Mathematical Biosciences and Engineering

MBE, 19 (1): 997–1025.
DOI: 10.3934/mbe.2022046
Received: 15 August 2021
Accepted: 01 November 2021
Published: 25 November 2021

http://www.aimspress.com/journal/MBE

Research article

Effect of dual-convolutional neural network model fusion for Aluminum profile surface defects classification and recognition

Xiaochen Liu[1,*], Weidong He[1], Yinghui Zhang[1], Shixuan Yao[2] and Ze Cui[3]

[1] School of Mechanical Engineering, Dalian Jiaotong University, Dalian 116028, China
[2] School of Software Engineering, Dalian University of Foreign Languages, Dalian 116044, China
[3] School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China

* **Correspondence:** Email: lxc_jason@163.com.

Abstract: Classifying and identifying surface defects is essential during the production and use of aluminum profiles. Recently, the dual-convolutional neural network(CNN) model fusion framework has shown promising performance for defects classification and recognition. Spurred by this trend, this paper proposes an improved dual-CNN model fusion framework to classify and identify defects in aluminum profiles. Compared with traditional dual-CNN model fusion frameworks, the proposed architecture involves an improved fusion layer, fusion strategy, and classifier block. Specifically, the suggested method extracts the feature map of the aluminum profile RGB image from the pre-trained VGG16 model's *pool5* layer and the feature map of the maximum pooling layer of the suggested A4 network, which is added after the Alexnet model. then, weighted bilinear interpolation unsamples the feature maps extracted from the maximum pooling layer of the A4 part. The network layer and upsampling schemes ensure equal feature map dimensions ensuring feature map merging utilizing an improved wavelet transform. Finally, global average pooling is employed in the classifier block instead of dense layers to reduce the model's parameters and avoid overfitting. The fused feature map is then input into the classifier block for classification. The experimental setup involves data augmentation and transfer learning to prevent overfitting due to the small-sized data sets exploited, while the K cross-validation method is employed to evaluate the model's performance during the training process. The experimental results demonstrate that the proposed dual-CNN model fusion framework attains a classification accuracy higher than current techniques, and specifically 4.3% higher than Alexnet, 2.5% for VGG16, 2.9% for Inception v3, 2.2% for VGG19, 3.6% for Resnet50, 3% for Resnet101, and 0.7% and 1.2% than the conventional dual-CNN fusion framework 1 and 2, respectively, proving the effectiveness of the proposed strategy.

Keywords: Aluminum profile surface defects; feature extraction; CNN model fusion framework; transfer learning; global average pooling

1. Introduction

The aluminum profile is a relatively common material in infrastructure construction and industrial manufacturing that is lightweight, has high strength, corrosion resistance, formability, and is recyclable [1]. It is extensively used in rail transit, construction facilities, automobile manufacturing, equipment manufacturing, medical equipment, and other industries [2]. In the production or use process of aluminum profiles, due to external factors, it may present the defects of inconsistency in various sizes and shapes, seriously affecting the safety and reliability of aluminum profiles. Therefore, detecting and ensuring the surface quality of aluminum profiles is significant to improve the product's service life [3].

In the past, object defects were commonly detected manually. It was a simple, highly repetitive, cost-wasting and labor work, where accuracy and stability could not be guaranteed. With the advancement of optical instruments, numerous scholars have used machine vision to realize defect recognition and improve the detection stability and recognition rate [4]. For example, Gao et al. [5] exploit thermal imaging technology to propose a low-rank tensor sparse mixture Gaussian (MoG) decomposition algorithm for natural crack detection. Their method reduces noise interference and extracts crack information to realize metal defect detection. Luo et al. [6] suggest a hybrid spatial and temporal deep learning architecture for automatic thermography defects detection that extracts internal defect information of composite materials with complex shapes and patterns. Accordingly, Hu et al. [7] developed a hybrid multi-dimensional feature fusion structure involving spatial and temporal segmentation appropriate for automated thermography defect detection of composite materials. Ahmed et al. [8] use the optical pulse thermal imaging diagnosis system and propose a joint sparse low-rank matrix decomposition algorithm to separate weak defect information from intense noise in composite materials and improve defect resolution. Sun et al. [9] investigate weld defect detection and classification based on machine vision. They categorize the weld defects and suggest a modified background subtraction method based on Gaussian mixture models to extract the feature areas of the weld defects, which are then employed to design classification algorithms. Zhang et al. [10] design an image acquisition system to simultaneously collect weld images and propose a new CNN classification model with 11 layers to identify weld penetration defects based on weld imagery. Bao et al. [11] propose a Triplet-Graph Reasoning Network (TGRNet), which combines surface defect triples (including a triple encoder and triple loss) to segment the background and defect areas, and separates them into metal and non-metal classes (leather and tile). For this method, the data is centralized to verify the network's effectiveness. Shervan et al. [12] focus on the surface defect detection problem, considering a new noise-resistant and multi-resolution version of LBP to extract surface features. Additionally, the authors propose a surface defect detection algorithm that is invariant to the texture descriptor. The effectiveness of this technique is verified in architectonic stone and Fabric Textile. Jong et al. [13] suggest a new convolutional variational autoencoder (CVAE) to generate sufficient defect data. Defect classification algorithm based on deep CNN for metal surface defect detection has also been proposed. Ihor et al. [14] design an automated method for detecting and classifying three types of surface defects in rolled metal, and use Resnet50 for feature extraction and defect classification.

Guan et al. [15] utilize VGG19 to extract steel surface defects and suggest different feature layers originating from the defect weight model. Then the authors employ SSIM and a decision tree to evaluate the image quality and adjust the network's structure and classify steel surface defects.

The latter method uses image processing and deep learning methods to extract the defect features of various objects effectively, and to a certain extent, provides insights for the method developed in this paper. Since this article mainly considers identifying and classifying defects on the aluminum profiles surface, the following works introduce the related literature. Defect recognition exploiting conventional machine vision mainly includes image capturing, feature extraction and definition, image preprocessing, and defect-recognition [16]. In this regard, the defect recognition accuracy is seriously affected by the accuracy of the feature extraction process and the method defining the features. Liu et al. [17] employ the gray-level co-occurrence matrix algorithm and the Gabor wavelet transform method to extract the surface texture features of aluminum profiles. They classify the features based on the radial basis function kernel SVM (Support Vector Machines) classification algorithm. Chondronasios et al. [18] propose a new technology based on the gradient-only co-occurrence matrix (GOCM) and the Sobel operator to extract and define the surface features of aluminum profiles.The authors use two-layer ANNs to classify the surface defects of aluminum profiles. Although traditional machine vision-based methods utilize image processing for surface defect feature extraction and defects classification, the extraction and defects definition requires manual processing and empirical judgment by engineers [19], which lacks robustness and is not conducive to operation.

Recently, deep learning has been extensively used in various application, including feature extraction and classification of aluminum profiles surface defects, due to its ability to learn image features automatically. In the context of aluminum profiles surface defects, Li et al. [20] rely on the adaptive threshold method to binarize the surface image of the aluminum plate, extract image features, and implement surface defect classification through a three-layer BP neural network. Wei et al. [21] utilize Resnet101 as the primary network and propose a multi-scale defect detection network based on deep learning to identify and classify surface defects of aluminum profiles. Neuhauser et al. [22] propose a VGG16 based architecture suitable for actual industrialization exploiting transfer learning. and data augmentation to increase the data set, avoiding model overfitting. Zhang et al. [23] design an attention mechanism to detect surface defects of aluminum profiles. This method initially exploits the category representation network to extract the common category feature map (CCM). Then, the attention module generates the proposed feature map (PM), and a rare category feature map (RCM) is formed through CCM and PM. After that, the score of the defect category is obtained through CCM and RCM spatial pooling for defects identification. Chen et al. [24] propose an aluminum profiles surface defect detection method relying on a deep self-attention mechanism (DSAM) under hybrid noise conditions. This technique employs the residual learning strategy to obtain the defect feature map from the image, adds the corresponding weight matrix to the defect feature map to achieve fine feature extraction, and finally adds a softmax classification layer for defect recognition. Liu et al. [25] develop a semi-supervised anomaly detection method, entitled Dual Prototype Auto-Encoder (DPAE). During the training phase, a dual prototype loss and reconstruction loss are introduced to encourage the latent vector generated by the encoder to be closer to its own prototype. Finally, the distance between the image's latent vectors is used to detect and identify the surface defects of the aluminum profile.

The above works exploit deep learning to identify and classify the surface defects of aluminum profiles and achieve good experimental results. Additionally, compared with traditional machine vision,

deep learning-based feature extraction is more robust. However, there are still some issues that need to be resolved. For example, current deep learning methods utilize an input source, a neural network model, and the characteristic information of a single information source extracted through the neural network, which cannot fully reflect the characteristics of the object examined [26].

To solve these problems, the defect classification accuracy can be enhanced through a dual-convolutional neural network(CNN) model fusion framework that extracts the input source features separately, which are then fused. A dual-CNN model fusion framework may have two forms, either employing two different input sources or the same input source. In the former case, the same neural network model extracts features of different input sources and then merges them for classification [26–31]. This case involves neural network models with specific structure differences, e.g., CNN convolution kernel size and number, and several operation differences in the model learning process. In the same input source case, the classification performance varies depending on the extracted features [32]. For this fusion scheme, two different convolutional neural networks separately extract features and then merge them, aiming during the design, the extracted features to complement each other [32,33].

Duan et al. [26]propose a dual-CNN model fusion framework based on gradient images to identify and classify the surface defects of aluminum profiles. The original and gradient images are used as two different input sources, while both neural network models use Alexnet and realize feature fusion through wavelet transform fusion. Then the fused features are input into the SVM classifier block for defect classification, Akilan et al. [33] use the VGG16 and Alexnet networks to extract features from two identical input sources, and employ PCA (Principal Component Analysis) and energy normalization to form a feature space. This work also utilizes algorithm rules (Sum, Average, Max, Min) to fuze the features, with several rules being evaluated to select the optimal fusion strategy. The fuzed features are then input into an SVM classifier block for classification. Experimental results employing this method demonstrate that the Sum strategy is effective in most data sets. The first fusion framework mentioned above combines the output features of the first dense layer of the two Alexnet models, while the second fusion framework combines the output features of the first dense layer of VGG16 and the second dense layer of Alexnet. Both model fusion frameworks have a common attribute: the fused features are first input to the first dense layer of the classifier block, and then classification is achieved through multiple network layers. (the fusion framework will be introduced in the next part of this article). It should be noted that given the lack of research on recognizing and classifying aluminum profile surface defects utilizing a dual-CNN fusion framework, this article mainly refers to methods applied in other fields aiming to suggest the necessary improvements to facilitate a solution appropriate for aluminum profiles.

This article proposes an improved dual-CNN model fusion framework that uses the same input source and different convolutional neural networks (VGG16 and Alexnet). We add multiple network layers before the feature fusion process and after the Alexnet network (including convolution, pooling, and activation). The RGB image feature map is extracted from the last maximum pooling layer in the pre-trained VGG16 and the last maximum pooling layer in the network layer added after the Alexnet network. Then, we use weighted bilinear interpolation to upsample the the maximum pooling layer feature maps of the network layer added after Alexnet to ensure that the feature maps output by the two models have the same dimensions. Feature map fusion relies on the improved wavelet transform fusion method. Finally, our method develops a classifier block (see Section 2) utilizing a global average pooling layer instead of a dense layer.

