Continuous Memory Representation for Anomaly Detection

Joo Chan Lee[1*], Taejune Kim[1,2*], Eunbyung Park[1†], Simon S. Woo[1†], and Jong Hwan Ko[1†]

[1] Sungkyunkwan University
[2] Robotics Lab, Hyundai Motor Company

Abstract. There have been significant advancements in anomaly detection in an unsupervised manner, where only normal images are available for training. Several recent methods aim to detect anomalies based on a memory, comparing or reconstructing the input with directly stored normal features (or trained features with normal images). However, such memory-based approaches operate on a discrete feature space implemented by the nearest neighbor or attention mechanism, suffering from poor generalization or an identity shortcut issue outputting the same as input, respectively. Furthermore, the majority of existing methods are designed to detect single-class anomalies, resulting in unsatisfactory performance when presented with multiple classes of objects. To tackle all of the above challenges, we propose CRAD, a novel anomaly detection method for representing normal features within a "continuous" memory, enabled by transforming spatial features into coordinates and mapping them to continuous grids. Furthermore, we carefully design the grids tailored for anomaly detection, representing both local and global normal features and fusing them effectively. Our extensive experiments demonstrate that CRAD successfully generalizes the normal features and mitigates the identity shortcut, furthermore, CRAD effectively handles diverse classes in a single model thanks to the high-granularity continuous representation. In an evaluation using the MVTec AD dataset, CRAD significantly outperforms the previous state-of-the-art method by reducing 65.0% of the error for multi-class unified anomaly detection. Our project page is available at https://tae-mo.github.io/crad/.

Keywords: Anomaly detection · Continuous memory representation

1 Introduction

With the recent advances in deep neural networks, anomaly detection (AD) has been applied for a wide range of applications such as manufacturing industry [19, 28], video surveillance [31, 33], and medical imaging [32, 35]. Despite its success,

* Equal contribution.
† Corresponding authors.

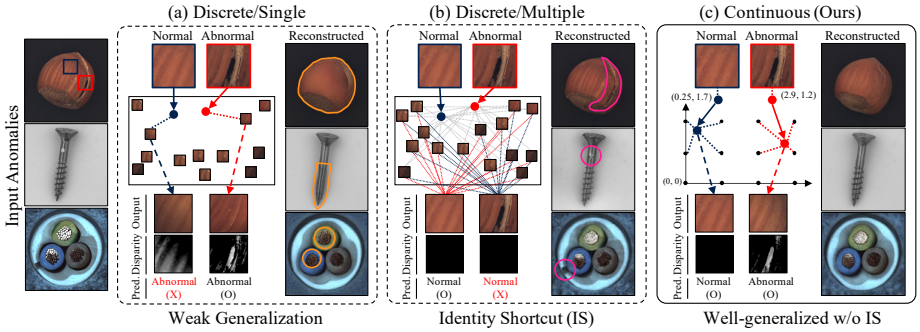

Fig. 1: Conceptual diagram and qualitative results of existing methods and ours. (a) and (b) use single and multiple normal features in a discrete memory, respectively, while our method (c) exploits continuous feature memory. We visualize the anomaly detection process with the normal (navy) and abnormal (red) patches of the top-left reference image. 'Pred.' indicates the prediction based on the disparity, and wrong predictions are marked as (X) with red color. We present the reconstruction results based on the reference abnormal images.

there are still several limitations hindering the broader applicability across many practical scenarios. In particular, one major bottleneck is the collection of a sufficient amount of anomalous data, which is scarce by definition of an anomaly. Furthermore, many AD systems often require considerable effort for pixel-wise labeling of 'normal' versus 'abnormal' data. Due to these challenges, there has been a growing interest in developing methods in an unsupervised manner where we train AD models solely with normal data [28].

As notable examples, PatchCore [28], PaDiM [8], and SPADE [7] proposed using additional memory that directly stores normal features (or distributions). During inference, these memory-based methods detect anomalies based on the distance between the testing input and its nearest neighbor in their memory, as shown in Fig. 1(a). While showing promising performance, these methods require storing a wide array of diverse normal features in memory, resulting in high space complexity and resource-intensive search operations. Moreover, these methods are often ineffective in identifying the characteristics of global anomalies due to their diversity, leading to suboptimal AD performance (see Tab. 1).

Another line of work [13, 14, 23] focuses on producing generalized normal features. Unlike the aforementioned approaches that use the nearest neighbor technique, these methods combine multiple normal features from the memory using an attention mechanism (i.e., referring to multiple discrete features), given a normal or abnormal input (Fig. 1(b)). They assume that the model always generates normal features, regardless of whether the inputs are normal or abnormal, thus anomalous regions can be detected based on the disparity between the inputs and outputs. Because these models gather diverse normal features from the memory via attention, they have exhibited increased robustness to test data,

leading to improved generalization performance. However, such strength may turn into a drawback when testing abnormal inputs. If these abnormal inputs can be reconstructed using a combination of normal features, the model could potentially generate outputs that are identical to the abnormal inputs. This issue, referred to as an identity shortcut (IS) by UniAD [37], prevents the models from detecting anomalies due to the minimal difference between the abnormal input and produced output.

Furthermore, a significant limitation exists in most of the approaches discussed earlier, as they are primarily designed to handle only one class of objects per model. When these methods are extended to multi-class scenarios, where a single model handles multiple classes, a significant performance drop has been observed [37], even with state-of-the-art methods. To mitigate this limitation, memory-based methods may incorporate sufficient memory to accommodate multi-class normality, yet this simultaneously increases memory consumption and search latency. Moreover, attention-based methods experience more severe challenges with the IS problem, as the greater number and diversity of aggregated features tend to more easily represent anomalies. These challenges necessitate training distinct models for each class, which increases training complexity, memory usage, computational overheads, and even data preparation efforts in a practical implementation.

