Applied Intelligence (2022) 52:8243–8258
https://doi.org/10.1007/s10489-021-02917-y

Defect classification on limited labeled samples with multiscale feature fusion and semi-supervised learning

Jiahuan Liu[1] · Fei Guo[1] · Yun Zhang[1] · Binkui Hou[1] · Huamin Zhou[1]

Accepted: 11 October 2021 / Published online: 24 October 2021

Abstract
Defect inspection is an essential part of ensuring the quality of industrial products. Deep learning has achieved great success in defect inspection when a large number of labeled samples are available. However, it is infeasible to collect and label numerous samples in many manufacturing processes. Meanwhile, deep learning methods cannot conform to the high defect recognition accuracy of strict production requirements when the labeled samples are scarce but varied. This paper proposed a novel convolutional neural network architecture and a semi-supervised learning strategy using soft pseudo labels and a mutual correction classifier to improve the defect inspection accuracy when labeled samples are scarce. The effectiveness of the proposed method is verified on a famous industrial defect inspection benchmark dataset and a practical dataset containing images collected from actual injection molding production lines. The results indicate that the proposed method achieves an accuracy of 99.03% on the benchmark defect dataset, which is approximately 13.2% higher than other methods when the training dataset contains only 45 labeled images and 135 unlabeled samples per category. The best accuracy on the benchmark dataset obtained by the proposed method reaches 99.72%. Besides, an average accuracy of 99.25% is achieved with only 20 labeled samples and 180 unlabeled samples per category in the practical defect inspection task. Visualization methods prove that the performance improvement comes from the proposed multiscale architecture and the semi-supervised learning strategy. The proposed method can be used in practical defect inspection applications of industrial manufacturing, such as steel rolling, welding, and injection molding.

Keywords Defect inspection · Semi-supervised learning · Multiscale feature · Convolutional neural network

1 Introduction

High-accuracy defect inspection is a common challenge in various industrial manufacturing industries [1, 2]. Affected by the fluctuation of process parameters and the production environment, different types of defects inevitably exist on product surfaces, such as scratches and crazing on a steel surface [3, 4] and burrs on a plastic product surface [5]. These defects can affect product performance and aesthetics, causing considerable economic losses. Conventional manual visual inspection is usually affected by subjective factors, which results in low accuracy and reliability. Therefore, automatic surface inspection (ASI) methods are increasingly widely used to improve the accuracy of defect inspection.

Existing ASI methods [6, 7] are mainly based on feature engineering, where the image features are hand-crafted and designed by experienced practitioners. These methods are highly dependent on the engineers, leading to poor robustness and low accuracy. As machine learning methods have achieved remarkable performance in many classification tasks [8, 9], many studies have been done to apply convolutional neural networks (CNNs) to defect inspection tasks where a large number of labeled samples are available [10–12]. However, it is extremely time-consuming and labor-intensive to collect and label enough samples manually for deep learning model training in industrial manufacturing. Therefore, it is a great challenge to use a few manually labeled samples to achieve sufficiently high defect inspection accuracy to meet the manufacturing requirements.

Semi-supervised learning (SSL) methods that can utilize both labeled samples and unlabeled samples for model

✉ Binkui Hou
houbinkui@hust.edu.cn

[1] State Key Laboratory of Materials Processing and Die & Mould Technology, School of Materials Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

training have been widely adopted to improve the model performance in many classification tasks [13, 14]. The first method is to assign class labels for unlabeled samples in some way (known as pseudo labels). Then, the unlabeled samples combined with pseudo labels are included in model training [15, 16]. Hao Wu et al. [15] proposed to predict labels for unlabeled hyperspectral images with clustering methods first. Then, these unlabeled images and the original labeled images are all used to train deep convolutional recurrent neural networks for classification. This method obtained better performance in the pixel-wise classification task of hyperspectral images in real datasets. However, although performance improvement was obtained, the overall classification accuracy was not high enough because the learned image features of models were damaged once numerous unlabeled samples were wrongly labeled by clustering methods. Another method is to generate more samples with a few labeled samples using a generative adversarial network (GAN). In practice, Yu He et al. [17] adopted a GAN to generate samples using labeled samples and used the discriminator of the GAN to predict class labels for these generated unlabeled samples. In this way, they obtained a more than 26% accuracy increase in solving the problem of steel surface defect detection, achieving an accuracy of 96.06% in the case of 90 labeled samples and 90 unlabeled samples in the training datasets. Similarly, when labeled samples were insufficient, the diversity of generated samples was not enough, leading to over-fitting classification models and decreased recognition accuracy. Finally, a third method trains a convolutional autoencoder (CAE) [18] or a GAN [19] with all the training samples for image feature extraction. Then the encoder of the CAE or the discriminator of the GAN serves as a classifier for defect inspection tasks trained on some labeled samples. He Di et al. [3]. proposed training a CAE on unlabeled samples, and the encoder was taken as the discriminator of a subsequent GAN. Using 21,000 unlabeled samples from other similar tasks training the CAE, an accuracy of 96.7% was obtained in defect detection of cold-rolled strips.

Although these existing methods have achieved great performance, their accuracy still cannot meet the strict requirements of some high-precision manufacturing processes requiring more than 99% accuracy. In addition, because of workload limits, collecting and labeling many valid training samples with correct category labels are difficult and sometimes infeasible during steel rolling, welding, injection molding, and other processes. Thus, the labeled samples available for recognition model training are very scarce [3], whereas many unlabeled samples are idle and cannot be used effectively. Moreover, the appearances of the defective products change significantly due to the fluctuation of the process parameters, leading to great intra-class differences [4] in the training datasets. The intra-class difference means that the appearances of samples in the same category are very different and that some samples are even similar to samples of other categories. Since these samples located at the boundary of the two categories are difficult to distinguish, the intra-class difference increases the difficulty of the defect recognition task. In summary, the major challenge in the industrial defect inspection task is to achieve high accuracy of more than 99% with a few labeled samples when the defective samples are diverse and hard to recognize.

To address this challenge, we propose a new semi-supervised learning strategy and a novel multiscale convolutional neural network (MS-CNN) architecture to improve the accuracy of defect recognition. In this method, an MS-CNN architecture is proposed to extract multiscale image semantic features to adapt to the defects of different sizes in the input images. The stronger representation capability provided by the multiscale features improves the ability of the models to distinguish similar samples of different categories. In addition, a method of soft pseudo labels combined with threshold selection is proposed to improve the utilization efficiency of unlabeled samples in the semi-supervised learning strategy. Moreover, a mutual correction classification module where two independent classifiers mutually correct their respective predicted pseudo labels is proposed to improve the accuracy of the pseudo labels and reduce the damage of noisy pseudo labels to the features learned by the models. Meanwhile, data augmentation techniques are applied to increase the number of labeled samples to further improve the recognition accuracy. The effectiveness and performance improvement of the proposed method are verified on a general defect classification dataset and a practical defect recognition dataset of an injection molding product.

The remainder of this paper is organized as follows. Section 2 introduces the proposed methodology. Section 3 describes the datasets and experiments design. And the results of the experiments are discussed in Section 4. At last, conclusions are summarized in Section 5.

2 Methodology

The details of the proposed semi-supervised defect recognition method using MS-CNN architecture and mutual correction classifier are introduced in this section. The content includes multiscale feature extraction architecture, pseudo labels generation method, mutual correction classifier module, data augmentation method, and model training details.

2.1 Multiscale CNN architecture

To improve the discrimination of image features extracted by CNN models, a multiscale CNN architecture that combines image features of different scales is proposed in this study. Common CNN architectures, such as VGG-16 [20], have just one pathway from the input images to the output results. Then, the predictions are only based on the image features of the last convolutional layer. When the defects in images are subtle, or samples of different categories are similar, the features extracted by a single pathway architecture lose much discriminative information due to the continuous down-sampling of pooling layers, resulting in insufficient accuracy. Therefore, motivated by some object detection frameworks [21, 22], we propose a novel multiscale CNN architecture that can extract sophisticated image features. Specifically, we add two extra lateral connections to the main convolutional pathway of a CNN architecture to combine image features extracted by layers of different semantic levels. The sketch of the proposed MS-CNN is shown in Fig. 1.

As shown in Fig. 1, the proposed MS-CNN architecture has two extra lateral connections at the end of the third and the fourth convolutional layers (marked as Conv3 and Conv4, respectively). These lateral connections merge feature maps of layers Conv3 and Conv4 to produce two feature maps with different receptive fields. Then, the output feature maps obtained by the Conv5 layer are combined with those of two lateral connections using channel-wise addition. Thus, these image features extracted by different convolutional layers have complicated semantics at different levels only built from a single input image scale. Besides, as in Fig. 1, the fusion of these features is performed by channel-wise addition operation, which is defined in Eq. (1):

$$R_{i,j,k} = M^{L_1}_{i,j,k} + M^{L_2}_{i,j,k} + M^{L_3}_{i,j,k}, \tag{1}$$

where M is the feature map matrixes of different layers, L_1, $L_{2,}$ and L_3 denote the different layers, subscript i and j mean row number and column number of matrix M, respectively. R is the results of feature fusion, k is the number of channels of different feature maps (e.g., 512 of Conv3 in Fig. 1).

The image samples of industrial products are mostly grayscale images that contain fewer information and features than natural images. Therefore, the significant features of small objects in industrial images may disappear after spatial resolution down-sampling of the feature maps, leading to the confusion of similar samples. However, the proposed MS-CNN architecture preserves the salient features of subtle areas by connecting the low-level semantic features to the high-level semantic features. Thus, the MS-CNN models can learn features of different scale defect areas from a single scale input image simultaneously, improving the information richness of image features and the discrimination of the models.

