sensors

MDPI

Article

Two-Stream Network One-Class Classification Model for Defect Inspections

Seunghun Lee [1], Chenglong Luo [1], Sungkwan Lee [2] and Hoeryong Jung [1,*]

[1] Division of Mechanical and Aerospace Engineering, Konkuk University, 120 Neungdong-ro, Gwangjin-gu, Seoul 05029, Republic of Korea; erioer95@konkuk.ac.kr (S.L.); luo0611@konkuk.ac.kr (C.L.)
[2] Sambo Technology, 90 Centum Jungang-ro, Haeundae-gu, Busan 48059, Republic of Korea; sales@sambotechnology.co.kr
* Correspondence: junghl80@konkuk.ac.kr; Tel.: +82-2-450-3903

Abstract: Defect inspection is important to ensure consistent quality and efficiency in industrial manufacturing. Recently, machine vision systems integrating artificial intelligence (AI)-based inspection algorithms have exhibited promising performance in various applications, but practically, they often suffer from data imbalance. This paper proposes a defect inspection method using a one-class classification (OCC) model to deal with imbalanced datasets. A two-stream network architecture consisting of global and local feature extractor networks is presented, which can alleviate the representation collapse problem of OCC. By combining an object-oriented invariant feature vector with a training-data-oriented local feature vector, the proposed two-stream network model prevents the decision boundary from collapsing to the training dataset and obtains an appropriate decision boundary. The performance of the proposed model is demonstrated in the practical application of automotive-airbag bracket-welding defect inspection. The effects of the classification layer and two-stream network architecture on the overall inspection accuracy were clarified by using image samples collected in a controlled laboratory environment and from a production site. The results are compared with those of a previous classification model, demonstrating that the proposed model can improve the accuracy, precision, and F1 score by up to 8.19%, 10.74%, and 4.02%, respectively.

Keywords: defect inspection; machine vision; one-class classification; two-stream network

check for updates

Citation: Lee, S.; Luo, C.; Lee, S.; Jung, H. Two-Stream Network One-Class Classification Model for Defect Inspections. *Sensors* **2023**, *23*, 5768. https://doi.org/10.3390/s23125768

Academic Editors: Jianxiong Zhu, Zhijie Xia and Longhui Qin

Received: 18 April 2023
Revised: 14 June 2023
Accepted: 19 June 2023
Published: 20 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Defect inspection is important in the manufacturing industry and is required to ensure consistent product quality and improve the costs and efficiency of the entire manufacturing process. Human visual inspection, however, is time-consuming, labor-intensive, and prone to human errors. In contrast, machine vision inspection using cameras, optics, and inspection software enables fast and robust low-cost inspection. Therefore, it has been increasingly adopted in various manufacturing industries [1–5]. For decades, numerous studies on machine vision inspection have been conducted [6–10], but traditional inspection techniques still face challenges in dealing with variations in environmental conditions and part appearance.

Recently, inspection algorithms integrating artificial intelligence (AI) techniques have shown promise and improved the accuracy and robustness of defect inspection. These algorithms have been employed in various manufacturing industries, including textile [11], fabric [8–10], and steel surface [4,12]. Defect inspection using deep learning algorithms achieved enhanced accuracy and robustness by learning features from the large training dataset. A number of prominent architectures and pre-trained models, such as AlexNet [13], VGGNet [14], ResNet [15], and MobileNet [16], have emerged, and these are accompanied by various techniques to enhance inspection performance. Wei et al. achieved an inspection accuracy of 98.5% using convolutional neural network (CNN)-based algorithms with

image preprocessing, such as noise reduction and binarization, to detect defective products in the textile industry [17]. Yang et al. used the you only look once (YOLO) v5 object detection algorithm to detect and identify welding defects on steel pipes. The proposed model achieved an accuracy of 97.8%, demonstrating its potential for real-time welding defect detection [18]. Kim et al. presented a skip-connected convolutional autoencoder for advanced printed circuit board (PCB) inspection. The proposed unsupervised autoencoder model delivered promising performance, with a detection rate of up to 98% in 3900 defect and non-defect images [19]. Tang et al. proposed a skip autoencoder to improve the accuracy of anomaly detection and address labeling issues. Leveraging a pre-trained feature extractor and skip connections, the proposed method achieved better performance, showing a maximum area under the curve (AUC) of 0.98 [20]. Upadhyay et al. developed a U-Net-based deep learning framework to detect engine defects. They applied a hybrid motion deblurring method for image sharpening and denoising, combined with a customized generative adversarial network (GAN) model, to remove the blur effect based on classic computer vision techniques. The deep learning framework achieved precisions and recalls of over 90% [21]. Yoon Jong-Pil et al. presented a defect classification approach based on a convolutional variational autoencoder (CVAE) and deep CNN for metal surface defect inspection. The proposed conditional CVAE achieved a maximum completion of 0.9969 [22].

Although AI-based inspection provides superior performance compared to traditional methods, several limitations remain in applying this approach to practical situations. One major challenge is the performance degradation caused by data imbalance. AI-based inspection requires a large training dataset. However, practically, the collected data often suffer from class imbalance, where certain classes have considerably fewer samples than others. In defect inspection, collecting sufficient defective samples is difficult because the defect rate is quite low (usually under 1–5%) in general manufacturing processes. To address this issue, various methods have been proposed, including data augmentation [23–25], synthetization [19,26], and an adjustment of the weight or loss function of the network [27]. Wang et al. proposed a novel loss function called 'mean false error' together with its improved version called 'mean squared false error' for deep network training using imbalanced datasets [28]. Mao et al. improved data imbalance by extending the training dataset using a GAN model and achieved up to 86.8% accuracy [29].

One-class classification (OCC), which identifies objects belonging to a specific class given only positive samples of that class, is attracting attention as a solution to this problem [30–40]. Unlike general machine-learning-based classification algorithms, the OCC model aims to learn a classification boundary that separates the target class from other classes in the input space. OCC can thus be utilized effectively to solve data imbalance problems, as it does not require negative samples and can be trained only using positive samples. Shin et al. proposed a one-class support vector machine (SVM) model to detect mechanical defects in electronic devices, achieving up to 93.9% accuracy compared to the multilayer perceptron method [31]. Ruff et al. proposed a deep support vector data description that extracts the similarity between patterns of general categories and new data. The proposed method achieved up to 99.7% average AUCs on MNIST and CIFAR-10 [34]. Lee et al. proposed a one-class deep-learning-based fault-detection module for imbalanced industrial time-series data. Using four different networks, i.e., MLP, ResNet, LSTM, and ResNet-LSTM, for prediction, they achieved an excellent fault prediction accuracy of 96% [36]. Goyal et al. developed a deep robust one-class classification (DROCC) to help address the representation collapse problem. The DROCC achieved an average accuracy of 74.2% using the CIFAR-10 dataset [37].

The representation collapse problem is a major issue in OCC, and it can arise when the diversity of the training data is insufficient, or the data follow a repetitive pattern. In such cases, the decision boundary is fitted too tightly to the training dataset, leading to a decrease in the generalization performance for new data. In practical applications, the

environmental conditions for collecting training and test samples may not be the same, which can lead to false positive errors, resulting in overall performance degradation.

