Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.027035

ARTICLE

# Classification-Detection of Metal Surfaces under Lower Edge Sharpness Using a Deep Learning-Based Approach Combined with an Enhanced LoG Operator

**Hong Zhang[1,*], Jiaming Zhou[1], Qi Wang[1], Chengxi Zhu[1] and Haijian Shao[2]**

[1]School of Electrical Information Engineering, Jiangsu University of Technology, Changzhou, 213001, China

[2]Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, NV 89154, USA

*Corresponding Author: Hong Zhang. Email: hazh0216@jsut.edu.cn

Received: 11 October 2022 Accepted: 06 February 2023 Published: 28 June 2023

## ABSTRACT

Metal flat surface in-line surface defect detection is notoriously difficult due to obstacles such as high surface reflectivity, pseudo-defect interference, and random elastic deformation. This study evaluates the approach for detecting scratches on a metal surface in order to address a problem in the detection process. This paper proposes an improved Gauss-Laplace (LoG) operator combined with a deep learning technique for metal surface scratch identification in order to solve the difficulties that it is challenging to reduce noise and that the edges are unclear when utilizing existing edge detection algorithms. In the process of scratch identification, it is challenging to differentiate between the scratch edge and the interference edge. Therefore, local texture screening is utilized by deep learning techniques that evaluate and identify scratch edges and interference edges based on the local texture characteristics of scratches. Experiments have proven that by combining the improved LoG operator with a deep learning strategy, it is able to effectively detect image edges, distinguish between scratch edges and interference edges, and identify clear scratch information. Experiments based on the six categories of meta scratches indicate that the proposed method has achieved rolled-in crazing (100%), inclusion (94.4%), patches (100%), pitted (100%), rolled (100%), and scratches (100%), respectively.

## KEYWORDS

Deep learning; gaussian-laplace algorithm; texture feature; scratch detection

## Nomenclature

| | |
|---|---|
| $\sigma$ | Spatial distribution coefficient of the Gaussian function |
| $*$ | Convolutional operation |
| $h(x, y)$ | Image obtained after smoothing filtering |
| $\nabla^2$ | Represents the Laplace operator |
| $\nabla^2 G(x, y)$ | LoG operator |
| $h\left(x, y, \sigma_x, \sigma_y, \alpha\right)$ | Two-dimensional Gaussian function |
| $\beta_n$ | $n$th weight value of the $n$th template |

$\sigma_{xn}$ The standard deviation in the $X$ direction
$\sigma_{yn}$ The standard deviation in the $Y$ direction

## 1 Introduction

The production and operation environment of metal materials is relatively hostile, resulting in various surface blemishes and faults that negatively influence product quality and corporate productivity. Consequently, it is essential to automatically detect flaws on metal surfaces. Scratches are prominent manufacturing defects, and the detection of surface scratches is prevalent in the quality inspection of modern industrial products [1–3]. In-line surface defect identification of metal planar materials is confronted with the following formidable manufacturing challenges: (1) High surface reflectivity under lower edge sharpness: The smooth surface of steel, aluminum, copper strip, and other extremely thin strips makes it simple to bring together large areas of light and shadow., grayscale inconsistencies increase the likelihood of false edge detection. (2) Pseudo-defect interference: Pseudo-defects (such as water droplets, water cloth, rainlines, water mist, and other true defects formed during laminar flow cooling) can frequently trigger false alarms in the detecting equipment. Random elastic deformation will cause random image distortion in charge-coupled device (CCD) cameras [4]. This is caused by things like continuous rolling equipment vibration, insufficient rolling speed, side guide plate ectopicality, rolling speed fluctuations, atmospheric turbulence-like effects, and other things [5].

The utilization of computer vision-based product fault detection technologies is expanding. Image edge detection is a crucial component of Technology for detecting product defects. Picture edge is a characteristic of image segmentation and an essential source of texture data. Because noise and edges are both high-frequency components in the frequency domain and the spatial domain is manifested by a large abrupt change in gray value, it is frequently difficult to avoid enhancing the noise during edge detection, which frequently causes issues with image segmentation and target detection and extraction [6–8]. The surface defect detection methods of metal plane materials are mainly divided into four categories: methods based on statistical and spectral analysis. methods based on machine learning. and hybrid methods based on deep learning.

1) Detection methods based on statistical and spectral analysis. From the perspective of statistical methods, visual textures are random events. Statistical approaches investigate the regular and periodic distribution of pixel intensities by analyzing the statistical characteristics of the spatial distribution of pixels in order to discover flaws on the surface of flat metal surfaces. In the case of light variations and pseudo-defect interference, statistical approaches have limited application. Yang et al. [9] have discovered that images with different characteristics are more likely to be isolated in the transformation domain and that it is possible to find better defect detection methods than direct processing methods in the pixel domain Consequently, the use of approaches based on spectrum analysis is growing. In addition, a novel structural model with parameter learning projects the original texture distribution of the image block onto a low-dimensional distribution, making it simpler to identify faults.

2) Machine learning-based detection methods. The essence of machine learning is the analysis and learning of data (features) followed by making accurate decisions or predictions. In actual industrial settings, machine vision Technology is frequently used to detect flaws on metal assembly line surfaces. Typically, these techniques involve conventional image processing and deep learning designed to identify and analyze manufacturer-collected issues. The primary goal of conventional image processing techniques is to manually create features to describe surface flaws. Common manual features include LBP (Local Binary Mode) [10,11], HOG (Histogram of Directional Gradient) [12,13], GLCM (Grayscale Co-Occurrence Matrix) [14,15], and other statistical features. The goal is to identify

the proper features to represent the defect information in the input image of the metal surface. Since standard image processing algorithms frequently require sophisticated threshold settings for fault identification, they cannot be immediately implemented in the real world. Moreover, classic detection approaches are less effective and precise in complicated real-world contexts. They are sensitive to factors such as lighting conditions and the background. if these factors change, the threshold settings must be adjusted. Otherwise, the algorithm will not function in new contexts due to its lack of adaptability and robustness.

3) Deep learning-based hybrid detection method. The detection algorithm based on deep learning is more effective than manual detection in product surface quality control, and it can also reduce the workload of inspectors, lower the production costs of businesses, improve the precision and positioning of defect detection, enhance production efficiency, and promote manufacturing automation and intelligent development. Zhang et al. [16] developed a deep learning method combined with scratch tests to investigate plastic material properties and presented an effective strategy for determining the surface properties of the materials. Combining deep learning with related image inspection techniques is more likely to improve the detection accuracy of deep learning in various application contexts [17–19]. The conventional edge detection method uses the local differential method of the image, such as the Sobel operator, the Canny operator, the Pretalk operator, and the Laplace operator, to get information about the image's edges. These classical techniques are simple and straightforward to construct, but edge continuity is low, noise sensitivity is high, and positioning accuracy is generic. In recent years, there have been numerous techniques based on deep learning. Although these edge detection methods have a high detection accuracy and excellent search capability, they require many training samples for model training, and their efficiency is frequently poor [20–23]. The classic LoG (Gaussian-Laplace operator) [24,25] has the advantages of strong anti-interference ability, high boundary positioning, and good continuity, and it may also contribute to noise reduction. Nevertheless, the typical LoG operator is not precisely positioned during edge detection. double edges will arise during detection. and the detection edges under different scale factors are not distinct and discontinuous. Therefore, this paper proposes an improved LoG operator optimized by weight parameters, which can synthesize multi-angle edge details according to the characteristics and detection requirements of the detection target, and make the edge of the detection target clearer and more continuous by increasing the weight in a certain direction in a targeted manner and using deep learning techniques to extract the color, shape, texture, and other related deep features based on the local text. Using the difference between the gray value of the recovered target's local texture and its local background, this method further evaluates and identifies the edge information found by edge detection. It then carefully removes the image's scratches.

The rest of this paper is structured as follows: Section 2 describes the proposed detection strategy for metal surface scratches, including the improved LoG for detection analysis, LoG operator improvement and optimization strategy, and deep learning model structure design strategy. In Section 3, two experiments are done based on the difference in gray value between the target of detection and the local background, and this paper is concluded in Section 4.

## 2 The Proposed Detection Method for Metal Surface Scratch

This section introduces the proposed detection for metal surface scratches, including the improved LoG for detection analysis, LoG operator improvement and optimization strategy, and deep learning model structure design strategy, and Fig. 1 depicts the processing flowchart for this paper.

