IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 May 2023, accepted 9 June 2023, date of publication 15 June 2023, date of current version 21 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3286548

RESEARCH ARTICLE

Interpretability-Aware Industrial Anomaly Detection Using Autoencoders

RUI JIANG[1], YIJIA XUE[1], AND DONGMIAN ZOU[1,2], (Member, IEEE)
[1]Division of Natural and Applied Sciences, Duke Kunshan University, Kunshan, Jiangsu 215316, China
[2]CMCS/DSRC, Duke Kunshan University, Kunshan, Jiangsu 215316, China

Corresponding author: Dongmian Zou (dongmian.zou@duke.edu)

The research results of this article are sponsored by the Kunshan Municipal Government research funding.

ABSTRACT The past decade has witnessed wide applications of deep neural networks in anomaly detection. However, the dearth of interpretability in neural networks often hinders their reliability, especially for industrial applications where practical users heavily rely on interpretable methods to provide explanations for their decision-making. In this paper, we propose a reconstruction-based approach to unsupervised detection of anomalies in industrial defect data. Our algorithm employs an interpretability score during both the training and test phases. Specifically, we train an autoencoder with a loss function that incorporates an interpretability-aware error term. After training, the autoencoder processes a specific feature from the difference between the test image and the average of training images and produces an attention map that is used for detecting the anomalies. Our method not only achieves competitive performance compared with non-interpretability-aware methods but also produces attention maps that facilitate a direct explanation of detection results, which can potentially be useful for industrial practitioners.

INDEX TERMS Anomaly detection, autoencoders, interpretability, visual explanation.

I. INTRODUCTION

Anomaly detection is an important research field in machine learning that aims to detect unusual patterns within given data [1], [2], [3]. It is widely used in various fields, such as network intrusion detection [4], signal processing [5], abnormal behavior detection [6], and medical image analysis [7]. Early anomaly detection algorithms were primarily used in the field of data mining [8]. However, in recent years, with the development of computer vision and related technologies, there has been an increasing interest in applying anomaly detection to the field of image processing [9], [10], [11]. In particular, many research works have introduced techniques that utilize deep learning to detect anomalies in images [12], [13].

In industrial applications, anomaly detection is crucial for detecting visual defects in products. Industrial anomaly detection aims to find visible defects in the appearance of various industrial products, including fabrics, chips, pharmaceuticals, and even building materials [14], [15], [16], [17], [18]. These defects, though minor, may seriously affect the normal function of the product. In industrial anomaly detection, it is usually easy to obtain data that show a normal pattern, whereas it is often challenging to obtain data that represent possible defects. Therefore, the most natural scenario is the unsupervised learning setting, where the task is to use unannotated samples or normal samples to build a detection model and detect anomaly samples that differ from the expected pattern [19].

Benefiting from their powerful capability of feature extraction and representation learning, methods based on convolutional neural networks (CNNs) can greatly improve detection and localization accuracy [19], [20]. For high-resolution industrial application datasets for industrial applications, there already exist powerful anomaly detectors [21], [22]. However, in addition to robustness and model performance, model interpretability is crucial for decision-making, given the strict inspection for product quality and safety, especially in the manufacturing field [23], [24]. Although various methods exist for understanding CNNs [25], [26], they do not form a part of the anomaly detection model and thus cannot guarantee that the anomaly detection model is capable

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

of correctly interpreting the results. In this paper, we innovatively use a gradient-based interpretation method [27] to include the heatmap used to explain a model as a loss, so that the model will extract features that are crucial for explaining the anomaly detection. In other words, our model promotes making decisions that align with human intuition, which are more explainable and helpful in industrial applications.

One famous class of approach in the anomaly detection area is reconstruction-based, or more specifically, based on the neural network architecture of an Autoencoder (AE) [28], [29]. The encoder transforms the input image into a latent variable, and the decoder maps the latent variable to a reconstructed version of the input image. The anomalous images can be detected since they usually have a larger reconstruction error between the input image and the reconstructed image. However, the reconstruction loss is usually unaware of the anomaly detection task and may produce uninterpretable results, e.g., a larger reconstruction error for normal pixels. In this paper, we address this problem from two aspects. First, we introduce a novel interpretability-aware loss to AE. In particular, we discourage attention over a large region where the attention is from an explainable anomaly heatmap. Second, we replace the reconstruction error used in decision-making with the heatmap for each image, which can be used for obtaining a localization map for interpreting the results. Specifically, the attention map is obtained by back-propagating the difference between averaged normal images and the candidate image, which is then processed to produce the anomaly score.

We summarize our contributions as follows.

- We introduce a novel interpretability-aware loss term for CNNs, which can be flexibly used in various models and produces interpretable anomaly detection results.
- We use this loss term to improve AE for anomaly detection. Accordingly, we further propose novel anomaly scores that are derived based on an explainable heatmap.
- The proposed anomaly score places greater emphasis on the defect area and can also detect multiple similar defects in a single image. This makes our model highly applicable and relevant for industrial practitioners.
- We conduct extensive experiments on industrial image datasets. The results, both quantitatively (in AUC scores) and qualitatively, show the effectiveness of our proposed model.

II. RELATED WORKS

We survey related works on unsupervised anomaly detection and interpretable CNN in §II-A and §II-B, respectively.

A. UNSUPERVISED ANOMALY DETECTION

Unsupervised anomaly detection, also known as novelty detection, is a critical machine learning task used to identify anomalous samples by constructing a model based on normal samples only [30]. Among the existing approaches to unsupervised anomaly detection, the most related works can be categorized into two main categories: classification-based and reconstruction-based [31], [32], [33], [34]. Classification-based anomaly detection approaches aim to extract highly discriminative features from normal samples to identify anomalous samples [35], [36]. Recent examples of classification-based anomaly detection methods include OC-SVM [37], [38] and Deep SVDD [36]. On the other hand, the objective of the reconstruction-based approach is to reconstruct samples based on the extracted features, with the anticipation that anomalous samples will receive worse reconstruction results compared to normal samples based on the training information [28]. Compared to relatively early models such as K-means [39], recent reconstruction-based approaches adopt neural networks, especially autoencoders [40], [41], [42], variational autoencoders (VAEs) [43], [44], and GANs [30], [34]. Nevertheless, none of the above approaches consider an interpretability loss as our model. When used in industrial contexts, these approaches can hardly produce meaningful interpretations.

One work that is specifically related to ours is [45], where the GradCAM attention map is integrated with a VAE model to visually explain the principle behind anomaly detection. However, their method requires the use of the special VAE architecture with latent space parametrization representing the mean and variance of the posterior. In contrast, our model relies solely on the reconstruction of an AE and contains a novel component in the loss function that incorporates the GradCAM output. This component is used for deriving the anomaly score. As a result, our approach allows us to obtain more interpretable results in anomaly detection, which is particularly useful in industrial applications.