Compared with traditional dual-CNN model fusion frameworks [26,33], we extract the feature

maps of the largest pooling layer at the end of the proposed CNN model, fuse these feature maps, and use global average pooling for classification rather than dense layers. This strategy preserves more local feature information extracted from the image and reduces the model's dimensionality, making the network easier to train, avoiding too many weight parameters when the feature map enters the dense layer, which leads to overfitting during the model training process [34–36]. Regarding the feature fusion strategy, the improved dual-CNN model fusion framework uses an improved wavelet transform that combines the Canny operator and the area energy method (see Section 2).

The remainder of this article is organized as follows. Section 2 introduces the dual-CNN model fusion framework, network layer function, up-sampling, feature fusion methods, and model training methods proposed in this paper. Section 3 describes the experimental setup and the evaluation metrics, while Section 4 presents the experimental results and analysis. Finally, Section 5 concludes this work.

To improve readability, some of the abbreviations presented throughout the text are defined as follows. Support Vector Machines (SVM) is a class of generalized linear classifiers that classifies binary data in a supervised learning manner. Its decision boundary is the maximum hyperplane margin solved for the learning sample. Principal Component Analysis (PCA) is a standard data analysis method, often used for dimensionality reduction of high-dimensional data that can be utilized to extract data's main feature components. Local Response Normalization (LRN) is a local normalization method that primarily prevents the neural network model from overfitting during the training process.

2. Methods

This section mainly introduces the related methods utilized in the experiments of Section 3, including the dual-CNN model fusion framework, the definition of the relevant network layers, the feature map upsampling method, the feature fusion strategy, transfer learning, data augmentation, and model performance evaluation methods.

2.1. Dual-convolution neural network model fusion frameworks

A traditional convolutional neural network framework includes a neural network model and a single classifier to extract feature information from the input source. This framework is called a single convolutional neural network. In contrast, the multi-convolutional neural network model fusion framework involves multiple convolutional neural network models that extract several features from given training data and inputs the fused features into a single classifier for classification [28]. The dual-CNN model fusion framework includes two network models. The input source features are extracted from the two models, are fused [29,30,32], and are then input into a single classifier for classification. Figures 1 and 2 illustrate the two different dual-CNN model fusion strategies [26,33].

The aluminum profile images are the input source of both fusion frameworks, which are employed to analyze the changes of the corresponding feature maps. The input source of Figure 1 involves the raw image (224X224X3) with the CNN network structure involving the pre-trained VGG16 and Alexnet models. The input source of Figure 2 considers two different images, namely the original (224×224×3) and its variant after image processing (224×224×3), i.e., gradient processing to form a gradient image and enhance the image's edge information [26]. This CNN network structure exploits two Alexnet models that independently exploit each input image. Figure 1 highlights that the first dual-convolutional network model fusion framework combines the output features of the first dense layer

of VGG16 and the second dense layer of Alexnet. The architecture presented in Figure 2 combines the output features of the first dense layer of each Alexnet model, and the fused feature map is input to the classifier block through the dense layer for classification. The CNN's feature maps from the largest convolutional layer input to the dense layer in Figures 1 and 2 are 7×7×512 and 6×6×256, respectively. The output dimension is 4096×1, and the number of weight parameters is 102760448 and 37748736, respectively. It should be noted that the excessive number of weight parameters during training increases the possibility of model overfitting [34].

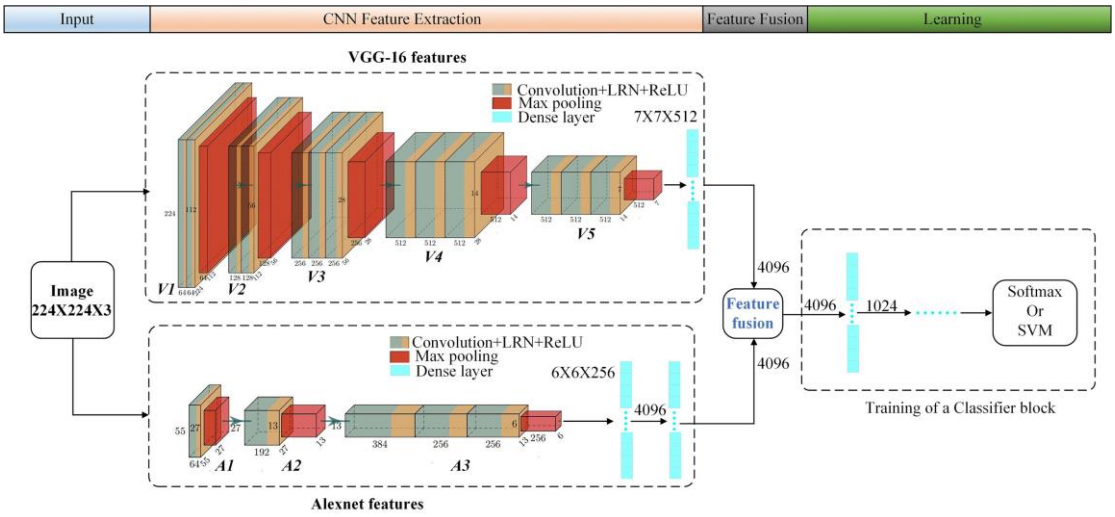


Figure 1. Fusion framework 1.

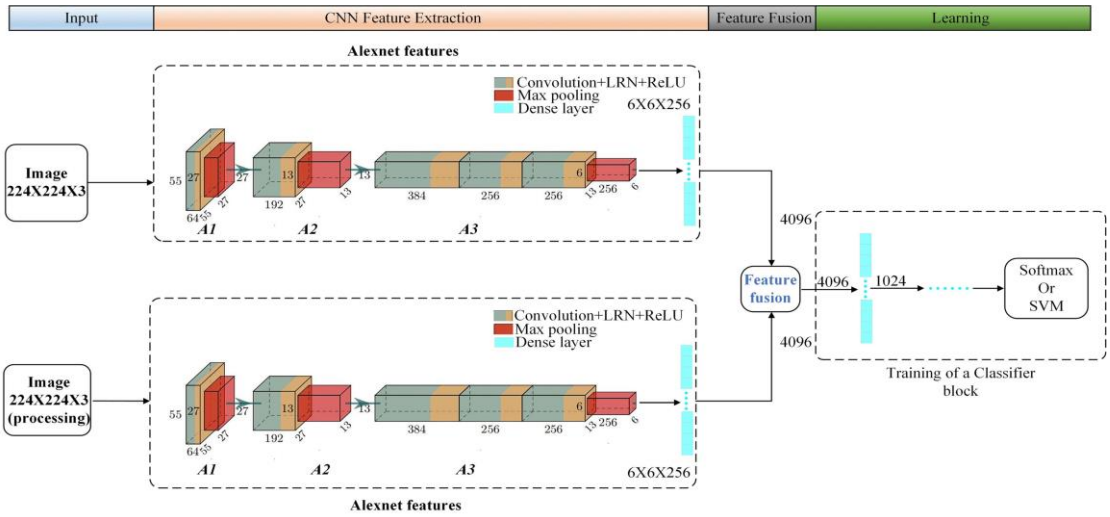


Figure 2. Fusion framework 2.

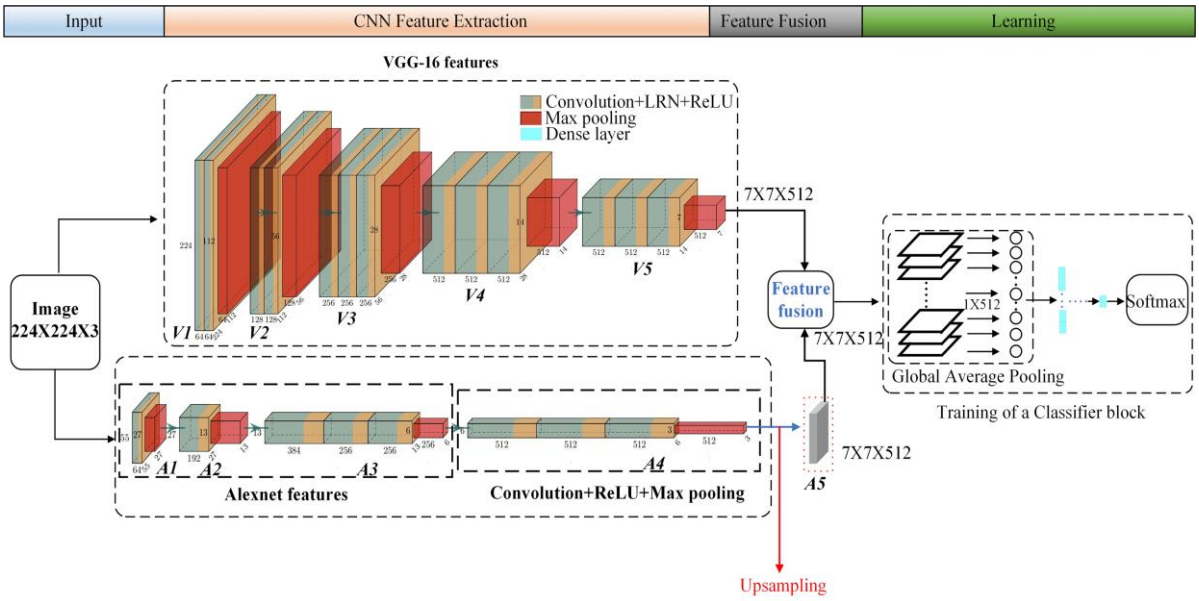


Figure 3. Proposed fusion framework.

The proposed dual-CNN model fusion framework is illustrated in Figure 3. The image input dimension is 224×224×3, and the feature map is extracted from the pre-trained VGG16 and the Alexnet models. The output feature map acquired from the V5 part of the VGG16 model is 512×7×7, while the network layer A4 added after the Alexnet model ensures 512 output feature map channels. Then we upsample the feature map generated by the largest pooling layer in the A4 part, from 3×3 to 7×7, and the Feature Fusion part performs feature fusion preserving the feature maps' channel number and size. The fusion feature map is 512×7×7 and is directly input to the global average pooling layer. Considering the latter layer, global average pooling is performed only on the feature map without training the weight parameters to avoid overfitting due to the excessive number of parameters. The output size is 512×1×1, and finally, the classifier block classifies the output result. To the best of our knowledge, this dual-CNN model fusion framework employing global average pooling instead of dense layers to build classifier blocks has not been applied yet to classify and detect aluminum defects. The feature fusion, upsampling, and global average pooling schemes will be introduced in detail in sections 1.2 and 1.3 of the main text.

The above dual-CNN model fusion framework exploits VGG16 and Alexnet as the primary neural network models. Both have demonstrated outstanding results in image classification and target detection tasks and their generalization performance to migrate to other image data domains [19,37,38]. Furthermore, other existing CNN models may impose some training complexity due to complex connections and deeper structures, while VGG16 and Alexnet are "straight-type" structures. As the number of network layers increases, the extracted features represent finer details, better-facilitating feature fusion. Additionally, Alexnet is the 2012 ImageNet competition champion containing five convolutional layers (including the activation and pooling layers), three fully connected layers, and the classifier output category is 1000. The VGG16 model is the champion of the 2014 ILSVRC competition classification project, with the model containing 16 convolutional layers (including the activation and pooling layers) and three fully connected layers.

