarities, between data representations as a pseudo-label. Unlike previous work, ReConPatch achieves robust anomaly detection performance without extensive input augmentation. Our method achieves the state-of-the-art anomaly detection performance (99.72%) for the widely used and challenging MVTec AD dataset.

applications. These approaches aim to learn how to classify anomalies from normal cases based on previously collected data. However, anomaly detection is particularly challenging because defects are rarely observed and unknown types of defects can occur. This situation, in which the majority of cases are marked as normal and abnormal cases are scarce in the collected data, has led to the improvements in one-class classification.

The key concept of one-class classification for anomaly detection is to train a model to learn a distance metric between data and detect anomalies at a large distance from the nominal data. In an effort to learn the metric, reconstruction-based approaches have been proposed to detect anomalies by measuring the reconstruction errors using auto-encoding models [8, 20, 26] or generative adversarial networks (GANs) [21, 25]. As the variety of data is not sufficiently rich to estimate a reliable nominal distribution from scratch, recent works have shown that leveraging the common visual representation, obtained from a natural image dataset [10], can result in high anomaly detection performance [3, 7]. Although pre-trained models can provide rich representations without adaptation, such representations are not sufficiently distinguishable to identify subtle defects in industrial images. The distribution shift between natural and industrial images also makes it difficult to extract anomaly-specific features. For improvements in anomaly detection performance, training a model to learn a representation space that effectively discriminates borderline anomalies is essential.

To alleviate the distribution shift between the pre-trained and the industrial datasets, prominent features for anomaly detection can be distilled by training a student model to reproduce the representation of the pre-trained model using a teacher supervision [5]. Attaching a normalizing flow [11] at the end of the pre-trained model is another approach to exploit the pre-trained representation and estimate the distribution of normality [23]. Unfortunately, existing methods still require extensive handcrafted input augmentation, such as random crop, random rotation, or color jitter. Particu-

1. Introduction

Anomaly detection in industrial manufacturing is key to identify the defects in products and maintain their quality. Anomalies can include incorrect parts, misaligned components, or damage to the product. Machine learning approaches to anomaly detection have been widely studied owing to an increasing demand for automation in industrial

*Correspondence to: bj.kang@lgresearch.ai

larly in case of industrial images, data augmentation should be carefully designed by the user expertise.

In this paper, we introduce unsupervised metric learning framework for anomaly detection by enhancing the arrangement of the features, *ReConPatch*. Contrastive learning-based training schemes present weaknesses in terms of modeling variations within nominal instances, which may increase the false-positive rate of the anomaly detection. To this end, ReConPatch utilizes the contextual similarity [16] among features obtained from the model as a pseudo-label for the training. Specifically, our method efficiently adapts feature representation by training only a simple linear transformation, as opposed to training the entire network. By doing so, we are able to learn a target-oriented feature representation which achieves higher anomaly detection accuracy without input augmentation, making our method a practical and effective solution for anomaly detection in various industrial settings.

2. Related Work

Unsupervised machine learning approaches in anomaly detection using neural networks have been widely analyzed. Deep Support Vector Data Description (SVDD) trains a neural network to map each datum to the hyperspherical embedding and detect anomalies by measuring the distance from the center of the hypersphere [24]. Patch SVDD has been developed as a patch-wise extension of Deep SVDD, utilizing the features of each spatial patch from the convolutional neural network (CNN) feature map to enhance localization and enable fine-grained examination [30]. The reconstruction-based approach assumes that normal data can be accurately reconstructed or generated by training a model using a nominal dataset, whereas abnormal data cannot. Based on this assumption, an anomaly score is calculated as the error between the original input and the reconstructed input. Auto-encoding models are used for the reconstruction model [8, 20, 26]. With the improvements in GANs, several approaches have also shown the effectiveness of GANs in anomaly detection [21, 25]. When training a model from scratch, variety and abundance should be guaranteed, which is mostly not available for anomaly detection.

To alleviate the shortage of data in anomaly detection, attempts have been made to utilize a common visual representation trained with a rich natural image dataset [10]. Previous studies measure the distance between the representations of input data and their nearest neighbors to detect the anomalies [3] and compares hierarchical sub-image features to localize anomalies [7]. A memory bank is introduced in SPADE to efficiently store the representatives to be compared with [7].

DifferNet [23] provides a normalizing flow [11] that is helpful in training a bijective mapping between the pre-

trained feature distribution and the well-defined density of the nominal data, which is used to identify the anomalies. A condition normalizing flow using positional encoding is proposed by CFLOW-AD [12]. As the normalizing flow is trained to map features to the nominal distribution, this method is vulnerable to the outliers in the training dataset.

PatchCore proposes a locally aware patch feature and efficient greedy subsampling method to define the core-set [22]. The coupled-hypersphere-based feature adaptation (CFA) trains a patch descriptor that maps features onto the hypersphere, which is centered on the nearest neighbor in the memory bank [17]. PaDim estimates a Gaussian distribution of patch features at each spatial location to detect and localize out-of-distributions (OODs) as anomalies [9]. PNI is developed to train a neural network to predict the feature distribution of each spatial location and its neighborhoods [2].

