The International Journal of Advanced Manufacturing Technology (2022) 123:1999–2015
https://doi.org/10.1007/s00170-022-10335-8

ORIGINAL ARTICLE

A new lightweight deep neural network for surface scratch detection

Wei Li[1] · Liangchi Zhang[2,3,4] · Chuhan Wu[1] · Zhenxiang Cui[5] · Chao Niu[5]

Received: 7 July 2022 / Accepted: 15 October 2022 / Published online: 26 October 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
This paper aims to develop a lightweight convolutional neural network, *WearNet*, to realise automatic scratch detection for components in contact sliding such as those in metal forming. To this end, a large surface scratch dataset obtained from cylinder-on-flat sliding tests was used to train the *WearNet* with appropriate training parameters such as learning rate, gradient algorithm and mini-batch size. A comprehensive investigation on the network response and decision mechanism was also conducted to show the capability of the developed *WearNet*. It was found that compared with the existing networks, *WearNet* can realise an excellent classification accuracy of 94.16% with a much smaller model size and faster detection speed. Besides, *WearNet* outperformed other state-of-the-art networks when a public image database was used for network evaluation. The application of *WearNet* in an embedded system further demonstrated such advantages in the detection of surface scratches in sheet metal forming processes.

Keywords Surface scratch detection · Convolutional neural network · Contact sliding · Sheet metal forming

1 Introduction

In industrial production, the detection of workpiece surface defects is essential to ensure product quality [1]. For example, in a metal forming process, surface scratch in contact sliding has been critical because it downgrades the surface quality of a workpiece and the service life of a tooling system. However, this has been traditionally a manual process, which is very inefficient, inaccurate and unreliable [2]. Recent advances in artificial intelligence provide a promising approach to tackling tough engineering problems, such as fault detection [3], nonlinear system control [4, 5] and surface defect detection [6]. For instance, the image features of surface defects can be learned by machine learning techniques such as the support vector machine (SVM) and the artificial neural network (ANN). The SVM algorithm has been utilised to analyse and classify surface defects on steel surfaces [7] and cutting tools [8]. Similar applications of the ANN algorithm have been noted for defect recognition in cold rolling [9] and colour-filter production [10]. For example, the performance of different neural networks for surface crack detection in fracture experiments was tested and compared [11]. However, the separated feature extraction and classification operations have significantly restricted the detection efficiency.

The CNN-based deep learning technology has demonstrated its capability in image classification because it can automatically detect and extract high-level image features from the labelled image data [12]. Several CNN networks have been developed and applied to classify the image data, e.g. AlexNet [13], VGG-16 [14], GoogleNet [15] and EfficientNet [16]. The applications of CNN networks for the detection of surface defects [17, 18], rolling bearing degradation [19] and roll marks on hot-rolled steel plates/strips [20] were also noted. For example, an image detection model based on R-CNN network was proposed to identify the wear location and wear mechanism in tribological tests [21].

✉ Liangchi Zhang
zhanglc@sustech.edu.cn

[1] School of Mechanical and Manufacturing Engineering, The University of New South Wales, Kensington, NSW 2052, Australia

[2] Shenzhen Key Laboratory of Cross-Scale Manufacturing Mechanics, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China

[3] SUSTech Institute for Manufacturing Innovation, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China

[4] Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China

[5] Baoshan Iron & Steel Co., Ltd., Shanghai 200941, China

Besides, different optimisation algorithms have been proposed to train the CNN networks appropriately. For instance, an Adam optimiser with power-exponential learning rate was proposed to control the iteration direction and step size in order to tackle the problems of local minima and overshoot in network training [22]. Although the CNN networks have been widely used for image classification with high accuracy in recent studies, their large model sizes and complicated structures limit the classification speed and bring about high latency.

Therefore, lightweight CNN networks, e.g. SqueezeNet [23], MobileNet [24] and ShuffleNet [25], have been developed to decrease the network parameter number and model size without sacrificing the classification accuracy. For example, a new fire module was utilised in the SqueezeNet to considerably reduce the computation consumption and communication cost [23]. Therefore, it can be feasibly built in the hardware with limited memory, e.g. mobile devices, to complete the real-time object detection in automatic vision-based systems [26]. However, a lightweight CNN network for detecting the surface scratch in contact sliding is yet unavailable.

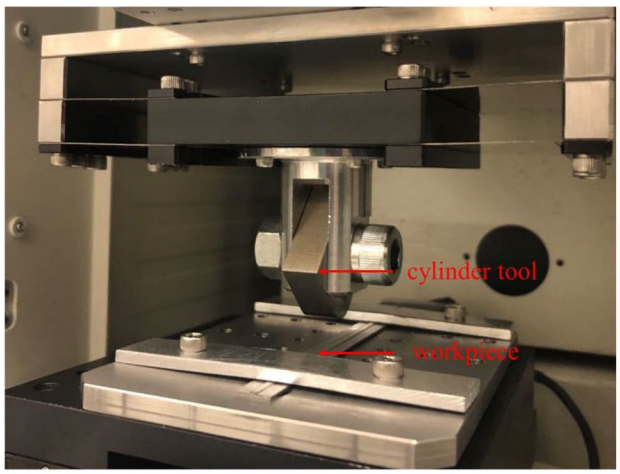
Recently, the embedded system has been used to deploy CNN networks to complete real-time recognition and detection tasks, e.g. vehicle plate recognition [27], fire detection [28], handwriting recognition [29] and action recognition [30, 31]. Normally, the embedded hardware has limited computation capacity and on-board memory; thus, lightweight CNN architectures are more feasible to be deployed in the embedded environment. For example, an anamorphic depth lightweight CNN, Anam-Net [32], was proposed to segment anomalies in COVID-19 chest CT images. Therefore, it is expected that deploying a lightweight CNN network for surface scratch detection in sheet metal forming will help to improve the level of automation and efficiency.

This paper aims to develop a lightweight CNN structure, called *WearNet*, for surface scratch detection in contact sliding. A customised convolutional block will be developed to reduce the training parameter number and network layers as well as to simplify the network structure but retain classification accuracy. To train the *WearNet*, cylinder-on-flat sliding tests will be conducted to provide a large-scale surface scratch dataset. The network response and decision mechanism will be investigated to reveal the *WearNet* capability. The *WearNet* will then be compared with the existing advanced CNN structures to demonstrate its advantages in classification accuracy, model size and computation efficiency. In addition, the performance of the developed *WearNet* will be compared against other existing CNN networks based on a public image database, i.e. the NEU surface defect database [20]. Finally, a Linux-based embedded system will be utilised as the deploying environment to further test the detection performance of *WearNet*.

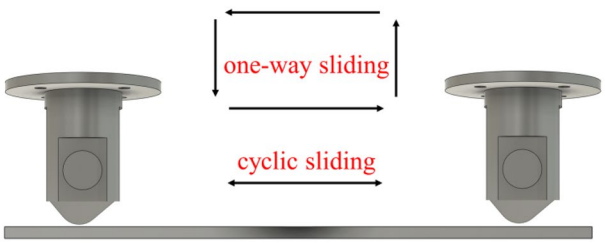
2 Image database of surface scratches

2.1 Experimental setup for data collection

To develop a reliable CNN-based detection model, a large-scale surface scratch database is essential. To extend the database scale, cylinder-on-flat sliding tests (see Fig. 1) were conducted under a wide range of operation conditions listed in Table 1. The cylinder-on-flat sliding setup has been used to mimic the contact conditions encountered in a metal forming process [33, 34]. To match industrial production conditions, two typical types of high-strength steel, DP980 and QP980, and DC53 tools with nitriding and vacuum heat treatment were selected as the pair of sliding contact. Both the one-way and cyclic sliding tests were carried out to mimic practical sliding types. The ranges of testing parameters



(a) cylinder-on-flat sliding system



(b) one-way and cyclic sliding

Fig. 1 Illustration of the experimental setup for data collection

Table 1 Conditions of the contact sliding experiments

No.	Cylinder tool			Workpiece	Load (N)	Speed (mm/s)	Width (mm)	Sliding
	Material	Surface	Radius (mm)					
1	DC53	Nitriding	8–12	DP980	20–40	2	1.5	Cyclic
2	DC53	Nitriding	8–12	DP980	20–40	3	1.2	One-way
3	DC53	Nitriding	8–12	QP980	20–40	3	1.2	One-way
4	DC53	VHT	8–12	QP980	20–40	3	1.2	One-way

(tool radius, normal load, sliding speed, contact width) are listed in Table 1.

2.2 Image data processing

After each sliding test, the surface topography was measured by a digital microscope, OLYMPUS DSX 510. The measuring size of each image is 750 × 750 µm. Both of the surface images of DP980 and QP980 workpieces were divided into five categories (see Fig. 2):

1. The surface images prior to contact sliding were labelled as *intact surface*.
2. After certain cycles in a sliding test, if material transfer was identified on the workpiece surface without obvious scratches, the measured surface images were denoted as *material transfer*.
3. The images with the maximum depth of scratching (h_{max}) below 2 µm, $2 < h_{max} < 4$ µm and $h_{max} > 4$ µm were called as *minor, mild* and *severe scratch*, respectively.

The surface images, Fig. 2c–e, were identified based on the h_{max}. This is because h_{max} plays an important role in determining the severity of scratching damage [35, 36]. Overall, the database with a total of 10,500 surface images was identified by ten labels, as shown in Table 2. These images were randomly divided into training, validation and testing datasets with a ratio of 4:1:1 (7000:1750:1750). The image resolution was normalised to 227 × 227 prior to the training.

