IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 12 August 2024, accepted 28 August 2024, date of publication 2 September 2024, date of current version 11 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3453371

RESEARCH ARTICLE

Environmentally-Robust Defect Classification With Domain Augmentation Framework

SUNGHO LEE AND JAEWOONG SHIM
Department of Data Science, Seoul National University of Science and Technology, Nowon-gu, Seoul 01811, Republic of Korea

Corresponding author: Jaewoong Shim (jaewoong@seoultech.ac.kr)

This work was supported in part by Korea Institute for Advancement of Technology (KIAT) through Korean Government [Ministry of Trade, Industry and Energy (MOTIE)] (the Competency Development Program for Industry Specialist) under Grant P0017123, and in part by the National Research Foundation of Korea (NRF) through Korean Government (MSIT: Ministry of Science and ICT) under Grant RS-2022-00165783.

ABSTRACT Visual defect classification is a critical process in manufacturing systems, aiming to achieve high-quality production and reduce costs. Although deep learning-based defect classification models have achieved significant success, their performance can be significantly diminished due to 'environment shifts'—variations in manufacturing environments across multiple production lines. To address this challenge, we propose a domain augmentation framework to construct an environmentally-robust defect classification model, delivering high performance across various manufacturing environments using a training dataset from only a single production line. In our framework, each environment is treated as a separate domain, and multiple augmented domains are created using image transformation functions. Subsequently, a defect classification model is trained using a multi-source domain generalization (DG) method with these augmented domains. This approach mitigates the single-source DG problem to a multi-source DG problem, enabling the adoption of multi-source DG methods, which leads to performance improvements. The effectiveness of the proposed framework is demonstrated through experiments on a dataset provided by a Korean manufacturing company.

INDEX TERMS Visual defect classification, environment shift, domain generalization, image augmentation, manufacturing system.

I. INTRODUCTION

Defect classification is a vital process in manufacturing process that involves not only the detection of defects but also the classification of defect types. This process prevents defective products from being shipped and provides clues to the cause of defects through the analysis of defect types, significantly contributing to product quality assurance, reduction of product recall costs, and enhancement of customer satisfaction. Traditional defect classification involved direct human visual inspection of the produced goods, which incurs considerable time expenditure, decreased inspection accuracy due to the possibility of human error, and a finite volume of goods that can be inspected due to the daily labor time constraints of humans. To overcome these limitations, automatic defect classification was required [1], [2].

In recent years, considerable research has been conducted to apply machine learning to automate defect classification [3]. A classification model is constructed by learning from a training dataset composed of previously collected images labeled with their defect type. The trained model is then used to classify new product images into defect types. To utilize traditional machine learning algorithms on defect classification, feature extraction from images using various methods must precede [4], [5]. However, with recent advancements in deep learning, the need for separate feature extraction has been eliminated, and this has demonstrated excellent performance in visual defect classification.

In manufacturing systems, the same product is typically produced through multiple production lines. Each production line, however, has slightly different manufacturing environ-

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

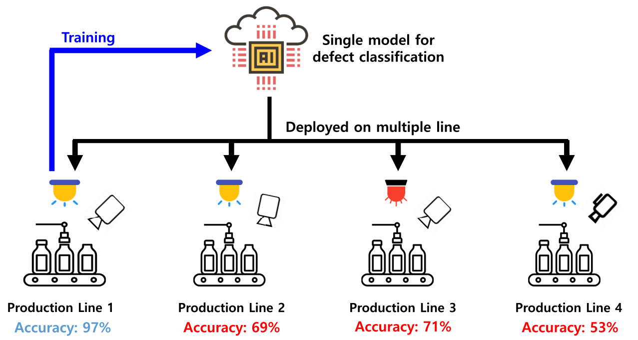

FIGURE 1. Performance degradation of defect classification model in different manufacturing environments.

ments, resulting in variations in elements such as lighting intensity and type, camera positioning and angles, and so on. As a result, the images of the products captured are subtly different from each other. Moreover, even within the same production line, the imaging environment can change slightly over time. We will refer to these variations, which are not present in the training data, as 'environment shifts'. Such shifts can significantly degrade the performance of deep learning-based defect classification models, due to their underlying assumption of a consistent environment for prediction. Therefore, as illustrated in Figure 1, a defect classification model built on a single production line generally exhibits lower performance when deployed on multiple production lines.

Ideally, a unique defect classification model should be built and deployed for each production line, and the model should be frequently retrained. However, due to high costs, it is practically challenging to acquire sufficient labeled training data for each production line to build an individual model. Especially, it becomes even more difficult to acquire labeled training data and build a model when a new production line is established. In real-world manufacturing systems, it is common to obtain a training dataset from a single environment due to various cost constraints. Therefore, there is a need for a robust defect classification model capable of overcoming environment shifts and delivering high performance across various manufacturing environments, even if it is built using labeled datasets from only a single production line.

To create an environmentally-robust defect classification model, domain generalization (DG) methods can be utilized, treating each different manufacturing environment as a separate domain [6], [7]. DG is a research field that aims to overcome the domain gap between training and test data by constructing a domain-invariant model [8], [9]. One of the most common and basic methods is to augment training images during the training procedure. Although various DG methods have been developed, most of them assume a multi-source DG situation where data from multiple domains are available in the training data. Training a model using data from only a single production line corresponds to a single-source DG situation where data from only a single domain is available. Unfortunately, the single-source DG problem is much more challenging than the multi-source DG problem, and therefore, the methodologies that can be attempted are limited [8], [10], [11].

In this paper, we propose a domain augmentation framework for the construction of a robust defect classification model that can achieve high performance in various manufacturing environments using a training dataset from only a single production line. The proposed framework first applies several image transformation functions to the training dataset, designating each augmented dataset as a different domain to create multiple domains. A defect classification model is then trained using a multi-source DG method with these augmented domains. Although the problem we aim to solve originates from a single-source situation, our framework effectively transforms it into a multi-source situation, enabling the use of multi-source DG methods, which results in performance improvements. This is a general framework that can be utilized with any multi-source DG method and image transformation function, which can be selected based on prior knowledge of potential environment shifts. The effectiveness of the proposed framework is investigated through experiments using a dataset provided by a Korean manufacturing company.