3. Method

Our proposed method, ReConPatch, focuses on learning a representation space that maps patch features with similar nominal characteristics to be grouped closely in an unsupervised learning manner. Although previous work [22] has shown the effectiveness of selecting representative nominal patch features using a pre-trained model, this model still presents a representation biased to the natural image data, which has a gap with the target data. The main concept of our proposed approach is to train the target-oriented features that spread out the distributions of patch features according to the variations in normal samples, and gathers the similar features.

3.1. Overall structure

As shown in Fig. 1, our framework consists of the training and the inference phases. In the training phase, we first collect the feature map $\Phi(x) \in \mathbb{R}^{C \times H \times W}$ for each input x in the training data using the pre-trained CNN model $\Phi(\cdot)$ [12, 7, 9, 22, 17, 2]. Patch-level features $\mathcal{P}(x, h, w) \in \mathbb{R}^{C' \times 1}$ then generated by aggregating the feature vectors of the neighborhood within a specific patch size s in the same approach employed in PatchCore [22].

ReConPatch utilizes two networks to train representations of the patch-level features. One of these is a network for patch-level feature representation learning, which is trained using the relaxed contrastive loss \mathcal{L}_{RC} in Eq. 7. The representation network is composed of a feature representation layer f and the projection layer g respectively. When computing the \mathcal{L}_{RC} , pseudo-labels should be provided for every pair of features. The other network is used to calculate pairwise and contextual similarities between patch-level feature pairs. In addition, the similarity calcula-

¹ C and C' can be different according to the aggregation.

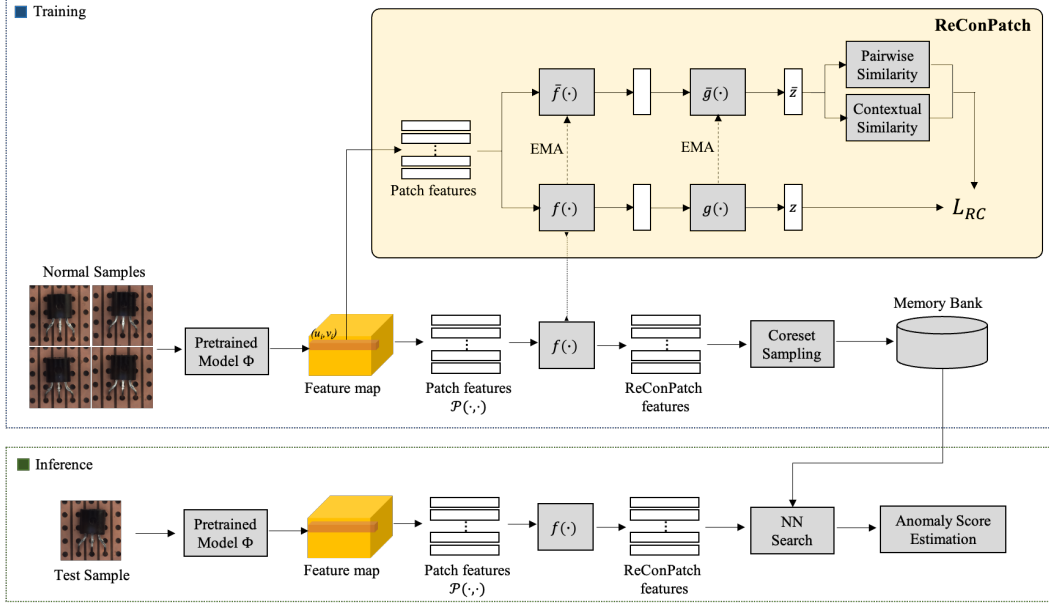


Figure 1: Overall structure of the anomaly detection using ReConPatch. ReConPatch consists of two networks to train representations of the patch-level features, which includes the feature representation layer f , \bar{f} and projection layer g , \bar{g} respectively. Upper networks (\bar{f}, \bar{g}) are used to calculate pairwise and contextual similarities between patch-level feature pairs, while the bottom networks (f, g) used for the representation learning of patch-level features is trained through relaxed contrastive loss \mathcal{L}_{RC} .

tion network is gradually updated by an exponential moving average (EMA) of the representation network. To distinguish the two networks, the layers in the latter network is denoted as \bar{f} and \bar{g} respectively.

After training the representation, the patch-level features extracted from the pre-trained CNN are transformed into target-oriented features using the feature representation layer f [6]. The representative features are selected using the coreset subsampling approach based on the greedy approximation algorithm [28] and stored in a memory bank. In the inference phase, the features of a test sample are extracted using the same process as training, and the anomaly score is calculated by comparing the features with the normal representative in the memory bank.