To address all of the issues above, we propose a novel continuous memory representation for anomaly detection (CRAD), where we use an external grid-based representation for normal features (Fig. 1(c)). Given that the input to the grid is a spatial feature from an image rather than coordinates required for conventional grids, we need a specially designed framework for handling feature-based inputs. In light of this requirement, we transform input spatial features into low-dimensional coordinates, based on which we interpolate neighboring normal features in the grid. This continuous memory, unlike other discrete counterparts, allows for instant ($\mathcal{O}(1)$ time complexity) retrieval of the normal feature, while the interpolation technique mitigates the weak generalization issue. Also, as our approach does not rely on innumerable features across the entire memory, it reduces the risk of generating entirely new features (unseen anomalies in our context), thus helping to avoid the IS problem. These advantages are even stronger in the multi-class scenario, where a larger memory space would be needed, by avoiding the tremendous computation associated with searching or aggregating every feature.

Deploying the continuous memory, we additionally include specific designs on CRAD tailored for AD. CRAD incorporates two distinct continuous memories to represent normal features from both a local and global perspective. Through the integration of these representations, CRAD adeptly identifies coarse-to-fine anomalies. Furthermore, we implement coordinate jittering to enhance the generalization capability of the grid, facilitating the update of a broader range of grid values with each input coordinate. A feature refinement process further improves CRAD, minimizing false detections (i.e., false positives) by ensuring that the normal regions in the fused output remain consistent with the original input.

CRAD not only addresses weak generalization and IS issues but also provides several additional advantages. Whereas discrete memories struggle to represent global features (e.g., an entire input feature), our continuous memory successfully captures their structural characteristics (Tab. 1). Therefore, it enables identifying anomalies across a wider range of classes, each featured by distinct structures. Furthermore, different from discrete memories with limited entries, CRAD represents an infinite number of normal features in the continuous memory. Thus, CRAD achieves high performance with compact memory, resulting in high parameter efficiency. To the best of our knowledge, this is the first work leveraging continuous memory to effectively represent normal features for AD.

In the extensive experiments, we demonstrate the superiority of continuous memory over the existing discrete spaces in terms of accuracy and efficiency (both for computation and parameter). Furthermore, experimental results on the MVTec AD dataset [3] show that CRAD achieves state-of-the-art performance in a unified setting (multi-class AD with a single model) for anomaly detection, which even outperforms the state-of-the-art models trained for each respective class. With the comprehensive analysis, we demonstrate CRAD is an effective solution for AD, overcoming the limitations of existing methods.

2 Related Work

2.1 Unsupervised Anomaly Detection

Confronted with the difficulties in collecting and annotating anomalous data, recent works have focused on an unsupervised approach, where only normal images are available for training. Several studies have explored how a model trained only on normal data behaves differently when exposed to anomalous test inputs. For instance, reconstruction-based methods [18, 27, 37, 40] utilize auto-encoders [2, 5, 36] or GANs [1, 24, 29, 32] to reconstruct the normal feature regardless of input's normality, and then compare the reconstructed outcomes and original inputs to detect and localize the anomalies. Similarly, distillation-based methods [4, 9, 30, 34] exploit the disparity between the output of student and teacher networks on anomalous input.

Other methods leverage auxiliary memory to retain normal features, where they are classified into the following two categories: reference to a single discrete feature (Fig. 1(a)) and reference to a combination of discrete features (Fig. 1(b)). The former methods [7, 8, 26, 28] detect anomalies by measuring the distance between the input and stored features (or feature distributions) extracted by a pre-trained network. For instance, PatchCore [28] and SPADE [7] are designed to store the representative normal features in a memory bank and use the nearest neighbor search for anomaly scoring. However, the above methods present a significant challenge to represent features that are not already stored in their memory. Therefore, when faced with a complex and diverse range of inputs, they result in limited performance or require tremendous memory usage to achieve satisfactory performance. Moreover, an increase in the number of stored fea-

tures can significantly delay the time to reference all the stored ones, further hampering their effectiveness.

The latter methods [13, 14, 23] employ attention-like techniques to take a weighted sum of all normal features in the discrete space based on their similarity to the input. While these approaches exhibit superior generalization capabilities compared to the former methods, they generalize not only to normal features but also to abnormal features, using combinations of all features in the memory. This causes the input anomalies to be reconstructed, coined as the IS problem, reducing the AD performance by hindering the model from recognizing the disparity, as depicted in Fig. 1(b).

2.2 Unified Model for Multiple Classes

While the approaches mentioned above exhibit promising performance in identifying anomalies within a single class, they might not be easily and practically deployable due to various issues. When targeting multi-class objects, the number of required models increases, resulting in multiplied memory and computational overhead. Moreover, training numerous models in proportion to the number of object classes further complicates their practical implementation. Conversely, when these methods are applied to address multiple classes with a single model to avoid the above issues, they suffer from a significant performance drop [37]. This is because multi-class data pose a more complex problem for models originally designed for a single class, where more classes entail more complex and diverse underlying class distributions.

Recently, UniAD [37] introduced a framework capable of detecting and localizing anomalies in multi-classes setting with a single model. UniAD defines the IS problem, which means the reconstruction-based models tend to be trained as an identity function, thereby outputting the same as input even if the input contains anomalies. This hinders the model from identifying anomalies based on the disparity between the input and output. To mitigate the IS issue, UniAD introduces a learnable query with neighbor-masked attention (NMA). NMA restricts each query feature from attending an input feature in the same and neighboring location. However, UniAD shows limited performance due to the lack of a special design for multi-class scenarios, such as employing fixed queries regardless of the input's class or visual characteristics. Although several recent works have explored on unified AD framework by using synthesized anomalies [39] and vector quantization [21], they still show limited detection performance.

