applied sciences

MDPI

Article

Data Augmentation Methods Applying Grayscale Images for Convolutional Neural Networks in Machine Vision

Jinyeong Wang and Sanghwan Lee *

Department of Mechanical Convergence Engineering, Hanyang University, Seoul 04763, Korea; ytrqwe12@hanyang.ac.kr
* Correspondence: shlee@hanyang.ac.kr

Abstract: In increasing manufacturing productivity with automated surface inspection in smart factories, the demand for machine vision is rising. Recently, convolutional neural networks (CNNs) have demonstrated outstanding performance and solved many problems in the field of computer vision. With that, many machine vision systems adopt CNNs to surface defect inspection. In this study, we developed an effective data augmentation method for grayscale images in CNN-based machine vision with mono cameras. Our method can apply to grayscale industrial images, and we demonstrated outstanding performance in the image classification and the object detection tasks. The main contributions of this study are as follows: (1) We propose a data augmentation method that can be performed when training CNNs with industrial images taken with mono cameras. (2) We demonstrate that image classification or object detection performance is better when training with the industrial image data augmented by the proposed method. Through the proposed method, many machine-vision-related problems using mono cameras can be effectively solved by using CNNs.

Keywords: machine vision; data augmentation; deep learning; convolutional neural networks; transfer learning

check for updates

Citation: Wang, J.; Lee, S. Data Augmentation Methods Applying Grayscale Images for Convolutional Neural Networks in Machine Vision. *Appl. Sci.* **2021**, *11*, 6721. https://doi.org/10.3390/app11156721

Academic Editor: Rocco Furferi

Received: 31 May 2021
Accepted: 21 July 2021
Published: 22 July 2021

1. Introduction

With the increasing demand for machine vision to automate the surface inspection of factories, the requirement for higher inspection speed and accuracy has also increased. Machine vision refers to any software or hardware that utilizes visual information of the inspection target to perform the inspection. Conventional machine vision [1–3] is capable of inspecting formalized defects through rule-based inspections. However, detecting non-formalized defects is challenging to conventional machine vision applications.

In 2012, AlexNet [4] won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) by using CNNs. Since then, CNNs have been applied in various fields that use image data. Machine vision researchers are also conducting studies to detect defects by applying CNNs [5–21]. Generally, when applied to machine vision, CNNs perform one of the following three tasks: (1) classification into normal or defective at a specific part, (2) detection of standard and defective parts, and (3) segmentation of the defective area.

Most of the machine vision applications use mono cameras because they utilize structural features of the inspection target. As shown in Figure 1, the color camera typically obtains a three-channel RGB image via interpolation after shooting with the Bayer pattern. However, the mono camera does not require interpolation after acquiring. As a result, mono cameras with the same pixel have a better resolution than color cameras. Because most machine visions do not require color information for defect inspection, they leverage mono cameras to obtain grayscale images.

Regarding the application of CNN-based machine vision with mono cameras, Yang et al. [16] confirmed that transfer learning from the network that won the ILSVRC competition performs higher classification accuracy than one trained from scratch. In image classification with grayscale images, Xie and Richmond [22] showed that transfer learning from the network pretrained with grayscale ILSVRC data shows better classification accuracy than transfer learning from the network pretrained with original ILSVRC data.

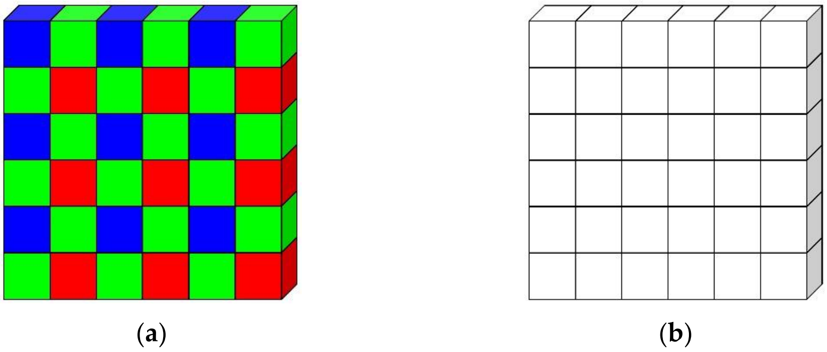


Figure 1. Structure of the camera sensor: (**a**) Bayer pattern of the color camera and (**b**) mono camera.

Burduja et al. [23] performed intracranial hemorrhage detection by using color images merged from three grayscale images that extracted different features from one CT image.

To solve the machine vision issues due to scarce data on defective products, Yun et al. [17] performed a data augmentation through a conditional convolutional variable autoencoder (CCVAE) for defect classification. However, if surface defect inspection is performed with object detection, the application of CCVAE-based data augmentation is limited.

In general computer vision, CNNs are trained by using large amounts of data, such as a million images for a thousand classes provided by ILSVRC or 110,000 images for 80 classes provided by COCO [24]. However, collecting that amount of balanced dataset for training each application in machine vision is less productive. Eventually, most of the CNN applications are trained by the imbalanced small amount of data. Therefore, reliable methods to train the surface inspection networks with these small datasets must be devised.

In this study, we devised a data augmentation method that can be easily applied when preparing CNN-based machine vision systems, using mono cameras. Our proposed method does not leverage neural networks, so that it can perform data augmentation quickly. We also demonstrate that it can be applicable for imbalanced datasets. Experiments show that our proposed method is effective for both image classification and object detection processes. The data augmentation method developed in this study is based on the following methods: (1) imitating the various changes that can occur while acquiring images from mono cameras in machine vision systems; (2) extracting structural features of the images, which are the primary purpose of using the mono cameras; and (3) merging them into color images.

2. Materials and Methods

2.1. Dataset

2.1.1. The NEU-DET Dataset

The NEU-DET dataset [3], which was collected by the Northeastern University, is a dataset for detecting six types of defects on metal surfaces. Object annotations for defect detection are provided, but we used them as a dataset for image classification in this paper. Each class has 240 images for training and 60 images for validation, and each image is 200 × 200 pixels. Figure 2 shows some samples of the NEU-DET dataset.

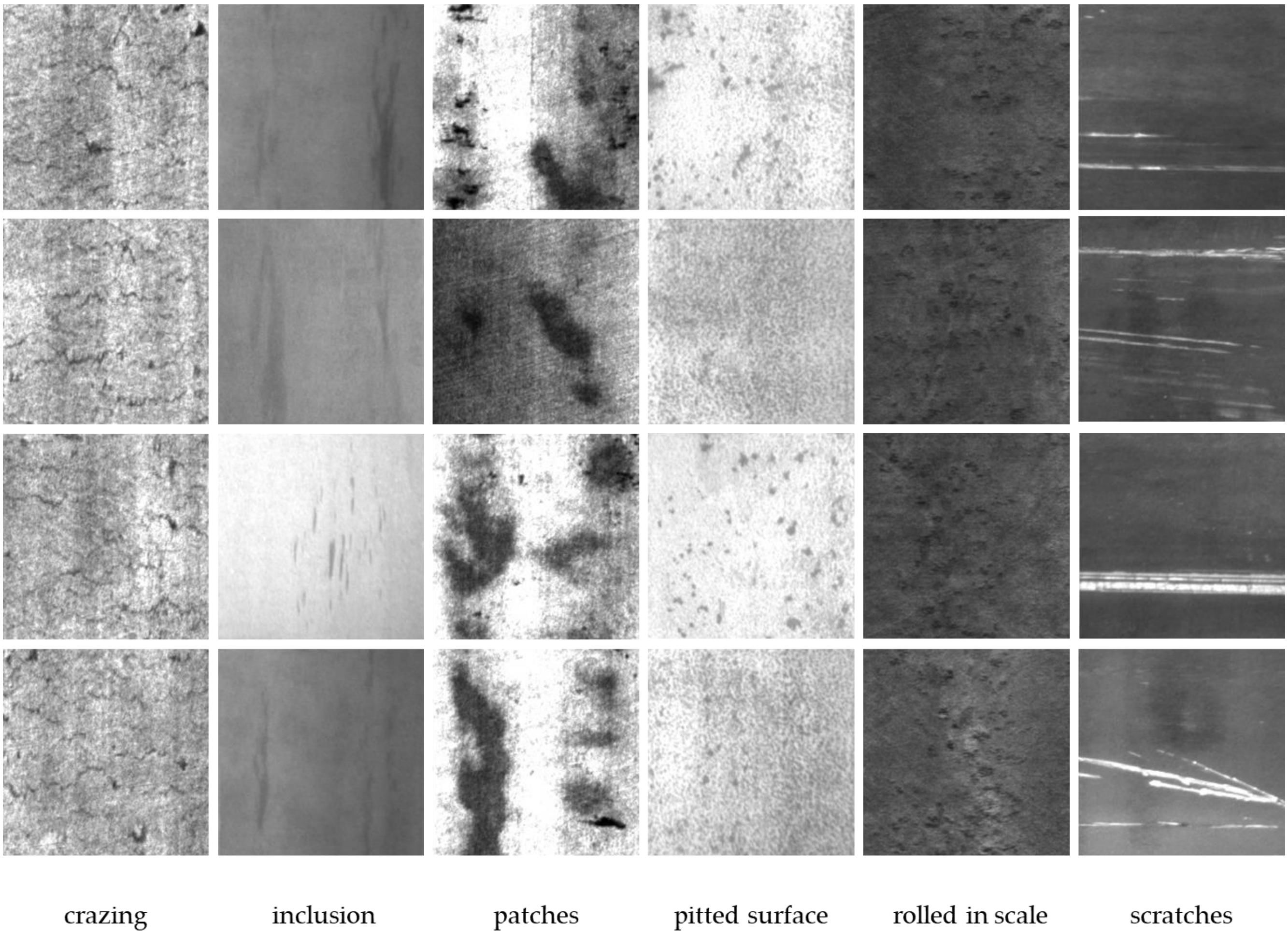


Figure 2. Surface defect samples in the NEU-DET dataset.