As shown in Fig. 1, although we add two extra lateral connections, we reduce the number of neurons in the fully connected layer. Therefore, the total parameters are only

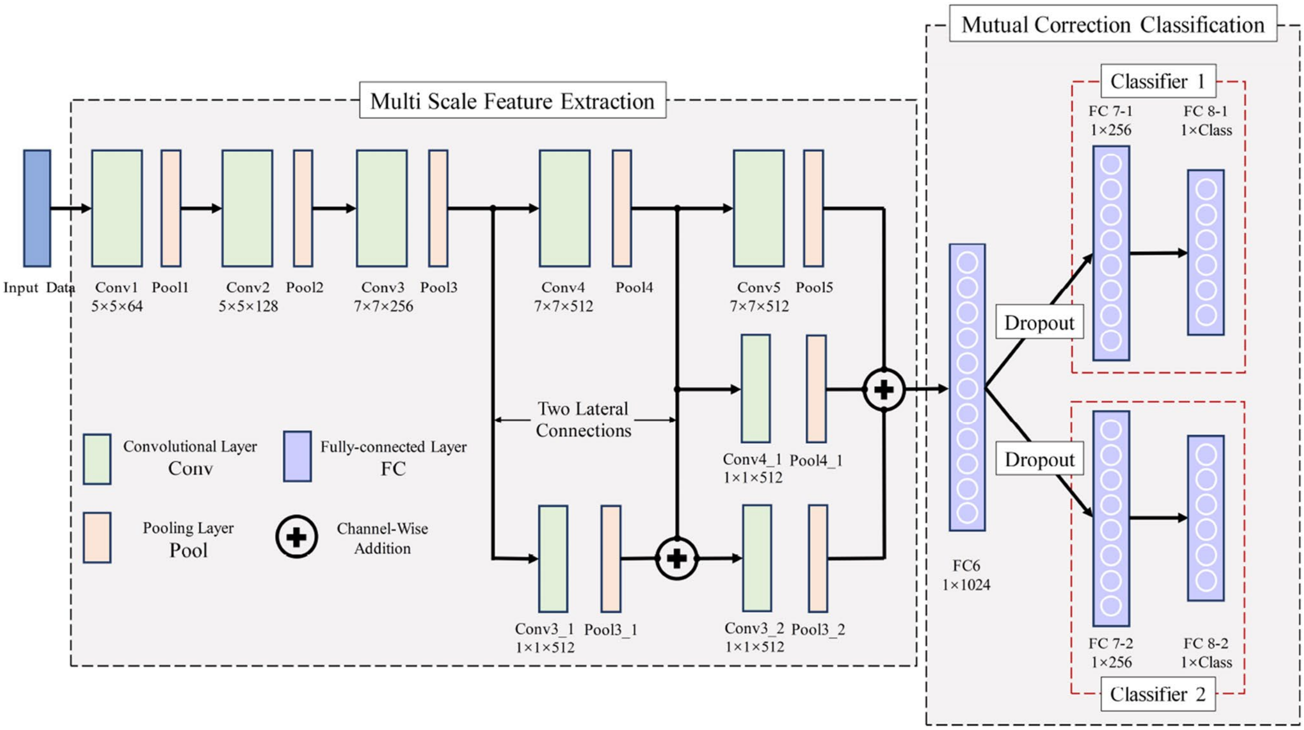


Fig. 1 Sketch of the proposed multiscale CNN

approximately 24.3% of the original VGG16 architecture, reducing the calculation time and power consumption in the actual application. In addition, the neurons in CNNs work continuously, and the power consumption of computing hardware is still high. Therefore, some research [23, 24] uses a spiking neural network (SNN) to reduce the active time of neurons to reduce power consumption.

2.2 Semi-supervised learning method

A new semi-supervised training strategy based on pseudo labels is proposed to utilize a large number of unlabeled samples during model training. Specifically, before a batch of unlabeled samples is used for the next step model training, these samples are first fed into the model obtained in the last training step, and the predicted results of the previous model are regarded as pseudo labels of these samples. The illustration of the proposed semi-supervised learning strategy is shown in Fig. 2.

As shown in Fig. 2, the unlabeled samples in a batch of training samples are first fed into the model obtained in training step ***s*-1**. Subsequently, the model predicts category probability for these samples in inference mode. These predicted results are utilized as pseudo labels of these unlabeled samples. Then, the unlabeled samples with high prediction confidence are selected for next-step model training together with all the original labeled samples. The prediction of pseudo labels and the training of the model iterate alternately until the model converges.

Common pseudo label methods, such as the study in Ref. [25], mostly use "hard label" as supervision information, which is a one-hot form and consists of only 0 and 1. However, these "hard labels" discard the information of categories with low predicted probability, resulting in an insufficient representation capacity of correlation/ambiguity among different samples [26]. In addition, when the pseudo labels are incorrect, the overconfidence of "hard label" may lead to a degradation in generalization ability. Therefore, this study proposes to use the predicted class probability of the model obtained in the last training step as the pseudo label, which is the so-called "soft label." Compared to "hard label," "soft label" is similar to label smoothing strategy and preserves much more valuable information because category information with small probability may imply similar features among the input images, especially when these images are on the decision boundary of two similar categories. In addition, to prevent incorrect pseudo labels from harming the learned image features, a threshold mechanism is adopted to select the unlabeled samples with high prediction confidence to calculate the final loss function.

A new loss function is proposed to implement the proposed semi-supervised learning strategy. We compute the cross-entropy loss function for labeled samples and unlabeled samples, and the new loss function is a linear combination of them. The cross-entropy loss function used in this study is defined in Eq. (2):

$$Loss_{total} = \alpha * Loss_{Label} + \beta * Loss_{unLabel}, \tag{2}$$

where α and β are the weights of the loss of labeled samples and unlabeled samples.

In training step $\boldsymbol{s}$, the targets of the labeled samples are true manual labels, and the targets of the unlabeled samples are "soft labels" predicted by models obtained in the last training step ***s*-1**. Thus, the loss values of labeled samples are computed as Eq. (3):

$$Loss_{Label} = -\sum_{i=1}^{N^L}\sum_{j=1}^{C} y_{ij}\log(p_{ij}^L), \tag{3}$$

where N^L is the number of labeled samples in the training datasets, C is the number of categories, y_{ij} indicates the true manual label of j-th class of i-th sample, p_{ij}^L is the output of j-th class of i-th labeled sample predicted by the model in

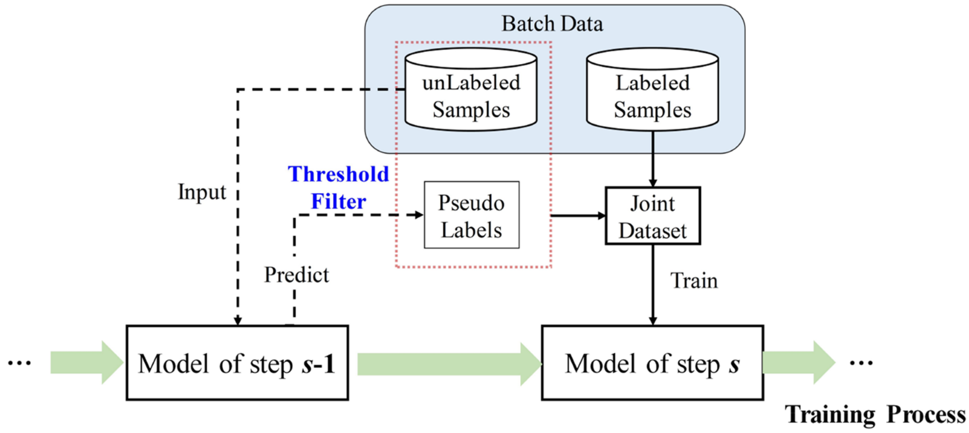


Fig. 2 Illustration of the proposed semi-supervised training strategy

training step s. Similarly, the loss values of unlabeled samples are computed as Eq. (4):

$$Loss_{unLabel} = -\sum_{i=1}^{N^U}\sum_{j=1}^{C}(p_{ij}^{last} > \tau)p_{ij}^{last}\log(p_{ij}^{U}), \tag{4}$$

where N^U is the number of unlabeled samples in the training datasets, C is the number of categories, p_{ij}^{last} is the soft label predicted by the model obtained in training step $\boldsymbol{s}$**-1**, p_{ij}^{U} is the output of j-th class of i-th unlabeled sample predicted by the model in training step $\boldsymbol{s}$. The item $\mathrm{I}\left(p_{ij}^{last} > \tau\right)$ acts as a mask to select unlabeled samples whose predicted pseudo labels have high confidence. The value of this mask item is 1 when the maximum of a predicted pseudo label is greater than a threshold τ (0.9 in this study); otherwise, it is 0.

2.3 Mutual correction classifier

When existing semi-supervised learning methods include unlabeled samples in model training, their labels are pseudo labels predicted by classification models themselves, and there is no human intervention to judge whether these labels are correct. These semi-supervised methods have no error correction measures to correct these pseudo labels when they make mistakes. If these incorrect pseudo labels are used as the supervision information of unlabeled samples, the models will be confused, and the learned image features will be harmed. Even if the loss values of the labeled samples are used as a penalty term to suppress the damage caused by these incorrect labels, the performance of the models will still be significantly affected, resulting in insufficient overall accuracy.