B. INTERPRETING CONVOLUTIONAL NEURAL NETWORKS

The task of explaining CNNs has received considerable attention in recent years because it provides an understanding of the model's authenticity and enhances the reliability of its outcomes [46], [47]. Two commonly used general approaches to visual-attention-based CNN visualization are the response-based method and the gradient-based method [48], [49]. Response-based methods such as SAGAN [50], ABN [51], and Class Activation Mapping (CAM) [25] modify the original CNN architectures for auxiliary information but require specific CNN architectures. For instance, CAM implements the visualization of CNNs by modifying the model structures with a global average pooling layer. However, CAM has restrictions in that it requires a global average pooling layer to be applied to the convolutional feature maps. On the other hand, the gradient-based approach utilizes the gradients calculated through backpropagation. Similar to CAM, the Gradient-weighted Class Activation Mapping (GradCAM) [27] generates a weighted attention map for CNN models, but based on gradients computed through backpropagation. GradCAM can be implemented without any

restrictions on CNN architectures. However, GradCAM has mostly been adopted only for validation and visualization purposes after training the CNN model.

CNN interpretation has been found beneficial in various applications, such as 3D object recognition [52], diagnosis [53], human activity recognition [54], and metric learning [55]. In particular, CNN interpretation is crucial in industrial applications, which serves as a motivation for the current work. For instance, in [56], a visualized feature map is extracted using GradCAM to meet the requirements of process engineers. Similarly, GradCAM feature maps are also extracted for power equipment maintenance [57] and electromechanical system diagnosis [58]. Other visualization techniques have also been adopted. For instance, in [59], t-SNE is adopted to visualize the wafer defect maps. Our method differs from the above works because we not only use GradCAM for interpretation but also incorporate it as part of the training process to improve our neural network model.

III. APPROACH

We review autoencoders and their losses in §III-A. We then introduce our novel interpretability-aware loss in §III-B and our attention map used for anomaly detection in §III-C.

A. AUTOENCODER WITH STRUCTURAL SIMILARITY LOSS FOR IMAGE ANOMALY DETECTION

Given input images $\{\mathbf{x}^{(t)}\}_{t=1}^{N}$, in $\mathbb{R}^{c\times h\times w}$, the encoder E maps the images to low-dimensional latent variables $\{\mathbf{z}^{(t)}\}_{t=1}^{N} \subset \mathbb{R}^{d}$. The decoder D uses $\{\mathbf{z}^{(t)}\}_{t=1}^{N}$ to reconstruct the image, transforming $\mathbf{z}^{(t)}$ into $\hat{\mathbf{x}}^{(t)}$, which have the same dimensionality as and are similar to $\{\mathbf{x}^{(t)}\}_{t=1}^{N}$. Here, d represents the dimensionality of the latent variable $\{\mathbf{z}^{(t)}\}_{t=1}^{N}$, while c, h, and w respectively represent the numbers of channels, height, and width of the image. The parameters of E and D are learned by minimizing the reconstruction loss. Traditionally, the reconstruction error is computed using pixel-wise evaluation metrics, such as the ℓ^2 loss, to generate an anomaly score map based on the discrepancy between the input image and its reconstruction. However, it has been shown in [27] that incorporating a structural similarity loss in autoencoder architectures enhances the model's ability to capture inter-dependencies between image regions. Consequently, this approach effectively identifies complex structural defects in images.

The Structural Similarity Index (SSIM) [60] is a method used to compare two images to determine their similarity. It compares local patterns of pixel intensities in two images, denoted as $\mathbf{x}$ and $\mathbf{y}$, based on three components: luminance $l(\mathbf{x}, \mathbf{y})$, contrast $c(\mathbf{x}, \mathbf{y})$, and structure $s(\mathbf{x}, \mathbf{y})$. These components are defined by

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1}{\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1}, \tag{1}$$

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}} + C_2}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2}, \tag{2}$$

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{\mathbf{xy}} + C_3}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}} + C_3}, \tag{3}$$

respectively, where $\mu_{\mathbf{x}}$ denotes the average pixel luminance of the image $\boldsymbol{x}$, $\sigma_{\mathbf{x}}$ denotes the standard deviation of the pixel luminance of the image $\boldsymbol{x}$. Here, the constants C_1, C_2, and C_3 are included to avoid zero denominators, with C_3 set to $C_2/2$. The SSIM index is then defined as a function of l, c, and s, or more specifically,

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{\left(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1\right)\left(2\sigma_{\mathbf{xy}} + C_2\right)}{\left(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1\right)\left(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2\right)}. \tag{4}$$

B. GENERATING INTERPRETABILITY-AWARE LOSS

In this section, we introduce our Interpretability-Aware (IA) loss L_{IA}. Unlike the vanilla GradCAM [27], which backpropagates the CNN based on the score for a specific classification type, our proposed loss can be implemented for non-classification tasks. Specifically, we backpropagate the latent variable $\mathbf{z}$ generated by the encoder (i.e., $\mathbf{z} = E(\mathbf{x})$) until the gradient reaches a specified layer of the encoder. We remark that $\mathbf{z}$ can be any feature in a CNN, allowing our proposed loss to be applied flexibly to various CNN architectures. In addition to AE, we will also demonstrate its application in a classifier in our experiments.

To obtain the IA loss L_{IA} for an input image $\mathbf{x}$, encoded to a latent variable $\boldsymbol{z}$, we backpropagate the gradient of each entry z_i of $\boldsymbol{z}$, $i = 1, \cdots, d$, with respect to the feature maps $\mathbf{A}_j$ in the j-th layer of the encoder. This process generates the GradCAM attention map $\mathbf{M}_j$, which is obtained through a linear combination of feature maps $\mathbf{A}_j$ with ReLU activation:

$$\mathbf{M}_j = \mathrm{ReLU}\left(\sum_k \alpha_{ij}^k \mathbf{A}_j^k\right), \tag{5}$$

where k is the channel index. In (5), $\alpha_{ij}^k \in \mathbb{R}$ is obtained by applying the global average pooling operation to the gradient of z_i with respect to A_j^k:

$$\alpha_{ij}^k = \mathrm{AvgPool}\left(\frac{\partial z_i}{\partial \mathbf{A}_j^k}\right). \tag{6}$$

To ensure that the attention maps $\mathbf{M}_{ij}$ generated from different layers are comparable, we conduct bilinear interpolation upsampling operations to bring them to the same size of 256×256. The IA loss L_{IA} is derived from the upsampled attention maps. Specifically, we first calculate the mean of all the pixel values of the map $\mathbf{M}_{ij}$ as follows:

$$\mu = \sum_{i=1}^{d}\sum_{s=1}^{h}\sum_{t=1}^{w} \frac{M_{ij}^{st}}{dhw}, \tag{7}$$

and then incorporate it into the regularization term as follows:

$$L_{\mathrm{IA}} = \lambda\mu^2, \tag{8}$$

where λ is the regularization coefficient of the IA loss.

