



# Article System Invariant Method for Ultrasonic Flaw Classification in Weldments Using Residual Neural Network

Jinhyun Park<sup>1</sup>, Seung-Eun Lee<sup>1</sup>, Hak-Joon Kim<sup>1,\*</sup>, Sung-Jin Song<sup>1</sup> and Sung-Sik Kang<sup>2</sup>

- <sup>1</sup> Department of Mechanical Engineering, College of Engineering, Sungkyunkwan University, Suwon 16419, Korea; bingsu09@skku.edu (J.P.); nmop00@skku.edu (S.-E.L.); sjsong@skku.edu (S.-J.S.)
- <sup>2</sup> Korea Institute of Nuclear Safety, Daejeon 34142, Korea; sskang@kins.re.kr

\* Correspondence: hjkim21c@skku.edu; Tel.: +82-31-290-7460

Abstract: The industrial use of ultrasonic flaw classification using neural networks in weldments must overcome many challenges. A major constraint is the use of numerous systems, including a combination of transducers and equipment. This causes high complexity in the datasets used in the training of neural networks, which decreases performance. In this study, the performance of a neural network was enhanced using signal processing on an ultrasonic weldment flaw dataset to achieve system invariance. The dataset contained 5839 ultrasonic flaw signals collected by various types of transducers connected to KrautKramer USN60. Every signal in the dataset was from 45 FlawTech/Sonaspection weldment specimens with five types of flaw: crack, lack of fusion, slag inclusion, porosity, and incomplete penetration. The neural network used in this study is a residual neural network with 19 layers. The performance evaluation of the same network structure showed that the original database can achieve  $62.17\% \pm 4.13\%$  accuracy, and that the invariant database using the system invariant method can achieve  $91.45\% \pm 1.77\%$  accuracy. The results demonstrate that using a system invariant method for ultrasonic flaw classification in weldments can improve the performance of a neural network with a highly complex dataset.



## 1. Introduction

Ultrasonic testing is a non-destructive testing technique that is commonly used in many industries. Because most defects occur in weldments, ultrasonic inspections are usually focused on weldments, and thus detecting and classifying ultrasonic flaw signals is a long-standing task in the field of ultrasonic testing. Determining flaws requires numerous experiences, and it is difficult to perform non-destructive testing efficiently because there is the possibility of human misjudgment. To solve this problem, many studies on ultrasonic testing to determine the presence, type, shape, and size of flaws using artificial intelligence are being conducted.

Song et al. [1] used probabilistic neural networks on ultrasonic weldment defect signals. Masnata et al. [2] used a neural network and Fisher discriminant analysis to classify cracks and volumetric defects. Polikar et al. [3] applied a neural network to develop a frequency-invariant weldment defect classification system for pipes. Margrave et al. [4] used neural networks to detect flaws in steel pipes. Song et al. [5] developed an intelligent ultrasonic evaluation system (IUES) for real-time weldment flaw classification. Simas Filho et al. [6] developed a flexible decision support system based upon neural network and principal component analysis to evaluate fiber-metal laminates. Cruz et al. [7] also developed a decision support system that selects parameters for the neural network to detect flaws in steel-welded joints. Meng et al. [8]. used a combination of wavelet transform and a convolutional neural network to classify void and delamination defects in CERP specimens. With the exception of [4], all these methods are limited due to their complexity [9–11].



Citation: Park, J.; Lee, S.-E.; Kim, H.-J.; Song, S.-J.; Kang, S.-S. System Invariant Method for Ultrasonic Flaw Classification in Weldments Using Residual Neural Network. *Appl. Sci.* 2022, *12*, 1477. https://doi.org/ 10.3390/app12031477

Academic Editor: Marco Scalerandi

Received: 8 December 2021 Accepted: 12 January 2022 Published: 29 January 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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/). A large amount of data must be collected to ensure the performance of artificial intelligence (AI). It is highly difficult to collect a large amount of real inspection data in the ultrasonic testing research field. Many studies collect data from specimens to solve this problem; however, there is a limit on the amount of data that one research group can collect. This works well for systems with little variation, such as those that use an ultrasonic transducer from the same manufacturer. However, inspectors use various types of transducer and equipment for industrial ultrasonic testing. The limitation of dataset size can be a problem in this scenario for the following reasons. First, the AI may not sufficiently train signals from each system. Second, flaw signals from various systems may interfere with the training process, finding global minimum. To solve this problem, this study proposes a system invariant technique to reduce the complexity of ultrasonic weldment flaw databases and compares the performance of two different databases, one without a system invariant method and the other with a system invariant method, using the same neural network architecture.