2.3 Grid Representation

In the revolution of neural fields or neural representations that parameterize signals by a function of coordinates, grid representation has been demonstrated to be effective in various tasks, including image and video processing [11,16], 3D reconstruction [15,22], and novel view synthesis [6,10,20]. The grid structure is capable of efficiently representing high-frequency components without spectral

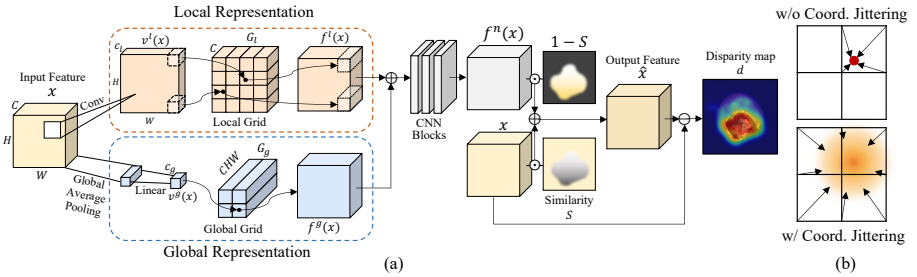

Fig. 2: (a) The detailed architecture of CRAD and (b) visualization of coordinate jittering. The input x is firstly transformed into pixel-wise and feature-wise coordinates. After the normal features are sampled from local and global representations, they are fused by CNN blocks. The final reconstruction is acquired through the proposed feature refinement process.

bias [17,25], and effectively generalizing features by offering a continuous feature space.

In this work, we propose incorporating grid representation to achieve high-performance AD. Our key contribution involves representing the normal features in a continuous space by substituting the discrete feature memory to the continuous grid in order to resolve the challenging issues discussed above while achieving high performance.

3 CRAD

Background. To help the readers understand CRAD, we first describe the grid operation. A grid is trained as a function of coordinates with infinite resolution, outputting coordinate-corresponding features. The output feature in infinite resolution is aggregated by nearby features in the grid, based on the distance between the coordinate of the input and neighboring features. For example, when we take 1D grid sampling $\phi(\cdot; G) : \mathbb{R} \rightarrow \mathbb{R}^C$, the output feature with channel C is interpolated by neighboring values of the 1D grid $G \in \mathbb{R}^{R \times C}$, which is mathematically formulated as follows:

$$\begin{aligned} \phi(v; G) = |v - n|G[m] + |v - m|G[n], \\ m = \lfloor v \rfloor,\ n = \lceil v \rceil, \end{aligned} \tag{1}$$

where $v \in \mathbb{R}$ is an arbitrary input coordinate normalized to the grid resolution R, and $G[i]$ denotes the feature from index i of the grid G. m and n are indices to be referenced, and $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote floor and ceiling operation, respectively. The above equation can be simply extended to a higher dimension D by interpolating 2^D values of a D-dimensional grid (e.g., $2^D = 4$ values in a 2D grid in Fig. 1(c)).

Overview. The motivation of our work is to effectively represent the normal features in continuous memory using the grid operation, distinct from the discrete memories. In an unsupervised manner, CRAD detects anomalous images and regions based on the discrepancy between the input feature and output normal feature, as described in Fig. 2(a). Therefore, the primary objective of CRAD is to effectively retain the normal components (e.g., shapes or textures) of the original input feature while eliminating any anomalies presented within the feature.

To this end, we represent normal features in the continuous memory during the training phase, coined as normal representation, which is used for replacing abnormal features in the testing phase. We describe how CRAD represents normal features in the continuous memory and acquires the output feature $\hat{x}$, based on the input feature x extracted from a pre-trained backbone, where x, $\hat{x} \in \mathbb{R}^{C\times H\times W}$, and C, H, W are the channel, height, and width of the feature, respectively.

3.1 Normal Representation

The fundamental concept of CRAD is to transform the input feature into specific coordinates of continuous values, which are subsequently mapped to feature grids. In particular, we design to represent the normal features from local and global perspectives. By combining the distinctive features from each perspective, the resulting feature can provide a strong representation of the input, capturing both fine-grained details as well as broader overall structures.

Local representation. As shown in Fig. 2(a), CRAD samples each pixel of the feature, which characterizes each patch of the image, to represent the local feature. Then, the channels of each pixel are transformed to corresponding coordinates (a low-dimensional vector) by convolutional layers with a kernel size of 1, followed by hyperbolic tangent activation. Using these pixel-wise coordinates, we obtain normal features sampled from the local grid representation. More formally, we define a function $v^l(\cdot) : \mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{C_l\times H\times W}$, which generates pixel-wise coordinates based on the input feature, where C_l is the dimension of the produced coordinates. Given the pixel-wise coordinates $v^l_{h,w}(\cdot) \in \mathbb{R}^{C_l}$, normal features are sampled from C_l-dimensional grid G_l, which has the resolution of each dimension R_l and channel of C. The equation for local representation $f^l(x) : \mathbb{R}^{C\times H\times W} \rightarrow \mathbb{R}^{C\times H\times W}$ is written as follows:

$$f^l_{h,w}(x) = \phi(v^l_{h,w}(x); G_l), \tag{2}$$

where $\phi(\cdot; G_l) : \mathbb{R}^{C_l} \rightarrow \mathbb{R}^C$ represents sampling feature from grid G_l by bilinearly interpolating the grid values based on the coordinates.

As each pixel of the feature characterizes a patch in an image, the local representation ensures retaining normal patches and replacing abnormal patches with normal patches that have similar local context. Hence, when a normal patch is fed, even though there is no exact match in the training patches, a corresponding normal feature can be represented by interpolating normal features mapped at nearby coordinates.

In addition, for an abnormal patch, CRAD finds a normal feature that is the most representative of the abnormal patch based on the reduced coordinates. As the grid has never been exposed to abnormal features during training, it is unable to represent abnormal features by interpolating nearby normal features. This is the core idea of how we can effectively resolve the identity shortcut (IS) issue frequently found in the existing methods that aggregate numerous features based on similarities using attention mechanisms [13, 23].

Global representation. Anomalous regions can exist not only locally within an image but also at a global scale. To handle such global anomalous cases, CRAD maintains another grid representation to capture the global feature of an image. Similar to the local representation, we formulate the function to obtain global feature coordinates $v^g(\cdot) : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C_g}$, where C_g is the reduced dimension of coordinates. For the function $v^g(\cdot)$, we employ global average pooling and linear layers, as shown in Fig. 2(a). The feature-wise coordinates are mapped to each normal feature by C_g-dimensional grid G_g that has the resolution of each dimension R_g. An element of the grid G_g is a CHW dimensional vector, which is reshaped to $C \times H \times W$ tensor once sampled. The equation for global representation $f^g(x) : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{C\times H\times W}$ is expressed as follows:

$$f^g(x) = \mathsf{reshape}(\phi(v^g(x); G_g)), \tag{3}$$

where $\phi(\cdot; G_g) : \mathbb{R}^{C_g} \to \mathbb{R}^{CHW}$ represents sampling feature from grid G_g by bilinear interpolation, and $\mathsf{reshape}(\cdot) : \mathbb{R}^{CHW} \to \mathbb{R}^{C\times H\times W}$ denotes the reshape operation.

