



A deep transfer learning model for inclusion defect detection of aeronautics composite materials

Yanfeng Gong^a, Hongliang Shao^b, Jun Luo^{a,*}, Zhixue Li^a

^a Key Laboratory of Optoelectronic Technology and System of the Ministry of Education, Chongqing University, Chongqing 400044, China

^b Shanghai Satellite Equipment Research Institute, Shanghai 200240, China

ARTICLE INFO

Keywords:

Inclusion defect detection
Aeronautics composite materials
Transfer learning
Feature extraction

ABSTRACT

Composite materials are increasingly used as structural components in military and civilian aircraft. To ensure their high reliability, numerous non-destructive testing (NDT) techniques have been used to detect defects during production and maintenance. However, most of these techniques are non-automatic, with diagnostic results determined subjectively by operators. Some deep learning methods have been proposed to identify defects in images obtained through NDT, but they need labeled image samples with defects, which can be expensive or unavailable. We propose a deep transfer learning model to accurately extract features for the inclusion of defects in X-ray images of aeronautics composite materials (ACM), whose samples are scarce. We researched an automatic inclusion defect detection method for X-ray images of ACM using our proposed model. Experimental results show that the model can reach 96% classification accuracy (F1_measure) with satisfactory detection results.

1. Introduction

Composites are composed of multiple base materials, each maintaining its own structure and properties rather than forming a combined alloy [1]. These materials, such as glass fiber reinforced polymers (GFRP) and carbon fiber reinforced polymers (CFRP), are increasingly used as structural components in high-performance military aircraft because of their high strength, low weight, low coefficient of thermal expansion, high fatigue resistance, inherent corrosion resistance, and low electromagnetic reflectance [2,3]. However, defects can occur during their manufacture, including micro-cracks, fiber breakage, voids, porosity, delamination, and inclusions [3]. These defects can lead to serious accidents when used in products. Thus the detection of defects is required for their high quality and reliability. There are various methods to evaluate composite materials, including ultrasonic, thermographic, infrared thermography, radiographic, visual detection, acoustic emission, acousto-ultrasonic, shearography, optical, electromagnetic, liquid penetrant, and magnetic particle testing [1]. However, no technique can accurately and effectively identify all types of defects. For instance, terahertz time-domain spectroscopy has been researched for nondestructive evaluation of aircraft composites [4–7]. It is usually used to identify surface defects and some internal defects, such as voids, delamination, punctures, mechanical

damage, and heat damage [2]. X-ray computed tomography is mainly utilized to evaluate impact damage such as matrix cracking and delamination [8–11], and internal defects such as porosity, voids, and micro-cracking [12–14] in composite structures. This paper focuses on image processing of slag inclusion defect detection in CFRP. Slag inclusion usually occurs due to granular slags that are accidentally included in parent material during manufacture. Such defects may easily form sites of stress concentration and cause more serious defects such as cracks, delamination, and disbands, which may severely degrade the mechanical properties of composites. NDT techniques that are commonly applied for slag inclusion defect detection include ultrasonic [19], infrared thermography [15,16], laser shearography [17], radiography [18,19], microwave [20], and terahertz [6,7]. The X-ray testing includes two ways: digital radiography (DR) and computed tomography (CT). The X-ray DR testing and ultrasound methods have attracted considerable interest [18] in inclusion defects detection. The ultrasound method has good processing speed and strong penetrating power, but lateral and axial spatial resolution are limited, and liquid coupling is required, which may contaminate the samples [21]. Although the X-ray DR testing method has ionizing radiation that is harmful to human body, it can obtain images of objects with high resolution in real-time [18,22]. Despite the high accuracy of defect detection in some of the above NDT methods, they are all manual, with

* Corresponding author.

E-mail address: luojun@cqu.edu.cn (J. Luo).

diagnostic results determined subjectively by operators. Thus the results are affected by the experience and knowledge of experts. Detection errors can greatly increase, especially after the operator works long hours. Moreover, the detection process is time-consuming and inefficient. We choose the X-ray DR testing as the way to acquire image samples in our proposed method.

To reduce the effects of operator subjectivity and improve the efficiency of defect detection, methods based on modern computer vision, such as deep learning, are utilized to detect defects in composites. Liang et al. [23] designed an artificial neural network (ANN) with two hidden layers to identify holes and cracks in composite structures. ANNs were used for structural damage detection [24,25] and have been proved efficient and satisfactory when the volume of measurement data is enormous. Anderson and Lemoine [26] utilized an ANN to construct an electric potential-based damage detection technique that is applicable to electrically conductive composite structures and electroactive membranes. Ye et al. [27] proposed ANN-based damage identification in carbon fiber-reinforced composite laminates using the hierarchical development of a training database to mitigate the impact of the large amounts of data required. However, a detection system using only an ANN can easily fall into a local optimum and has low classification accuracy. Thus Lee et al. [28] proposed a method combining ANN with SVM for defect diagnostics of a gas turbine engine. A genetic algorithm was adopted for global optimization to prevent falling into local optima when combined with neural networks for damage detection [29,30]. Recently, with the development of deep learning, more artificial intelligence models, such as deep neural networks [31–35], multi-layer perceptron [36], and recurrent neural networks [37], have been adopted for damage detection.

Through the above literature review, it can be seen that artificial intelligence models satisfy the condition that training data are not scarce. Moreover, most of the data are simulated, not real, which can lead to failure in application because of different data distributions. Also, when the training dataset is small, the trained model will have poor generalization. Small-scale data cannot represent the distribution to which they belong, and the training and testing data must have the same distribution. However, large-scale training samples are not available or are expensive to collect in some scenarios, such as aerospace, where the production process is strictly controlled, and manufactured parts must be guaranteed to be defect-free before assembly.

To complete the automatic defect detection of ACM, including a diagnostic result from the acquired images, we propose a deep transfer learning model based on X-ray NDT. We research a defect detection method using the proposed model, which is applicable to scarce training data. The framework is shown in Fig. 1. Firstly, we collect a batch of image samples of ACM using X-ray NDT for use as the target domain according to the requirements of transfer learning, while the source domain adopts another dataset that is different but similar. Then, deep transfer learning neural networks are constructed and trained to obtain an accurate and efficient inclusion identification model. Eventually, this well-trained model is applied to detect inclusion defects in images

obtained by X-ray NDT. The sliding-window approach is adopted, combining with a heat map during detection.

The main contributions of this paper are as follows.

- 1) A deep transfer learning model is proposed. The proposed neural network model consists of four modules: feature extractor, label classifier, domain classifier, and distance metrics. The model trained through these four modules enables the feature extractor to obtain domain-invariant features and ensures that the label classifier has good inclusion defect detection performance.
- 2) A new method to detect inclusion defects of ACM is explored based on X-ray NDT. Conventional machine vision-based defect detection methods for composite materials mostly depend on large-scale training data or simulated data. The proposed method can be directly applied to inclusion defect detection for original X-ray images of ACM, highly accurately and in real time, even under the condition of scarce image samples.
- 3) This is the first application of deep transfer learning to automatically detect the defects of ACM, whose X-ray image samples are scarce.