## 2. Ultrasonic Weldments Flaw Databases

To properly train neural networks, a database of well-inspected ultrasonic weldments flaw signals was developed. The signals were collected from 45 FlawTech/Sonaspection weldment specimens, including five types of flaw: crack, lack of fusion, slag inclusion, porosity, and incomplete penetration. The signals were collected with 21 different ultrasonic transducers to make the system complex. This group of transducers included 2, 4, and 5 MHz central frequencies, 45, 60, and 70 degree inspection angles, and four different manufacturers (KrautKramer, GE, Olympus, and TKS). All the inspections were performed using KrautKramer and USN-60 flaw detector as the ultrasonic pulser-receiver instrument and a 50 MHz sampling frequency. The list of transducers is presented in Table 1.

Table 1. List of transducers.

Manufacturer	K	KrautKramer		G	E	Olympus	TKS
	MWB 45-2	WB 45-2	WB 45-4	MWB 45-4	SWB 45-5	A430S (45°, 2 MHz)	4C14 X 14A45
	MWB 60-2	WB 60-2	WB 60-4	MWB 60-4	SWB 60-5	A430S (60°, 2 MHz)	4C14 X 14A60
	MWB 70-2	WB 70-2	WB 70-4	MWB 70-4	SWB 70-5	A430S (70°, 2 MHz)	4C14 X 14A70

To construct the database, 5839 signals were acquired for each flaw and, subsequently, 2048 points were extracted with the flaw signal (Figure 1). The number of each weldment flaw is presented in Table 2.



Figure 1. Flaw A scan signal example for database construction: (a) crack; (b) porosity.

Title 1	Flaws	No. of Signals
1	Crack	2899
2	Lack of Fusion	1196
3	Slag Inclusion	634
4	Porosity	493
5	Incomplete Penetration	617
	Total	5839

Table 2. Ultrasonic weldment flaw database.

To check the performance of the neural networks for training, the database was separated. To balance the learning between each flaw, 40% of the crack signal and 10% of other flaw signals were randomly divided into the test signal. The compositions of the training and testing signals are described in Table 3.

Table 3. Ultrasonic weldment flaw database.

Weldment Flaw Database				
Training Database Testing Database				
Crack	1740	Crack	1159	
Lack of Fusion	1076	Lack of Fusion	120	
Slag Inclusion	571	Slag Inclusion	63	
Porosity	444	Porosity	49	
Incomplete Penetration	555	Incomplete Penetration	62	
Total	4386	Total	1453	

## 3. Database Augmentation

Although 4386 signals were obtained, considering the complexity of the system, the amount of data was not sufficient. Therefore, the number of training data was increased through augmentation. In terms of data augmentation, noise addition and time shifting were effective in various studies [3,10].

Because ultrasonic testing is usually performed in harsh environments, the possibility of the testing results being corrupted by electric noise, presented as white Gaussian noise (WGN), exists. Additive WGN was used to initiate this. The mathematical details of WGN are described in [12,13]. The WGN is added with two different signal-to-noise ratios (SNR) levels i.e., SNR 20 and SNR 15 (Figure 2). The number of testing signals increased threefold from 4386 to 13,158, including the original signal.



Figure 2. Additive WGN: (a) non-; (b) SNR 15.

Another augmentation technique was adjusted after developing the augmentation database with WGN. In terms of ultrasonic testing signals, time shifting is equivalent to changing the distance between the transducer and the flaw location [10]. Therefore, new signals were created that shifted 20 and 10 points backward, and 20 and 10 points forward (Figure 3). Because the system features a sampling frequency of 50 MHz, one point on the time axis is 0.02  $\mu$ s. Because of time shifting, including the original signal, the testing signal number increased fivefold, from 13,158 to 65,790. This augmented dataset is referred to as the original database and is described in Table 4.