3.2. Patch-level feature representation learning

The objective of ReConPatch is to learn target-oriented features from patch-level features, thereby enabling more effective discrimination between normal and abnormal features. To accomplish this goal, a patch-level features representation learning approach is applied to aggregate highly similar features while repelling those with low similarity. As a lack of labeled data is a common challenge in anomaly detection datasets, we employ pseudo-labels to indicate the degree of proximity between patch-level features. To address this issue, the similarity between patch-level features

using the pairwise similarity and the contextual similarities is measured to be used as pseudo-labels.

For two arbitrary patch-level features p_i and p_j obtained by $\mathcal{P}(x, h, w)$, let the projected representation be $\bar{z}_i = \bar{g}(f(p_i))$ and $\bar{z}_j = \bar{g}(f(p_j))$. The pairwise similarity between two features, $\omega_{ij}^{Pairwise}$, is then provided by

$$\omega_{ij}^{Pairwise} = e^{-\|\bar{z}_i - \bar{z}_j\|_2^2 / \sigma} \quad (1)$$

where σ is the bandwidth of the Gaussian kernel, which can be adjusted to tune the degree of smoothing in the similarity measure [15, 16]. We note that Eq. 1 is used to measure the Gaussian kernel similarity between p_i and p_j , which is widely used to measure anomaly scores. However, the pairwise similarity is insufficient to consider the relationships among groups of features. As depicted in Fig. 2, for example, cases (a) and (b) have the same pairwise similarity. In (a) case, \bar{z}_i and \bar{z}_j belong to different groups of features; therefore, they should be separated. By contrast, in (b), they belong to the same group and should be gathered.

This leads to the simultaneous measure of contextual similarity, which consider the neighborhood of an embedding vector. Let k -nearest neighborhood of the feature index i is given as a set of indices, $\mathcal{N}_k(i) = \{j | d_{ij} \leq d_{il}\}$ where l is k -th nearest neighbor and d_{ij} denotes the Euclidean distance between the two embedding vectors ($d_{ij} = \|\bar{z}_i - \bar{z}_j\|_2^2$).

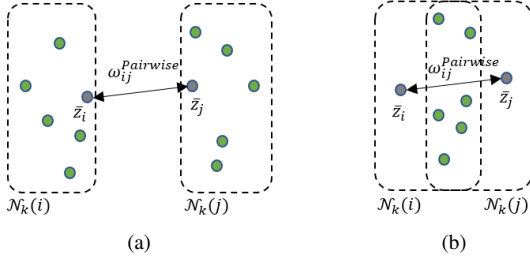


Figure 2: Illustrative examples of similarity measures in the representation space. The pairwise similarity $\omega_{ij}^{Pairwise}$ between \bar{z}_i and \bar{z}_j is identical in both (a) and (b). In (a), the k -nearest neighbors $\mathcal{N}_k(i)$ and $\mathcal{N}_k(j)$ do not enclose each other. Therefore, $\omega_{ij}^{Contextual}$ has a lower value, and the \bar{z}_i and \bar{z}_j pair should become apart. By contrast, as $\mathcal{N}_k(i)$ and $\mathcal{N}_k(j)$ enclose each other in (b) case, $\omega_{ij}^{Contextual}$ takes a higher value, so that \bar{z}_i and \bar{z}_j pair should attract each other.

Two patch-level features can be regarded as contextually similar if they share more nearest neighbors in common [18]. The contextual similarity $\tilde{\omega}_{ij}^{Context}$ between two patch-level features p_i and p_j is then defined as

$$\tilde{\omega}_{ij}^{Contextual} = \begin{cases} \frac{|\mathcal{N}_k(i) \cap \mathcal{N}_k(j)|}{|\mathcal{N}_k(i)|}, & \text{if } j \in \mathcal{N}_k(i) \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

In addition, the approach developed in this study adopts the idea of query expansion, which is widely used to improve the information retrieval, by expanding the query to the neighbors of neighbors [16, 18]. $\tilde{\omega}_{ij}^{Context}$ is redefined by averaging the similarities over the set of k -nearest reciprocal neighbors.

$$\mathcal{R}_k(i) = \{j | j \in \mathcal{N}_k(i) \text{ and } i \in \mathcal{N}_k(j)\} \quad (3)$$

$$\hat{\omega}_{ij}^{Contextual} = \frac{1}{|\mathcal{R}_{\frac{k}{2}}(i)|} \sum_{l \in \mathcal{R}_{\frac{k}{2}}(i)} \tilde{\omega}_{lj}^{Contextual}. \quad (4)$$

Because $\hat{\omega}_{ij}^{Contextual}$ is asymmetric, the contextual similarity is finally defined as the average bi-directional similarity of a pair, which is given by

$$\omega_{ij}^{Contextual} = \frac{1}{2} (\hat{\omega}_{ij}^{Contextual} + \hat{\omega}_{ji}^{Contextual}). \quad (5)$$

The final similarity between two patch-level features p_i and p_j is then defined as a linear combination of two similarities with a quantity $\alpha \in [0, 1]$,

$$\omega_{ij} = \alpha \cdot \omega_{ij}^{Pairwise} + (1 - \alpha) \cdot \omega_{ij}^{Contextual}. \quad (6)$$

Patch-level features do not have explicit labels because each patch image is correlated with neighboring patches.