2. Proposed method

2.1. Acquisition of sample images

In accordance with the purpose of this work, X-ray NDT was adopted to acquire ACM images to train the proposed deep transfer learning model. The radiation source was the MXR-225 tube manufactured by Comet AG. The tube voltage and current intensity were 65 kV and 3 mA, respectively. The X-ray real-time imaging system was the XYD-225, as shown in Fig. 2. As supervised transfer learning, the proposed model only needs a small number of labeled ACM image samples, in which each category has only dozens of samples. We collected a batch of ACM image samples from the Shanghai Satellite Equipment Research Institute, including defective images with inclusion and non-inclusion images, which were acquired using the above equipment. There were 100 images with defects, and some are shown in Fig. 3, where Fig. 3(a) shows the original ACM whose width is 40 mm, and Fig. 3(b)–(d) show some corresponding X-ray images containing slag inclusion defects. There are several slag inclusion defects in each X-ray image, which are circled using red square frames. The size of these defects are mostly about 0.5 mm in diameter. As the density of slags is bigger than that of ACM, the grey level of corresponding images is higher than that of surrounding regions in the conventional imaging mode. While we adopted the reverse mode, the images of slags become dark dots whose contrast is lower than that of background, and the more the X-rays absorbed, the darker the resulting image in the panel. Because the slag is granular, its image appears as a spot. For each image, we cropped the patches to 32×32 pixels containing defects as the target domain sample set. There were 200 patch images, labeled as “1” (inclusion). To balance different categories, we

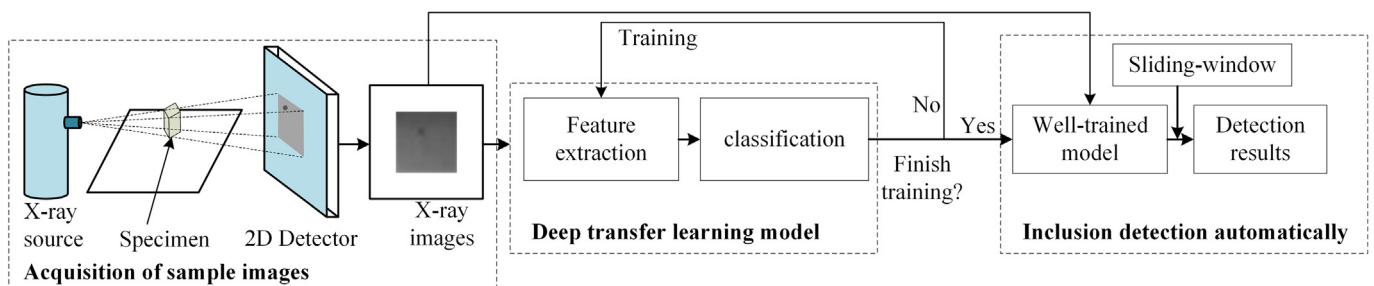


Fig. 1. Overall framework of the proposed method.

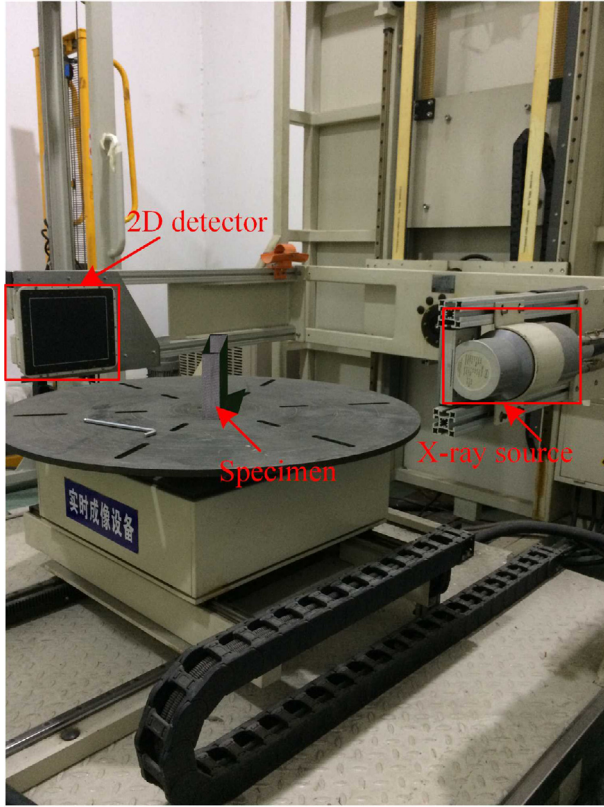


Fig. 2. Equipment used for imaging.

cropped another 200 patch images without defects and labeled them as “0” (non-inclusion). Some patch images are shown in Fig. 4. For the source domain dataset, we used the patch images of the group

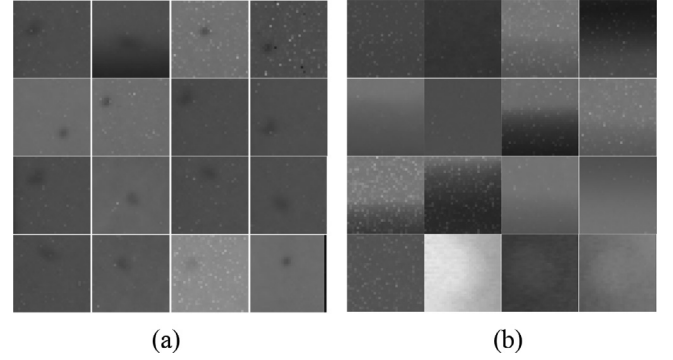


Fig. 4. Some patch image samples in the target domain: (a) Images with inclusions; (b) Images without inclusions.

“welds” from GDXray [41], which is a public database taken by the BAM Federal Institute for Materials Research and Testing, Berlin, Germany. As shown in Fig. 5, the inclusion defects in the images of the group “welds” had characteristics similar to those of the collected ACM images, but they had different marginal probability distributions, as validated in later experiments. Thus it is our opinion that these two domains constitute a source and target domain pair, according to the theory of transfer learning. The group “welds” contains 88 entire X-ray images. Similar to the way we obtained the target domain dataset, we cropped the patches to size 32×32 pixels containing defects from the group “welds” to construct the source domain dataset, which contained 400 patch images with defects. We used data expansion [31] to reinforce the source domain dataset, which eventually contained 1200 patch images with defects. The dataset is shown in Table 2; 2300 patch images in the source domain dataset and 40 patch images in the target domain dataset were used for training, during which 100 patch images in each of these datasets were used for validation. Since our task is to detect inclusion defects in the target domain, only samples in the tar-

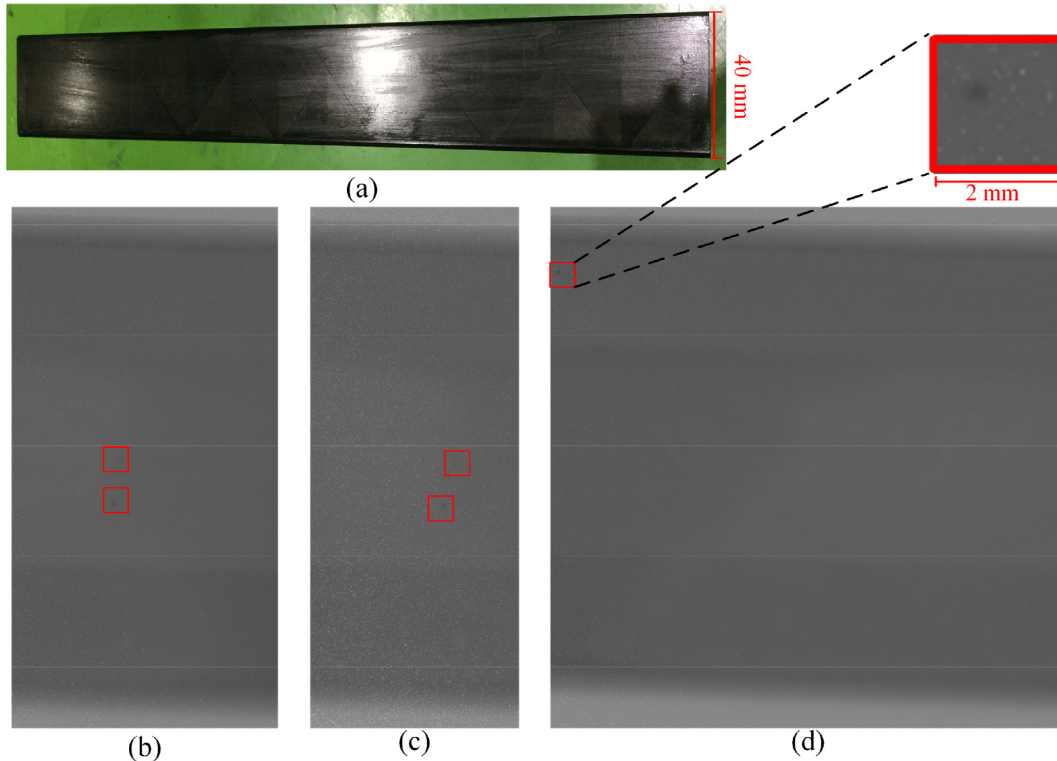


Fig. 3. Several X-ray image samples of ACM with inclusion defects.

