IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 March 2024, accepted 17 May 2024, date of publication 28 May 2024, date of current version 4 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406376

RESEARCH ARTICLE

Anomaly Detection Using Normalizing Flow-Based Density Estimation and Synthetic Defect Classification

SEUNGMI OH AND JEONGTAE KIM, (Member, IEEE)
Department of Electronic and Electrical Engineering, Ewha Womans University, Seoul 03760, South Korea
Graduate Program in Smart Factory, Ewha Womans University, Seoul 03760, South Korea

Corresponding author: Jeongtae Kim (jtkim@ewha.ac.kr)

This work was supported by the National Research Foundation (NRF), South Korea, through Project Brain Korea 21 Fostering Outstanding Universities for Research (BK21 FOUR) and a grant funded by Korean Government [Ministry of Science and ICT (MSIT)], South Korea, under Grant 2022R1F1A1075009 and Grant RS-2023-00252007.

ABSTRACT We propose a novel deep learning-based anomaly detection (AD) system that combines a pixelwise classification network with conditional normalizing flow (CNF) networks by sharing feature extractors. We trained the pixelwise classification network using synthetic abnormal data to fine-tune a pretrained feature extractor of the CNF networks, thereby learning the discriminative features of the in-domain data. After that, we trained the CNF networks using normal data with the fine-tuned feature extractor to estimate the density of normal data. During inference, we detected anomalies by calculating the weighted average of the anomaly scores from the pixelwise classification and CNF networks. Because the proposed system not only has learned the properties of in-domain data but also aggregated the anomaly scores of the classification and CNF networks, it showed significantly improved performance compared to existing methods in experiments using the MvTecAD and BTAD datasets. Moreover, the proposed system does not increase computations intensively since the classification and the density estimation systems share feature extractors.

INDEX TERMS Industrial inspection, machine vision, deep learning, anomaly detection, synthetic defect generation, density estimation, normalizing flow network, fine-tuning network.

I. INTRODUCTION

To detect abnormal (or defective) samples using supervised learning, a sufficient number of abnormal samples are required to train a network. However, it is often difficult to acquire sufficient abnormal data because they rarely occur for events such as quality control for industrial inspection [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], video surveillance [19], [20], disease diagnosis using medical images [21], [22], [23], [24], and abnormal electrocardiogram vital-sign detection [25], [26]. Therefore, anomaly detection (AD) methods, which train a network using only normal data and detect test samples deviating from the normal data as anomalies, have been widely studied [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]. As a matter of fact, AD is very challenging because a network trained using only normal data may have difficulty detecting anomalies similar to the normal data.

Methods for AD can be categorized into reconstruction-based [1], [2], [3], [4], [5], [6], [7], [8], [19], [20], [21], [22], [23], distance-based (or embedding-based) [12], [13], [14], [15], [16], [17], [18], [27], and anomaly simulation-based (or self-supervised learning-based) [8], [9], [10], [11], [24], [28] methods. Reconstruction-based methods train a network to reconstruct the input using only normal data and identify anomalies based on reconstruction errors.

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Huang.

The idea behind these methods is that the network does not reconstruct abnormal data well because it was trained using only normal data. Many researchers have investigated these methods because they are simple and intuitive [8], [12]. However, reconstruction-based methods have struggled to detect abnormal data because they often reconstruct the abnormal data well [29], [30], [31]. Many studies [3], [4], [5], [6] have referred to this as an overgeneralization problem and have endeavored to solve it.

Distance-based methods estimate the density of the features of normal data and determine samples whose statistical distances are larger than a certain threshold as abnormal samples. Usually, these methods [12], [13], [14], [15], [16], [17], [18] utilize a pretrained network for classification of a large-scale dataset (e.g., ImageNet [32]) as feature extractors. Thanks to the discriminative features of the pretrained network, recent studies [15], [16], [17], [18] reported that the methods show satisfactory performance for some applications. However, we strongly believe that the pretrained network is not effective in extracting suitable features for AD for inspection of industrial products because of the large domain gap between the industrial product and ImageNet datasets.

To obtain discriminative features of the in-domain data, anomaly simulation-based [8], [9], [10], [11], [24], [28] methods have been studied to approximate real-world abnormal data by generating synthetic abnormal data with spatial irregularities. These methods train a network to classify normal and synthetic abnormal data at the image or pixel levels. Therefore, the network can elaborately identify abnormal regions because it learns the discriminative features of the in-domain data. However, they are limited because synthetic abnormal data may not represent all the manifestations of abnormal data in the real world. This causes an overfitting problem, in which the network struggles to detect anomalies in the test phase owing to biases in the synthetic abnormal data [11].

Considering the limitations of existing AD methods, we propose a method that combines a pixelwise classification network with conditional NF (CNF) networks [33] by sharing feature extractors to enhance performance for AD. In addition, to effectively obtain discriminative features from the in-domain data, we propose a hybrid training algorithm that trains the pixelwise classification network using synthetic abnormal data and trains the CNF networks using only normal data. We expect the proposed method to show improved performance for two reasons. First, we expect to improve the performance of the CNF networks because of the discriminative features of the in-domain data from the fine-tuned feature extractor. Second, we believe that aggregating the predictions of the pixelwise classification and the CNF networks enhances the performance because of the positive impact of network ensembles [34], [35], [36] and collaborative effects. In addition, the proposed method showed better performance than existing methods of AD without significantly increasing the complexity of the network thanks to the sharing of feature extractors of the pixelwise classification and CNF networks. We empirically demonstrate our arguments using various metrics and extensive experiments on the MVTec AD [37] and Bean-Tech AD (BTAD [38]) datasets.

The main contributions of this study are summarized as follows:

- To the best of our knowledge, this study is the first attempt to enhance the performance of AD by fine-tuning the feature extractor of CNF networks by pixelwise classification using synthetic abnormal data. We experimentally demonstrated that the fine-tuned feature extractor is helpful in the extraction of valuable features for AD, owing to its discriminative ability and in-domain knowledge. In addition, we confirmed that the fine-tuned feature extractor by pixelwise regression degrades performance for AD because it extracts only normal features, even though a test image has anomalous regions.
- We propose a network architecture that combines a pixelwise classification network and CNF networks through a shared feature extractor to improve performance of AD. Moreover, to utilize their collaborative effects, we propose a hybrid training algorithm that trains the pixelwise classification network using synthetic abnormal data and trains CNF networks using only normal data. To the best of our knowledge, combining the pixelwise classification network and CNF networks with the shared feature extractor for performance improvements has not been attempted for AD.

The remainder of this paper is organized as follows. Section II reviews existing AD methods. Section III describes the proposed method in detail. Section IV presents the results that empirically demonstrate our arguments through extensive experiments. Finally, Section V summarizes and concludes the study.

II. RELATED WORK

A. RECONSTRUCTION-BASED METHODS

Reconstruction-based methods identify abnormal data based on reconstruction errors from a network trained to reconstruct the input using only normal data. The underlying concept of these methods is that representations learned only from normal data may not describe the features of abnormal data [29]. Many researchers [1], [2], [19], [20], [22] have extensively utilized autoencoder structures and generative networks because of their simplicity. Autoencoder structures learn intrinsic features of normal data by compressing an input image into low-dimensional embeddings to prevent the copying of the input [1]. Generative networks for AD are trained to generate normal data and detect anomalies based on the quality of the generated fake images [2], [19], [20], [21], [22]. However, these methods have an overgeneralization problem in that the network reconstructs or generates abnormal data well. Therefore, some researchers [3], [4], [5], [6] have attempted to utilize in-painting methods

for AD, expecting the network to consider the context of the data to alleviate the overgeneralization problem.

Despite these attempts, researchers of the discriminatively trained reconstruction anomaly embedding model (DRAEM) [8] and masked swin transformer U-net (MSTUNet) [7] have argued that reconstructive networks trained with only normal data still have the overgeneralization problem. They attempted to mitigate the overgeneralization problem by utilizing synthetic abnormal data, thereby improving the performance. Moreover, to prevent the problem of overfitting to synthetic anomalies, they sequentially trained a reconstructive network to reconstruct normal data from synthetic abnormal data and a discriminative network to predict synthetic abnormal regions from the concatenated input of the reconstructed normal and synthetic abnormal data. However, we believe that these methods still suffer from the overfitting problem because the final prediction of the system relies strongly on synthetic abnormal data.

B. DISTANCE-BASED METHODS

Distance-based methods estimate the density of the normal features extracted from a pretrained network for classification on a large-scale dataset (e.g., ImageNet [32]) and identify abnormal samples based on statistical distances from the normal data. Researchers have hypothesized that the discriminative ability of the pretrained network is also helpful for AD [27]. The patch distribution modeling method (PaDiM) [12] is a method that estimates the distribution of each normal feature vector as a multivariate Gaussian distribution without an additional training process for AD. Some researchers [13], [14], [15] have leveraged an NF network consisting of a sequence of invertible transformations with parameters determined by small network blocks to estimate the complex distribution of normal features [33], [39]. The CNF framework for AD (CFlow-AD) [15] utilizes CNF networks [33] to estimate the distribution based on the position information to improve the performance at the pixel level. However, we believe that the pretrained feature extractor is not effective in providing suitable features for AD to CNF networks because of the domain gap between the ImageNet and in-domain datasets.

In addition, some researchers [16], [17] applied the knowledge distillation framework to AD by considering the pre-trained network as a teacher network and training a student network to mimic the feature maps of the teacher network for normal data. Subsequently, anomalies were detected based on the distance between the feature embeddings of the teacher and student networks. The collaborative discrepancy optimization (CDO) method [18] utilizes synthetic abnormal data that are randomly generated by inserting rectangular patches with random values sampled from a Gaussian normal distribution into normal data. It trains the student network to enforce l2 distances between the features extracted from the student and teacher networks such that they are sufficiently small for normal data and sufficiently large for abnormal data. Similar to focal loss [40], this method utilizes an overlap optimization module to correct the outliers in normal and synthetic abnormal data during training. However, we argue that these methods are not effective at considering in-domain knowledge because they train the student network to mimic the pretrained network on the ImageNet dataset. In addition, we presume that CDO may be susceptible to overfitting with synthetic abnormal data because it trains the student network to discern synthetic abnormal regions from normal regions.