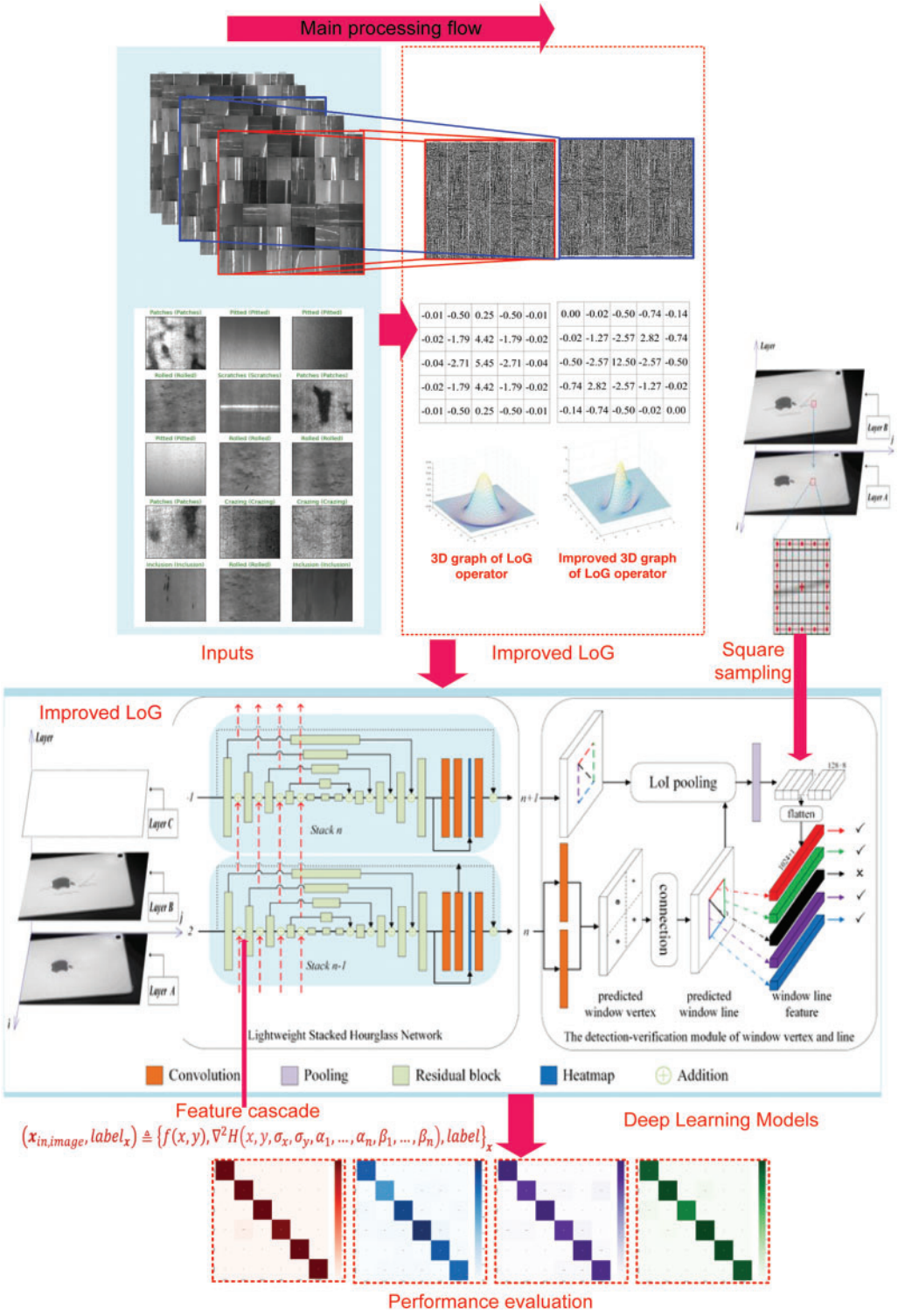



**Figure 1:** Processing diagram of this paper

Based on the difference between the gray value of the detection target and the gray value of the local backdrop, the processing approach is divided into two parts in this study. 1) The Gaussian smoothing filter and the Laplace sharpening filter are used in conjunction with the improved LoG operator to conduct Gaussian smoothing on the image and to recover edge detail information from the image. This method is effective for edge classification detection when the gray value of the detection target and the surrounding background are significantly different. 2) Despite the fact that the LoG operator can extract stronger semantic information and richer detailed information because of not clear and continuous when dealing with weak scratches on the metal surface, the resolution is very low and the perception of details is poor. This study combines the benefits of ResNet with the upgraded LoG operator to cascade features into a new enhanced image with comprehensive information, location information, and a higher resolution as input. The ResNet network is used to connect the shallow and high-level feature maps and to convey the shallow information to the deep Layer in order to alleviate the problem of easily overlooked detail scratches in the deep feature map and improve the accuracy of classification detection. In addition, the confusion matrix on the testing (validation) dataset for the four benchmark methods and the suggested method (improved LoG + ResNet) is presented to assess the final performance.

### *2.1 The Improved LoG for Detection Analysis*

The LoG operator is derived by locating the second-order derivative of the Gaussian kernel function, which is represented in the image as first Gaussian smoothing of the image and then Laplace second-order derivation. Set $f(x, y)$ as the original image, use the two-dimensional Gaussian function $G(x, y)$ to convolution, the original image, and the filtering result is expressed as by Eq. (1),

$$h(x, y) = G(x, y) * f(x, y) \tag{1}$$

where $*$ is the symbol of convolutional operation, $G(x, y) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$, $h(x, y)$ is the image obtained after smoothing filtering, $\sigma$ is the spatial distribution coefficient of the Gaussian function. Then Eq. (2) is obtained by applying the Laplace operator to the image,

$$\nabla^2 h(x, y) = \nabla^2 G(x, y) * f(x, y) \tag{2}$$

where $\nabla^2$ represents the Laplace operator, i.e., $\nabla^2 G(x, y)$ is the LoG operator. Note that the convolution principle of LoG is equivalent to differentiating the Gaussian function and then convolving with the original image, that is, using a Gaussian low-pass filter on the image before using the Laplace operator.

$$\nabla^2 G(x, y) = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} \tag{3}$$

$$\nabla^2 G(x, y) = \frac{1}{2\pi\sigma^4}\left[\frac{x^2+y^2}{2\sigma^2} - 2\right] exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4}$$

Eq. (4) is then derived by substituting Eq. (1) into Eq. (3). The LoG operator combines the Gaussian smoothing filter with the Laplace sharpening filter, which smoothes the fine edges while acting as a noise suppressor. As a result, the edge features are lost to a certain amount, and when dealing with weak scratches, the edges are not distinct and continuous. The LoG operator has rotational symmetry because a change in the value of the two-dimensional Gaussian function of the classical LoG operator $\sigma$ has no effect on the edge detection scale of other angles. Sometimes, only the edge information of a particular angle is required to detect the target's properties, or the edge of a

particular angle must be emphasized. This study improves the standard LoG operator so that it affects edge recognition in different ways depending on the direction of the image. This makes edge details stand out at certain angles.

### *2.2 LoG Operator Improvement and Optimization Strategy*

On the basis of the traditional LoG operator, the scale parameter is introduced, and the standard deviation in the X direction is denoted as $\sigma_x$, and the standard deviation in the Y direction is denoted as $\sigma_y$, then the Gaussian function with scale parameters $G\left(x, y, \sigma_x, \sigma_y\right)$ can be represented as Eq. (5),

$$G\left(x, y, \sigma_x, \sigma_y\right) = exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{x^2}{2\sigma_y^2}\right) \times \left(2\pi\left(\frac{\sigma_x}{2} + \frac{\sigma_y}{2}\right)^2\right)^{-1} \tag{5}$$

$$\nabla^2 G\left(x, y, \sigma_x, \sigma_y\right) = \left(\left(\frac{x^2}{\sigma_x^2} + \frac{x^2}{\sigma_y^2}\right) - 2\right) \times exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{x^2}{2\sigma_y^2}\right) \times \left(2\pi\left(\frac{\sigma_x}{2} + \frac{\sigma_y}{2}\right)^4\right)^{-1} \tag{6}$$

LoG edge detection operator $\nabla^2 G\left(x, y, \sigma_x, \sigma_y\right)$ with scale parameters is thus obtained based on Eq. (6). The projection of the traditional LoG operator three-dimensional function in the XOY plane is a circle, after the introduction of scale parameters, the projection of the LoG operator three-dimensional function diagram in the XOY plane is an ellipse. The two-dimensional Gaussian function $h\left(x, y, \sigma_x, \sigma_y, \alpha\right)$ with scale and angle parameters is denoted as Eq. (7),

$$h\left(x, y, \sigma_x, \sigma_y, \theta\right) = \aleph\left(\sigma_x, \sigma_y\right) exp\left(-\frac{1}{2}\left(\frac{xcos\theta + ysin\theta}{\sigma_x}\right)^2 - \frac{1}{2}\left(\frac{ycos\theta - xsin\theta}{\sigma_y}\right)^2\right) \tag{7}$$

where $\aleph\left(\sigma_x, \sigma_y\right) = \left(2\pi\left(\frac{\sigma_x}{2} + \frac{\sigma_y}{2}\right)^2\right)^{-1}$, then perform the Laplace transform $\nabla^2 G\left(x, y, \sigma_x, \sigma_y\right)$ on the Eq. (7) to obtain a two-dimensional Gaussian function with scale and angle parameters. The result is an anisotropic LoG operator $\nabla^2 G\left(x, y, \sigma_x, \sigma_y\right)$ that detects edges in different directions when values are different. The three-dimensional image no longer has rotational symmetry when $\left(\sigma_x, \sigma_y, \theta\right) = (1.0, 0.5, \pi/4)$, the X direction is different from the Y direction scale, and the direction of the long axis of the bottom projection is angled from the X axis. Edge detection is performed on the convolution template obtained by the Eq. (8), and when the angle parameter takes different values, the edge of different angles can be detected. In order to extract scratches at a specific angle according to the actual detection task, highlight the edge features of a certain angle, and synthesize the multi-directional edge details, this paper introduces the weight parameter [10] based on the anisotropic LoG operator.

$$\begin{aligned}\nabla^2 H\left(x, y, \sigma_x, \sigma_y, \alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n\right) = f\left(x, y\right) \times \left(\beta_1 \nabla^2 G\left(x, y, \sigma_{x1}, \sigma_{y1}, \alpha_1\right) + \ldots\right. \\ \left. + \beta_n \nabla^2 G\left(x, y, \sigma_{xn}, \sigma_{yn}, \alpha_n\right)\right)\end{aligned} \tag{8}$$

where $\beta_n, \sigma_{xn}, \sigma_{yn}$ represent the $n$th weight value of the nth template, the standard deviation in the X direction, and the standard deviation in the Y direction, respectively.