2.1.2. Brake Pad Dataset

The machine vision system structure is shown in Figure 3a, and the brake pad image for inspection is shown in Figure 3b. The brake pad image was obtained by using a 2.5-megapixel complementary metal–oxide–semiconductor (CMOS) sensor mono camera. The type of product that performs the inspection is shown in Figure 4.

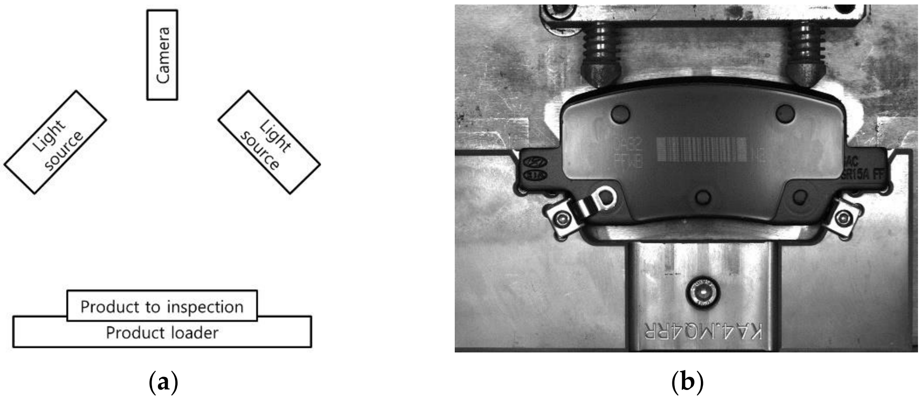
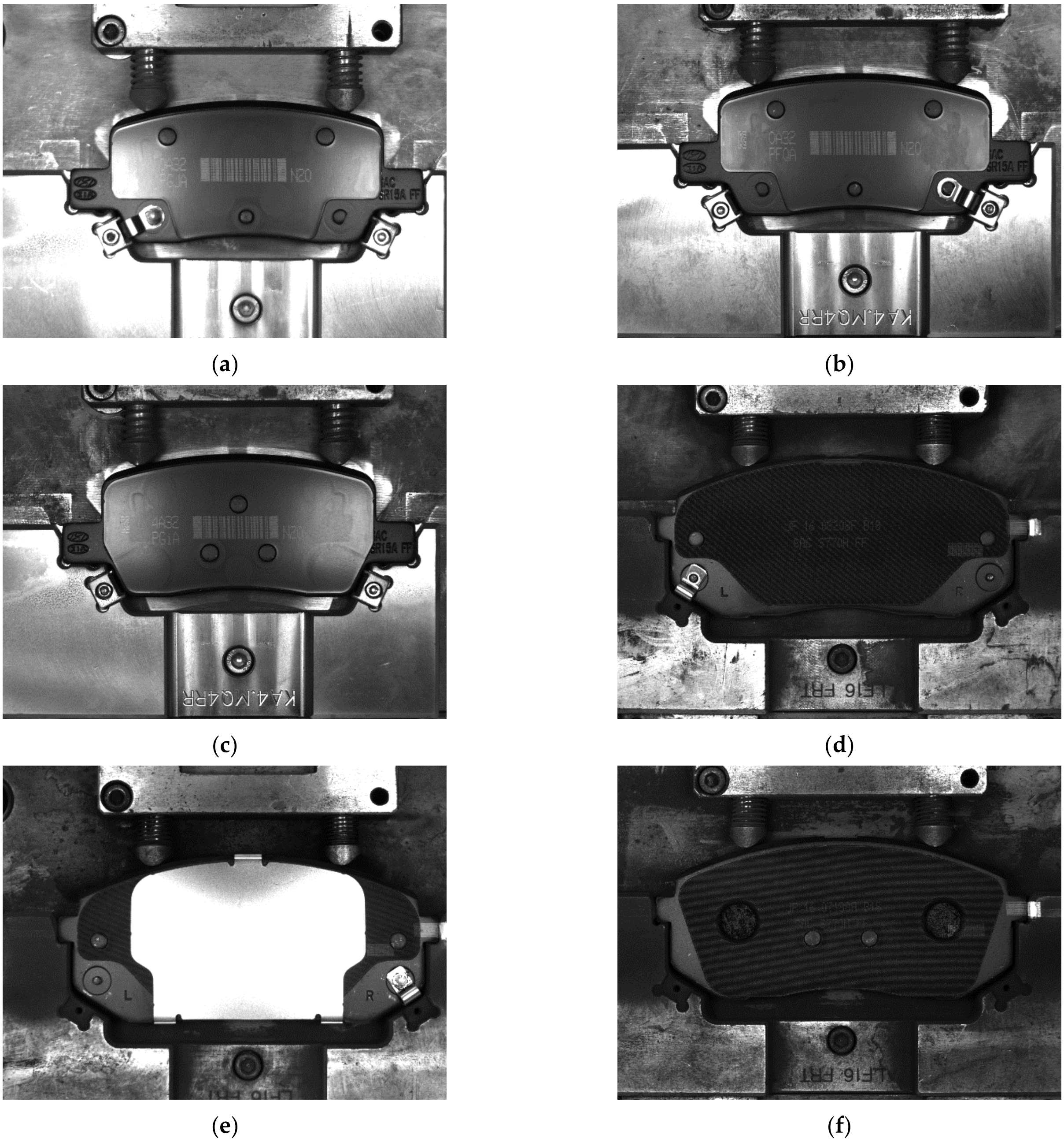


Figure 3. (**a**) System configuration and (**b**) captured image of the brake pad.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 4. Data types: (**a**) product type 1, riveted sensor in left; (**b**) product type 1, unriveted sensor in right; (**c**) product type 2; (**d**) product type 3, unriveted sensor in left; (**e**) product type 3, riveted sensor in right; and (**f**) product type 4.

The total number of original images was 545, of which 490 images were used for training and 55 for validation. Table 1 shows the number of each object to detect.

Table 1. The number of objects to detect.

	Protruding Part	Unriveted Sensor	Riveted Sensor
Training dataset	1236	320	112
Validation dataset	138	37	12

The primary defect types are shown in Figure 5. The following procedure can be applied to inspect them:

1. Inspecting the location of the protruding part of the product to inspect whether the product is loaded in the wrong location, as shown in Figure 5b.
2. Inspecting whether the metal sensor is located correctly in the specified location of the product, as shown in Figure 5a,c.
3. Inspecting whether the riveting is performed correctly to secure the sensor. Figure 5d shows an incorrectly riveted product.