2.2. A brief overview of convolutional neural networks

(1) Convolution layer and pooling layer

The convolution layer extracts data features from the input image through convolution operations [39]. Convolution is a linear operation between the input image and the convolution kernel involving a dot product operation within the convolutional process between the convolution kernel and the input image's receptive field. The convolution kernel size is increasing with a specific step size to match the various receptive field sizes. The convolution function is:

$$I_j = \sum x_i * k_j + b_j \tag{1}$$

where $*$ denotes the convolutional operation, k_j and b_j are the weight and bias vectors of the convolutional kernel j, respectively, and x_i denotes the input of the convolutional layer.

The pooling layer aims to downsample the feature map, compress the model features, and simplify the network complexity. The pooling process can be distinguished into Max pooling and Mean pooling. In the proposed CNN network structure, the model uses maximum pooling in the early stage to reduce redundant features and extract texture and other features, while in the later stages, average pooling retains the image background features [40].

(2) Global average pooling layer

Global Average Pooling (GAP) is a method for spatial dimensionality reduction through pooling. Employing global average pooling rather than dense layers affords to reduce the model parameters, avoids over-fitting, and improves the entire network's generalization ability. Additionally, the spatial\semantic information extracted by each convolution and pooling layer is preserved [35,41,42]. In this paper, the feature map produced by our method dimension after fusion is of size 512×7×7. After fusion, we reduce the model's parameters utilizing global average pooling to calculate the average feature map of all pixels within each channel. The final output model is 512×1×1, with the corresponding schematic diagram of the global average pooling illustrated in Figure 4. GAP replaces the dense layer that generates many parameters after the feature fusion process. Since the global average pooling layer has no parameters, it can prevent the layer from overfitting, integrate global spatial information, and have better robustness to the spatial translation of the input image.

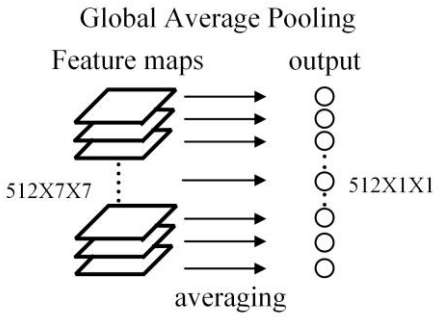


Figure 4. The diagram of global average pooling.

(3) Dense layer with dropout

The dense layer resizes the features extracted by the convolutional and pooling operation and

guarantees that features can be mapped regardless of their sizes. When the training sample size is small and the model parameters are many, over-fitting is prone to occur, and the model's generalization ability is weakened. Dropout reduces the possibility of overfitting, achieving the regularization effect [43], i.e., during the forward propagation process, the activation value of the neuron stops according to the defined Dropout date. Dropout reduces the complex cooperative adaptation relationship between neurons and avoids overfitting. We adopt [44] and set the Dropout date parameter to 0.5.

(4) Batch normalization layer

Adding batch normalization to the training process of the CNN model can achieve a stable activation value distribution, ensure the input data distribution per layer is relatively stable, and accelerate the model's learning process. Batch normalization reduces the model's sensitivity to the network's parameters, simplifies the tuning process, stabilizes the network learning, and has a particular regularization effect in the model training process [45]. Thus, we use batch normalization to maintain all layer inputs on the same range in the classifier block.

(5) Activation layer and softmax

The CNN model implements linear operations through convolutional layers in the forward propagation process. Multiple linear transformations in the network cause data expansion and insufficient model classification capabilities. The activation layer completes the nonlinear data transformation, performs data normalization, prevents overflow caused by excessive data, and increases the network's capabilities. ReLU (Rectified Linear Unit) was introduced as a nonlinear activation function, increasing network nonlinearity, preventing gradients from disappearing, and reducing network training time [46].

The Softmax layer is placed at the end of the model, and its function is to map the generated sample label space to (0,1) as the result of the classification task. The Softmax function is given by:

$$P(y^{(i)} = n \mid x^{(i)}; W) = \begin{bmatrix} P(y^{(i)} = 1 \mid x^{(i)}; W) \\ P(y^{(i)} = 2 \mid x^{(i)}; W) \\ \cdots \\ P(y^{(i)} = n \mid x^{(i)}; W) \end{bmatrix} = \frac{1}{\sum_{j=1}^{n} e^{W^T x^{(i)}}} \begin{bmatrix} e^{W_1^T x^{(i)}} \\ e^{W_2^T x^{(i)}} \\ \cdots \\ e^{W_n^T x^{(i)}} \end{bmatrix} \tag{2}$$

where $e^{W_n^T x^{(i)}}$ is the softmax layer input, $P(y^{(i)} = n \mid x^{(i)}; W)$ represents the probabilities of the *i*th training example，and "n" denotes the model output. class cardinality with the sum of the class probabilities being one. Finally, the proposed dual-CNN fusion framework outputs the probability values of 4th four categories through a softmax layer. Thus we set n = 4. We employ cross-entropy as the loss function in softmax to determine how close the actual output is to the expected output, as in multi-classification tasks, the experimental effect of cross-entropy is closer to the ideal value. The cross-entropy loss function is:

$$L_{\log}(Y, P) = -\log \Pr(Y \mid P) = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} y_{i,k} \log p_{i,k} \tag{3}$$

where $y_{i,k}$ is the true label value, $p_{i,k}$ represents the probability value corresponding to the k-th label under the ith sample, and N is the total number of samples.

(6) Classifier block

Figure 5 presents the classifier block, including a global average pooling, dense, batch normalization, ReLU, dropout, and softmax layer. If too many parameters exist in the dense layer, utilizing global average pooling instead of a dense layer reduces the model weight parameters and avoid overfitting. Accordingly, batch normalization maintains all layer inputs in the same range, dropout prevents overfitting due to being a regularization technique, and Softmax defines the output category probability.

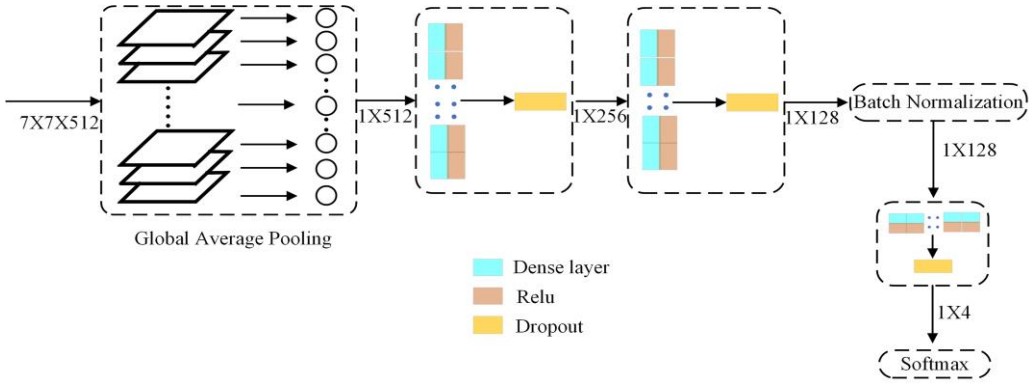


Figure 5. Classifier block.

2.3. Upsampling and Feature fusion

(1) Upsampling methods

When the VGG16 and Alexnet models perform feature map fusion, the feature map size is inconsistent. Thus, we upsample the feature map of the Alexnet model to expand its size. Commonly upsampling methods include deconvolution, depooling, and interpolation [34]. The interpolation method is simple to operate and easy to implement, and thus in this work, we employ interpolation for upsampling. Standard interpolation methods mainly include the Nearest Neighbor, Bilinear, and Bicubic Interpolation [47]. The Nearest Neighbor interpolation algorithm is processing efficient but imposes noticeable distortion, mosaic, and aliasing [48]. Therefore, this article mainly compares the effects of Bilinear, Bicubic, and Weighted Bilinear interpolation.

Bilinear interpolation (BI): Figure 6 presents a schematic diagram of a bilinear interpolation process, with the pixel value at point p being the one to determine. Q_{12} and Q_{22} are pixels with known pixel values in the same direction. The pixel value of R_2 can be obtained by linear interpolation between Q_{12} and Q_{22}, and the pixel value of R_1 by linearly interpolating Q_{11} and Q_{21}. Finally, the pixel value of point p can be calculated by linearly interpolating R_1 and R_2. For this process, the involved formulas are Eqs 4–6. Specifically, the output of function f is the p's pixel value. Given the known value of $Q_{11}(x_1, y_1)$, $Q_{12}(x_1, y_2)$, $Q_{21}(x_2, y_1)$, $Q_{22}(x_2, y_2)$, where x and y are pixel coordinates, by using bilinear interpolation, we first Interpolate Q_{11} and Q_{21} in the x direction to get:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad \text{where } R_1 = (x, y_1) \tag{4}$$

According to Q_{12} and Q_{22}:

$$f(R_2) \approx \frac{x_2-x}{x_2-x_1} f(Q_{12}) + \frac{x-x_1}{x_2-x_1} f(Q_{22}) \quad \text{where } R_2 = (x, y_2) \tag{5}$$

Then interpolate R_1 and R_2 in the y direction to get:

$$f(P) \approx \frac{y_2-y}{y_2-y_1} f(R_1) + \frac{y-y_1}{y_2-y_1} f(R_2) \tag{6}$$

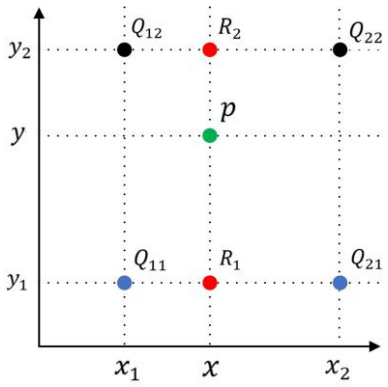


Figure 6. Schematic diagram of bilinear interpolation.

Weighted Bilinear interpolation (WBI): This method optimizes the interpolation effect by adding weight values in the x and y directions. The value of the unknown pixel point p is calculated by interpolating the pixels in the x and y directions. Adding weight values can adjust the linear relationship of the fitted data, improve the linearity of the boundary image data changes to a certain extent, and ensure that the image boundary texture is evident [49].

$$\begin{cases} w_x = \frac{f(Q_{21})+f(Q_{22})-f(Q_{11})-f(Q_{12})}{2*(x_2-x_1)} \\ w_y = \frac{f(Q_{12})+f(Q_{22})-f(Q_{11})-f(Q_{21})}{2*(y_2-y_1)} \end{cases} \tag{7}$$

where w_x and w_y are the weight values in the x and y directions, respectively. Eq 8 presented next is the calculation formula for the pixel value of the p point after the weight is added:

$$f_w(P) = \frac{w_y}{w_x+w_y}\left(\frac{f(R_1)+f(R_2)}{2}\right) + \frac{w_x}{w_x+w_y} f(P) \tag{8}$$

Bicubic interpolation (BCI): The difference between bicubic and bilinear interpolation is the increase in fitting data. Assuming that the original image size is (m, m) and the interpolated target image size is (M, M), we first determine the image ratio relationship m/M = 1/K, and the unknown point *P*(X, Y) corresponds to the original image in the target image. For the coordinates *p* (X/K, Y/K) on the image, the bicubic interpolation needs to find the nearest 16 pixels around point *p*. Then the bicubic function is constructed to calculate the weight of the 16 nearest pixels, and the pixel contribution value is obtained by the product of the weight and the pixel value [48].

$$w(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & \text{for } |x| \le 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & \text{for } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where w(x) is the bicubic function that obtains the coefficients corresponding to the 16 adjacent pixels to pixel p, and $a = -0.5$. The weight of 16 pixels can be calculated from Eq 9, and the pixel value of P can be calculated from:

$$P(X,Y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} * W(i) * W(j) \quad (10)$$

where a_{ij} represents the pixel to be fitted, $W(i)$ the weight on the abscissa of a_{ij}, and $W(j)$ the weight on the ordinate of a_{ij}.