Note that due to the ReLU operation in (5), each pixel value M_{ij}^{st} is non-negative. Consequently, μ can be viewed

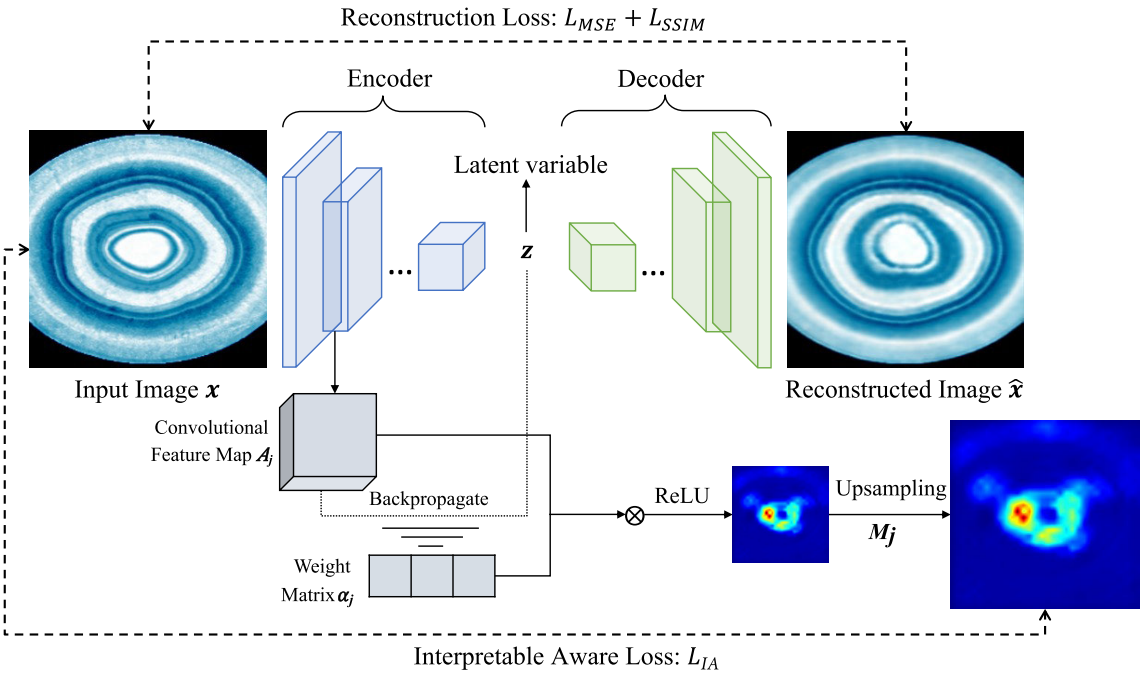

FIGURE 1. Illustration of the training stage of AE using both the reconstruction loss and the interpretability-aware loss. The upper part shows the AE, which is naturally endowed with the reconstruction loss. The lower part shows backpropagation of the latent variable with respect to the encoder layer and produces an attention map for calculating the interpretability-aware loss.

as a LASSO [61] term essentially, promoting sparsity in the attention map. This sparsity encourages the attention to focus on a small number of pixels, which is particularly important in industrial applications where defective regions in products are typically limited in size. It is worth noting that this approach differs from using an ℓ^2-like loss, which is commonly employed to prevent overfitting.

As illustrated in Fig. 1, training the AE entails minimizing the sum of the SSIM loss, the MSE loss, and the IA loss. To distinguish our approach from other AEs, we refer to our model as IAAE. Once trained, the IAAE exhibits focused attention that is utilized in anomaly detection, as explained in the next section. Algorithm 1 summarizes the steps for training the IAAE.

Algorithm 1 Training IAAE

Input: Training data $\{\mathbf{x}^{(i)}\}_{i=1}^{L}$; initialized parameters $\boldsymbol{\theta}$ of AE with encoder E and decoder D; number of epochs K; batch indices $\mathcal{I}$; learning rate α

Output: Trained parameters $\boldsymbol{\theta}$

for $k = 1, \cdots, K$ **do**
  for each batch $\{\mathbf{x}^{(i)}\}_{i\in\mathcal{I}}$ **do**
    $\mathbf{z}^{(i)} = E(\mathbf{x}^{(i)})$, $\hat{\mathbf{x}}^{(i)} = D(\mathbf{z}^{(i)})$
    $L_{\text{SSIM}} = \text{SSIM}(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(i)})$, according to (4)
    $L_{\text{MSE}} = |\mathcal{I}|^{-1} \sum_{i\in\mathcal{I}} \left\|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\right\|^2$
    Compute L_{IA} according to (5)–(8)
    $L = L_{\text{SSIM}} + L_{\text{MSE}} + L_{\text{IA}}$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L$
  end for
end for

C. GENERATING INTERPRETABILITY-AWARE ATTENTION MAP

For industrial products, the shooting conditions of their images may vary, making it inappropriate to retrieve a normal sample from the training stage and compare it directly with a test sample. To address this issue, we leverage an AE trained using an IA loss and utilize an Interpretability-Aware Attention Map (IAAM) for anomaly detection, as described below. It is important to note that IAAM differs from the GradCAM attention map discussed in the previous section. To be specific, IAAM focuses on the differences between a test sample and normal images and thus provides more suitable scores for anomaly detection. In contrast, GradCAM attention solely applies to the images themselves.

The first step in constructing the IAAM is to obtain a difference map between an input image $\mathbf{y}^*$ and the pixel-wise mean of all normal examples used in training, denoted as $\bar{\mathbf{x}}$. Given that our model is applied to images of industrial products, the prior assumption is that in each anomalous image, the area of the defect is concentrated and often small compared to the entire product image. To locate these small, compact defects more accurately, we amplify the differences between $\mathbf{y}^*$ and $\bar{\boldsymbol{x}}$. The detailed difference map is defined as

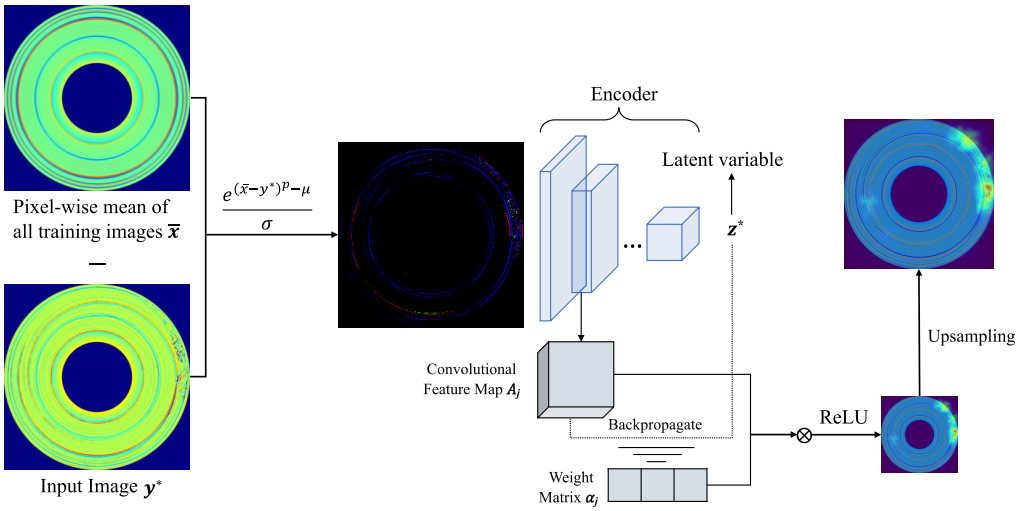

FIGURE 2. Illustration of the IAAM generating process. Each test image is compared with the pixel-wise mean of training images, which produces a difference map, used as input of the trained AE. The GradCAM attention map is then extracted from backpropagating the latent variable to the encoder layer. The pixel values of the attention map are then used for anomaly detection.