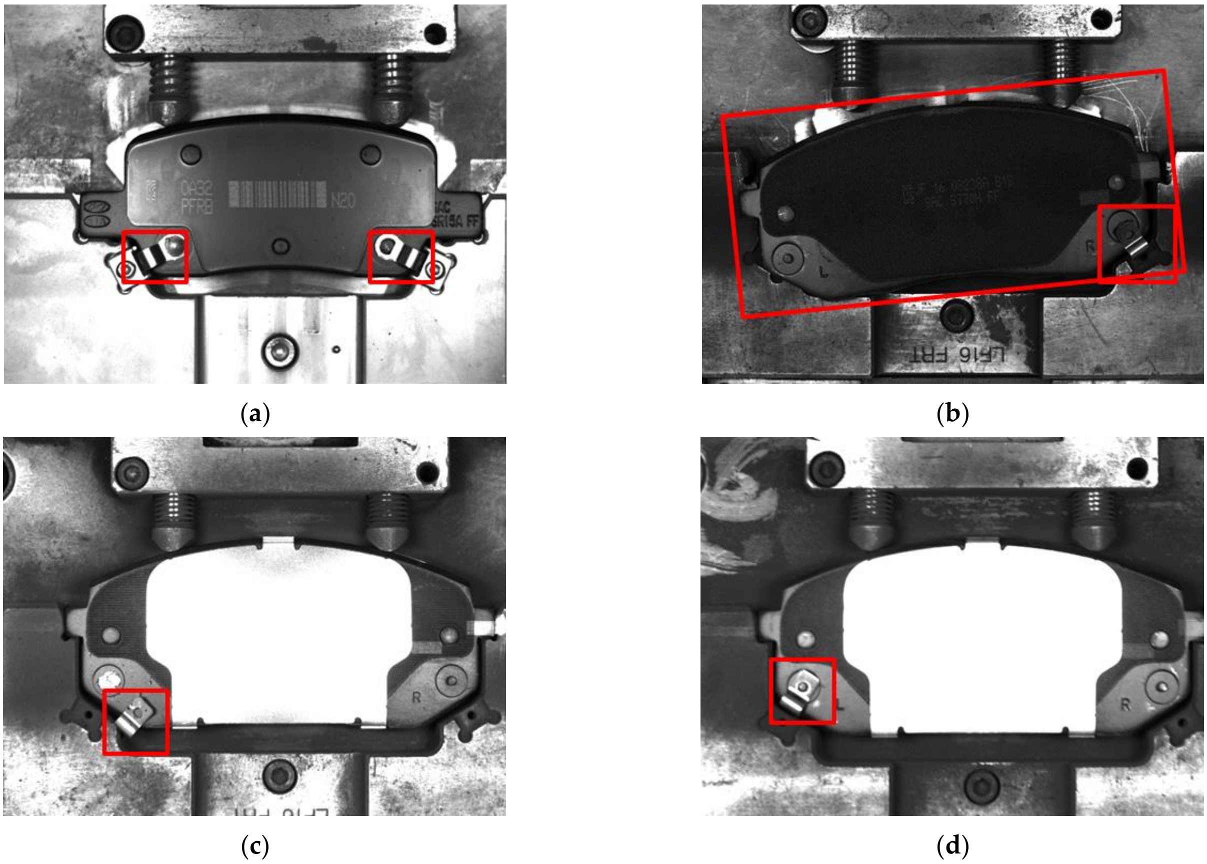

(**a**) (**b**) (**c**) (**d**)

Figure 5. Defective case: (**a**) ill-positioned sensor, (**b**) ill-positioned sensor and wrong placement of the product, (**c**) ill-positioned sensor and unsuitable rivet, and (**d**) riveting not performed.

Object detection via CNNs was performed to inspect these defects. The objects for detection are as follows: (1) protruding part, (2) unriveted sensor, and (3) riveted sensor.

2.2. Proposed Data Augmentation Method

The proposed data augmentation method was performed in two steps. First, we imitated the characteristics of the camera and extracted the structural features of the inspection target. In this step, all the images after augmentation were one-channel grayscale images. Then, we combined the corresponding images to generate several three-channel color images. Four types of data, including the original data, were prepared to validate the superiority of the proposed data augmentation method.

1. Original images (original).
2. Augmented one-channel grayscale images with original images (one-channel).
3. Grayscale images were converted after augmentation, using the proposed method with original images (three-channel, gray).
4. Color images augmented by using the proposed method with original images (three-channel, color).

Then neural networks were trained, using each dataset, and their performances were compared.

We used OpenCV in Python for data augmentation, and the implementation proposed in this paper is opened on a Github repository (github.com/jinfree/GrayscaleImageAugmentation) (accessed on 9 July 2021), under an AGPLv3 license.

2.2.1. One-Channel Augmentation

This section discusses the first of the two steps of data augmentation. We performed one-channel augmentation via four approaches: random pixel noise, bright adjustment, blur, and edge extraction. Edge extraction is conducted to extract structural information of the inspection target. The other approaches imitate image changes that can occur when acquiring images from the CMOS camera.

Pixel Noise

As shown in Figure 6, there are two types of image sensors used in machine vision, namely charge-coupled device (CCD) and CMOS. A CCD is a sensor that accumulates and transmits charges generated by using light energy and eventually converts them into electrical signals. CMOS sensors immediately amplify and transmit the charges generated by using light energy into electrical signals. CMOS sensors outperform CCD sensors regarding the number of frames per second, resolution, and power consumption. As a result, the CMOS sensor is used for high-resolution machine vision cameras; however, it has the disadvantage of pixel-level noise, as shown in Figure 7. We performed data augmentation by imitating such pixel noise; the pseudo-code is shown in Algorithm 1.

Algorithm 1. Pseudo-code of applying pixel noise to the given image.

```
Input: Original grayscale image
Output: Grayscale image with pixel noise.
for x in range of 0 to width of image
    for y in range of 0 to height of image
        value = image[x, y] + random number in range of $-10$ to 10
        if value > 255
            value = 255
        else if value < 0
            value = 0
        image[x, y] = value
    end for
end for
```

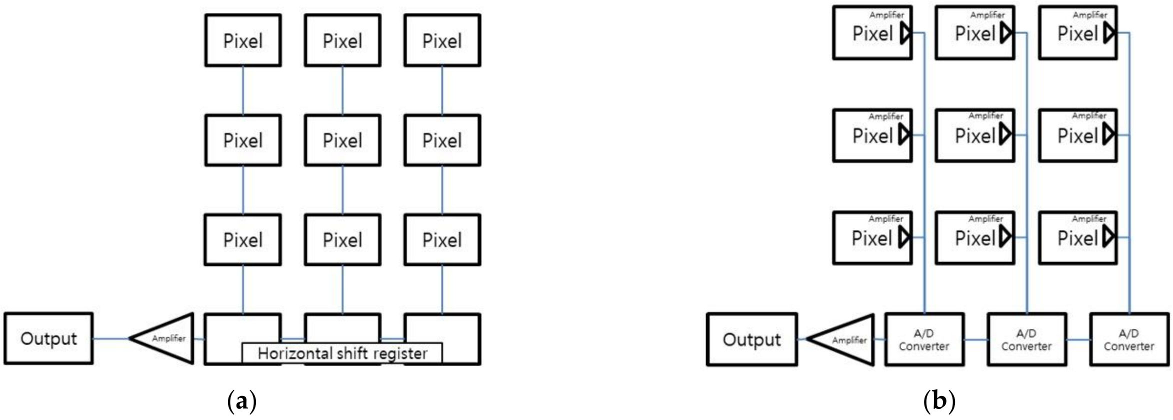
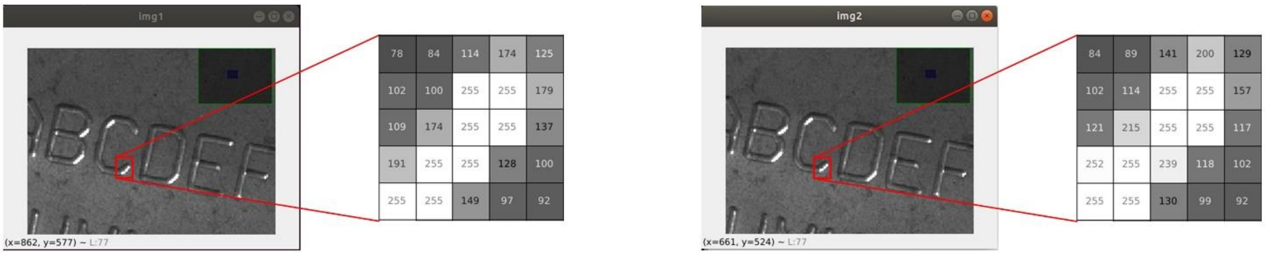


Figure 6. Difference between CCD and CMOS sensors: (**a**) CCD sensor array and (**b**) CMOS sensor array.



78	84	114	174	125
102	100	255	255	179
109	174	255	255	137
191	255	255	128	100
255	255	149	97	92

(**a**)

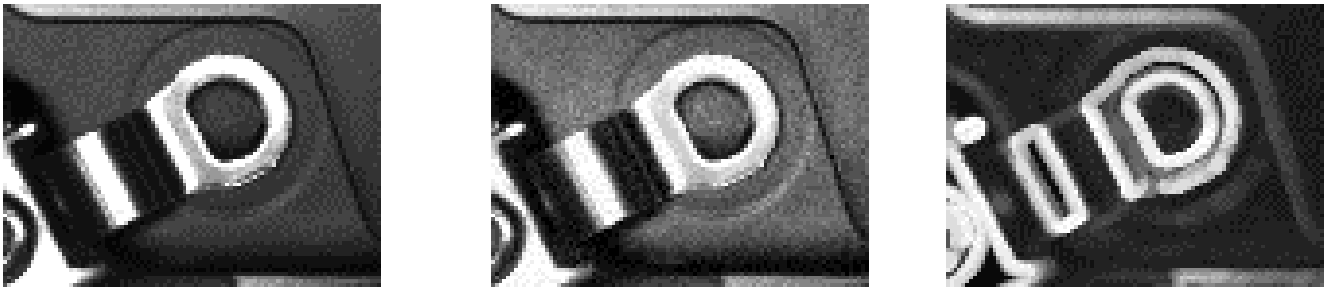
84	89	141	200	129
102	114	255	255	157
121	215	255	255	117
252	255	239	118	102
255	255	130	99	92

(**b**)

Figure 7. Pixel noise in the images captured by using the CMOS camera; images were captured at intervals of 1 s, under the same aperture, exposure, and lighting conditions. (**a**) Pixel values of the image taken first (**b**) Pixel values of the images taken after one second.