### *2.3 Deep Learning Model with Improved LoG Operator*

Convolutional neural networks (CNNs) are superior to traditional neural networks as a deep learning machine learning technique. Convolutional Layers make up a convolutional neural network. Convolutional neural networks consist of three neural Layers: convolutional neural Layer, pooling

neural Layer, and fully connected neural Layer. Under certain conditions, the output of the convolutional Layer is derived by filtering the input. The constituents of convolutional Layers are rectangular meshes or neural cubes. Thus, input and output Layers with filters can be rectangular meshes or cubic neuron modules. A new neuron is formed at each location in the filter by calculating the pixel's weighted volume. The volume of the output neuron in the convolutional Layer is regulated by three hyperparameters: depth, stride length, and zero fill. The pooling Layer does the subsequent action following every convolutional Layer. Pooling Layers are used to decrease the number of input neurons. This Layer generates a single value for each small rectangular block sampled from the convolutional Layer. Typically, the maximum pool method is employed. This is defined for a single type of pooling Layer by Eq. (9),

$$h_j^l(x,y)_{maxpooling} = max_{\overline{x}\in N,\overline{y}\in N(y)} h_j^{l-1}\left(\overline{x},\overline{y},\nabla^2 H\left(x,y,\sigma_x,\sigma_y,\alpha_1,\ldots,\alpha_n,\beta_1,\ldots,\beta_n\right)\right) \tag{9}$$

Using sampling filters that apply filters to each Layer, pooling Layers are generated. In most cases, the input Layer size is $4*4$, the filter size is $2*2$, and the stride size is 2. The fully linked Layer is the final Layer, which consists of all the neurons that came before it. From all input neurons to all output neurons, the fully linked Layer frequently reduces the quantity of spatial information. Thanks to the CNN network, all of the levels are ultimately interconnected. This makes it difficult for the network to go deep and expand, and gradient bursting or gradient disappearance may occur. ResNet uses a rapid connection by bypassing at least two levels. The residual building blocks use quick connections to bypass the convolutional Layers, which effectively alleviates the problem of vanishing gradients or gradient exploding caused by the increase in depth in the neural network, thereby constructing a CNN structure with greater flexibility to improve the defect detection rate. ResNet 50 is a widely used deep learning model with 48 convolutional Layers along with 1 MaxPool and 1 average pool Layer. Its residual building blocks consist of convolutional Layers (Conv), batch normalization (BN), rectified linear elements (ReLU) activation functions and shortcuts. The ResNet-50 architecture is comprised of the following components, one Layer consists of the convolution of kernel sizes $7*7$ and 64 distinct cores, and the highest pooling step is 2. In turn, there are three hidden Layers with $1*1*64$, $3*3*64$, and $1*1*256$ convolutional kernel sizes, respectively. Then there are 4 replicates, and the convolutional kernel sizes for the 12 hidden Layers are $1*1*128$, $3*3*128$, and $1*1*512$, respectively. This is followed by six replicates, including 18 hidden convolutional Layers with kernel sizes of $1*1*256$, $3*3*256$, and $1*1*1024$, respectively. The sizes of the convolutional kernels were $1*1*512$, $3*3*512$, and $1*1*2048$, in that order. Classify by developing a 1000-node average pooled, fully linked Layer-connected softmax function. Consequently, a 50-Layer deep convolutional ResNet network model has been developed for metal surface scratch identification. The network depth of the deep learning model can enable the model to extract conceptual information at a higher level and boost the model's capability. However, as the number of hidden Layers in neural networks increases, the correlation between backpropagation gradients deteriorates and approaches that of white noise, which can easily result in a degradation of performance in deep learning. Specifically, the original metal scratch image has a local correlation, and if the gradient of the hidden Layer is similar to white noise, then the gradient update at this moment is equivalent to random perturbation, which cannot substantially increase classification detection accuracy. Using gradient correlation measurements, ResNet is exceptional at maintaining gradient correlation. As a result, as the gradient is propagated backwards by the network, there will be no gradient exploding or disappearing, and the gradient of each Layer will be updated.

$$\left(\boldsymbol{x}_{in,image}, label_x\right) \triangleq \left\{f(x,y), \nabla^2 H\left(x,y,\sigma_x,\sigma_y,\alpha_1,\ldots,\alpha_n,\beta_1,\ldots,\beta_n\right), label\right\}_x \tag{10}$$

The original metal scratch image's low-level features have a better resolution and include more positional and detailed information, but because they undergo less convolution, their semantics are diminished and they are more noisy. The LoG operator extracts features with better semantic information and more detailed information, but at a very low resolution and with poor detail perception. Especially if a bright spot appears in a darker area of the image, the Laplace operator will produce a "brighter, brighter" effect, which might serve to sharpen the image. Therefore, this paper combines the advantages of ResNet with the improved LoG operator, as described in Eq. (10), and cascades the features to form a new enhanced image containing detailed information, position information, and higher resolution as input. It can fuse feature maps with strong low-resolution semantic information and feature maps with weak high-resolution semantic information but rich spatial information under the premise of requiring less computation. Then, the ResNet network is used to connect the shallow and high-level feature maps and transmit the shallow information to the deep Layer to solve the problem of detail scratches that are easily overlooked in the deep feature map.

## 3 Experiments

In this section, in order to visualize the processing effect of the method and evaluate the performance of the model, we test two different types of data sets.

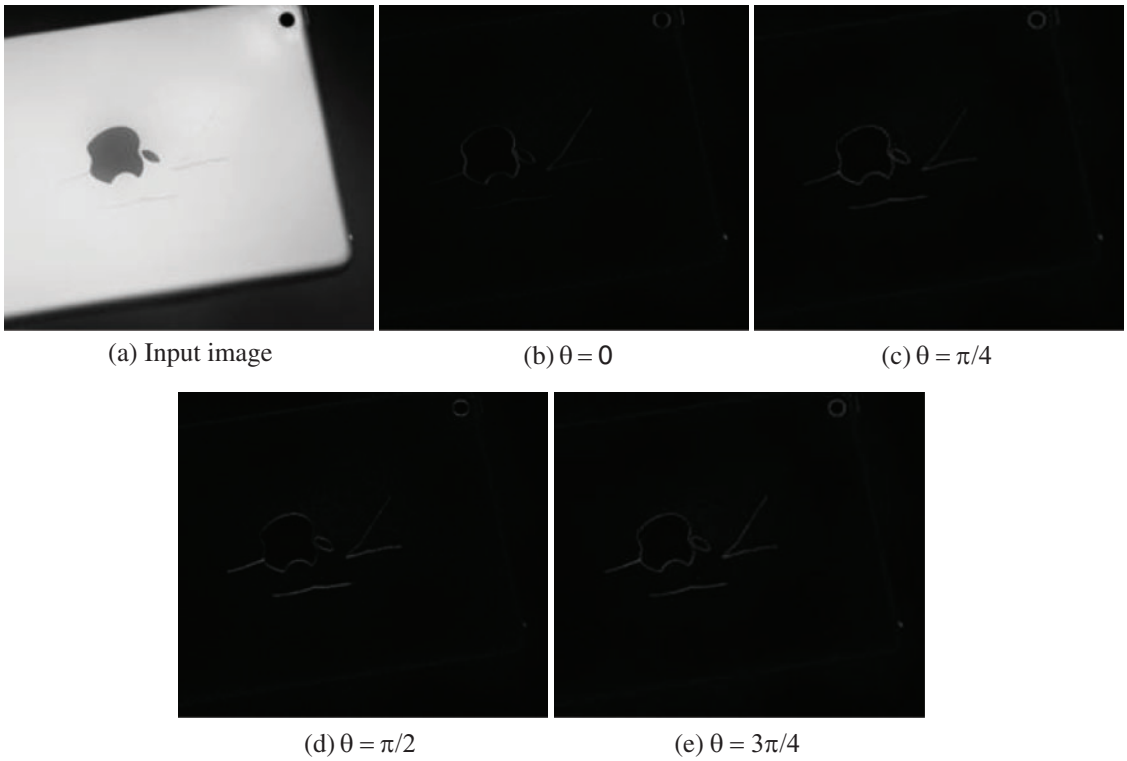
### *3.1 Performance Analysis of the Improved LoG Operator Edge Feature Extraction*

In order to visually demonstrate the advantages of improving the LoG operator, this paper selects four sets of parameters $(\sigma_x, \sigma_y, \theta)$, and for generating convolutional templates, respectively, the input image is edge-detected, and then the detection results of each angle are weighted to obtain the final image. The parameter combinations selected in this paper are, $(\sigma_x = 1, \sigma_y = 0.5, \theta = 0, \pi/4, \pi/2, 3\pi/4)$. These combinations of parameters are substituted into the discretization formula of Eq. (11), and the resulting initial template is normalized and the coefficient sum is 0 steps to obtain the final convolution template. The corresponding templates for the combination of parameters are provided in Table 1.

**Table 1:** Improve the template of the LoG operator

| $\partial x = 1.0, \partial y = 0.5, \theta = 0$ | | | | | $\partial x = 1.0, \partial y = 0.5, \theta = 3\pi/4$ | | | | |
|---|---|---|---|---|---|---|---|---|---|
| −0.01 | −0.50 | 0.25 | −0.50 | −0.01 | 0.00 | −0.02 | −0.50 | −0.74 | −0.14 |
| −0.02 | −1.79 | 4.42 | −1.79 | −0.02 | −0.02 | −1.27 | −2.57 | 2.82 | −0.74 |
| −0.04 | −2.71 | 5.45 | −2.71 | −0.04 | −0.50 | −2.57 | 12.50 | −.57 | −0.50 |
| −0.02 | −1.79 | 4.42 | −1.79 | −0.02 | −0.74 | 2.82 | −2.57 | −1.27 | −0.02 |
| −0.01 | −0.50 | 0.25 | −0.50 | −0.01 | −0.14 | −0.74 | −0.50 | −0.02 | 0.00 |

In order to verify experimentally the used template shown in Figs. 3, 2a is the input image (single-channel grayscale map) used in this section. The input images are convolved separately from the obtained templates, and the results are shown in Figs. 2b–2e, respectively. From the experimental results, it can be seen that the edge along the angle $\theta$ is clearer, while the edge information in the direction perpendicular to the angle $\pi/2$ and $3\pi/4$ are blurred, and when the angle parameters are selected, the detected scratch information is clearer. Combined with the observation of Fig. 2, when the long axis of the ellipse is perpendicular to the tangent direction of the edge profile, the edge detection works best.