The main contributions of the proposed framework are summarized as follows.

- The framework can overcome environment shifts and achieve high defect classification performance on new production lines using a dataset from only a single existing production line, thus eliminating the need for additional data collection or model training costs.
- It alleviates the challenging single-source DG problem by converting it into a multi-source DG problem. This allows for the utilization of a wider range of methods, potentially leading to performance improvement.
- The proposed framework is straightforward yet effective and can be utilized with any image transformation function and any multi-source DG method. This flexibility allows for selection based on prior knowledge of potential environment shifts, thereby enhancing its applicability.

The rest of this paper is structured as follows. In section II, we offer a thorough review of the relevant literature. In section III, we present our proposed framework. In section IV and section V, we outline the experimental setup and discuss the results, respectively. Finally, the conclusions of our research are summarized in section VI.

II. RELATED WORK

A. IMAGE AUGMENTATION

Deep learning technologies have recently achieved remarkable results across a wide range of applications. Particularly, they have seen significant success in the field of computer vision, which has been made possible by the construction of large labeled datasets like ImageNet [12]. However, creating such large labeled datasets in real-world situations, such as in manufacturing, is very challenging due to various practical constraints. Image augmentation is one of the primary methods to overcome these challenges. By applying various image transformations that can preserve labels to a given dataset, image augmentation effectively increases the amount of training data. This alleviates the problem of having a small training dataset and improves the generalization performance of the trained model. In fact, image augmentation has been extensively studied for defect detection and classification in manufacturing systems [13], [14].

Image augmentation methods can be divided into two main types: generative model-based methods and explicit transformation-based methods [15]. Generative model-based methods involve using another deep learning architecture to learn the distribution of training data and then generate new images based on the trained model. Representative architectures for generative models in this area include variational autoencoders (VAE) [16], generative adversarial networks (GAN) [17], and diffusion models [18]. These methods have the advantage of providing appropriate transformations automatically without an explicit function. However, they require high computational cost and a substantial amount of dataset needed for training the generative model. Moreover, they generate images only within the distribution range of the original dataset, making it difficult to achieve diversity. On the other hand, explicit transformation-based methods change given training images by applying various existing transformations. This includes 1) geometric transformations such as rotation and flipping, 2) photometric transformations such as adjusting brightness, contrast, and saturation, 3) noise injection, and 4) applying filters such as the Gaussian blur [19] and Laplacian filters [20]. These augmentation methods are affordable and can be effective when selective transformation methods are used based on prior knowledge.

Depending on when the transformation is applied to the training images, image augmentation can be categorized into offline augmentation and online augmentation [21]. Offline augmentation is a method where augmentation is applied to the entire training set before model training begins. This method requires additional storage space and has the disadvantage of reduced diversity compared to online augmentation. Therefore, in most applications, online augmentation is primarily used, where image transformations are applied during model training dynamically every epoch to greatly enhance the diversity of the data. However, in our framework, we have adopted the offline augmentation method to create augmented domains with explicit domain labels.

B. DOMAIN GENERALIZATION

Deep learning models typically assume that the training and test datasets are drawn from the same distribution. However, in real-world applications, mismatches between the distribution of the training dataset and the test dataset are common [22]. In such situations, the performance of deep learning models trained using conventional methods tends to decline. To overcome this, DG method has been studied. The goal of DG is to build a model that can perform well on test sets from various distributions (target domains), using only the training data from some specific distributions (source domains) [8], [23], [24].

DG can be categorized into single-source DG and multi-source DG, based on the number of domains in the training dataset. Unlike multi-source DG, where the training dataset is composed of multiple domains, single-source DG obtains the training dataset from only one domain. Naturally, single-source DG is more challenging than multi-source DG, and therefore, the methodologies studied are relatively scarce [8], [10], [25]. Nevertheless, in many real-world situations, obtaining a labeled training set from multiple domains is difficult due to data acquisition issues, making single-source DG more practical.

DG methods can be divided into three strategies [8]:

- Augmentation based strategy: This strategy involves transforming the original image data to simulate domain shifts [26], [27]. For instance, MixStyle [26] enhances generalization performance by mixing training images that have the same label but belong to different domains.
- Regularization strategy: This strategy aims to minimize discrepancies across domains by constraining the model's weights and complexity during training [28], [29], [30]. VReX [30] introduces a penalty to the loss function to reduce differences in risk across training domains, thereby reducing the model's sensitivity. GroupDRO [29] assigns individual weights to the loss for each domain to minimize the worst-case loss over a set of domains. IBN-Net [28] employs instance normalization and batch normalization to limit the learning of domain-specific information. Due to their inherent characteristics, both VReX and IBN-Net can be seamlessly applied in a single-source DG situation.
- Domain alignment strategy: The essence of this strategy is to reduce the discrepancy between source domains, aiming to learn domain-invariant information [31], [32], [33]. DANN [32] fosters the training of domain-invariant features by using adversarial training between a domain classifier and a class classifier. Deep CORAL [31] trains a nonlinear function that can align correlations of layer activations between domains. DAN [33] leverages an adaptation layer to extract domain-invariant features. Additionally, causal learning could be adopted, which focuses on resolving mismatches in causal relationships between domains to improve the model's robustness. By identifying and utilizing causally relevant features, this approach can enhance the generalization capability of the model [22], [25], [34], [35].

III. PROPOSED FRAMEWORK

The objective of this study is to construct a defect classification model that can robustly deliver high performance across various manufacturing environments. In real-world manu-

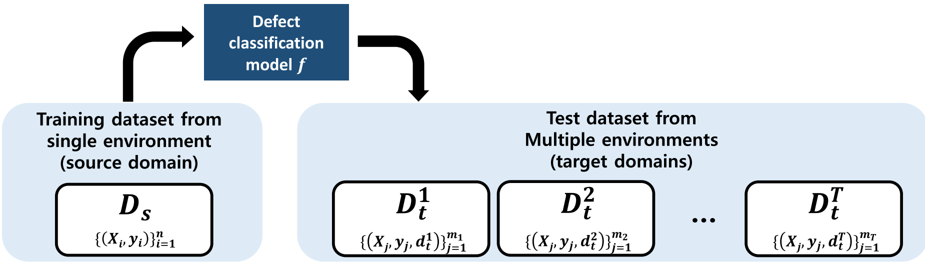

FIGURE 2. Formulation of the problem addressed in this study.