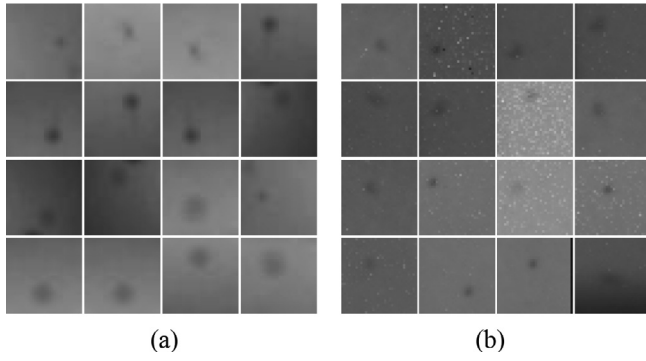


Fig. 5. Comparison of image samples in source domain and target domain: (a) Some patch images in the group “welds” from GDxray; (b) Some patch images of ACM.

get domain were used to test the performance of the proposed model. There were 260 test patch images, consisting of 130 in each category.

2.2. Transfer learning

Features can be fetched hierarchically from the lower layer to the higher layer in a typical deep convolutional neural network (DCNN) [38]. Once we have a well-trained DCNN model, the semantic representation of an image can be obtained. However, an accurate conventional DCNN model needs large numbers of samples for training, and the testing and training images must have the same probability distribution [40,40]. In aerospace, it is difficult to obtain so many training samples of composite materials with inclusion defects, but transfer learning of DCNN can meet the challenge. Through transfer learning, knowledge learned in one domain can be applied to a different but related domain, and we call these the source and target domain, respectively. We introduce some notation to clearly describe the problem. D_t is the target domain, and χ_t is the corresponding feature space. The marginal probability distribution is $P(X_t)$, where $X_t = \{x_{t1}, \dots, x_{tm}\} \in \chi_t$, and x_{ti} is the i th instance [39]. The target domain is $D_t = \{\chi_t, P(X_t)\}$, in which the training and testing data consist of pairs $\{x_{ti}, y_{ti}\}$, where $x_{ti} \in X_t$ and $y_{ti} \in \gamma_t$ (γ_t is the label space of the target domain). We denote the source domain by the subscript s ; hence, the domain is $D_s = \{\chi_s, P(X_s)\}$, with data pairs $\{x_{si}, y_{si}\}$. The training and testing data are from the same domain in traditional DCNN, as shown in Fig. 6(a). Transfer deep learning allows the training and testing data to come from different but related domains [40,40], as shown in Fig. 6(b). Hence, DCNN with transfer learning can learn domain-invariant features.

Collecting large amounts of X-ray images with inclusion defects from aeronautic composite material parts is difficult and expensive because the parts are of high quality. Small-scale sample data can easily cause overfitting, thus traditional DCNN is unsuitable. Data from another different but related domain are used to assist in training those scarce data with transfer learning. We use labeled sample data from aeronautic composite materials as the target domain, and labeled sample data from the welding database [41] as the source domain. These two domains have different marginal probability distributions ($P(X_s) \neq P(X_t)$) [39]. However, defective samples in these two domains have similar characteristics; hence, they have the same feature space ($\chi_s = \chi_t$). As their categories are the same, they have the same label space ($\gamma_s = \gamma_t$). The objective of training the DCNN with transfer learning is a model that can obtain domain-invariant features when inputting different data from these two domains. This means that with the well-trained model, the same label will be output when two defective samples are input, one from ACM and the other from the welding dataset. The key issue is how to construct a transfer learning model to

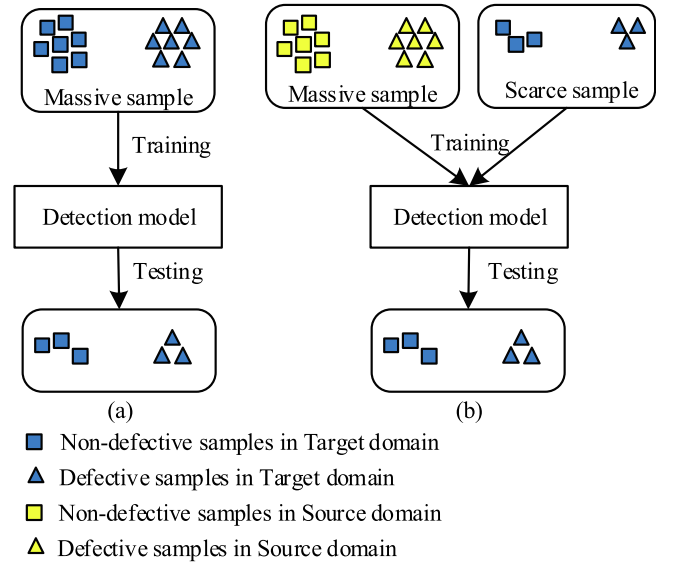


Fig. 6. Deep learning for defect detection: (a) Traditional DCNN; (b) DCNN with transfer learning.

learn domain-invariant features to recognize the inclusion defects in aviation composite materials.

2.3. Deep transfer learning model

As shown in Fig. 7, the proposed transfer learning model consists of four modules: feature extractor, label classifier, domain classifier, and distance metrics. The feature extractor and label classifier constitute the label classification, while the feature extractor and domain classifier constitute the domain classification. Thus the label and domain classifications share the feature extractor. The label classification can fetch features from input image samples and try to classify them into correct class labels. The domain classification tries to achieve similar feature distributions between the source and target domains. The distance metrics include two specific distances: 1) the distance between samples from different domains but the same labels; and 2) the distance between samples from different domains and labels. The label classification, domain classification, and distance metrics jointly complete domain adaptation, through which domain-invariant features can be fetched when images from different domains are fed into the feature extractor. The structure of the proposed model can be seen in Fig. 7, where C1, C2, and C3 denote the three 2-D convolutional layers of the feature extractor; F1 and F2 denote the fully connected layers of the feature extractor; and FL and FD denote the fully connected layers of the label classifier and domain classifier, respectively. As shown in Table 1, the parameters of the entire network model are small in number (130,144), which ensures the real-time performance of the proposed model.

2.3.1 Label classification

Label classification was achieved by a convolutional neural network (CNN) with C1, C2, C3, F1, F2, and FL, among which the three convolutional layers C1, C2, C3 and the first two fully connected layers F1 and F2 were used for the feature extractor, and the last fully connected layer (FL) for the label classifier.

During the training process, the input of C1 was in the form of image pairs (x_s, x_t) , where x_s was from the source domain, and x_t from the target domain. In forward propagation, corresponding features could be learned by applying the convolutional kernel on the inputs through a convolution operation in each convolutional layer, each con-

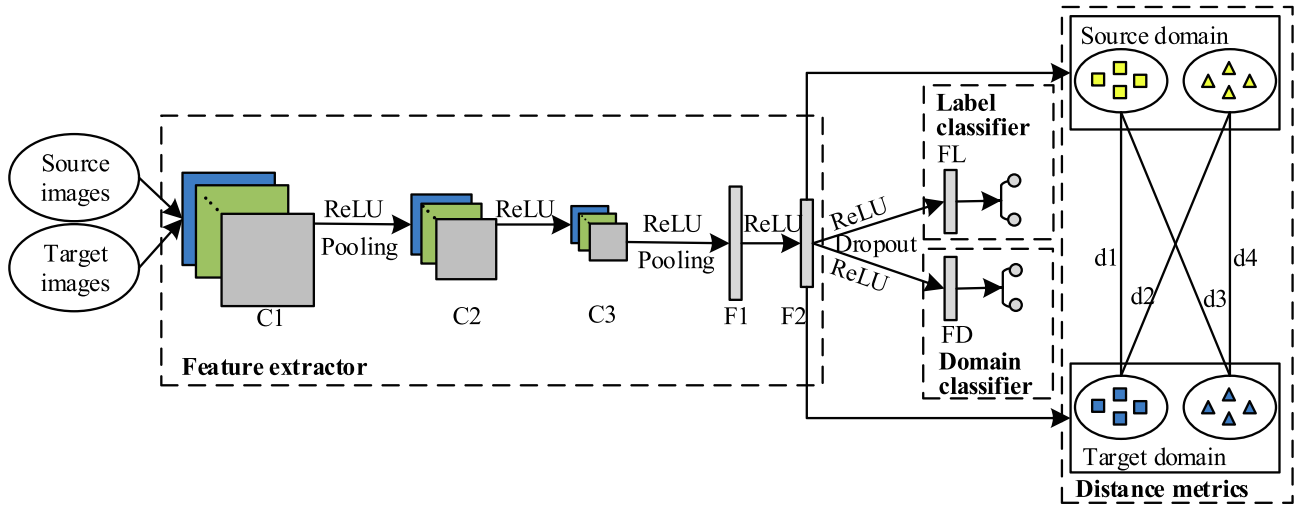


Fig. 7. Architecture of the proposed transfer learning model.