3 *WearNet* for surface scratch detection

3.1 Structure of *WearNet*

The existing advanced CNN networks were designed to classify the 1000 labels of ImageNet [37, 38] with a database of over 14 million images. For the surface scratch identification in the current study, the image database was much smaller than ImageNet and fewer image labels were utilised. As such, a lightweight *WearNet* was developed based on a novel convolutional block to prevent overfitting, to effectively minimise the network parameters and to reduce the model size. The architecture and specifications of *WearNet* are outlined as follows (see Table 3 and Fig. 3):

1. Convolutional layer: this plays an important role in extracting the image features from the input image data, which is achieved by the convolution kernel. The con-

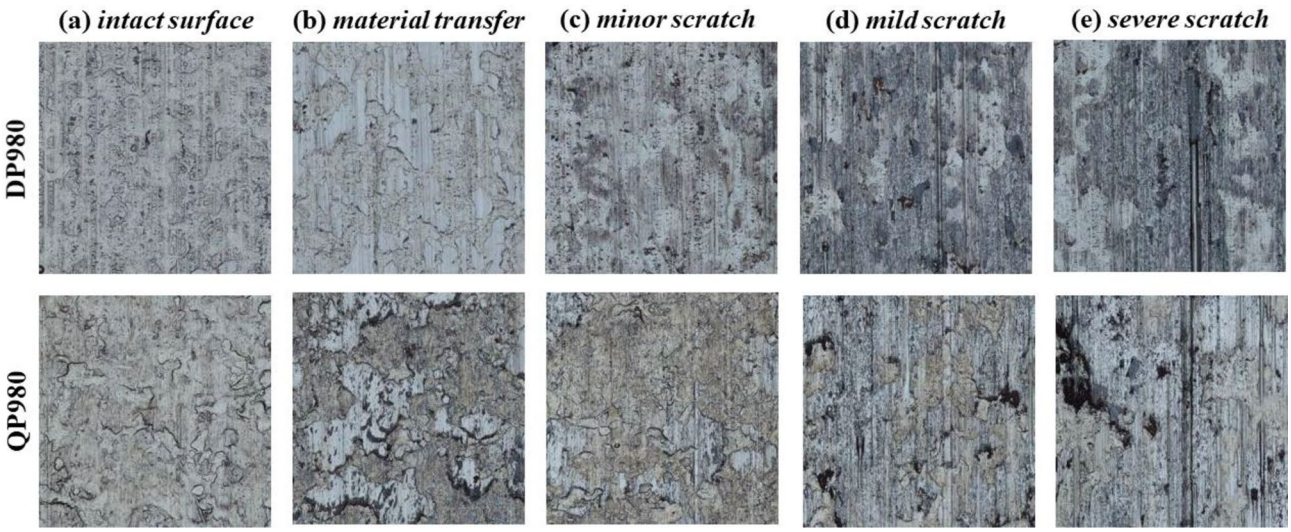


Fig. 2 Surface scratch labels in the image database: **a** *intact surface*, **b** *material transfer*, **c** *minor scratch*, **d** *mild scratch*, **e** *severe scratch*

Table 2 Image numbers of different surface image labels

	Intact surface	*Material transfer*	*Minor scratch*	*Mild scratch*	*Severe scratch*
DP980	720	1320	1200	1800	660
QP980	660	1320	1200	900	720

volution kernel is a square filter, which can scan the input image and output feature maps. The kernel sizes for conv-1 and conv-2 are 3×3 and 1×1, respectively. Each convolutional layer is followed by a ReLU activation function.

2. Max-pooling layer: The role of the pooling layer is to reduce the feature map size by downsampling. There are two common methods to conduct pooling: average pooling and max pooling. The max-pooling is more suitable for image feature processing as it preserves the maximum output in a rectangular region. Therefore, the max-pooling strategy was selected in this paper. Besides, the network conducts max pooling with a stride of 2 to ensure late downsampling.
3. Convolutional block: This block takes advantage of separable convolution and squeeze-expand operations, as shown in Fig. 4. It starts with a squeeze convolution layer with 1×1 filters, which helps to restrict the total number of input channels, n_1, fed into the following expand module. The expand module consists of batch normalisation, separable convolution and expand convolution using a 3×3 kernel. In the separable convolution, a channel-wise 3×3 spatial convolution is followed by a point-wise 1×1 convolution, which can bring about higher computation efficiency as fewer convolution operations will be conducted. In the concatenation layer, the channel number increases from n_1 to $4\times n_1$. As such, the network parameters and model size are decreased significantly.
4. Dropout layer: The dropout layer is to avoid the overfitting problem in the network training process [39]. The strategy is to deactivate some hidden layer nodes in the neural network and minimise their effects in the current training step. In this study, a dropout layer with a ratio of 0.5 was applied after max pool-3.
5. GAP layer: The global average pooling (GAP) layer is used to replace the fully connected layer in the traditional CNN networks. The GAP layer averages each feature map to enforce the correspondence between feature maps and image categories. There is no optimisation for the parameters, which further reduces the network parameters and minimises the overfitting problem.

Table 3 Specifications of the *WearNet* network

Layer	Kernel size	Input size	Output channels
conv-1	3×3	$227\times227\times3$	32
max pool-1	3×3	113×113	32
conv-block-1	\	56×56	128
conv-block-2	\	56×56	128
max pool-2	3×3	56×56	128
conv-block-3	\	28×28	256
conv-block-4	\	28×28	256
max pool-3	3×3	28×28	256
dropout	\	14×14	256
conv-2	1×1	14×14	10
GAP	\	14×14	10
softmax	\	1×1	10

In this study, the *WearNet* was developed by using the customised convolutional block. The network layers and parameters were effectively minimised to bring about a smaller mode size and higher classification speed. The comparison among the *WearNet* and other CNN networks was conducted on an embedded system to demonstrate the distinguished performance of the proposed *WearNet* for practical applications.

3.2 Training details

The *WearNet* was trained and evaluated by MATLAB on a PC with an Intel i5-10,600 (3.3 GHz and 16 GB RAM) and an NVIDIA RTX 3080 GPU (10 GB). Deep Learning Toolbox in MATLAB can provide a friendly framework for building network structures, setting training parameters and monitoring training processes. GPU computing was utilised in the network training to speed up the iteration. In general, there are three learning algorithms in the machine learning area, including supervised learning [40], unsupervised learning [41] and reinforcement learning [42]. In this paper, the supervised learning algorithm was adopted and the dataset consisting of labelled images listed in Table 2 was used for network training. The selection of training parameters (e.g. learning rate, gradient algorithm and mini-batch size) is discussed in the following section.

3.3 Evaluation protocol

The evaluations of deep neural networks are based on the following aspects:

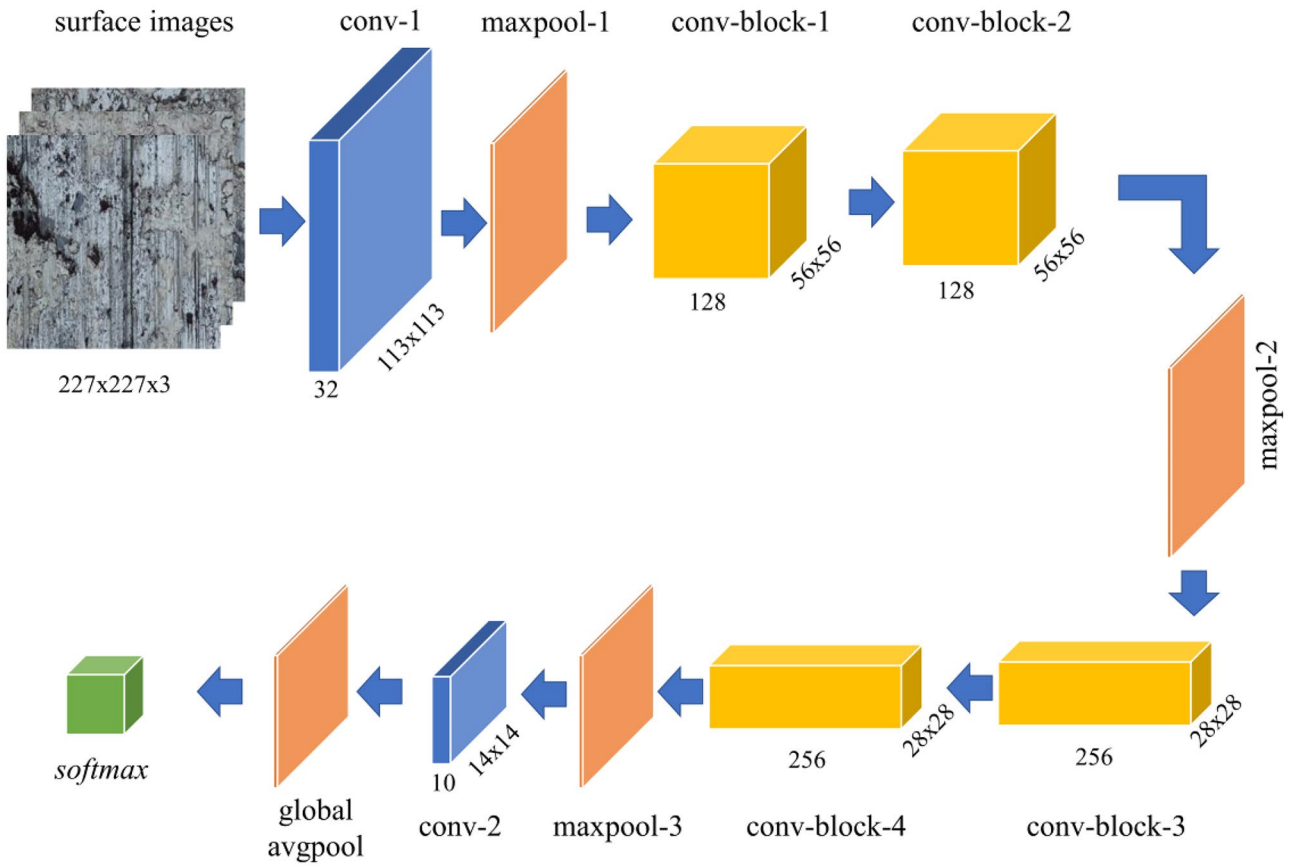


Fig. 3 Architecture of the *WearNet* network

1. Training time T: The training time is related with the network architecture, database scale and training platform as well as the training parameters.
2. Classification accuracy p: The prediction result is considered accurate when the predicted category with the highest confidence is consistent with the ground truth. Thus, the classification accuracy (p) can be given by:

$$p = N_a/N \tag{1}$$

where N_a and N refer to the numbers of accurately classified images and total images, respectively.