Moreover, the goal is to obtain unique target-oriented features rather than clearly distinguishing them. Thus, relaxed contrastive loss [15] was adopted, in which inter-feature similarity is considered as pseudo-labels. Let $\delta_{ij} = \|z_i - z_j\|_2 / (\frac{1}{N} \sum_{n=1}^N \|z_i - z_n\|_2)$ denote the relative distance between embedding vectors in a mini-batch. The relaxed contrastive loss is given by

$$\mathcal{L}_{RC}(z) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \omega_{ij} (\delta_{ij})^2 + \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N (1 - \omega_{ij}) \max(m - \delta_{ij}, 0)^2 \quad (7)$$

where z is the embedding vectors inferred by $g(f(p))$, N is the number of instances in a mini-batch, and m is the repelling margin. ω_{ij} in Eq. 7 determines the weights of the attracting and repelling loss terms.

While representation learning networks f and g are trained with relaxed contrastive loss, the similarity calculation network \tilde{f} and \tilde{g} are slowly updated with an the EMA of the parameters in f and g respectively. Fast training of the similarity calculation network reduces the consistency of the relationships between the patch-level features, leading to unstable training. Let $\theta_{\tilde{f}, \tilde{g}}$ be the parameters of the similarity calculation network and $\theta_{f, g}$ be the parameters of the representation learning network. $\theta_{\tilde{f}, \tilde{g}}$ is then updated by

$$\theta_{\tilde{f}, \tilde{g}} \leftarrow \gamma \cdot \theta_{\tilde{f}, \tilde{g}} + (1 - \gamma) \cdot \theta_{f, g} \quad (8)$$

where γ is the hyper-parameter that adjusts the rate of momentum update.

3.3. Anomaly detection with ReConPatch

Anomaly scores are calculated in the same manner as in the case of PatchCore [22]. After training, the coreset is subsampled from the newly trained feature representation $f(\cdot)$ using the greedy approximation algorithm [28] and stored in memory bank \mathcal{M} . The coreset takes a role of the representative feature, which is used to compute the anomaly score. The pixel-wise anomaly score is then obtained by calculating the distance between the feature generated the feature representation layer $f(p_t)$ and its nearest coreset r^* within the memory bank.

$$s'_t = \min_{r^* \in \mathcal{M}} \mathcal{D}(f(p_t), r^*) \quad (9)$$

$$s_t = \left(1 - \frac{e^{s'_t}}{\sum_{r' \in \mathcal{N}_b(r^*)} e^{\mathcal{D}(f(p_t), r')}} \right) s'_t \quad (10)$$

where $r^* \in \mathcal{M}$ is the nearest neighbor coreset and $\mathcal{N}_b(r^*)$ is the b -nearest neighbor coresets of r^* in the memory bank. In addition, the image-wise anomaly score is computed as

the maximum score over the anomaly scores calculated for every patch feature in the image.

The accuracy of anomaly detection can be further improved by score-level fusion from multiple models. Because each model has a different distribution of scores, score normalization is necessary to evenly fuse the score levels of each model. The anomaly score is normalized to the modified z-score [1], defined as

$$\bar{s}_t = \frac{s_t - \tilde{s}}{\beta \cdot MAD}, \quad (11)$$

where \tilde{s} and MAD are the median value of the anomaly scores and the absolute deviation over the entire dataset for training, and β is a constant which is set to 1.4826 in our method.

4. Experiments

4.1. Experimental setup

Dataset In this study, we used the MVTec AD [4] dataset for our experiments. This dataset is widely used as an industrial anomaly detection benchmark. The dataset consists of 15 categories, with 3,629 training images and 1,725 test images. The training dataset includes only normal images, whereas the test dataset includes both normal and anomalous images. Each category in the test dataset has labels for normal and abnormal images, and anomaly ground truth mask labels for segmentation evaluation. For the single-model performance comparison, we performed the same pre-processing as described in previous work [7, 9, 17, 22]. Specifically, we resized each image to 256×256 and then center-cropped to 224×224 . For the ensemble model, the same pre-processing was used as in [22], each image was resized to 366×366 and then center-cropped to 320×320 . In addition, to compare with PNI [2], we resized each image to 512×512 and then center-cropped to 480×480 . No data augmentation was applied to any category.

Metrics To evaluate the performance of our proposed model, anomaly detection and segmentation performance was compared using the area under the receiver operation characteristic (AUROC) curve metric, following [7, 9, 17, 22]. For detection performance evaluation, we measure the image-level AUROC by using the model output anomaly score and the normal/abnormal labels of the test dataset. For segmentation, we measure the pixel-level AUROC using the anomaly scores obtained from the model output for all pixels and the anomaly ground truth mask labels.

