Journal of Big Data

RESEARCH

Open Access

Contrastive self-supervised representation learning framework for metal surface defect detection

Mahe Zabin[1], Anika Nahian Binte Kabir[2], Muhammad Khubayeeb Kabir[2], Ho-Jin Choi[1*] and Jia Uddin[3]

*Correspondence:
hojinc@kaist.ac.kr

[1] School of Computing, Korea Advanced Institute of Science and Technology, KAIST, Daejeon, South Korea
[2] Department of Computer Science and Engineering, School of Data and Sciences, Brac University, Dhaka, Bangladesh
[3] AI and Big Data Department, Endicott College, Woosong University, Daejeon, South Korea

Abstract

Automated detection of defects on metal surfaces is crucial for ensuring quality control. However, the scarcity of labeled datasets for emerging target defects poses a significant obstacle. This study proposes a self-supervised representation-learning model that effectively addresses this limitation by leveraging both labeled and unlabeled data. The proposed model was developed based on a contrastive learning framework, supported by an augmentation pipeline and a lightweight convolutional encoder. The effectiveness of the proposed approach for representation learning was evaluated using an unlabeled pretraining dataset created from three benchmark datasets. Furthermore, the performance of the proposed model was validated using the NEU metal surface-defect dataset. The results revealed that the proposed method achieved a classification accuracy of 97.78%, even with fewer trainable parameters than the benchmark models. Overall, the proposed model effectively extracted meaningful representations from unlabeled image data and can be employed in downstream tasks for steel defect classification to improve quality control and reduce inspection costs.

Keywords: Metal surface defects, Lightweight convolutional encoder, Semi-supervised learning, Self-supervised learning

Introduction

In the context of Industry 4.0, smart industrial monitoring plays a pivotal role across various industries, particularly when coupled with real-time automated diagnosis using artificial intelligence (AI) [1]. Strip steel, a fundamental material utilized in diverse sectors such as automobiles, military applications, tubes, appliances, refrigerators, washers and dryers, building materials, and electrical components, is susceptible to quality degradation due to multiple factors including production technology and rolling equipment [2, 3]. For instance, rolled-in scale defects on steel surfaces result from the peeling of the oxide film during rolling, while scratches may be inflicted by the friction between the roll and surface. Implementing effective defect inspection strategies is crucial to ensure the safe functionality of components and to minimize raw material wastage. These defects can exhibit various characteristics, including single-point occurrences, continuous

patterns, periodic types such as lines, scratches, spots, and holes, with intermittent, horizontal, and vertical distributions [4, 5].

Thus, conducting early inspections within the production pipeline can reduce production costs in the final stages [6]. Several methods have been employed for surface defect detection, encompassing manual detection [7], morphological image detection [8], machine vision-based detection [9–11], and magnetic flux leakage detection [12].

Manual inspection often suffers from varying interpretations of defects, leading to inconsistencies in identifying defects of different shapes and sizes. This process is time-consuming, labor-intensive, and prone to inaccuracies. To address these challenges, computer vision-based techniques have been increasingly employed. Traditional approaches for defect detection in computer vision involve image processing techniques such as edge detection using Sobel filtering on grayscale images, combined with classification using multilayer perceptron and support vector machine (SVM) classifiers [13]. Early studies focused on morphological operations, handcrafted feature extraction, and utilization AI algorithms [14–17]. Notably, Song et al. proposed a method that utilizes adjacent evaluations to complete local binary patterns, demonstrating reasonable classification of metal surface defects even in the presence of Gaussian noise. However, the ambiguity and similarity among steel surface defects limits their detection.

More recently, the adoption of machine learning (ML) and deep learning (DL) models has demonstrated promise in detecting defects in steel surfaces. Although ML models can learn low-level features, they tend to overlook intricate details. The advancements in DL architectures have enabled automatic detection of defects with improved precision. However, the increased depth of DL models introduces challenges, such as reduced inference speed and the requirement for larger labeled datasets. Furthermore, the lack of large labeled datasets limits the suitability of deep architectures for surface defect detection.

The objective of this study is to overcome the limitations associated with industrial defect classification, such as a scarcity of samples representing each defect within a given dataset and the laborious process of curating a labeled dataset.

Contributions

This study presents advancements over previous state-of-the-art self-supervised learning methods applied in the domain of industrial steel defect detection. We explored memory-efficient and adaptable self-supervised learning frameworks and conducted a comparative analysis against baseline methods. The results surpassed those achieved by existing supervised learning approaches. The framework employed in this study utilizes existing metal surface defect datasets and investigates the effect of transferring learned representations to a specific defect detection task. These representations are derived from the mapping of input features created by the intermediate layer and projection head of the convolutional encoder. However, the detection and recognition of small and complex targets remain persistently challenging. Issues such as limited labeled samples, increased computational time, low accuracy, imbalanced dataset problems, and interference from edge-lighting commonly arise. Metal surface defects often exhibit complexity and task-specific characteristics, further exacerbating the scarcity of samples. Our hypothesis posits that the encoder can gain a classification advantage by leveraging

existing surface defect datasets containing diverse defect types and transferring the acquired representations to a specific defect detection task.

Overall, this study makes the following contributions:

- A straightforward framework incorporating contrastive learning representations and nearest-neighbor contrastive learning was presented for steel defect detection. This framework integrates an augmentation framework and a lightweight encoder. The representation learning process utilizes a large unlabeled dataset, while fine-tuning and classification tasks are performed using a smaller labeled dataset.
- In the experimental evaluation, multiple interchangeable lightweight encoders were tested and compared against baseline classification tasks. The performance of the model was compared with that of five state-of-the-art surface defect classification models.

The remaining sections of this paper are organized as follows: ("Related Works") section discusses related studies pertaining to steel defect detection. ("Proposed Methodology") section presents the proposed methodology, providing a detailed description of the self-supervised learning framework. In ("Experimental Results and Discussion") section, the experimental results and analysis are presented. Finally, ("Conclusions") section concludes the paper, outlining the limitations of the study, and offers directions for future research.

Related works

In recent years, researchers have explored various approaches, including traditional machine learning, deep learning, and adaptive learning-based methods, to steel surface defect detection using crack image datasets. Numerous studies have focused on applying basic computer vision-based machine learning (ML) models for steel surface defect detection (SSDD) using surface crack image datasets. For instance, in [18], a random forest (RF) classifier was employed for SSDD, accompanied by feature extraction methods such as gray-level co-occurrence matrix (GLCM), wavelet, and histogram of gradients (HOG). The performance of the model was evaluated using an SVM classifier with limited parameter variations. Another study by Bin et al. [19] proposed a machine vision model for SSDD that extracted invariant moment features from steel cracks. Furthermore, in [20], several ML algorithms, including SVM, K-nearest neighbors, Gaussian process, decision tree, RF, artificial neural network, naive Bayes, and AdaBoost were experimented with for rapid surface defect identification. However, advanced ML techniques such as deep learning, transfer learning, and adaptive learning offer significant performance improvements. In the subsequent subsections, we summarize the state-of-the-art techniques applied to these robust and advanced ML approaches. A concise overview of cutting-edge studies is provided in tabular form at the conclusion of this section.

Deep learning based methods

Furthermore, researchers have explored the application of Deep Convolutional Neural Network (CNN) architectures for steel surface defect detection (SDD) utilizing surface

crack image datasets. Božič et al. [21] employed an end-to-end training approach with a two-stage neural network for detecting various defects, including steep defects. A unique loss function was utilized to address the uncertainty associated with region-based annotations and compared with precise pixel-level annotations. Similarly, Chi et al. [22] employed a DCNN architecture for steel-crack detection, achieving notably higher accuracy in the experimental evaluation compared to other crack datasets. Despite claiming applicability for online steel prediction, their experiments did not consider the computational complexity and training period.

Masci et al. [23] employed MaxPooling with a CNN for SSD and compared its performance with traditional approaches. In another study [24], a DCC-Center Net architecture was employed for SSD detection, where keypoint estimation was used to identify center points and regress defect properties. However, the model exhibited limitations in detecting obscure defects because of its segmentation method and network architecture. Furthermore, in [25], a domain-adaptation adaptive CNN was utilized for SSD, employing adaptive learning rates based on loss and weight. Although the proposed model demonstrated improved results compared to conventional CNN models, it was evaluated using a small dataset.

Bhatt et al. [26] conducted a review of DL-based techniques for steel surface defect detection (SSDD), focusing on classification and localization. Yang et al. [27] proposed a CNN model with multiple convolutional layers, each utilizing varying kernel sizes to enhance the receptive field. Benbarrad et al. [11] employed a CNN architecture on compressed steel images for SSD classification, demonstrating comparable performance to that of the uncompressed images. Liu et al. [28] utilized a hybrid architecture combining long short-term memory (LSTM) and CNN, whereas Singh et al. [66] employed a ResNet101-SVM architecture for surface defect classification. For the automatic detection of small and complex steel defects, a deformable convolution network with multi-scale feature fusion was applied [29]. However, due to the several steps involved in the model, its detection time was relatively higher compared to other models.