(2) Feature fusion methods

Currently, several feature fusion methods exist, with the most common ones being sum, maximum, and wavelet transform fusion [26], with each feature fusion method having a particular impact on the experiment's accuracy. During the experiment, we compare the classification accuracy of various fusion methods, including sum, maximum, wavelet transform, and improved wavelet transform fusion. These methods are introduced next, and their interplay on the classification accuracy is presented in Section 4.

Sum fusion (SF): This is a standard feature fusion method, which is often utilized in image pixel-level feature fusion and feature-level fusion schemes [26,33]. The summation fusion involves adding the corresponding pixels in the same dimension of the two feature maps. The summation and fusion formulas are:

$$\begin{cases} F = [F_1, F_2, F_3 \dots, F_k] \\ F_k = F_k^{V(i,j)} + F_k^{A(i,j)} \end{cases} \quad (11)$$

where F represents the total fusion feature map of size 512X7X7, k = 1,2,3…,512 represents the feature map channels, with a feature map size per channel of 7X7 (the dimension remains unchanged after the feature map fusion process completes). $V(i,j)$ represents the pixel (i,j) value for the k-channel in the VGG16 feature map, $i = j$ = 1,2,3….,7, and $A(i,j)$ denotes the pixel (i,j) value of the upsampled Alexnet network feature map for the kth-channel.

Maximum fusion (MF): This method compares the corresponding pixels of the same dimension in two feature maps and selects the largest one as the fused pixel. The maximum fusion formula is:

$$\begin{cases} F = [F_1, F_2, F_3 \dots, F_k] \\ F_k = Max[F_k^{V(i,j)}, F_k^{A(i,j)}] \end{cases} \quad (12)$$

Improved Wavelet transform fusion (IWTF): This scheme performs wavelet transformation on two original images, transforms them into high-frequency and low-frequency image signal components, then fuses these components of different feature domains to obtain a new wavelet tower. Finally, it performs fusion transformation through an inverse wavelet.

Wavelet transform fusion (WTF) manages a very appealing reconstruction ability ensuring no information loss and redundant information in the signal's decomposition process [50]. Let the coefficients of images A and B be (cA, dA_i^{ε}) and (cB, dB_i^{ε}) after i-layer wavelet decomposition, and

the coefficients corresponding to the image after fusion be (cF, dF_i^{ε}). c represents the low-frequency coefficient of the image in the i-th layer and d the high-frequency coefficient of the image in the ε direction of the i-th layer. The low-frequency information includes the image's outline, and the high-frequency information includes the image's details. The traditional wavelet transform fusion method uses weighted average fusion at low frequencies (Eq 13) and employs the most considerable absolute value of coefficients at high frequencies (Eq 14). Finally, (x, y) indicates the coefficient location. In Eq 15, F represents the total fusion feature map of size 512X7X7, with k = 1,2,3,…,512 representing the feature map channels of size 7X7 per channel. After the feature map fusion process completes, its dimension is preserved. F_k^V and F_k^A denote the feature map under the k-th channel for VGG16 and Alexnet, respectively. The corresponding feature map channels generated by VGG16 and Alexnet undergo a wavelet transform fusion as follows:

$$cF(x,y) = \frac{1}{2}[cA(x,y) + cB(x,y)] \tag{13}$$

$$dF_j^{\varepsilon}(x,y) = \begin{cases} dA_j^{\varepsilon}(x,y), \left|dA_j^{\varepsilon}(x,y)\right| \geq \left|dB_j^{\varepsilon}(x,y)\right| \\ dB_j^{\varepsilon}(x,y), \left|dA_j^{\varepsilon}(x,y)\right| < \left|dB_j^{\varepsilon}(x,y)\right| \end{cases} \tag{14}$$

$$\begin{cases} F = [F_1, F_2, F_3 \ldots, F_k] \\ F_k = \text{Wavelet transform fusion}[F_k^V, F_k^A] \end{cases} \tag{15}$$

This paper employs the db4 wavelet to decompose the original image involving three wavelet layers and obtains the image's high and low-frequency coefficients. The low-frequency coefficients are fused through a weighted average scheme, while for the high-frequency coefficients, the Canny operator is applied to perform edge detection, extract the edge area information, and reduce subsequent image fusion data. In the edge area, the area energy selects the high-frequency coefficients [51–53], and finally, the fused wavelet coefficients are subjected to wavelet inverse transformation to realize fusion. The edge region extracted by the Canny operator is divided into MxN regions, which in this work is M = N = 2. Then, we employ Eq 16 to find the average wavelet energy of each area block, and finally, Eq 17 to determine the high-frequency coefficient. The fusion flow chart utilizing Wavelet transform is illustrated in Figure 7.

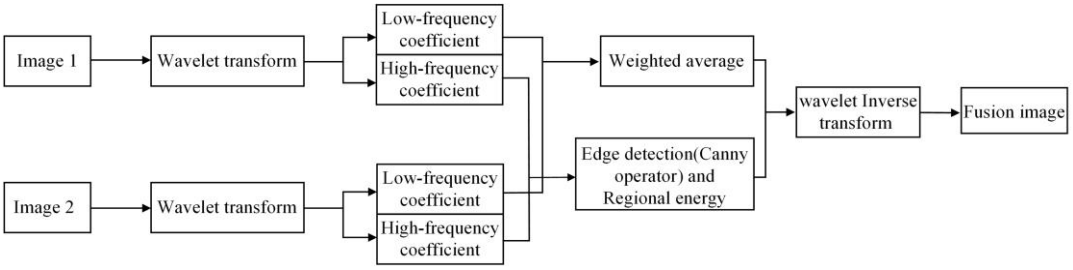


Figure 7. Wavelet transform fusion.

$$\begin{cases} E_A = \sum_{x=1}^{M} \sum_{y=1}^{N} G_A(x,y)^2 / M * N \\ E_B = \sum_{x=1}^{M} \sum_{y=1}^{N} G_B(x,y)^2 / M * N \end{cases} \tag{16}$$

$$G_F = \frac{E_A}{E_A+E_B} * G_B(x,y) + \frac{E_B}{E_A+E_B} * G_A(x,y) \quad (17)$$

where E_A, E_B represent the average wavelet energy and $G_A(x,y)$, $G_B(x,y)$ the high-frequency coefficients of the images A and B in the current area block, respectively. Eq 17 expresses the weighted addition between the high-frequency coefficients of images A and B and the average energy， while G_F the high-frequency coefficient after fusion.

2.4. K-fold cross-validation

K-fold cross-validation is a way to build a model and verify its parameters when fewer data sets are utilized in a deep learning scheme. The sample data are combined into different training and validation sets, where the training set trains the model, and the validation evaluates the model's accuracy, preventing the model from overfitting [54]. K-fold cross-validation divides the sample data into K random subsets, where K-1 subsets are employed as the training set, and the remaining one is the validation set. Since the sample data is divided into training sets, there are K choices, and thus the training and the verification errors need to be calculated each time. Finally, the K calculations of the model's training and verification errors are averaged to obtain the cross-validation errors [55] that are ultimately used to evaluate the model's performance. Figure 8 presents the K-fold cross-validation graph. In our research, we set K = 5.

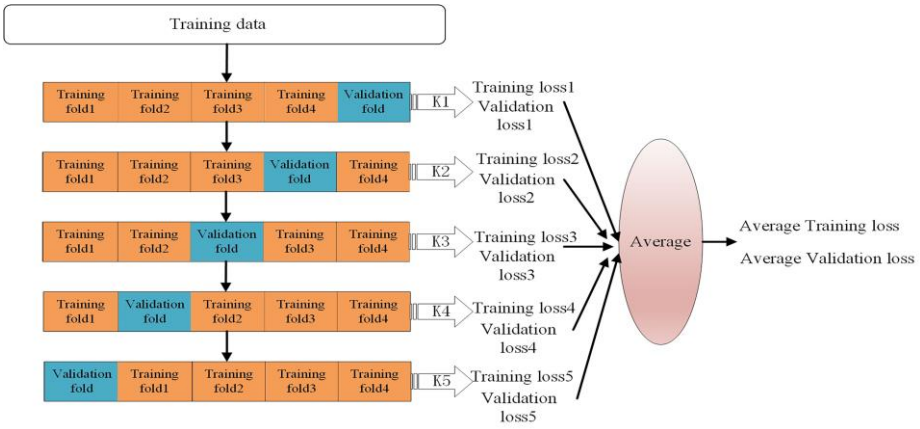
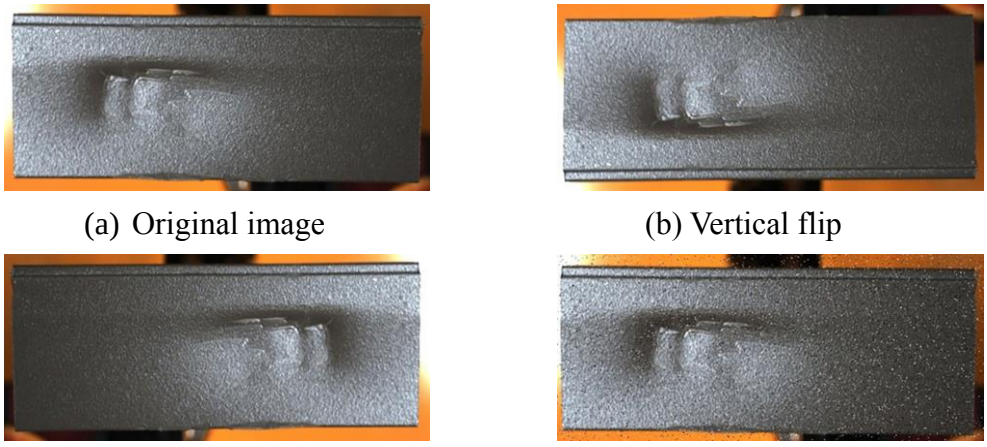


Figure 8. The diagram of K-fold cross-validation.

2.5. Transfer learning and data augmentation

Transfer learning is used in deep learning to solve model overfitting and the poor robustness caused by insufficient or few data sets [56]. Transfer learning allocates parameters generated in the model training process under one data set to model another by realizing parameter sharing. This work considers the VGG16 and Alexnet as the basic pre-trained models for object detection on the ImageNet dataset. During this pre-training period, millions of parameters are learned to obtain standard visual features fed to our convolutional neural network. However, in the proposed dual-CNN model fusion framework, we freeze the feature layer parameters of the VGG16 and Alexnet models, extract the feature layer features, and train the parameters of the classifier block.

During the CNN training process, exploiting data sets with only a few samples per class imposes the model to overfit and reduces its test accuracy and generalization ability. In this work, the aluminum profile data set exploited originates from a factory that uses a digital camera for image collection, and thus the number of data sets is insufficient. Therefore, data enhancement is applied to the original data set to expand the data set and improve the model's robustness [57,58]. Data augmentation methods include dimming, horizontal rotation, vertical rotation, and noise addition (Gaussian noise and salt and pepper noise). Adding noise simulates low image quality due to external factors during the actual image acquisition, transmission, and storage. we mainly use horizontal and vertical rotation and salt and pepper noise. The horizontal and vertical rotation involves 180-degrees rotation from left to right and bottom to top, respectively. Considering noise, the signal-to-noise ratio is set to 0.95, 0.9, 0.75. This article has 3568 images enhanced by horizontal rotation, while for the experiments, we create 3573 vertically rotated images and 3571 images with salt and pepper noise. Examples of the three data enhancement methods are illustrated in Figure 9.