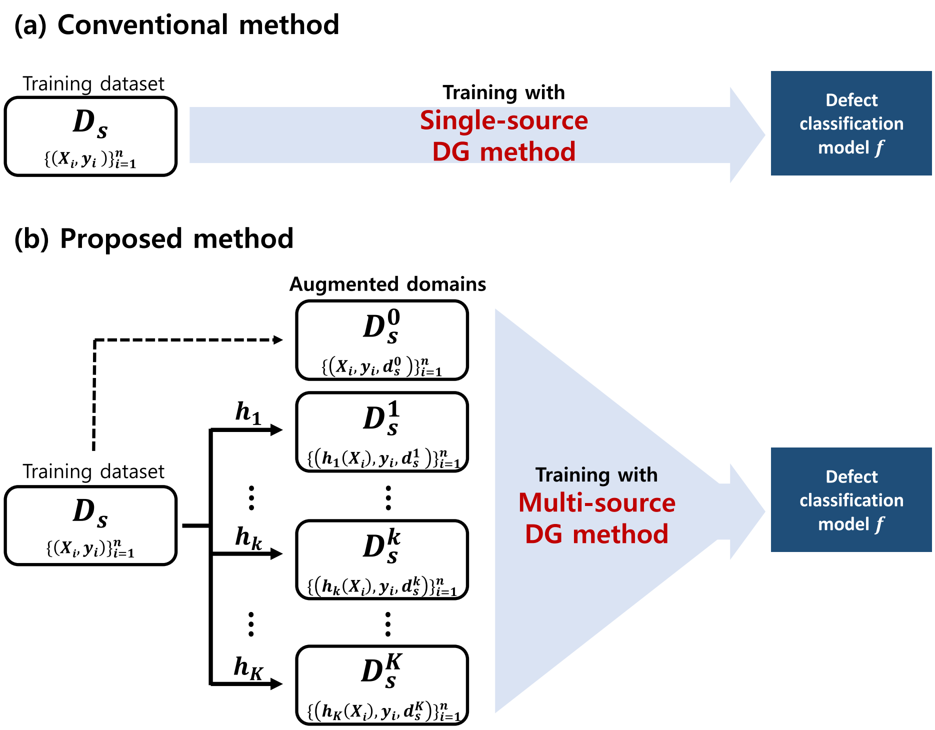

FIGURE 3. Schematic comparison of conventional method and proposed method.

facturing systems, it is common to obtain training dataset from a single environment due to various cost constraints. Therefore, we assume that the training dataset consists of only one environment, but the test dataset consists of multiple environments. A defect classification model is constructed using a training dataset from a single environment, and applied to multiple manufacturing environments. For the purposes of this problem resolution, we will treat each different manufacturing environment as a separate domain, as depicted in Figure 2. A list of notations used in this paper is depicted in Table 1.

The conventional approach in this situation is depicted in Figure 3(a). The simplest and most widely used method is image augmentation, which involves applying various transformations to the training image X during the model training process. In addition to this, several single-source DG methods can be used to train the model f. However, single-source DG methods are relatively limited in variety compared to multi-source DG methods, making it challenging to achieve high performance.

TABLE 1. Notations used in this paper.

Notation	Description
X_i	Image of the i-th product
y_i	Label (defect type) of the i-th product
D_s	Training dataset
D_s^k	Dataset of k-th augmented domain
D_t	Test dataset of target domain
d_s^k	Domain label of k-th augmented domain
d_t	Domain label of target domain
n	Number of instances in the source domain
m	Number of instances in the target domain
K	Number of augmented domains
h_k	Image transformation function for k-th augmented domain
f	Defect classification model

We propose a domain augmentation framework for the construction of a environmentally-robust defect classification model. Figure 3(b) illustrates the procedure of the proposed

Algorithm 1 Domain Augmentation Framework

input: training dataset D_s, Image transformation functions $h_1, h_2, \ldots, h_K$
output: defect classification model f
procedure
  $D_{tr} \leftarrow \{(X_i, y_i, d_s^0) | (X_i, y_i) \in D_s\}$
  for $k = 1, \ldots, K$ **do**
    $D_{tr} \leftarrow D_{tr} \cup \{(h_k(X_i), y_i, d_s^k) | (X_i, y_i) \in D_s\}$
  end for
  Train f with D_{tr} using multi-source DG method
end procedure

framework. This framework involves a two-step procedure, which will be elaborated in the next subsections: 1) Application of image transformation functions to generate multiple augmented domains. 2) Training a defect classification model via multi-source DG methods by using the various augmented domains. By augmenting the domain with image transformation functions, we can mitigate the single-source DG situation to a multi-source DG situation. This allows us to utilize multi-source DG methods, potentially leading to performance improvements. This strategy effectively broadens the variety of methodologies available for model training, overcoming the limitations of single-source DG methods. This is a general framework that can be utilized with any image transformation function and multi-source DG method. The proposed domain augmentation framework is described in Algorithm 1.

A. DOMAIN AUGMENTATION

Given a single domain training dataset $D_s = \{(X_i, y_i)\}_{i=1}^n$, the objective of this step is to create multiple augmented domains for training. The difference from conventional image augmentation methods is that different domain labels d_s^k are assigned for different types of image transformation functions h_k.

Firstly, the original dataset D_s is considered as dataset D_s^0 by assigning all instances with domain label d_s^0, i.e., $D_s^0 = \{(X_i, y_i, d_s^0)\}_{i=1}^n$. Next, to create datasets for multiple domains, for all instances in D_s, we apply the image transformation function h_k to the image X, preserve the label y, and assign the domain label d_s^k to generate $D_s^k = \{(X_i, y_i, d_s^k)\}_{i=1}^n$. If a total of K image transformation functions are used, K datasets are created. As a result, a total of $K+1$ datasets with different domain labels, $D_s^0, D_s^1, D_s^2, \ldots, D_s^K$, are combined to form a new training dataset D_{tr}, which is used to train the model.