The global representation not only effectively replaces global anomalies as a whole but also distinguishes the class-wise distribution for the unified setting. Based on the image-wise features, the reduced coordinates are well distributed on the continuous space, modeling the decision boundary of complex distribution (see Fig. 3(b) and Sec. 4.2 for more information).

Fused representation. We combine the local and global representations $f^l(x)$ and $f^g(x)$ to effectively learn the normal representation $f^n(x)$, as shown in Fig. 2(a). The local and global representations are concatenated and then fed into the following convolution networks $\psi(\cdot) : \mathbb{R}^{2C\times H\times W} \to \mathbb{R}^{C\times H\times W}$ to reconstruct $f^n(x)$ as follows:

$$f^n(x) = \psi(\mathsf{concat}(f^l(x), f^g(x))), \tag{4}$$

where $\mathsf{concat}(\cdot, \cdot)$ denotes the concatenation of two features along with the channel axis. By fusing the local and global representation, CRAD can represent normal features from fine-grained details to broader contexts, resulting in higher performance compared to the cases using only either of them (see ablation study in Sec. 4.4).

3.2 Feature Refinement

Despite the fusion of local and global normal representation, deviations for the normal regions between $f^n(x)$ and x can still exist, which can lead to false

detection (i.e., false positives). Hence, in feature refinement, we aim to refine $f^n(x)$ in the regions that are supposed to be normal but deviate from x, with the goal of reducing false positives. To identify such regions, we evaluate the pixel-wise similarity between x and $f^n(x)$ by combining both Mean Squared Error (MSE) and cosine similarity. These two metrics offer a comprehensive view of the differences between normal and abnormal features, where MSE captures the absolute intensity disparities while cosine similarity characterizes structural and positional similarity. By considering the combined similarity $S \in \mathbb{R}^{H \times W}$, we can reconstruct $\hat{x}$ as follows:

$$\hat{x}_{h,w} = S_{h,w} x_{h,w} + (1 - S_{h,w}) f^n_{h,w}(x), \tag{5}$$

$$S_{h,w} = \lambda_1 \mathbb{1}[\mathsf{mse}(x_{h,w}, f^n_{h,w}(x)) < k] + \lambda_2 \mathsf{cosim}(x, f^n(x)), \tag{6}$$

where h, w are the indices of the spatial feature, $\mathbb{1}[\cdot]$ is the indicator function and $\mathsf{mse}(\cdot,\cdot)$ and $\mathsf{cosim}(\cdot,\cdot)$ are the MSE and cosine similarity, respectively. To use MSE as a measure of similarity, we convert the MSE value to either 0 or 1, depending on whether it surpasses the threshold k or not.

3.3 Training and Inference

Coordinate jittering. To achieve a more generalized grid representation, we apply Gaussian noise to vectorized local coordinates $v^l(x)$ in the training phase. For instance, without jittering, a coordinate affects up to four grid values in a 2D grid, as shown in Fig. 2(b). In contrast, when perturbating the coordinate, we can update more grid values with bell-shaped distribution in each iteration, producing a more generalized grid.

Training. Given x and $\hat{x}$ derived from CRAD, we employ the MSE loss as an objective function, as follows:

$$\mathcal{L} = \frac{1}{CHW} \left\| x - \hat{x} \right\|_2^2 . \tag{7}$$

Based on Eq. (7), we learn the entire model in an end-to-end manner, including the grids initialized by Xavier normal initialization [12]. As x is always a normal input in the training phase, the grids are learned to represent normal features.

Inference. To perform anomaly detection and localization through the disparity between x and $\hat{x}$, an anomaly score map $d \in \mathbb{R}^{H \times W}$ is formulated as follows:

$$d_{h,w} = \left\| x_{h,w} - \hat{x}_{h,w} \right\|_2 , \tag{8}$$

where h and w indicate the location of each pixel. To match with the corresponding ground truth, d is interpolated into the original shape of the input. An anomaly score for each image is obtained by taking the max value from the average-pooled d, and the interpolated anomaly map itself is used for the pixel-wise anomaly score.

4 Experimental Results

4.1 Experimental Setup

We used MVTec AD [3] and VisA [41] datasets, which are representative datasets for real-world unsupervised AD. We evaluated the performance of anomaly detection by the Area Under the Receiver Operator Curve (AUROC). Following previous studies, we computed the class-average AUROC for detection and pixel-wise AUROC for localization. We implemented CRAD in the PyTorch framework, and we used the NVIDIA A5000 GPU for all evaluations. We trained our models for 50 epochs, thrice with different seeds (0,1,2), with a batch size of 64. We describe the detailed implementation of CRAD in the supplementary materials.

We evaluated the performance under two different scenarios: 1) a unified setting where a single model is used for anomaly detection across multiple classes, and 2) a separate setting in which we utilize respective models for different classes. When training a unified model across all methodologies, we maintained the model size to be consistent with each separate model.

4.2 Effectiveness of Continuous Memory Representation

Improved performance. To assess the efficacy of the continuous memory, we implemented two baselines with discrete memories under the same overall detection framework of CRAD as follows: 1) referring to a single feature from discrete space (Fig. 1(a)) through vector quantization (VQ), and 2) referring to a combination of multiple discrete features (Fig. 1(b)) with an attention module. As shown in Tab. 1, CRAD, providing a continuous memory, outperforms the other baselines for both local and global representation. When we expand the memory size for local representation, the attention shows the performance drop, suffering from a more severe IS. Although VQ shows performance improvement with larger memory entries, it still falls short of CRAD even with the quadrupled feature space. Furthermore, the baselines consistently underperform in global representation, indicating their inability to represent structural information of the entire feature.