**Figure 3.** Time shifting of signals: (**a**) 10 points (0.2 µs) forward and backward shift; (**b**) 20 points (0.4 µs) forward and backward shift.

Table 4. Augmented	database	(original	datał	base).

	Origi	nal Database		
Training Database Testing Database				
Crack	26,100	Crack	1159	
Lack of Fusion	16,140	Lack of Fusion	120	
Slag Inclusion	8565	Slag Inclusion	63	
Porosity	6660	Porosity	49	
Incomplete Penetration	8325	Incomplete Penetration	62	
Total	65,790	Total	1453	

## 4. Artificial Neural Networks

Artificial neural networks (ANNs) are computational neural models inspired by a sequence of biological neurons. ANNs operate on the basis of the backpropagation algorithm proposed by Rumelhart et al. [14]. In the early stage of ANNs, a significantly simplified form of neural network, known as a multilayer perceptron (MLP), was developed, which led to fully connected neural networks (FCNNs) [15]. FCNNs are composed of input, hidden, and output layers. Each layer contains nodes with weights, and each node in the layer is connected to every node in the next layer. Deep neural networks are networks with more than two hidden layers.

The process of training a neural network involves adjusting the weight and connections between the layers. First, the data enter the input layer and pass through the hidden layers, while the dot product between the input data and weight matrix, whose size depends on the shape of the hidden layers, is computed. While computing, the activation function is used for tuning the training direction. After calculating the output, a loss function is computed, which gives the error between the true label and the predicted output. The error is then inverted using a backpropagation algorithm, and the weights are slightly updated to reduce this error. This process can be repeated until the error becomes sufficiently small or saturated [16–19].

#### 4.1. Convolutional Neural Network

Convolutional neural networks (CNNs) were first proposed by Lecun et al. [20]. They later gained popularity in the computer vision community [21]. CNNs feature a convolutional layer that is connected to a specific spatial region in the previous layer and is based on the size of a defined filter, in contrast to FCNNs. This architecture allows the network to focus on lower-level features before assembling them into higher-level features. CNNs also include pooling layers, which help to reduce the computational load by downsampling the input data, but increase the complexity of the network [22].

#### 4.2. Residual Neural Network

Residual neural networks (ResNets), first proposed by Kaiming He et al. [23], are based on CNN but include shortcuts between the input and output of the unit layer. If the input data are referred to as x, the output of ResNets is described by Equation (1):

$$F(x) + x \tag{1}$$

where F is the weight of computation. The calculational direction of ResNets minimizes the value of F(x) + x, which means F(x) becomes 0 since x is a fixed (input) value here. Conventional CNNs feature a weak spot in significantly deep networks, as their performance degrades after a certain depth. However, significantly deep networks are required to train with large amounts of data. ResNets solve this problem through their distinct network architecture as previously described, obtaining the best results by training 152 layers of ResNet using the CIFAR-10 dataset [24–27].

#### 4.3. Residual Neural Network Architecture

The ResNet used in this study was designed in Keras, based on Tensorflow (Google open-source software for deep learning). The algorithm included an input layer, nineteen convolutional layers, and three stages of ResNet blocks. Each stage features the same architecture, but a different number of feature maps. The detailed architecture of the stage with feature map FM is described in Table 5. The final ResNet architecture used when these unit stages were combined is shown in Table 6.

Table 5. Detailed architecture of unit stage of ResNet.

Adopted ResNet Stage					
	Layer Type	Kernel Size/Stride	Feature Maps	Activation Function	Description
1	Input layer	-	-	-	х
2	Conv 1	$(1 \times 3)/(1 \times 1)$	FM	Relu	y = Conv 1 (x)
3	Dropout	0.5	-	-	y = Dropout(y)
4	Conv 2	$(1 \times 3)/(1 \times 1)$	FM	Relu	y = Conv 2 (y)
5	Dropout	0.5	-	-	y = Dropout (y)
6	Shortcut	-	-	-	$\mathbf{x} = \mathbf{x}$
7	Conv s	$(1 \times 1)/(1 \times 1)$	FM	Relu	x = Conv s (x)
8	Add shortcut	-	-	Relu	$\mathbf{x} = \mathbf{x} + \mathbf{y}$
9	Conv 3	$(1 \times 3)/(1 \times 1)$	FM	Relu	y = Conv 3 (x)