Implementation details. For the single model, ImageNet pre-trained WideResNet-50 [31] was employed as the feature extractor. The f layer output size was set to 512, and the coreset subsampling percentage was set to 1%. Our proposed ReConPatch was trained for 120 epochs per each

Method	Ours-25%	Ours-10%	Ours-1%
Detection	99.24	99.27	99.49
Segmentation	98.01	98.07	98.07

Table 1: Results of the ablation study on coreset subsampling percentage using our proposed ReConPatch model with a WideResNet-50 backbone on the MVTec AD dataset.

Dimension	1024	512	256	128	64
Detection	99.49	99.56	99.53	99.52	99.14
Segmentation	98.07	98.07	98.03	97.94	97.68

Table 2: Ablation study results for the f layer dimension on the MVTec AD dataset using our proposed ReConPatch model with a WideResNet-50 backbone.

Metric	Detection	Segmentation
WRN-50, $s = 5$, 512 dim, layer (2+3), Imagesize 224		
AUROC	98.84	97.82
WRN-50, $s = 5$, 512 dim, layer (1+2+3), Imagesize 224		
AUROC	98.7	98.18

Table 3: Ablation study results obtained using our proposed ReConPatch model with more hierarchy levels, larger patch size on the MVTec AD dataset.

category. Without specific instructions, hierarchy levels² 2 and 3 were used with a patch size of $s = 3$ to generate the patch-level features. Particularly for the segmentation evaluation in Table 5, hierarchy levels 1, 2, and 3 were used with a patch size of $s = 5$, which is identified as the best performance through the ablation. In addition, for the comparison with PNI [2] using WideResNet-101, hierarchy levels 2 and 3 were used with a patch size of $s = 5$.

For the ensemble model, ImageNet pre-trained WideResNet-101 [31], ResNext-101 [29], and DenseNet-201 [14] are used as feature extractors for comparison with the PatchCore [22]. The f layer output size was set to 384, and we applied a coreset subsampling with percentage of 1% to all models in the ensemble. We trained ReConPatch for 60 epochs for each category. Hierarchy levels 2 and 3 were used for feature extraction in each model, and a patch size of $s = 3$ was applied to generate the patch-level features. Furthermore, to compare with PNI [2] using 480×480 image size, different parameters were applied. The f layer output size was set to 512, and a patch size of $s = 5$ was used. In this case, we trained each category for 120 epochs. ReConPatch was trained using AdamP [13] optimizer with a cosine annealing [19] scheduler. The learning rate was set to $1e-5$ for a single model and $1e-6$ for

²Hierarchy levels denote residual blocks in WideResNet architecture, which is same in [22].

the ensemble model, with a weight decay of $1e-2$. In the models using a 480×480 image size, the learning rate was specifically set to $1e-6$.

4.2. Ablation study

In this study, we aim to investigate the optimal configuration of ReConPatch through ablation studies. The first ablation was performed to determine the optimal coreset subsampling percentage. To this end, we compared anomaly detection and segmentation AUROC metrics using three subsampling percentages: 25%, 10%, and 1%, which were the same percentages used in PatchCore [22]. The pre-trained WideResNet-50 [31] backbone was used as the baseline for this experiment and the output dimension of the f layer is set to 1024. The results are presented in Table 1. We observe that the subsampling percentage of 1% provides the best performance. In addition, experiments were performed for various output dimension of the f layer (1024, 512, 256, 128, and 64), to determine the optimal dimension. The experiments were conducted with coreset subsampling set to 1%. The results are presented in Table 2, indicating that the highest performance was achieved with the dimension of 512, beyond which, the performance decreased after the dimension of 512.

Table 3 shows the results of an ablation study using more hierarchy levels, larger patch size on the MVTec AD [4] dataset with our proposed ReConPatch model. This study aimed to improve segmentation performance by utilizing more diverse information on the patch features. The results showed a decrease in detection performance, although the segmentation performance increased up to 98.18% when the patch size was increased to 5 and hierarchy levels 1, 2, and 3 are used.

4.3. Quantitative results

In this subsection, we evaluate the anomaly detection performance of our proposed method by comparing it with previous works that used the same pre-trained model and image size [7, 9, 17, 22]. We also include the performance of concurrent methods PNI [2] and CFLOW-AD [12] in Tables 4 and 5. In case of PNI [2], a WideResNet-101 model with an image size of 480×480 was used. To improve its performance, a refinement network was included, which was trained in a supervised manner using artificially created defect dataset. For CFLOW-AD [12], a WideResNet-50 model with an image size of 256×256 is used. The evaluation results used in CFLOW-AD were the best performances obtained for each category when using the image size of 256×256 .