Table 1: Performance evaluation of the different feature memories in the unified setting. #Entry denotes the number of features in each memory and Persp. indicates the perspective (local or global).

Persp.	Method	#Entry	Detection	Localization
Local	VQ	64	96.9±0.65	96.1±0.05
		256	97.8±0.23	96.0±0.08
	Attention	64	95.9±1.1	96.2±0.25
		256	93.9±1.8	95.1±0.76
	CRAD	64	**98.6±0.07**	**97.5±0.04**
Global	VQ	16	81.0±0.97	89.9±0.42
		64	82.2±0.56	91.4±1.0
	Attention	16	77.9±2.3	86.7±3.4
		64	82.1±0.54	90.6±0.19
	CRAD	16	**92.3±0.60**	**95.7±0.17**

Visualization of coordinates. Although the quantitative results above clearly demonstrate the effectiveness of the continuous normal representation of CRAD, we additionally visualize the generated and mapped coordinates in Fig. 3. The normal and abnormal areas with a similar local characteristic are mapped at a near distance in the local feature space (e.g., the patch 317 and 485 in Fig. 3(a)). Similarly, the global coordinates of the two input images are mapped to almost

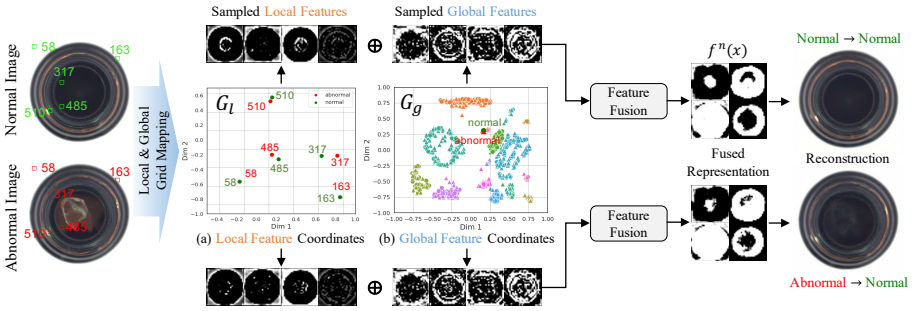

Fig. 3: Visualization of CRAD's pipeline. Each marker in (a) corresponds to the patch on the left image that has the same number and color. Each marker in (b) corresponds to a single image from the test dataset, where different colors represent distinct classes, and circles and triangles denote the normal and abnormal images, respectively. 'Dim 1' and 'Dim 2' are the two dimensions of 2D grids.

the same location at the global feature space (Fig. 3(b)). These results indicate that the model successfully learns to generate coordinates corresponding to each input feature, and the local and global grids can represent the normal features from each perspective effectively.

Class	US [4]	PaDiM [8]	MKD [30]	DRAEM [38]	RD4AD [9]	PatchCore [28]	UniAD [37]	HVQ-T [21]	CRAD (Ours)
Bottle	84.0/99.0	97.9/99.9	98.7/99.4	97.5/99.2	98.7/100	100/100	99.7/100	100/-	100±0.00/100
Cable	60.0/86.2	70.9/92.7	78.2/89.2	57.8/91.8	85.0/95.0	99.7/99.4	95.2/97.6	99.0/-	99.1±0.34/99.7
Capsule	57.6/86.1	73.4/91.3	68.3/80.5	65.3/98.5	95.5/96.3	90.9/97.8	86.9/85.3	95.4/-	97.0±0.05/98.4
Hazelnut	95.8/93.1	85.5/92.0	97.1/98.4	93.7/100	87.1/99.9	100/100	99.8/99.9	100/-	100±0.06/100
Metal Nut	62.7/82.0	88.0/98.7	64.9/73.6	72.8/98.7	99.4/100	99.9/100	99.2/99.0	99.9/-	100±0.00/100
Pill	56.1/87.9	68.8/93.3	79.7/82.7	82.2/98.9	52.6/96.6	96.9/96.0	93.7/88.3	95.8/-	98.6±0.36/98.7
Screw	66.9/54.9	56.9/85.8	75.6/83.3	92/93.9	97.3/97.0	90.1/97.0	87.5/91.9	95.6/-	97.6±0.33/98.6
Toothbrush	57.8/95.3	95.3/96.1	75.3/92.2	90.6/100	99.4/99.5	100/99.7	94.2/95.0	93.6/-	99.2±0.73/96.1
Transistor	61.0/81.8	86.6/97.4	73.4/85.6	74.8/93.1	92.4/96.7	99.7/100	99.8/100	99.7/-	99.8±0.18/99.9
Zipper	78.6/91.9	79.7/90.3	87.4/93.2	98.8/100	99.6/98.5	94.7/99.5	95.8/96.7	97.9/-	99.2±0.13/99.6
Carpet	86.6/91.6	93.8/99.8	69.8/79.3	98.0/97.0	97.1/98.9	97.1/98.7	99.8/99.9	99.9/-	99.9±0.05/100
Grid	69.2/81.0	73.9/96.7	83.8/78.0	99.3/99.9	99.7/100	96.3/97.9	98.2/98.5	97.0/-	100±0.0/100
Leather	97.2/88.2	99.9/100	93.6/95.1	98.7/100	100/100	100/100	100/100	100/-	100±0.00/100
Tile	93.7/99.1	93.3/98.1	89.5/91.6	99.8/99.6	97.5/99.3	99.0/98.9	99.3/99.0	99.2/-	100±0.00/100
Wood	90.6/97.7	98.4/99.2	93.4/94.3	99.8/99.1	99.2/99.2	99.5/99.0	98.6/97.9	97.2/-	99.6±0.51/99.2
Mean	74.5/87.7	84.2/95.5	81.9/87.8	88.1/98.0	93.4/98.5	97.6/99.0	96.5/96.6	98.0/-	**99.3±0.08**/**99.4**

Table 2: Quantitative results for anomaly detection, evaluated with AUROC metric on MVTec-AD. All methods are evaluated under the unified and separate settings.