Konovalenko et al. [30, 64] utilized a residual neural network (RNN), whereas Hao et al. [31] employed a modified RNN with attention blocks for defect classification in strip steel. RNNs are computationally demanding, slower, and require larger amounts of data compared to CNNs. Zhou et al. [32] proposed an effective training approach for defect detection, incorporating knowledge distillation, attention mechanisms, and feature fusion, achieving over 90% area under receiver operating characteristic (ROC) curve. Anvar et al. [33] conducted experiments with ShuffleDefectNet on the NEU metal surface defect dataset, achieving impressive generalization performance.

Hao et al. [34] proposed DF-ResNeSt50, a split-attention network for SSDD, which demonstrated enhanced and optimized data augmentation capabilities. In another study [35], a Multi-SE-ResNet34 architecture incorporating squeeze convolution layers and excitation blocks was utilized for SSDD, achieving improved accuracy. However, the computational complexity of the network is high. Two neural networks, UNet and Xception, were employed in [36] for steel defect detection, where UNet was used for segmentation and Xception for classification. Xception, known for its depth-wise separable convolutions, introduces significant computational complexity. In [37], the RepVGG algorithm with spatial attention was applied to hot-rolled SSDD, showcasing improved

classification performance compared to ResNet, VGG, and MobileNet. Nevertheless, the model performs inadequately when detecting small-sized oxide scales on plates.

Anchor-free feature extraction using YOLO-V3 has been employed in steel defect detection, which reduced the computation time compared to state-of-the-art models. However, its detection accuracy reached only 70%, and it struggled to detect high-resolution images with extremely small defects. More recently, CP-YOLOv3-dense [38] has been applied for SDD, utilizing multilayer convolutional features for predictions and demonstrating improved results compared to the YOLO-V3 model. Lightweight architectures such as Ghost-CBAM-YOLOv4 (GCB-Net) [39] and an improved YOLOv5-based transformer (MSFT-YOLOv5) [9] have been employed for SSDD. Additionally, a lightweight YOLOv5 architecture with adaptive bounding box annotation is proposed in [40]. Although YOLO models exhibit higher detection accuracy, their performance is heavily reliant on image labeling, which is a time-consuming task.

Transfer learning based methods

In a recent study [41], ResNet50 deep architecture- a hybrid architecture combining ResNet50 and enhanced fast-region CNN [42]—has been utilized for detecting hot-rolled steel defects. However, these models exhibited a considerable number of false positives and high computational complexity. Wan et al. [43] employed a VGG19 architecture with pretrained image weights, proposing an improved VGG-19 network for steel surface screening and defect generalization. Additionally, Feng et al. [44] introduced a vision transformer model with a deep VGGNet architecture, achieving enhanced accuracy. Note that VGG architectures are computationally demanding owing to their large number of trainable parameters.

In another study [45], a TL-SDD model was employed, utilizing the transfer of common defect class knowledge to detect rare defects. However, the model was evaluated using only one rare type of defect. In [46], a transfer learning-based U-Net, composed of ResNet and DenseNet encoders, was applied to SDD with random initialization and image weights. Unfavorably, this model demonstrated unsatisfactory performance for rare and complex defects. Waqas et al. [63] employed seven pretrained CNN architectures for crack detection: GoogLeNet, MobileNet-V2, Inception-V3, ResNet18, ResNet50, ResNet101, and ShuffleNet. Although Inception-V3 outperformed the other models, it achieved an overall accuracy of 88.5% with 24 M trainable parameters.

Adaptive learning based methods

The challenges associated with routine annotation and the associated costs remain unaddressed in DL/DTL architectures. Therefore, the adoption of semi-supervised learning techniques becomes imperative. Self-supervised learning develops a network to learn from a large volume of unlabeled data, enabling it to recognize crucial patterns before fine-tuning with a fraction of labeled data. Mayuravaani et al. [47] proposed a semi-supervised learning technique utilizing a CNN to predict the weights of unlabeled data. In the subsequent step, the predicted labels, combined with labeled data, were employed to train their architectures. The unlabeled data underwent training using the contractive autoencoder (CAE), followed by the utilization of a semi-supervised generative adversarial network (GAN) to train the

network. In another study [48], an improved semi-supervised multitask GAN (iSSMT-GAN) has been introduced for defect detection, avoiding issues such as gradient disappearance or overfitting while generating high-quality image features. Although the model demonstrated enhanced accuracy compared to other state-of-the-art DL models, its validation was limited to a surface defect dataset.

Zhang et al. [49] proposed CADN, a weakly supervised learning method for surface detection that simultaneously detects defects and classifies images, utilizing a knowledge distillation strategy to expedite the training process. However, in this particular model, a lighter version of CADN was employed for knowledge distillation. In ref. [50], a two-layer DL model was employed for SSDD, with one layer dedicated to pixel-level segmentation and the other for image label classification. This model was trained using a combination of fully (pixel-level) and weakly (image-level) labeled data. To enhance the awareness of the network toward subtle anomalies, a saliency map-guided defect segmentation technique was employed in conjunction with self-supervised learning [51]. Zhao et al. [52] introduced a feature-attention convolution module combined with a few-shot learning approach. The module effectively captured long-range relationships between features, leading to improved performance across 5-way 1-shot and 5-way 5-shot tasks.

Hu et al. applied an efficient CNN model with object level attention mechanism for defect detection on radiography images [69]. Although the model works with small dataset, fine-tuning the hyper-parameters of CNN varies experiment to experiment and the radiographic images with low resolution are not sufficient for detecting small surface defects. In ref. [70], Lou et al. applied cross-attention transformer encoder-decode network (CAT-EDNet) for strip steel surface defect detection. The model adaptively allocates the aggregation weights that represent differentiated channel-wise information entropy values. Although the model maintains the details of defects boundaries in noise scenario, inference time is still an issue with this model to meet the real time defect detection. In ref. [71], a multi-label defect classification method is proposed with a novel self-purification module (SPM) consisting of intraclass purification (ICP) and interclass decorrelation (ICD). The ICP purify the features from the task-aware information and ICD eliminates the cross-correlation and defect-irrelevant components. The model is computational burden due to its multiple labels.

Several state-of-the-art representation learning frameworks incorporate memory mechanisms to retrieve representations. One such framework is contrastive multiview coding (CMC), which learns representations by maximizing the shared information between two views [53]. However, this approach is considered inefficient as the memory bank requires periodic updates. Another approach, autoregressive contrastive predictive coding (CPC), utilizes an encoder trained using the noise contrastive estimation loss [54]. Unsupervised learning with momentum contrast is based on the functions of the momentum encoder [55], comparing an encoded query image to multiple key images to maximize mutual information. In comparison, a simple framework for contrastive learning and nearest-neighbor self-supervised contrastive learning demonstrated superior results compared to CPC, MOCO, and CMC Table 1.

Table 1 Summary of the state-of-the-art defect classification papers

Category	Objectives and features	Limitations
Machine Learning [18–20]	Surface defect classification; hand-crafted feature extraction	Extracted features are sensitive to noise and variability, computationally intensive
Deep Learning [32, 33],	Lightweight architecture for defect detection; transfer of valuable defect features to student model, handle complex and varied defect patterns extracted using evolving convolutional architectures	Requiring large volumes of labeled examples, overfitting and poor generalization on rare classes, does not aid detection of rare and newer defects
TL-based methods [9, 28, 43, 44, 46]	Use of small dataset, saves training period; usage of prior knowledge towards target domain	If source and target tasks are not adequately representative, they become redundant; negative transfer
Adaptive Learning [50, 52, 53, 55, 57]	Defect detection with limited samples; process unlabeled data to obtain useful defect representations	Poor results for rare surface defects, computational complexity issue for real-time monitoring

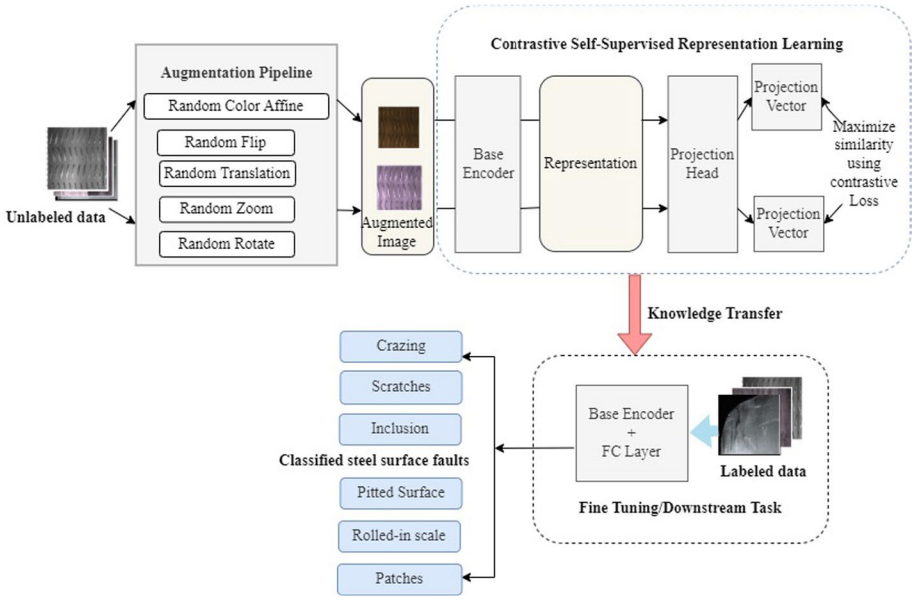

Fig. 1 Detailed methodology of the model with contrastive self-supervised representation learning and fine-tuning

Proposed methodology

In this section, we provide a detailed description of the dataset used in our study, including the unlabeled dataset employed for self-supervised learning and the labeled dataset utilized for fine-tuning.