follows. For each $\mathbf{y}^*$, let i be an index for the pixels (ranging from 1 to $n = c \cdot h \cdot w$). We first calculate the mean of all the exponential differences

$$\mu^* = \frac{1}{n}\sum_{i=1}^{n} e^{(\bar{x}_i - y_i^*)^p}, \tag{9}$$

where the power p, applied to $\bar{x}_i - y_i^*$, is an energy measurement index, with a larger value of p indicating a greater emphasis on the differences. Similarly, we calculate the standard deviation of all the exponential differences as follows:

$$\sigma^* = \sqrt{\frac{\sum_{i=1}^{n}\left(e^{(\bar{x}_i - y_i^*)^p} - \mu^*\right)^2}{n-1}}. \tag{10}$$

At the end, we obtain the standardized image $\tilde{\mathbf{y}}^*$ whose entries are given by

$$\tilde{y}_i^* = \frac{e^{(\bar{x}_i - y_i^*)^p} - \mu^*}{\sigma^*}. \tag{11}$$

Once $\tilde{\mathbf{y}}^*$ is obtained, similarly to how we obtain the GradCAM attention map, we first encode it into a latent variable $\mathbf{z}^*$, and then backpropagate the gradient of $\mathbf{z}^*$ with respect to the feature maps $\mathbf{A}_j^*$ in one of the encoder layers to generate the interpretability-aware attention map $\mathbf{M}_j^*$. The procedure for obtaining the IAAM is summarized in Fig. 2.

The IAAM obtained through the above procedures can be used in anomaly detection tasks. Specifically, we use the sum of the pixel values in $\mathbf{M}_j^*$ as the anomaly score, which is compared to a threshold. If the anomaly score is larger than the threshold, then $\mathbf{y}^*$ is considered an anomalous sample. Algorithm 2 outlines the necessary steps for performing anomaly detection.

Algorithm 2 Generate IAAM for Anomaly Detection

Input: Test data $\{\mathbf{y}^{(j)}\}_{j=1}^{N}$; Training data $\{\mathbf{x}^{(i)}\}_{i=1}^{L}$; AE trained using Algorithm 1 with the encoder E; image-wise threshold ϵ_T; number of pixels n;
Output: Interpretability-Aware Attention Map $\mathbf{M}_j$

$\bar{\mathbf{x}} = L^{-1}\sum_{i=1}^{L}\mathbf{x}^{(i)}$
for $j = 1, \ldots, N$ **do**
  $\mu^{(j)} = n^{-1}\sum_{i=1}^{n} e^{(\bar{x}_i - y_i^{(j)})^p}$
  $\sigma^{(j)} = \sqrt{(n-1)^{-1}\sum_{i=1}^{n}\left(e^{(\bar{x}_i - y_i^{(j)})^p} - \mu^{(j)}\right)^2}$
  $\tilde{\mathbf{y}}^{(j)} = \sigma_j^{-1}\left(e^{(\bar{\mathbf{x}} - \mathbf{y}^{(j)})^p} - \mu^{(j)}\right)$ (broadcast operation)
  $\mathbf{z}^{(j)} = E(\tilde{\mathbf{y}}^{(j)})$
  if GlobalSumPool$(\mathbf{M}_j) \le \epsilon_T$ **then**
    $\mathbf{y}^{(j)}$ is a normal sample
  else
    $\mathbf{y}^{(j)}$ is an anomalous sample
  end if
end for

IV. EXPERIMENT RESULTS

We validate the effectiveness of our IA loss by visualizing the results of a simple task in §IV-A. We compare our model with baseline methods and discuss its performance in §IV-B. All experiments reported in this paper were conducted on a GPU server with NVIDIA GeForce RTX 3090 GPUs (24G memory).

A. QUALITATIVE VALIDATION OF THE IA LOSS

We first qualitatively validate the application of the IA loss by visualizing the attention maps from models trained by minimizing a loss function with and without an additional IA

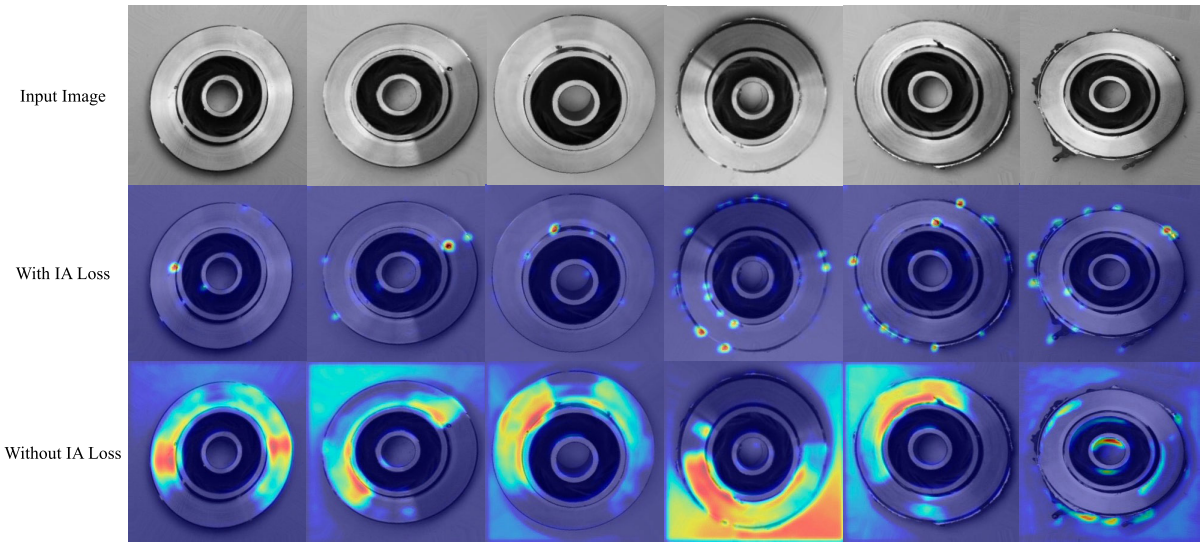

FIGURE 3. Visualization of examples from the Casting Dataset and their GradCAM attention maps. The first row shows the input test examples which represent defective products. The second and third rows show output attention maps from models trained with and without our proposed IA loss, respectively.