To overcome the challenges described above, this study proposes a mutual correction classifier to correct the pseudo labels that may be wrong. At the end of the MS-CNN architecture, we design two independent classifiers after the flattened image features to predict category probability results for input images independently. The dropout [27] mechanism introduced in the neural connections between the flatten image features and these classifiers can guarantee that the predictions made by the two classifiers are different to avoid symmetry in the results. The average values of the predictions of two classifiers are used as final outputs of the MS-CNN models. The schematic diagram of the working principle of the mutual correction classifier is shown in Fig. 3.

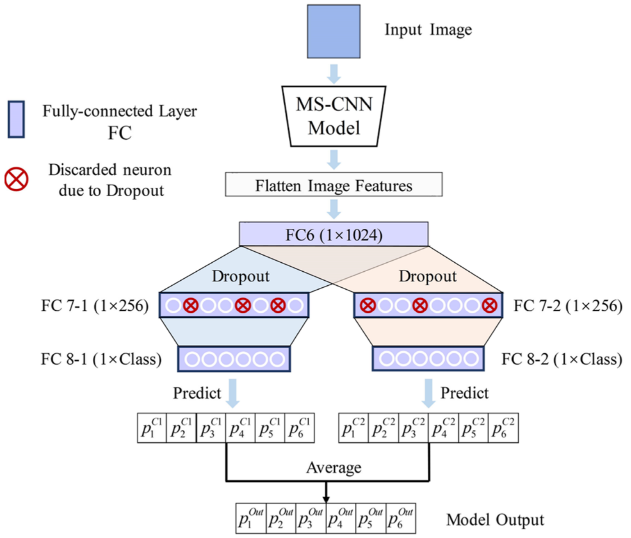


Fig. 3 Mechanism of proposed mutual correction classifier

As shown in Fig. 3, the working principle of the mutual correction mechanism is as follows. In the model training process, the average of the predicted probability results of two classifiers is used as the pseudo label of the unlabeled input images for the next model training step. When one of the classifiers makes an incorrect prediction while the other one is correct, the average of the two results as the final output can correct the incorrect prediction in the right direction, reducing the destructive impact of the incorrect results. Due to the use of soft labels, the incorrect prediction still preserves much category information, which helps the model learn image features. On the other hand, if both classifiers predict incorrectly, the loss of labeled samples in the loss function serves as a penalty term to force the model to relearn effective image features to minimize the total loss.

As described above, it can be found that this mutual correction mechanism fails when the pseudo labels are hard labels. For example, in a binary classification problem, one classifier (classifier 1) may predict category probability of [0.9, 0.1] while the other (classifier 2) gives a result of [0.8, 0.2]. Therefore, the final model output result, which is the average of these two predictions, is [0.55, 0.45], and the hard label is [1, 0]. This label is the same as the hard label of classifier 1, and classifier 2 is blocked and vice versa.

In general, the proposed mutual correction classifier can automatically correct the pseudo labels predicted by the models, improving the accuracy of these pseudo labels. Therefore, the proposed MS-CNN models can utilize a large number of unlabeled samples to learn image features and improve the performance of the models finally.

2.4 Data augmentation

Data augmentation methods [28, 29] are used to increase the number of labeled samples when the labeled samples are extremely scarce because CNN models usually need a large number of training samples to get good performance and prevent over-fitting. With the augmentation method, the number and the diversity of the labeled samples can be both increased. These additional samples force the MS-CNN models to learn more efficient and robust features. Therefore, the probability of incorrect prediction of pseudo labels of unlabeled samples can be reduced, and the proposed MS-CNN models have a stronger ability to against the instability caused by noisy labels when including unlabeled samples in model training. In addition, data augmentation can also be regarded as a regularization to reduce the dependence on some specific features [30].

Moreover, due to environmental interference, data distribution obtained during sample collection may differ from actual scenarios. When a well-trained CNN model is applied to practical inspection processes, illumination changes and electronic noise will cause a degradation of image quality, so the inspection model needs to be highly robust to these corruptions. Different data augmentation methods can increase the number of image samples and simulate image degradations. Thus, the robustness of inspection models can be improved when applied in actual defect inspection.

In this work, only random flip, random changes in brightness and contrast, and Gaussian noise injection are used to increase the number of labeled training samples. Considering the situations that may be encountered in actual defect inspection tasks, the reasons for selecting these augmentation methods are summarized in Table 1.

The degree of data augmentation varies with the number of labeled samples used for model training. Specifically, when there are more labeled samples, e.g., 50% of the whole datasets, these labeled samples are expanded approximately two times. Moreover, when there are fewer labeled samples, e.g., 10% of the whole datasets, these labeled samples are expanded approximately five times.

2.5 MS-CNN Model training

Since it is challenging to collect tens of thousands of training samples in industrial production, a model-based transfer learning method [31, 32] is adopted to help the models learn efficient image features and prevent over-fitting. Transferred parameters of pre-trained models have excellent representation capability applicable in new vision tasks such as defect inspection of industrial products. Thus, transfer learning methods can significantly reduce the required training samples and improve classification performance.

Table 1 Reason for selecting these augmentation techniques in this work

Methods	Reasons
Random Flip	Random placement of the products
Random brightness change	Illumination change in the environment
Gaussian noise injection	Electronic noise in image acquisition

Since the main pathway of the proposed MS-CNN architecture is similar to the typical VGG16 model, the convolutional parameters of a well-trained VGG16 model on the open-source ImageNet dataset [33] are transferred directly to our MS-CNN model. The proposed MS-CNN architecture contains five convolutional layers in the main convolutional pathway and two additional lateral connections between high-level convolutional layers. The parameters of the five convolutional layers in the main pathway are directly transferred from the VGG16 model. The parameters of the remaining convolutional layers and the fully connected layers are all initialized with a random Gaussian distribution. The transfer learning strategy used in this work is performed as Fig. 4.

In a deep CNN model, the low-level convolutional layers extract general image features that widely exist in both natural images and industrial images, while the high-level convolutional layers extract complicated image features specific to different tasks [34]. Therefore, the transferred parameters of the low-level convolutional layers remain unchanged, while the parameters of the high-level layers are fine-tuned during the training phases of the new MS-CNN models. Specifically, as shown in Fig. 4, the parameters in the Conv1 and Conv2 layers are transferred from the original VGG16 model and remain frozen while the parameters in Conv3-Conv5 are updated during the training phases. The remaining convolutional parameters in Conv3_1, Conv3_2, and Conv4_1 and the parameters of all the fully-connected layers are randomly initialized with a Gaussian distribution. Then these parameters are trained with different industrial datasets.

During MS-CNN training phases, some hyper-parameters should be selected carefully to achieve the best model performance, including learning rate, batch size, training epochs, and dropout probability. The choices of these hyper-parameters are summarized in Table 2. We briefly explain the reasons for the choices of these training hyper-parameters as follows. For learning rate, since our MS-CNN models are trained on small datasets with transfer learning methods, it is set to $1 \sim 3 \times 10^{-4}$ in this work empirically. Limited by the performance of GPU used in this work, we choose the maximal batch size of 32 in training phases. In addition, since there are three stages of semi-supervised training of the MS-CNN models, about 40 epochs are needed to guarantee the convergence of models. Finally, according to experimental results, we choose 0.7 as the keep probability of the dropout method to obtain the optimal models.

Moreover, since our loss function is a linear combination of labeled loss and unlabeled loss, balance factors of α and β are also considered hyper-parameters. The choice of balance factors is implemented by range search according to model performance.

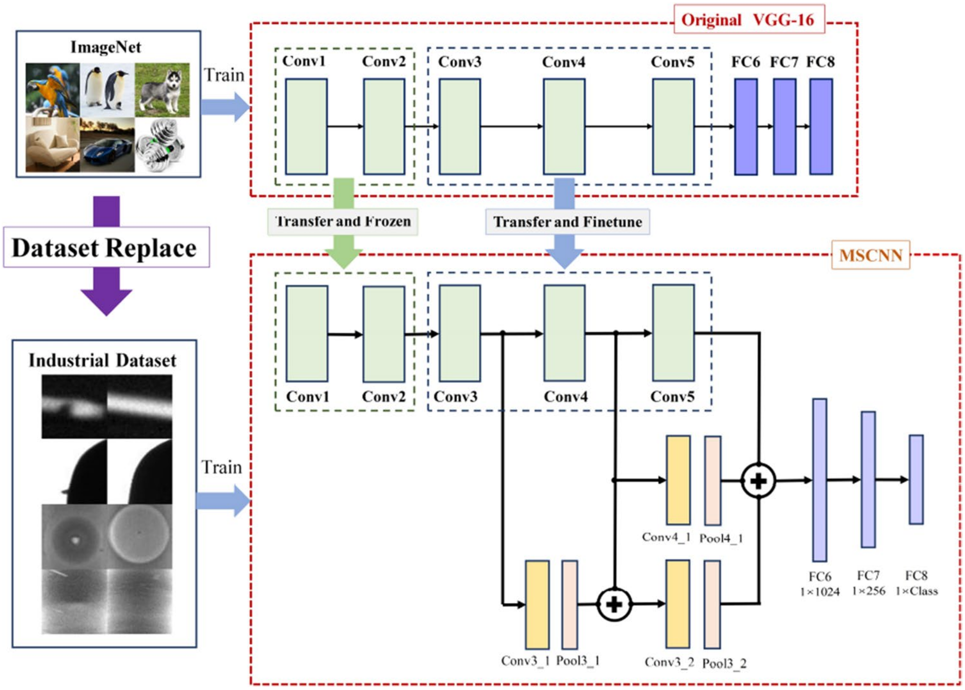


Fig. 4 Schematic diagram of transfer learning strategy applied in this paper

We summarize the overall training procedure of the proposed semi-supervised learning strategy in **Algorithm 1**. The training phase is separated into three stages, as mentioned before. In the first stage, only labeled samples contribute to the model loss function, and the weight factor β in Eq. (2) is set to 0 to disable the loss contribution of unlabeled samples. In the next stage, the training process is divided into two sub-stages. In the first sub-stage, the value of β increases linearly slowly with training steps. Unlabeled samples are included in the model training process and contribute to the total loss function in a slow-growth way to prevent these samples with incorrect pseudo labels from damaging learned features of the models. In the second sub-stage, the value of β increases faster, and the contribution of unlabeled samples to the total loss gets larger. In the third stage, the value of β remains unchanged for final model fine-tuning on all the training samples.