Contrast Limited Adaptive Histogram Equalization (CLAHE)

Even if the optical system that inspects the product is configured to minimize the effect of external light sources, the brightness of the captured image is sometimes different because of the external reflective light. Neural networks can get robust against brightness changes by adjusting the brightness distribution of the dataset. To equalize the brightness distribution, we used the CLAHE algorithm published by Pizer et al. [25]. Figure 8a,b shows the difference in brightness before and after the application of CLAHE, respectively.



(**a**) (**b**) (**c**)

Figure 8. (**a**) Patch of the original image. (**b**) Patch of the CLAHE-applied image. (**c**) Patch of the morphological gradient-applied image.

Gaussian Blur

If the focus of the lens is not aligned, the image of the inspection target is blurred. To ensure that the CNNs are robust to image blurring resulting from an inexperienced operator's lens manipulation, blur was applied via the Gaussian kernel generated through Equation (1). We applied the GaussianBlur function of OpenCV Python, and ksize = (11, 11), sigmaX = 11, and sigmaY = 11 were used as input factors.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

Morphological Gradient

Edges are extracted as structural features of the inspection target. Generally, the Canny Edge algorithm [26] is used for edge detection. However, we performed morphological gradient operations to preserve the importance of information while extracting all the structural information from the image under examination in the form of edges. We used the getStructuringElement function to obtain the kernel and morphologyEx to perform the morphological gradient operation. We used the input parameters of the getStructuringElement function as flag = cv2.MORPH_ELIPSE and ksize = (11, 11). Additionally, op = cv2.MORPH_GRADIENT for the morphologyEx function. Figure 8c shows the results of the morphological gradient operation. Input and output images are both grayscale images.

2.2.2. Three-Channel Augmentation

When reading images from OpenCV, the channel order is Blue–Green–Red. However, other image processing libraries read images in the order of Red–Green–Blue. To use the data independently of the libraries that read the images, we used images applied with morphological gradients in the Green channel. The Red and Blue channels combine the rest of the one-channel augmented images and the original images to create the three-channel images.

Figure 9a depicts an example of the order of one-channel images entered into each channel while creating a three-channel image. Figure 9b is a color image combined according to the proposed method. In addition, we prepared the data transformed into grayscale images, as shown in Figure 9c, to verify that neural networks are well-trained when trained with data of different features in all three channels and not well-trained only because of a large amount of data.

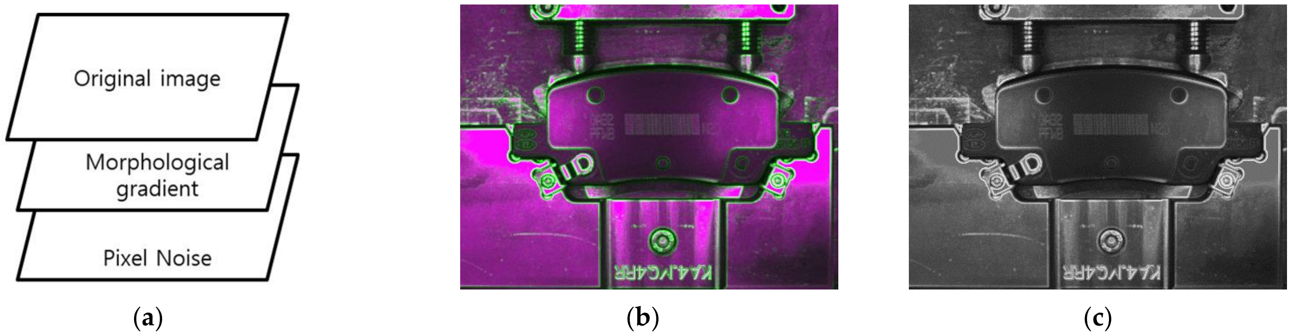


Figure 9. (**a**) One of the channel structures proposed in this study. (**b**) Color image produced by using data augmentation. (**c**) Grayscale images used to verify that there are effects other than the increased number of data.

The method used to preprocess one channel and combine it when augmenting to three channels is shown in Table 2. The number of data of the NEU-DET dataset and the brake pad dataset after the augmentation is shown in Tables 3 and 4.

Table 2. Proposed data augmentation methods.

Augmentation Methods	Data Augmentation Methods of Each Image
One-channel Augmentation	Pixel Noise CLAHE Gaussian Blur Morphological gradient
Three-channel Augmentation	Original + Morphological gradient + Pixel noise Original + Morphological gradient + Gaussian blur Original + Morphological gradient + CLAHE Pixel noise + Morphological gradient + Gaussian blur Pixel noise + Morphological gradient + CLAHE Gaussian blur + Morphological gradient + CLAHE

Table 3. Number of datasets after the augmentation, NEU-DET dataset.

Dataset Configuration	# of Training Datasets	# of Validation Datasets
Original dataset	240	60
Original dataset + one-channel mixed dataset	1200	300
Original dataset + three-channel mixed dataset	1680	420

Table 4. Number of datasets after the augmentation, brake pad dataset.

Dataset Configuration	# of Training Datasets	# of Validation Datasets
Original dataset	490	55
Original dataset + one-channel mixed dataset	2450	275
Original dataset + three-channel mixed dataset	3430	385

2.3. Networks

Unlike the ordinary CNN-based computer vision tasks, the machine vision problem has relatively few classes required to be classified or detected. Owing to the small number of classes to be inspected, the accuracy of the relatively simple neural networks is not significantly lower than that of the complex neural networks. It is more economical to increase the inspection speed in the production process at the factory. As a result, we focused on inspection speed, and the neural networks are chosen based on the inference speed in this paper.

Four types of datasets were trained by using the same hyperparameters.

2.3.1. Image Classification Networks

Image classification networks are trained by using the NEU-DET dataset. MobileNetV2 by Sandler [27] and Resnet18 by He et al. [28] are transfer-learned, using the prepared data. The framework to train both networks is Pytorch, and GPU is GTX 1080Ti.

Hyperparameters used in the training of both neural networks are shown in Table 5.

Table 5. Hyperparameters to train image classification networks using the NEU-DET dataset.

Learning Rate	Batch Size	Optimizer	Epochs
0.001	16	SGD	2

2.3.2. Object Detection Networks

There are two types of object detection networks, which are of two types and are shown in Figure 10. Figure 10a shows the architecture of the two-stage detector, which involves the following steps: (1) image input, (2) feature extraction, (3) region proposal, and (4) object classification. Although the object detection accuracy was high, the inference speed was relatively slow. Figure 10b shows the structure of the one-stage detector, which goes through the steps of image input, feature extraction, and object detection. It has the advantage of being less accurate albeit faster in inference than the two-stage detectors.

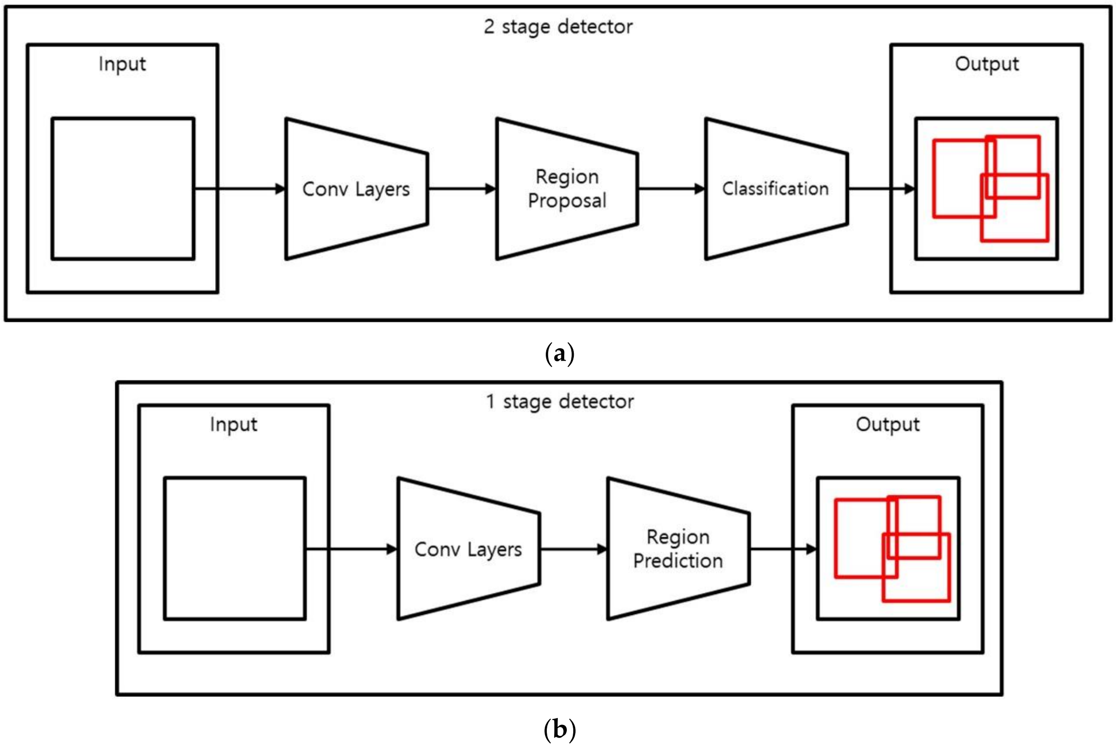


Figure 10. Generalized object detection architecture: (**a**) two-stage detector and (**b**) one-stage detector.