3. Recognition rate r: For a specific image class c, if M_a and M donate the numbers of images classified as class c correctly and the total image of class c, the recognition rate (r) of class c can be defined as

$$r = M_a/M \tag{2}$$

4. Classification time t: The classification time plays an essential role in evaluating the network performance, particularly for the industrial production involving fast recognition. In this study, the average classification time (t) for each surface image was calculated for further analysis and comparison.
5. Model size: This determines the applicability of the *WearNet* in production. Typically, larger CNN architectures require more transition bandwidth and communication costs.

Classification accuracy is one of the most important evaluation metrics, as it indicates the overall classification performance of a CNN network. However, the classification accuracy alone can be misleading, if the numbers of surface images in individual classes are unequal. Therefore, the recognition rate and confusion matrix are used to check the network performance on each image class and to figure out how the CNN network is confused when making classification decisions. The training and classification time will play an important role when the computation efficiency of different networks is investigated. Besides, the model size should be taken into consideration when the CNN network is deployed in the embedded environment, as the on-board memory of an embedded device is usually limited.

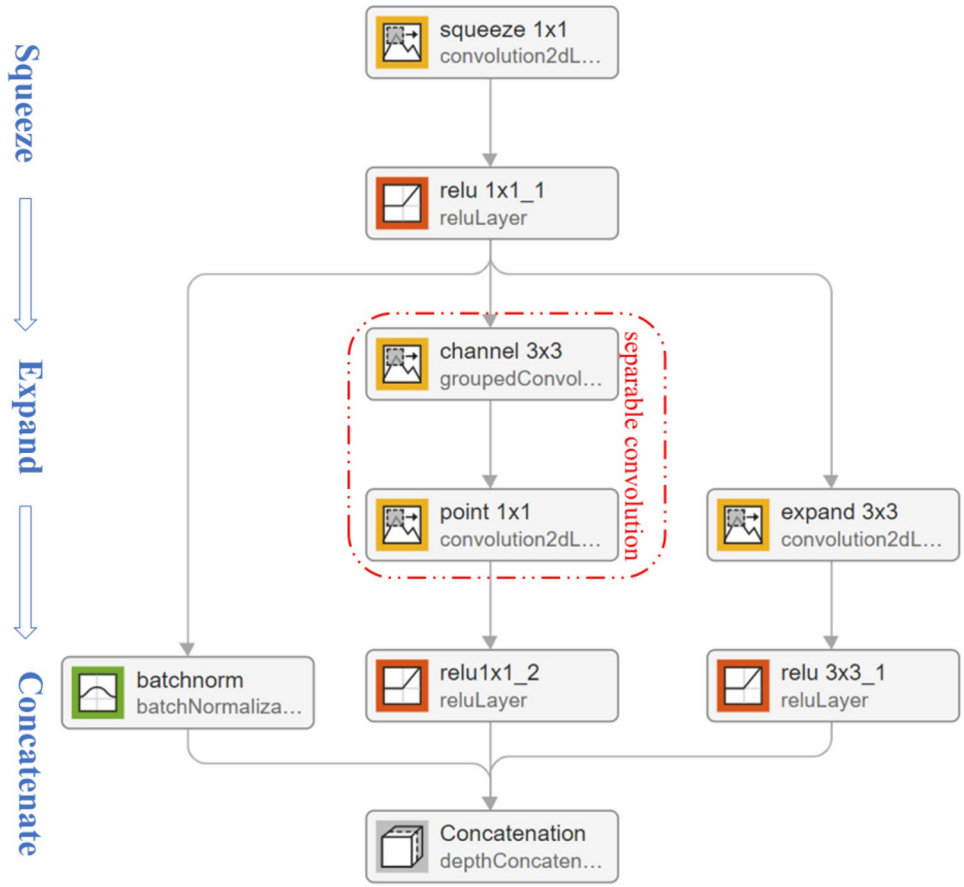


Fig. 4 Illustration of a customised convolutional block in the *WearNet* structure

4 Results and discussions

In training the *WearNet*, it is crucial to select appropriate training parameters. This should be done by considering the network structure, the surface image database and the computation resource available. In this section, the selection of the optimised training parameters for *WearNet* was explored. Then, the *WearNet* was investigated by focusing on the network response, layer activations and network decision mechanism. The comparison between the *WearNet* and other CNN networks was conducted by using the evaluation protocol in the last section.

4.1 Selection of training parameters

The effects of different training parameters, e.g. learning rate, gradient algorithm and mini-batch size, were investigated in this paper, as they were reported to have a considerable influence on the training results in the literature [43, 44]. In the training experiments, it was found that the validation accuracy usually reached a stable stage after around 150 training epochs. Therefore, a series of network training experiments were conducted with the maximum epoch number of 200. The epoch refers to the entire image database being fully trained once. The mini-batch size (N_b) refers to the number of image data used for networking training in a single iteration. Generally, the network parameters are trained and updated by a greater number of times for a higher epoch number. With a given training database size D, the number N_i of total iterations can be given by

$$N_i = (D/N_b) \times 200 \tag{3}$$

1. Learning rate: The learning rate determines the converging speed of iteration. In the network training, it is essential to find an optimal value of learning rate to achieve a reasonable balance between the training speed and validation accuracy. Different learning rates, ranging from 0.001 to 0.00001, were tested and compared, as shown in Fig. 5. Besides, a piecewise learning rate from 0.001 to 0.0001 was also utilised in the training experiments. The descent algorithm and mini-batch size were set as stochastic gradient descent method (SGDM) and 16, respectively. According to the training results, it is concluded that a larger learning rate enables the model to learn faster but brings with it a risk of sub-optimal

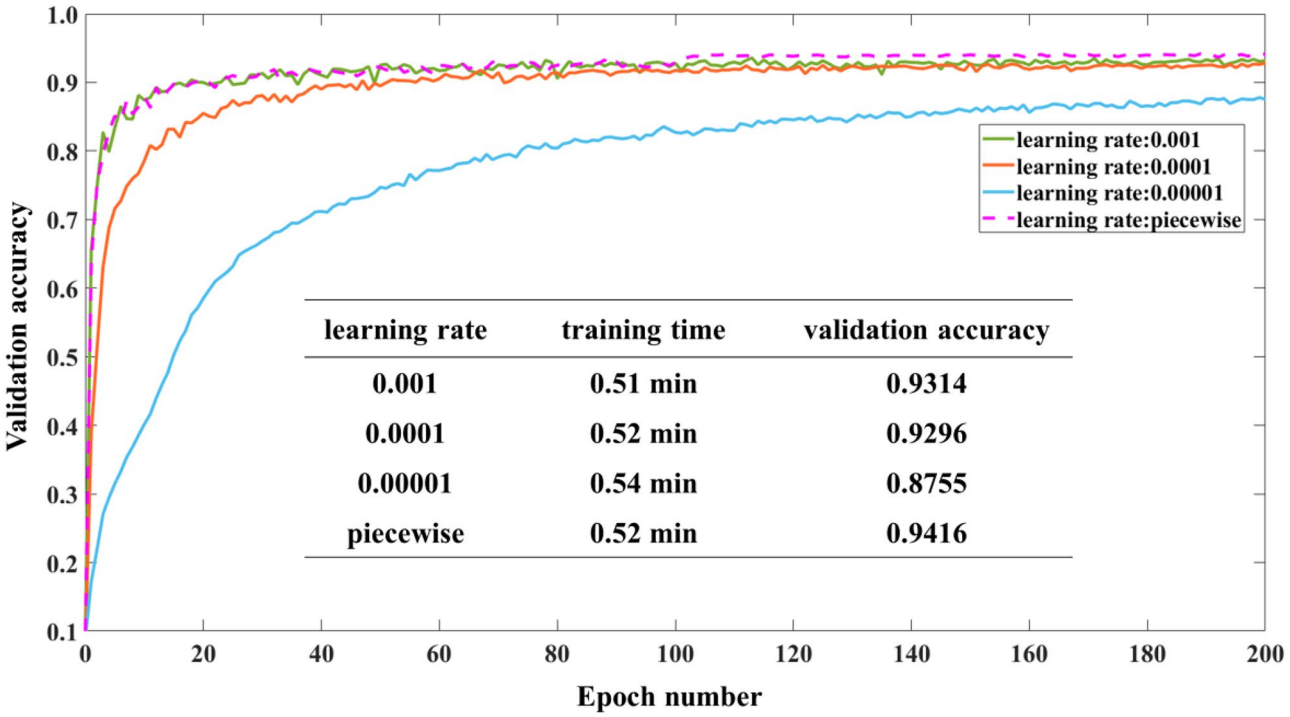


Fig. 5 Iteration process of networking training with different learning rates

results. When the learning rate becomes smaller, the convergence speed becomes lower in the initial stage, and it takes a longer time to reach the stable stage. In particular, if the learning rate is too small (e.g. 0.00001), the final validation accuracy is relatively lower after 200 training epochs. Therefore, the piecewise learning rate from 0.001 to 0.0001, which combines the helpful characteristics of the larger and smaller learning rates, can bring about a fast convergence in the beginning and ensure a high validation accuracy in the final stage. Figure 5 also presents the average training time for single epoch, indicating that the influence of learning rate variations on training time is negligible.

2. Gradient algorithm: The gradient descent algorithm is used in the training of deep neural networks [45, 46]. This section compares the training performance of two typical gradient algorithms, SGDM and Adam, and selects the appropriate algorithm by considering the convergence speed, computation efficiency and generalisation ability. Compared with the traditional gradient descent algorithms, SGDM computes the gradient of the loss function only by a small random subset, instead of the whole dataset, and performs a parameter update, which can help to improve the computation efficiency. The Adam algorithm utilises squared gradients to scale the learning rate and takes advantage of momentum by the moving average of gradient. Figure 6 presents the training process of the two algorithms with the batch size and learning rate fixed at 16 and piecewise, respectively.