For this domain augmentation step, any general image transformation can be used as h. If there is prior knowledge about various manufacturing environments, image transformation functions related to that can be included and utilized on this step.

B. MULTI-SOURCE DOMAIN GENERALIZATION

The new training dataset D_{tr}, created in subsection III-A, contains $K+1$ different domains, with each instance having its corresponding domain label. As the original single-source DG situation has been circumvented to a multi-source DG situation, we now employ a multi-source DG method for training the defect classification model f. For this step, any general multi-source DG method can be utilized. This allows for a wide range of methodologies to be employed, enhancing the potential for achieving a high-performance model.

IV. EXPERIMENTAL DESIGN

A. DATASET DESCRIPTION

TABLE 2. Description of each defect class.

class	description
OK	normal product with no defect
Scratch	scraped or dug into by a sharp object
FM	foreign materials in pin hole
Pin	pin bent or deformed
Dent	hollow on black surface
Glue	some sticky on black surface

To demonstrate the effectiveness of the proposed framework, we used a dataset provided by a Korean manufacturing company. This dataset comprises images of D-Sub connectors, each with a size of (512, 288), and each image corresponds to one of six classes: OK, Scratch, FM, Pin, Dent, and Glue. Representative samples for each class are illustrated in Figure 4, and a detailed description of each class is provided in Table 2.

This D-Sub connector dataset contains images taken in a **Default** environment, and it also contains images from various other shooting environments. Specifically, there are three environmental factors that can affect the images: **Color** (changes due to light scattering), **Brightness** (changes due to illumination levels), **Focus** (changes due to camera focus). Based on a **Default** environment, each factor independently has four configuration settings depending on its intensity. Representative examples of each configuration settings are shown in Figure 5. With the four configuration settings for each of the three factors, a total of 12 different environments are provided. Along with the **Default** environment, there are a total of 13 environments.

This dataset is structured into training, validation, and test sets to align with the problem situation we are considering. The training and validation datasets contain the images from only the **Default** environment. In contrast, the test dataset includes the images from all 13 environments. Specifically, 40% of the test set corresponds to the **Default** environment, while the remaining 60% is evenly distributed across the other 12 environments, spanning all classes. Thus, high performance on this test dataset would demonstrate the model's robustness against various environments in

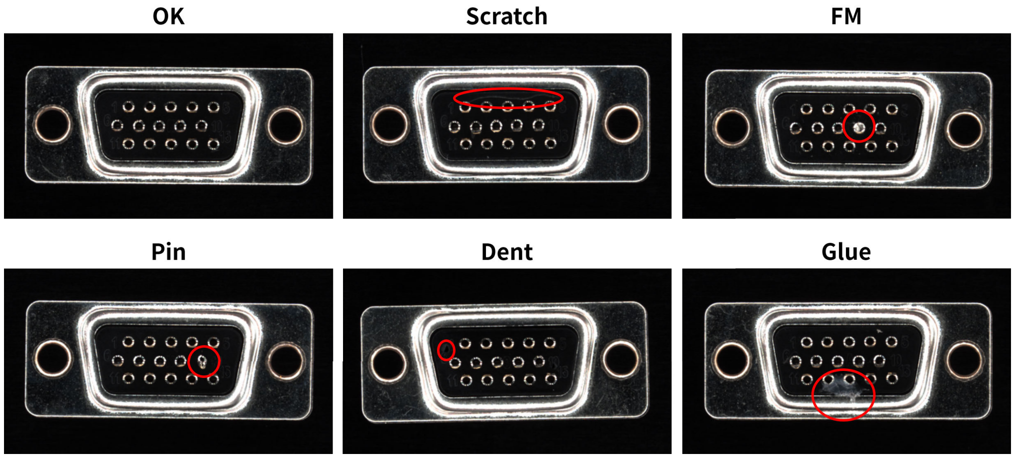

FIGURE 4. Representative examples of each defect class. For clarity, the defect area has been indicated with a red circle.

FIGURE 5. Representative examples of each 13 different environments from OK class. Based on the **Default** environment, there are three environmental factors (**Color**, **Brightness**, and **Focus**), each with four configuration settings depending on its intensity.

real-world manufacturing situations. The detailed distribution of the datasets is summarized in Table 3.

TABLE 3. Detailed distribution of datasets.

Dataset	Environment	Class						Total
		OK	Scratch	FM	Pin	Dent	Glue	
Training	**Default**	160	20	20	20	20	20	260
Validation	**Default**	40	4	4	4	4	4	60
	Default	400	48	48	48	48	48	
	Color-1	50	6	6	6	6	6	
	Color-2	50	6	6	6	6	6	
	Color-3	50	6	6	6	6	6	
	Color-4	50	6	6	6	6	6	
	Brightness-1	50	6	6	6	6	6	
Test	**Brightness-2**	50	6	6	6	6	6	1600
	Brightness-3	50	6	6	6	6	6	
	Brightness-4	50	6	6	6	6	6	
	Focus-1	50	6	6	6	6	6	
	Focus-2	50	6	6	6	6	6	
	Focus-3	50	6	6	6	6	6	
	Focus-4	50	6	6	6	6	6	

B. EXPERIMENTAL SETTINGS

To demonstrate the effectiveness of the domain augmentation framework, we set four different baseline methods to be compared.

- **Vanilla**: The model was trained based on the cross-entropy loss of the given training dataset, as in a conventional situation that does not consider environment shifts. It does not involve any image augmentation or DG methods.
- **Offline aug**: The model was trained using only offline image augmentation without any additional DG method. This can be seen as an ablation study from our proposed framework, where only the first step is applied.
- **Online aug**: The model was trained using only online image augmentation without any additional DG method.
- **Single DG**: The model was trained by using a single-source DG method.