In this paper, we propose a two-stream network OCC model for defect detection that attempts to address the representation collapse problem, which has been a critical issue when applying the OCC model to practical applications. The proposed two-stream network model alleviates the representation collapse problem by introducing two feature extractor networks, i.e., global and local feature extractor networks. The global feature extractor network, which is designed to learn a general feature of the target class, can extract a feature vector that is not affected by variations in environmental conditions. The local feature extractor network is designed to capture features specific to the training dataset, and it extracts the target class-oriented feature vectors. Two feature vectors output from each network are merged and passed through the following classification layer for the final decision. Three types of classification layers, i.e., a one-dimensional (1D) convolution layer, a fully connected layer, and an SVM layer, were tested for the target class classification to determine the optimal classification layer. The proposed two-stream OCC model was verified by using an image dataset obtained from the practical application of automotive airbag bracket inspection. The main contributions of this paper are as follows:

- A two-stream network architecture composed of global and local feature extractor networks is proposed to resolve the representation collapse problem of the OCC model.
- The classification performances of three types of classification layers, i.e., 1D convolution, fully connected, and SVM layers, are described to elucidate the type that yields the optimal classification performance.
- The performance of the proposed OCC model is verified using the practical application of automotive airbag bracket inspection.

2. Materials and Methods

2.1. Two-Stream Network OCC Model

OCC involves training a model using data from a single class and capturing its feature vectors. Although OCC is effective in capturing the distribution of given target data, its ability to recognize new data with different characteristic distributions may be diminished. To address this limitation, which is called representation collapse, this paper proposes a two-stream network OCC model. The main idea is to introduce a global feature extractor network to alleviate the issue of decision boundary collapse relative to the training data. By merging a global feature vector representing object-oriented general characteristics with a class-oriented local feature vector, the two-stream network model prevents the decision boundary from being overfitted to the training data and balances both features to identify an appropriate decision boundary.

Figure 1 shows the two-stream network OCC model proposed in this paper. It consists of two types of feature extractor networks, i.e., global and local feature extractor networks. The global feature extractor network is designed to capture all characteristics of the inspection objects, such as geometrical and topological characteristics. Generally, the global feature is an object-oriented characteristic, and it can be consistently extracted regardless of variations in environmental conditions. The local feature extractor is responsible for extracting the target class-oriented characteristics from the training datasets. The local feature describes the surface characteristics of inspection objects, such as colors and textures. Unlike the global feature, the local feature presented in the image can be influenced by environmental conditions. The two feature vectors obtained from each feature extractor network are merged as a single feature vector and passed through the classification layer.

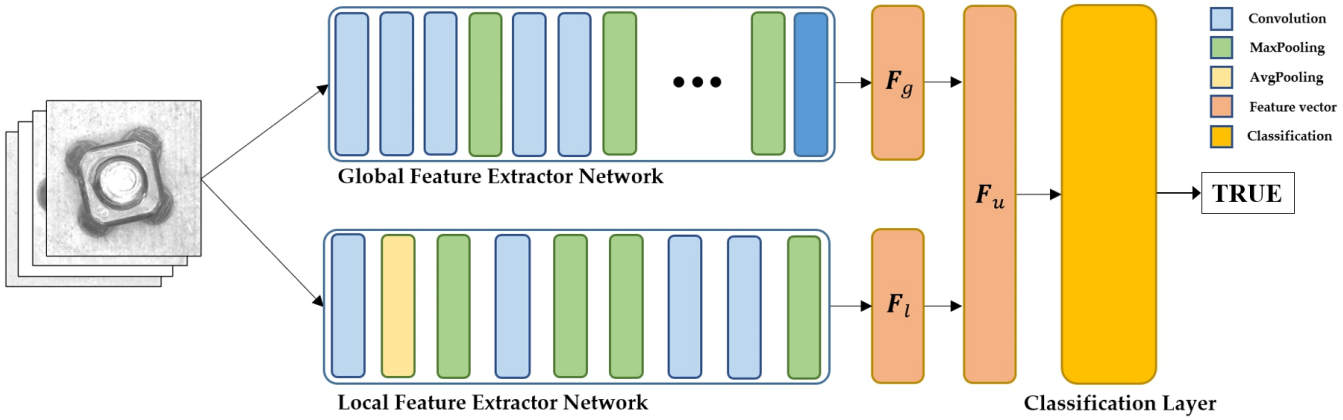


Figure 1. Two-stream network consisting of global and local feature extractor networks followed by a classification layer.

The global feature extractor network is implemented using an Inception V3 network model that consists of a deep neural network architecture including 94 convolution layers and 20,861,480 parameters. It includes three inception modules, which are composed of multiple parallel paths with different filter sizes, to create a rich set of features that capture different aspects of the input image. The inception modules and auxiliary classifiers in the global feature extractor network alleviate the overfitting problem and improve the consistency of feature extraction. The details of the global feature extractor network are presented in Table 1. The global feature extractor network is pre-trained using an ImageNet dataset separately from the other parts of the entire two-stream network. In the entire model training process, the weights of the global feature extractor network are fixed as the pre-trained value to prevent the feature vector from being biased relative to the training dataset. The global feature vector, $\boldsymbol{F}_g$, extracted from the global feature extractor network can be expressed as

$$\boldsymbol{F}_g = \boldsymbol{K}_g * \boldsymbol{I}, \quad \boldsymbol{F}_g \in \mathbb{R}^D, \tag{1}$$

where $\boldsymbol{I}$ represents the image, $\boldsymbol{K}_g$ denotes the global feature extractor network, and D is the dimension of the global feature vector.

Table 1. Detailed configuration of the global feature extractor network.

Type	Stride	Filter Size	Input Size
Conv2d	2	3 × 3, 32	111 × 111 × 32
Conv2d	2	3 × 3, 32	109 × 109 × 32
Conv2d	2	3 × 3, 64	109 × 109 × 64
MaxPooling2d	2	3 × 3, x	54 × 54 × 64
Conv2d	2	3 × 3, 1	52 × 52 × 80
Conv2d	2	3 × 3, 2	26 × 26 × 192
Conv2d	2	3 × 3, 1	26 × 26 × 288
3× Inception	inception module structure [41]		13 × 13 × 768
5× Inception	inception module structure [41]		6 × 6 × 1280
2× Inception	inception module structure [41]		6 × 6 × 2048
MaxPooling2d	2	1	5 × 5 × 2048

The local feature extractor network is composed of four convolution layers and three max-pooling layers including 3,796,480 parameters as presented in Table 2. A simple CNN structure is used for the local feature extractor network to capture the features specific to the target dataset. The local feature extractor network captures the target data-oriented local feature vector $\boldsymbol{F}_l$, which can be determined by applying

$$\boldsymbol{F}_l = \boldsymbol{K}_l * \boldsymbol{I}, \quad \boldsymbol{F}_l \in \mathbb{R}^D, \tag{2}$$

where $\boldsymbol{I}$ represents the image, and $\boldsymbol{K}_g$ denotes the local feature extractor network. The dimension of the local feature vector is identical to that of the global feature vector. The global and local feature vectors are merged as a single feature vector, $\boldsymbol{F}_u$, as follows and passed through the classification layer:

$$\boldsymbol{F}_u = \boldsymbol{F}_g \oplus \boldsymbol{F}_l,\ \boldsymbol{F}_u \in \mathbb{R}^{2D} \tag{3}$$

where $\boldsymbol{F}_u$ is the unified feature vector. $\boldsymbol{F}_u$ is passed through the classification layer to determine the final decision of the defect inspection. Three types of classification layers, including a 1D convolution layer, a fully connected layer, and an SVM layer, were implemented to validate the effect of the classification layer on the overall inspection performance and to identify the optimal classification layer. The details of each classification layer are presented in Table 3.