It should be noted that in all the three training stages, all the labeled samples and unlabeled samples in the training datasets are used for model training, and only the parameter β is different.

Algorithm 1 the proposed new semi-supervised training strategy based on pseudo labels

```
Initialize the MS-CNN model with transferred parameters and default Gaussian initializer.
Set values of $\alpha$ and $\beta$: $\alpha \leftarrow 1.0$, $\beta^* \leftarrow 0.5$; Set schedule $2 \times T_2 < T_3 + T_1$
for $t = 1$ to $T$, do
    // only labeled samples contribute to the total loss function
    if $t <= T_1$
        $\beta = 0$
    // gradually including unlabeled samples in model training, first slowly then rapidly
    if $T_1 < t <= T_2$
        $\beta = (t - T_1) / (T_2 - T_1) \times \beta^* \times 0.5$
    if $T_2 < t < T_3$
        $\beta = (t + T_3 - 2 \times T_2) / (T_3 - T_2) \times \beta^* \times 0.5$
    // finetuning the models
    if $t >= T_3$
        $\beta = \beta^*$
    //get pseudo labels of mini-batch of unlabeled samples
    Compute the class probability of samples in mini-batch with the model obtained in the last training step
    // in the first stage, unlabeled samples do not contribute to the loss values though they are computed
    Compute total loss in Equation (2) with a mini-batch of labeled samples and unlabeled samples.
    Performing backward propagation to get gradients and update the MS-CNN parameters.
end for
```

According to the study in Ref. [36], deep neural network models are content-aware and tend to learn patterns in the training samples first. If there is noise in the labels and the labels are not completely random, DNN models still try to learn features of the training samples with correct supervised information. Therefore, since the MS-CNN models have been trained on the labeled samples with correct manual labels, some of the unlabeled samples can be classified correctly. Then the models try to learn patterns and features from these unlabeled samples with correct pseudo labels gradually. This learning process is gradually strengthened with longer training time and larger contributions of

Table 2 Choices of training hyper-parameters used in this work

Parameters	Learning Rate	Optimizer	Batch Size	Epochs	Dropout
Choice	$1 \sim 3 \times 10^{-4}$	Adam [35]	32	40	0.7

unlabeled samples and finally improves the performance of the MS-CNN models.

3 Experiments setup

In this work, the proposed novel MS-CNN architecture and the new semi-supervised learning strategy based on pseudo labels are first verified on a surface defect recognition benchmark dataset, NEU. Then an injection molding product image dataset (marked as IMP) collected from an existing injection molding production line is used to illustrate practical applications of the proposed method.

3.1 Description of the datasets

The NEU dataset used in this work is generally used as a benchmark dataset of industrial defect detection tasks since it is challenging and representative [3, 17, 25, 37]. This dataset contains six kinds of typical surface defects of hot-rolled steel strips, i.e., rolled-in scale (Rs), patches (Pa), crazing (Cr), pitted surface (Ps), inclusion (In), and scratches (Sc). There are 300 grayscale images with a resolution of 200 × 200 pixels (width × height) for each class in the NEU dataset. Some image samples are shown in Fig. 5(a). As shown in Fig. 5(a), some image samples look similar, e.g., CR, PS, and RS. Thus, the main challenge of this dataset is that some inter-class defects have similar aspects, while some defects have huge intra-class differences due to the image sampling angles. For example, the scratch defects may be horizontal scratch, vertical scratch, and slanting scratch. Another challenge is that this dataset suffers from the influence of illumination and material changes.

The IMP dataset is collected from a white plastic injection molding product with dimensions of approximately 35 mm × 6 mm × 2.5 mm (length × width × height), and some image samples are shown in Fig. 5(c). There are four different kinds of defects on the upper and lower surfaces and four edges of this product: pits on the upper surface, pits on the side, burrs, and injection point residual. We designed an adjustable image acquisition platform to capture defect images with CCD cameras, as shown in Fig. 6(a). This platform was only used for the design and verification of the image acquisition scheme. In actual defect inspection tasks, the appropriate scheme obtained with this platform was deployed to the online inspection equipment. Then, as shown in Fig. 6(b), some defects appear in the fixed place of the products due to the defect generation mechanism and process factors, including pits on the upper surface, residuals on the lower surface, and burrs. However, pits on the front and back sides (angle of view in Fig. 6) occur at random on both the entire side surfaces. Thus, image patches that may contain defective areas are cropped out automatically by corresponding image preprocessing algorithms. Due to the different positions of the defects, the size of the defective image patches is also different. The image sample resolution of different defects is listed as follows: burr patches are 60 × 120 pixels (width × height, the same below), pit-up patches are 100 × 120 pixels, pit-side patches are 70 × 70 pixels, and residual patches are 70 × 70 pixels.

In addition, considering the defect types of the product used in this paper, input images with relatively small resolutions can also work well. However, in some cases with more complex defects and higher requirements, the resolutions of the input images need to be increased (millions of pixels), which will increase the amount of calculation and affect the real-time performance. In improving the real-time

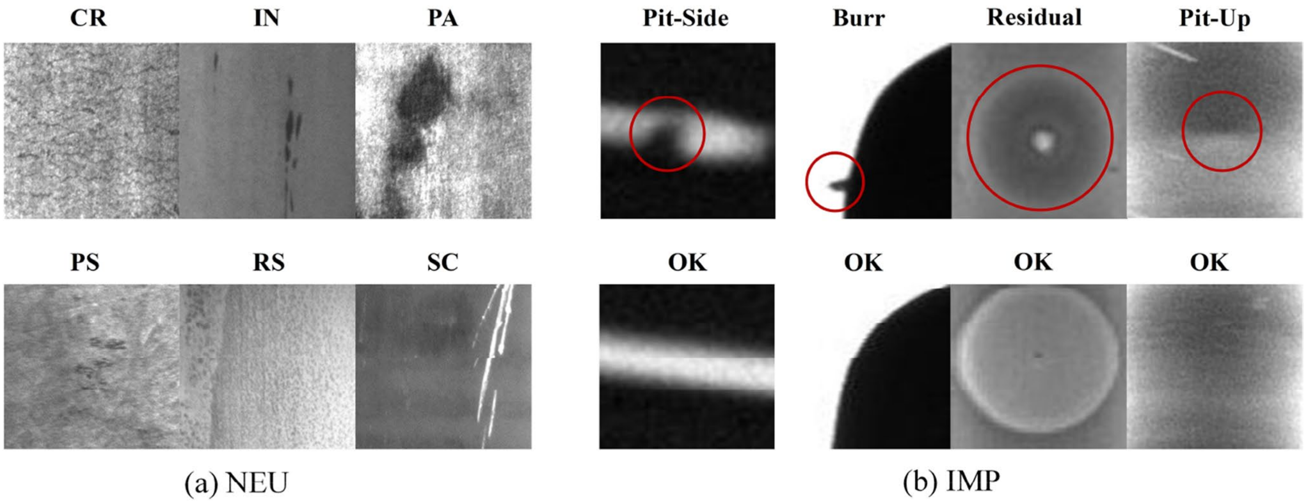
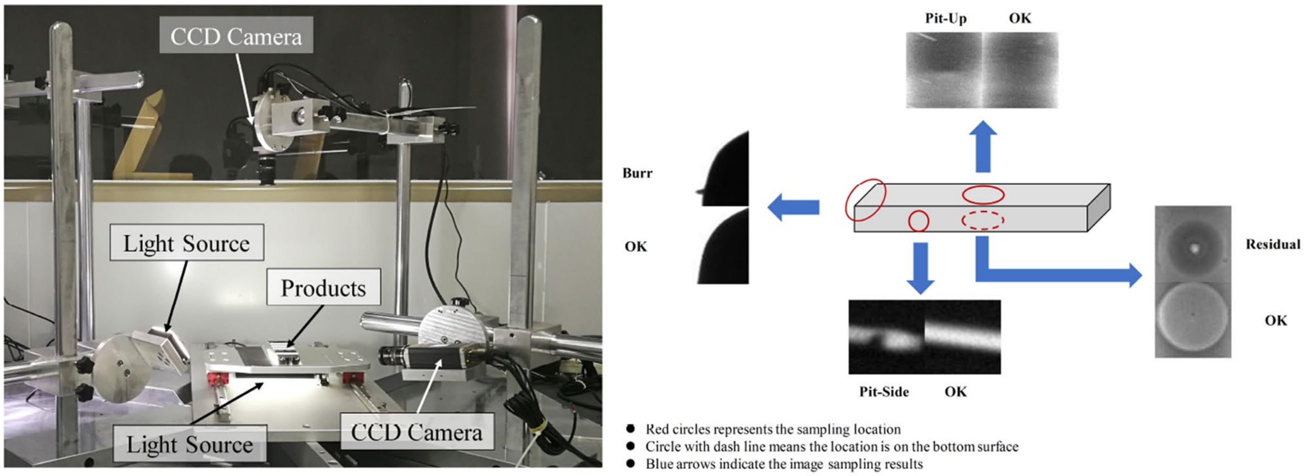