Because of the random gradient computation, the SGDM usually has a lower convergence speed in the beginning and reaches its stable stage after a higher number of iterations compared with the Adam algorithm. However, the former consumes less training cost and leads to a higher validation accuracy than the latter. This is because more frequent updates are conducted for SDGM. Hence, there are more chances to jump out of a local minimal and search for better solutions. Hence, the SGDM algorithm was adopted in this study.

3. Mini-batch size: In training the CNN networks, the scale of image database is usually very large. The computational cost will be unaffordable if the whole database is swept in each iteration. Hence, a proper selection of a mini-batch size is important to reduce the training cost and refine classification performance [47]. By using the SGDM and piecewise learning, Fig. 7 demonstrates how the mini-batch size affects the training cost and validation accuracy. In general, the influence of batch size variations is more significant at the beginning of network training.

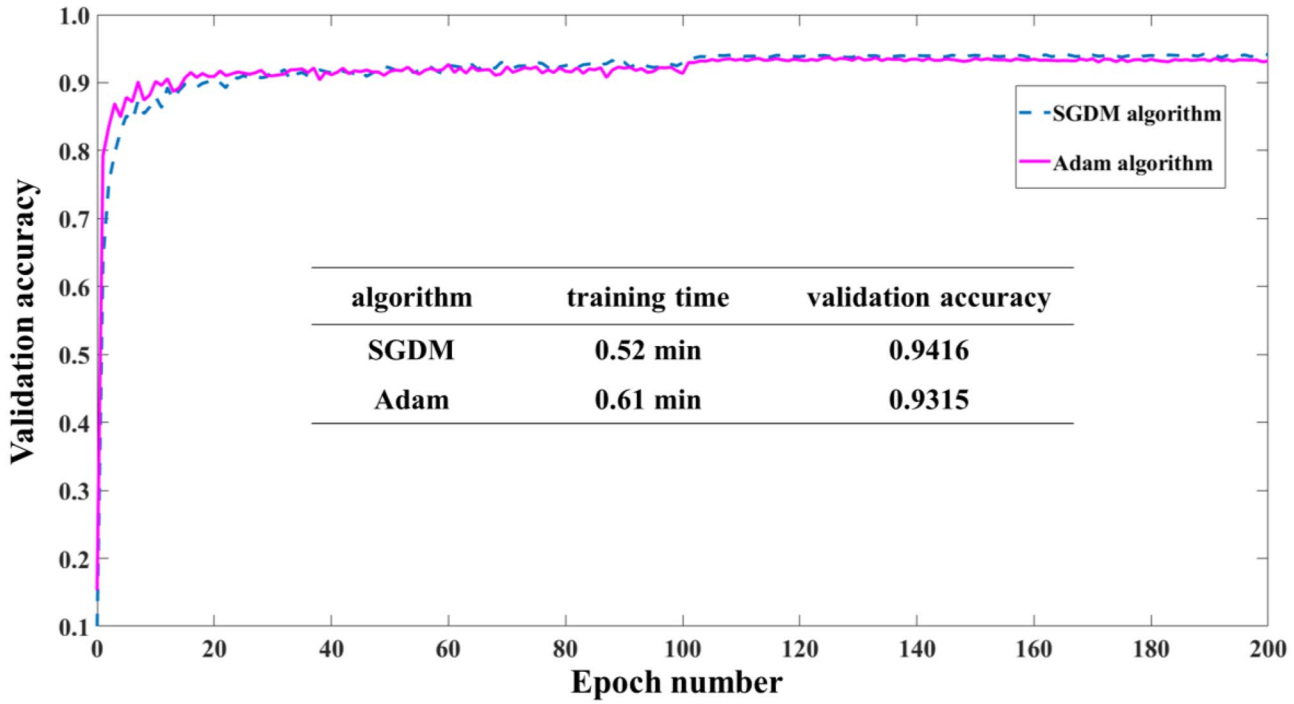


Fig. 6 Iteration process of networking training with different gradient algorithms

It is noted that a smaller batch size usually brings about a faster convergence speed because the network parameters are updated more frequently within each epoch. Meanwhile, more iteration steps related to a smaller size also bring about more training time. However, overfitting should be taken into consideration if the batch size is too small. It is also found that a larger batch size may lead to poorer generalisation ability. For example, as shown in Fig. 7, the validation accuracy drops gradually when the batch size increases from 16 to 64. With considering the balance between the training cost and validation accuracy listed in Fig. 7, the mini-batch size will be fixed at 16.

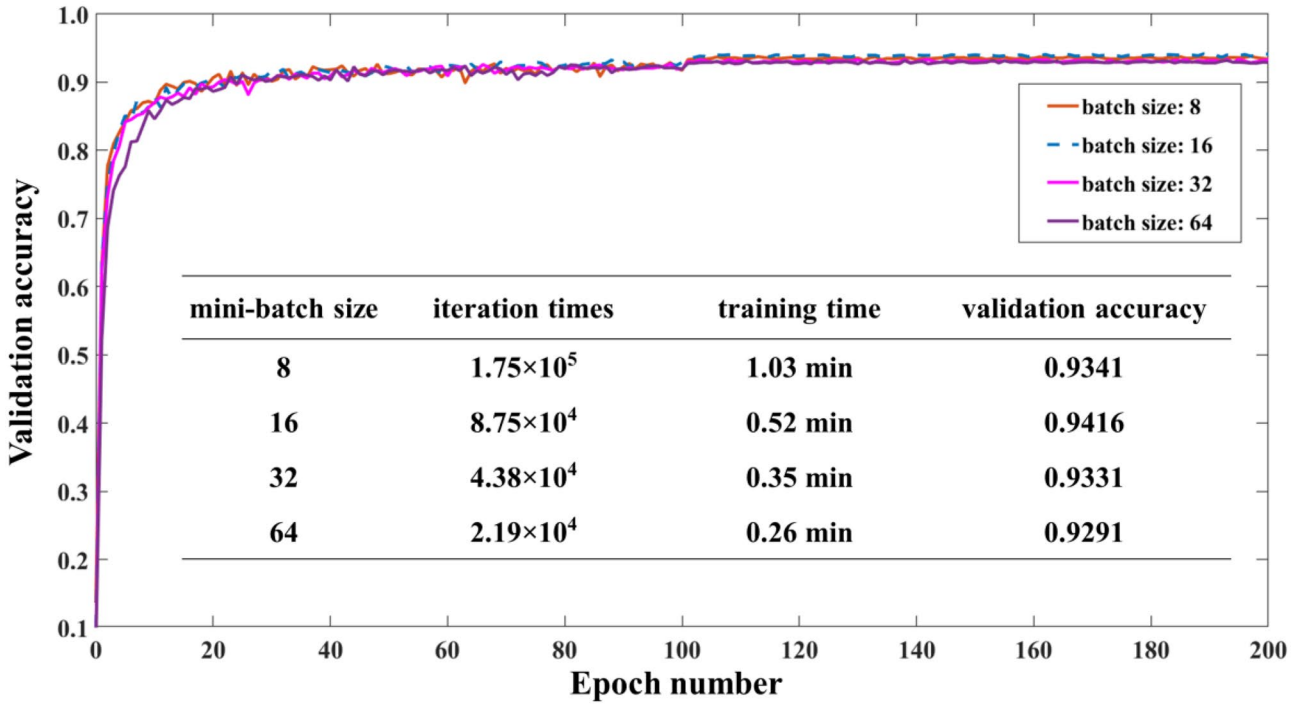


Fig. 7 Iteration process of networking training with different mini-batch sizes

Table 4 Surface scratch images used for *t-SNE* plotting

	Intact surface	*Material transfer*	*Severe scratch*
QP980	50	50	50
DP980	50	50	50

4.2 Examination of developed *WearNet*

With the above selected parameters, the *WearNet* for surface scratch detection was trained. The network responses, layer activations and decision mechanism of the *WearNet* are as follows:

1. Network responses: To examine the *WearNet* responses, a *t*-distributed stochastic neighbour embedding (*t-SNE*) function [48] was used in the study. Three hundred surface images with six different labels, as listed in Table 4, were used to investigate the responses of different layers, i.e. maxpool-1, conv-2 and *softmax*, in the *WearNet*. Figure 8 illustrates the *t-SNE* plots for three different layers where the six colours of these solid dots refer to the six image labels. For the maxpool-1 layer, the labels were not correctly grouped because only low-level features, e.g. colours and edges, were operated in such an early layer. The conv-2 layer can refine the cluster of these labels to some extent, but the accuracy was not satisfactory. For the *softmax* layer, the *t-SNE* plotting demonstrates that an appropriate classification of these different labels was achieved as the network went deeper, which validated the high accuracy of the developed *WearNet*.

2. Examination of layer activations: The layer activations play an important role in training the *WearNet*. To check which features the network has learned and whether the representative features have been correctly detected and preserved, it is necessary to visualise and examine the activation maps within different layers, i.e. conv-1, squeeze layer in conv-block-2 and conv-2. A testing image, QP980 surface with *severe scratch*, was fed into the trained *WearNet*.