Table 2. Detailed configuration of the local feature extractor network.

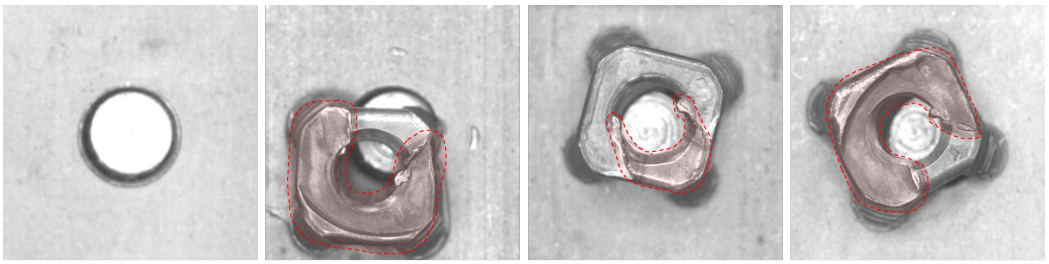
Type	Stride	Filter Size	Output Size
Conv2d	2	7×7	$112 \times 112 \times 512$
MaxPooling2D	2	2×2	$56 \times 56 \times 512$
Conv2d	2	5×5	$28 \times 28 \times 256$
MaxPooling2D	2	2×2	$14 \times 14 \times 256$
Conv2d	1	3×3	$12 \times 12 \times 128$
Conv2d	1	3×3	$10 \times 10 \times 128$
MaxPooling2D	2	2×2	$5 \times 5 \times 128$

Table 3. Detailed configuration of three types of classification layers.

Classification Layers	Architecture			
	Type	Stride	Filter Size	Output Size
1D Convolution Layer	Covn1d	1	1×1	$5 \times 5 \times 128$
	Conv1d	1	1×1	$5 \times 5 \times 64$
	Conv1d	1	1×1	$5 \times 5 \times 1$
Fully Connected Layer	Dense		#of nodes: 128	
	Dense		#of nodes: 128	
	Dense		#of nodes: 1	
SVM Layer	Dense		#of nodes: 128	
	Dense		#of nodes: 128	
	Dense		#of nodes: 1	

2.2. Model Verification

The two-stream network OCC model was verified using the image samples collected by the practical vision inspection system of an automotive airbag bracket. The airbag bracket was manufactured using projection welding, joining a nut on a bracket plate. Faults may have occurred in the welding procedure, resulting in several types of defects such as nut omissions, axial twisting, and surface abnormalities, as shown in Figure 2. These types of defects should be detected by the vision inspection system, and this study verifies the performance of the proposed two-stream network OCC model by evaluating the inspection accuracy using positive and negative airbag bracket samples.

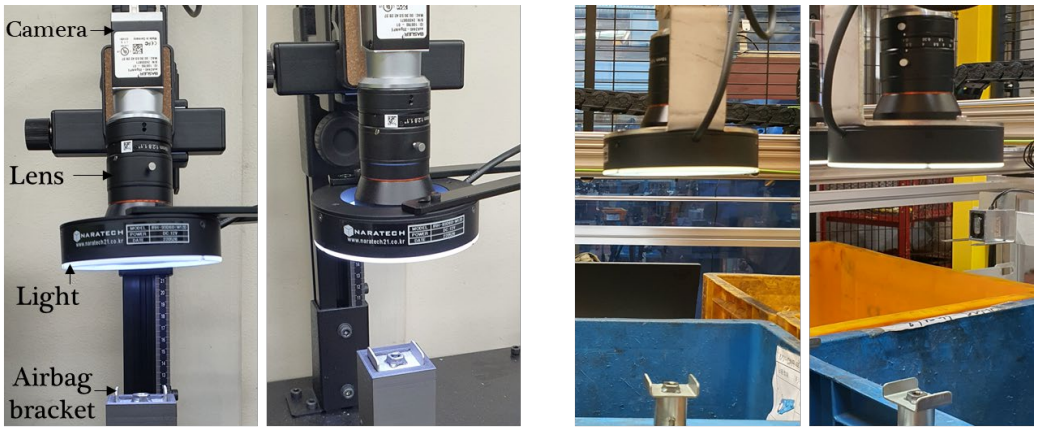


(a) (b) (c) (d)

Figure 2. Examples of welding defects in airbag bracket inspection. (**a**) Nut omission. (**b**–**d**) Surface abnormalities.

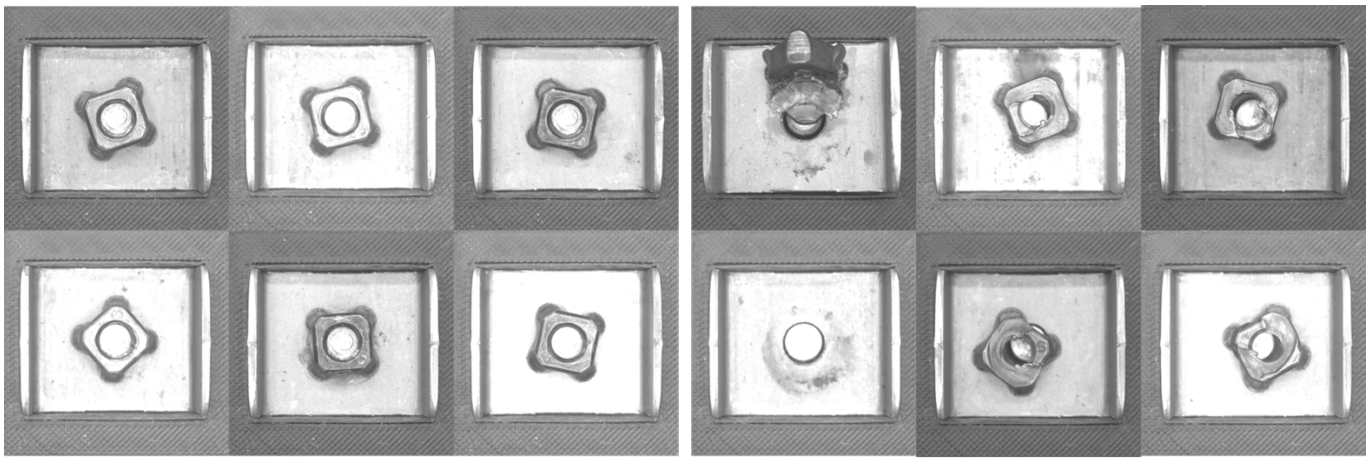
2.2.1. Data Collection

The image datasets for training and performance evaluation were collected in two different environments, i.e., a laboratory and a production site. The vision system, including the camera, lens, lighting, and kinematic configuration, was set identically in both environments, as shown in Figure 3a,b. An area scan monocamera (acA2440-20gm, Basler, Ahrensburg, Germany) with a resolution of 2448 × 2048 (5 MP) and a 16 mm lens (MVL-KF1628M, HIKROBOT, Zhejiang, China) was used as the vision system. The working distance between the lens and the airbag bracket was set to 10.0 cm. A total of 870 images of airbag bracket samples, including 696 positive and 174 negative images, were collected in the laboratory setup, and 136 images, including 122 positive and 14 negative images, were captured in the production site setup. Subsequently, 80% of the images collected in the laboratory were used to train the two-stream network model, and the remaining 20% were used for model verification. The images collected on the production site were used only for model verification. Figures 4 and 5 show the airbag bracket image samples collected in the laboratory and on the production site, respectively.