For multi-source DG methods to be used for our proposed framework, we adopted Deep CORAL [31], ERM [36], GroupDRO [29], IRM [37], MixStyle [26], IBN-net [28], VReX [30], and DANN [32]. For the baseline **Single DG**, IBN-net and VReX methods were utilized since they are also applicable on single-source situation.

FIGURE 6. Representative examples of each type of image transformation to be used for creating augmented domains.

As an image transformation function h, we adopted the following eight types of transformation from the open-source library, Imgaug.[1] Each transformation will be used for creating augmented domains.

- CoarseDropout: This transformation sets a certain fraction of pixels in images to zero, with a probability ranging from 0.005 to 0.025%.
- AdditiveGaussianNoise: This transformation adds Gaussian noise $N(0, s)$ to each pixel of the image, where s is varied between 0 and 2.55.
- RandomFlip: This transformation randomly applies a flip to input images either horizontally or vertically.
- GaussianBlur: This transformation applies a Gaussian kernel-based blur to the image, with the blur intensity, defined by σ, varying randomly between 0.2 and 2.
- Affine: This transformation first applies random rotations between -25 and 25 degrees to make an rotated image. Then, it creates final image with alpha-blending, $0.25 \cdot$ [rotated image] $+ 0.75 \cdot$ [original image].
- MultiplyHue: This transformation operates in the HSV color space, multiplying the H channel pixel values by a random factor between 0.25 and 4. Then, it also adds a random value to each pixel with EnhanceBrightness.
- GammaContrast: This transformation modifies the contrast of images using a formula $255 \times \left(\frac{v}{255}\right)^{\gamma}$, where v represents the pixel value and γ is randomly sampled from [0.4, 2].
- HomomorphicFilter: This transformation, called Homomorphic Filtering [38], works by balancing light and dark areas in an image, making it clearer and more detailed by adjusting light levels and enhancing contrasts.

[1] https://github.com/aleju/imgaug

Representative examples of these transformations are shown in Figure 6. For the main results presented in subsection V-A, three methods - 'MultiplyHue,' 'Affine,' and 'RandomFlip' - were utilized to create augmented domains. These methods were chosen because they demonstrated the highest performance for the proposed framework. A detailed discussion regarding the types and number of transformation functions will be presented in subsection V-B.

In all experiments, ResNet50 was adopted as a backbone network architecture. The SGD optimizer was set with a learning rate of 0.01, and cosine annealing was applied. The batch size was set to 32. Training was conducted for a total of 100 epochs, and early stopping was applied if there was no improvement in validation loss for 4 consecutive epochs. All other settings were consistent with the default settings as specified in the original papers. All experiments were repeated 5 times and the average was reported. We used PyTorch 1.71, running on a GPU of RTX 3090.

V. RESULTS AND DISCUSSION

A. MAIN RESULTS

Table 4 presents the results of comparing the proposed framework with four other baselines. As shown in the table, the proposed domain augmentation framework with IBN-net demonstrated the highest overall accuracy of 0.8941 on the test dataset. Moreover, all other methods using the domain augmentation framework also showed relatively high performance, with the average accuracy of the seven methods being 0.8705, which was more effective compared to all other baseline methods. The domain augmentation framework demonstrated significantly better performance, even though the actual amount of data used for training was the same as that of **Offline aug**, which can be seen as an ablation study of the proposed framework. Therefore, it can be concluded that the improved robustness of the model is not merely due to the increased dataset size, but rather a result of the effectiveness of our framework. When examining the accuracy for each environment in Table 4, all methodologies tend to decrease in accuracy when there are changes in **Color**, **Brightness**, and **Focus** compared to the **Default** environment directly learned from the training set. However, the decrease was smallest when using the domain augmentation framework. This demonstrates that the domain augmentation framework is effective in constructing a robust defect classification model that can handle environmental changes.

Among the multi-source DG methods employed in the proposed framework, IBN-net [28] showed the best results. It effectively learns features that are invariant to subtle

TABLE 4. Comparison of accuracy across each environment in the test dataset (mean ± standard deviation).

Environmental factors	Vanilla	Offline aug	Online aug	Single DG			Domain augmentation framework (proposed)							
				IBN-net	VReX	Avg	Deep Coral	ERM	GroupDRO	IBN-net	MixStyle	VReX	DANN	Avg
Default	87.0±3.14	77.01±2.24	79.01±1.02	88.46±1.96	86.21±2.64	87.34	88.34±0.75	91.23±1.20	91.46±3.67	**93.43±2.42**	92.11±2.42	84.68±1.94	91.81±2.77	90.44
Color	66.84±4.49	73.14±2.50	72.16±1.47	86.81±2.99	81.46±2.24	84.14	88.16±0.98	89.13±0.63	89.46±3.26	**93.19±2.73**	93.07±3.49	83.98±1.26	88.17±2.38	89.31
Brightness	63.22±3.12	72.19±3.03	73.44±1.36	76.03±4.15	72.24±7.41	74.14	86.91±1.02	87.5±3.37	87.66±2.87	90.17±2.10	88.18±2.10	81.67±0.89	**90.39±2.95**	87.5
Focus	52.49±16.75	69.71±2.06	70.67±2.33	**85.64±3.19**	79.07±3.16	82.16	84.31±1.17	81.64±1.72	81.69±3.19	83.88±3.52	85.41±2.15	80.61±3.72	80.67±19.2	82.6
Overall	73.29±3.66	78.75±1.67	82.1±1.36	84.56±1.41	80.2±1.47	82.38	87.61±1.10	87.36±0.98	87.31±2.79	**89.41±2.42**	88.31±1.90	84.54±2.04	86.84±2.29	87.05