Table 1

Symbolic representation and corresponding parameters in model.

| Symbol | Operation | Parameters |
|--------|-----------------|---|
| C1 | 2-D Convolution | $32 \times 3 \times 3 \times 3 = 864$ |
| C2 | 2-D Convolution | $32 \times 32 \times 3 \times 3 = 9216$ |
| C3 | 2-D Convolution | $32 \times 32 \times 3 \times 3 = 9216$ |
| F1 | Linear | $128 \times 800 = 102400$ |
| F2 | Linear | $64 \times 128 = 8192$ |
| FL | Linear | $2 \times 64 = 128$ |
| FD | Linear | $2 \times 64 = 128$ |

connected with an activation function and pooling layer, except for C2. The formulation is

$$f_j^c = \text{MaxPool2d} \left(\text{ReLU} \left(\sum_{i=1}^n w_i * f_{j-1}^i + b_c \right) \right), \quad (1)$$

where f_j^c is the c th channel of feature map f_j of the j th convolutional layer; f_{j-1}^i is the i th channel of feature map f_{j-1} in the $(j-1)$ th convolutional layer; $*$ is a 2-D convolutional operator; w_i is the i th convolutional kernel corresponding to f_{j-1}^i ; b_c is the corresponding bias vector; $\text{ReLU}(\cdot)$ is an activation function; and $\text{Maxpool2d}(\cdot)$ is a maximize pooling function.

Following the last convolutional layer C3, two fully connected layers F1 and F2 were utilized to learn more specific semantic features. Then a single fully connected layer FL and a softmax function were used for the label classifier, formulated as

$$f = \text{softmax}(w_l^T \cdot f_l + b_l) \quad (2)$$

$$\text{softmax}(z) = \frac{1}{1 + e^{-z}}, \quad (3)$$

where f_l is the feature vector of FL, w_l is the weight matrix corresponding to f_l , and b_l is the corresponding bias vector.

Table 2

Composition of image dataset in source domain and target domain.

| | Source domain dataset | | Target domain dataset | | |
|---------------|-----------------------|------------|-----------------------|------------|---------|
| | Training | Validation | Training | Validation | Testing |
| Inclusion | 1150 | 50 | 20 | 50 | 130 |
| Non-inclusion | 1150 | 50 | 20 | 50 | 130 |
| Total | 2300 | 100 | 40 | 100 | 260 |

2.3.2 Domain classification

A domain adaptation technique was used to make the feature extractor learn domain-invariant features when inputting data of two different domains. An adversarial network was constructed, consisting of a feature extractor and domain classifier. The feature extractor tries to make the feature distributions of two domains as similar as possible, and the domain classifier tries to correctly classify them into two classes, those being the source domain and target domain.

Domain classification uses the same feature extractor as label classification. Therefore, the formulas of the feature extractor in domain classification and label classification are the same. The domain classifier just contains one fully connected layer FD, whose input consists of the results of F2 processed by ReLU and dropout. A softmax function was adopted for the last domain classification result:

$$d = \text{softmax}(w_d^T \cdot f_d + b_d), \quad (4)$$

where f_d is the feature vector of FD, w_d is the weight matrix corresponding to f_d , and b_d is the corresponding bias vector.

2.3.3 Distance metrics

The above domain classification can only guarantee that the extracted feature distributions of two different domains are similar. Distance metrics were adopted to make the extracted features more class distinguishable. The distance metrics try to minimize the distance of samples from different domains but with the same label, such as d1 and d4 in Fig. 7, and simultaneously to maximize the distance of samples from different domains and labels, such as d2 and d3 in Fig. 7. This strategy can align the semantic features from different domains to further improve the performance of domain adaptation. The distance of the feature pairs from the feature extractor is calculated as

$$d(f_{F2}^s, f_{F2}^t) = \begin{cases} \frac{1}{2} \|f_{F2}^s - f_{F2}^t\|^2, & \text{if } y_s = y_t \\ \frac{1}{2} \max(0, 1 - \|f_{F2}^s - f_{F2}^t\|^2), & \text{if } y_s \neq y_t \end{cases}, \quad (5)$$

where f_{F2}^s is the feature vector of F2 in the source domain, f_{F2}^t is the feature vector of F2 in the target domain, and y_s and y_t are the corresponding class labels of f_{F2}^s and f_{F2}^t , respectively.

2.4. Training process

To learn the features that have both label-discrimination and domain-invariance, we jointly optimized the label classifier and domain classifier [42] during training. There are three objective loss functions for optimization: label classification loss, domain classification loss, and distance metrics loss. As the label classification includes a feature extractor and label classifier, we used θ_f and θ_l , respectively, as the parameters of the feature extractor and label classifier, the optimal parameters being θ_f^* and θ_l^* , respectively. Similarly, for domain classification, we used θ_d as the parameters of the domain classifier, with the optimal parameter being θ_d^* . The feature extractor is denoted as $G_f(x; \theta_f)$, the label classifier as $G_l(x; \theta_l)$, and the domain classifier as $G_d(x; \theta_d)$. The cross-entropy was used to measure the loss of these three objective functions.

2.4.1 Label classification loss

The purpose of label classification is to predict, as accurately as possible, the labels of input images in the source domain. Therefore, the label classification loss should be minimized. Defect detection in aeronautic composite materials is a kind of binary classification problem, whose detection results are either inclusion defects or non-inclusion defects. The label classification loss is formulated as

$$L_l(\theta_f^*, \theta_l^*) = \min_{\theta_f, \theta_l} \{ [E_{x \sim P(x_s)} [-y \cdot \log G_l(G_f(x; \theta_f); \theta_l)]] \}. \quad (6)$$

2.4.2 Domain classification loss

The purpose of domain classification is to make the distribution of these two domains as similar as possible. This can be completed using a strategy of adversarial networks [43]. Thus the domain classification is optimized by maximizing the loss of the domain classifier to seek the optimal parameters θ_f of the feature extractor, and simultaneously minimizing the loss of the domain classifier to seek the optimal parameters θ_d [42]. The formulation is

$$L_d(\theta_f^*, \theta_d^*) = \max_{\theta_f} \min_{\theta_d} \{ E_{x \sim P(x_s)} [-\log G_d(G_f(x; \theta_f); \theta_l)] + E_{x \sim P(x_t)} [-\log(1 - G_d(G_f(x; \theta_f); \theta_d))] \}. \quad (7)$$

2.4.3 Distance metrics loss

The distance metrics strategy includes minimizing the distance of features from different domains but with the same labels, and maximizing the distance of features from different labels. Euclidean distance was adopted to compute these metrics. Specifically, we completed the distance metrics by optimizing the loss function

$$D(\theta_f^*) = \min_{\theta_f} D(x_s, x_t) = \begin{cases} \frac{1}{2} \| G_f(x_s; \theta_f) - G_f(x_t; \theta_f) \|^2, & \text{if } y_s = y_t \\ \frac{1}{2} \max(0, 1 - \| G_f(x_s; \theta_f) - G_f(x_t; \theta_f) \|^2), & \text{if } y_s \neq y_t \end{cases} \quad (8)$$

where y_s and y_t are the labels of input samples x_s and x_t , respectively, where $\| \cdot \|$ denotes the 2-norm.