(a) Original image (b) Vertical flip

(c) Horizontal flip (d) Salt and Pepper noise

Figure 9. The data augmentation.

3. Experimental setup

This section mainly introduces the visual acquisition equipment, model training environment, experimental data, and the qualitative evaluation indicators involved in the experiment process.

3.1. Machine vision setup

Figure 10 illustrates the designed image capturing device, including an ABB120 robotic arm, light shield, LED strip light source, background board, camera, and conveyor belt. The hood (0.25m × 0.25m × 0.65m) is designed to create a suitable lighting environment and avoid substantial light interference during image capturing. The LED strip light source with an adjustable brightness improves the image surface collection effect. Experimental tests have proved that choosing the orange color for the background plate can enhance the contrast between the image and the background. The image acquisition equipment uses a Hikvision industrial camera (model: MV-CE060-10UM) with a resolution of 3072 × 2048, which is installed 60mm under the hood. The image acquisition process is: the robotic arm utilizing an end effector grabs the aluminum profile workpiece (known position) and

places it horizontally under the camera. The trigger time is set to capture the first image, and then the robotic arm changes the posture and position of the aluminum profile into a known orientation to capture the second image. After that, the end joint rotates 180 degrees to capture the third image, and in total, three images per aluminum profile workpiece are captured. Finally, the aluminum profile workpieces are sorted, the robotic arm places them on the conveyor belt, and the PLC controls the conveyor belt to move them to their designated position.

Figure 11 presents the collected surface image of the aluminum profile workpiece. Side 1 is the first image taken horizontally under the camera when the robot arm grabs the aluminum profile workpiece. Side 2 is the second image taken after the robot arm changes its posture, while Side 3 refers to the third image after the end joint rotates 180 degrees.

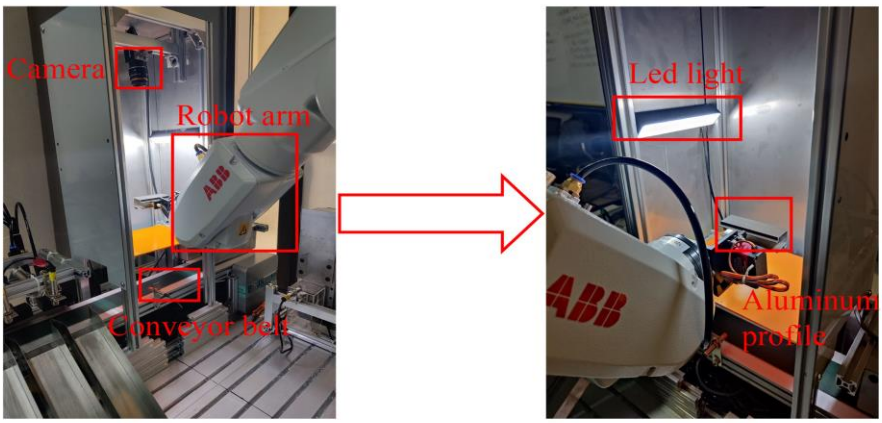


Figure 10. The designed image acquisition setup for automatic online classification.

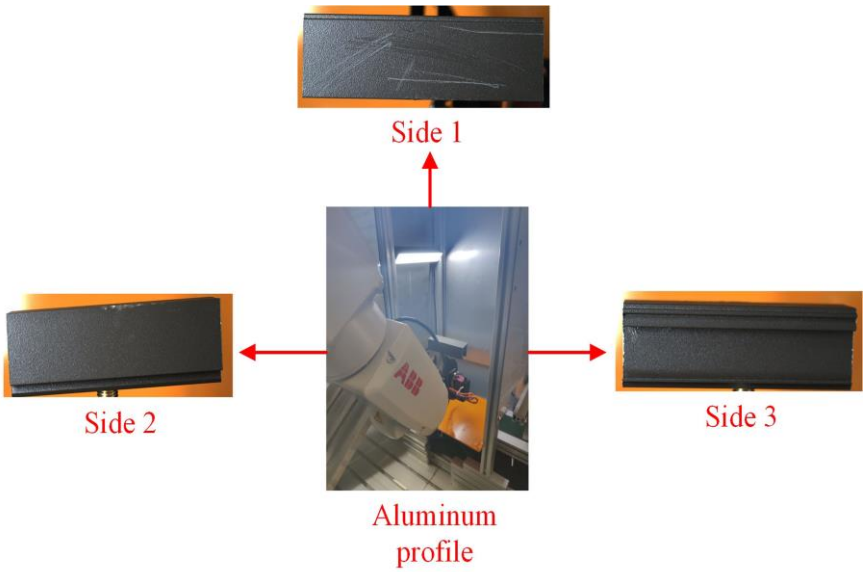
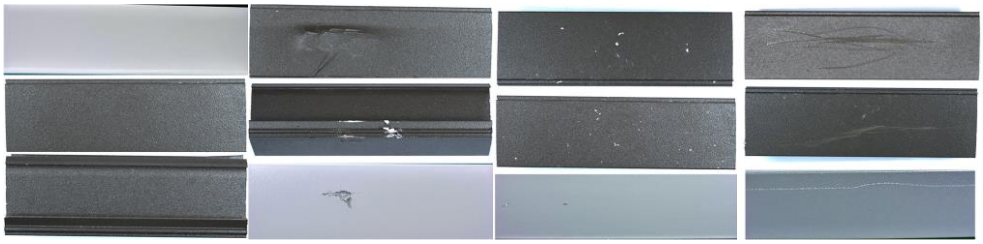


Figure 11. The aluminum profile surface image.

3.2. Experimental development environment and dataset

For the trials, we utilize the Tensorflow deep learning framework. The CNN models are trained employing an NVIDIA GeForce GTX 1060 6GB GPU, and the software environment is python 3.8.3. The CNN model employs the Adam optimizer with a learning rate and learning rate decay of 0.001 and 1e-5, respectively, a batch size of 16 and 50 epochs for the network training. The remaining parameters are the default ones of Tensorflow.

The experiment exploits the images collected from the aluminum profile workpiece utilizing the image acquisition device of Figure 10. The entire data set includes three single defect sample types and one non-defect sample type, which adopt the classification categories of. Specifically, in Figure 12(a), the surface is smooth and flat, i.e., the Intact class. In Figure 12(b), an external force affects the surface, and the damaged area is large, i.e., Bruise class. The small dirty spots on the surface in Figure 12(c) are Dirty spots (DS) class. In Figure 12(d), irregular scratches appear on the surface that are unevenly distributed, which is classified as a Scratch. Table 1 presents the number of samples in various categories. The dataset contains 14282 surface images, including the original collected images and the images after data augmentation. The data set is divided into a test set (1430 images) and a training set (12852 images) following a 1:9 ratio. The training set undergoes a 5-fold cross-validation process during which the training samples are10280 and the verification samples are 2572.



(a) Intact (b)Bruise (c) Dirty spots(DS) (d) Scratch

Figure 12. The sample images of aluminum profile surface.

Table 1. Classification and statistics of aluminum profile workpiece data set

Category	Number of original images collected		Number of images enhanced by data	
	train	test	train	test
Intact	828	93	2484	276
Bruise	522	59	1566	174
Dirty spots(DS)	980	108	2944	328
Scratch	882	98	2646	294

3.3. Quantitative evaluation metrics

In the training and verification phase, the training effect is monitored by displaying the classification accuracy (CA) and cross-entropy loss (CEL) changes in real-time [59,60]. In the testing phase, the robustness and generalization of the model are verified through indicators such as confusion matrix, ACC, PPV, TPR, and F-score. All indicators are described in detail below.

Classification Accuracy (CA): The ratio of the correctly predicted samples to the total number of actual samples. The subsequent trials consider Training accuracy, Validation accuracy, and Test accuracy.

Cross-Entropy Loss (CEL): Assesses the gap between the actual and prediction classes. The experiments involve Training loss and Validation loss.

Confusion matrix: The predicted results of all categories and the real results are placed in the same table based on their category. This table highlights the number of correct and incorrect identifications per category.

We use the positive predictive value (PPV) or precision, true positive rate (TPR) or recall, F-score, and accuracy (ACC) to evaluate the model's performance [59]. These metrics are calculated utilizing Eqs 18–21, respectively. For class x, $\mathrm{TP_X}$ represents the number of correctly predicted x, $\mathrm{PPV_X}$ the number of correctly predicted x divided by the total number of predictions belonging to x, and $\mathrm{TPR_X}$ is the number of predicted x divided by the total number of actual x. The F-score is utilized to combine PPV and TPR metrics into one metric using the harmonic mean, which is detailed in Eq 20. Finally, Eq 21 shows the ACC definition, where n represents the number of categories and l_i the number of each category.

$$\mathrm{PPV_X} = \frac{\mathrm{TP_X}}{\mathrm{TotalPredicted\ X}} \tag{18}$$

$$\mathrm{TPR_X} = \frac{\mathrm{TP_X}}{\mathrm{TotalActual\ X}} \tag{19}$$

$$\text{F-score}_{\,x} = \frac{1}{0.5/\mathrm{TPR_X} + 0.5/\mathrm{PPV_X}} \tag{20}$$

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} TP_i/I_i}{n} \tag{21}$$

4. Experiments results and analysis

This section describes and analyzes the experimental results. During the training process, the K-fold cross-validation method determines the best model through analysis by evaluating all models' performance (Section 4.1). We combine various interpolation and feature fusion methods for model training and testing and then analyze them to determine the optimal combination (Section 4.2). We also compare the performance indicators of various fusion frameworks during the training and testing process (Section 4.3), and finally, we analyze the current research deficiencies and propose future research directions (Section 4.4.).

4.1. K-fold cross-validation and performance evaluation

The VGG16 model's maximum pooling layer of the V5 part and Alexnet model's A4 part were selected as the feature map fusion positions during the experiments. After convolution, using maximum pooling reduces redundant features and extracts texture features [61]. To upsample the feature map of the maximum pooling layer of the A4 part, we use weighted bilinear interpolation, while the wavelet transform fusion method is employed for feature map fusion. Table 2 shows the classification accuracy (CA) and cross-entropy loss (CEL) values of each fold of our proposed fusion scheme. According to Table 2, the average CA and CEL values during training are 0.977 and 0.109,

respectively, while the corresponding validation image data set's results are 0.970 and 0.124, respectively. During training, the best efficiency is attained in the fourth fold of the 5-fold process. Figure 13 illustrates the change curve of CA and CEL during the 4-fold training. Specifically, when the epoch is greater than 20, the gap between the verification accuracy and the loss curve tends to stabilize, and there is no significant change in the accuracy and loss values as the loss slowly decreases throughout the training process. The model's CA in the training and validation data sets are 0.983 and 0.971, while CEL is 0.097 and 0.116, respectively. Throughout the experiments, the same cross-validation method is used to validate the competitor models. Section 4.2 introduces in detail the experimental results by evaluating three interpolation methods and four feature fusion schemes.

Table 2. Comparison of the proposed fusion frameworks in each fold.