TABLE 1. Hyperparameters for training the ResNet-50 model.

Hyperparameter	Value
Learning rate	1×10^{-3}
Weight decay	1×10^{-5}
Batch size	128

TABLE 2. AUC scores for the Casting Dataset. We report the average score from 10 experiments with random initialization.

Loss	AUC Score
With IA loss	**0.975**
Without IA loss	0.964

loss, respectively. To this end, we first train a simple classifier using the above two alternatives and visualize the GradCAM attention maps. It is important to note that the GradCAM serves as an explanation of the classification results for practical users. By comparing the explanatory power of these attention maps, we can observe the benefits of our IA loss.

The dataset we use is the Casting Dataset [62], which consists of 7,348 grayscale images with dimensions of 300 × 300 pixels. The dataset primarily contains products from the casting manufacturing process. The training set consists of 2,875 normal images and 3,758 defect images, while the test set contains 262 normal images and 453 defect images. Defects in the dataset encompass various types, such as blow holes, pinholes, burr, shrinkage defects, mould material defects, pouring metal defects, metallurgical defects, and others. The inspection process for these products is typically carried out manually, which is time-consuming and subject to human error. Anomalies in this dataset typically manifest small areas within the product images, and there may be multiple similar defects within the same image.

Our focus is on validating the effectiveness of our proposed IA loss. To achieve this, we train two classification models which distinguish normal and anomalous examples, using the same architecture but different losses. The first model utilizes a standard binary cross entropy (BCE) loss in addition to our proposed IA loss, while the second model only employs the BCE loss. The classifier is taken to be a ResNet-50 model, trained from scratch using labeled data from the Casting Dataset. Table 1 displays the hyperparameters utilized during the training process.

In Table 2, we present the results of defect product detection using the area under the receiver operating characteristic curve (AUC) as the evaluation metric. We compare the performance of models trained with and without the IA loss. We observe from the table that, although the model trained with the IA loss performs better than the model without the IA loss, both models achieve high AUC scores. This implies that practical users may find both models effective in detecting defective products. However, in order to understand why a product is considered defective, an interpretability method needs to be applied. Next, we report the results obtained by observing the GradCAM attention maps for selected examples from the Casting Dataset.

Fig. 3 presents the examples from the test data in the first row, followed by the GradCAM attention maps obtained from models trained with the IA loss in the second row and without the IA loss in the third row. From the visualization, we observe notable differences between the two sets of attention maps. In the case of the model without IA loss, the attention areas appear large and ambiguous, indicating that the model may struggle to accurately identify the correct reason for the detection. Consequently, the results from this model may be deemed untrustworthy, providing no guidance for improving the manufacturing processes. In contrast, the attention maps generated by the model with IA loss exhibit more focused and localized hot areas. Comparing the first

TABLE 3. Architecture of the AE used in the experiment. The index of layers refers to the convolution layer. After each convolution layer, a LeakyReLU activation function with a slope of 0.2 is applied. For convolution layers 1-6 and 11-16, batch normalization (BatchNorm) is applied to the output activations.

Layer	Input shape	Output shape	Kernel	Stride	Padding
1	(256, 256, 3)	(64, 64, 32)	4×4	2	1
2	(64, 64, 32)	(32, 32, 32)	4×4	2	1
3	(32, 32, 32)	(16, 16, 32)	4×4	2	1
4	(16, 16, 32)	(16, 16, 32)	3×3	1	1
5	(16, 16, 32)	(8, 8, 64)	4×4	2	1
6	(8, 8, 64)	(8,8,64)	3×3	1	1
7	(8, 8, 64)	(4,4,128)	4×4	2	1
8	(4, 4, 128)	(4,4,64)	3×3	1	1
9	(4, 4, 64)	(4,4,32)	3×3	1	1
10	(4, 4, 32)	(1,1,100)	8×8	1	0
11	(1, 1, 100)	(4, 4, 32)	8×8	1	0
12	(4, 4, 32)	(4, 4, 64)	3×3	1	1
13	(4, 4, 64)	(4, 4, 128)	3×3	1	1
14	(4, 4, 128)	(8, 8, 64)	4×4	2	1
15	(8, 8, 64)	(8, 8, 64)	3×3	1	1
16	(8, 8, 64)	(16, 16, 32)	4×4	2	1
17	(16, 16, 32)	(16, 16, 32)	3×3	1	1
18	(16, 16, 32)	(32, 32, 32)	4×4	2	1
19	(32, 32, 32)	(64, 64, 32)	4×4	2	1
20	(64, 64, 32)	(256, 256, 3)	4×4	2	1

and second rows, we can observe that the anomaly areas more closely correspond to human intuition. Additionally, the model is capable of identifying multiple defects within a single image. Evidently, our proposed IA loss assists the model in effectively focusing on the true defect areas, thereby enhancing its trustworthiness.

B. QUANTITATIVE RESULTS FOR ANOMALY DETECTION

1) EXPERIMENTAL SET UP

In this section, we evaluate the effectiveness of our proposed method by training an AE using Algorithm 1 and obtaining the IAAM for anomaly detection based on Algorithm 2. During the training stage, only normal samples are utilized, while a combination of normal and anomalous samples is used for testing.

For our experiments, we utilize the BeanTech Anomaly Detection (BTAD) Dataset [63], which is an industrial anomaly detection dataset with pixel-level annotations. This dataset consists of RGB images representing three different industrial products, with 400 training images for Product 1, 1,000 training images for Product 2, and 399 training images for Product 3.

To facilitate our experiments, we crop the images into patches of size 256 × 256. The precise architecture of the autoencoder network used in all experiments is provided in Table 3. We employ the Adam optimizer [64] for training, and the specific hyperparameters utilized during the training stage are presented in Table 4.

We also implement other popular anomaly detection models using the same setting to serve as the benchmark, which includes One-Class Support Vector Machine (OC-SVM) [65], Local Outliers Factor (LOF) [66], K-Means [67], 1-Nearest Neighbors (1NN) [67], GANomaly [30], L2-AE [28], SSIM-AE [28]. For all neural network-based methods, we use the same hyperparameters for learning rate, batch size, etc.

TABLE 4. Training hyperparameters of the proposed model. The power of energy measurement index refers to the power used in equation (10).

Hyperparameter	Value
Learning rate	2×10^{-4}
Weight decay	1×10^{-5}
Regularization coefficient	1
Batch size	128
Epoch number	400
Backward layer	9
Energy measurement index	9
Regularization coefficient	1
Window size of SSIM loss	11

2) RESULTS

In our evaluation, we use the AUC metric to measure the performance of our proposed method and comparable methods, which is in line with previous works. Table 5 presents a comparison of the performance results for anomaly detection. We conduct the experiments using the same settings for three times and record the mean and standard deviation of the results for each run.