2.4.4 Total loss

According to the above analysis, θ_f^* and θ_d^* can be solved for directly by minimizing the label classification loss L_l and domain classification loss L_d . To obtain θ_f^* , L_d should be maximized, and L_l and the distance metrics loss D should simultaneously be minimized.

To jointly optimize the above three objectives instead of alternately training them, we represented the total loss as

$$L(\theta_f, \theta_l, \theta_d) = \lambda_1 \cdot L_l(\theta_f, \theta_l) - \lambda_2 \cdot L_d(\theta_f, \theta_d) + \lambda_3 \cdot D(\theta_f), \quad (9)$$

where λ_1 , λ_2 , and λ_3 are the meta-parameters used to control the trade-off among the three kinds of loss.

The optimal parameters θ_f^* , θ_l^* , and θ_d^* are then

$$\theta_f^*, \theta_l^* = \arg \min_{\theta_f, \theta_l} L(\theta_f, \theta_l, \theta_d) \quad (10)$$

$$\theta_d^* = \arg \max_{\theta_d} L(\theta_f, \theta_l, \theta_d) = \arg \min_{\theta_d} L_d(\theta_f, \theta_d). \quad (11)$$

We adopted stochastic gradient descent (SGD), as commonly used in deep learning, to search the optimal points. However, because of the $-\lambda_2$ before $L_d(\cdot)$ in the formula (9), SGD cannot be directly used; thus the strategy of reversing the gradient was adopted in the back-propagation of the domain classification [42]. Specifically, a gradient reversal layer (GRL) was added between the feature extractor and domain classifier. GRL works only in backpropagation during optimization. Accordingly, formula (9) can be expressed as

$$\tilde{L}(\theta_f, \theta_l, \theta_d) = \lambda_1 \cdot L_l(\theta_f, \theta_l) + \lambda_2 \cdot \tilde{L}_d(\theta_f, \theta_d) + \lambda_3 \cdot D(\theta_f) \quad (12)$$

where $\tilde{L}_d(\theta_f, \theta_d)$ denotes the application of GRL to $L_d(\theta_f, \theta_d)$, with the result that $\partial L_d / \partial \theta_f$ is multiplied by $-\lambda_2$ in backpropagation and remains an identity transform in forward propagation. Therefore, the corresponding gradients of these parameters can be expressed as

$$\frac{\partial \tilde{L}}{\partial \theta_f} = \lambda_1 \cdot \frac{\partial L_l}{\partial \theta_f} + \left(-\lambda_2 \cdot \frac{\partial \tilde{L}_d}{\partial \theta_f} \right) + \lambda_3 \cdot \frac{\partial D}{\partial \theta_f} \quad (13)$$

$$\frac{\partial \tilde{L}}{\partial \theta_l} = \lambda_1 \cdot \frac{\partial L_l}{\partial \theta_l} \quad (14)$$

$$\frac{\partial \tilde{L}}{\partial \theta_d} = \lambda_3 \cdot \frac{\partial \tilde{L}_d}{\partial \theta_d}. \quad (15)$$

Accordingly, θ_f , θ_l , and θ_d can be updated by the following expressions:

$$\theta_f \leftarrow \theta_f - \varepsilon \cdot \frac{\partial \tilde{L}}{\partial \theta_f} \quad (16)$$

$$\theta_l \leftarrow \theta_l - \varepsilon \cdot \frac{\partial \tilde{L}}{\partial \theta_l} \quad (17)$$

$$\theta_d \leftarrow \theta_d - \varepsilon \cdot \frac{\partial \tilde{L}}{\partial \theta_d}, \quad (18)$$

where ε is the learning rate, which varies during the iteration.

Once the model is trained, it can extract domain-invariant features, and the corresponding labels of samples in the target domain can be accurately predicted. The model can then be used to inspect the X-ray images of ACM.

3. Experiment validations

3.1. Experiment setup

We evaluated and compared the performance of the proposed method with several other common methods using the above datasets. We trained the model on the sample data from the source and target domains and tested the model on the target domain sample data. We detected an entire X-ray ACM image using the sliding-window method and recognized the defects through the well-trained transfer learning model. All experiments were conducted in the PyTorch framework on a computer with 16 GB RAM, six Intel i7 processors, a GTX 1080ti GPU, and a 64-bit Ubuntu-16.04 operating system, using Python 3.5.

Table 3

Detection results of various common models, where the bold means the best result in each column.

| Method | Accuracy | Precision | Recall | F1_measure | Transfer method |
|-----------------------------|--------------|--------------|--------------|-------------|------------------|
| Just training source domain | 0.819 | 0.721 | 0.909 | 0.804 | None |
| DANN [42] | 0.931 | 0.932 | 0.896 | 0.914 | Adversarial |
| CCSA [4] | 0.92 | 0.984 | 0.818 | 0.893 | MMD |
| DCTLN [44] | 0.957 | 0.96 | 0.935 | 0.947 | Multi-adaptation |
| Ours | 0.968 | 0.973 | 0.948 | 0.96 | Multi-adaptation |

While training the model, we randomly selected one patch from the source domain dataset and another from the target domain dataset to comprise an image pair. As there were 2400 source images and 40 target images for training, there were 96,000 image pairs ($2400 \times 40 = 96,000$). The image pairs were divided into several batches to be fed into the network for training, each batch consisting of 256 image pairs. We gradually changed the adaptation factor λ using the following formulation [42]:

$$\lambda = \frac{2}{1 + \exp(-10 \cdot p)} - 1 \quad (19)$$

$$p = \frac{b_i + e_i \cdot b_t}{e_t \cdot b_t}, \quad (20)$$

where b_i is the i th batch, e_i is the i th epoch, b_t is the number of batches in one epoch, and e_t is the total number of iteration epochs, which was set to 100.

For all layers in the network, we trained the feature extractor, label classifier, domain classifier, and distance metrics from scratch, using stochastic gradient descent (SGD) with an initial learning rate of 0.01 and momentum of 0.9.

Four other methods were tested on the same dataset for comparison with the proposed model. The results are shown in Table 3, where

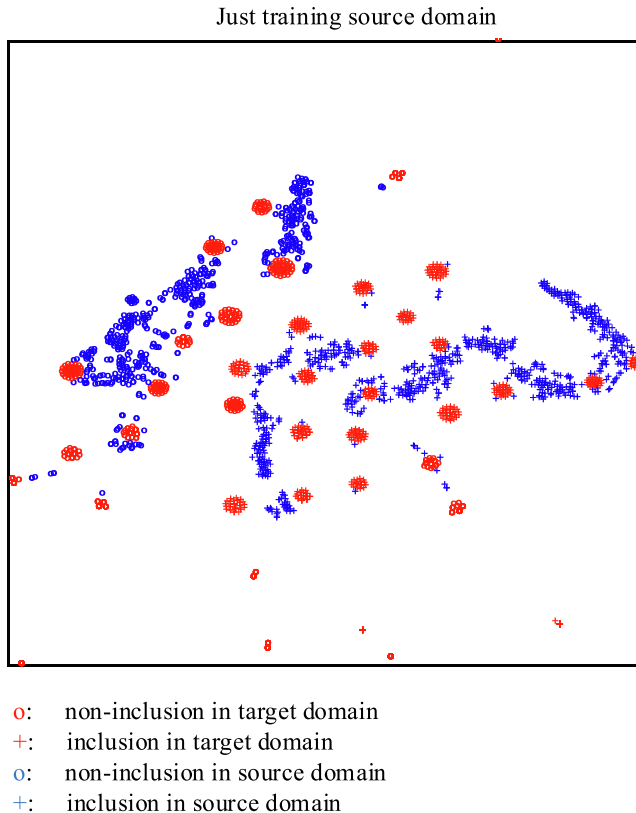
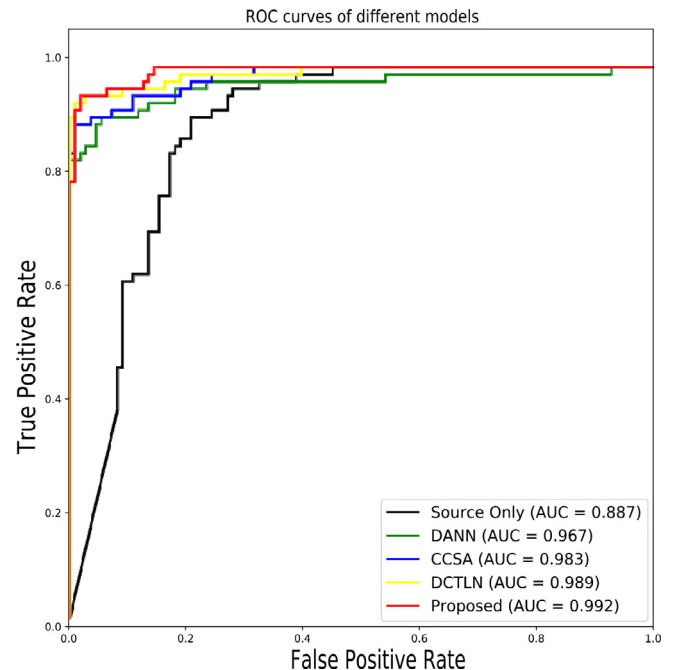
“just training source domain” means samples in the target domain were not trained, which was used to show the necessity of transfer learning. DANN [42] is an adversarial network for domain adaptation, in which feature learning, domain adaptation, and classifier learning are performed jointly using a gradient reversal layer. CCSA [4] tries to align the distributions of sample features from different domains but with the same labels, and separates the distributions of sample features from different domains and with different labels. DCTLN [44] combines DANN and maximum mean discrepancy (MMD). For fair comparisons, all methods had the same numbers of convolutional layers and fully connective layers. Accuracy, precision, recall, and F1_measure were adopted to evaluate the performance of each method. Accuracy indicates how many ACM image samples are correctly detected, precision is the proportion of detected inclusion defects that are correctly identified, and recall represents how many inclusion defects samples are detected from all those with inclusion defects. Their formulas are as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (21)$$