(a) Input image (b) $\theta = 0$ (c) $\theta = \pi/4$

(d) $\theta = \pi/2$ (e) $\theta = 3\pi/4$

**Figure 2:** The experimental results when $\theta$ under different values

Through the above analysis, in order to obtain a better detection effect, when selecting the weight factor, the weight in the direction $\pi/2$ and $3\pi/4$ can be appropriately increased, the edge of the four angles $\theta = 0, \pi/4, \pi/2, 3\pi/4$ of the detection result image, respectively, take the weight parameters $(\beta_1, \beta_2, \beta_3, \beta_4) = (0, 0.2, 0.4, 0.8)$, substitute parameters to weight the image fusion, the final image obtained is shown in Figs. 3a, 3b is the detection result of the traditional LoG operator. Fig. 3c is the threshold estimation and morphological analysis results, and Figs. 3d, 3e are the identification results of the input image and magnified edge detection in detail, respectively.

(a) Improved LoG operator (b) Traditional LoG operator (c) Threshold estimation & morphological analysis

**Figure 3:** (Continued)

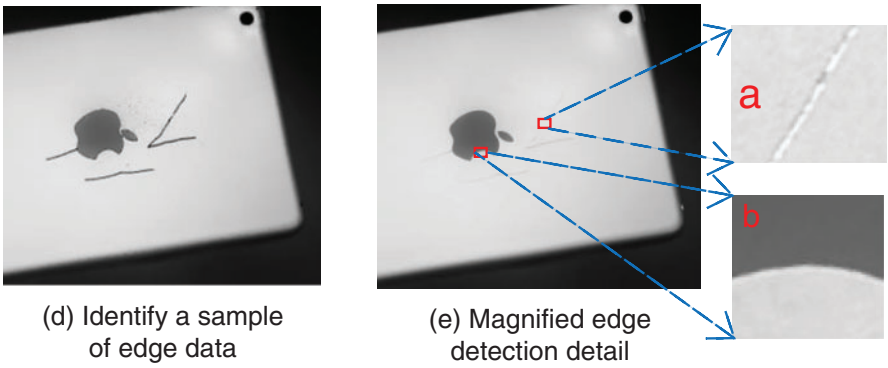



**Figure 3:** Experimental results comparison of traditional and improved LoG operator

As observed in the illustration, the weighted result properly combines the edge information from many angles, and the edge at the scratch is crisper and more continuous than the edge result obtained by the standard LoG operator, which is consistent with the experimental predictions. The test results show that the improved LoG can find edges from different angles and extract edge features that match the requirements for finding edges.

### *3.2 The Performance Evaluation of the Proposed Approach*

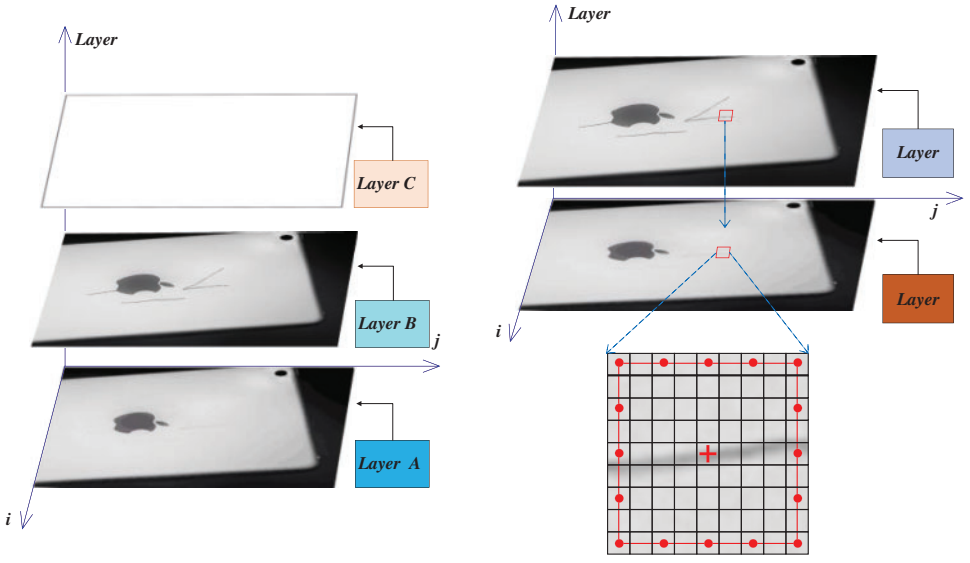
When the degree of difference between the gray value of the detection target and its local background is different, the difficulty of metal scratch detection is also different, and this section is divided into two parts to describe separately.

#### *3.2.1 Texture Feature Extraction and Analysis*

The larger the difference between a scratch and a certain feature parameter of other interfering information, the more precisely this parameter can be used to extract targets. Observing the results of edge detection reveals, however, that the edge of the scratch is comparable to the area size of the interference edge, and that there is a connection between the scratch edge and the interference edge. these characteristics will diminish the detection effect of the method. Using the form and color characteristics of a detection target to filter and extract features is the most prevalent technique in feature extraction. As a result of the brighter color on both sides of the scratch in a typical metal scratch, the grayscale value is greater. the grayscale values are notably different on both sides of the non-scratch edge, with higher values on one side and lower values on the other. Based on this property and the improvement of LoG edge recognition, this work gives a local texture screening method that filters out scratch information in an effective way.

*i*) Mark edge information. The edge detection result image of the improved LoG operator is processed by threshold processing and morphological operation to prevent noise and many interfering edges in the processed image. The grayscale of this figure is then reversed and overlaid into the original grayscale image, and the edge of the image that is marked with the edge is marked with black and its grayscale value is 0.

*ii*) Create Layer space. As shown in Fig. 4a, a Layer space is established with the input image (denoted as Layer *A*), the tag edge image (denoted as Layer *B*) and the blank output image (denoted as Layer *C*), and the subsequent steps will be performed in this Layer space, image *A* will be used as a sampled image, image *B* will be used as a lookup image for suspicious pixels, and image *C* will be used as the output image.




(a) Layer space (b) Square sampling area

**Figure 4:** Schematic diagram of Layer space and sampling

*iii*) Find suspicious pixels. Set a suspicious threshold of U that is used to find suspicious target pixels in Layer $B$. Since the edge pixel grayscale value of Layer $B$ in this case is 0, the threshold $U$ can be taken from 1. Each pixel is traversed sequentially on Layer $B$, and if the grayscale value of the point is greater than $U$, the point is considered a non-suspicious pixel, and the grayscale value of the pixel is assigned a value of 0 on Layer $C$, and the resulting grayscale image appears black, indicating that the point is not an edge area and is not scratch information. If the grayscale value of a pixel is less than $U$, the point is considered a suspicious pixel and detection continues. The $I_C(i,j)$ is denoted by Eq. (11),

$$I_C(i,j) = \begin{cases} 0, if I_B(i,j) > U \\ Suspicious\ pixels, else \end{cases} \tag{11}$$

where $I_B(i,j)$ and $I_C(i,j)$ represent the Grayscale values corresponding to the pixels of the $i$-row, $j$-column on Layer $B$ and Layer $C$, respectively, signal that the pixel is suspicious and that processing should proceed.

*iv*) Local sampling. On Layer $A$, a square sample region with a side length of $(2K+1)$ pixels is built, centered on the corresponding pixel coordinates of the suspicious pixels, where $K$ can be chosen based on the image size and pixel width of the detection target. After constructing the square sampling area, the pixels on the square's perimeter are uniformly sampled on Layer $A$, and the number of sample points is set to $S$. Fig. 5b depicts a diagrammatic representation of the created square sample area and sampling point. Select an appropriate demarcation threshold $V$, which is the demarcation between high and low gray values at the sample point on Layer $A$, and vote on the number of occurrences of high and low gray values at the sample point. Vote results for light grayscale values are recorded as $L_1$, and voting results for dark grayscale values are recorded as $L_2$.

*v*) Suspicious point discrimination. Introduce a voting threshold of $W$, the value of $W$ is between $O$ and $S$, generally setting is $W = 2S/3$. When the value of the voting result $L_1$ is greater than $L_2$, and $L_1$ is greater than the voting threshold $W$, it means that in the field of the point, the pixel of the

high gray value is larger than the pixel of the low gray value, and reaches a certain proportion, which is in line with the texture characteristics at the scratch, then this pixel is determined to be the edge of the scratch, and the grayscale value of the point is assigned a value of 255 on Layer $C$. Otherwise, if the point is not in line with the texture characteristics of the scratch, the pixel is determined to be a non-scratch edge, and the grayscale value of the pixel is assigned to 0 on Layer $C$. The resulting Layer $C$ is the output image. The above procedure can be expressed as follows by Eq. (12),

$$I_C(i,j) = \begin{cases} 255, if\ (L_1 > L_2, L_1 > W) \\ 0, else \end{cases} \tag{12}$$

where $I_C(i,j)$ is the grayscale value of the pixels of the $i$-row and $j$-column on Layer $C$.

vi) Noise removal. The above process has removed most of the interference edges, but still leaves some smaller noise areas. In order to remove these noises, select the appropriate $S$ and communication domain filter values $Q$ according to the image size, treat the communication domain area less than $Q$ as noise, and assign a value of 0, to remove the noise in the foreground pixels of the scratch image and obtain the final scratch detection results.

The local texture screening method primarily uses the local texture characteristics of the scratch to judge and identify the edge information, and the core basis for its judgment is that the texture and grayscale value of the local background of the extracted target are not significantly different, whereas the grayscale value of the local background of the interference information has obvious differences, and the target and interference information can be distinguished. To ease the local texture screening method for locating suspicious target pixels, you can first use an image segmentation algorithm, such as threshold segmentation and edge detection, to enlarge and detect the grayscale difference between the target and the local background information. In the preceding example, the edges of the input image are detected with the improved LoG operator due to the texture characteristics of the image and the detection task. After processing, the difference between edge information and its local background is emphasized, and the local texture screening method is used to find targets.