Adopted ResNet Stage					
	Layer Type	Kernel Size/Stride	Feature Maps	Activation Function	Description
10	Dropout	0.5	-	-	y = Dropout (y)
11	Conv 4	$(1 \times 3)/(1 \times 1)$	FM	Relu	y = Conv 4 (y)
12	Dropout	0.5	-	-	y = Dropout (y)
13	Shortcut	-	-	-	x = x
14	Add shortcut	-	-	Relu	$\mathbf{x} = \mathbf{x} + \mathbf{y}$
15	Conv 5	$(1 \times 3)/(1 \times 1)$	FM	Relu	y = Conv 5 (x)
16	Dropout	0.5	-	-	y = Dropout (y)
17	Conv 6	$(1 \times 3)/(1 \times 1)$	FM	Relu	y = Conv 6 (y)
18	Dropout	0.5	-	-	y = Dropout (y)
19	Shortcut	-	-	-	x = x
20	Add shortcut	-	-	Relu	$\mathbf{x} = \mathbf{x} + \mathbf{y}$

Table 5. Cont.

Table 6. Architecture of ResNet.

Adopted ResNet Architecture				
	Layer Type	Kernel Size/Stride	Feature Maps	Output Size
1	Input Layer	-	-	2048
2	Conv 1	$(32)/(1 \times 8)$	64	(256 × 64)
3	Dropout	0.5	-	(256 × 64)
4	Activation	Relu	-	$(256 \times 64)$
5	Max Pool	$(3)/(1 \times 2)$	-	(128 × 64)
6	Stage 1	-	64	(128 × 64)
7	Max Pool	$(3)/(1 \times 2)$	-	$(64 \times 64)$
8	Stage 2	-	128	(64 × 128)
9	Max Pool	$(3)/(1 \times 2)$	-	(32 × 128)
10	Stage 3	-	256	(32 × 256)
11	Dense Layer	300	-	300
12	Dropout	0.5	-	-
13	Output Layer	5	-	5

The number of nodes in the input layer was maintained at the same level as the sampling point of the input signal, which was 2048. The first convolutional layer contained 64 kernels of size  $1 \times 32$ , which remained large because of their performance for ultrasonic and vibration signals. [10,11,28]. The pooling layer reduces the number of layers but does not affect performance [22]. After the data have passed through stage 3, there is one fully connected (dense) layer and an output layer. The number of nodes in the fully connected layer was chosen as the best performance over multiple trials. The number of nodes in the output layer was kept equal to the number of flaws for classification. The activation function used in the layers was rectified linear unit (*Relu*), described by Equation (2)

$$Relu = \begin{cases} (x < 0) f(x) = 0\\ (x \ge 0) f(x) = x \end{cases}$$
(2)

because of its state-of-the-art performance in deep learning [29–31]. The loss function used is sparse categorical cross-entropy, described by Equation (3):

$$-\sum_{i=1}^{M} y_i \log(p_i) \tag{3}$$

where M is the number of data, y is a vector of the true label, and p is the predicted label. Sparse categorical cross-entropy gives the result as integer form, and it is widely used for multiclass classification problems. The activation function applied to the output layer is softmax, described by Equation (4):

$$\sigma(y)_i = \frac{e^{y_i}}{\sum_{k=1}^K e^{y_k}} \tag{4}$$

where *y* is the input vector to the output layer, and *i* indexes the output nodes, i = 1, 2, K. In this study, the number of classes is 5, so *K* is 5. Additionally, dropout normalization was used to avoid overfitting. The network is then trained to minimize cross-entropy loss.

ResNet is trained with optimizer adaptive moment estimation (Adam) for 500 epochs. Adam is an algorithm for the first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The hyper-parameters feature intuitive interpretations and typically require little tuning. The method is suitable for the classification of flaw signals in complex systems since it is appropriate for nonstationary objectives and problems with very noisy and/or sparse gradients [32].