Fig. 5 Image samples of the datasets used in this work (Red circles mark the defective regions)


CCD Camera
Light Source
Products
Light Source
CCD Camera
Pit-Up
OK
Burr
OK
Residual
OK
Pit-Side
OK
● Red circles represents the sampling location
● Circle with dash line means the location is on the bottom surface
● Blue arrows indicate the image sampling results

(a) Image Acquisition Platform

(b) Image Sampling Location

Fig. 6 Image acquisition platform and image sampling location of defective areas (Defect images are resized)

performance of neural networks, Shuangming Yang et al. proposed a series of very effective methods to improve the computational efficiency and throughput of the models, including: biological-inspired cognitive supercomputing system [38], neuromorphic fault-tolerant context-dependent learning framework [39], and large-scale cerebellar network model [40].

Due to the influences of the production process and product dimensions, the sizes of these defects are very small. For example, the length of a burr is only approximately 0.05 mm, and the equivalent radius of a pit defect is merely 0.1 mm. In the defect image patches, the burr defect is only approximately 4 pixels. Another challenge of this detection task is that the different states of CNC machining make the manifestations of residual defects very different, with more than 20 different appearances. In addition, pit defects on the side surfaces of the product also have different appearances due to different foreign materials in the mold. Different appearances of these two defects are shown in Fig. 7. In summary, the difficulty of this inspection task is that the defects are very small, and some defects have great intra-class differences.

Since the NEU dataset contains 300 images for each category, this study divides the dataset into a training dataset and a test dataset at a ratio of 3:2 following Ref. [17], i.e., the training dataset contains 180 images per category. For the IMP dataset, we collected hundreds of images for each class. The training dataset adopted in this work contains 200 images per category, and the remaining images are all used as the test dataset. The approximate quantity allocation of the NEU dataset and the IMP dataset is shown in Table 3.

3.2 Details of model training

This section discusses the details of the MS-CNN model training with the proposed semi-supervised learning strategy on the datasets described in Section 3.1. All the experiments are conducted with open source python libraries and deep learning framework TensorFlow [41]. An NVIDIA GeForce 1080 GPU (RAM 8 GB) is applied to speed up the model training processes.

Since this study aims to improve the inspection accuracy of industrial defects in the cases where numerous labeled samples are not available, we investigate the model performance changes with a gradual reduction in the proportion of labeled samples. Thus, different ratios of labeled samples in all the training samples are selected to train the MS-CNN models, and the performance of these models is tested on the same test dataset. For the NEU dataset and the IMP dataset, the ratios of labeled samples in all training samples are defined in Table 4.

During the experiments on these datasets, all the labeled training samples are used to train a common CNN model with supervised learning to obtain the baseline performance first. Then, different sub-training datasets containing labeled samples randomly selected from the original datasets with the ratios listed in Table 4 are adopted to train models using the proposed semi-supervised learning strategy. The transfer learning method discussed in Section 2.5 is applied to reduce training time and improve performance. In addition, to avoid the bias of dataset partitioning, fivefold cross-validation is used to perform the training and test experiments on all the datasets. Since data augmentation simulates the noise interference encountered in actual recognition scenarios, cross-validation can investigate the

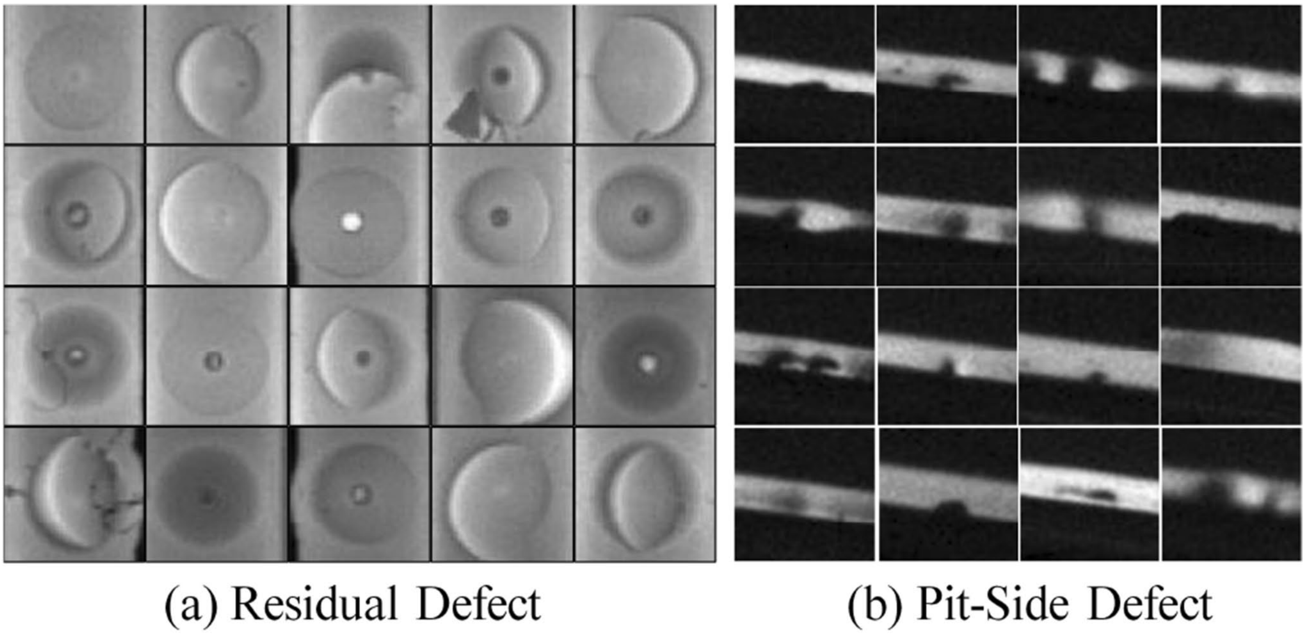


Fig. 7 Different appearances of the residual defect and the pit-side defect in the IMP dataset

robustness of the proposed method to changes in datasets and environmental noise.

For model performance evaluation on the test datasets, since every category of defects should be correctly classified concurrently by a single model, the average accuracy obtained on every individual category is used as a performance indicator (notated as *AveAcc*). This indicator is more convincing than simple overall average accuracy, especially when the test datasets are unbalanced, and it is also referred to as balanced classification accuracy (BCA) in some studies [42, 43]. The indicator *AveAcc* is computed as Eq. (5):

$$AveAcc = \frac{1}{N}\sum_{i}^{N}\frac{m_C^i}{m_A^i}, \tag{5}$$

where N is the number of categories in datasets, m_C^i is the number of correctly classified samples of i-th category, and m_A^i is the number of all the samples of i-th category.

In addition, in some cases, only the average accuracy (*AveAcc*) is not enough to illustrate the model performance. Thus, in the following work of this study, other performance indicators, such as precision, recall, and F1 score, are also considered to investigate the performance improvements of the proposed models obtained on the benchmark dataset.

4 Results and discussion

In this section, the performance of the proposed method on the benchmark dataset NEU was evaluated and compared with results in some previous work. In addition, ablation studies on the NEU dataset were conducted to investigate the performance improvement of the proposed MS-CNN architecture and the semi-supervised learning strategy. Then, visualization methods were adopted to probe the reason for improvement on the NEU dataset of the proposed method. Finally, results of the proposed method on the IMP dataset were illustrated to prove that the proposed method was also effective in solving practical defect inspection problems.

Table 3 Approximate quantity allocation of the datasets used in this work (Images per category)

Dataset	NEU	IMP
Training Dataset	180	200
Test Dataset	120	200~450

Table 4 Ratios of labeled samples in all the datasets in this work

Dataset	Different ratios and amount per category (ratio/amount)				
NEU	100% / 180	50% / 90	25% / 45	5% / 9	–
IMP	100% / 200	50% / 100	25% / 50	10% / 20	5% / 10

4.1 Results on the NEU dataset

The proposed MS-CNN architecture and the semi-supervised learning strategy were first verified on a benchmark dataset NEU to investigate the performance improvements compared with existing methods. In the comparison results, in addition to the average accuracy, the macro average precision (abbreviated as *Prec.*), macro average recall rate (abbreviated as *Recall*), and macro average F1 score (abbreviated as *F1.*) mentioned in Ref. [37] were also used to compare performance in more detail. Comparison results of the proposed method and previous work on the NEU dataset are summarized in Table 5. It should be noted that all the results of the cited papers came from these papers directly. For the results of Yu et al. [17], the meaning of 3 × and 5 × was that they expanded the number of the labeled samples in the NEU dataset to three and five times with GAN models, respectively. In addition, when the MS-CNN models were trained on the 100% labeled samples, no data augmentation (DA) was used.