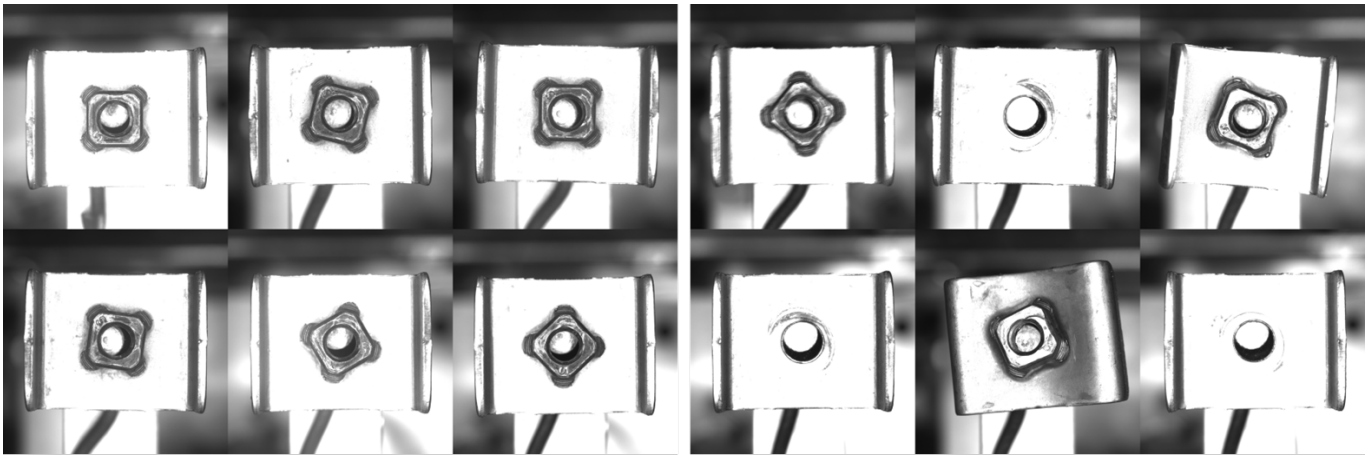
(a) (b)

Figure 3. Vision system setup for collecting image samples for the model's verification. (**a**) Vision system setup in the laboratory and (**b**) vision system setup on the production site.



(a) (b)

Figure 4. Positive and negative image samples collected in the laboratory. (**a**) Positive and (**b**) negative image samples.



(a) (b)

Figure 5. Positive and negative image samples collected at the production site. (**a**) Positive and (**b**) negative image samples.

2.2.2. Training

The region of interest (ROI) for the airbag bracket's inspection can be defined as the rectangle centered at the bracket's center that tightly encloses the nut region. The ROI was cropped in raw image samples and resized to 750 × 750 for model training and verification. To enlarge the training dataset, several variations were applied to the raw images: The center of the cropped region was randomly set within 100 pixels at the center of the bracket to reflect possible variations in the bracket position, and each image was rotated by 90°, 180°, and 270° and flipped. A total of 3470 image samples were used for training. Figure 6 shows the dataset enlargement procedure applied for model training. The Adam optimizer and Huber loss function were used for training, and the maximum number of epochs was set to 100.

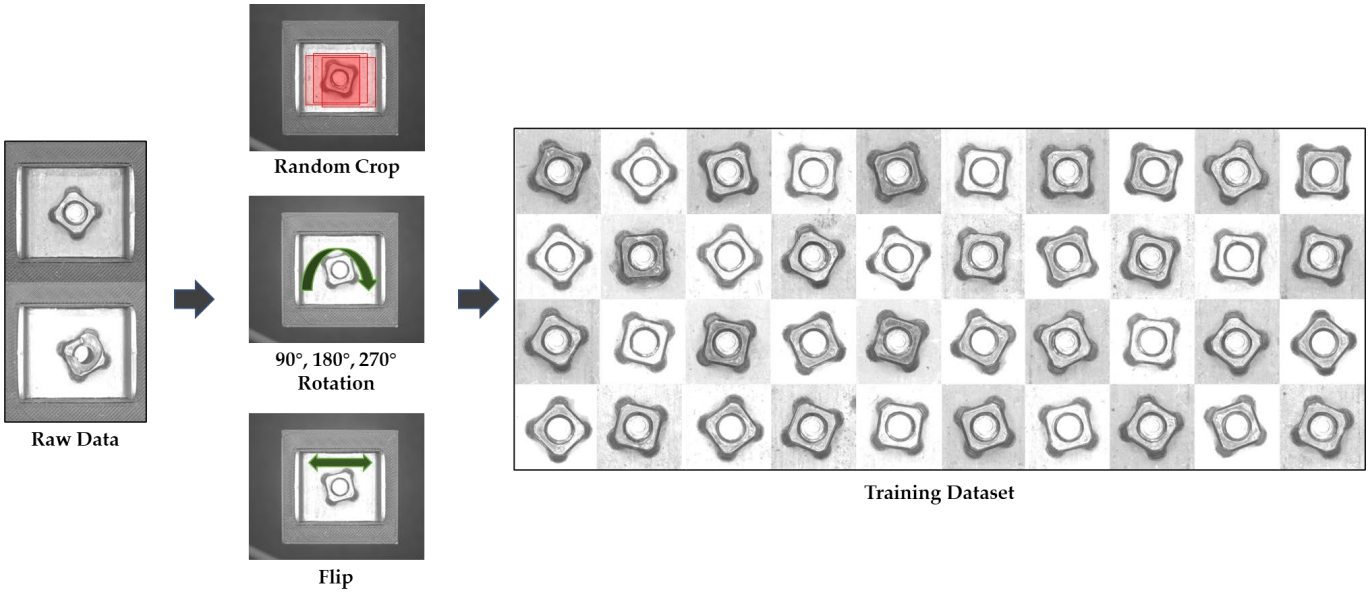


Figure 6. Dataset enlargement for model training by random cropping, rotating, and flipping raw images.

2.2.3. Evaluation Metrics

The performance of the proposed two-stream network model was evaluated by four metrics: accuracy, precision, recall, and F1 score. These evaluation metrics can be determined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \ \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \ \text{F1score} = 2 \cdot \frac{\text{Precision} \cdot \text{recall}}{\text{Precision} + \text{recall}} \quad (4)$$

where TP, TN, FP, and FN represent the true positive, true negative, false positive, and false negative, respectively.

3. Results

The performance of the proposed two-stream network model was evaluated from three perspectives: the effect of the classification layer, the effect of the two-stream network architecture, and performance in comparison with those of previous methods. In the performance evaluation, the two-stream model was trained only with the datasets gathered in the laboratory, and it was tested using two datasets gathered in the laboratory and on the production site.

3.1. Performance Evaluation in Terms of the Classification Layer

The proposed two-stream network OCC model was implemented with three types of classification layers: 1D convolution, fully connected, and SVM layers. Figure 7 and Table 4 present the experimental results of the two-stream network model according to the selected classification layer, as evaluated using laboratory datasets. In total, 140 positive and 34 negative images collected in the laboratory were used in this experiment. The confusion matrices in Figure 6 demonstrate that the SVM and 1D convolution layers achieved the best performance in classifying the TP (136/140) and TN (23/34) labels, respectively. The 1D convolution layer showed the best accuracy, precision, and F1 score of 0.8966, 0.9236, and 0.9366, respectively, whereas the SVM layer yielded the highest recall of 0.9714, as shown in Table 4.