There are 32 channels in the first convolution layer (conv-1). The 32 image features corresponding to the 32 channels are shown in Fig. 9a. Similarly, Fig. 9b, c present the feature

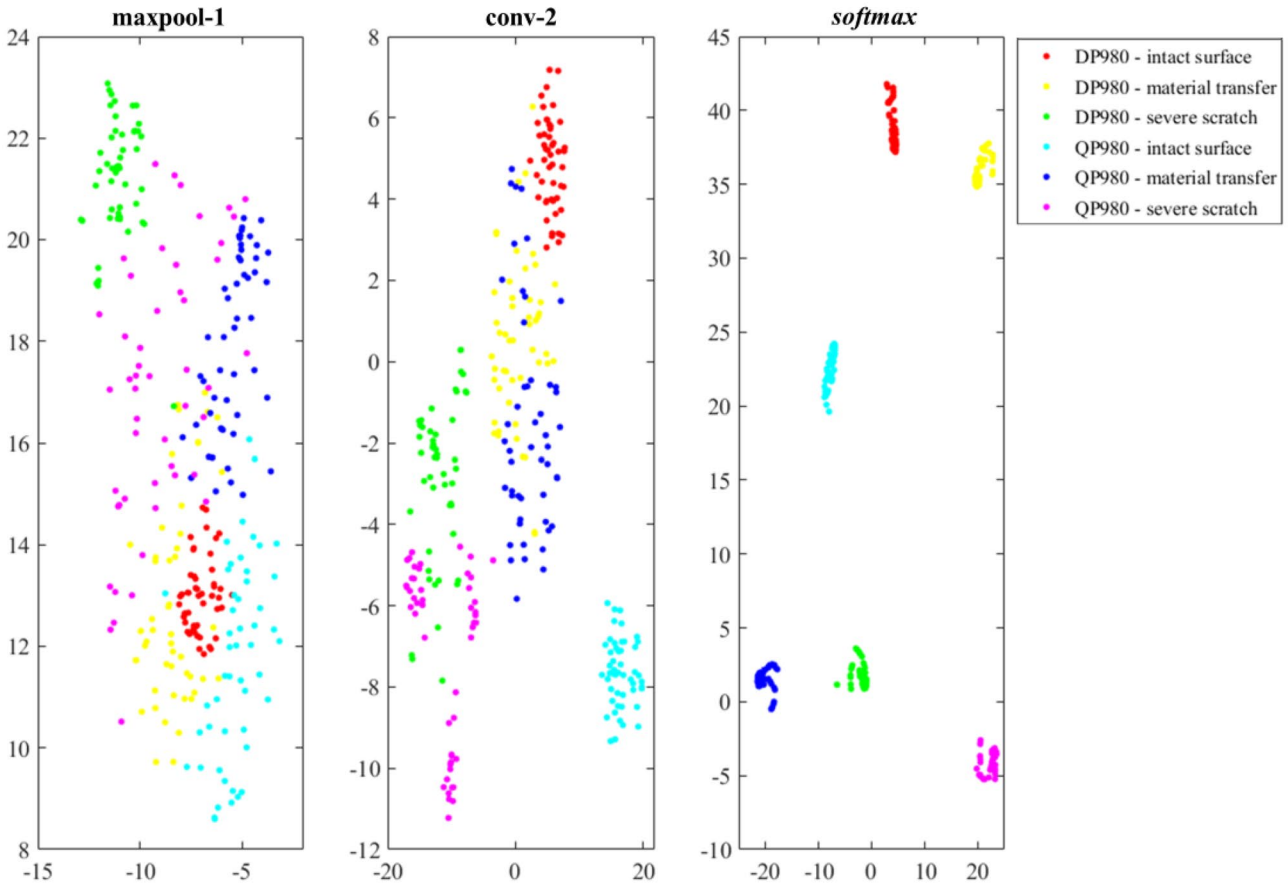
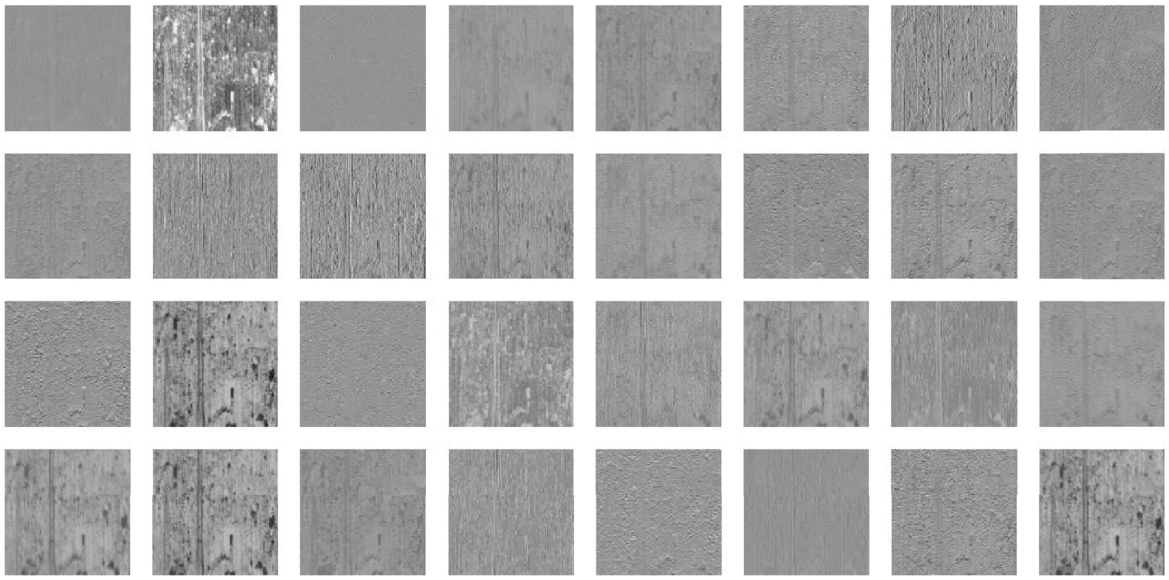
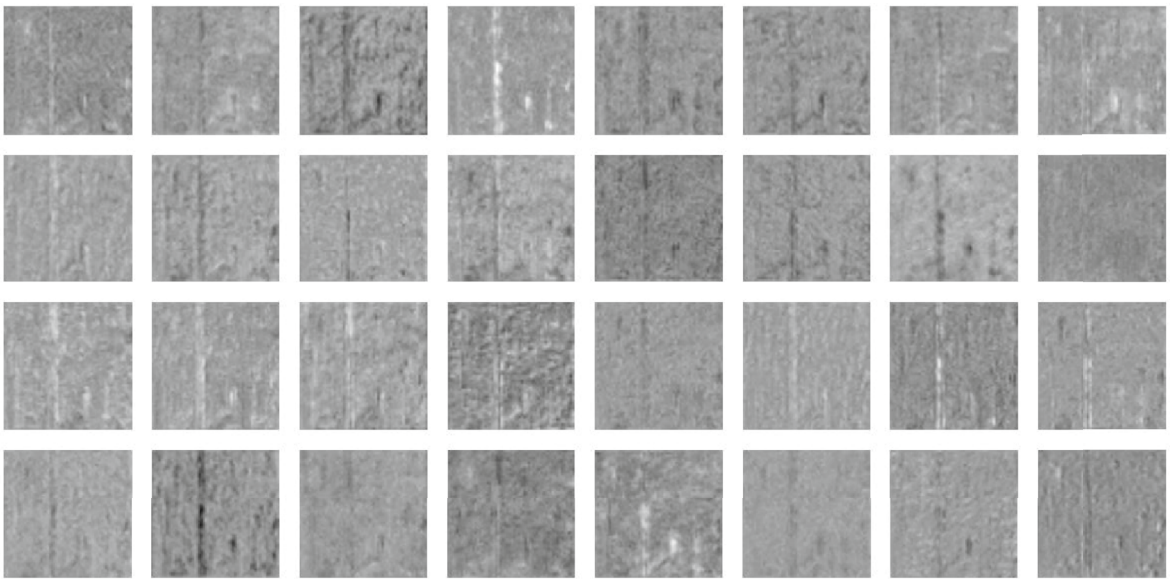


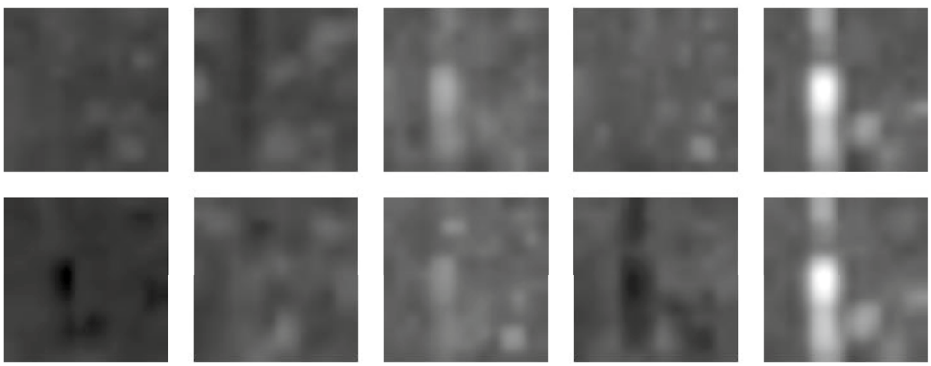
Fig. 8 *t-SNE* plotting for different layers in *WearNet*: maxpool-1, conv-2 and *softmax*



(a) conv-1 layer with 32 channels

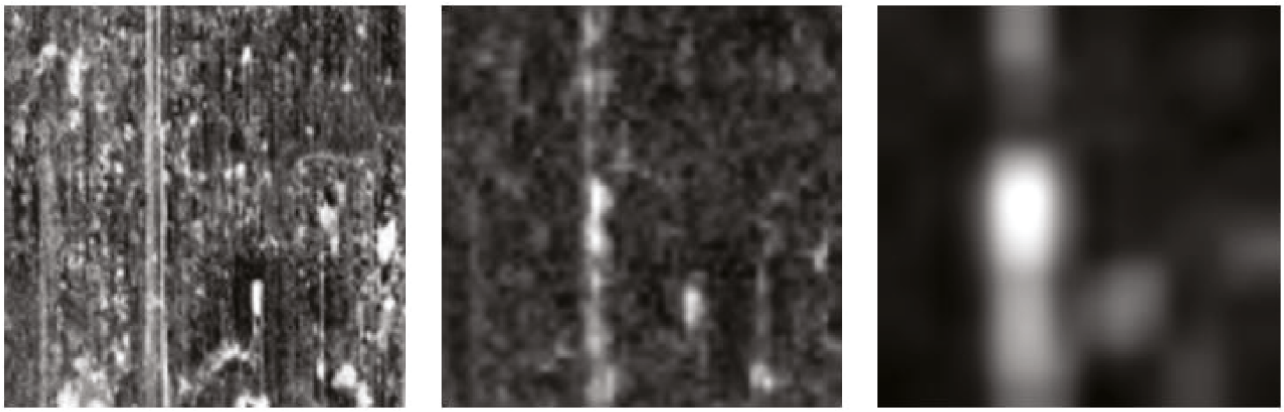


(b) squeeze layer in conv-block-2 with 32 channels



(c) conv-2 layer with 10 channels

Fig. 9 Feature maps from three different layers



(a) conv-1 layer **(b)** squeeze layer in conv-block-2 **(c)** conv-2 layer

Fig. 10 Image features with the largest activations in three different layers

maps of the squeeze layer in the conv-block-2 and the last convolution layer (conv-2). The numbers of channels of the squeeze layer and conv-2 layer are 32 and 10, respectively. Figure 10 illustrates the image features with the largest activations in the three layers. Clearly, the discriminative features for characterising the *severe scratch* were extracted step by step as the neural network went deeper. For example, the deep and long ploughings were typical features for severe scratching images (Fig. 10).