The neural network trained for object detection uses YOLOv4 [29] and YOLOv4-tiny. The framework to train both networks is Darknet, and GPU is GTX 1060; the generalized architecture of YOLOv4 and YOLOv4-tiny is shown in Figure 11.

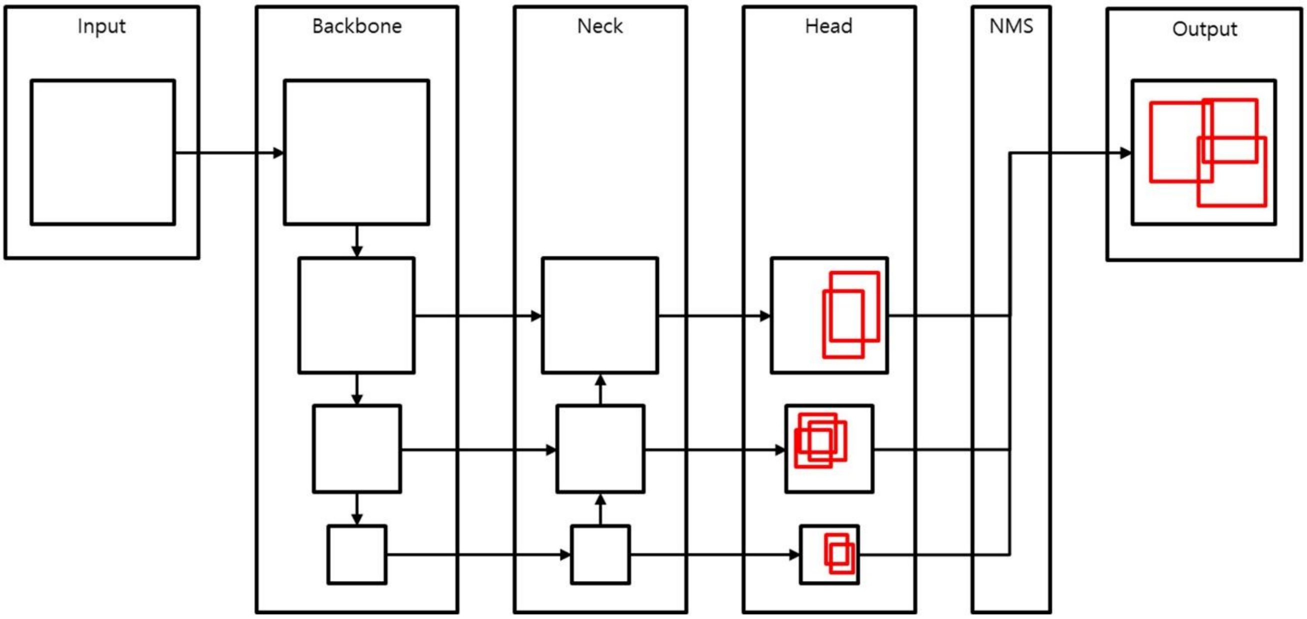


Figure 11. YOLOv4 and YOLOv4-tiny generalized architecture.

Hyperparameters used in the training of both neural networks are shown in Table 6.

Table 6. Hyperparameters to train object detection networks using the brake pad dataset.

Networks	Learning Rate	Batch Size	Subdivisions	Epochs
YOLOv4	0.0013	64	32	10
YOLOv4-tiny	0.00261	64	16	10

3. Results

3.1. Evaluation Metrics

3.1.1. Image Classification Metrics

Classification accuracy and F1 scores on the validation datasets were used for the evaluation of the trained networks. Classification accuracy is the ratio of results classified as correct for all the classification results and is calculated by using Equation (2). The F1 score is a harmonic mean of precision and recall, an indicator that allows a more accurate evaluation of the networks when the data label is unbalanced. F1 score is calculated using Equation (3). Precision and recall are calculated by using Equations (4) and (5), respectively. The definitions of *TP*, *FP*, *FN*, and *TN* are tabulated in Table 7.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$\text{F1}_{Score} = \frac{2 \times precision \times recall}{precision + recall} \tag{3}$$

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$recall = \frac{TP}{TP + FN} \tag{5}$$

Table 7. Details of *TP*, *FP*, *FN*, and *TN*.

		Prediction	
		Positive	**Negative**
Ground truth	Positive	*TP* (true positive)	*FN* (false negative)
	Negative	*FP* (false positive)	*TN* (true negative)

3.1.2. Object Detection Metrics

We can use the mean average precision (mAP) as an evaluation metric for object detection neural networks. mAP metrics include mAP@0.5 and mAP@0.5:0.95. Everingham et al. [30] used mAP@0.5 at the Pascal VOC competition and mAP@0.5:0.95 at the COCO Object Detection competition [24]. Moreover, mAP@0.5 is the average value of the class-wise average precision (AP) for an intersection over union (IoU) threshold of 0.5. Similarly, mAP@0.5:0.95 is the average value of 10 APs for the IoU threshold of 0.5–0.95 with an interval of 0.05. IoU refers to the superposition ratio of the predicted object box to the ground-truth object box by the object detection neural network and is calculated as follows:

$$\text{IoU} = \frac{box_{prediction} \cap box_{groundtruth}}{box_{prediction} \cup box_{groundtruth}} \tag{6}$$

AP is the area below the line on the precision–recall graph.

In the object detection task of machine vision, it is essential to locate the object accurately. Therefore, the value of mAP@0.5:0.95 is more important than the value of mAP@0.5 because mAP@0.5:0.95 needs to compute a high IoU ratio while mAP@0.5 does not.

3.2. *Quantitative Results*

The image classification networks and object detection networks were trained using four prepared datasets for comparison, including the data augmentation method proposed in this study. The types of datasets are as follows: (1) original dataset, (2) one-channel dataset with the original dataset, (3) three-channel grayscale dataset with the original dataset, and (4) three-channel color dataset with the original dataset.

During the evaluation process, the trained network was validated (1) using the original validation data with the augmented validation data and (2) only using the original validation data.

The networks were trained ten times for each experimental condition to verify the reproducibility and repeatability of each metric. Subsequently, we showed average, standard deviation, and boxplot for each metric.

3.2.1. Image Classification

We train two neural networks with the NEU-DET dataset to demonstrate that the proposed data augmentation method affects image classification tasks.