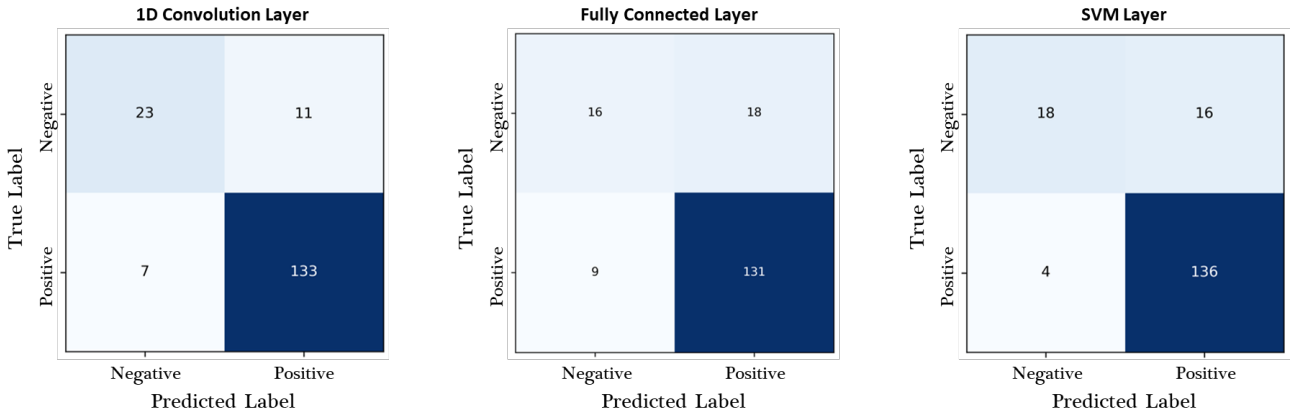


Figure 7. Confusion matrices of the two-stream network with three types of classification layers, as evaluated by using the laboratory dataset. (**a**) One-dimensional convolution layer, (**b**) fully connected layer, and (**c**) SVM layer.

Table 4. Results of performance evaluation according to the classification layer.

	Model	Accuracy	Precision	Recall	F1 Score
Laboratory dataset	1D conv.	0.8966	0.9236	0.9500	0.9366
	Fully conn.	0.8448	0.8792	0.9357	0.9066
	SVM	0.8851	0.8947	0.9714	0.9315
Production site dataset	1D conv.	0.9706	1.0000	0.9672	0.9833
	Fully conn.	0.9632	0.9835	0.9754	0.9794
	SVM	0.8015	0.8926	0.8852	0.8889

Figure 8 shows the experimental results, as evaluated by using the production site dataset. In total, 122 positive and 14 negative images collected on the production site were used in this experiment. The confusion matrices in Figure 8 demonstrate that the fully connected and 1D convolution layers achieved the best performance in classifying the TP (119/122) and TN (14/14) labels, respectively. The 1D convolution layer showed the best accuracy, precision, and F1 score of 0.9706, 1.0000, and 0.9833, respectively, whereas the fully connected layer achieved the highest recall of 0.9754, as shown in Table 4.

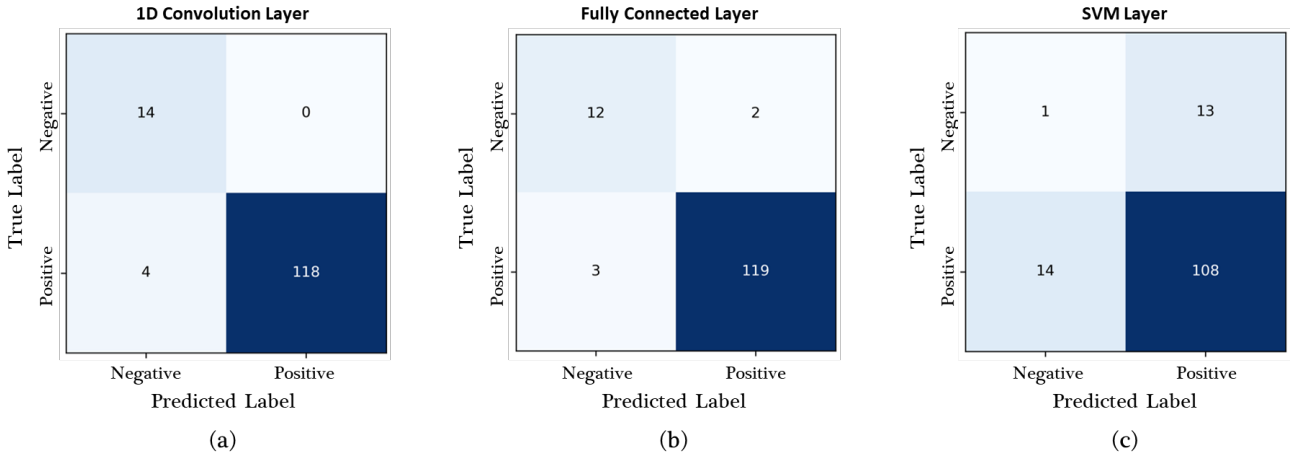


Figure 8. Confusion matrices of the two-stream network with three types of classification layers, as evaluated by using the production site dataset. (**a**) One-dimensional convolution layer, (**b**) fully connected layer, and (**c**) SVM layer.

3.2. Performance Evaluation of the Two-Stream Network Model

The performance of the two-stream network model was compared with those of models without one of the global and local feature extractor networks in this experiment. The 1D convolution layer was used for the classification layer in this experiment. Table 5 presents a comparison of the performance of the two-stream network model with those of the single-stream network. The performance evaluation was conducted for both the laboratory and production site datasets. The global feature extractor network model exhibited the lowest performance for both datasets, with an accuracy of 0.8621, a precision of 0.8580, and an F1 score of 0.9205 for the laboratory dataset; and an accuracy of 0.8971, a precision of 0.9030, and an F1 score of 0.9453 for the production site dataset. The 2S-1DOC model exhibited the highest performance for both datasets, with an accuracy, precision, and F1 score of 0.8966, 0.9236, and 0.9366, respectively, for the laboratory dataset; and an accuracy, precision, and F1 score of 0.9706, 1.0000, and 0.9833, respectively, for the production site dataset. Figure 9 presents the t-distributed stochastic neighbor embedding (t-SNE) plots of the feature vectors output from the local, global, and two-stream network. The t-SNE plot visualizes the similarity between feature vectors by mapping high-dimensional feature vectors to a lower-dimensional space (2D). The feature vectors of the two-stream network, which combines the characteristics of the local and global feature extractor networks, clearly distinguish the true and false samples with a single decision boundary.

Table 5. Comparison of the performances of the two-stream and single-stream network models.