Sampled features from the grids (normal vs. abnormal). In addition, we visualize the sampled features from the local and global grids based on the learned coordinates. Fig. 3 shows that the sampled features (from the local and global grids) with near coordinates share similar characteristics whether the input image is normal or not. Furthermore, the fused normal representations of both normal and abnormal inputs are reconstructed into normal images. Specif-

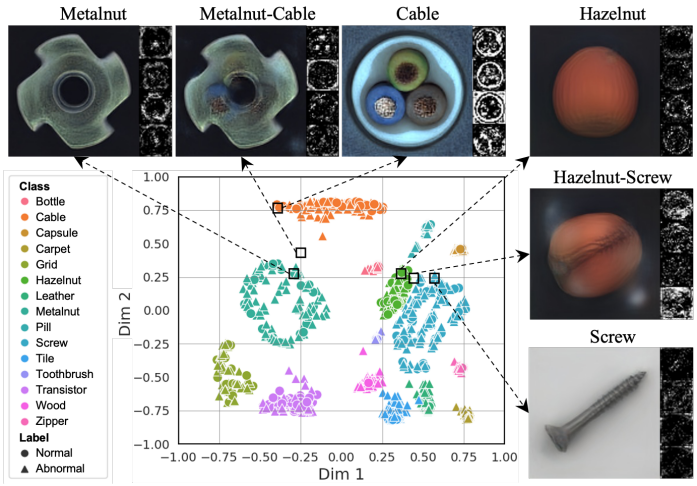

Fig. 4: Visualization of the contents mapped at a continuous grid. We manually select six global coordinates and visualize the corresponding sampled normal features.

Class	US [4]	PaDiM [8]	MKD [30]	DRAEM [38]	RD4AD [9]	PatchCore [28]	UniAD [37]	HVQ-T [21]	CRAD (Ours)
Bottle	67.9/97.8	96.1/98.2	91.8/96.3	87.6/99.1	97.7/98.7	98.4/98.6	98.1/98.1	98.3/-	98.2±0.10/98.6
Cable	78.3/91.9	81.0/96.7	89.3/82.4	71.3/94.7	83.1/97.4	96.7/98.5	97.3/96.8	98.1/-	98.4±0.17/98.3
Capsule	85.5/96.8	96.9/98.6	88.3/95.9	50.5/94.3	98.5/98.7	94.8/98.9	98.5/97.9	98.8/-	98.7±0.08/98.6
Hazelnut	93.7/98.2	96.3/98.1	91.2/94.6	96.9/99.7	98.7/98.9	98.6/98.7	98.1/98.8	98.8/-	98.5±0.17/98.9
Metal Nut	76.6/97.2	84.8/97.3	64.2/86.4	62.2/99.5	94.1/97.3	98.3/98.4	94.8/95.7	96.3/-	97.5±0.36/97.3
Pill	80.3/96.5	87.7/95.7	69.7/89.6	94.4/97.6	96.5/98.2	97.3/97.6	95.0/95.1	97.1/-	98.2±0.03/98.0
Screw	90.8/97.4	94.1/98.4	92.1/96.0	95.5/97.6	99.4/99.6	98.0/99.4	98.3/97.4	98.9/-	99.3±0.04/99.2
Toothbrush	86.9/97.9	95.6/98.8	88.9/96.1	97.7/98.1	99.0/99.1	98.4/98.7	98.4/97.8	98.6/-	98.8±0.04/98.7
Transistor	68.3/73.7	92.3/97.6	71.7/76.5	64.5/90.9	86.4/92.5	94.9/96.4	97.9/98.7	97.9/-	98.1±0.14/98.3
Zipper	84.2/95.6	94.8/98.4	86.1/93.9	98.3/98.8	98.1/98.2	95.8/98.9	96.8/96.0	97.5/-	97.8±0.06/97.9
Carpet	88.7/93.5	97.6/99.0	95.5/95.6	98.6/95.5	98.8/98.9	98.9/99.1	98.5/98.0	98.7/-	98.6±0.06/98.7
Grid	64.5/89.9	71.0/97.1	82.3/91.8	98.7/99.7	99.2/99.3	96.9/98.7	96.5/94.6	97.0/-	98.0±0.05/98.0
Leather	95.4/97.8	84.8/99.0	96.7/98.1	97.3/98.6	99.4/99.4	99.3/99.3	98.8/98.3	98.8/-	98.9±0.07/99.1
Tile	82.7/92.5	80.5/94.1	85.3/82.8	98.0/99.2	95.6/95.6	95.9/95.9	91.8/91.8	92.2/-	94.4±0.16/94.6
Wood	83.3/92.1	89.1/94.1	80.5/84.8	96.0/96.4	96.0/95.3	94.4/95.1	93.2/93.4	92.4/-	93.8±0.09/93.8
Mean	81.8/93.9	89.5/97.4	84.9/90.7	87.2/97.3	96.0/97.8	97.1/98.1	96.8/96.6	97.3/-	**97.8±0.12**/97.9

Table 3: Quantitative results for anomaly localization, evaluated with AUROC metric on MVTec-AD. All methods are evaluated under the unified and separate settings.

ically, CRAD preserves the fine-grained details of the bottle (normal region), while it reconstructs the corresponding normal state of the anomalous region that has never been encountered during training. This result demonstrates that the continuous feature space efficiently tackles the two major challenges in discrete feature space: weak generalization and IS.