#### 4.4. ResNet Performance Evaluation

The performance of ResNet was then evaluated on the original database (Figure 4). The results show that highly complex datasets with a comparably low amount of data can be a significant penalty when training a neural network. In epoch 461, the training accuracy increased to 92.63%, whereas the testing accuracy reached 70.75% in epoch 241. Here, the meaning of testing accuracy is simply how much data were predicted to be the same as the true label. For a more detailed analysis of the testing accuracy, the average accuracy after epoch 100 was 62.17%, with a standard deviation of 4.13%. This means that training ResNet with the original database can result in a network performance of 62.17%  $\pm$  4.13%.



**Figure 4.** Performance of ResNet trained with original database: (**a**) training accuracy curve; (**b**) testing accuracy curve.

Prior research demonstrates that for simpler databases acquired with KrautKramer transducers, the testing accuracy exceeds 90% [10,11]. This low testing accuracy has two possible explanations. One is that the network is overfitted, and the other is that the

complexity of the database is too high. However, since this network already includes many regularization techniques that are widely known to solve overfitting, it is difficult to regard it as a problem of overfitting alone. Or, at least, it is difficult to solve this problem with only the regularization technique. Therefore, reducing the complexity of the database down can be a possible solution for this problem.

#### 5. System Invariant Method

Because the database was obtained from 21 different transducers, the neural network may be confused by other types of signals because of the unique characteristics of each transducer. There are several approaches to making ultrasonic signals invariant to system or frequency to improve the performance of the neural network training. Polikar et al. created a frequency invariant system to enhance the performance of a neural network for ultrasonic signals by normalizing the frequency [3]. Song et al. developed an intelligent system for ultrasonic flaw classification in weldments [5]. During the process, the system was normalized using a reference signal in the frequency domain.

## 5.1. Principle of System Invariant Method

Weldment flaws signal  $V_f$  include the system, flaw scattering pattern, and specimen as variables. Among these, systems and specimens increase the complexity of the database, which can inhibit performance in classifying neural network flaws, so the database must be refined by extracting only the flaw scattering patterns. The schematic experiment setup is shown in Figure 5.



Figure 5. Schematics of the experimental setup to obtain a flaw signal in the weldment.

Because the ultrasound system is linear time-invariant (LTI),  $V_f$  can be expressed in a time domain with convolution between system variables, as in Equation (5) [33]:

$$V_f(t) = \beta(t) * C_1(t) * T_{12}(t) * C_2(t) * A(t) * C_2(t) * T_{21}(t) * C_1(t)$$
(5)

where  $\beta$  is system factor,  $C_1$  is the diffraction correction in the wedge,  $T_{12}$  is the transmission coefficient of the wedge to the specimen,  $C_2$  is the diffraction correction in the specimen, A is the scattering pattern from the flaw, and  $T_{21}$  is the transmission coefficient of the specimen to the wedge.

To eliminate system and specimen parameters, signals were acquired from a reference specimen using the same system used for the weldment signal acquisition. A reference signal was acquired using a circular reference block to obtain the same ultrasound path for every inspection angle for each transducer–system combination. Here, 21 reference signals



by 21 different types of transducers were obtained using the same ultrasonic pulser-receiver device. The schematic experimental setup for the reference signal is shown in Figure 6.

Figure 6. Schematics of experiment setup to obtain reference signal.

Furthermore, the reference signal  $V_R$  can be described in a time domain as a convolution between system factors, as in Equation (6):

$$V_R(t) = \beta(t) * C_1(t) * T_{12}(t) * C'_2(t) * R(t) * C'_2(t) * T_{21}(t) * C_1(t)$$
(6)

where  $C'_2$  is diffraction correction in the reference specimen and *R* is the reflection coefficient from the round surface. In the frequency domain, the signals  $V_f$  and  $V_R$  can be combined by simple multiplication, not convolution [34,35]. Using fast Fourier transforms (FFT),  $V_f$  and  $V_R$  are described in the frequency domain as Equations (7) and (8):

$$V_f(\omega) = \beta(\omega)C_1(\omega)T_{12}(\omega)C_2(\omega)A(\omega)C_2(\omega)T_{21}(\omega)C_1(\omega)$$
(7)

$$V_R(\omega) = \beta(\omega)C_1(\omega)T_{12}(\omega)C'_2(\omega)R(\omega)C'_2(\omega)T_{21}(\omega)C_1(\omega)$$
(8)