	Model	Accuracy	Precision	Recall	F1 Score
Laboratory dataset	Local	0.8736	0.8933	0.9571	0.9241
	Global	0.8621	0.8580	0.9929	0.9205
	Two-stream	0.8966	0.9236	0.9500	0.9366
Production sitedataset	Local	0.9310	0.9923	0.9214	0.9556
	Global	0.8971	0.9030	0.9918	0.9453
	Two-stream	0.9706	1.0000	0.9672	0.9833

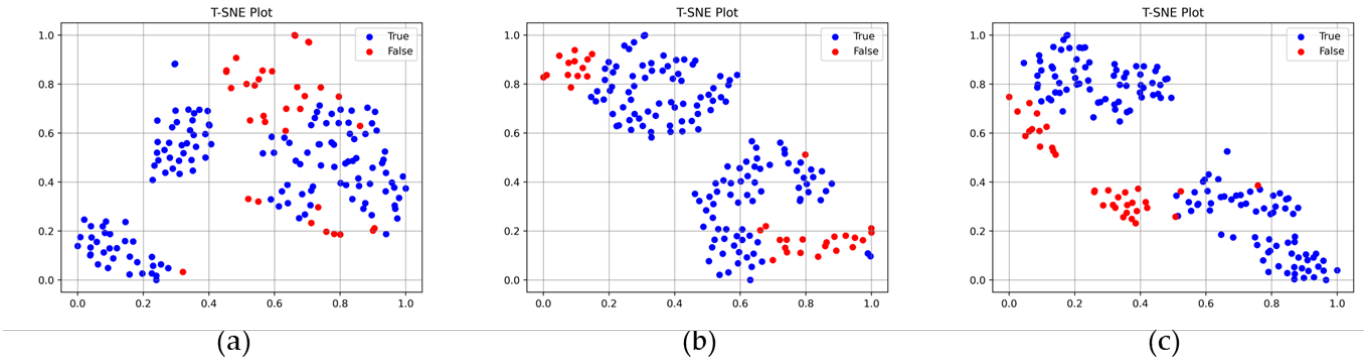


Figure 9. t-SNE plots of various classification models. Blue and red dots represent feature vectors of true and false samples, respectively. (**a**) Global feature extractor network, (**b**) local feature extractor network, and (**c**) two-stream network.

3.3. Performance Comparison with Previous Models

Table 6 compares the performance of the two-stream network model and previous image classification models. Six representative classification models, InceptionV3 [41], ResNet101V2 [14], Xception [42], MobileNetV2 [15], VGG-16 [13], and PaDiM [43], were tested for the performance comparison. The two-stream network model presented the highest accuracy and precision of 0.8966 and 0.9236, respectively. However, ResNet101V2, Xception, MobileNetV2, and VGG-16 yielded the highest recall result of 1.000, and PaDiM shows the highest F1 score result of 0.9388. The InceptionV3 model presented the lowest

accuracy, precision, and F1 scores of 0.8621, 0.8580, and 0.9205, respectively. Figure 10 presents the t-SNE plots of the feature vectors of the two-stream network model and previous models. As shown in the figure, the proposed two-stream network most clearly distinguished the true and false samples compared to previous models.

Table 6. Performance comparison with previous models using the laboratory dataset.

Model	Accuracy	Precision	Recall	F1 Score
InceptionV3	0.8621	0.8580	0.9929	0.9205
ResNet101V2	0.8736	0.8642	1.0000	0.9272
Xception	0.8678	0.8589	1.0000	0.9241
MobileNetV2	0.8736	0.8642	1.0000	0.9272
VGG-16	0.8678	0.8589	1.0000	0.9241
PaDiM	0.8966	0.8961	0.9857	0.9388
Proposed model	0.8966	0.9236	0.9500	0.9366

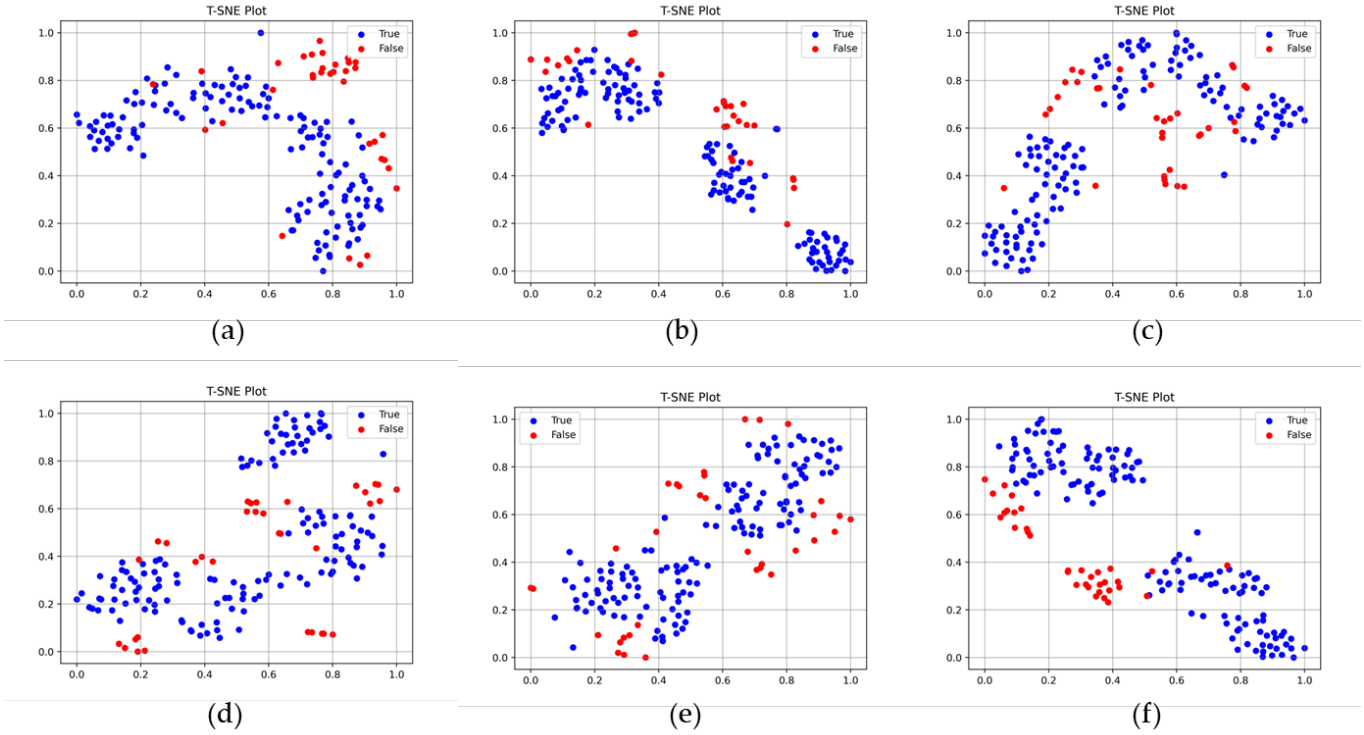


Figure 10. Comparison of the t-SNE plots of the previous and proposed models. (**a**) InceptionV3, (**b**) ResNet101V2, (**c**) Xception, (**d**) MobileNetV2, (**e**) VGG-16, and (**f**) proposed model.

Table 7 presents a comparison of the results obtained using the proposed and previous models and the production site dataset. The two-stream network model presents the highest accuracy, precision, and F1 scores of 0.9706, 1.0000, and 0.9833, respectively.

Table 7. Performance comparison with previous models using the production site dataset.