Decision boundary of multiple classes. Fig. 4 describes the coordinate distribution using a model trained with global representation. The images from each class form clusters in the continuous memory space, effectively modeling the decision boundaries between classes. This implies that the continuous memory

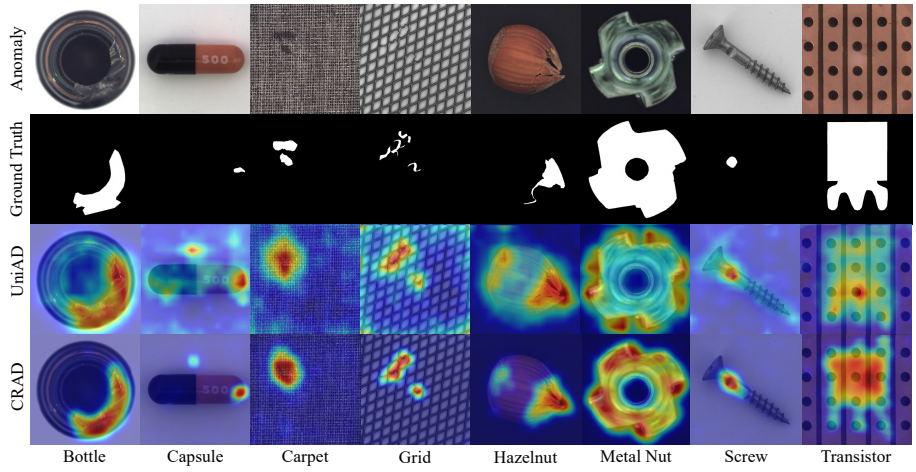

Fig. 5: Qualitative results of CRAD on MVTec AD. Each row of the figure represents anomaly images, corresponding ground truths, results from UniAD, and our results.

can represent well-defined structural features. Furthermore, the reconstructions of the sampled features at the decision boundary show combined characteristics of near classes, demonstrating the high granularity of continuous features. Leveraging the advantages of the continuous feature memory, CRAD can model correct decision boundaries in complex multi-class distributions with compact representations, leading to high performance in a unified setting.

4.3 Anomaly Detection and Localization

We evaluate CRAD in comparison with recent state-of-the-art methods in both unified and separate settings, focusing on detection (Tab. 2) and localization (Tab. 3) performance on MVTec AD. In the unified setting, the methods not specifically designed for multiple classes exhibit a significant performance drop compared to their performance in the separate setting. In contrast, UniAD, HVQ-Trans, and CRAD maintain the performances of their separate models in the unified setting. Among these, CRAD notably outperforms UniAD and HVQ-Trans, achieving state-of-the-art performance in the unified setting. Specifically, CRAD successfully reduces the error rate of HVQ-Trans from 2.0% to 0.7%, bringing a total error reduction of 65.0%. For detection, the unified CRAD even outperforms separate models of PatchCore, which is the previous state-of-the-art in single-class AD. Similarly, for localization, CRAD achieves the best performance in the unified setting and matches PatchCore in a separate setting. Fig. 5 showcases the qualitative results of UniAD and CRAD in the unified setting, highlighting CRAD's superior prediction quality with fewer noisy areas. We additionally evaluate CRAD on VisA, which is a more challenging dataset. As

Class		Detection				Localization			
		UniAD [37]	PatchCore [28]	OmniAL [39]	CRAD (ours)	UniAD [37]	PatchCore [28]	OmniAL [39]	CRAD (ours)
Complex Structure	PCB1	94.8/90.2	97.6/98.5	77.7/96.6	96.8/95.4	99.3/99.2	99.7/99.8	97.6/98.7	99.5/99.5
	PCB2	92.5/84.2	96.7/97.2	81.0/99.4	92.9/92.7	97.6/96.5	98.0/98.7	93.9/83.2	97.6/97.0
	PCB3	86.6/90.7	97.3/98.5	88.1/96.9	95.2/96.1	98.1/98.0	99.3/99.4	94.7/98.4	98.7/98.6
	PCB4	99.3/97.4	99.7/99.7	95.3/97.4	99.4/98.6	97.6/97.2	97.7/98.2	97.1/98.5	98.6/98.4
Multiple Instances	Candle	97.0/90.2	94.7/99.4	86.8/85.1	96.3/96.6	99.1/99.0	98.3/99.3	95.8/90.5	99.2/99.2
	Capsules	70.7/80.3	75.0/76.3	90.6/87.9	90.5/91.5	98.1/98.5	99.1/99.2	99.4/98.6	99.5/99.5
	Macaroni1	90.4/90.2	94.7/97.4	92.6/96.9	96.6/96.0	99.1/99.0	99.0/99.7	98.6/98.9	99.1/99.1
	Macaroni2	82.8/77.4	78.6/76.7	75.2/89.9	88.7/90.4	97.7/97.4	96.1/98.6	97.9/99.1	98.8/99.0
Single Instance	Cashew	93.8/92.9	97.3/97.8	88.6/97.1	95.5/96.4	98.9/99.2	98.1/98.7	95.0/98.9	97.4/98.0
	Chewinggum	99.3/98.3	98.5/98.8	96.4/94.9	99.5/98.9	99.1/98.5	98.9/98.9	99.0/98.7	98.3/98.4
	Fryum	88.8/84.4	95.4/96.0	94.6/97.0	94.5/93.7	97.7/96.7	89.8/92.4	92.1/89.3	96.6/96.3
	Pipe fryum	97.0/91.8	99.2/99.8	86.1/91.4	96.6/98.3	99.3/99.3	97.5/98.9	98.2/99.1	99.4/99.4
Mean		91.1/89.0	93.7/94.7	87.8/94.2	**95.2**/**95.4**	98.5/98.2	97.5/**98.5**	96.6/96.0	**98.6**/**98.5**

Table 4: Quantitative results for anomaly detection and localization, evaluated on VisA. All methods are evaluated under the unified and separate settings.

shown in Tab. 4, CRAD outperforms other state-of-the-art methods for detection.

4.4 Ablation Study

We conducted an ablation study on CRAD to assess the impact of individual proposals. Tab. 5 shows that our key contribution is the normal representation from both local and global contexts, which independently yields comparable performance. Notably, a model with only local representation outperforms UniAD and HVQ-Trans. The integration of both representations achieves improved performance, with additional gains from feature refinement and coordinate jittering.

Table 5: Ablation studies in the unified setting using MVTec AD.