$$Precision = \frac{TP}{TP + FP} \quad (22)$$

$$Recall = \frac{TP}{TP + FN} \quad (23)$$

$$F1_measure = \frac{2 * Precision * Recall}{Precision + Recall}, \quad (24)$$

**Fig. 8.** Visualization of features just trained in the source domain.**Fig. 9.** ROC curves of different models.

where

TP = ACM images both labeled and detected as inclusion;
 TN = ACM images both labeled and detected as non-inclusion;
 FP = ACM images labeled as non-inclusion but detected as inclusion;
 FN = ACM images labeled as inclusion but detected as non-inclusion.

3.2. Experiment results and analysis

From Table 3, it can be seen that the proposed model performs better than the other models in terms of accuracy (96.8%), recall (94.8%), and F1_measure (96%), except that its precision is slightly lower than that of CCSA. This shows our model's ability to detect inclusion defects. When just the source domain dataset was trained, the precision was 0.721 and the F1_measure was 0.804, which was inferior to those with transfer learning. To explain why to just train the source

domain dataset yielded the poorest performance, we visualized the extracted features using t-distributed stochastic neighbor embedding (t-SNE) [45], where high-dimensional semantic features were displayed in a 2-D space. The results are shown in Fig. 8, where the features of non-inclusion class in the source domain are plotted with a blue "o," features of the inclusion class in the source domain are plotted in a blue "+," features of the non-inclusion class in the target domain are plotted in a red "o," and features of the inclusion class in the target domain are plotted in a red "+." The visualization of extracted features in Fig. 8 shows that the features of different categories in the source domain are class discriminative, while those of different categories in the target domain are mixed and just slightly class discriminative. This indicates that the marginal probability distributions between ACM image samples and the group of "welds" samples from GDXray are indeed different ($P(X_s) \neq P(X_t)$), although these two datasets look alike. Thus to apply the model that is just trained in the source domain dataset for defect detection in the target domain dataset is unsuitable.

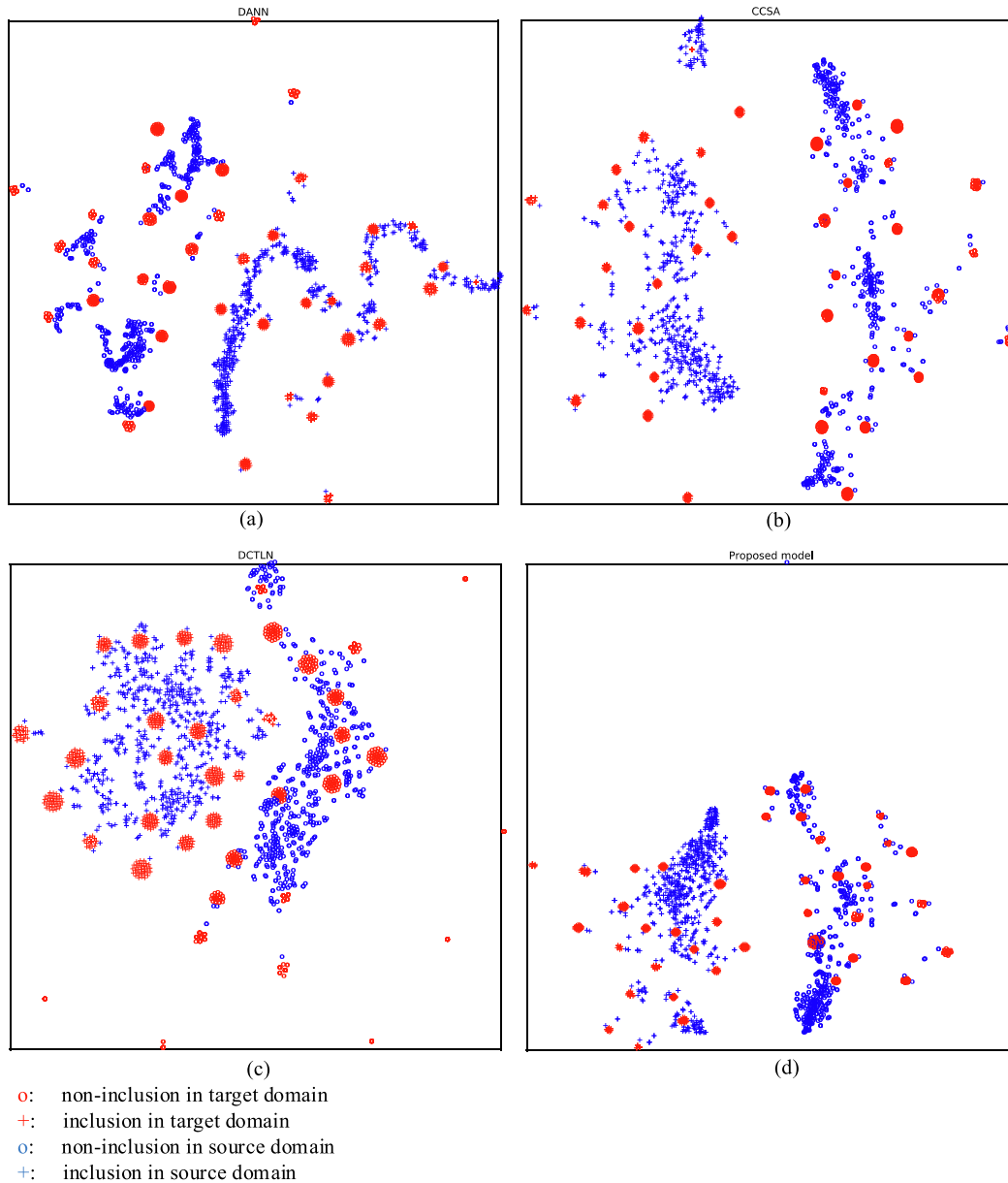


Fig. 10. Feature visualization using t-SNE: (a) DANN; (b) CCSA; (c) DCTLN; (d) Proposed model.