From the results, it is clear that our method excels the benchmarks for all three products. In particular, it is better than AE models trained without IA loss. At the same time, it is consistently better than models not based on AE, including traditional models and GAN-based models.

3) ANALYSIS ON TWO IMPORTANT FEATURES

To further explore and validate the influence of different features on the model performance, we conduct a sensitivity analysis to compare the effects caused by alternative choice of hyperparameters. Specifically, there are two important features to our model: first, the layers towards which the GradCAM backpropagates in the training and testing processes respectively; second, the energy measurement index i.e., the exponential order p used in (9)–(11) for computing the IAAM. Next, we discuss the effect of changing these two features and show the numerical results according to changes. In addition to our primary focus on detection, we present the pixel-wise results in this section to provide a more comprehensive analysis.

a: GradCAM USING BACKPROPAGATION to DIFFERENT CONVOLUTIONAL LAYERS

The focus of GradCAM varies depending on the layer to which it backpropagates. Ablation studies conducted in [27] suggest that deeper convolutional layers tend to capture more high-level and abstract features of the image, while shallow layers tend to capture more local and basic features. In our context, the selection of the convolutional layer involves a tradeoff during training. Choosing a deep layer sacrifices resolution since deep layer features have smaller sizes and require upsampling before producing the GradCAM attention maps. On the other hand, choosing a shallow layer sacrifices

TABLE 5. Image-level AUC results for the BTAD Dataset. We report the results for all the individual products, as well as the mean of all three products.

Product	OC-SVM	LOF	K-Means	1NN	GANomaly	L2-AE	SSIM-AE	IAAE (Ours)
1	0.798±0.007	0.489±0.003	0.765±0.003	0.786±0.008	0.825±0.005	0.731±0.000	0.826±0.005	**0.948**±0.018
2	0.382±0.002	0.420±0.001	0.392±0.010	0.591±0.005	0.556±0.013	0.477±0.008	0.547±0.021	**0.605**±0.004
3	0.647±0.007	0.487±0.004	0.411±0.004	0.627±0.010	0.436±0.003	0.291± 0.017	0.261±0.018	**0.665**±0.004
Mean	0.609± 0.005	0.465±0.003	0.522±0.006	0.668±0.007	0.605±0.007	0.549±0.008	0.528±0.015	**0.739**±0.009

TABLE 6. AUC scores for GradCAM using different convolutional layers on the BTAD Dataset. For each setting, there are two rows: the top row reports the pixel-wise score, and the bottom row reports the image-wise score.

Test layer \ Training layer	Encoder.0	Encoder.3	Encoder.6	Encoder.9	Encoder.12	Encoder.15	Encoder.22
Encoder.0	0.886 0.936	0.805 0.954	0.878 0.905	0.917 0.974	0.914 0.962	0.906 0.957	0.712 0.945
Encoder.3	0.875 0.934	0.750 0.513	0.864 0.907	0.897 0.936	0.901 0.924	0.904 0.968	0.772 0.948
Encoder.6	0.875 0.934	0.776 0.845	0.856 0.927	0.894 0.943	0.864 0.810	0.891 0.955	0.705 0.908
Encoder.9	0.883 0.935	0.812 0.943	0.859 0.948	0.882 0.935	0.850 0.868	0.830 0.862	0.555 0.768
Encoder.12	0.884 0.936	0.790 0.901	0.874 0.922	0.886 0.910	0.870 0.850	0.878 0.925	0.627 0.919
Encoder.15	0.885 0.936	0.784 0.753	0.870 0.932	0.883 0.928	0.854 0.756	0.865 0.877	0.569 0.745
Encoder.22	0.886 0.938	0.837 0.914	0.910 0.940	0.921 0.965	0.913 0.967	0.910 0.978	0.783 0.948

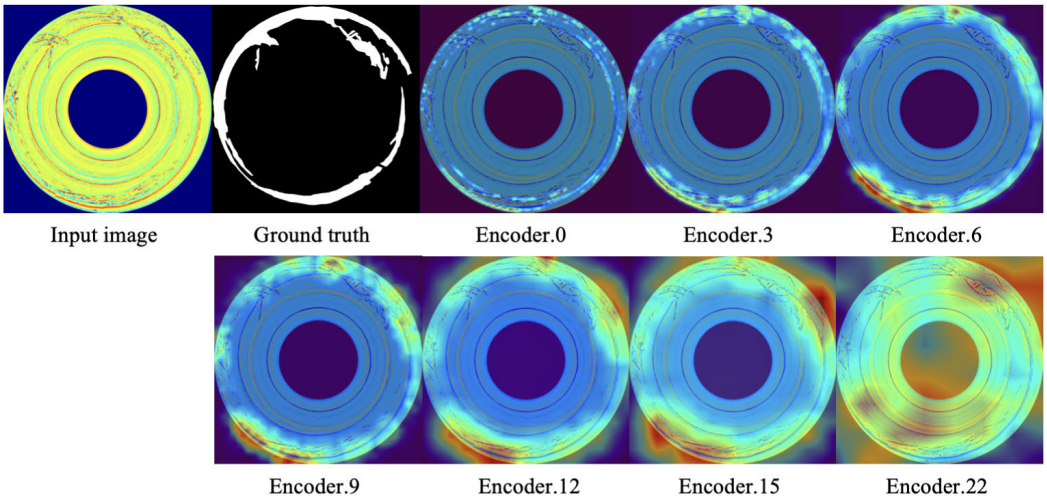

FIGURE 4. Illustration of heat maps for different layers towards which GradCAM backpropagates in the testing stage. The training layer is Encoder.9.

explanatory power since the features contain fewer semantics. Therefore, we expect a layer in the middle to be most suitable for our task.

To validate our choice, we compare alternative models by adjusting the layers used for GradCAM backpropagation during the training and testing stages, while keeping the other hyperparameters the same. We consider convolutional layers 0, 3, 6, 9, 12, 15, and 22. Table 6 displays the results for Product 1 of the BTAD Dataset. The quantitative comparison suggests that the choice of the convolutional layer does impact the model's performance, but it is not very sensitive, especially with respect to the layer used in the testing phase. Regarding the training phase, models trained on deeper convolutional layers generally perform better in terms of localization and classification tasks, indicating that deeper layers contain more useful semantics. However, using a very deep layer (Encoder.22) for training and extracting the attention map results in significantly worse performance due to the very low resolution of the attention map.