In order to verify whether the above hypothesis can be true, this paper selects two cases of target detection for experiments, as shown in Figs. 5a, 5b are the input images of "hair detection" and "weld inspection" (single-channel grayscale image), and the images obtained after processing by local texture screening method are shown in Figs. 5c, 5d, and the selected parameter values are shown in Table 2.

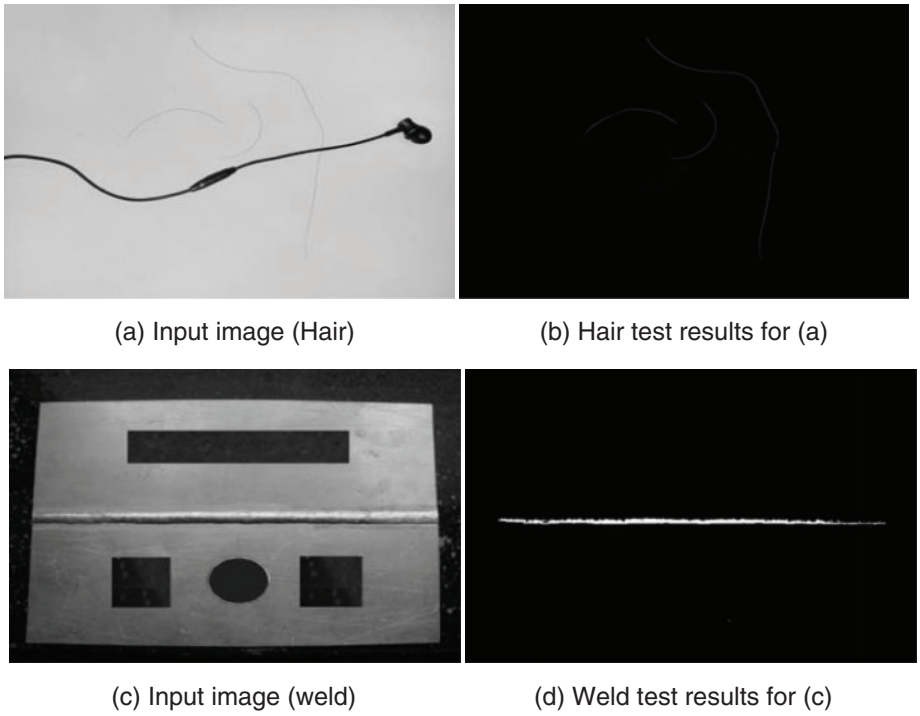
As can be seen from Fig. 5, in other object detection tasks, this method can also effectively eliminate interference information, successfully detect the target, and obtain very desirable results. $S$, $L$ and $W$, respectively represent the area size of the available area corresponding to the sampling length $u$ and width $v$, the length and width of the smallest circumscribed rectangle as tabulated in Table 2, and the circularity of metal scratch detection (indicating the degree to which the shape of the target area is close to a circle), could be obtained by using the area $S$ and the perimeter $C$ of the area. The circularity $R$ and rectangularity $J$ are estimated by Eqs. (13) and (14), respectively.

$$R = 4\pi S \cdot C^{-2} \tag{13}$$

$$J = S \cdot (L \cdot W)^{-1} \tag{14}$$

Especially when the difference between the gray value of the detection target and its local background is more obvious, the use of LoG for edge detection can better complete the detection of metal surface scratches. The great benefit of this research is that even very thin hair strands and metal scratches can still be recognized with high precision, especially when texture information is

analyzed, which is helpful in developing high-precision metal scratch detection models. The approach's major limitations are that the metal scratch data evaluated is essentially a 2D image, and the detecting accuracy range of scratch size cannot be judged only on the image's grayscale value. Because images acquired under different light sources, the angle of acquisition, and the image processing method will all cause the size of metal scratches to vary under different conditions.



(a) Input image (Hair) (b) Hair test results for (a)

(c) Input image (weld) (d) Weld test results for (c)

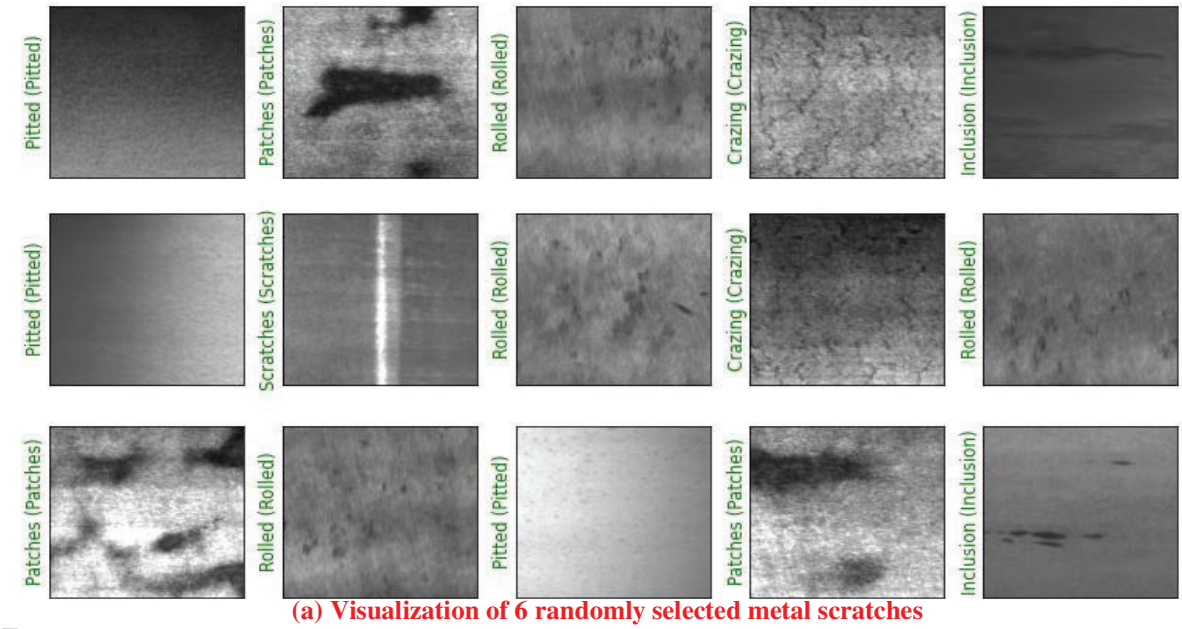
**Figure 5:** Object detection experiments and results

**Table 2:** Parameters for the local texture screening method

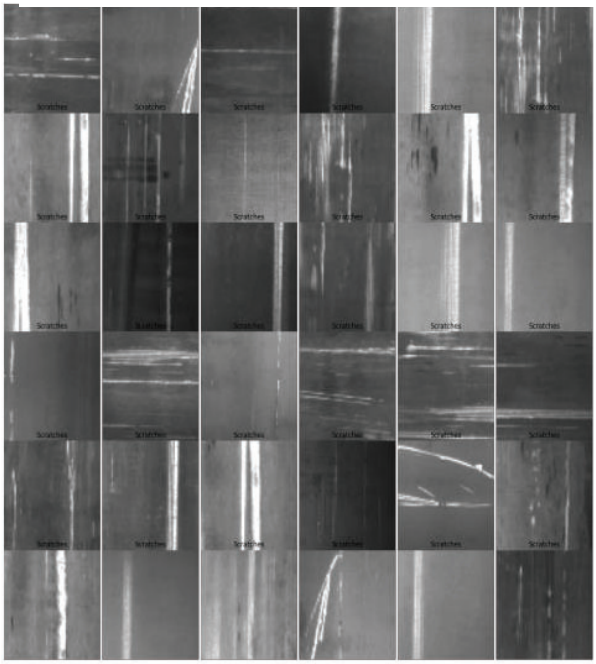
| Image | $U$ | $V$ | $S$ | $W$ | $K$ | $Q$ |
|---|---|---|---|---|---|---|
| Hairline detection | 115 | 130 | 16 | 11 | 2 | 0 |
| Weld detection | 75 | 100 | 16 | 11 | 7 | 0 |

### *3.2.2 Performance Comparison of the Deep Learning*

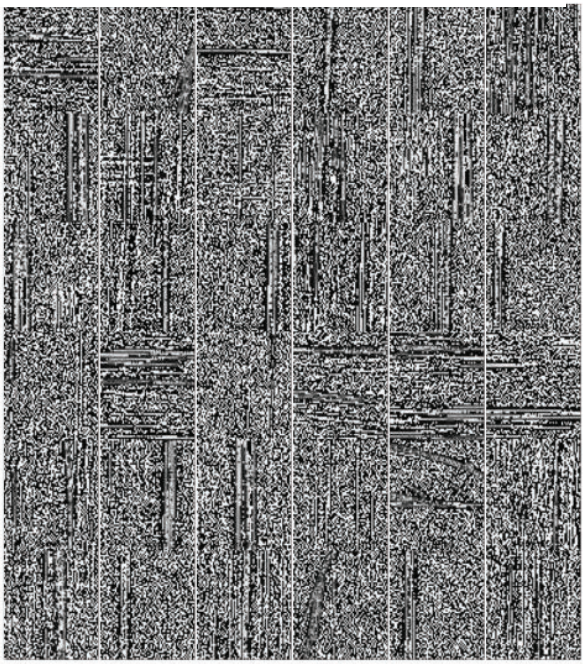
The purpose of this paper is to detect accurate scratch information. Although the improved LoG operator successfully detected the image's edges and to some extent suppressed the noise, there is still some noise that is difficult to remove, and there is a great deal of non-scratch interference information in the image, such as product marks, metal shell edges, etc. The visualization of 6 randomly selected metal scratches, metal scratches with low contrast and ones processed by LoG operator are shown in Fig. 6. The confusion matrix on testing (validation) dataset by Googlenet [26], Xception [27], Inception [28], ResNet [29] and the Proposed approach (improved LoG + ResNet) are shown in Tables 3–7, respectively.