Furthermore, we explain the functioning of the augmentation pipeline with the help of sample figures. We then delve into the contrastive loss function employed in our framework. Finally, we present the specifics of the base encoders used as classifiers for self-supervised representation learning. A block diagram illustrating the utilization of contrastive self-supervised representation learning for steel crack classification is depicted in Fig. 1.

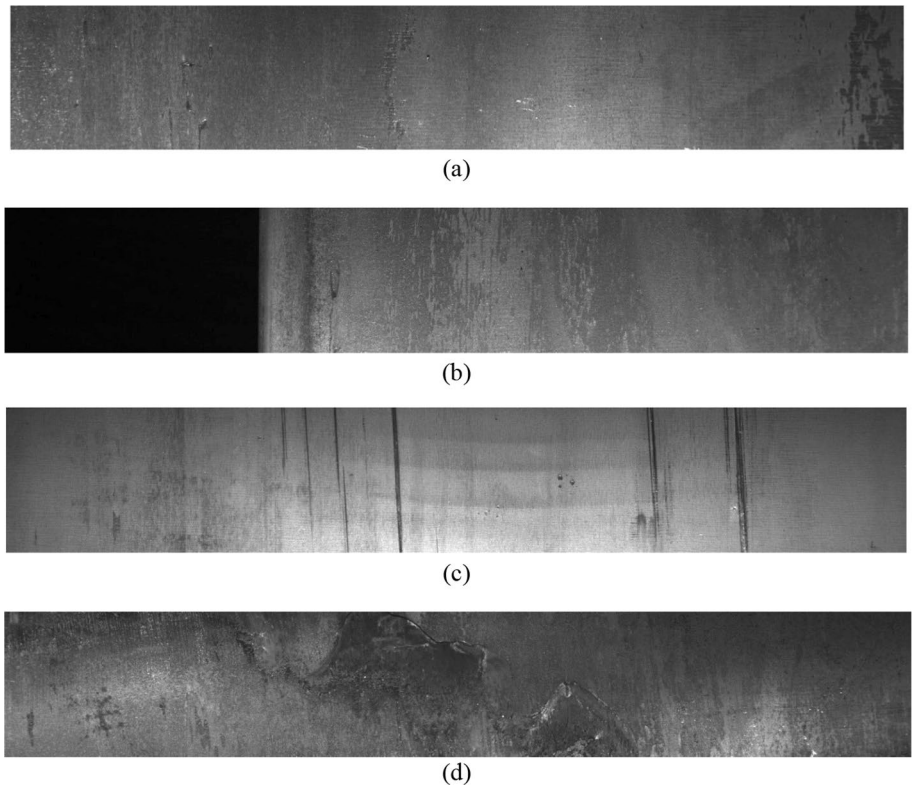

Fig. 2 Sample images of the Severstal steel dataset, (**a**) Type 1, (**b**) Type 2, (**c**) Type 3, and (**d**) Type 4, respectively

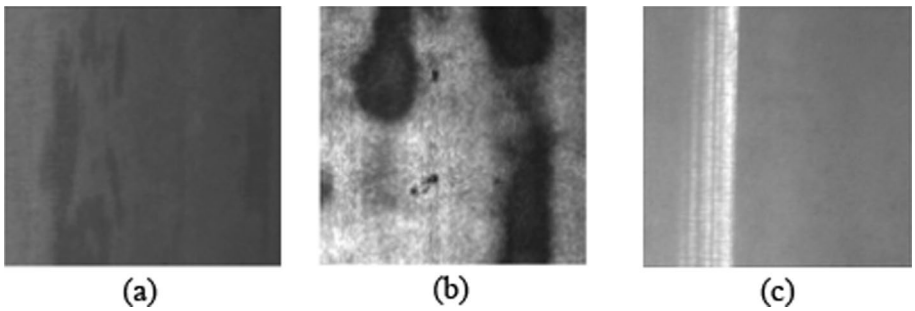

Fig. 3 Sample images of SD-Saliency-900 dataset (**a**) inclusion, (**b**) patches, and (**c**) scratches, respectively

Dataset

The self-supervised representation learning phase involved three public datasets: Severstal Steel (SS) Defect, SD Saliency 900, and GC-10 DET. The SS dataset captured defect classes using a high-frequency camera and consisted of four defect classes, as illustrated in Fig. 2. Among these datasets, the SS dataset was the largest, containing 18,076 images of defective and non-defective steel. Defects exhibited varying shapes, sizes, and appearances. The SD-Saliency-900 dataset comprised three defect classes: inclusions, patches, and scratches. Sample defects from the SD-Saliency-900 dataset are portrayed in Fig. 3. GC-10 DET is a comprehensive metal defect image dataset that encompasses ten different categories of defects. Within the GC-10 DET dataset, a few images exhibited

multiple types of defects, rendering them suitable for localization tasks and representation learning. The NEU metal surface defect dataset comprises six defect types, namely rolled-in scales, patches, crazing, pitted surfaces, inclusions, and scratches.

For the experimental evaluation and fine-tuning, we utilized the NEU metal surface defect dataset. Figure 4 showcases sample images from this dataset, which comprised approximately 1,800 grayscale images. During the pretraining phase, a combined dataset of 21,222 strip steel defect images from three public datasets, covering 16 types of defects, was employed.

Augmentation pipeline

The stochastic nature of the data augmentation pipeline, as depicted in Fig. 1, played a significant role in the training process. The selection of data augmentation techniques had a substantial impact on the training outcomes.

The pipeline involved random flipping and random cropping of images. Additionally, random color distortions were applied by manipulating the color spaces. Figure 5 illustrates a selection of randomly augmented images utilized in the simple contrastive representation learning framework. The nearest neighbor representation learning framework utilized common augmentation techniques such as random cropping and flipping.

Contrastive loss

Contrastive loss was originally proposed by Hadsell et al. [56] in the context of dimensionality reduction. The general form of the contrastive loss function is expressed as,

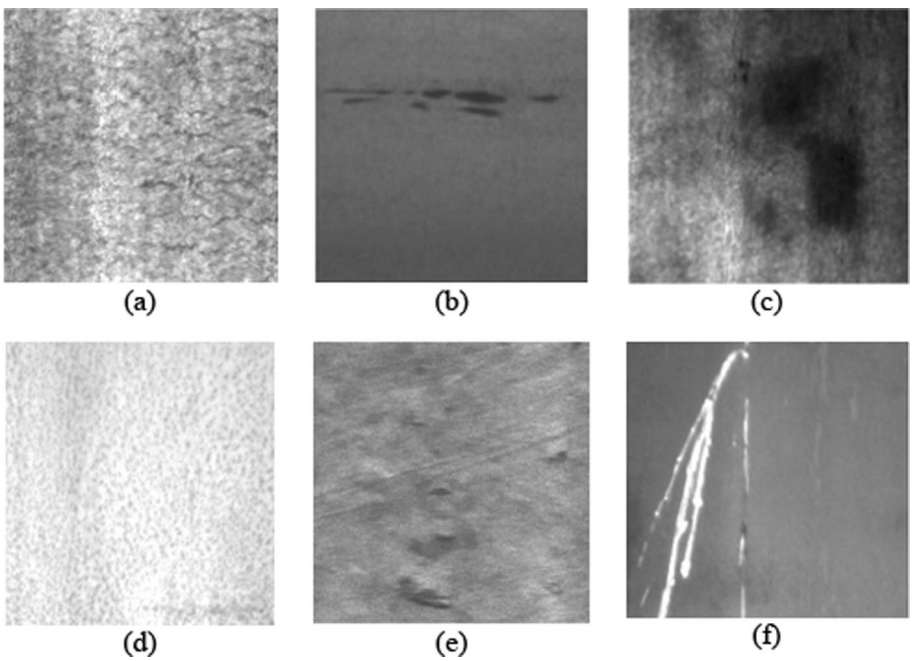

Fig. 4 Sample images of NEU metal surface defect dataset: (**a**) crazing, (**b**) inclusion, (**c**) patches, (**d**) pitted surface, (**e**) rolled-in scale, and (**f**) scratches

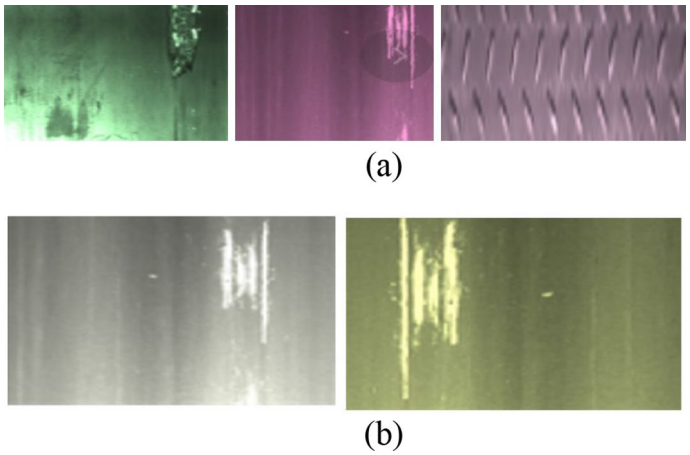

Fig. 5 **a** Random color distortions and (**b**) flipped image and color distortions added on (**a**)

$$L(W,(Y,X_1,X_2)) = (1-Y)L_S\left(D_w^i\right) + YL_D\left(D_w^i\right) \tag{1}$$

where X_1 and X_2 represent two images that are either similar or dissimilar. The term D_w denotes the similarity between two data points, L_S and L_D representing the loss functions for the similar and dissimilar cases. In our framework, we employed a simple contrastive learning approach that utilizes a specific instance of contrastive loss known as the normalized temperature-scaled cross-entropy loss (NT-Xent). The initial step of the loss function involves calculating the cosine similarity between two augmented images in a pair, denoted by s_i and s_j. The variable temperature, t, allows for similarities within a certain range, typically from -1 to 1.

$$s_{i,j} = \frac{z_i^T z_j}{t\left(||z_i||||z_j||\right)} \tag{2}$$

For each minibatch, the similarities between augmented pair are computed using Eq. 2, where z_i and z_j represent a pair of outputs from the projection head g, as depicted in Fig. 6. These similarities are then passed through a Softmax function, yielding the probability of the pairs being similar.