In Table 5, it can be seen that the proposed method achieved better performance than all the previous work for all the configurations. Specifically, in the case of 25% labeled samples, the proposed method achieved a performance improvement of up to 13.2% compared with Ref. [3]. In addition, compared with the study in Ref. [17], we achieved an accuracy improvement of approximately 3.44% with similar model complexity. Moreover, a recent study in Ref [37]. proposed an inspection model using graph convolutional network (GCN) and adopted a complicated model training strategy. This method obtained comparable accuracy to ours, but its inference time is nearly 20 times that of our model due to the high complexity of their GCN model.

The performance improvement of the proposed method was most obvious when there were fewer labeled samples (25%), which indicated that the proposed method could use numerous unlabeled samples for model feature learning more effectively. Although the performance improvement of the proposed method varied with the number of labeled samples, we still achieved the best results on all performance indicators of all the experimental configurations, including the fully supervised learning method on the 100% labeled samples.

Considering the major challenges of the lack of labeled samples and great intra-class differences in the NEU datasets, we may explore the reason for the performance improvement of the proposed method. The CAE used in Ref. [3] for image features learning and the GAN used in Ref [17]. for training samples generation were indeed helpful for CNN models learning useful image features for classification. However, there was no strategy in their methods to deal with great intra-class differences, which was critical for recognizing similar samples. In addition, generated ambiguous samples with incorrect labels in Ref. [17] would damage the image features of the CNN models, leading to performance degradation. The results of 3 × and 5 × in the cases of 100% training samples confirmed this conjecture, which was 98.53% accuracy of 5 × and 99.56% accuracy of 3 ×, as shown in Table 5. For the work in Ref. [25], the misleading information caused by overconfidence of the incorrect hard pseudo labels would harm the features learned by the models, resulting in low accuracy. In contrast, the soft pseudo labels and mutual correction classifier proposed in this work could alleviate this problem and improve the final performance. Furthermore, the latest work in Ref. [37] utilized GCN models to improve the class separation of the samples and achieved quite a high accuracy. However, the proposed method achieved higher performance using threshold selection and a mutual correction classifier to obtain better pseudo labels. The simplicity of the proposed MS-CNN architecture also made the proposed method much faster than the GCN models.

4.2 Ablation studies on the NEU and STL datasets

Ablation studies were conducted on the NEU dataset to investigate how the innovations proposed in this study contributed to the final performance improvement. Thus, comparison experiments were designed to study the difference in accuracy increase that came from the proposed MS-CNN architecture, the proposed semi-supervised learning strategy, and the data augmentation. In the configurations

Table 5 Comparison Results on the NEU dataset obtained on various ratios of labeled samples

Methods		25%				50%				100%
		AveAcc	*Prec*	*Recall*	*F1*	*AveAcc*	*Prec*	*Recall*	*F1*	*AveAcc*
CAE-SGAN [3]		85.83	-	-	-	94.87	-	-	-	98.96
cDCGAN [17]	3 ×	89.58	-	-	-	96.06	-	-	-	99.56
	5 ×	95.59	-	-	-	97.92	-	-	-	98.33
PLCNN [25]		92.78	92.67	93.36	93.01	96.53	94.67	94.85	94.76	-
MMGCN [37]		98.06	98.00	98.06	98.03	98.75	98.83	98.76	98.80	99.72
Proposed Method		98.19	98.19	98.19	98.18	98.89	98.94	98.89	98.89	**99.72**
Proposed Method + DA		**99.03**	**99.03**	**99.03**	**99.02**	**99.44**	**99.46**	**99.44**	**99.44**	**-**

The optimal accuracy is marked in bold

of comparison experiments, the abbreviation "SL" meant supervised learning strategy, "SSL" meant the proposed semi-supervised learning strategy in this study, and "DA" meant data augmentation. In addition, the results achieved with VGG-16 and Inception V3 [44] models using a fully supervised learning strategy on different ratios of labeled samples were also used as a comparison. The results of ablation studies on the NEU dataset were shown in Table 6, where the "5%" indicated that the training dataset contained 5% labeled samples. When the training datasets contained 100% labeled samples, the MS-CNN models were trained with a fully supervised learning method instead of semi-supervised learning, and no data augmentation was used.

From the results in Table 6, it can be seen that the three innovations of the proposed method all brought different degrees of performance improvement. First, as seen from experiments NOs.1–3, the proposed MS-CNN architecture obtained higher accuracy than vanilla VGG16 models, for example, up to 4.31% accuracy increase in the case of only 5% labeled samples. These gains indicated that combining multiscale image features extracted by different high-level convolutional layers was indeed helpful for subsequent classifiers to make better decisions. In addition, the conventional deeper multiscale architecture, Inception V3, obtained inferior accuracy to the proposed MS-CNN, which indicated that the overly abstract image features extracted by a deeper model lost many discriminative features, which made it difficult to distinguish similar samples.

Then, from the results of experiments NO.3 and NO.4 of Table 6, a common semi-supervised learning strategy based on "hard labels" could improve the accuracy to a certain extent when the labeled samples were scarce. However, the simple label prediction strategy introduced much noise to the pseudo labels of unlabeled samples. Since CNN models tended to learn the patterns contained in datasets [36], some correct pseudo labels still played a positive role in promoting the learning of the image features.

Next, from experiments NO.4 and NO.5, it was intuitive that the proposed semi-supervised learning strategy achieved higher performance than "hard labels" in all the cases, especially when the training dataset contained only 5% labeled samples. The results indicated that the proposed semi-supervised learning strategy based on the soft pseudo labels and the mutual correction classifier could provide more informative supervision for the unlabeled samples during training phases. By including more unlabeled samples with more accurate pseudo labels into model training, the MS-CNN models learned image features with better representation capacity.

Finally, in experiment NO.5 in Table 6, it can be seen that the data augmentation method increased accuracy to varying degrees according to the number of the labeled samples. When the labeled samples were extremely scarce, e.g., only 5%, accuracy improvement reached approximately 5.01%. With more labeled samples provided, gains contributed by data augmentation decreased from 5.01% (9 samples per category) to 0.64% (90 samples per category).

Furthermore, from experiment NO.6, the standard deviation of the results obtained by the proposed method was relatively low, which indicated that the proposed method still had good robustness to the changes of the training datasets and the environmental noise generated by data augmentation.

4.3 Visualization results on the NEU dataset

In this section, some visualization methods were applied to the NEU dataset as an example to probe the reasons for the performance improvement of the proposed method. We first plotted some feature maps extracted by the proposed MS-CNN model and the VGG16 model to investigate the difference of the learned image features. For MS-CNN architecture, we provided the feature maps of the Conv3 layer and the multiscale feature fusion results (marked as Conv_Combine for simplicity) of pool5, pool4_1, and pool3_2 in Fig. 1 calculated on different input images of the NEU dataset. For the VGG-16 architecture, feature maps extracted by the Conv3 and Conv5 layers were illustrated. These two architectures were all trained on the datasets containing all the labeled samples with a fully supervised learning strategy. The visualization results were shown in Fig. 8.

Table 6 Results of ablation studies on the NEU dataset

NO	Configurations	*AveAcc* (%)			
		9×6 (5%)	45×6 (25%)	90×6 (50%)	180×6 (100%)
1	VGG16 + SL	78.47 ± 3.26	92.36 ± 1.74	94.72 ± 1.08	96.62 ± 0.85
2	Inception V3 + SL	80.39 ± 4.78	95.06 ± 2.12	96.89 ± 1.76	99.28 ± 0.22
3	MS-CNN + SL	82.78 ± 1.63	95.14 ± 0.85	97.22 ± 0.62	99.72 ± 0.27
4	MS-CNN + SSL (Hard Label)	84.05 ± 1.81	96.72 ± 0.74	97.61 ± 1.37 37	(**99.72 ± 0.27**)
5	MS-CNN + SSL (Proposed)	86.88 ± 1.82	98.19 ± 0.60	98.72 ± 0.35	(**99.72 ± 0.27**)
6	MS-CNN + SSL + DA	**91.89** ± 0.85	**98.94** ± 0.41	**99.36** ± 0.34	(**99.72 ± 0.27**)