MobileNetV2 Results

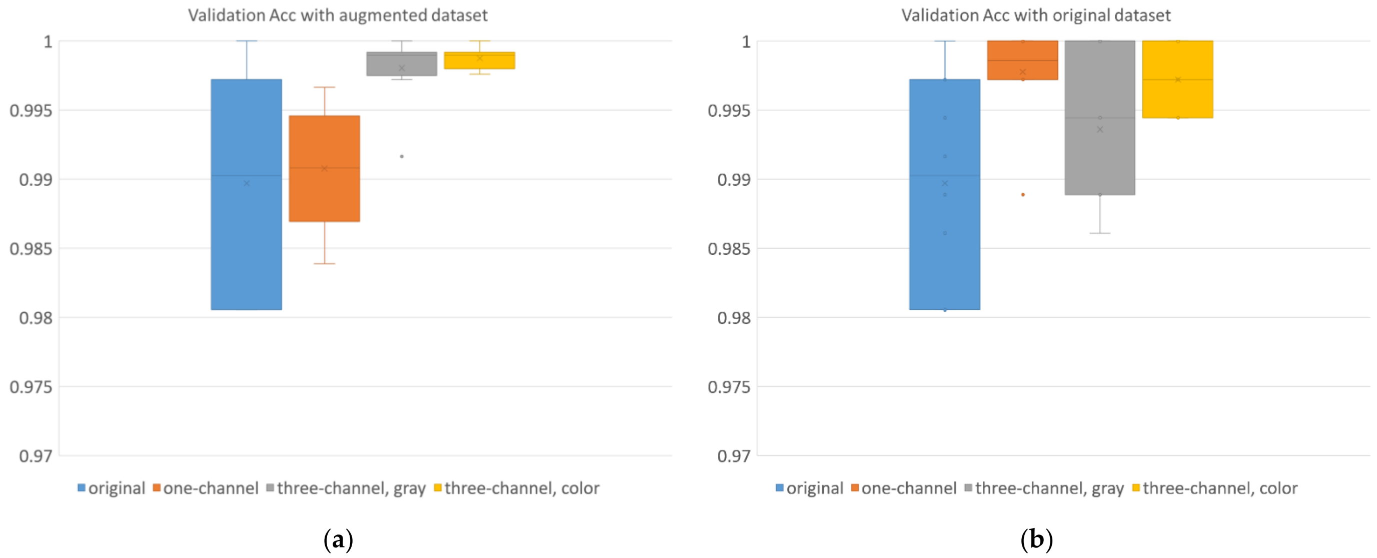
With augmented validation data and original validation data, the average and standard deviation of the classification accuracy and the F1 score are obtained and shown in Table 8.

Table 8. Average and standard deviation of MobileNetV2 validation result.

		Original		**One-Channel**		**Three-Channel Grayscale**		**Three-Channel Color**	
		Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
Augmented dataset	accuracy	0.990	0.008	0.991	0.004	0.998	0.002	0.999	0.001
	F1 score	0.990	0.007	0.991	0.004	0.998	0.002	0.999	0.001
Original dataset	accuracy	0.990	0.008	0.998	0.003	0.994	0.002	0.997	0.002
	F1 score	0.990	0.007	0.998	0.003	0.994	0.005	0.997	0.002

Due to balanced datasets, the accuracy and F1 score tend to be the same. Moreover, the networks trained by the proposed three-channel augmented color dataset has higher accuracy and lower standard deviation. In the validation results with the original dataset, the average accuracy of the networks trained with the one-channel augmented dataset is higher than that of networks trained with the three-channel augmented color dataset. However, the networks trained with the three-channel augmented color dataset have a lower standard deviation.

Figure 12 shows the boxplots of validation accuracy on each dataset. Figure 12a is the validation results with the augmented dataset, and Figure 12b is the validation results with the original dataset. The network trained by the three-channel augmented color dataset shows good accuracy than the network trained by the original dataset in both validation results. The networks trained with the one-channel augmented dataset do not have higher accuracy in Figure 12a. However, although it has an outlier that cannot assume consistent performance, it has higher accuracy in Figure 12b. The networks trained with the three-channel augmented grayscale dataset have an outlier in Figure 12a.



(**a**) (**b**)

Figure 12. Boxplots of validation accuracy with MobileNetV2: (**a**) results validate with the augmented dataset; (**b**) results validate with the original dataset.

By training with the three-channel augmented color dataset, we can assume that the performance of networks will not fall below expected performance in both cases.

Resnet18 Results

With augmented validation data and original validation data, the average and standard deviation of the classification accuracy and the F1 score are obtained and shown in Table 9.

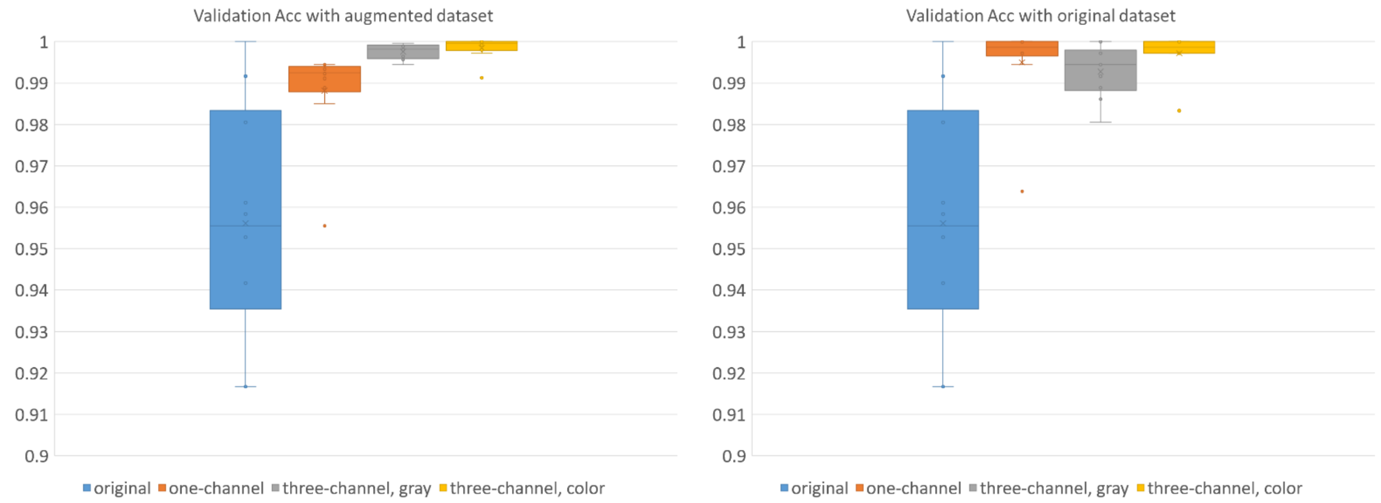
Table 9. Average and standard deviation of Resnet18 validation result.

		Original		**One-Channel**		**Three-Channel Grayscale**		**Three-Channel Color**	
		Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
Augmented dataset	accuracy	0.956	0.029	0.988	0.012	0.998	0.002	0.998	0.003
	F1 score	0.960	0.025	0.988	0.011	0.998	0.002	0.998	0.003
Original dataset	accuracy	0.956	0.029	0.995	0.011	0.993	0.002	0.997	0.005
	F1 score	0.960	0.025	0.995	0.010	0.993	0.006	0.997	0.005

The validation accuracy of Resnet18 is lower than that of MobileNetV2. ResNet18 has more parameters to train than MobilenetV2. Since only two epochs have been trained, it can be expected that Resnet18 is not optimized parameters to classify the NEU-DET dataset. However, the tendency of validation results via trained datasets can be confirmed.

Similar to the validation results of MobileNetV2, accuracy and F1 score tend to be the same. Moreover, the average validation accuracy of the networks trained with the three-channel augmented color dataset is higher than other results.

Figure 13 shows the boxplots of validation accuracy on each dataset. Figure 13a is the validation results with the augmented dataset, and Figure 13b is the validation results with the original dataset. The network trained by the three-channel augmented color dataset shows good accuracy than the network trained by the original dataset in both validation results.



(**a**) (**b**)

Figure 13. Boxplots of validation accuracy with Resnet18: (**a**) results validate with the augmented dataset; (**b**) results validate with the original dataset.

The validation results of Resnet18 also show that the networks trained with the three-channel augmented color dataset have high average accuracy.

As a result, the proposed data augmentation method was effective for the image classification task, which uses the grayscale image captured by mono cameras for surface inspection.

3.2.2. Object Detection

We trained two neural networks with the brake pad dataset to demonstrate that the proposed data augmentation method affects object detection tasks.

YOLOv4 Results

We tabulated average and standard deviations for all mAPs of trained YOLOv4 networks to determine how mAP changes with the IoU threshold. Each mAP and mAP@0.5:0.95 by augmented data are shown in Table 10, and corresponding results by the original data are shown in Table 11.

Table 10. YOLOv4 object detection validation results with the augmented dataset.