The loss function for positive pairs of augmented images was calculated by taking the negative logarithm of the Softmax function, as.

$$l_{i,j} = -log\frac{exp\left(s_{i,j}\right)}{S_{k=1}^{2N} l_{[k!=i]} exp\left(s_{i,k}\right)} \tag{3}$$

where N denotes the minibatch size. The overall loss for all pairs in the minibatch was calculated using the equation expressed in Eq. 4, where N denotes the minibatch size.

$$L = \frac{1}{2N} S_{k=1}^{N}[l(2k-1,2k) + l(2k,2k-1)] \tag{4}$$

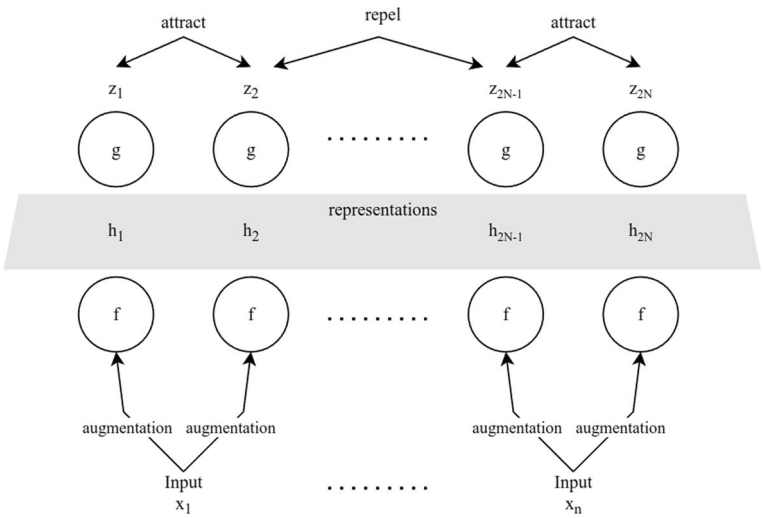

Fig. 6 Simple framework for contrastive learning representations

Self-supervised learning frameworks

In their work, Chen et al. [57] proposed a straightforward framework for contrastive learning representations, utilizing random cropping with resizing, random color distortions, and Gaussian blurring to transform unlabeled images. Herein, we implemented an augmentation sequence to generate augmented images. Thereafter, these images were used to extract representations using various encoders, such as a stack of convolutional layers, RNNs, ShuffleNet, or SqueezeNet, denoted as f in Fig. 6. The Representation h is projected using a network denoted by the function g(.). The projection head aided in mapping the representations to a vector space, typically consisting of a nonlinear hidden layer that generated the representations z. At this stage, the NT-Xent loss function is applied, considering the similarities between pairs. The resulting pretrained network can then be utilized to transfer the knowledge of the learned representations. In our case, encoder representations were employed for transfer learning. Figure 1 provides an abstract overview of the framework.

Dwibedi et al. [58] introduced nearest-neighbor contrastive learning as a representation learning framework. Unlike using augmented variations, this framework maximizes similarity by considering different images from the same instance or across samples. An encoder is employed to sample the nearest neighbors and form a support set. This support set is constructed in a latent space, where the samples are treated as positive examples, and the embeddings are continually updated. In contrast, SimCLR focuses on maximizing similarity between multiple views of the same image. The support set is represented as a two-dimensional (2D) matrix, with one dimension representing the queue size and the other dimension representing the embedding size. Note that the effectiveness of generalization heavily depends on the chosen augmentation sequence. Thus, representation learning using this framework is highly sensitive to the choice of augmentation techniques.

The nearest-neighbor contrastive learning technique utilizes a loss function based on noise contrastive estimation (NCE) [59], known as Contrastive Predictive Coding

(CPC), which was proposed by Van den Oord et al. [54]. In essence, the InfoNCE loss brings positive pairs closer together in the embedding space while simultaneously learning the differences between negative pairs. The following equation represents the version of InfoNCE loss employed by NNCLR:

$$L_i^{NNCLR} = -log \frac{\exp\left(NN(z_i, Q).\frac{z_i^+}{t}\right)}{S_{k=1}^{n} \exp\left(NN(z_k, Q).\frac{z_i^+}{t}\right)}, \tag{5}$$

where $NN(z_i, Q)$ represents the l_2 norms of z and element q in the support set. This equation calculates the nearest-neighbor relationship as follows:

$$NN(z, Q) = argmin||z - q||_2. \tag{6}$$

The encoders utilized in the experiments varied from simple convolutional encoders to more efficient and lightweight CNNs suitable for defect detection. A basic convolutional encoder consisting of four 2D convolutional layers with 128 filters was stacked. The resulting array was then passed through a dense layer of 128 units, using the rectified linear unit activation function. To address potential issues of vanishing gradients in deeper architectures, the number of convolutional layers was increased, and standard skip connections were added to provide alternative paths for backpropagation. This technique, known as skip-ConvNet, is depicted in Fig. 7, illustrating the architecture of skip-ConvNet and the lightweight convolutional encoder.

SqueezeNet, proposed by Iandola et al. [60], achieved results comparable to AlexNet but with fewer parameters. It is a compact network with a fire module that serves two functions. First, the 1×1 convolutional filters squeeze the feature maps, and the output is fed into expanding layers, which consist of convolutional layers with 3×3 filters.

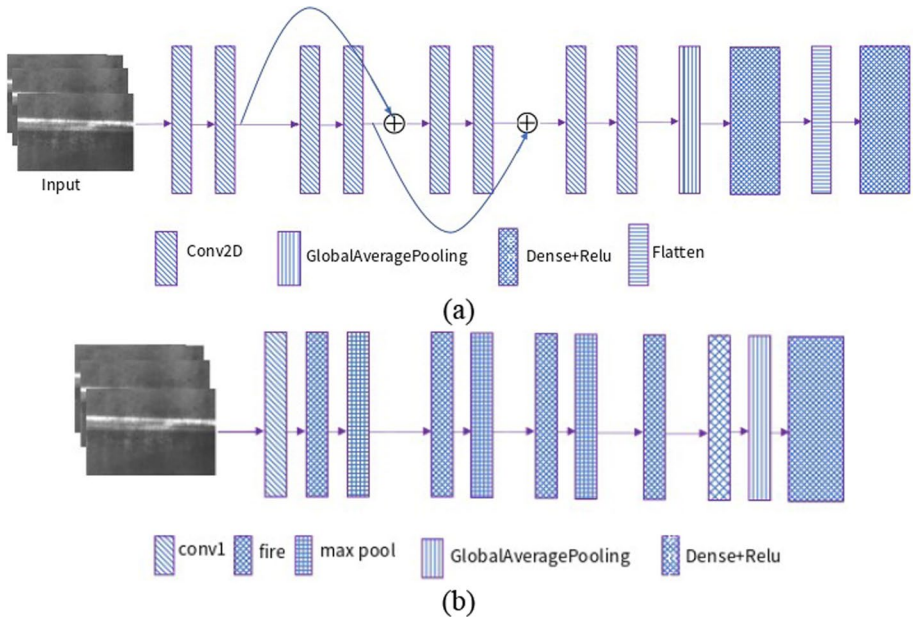

Fig. 7 Layers information: **a** Convnet with skip connections; **b** Convolutional neural network

ShuffleNet, proposed by Zhang et al. [61], is a computationally efficient CNN originally composed of four stages, as depicted in Fig. 8b. Inspired by the bottleneck units of residual networks, it incorporates ShuffleNet units in each stage, as depicted in Fig. 8a, b. ShuffleNet-v2, introduced by Ma et al. [62], demonstrated that dependency on group convolutions can reduce computation speed. ShuffleNet-v2 utilizes channel-split operations, splitting feature channels into two branches. After convolution operations, the channel branches are concatenated.

The channel shuffle operation facilitates the exchange of information among feature channels. In this study, ShuffleNet-v2 basic units and down-sampling units were utilized as building blocks in three stages. In the initial stage, the input passed through a convolutional layer with a kernel size of three. A max-pooling layer with a kernel size of three and a stride of two was applied. The building blocks of ShuffleNet, represented as Units (a) and (b) in Fig. 8, were repeated twice in stages 2 and 3.