Model	Accuracy	Precision	Recall	F1 Score
InceptionV3	0.8971	0.9030	0.9918	0.9453
ResNet101V2	0.9118	0.9104	1.0000	0.9531
Xception	0.9191	0.9173	1.0000	0.9569
MobileNetV2	0.9044	0.9037	1.0000	0.9494
VGG16	0.8971	0.9030	0.9918	0.9453
PaDiM	0.5882	1.0000	0.5410	0.7021
Proposed model	0.9706	1.0000	0.9672	0.9833

4. Discussion

In the manufacturing sector, defect inspection using AI technology has been extensively studied to optimize labor costs and process automation. However, due to the difficulty of collecting data in the field and data imbalances, OCC has recently attracted attention for various applications. OCC is efficient in applications where data are unbalanced, but it has a critical limitation in that the features are compressed in the training data, resulting in false-positive errors. To overcome this limitation, we developed a two-stream network OCC model consisting of local and global feature extractor networks followed by a classification layer. The performance of the proposed model was validated using a practical example of automotive-airbag bracket-welding defect inspection. The image datasets of the airbag bracket collected in two different environments, i.e., a laboratory and a production site, were used for the training and validation of the proposed model. For the dataset collected in the laboratory, our model achieved results of 0.8966, 0.9236, 0.9500, and 0.9366 for the accuracy, precision, recall, and F1 score, respectively. For the production site dataset, the model achieved results of 0.9706, 1.0000, 0.9672, and 0.9833 for the accuracy, precision, recall, and F1 score, respectively.

The inspection performance of the entire model could be affected by not only the performance of the feature extraction layer but also that of the classification layer. Three types of classification layers, 1D convolution, fully connected, and SVM layers, were tested to investigate the effect of the classification layer and to identify the optimal classification presenting the best inspection performance. The 1D convolution layer showed the best accuracy, precision, and F1 score for both laboratory and production site datasets. The fully connected layer yielded slightly better performances than the 1D convolution layer only in terms of recall. In the performance comparison between the laboratory and production site datasets, the SVM layer exhibited a decrease in the accuracy, precision, recall, and F1 score by 9.44%, 0.24%, 8.87%, and 4.58%, respectively, for the production site dataset compared with the laboratory dataset. By contrast, the 1D convolution layer showed an increase of 8.37% in accuracy, 8.54% in precision, 1.77% in recall, and 5.01% in the F1 score for the production site dataset compared to the laboratory dataset. These results indicate that the classification by the 1D convolution layer is more appropriate for alleviating the representation's collapse than that by other layers.

Compared with the single-stream network model, the two-stream network model showed an increase of up to 7.35% in accuracy, 9.70% in precision, and 3.80% in F1 score, proving that the two-stream model achieved a better performance than the existing single-stream model. In addition, the proposed two-stream model exhibited performance improvements in the production site dataset's results compared with the laboratory dataset results, with an increase in accuracy of 8.25%, precision of 8.27%, recall of 1.81%, and F1 score of 4.99%, demonstrating that the proposed model maintains the inspection performance for the datasets gathered under different environmental conditions than the training datasets. This finding proves that the two-stream network architecture contributes to reducing the performance degradation caused by representation collapse.

The effect of the two-stream network on performance improvement is clearly presented by the t-SNE plots shown in Figure 9. In Figure 9a, the feature vectors produced by the global feature extractor network provide a rough classification of the true and false samples,

and there is some overlap observed among certain portions of the samples. The lack of a distinct decision boundary can be attributed to the global feature extractor network's emphasis on capturing general features. In contrast, the feature vector generated by the local feature extractor network depicted in Figure 9b exhibits clear differentiation between true and false samples. Nevertheless, determining a single decision boundary is challenging as false samples are divided into two separate clusters. By combining the characteristics of the global and local feature extractor networks, the feature vector generated by the two-stream network depicted in Figure 9c effectively discriminates between true and false samples using a single decision boundary.

The comparison between the proposed two-stream network model and the previous model confirmed its enhanced classification performance. In the performance comparison with the previous model, the proposed two-stream model showed the best performance for most performance indices, including the accuracy, precision, and F1 score for production site datasets. The improvements in accuracy, precision, and F1 score were up to 65.01%, 10.74%, and 40.05%, respectively. The PaDiM method demonstrated proficient classification performance within the laboratory dataset. However, its performance significantly deteriorated when applied to the production site's dataset, which has distinct environmental conditions compared to the training dataset. To understand the rationale behind the performance improvement in the proposed two-stream network model, we examined the t-SNE plots presented in Figure 10. The feature vectors of the previous model did not exhibit clear classification boundaries for true and false samples. In contrast, the feature vectors generated by the proposed model provided the most distinct differentiation between true and false samples. The significance of this enhancement in classification features lies in its ability to alleviate the inherent bias toward true samples, which frequently possess larger datasets in comparison to false samples. The biased predictions of previous models toward true samples had a detrimental impact on precision performance, resulting in its degradation.

The two-stream network OCC model proposed in this study exhibited high classification performance with respect to both the laboratory and production site datasets. However, the validation was not sufficient for verifying the classification performance of negative samples because not enough defective samples were collected at the production site. In future studies, sufficient negative samples must be collected, and the performance of the proposed model should be further validated with those samples.

5. Conclusions

In this paper, we proposed a two-stream network OCC model to resolve the representation collapse problem of OCC models. The performance of the proposed model was validated in terms of the classification layer and network architecture, and comparisons were carried out using previous methods that implement image samples collected in the practical example of airbag bracket inspection. The performance results clearly indicated that the proposed model effectively addressed the representation collapse problem, resulting in enhanced inspection accuracy in comparison to existing classification models. Moreover, the classification performance of the proposed two-stream model exhibited an impressive improvement of up to 10% compared to previous classification models. This performance improvement can be accomplished using the novel two-stream network, which seamlessly integrates both general and data-specific features. The practical applications of defect inspection can greatly benefit from the implementation of the two-stream network model presented in this paper. Its incorporation is poised to make valuable contributions toward enhancing performances in vision inspection tasks.