(a) Visualization of 6 randomly selected metal scratches



(b) Visualization of metal scratches with low contrast



(c) Visualization of metal scratches processed by LoG operators

**Figure 6:** Visualization of the NEU metal surface defects dataset

**Table 3:** The confusion matrix on testing (validation) dataset by Googlenet

| | Crazing | Inclusion | Patches | Pitted | Rolled | Scratches |
|---|---|---|---|---|---|---|
| Crazing | 19(17) | 0(0) | 0(0) | 0(0) | 0(0) | 0(0) |
| Inclusion | 0(0) | 15(17) | 0(0) | 0(0) | 0(0) | 0(0) |
| Patches | 0(0) | 0(1) | 15(16) | 0(0) | 0(0) | 0(0) |
| Pitted | 1(1) | 1(0) | 1(0) | 17(16) | 1(0) | 0(0) |
| Rolled | 1(0) | 0(0) | 1(0) | 0(0) | 18(17) | 0(0) |
| Scratches | 0(0) | 1(0) | 0(0) | 1(0) | 0(1) | 14(16) |

**Table 4:** The confusion matrix on testing (validation) dataset by Xception

| | Crazing | Inclusion | Patches | Pitted | Rolled | Scratches |
|---|---|---|---|---|---|---|
| Crazing | 17(17) | 0(0) | 0(0) | 0(0) | 0(0) | 0(0) |
| Inclusion | 0(0) | 15(17) | 0(0) | 1(0) | 1(0) | 0(0) |
| Patches | 0(0) | 0(0) | 17(17) | 0(0) | 0(0) | 0(0) |
| Pitted | 0(1) | 1(2) | 1(0) | 14(14) | 0(1) | 0(0) |
| Rolled | 0(0) | 0(0) | 0(0) | 0(0) | 20(0) | 0(0) |
| Scratches | 0(0) | 1(0) | 1(1) | 0(0) | 0(0) | 17(16) |

**Table 5:** The confusion matrix on testing (validation) dataset by Inception

| | Crazing | Inclusion | Patches | Pitted | Rolled | Scratches |
|---|---|---|---|---|---|---|
| Crazing | 16(17) | 0(0) | 0(0) | 0(0) | 0(0) | 0(0) |
| Inclusion | 0(0) | 12(14) | 0(1) | 1(1) | 1(0) | 0(2) |
| Patches | 1(0) | 0(0) | 18(17) | 1(0) | 0(0) | 1(0) |
| Pitted | 1(1) | 1(1) | 1(0) | 19(14) | 1(2) | 0(0) |
| Rolled | 0(0) | 0(0) | 1(0) | 0(0) | 16(16) | 1(1) |
| Scratches | 0(0) | 1(0) | 1(0) | 1(1) | 0(1) | 15(16) |

**Table 6:** The confusion matrix on testing (validation) dataset by ResNet

| | Crazing | Inclusion | Patches | Pitted | Rolled | Scratches |
|---|---|---|---|---|---|---|
| Crazing | 17(17) | 0(0) | 0(0) | 0(0) | 0(0) | 0(0) |
| Inclusion | 0(0) | 18(17) | 0(0) | 1(0) | 0(0) | 0(0) |
| Patches | 1(1) | 0(0) | 14(16) | 0(0) | 0(0) | 0(0) |
| Pitted | 0(0) | 1(0) | 0(0) | 18(17) | 0(0) | 0(0) |
| Rolled | 0(0) | 0(0) | 0(0) | 0(0) | 18(17) | 0(0) |
| Scratches | 0(0) | 0(0) | 0(0) | 0(1) | 0(0) | 17(16) |

**Table 7:** The confusion matrix on testing (validation) dataset by proposed approach

| | Crazing | Inclusion | Patches | Pitted | Rolled | Scratches |
|---|---|---|---|---|---|---|
| Crazing | 17(17) | 1(0) | 0(0) | 1(0) | 0(0) | 0(0) |
| Inclusion | 0(0) | 18(17) | 0(0) | 0(0) | 0(0) | 0(0) |
| Patches | 0(0) | 0(0) | 13(0) | 0(0) | 0(0) | 0(0) |
| Pitted | 0(0) | 2(1) | 0(0) | 14(16) | 0(0) | 0(0) |
| Rolled | 0(0) | 0(0) | 0(0) | 0(0) | 17(17) | 0(0) |
| Scratches | 0(0) | 1(0) | 0(0) | 0(0) | 0(0) | 20(17) |

Feature extraction is necessary for removing interference information and properly extracting the detection target. Especially when the difference between the gray value of the detection target and its local background is not obvious, as depicted in Fig. 6b, the use of LoG for edge detection, or even when combined with texture analysis, cannot effectively improve the detection accuracy, the used data set NEU Metal Surface Defects Database (www.kaggle.com/datasets/fantacher/neu-metal-surface-defects-data) which contains six kinds of typical surface defects of the hot-rolled (Sc). The collection contains 1,800 grayscale photos, 300 examples for each of six types of common surface flaws. It is required to utilize the LoG technique in conjunction with deep learning to develop a high-precision metal surface scratch detection model since the detection target is close to the grayscale value of its local background.

The performance configuration is Ubuntu 18.04.6 (GPU 2080Ti 11G, RAM 64 G), Windows 10 (GPU 6 G, RAM 32 G), Visual Studio 2022 (C++), Python 3.6, Matlab 2019b. To evaluate the final six classification detections, the performance of Googlenet [26], Xception [27], Inception [28], and ResNet [29] is applied. Based on deep learning's superior classification performance, the Pytorch and Keras frameworks are used to develop a deep learning network classification model for the classified findings. The maximum epoch length is 10, and the maximum learning rate is 0.001. Crossectropy and the Adam method are utilized to quantify model loss and optimize model architecture, respectively. The confusion matrix of the Googlenet, Xception and Inception are shown in Fig. 7, and the confusion matrix of the ResNet and proposed approach (ResNet combined the improved LoG operator) are shown in Fig. 8.