For the SimCLR and NNCLR frameworks, images were resized to 128×800 and 200×500 pixels, respectively. All experiments were conducted and trained on a T4 GPU system with 52 GB of RAM. The learning rate was set to 0.0001 with a batch size of 32. The network was trained using the Adam optimizer. Subsequently, the networks were trained for 100 epochs.

Experimental results and discussion

The experimental results obtained using the proposed method are described in the following subsections. To evaluate the model performance, we trained the best-performing architectures using similar parameters without self-supervised contrastive pretraining. In this section, we measure the accuracy, precision, recall, and F1 scores for all the architectures. Additionally, we discuss the validation accuracy, loss curves, and overall outcomes. Accuracy and F1 scores are essential metrics for binary and multiclass classification problems. Accuracy represents the percentage of correctly predicted labels out of the total number of samples, which is calculated as follows:

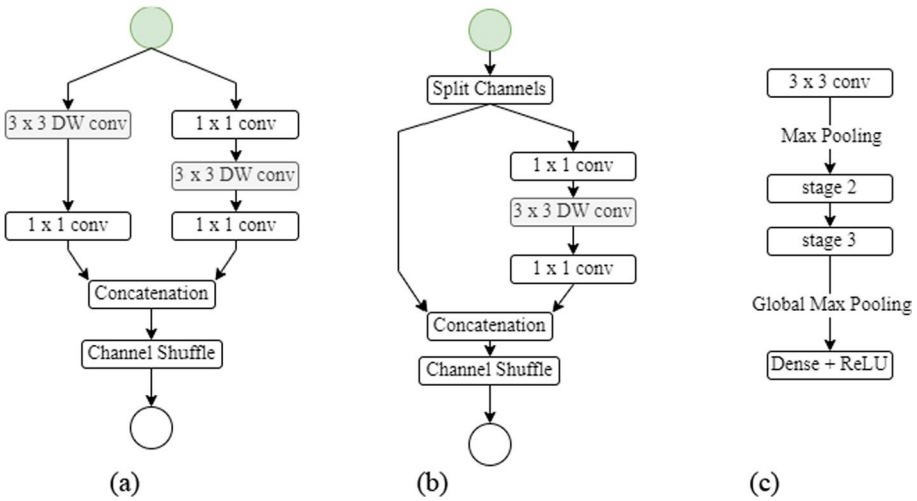

Fig. 8 ShuffleNet architecture that includes basic building blocks depicted in (**a**, **b**); **c** high-level illustration of the stages of ShuffleNet

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision is the ratio of correctly identified positive cases to all predicted positive cases:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Recall measures true positives (TP) from all actual positive cases:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

where TP stands for true positive, TN is for true negative, FP is for false positive, and FN is for false negative.

The F1-score is a reliable metric in practical scenarios with imbalanced class distributions, defined as the harmonic mean of precision and recall:

$$F1 - Score = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

The area under the curve and receiver operating characteristic (AUC–ROC) metrics demonstrate the capability of the model to differentiate among classes. The ROC plots sensitivity (or recall) against the false-positive rate, representing a probability distribution curve. A high sensitivity at a minimal false-positive rate indicates good discriminative power. Models with higher generalization ability have a greater area under the ROC curve.

Simple framework for contrastive representation learning

The encoders were pretrained on a large dataset comprising 16 types of defects for 100 epochs, optimized using the Adam optimizer. Callbacks such as Early Stopping with a patience set to 15 epochs were used to train the models for the optimal number of epochs and prevent overfitting. Two different losses, contrastive and linear probe losses, were used to evaluate the performance of the model. Contrastive accuracy indicated the proportion of cases in which the original images were similar to their altered versions in a batch of samples containing augmented images. The quality of the representations was evaluated using linear probing accuracy and probing loss in each epoch. Linear probe metrics were computed by using a classification layer on top of the frozen encoder during the pretraining phase. The proposed approach utilized a lightweight deployable DL architecture as an encoder suitable for resource-constrained settings.

To prevent overgeneralization during fine-tuning tasks, we employed validation loss-based early stopping as a callback metric. Contrast accuracy and loss were monitored in this study.

Several DL architectures suffer from overgeneralization, high variance, and resource constraints when dealing with small target defects. In contrast, lightweight CNNs enable more efficient cross-server communication during the training process,

making them suitable for small defect datasets. The selected encoders require less bandwidth when exporting the models to different hardware.

During the pretraining stage, the simplified ConvNet encoder demonstrated a contrastive accuracy of 91%, whereas the three-stage ShuffleNet achieved an accuracy of 86%. On top of the pretrained encoder, a dense layer with six units and Softmax activation applied. The results of fine-tuning using different encoders are summarized in Table 2. ConvNet and Skip-ConvNet achieved accuracies of 97.78% and 97.41%, respectively, with the latter having approximately one-sixth of the parameters. However, fire modules in SqueezeNets exhibited certain drawbacks such as low accuracy and high complexity, despite comprising lower number of architectural parameters. The fire module combines 1×1 and 3×3 kernels, resulting in smaller receptive fields compared to architectures with stacked 3×3 convolutional layers. Deep convolutional networks such as ResNet50 can introduce high variance and mild overfitting for datasets with small sample sizes, while incorporating more model parameters. After pretraining, ResNet50 achieved an accuracy of 91.11% on the NEU metal defect dataset.

ShuffleNet significantly improves the framework performance owing to its channel shuffle mechanism and group convolutions. The network was customized to achieve optimal complexity for the framework, resulting in a half the number compared to Skip-ConvNet. ConvNet achieved an accuracy of 97.78% by using stacked convolutional layers with 3×3 kernels, allowing gradual abstraction of the input image and extraction of low-level features. The network avoids overgeneralization, primarily because of its smaller depth and pooling layers. By omitting pooling layers, ConvNet captures the entire feature space, including both high- and low-level features, for the pretraining and classification tasks. However, ShuffleNet demonstrate superior efficiency and effectiveness with fewer model parameters, albeit with a slightly lower accuracy. The encoders were pretrained on the unlabeled pretraining dataset and then fine-tuned on the NEU metal surface defect dataset. The results achieved by different encoders on the target dataset after self-supervised pretraining are summarized in Table 2. For comparison, we conducted experiments using the three base encoders as standalone classifiers and a simple augmentation pipeline, treating them as baselines for downstream classification tasks. In principle, we aimed to evaluate whether the encoders would yield lower accuracy without the self-supervised pretraining framework.

Table 2 Self-supervised pretraining and finetuning using SimCLR framework

Encoders	Accuracy (%)	F1-score	Parameters
SqueezeNet	88.89	0.88	0.13 M
ResNet50	91.11	0.91	49.80 M
ShuffleNet	97.04	0.97	0.41 M
Skip-ConvNet	97.41	0.97	0.92 M
ConvNet	97.78	0.98	6.01 M

Table 3 Baseline Classification Results

Encoders	Accuracy (%)	F1-score
ConvNet	91.00	0.91
Skip-ConvNet	92.00	0.92
ShuffleNet	94.81	0.95

Therefore, the hyperparameters of the encoders remained unchanged. The baseline classification results listed in Table 3 displayed a marginal but significant decrease in test accuracy for the networks.

The confusion matrix for the fine-tuned ConvNet encoder, pretrained using the SimCLR framework, is illustrated in Fig. 9. Four out of six classes were correctly classified, with no false positives, whereas the remaining two classes had three to four false positives. Validation loss and accuracy curves for the baseline, pretraining,

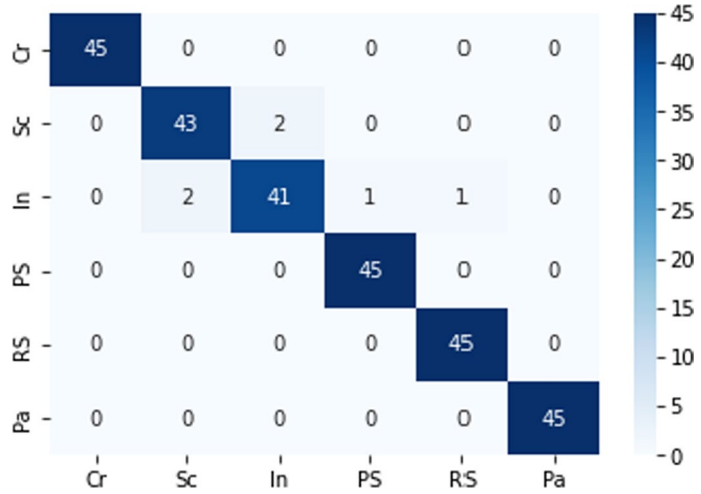

Fig. 9 Confusion matrix of simple convolutional encoder

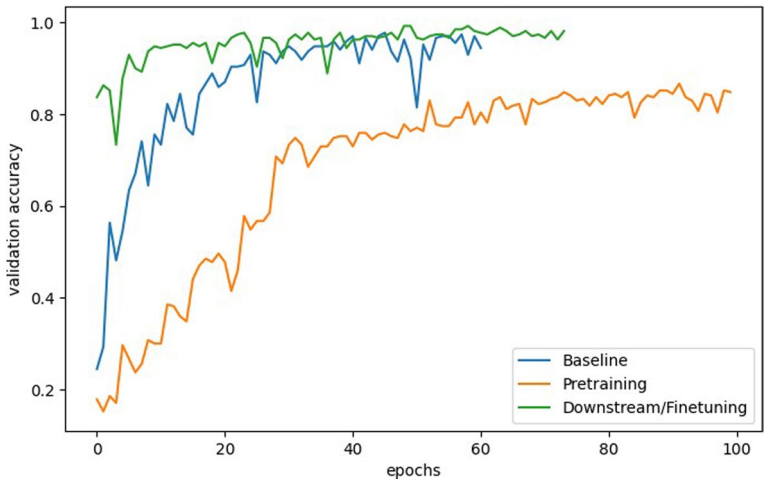

Fig. 10 Validation accuracy curve of convolutional encoder: baseline, pretraining, and downstream/finetuning