The optimal accuracy is marked in bold

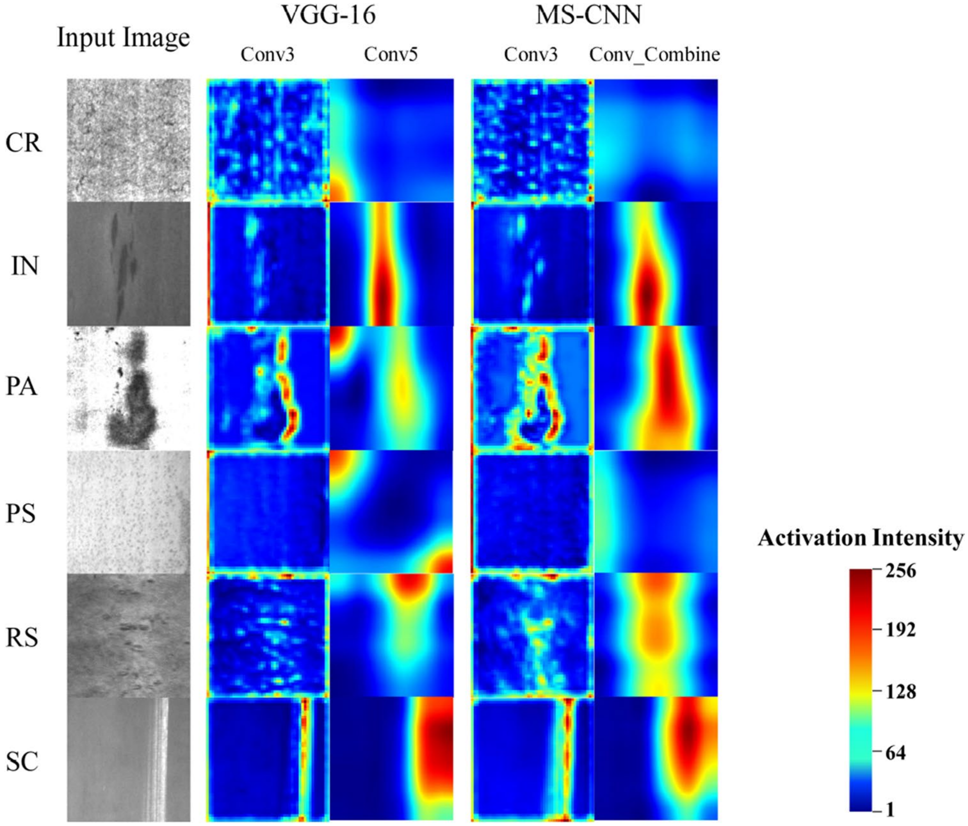
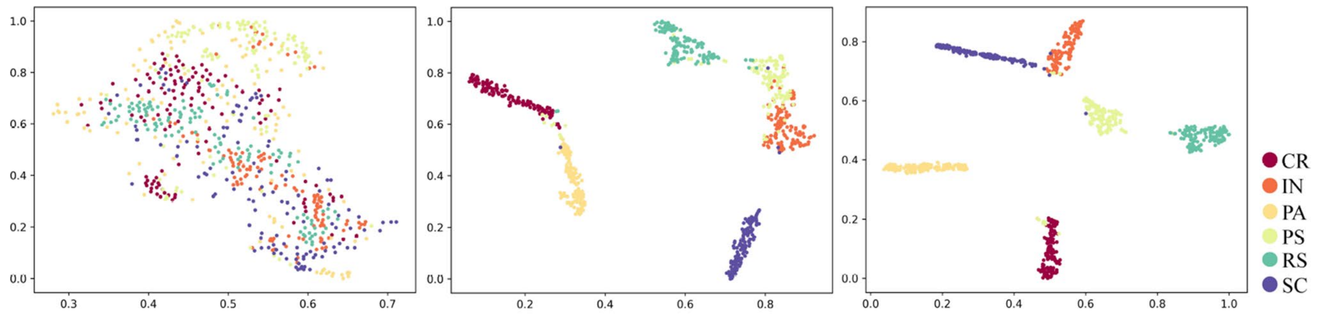


Fig. 8 Some feature maps obtained by VGG-16 and MS-CNN architecture on the NEU dataset

In Fig. 8, it can be seen that the proposed MS-CNN architecture obtained better image features compared with the typical VGG16 model. As shown in Fig. 8, high activation response areas (red and yellow areas) were consistent with the defective regions in the input images. For example, for the input image of class “PA,” high response areas in the feature map extracted by the Conv3 layer of the MS-CNN architecture exactly corresponded to the boundary of the defective region in the input image. However, the feature maps extracted by the Conv3 layer of the VGG-16 models had incomplete activation areas. For the feature maps of Conv5 of VGG-16 and Conv_Combine of MS-CNN, which were all the last feature maps before the classifier, the advantage of feature maps obtained by the MS-CNN was more noticeable, especially in class “PA,” “RS” and “SC.” These feature maps had more abundant information and matched the defective areas better, resulting in higher model accuracy.

Then, to study the performance of the proposed method in solving the challenge of large intra-class differences existing in the NEU datasets, the t-SNE [45] method was adopted to visualize the feature distribution. The original images in the NEU test datasets were flattened into one-dimensional vectors, and the t-SNE method was used for dimension reduction and direct visualization for comparison. Then, the feature vectors obtained by the penultimate fully connected layer in the proposed MS-CNN architecture and VGG-16 were visualized with the t-SNE method. The MS-CNN models were trained on 25% of labeled samples and 75% of unlabeled samples, while the VGG-16 models were trained only on 25% of labeled samples. All the results of visualization of the t-SNE were used to plot images, as shown in Fig. 9.

In Fig. 9, after training, the VGG16 architecture with a fully supervised method and the MS-CNN architecture with a semi-supervised method all clearly divided the original dataset into six classes. However, the proposed method got better performance. The feature points obtained by the proposed method in Fig. 9(c) were far apart, and the interference between point clouds was smaller than that of the VGG16 architecture trained with the fully supervised method in Fig. 9(b). For example, for classes “PS” and “IN,” the light-green point cloud and orange point cloud on the right in Fig. 9(b) had much overlap, which indicated that VGG-16 models trained on only 25% of labeled samples could not distinguish the two categories very well. However, the results in Fig. 9(c) indicated that the proposed method could solve the problem better and that the point clouds of classes “PS” and “IN” had little intersection. In addition, the accuracies obtained by these two methods were 92.36% and 98.33%, respectively, on the training datasets of 25% labeled samples. Both the accuracy and visualization results illustrated that the proposed method had an excellent



(a) Original test dataset (b) Results of VGG16+SL (c) Results of MS-CNN+SSL

Fig. 9 The t-SNE results of the layer FC7 of the proposed MS-CNN architecture

performance in improving classification accuracy when labeled samples were scarce.

Although a few points of class "PS" were confused with class "CR" in Fig. 9(c), i.e., the dark-red point cloud in the middle and lower parts of Fig. 9(c) contained several yellow-green dots (PS), the proposed method could still achieve a high accuracy of 98.33%. The results showed that the proposed method would not be confused by similar samples of different classes in the NEU dataset, nor would it misclassify the samples with large differences in the same class. Considering that the main challenge of the NEU dataset was that inter-class samples had similar appearances for some defects, while some defects had large intra-class differences, the results in Fig. 9 indicated that the proposed method could address this challenge to some extent.

The above visualization experiments showed that with the MS-CNN architecture and semi-supervised learning strategy, the proposed method obtained better high-dimensional convolutional image features, which were helpful for the models to distinguish similar samples, and improved the accuracy. The proposed methods could also handle the challenges of the NEU dataset and achieved better accuracy, exceeding other common methods.

4.4 Practical application on the IMP Dataset

This section discussed the results of the proposed method in solving practical surface defect recognition problems of a plastic injection molding product. An automatic defect inspection system was developed in this task, which cropped possible defective image patches from the original CCD image and fed these patches to the MS-CNN models for classification. The recognition accuracy of the proposed method was shown in Table 7, where the "5% (10)" meant that the models were trained with 5% labeled samples and 95% unlabeled samples of the original training dataset, and there were ten labeled samples in each category.

In Table 7, it can be seen that the proposed method also obtained great performance in the real-world defect recognition task. First, from experiments NO.1 and NO.2, the original VGG16 architecture achieved a pretty high accuracy when the models were trained on 100% of the training dataset because the image features in the IMP dataset were much simpler than that of the NEU dataset. However, the proposed MS-CNN architecture still got better performance than the vanilla VGG16 model. This improvement proved the effectiveness of the multiscale features extracted by the proposed MS-CNN architecture again.

Then, from the results of experiment NO.3 in Table 7, with only 10% labeled samples of the original training dataset (i.e., 20 images per category), we could still reach a high accuracy of approximately 99.25%, better than 96.12% achieved by the VGG-16 models trained with a fully supervised learning strategy. This indicated that the proposed semi-supervised learning strategy based on the soft pseudo label and the mutual correction classifier could utilize unlabeled samples to improve inspection accuracy. Thus, we could save a lot of time and effort on labeling samples in practical applications and accelerate the research process of the detection algorithms.

Table 7 Results of comparison experiments on the Industrial Dataset

NO	Configurations	*Ave_Acc* (%)				
		5% (10)	10% (20)	25% (50)	50% (100)	100% (200)
1	VGG16 + SL	95.30 ± 2.34	96.12 ± 1.93	96.72 ± 1.58	98.14 ± 1.03	98.81 ± 0.38
2	MSCNN + SSL	97.02 ± 1.42	98.84 ± 0.95	**99.39** ± 0.59	**99.40** ± 0.43	**99.51 ± 0.22**
3	MSCNN + SSL + DA	**98.82** ± 1.61	**99.25** ± 1.33	98.62 ± 0.74	98.94 ± 0.81	**(99.51 ± 0.22)**

The optimal accuracy is marked in bold

Finally, experiments NO.2 and NO.3 in Table 7 illustrated that the data augmentation method did not always work well. In the cases where labeled samples were 25% and 50%, i.e., more labeled samples were available, we found slight performance degradations that were 0.77% and 0.46% decrease in case of 25% and 50%, respectively. This may be because the noise introduced by the data augmentation method damaged the image features that the models had learned on labeled samples. In addition, more duplicated samples could also lead to model over-fitting, which was similar to the results in Ref. [17].

Despite this problem, the proposed method still achieved excellent performance on all the defect categories of the IMP dataset. These results showed that the proposed method also worked well in the defect inspection of practical productions. Moreover, in deploying the MS-CNN models to actual inspection scenarios, some other digital neuromorphic computing, such as SNN [46, 47], was considered to reduce the computing time and power consumption.