Fold	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
1	0.973	0.121	0.965	0.129
2	0.980	0.109	0.968	0.128
3	0.975	0.118	0.966	0.133
4	**0.983**	**0.097**	**0.971**	**0.116**
5	0.976	0.104	0.981	0.112
Average	0.977±0.006	0.109±0.012	0.970±0.011	0.124±0.012

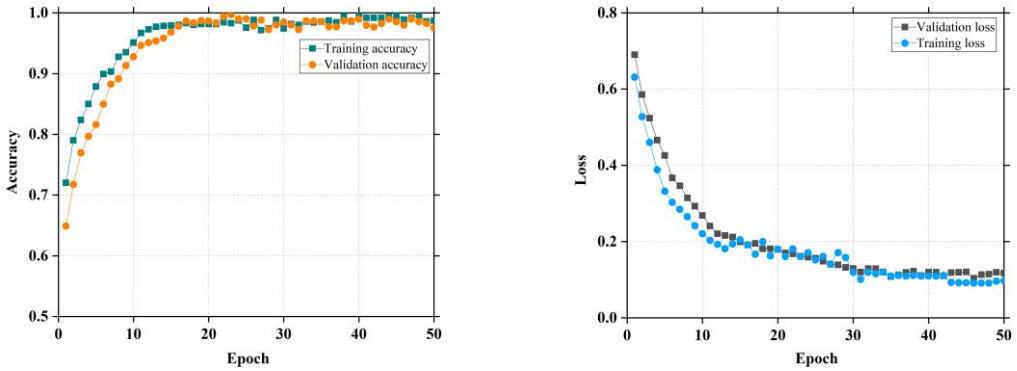


Figure 13. the accuracy and loss curve of WBI and IWTF combination during the training process.

4.2. Comparison of different interpolation methods under different feature fusion methods

This section evaluates three interpolation and four feature fusion methods, which are cross-combined, and each combination is applied to the proposed dual-CNN model fusion framework. The fused feature map is input into the classifier block for classification. Under the same experimental conditions, the performance of the models for each combination is compared in the training and test set.

We use CA and CEL during the training process to analyze the model's performance under different combinations, with the specific data shown in Table 3. The latter table shows the CA and CEL when the optimal model is obtained after five cross-validations of different combinations during the training process. Figure 14 shows the change trend curve of CA and CEL when the optimal model is obtained after cross-validating different combinations during the training process. Table 3 and

Figure 13 highlight that the combination of weighted bilinear interpolation and improved wavelet transform works best in the training set. The CA of the model in the training and validation data sets are 0.983 and 0.971, respectively, and the CEL is 0.097 and 0.116, respectively.

Table 3. Performance comparison of different interpolation and different feature fusion method combinations during the training process.

Combinations	Accuracy		Loss	
	Training	Validation	Training	Validation
BI+SF	0.911	0.910	0.382	0.326
BI+MF	0.917	0.927	0.278	0.266
BI+WTF	0.943	0.938	0.147	0.161
BI+IWTF	0.961	0.952	0.121	0.119
WBI+ SF	0.939	0.935	0.174	0.181
WBI+ MF	0.920	0.905	0.325	0.297
WBI+ WTF	0.956	0.951	0.129	0.132
WBI+ IWTF	**0.983**	**0.971**	**0.097**	**0.116**
BCI+ SF	0.935	0.928	0.212	0.262
BCI+ MF	0.916	0.909	0.316	0.299
BCI+ WTF	0.953	0.948	0.136	0.144
BCI+ IWTF	0.968	0.957	0.117	0.123

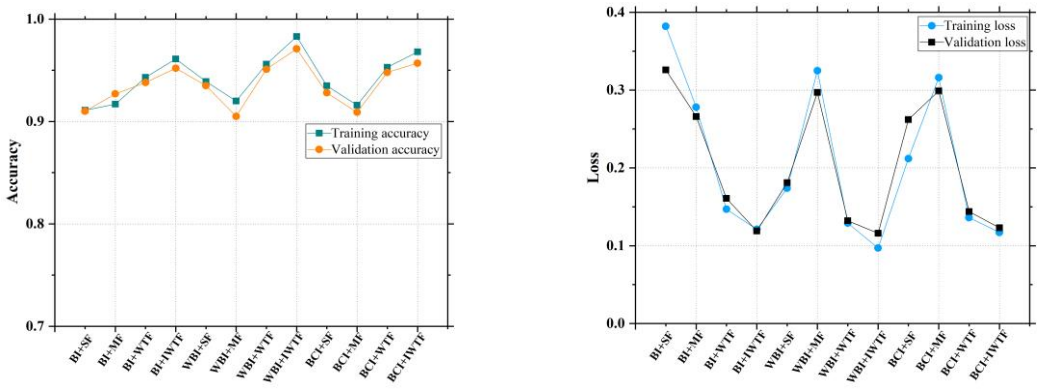


Figure 14. Comparison of training accuracy of different interpolation and different feature fusion method combinations.

Comparing the Test accuracy of various combinations in the test set, Figure 15 highlights that the combination of the weighted bilinear interpolation and improved wavelet transform works best in the test set, managing a Test accuracy of 0.951. Analyzing the experimental effects on the various combinations examined indicates that the improved wavelet transform combined with any interpolation method affords better performance than any other feature fusion method. Nevertheless, in most cases, the weighted bilinear interpolation has higher experimental accuracy.

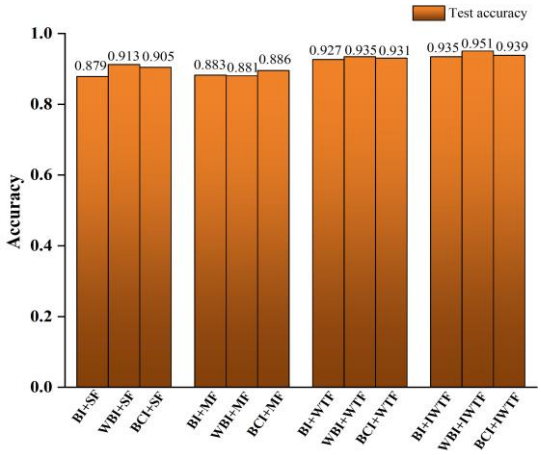


Figure 15. Comparison of test accuracy of different interpolation and different feature fusion method combinations.

4.3. Comparison of the proposed fusion frameworks with other fusion frameworks

This section challenges the proposed dual-CNN model fusion framework against the other two fusion frameworks presented in Section 2. All feature fusion methods are applied under the same experimental conditions as mentioned in the previous trials. Figure 16 presents the test accuracy of the various fusion frameworks. Among them, the proposed dual-CNN model fusion framework and the other two frameworks presented in Section 2 have the highest accuracy when combined with the improved wavelet transform fusion strategy, achieving an accuracy of 0.951, 0.944, and 0.939, respectively. For the data set utilized in this paper, we also compare the test accuracy of the three fusion frameworks under the same fusion method. In most cases, the proposed dual-CNN model fusion framework manages a higher accuracy than the other two fusion frameworks.

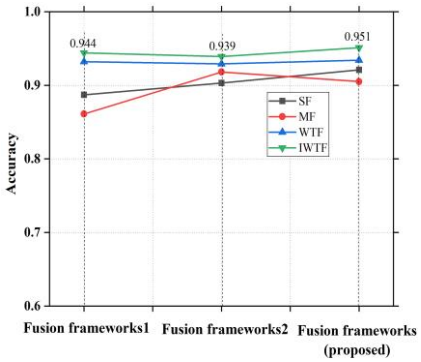
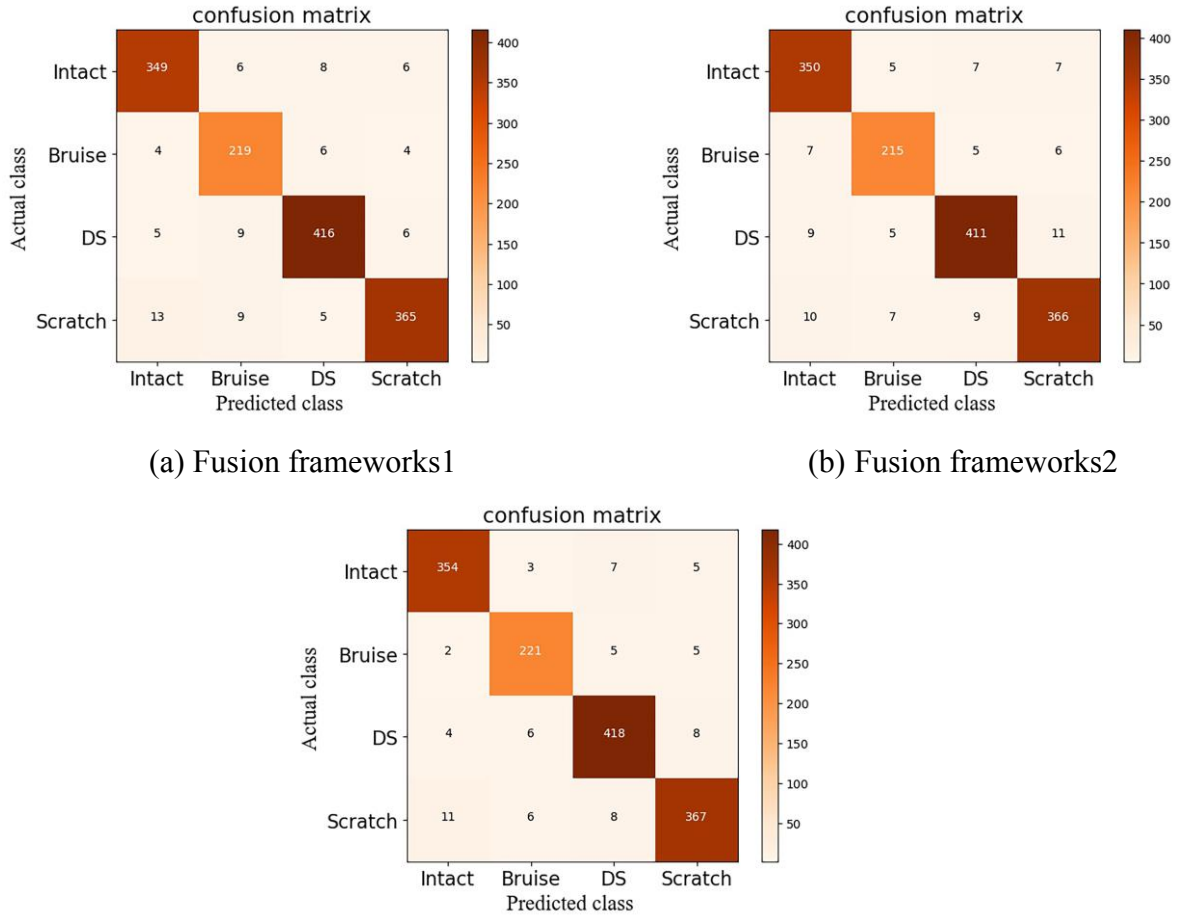


Figure 16. Comparison of test accuracy of different fusion frameworks.