Local	Global	Refine	Jitter	Detect	Localize
	✓			92.3	95.7
✓				98.6	97.5
✓	✓			98.8	97.7
✓	✓	✓		99.1	97.8
✓	✓	✓	✓	99.3	97.8

5 Conclusion

In this work, we have proposed a novel anomaly detection architecture, CRAD, which represents normal features in the continuous memory, unlike prior approaches limited to discrete feature space. CRAD successfully represents local as well as global features in the continuous space while overcoming the limitations of existing methods, such as weak generalization, identity shortcut, and high computational/parameter complexity. Through extensive experiments, we have demonstrated the effectiveness of CRAD qualitatively and quantitatively. Although CRAD demonstrates its superior generalization capability compared to existing methods, we found a limitation that this cannot be the case with extremely limited data (i.e., 1- or zero-shot), more discussed in the supplementary materials. We believe that it can be further addressed and our work paves the way for future advancements in anomaly detection.

Acknowledgements

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) funded by the Korea Government (MSIT) under Grant RS-2021-II212068 (Artificial Intelligence Innovation Hub) and Grant RS-2022-II220688 (AI Platform to Fully Adapt and Reflect Privacy-Policy Changes); in part by the Culture, Sports, and Tourism R&D Program through the Korea Creative Content Agency funded by the Ministry of Culture, Sports and Tourism in 2024 under Grant RS-2024-00348469 (Research on neural watermark technology for copyright protection of generative AI 3D content); and in part by the SEMES-Sungkyunkwan University collaboration funded by SEMES.

References

1. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Asian Conference on Computer Vision. pp. 622–637 (2019)
2. Baur, C., Wiestler, B., Albarqouni, S., Navab, N.: Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. pp. 161–169 (2019)
3. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9592–9600 (2019)
4. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4183–4192 (2020)
5. Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., Steger, C.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv preprint arXiv:1807.02011 (2018)
6. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision (2022)
7. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint arXiv:2005.02357 (2020)
8. Defard, T., Setkov, A., Loesch, A., Audigier, R.: Padim: a patch distribution modeling framework for anomaly detection and localization. In: International Conference on Pattern Recognition. pp. 475–489 (2021)
9. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9737–9746 (2022)
10. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022)
11. Gao, J., Wang, Z., Xuan, J., Fidler, S.: Beyond fixed grid: Learning geometric image representation with a deformable grid. In: European Conference on Computer Vision. pp. 108–125 (2020)

12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 249–256 (2010)
13. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1705–1714 (2019)
14. Hou, J., Zhang, Y., Zhong, Q., Xie, D., Pu, S., Zhou, H.: Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8791–8800 (2021)
15. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T.: Local implicit grid representations for 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6001–6010 (2020)
16. Lee, J.C., Rho, D., Ko, J.H., Park, E.: Ffnerv: Flow-guided frame-wise neural representations for videos. In: Proceedings of the ACM International Conference on Multimedia (2023)
17. Lee, J.C., Rho, D., Nam, S., Ko, J.H., Park, E.: Coordinate-aware modulation for neural fields. In: International Conference on Learning Representations (2024)
18. Liang, Y., Zhang, J., Zhao, S., Wu, R., Liu, Y., Pan, S.: Omni-frequency channel-selection representations for unsupervised anomaly detection. IEEE Transactions on Image Processing **32**, 4327–4340 (2023)
19. Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F., Jin, Y.: Deep visual anomaly detection in industrial manufacturing: A survey. arXiv preprint arXiv:2301.11514 (2023)
20. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Advances in Neural Information Processing Systems **33**, 15651–15663 (2020)
21. Lu, R., Wu, Y., Tian, L., Wang, D., Chen, B., Liu, X., Hu, R.: Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection. In: Advances in Neural Information Processing Systems. vol. 36, pp. 8487–8500 (2023)
22. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)
23. Park, H., Noh, J., Ham, B.: Learning memory-guided normality for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14372–14381 (2020)
24. Pidhorskyi, S., Almohsen, R., Doretto, G.: Generative probabilistic novelty detection with adversarial autoencoders. Advances in neural information processing systems **31** (2018)
25. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning. pp. 5301–5310 (2019)
26. Rippel, O., Mertens, P., Merhof, D.: Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In: International Conference on Pattern Recognition. pp. 6726–6733 (2021)
27. Ristea, N.C., Madan, N., Ionescu, R.T., Nasrollahi, K., Khan, F.S., Moeslund, T.B., Shah, M.: Self-supervised predictive convolutional attentive block for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13576–13586 (2022)

28. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14318–14328 (2022)
29. Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E.: Adversarially learned one-class classifier for novelty detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3379–3388 (2018)
30. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution knowledge distillation for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14902–14912 (2021)
31. Santhosh, K.K., Dogra, D.P., Roy, P.P.: Anomaly detection in road traffic using visual surveillance: A survey. ACM Computing Surveys **53**(6), 1–26 (2020)
32. Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. Medical Image Analysis **54**, 30–44 (2019)
33. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6479–6488 (2018)
34. Wang, G., Han, S., Ding, E., Huang, D.: Student-teacher feature pyramid matching for anomaly detection. arXiv preprint arXiv:2103.04257 (2021)
35. Xiang, T., Zhang, Y., Lu, Y., Yuille, A.L., Zhang, C., Cai, W., Zhou, Z.: Squid: Deep feature in-painting for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23890–23901 (2023)
36. Ye, F., Huang, C., Cao, J., Li, M., Zhang, Y., Lu, C.: Attribute restoration framework for anomaly detection. IEEE Transactions on Multimedia **24**, 116–127 (2020)
37. You, Z., Cui, L., Shen, Y., Yang, K., Lu, X., Zheng, Y., Le, X.: A unified model for multi-class anomaly detection. In: Advances in Neural Information Processing Systems. vol. 35, pp. 4571–4584 (2022)
38. Zavrtanik, V., Kristan, M., Skočaj, D.: Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8330–8339 (2021)
39. Zhao, Y.: Omnial: A unified cnn framework for unsupervised anomaly localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3924–3933 (2023)
40. Zhou, K., Xiao, Y., Yang, J., Cheng, J., Liu, W., Luo, W., Gu, Z., Liu, J., Gao, S.: Encoding structure-texture relation with p-net for anomaly detection in retinal images. In: European Conference on Computer Vision. pp. 360–377 (2020)
41. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: European Conference on Computer Vision. pp. 392–408 (2022)