We plot the ROC curves of the above five models to verify the advantages of our model, as shown in Fig. 9. In the ROC curves, x-coordinates denote the false-positive rate (FPR), which is identical to recall in Table 3, and y-coordinates denote the true-positive rate (TPR), which represents how many false inclusion defects samples are detected when all non-inclusion defects are tested. The formulas are as follows:

$$TPR = \frac{TP}{TP + FN} \quad (25)$$

$$FPR = \frac{FP}{TN + FP} \quad (26)$$

AUC was computed for each ROC curve. This is the score of the area under the corresponding ROC curve, whose value is from 0 to 1. The results show that the ROC curve and AUC score of just the training source domain are most unsatisfactory and are plotted in black. In addition, the proposed model has a better ROC curve and AUC score than the others, which is plotted in red, with a corresponding AUC score of 0.992. Looking at Table 3, Figs. 8 and 9, it can be seen that the models without transfer learning perform much worse than those with transfer learning. This shows the importance of transfer learning, and among the transfer learning models, ours performs best. To more intuitively display the extracted features, we visualize them using t-SNE in Fig. 10. Fig. 10(a)–(d) plot features extracted through the DANN model, CCSA model, DCTLN model, and our model, respectively. From Fig. 10(a), it is seen that DANN does not perform well enough in extracting class discriminative features. CCSA only adopts MMD for transfer learning, and it shows good performance. However, the class clustering is not tight enough, as shown in Fig. 10(b). As a multiadaptation method combining adversarial networks and MMD, DCTLN obtains tighter class clustering. In contrast, our model combines adversarial networks and distance metrics, resulting in better clustering of learned features from different classes.

In summary, our proposed model has the best accuracy of 96.8%, best recall of 94.8%, best F1_measure of 96%, and best AUC score of 0.992.

To validate the performance of the proposed method at inclusion defect detection in ACM, we detected the X-ray images of ACM using the sliding-window method and recognized the defects through the well-trained transfer learning model [34]. The size of the detection window was 32×32 pixels, which was the same as that of patch images used in the training procedure. During detection, the detection window slid from the upper-left to the bottom-right of the entire image with a step of 16 pixels, as shown in Fig. 11. Each detection window was input to the well-trained model and classified into the label “1” or “0,” indicating inclusion or non-inclusion, respectively. As we set the sliding step to 16 pixels, some defects could be contained in multiple detection windows and detected many times. Therefore, a heat map was utilized to mark the defects, where places with defects had higher values [34]. The detection results are shown in Fig. 12.

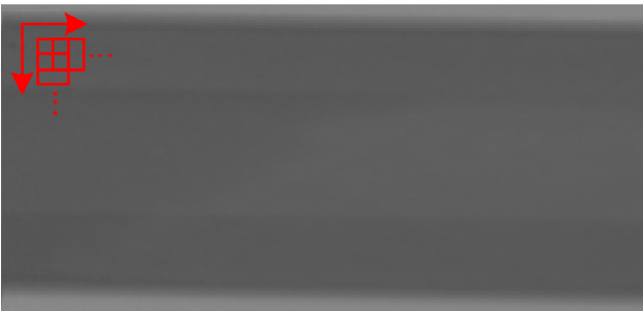


Fig. 11. Schematic of sliding-window during detection.

Fig. 12(a) shows the original X-ray image waiting for detection, where two inclusions exist, indicated by red arrows. After using the sliding-window approach, the corresponding heat map is shown in Fig. 12 (b). It can be seen that places with inclusion defects have higher heat values. This is validated by showing the results of each sliding-window during detection without using the heat map. As shown in Fig. 12(c), for each inclusion defect, there are two sliding-windows passing over them whose detection results are marked in green rectangles, which can be seen to match the heat map. Fig. 13 shows another detection example using the proposed model and sliding-window approach, showing that our method could successfully detect the inclusion defects. In addition to accurate recognition, the detection time of one X-ray image of size 579×296 pixels is 0.24 s, which meets

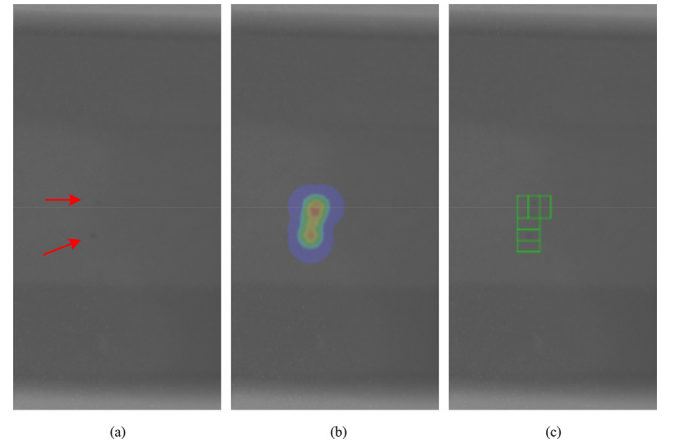


Fig. 12. Inclusion defect detection using sliding-window approach: (a) X-ray image of ACM with inclusion defects; (b) Corresponding heat map results; (c) Detection results without heat map.

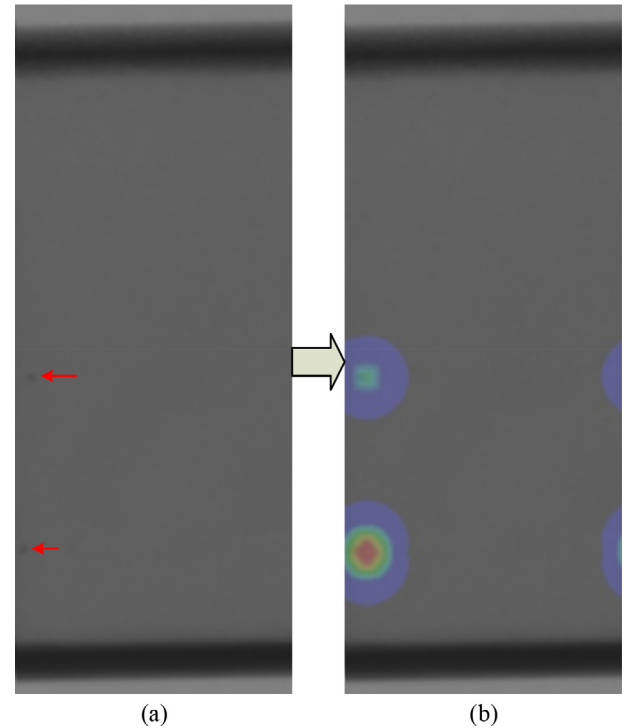


Fig. 13. Another detection example using proposed model and sliding-window approach: (a) X-ray image of ACM with inclusion defects; (b) Corresponding heat map results.

real-time requirements. This can be explained by the small number of parameters of this model. As shown in Table 1, there are 130,144 parameters of the feature extractor and label classifier ($864 + 9216 + 9216 + 102,400 + 8192 + 128$). DCNN with so few parameters, and such high detection precision can also be used in other applications.

4. Conclusion

We applied transfer learning to the automatic inclusion defect detection of ACM, whose defective samples are extremely scarce, and proposed a new deep transfer learning model. To show the advantages of our proposed model, comparison experiments on four widely used methods, i.e., traditional ANN (just training the source domain), DANN, CCSA, and DCTLN, were conducted using the same X-ray image samples of ACM. These X-ray image samples were acquired through an X-ray DR system. Our model obtained an accuracy of 96.8%, recall of 94.8%, F1_measure of 96%, and AUC score of 0.992, exceeding the performance of the other models. These results validate that our model can accurately extract features and obtain higher recognition accuracies for inclusion defects in aerospace, even with scarce samples. Using the proposed model, we further introduced the sliding-window approach to the localization of inclusion defects, and we conducted corresponding experiments using X-ray image samples of ACM. The experimental results showed that our method can accurately detect inclusion defects in X-ray images of ACM. In addition, the proposed model is real-time, with a detection time of 0.24 s of one common-size X-ray image of ACM. Therefore, the proposed method is applicable to the inclusion defect detection of X-ray images of ACM. X-ray images must be cropped to patch images of small size to train the model, and this is time-consuming. Hence, to directly train the model using entire X-ray images will be pursued in our future studies.

CRedit authorship contribution statement

Yanfeng Gong: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Hongliang Shao:** Writing - review & editing, Validation, Formal analysis, Investigation, Resources, Data curation, Funding acquisition. **Jun Luo:** Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration. **Zhixue Li:** Software.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Research on Intelligent Recognition Technology of Spacecraft Composite Defects funded by the Shanghai Institute of Satellite Equipment (No. JG20180209). Thanks for the X-ray image samples of aeronautics composites materials they provided. And we thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.compstruct.2020.112681>.