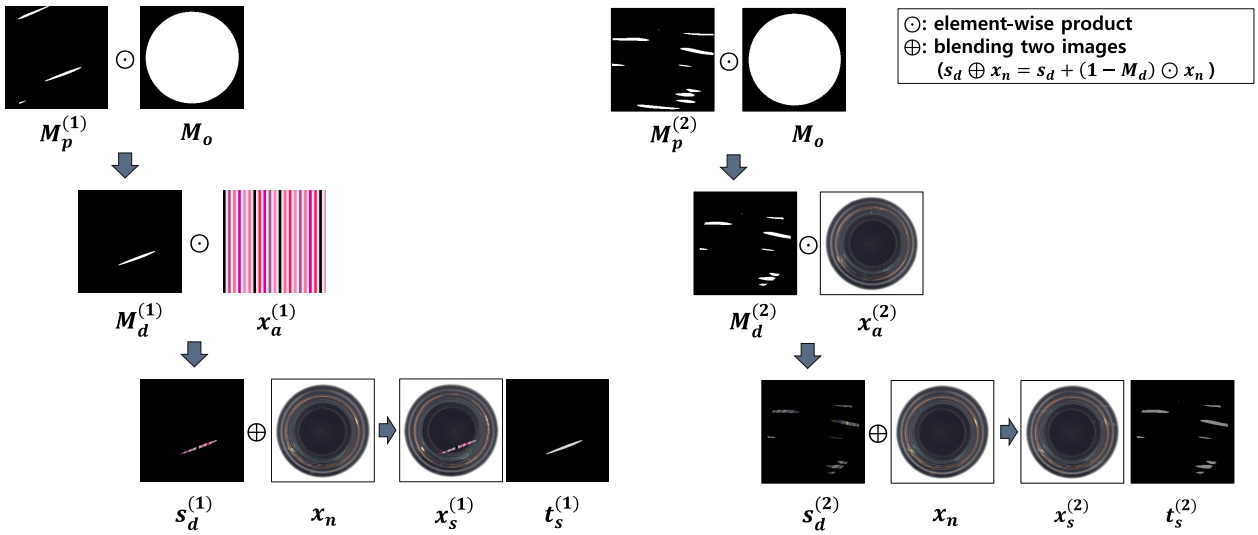
C. ANOMALY SIMULATION-BASED METHODS

Anomaly simulation-based methods generate synthetic anomalous data using normal data and train a network to classify them. Researchers who have studied these methods generated synthetic abnormal data by applying random transformations to normal data [9], [10], [11], [24], assuming that the main characteristic of the anomalies is spatial irregularity. Therefore, these methods are also known as self-supervised learning-based methods because they learn data representations by predicting the geometric transformations of normal data [10], [11]. CutOut [9] randomly selects rectangular patches of an input image and fills them in gray. However, the appearance of the synthetic abnormal data generated by CutOut is unrealistic and monotonous. To generate realistic synthetic abnormal data, CutPaste [10] randomly selects patches of a normal image and inserts them into random regions of the image. Furthermore, some researchers have proposed the natural synthetic anomaly generation method (NSA [11]) using the Poisson image editing method, which seamlessly blends one image with another.

In addition to these methods, some researchers [8], [28] have utilized another dataset to generate synthetic anomalous data. DRAEM [8] proposed a method for generating defect segments by applying a mask generated using Perlin noise [41] to an image from a describable texture dataset (DTD [42]). The memory-based segmentation network (MemSeg [28]) developed this method to generate more realistic synthetic defect data by adding the binarization process of the image to separate the background from an object for inspection. A network trained with synthetic data generated using these methods can improve performance for AD because it learns features distinct from normal data. However, the network has an overfitting problem because synthetic abnormal data may not cover the appearance of abnormal data in the real world [11].

III. PROPOSED METHOD

This section introduces the proposed method that combines a pixelwise classification network with CNF networks by sharing feature extractors. We explain the synthetic defect data generation method used to train the pixelwise classification network, the network architecture of the proposed method, and the hybrid training algorithm and score map aggregation method used to enhance performance for AD.



(a) Synthetic defects with apparent differences

(b) Synthetic defects with subtle differences

FIGURE 1. Synthetic defect data generation process in the proposed method.

A. SYNTHETIC DEFECT DATA GENERATION METHOD

We generated a synthetic defect dataset to train the pixelwise classification network using MemSeg's synthetic defect image generation method [28] and a label-smoothing strategy inspired by NSA [11]. In Fig.1, we describe the synthetic defect data generation process based on the extent of the differences in appearance between a synthetic defect image and a normal image. Fig.1(a) shows the generation process for synthetic defect data with apparent differences using data ($\mathbf{x_a}^{(1)}$) from another dataset (e.g., DTD [42]). Fig.1(b) illustrates the generation process for synthetic defect data with subtle differences using other normal data ($\mathbf{x_a}^{(2)}$) from the in-domain dataset.

To generate the synthetic defect data, we first generated a pattern mask ($\mathbf{M_p}$) using Perlin noise [41] to create irregularities in the defect patterns [8], [28]. In addition, we obtained an object mask ($\mathbf{M_o}$) by applying Otsu's method [43] and morphological operations to the grayscale normal image [28]. To prevent the generation of synthetic defects in the background, we produced a synthetic defect mask ($\mathbf{M_d}$) by applying an element-wise product of the pattern and object masks [28]. The synthetic defect regions ($\mathbf{s_d}$) were randomly sampled by masking an image ($\mathbf{x_a}$) that was used as the source of defect regions with the defect mask [8], [28]. Subsequently, we generated synthetic defect data ($\mathbf{x_s}$) by blending these regions ($\mathbf{s_d}$) with the normal data ($\mathbf{x_n}$).

To assign the ground truth for the synthetic defect data, we generated soft labels using the label-smoothing approach inspired by NSA [11] to consider the extent of abnormal appearances. One-hot encoding, which is typically used as the ground-truth probability, can be unsuitable for synthetic defect data because of the inconsistent quality of randomly generated synthetic defects. To generate soft labels ($\mathbf{t_s}^{(1)}$, $\mathbf{t_s}^{(2)}$), we first normalized the input defect-free image ($\mathbf{x_n}$) and the synthetic defect image ($\mathbf{x_s}$) between 0 and 1, and calculated the differences in the RGB values. Subsequently, we generated the pixelwise ground-truth probability of the synthetic defects by applying the sigmoid function to the averaged values of the color differences. Therefore, the network can learn to distinguish synthetic defect regions with subtle ($\mathbf{x_s}^{(2)}$) and apparent ($\mathbf{x_s}^{(1)}$) changes, as shown in Fig.1. We expected this to prevent the network from making incorrect predictions with high confidence during the testing phase.

B. NETWORK ARCHITECTURE

1) OVERALL NETWORK ARCHITECTURE

We describe the overall network architecture of the proposed method in Fig.2. The network used in the proposed method consists of an encoder-decoder network for pixelwise classification and CNF networks for estimating the distribution of normal data. We used this network architecture for the following two reasons: First, the feature extractor fine-tuned by pixelwise classification using synthetic defect data improves the performance of the CNF networks because of its discriminative abilities and domain gap reduction. Second, the aggregated prediction of the pixelwise classification and CNF networks improves the performance because of their collaborative effects and the positive effects of network ensembles [34], [35], [36].

We first constructed a pixelwise classification network to fine-tune the pretrained network on the ImageNet dataset to obtain in-domain knowledge and discriminative abilities. We propagated multilevel feature maps with three different resolutions from the encoder to the decoder using skip connections [44]. In addition, to identify defects with small regions and structural differences, we used three independent

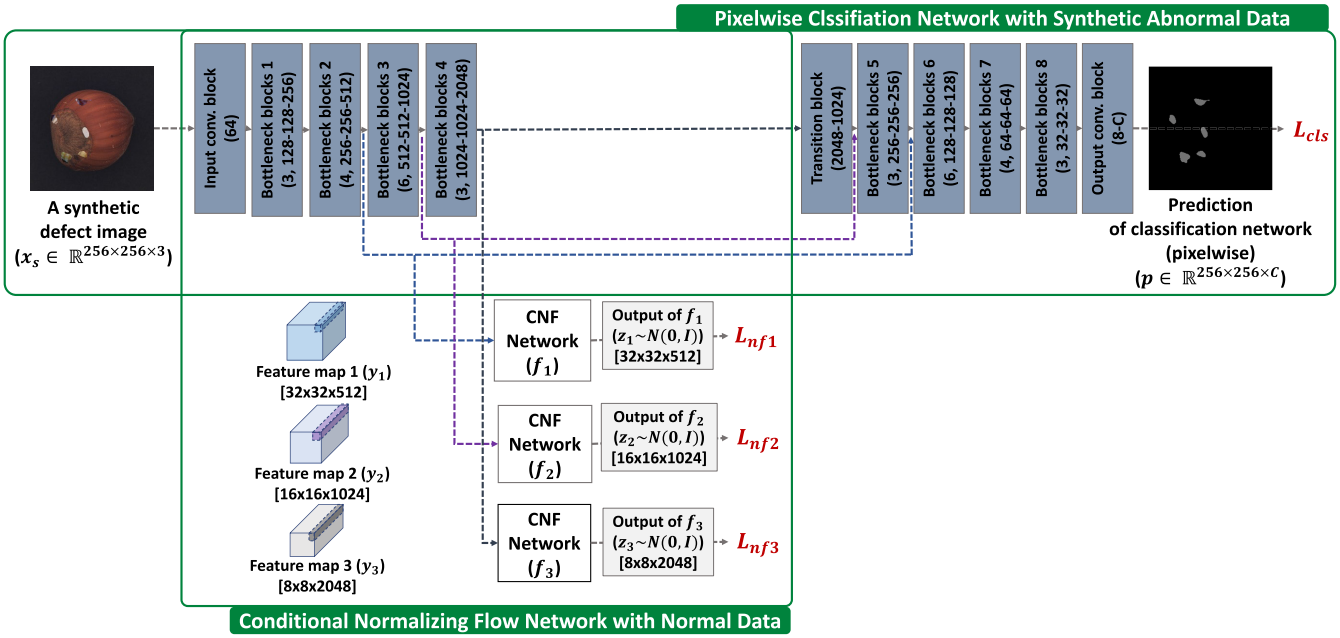


FIGURE 2. Network architecture of the proposed method comprising a U-Net structured network for pixelwise classification and CNF networks for density estimation.