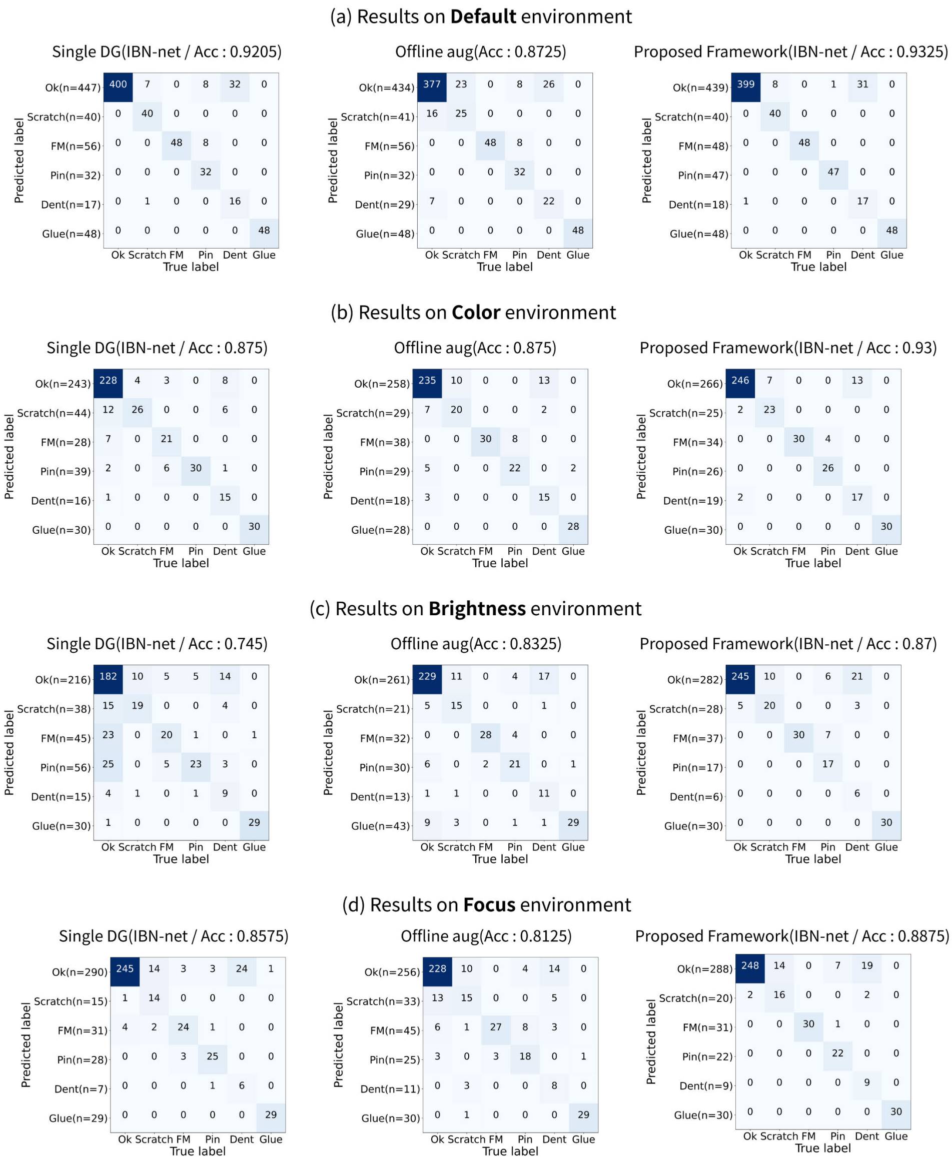

FIGURE 7. Confusion matrices for each method on the test dataset from a single run out of five repeated experiments: (a) results on the Default environment, which is the same environment as the training dataset. (b) results on the Color environments. (c) results on the Brightness environments. (d) results on the Focus environments.

variations by utilizing both instance normalization and batch normalization. This makes it particularly effective for addressing the subtle environmental shifts we are considering, regardless of the type of image transformation functions employed in our framework. On the other hand, VReX showed the worst results. This is likely because the VReX assumption [30]—that variation across source domains is representative of test set variation—does not align well with our situation, where variations between domains are subtle yet diverse.

Figure 7 presents the confusion matrices for **Offline aug**, **Single DG** (IBN-net), and the proposed framework

TABLE 5. Performance comparison of each image transformation function within the domain augmentation framework. To assess the effect of individual image transformation functions, only a single augmented domain was added to the original source domain. (mean ± standard deviation).

DG method	Image transformation function							
	Coarse Dropout	Additive GaussianNoise	Random Flip	Gaussian Blur	Affine	Multiply Hue	Gamma Contrast	Homomorphic Filter
Deep Coral	80.73±1.43	80.14±2.62	83.66±3.23	80.04±3.80	84.58±1.83	**88.41±0.80**	85.15±1.95	84.47±2.26
ERM	80.46±3.41	79.53±4.05	84.51±2.68	81.72±1.85	81.45±3.85	**89.60±1.82**	82.79±1.82	83.19±1.24
GroupDRO	81.14±3.51	74.13±4.43	85.93±1.71	80.90±2.92	85.90±2.10	**87.54±3.39**	63.77±1.04	83.64±2.01
IBN-Net	77.70±6.53	79.29±2.53	85.68±1.60	78.75±4.69	86.65±2.39	**86.67±4.28**	84.64±1.68	83.41±2.63
Mix-Style	83.56±1.96	80.49±1.36	85.13±2.5	84.87±3.09	86.43±1.25	**87.66±1.97**	83.96±1.37	86.29±2.04
VReX	80.65±3.09	71.00±12.19	77.12±5.15	**83.08±2.15**	79.48±4.14	81.84±4.28	54.06±17.36	76.72±4.19
DANN	81.08±3.32	78.47±5.31	84.11±3.53	81.17±4.90	83.95±2.64	**86.87±3.14**	83.88±2.31	85.67±1.96
Avg	80.76	77.58	83.73	81.51	84.06	86.94	76.89	83.34
Avg Rank	6.14	7.29	3.43	5.43	3.43	1.14	5.29	3.86

(IBN-net) on the test set from a single run out of five repeated experiments. The top part shows the results for the **Default** environment, while (b) through (d) shows the results for each environmental factor. Similar to the results in Table 4, the proposed framework demonstrates relatively robust performance compared to other baselines in all environments. Notably, while **Single DG** exhibits competitive performance in the **Default** environment comparable to the proposed framework, it experiences a more significant performance decrease in other environments. While the proposed framework demonstrated strong performance across most environments, we observed that a significant number of Dent and Scratch instances were misclassified as OK, particularly in the **Focus** environment. Figure 8 presents these misclassified samples. As can be seen, the Dent and Scratch defects are so tiny that they become nearly indistinguishable in the **Focus** environment, making them appear almost as if they belong to the OK class, even to the human eye. Addressing such tiny defects under environmental shifts is a significant challenge, highlighting the need for further methodological development.