The performance of the ReConPatch in Tables 4 and 5 was obtained using 1% coreset subsampling and f layer dimensions of 512, which is determined according to Table 2. Table 4 compares the anomaly detection performance

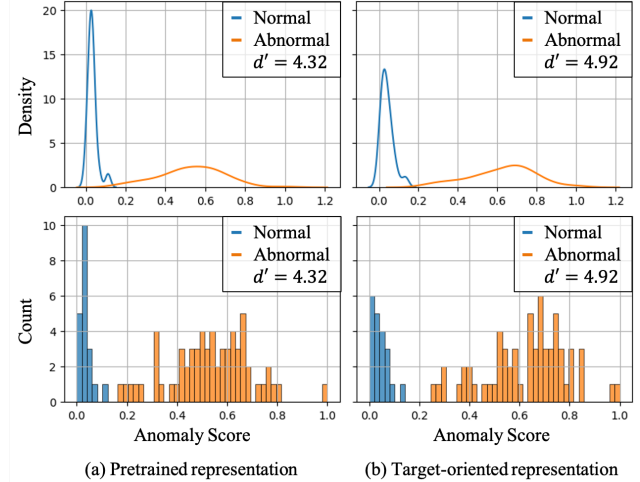


Figure 3: The histogram displays the normal and abnormal distributions of the anomaly score for the bottle class in the MVTec AD [4] dataset. The left histogram represents the patch feature results, and the right histogram represents the f layer output, which is the ReConPatch feature.

of a single model for each category of the MVTec AD [4] dataset, evaluated with image-level AUROC. Our proposed ReConPatch achieved an image-level AUROC of 99.56%, which outperformed CFA [17] (at 99.3%). Furthermore, ReConPatch provided higher performance than the state-of-the-art PNI [2] with WideResNet-101 [31], which achieved the performance of 99.62%.

Table 5 presents the anomaly segmentation performances evaluated using pixel-level AUROC. Our proposed approach focused on improving the anomaly detection performance; as a result, the segmentation performance may not be as high as its detection performance. However, we achieved a higher performance of 98.18% compared to PatchCore [22], indicating that the addition of ReConPatch feature in the f layer contributed to the improved segmentation performance.

Table 6 presents the performance of our ensemble model, which was evaluated using the modified z-score in Eq. 11 for each output from WideResNet-101 [31], ResNext-101 [29], and DenseNet-201 [14] models. Our model achieved state-of-the-art performance in anomaly detection task with AUROC of 99.72% on the MVTec AD dataset using an image size of 480×480 . We note that our model still outperforms the PNI [2] using a smaller image size of 320×320 , achieving an AUROC of 99.67% compared to AUROC of 99.63%. Furthermore, we outperformed PatchCore [22] in terms of anomaly segmentation performance, with an improved performance of 98.36% AUROC.

Backbone	WRN-101		WRN-50					
Image size	480×480	480×480	256×256	224×224	224×224	224×224	224×224	224×224
↓ Class\Method →	PNI [2] (w/ refine)	Ours	CFLOW-AD [12]	SPADE [7]	PaDiM [9]	PatchCore [22]	CFA [17]	Ours
Bottle	100	100	100	-	-	100	100	100
Cable	99.76	99.66	97.59	-	-	99.5	99.8	99.83
Capsule	99.72	99.76	97.68	-	-	98.1	97.3	98.8
Hazelnut	100	100	99.98	-	-	100	100	100
Metal nut	100	100	99.26	-	-	100	100	100
Pill	96.89	96.21	96.82	-	-	96.6	97.9	97.49
Screw	99.51	99.84	91.89	-	-	98.1	97.3	98.52
Toothbrush	99.72	100	99.65	-	-	100	100	100
Transistor	100	100	95.21	-	-	100	100	100
Object classes	99.51	99.5	97.56	-	-	99.14	99.14	99.4
Carpet	100	100	98.73	-	-	98.7	97.3	99.6
Grid	98.41	99.5	99.6	-	-	98.2	99.2	100
Leather	100	100	100	-	-	100	100	100
Tile	100	100	99.88	-	-	98.7	99.4	99.78
Wood	99.56	99.47	99.12	-	-	99.2	99.7	99.65
Zipper	99.87	99.89	98.48	-	-	99.4	99.6	99.76
Texture classes	99.64	99.81	99.3	-	-	99.03	99.2	99.8
Average	99.56	99.62	98.26	85.5	95.3	99.1	99.3	99.56

Table 4: Anomaly detection performance (Image-level AUROC) of the MVTec AD [4] dataset.