CNF networks for feature maps with different resolutions conveyed by the fine-tuned feature extractor.

2) PIXELWISE CLASSIFICATION NETWORK

The convolutional blocks used to construct the pixelwise classification network are shown in Fig.2. We used the pre-trained WideResNet-50-2 [45] on the ImageNet dataset as an encoder for the pixelwise classification network. The encoder contains one input convolution block and four bottleneck blocks. The input convolution block contains a 7×7 convolution layer with strides of 2, a batch normalization layer, ReLU activation, and a max pooling layer with strides of 2. The bottleneck blocks (N, c1-c2-c3) denote a set of N bottleneck blocks, where c1, c2, and c3 represent the number of output channels for 1×1, 3×3, and 1×1 convolutions in the bottleneck block, respectively. The bottleneck block was proposed in ResNet [44] to effectively learn features in deep neural networks. This block comprises two 1×1 convolutional layers and a 3×3 convolutional layer. The former 1×1 convolution layer reduces the number of channels, whereas the latter 1×1 convolution layer restores the dimensions of an input feature. A 3×3 convolution layer between the two 1×1 convolution layers can extract a valuable feature from the compressed input feature.

We designed a decoder network to be symmetric to the encoder but made it shallow by reducing the number of channels to $\frac{1}{4}$ of the number of channels for the encoder. Therefore, the decoder contains four bottleneck blocks, one transition block, and one output block. The transition block reduces the number of channels using two combinations of 1×1 convolutional layers and ReLU activation. The output block contains a 3×3 convolution layer, a batch normalization layer, ReLU activation, and one convolution layer to reduce the number of channels to the number of classes.

The loss function of the pixelwise classification network using ground-truth probability with our label-smoothing strategy is defined as follows:

$$L_{cls} = -\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{c=1}^{C} t_{ij}^{c}\log p_{ij}^{c}, \tag{1}$$

where p_{ij}^{c} represents the predicted probability of class c for a pixel at position (i,j); t_{ij}^{c} represents the target probability of class c, which is the smoothed label in the synthetic defect regions; C is the number of classes; and H and W denote the height and width of the output, respectively.

3) CONDITIONAL NORMALIZING FLOW NETWORK

The CNF network (f_i) for a feature map ($\mathbf{y}_i$) receives feature vectors ($\mathbf{y}_i^k$) with channel dimension C_i at each position k of the i^{th} feature map and a position-embedding vector ($\mathbf{c}_i^k$) with dimension D generated by using the unique sinusoidal harmonics for its spatial location [15], [46]. Each CNF network consists of six invertible blocks, each of which is constructed as shown in Fig.3. First, the input feature vector is split into two subvectors: ($\mathbf{y}_{i,1}^k$ and $\mathbf{y}_{i,2}^k$). Subsequently, it generates the scale ($\mathbf{s}_{i,1}^k, \mathbf{s}_{i,2}^k$) and shift ($\mathbf{t}_{i,1}^k, \mathbf{t}_{i,2}^k$) parameters of each subvector by feeding the other subvector and the position embedding vector into fully connected layers and nonlinear functions to estimate complex distributions [15], [47]. Therefore, the invertible transformation of the CNF network can differ, even if the two feature vectors at different positions are the same. This allows the network to estimate

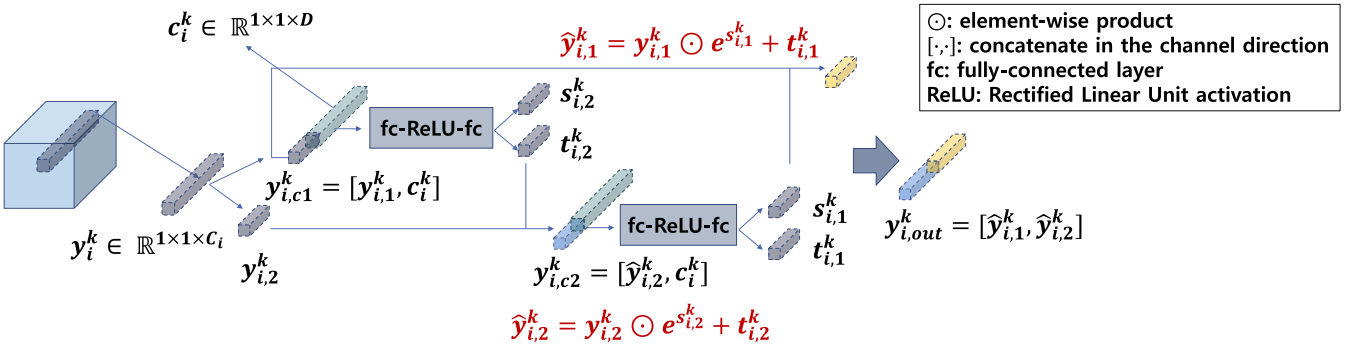


FIGURE 3. Invertible transform block of the CNF network of the proposed method.

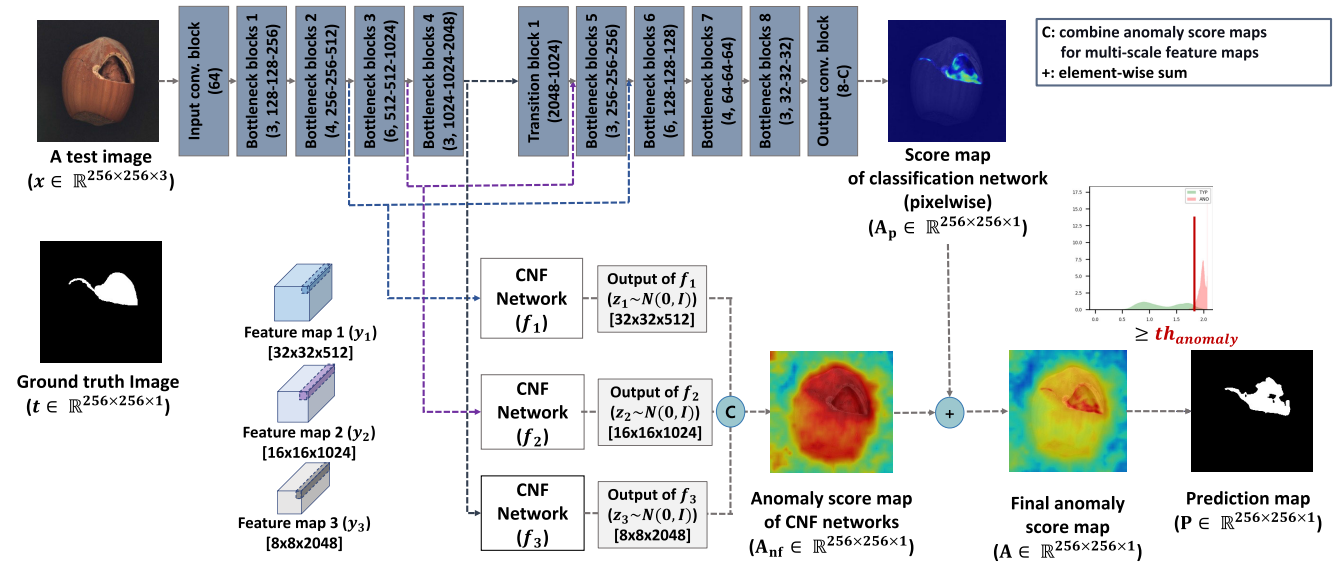


FIGURE 4. Inference process of the proposed method.

the distribution of normal features by considering spatial information, thereby enhancing the performance at the pixel level [15].

We trained each CNF network using the loss function by applying the negative log-sigmoid function to the log-likelihood of the normal feature vectors to prevent divergence [15], as in

$$\begin{aligned} L_{nfi} &= -\frac{1}{m_f}\sum_{k=1}^{m_f}\log\sigma(\log p(\mathbf{y}_i^k)) \quad (i=1,2,3) \\ &= -\frac{1}{m_f}\sum_{k=1}^{m_f}\log\sigma(\log p(\mathbf{z}_i^k)+\log|\det(\frac{\partial \mathbf{z}_i^k}{\partial \mathbf{y}_i^k})|) \\ &= -\frac{1}{m_f}\sum_{k=1}^{m_f}\log\sigma(-\frac{||\mathbf{z}_i^k||_2^2}{2}+\mathbf{1}^T\mathbf{s}_i^k+const.), \end{aligned} \tag{2}$$

where m_f represents the size of the minibatch input feature vectors for the CNF network, $\sigma(x)$ is the sigmoid function, $\mathbf{1}$ is a vector with the same dimensions as the scale parameter and all components of it are 1, and $\mathbf{z}_i^k$ denotes the output vector of the CNF network at position k for the i^{th} feature map.

C. TRAINING AND INFERENCE PROCESS

1) HYBRID TRAINING ALGORITHM

It has been reported that jointly training several networks using the same input data results in performance improvements in AD [10], [34], [35], [36]. Unlike these methods, the proposed method is the hybrid training algorithm that induces collaborative effects between pixelwise classification and CNF networks. Alg.1 illustrates the hybrid training process, which first trains the classification network with synthetic defect data and then trains the CNF networks with defect-free data. By training each network with different datasets, we expect to obtain collaboration effects in which the pixelwise classification network elaborately detects abnormal regions and the CNF networks detect abnormal regions with different appearances from the synthetic data. We also expect

the discriminative and in-domain features extracted from the fine-tuned feature extractor to enhance the performance of CNF networks in AD.

Therefore, we first trained the pixelwise classification network with synthetic defect data to fine-tune the feature extractor of the CNF networks for discriminative features and in-domain knowledge. As illustrated in Alg.1, we randomly generated synthetic defect data for every N_s epoch to prevent the network from overfitting to the synthetic defect data. In addition, we fine-tuned the encoder after training the decoder sufficiently to learn the discriminative features of the in-domain data while maintaining the rich discriminative features of the pretrained feature extractor. Subsequently, we trained the CNF networks with the fine-tuned feature extractor using only normal data, assuming that the CNF networks trained with normal data were not biased towards the synthetic defect data.

2) SCORE MAP AGGREGATION FOR INFERENCE

Fig.4 shows the inference process, i.e., weighted averaging the anomaly scores from the pixelwise classification network and CNF networks. We expect that the combined predictions will enhance performance owing to the positive impact of network ensembles [34], [35], [36] and the collaborative effects induced by the hybrid training process. We illustrate the detailed process used to obtain the anomaly score map of each network and to aggregate the score maps in Alg.2.