3. Decision mechanism: To figure out how the *WearNet* makes a reliable classification decision, the gradient-weighted class activation mapping, Grad-CAM [49], technique was employed. It utilises the gradient of the final classification scores associated with the convolutional features to determine the most influential part of a tested image for the classification. Figure 11 illustrates the Grad-CAM map for a test image from the class of QP980-*severe scratch*. The regime with the blue refers

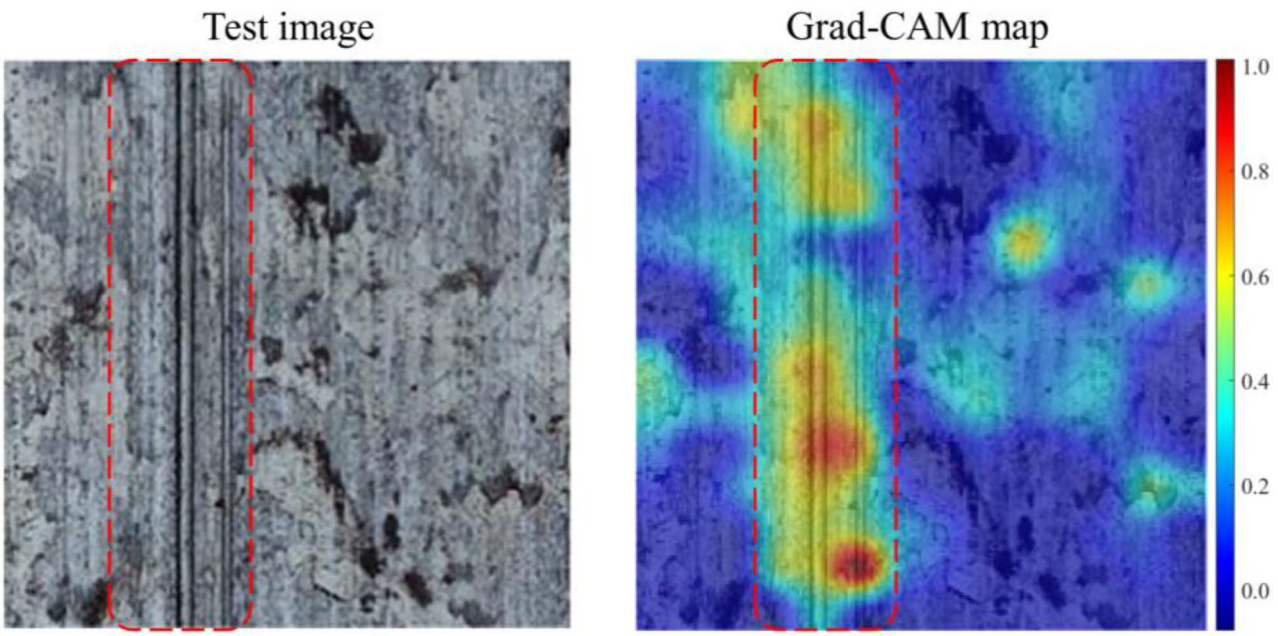


Fig. 11 Grad-CAM gradient map for a test image (QP980-*severe scratch*)

Table 5 Comparison in the network performance

	Training time	Validation accuracy	Test accuracy
AlexNet	0.75 min	0.9041	0.8926
EfficientNet	6.37 min	0.8833	0.8777
MobileNet	3.68 min	0.8998	0.8869
SqueezeNet	1.27 min	0.9233	0.9177
WearNet	0.52 min	0.9416	0.9297

to a low influence while the regime with the red denotes a high effect. In general, a larger gradient corresponds to the red zone on which the final score most relies. It should be noted that the red rectangle zone also refers to the area with a deep and long scratch, which has the greatest impact on classifying the test image as *severe scratch*.

4.3 Comparison with other CNN networks

A series of classification experiments were conducted to compare the performance of the *WearNet* with other state-of-the-art networks when the training conditions, i.e. image dataset (see Table 2), training platform and parameter settings, were identical. Table 5 compares the network performance in terms of training time, validation and testing accuracy. Here, the training time refers to the average iteration time for a single epoch. It can be found that the validation and test accuracy of *WearNet* outperform the others while the minimum training time is consumed. Figure 12 presents the evolution of validation accuracy during the network training, in which *WearNet* has the highest value throughout the

Table 6 Comparison in recognition rates of individual image labels

No.	Workpiece	Label	AlexNet	MobileNet	SqueezeNet	*WearNet*
1	DP980	*Intact surface*	0.95	0.98	0.99	0.99
2		*Material transfer*	0.99	0.99	0.99	0.99
3		*Minor scratch*	0.95	0.94	0.97	0.98
4		*Mild scratch*	0.93	0.88	0.94	0.94
5		*Severe scratch*	0.65	0.73	0.76	0.83
6	QP980	*Intact surface*	0.99	0.99	0.99	1.00
7		*Material transfer*	0.87	0.87	0.87	0.90
8		*Minor scratch*	0.82	0.79	0.83	0.85
9		*Mild scratch*	0.78	0.78	0.87	0.86
10		*Severe scratch*	0.91	0.93	0.93	0.96

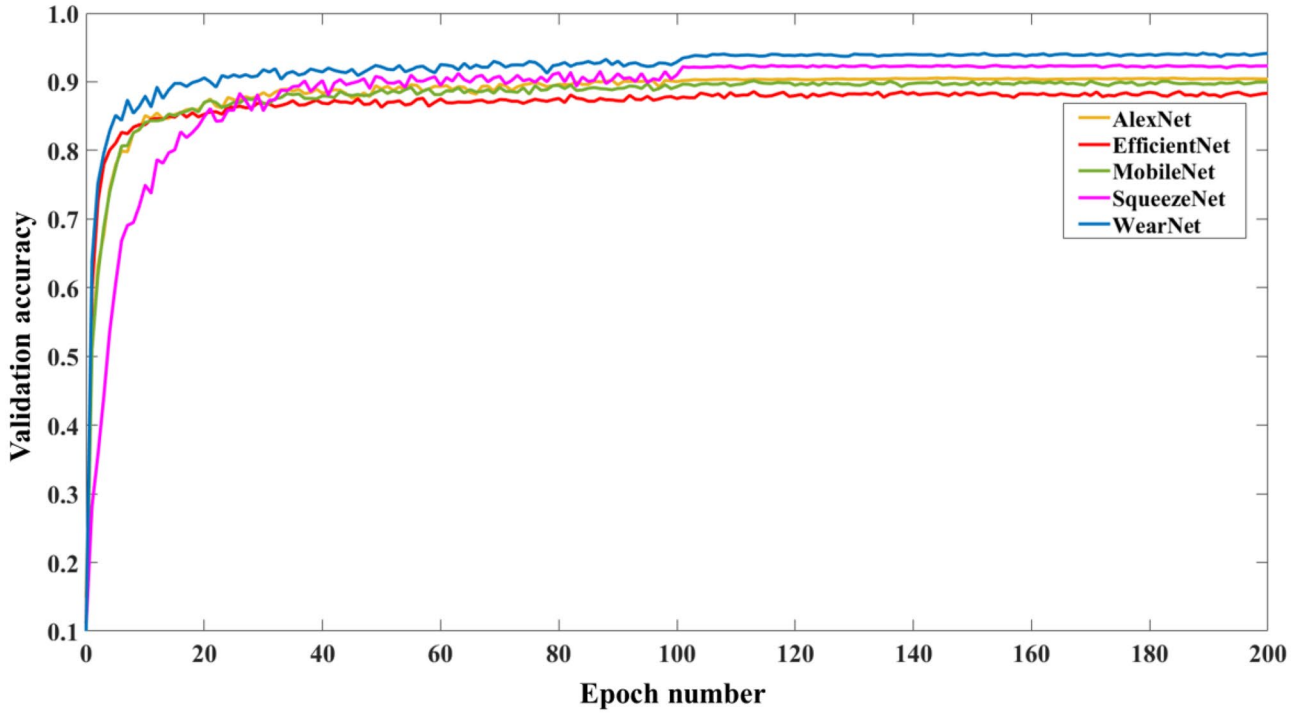
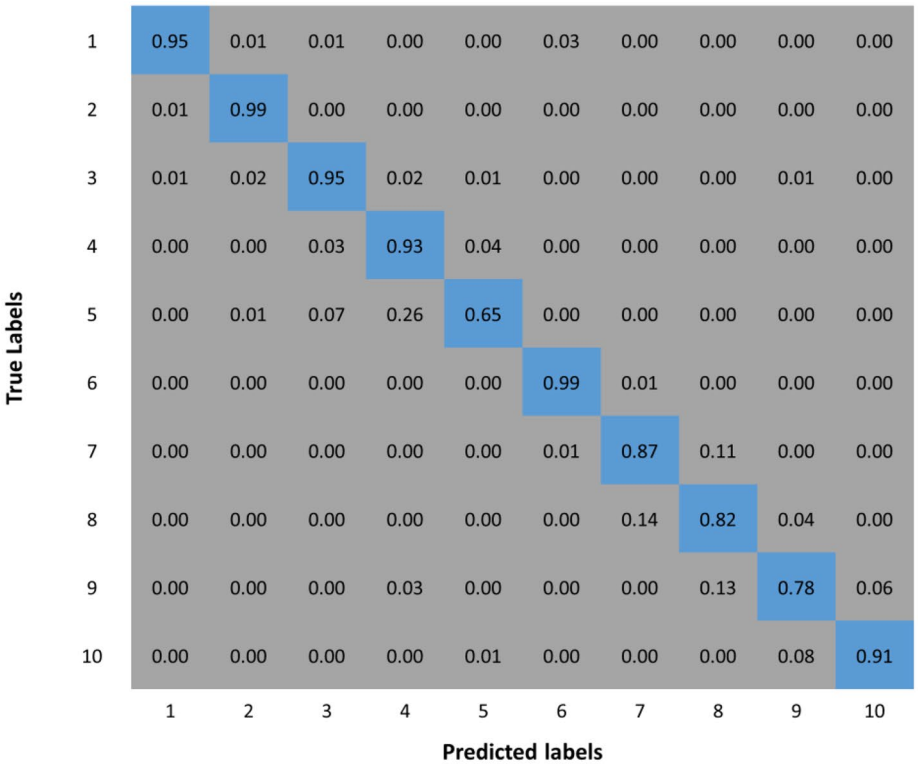
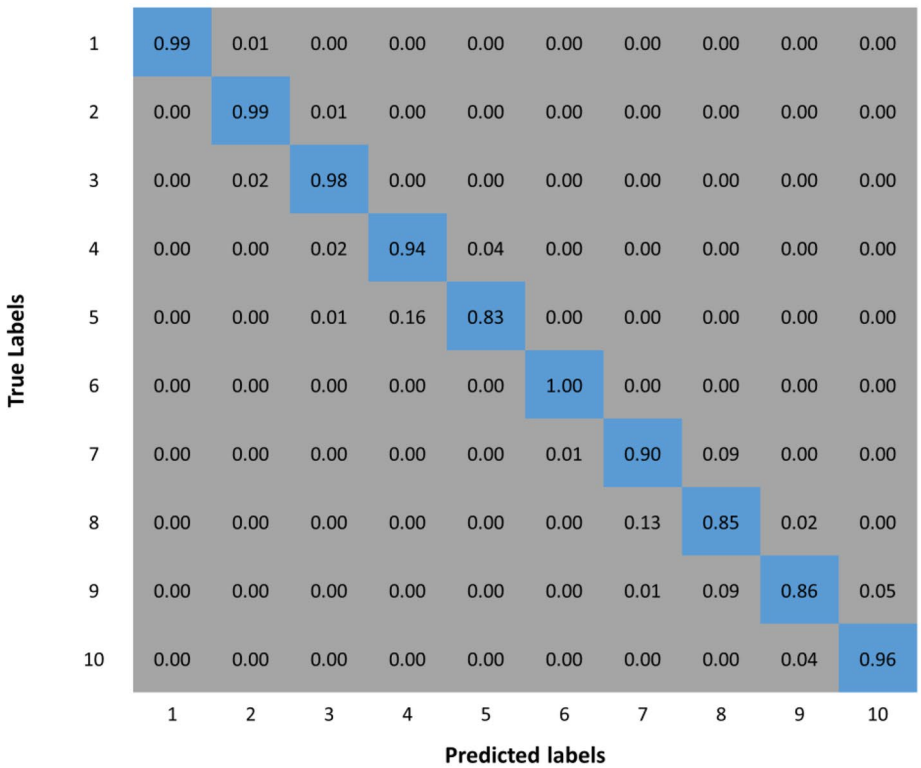