To ensure that our method is explainable during the testing phase, we visualize the performance of the alternative models in Fig. 4, while fixing the layer used during training to be Encoder.9. Testing on a shallower convolutional layer has the advantage of focusing on a smaller and concentrated area to depict the defects in the generated abnormal area. However, it may also mistakenly narrow down the estimated

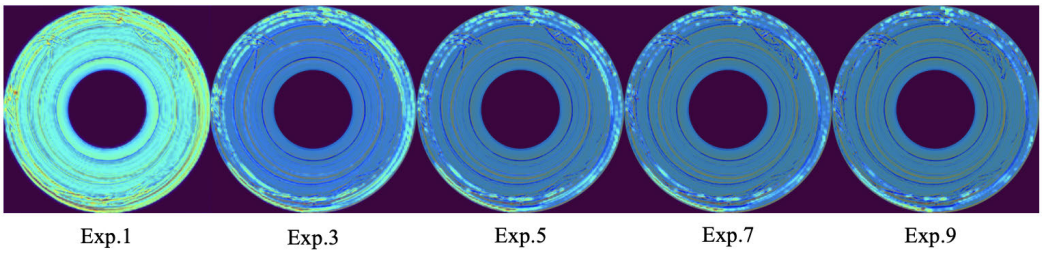

FIGURE 5. Illustration of heat maps for different energy measurement indices in the testing stage.

TABLE 7. AUC scores for anomaly detection using different energy measurement index *p*, validated on Product 1 of the BTAD Dataset. For each *p*, the top row reports the pixel-wise score, and the bottom row reports the image-wise score.

p	AUC Score
1	0.823 0.844
3	0.867 0.887
5	0.883 0.914
7	0.889 0.933
9	0.917 0.974

abnormal area when the defects are actually large. Therefore, to ensure that our model provides good interpretability when users examine the attention map, we choose the Encoder.9 convolutional layer for training and the Encoder.0 layer for visualization.

b: ENERGY MEASUREMENT INDEX

The energy measurement index p in (9)–(11) affects the concentration of the attention heat map generated by GradCAM during the testing phase. By increasing the exponential order p, the gradient becomes more polarized as it amplifies the already high gradients and increases the distance between these high gradients and the lower ones. Consequently, the GradCAM heat map exhibits a more concentrated hot area since it is derived from these gradients. This specific design aims to enhance the capability of IAAM in accurately identifying the defective part of abnormal industrial data. Furthermore, since the objective is to increase differentiation among pixels, we restrict p to odd numbers. The validation results for Product 1 of the BTAD Dataset using different values of p are presented in Table 7, and the corresponding heatmaps are visualized in Figure 5. It is evident that as the energy measurement index increases, both the pixel-level and image-level anomaly detection accuracy scores improve. This validates our choice of using a larger value of $p = 9$ for achieving better detection results and interpretability.

V. CONCLUSION

In this paper, we have presented an interpretable deep learning-based algorithm for the detection of anomalies in industrial products. Our algorithm leverages the capabilities of neural networks for anomaly detection while ensuring model interpretability, making it suitable for industrial users who require actionable insights. The experimental results have demonstrated that our algorithm surpasses the performance of baseline anomaly detection methods in terms of accuracy and interpretability. Particularly, the attention maps generated by our algorithm offer valuable insights into its functioning and can be leveraged to enhance its performance.

While our proposed algorithm holds significant potential for various industrial applications such as quality control, product inspection, and defect prevention, we would like to acknowledge two potential limitations. Firstly, different types of data may necessitate the adjustment of hyperparameters, which should be considered alongside the selection of an appropriate threshold during practical implementation. Secondly, our model utilizes GradCAM attention maps twice, both during training and testing, which may introduce additional computational complexity.

In the future, our focus will be on enhancing the scalability of our algorithm to handle larger datasets with more complex anomalies. We will also extend our investigations to other types of data, including audio or sensor data, where interpretability is equally vital. Additionally, we will explore how our interpretability-aware algorithm can foster effective collaboration between humans and machines in industrial settings.

ACKNOWLEDGMENT

(Rui Jiang and Yijia Xue contributed equally to this work.)

REFERENCES

[1] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *Proc. 20th Nat. Inf. Syst. Secur. Conf.*, vol. 377, Baltimore, MD, USA, 1997, pp. 366–380.

[2] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Inf. Sci.*, vol. 177, no. 18, pp. 3799–3821, Sep. 2007.

[3] S. Omar, A. Ngadi, and H. H. Jebur, "Machine learning techniques for anomaly detection: An overview," *Int. J. Comput. Appl.*, vol. 79, no. 2, pp. 33–41, Oct. 2013.

[4] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Netw.*, vol. 8, no. 3, pp. 26–41, May 1994.

[5] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.

[6] T. Xiang and S. Gong, "Video behavior profiling for anomaly detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 893–908, May 2008.

[7] A. Taboada-Crispi, H. Sahli, D. Hernandez-Pacheco, and A. Falcon-Ruiz, "Anomaly detection in medical image analysis," in *Handbook of Research on Advanced Techniques in Diagnostic Imaging and Biomedical Applications*. Hershey, PA, USA: IGI Global, 2009, pp. 426–446.

[8] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Proc. Comput. Sci.*, vol. 60, pp. 708–713, Jan. 2015.

[9] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[10] S. Matteoli, M. Diani, and G. Corsini, "A tutorial overview of anomaly detection in hyperspectral images," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 5–28, Jul. 2010.

[11] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9584–9592.

[12] G. Pang, C. Shen, L. Cao, and A. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2020.

[13] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Comput.*, vol. 22, no. S1, pp. 949–961, Jan. 2019.

[14] H. Yang, Y. Chen, K. Song, and Z. Yin, "Multiscale feature-clustering-based fully convolutional autoencoder for fast accurate visual inspection of texture surface defects," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1450–1467, Jul. 2019.

[15] P. Lall, P. Gupta, and A. Angral, "Anomaly detection and classification for PHM of electronics subjected to shock and vibration," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 2, no. 11, pp. 1902–1918, Nov. 2012.

[16] A. Goode, R. Sukthankar, L. Mummert, M. Chen, J. Saltzman, D. Ross, S. Szymanski, A. Tarachandani, and M. Satyanarayanan, "Distributed online anomaly detection in high-content screening," in *Proc. 5th IEEE Int. Symp. Biomed. Imag., From Nano Macro*, May 2008, pp. 249–252.

[17] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Bayesian skip-autoencoders for unsupervised hyperintense anomaly detection in high resolution brain mri," in *Proc. IEEE 17th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2020, pp. 1905–1909.

[18] G. Park, M. Lee, H. Jang, and C. Kim, "Thermal anomaly detection in walls via CNN-based segmentation," *Autom. Construct.*, vol. 125, May 2021, Art. no. 103627.

[19] B. Staar, M. Lutjen, and M. Freitag, "Anomaly detection with convolutional neural networks for industrial surface inspection," *Proc. CIRP*, vol. 79, pp. 484–489, Jan. 2019.

[20] Z. Tang, Z. Chen, Y. Bao, and H. Li, "Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring," *Struct. Control Health Monitor.*, vol. 26, no. 1, p. e2296, Jan. 2019.

[21] Z. Shi, X. Yu, Z. Jiang, and B. Li, "Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4511–4523, Aug. 2014.

[22] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The MVTec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 1038–1059, Apr. 2021.

[23] L. Puggini and S. McLoone, "An enhanced variable selection and isolation forest based methodology for anomaly detection with OES data," *Eng. Appl. Artif. Intell.*, vol. 67, pp. 126–135, Jan. 2018.