Unlike for the anomaly score map of the pixelwise classification network ($\mathbf{A_p}$), postprocessing is necessary to obtain the anomaly score map of the CNF networks ($\mathbf{A_{nf}}$) because the outputs of the CNF networks are not normalized in the range between 0 and 1. To achieve this, we first calculated the log-likelihood ($l_{i,k}$) of a feature vector ($\mathbf{y}_i^k$) at the k^{th} position of the i^{th} feature map for normal data as

$$l_{i,k} = \log p(\mathbf{y}_i^k) = -\frac{\|\mathbf{z}_i^k\|_2^2 + C_i \log(2\pi)}{2} + \mathbf{1}^T \mathbf{s}_i^k, \quad (3)$$

where C_i represents the channel dimensions of the i^{th} feature map extracted from the fine-tuned encoder. Instead of normalizing the log-likelihood $l_{i,k}$ to a range between 0 and 1 for the test dataset ($\mathbf{D}_{Test}$), as in CFlow-AD [15], we calculated the maximum value (u_i) of $l_{i,k}$ from the training dataset ($\mathbf{D}_{Train}$) to normalize $l_{i,k}$ as follows:

$$\mathbb{Z}_b^+ = \{1, 2, \ldots, b\},$$
$$u_i = \max_{j,k} l_{i,k}^{(j)}, \qquad \forall\, j \in \mathbb{Z}^+_{|\mathbf{D}_{Train}|}\, \forall\, k \in \mathbb{Z}^+_{S_i} \quad (4)$$

where $l_{i,k}^{(j)}$ is the log-likelihood for the j^{th} training image; $\mathbb{Z}_b^+$ represents the positive integer set with an upper bound (b); $|\mathbf{D}|$ denotes the number of elements in the set $\mathbf{D}$; and S_i is the product of H_i and W_i, which represent the height and width of the i^{th} feature map extracted from the fine-tuned encoder, respectively.

Algorithm 1 Training Process of the Proposed Method

INPUT
$\mathbf{D_n} = \{\mathbf{x_n}^{(j)}\}_{j=1}^{|\mathbf{D_n}|}$ is a normal dataset used for training.
$E(\cdot;\theta_\mathbf{E})$ is the shared feature extractor.
$D(\cdot;\theta_\mathbf{D})$ is the decoder network for pixelwise classification.
$f_i(\cdot,\cdot;\theta_{\mathbf{f}_i})$ is a CNF network with the i^{th} feature maps.
$\theta_{\mathbf{E_p}}$ parameters of the pretrained network on ImageNet.
N_s period to generate the synthetic defect dataset.
N_{pD}, N_p, N_f epochs for training the decoder-only, pixelwise classification, and CNF networks, respectively.

Step1: Train the pixelwise classification network with $\mathbf{D_s}$

Initialize $\theta_\mathbf{E}$ by $\theta_{\mathbf{E_p}}$
for $e = 1, \ldots, N_p$ **do**
  if $mod(e, N_s) == 1$ **then**
    $\mathbf{D_s} = \{\mathbf{x_s}^{(j)}, \mathbf{t_s}^{(j)}\}_{j=1}^{|\mathbf{D_n}|} \leftarrow AnomalySimulation(\mathbf{D_n})$
      ▷ generate synthetic defect dataset for training
  end if
  for $j = 1$ to $|\mathbf{D_s}|$ step m_p **do**
    $(\mathbf{x}, \mathbf{t}) \leftarrow$ sample minibatch of m_p data from $\mathbf{D_s}$
    $\mathbf{p} \leftarrow D(E(\mathbf{x};\theta_\mathbf{E});\theta_\mathbf{D})$
    $L_{cls} \leftarrow CrossEntropy(\mathbf{p}, \mathbf{t})$ ▷ Eq.(1)
    if $e \le N_{pD}$ **then**
      $\theta_\mathbf{D} \leftarrow optimizer(L_{cls}, \theta_\mathbf{D})$
    else
      $\theta_\mathbf{E} \leftarrow optimizer(L_{cls}, \theta_\mathbf{E})$
      $\theta_\mathbf{D} \leftarrow optimizer(L_{cls}, \theta_\mathbf{D})$
    end if
  end for
end for
$\hat{\theta_E}, \hat{\theta_D} \leftarrow \theta_\mathbf{E}, \theta_\mathbf{D}$

Step2: Train CNF networks with $\mathbf{D_n}$

for $e = 1, \ldots, N_f$ **do**
  for $j = 1$ to $|\mathbf{D_n}|$ step m_{nf} **do**
    $\mathbf{x} \leftarrow$ sample minibatch of m_{nf} data from $\mathbf{D_n}$
    $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \leftarrow E(\mathbf{x};\hat{\theta_E})$
    for $i = 1, 2, 3$ **do**
      $S_i \leftarrow$ spatial size of $\mathbf{y}_i$
      for $j_i = 1$ to S_i step m_f **do**
        $(\mathbf{y}, \mathbf{c}) \leftarrow \{\mathbf{y}_i^k, PE(k)\}_{k=j_i}^{\min\{j_i+m_f, S_i\}}$
          ▷ sample minibatch of m_f feature vectors and positional embedding vectors
        $\mathbf{z}, \mathbf{s} \leftarrow f_i(\mathbf{y}, \mathbf{c};\theta_{\mathbf{f}_i})$
        $L_{nfi} \leftarrow -\frac{1}{m_f}\sum_{k=1}^{m_f} \log\sigma(\mathbf{1}^T\mathbf{s}_i^k - \frac{\|\mathbf{z}_i^k\|_2^2}{2})$
          ▷ Eq.(2)
        $\theta_{\mathbf{f}_i} \leftarrow optimizer(L_{nfi}, \theta_{\mathbf{f}_i})$
      end for
    end for
  end for
end for
$\hat{\theta}_{\mathbf{f}_i} \leftarrow \theta_{\mathbf{f}_i}$

Algorithm 2 Inference Process of the Proposed Method

INPUT

$\mathbf{D}_{Train} = \{\mathbf{x_n}^{(j)}\}_{j=1}^{|\mathbf{D}_{Train}|}$ denotes the training data set.

$\mathbf{D}_{Test} = \{\mathbf{x}^{(j)}\}_{j=1}^{|\mathbf{D}_{Test}|}$ denotes the test dataset.

$E(\cdot;\hat{\theta_E})$ is the fine-tuned feature extractor.

$D(\cdot;\hat{\theta_D})$ is the trained decoder for pixelwise classification.

$f_i(\cdot,\cdot;\hat{\theta}_{\mathbf{f}_i})$ is the trained CNF network with i^{th} feature maps.

w weight to balance the anomaly scores of the pixelwise classification and CNF networks.

Step1: Obtain the anomaly score map ($\mathbf{A_p}$) of the pixelwise classification network

$\mathbf{A_p} \leftarrow D(E(\mathbf{x}^{(j)};\hat{\theta_E});\hat{\theta_D})$

Step2-1: Obtain the maximum value (u_i) of $\log p(\mathbf{y}_i^k)$ for $\mathbf{D}_{Train}$ to normalize the score of each CNF network

$u_1 = u_2 = u_3 = 0$ ▷ Initialize u_i
for $j = 1$ to $|D_{Train}|$ **do**
  $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \leftarrow E(\mathbf{x_n}^{(j)};\hat{\theta_E})$
  for $i = 1, 2, 3$ **do**
    $S_i, C_i \leftarrow$ spatial size, channel dimension of $\mathbf{y}_i$
    for $k = 1$ to S_i **do**
      $l_{i,k}^{(j)} \leftarrow \log p(\mathbf{y}_i^k)$ ▷ Eq.(3)
      $u_i \leftarrow \max\{u_i, l_{i,k}^{(j)}\}$ ▷ Eq.(4)
    end for
  end for
end for

Step2-2: Obtain the anomaly score map ($\mathbf{A_{nf}}$) of the CNF networks with $\mathbf{D}_{Test}$

$n_{max} = 0$ ▷ Initialize n_{max}
for $j = 1$ to $|D_{Test}|$ **do**
  $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \leftarrow E(\mathbf{x}^{(j)};\hat{\theta_E})$
  for $i = 1, 2, 3$ **do**
    $S_i, C_i \leftarrow$ spatial size, channel dimension of $\mathbf{y}_i$
    for $k = 1$ to S_i **do**
      $l_{i,k}^{(j)} \leftarrow \log p(\mathbf{y}_i^k)$ ▷ Eq.(3)
      $n_{i,k}^{(j)} \leftarrow \exp(l_{i,k}^{(j)} - u_i)$ ▷ Eq.(5)
    end for
    $\mathbf{N}_i^{(j)} = \{n_{i,k}^{(j)}\}_{k=1}^{S_i}$
    $\hat{\mathbf{N}}_i^{(j)} \leftarrow interpolation(\mathbf{N}_i^{(j)})$ ▷ Eq.(6)
  end for
  $\mathbf{N}^{(j)} \leftarrow \sum_{i=1}^{3} \hat{\mathbf{N}}_i^{(j)}$ ▷ Eq.(7)
  $n_{max}^{(j)} \leftarrow \max \mathbf{N}^{(j)}$
  $n_{max} \leftarrow \max\{n_{max}, n_{max}^{(j)}\}$
end for
$\mathbf{A_{nf}} = \mathbf{N} - n_{max}$ ▷ Eq. (8).