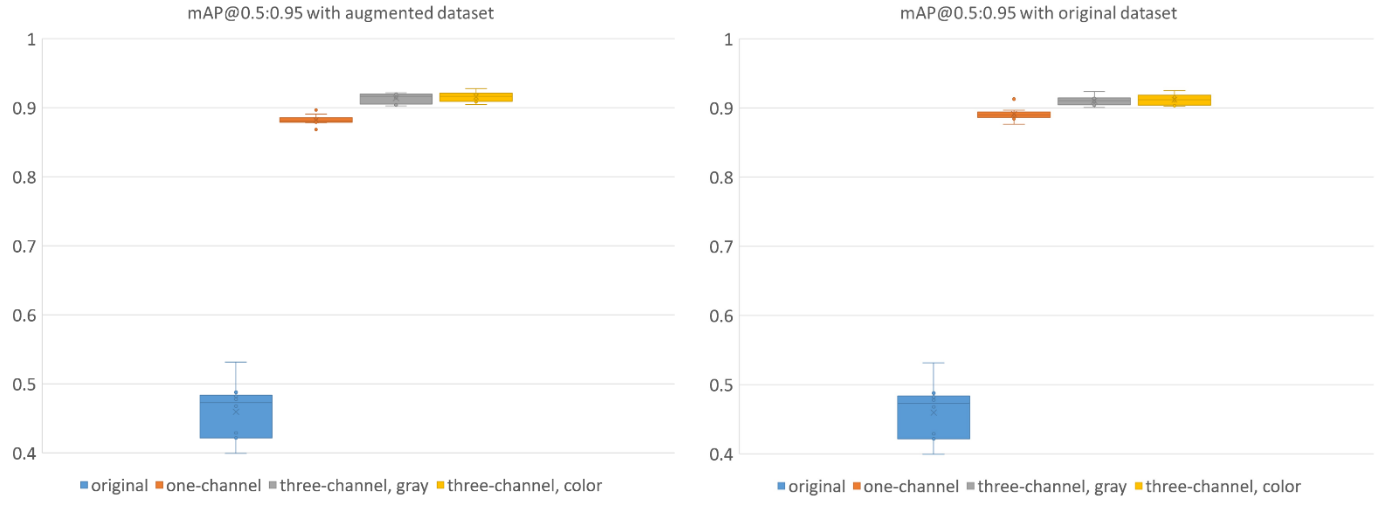
	Original		**One-Channel**		**Three-Channel Grayscale**		**Three-Channel Color**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
mAP@0.5	0.894	0.044	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.55	0.857	0.072	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.6	0.817	0.079	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.65	0.727	0.092	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.7	0.588	0.101	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.75	0.397	0.049	0.998	0.001	1.000	0.001	1.000	0.000
mAP@0.8	0.224	0.061	0.986	0.010	0.996	0.001	0.997	0.001
mAP@0.85	0.077	0.033	0.963	0.016	0.989	0.001	0.988	0.001
mAP@0.9	0.016	0.015	0.726	0.030	0.884	0.026	0.884	0.018
mAP@0.95	0.001	0.001	0.150	0.062	0.272	0.069	0.293	0.063
mAP@0.5:0.95	0.460	0.040	0.882	0.008	0.914	0.007	0.916	0.007

Table 11. YOLOv4 object detection validation results with the original dataset.

	Original		One-Channel		Three-Channel Grayscale		Three-Channel Color	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
mAP@0.5	0.894	0.044	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.55	0.857	0.072	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.6	0.817	0.079	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.65	0.727	0.092	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.7	0.588	0.101	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.75	0.397	0.049	0.999	0.002	0.999	0.001	1.000	0.000
mAP@0.8	0.224	0.061	0.986	0.016	0.996	0.001	0.997	0.000
mAP@0.85	0.077	0.033	0.972	0.017	0.989	0.001	0.988	0.002
mAP@0.9	0.016	0.015	0.767	0.022	0.885	0.034	0.860	0.045
mAP@0.95	0.001	0.001	0.192	0.086	0.234	0.052	0.275	0.064
mAP@0.5:0.95	0.460	0.040	0.892	0.009	0.910	0.007	0.912	0.008

All results show the phenomenon in which the IoU threshold increases and the mAP value decreases similarly. Table 10 shows the high mAPs and mAP@0.5:0.95 of YOLOv4 networks trained with the three-channel augmented color dataset. Nevertheless, in Table 11, some average mAP of YOLOv4 networks trained with the three-channel augmented grayscale dataset has a higher average mAP than YOLOv4 networks trained with the three-channel augmented color dataset. However, the YOLOv4 networks trained with the three-channel augmented color dataset has the highest mAP@0.5:0.95.

The boxplot of mAP0.5:0.95 obtained from both datasets is shown in Figure 14.



(**a**) (**b**)

Figure 14. Boxplots of validation mAP@0.5:0.95 with YOLOv4: (**a**) results validate with the augmented dataset; (**b**) results validate with the original dataset.

Figure 14 shows that the YOLOv4 networks can better infer performance when trained with the three-channel augmented color dataset than trained with the other datasets.

YOLOv4-Tiny Results

We also tabulated average and standard deviations for all mAPs of trained YOLOv4-tiny networks. Each mAP and mAP@0.5:0.95 by augmented data are shown in Table 12, and corresponding results by the original data are shown in Table 13.

Table 12. YOLOv4-tiny object detection validation results with the augmented dataset.

	Original		**One-Channel**		**Three-Channel Grayscale**		**Three-Channel Color**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
mAP@0.5	0.857	0.044	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.55	0.761	0.067	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.6	0.620	0.080	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.65	0.464	0.075	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.7	0.306	0.061	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.75	0.160	0.045	0.997	0.002	0.997	0.003	0.999	0.001
mAP@0.8	0.070	0.019	0.979	0.005	0.988	0.004	0.985	0.004
mAP@0.85	0.013	0.009	0.897	0.019	0.954	0.005	0.958	0.009
mAP@0.9	0.001	0.001	0.548	0.039	0.657	0.054	0.717	0.043
mAP@0.95	0.000	0.000	0.050	0.027	0.081	0.040	0.092	0.031
mAP@0.5:0.95	0.325	0.028	0.847	0.006	0.868	0.009	0.875	0.006

Table 13. YOLOv4-tiny object detection validation results with the original dataset.

	Original		**One-Channel**		**Three-Channel Grayscale**		**Three-Channel Color**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
mAP@0.5	0.857	0.044	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.55	0.761	0.067	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.6	0.620	0.080	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.65	0.464	0.075	1.000	0.000	1.000	0.000	1.000	0.000
mAP@0.7	0.306	0.061	1.000	0.000	1.000	0.001	1.000	0.000
mAP@0.75	0.160	0.045	0.996	0.003	0.999	0.003	0.998	0.002
mAP@0.8	0.070	0.019	0.983	0.005	0.989	0.006	0.986	0.005
mAP@0.85	0.013	0.009	0.923	0.029	0.939	0.017	0.951	0.015
mAP@0.9	0.001	0.001	0.593	0.049	0.645	0.079	0.704	0.050
mAP@0.95	0.000	0.000	0.077	0.052	0.068	0.049	0.097	0.036
mAP@0.5:0.95	0.325	0.028	0.857	0.011	0.864	0.012	0.874	0.007

Similar to the results of YOLOv4, Tables 12 and 13 show the tendency in which the IoU threshold increases and the mAP value decreases. In Table 12, mAP@0.8 of YOLOv4-tiny networks trained with the three-channel augmented color dataset is lower than that of networks trained with the three-channel augmented grayscale dataset. In Table 13, mAP@0.75 and mAP@0.8 of YOLOv4-tiny networks trained with the three-channel augmented color dataset are lower than that of networks trained with the three-channel augmented grayscale dataset. However, in most cases, YOLOv4-tiny networks trained with the three-channel augmented color dataset have the highest mAP value.

The boxplot of mAP0.5:0.95 obtained from both datasets is shown in Figure 15.

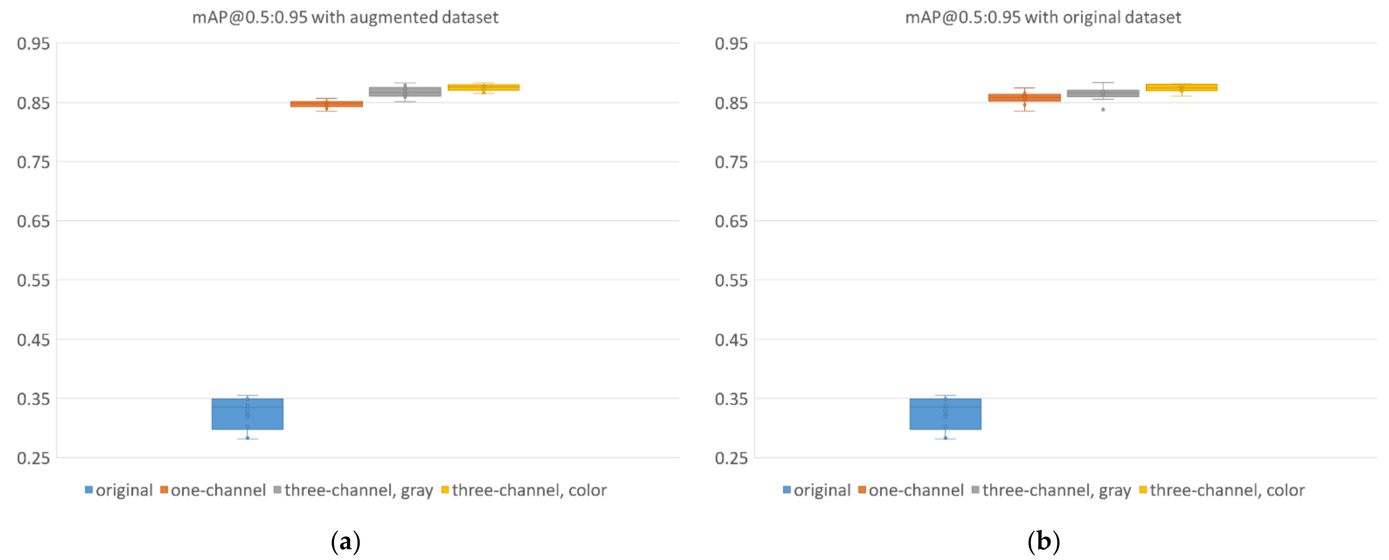


Figure 15. Boxplots of validation mAP@0.5:0.95 with YOLOv4-tiny: (**a**) results validate with the augmented dataset; (**b**) results validate with the original dataset.