Divide these two terms, flaw scattering pattern A can be extracted as Equation (9):

$$\frac{V_f(\omega)}{V_R(\omega)} = \left(\frac{C_2(\omega)}{C'_2(\omega)}\right)^2 \frac{A(\omega)}{R(\omega)}$$
(9)

The reflection *R* is affected only by the mechanical property of the two mediums and the incidence angle. Here, the round surface of the reference block fixes the incidence angle as normal when the transducer angle varies. Therefore,  $R_i$  is constant, while i represents each transducer with a different angle. Furthermore, the reflection coefficient for steel to air is significantly close to -1 in this case. This means that Equation (9) is described as Equation (10):

$$\frac{V_f(\omega)}{V_R(\omega)} \cong -\left(\frac{C_2(\omega)}{C'_2(\omega)}\right)^2 A(\omega) \tag{10}$$

Diffraction correction  $C(\omega)$  in far-field is represented as Equation (11):

$$C(\omega) = \frac{-ika^2}{2z} \tag{11}$$

where *k* is the wavenumber, *a* is the radius of the transducer and *z* is the flight distance.  $C_2$  and  $C'_2$  feature the same frequency and radius as the transducer, except for distance. Thus, Equation (10) reduces to:

$$\frac{V_f(\omega)}{V_R(\omega)} \cong -\left(\frac{z'}{z}\right)^2 A(\omega) \tag{12}$$

where z' is the distance between the transducer and the flaw of the weldment specimen, while z is the distance between the transducer and the round surface of the reference specimen. With constant  $a = -(z'/z)^2$ ,  $A(\omega)$  can be represented as Equation (13):

$$aA(\omega) \cong \frac{V_f(\omega)}{V_R(\omega)} \tag{13}$$

Applying inverse fast Fourier transform to  $A(\omega)$  can restore the time domain signal of the flaw scattering pattern. When training a neural network, each signal becomes normalized for increasing performance; the constant *a* disappears. The flow diagram of this process is shown in Figure 7.



Figure 7. Flow diagram of the system invariant method.

## 5.2. Applying System Invariant Method to Original Database

To reduce complexity, the system invariant method should be used to reconstruct the ultrasonic weldment signals in the original database. Examples of crack signals before and after using the system invariant method are shown in Figure 8. The system factor was removed from the original signals, leaving only the scattering pattern from flaws. The database that uses the system invariant method is referred to as the invariant database.

#### 5.3. ResNet Performance Evaluation with Invariant Database

Because the original database and the invariant database featured the same number and composition, training the same architecture of ResNet with the original database and invariant database is suitable for verifying the effect of the system invariant method.

The performance of ResNet was then evaluated on the invariant database, as shown in Figure 9. The results show that the system invariant method can reduce database complexity and enhance network performance. The training accuracy increased to 92.15% in epoch 446, which is comparable to the original database, whereas the testing accuracy was higher than 95.04% in epoch 491. For a more detailed analysis of the testing accuracy, the average accuracy after epoch 100 was 91.45%, with a standard deviation is 1.77%. This means that training ResNet with the invariant database can obtain a network performance of 91.45%  $\pm$  1.77%.



**Figure 8.** Examples of cracks and porosity signals when applying system invariant method, from left to right: original signal, flaw scattering pattern in the frequency domain, and invariant signal; (a) crack signal with Olympus A430S (45 degrees, 2 MHz); (b) crack signal with KrautKramer WB70-4 (70 degrees, 4 MHz); (c) porosity signal with Olympus A430S (45 degrees, 2 MHz); (d) porosity signal with KrautKramer WB70-4 (70 degrees, 4 MHz).



**Figure 9.** ResNet performance training with invariant database: (**a**) training accuracy curve; (**b**) testing accuracy curve; (**c**) comparison between accuracy and testing accuracy curves.