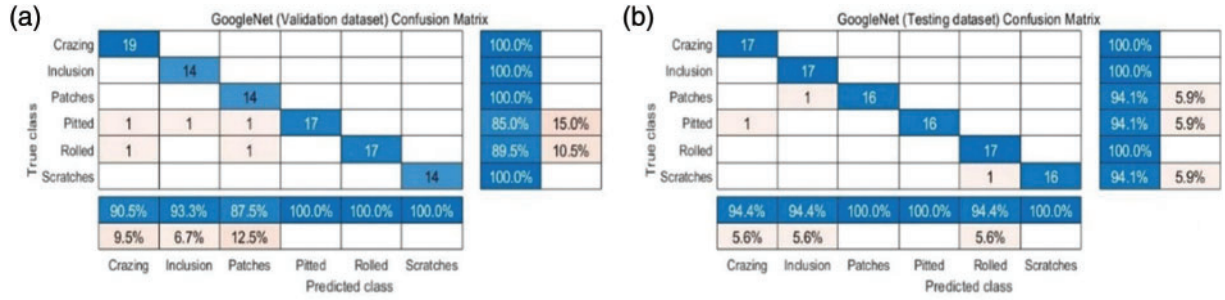



**Figure 7:** (Continued)

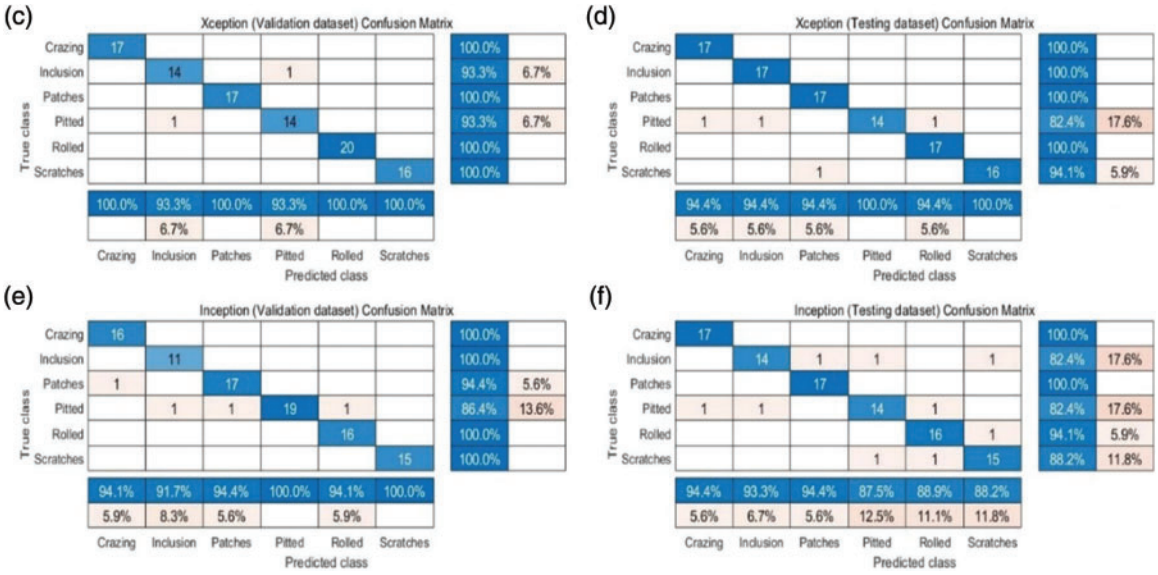



**Figure 7:** The confusion matrix of the Googlenet, Xception and Inception

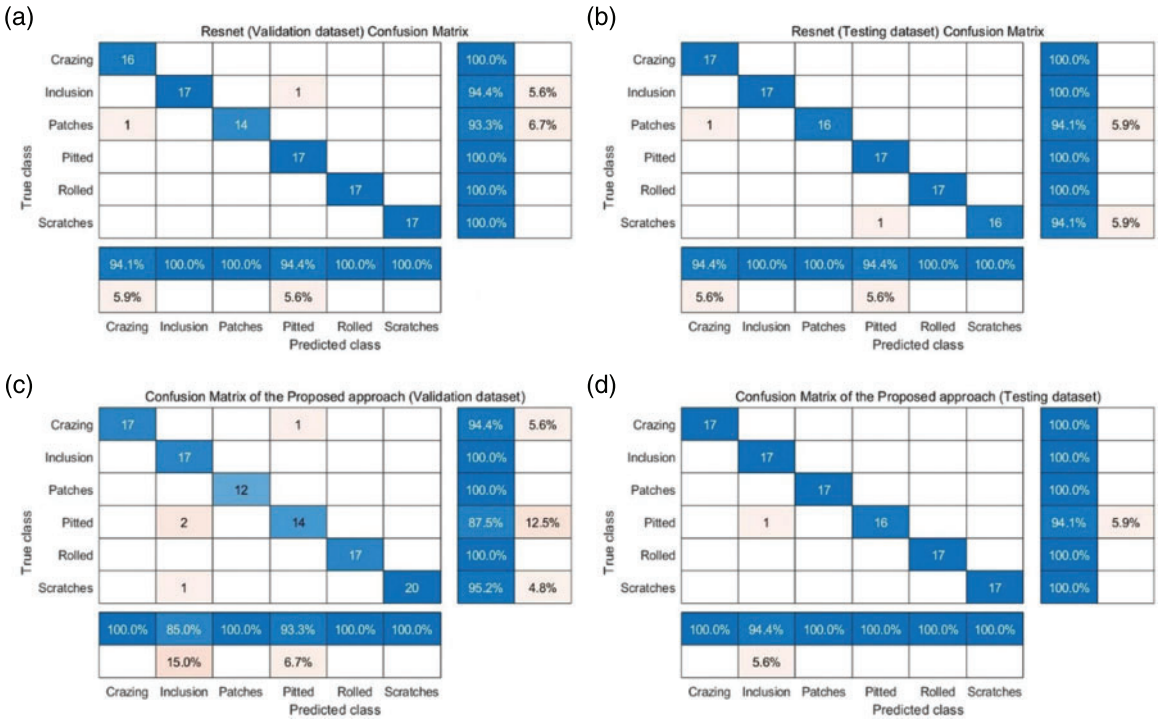



**Figure 8:** The confusion matrix of the ResNet and proposed approach

Tables 8, 9 show the classification accuracy of five deep learning models on the validation dataset and testing dataset, respectively. One can find that the detection accuracy of the metal surface scratch based on the proposed approach has been improved by 1.9%, 2.8%, 8.0% and 1.0% compared with the other four benchmark approaches. The comparison of the classification-detection accuracy on the

training, validation, and testing dataset, and the statistics are shown in Figs. 9, 10. The advantages of the proposed strategy and this paper in terms of model accuracy and model loss are readily apparent, as is the method proposed for detecting scratches on metal surfaces. It is unable to efficiently extract appropriate features from misclassified samples to compare with the characteristics of other scratches because the texture features of the two-dimensional image's scratches are so similar to one another. Based on the available dataset, analysis is performed using the higher dimension with a topology or a superior perspective.

**Table 8:** The classification-detection accuracy of the deep learning models on validation dataset

| Methods | Crazing | Inclusion | Patches | Pitted | Rolled | Scratches |
|---|---|---|---|---|---|---|
| Googlenet [26] | 90.5% | 93.3% | 87.5% | 100% | 100% | 100% |
| Xception [27] | 100% | 93.3% | 100% | 93.3% | 100% | 100% |
| Inception [28] | 94.1% | 91.7% | 94.4% | 100% | 94.1% | 100% |
| ResNet [29] | 94.1% | 100% | 100% | 94.4% | 100% | 100% |
| Proposed | 100% | 85% | 100% | 93.3% | 100% | 100% |

**Table 9:** The classification-detection accuracy of the deep learning models on testing dataset

| Methods | Crazing | Inclusion | Patches | Pitted | Rolled | Scratches | Ave. acc. |
|---|---|---|---|---|---|---|---|
| Googlenet [26] | 94.4% | 94.4% | 100% | 100% | 94.4% | 100% | 97.2% |
| Xception [27] | 94.4% | 94.4% | 94.4% | 100% | 94.4% | 100% | 96.3% |
| Inception [28] | 94.4% | 93.3% | 94.4% | 87.5% | 88.9% | 88.2% | 91.1% |
| ResNet [29] | 94.4% | 100% | 100% | 94.4% | 100% | 100% | 98.1% |
| Proposed | 100% | 94.4% | 100% | 100% | 100% | 100% | **99.1%** |