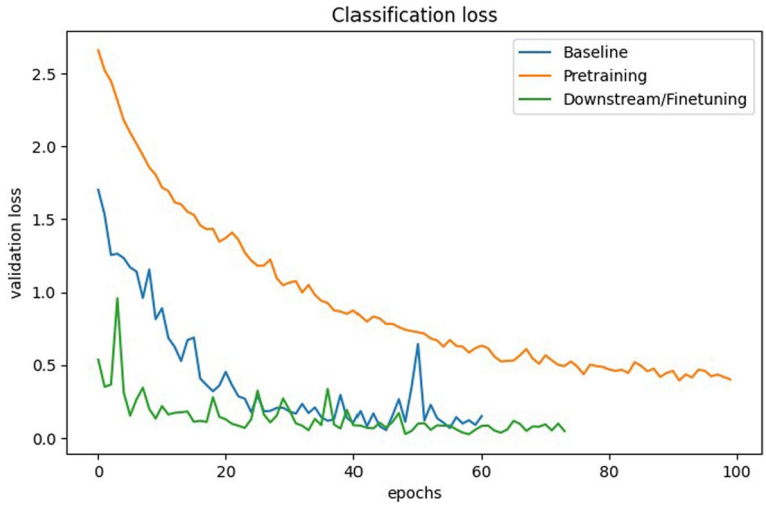

Fig. 11 Validation loss curve of ConvNet: baseline, pretraining, and downstream/finetuning

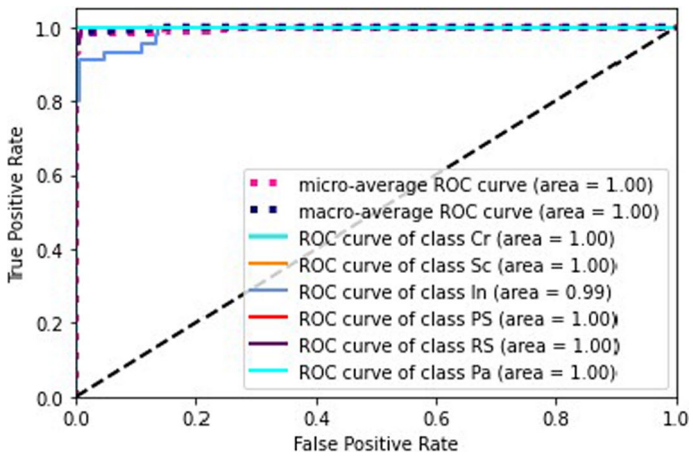

Fig. 12 ROC curves for the six defect classes generated from the ConvNet

and fine-tuning of the ConvNet encoder are illustrated in Figs. 10 and 11, respectively. After training on the pretraining dataset, the encoders exhibited a classification advantage. The pretrained ConvNet reached a higher validation accuracy faster compared to its baseline counterpart. The encoder achieved a lower validation loss due to the self-supervised pretraining, as illustrated in Fig. 11. Early Stopping with a patience value of 15 was used as a callback, resulting in the models converging at different epoch numbers. We used the early stopping function to reduce the model-overfitting and training time.

As an assessment of the practical reliability of the model, the ROC curves for the six defect types are plotted in Fig. 12, wherein the dotted boundary represents the diagonal in which the false positive rate equals the true positive rate. In principle, the performance of the classifier is indicated by the proximity of a curve to the diagonal (i.e., distance to the diagonal), based on which the pretrained classifier exhibited an extremely high true positive rate.

The convolutional architectures demonstrated reduced training time per step. ResNet50 took 55 ms per step due to its depth, while the squeeze net, with nine fire modules, had a longer training time of 35 ms per step compared to the convolutional encoder, skip ConvNet, and Shufflenet.

A simple CNN consists of convolutional operations, pooling, and fully connected layers. The proposed ConvNet includes a total of convolutional and fully connected layers. The total number of parameters for the four stacked convolutional layers can be represented as $4 \times k \times \left(f \times f \times c + 1\right)$, where f, c, and k denote the filter size, number of channels, and number of filters, respectively. The fully connected layer adds additional parameters. In the case of skip-Convnet, additive skip connections $n \times n$ introduce parameters in addition to the convolutional layers, where n represents the dimension of the matrix.

ResNet50 has 23.521 million trainable parameters, as reported in the literature. Therefore, shallower models exhibited significantly reduced training times, ranging from 8 to 10 ms per step.

Nearest neighbor contrastive learning of representations

The support set is a priority queue updated by removing old embeddings and introducing new representations. Table 4 summarizes the effect of self-supervised pretraining, depicting an improvement in test accuracy compared to the baseline results presented in Table 2.

The framework yielded optimal results with a queue size of 93,000 the ImageNet2012 dataset, which contained over one million images and required a large embedding space. However, in the experiments, a queue size of 10,000 was determined to be optimal for a simple ConvNet, and this size was used in the subsequent experiments.

Figure 13 provides the details of the confusion matrix for SkipConvNet using the NNCLR framework for different classes of defects. Similar to the SimCLR framework, the Inclusion (In) and Scratch (Sc) classes, each with three to four false positives, whereas the Split Surface (Ps) class contains only one false positive. Figure 14 displays a comparative reduction in validation loss after self-supervised pretraining. Interestingly, fine-tuning a pretrained encoder with the NNCLR framework exhibited trends similar to those observed with the SimCLR framework. As depicted in Fig. 13 and Fig. 15, the confusion matrix and ROC curve indicated a fair generalization ability for all classes. In this setting, the Inclusion class achieved an AUC of 0.97, which was slightly reduced in comparison to its previous value.

Table 4 Results of different base-encoders using the NNCLR framework

Encoders	Accuracy (%)	F1-score	Parameters
Simple ConvNet	95.56	0.96	5.85 M
Skip-ConvNet	97.04	0.97	0.92 M
ShuffleNet	95.92	0.96	0.40 M

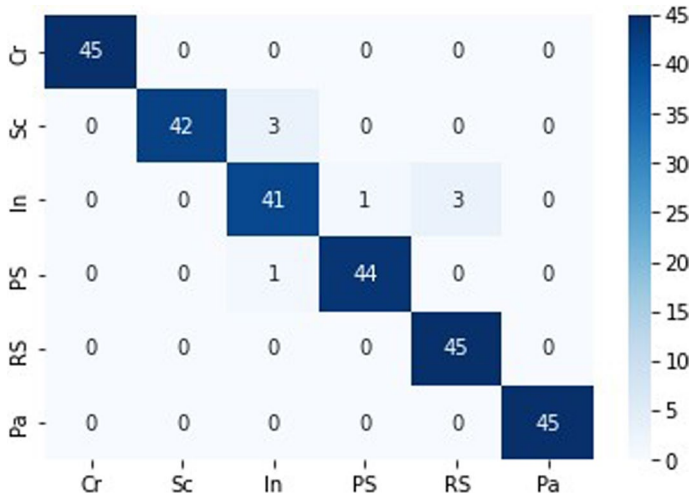

Fig. 13 Resulting confusion matrix of Skip-ConvNet architecture

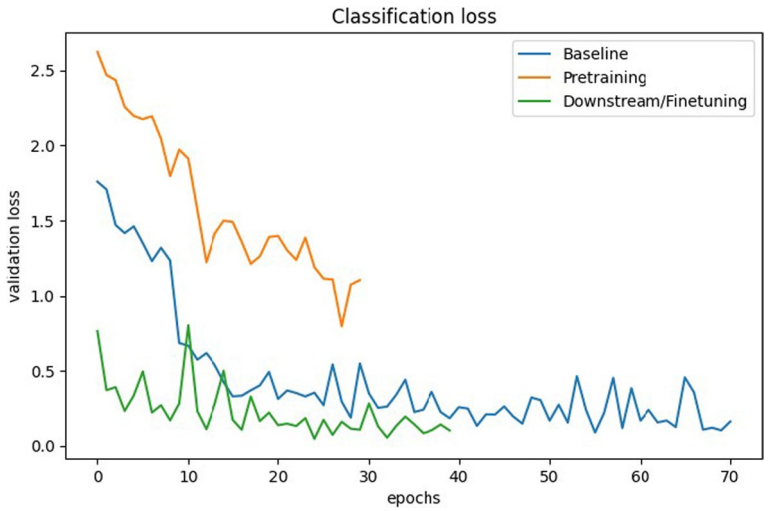

Fig. 14 Validation loss curve displaying a reduction in validation loss for a smaller number of epochs