Fig. 12 Training process of *WearNet* and other state-of-the-art networks



(a) AlexNet



(b) *WearNet*

Fig. 13 Confusion matrices for AlexNet and *WearNet*

Table 7 Comparison in the complexity of different CNN networks

	Layer number	Parameter quantity	Model size	Classification time
AlexNet	25	61.0 M	201 MB	1.36 ms
EfficientNet	290	5.31 M	18.4 MB	4.80 ms
MobileNet	154	3.50 M	8.19 MB	2.80 ms
SqueezeNet	68	1.24 M	2.59 MB	0.65 ms
WearNet	48	0.16 M	0.54 MB	0.58 ms

whole training process. Furthermore, Table 6 presents the recognition rates of individual image labels when different CNN networks are tested. In general, the recognition rates of most image labels are over 85%. When it comes to the labels DP980-*severe scratch* and QP980-*mild scratch*, the *WearNet* is still able to provide reliable classification results, while the recognition rates related to other networks drop significantly, especially AlexNet. Therefore, Fig. 13 shows the confusion matrices for AlexNet and *WearNet*, which can help to figure out how two CNN networks are confused when making classification decisions. It can be found that for the scratch images with the label of DP980-*severe scratch*, AlexNet is able to classify them with a recognition rate of only 65%, while around 7% and 26% are incorrectly classified as DP980-*minor scratch* and DP980-*mild scratch*, respectively. However, when the *WearNet* is employed, the classification error can be reduced significantly, as shown in Fig. 13b.

Table 7 compares the complexity of *WearNet* and other CNN networks in terms of layer number, parameter quantity, model size and classification time. It is found that the model size is closely related to the quantity of network parameters, while the number of network layers has a considerable impact on its classification time. As shown in Table 7, the structure of *WearNet* is simpler than that of its convolutional counterparts, which brings about the smallest model size and fastest classification speed. However, the *WearNet* still has excellent classification performance, which should be attributed to its well-designed lightweight architecture. In conclusion, the *WearNet* proposed in this study has shown its advantages in computational efficiency and model size, as well as in its excellent classification performance.

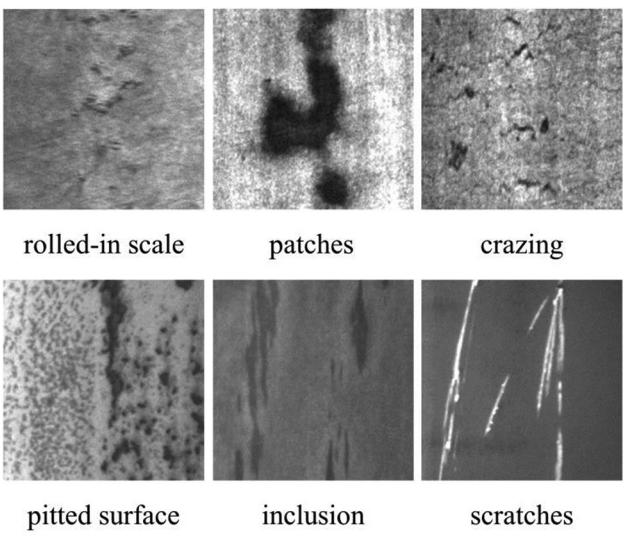


Fig. 14 Surface defects of hot-rolled steel strips in the NEU database

4.4 Classification performance on a public dataset

To comprehensively demonstrate the effectiveness of *WearNet*, its performance is investigated based on the NEU surface defect dataset. This dataset collects six kinds of typical surface defects on hot-rolled steel strips (see Fig. 14), i.e. rolled-in scale, patches, crazing, pitted surface, inclusion and scratches, with a set of 300 labelled images for each type. The surface defect images in the database are greyscale and will be converted into RGB images before being used for network training. All the images were randomly divided into training, validation and testing datasets with a ratio of 4:1:1 (1200:300:300). A smaller training epoch (100) was adopted due to the smaller size of the image database, while other training parameters were identical to those in the last section. Table 8 presents the classification performance of four different CNN networks. All the four networks achieved high validation and test accuracy (over 98%), but the *WearNet*

Table 8 Classification performance on NEU surface defect database

	Training time	Validation accuracy	Test accuracy	Classification time
AlexNet	0.11 min	0.9980	0.9947	7.45 ms
MobileNet	0.73 min	0.9833	0.9767	7.63 ms
SqueezeNet	0.13 min	0.9833	0.9800	3.97 ms
WearNet	0.12 min	0.9987	0.9967	3.62 ms

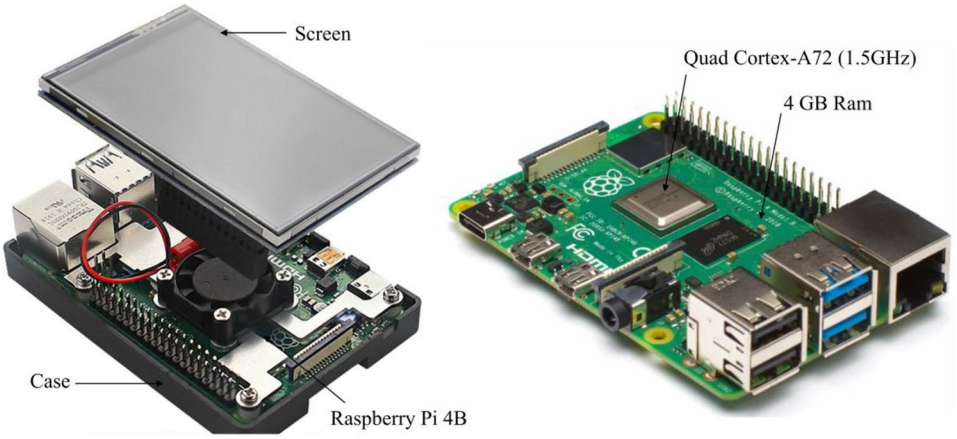


Fig. 15 Illustration of an embedded system

(a) system setup **(b)** structure of Raspberry Pi 4B

outperformed others in terms of training time and classification speed.

5 Deployment of CNN networks

Currently, embedded systems are widely used in the industry due to their advantages in high efficiency, good affordability, continuous production and low energy consumption. In this study, a Linux-based embedded system, in which Raspberry Pi 4B works as the core hardware (see Fig. 15), was used to further demonstrate the application of CNN networks. In this section, 600 surface images selected from the testing dataset were used in the surface defection test.