Each column of the confusion matrix represents the predicted category, and the total number of data per column is displayed as the predicted number of that category. Each row represents the true attribution category, with the total number of data per row representing the number of that category. The value in each column shows the actual amount of data predicted for that type. We evaluate the three fusion frameworks (the total number of images is 1430) employing the improved wavelet

transform fusion strategy on the test set. The latter set involves 369 images of the Intact category, 233 of the Bruise category, 436 of the DS class, and 392 of the Scratch class. Figure 17 depicts the difference in the confusion matrix under the three fusion frameworks. From Figure 17, we find that some scratches are easily misclassified as Intact. The reason may be that the scratches on the surface of the aluminum profile are not noticeable. Comparing the three confusion matrices of Figure 17, it is evident that the test accuracy of our proposed dual-CNN model fusion framework is higher than the other two fusion frameworks.



(a) Fusion frameworks1

(b) Fusion frameworks2

(c) Fusion frameworks(proposed)

Figure 17. Confusion matrix under different fusion frameworks.

Table 4 presents the accuracy rate (ACC), average PPV, average TPR, and average F-score of the three different fusion frameworks combined with the improved wavelet transform fusion strategy. The average PPV, average TPR, and average F-score represent the corresponding average metric over all categories. According to Table 4, the accuracy of our proposed architecture is higher than the other two modular fusion frameworks. The accuracy rate is 0.951, the average PPV is 0.949, the average TPR is 0.950, and the average F-score is 0.949.

Table 4. Average statistical parameters of various fusion frameworks.

method	Acc(%)	AP(%)	AT(%)	Af(%)
Fusion frameworks1	94.4	93.8	94.3	94.1
Fusion frameworks2	93.9	93.5	93.3	93.6
Fusion frameworks(proposed)	**95.1**	**94.9**	**95.0**	**94.9**

(Acc:Accuracy AP: Average PPV AT: Average TPR Af:Average F-score)

As a recap, Figures 1, 2 and 3, present the three competitor fusion architectures. In the proposed scheme (illustrated in Figure 3), we select the feature map of the largest pooling layer of the Alexnet model A4 and VGG16 model V5 for feature fusion. Then, we use global average pooling instead of the dense layer to build a classifier block affording fewer training parameters and reduce the model's space dimensionality to avoid over-fitting and improve classification accuracy. The performance difference between our method and the first fusion framework is mainly due to the different feature fusion positions and classifier blocks. Part of the features fused by the second fusion framework originates from the processed image, i.e., gradient processing.

Figure 18 illustrates the accuracy metrics between the dual-CNN model fusion framework and the single convolutional neural network framework. The latter figure indicates that the experimental accuracy of the dual-neural network after feature fusion is higher than that of the single neural network. Among them, the test accuracy of Alexnet is 0.908, and of VGG16 is 0.926. After feature fusion, the performance of the two convolutional neural networks is better, managing a test accuracy rate of 0.951, which is 0.043 higher than solely using the Alexnet model and 0.025 higher than the VGG16 model. Comparing the three single-convolutional neural network frameworks and the other two traditional dual-CNN model fusion frameworks, the experimental accuracy of our dual-CNN model fusion framework has been improved.

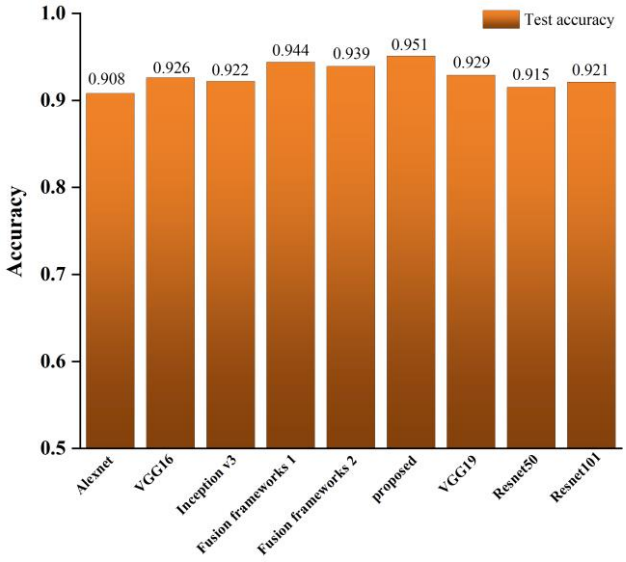
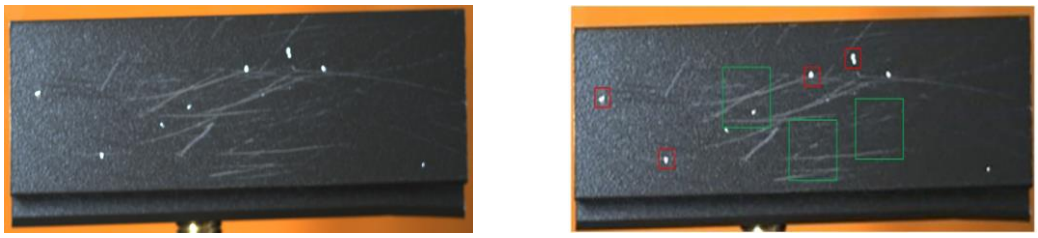


Figure 18. Comparison of test accuracy between dual-convolutional neural network model fusion framework and single convolutional neural network framework.

4.4. Insufficient research and future work

The suggested dual-CNN model fusion framework affects the surface defect recognition and classification of aluminum profile workpieces. It is important to note that the achieved experimental accuracy meets the requirements. However, it is limited to the identification and classification of a single defect. When multiple defects with inconsistent sizes on the aluminum profile workpiece surface exist, the dual-CNN model fusion framework will classify it according to the learned defect feature ratio and take the largest ratio as the classification output. This may lead to incorrect classification results. Figure 19(a) shows the classification result when multiple defects exist. In this example, the scratches on the surface of the workpiece are evenly distributed, and the size is larger than the dirty spots, so the scratches are output as the final classification.

Future work should also include other single-convolutional neural network frameworks for feature fusion to form a multi-convolutional neural network fusion framework. This strategy mainly realizes simultaneous recognition of multiple defects and defect marking positions to segment and highlight the defective parts. Figure 19(b) shows the location of multiple defects on the surface of the workpiece. The red squares represent dirty spots, and the green ones represent scratches.



(a) Actual:(DS+Scratch) Predicted:Scratch (b) Defect identification and location

Figure 19. Multiple defect classification and location.

5. Conclusions

This work considers aluminum defect detection and classification. Specifically, we propose an improved dual-CNN model fusion framework to extract different features of the same input source exploiting the pre-trained VGG16 and Alexnet models. Weighted bilinear interpolation ensures that the feature map generated by the last maximum pooling layer of the Alexnet and VGG16 models have the same dimensions. The improved wavelet feature fusion strategy is exploited to fuse feature maps effectively, while global average pooling replaces the dense layer to construct the classification block, i.e., classify and recognize the aluminum profile's surface defects.

Additionally, we analyze the structure of the conventional dual-CNN model fusion framework and the hidden layers' role. We also challenge several traditional upsampling methods combined with feature fusion strategies and select a set of optimal combinations (improved bilinear interpolation and improved wavelet transform fusion) as the configuration of the framework proposed in this paper. During the experiments, data augmentation and transfer learning methods are employed to prevent overfitting, and the K cross-validation method is used to evaluate the performance of the experimental model during the training process. Finally, we challenge the proposed framework against traditional dual-CNN model fusion frameworks and single-convolutional neural networks under the same

experimental conditions. Among them, the classification accuracy of our framework on the test set is 0.951, while the two conventional dual-CNN model fusion frameworks are 0.944 and 0.939, VGG16 manages 0.926, Alexnet 0.908, Inceptionv3 0.922, VGG19 0.929, Resnet50 0.915, and Resnet101 0.921. The experimental results highlight the contribution of exploiting an improved wavelet fusion strategy to achieve feature fusion after the maximum pooling layer of the two models. Additionally, the experimental results indicate the effectiveness of employing global average pooling instead of a dense layer to build the classifier block.

Acknowledgments

The authors would like to express their gratitude to EditSprings (https://www.editsprings.com/) for the expert linguistic services provided. The authors acknowledge the support from the National Key R&D Program of China (Grant No. 2017YFB1300700).

Conflict of interest

The authors declared that they have no conflicts of interest in this work.

References

1. Z. W. Liu, L. X. Li, J. Yi, S. K. Li, Z. H. Wang, G. Wang, Influence of heat treatment conditions on bending characteristics of 6063 aluminum alloy sheets, *T. Nonferr. Metal. Soc.*, **27** (2017), 1498–1506. doi: 10.1016/s1003-6326(17)60170-5.
2. S. Bingol, A. Bozaci, Experimental and Numerical Study on the Strength of Aluminum Extrusion Welding, *Materials (Basel)*, **8** (2015), 4389-4399. doi: 10.3390/ma8074389.
3. L. Donati , L. Tomesani, The effect of die design on the production and seam weld quality of extruded aluminum profiles, *J. Mater. Process. Technol.*, **164-165** (2005), 1025–1031. doi: 10.1016/j.jmatprotec.2005.02.156.
4. C. T. Mgonja, A review on effects of hazards in foundries to workers and environment, *IJISET: Int. J. Innov. Sci. Eng. Technol.*, **4** (2017), 326–334.
5. J. Ahmed, B. Gao, W. l. Woo, Sparse low-rank tensor decomposition for metal defect detection using thermographic imaging diagnostics, *IEEE T. Ind. Inform.,* **17** (2020), 1810–1820. doi: 10.1109/TII.2020.2994227.
6. Q. Luo, B. Gao, W. l. Woo, Y. Yang, Temporal and spatial deep learning network for infrared thermal defect detection, *NDT & E. Int.*, **108** (2019), 102164. doi: 10.1016/j.ndteint.2019.102164.
7. B. Z. Hu, B. Gao, W. l. Woo, L. F. Ruan, J. K. Jin, A Lightweight Spatial and Temporal Multi-Feature Fusion Network for Defect Detection, *IEEE T. Image Process.*, **30** (2020), 472–486. doi: 10.1109/TIP.2020.3036770.
8. J. Ahmed, B. Gao, W. l. Woo, Y. Zhu, Ensemble Joint Sparse Low-Rank Matrix Decomposition for Thermography Diagnosis System, *IEEE T. Ind. Electronics*, **68** (2020), 2648–2658. doi: 10.1109/TIE.2020.2975484.
9. J. Sun, C. Li, X. J. Wu, V. Palade, W. Fang, An effective method of weld defect detection and classification based on machine vision, *IEEE T. Ind. Inform.*, **15** (2019), 6322–6333. doi: 10.1109/TII.2019.2896357.