Step3: Aggregate $\mathbf{A_p}$ and $\mathbf{A_{nf}}$

$\mathbf{A} = \mathbf{A_{nf}} + w\mathbf{A_p}$ ▷ Eq.(9)

We set the normal score ($n_{i,k}^{(j)}$) of the feature vector for the j^{th} test image to the normalized $l_{i,k}^{(j)}$ for the training dataset by subtracting the maximum value (u_i) and applying an exponential function [15] as follows:

$$n_{i,k}^{(j)} = \exp(l_{i,k}^{(j)} - u_i), \qquad \forall j \in \mathbb{Z}^+_{|\mathbf{D}_{Test}|} \tag{5}$$

where $\mathbf{D}_{Test}$ denotes the test data set. Using the normal scores ($\mathbf{N}_i^{(j)}$) of the feature map, we obtained the normal score map ($\hat{\mathbf{N}}_i^{(j)}$) of the i^{th} feature map of the j^{th} test image as follows:

$$\begin{aligned} \mathbf{N}_i^{(j)} &= \{n_{i,k}^{(j)}\}_{k=1}^{S_i} \in \mathbb{R}^{H_i \times W_i}, \\ \hat{\mathbf{N}}_i^{(j)} &= g(\mathbf{N}_i^{(j)}) \in \mathbb{R}^{H \times W}, \end{aligned} \tag{6}$$

where H and W denote the height and width of the test image, respectively, and g denotes the bilinear interpolation function [15]. To finalize the normal score map ($\mathbf{N}^{(j)}$) of the test image, we summed the normal score maps of each feature map as follows:

$$\mathbf{N}^{(j)} = \sum_{i=1}^{3} \hat{\mathbf{N}}_i^{(j)}. \tag{7}$$

Then, we acquired an anomaly score map ($\mathbf{A_{nf}}$) from the CNF networks of a test image by subtracting the maximum normal score across the test dataset from a normal score map ($\mathbf{N}$) [15] as follows:

$$\begin{aligned} \mathbf{N}^{(j)} &= \{n_k^{(j)}\}_{k=1}^{S}, \\ \mathbf{A_{nf}} &= \mathbf{N} - \max_{j,k}(n_k^{(j)}), \end{aligned} \tag{8}$$

where S denotes the product of H and W.

Finally, we produced an anomaly score map ($\mathbf{A}$) for the proposed method using the weighted averaging anomaly score maps of the pixelwise classification and CNF networks as

$$\mathbf{A} = \mathbf{A_{nf}} + w\mathbf{A_p}, \tag{9}$$

where w is the weight balancing of the anomaly scores from the pixelwise classification and CNF networks. To evaluate performance for AD at the image level, we determined the maximum value of the aggregated anomaly score map as the anomaly score of the test image.

IV. EXPERIMENT

To verify the effectiveness of the proposed method, we compared its performance with those of other methods such as PaDiM [12], CFlow-AD [15], DRAEM [8], and CDO [18] on the MVTecAD and BTAD datasets [37], [38]. We also conducted ablation studies to empirically demonstrate our hypotheses.

A. DATASETS

1) MVTecAD DATASET

We evaluated our method using the MVTecAD dataset [37], which is a benchmark dataset for AD methods that focuses on quality inspection in the industry. It contains 5,354 high-resolution color images with sizes ranging

from 700 × 700 to 1,024 × 1.024 across 15 product categories. The dataset comprises 12–60 normal test images, 30–141 defect test images, and 60–391 normal training images for each category. In addition, the dataset contains 73 different defect types, including dents, scratches, contamination, and structural differences. Therefore, owing to these attributes, it is widely used to evaluate the generalization ability of various AD methods.

2) BTAD DATASET

Similar to the MVTecAD dataset, the BTAD dataset [38] is a benchmark dataset for AD methods used for quality inspection in the industry. It contains 2,830 real-world images of three industrial products showing body and surface defects. The image resolutions of products 1, 2, and 3 are 1,600 × 1.600, 600×600, and 800×600 pixels, respectively. There are 400, 1,000, and 399 normal training images for products 1, 2, and 3, respectively. The dataset consists of 21-400 normal test images and 41-200 defect test images for each product.

B. IMPLEMENTATION DETAILS

Each experiment and ablation study were conducted using 256 × 256 resized images for all categories in the MVTecAD dataset and all products in the BTAD dataset. As mentioned in Section III, we fine-tuned the pretrained WideResNet-50-2 after sufficiently training the shallow decoder, which has a symmetric structure with the encoder. To effectively acquire in-domain knowledge without losing the rich discriminative ability learned from the ImageNet dataset, we did not update the sample mean and sample variance of the batch normalization layers for the in-domain data. Every pixelwise classification network had the same architecture for all the experiments. Using the fine-tuned feature extractor, we trained CNF networks that comprised a sequence of six invertible transformations. To aggregate score maps from the pixelwise classification and CNF networks, we manually determined the balancing weight w for each category in a way such that the AUPR score was maximized. Hyperparameters such as the initial learning rate and training epochs for the pixelwise classification and CNF networks for each category of the MVTecAD dataset and each product of the BTAD dataset are available on the GitHub website.[1]

We implemented and evaluated DRAEM for the MVTecAD and BTAD datasets using the original source code without modifications [8]. However, we modified the original source code of CDO to utilize the same feature maps as the CNF networks in the proposed method for the BTAD dataset, although we did not modify it for the MVTecAD dataset [18]. For PaDiM [12], the same modification was adopted for both the MVTecAD and BTAD datasets. For a fair comparison, we set the number of invertible transformations in CFlow-AD to six and produced anomaly score maps for every method using the same process as our method. PaDiM, CFlow-AD, and CDO utilize the pretrained WideResNet-50-2 as a feature extractor, similar to the proposed method. However, the DRAEM constructs its network architecture without using the pretrained network [8].

Moreover, we conducted an ablation study to compare the performances of CNF networks that use the pretrained network, the fine-tuned network by pixelwise classification, and a network fine-tuned by pixelwise regression to evaluate the effectiveness of a feature extractor of the CNF networks. To achieve this, we designed a reconstruction network with the same architecture as the pixelwise classification network, except that it did not use skip connections. We believe that skip connections cause an overgeneralization problem in reconstruction-based methods. The threshold ($\text{th}_{\text{anomaly}}$) for generating the prediction map for the defect regions was set to the value that achieved the best f1-score, which was calculated as the harmonic mean of precision and recall at the pixel level. Each experiment was repeated three times with random initialization to prevent overestimation by each method.

C. PERFORMANCE METRICS

We selected three threshold-independent metrics to evaluate our method: the area under the receiver operating characteristic curve (AUROC), area under the precision-recall curve (AUPR), and area under the per-region overlap curve (AUPRO). The AUROC metric is the area of the true positive rate (*TPR*)-false positive rate (*FPR*) curve, which is a useful metric for evaluating general-purpose classifications. *TPR* (or recall) quantifies the number of correct predictions for the defective data from the total defective data, as in

$$TPR = \frac{TP}{TP + FP}, \tag{10}$$

where *TP* is the number of true positives, indicating correct predictions for abnormal or defective data, and *FP* is the number of false positives, representing incorrect predictions for normal data.

FPR is the ratio of the number of incorrect predictions for normal data to the total normal data:

$$FPR = \frac{FP}{FP + TN}, \tag{11}$$

where *TN* is the number of true negatives, indicating correct predictions for normal data.

However, the AUROC metric can be less informative for highly skewed datasets because it considers the correct prediction of the majority class (i.e., normal) [48]. Therefore, the AUPR metric is a superior method for classifying rare events because it does not consider the correct prediction of the majority class by constructing a curve with precision (*PRC*) instead of *FPR*. The precision metric is formulated using only the predictions for the minority class (i.e., defects) as in

$$PRC = \frac{TP}{TP + FN}, \tag{12}$$

where *FN* is the number of false negatives, indicating incorrect predictions for abnormal or defective data.

[1] https://github.com/seungmi-oh/AD-CLSCNFs

TABLE 1. Experimental results of the CNF networks according to a feature extractor. (The value in bold type represents the best performance.)

Network	CNF networks with the pretrained feature extractor on ImageNet (CFlow-AD, FE_{PRE}-CNFs)				CNF networks with the fine-tuned feature extractor by pixelwise regression (FE_{REG}-CNFs)				CNF networks with the fine-tuned feature extractor by pixelwise classification (FE_{CLS}-CNFs)			
Metric /Category	Img AUROC	Pix AUROC	Pix AUPR	AUPRO	Img AUROC	Pix AUROC	Pix AUPR	AUPRO	Img AUROC	Pix AUROC	Pix AUPR	AUPRO
Bottle	**100.00**	98.74	73.54	94.48	98.92	98.03	71.59	87.39	**100.00**	**98.80**	**74.08**	**94.51**
Cable	93.82	97.28	59.07	93.26	**98.25**	**98.42**	**64.65**	**95.25**	93.65	97.33	59.68	93.33
Capsule	97.22	99.06	49.42	94.50	84.86	97.54	26.84	85.41	**97.70**	**99.09**	**50.57**	**95.03**
Carpet	98.26	99.24	**66.23**	96.77	79.01	89.19	36.63	71.49	**99.35**	**99.25**	62.41	**97.29**
Grid	98.80	98.89	37.95	95.78	83.32	79.68	6.28	59.82	**99.50**	**98.92**	**38.05**	**96.04**
Hazelnut	**99.99**	**98.81**	**62.12**	96.71	93.93	97.51	54.00	87.59	99.95	**98.81**	61.75	**97.07**
Leather	**100.00**	99.59	57.99	**98.90**	95.42	97.70	46.36	90.19	**100.00**	**99.60**	**58.78**	98.85
Metal Nut	99.17	97.93	78.62	94.27	95.09	97.79	**80.80**	80.85	**99.69**	**98.07**	78.89	**94.48**
Pill	95.50	98.43	71.71	95.98	81.90	92.47	36.61	80.94	**96.73**	**98.64**	**75.29**	**96.35**
Screw	92.04	98.68	35.05	94.26	54.43	95.28	6.08	83.18	**92.33**	**98.85**	**36.25**	**94.86**
Tile	**99.64**	97.67	**76.89**	91.31	93.50	93.46	65.56	81.99	98.42	**97.93**	74.94	**92.62**
Toothbrush	84.35	**98.37**	**33.05**	**90.16**	**97.69**	98.24	31.31	85.91	83.52	98.29	31.96	89.64
Transistor	97.03	91.09	50.40	82.84	94.84	**95.93**	54.98	87.78	**99.79**	95.13	**57.30**	**89.28**
Wood	95.79	95.80	57.23	93.16	**97.34**	93.24	53.87	80.98	96.23	**96.34**	**58.54**	**94.09**
Zipper	98.99	99.01	56.40	96.55	91.87	96.46	32.23	86.91	**99.35**	**99.17**	**57.63**	**97.07**
Average	96.71	97.91	57.71	93.93	89.36	94.73	44.52	83.05	**97.08**	**98.28**	**58.41**	**94.70**

The AUPRO metric is a region-based metric representing the area of the *PRO–FPR* curve, unlike evaluation methods at the pixel or image level, such as the AUROC and AUPR metrics. The *PRO* metric is defined as

$$PRO = \sum_{j} \sum_{s} \frac{|P_j \cap T_{j,s}|}{|T_{j,s}|}, \tag{13}$$

where P_j denotes the set of pixels predicted as a defect in the j^{th} defective image and T_j is the set of pixels constituting the s^{th} defect segment of the j^{th} defective image. It can effectively evaluate the segmentation performance regardless of the size of the defect segments because the *PRO* metric is calculated as the expectation of the ratio of correctly predicted pixels to ground-truth pixels for every segment.