A further reason why the invariant method can improve network performance is that the test curve behaves similarly to the training curve, with only a minor shift up and down. This implies that the system invariant method can confuse the network into believing that training and testing data with the same flaw type are similar signals with which to classify.

#### 5.4. Performance Comparison

Table 7 and Figure 10 show a performance comparison. Comparing the training accuracies, the invariant database featured lower average accuracy, but a smaller standard deviation than the original database, making it difficult to determine which is better. It is more rational to consider this as an error that can occur naturally during the training process of a neural network, rather than a performance deviation.

Table 7. ResNet performance comparison between the original and the invariant database.

	Original Database		Invariant Database	
(Epoch > 100)	Training	Testing	Training	Testing
Average Accuracy (%)	92.63	62.17	92.15	91.45
Standard Deviation (%)	0.64	4.13	0.43	1.77



Figure 10. ResNet testing performance comparison between original and invariant database.

Comparing the test accuracies, there was a significant improvement in accuracy for the invariant database. The average accuracy improved by 29.28 pp from 62.17% to 91.45%, making it possible to classify ultrasonic weldment flaw signals more precisely; furthermore, the standard deviation improved by 2.36 pp from 4.13% to 1.77%, making the network performance more stable. Furthermore, the average training and test accuracies of the invariant database were almost similar, suggesting that the system invariant technique can effectively secure the system invariant.

#### 6. Summary

In this study, a system invariant method was developed and tested using a complex database obtained from 21 different transducers. The flaw signal database contained 65,790 training data and 1453 test data, referred to as the original database. By developing a system invariant method in the frequency domain and using it in the original database, an invariant database was constructed. Consequently, comparing performance using a residual neural network, the invariant database showed a significant improvement in accuracy compared to original database. The proposed system invariant method showed meaningful performance improvements against complex ultrasonic equipment. In future studies, practical research is needed in order to overcome difficulties that may arise when the system invariant technique is applied to the actual ultrasonic testing site.

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

**Funding:** This research was supported by the Nuclear Safety Research Program of the Korea Foundation of Nuclear Safety (KoFONS), funded by the Nuclear Safety and Security Commission (No. 1805005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

## Abbreviations

WGN	White Gaussian noise
ANN	Artificial neural network
FCNN	Fully connected neural network
CNN	Convolutional neural network
ResNet	Residual neural network