Figure 15 also shows that the YOLOv4-tiny networks can have outstanding inference performance when trained with the three-channel augmented color dataset.

As a result, the proposed data augmentation method effective in the grayscale image data captured by mono cameras in the surface inspection by object-detection tasks.

4. Discussion

In the experiments performed in this study, the NEU-DET dataset was used to train MobileNetV2 and Resnet18 for the image classification task, and the braked pad dataset was used to train YOLOv4 and YOLOv4-tiny for the object detection task. The image classification task and the object-detection task show that the proposed data augmentation method effectively trains the CNNs for machine vision systems using mono cameras.

This shows that the CNNs trained with the proposed three-channel augmented color dataset perform better than the CNNs trained using the other methods. Suppose the CNNs perform better only owing to the number of augmented data. In that case, there should be no difference in the performance of the CNNs trained with the three-channel augmented color dataset and the ones trained with the three-channel augmented grayscale dataset. However, the results show that the CNNs trained with the three-channel augmented color dataset preprocessed by using different methods for each channel performed better.

The reasons are as follows. (1) We imitate the possible variations in the image captured with mono cameras: random oscillation of pixel values in CMOS sensors, brightness changes caused by the light conditions, and blurring effect caused by improper lens alignment. Furthermore, we extract structural information needed for surface defects by extracting the edges. In most experimental results, validation results show that the network trained with the one-channel augmented dataset performs better than the network trained with the original dataset. These results imply that the data augmentation based on characteristics of machine vision is effective in training the CNN for surface defect inspection. (2) When transfer learning on typical CNNs, we assume that the input of CNN is a color image. Moreover, color images have different information for each channel. However, the machine vision system using mono cameras uses grayscale images. Moreover, existing machine vision studies have trained CNNs by opening them as color images so that the three channels have the same original grayscale information. In the work of Burduja et al. [23], they trained CNN by preprocessed color images merged from three grayscale images that extracted different features from one CT image. Based on this work, we train by synthesizing the various information used for inspection in the grayscale machine vision images into color images. The CNNs for surface inspection in the machine vision systems using mono cameras can be trained with a small amount of unbalanced dataset with the proposed data augmentation method.

5. Conclusions

This study proposes a data augmentation method for training high-performance CNNs in machine vision applications using mono cameras. There has been no research to utilize and apply the characteristics of the images to the CNNs, which can arise from mono cameras, in the industry. This work shows that the CNN-based machine vision using mono cameras can perform when trained with combined three-channel images from multiple variations of images.

Future work will include the application of defect inspection via instance segmentation and anomaly detection. The applicability of the proposed data augmentation method to instance segmentation and anomaly detection will be confirmed in future work.

Author Contributions: Conceptualization, J.W.; methodology, J.W.; software, J.W.; validation, J.W.; formal analysis, J.W.; investigation, J.W.; resources, J.W.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, J.W. and S.L.; visualization, J.W.; supervision, S.L.; project administration, S.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the SAERON AUTOMOTIVE CO., LTD. South Korea.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author. The data are not publicly available, due to security.

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

References

1. Jiang, B.; Wang, C.-C.; Tsai, D.-M.; Lu, C.-J. LCD surface defect inspection using machine vision. In Proceedings of the Fifth Asia Pacific Industrial Engineering and Management Systems Conference, Gold Coast, Australia, 12–15 December 2004; pp. 24.7.1–24.7.9.
2. Böttger, T.; Ulrich, M. Real-time texture error detection on textured surfaces with compressed sensing. *Pattern Recognit. Image Anal.* **2016**, *26*, 88–94. [CrossRef]
3. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [CrossRef]
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
5. Bergmann, P.; Fauser, M.; Sattlegger, D.; Steger, C. MVTec AD—A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9584–9592.
6. Chang, F.; Liu, M.; Dong, M.; Duan, Y. A mobile vision inspection system for tiny defect detection on smooth car-body surfaces based on deep ensemble learning. *Meas. Sci. Technol.* **2019**, *30*. [CrossRef]
7. Chien, J.-C.; Wu, M.-T.; Lee, J.-D. Inspection and Classification of Semiconductor Wafer Surface Defects Using CNN Deep Learning Networks. *Appl. Sci.* **2020**, *10*, 5340. [CrossRef]
8. Ding, F.; Zhuang, Z.; Liu, Y.; Jiang, D.; Yan, X.; Wang, Z. Detecting Defects on Solid Wood Panels Based on an Improved SSD Algorithm. *Sensors* **2020**, *20*, 5315. [CrossRef]
9. Lin, J.; Yao, Y.; Ma, L.; Wang, Y. Detection of a casting defect tracked by deep convolution neural network. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 573–581. [CrossRef]
10. Lu, M.; Chen, C.-L. Detection and Classification of Bearing Surface Defects Based on Machine Vision. *Appl. Sci.* **2021**, *11*, 1825. [CrossRef]
11. Ruiz, L.; Torres, M.; Gómez, A.; Díaz, S.; González, J.M.; Cavas, F. Detection and Classification of Aircraft Fixation Elements during Manufacturing Processes Using a Convolutional Neural Network. *Appl. Sci.* **2020**, *10*, 6856. [CrossRef]
12. Sun, X.; Gu, J.; Huang, R.; Zou, R.; Giron Palomares, B. Surface Defects Recognition of Wheel Hub Based on Improved Faster R-CNN. *Electronics* **2019**, *8*, 481. [CrossRef]

13. Urbonas, A.; Raudonis, V.; Maskeliūnas, R.; Damaševičius, R. Automated Identification of Wood Veneer Surface Defects Using Faster Region-Based Convolutional Neural Network with Data Augmentation and Transfer Learning. *Appl. Sci.* **2019**, *9*, 4898. [CrossRef]
14. Wang, T.; Chen, Y.; Qiao, M.N.; Snoussi, H. A fast and robust convolutional neural network-based defect detection model in product quality control. *Int. J. Adv. Manuf Tech.* **2018**, *94*, 3465–3471. [CrossRef]
15. Wen, S.; Chen, Z.; Li, C. Vision-Based Surface Inspection System for Bearing Rollers Using Convolutional Neural Networks. *Appl. Sci.* **2018**, *8*, 2565. [CrossRef]
16. Yang, Y.; Pan, L.; Ma, J.; Yang, R.; Zhu, Y.; Yang, Y.; Zhang, L. A High-Performance Deep Learning Algorithm for the Automated Optical Inspection of Laser Welding. *Appl. Sci.* **2020**, *10*, 933. [CrossRef]
17. 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]
18. Zhang, J.; Liu, H.; Cao, J.; Zhu, W.; Jin, B.; Li, W. A Deep Learning Based Dislocation Detection Method for Cylindrical Crystal Growth Process. *Appl. Sci.* **2020**, *10*, 7799. [CrossRef]
19. He, Y.; Song, K.; Meng, Q.; Yan, Y. An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1493–1504. [CrossRef]
20. Lv, X.; Duan, F.; Jiang, J.-J.; Fu, X.; Gan, L. Deep Active Learning for Surface Defect Detection. *Sensors* **2020**, *20*, 1650. [CrossRef]
21. Yang, J.; Li, S.; Wang, Z.; Dong, H.; Wang, J.; Tang, S. Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges. *Materials* **2020**, *13*, 5755. [CrossRef] [PubMed]
22. Xie, Y.; Richmond, D. Pre-training on Grayscale ImageNet Improves Medical Image Classification. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018; pp. 476–484.
23. Burduja, M.; Ionescu, R.T.; Verga, N. Accurate and Efficient Intracranial Hemorrhage Detection and Subtype Classification in 3D CT Scans with Convolutional and Long Short-Term Memory Neural Networks. *Sensors* **2020**, *20*, 5611. [CrossRef]
24. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Lecture Notes in Computer Science, Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; pp. 740–755.
25. Pizer, S.M.; Johnston, R.E.; Ericksen, J.P.; Yankaskas, B.C.; Muller, K.E. Contrast-limited adaptive histogram equalization: Speed and effectiveness. In Proceedings of the First Conference on Visualization in Biomedical Computing, Atlanta, GA, USA, 22–25 May 1990; pp. 337–345.
26. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]
27. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv* **2018**, arXiv:1801.04381.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
29. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
30. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]