D. EXPERIMENTAL RESULTS

1) MVTecAD DATASET

We conducted a comprehensive study of feature extractors for CNF networks on the MVTecAD dataset using the AUROC, AUPR, and AUPRO metrics. Table 1 shows the performance of CNF networks with the pretrained feature extractor (FE_{PRE}), the fine-tuned feature extractor by pixelwise classification (FE_{CLS}), and the fine-tuned feature extractor by pixelwise regression (FE_{REG}). The CNF networks with FE_{CLS} (FE_{CLS}-CNFs) yielded the highest imagewise AUROC (97.08%), pixelwise AUROC (98.28%), AUPR (58.41%), and AUPRO (94.70%) averages for all categories of the MVTecAD dataset. In contrast, the CNF networks with FE_{REG} (FE_{REG}-CNFs) showed the worst performance. This suggests that the discriminative ability of the feature extractor is critical for improving the performance of CNF networks in AD. The fact that FE_{CLS}-CNFs performed better than CNF networks with FE_{PRE} (FE_{PRE}-CNFs) indicates that FE_{PRE} may not extract valuable features for AD owing to the large domain gap. As a result, we empirically showed that a fine-tuned feature extractor to identify synthetic abnormal data is useful for improving performance of AD thanks to the discriminative features of in-domain data.

To further confirm the effect of the feature extractor, we visualized the score ($\mathbf{A_{nf}}$) and prediction maps ($\mathbf{P_{nf}}$) of CNF networks with different feature extractors in Fig.5. As shown in the first rows of Fig.5, $\mathbf{A_{nf}}$ of FE_{REG}-CNFs has too small anomaly scores in the defective regions to identify all defects. However, both $\mathbf{A_{nf}}$ of FE_{CLS}-CNFs and FE_{PRE}-CNFs accurately highlight all the defective regions. This implies that FE_{REG}-CNFs did not identify defect regions with subtle differences because FE_{REG} may extract normal features from defective images. By contrast, the second row of Fig.5 represents the necessity of in-domain features for AD. $\mathbf{P_{nf}}$ of FE_{PRE}-CNFs detected all three leads as defects, in contrast to $\mathbf{P_{nf}}$ of FE_{REG}-CNFs and FE_{CLS}-CNFs, which correctly detected the defective bent lead. The last row of Fig.5 shows that FE_{CLS} can help detect defective regions with subtle differences thanks to its discriminative ability and in-domain knowledge. Every $\mathbf{A_{nf}}$ highlights the overall screw object because the appearance of defects is difficult to identify. Nevertheless, $\mathbf{P_{nf}}$ of FE_{CLS}-CNFs correctly detected the defect regions. This supports our argument that in-domain knowledge and discriminative ability are meaningful for enhancing the performance.

We also conducted an ablation study to determine whether the predictions of the pixelwise classification network and CNF networks are complementary and whether sharing feature extractors is beneficial for performance improvement.

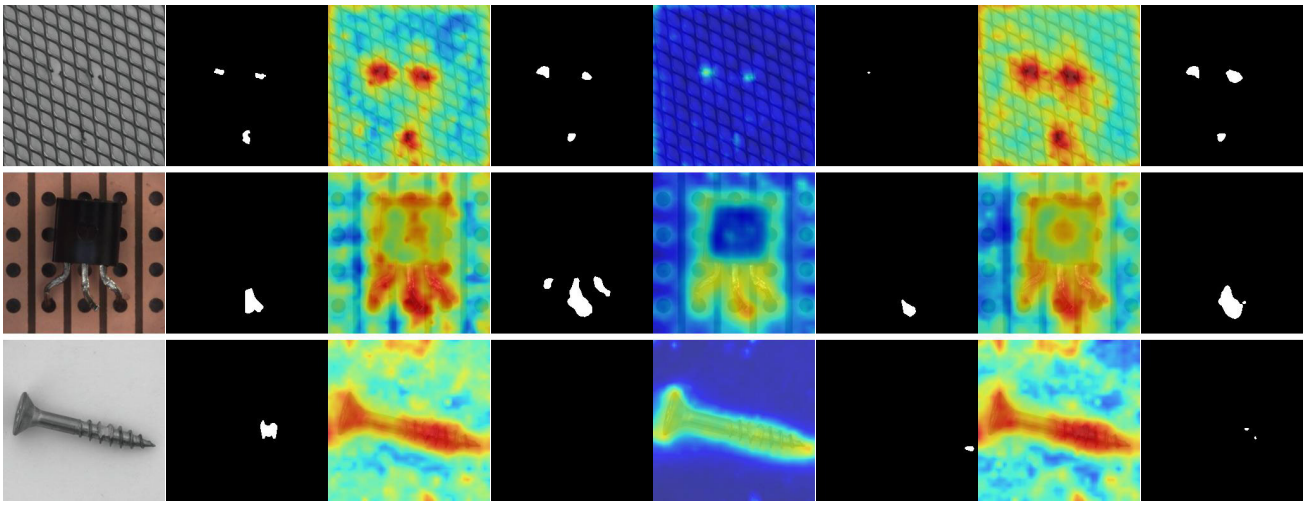


FIGURE 5. Anomaly score and prediction maps of the CNF networks with different feature extractors: (a) input image, (b) ground-truth image, (c) $\mathbf{A}_{nf}$ of FE_{PRE}-CNFs, (d) $\mathbf{P}_{nf}$ of FE_{PRE}-CNFs, (e) $\mathbf{A}_{nf}$ of FE_{REG}-CNFs, (f) $\mathbf{P}_{nf}$ of FE_{REG}-CNFs, (g) $\mathbf{A}_{nf}$ of FE_{CLS}-CNFs, and (h) $\mathbf{P}_{nf}$ of FE_{CLS}-CNFs.

TABLE 2. Experimental results of the proposed method for the MVTecAD dataset.

Score Map	Img AUROC (%)	Pix AUROC (%)	Pix AUPR (%)	AUPRO (%)
$\mathbf{A_p}$	91.96	92.32	55.27	73.39
$\mathbf{A}_{\mathbf{nf}}^{\mathbf{PRE}}$	96.71	97.91	57.71	93.93
$\mathbf{A_{nf}}$	97.08	98.28	58.41	94.70
$\mathbf{A}^{\mathbf{ns}}$	97.77	98.25	71.27	95.02
A	**98.09**	**98.59**	**72.09**	**95.69**

We compared the performance of the proposed method (CLS-CNFs-sFE), a network (CLS-CNFs-nsFE) that combined the pixelwise classification network (CLS) and FE_{PRE}-CNFs (i.e., CFlow-AD) not sharing feature extractors, and individual networks (CLS, FE_{CLS}-CNFs, and CFlow-AD). In Table 2, we denoted the anomaly score map of the CFlow-AD network as $\mathbf{A}_{\mathbf{nf}}^{\mathbf{PRE}}$ and that of CLS-CNFs-nsFE as $\mathbf{A}^{\mathbf{ns}}$. As illustrated in the table, aggregating the score maps of the pixelwise classification and CNF networks enhanced the performance compared to the individual networks for all metrics regardless of whether the feature extractor was shared. Notably, the AUPR metric, which is superior for evaluating the discerning ability of imbalanced settings, increases steeply for combined networks compared to individual networks. This implies that the pixelwise classification and CNF networks complement each other. Moreover, the proposed method exhibits the best performance for all metrics. This indicates that sharing feature extractors improves the performance because FE_{CLS} enhances the performance of CNF networks because of its discriminative ability and in-domain knowledge, which is consistent with the results in Table 1 and Fig.5.

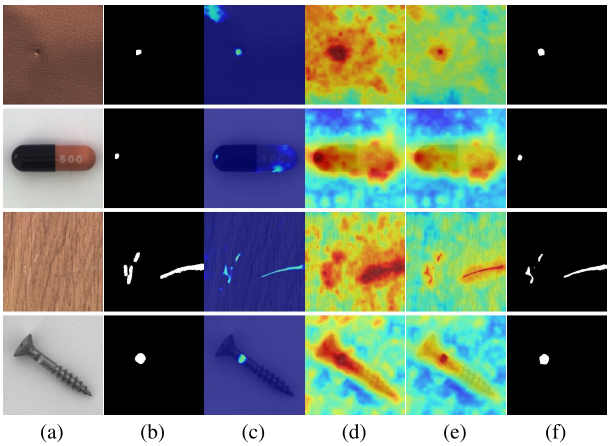


FIGURE 6. Anomaly score and prediction maps of the pixelwise classification network, CNF networks, and the proposed method for the MVTecAD dataset: (a) input image, (b) ground-truth image, (c) $\mathbf{A_p}$ (d) $\mathbf{A_{nf}}$, (e) **A**, and (f) **P**.

Fig.6 shows the qualitative results supporting our assumption that the prediction of CNF networks mitigates the overfitting problem of the pixelwise classification network. As indicated in the first and second rows in Fig.6(c), the pixelwise classification network assigned high anomaly scores to the normal regions owing to the overfitting problem. However, the anomaly score map (**A**) of the proposed method highlights the defect regions and the prediction map (**P**) of the proposed method correctly detects the defect regions without false detection. This indicates that aggregating with the prediction of CNF networks has a positive impact on the avoidance of false detections in pixelwise classification networks.