10. Z. F. Zhang, G. R. Wen, S. B. Chen, Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding, *J. Manuf. Process.*, **45** (2019), 208–216. Doi: 10.1016/j.jmapro.2019.06.023.
11. Y. Q. Bao, K. C. Song, J. Liu, Y. Y. Wang, Y. H. Yan, H. Yu, et al., Triplet-Graph Reasoning Network for Few-shot Metal Generic Surface Defect Segmentation, *IEEE Trans. Instrum. Meas.*, **70** (2021). doi: 10.1109/TIM.2021.3083561.
12. S. Fekri-Ershad, F. Tajeripour, Multi-resolution and noise-resistant surface defect detection approach using new version of local binary patterns, *Appl. Artif. Intell.*, **31** (2017), 395–410. doi: 10.1080/08839514.2017.1378012.
13. P. Y. Jong, C. S. Woosang, K. Gyogwon, S. K. Min, L. Chungki, J. L. Sang, Automated defect inspection system for metal surfaces based on deep learning and data augmentation, *J. Manuf. Syst.*, **55** (2020), 317–324. doi: 10.1016/j.jmsy.2020.03.009.
14. K. Ihor, M. Pavlo, B. Janette, B. Jakub, Steel surface defect classification using deep residual neural network, *Metals,* **10** (2020),846. doi: 10.3390/met10060846.
15. S. H. Guan, M. Lei, H. Lu, A steel surface defect recognition algorithm based on improved deep learning network model using feature visualization and quality evaluation, *IEEE Access,* **8** (2020), 49885–49895. doi: 10.1109/ACCESS.2020.2979755.
16. B. Zhang, M. M. Liu, Y. Z. Tian, G. Wu, X. H. Yang, S. Y. Shi, et al., Defect inspection system of nuclear fuel pellet end faces based on machine vision, *J. Nucl. Sci. Technol.*, **57** (2020), 617–623. doi: 10.1080/00223131.2019.1708827.
17. Z. H. Liu, H. B. Shi, X. F. Zhou, Aluminum Profile Type Recognition Based on Texture Features, *Appl. Mech. Mater.,* **556–562** (2014), 2846–2851. doi: 10.4028/www.scientific.net/AMM.556-562.2846.
18. A. Chondronasios, I. Popov, I, Jordanov., Feature selection for surface defect classification of extruded aluminum profiles, *Int. J. Adv. Manuf. Technol.*, **83** (2015), 33–41. doi: 10.1007/s00170-015-7514-3.
19. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90.
20. Q. H. Li, D. Liu, Aluminum Plate Surface Defects Classification Based on the BP Neural Network, *Appl. Mech. Mater.,* **734** (2015), 543–547. doi: 10.4028/www.scientific.net/AMM.734.543.
21. R. F. Wei, Y. B. Bi, Research on Recognition Technology of Aluminum Profile Surface Defects Based on Deep Learning, *Materials (Basel)*, **12** (2019), 1681. doi: 10.3390/ma12101681.
22. F. M. Neuhauser, G. Bachmann, P. Hora, Surface defect classification and detection on extruded aluminum profiles using convolutional neural networks, *Int. J. Mater. Form.*, **13** (2019), 591–603. doi: 10.1007/s12289-019-01496-1.
23. D. F. Zhang, K. C. Song, J. Xu, Y. He, Y. H. Yan, Unified detection method of aluminium profile surface defects: Common and rare defect categories, *Opt. Lasers Eng.,* **126** (2020), 105936. doi: 10.1016/j.optlaseng.2019.105936.
24. R. X. Chen, D. Y. Cai, X. L. Hu, Z. Zhan, S. Wang, Defect Detection Method of Aluminum Profile Surface Using Deep Self-Attention Mechanism under Hybrid Noise Conditions, *IEEE Trans. Instrum. Meas.,* (2021). doi: 10.1109/TIM.2021.3109723.
25. J. Liu, K. C. Song, M. Z. Feng, Y. H. Yan, Z. B. Tu, L. Liu, Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection, *Opt. Lasers Eng.,* **136** (2021), 106324. doi: 10.1016/j.optlaseng.2020.106324.

26. C. M. Duan, T. C. Zhang, Two-Stream Convolutional Neural Network Based on Gradient Image for Aluminum Profile Surface Defects Classification and Recognition, *IEEE Access*, **8** (2020), 172152-172165. doi: 10.1109/ACCESS.2020.3025165.
27. Y. L. Yu, F. X. Liu, A Two-Stream Deep Fusion Framework for High-Resolution Aerial Scene Classification, *Comput. Intell. Neurosci.*, **2018** (2018), 8639367. doi: 10.1155/2018/8639367.
28. C. Khraief, F. Benzarti, H. Amiri, Elderly fall detection based on multi-stream deep convolutional networks, *Multimed. Tools Appl.*, **79** (2020), 19537–19560. doi: 10.1007/s11042-020-08812-x.
29. W. Ye, J. Cheng, F. Yang, Y. Xu, Two-Stream Convolutional Network for Improving Activity Recognition Using Convolutional Long Short-Term Memory Networks, *IEEE Access*, **7** (2019), 67772–67780. doi: 10.1109/ACCESS.2019.2918808.
30. Q. S. Yan, D. Gong, Y. N. Zhang, Two-Stream Convolutional Networks for Blind Image Quality Assessment, *IEEE Trans. Image Process.*, **28** (2019), 2200–2211. doi: 10.1109/TIP.2018.2883741.
31. T. Zhang, H. Zhang, R. Wang, Y. D. Wu, A new JPEG image steganalysis technique combining rich model features and convolutional neural networks, *Math. Biosci. Eng.*, **16** (2019), 4069–4081. doi: 10.3934/mbe.2019201.
32. M. Uno, X. H. Han, Y. W. Chen, Comprehensive Study of Multiple CNNs Fusion for Fine-Grained Dog Breed Categorization, *2018 IEEE Int. Sym. Multim. (ISM)*, (2018), 198–203. doi: 10.1109/ISM.2018.000-7.
33. T. Akilan, Q. J. Wu, H. Zhang, Effect of fusing features from multiple DCNN architectures in image classification, *IET Image Process.*, **12** (2018), 1102–1110.
34. D. J. Li, H. T. Guo, B. M. Zhang, C. Zhao, D. H. Yu, Double vision full convolution network for object extraction in remote sensing imagery, *J. Image Graph.*, **25** (2020), 0535–0545.
35. M. Lin, Q. Chen, S. Yan, Network In Network, *arXiv preprint arXiv:*1312. 4400(2013).
36. K. M. He, X. Zhang, S. Q. Ren, J. Sun, Deep residual learning for image recognition, *Proc. IEEE confer. Computer vis. Pattern recognit.*, (2016), 770–778.
37. C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, *Proc. IEEE confer. Computer vis. Pattern recognit.*, (2015), 1–9.
38. K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv preprint arXiv*:1409. 1556 (2014).
39. Y. Lecun, Y. Bengio, Convolutional Networks for Images, Speech, and Time-Series, The *Handbook of Brain Theory & Neural Networks*, 3361 (10), 1995.
40. V. Suarez-Paniagua, I. Segura-Bedmar, Evaluation of pooling operations in convolutional architectures for drug-drug interaction extraction, *BMC Bioinformatics*, **19** (2018), 209. doi: 10.1186/s12859-018-2195-1.
41. X. L. Zhang, J. F. Xu, J. Yang, L. Chen, H. B. Zhou, X. J. Liu, et al., Understanding the learning mechanism of convolutional neural networks in spectral analysis, *Anal Chim Acta*, **1119** (2020), 41–51. doi: 10.1016/j.aca.2020.03.055.
42. S. W. Kwon, I. J. Choi, J. Y. Kang, W. I. Jang, G. H. Lee, M. C. Lee, Ultrasonographic Thyroid Nodule Classification Using a Deep Convolutional Neural Network with Surgical Pathology, *J. Digit. Imaging*, **33** (2020), 1202–1208. doi: 10.1007/s10278-020-00362-w.
43. G. E. Dahl, T. N. Sainath, G. E. Hinton, Improving deep neural networks for LVCSR using rectified linear units and dropout, *2013 IEEE Int. Conf. Acoustics, IEEE*, 2013. doi: 10.1109/ICASSP.2013.6639346.

44. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.,* **15** (2014), 1929–1958.
45. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *Int. Conf. Mach. Learn.*, *PMLR*, (2015), pp. 448–456.
46. V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, *lcml*, 2010.
47. P. Li, X. Liu, Bilinear interpolation method for quantum images based on quantum Fourier transform, *Int. J. Quantum Inf.*, **16** (2018), 1850031. doi: 10.1142/S0219749918500314.
48. D. Y. Han, Comparison of commonly used image interpolation methods, *Proc. 2nd Int. Conf. Comput. Sci. Electron. Eng. (ICCSEE 2013)*, 10 (2013).
49. X. Wang, X. Jia, W. Zhou, et al., Correction for color artifacts using the RGB intersection and the weighted bilinear interpolation, *Appl. Opt.*, **58** (2019), 8083–8091. doi: 10.1364/AO.58.008083.
50. J. F. Dou, Q. Qin, Z. M. Tu, Image fusion based on wavelet transform with genetic algorithms and human visual system, *Multimed. Tools Appl.*, **78** (2018), 12491–12517. doi: 10.1007/s11042-018-6756-0.
51. H. M. Lu, L. F. Zhang, S. Serikawa, Maximum local energy: An effective approach for multisensor image fusion in beyond wavelet transform domain, *Comput. Math. Appl.* **64** (2012), 996–1003. doi: 10.1016/j.camwa.2012.03.017.
52. B. Zhang, Study on image fusion based on different fusion rules of wavelet transform, *2010 3rd Int. Conf. Adv. Comput. Theo. Eng. (ICACTE)*, Vol. 3. IEEE, 2010. doi: 10.1109/ICACTE.2010.5579586.
53. S. L. Liu, Z. J. Song, M. N. Wang, WaveFuse: A Unified Deep Framework for Image Fusion with Discrete Wavelet Transform, *arXiv preprint arXiv*:2007. 14110(2020).
54. D. Kusumoto, M. Lachmann, T. Kunihiro, S. Yuasa, Y. Kishino, M. Kimura, et al., Automated Deep Learning-Based System to Identify Endothelial Cells Derived from Induced Pluripotent Stem Cells, *Stem Cell Rep.*, **10** (2018), 1687–1695. doi: 10.1016/j.stemcr.2018.04.007.
55. Su. P, Guo. S, Roys. S, F. Maier, H. Bhat, J. Zhuo, et al., Transcranial MR Imaging-Guided Focused Ultrasound Interventions Using Deep Learning Synthesized CT, *AJNR Am. J. Neuroradiol.*, **41** (2020), 1841–1848. doi: 10.3174/ajnr.A6758.
56. S. J. Pan, Q. Yang, A Survey on Transfer Learning, *IEEE Trans. Knowl. Data Eng.*, **22** (2010), 1345–1359. doi: 10.1109/TKDE.2009.191.
57. S. Medghalchi, C. F. Kusche, E. Karimi, U. Kerzel, S. K. Kerzel, et al., Damage Analysis in Dual-Phase Steel Using Deep Learning: Transfer from Uniaxial to Biaxial Straining Conditions by Image Data Augmentation, *JOM*, **72** (2020), 4420–4430. doi: 10.1007/s11837-020-04404-0.
58. X. R. Yu, X. M. Wu, C. B. Luo, P. Ren, Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework, *GISci. Remote Sens.*, **54** (2017), 741–758. doi: 10.1080/15481603.2017.1323377.
59. A. Taheri-Garavand, H. Ahmadi, M. Omid, S. S. Mohtasebi, K. Mollazade, G. M. Carlomagno, et al., An intelligent approach for cooling radiator fault diagnosis based on infrared thermal image processing technique, *Appl. Therm. Eng.*, **87** (2015), 434–443. doi: 10.1016/j.applthermaleng.2015.05.038.
60. M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, C. Pal, The Importance of Skip Connections in Biomedical Image Segmentation, *Deep learning and data labeling for medical applications,* Springer, Cham, 2016. 179–187. doi: 10.1007/978-3-319-46976-8_19.

61. Y-Lan. Boureau, Bach. F, Y. LeCun, Ponce. J, Learning mid-level features for recognition, *2010 IEEE Computer Society Conf. Comput. Vis. Pattern Recognit., IEEE*, (2010), 2559–2566. doi: 10.1109/CVPR.2010.5539963.

AIMS Press