Author Contributions: Conceptualization, H.J.; formal analysis, S.L. (Seunghun Lee) and C.L.; funding acquisition, H.J.; investigation, S.L. (Sungkwan Lee); methodology, S.L. (Seunghun Lee) and C.L.; project administration, H.J.; software, S.L. (Sungkwan Lee); validation, S.L. (Seunghun Lee) and C.L.; writing—original draft, S.L. (Seunghun Lee) and C.L.; writing—review and editing, H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by Konkuk University in 2018.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data cannot be provided due to the security reason.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gao, Y.; Li, X.; Wang, X.V.; Wang, L.; Gao, L. A Review on Recent Advances in Vision-based Defect Recognition towards Industrial Intelligence. *J. Manuf. Syst.* **2022**, *62*, 753–766. [CrossRef]
2. Tang, Z.; Tian, E.; Wang, Y.; Wang, L.; Yang, T. Nondestructive Defect Detection in Castings by Using Spatial Attention Bilinear Convolutional Neural Network. *IEEE Trans. Ind. Inform.* **2020**, *17*, 82–89. [CrossRef]
3. Muresan, M.P.; Cireap, D.G.; Giosan, I. Automatic vision inspection solution for the manufacturing process of automotive components through plastic injection molding. In Proceedings of the 16th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 3–5 September 2020.
4. Xu, Y.; Zhang, K.; Wang, L. Metal Surface Defect Detection Using Modified YOLO. *Algorithms* **2021**, *14*, 257. [CrossRef]
5. Zhao, Z.; Yang, X.; Zhou, Y.; Sun, Q.; Ge, Z.; Liu, D. Real-time detection of particleboard surface defects based on improved YOLOV5 target detection. *Sci. Rep.* **2021**, *11*, 21777. [CrossRef]
6. Aber Ronaghi, A.; Ren, J.; El-Gindy, M. Defect Detection Methods for Industrial Products Using Deep Learning Techniques: A Review. *Algorithms* **2023**, *16*, 95. [CrossRef]
7. Zheng, X.; Wang, H.; Chen, J.; Kong, Y.; Zheng, S. A Generic Semi-Supervised Deep Learning-Based Approach for Automated Surface Inspection. *IEEE Access* **2020**, *8*, 114088–114099. [CrossRef]
8. Zhu, Z.; Han, G.; Jia, G.; Shu, L. Modified DenseNet for Automatic Fabric Defect Detection With Edge Computing for Minimizing Latency. *IEEE Internet Things J.* **2020**, *7*, 9623–9636. [CrossRef]
9. Shao, L.; Zhang, E.; Ma, Q.; Li, M. Pixel-Wise Semisupervised Fabric Defect Detection Method Combined With Multitask Mean Teacher. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–11. [CrossRef]
10. Pourkaramdel, Z.; Fekri-Ershad, S.; Nanni, L. Fabric defect detection based on completed local quartet patterns and majority decision algorithm. *Expert Syst. Appl.* **2022**, *198*, 116827. [CrossRef]
11. Jeyaraj, P.R.; Nadar, E.R.S. Effective textile quality processing and an accurate inspection system using the advanced deep learning technique. *Text. Res. J.* **2020**, *90*, 971–980. [CrossRef]
12. Huang, Y.C.; Hung, K.C.; Lin, J.C. Automated Machine Learning System for Defect Detection on Cylindrical Metal Surfaces. *Sensors* **2022**, *22*, 9783. [CrossRef] [PubMed]
13. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
14. Qiu, Z.; Yao, T.; Mei, T. Learning spatio-temporal representation with pseudo-3d residual networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5533–5541.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 630–645.
16. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
17. Wei, W.; Deng, D.; Zeng, L.; Zhang, C. Real-time implementation of fabric defect detection based on variational automatic encoder with structure similarity. *J. Real-Time Image Process.* **2021**, *18*, 807–823. [CrossRef]
18. Yang, D.; Cui, Y.; Yu, Z.; Yuan, H. Deep Learning Based Steel Pipe Weld Defect Detection. *Appl. Artif. Intell.* **2021**, *35*, 1237–1249. [CrossRef]
19. Kim, J.; Ko, J.; Choi, H.; Kim, H. Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors* **2021**, *21*, 4968. [CrossRef]
20. Tang, T.W.; Hsu, H.; Huang, W.R.; Li, K.M. Industrial Anomaly Detection with Skip Autoencoder and Deep Feature Extractor. *Sensors* **2022**, *22*, 9327. [CrossRef]
21. Upadhyay, A.; Li, J.; King, S.; Addepalli, S. A Deep-Learning-Based Approach for Aircraft Engine Defect Detection. *Machines* **2023**, *11*, 192. [CrossRef]
22. Yun, J.P.; Shin, W.C.; Koo, G.; Kim, M.S.; Lee, C.; Lee, S.J. Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *J. Manuf. Syst.* **2020**, *55*, 317–324. [CrossRef]
23. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **2017**, *106*, 249–259. [CrossRef]
24. Bendre, N.; Marín, H.T.; Najafirad, P. Learning from Few Samples: A Survey. *arXiv* **2020**, arXiv:2007.15484.
25. Hasib, K.M.; Iqbal, M.S.; Shah, F.M.; Al Mahmud, J.; Popel, M.H.; Showrov, M.I.H.; Ahmed, S.; Rahman, O. A Survey of Methods for Managing the Classification and Solution of Data Imbalance Problem. *J. Comput. Sci.* **2020**, *16*, 1546–1557. [CrossRef]

26. Tsai, D.M.; Jen, P.H. Autoencoder-based anomaly detection for surface defect inspection. *Adv. Eng. Inform.* **2021**, *48*, 101272. [CrossRef]
27. Papadopoulos, A.A.; Rajati, M.R.; Shaikh, N.; Wang, J. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing* **2021**, *441*, 138–150. [CrossRef]
28. Wang, S.; Liu, W.; Wu, J.; Cao, L.; Meng, Q.; Kennedy, P.J. Training deep neural networks on imbalanced data sets. In Proceedings of the 2016 International Joint Conference on Neural Networks, Vancouver, BC, Canada, 24–29 July 2016; pp. 4368–4374.
29. Mao, W.L.; Chiu, Y.Y.; Lin, B.H.; Wang, C.C.; Wu, Y.T.; You, C.Y.; Chien, Y.R. Integration of Deep Learning Network and Robot Arm System for Rim Defect Inspection Application. *Sensors* **2022**, *22*, 3927. [CrossRef] [PubMed]
30. Tax, D.M.J.; Duin, R.P.W. Support Vector Data Description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
31. Shin, H.J.; Eom, D.H.; Kim, S.S. One-class support vector machines—An application in machine fault detection and classification. *Comput. Ind. Eng.* **2005**, *48*, 395–408. [CrossRef]
32. Mahadevan, S.; Shah, S.L. Fault detection and diagnosis in process data using one-class support vector machines. *J. Process Control.* **2009**, *19*, 1627–1639. [CrossRef]
33. Liu, B.; Xiao, Y.; Yu, P.S.; Cao, L.; Zhang, Y.; Hao, Z. Uncertain one-class learning and concept summarization learning on uncertain data streams. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 468–484. [CrossRef]
34. Ruff, L.; Görnitz, N.; Deecke, L.; Siddiqui, S.; Vandermeulen, R.A.; Binder, A.; Müller, E.; Kloft, M. Deep One-Class Classification; In Proceedings of International Conference on Machine Learning (ICML), Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018.
35. Perera, P.; Patel, V.M. Learning Deep Features for One-Class Classification. *IEEE Trans. Image Process.* **2019**, *28*, 5450–5463. [CrossRef]
36. Lee, J.; Lee, Y.C.; Kim, J.T. Fault detection based on one-class deep learning for manufacturing applications limited an imbalanced database. *J. Manuf. Syst.* **2020**, *57*, 357–366. [CrossRef]
37. Goyal, S.; Raghunathan, A.; Jain, M.; Simhadri, H.V.; Jain, P. DROCC: Deep robust one-class classification. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020.
38. Hayashi, T.; Fujita, H.; Hernandez-Matamoros, A. Less complexity one-class classification approach using construction error of convolutional image transformation network. *Inf. Sci.* **2021**, *560*, 217–234. [CrossRef]
39. Hayashi, T.; Cimr, D.; Studnička, F.; Fujita, H.; Bušovský, D.; Cimler, R. OCSTN: One-class time-series classification approach using a signal transformation network into a goal signal. *Inf. Sci.* **2022**, *614*, 71–86. [CrossRef]
40. Hayashi, T.; Fujita, H. One-class ensemble classifier for data imbalance problems. *Appl. Intell.* **2022**, *52*, 17073–17089. [CrossRef]
41. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2818–2826.
42. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
43. Defard, T.; Setkov, A.; Loesch, A.; Audigier, R. PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. *arXiv* **2020**, arXiv:2011.08785.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.