Moreover, the third and final rows in Fig.6 show that the pixelwise classification network supports the CNF networks to identify defective regions with high confidence. The CNF

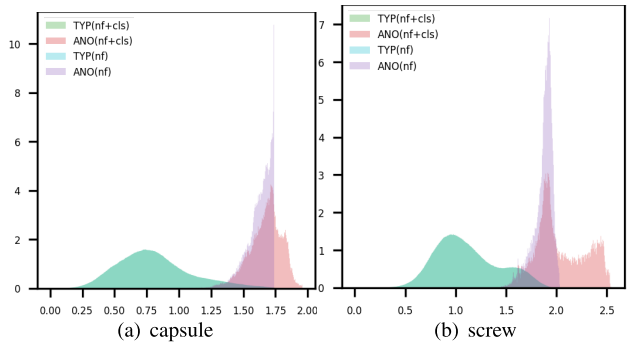


FIGURE 7. Histogram plot of the anomaly scores for the normal (TYP.) and defect pixels (ANO.) of all test images, where "nf" and "nf+cls" denote the prediction of FE_{CLS}-CNFs and the proposed method (CLS-CNFs-sFE), respectively.

TABLE 3. Computational time and memory size of the proposed method compared to those of the CFlow-AD method.

Network Architecture	Model size (MB)	Inference speed (fps)
FE (WideResnet-50-2)	268 (×0.36)	-
CLS	285 (×0.38)	59.25 (×1.45)
FE_{CLS}-CNFs	741 (×1.00)	40.87 (×1.00)
CLS-CNFs-sFE (proposed)	771 (×1.04)	37.94 (×0.93)
CLS-CNFs-nsFE	1,026 (×1.38)	35.77 (×0.88)

networks have the limitation that the score map for an object, which has similar appearances of normal and defective regions, highlights the overall object with high anomaly scores. The pixelwise classification network mitigates this limitation by accurately detecting the defective regions because it has learned subtle and distinct differences from normal regions using synthetic defect data. Because of this effect, the final anomaly score map of the screw becomes more intuitive than that of the CNF networks.

We also plotted the histogram of the pixelwise anomaly scores in Fig.7 to illustrate that the aggregated scores are helpful in separating normal and abnormal samples. The histogram (blue) of the anomaly scores for the normal test pixels of FE_{CLS}-CNFs is almost identical to that of the proposed method (green). However, the histogram (red) of the anomaly scores of the proposed method for the test defect pixels shifted in the direction distant from the histogram of the normal pixels compared with that (purple) of FE_{CLS}-CNFs. As a result, the distance between the modes of the histograms for the proposed method increased compared to that of the CNF networks. We believe that this is because the prediction of the pixelwise classification network separates the normal and defective samples.

We also compared the complexity of our method by measuring the model size and inference speed using methods in CFlow-AD [15] and CDO [18]. We measured the model size as the size of the floating-point parameters for each network [18]. And we set the batch size as 32 in measuring the inference speed for a fair comparison [15]. Table 3 is measured in the same environment with an Intel(R) Xeon(R) W-2245 CPU @ 3.90GH CPU, NVIDIA GTX 3090 GPU, and 128GB RAM for a fair comparison. As indicated in the table, the model size of the pixelwise classification network (CLS) was almost the same as that of the encoder (FE) because of the shallow decoder. Therefore, the proposed method, which combines the pixelwise classification network and CNF networks by sharing feature extractors, does not significantly increase the complexity of the network compared with CFlow-AD. However, the model size and inference speed of CLS-CNFs-nsFE were 1.38× larger and 0.88× slower than those of the CFlow-AD network, respectively. Therefore, sharing feature extractors is better than not sharing them, in terms of both the performance of AD and network complexity.

TABLE 4. Experimental results of the proposed method and existing methods for the MVTecAD dataset.

	Img AUROC (%)	Pix AUROC (%)	Pix AUPR (%)	AUPRO (%)
PaDiM	96.87	97.89	57.73	93.92
CFlow-AD	96.71	97.91	57.71	93.93
CDO	94.84	97.60	59.42	93.68
DRAEM	96.58	97.09	69.06	91.23
Proposed Method	**98.09**	**98.59**	**72.09**	**95.69**

To empirically demonstrate our arguments, we compared our method with the existing methods for AD. As illustrated in Table 4, the proposed method achieved the highest imagewise AUROC (98.09%), pixelwise AUROC (98.59%), AUPR (72.09%), and AUPRO (95.69%), averaged for all categories of the MVTecAD dataset. The fact that the AUPR values of CDO, DRAEM, and the proposed method are higher than those of PaDiM and CFlow-AD supports our argument that training a network with synthetic defect data is beneficial for detecting defect regions owing to its discerning ability. Additionally, we believe that the quality of the synthetic defect data influences the performance, considering the higher AUPR of DRAEM compared to CDO. The higher AUPRO values of PaDiM, CFlow-AD, and the proposed method compared to those of CDO and DRAEM support our assumption that the prediction of the distance-based method training with only normal data mitigates the overfitting problem with synthetic defect data. As shown in Fig.8, CDO and DRAEM did not detect some defective regions at all or over-detected normal regions as defect segments owing to the overfitting problem. Accordingly, it decreases AUPRO, which is used to evaluate performance at the segment level. As a result, the proposed method maintains the advantages of each method by combining pixelwise classification and CNF networks through a shared feature extractor.

2) BTAD DATASET

To investigate whether our arguments hold for other datasets, we conducted experiments using the BTAD dataset. We compared the performance of CLS, CFlow-AD, FE_{CLS}-CNFs,

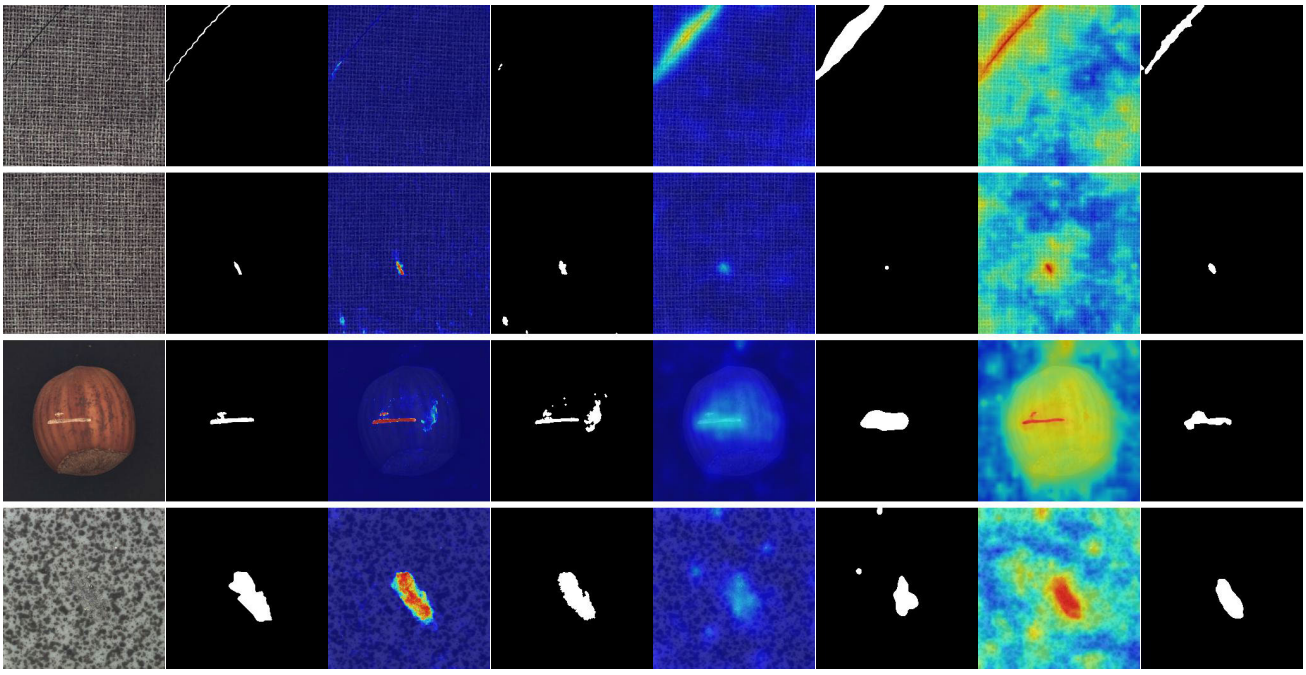


FIGURE 8. Anomaly score and prediction maps of the proposed method and existing methods for the MVTecAD dataset: (a) input image, (b) ground-truth image, (c) A of DRAEM, (d) P of DRAEM, (e) A of CDO, (f) P of CDO, (g) A of the proposed method, and (h) P of the proposed method.

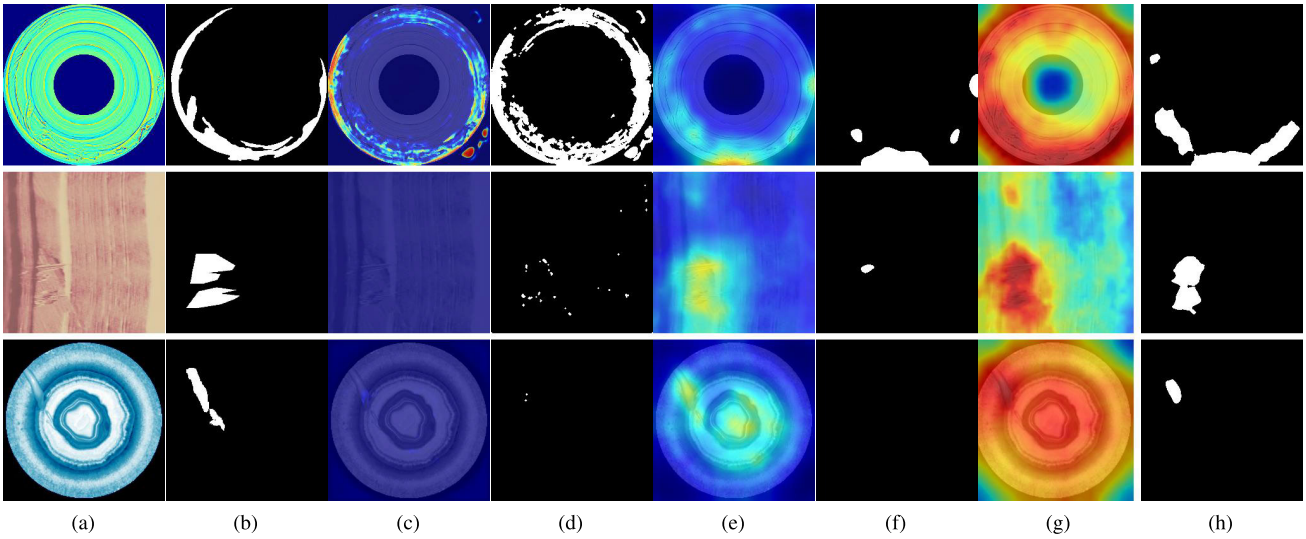


FIGURE 9. Anomaly score and prediction maps of the proposed method and existing methods for the BTAD dataset: (a) input image, (b) ground-truth image, (c) A of DRAEM, (d) P of DRAEM, (e) A of CDO, (f) P of CDO, (g) A of the proposed method, and (h) P of the proposed method.