Backbone	WRN-101		WRN-50					
Image size	480×480	480×480	256×256	224×224	224×224	224×224	224×224	224×224
↓ Class\Method →	PNI [2] (w/ refine)	Ours	CFLOW-AD [12]	SPADE [7]	PaDiM [9]	PatchCore [22]	CFA [17]	Ours
Bottle	98.87	98.78	98.76	98.4	98.3	98.6	-	98.2
Cable	99.1	98.86	97.64	97.2	96.7	98.4	-	99.3
Capsule	99.34	99.24	98.98	99	98.5	98.8	-	97.61
Hazelnut	99.37	99.07	98.82	99.1	98.2	98.7	-	98.94
Metal nut	99.29	99.29	98.56	98.1	97.2	98.4	-	95.76
Pill	99.03	98.66	98.95	96.5	95.7	97.4	-	95.35
Screw	99.6	99.59	98.1	98.9	98.5	99.4	-	98.79
Toothbrush	99.09	99.16	98.56	97.9	98.8	98.7	-	98.88
Transistor	98.04	96.18	93.28	94.1	97.5	96.3	-	99.65
Object classes	99.08	98.76	97.96	97.69	97.71	98.3	-	98.05
Carpet	99.4	99.29	99.23	97.5	99.1	99	-	98.75
Grid	99.2	98.73	96.89	93.7	97.3	98.7	-	99.04
Leather	99.56	99.48	99.61	97.6	99.2	99.3	-	96.02
Tile	98.4	97.15	97.71	87.4	94.1	95.6	-	98.92
Wood	97.04	95.16	94.49	88.5	94.9	95	-	98.9
Zipper	99.43	99.25	98.41	96.5	98.5	98.8	-	98.56
Texture classes	98.84	98.18	97.72	93.53	97.18	97.73	-	98.37
Average	98.98	98.53	97.87	96	97.5	98.1	98.2	98.18

Table 5: Anomaly segmentation performance (Pixel-level AUROC) of the MVTec AD [4] dataset.

4.4. Qualitative analysis

In Figure 3, we compared the distribution of anomaly scores obtained using patch features and the proposed ReConPatch features for anomaly detection, with the histograms for image-level anomaly scores. We examined the density and count distributions of the scores for the bottle class in the MVTec AD [4] dataset. As a result, we ob-

served that the score distribution of the ReConPatch feature was compressed for normal data and shifted further away from the normal distribution for abnormal data, compared to the score distribution of the patch feature. We also computed the discriminability index d' [27] between normal and abnormal data, which is used to measure how separable two distributions are. We verify that d' increased in the proposed

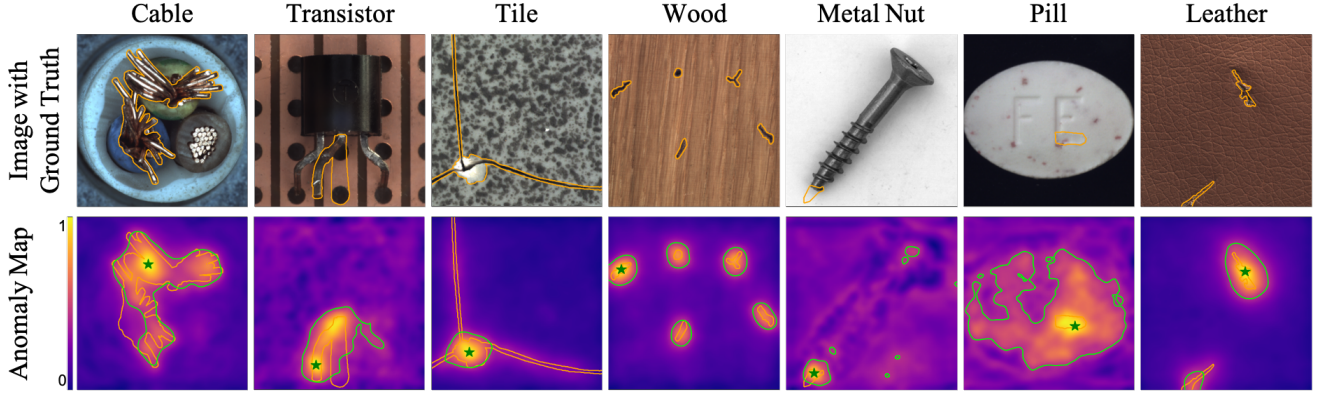


Figure 4: Examples of images with anomalies (top) and measured anomaly score maps (bottom) on MVTec AD dataset. The orange line depicts the ground truth of the anomalies and the green line depicts thresholds optimizing F1 scores of anomaly segmentation. The green star indicates the maximal location of the anomaly score in the heatmap.

Ensemble Backbone	WRN-101 & RNext-101 & DenseN-201			
Image size	480×480	480×480	320×320	320×320
Method	PNI [2] (w/ refine)	Ours	PatchCore [22]	Ours
Detection	99.63	99.72	99.6	99.67
Segmentation	99.06	98.67	98.2	98.36

Table 6: Comparison of ensemble model anomaly detection (Image-level AUROC) and segmentation (Pixel-level AUROC) performance.

representation.

$$d' = \frac{|\mu_{abnormal} - \mu_{normal}|}{\sqrt{(\sigma_{abnormal}^2 + \sigma_{normal}^2)/2}}. \quad (12)$$

In this experiment, the patch features were the same as the locally aware patch features proposed in PatchCore [22]. ReConPatch enables to obtain target-oriented features by training patch-level feature representations using the method proposed in section 3.2, which allows for an enhanced discrimination between normal and abnormal features. In Table 4, an image-level AUROC of 99.56% was achieved using ReConPatch, which is higher than the anomaly detection performance of PatchCore, which is 99.1%

We also provide the examples of the anomaly score map along with ground truth (orange line) overlaid input images in Fig. 4. The anomaly map indicates the regions of the input image where our algorithm has detected anomalies, with higher values indicating a higher likelihood of an anomaly being present. The green line indicates the threshold which is optimized by the F1 scores of anomaly segmentation. We curated 4 superior classes (cable, transistor, tile and wood) with higher performances and 3 inferior classes (metal nut, pill, and leather) for the analysis.