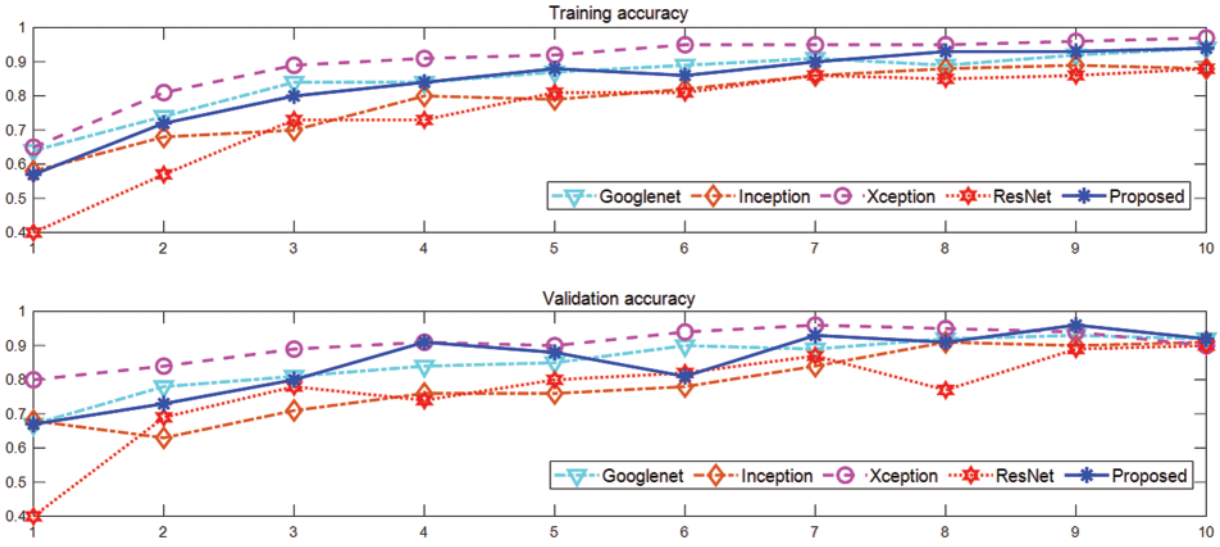



**Figure 9:** (Continued)

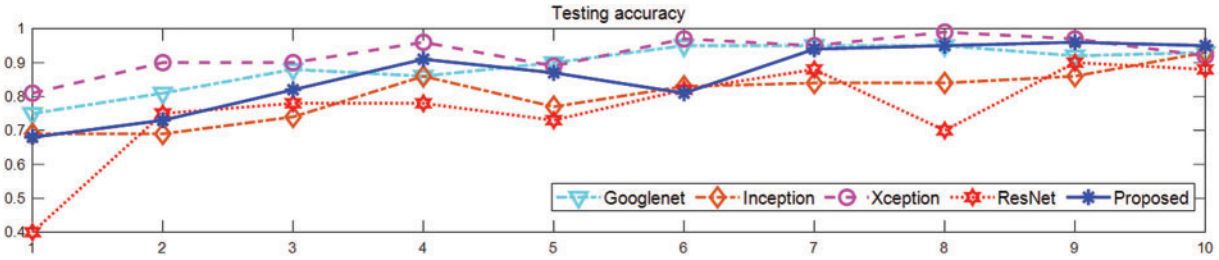



**Figure 9:** The classification-detection accuracy on the training, validation, and testing dataset

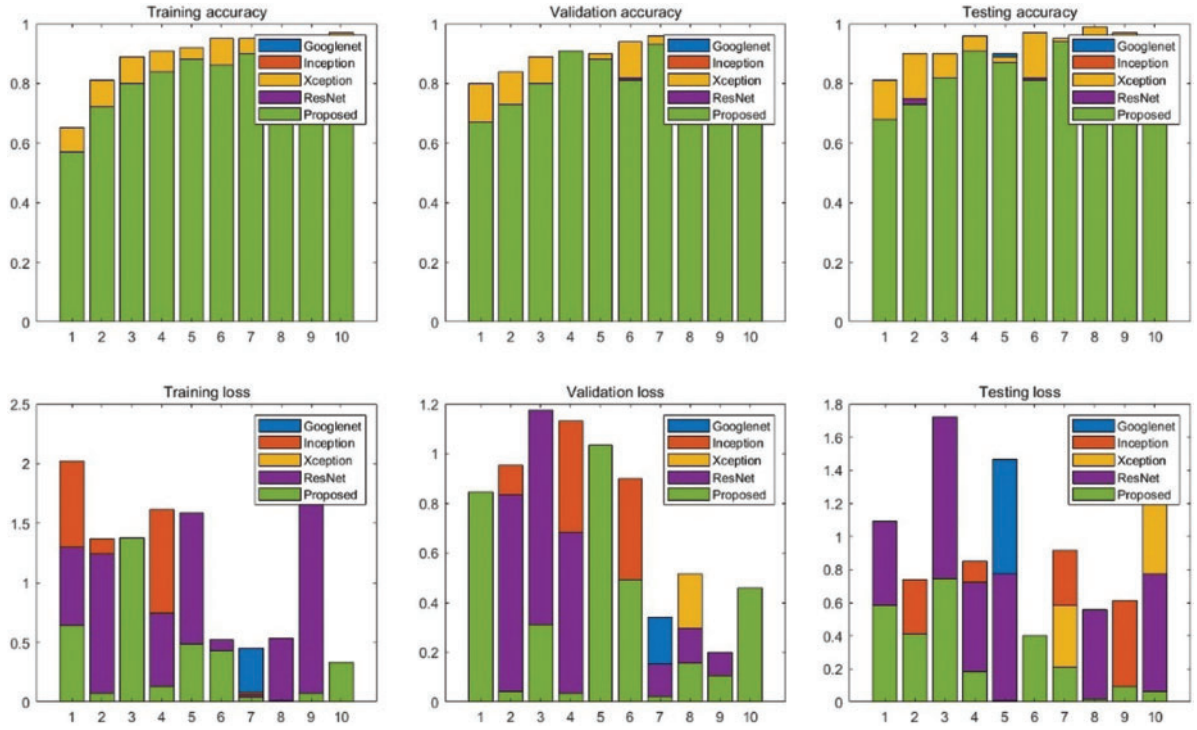



**Figure 10:** The classification-detection accuracy and loss on the training, validation, and testing dataset

## 4 Conclusions

This paper proposes a high-precision metal surface scratch classification-detection approach based on the enhanced LoG edge extraction operator and a deep learning technique. When the gray value of the detection target and its local background are significantly different, the enhanced LoG edge extraction operator is used to obtain the edge information of the image by selecting the edge based on the texture characteristics of the image and the target extraction task requirements. This research proposes a solution based on an improved LoG edge extraction operator and deep learning to overcome the challenge of removing interference edges during feature extraction. Experiments have shown that the improved LoG edge detection technique, in conjunction with local thresholds and deep learning, can efficiently remove interference information from the graph and reliably extract scratches. Experiments utilizing the proposed method based on the six types of metal scratches in the NEU Metal Surface Defects Database, namely rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS),

inclusion (In), and scratches (Sc), achieved a higher detection rate, i.e., an average detection accuracy of approximately 99.1%. Future research in this paper focuses on three-dimensional (3D) metal scratch identification of non-flat surfaces and employs a graph convolutional neural network (GCN) to assess scratch breadth and depth, which reflect the degree of deformation and damage, in order to better confront industrial applications.

**Acknowledgement:** Thanks to the help of four anonymous reviewers and journal editors, the logical organization and content quality of this paper have been improved.

**Funding Statement:** This project is supported by the National Natural Science Foundation of China (No. 62001197), Natural Sciences Research Grant for Colleges and Universities of Jiangsu Province (No. 22KJD470002) and Jiangsu Provincial Postgraduate Research and Practice Innovation Program (No. XSJCX21_58).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Li, H., Zhang, L., Zhang, Y., Fu, X., Shao, X. et al. (2023). A peridynamic approach for the evaluation of metal ablation under high temperature. *Computer Modeling in Engineering & Sciences, 134(3),* 1997–2019. https://doi.org/10.32604/cmes.2022.020792
2. Yan, X., She, D., Xu, Y., Jia, M. (2021). Deep regularized variational autoencoder for intelligent fault diagnosis of rotor-bearing system within entire life-cycle process. *Knowledge-Based Systems, 226(6),* 107142. https://doi.org/10.1016/j.knosys.2021.107142
3. Liao, Z., la Monaca, A., Murray, J., Speidel, A., Ushmaev, D. et al. (2021). Surface integrity in metal machining–Part I: Fundamentals of surface characteristics and formation mechanisms. *International Journal of Machine Tools and Manufacture, 162(289),* 103687. https://doi.org/10.1016/j.ijmachtools.2020.103687
4. Barak, L., Bloch, I. M., Botti, A., Cababie, M., Cancelo, G. et al. (2022). SENSEI: Characterization of single-electron events using a skipper charge-coupled device. *Physical Review Applied, 17(1),* 014022. https://doi.org/10.1103/PhysRevApplied.17.014022
5. Yang, Y., Zheng, P., Zeng, F., Xin, P., He, G. et al. (2023). Metal corrosion rate prediction of small samples using an ensemble technique. *Computer Modeling in Engineering & Sciences, 134(1),* 267–291. https://doi.org/10.32604/cmes.2022.020220
6. Bao, Y., Song, K., Liu, J., Wang, Y., Yan, Y. et al. (2021). Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. *IEEE Transactions on Instrumentation and Measurement, 70,* 1–11. https://doi.org/10.1109/TIM.2021.3083561
7. Lu, Q., Lin, J., Luo, L., Zhang, Y., Zhu, W. (2022). A supervised approach for automated surface defect detection in ceramic tile quality control. *Advanced Engineering Informatics, 53(2),* 101692. https://doi.org/10.1016/j.aei.2022.101692
8. Wang, J., Noble, J. H., Dawant, B. M. (2019). Metal artifact reduction for the segmentation of the intra cochlear anatomy in CT images of the ear with 3D-conditional GANs. *Medical Image Analysis, 58(2),* 101553. https://doi.org/10.1016/j.media.2019.101553
9. Yang, L., Zhou, F., Wang, L. (2022). A scratch detection method based on deep learning and image segmentation. *IEEE Transactions on Instrumentation and Measurement, 71,* 1–12. https://doi.org/10.1109/TIM.2022.3220285

10. Su, B., Chen, H., Zhu, Y., Liu, W., Liu, K. (2019). Classification of manufacturing defects in multicrystalline solar cells with novel feature descriptor. *IEEE Transactions on Instrumentation and Measurement, 68(12),* 4675–4688. https://doi.org/10.1109/TIM.2019.2900961
11. Cao, B. F., Li, J. Q., Qiao, N. S. (2020). Nickel foam surface defect detection based on spatial-frequency multi-scale MB-LBP. *Soft Computing, 24(8),* 5949–5957. https://doi.org/10.1007/s00500-019-04513-2
12. Matern, F., Riess, C., Stamminger, M. (2019). Gradient-based illumination description for image forgery detection. *IEEE Transactions on Information Forensics and Security, 15,* 1303–1317. https://doi.org/10.1109/TIFS.2019.2935913
13. Wang, Y., Zhu, X., Wu, B. (2019). Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier. *International Journal of Remote Sensing, 40(19),* 7356–7370. https://doi.org/10.1080/01431161.2018.1513669
14. Chai, H. Y., Wee, L. K., Swee, T. T., Hussain, S. (2011). Gray-level co-occurrence matrix bone fracture detection. *WSEAS Transactions on Systems, 10(1),* 7–16.
15. Mishra, S., Majhi, B., Sa, P. K., Sharma, L. (2017). Gray level co-occurrence matrix and random forest based acute lymphoblastic leukemia detection. *Biomedical Signal Processing and Control, 33(1),* 272–280. https://doi.org/10.1016/j.bspc.2016.11.021
16. Zhang, J., Qin, J., Li, Y., Lu, C., Liu, H. et al. (2022). Extraction of the plastic properties of metallic materials from scratch tests using deep learning. *Mechanics of Materials, 175,* 104502. https://doi.org/10.1016/j.mechmat.2022.104502
17. Zhou, X., Sun, J., Tian, Y., Lu, B., Hang, Y. et al. (2020). Hyperspectral technique combined with deep learning algorithm for detection of compound heavy metals in lettuce. *Food Chemistry, 321(2),* 126503. https://doi.org/10.1016/j.foodchem.2020.126503
18. Block, S. B., da Silva, R. D., Dorini, L. B., Minetto, R. (2020). Inspection of imprint defects in stamped metal surfaces using deep learning and tracking. *IEEE Transactions on Industrial Electronics, 68(5),* 4498–4507. https://doi.org/10.1109/TIE.2020.2984453
19. Takahashi, T., Nozaki, K., Gonda, T., Mameno, T., Ikebe, K. (2021). Deep learning-based detection of dental prostheses and restorations. *Scientific Reports, 11(1),* 1–7. https://doi.org/10.1038/s41598-021-81202-x
20. Kwon, O., Kim, H. G., Ham, M. J., Kim, W., Kim, G. H. et al. (2020). A deep neural network for classification of melt-pool images in metal additive manufacturing. *Journal of Intelligent Manufacturing, 31(2),* 375–386. https://doi.org/10.1007/s10845-018-1451-6
21. Yan, X., Liu, Y., Jia, M. (2020). Multiscale cascading deep belief network for fault identification of rotating machinery under various working conditions. *Knowledge-Based Systems, 193(7),* 105484. https://doi.org/10.1016/j.knosys.2020.105484
22. Liu, S., Xia, Y., Shi, Z., Yu, H., Li, Z. et al. (2021). Deep learning in sheet metal bending with a novel theory-guided deep neural network. *IEEE/CAA Journal of Automatica Sinica, 8(3),* 565–581. https://doi.org/10.1109/JAS.2021.1003871
23. John, R. A., Acharya, J., Zhu, C., Surendran, A., Bose, S. K. et al. (2020). Optogenetics inspired transition metal dichalcogenide neuristors for in-memory deep recurrent neural networks. *Nature Communications, 11(1),* 1–9. https://doi.org/10.1038/s41467-020-16985-0
24. Fu, J., Zhao, J., Li, F. (2021). Infrared sea-sky line detection utilizing self-adaptive laplacian of gaussian filter and visual-saliency-based probabilistic Hough transform. *IEEE Geoscience and Remote Sensing Letters, 19,* 1–5.
25. Gunn, S. R. (1999). On the discrete representation of the Laplacian of Gaussian. *Pattern Recognition, 32(8),* 1463–1472. https://doi.org/10.1016/S0031-3203(98)00163-0
26. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. et al. (2014). *Going deeper with convolutions*. Ithaca, NY, USA: Cornell University Press. https://arxiv.org/abs/1409.4842v1

27. Chollet, F. (2017). *Xception: Deep learning with depthwise separable convolutions*. Ithaca, NY, USA: Cornell University Press. https://arxiv.org/abs/1610.02357
28. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2015). *Rethinking the inception architecture for computer vision*. Ithaca, NY, USA: Cornell University Press. https://arxiv.org/abs/1512.00567v3
29. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. Las Vegas, NV, USA. https://doi.org/10.1109/CVPR.2016.90