Table 9 compares the classification of surface scratch in the embedded system with four different CNN structures. The folder size refers to the total size of whole configuration files, which enables the embedded system to run detection programmes independently. In the embedded environment, the folder size and classification time of the *WearNet* were significantly lower than others, while its testing accuracy was still the highest. Hence, it is expected that *WearNet* will have promising prospects in industrial production.

Table 9 Comparison of network performance in embedded system

	Test accuracy	Classification time	Folder size
AlexNet	0.8962	308 ms	419 MB
MobileNet	0.9029	554 ms	17.5 MB
SqueezeNet	0.9170	272 ms	6.35 MB
WearNet	0.9274	225 ms	1.24 MB

6 Conclusions

In this study, a lightweight CNN structure, called *WearNet*, has been developed based on the well-designed convolutional block. The *WearNet* is designed for surface scratch detection in contact sliding, and the surface scratch images in the database are collected from cylinder-on-flat tribological tests. A detailed investigation on the parameter selection for network training and examinations on the network response and decision mechanism have been carried out. The performance comparison between the *WearNet* and other commonly used CNN networks has been conducted by using different databases. The main contributions of this paper are summarised as follows:

1. The developed lightweight *WearNet* has minimised network layers and parameters, with distinguished advantages in model size and classification speed, while guaranteeing high classification accuracy and recognition rate.
2. Training parameter variations have a significant influence on the network training process. The selected combination of training parameters manages to achieve a good balance between computation consumption and network performance.
3. The developed *WearNet* is able to extract and learn discriminative features for surface scratch classification step by step. The examination results demonstrate the excellent capability of *WearNet* to correctly classify different scratch images with appropriate labels.
4. The application of *WearNet* in an embedded system shows that *WearNet* has promising application prospects in production.

Author contribution LZ initiated and supervised the project. WL carried out the experiments for data collection, developed and optimised the deep neural network under the supervision of CW and LZ. WL prepared the manuscript draft; CW and LZ revised the manuscript and contributed to the discussions. ZC and CN provided the experiment materials and made helpful suggestions on the experiment planning. All authors have read and agreed to the published version of the manuscript.

Funding This research was supported by the Baosteel Australia Research and Development Centre (BAJC) portfolio with Project BA17001, the ARC Hub for Computational Particle Technology IH140100035, the Chinese Guangdong Specific Discipline Project 2020ZDZX2006 and the Shenzhen Key Laboratory Project of Cross-scale Manufacturing Mechanics ZDSYS20200810171201007.

Availability of data and materials Queries about data and materials should be addressed to L.Z (zhanglc@sustech.edu.cn).

Code availability Queries about codes should be addressed to L.Z (zhanglc@sustech.edu.cn).

Declarations

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent to publication Not applicable.

Conflict of interest The authors declare no competing interests.

References

1. Sun J, Wang P, Luo YK et al (2019) Surface defects detection based on adaptive multiscale image collection and convolutional neural networks. IEEE Trans Instrum Meas 68:4787–4797
2. Zheng X, Zheng S, Kong Y et al (2021) Recent advances in surface defect inspection of industrial products using deep learning techniques. Int J Adv Manuf 113:35–58
3. Zhang X, Wang H, Stojanovic V et al (2022) Asynchronous fault detection for interval type-2 fuzzy nonhomogeneous higher-level markov jump systems with uncertain transition probabilities. IEEE Trans Fuzzy Syst 30:2487–2499
4. Song X, Sun P, Song S et al (2022) Event-driven NN adaptive fixed-time control for nonlinear systems with guaranteed performance. J Franklin Inst 359(9):4138–4159
5. Xu Z, Li X, Stojanovic V et al (2021) Exponential stability of nonlinear state-dependent delayed impulsive systems with applications. Nonlinear Anal Hybrid Syst 42:101088
6. Wang S, Wang H, Yang F et al (2022) Attention-based deep learning for chip-surface-defect detection. Int J Adv Manuf Technol 121:1957–1971
7. Jia H, Murphey YL, Shi J et al (2004) An intelligent real-time vision system for surface defect detection. Proc IEEE Int Conf Pattern Recognit 239–42
8. García-Ordás MT, Alegre E, González-Castro V et al (2017) A computer vision approach to analyze and classify tool wear level in milling processes using shape descriptors and machine learning techniques. Int J Adv Manuf 90:1947–1961
9. Yazdchi MR, Mahyari AG, Nazeri A (2008) Detection and classification of surface defects of cold rolling mill steel using morphology and neural network. Proce IEEE Int Conf Comput Intell Model Control Autom 1071–1076
10. Tseng DC, Chung IL, Tsai PL et al (2011) Defect classification for LCD color filters using neural-network decision tree classifier. Int J Innov Comput Inf Control 7:3695–3707
11. Müller A, Karathanasopoulos N, Roth CC et al (2021) Machine learning classifiers for surface crack detection in fracture experiments. Int J Mech Sci 209:106698
12. Nasir V, Sassani F (2021) A review on deep learning in machining and tool monitoring: methods, opportunities, and challenges. Int J Adv Manuf Technol 115:2683–2709
13. Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
14. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
15. Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. Proc IEEE Conf Comput Vis Pattern Recognit 1–9
16. Tan M, Le QV (2019) EfficientNet: rethinking model scaling for convolutional neural networks. arXiv:1905.1194
17. He Z, Liu Q (2020) Deep regression neural network for industrial surface defect detection. IEEE Access 8:35583–35591
18. Tabernik D, Šela S, Skvarč J et al (2020) Segmentation-based deep-learning approach for surface-defect detection. J Intell Manuf 31:759–776
19. Hu M, Wang G, Ma Z et al (2021) Bearing performance degradation assessment based on optimized EWT and CNN. Measurement 172:108868
20. Song K, Yan Y (2013) A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci 285:858–864
21. Wang M, Yang L, Zhao Z et al (2022) Intelligent prediction of wear location and mechanism using image identification based on improved Faster R-CNN model. Tribol Int 169:107466
22. Wang Y, Xiao Z, Cao G (2022) A convolutional neural network method based on Adam optimizer with power-exponential learning rate for bearing fault diagnosis. J Vibroeng 24(2):666–678
23. Iandola FN, Han S, Moskewicz MW et al (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. arXiv:1602.07360
24. Howard AG, Zhu M, Chen B et al (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
25. Zhang X, Zhou X, Lin M et al (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. Proc IEEE/CVF Conf Comput Vis Pattern Recognit 6848–6856
26. Lee HJ, Ullah I, Wan W et al (2019) Real-time vehicle make and model recognition with the residual SqueezeNet architecture. Sensors 19:982
27. Izidio DM, Ferreira A, Medeiros HR et al (2020) An embedded automatic license plate recognition system using deep learning. Des Autom Embed Syst 24:23–43
28. Saponara S, Elhanashi A, Gagliardi A (2021) Real-time video fire/smoke detection based on CNN in antifire surveillance systems. J Real Time Image Process 18:889–900
29. Ghaffari S, Sharifian S (2016) FPGA-based convolutional neural network accelerator design using high level synthesize. Proc IEEE Int Conf Signal Process Intell Syst 1–6
30. Ullah A, Muhammad K, Ding W et al (2021) Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. Appl Soft Comput 103:107102
31. Lee H, Kim Y, Kim M et al (2021) Low-cost network scheduling of 3D-CNN processing for embedded action recognition. IEEE Access 9:83901–83912
32. Paluru N, Dayal A, Jenssen HB et al (2021) Anam-Net: anamorphic depth embedding-based lightweight CNN for segmentation

of anomalies in COVID-19 chest CT images. IEEE Trans Neural Netw Learn Syst 32:932–946
33. Li W, Zhang LC, Wu CH et al (2022) Influence of tool and workpiece properties on the wear of the counterparts in contact sliding. J Tribol 144:021702
34. Li W, Zhang LC, Wu CH et al (2022) Debris effect on the surface wear and damage evolution of counterpart materials subjected to contact sliding. Adv Manuf 10:72–86
35. Li W, Zhang LC, Chen XP et al (2020) Fuzzy modelling of surface scratching in contact sliding. IOP Conf Ser Mater Sci Eng 967:012022
36. Li W, Zhang LC, Chen XP et al (2021) Predicting the evolution of sheet metal surface scratching by the technique of artificial intelligence. Int J Adv Manuf 112:853–865
37. Deng J, Dong W, Socher R et al (2009) Imagenet: a large-scale hierarchical image database. Proc IEEE Conf Comput Vis Pattern Recognit 248–255
38. Guillaumin M, Küttel D, Ferrari V (2014) Imagenet auto-annotation with segmentation propagation. Int J Comput Vis 110:328–348
39. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
40. Barrionuevo GO, Ramos-Grez JA, Walczak M et al (2021) Comparative evaluation of supervised machine learning algorithms in the prediction of the relative density of 316L stainless steel fabricated by selective laser melting. Int J Adv Manuf Technol 113:419–433
41. Dou J, Xu C, Jiao S et al (2020) An unsupervised online monitoring method for tool wear using a sparse auto-encoder. Int J Adv Manuf Technol 106:2493–2507
42. Xin X, Tu Y, Stojanovic V et al (2022) Online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems. Appl Math Comput 412:126357
43. Xia C, Pan Z, Li Y et al (2022) Vision-based melt pool monitoring for wire-arc additive manufacturing using deep learning method. Int J Adv Manuf Technol 120:551–562
44. Kuo JK, Wu JJ, Huang PH et al (2022) Inspection of sandblasting defect in investment castings by deep convolutional neural network. Int J Adv Manuf Technol 120:2457–2468
45. Kingma DP, Ba J (2017) Adam: a method for stochastic optimization. arXiv:1412.6980
46. Cheridito P, Jentzen A, Rossmannek F (2021) Non-convergence of stochastic gradient descent in the training of deep neural networks. J Complex 64:101540
47. Radiuk PM (2017) Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. Inf Technol Manag 20:20–24
48. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
49. Selvaraju RR, Cogswell M, Das A et al (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. Proc IEEE Int Conf Comput Vis 618–626

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.