Despite the complexity of the ground truth in some cases, our method shows is able to correctly identify the locations of the anomalies. Although the anomaly maps for the inferior classes may show somewhat noisy results, the location of the maximal anomaly score, denoted by a green star, consistently aligns with the ground truth anomalies. This alignment supports the high performance of our method in anomaly detection.

5. Conclusion

In this paper, we introduce the ReConPatch to learn a target-oriented representation space, which can effectively distinguish the anomalies from the normal dataset. ReConPatch effectively trains the representation by applying the metric learning with softly guided by the similarity over the nominal features. When measuring the similarity, ReConPatch takes advantage of two different similarity measures, the pairwise and the contextual similarity. We note that two similarity measure are well-aligned with the anomaly detection literature, in which the Gaussian kernel distance and the nearest neighbor are widely used. Brining together all the contributions, ReConPatch shows the state-of-the-art performance on the MVTec anomaly detection dataset. We believe that ReConPath would contribute to the improvements in anomaly detection since it does not require extensive data augmentation and enables dimension reduction without significant loss of performance. This can lead to more efficient and accurate anomaly detection, as well as potentially opening up new avenues for research in this area.

References

- [1] Vaibhav Aggarwal, Vaibhav Gupta, Prayag Singh, Kiran Sharma, and Neetu Sharma. Detection of spatial outlier by using improved z-score test. pages 788–790, 2019. [5](#)
- [2] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. Image anomaly detection and localization with position and neighborhood information. *arXiv preprint arXiv:2211.12634*, 2022. [2](#), [5](#), [6](#), [7](#), [8](#)
- [3] Liron Bergman, Niv Cohen, and Yedid Hoshen. Deep nearest neighbor anomaly detection. *arXiv preprint arXiv:2002.10445*, 2020. [1](#), [2](#)
- [4] 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. [5](#), [6](#), [7](#)
- [5] 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](#)
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. [3](#)
- [7] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. [1](#), [2](#), [5](#), [6](#), [7](#)
- [8] Diana Davletshina, Valentyn Melnychuk, Viet Tran, Hitansh Singla, Max Berrendorf, Evgeniy Faerman, Michael Fromm, and Matthias Schubert. Unsupervised anomaly detection for x-ray images. *arXiv preprint arXiv:2001.10883*, 2020. [1](#), [2](#)
- [9] Thomas Defard, Aleksandr Setkov, Angélique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV*, pages 475–489. Springer, 2021. [2](#), [5](#), [6](#), [7](#)
- [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](#), [2](#)
- [11] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *5th International Conference on Learning Representations*, 2017. [1](#), [2](#)
- [12] 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. [2](#), [6](#), [7](#)
- [13] Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. AdamP: Slowing down the slowdown for momentum optimizers on scale-invariant weights. *arXiv preprint arXiv:2006.08217*, 2020. [5](#)
- [14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. [5](#), [6](#)
- [15] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Embedding transfer with label relaxation for improved metric learning. pages 3967–3976, June 2021. [3](#), [4](#)
- [16] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Self-taught metric learning without labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7431–7441, 2022. [2](#), [3](#), [4](#)
- [17] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access*, 10:78446–78454, 2022. [2](#), [5](#), [6](#), [7](#)
- [18] Christopher Liao, Theodoros Tsiligkaridis, and Brian Kulis. Supervised metric learning to rank for retrieval via contextual similarity optimization. *arXiv preprint arXiv:2210.01908*, 2022. [4](#)
- [19] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. [5](#)
- [20] Duc Tam Nguyen, Zhongyu Lou, Michael Klar, and Thomas Brox. Anomaly detection with multiple-hypotheses predictions. In *International Conference on Machine Learning*, pages 4800–4809. PMLR, 2019. [1](#), [2](#)
- [21] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. *Advances in neural information processing systems*, 31, 2018. [1](#), [2](#)
- [22] 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. [2](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [23] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but different: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1907–1916, 2021. [1](#), [2](#)
- [24] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018. [2](#)
- [25] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3379–3388, 2018. [1](#), [2](#)
- [26] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pages 4–11, 2014. [1](#), [2](#)
- [27] Adrian J Simpson and Mike J Fitter. What is the best index of detectability? *Psychological Bulletin*, 80(6):481, 1973. [7](#)
- [28] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-GAN: Speeding up GAN training using core-sets. 119:9005–9015, 13–18 Jul 2020. [3](#), [4](#)

- [29] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 5, 6
- [30] 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. 2
- [31] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5, 6