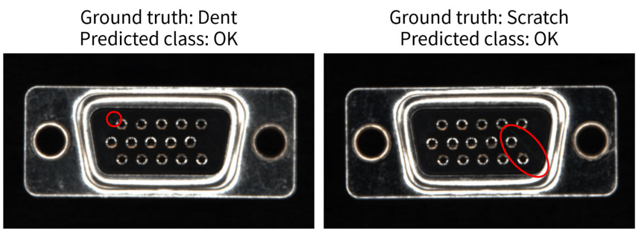

FIGURE 8. Representative misclassified samples of the proposed framework in the Focus environment.

B. ABLATION STUDY

In this subsection, we examined how the performance of the domain augmentation framework varies depending on the type of image transformation function h and the number of the functions K. Firstly, we investigated the effect of each image transformation function h within the context of our framework. To evaluate the effect of each of the eight image transformation functions introduced in subsection IV-B, we trained models with domain augmentation framework where $K = 1$, i.e., creating a single augmented domain with each of the eight image transformation functions.

As shown in Table 5, the effectiveness of each transformation function type can be ranked as follows: MultiplyHue, Affine, Randomflip, HomomorphicFilter, GammaContrast, GaussianBlur, CoarseDropout, and AdditiveGaussianNoise. This order was established based on the average rank, and in the event of a tie, as was the case with Affine and Randomflip, the average accuracy was used. MultiplyHue showed the highest performance as a single transformation function in the domain augmentation framework. MultiplyHue can create an augmented domain very similar to the change of the **Color** environmental factor, which appears to have had a positive effect. Therefore, if there is prior knowledge about the actual manufacturing environment, it can be reflected in the image transformation function of the domain augmentation framework to effectively improve performance.

Subsequently, we analyzed the performance changes depending on the number of newly generated domains K, that is, the number of transformation function h. Following the effectiveness order of image transformation functions identified in the previous experiment, we adjusted K by incrementally adding augmented domains. For $K = 1$, only MultiplyHue was utilized. For $K = 2$, Affine was added. For $K = 3$, Randomflip was included. This pattern persisted until $K = 8$, where all eight transformation functions were employed.

The result is depicted in Figure 9. The performance changes for each multi-source DG method are shown as solid lines, and the average is depicted as a red dotted line. The IBN-net, which showed the highest performance, performed best when K was 2, 3, or 4. Observing the average performance trend, there is an initial increase in performance as K grows up to 3, followed by a gradual decline due to the inclusion of less effective transformation functions. However, the extent of this performance variation is relatively minor, indicating that the performance is not significantly impacted by the number of augmented domains employed.

The findings from our series of experiments on the proposed framework can be summarized as follows:

- Utilizing appropriate image transformation functions based on prior knowledge is important.

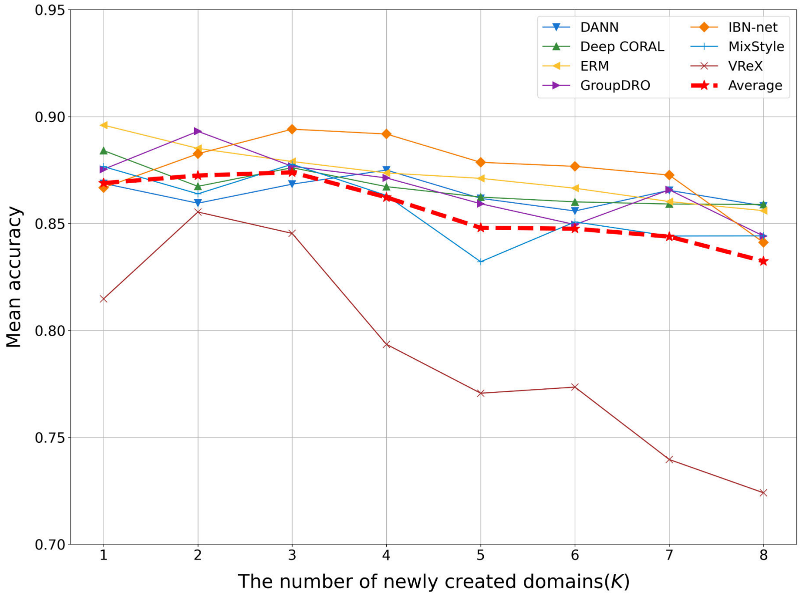

FIGURE 9. Performance of the domain augmentation framework as the number of newly created domains K increased.

- Three or four augmented domains are sufficient to achieve high performance, as adding more augmented domains does not provide significant additional improvements.
- The proposed framework can accommodate any multi-source DG method; IBN-net methods were effective, while VReX methods were relatively less effective in overcoming environment shifts.

VI. CONCLUSION

This study primarily focuses on the performance degradation of the defect classification model across diverse manufacturing environments. Aiming to develop an environmentally-robust defect classification model across diverse production lines, we introduced the domain augmentation framework. In this framework, each environment is treated as a separate domain, and by augmenting the domain using image transformation functions, the single-source DG problem is mitigated to a multi-source DG problem. Consequently, this allows for the application of more diverse and effective multi-source DG methods, overcoming the limitations inherent in single-source DG methods. This is a versatile framework that can be utilized with any image transformation function and multi-source DG method, making it capable of incorporating prior knowledge specific to manufacturing environments. Through experiments using real-world data, we demonstrated that the proposed framework can achieve high defect classification performance across diverse environments.

In future work, we plan to explore automated methods for identifying appropriate image transformation functions for our framework. This could include the use of reinforcement learning to find optimal augmented domains [39] or the introduction of conditional GANs [40] and diffusion models [15] to create augmented domains and enhance diversity. We anticipate that integrating these advanced techniques into our framework will effectively address environmental shifts in defect classification, as well as demonstrate broader applicability in managing general domain shifts. Moreover, we aim to develop a test time domain adaptation method to maximize the use of limited information from new manufacturing environments. Specifically, we plan to devise a methodology that assists the model in immediately adapting to a new environment at test time by providing limited information about the new environment in text form.