5 Conclusion

In this work, a novel MS-CNN architecture and a new semi-supervised learning strategy based on the soft pseudo label and a mutual correction classifier are proposed to improve the industrial defect recognition accuracy in cases where labeled samples are scarce and have large intra-class differences. The advantages of the proposed methods are first verified on a surface defect recognition benchmark dataset, the NEU dataset. The results on the NEU dataset show that the classification accuracy of the proposed method is much higher than that of existing methods, especially with much fewer labeled samples, obtaining an accuracy of up to 99.03% on only 45 labeled samples per category. Ablation studies and the visualization results indicate that the accuracy improvement can be attributed to better image features obtained by multiscale feature fusion of the proposed MS-CNN architecture and the massive unlabeled samples utilized with the proposed semi-supervised learning method. The data augmentation method also makes a great contribution to performance improvement, especially in cases where labeled samples are extremely scarce. In addition, the performance of the proposed method is also tested on an IMP dataset that comes from a real-world injection molding product defect recognition task. The results illustrate that when trained on only 20 labeled images and 180 unlabeled images per category, the proposed method still achieves an average accuracy of approximately 99%.

However, some limitations still exist in the proposed method. For example, when trained on more complicated datasets or fewer labeled samples, the proposed method cannot achieve quite high accuracy. We may incorporate more efficient label prediction methods into our work in the future to further improve the accuracy of pseudo labels. We will improve our study to make it suitable for more application scenarios of industrial defect inspection, such as welding, optic glass manufacture and so on.

Acknowledgements The authors would like to acknowledge financial support from Key-Area Research and Development Program of Guangdong Province (Grant No. 2019B090918001) and National Program on Key Basic Research Project (Grant No. 2019YFB1704900).

Funding Key-Area Research and Development Program of Guangdong Province (Grant No. 2019B090918001).

National Program on Key Basic Research Project (Grant No. 2019YFB1704900).

Declarations

Conflicts of interest/ Competing interests 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.

References

1. Zheng X, Chen J, Wang H et al (2021) A deep learning-based approach for the automated surface inspection of copper clad laminate images. Appl Intell 51:1262–1279
2. Wu J, Le J, Xiao Z, et al (2021) Automatic fabric defect detection using a wide-and-light network[J]. Applied Intelligence 51(7):4945–4961
3. Di H, Ke X, Peng Z, Dongdong Z (2019) Surface defect classification of steels with a new semi-supervised learning method. Opt Lasers Eng 117:40–48
4. Dong H, Song K, He Y, et al (2019) PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection[J]. IEEE Transactions on Industrial Informatics 16(12):7448–7458
5. Lee H, Ryu K (2020) Dual-Kernel-Based Aggregated Residual Network for Surface Defect Inspection in Injection Molding Processes. Appl Sci 10:8171
6. Liu B, Huang P, Zeng X, Li Z (2017) Hidden defect recognition based on the improved ensemble empirical decomposition method and pulsed eddy current testing. Ndt E Int 86:175–185
7. Jian C, Gao J, Ao Y (2017) Automatic surface defect detection for mobile phone screen glass based on machine vision. Appl Soft Comput 52:348–358
8. Xiao Q, Dai J, Luo J, Fujita H (2019) Multi-view manifold regularized learning-based method for prioritizing candidate disease miRNAs. Knowledge-Based Syst 175:118–129
9. Hayashi T, Fujita H, Hernandez-Matamoros A (2021) Less complexity one-class classification approach using construction error of convolutional image transformation network. Inf Sci (Ny) 560:217–234
10. Ren R, Hung T, Tan KC (2017) A generic deep-learning-based approach for automated surface inspection. IEEE Trans Cybern 48:929–940
11. Fujita H, Cimr D (2019) Decision support system for arrhythmia prediction using convolutional neural network structure without preprocessing. Appl Intell 49:3383–3391

12. Lin X, Wang X, Li L (2020) Intelligent detection of edge inconsistency for mechanical workpiece by machine vision with deep learning and variable geometry model. Appl Intell 50:2105–2119
13. Okaro IA, Jayasinghe S, Sutcliffe C et al (2019) Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. Addit Manuf 27:42–53
14. Qi G-J, Luo J (2020) Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. IEEE Transactions on Pattern Analysis and Machine Intelligence. https://doi.org/10.1109/TPAMI.2020.3031898
15. Wu H, Prasad S (2017) Semi-supervised deep learning using pseudo labels for hyperspectral image classification. IEEE Trans Image Process 27:1259–1270
16. Li Z, Ko B, Choi H-J (2019) Naive semi-supervised deep learning using pseudo-label. Peer-to-Peer Netw Appl 12:1358–1368
17. He Y, Song K, Dong H, Yan Y (2019) Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. Opt Lasers Eng 122:294–302
18. Protopapadakis E, Doulamis A, Doulamis N, Maltezos E (2021) Stacked autoencoders driven by semi-supervised learning for building extraction from near infrared remote sensing imagery. Remote Sens 13:371
19. Zhan Y, Hu D, Wang Y, Yu X (2017) Semisupervised hyperspectral image classification based on generative adversarial networks. IEEE Geosci Remote Sens Lett 15:212–216
20. Luo Q, Gao B, Woo WL, Yang Y (2019) Temporal and spatial deep learning network for infrared thermal defect detection. NDT E Int 108:102164
21. Yang L, Wang Z, Gao S (2019) Pipeline magnetic flux leakage image detection algorithm based on multiscale SSD network. IEEE Trans Ind Informatics 16:501–509
22. Tulbure A-A, Tulbure A-A, Dulf E-H (2021) A review on modern defect detection models using DCNNs–Deep convolutional neural networks. J Adv Res. https://doi.org/10.1016/j.jare.2021.03.015
23. Lobov SA, Mikhaylov AN, Shamshin M et al (2020) Spatial properties of STDP in a self-learning spiking neural network enable controlling a mobile robot. Front Neurosci 14:88
24. Yang S, Gao T, Wang J et al (2021) Efficient spike-driven learning with dendritic event-based processing. Front Neurosci 15:97
25. Gao Y, Gao L, Li X, Yan X (2020) A semi-supervised convolutional neural network-based method for steel surface defect recognition. Robot Comput Integr Manuf 61:101825
26. Wang F, Zhu L, Li J, et al (2021) Unsupervised soft-label feature selection. Knowledge-Based Syst 219:106847
27. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
28. Han D, Liu Q, Fan W (2018) A new image classification method using CNN transfer learning and web data augmentation. Expert Syst Appl 95:43–56
29. Moradi R, Berangi R, Minaei B (2020) A survey of regularization strategies for deep models. Artif Intell Rev 53:3947–3986
30. Baek K, Bang D, Shim H (2021) GridMix: Strong regularization through local context mapping. Pattern Recognit 109:107594
31. Shao S, McAleer S, Yan R, Baldi P (2018) Highly accurate machine fault diagnosis using deep transfer learning. IEEE Trans Ind Informatics 15:2446–2455
32. Li C, Zhang S, Qin Y, Estupinan E (2020) A systematic review of deep transfer learning for machinery fault diagnosis. Neurocomputing 407:121–135
33. Deng J, Dong W, Socher R, et al (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee, pp 248–255
34. Li X, Zhang W, Ding Q (2019) Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. Reliab Eng Syst Saf 182:208–218
35. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980
36. Krueger D, Ballas N, Jastrzebski S, Arpit D, Kanwal MS, Maharaj T, Bengio E, Fischer A, Courville AC (2017) Deep nets don't learn via memorization. ICLR
37. Wang Y, Gao L, Gao Y, Li X (2021) A new graph-based semi-supervised method for surface defect classification. Robot Comput Integr Manuf 68:102083
38. Yang S, Wang J, Hao X et al 2021 BiCoSS: toward large-scale cognition brain with multigranular neuromorphic architecture IEEE Trans Neural Networks Learn Syst 1 15 https://doi.org/10.1109/TNNLS.2020.3045492
39. Yang S, Wang J, Deng B et al 2021 Neuromorphic context-dependent learning framework with fault-tolerant spike routing IEEE Trans Neural Networks Learn Syst 1 15 https://doi.org/10.1109/TNNLS.2021.3084250
40. Yang S, Wang J, Zhang N et al 2021 CerebelluMorphic: large-scale neuromorphic model and architecture for supervised motor learning IEEE Trans Neural Networks Learn Syst 1 15 https://doi.org/10.1109/TNNLS.2021.3057070
41. Abadi M, Agarwal A, Barham P, et al (2016) Tensorflow: Large-scale machine learning on heterogeneous distributed systems[J]. arXiv preprint arXiv:1603.04467
42. Cui Y, Wu D, Huang J (2020) Optimize TSK fuzzy systems for classification problems: Minibatch gradient descent with uniform regularization and batch normalization[J]. IEEE Transactions on Fuzzy Systems 28(12):3065–3075
43. Zhang X, Wu D (2019) On the vulnerability of CNN classifiers in EEG-based BCIs. IEEE Trans Neural Syst Rehabil Eng 27:814–825
44. Szegedy C, Vanhoucke V, Ioffe S, et al (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 2818–2826
45. Chen H, Hu Q, Zhai B et al (2020) A robust weakly supervised learning of deep Conv-Nets for surface defect inspection. Neural Comput Appl 32:11229–11244
46. Yang S, Deng B, Wang J et al (2019) Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. IEEE Trans neural networks Learn Syst 31:148–162
47. Yang S, Wang J, Deng B et al (2018) Real-time neuromorphic system for large-scale conductance-based spiking neural networks. IEEE Trans Cybern 49:2490–2503

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.