[24] B. Hayes and J. A. Shah, "Interpretable models for fast activity recognition and anomaly explanation during collaborative robotics tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6586–6593.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[26] N. Cao, C. Lin, Q. Zhu, Y. Lin, X. Teng, and X. Wen, "Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 23–33, Jan. 2018.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[28] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 372–380.

[29] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *Proc. Wireless Telecommun. Symp. (WTS)*, Apr. 2018, pp. 1–5.

[30] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Computer Vision—ACCV*. Berlin, Germany: Springer, 2018, pp. 622–637.

[31] P. Mishra, C. Piciarelli, and G. L. Foresti, "A neural network for image anomaly detection with deep pyramidal representations and dynamic routing," *Int. J. Neural Syst.*, vol. 30, no. 10, Oct. 2020, Art. no. 2050060.

[32] P. Mishra, C. Piciarelli, and G. L. Foresti, "Image anomaly detection by aggregating deep pyramidal representations," in *Proc. Int. Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2021, pp. 705–718.

[33] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proc. ICML*, 2012, pp. 37–49.

[34] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3379–3388.

[35] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," 2018, *arXiv:1802.06360*.

[36] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Muller, and M. Kloft, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[37] K.-L. Li, H.-K. Huang, S.-F. Tian, and W. Xu, "Improving one-class SVM for anomaly detection," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2003, pp. 3077–3081.

[38] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.

[39] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.

[40] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*. Berlin, Germany: Springer, 2002, pp. 170–180.

[41] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. MLSDA 2nd Workshop Mach. Learn. Sensory Data Anal.*, Dec. 2014, pp. 4–11.

[42] C.-H. Lai, D. Zou, and G. Lerman, "Robust subspace recovery layer for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–28.

[43] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *Proc. 16th Eur. Conf.* Glasgow, U.K.: Springer, 2020, pp. 485–503.

[44] C.-H. Lai, D. Zou, and G. Lerman, "Robust variational autoencoding with Wasserstein penalty for novelty detection," in *Proc. 26th Int. Conf. Artif. Intell. Statist.*, vol. 206, F. Ruiz, J. Dy, and J.-W. Van De Meent, Eds. Apr. 2023, pp. 3538–3567.

[45] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. Camps, "Towards visually explaining variational autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8639–8648.

[46] H. Wang, X. Wu, Z. Huang, and E. P. Xing, "High-frequency component helps explain the generalization of convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8681–8691.

[47] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[48] X. Bai, X. Wang, X. Liu, Q. Liu, J. Song, N. Sebe, and B. Kim, "Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments," *Pattern Recognit.*, vol. 120, Dec. 2021, Art. no. 108102.

[49] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14298–14308.

[50] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

[51] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10697–10706.

[52] N. U. Islam and S. Lee, "Interpretation of deep CNN based on learning feature reconstruction with feedback weights," *IEEE Access*, vol. 7, pp. 25195–25208, 2019.
[53] S. Wang, Y. Yin, D. Wang, Y. Wang, and Y. Jin, "Interpretability-based multimodal convolutional neural networks for skin lesion diagnosis," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12623–12637, Dec. 2022.
[54] E. Kim, "Interpretable and accurate convolutional neural networks for human activity recognition," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 7190–7198, Nov. 2020.
[55] M. Amirian and F. Schwenker, "Radial basis function networks for convolutional neural networks to learn similarity distance metric and improve interpretability," *IEEE Access*, vol. 8, pp. 123087–123097, 2020.
[56] C. Chien, W. Hung, and E. T. Liao, "Redefining monitoring rules for intelligent fault detection and classification via CNN transfer learning for smart manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 35, no. 2, pp. 158–165, May 2022.
[57] S. Mantach, P. Gill, D. R. Oliver, A. Ashraf, and B. Kordi, "An interpretable CNN model for classification of partial discharge waveforms in 3D-printed dielectric samples with different void sizes," *Neural Comput. Appl.*, vol. 34, no. 14, pp. 11739–11750, Jul. 2022.
[58] F. Arellano-Espitia, M. Delgado-Prieto, V. Martínez-Viol, J. Saucedo-Dorantes, and R. A. Osornio-Rios, "Diagnosis electromechanical system by means CNN and SAE: An interpretable-learning study," in *Proc. IEEE 5th Int. Conf. Ind. Cyber-Physical Syst. (ICPS)*, May 2022, pp. 1–6.
[59] S. Wang, Z. Zhong, Y. Zhao, and L. Zuo, "A variational autoencoder enhanced deep learning model for wafer defect imbalanced classification," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 11, no. 12, pp. 2055–2060, Dec. 2021.
[60] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[61] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., Ser. B*, vol. 58, no. 1, pp. 267–288, Jan. 1996.
[62] R. Dabhi, "Casting product image data for quality inspection," Kaggle. Accessed: May 31, 2022. [Online]. Available: https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product
[63] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "VT-ADL: A vision transformer network for image anomaly detection and localization," in *Proc. IEEE 30th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2021, pp. 01–06.
[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[65] Y. Tian, M. Mirzabagheri, S. M. H. Bamakan, H. Wang, and Q. Qu, "Ramp loss one-class support vector machine; A robust and effective approach to anomaly detection problems," *Neurocomputing*, vol. 310, pp. 223–235, Oct. 2018.
[66] Z. Cheng, C. Zou, and J. Dong, "Outlier detection using isolation forest and local outlier factor," in *Proc. Conf. Res. Adapt. Convergent Syst.*, Sep. 2019, pp. 161–168.
[67] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.

RUI JIANG received the dual B.S. degree in data science from Duke Kunshan University and Duke University, in 2023. She is currently pursuing the master's degree in electrical and computer engineering with Duke University. Her research interests include leveraging machine learning and artificial intelligence techniques to enhance data analysis and facilitate more effective decision-making processes.

YIJIA XUE received the dual B.S. degree in data science from Duke Kunshan University and Duke University, in 2023. She is currently pursuing the master's degree in data science with Brown University. Her research interests include advancing the understanding and ethical implications of artificial intelligence, she is dedicated to exploring innovative approaches that promote transparency, interpretability, and fairness in machine learning models.

DONGMIAN ZOU (Member, IEEE) received the B.S. degree (Hons.) in mathematics from The Chinese University of Hong Kong, in 2012, and the Ph.D. degree in applied mathematics and scientific computation from the University of Maryland, College Park, in 2017.

From 2017 to 2020, he was a Postdoctoral Researcher with the Institute for Mathematics and its Applications and the School of Mathematics, University of Minnesota, Twin Cities. He joined Duke Kunshan University, in 2020, where he is currently an Assistant Professor in data science with the Division of Natural and Applied Sciences. He is also affiliated with the Zu Chongzhi Center for Mathematics and Computational Sciences (CMCS) and the Data Science Research Center (DSRC). His research interests include the intersection of applied harmonic analysis, machine learning, and signal processing.