## References

- Song, S.J.; Schmerr, L.W. Ultrasonic flaw classification in weldments using probabilistic neural networks. *J. Nondestruct. Eval.* 1992, 11, 69–77. [CrossRef]
- Masnata, A.; Sunseri, M. Neural network classification of flaws detected by ultrasonic means. NDT E Int. 1996, 29, 87–93. [CrossRef]
- 3. Polikar, R.; Udpa, L.; Udpa, S.S.; Taylor, T. Frequency invariant classification of ultrasonic weld inspection signals. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **1998**, 45, 614–625. [CrossRef] [PubMed]
- 4. Margrave, F.W.; Rigas, K.; Bradley, D.A.; Barrowcliffe, P. The use of neural networks in ultrasonic flaw detection. *Measurement* **1999**, 25, 143–154. [CrossRef]
- Song, S.J.; Kim, H.J.; Cho, H. Development of an intelligent system for ultrasonic flaw classification in weldments. *Nucl. Eng. Des.* 2002, 212, 307–320. [CrossRef]
- 6. Simas Filho, E.F.; Silva, M.M.; Farias, P.C.; Albuquerque, M.C.; Silva, I.C.; Farias, C.T. Flexible decision support system for ultrasound evaluation of fiber–metal laminates implemented in a DSP. *NDT E Int.* **2016**, *79*, 38–45. [CrossRef]
- 7. Cruz, F.C.; Simas Filho, E.F.; Albuquerque, M.C.; Silva, I.C.; Farias, C.T.; Gouvêa, L.L. Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing. *Ultrasonics* **2017**, *73*, 1–8. [CrossRef]
- 8. Meng, M.; Chua, Y.J.; Wouterson, E.; Ong, C.P.K. Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks. *Neurocomputing* **2017**, 257, 128–135. [CrossRef]
- 9. Munir, N.; Kim, H.J.; Song, S.J.; Kang, S.S. Investigation of deep neural network with drop out for ultrasonic flaw classification in weldments. *J. Mech. Sci. Technol.* 2018, 32, 3073–3080. [CrossRef]
- 10. Munir, N.; Kim, H.J.; Park, J.; Song, S.J.; Kang, S.S. Convolutional neural network for ultrasonic weldment flaw classification in noisy conditions. *Ultrasonics* 2019, 94, 74–81. [CrossRef]
- 11. Munir, N.; Park, J.; Kim, H.J.; Song, S.J.; Kang, S.S. Performance enhancement of convolutional neural network for ultrasonic flaw classification by adopting autoencoder. *NDT E Int.* **2020**, *111*, 102218. [CrossRef]
- 12. Rimoldi, B. Principles of Digital Communication: A Top-Down Approach; Cambridge University Press: Cambridge, UK, 2016.
- 13. Gallager, R.G. *Principles of Digital Communication;* Cambridge University Press: Cambridge, UK, 2008.
- 14. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. *Learning Internal Representations by Error Propagation*; California University San Diego La Jolla Inst for Cognitive Science (ICS-8506): San Diego, CA, USA, 1985.
- 15. Rosenblatt, F. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms; Spartan Books: Washington, DC, USA, 1961.
- 16. Géron, A. Hands on Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems; O'Reilly Media Inc.: Newton, MA, USA, 2017.
- 17. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press: Cambridge, MA, USA, 2016.
- 18. Nielsen, M.A. Neural Networks and Deep Learning; Determination Press: San Francisco, CA, USA, 2015.
- 19. Li, F.F.; Karpathy, A.; Johnson, J.; Yeung, S. *CS231n: Convolutional Neural Networks for Visual Recognition*; Stanford University: Stanford, CA, USA, 2016.
- Lecun, Y.; Bottou, L.; Benjio, Y.; Haffner, P. Gradient-based Learning Applied to Document Recognition. *Proc. IEEE* 1998, 86, 2278–2324. [CrossRef]
- 21. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. Nature 2015, 521, 436–444. [CrossRef] [PubMed]
- 22. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv* 2014, arXiv:1412.6806.
- He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images; Technical Report No. TR-2009; Computer Science Department, University of Toronto: Toronto, ON, Canada, 2009; Available online: https://www.cs.toronto.edu/~{}kriz/learningfeatures-2009-TR.pdf (accessed on 11 January 2022).
- 25. Wu, S.; Zhong, S.; Liu, Y. Deep residual learning for image steganalysis. Multimed. Tool. Appl. 2018, 77, 10437–10453. [CrossRef]
- 26. Shibata, N.; Tanito, M.; Mitsuhashi, K.; Fujino, Y.; Matsuura, M.; Murata, H.; Asaoka, R. Development of a deep residual learning algorithm to screen for glaucoma from fundus photography. *Sci. Rep.* **2018**, *8*, 14665. [CrossRef]
- 27. Cheng, X.; Zhang, Y.; Chen, Y.; Wu, Y.; Yue, Y. Pest identification via deep residual learning in complex background. *Comput. Electron. Agric.* 2017, 141, 351–356. [CrossRef]

- 28. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [CrossRef]
- Nair, V.; Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010.
- 30. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. Proc. icml 2013, 30, 3.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 2017, 60, 84–90. [CrossRef]
- 32. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. arXiv 2014, arXiv:1412.6980.
- 33. Schmerr, L.W. Fundamentals of Ultrasonic Nondestructive Evaluation; Springer: New York, NY, USA, 2016; pp. 650–681.
- 34. McGillem, C.D.; Cooper, G.R. *Continuous and Discrete Signal and System Analysis*, 2nd ed.; Holt, Rinehart & Winston: New York, NY, USA, 1984; p. 118. ISBN 0030617030.
- 35. Weisstein, E.W. Convolution Theorem. From MathWorld—A Wolfram Web Resource. Available online: https://mathworld. wolfram.com/ConvolutionTheorem.html (accessed on 11 January 2022).