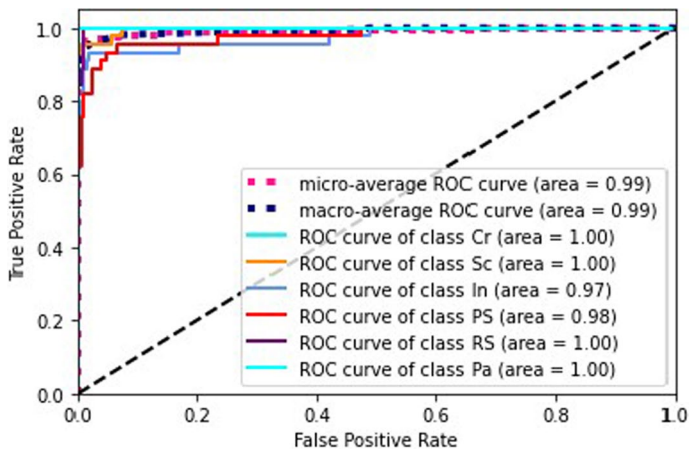

Fig. 15 ROC curve for skip-ConvNet architecture

Comparative analysis with surface defect classification methods

The networks were trained using a learning rate of 0.001 and different numbers of epochs to achieve optimal results. Basic augmentation techniques such as random cropping and rotation were applied. The models were initialized ImageNet weights and trained using RMSprop optimization. Increasing the number of layers in a network can potentially improve its generalization capability by optimizing more parameters. However, lightweight models are more preferred for smaller datasets, as they can avoid overfitting and reduce computational resource requirements. Nonetheless, the convolutional encoders pretrained on separate defect datasets without labels delivered favorable performance on the NEU metal surface defect dataset.

The results of the supervised approaches are summarized in Table 5. Models with varying complexities were selected for comparison with the self-supervised models. Zeeshan et al. utilized the VGG-19 architecture, which exhibited the highest number of trainable parameters among the models, leading to lower generalization ability. Qayyum et al. employed the InceptionV3 architecture for steel crack identification, demonstrating superior model adaptation and higher accuracy. Konovalenko et al. utilized the deep residual network ResNet152, which achieved moderate performance with over 86% accuracy compared to the other models.

Liu et al. and Singh et al. [28, 66] employed hybrid architectures for SSD. Although these architectures successfully extracted dominant features for other applications, they achieved approximately 70% accuracy for our dataset. This might be attributed to the insufficiency of labeled data and the similarity of extracted features among different surface defects.

In ref. [68], Smith et al. used a vision transformer-based encoder-decoder model called AnoViT proposed by Lee et al. [67]. for surface defect detection. It has 13.97 M trainable parameters. The model additionally can learn the global relationship between image patches that is capable of both image fault detection and localization. The AnoViT constructs a feature map maintaining the existing location information of individual patches by embedding all patches passed through multiple self-attention layers. For the NEU-CLS dataset, the model has exhibited an average 93% of accuracy, which is comparatively higher than other state-of-art models.

Table 5 Comparison of proposed model with the state-of-art deep learning models using the NEU-CLS dataset

Model	Architectures	Trainable parameters	Accuracy (%)	F1-score
Qayyum et al. [63]	InceptionV3	24 M	92.96	0.93
Konovalenko et al. [64]	ResNet152	60 M	86.29	0.86
Zeeshan et al. [65]	VGG19	138 M	61.85	0.55
Singh et al. [66]	ResNet101-SVM	44.5 M	70.23	0.70
Liu et. al. [28]	CNN-LSTM	1.2 M	67.5	0.68
Lee et. al [67]	AnoViT	13.97 M	93.0	0.93
Proposed	SimCLR-Skip-ConvNet	0.9 M	97.41	0.97

Conclusions

This study adopted a state-of-the-art self-supervised contrastive learning framework for defect detection on industrial metal surfaces. While the original framework employed ResNet-50 as the base encoder, our experiments employed lightweight convolutional encoders to address the challenges of limited training data on new and rare defect types.

However, the proposed framework presents a few concerns such as sensitivity to the opted augmentation techniques and the potential performance improvement with extremely large batch sizes, as discussed in this study. Nonetheless, the proposed model demonstrated high generalization capability with reasonable batch sizes and a relatively small dataset. In particular, self-supervised representation learning proves advantageous in the fine-tuning task, with SimCLR on Simple ConvNet achieving 97.78% accuracy and NNCLR on Skip-ConvNet achieving 97.04% accuracy. Thus, this study establishes that pretraining lightweight architectures using contrastive learning frameworks can produce near-gold standard results when fine-tuned for steel surface-defect classification. The limitations of this study include specific types of the considered defect, which may impact the generalization of the model to other scenarios. Thus, further research is required to explore its performance on a wider range of defect classes and real-world production environments. In future research, the model can be tested by implementing a conceptual model within an IoT framework in an industrial setting.

Acknowledgements
Not applicable.

Author contributions
MZ designed the model and the computational framework, MZ and AN wrote the manuscript with input from all authors, MK designed the figures, HC supervised and managed the fund, JU reviewed the manuscript.

Funding
This research was supported and funded by the Korean National Police Agency. [Project Name: XR Counter-Terrorism Education and Training Test Bed Establishment / Project Number: PR08-04-000-21]

Availability of data and materials
Three publicly available datasets [Severstal Steel Defect dataset (https://www.kaggle.com/competitions/severstal-steel-defect-detection/data), SD Saliency 900 (https://www.kaggle.com/datasets/alex000kim/sdsaliency900), and GC10-DET (https://www.kaggle.com/competitions/severstal-steel-defect-detection/data)] was used to pretrain the models. The NEU metal-surface defect dataset (https://www.kaggle.com/datasets/kaustubhdikshit/neu-surface-defect-database) was used to validate the model. The following repository contains code for results to be reproduced: https://github.com/MaheZ20Kaist/Contrastive-Learning-Metal-Surface.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that no conflicts of interest are associated with this publication.

Received: 9 December 2022 Accepted: 17 September 2023
Published online: 26 September 2023

References

1. Hao R, Lu B, Cheng Y, Li X, Huang B. A steel surface defect inspection approach towards smart industrial monitoring. J Intell Manuf. 2021;32(7):1833–43. https://doi.org/10.1007/s10845-020-01670-2.