REFERENCES

[1] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine learning for industrial applications: A comprehensive literature review," *Expert Syst. Appl.*, vol. 175, Aug. 2021, Art. no. 114820.

[2] C. Liu, S. Li, F. Chang, and Y. Wang, "Machine vision based traffic sign detection methods: Review, analyses and perspectives," *IEEE Access*, vol. 7, pp. 86578–86596, 2019.

[3] S. B. Jha and R. F. Babiceanu, "Deep CNN-based visual defect detection: Survey of current literature," *Comput. Ind.*, vol. 148, Jun. 2023, Art. no. 103911.

[4] J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, "Machine learning-based imaging system for surface defect inspection," *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 3, no. 3, pp. 303–310, Jul. 2016.

[5] Q. Li, Z. Luo, H. Chen, and C. Li, "An overview of deeply optimized convolutional neural networks and research in surface defect classification of workpieces," *IEEE Access*, vol. 10, pp. 26443–26462, 2022.

[6] D. Wang, Q. Liu, D. Wu, and L. Wang, "Meta domain generalization for smart manufacturing: Tool wear prediction with small data," *J. Manuf. Syst.*, vol. 62, pp. 441–449, Jan. 2022.

[7] M. Azamfar, X. Li, and J. Lee, "Deep learning-based domain adaptation method for fault diagnosis in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 3, pp. 445–453, Aug. 2020.

[8] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4396–4415, Apr. 2023.

[9] H. Zheng, R. Wang, Y. Yang, Y. Li, and M. Xu, "Intelligent fault identification based on multisource domain generalization towards actual diagnosis scenario," *IEEE Trans. Ind. Electron.*, vol. 67, no. 2, pp. 1293–1304, Feb. 2020.
[10] Z. Wang, Y. Luo, R. Qiu, Z. Huang, and M. Baktashmotlagh, "Learning to diversify for single domain generalization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 814–823.
[11] X. Peng, F. Qiao, and L. Zhao, "Out-of-Domain generalization from a single source: An uncertainty quantification approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 3, pp. 1775–1787, 2022.
[12] 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.
[13] J. P. Yun, W. C. Shin, G. Koo, M. S. Kim, C. Lee, and S. J. Lee, "Automated defect inspection system for metal surfaces based on deep learning and data augmentation," *J. Manuf. Syst.*, vol. 55, pp. 317–324, Apr. 2020.
[14] S. Jain, G. Seth, A. Paruthi, U. Soni, and G. Kumar, "Synthetic data augmentation for surface defect detection and classification using deep learning," *J. Intell. Manuf.*, vol. 33, no. 4, pp. 1007–1020, Apr. 2022.
[15] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
[16] D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014.
[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.
[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.
[19] E. S. Gedraite and M. Hadad, "Investigation on the effect of a Gaussian blur in image filtering and segmentation," in *Proc. ELMAR-*, Sep. 2011, pp. 393–396.
[20] S. Paris, S. W. Hasinoff, and J. Kautz, "Local Laplacian filters: Edge-aware image processing with a Laplacian pyramid," *ACM Trans. Graph.*, vol. 30, no. 4, p. 68, 2011.
[21] Z. Tang, Y. Gao, L. Karlinsky, P. Sattigeri, R. Feris, and D. Metaxas, "OnlineAugment: Online data augmentation with less domain knowledge," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 313–329.
[22] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proc. IEEE*, vol. 109, no. 5, pp. 612–634, May 2021.
[23] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[24] Y. Zhang, Y. Zhang, Q. Xu, and R. Zhang, "Learning robust shape-based features for domain generalization," *IEEE Access*, vol. 8, pp. 63748–63756, 2020.
[25] X. Xiang, Y. Liu, G. Fang, J. Liu, and M. Zhao, "Two-stage alignments framework for unsupervised domain adaptation on time series data," *IEEE Signal Process. Lett.*, vol. 30, pp. 698–702, 2023.
[26] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with MixStyle," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[27] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 561–578.
[28] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via IBN-Net," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 464–479.
[29] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2019.
[30] D. Krueger, E. Caballero, J. H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation (REx)," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5815–5826.
[31] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 443–450.
[32] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.
[33] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 97–105.
[34] Y. Liu, Z. Xia, M. Zhao, D. Wei, Y. Wang, S. Liu, B. Ju, G. Fang, J. Liu, and L. Song, "Learning causality-inspired representation consistency for video anomaly detection," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 203–212.
[35] T. Liang, T. Liu, J. Wang, J. Zhang, and P. Zheng, "Causal deep learning for explainable vision-based quality inspection under visual interference," *J. Intell. Manuf.*, pp. 1–22, Jan. 2024.
[36] V. Vapnik, *The Nature of Statistical Learning Theory*. Cham, Switzerland: Springer, 1999.
[37] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," 2019, *arXiv:1907.02893*.
[38] M.-J. Seow and V. K. Asari, "Ratio rule and homomorphic filter for enhancement of digital colour image," *Neurocomputing*, vol. 69, nos. 7–9, pp. 954–958, Mar. 2006.
[39] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, "Population based augmentation: Efficient learning of augmentation policy schedules," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2731–2741.
[40] 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.

SUNGHO LEE received the bachelor's degree in industrial engineering (ITM) from Seoul National University of Science and Technology, where he is currently pursuing the master's degree in data science. He interned at the Electronics and Telecommunications Research Institute (ETRI), where he worked on defense ICT Projects. His research interests include domain generalization, active learning, and vision-language models.

JAEWOONG SHIM received the B.S., M.S., and Ph.D. degrees in industrial engineering from Seoul National University, in 2014, 2016, and 2021, respectively. He worked as an Engineer at the Quality Assurance Team, Samsung Electronics, and the AI Team Lead at RTM. He is working as an Assistant Professor of industrial engineering with Seoul National University of Science and Technology. His research interests include cost-effective learning, anomaly detection, and active learning.