CLS-CNFs-nsFE, and the proposed method (CLS-CNFs-sFE) to determine whether the predictions of the pixelwise classification network and CNF networks are complementary and whether the sharing of feature extractors is beneficial for performance improvement. Table 5 shows that combining the score maps of the pixelwise classification and CNF networks improves the performance of all metrics, regardless of whether the feature extractor is shared. This indicates that the combined score map improves the performance because of the positive impact of network ensembles. Notably, the proposed method exhibited the best performance for all metrics. This implies that the sharing of feature extractors improves the performance because FE_{CLS} improves the performance of CNF networks because of the discriminative features of in-domain data, which is consistent with the results of experiments on the MVTecAD dataset.

We also compared the performance of our method and existing methods on the BTAD dataset to empirically

TABLE 5. Experimental results of the proposed method on the BTAD dataset.

Score Map	Img AUROC (%)	Pix AUROC (%)	Pix AUPR (%)	AUPRO (%)
$\mathbf{A_p}$	83.65	90.92	28.02	63.70
$\mathbf{A_{nf}^{PRE}}$	95.37	97.01	51.23	71.94
$\mathbf{A_{nf}}$	95.96	97.07	51.31	72.32
$\mathbf{A^{ns}}$	95.35	97.08	53.67	72.14
A	**95.97**	**97.12**	**53.76**	**72.66**

TABLE 6. Experimental results of the proposed and existing models on the BTAD dataset.

	Img AUROC (%)	Pix AUROC (%)	Pix AUPR (%)	AUPRO (%)
DRAEM	91.11	84.54	16.22	44.63
PaDiM	93.04	93.85	25.22	60.97
CFlow-AD	95.37	97.01	51.23	71.94
CDO	92.64	96.25	46.75	69.40
Proposed Method	**95.97**	**97.12**	**53.76**	**72.66**

demonstrate our arguments. As shown in Table 6, the proposed method achieved the highest average imagewise AUROC (95.97%), pixelwise AUROC (97.12%), AUPR (53.76%), and AUPRO (72.66%) for all products of the BTAD dataset, which is consistent with the results of the experiments on the MVTecAD dataset. Unlike the results for the MVTecAD dataset, DRAEM and CDO did not perform as well as CFlow-AD for all metrics. Because DRAEM and CDO generate synthetic defect data without using in-domain data, they may not be effective for the BTAD dataset whose data have similar appearances of normal and defective regions. Furthermore, Fig.9 indicates that CDO and DRAEM fail to detect some defective regions or incorrectly predict normal regions as defect segments owing to the overfitting problem. The proposed method, on the other hand, correctly detected defect regions as it has learned the discriminative features from the in-domain data and preserved the advantages of the pixelwise classification and CNF networks by aggregating the score maps.

V. CONCLUSION

We propose a novel AD method that combines a pixelwise classification network with CNF networks by sharing feature extractors. We were able to fine-tune a pretrained feature extractor by training the pixelwise classification network to identify abnormal regions in the synthetic abnormal data. By doing so, we expect the feature extractor network to learn the discriminative features of in-domain data. After training the feature extractor, we estimated the density of normal data using the CNF networks. During inference, by aggregating the anomaly scores from the pixelwise classification and CNF-based density, we were able to mitigate the problem of overfitting to the synthetic defect data. Furthermore, we believe that network ensembles have a positive impact on performance. In experiments using the MVTecAD and BTAD datasets, the proposed method showed significantly improved performance compared to existing methods. Further, in our ablation study, it was confirmed that the performance of the combined network is superior to that of individual networks while the computation did not increase significantly thanks to the sharing of feature extractors.

REFERENCES

[1] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," 2018, *arXiv:1807.02011*.
[2] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, Dec. 2018, pp. 622–637.
[3] V. Zavrtanik, M. Kristan, and D. Skoč aj, "Reconstruction by inpainting for visual anomaly detection," *Pattern Recognit.*, vol. 112, Apr. 2021, Art. no. 107706.
[4] X. Yan, H. Zhang, X. Xu, X. Hu, and P.-A. Heng, "Learning semantic context from normal samples for unsupervised anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, May 2021, pp. 3110–3118.
[5] Z. Li, N. Li, K. Jiang, Z. Ma, X. Wei, X. Hong, and Y. Gong, "Superpixel masking and inpainting for self-supervised anomaly detection," in *Proc. BMVC*, 2020.
[6] C. Huang, Q. Xu, Y. Wang, Y. Wang, and Y. Zhang, "Self-supervised masking for unsupervised anomaly detection and localization," *IEEE Trans. Multimedia*, vol. 25, pp. 4426–4438, Oct. 2022.
[7] J. Li, G. Li, T. Xie, and Z. Wu, "MST-UNet: A modified Swin transformer for water bodies' mapping using Sentinel-2 images," *J. Appl. Remote Sens.*, vol. 17, no. 2, May 2023, Art. no. 026507.
[8] V. Zavrtanik, M. Kristan, and D. Skocaj, "DRÆM—A discriminatively trained reconstruction embedding for surface anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8330–8339.
[9] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.
[10] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "CutPaste: Self-supervised learning for anomaly detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9659–9669.
[11] H. M. Schlüter, J. Tan, B. Hou, and B. Kainz, "Natural synthetic anomalies for self-supervised anomaly detection and localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2022, pp. 474–489.
[12] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Proc. Int. Conf. Pattern Recognit.* Cham, Switzerland: Springer, Jan. 2021, pp. 475–489.
[13] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but DifferNet: Semi-supervised defect detection with normalizing flows," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1906–1915.
[14] H. Shi, Y. Zhou, K. Yang, X. Yin, and K. Wang, "CSFlow: Learning optical flow via cross strip correlation for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2022, pp. 1851–1858.
[15] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1819–1828.
[16] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student–teacher anomaly detection with discriminative latent embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4182–4191.
[17] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee, "Multiresolution knowledge distillation for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14897–14907.
[18] Y. Cao, X. Xu, Z. Liu, and W. Shen, "Collaborative discrepancy optimization for reliable image anomaly localization," *IEEE Trans. Ind. Informat.*, vol. 19, no. 11, pp. 10674–10683, Nov. 2023.

[19] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 1577–1581.

[20] T.-H. Vu, J. Boonaert, S. Ambellouis, and A. Taleb-Ahmed, "Multi-channel generative framework and supervised learning for anomaly detection in surveillance videos," *Sensors*, vol. 21, no. 9, p. 3179, May 2021.

[21] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, May 2017, pp. 146–157.

[22] D. Zimmerer, S. A. A. Kohl, J. Petersen, F. Isensee, and K. H. Maier-Hein, "Context-encoding variational autoencoder for unsupervised anomaly detection," 2018, *arXiv:1812.05941*.

[23] H. Zhao, Y. Li, N. He, K. Ma, L. Fang, H. Li, and Y. Zheng, "Anomaly detection for medical images using self-supervised and translation-consistent features," *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3641–3651, Dec. 2021.

[24] J. Tan, B. Hou, J. Batten, H. Qiu, and B. Kainz, "Detecting outliers with foreign patch interpolation," 2020, *arXiv:2011.04197*.

[25] M. Hu, Z. Ji, K. Yan, Y. Guo, X. Feng, J. Gong, X. Zhao, and L. Dong, "Detecting anomalies in time series data via a meta-feature based approach," *IEEE Access*, vol. 6, pp. 27760–27776, 2018.

[26] Z. Ji, Y. Wang, K. Yan, X. Xie, Y. Xiang, and J. Huang, "A space-embedding strategy for anomaly detection in multivariate time series," *Expert Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117892.

[27] O. Rippel, P. Mertens, and D. Merhof, "Modeling the distribution of normal data in pre-trained deep features for anomaly detection," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6726–6733.

[28] M. Yang, P. Wu, and H. Feng, "MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities," *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105835.

[29] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: One-class novelty detection using GANs with constrained latent representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2893–2901.

[30] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 481–490.

[31] S. Pidhorskyi, R. Almohsen, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 6823–6834.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[33] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling, "Learning likelihoods with conditional normalizing flows," 2019, *arXiv:1912.00042*.

[34] J. Wang and R. Zuo, "Model averaging for identification of geochemical anomalies linked to mineralization," *Ore Geol. Rev.*, vol. 146, Jul. 2022, Art. no. 104955.

[35] X. Dong, C. J. Taylor, and T. F. Cootes, "Defect classification and detection using a multitask deep one-class CNN," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1719–1730, Jul. 2022.

[36] C. Zhang, Y. Wang, and W. Tan, "MTHM: Self-supervised multitask anomaly detection with hard example mining," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[37] 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.

[38] 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.

[39] E. G. Tabak and E. Vanden-Eijnden, "Density estimation by dual ascent of the log-likelihood," *Commun. Math. Sci.*, vol. 8, no. 1, pp. 217–233, 2010.

[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[41] K. Perlin, "An image synthesizer," *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 287–296, Jul. 1985.

[42] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3606–3613.

[43] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[45] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[47] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," 2016, *arXiv:1605.08803*.

[48] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.

SEUNGMI OH received the B.S. degree in electronic and electrical engineering from Ewha Womans University, Seoul, South Korea, in 2020, where she is currently pursuing the Ph.D. degree. Her research interest includes deep learning in machine vision.

JEONGTAE KIM (Member, IEEE) received the B.S. and M.S. degrees in control and instrumentation engineering from Seoul National University, Seoul, South Korea, in 1989 and 1991, respectively, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 2004. From 1991 to 1998, he was with Samsung Electronics, South Korea, where he was engaged in the development of digital camcorders and TVs. Since 2004, he has been with the Department of Electronic and Electrical Engineering, Ewha Womans University, Seoul, as a Professor. His research interests include machine learning, computer vision, and radar-signal processing.