References

- [1] Gholizadeh S. A review of non-destructive testing methods of composite materials. *Procedia Struct Integrity* 2016;1:50–7.

- [2] Stoik Christopher, Bohn Matthew, Blackshire James. Nondestructive evaluation of aircraft composites using reflective terahertz time domain spectroscopy.. *Ndt & E Int* 2010;43(2):106–15.
- [3] Poudel A, Shrestha SS, Sandhu JS, Chu TP, Pergantis CG. Comparison and analysis of Acoustography with other NDE techniques for foreign object inclusion detection in graphite epoxy composites. *Compos B* 2015;78:86–94.
- [4] Motian S, Piccirilli M, Adjeroh DA, Doretto G. Unified Deep Supervised Domain Adaptation and Generalization; 2017.
- [5] Ospald F, Zouaghi W, Beigang R, Matheis C, Jonuscheit J, Recur B, et al. Aeronautics composite material inspection with a terahertz time-domain spectroscopy system. *Opt Eng* 2013;53(3):123–8.
- [6] Zhang J, Wang J, Han X, Cui HL, Shi C, Zhang J, et al. Noncontact detection of Teflon inclusions in glass-fiber-reinforced polymer composites using terahertz imaging. *Appl Opt* 2016;55(36):10215–22.
- [7] Mieloszyk M, Majewska K, Ostachowicz W. THz spectroscopy application for detection and localisation of water inclusion in glass composite. *Compos Struct* 2018;192:537–44.
- [8] Tan KT, Watanabe N, Iwahori Y. X-ray radiography and micro-computed tomography examination of damage characteristics in stitched composites subjected to impact loading. *Compos B Eng* 2011;42(4):874–84.
- [9] Léonard F, Stein J, Soutis C, Withers P. The quantification of impact damage distribution in composite laminates by analysis of X-ray computed tomograms. *Compos Sci Technol* 2017;152:139–48.
- [10] Garcea SC, Wang Y, Withers PJ. X-ray computed tomography of polymer composites. *Compos Sci Technol* 2017.
- [11] Aymerich F, Francesconi L. Damage mechanisms in thin stitched laminates subjected to low-velocity impact. In: Lopresto V, Langella A, Abrate S, editors. International symposium on dynamic response and failure of composite materials. p. 133–40.
- [12] Little JE, Yuan X, Jones MI. Characterisation of voids in fibre reinforced composite materials. *NDT and E Int* 2011;46(3):122–7.
- [13] Nikishkov Y, Airolidi L, Makeev A. Measurement of voids in composites by X-ray Computed Tomography. *Compos Sci Technol* 2013;89(1):89–97.
- [14] Schilling PJ, Karedla BPR, Tatiparthi AK, Verges MA, Herrington PD. X-ray computed microtomography of internal damage in fiber reinforced polymer matrix composites. *Compos Sci Technol* 2005;65(14):2071–8.
- [15] Toscano C, Meola C, Iorio M, Carlomagno G. Porosity and inclusion detection in CFRP by infrared thermography. *Adv Opt Technol* 2012;2012.
- [16] Chu T, Don J, Pan Y, Poudel A. Defect characterization in commercial carbon-carbon composites. *World J Eng* 2012;9(6):481–6.
- [17] Gregory R. Composite repair inspection by laser shearography. *Insight - Non-Destr Test Cond Monit* 2003;45(3):183–5.
- [18] Ma B, Zhou Z, Zhao H, Zhang D, Liu W. Characterisation of inclusions and disbonds in honeycomb composites using non-contact non-destructive testing techniques. *Insight - Non-Destr Test Cond Monit* 2015;57(9):496–503.
- [19] Wang J, Zhang J, Chang T, Cui HL. A Comparative study of non-destructive evaluation of glass fiber reinforced polymer composites using terahertz, X-ray, and ultrasound imaging. *Int J Precis Eng Manuf* 2019;1–10.
- [20] Ramachandran R, Arunachalam K. Detection of dilute inclusion concentration in polymer metal composite using microwaves. In: International Microwave & Rf Conference; 2014. p.1–4.
- [21] Dong J, Kim B, Locquet A, Mckee P, Declercq N, Citrin DS. Nondestructive evaluation of forced delamination in glass fiber-reinforced composites by terahertz and ultrasonic waves. *Compos B Eng* 2015;79:667–75.
- [22] Mery. Computer vision technology for X-ray testing. *Insight Non-Destr Test Cond Monit* 2014.
- [23] Liang Y, Hwu C. On-line identification of holes/cracks in composite structures. *Smart Mater Struct* 2001;10(4):599.
- [24] Kim YY, Kapania RK. Neural networks for inverse problems in damage identification and optical imaging. *AIAA J* 2015;41(4):732–40.
- [25] Jiang SF, Zhang CM, Zhang S. Two-stage structural damage detection using fuzzy neural networks and data fusion techniques. *Expert Syst Appl* 2011;38(1):511–9.
- [26] Anderson T, Lemoine G, Ambur D. An artificial neural network based damage detection scheme for electrically conductive composite structures. In: 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; 2003. p. 1997.
- [27] Ye L, Su Z, Yang C, He Z, Wang X. Hierarchical development of training database for artificial neural network-based damage identification. *Compos Struct* 2006;76(3):224–33.
- [28] Lee SM, Roh TS, Choi DW. Defect diagnostics of gas turbine engine using hybrid SVM-artificial neural network method. *J Mech Sci Technol* 2009;23(2):559–68.
- [29] Zheng SJ, Li ZQ, Wang HT. Research on delamination monitoring for composite structures based on HHGA-WNN2009.
- [30] Lopes PDS, Jorge AB, Jr SoSeC. Detection of holes in a plate using global optimization and parameter identification techniques. *Inverse Prob Sci Eng* 2010;18(4):439–63.
- [31] Mery D, Arteta C. Automatic defect recognition in X-ray testing using computer vision. *Appl Comput Vision* 2017;1026–35.
- [32] Min M, Chua YJ, Wouterson E, Ong CPK. Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks. *Neurocomputing* 2017;257:01522.
- [33] 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.
- [34] Hou W, Wei Y, Guo J, Jin Y, Zhu Ca. Automatic detection of welding defects using deep neural network. *J Phys: Conf Ser: IOP Publ* 2018;933(1):012006.

- [35] Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M, Inman DJ. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J Sound Vib* 2017;388:154–70.
- [36] Cruz FC, Filho EFS, Albuquerque MCS, Silva IC, Farias CTT, Gouvêa LL. Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing. *Ultrasonics* 2017;73:1–8.
- [37] Elenchezian MRP, Vadlamudi V, Md Raihan R, Reifsnider K. Damage precursor identification in composite laminates using data driven approach. In: *AIAA Scitech 2019 Forum*; 2019. p. 0401.
- [38] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: *International Conference on Neural Information Processing Systems*; 2014. p. 3320–8.
- [39] Pan SJ, Yang Q. A survey on transfer learning. *IEEE Trans Knowl Data Eng* 2009;22(10):1345–59.
- [40] Tan C, Sun F, Tao K, Zhang W, Chao Y, Liu C. A Survey on Deep Transfer Learning; 2018. p. 270–9.
- [41] Mery D, Rizzo V, Zscherpel U, Mondragón G, Lillo In, Zuccar I, et al. GDxray: the database of X-ray images for nondestructive testing. *J Nondestruct Eval* 2015;34(4):42.
- [42] Ganin Y, Lempitsky V. Unsupervised Domain Adaptation by Backpropagation; 2015.
- [43] Goodfellow IJ, Pouget-Abadie J, Mirza M, Bing X, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: *International Conference on Neural Information Processing Systems*; 2014. p. 2672–2680.
- [44] Guo L, Lei Y, Xing S, Yan T, Li N. Deep convolutional transfer learning network: a new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Trans Ind Electron* 2018;66(9):7316–25.
- [45] Laurens VDM. Barnes-Hut-SNE. *Comput Sci* 2013;1301:3342.