2. Ning Z, Mi Z. Research on surface defect detection algorithm of strip steel based on improved YOLOV3. In Journal of Physics: Conference Series. IOP Publishing. 2021. Vol. 1907, No. 1, p. 012015. https://doi.org/10.1088/1742-6596/1907/1/012015
3. Schönbauer BM, Yanase K, Endo M. Influences of small defects on torsional fatigue limit of 17–4PH stainless steel. Int J Fatigue. 2017;100:540–8. https://doi.org/10.1016/j.ijfatigue.2016.12.021.
4. Zhang C, Wang Z, Liu B, Xiaolei W. Steel plate defect recognition of deep neural network recognition based on space-time constraints. Adv Multimedia. 2022. https://doi.org/10.1155/2022/9595286.
5. Ren Z, Fang F, Yan N, Wu Y. State of the art in defect detection based on machine vision. Int J Prec Eng Manufact Green Technol. 2021. https://doi.org/10.1007/s40684-021-00343-6.
6. Hauck Z, Rabta B, Reiner G. Impact of early inspection on the performance of production systems–insights from an EPQ model. Appl Math Model. 2022;107:670–87. https://doi.org/10.1016/j.apm.2022.03.003.
7. Saiz FA, Barandiaran I, Arbelaiz A, Graña M. Photometric stereo-based defect detection system for steel components manufacturing using a deep segmentation network. Sensors. 2022;22(3):882. https://doi.org/10.3390/s22030882.
8. Saeedi J, Dotta M, Galli A, Nasciuti A, Maradia U, Boccadoro M, Gambardella LM, Giusti A. Measurement and inspection of electrical discharge machined steel surfaces using deep neural networks. Mach Vision Appl. 2021;32(1):1–15. https://doi.org/10.1007/s00138-020-01142-w.
9. Guo Z, Wang C, Yang G, Huang Z, Li G. MSFT-YOLO: improved YOLOv5 based on transformer for detecting defects of steel surface. Sensors. 2022;22(9):3467. https://doi.org/10.3390/s22093467.
10. Akhyar F, Furqon EN, Lin CY. Enhancing precision with an ensemble generative adversarial network for steel surface defect detectors (EnsGAN-SDD). Sensors. 2022;22(11):4257. https://doi.org/10.3390/s22114257.
11. Benbarrad T, Eloutouate L, Arioua M, Elouaai F, Laanaoui MD. Impact of image compression on the performance of steel surface defect classification with a CNN. J Sens Actuator Netw. 2021;10(4):73. https://doi.org/10.3390/jsan10040073.
12. Yang L, Huang X, Ren Y, Zhang Y. Study on steel plate scratch detection based on improved MSR and phase consistency. Signal Image Video Process. 2022. https://doi.org/10.1007/s11760-022-02211-5.
13. Borselli A, Colla V, Vannucci M, Sant'Anna PCSS, Valdera PSA, Piaggio VR. Surface defects classification in steel products: A comparison between different artificial intelligence-based approaches. In Proceedings of the 11th IASTED International Conference on Artificial Intelligence and Applications, AIA 2011 2011. (pp. 2011–717). https://doi.org/10.2316/P.2011.717-068
14. Zheng H, Kong LX, Nahavandi S. Automatic inspection of metallic surface defects using genetic algorithms. J Mater Process Technol. 2002;125:427–33. https://doi.org/10.1016/S0924-0136(02)00294-7.
15. Hu H, Liu Y, Liu M, Nie L. Surface defect classification in large-scale strip steel image collection via hybrid chromosome genetic algorithm. Neurocomputing. 2016;181:86–95. https://doi.org/10.1016/j.neucom.2015.05.134.
16. Song K, Yan Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci. 2013;285:858–64. https://doi.org/10.1016/j.apsusc.2013.09.002.
17. Choi DC, Jeon YJ, Kim SH, Moon S, Yun JP, Kim SW. Detection of pinholes in steel slabs using Gabor filter combination and morphological features. ISIJ Int. 2017;57(6):1045–53. https://doi.org/10.2355/isijinternational.ISIJINT-2016-160.
18. Chaudhari CV. Steel surface defect detection using glcm, gabor wavelet, hog, and random forest classifier. Turkish J Comput Mat Educat. 2021;12(12):263–73.
19. Xue B, Wu Z. Key technologies of steel plate surface defect detection system based on artificial intelligence machine vision. Wirel Commun Mob Comput. 2021. https://doi.org/10.1155/2021/5553470.
20. Chen L, Yao X, Xu P, Moon SK, Bi G. Rapid surface defect identification for additive manufacturing with in-situ point cloud processing and machine learning. Virt Phy Proto. 2021;16(1):50–67. https://doi.org/10.1080/17452759.2020.1832695.
21. Božič J, Tabernik D, Skočaj D. End-to-end training of a two-stage neural network for defect detection. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 5619–5626). IEEE. 2021. https://doi.org/10.48550/arxiv.2007.07676
22. Zhang C, Wang Z, Liu B, Xiaolei W. Steel plate defect recognition of deep neural network recognition based on space-time constraints. Adv Multi. 2022. https://doi.org/10.1155/2022/9595286.
23. J Masci, U Meier, D Ciresan, J Schmidhuber, G Fricout. Steel defect classification with Max-Pooling Convolutional Neural Networks. The 2012 International Joint Conference on Neural Networks (IJCNN). 2012. pp. 1–6, https://doi.org/10.1109/IJCNN.2012.6252468.
24. Tian R, Jia M. DCC-CenterNet: A rapid detection method for steel surface defects. Measurement. 2022;187: 110211. https://doi.org/10.1016/j.measurement.2021.110211.
25. Zhang S, Zhang Q, Gu J, Su L, Li K, Pecht M. Visual inspection of steel surface defects based on domain adaptation and adaptive convolutional neural network. Mech Syst Sig Process. 2021;153: 107541. https://doi.org/10.1016/j.ymssp.2020.107541.
26. Chen Y, Ding Y, Zhao F, Zhang E, Wu Z, Shao L. Surface defect detection methods for industrial products: a review. Appl Sci. 2021;11(16):7657. https://doi.org/10.3390/app11167657.
27. Yang J, Fu G, Zhu W, Cao Y, Cao Y, Ying Yang M. A Deep Learning-based surface defect inspection system using multiscale and channel-compressed features. In IEEE Transactions on Instrumentation and Measurement, 2020. vol. 69, no. 10, pp. 8032–8042. https://doi.org/10.1109/TIM.2020.2986875.
28. Liu Y, Xu K, Xu J. Periodic surface defect detection in steel plates based on deep learning. Appl Sci. 2019;9(15):3127. https://doi.org/10.3390/app9153127.
29. Zhao W, Chen F, Huang H, Li D, Cheng W. A new steel defect detection algorithm based on deep learning. Comput Int Neurosci. 2021. https://doi.org/10.1155/2021/5592878.
30. Konovalenko I, Maruschak P, Brezinová J, Viňáš J, Brezina J. Steel surface defect classification using deep residual neural network. Metals. 2020;10(6):846. https://doi.org/10.3390/met10060846.
31. Hao Z, Wang Z, Bai D, Tao B, Tong X, Chen B. Intelligent detection of steel defects based on improved split attention networks. Front Bioeng Biotechnol. 2022;13(9):810876. https://doi.org/10.3389/fbioe.2021.810876.

32. Zhou Q, Wang H, Wang Y. Defect detection method based on knowledge distillation. IEEE Access. 2023. https://doi.org/10.1109/ACCESS.2023.3252910.
33. Anvar A, Cho YI. Automatic metallic surface defect detection using shuffledefectnet. J Korea Soc Comput Informat. 2020;25(3):19–26. https://doi.org/10.9708/JKSCI.2020.25.03.019.
34. Hao Z, Wang Z, Bai D, Tao B, Tong X, Chen B. Intelligent detection of steel defects based on improved split attention networks. Front Bioeng Biotechnol. 2021. https://doi.org/10.3389/fbioe.2021.810876.
35. Hao Z, Li Z, Ren F, Lv S, Ni H. Strip steel surface defects classification based on generative adversarial network and attention mechanism. Metals. 2022;12(2):311. https://doi.org/10.3390/met12020311.
36. Boikov A, Payor V, Savelev R, Kolesnikov A. Synthetic data generation for steel defect detection and classification using deep learning. Symmetry. 2021;13(7):1176. https://doi.org/10.3390/sym13071176.
37. Feng X, Gao X, Luo L. X-SDD: a new benchmark for hot rolled steel strip surface defects detection. Symmetry. 2021;13(4):706. https://doi.org/10.3390/sym13040706.
38. Zhang J, Kang X, Ni H, Ren F. Surface defect detection of steel strips based on classification priority YOLOv3-dense network. Ironmaking Steelmaking. 2021;48(5):547–58. https://doi.org/10.1080/03019233.2020.1816806.
39. Fan B, Li W. Application of GCB-Net based on Defect Detection Algorithm for Steel Plates. 2022. https://doi.org/10.21203/rs.3.rs-1550068/v1
40. Yang L, Huang X, Ren Y, Huang Y. steel plate surface defect detection based on dataset enhancement and lightweight convolution neural network. Machines. 2022;10(7):523. https://doi.org/10.3390/machines10070523.
41. Feng X, Gao X, Luo L. A ResNet50-based method for classifying surface defects in hot-rolled strip steel. Mathematics. 2021;9(19):2359. https://doi.org/10.3390/math9192359.
42. Wang S, Xia X, Ye L, Yang B. Automatic detection and classification of steel surface defect using deep convolutional neural networks. Metals. 2021;11(3):388. https://doi.org/10.3390/met11030388.
43. Wan X, Zhang X, Liu L. An improved VGG19 transfer learning strip steel surface defect recognition deep neural network based on few samples and imbalanced datasets. Appl Sci. 2021;11(6):2606. https://doi.org/10.3390/app11062606.
44. Feng X, Gao X, Luo L. An improved vision transformer-based method for classifying surface defects in hot-rolled strip steel. In Journal of Physics: Conference Series. IOP Publishing. 2021. Vol. 2082, No. 1, p. 012016. https://doi.org/10.1088/1742-6596/2082/1/012016
45. Cheng J, Guo B, Liu J, Liu S, Wu G, Sun Y, Yu Z. TL-SDD: A Transfer Learning-Based Method for Surface Defect Detection with Few Samples. In 2021 7th International Conference on Big Data Computing and Communications (BigCom). IEEE. 2021. pp. 136–143. https://doi.org/10.48550/arXiv.2108.06939
46. Damacharla P, Rao A, Ringenberg J, Javaid AY. TLU-net: a deep learning approach for automatic steel surface defect detection. In 2021 International Conference on Applied Artificial Intelligence (ICAPAI). IEEE. 2021. pp. 1–6. https://doi.org/10.48550/arXiv.2101.06915
47. Mayuravaani M, Manivannan S. A semi-supervised deep learning approach for the classification of steel surface defects," 2021 10th International Conference on Information and Automation for Sustainability (ICIAfS). 2021. pp. 179–184, https://doi.org/10.1109/ICIAfS52090.2021.9606143.
48. Zhu L, Baolin D, Xiaomeng Z, Shaoliang F, Zhen C, Junjie Z, Shumin C. Surface defect detection method based on improved semisupervised multitask generative adversarial network. Sci Program. 2022. https://doi.org/10.1155/2022/4481495.
49. Zhang J, Su H, Zou W, Gong X, Zhang Z, Shen F. CADN: a weakly supervised learning-based category-aware object detection network for surface defect detection. Patt Recog. 2021;109: 107571. https://doi.org/10.1016/j.patcog.2020.107571.
50. Fu G, Sun P, Zhu W, Yang J, Cao Y, Yang MY, Cao Y. A deep-learning-based approach for fast and robust steel surface defects classification. Opt Lasers Eng. 2019;121:397–405. https://doi.org/10.1016/j.optlaseng.2019.05.005.
51. Yang H, Zhu Z, Lin C, Hui W, Wang S, Zhao Y. Self-supervised surface defect localization via joint de-anomaly reconstruction and saliency-guided segmentation, in ieee transactions on instrumentation and measurement, 2023. vol. 72, pp. 1–10, Art no. 5014710, https://doi.org/10.1109/TIM.2023.3273681.
52. Zhao W, Song K, Wang Y, Liang S, Yan Y. FaNet: feature-aware network for few shot classification of strip steel surface defects. Measurement. 2023;208:112446. https://doi.org/10.1016/j.measurement.2023.112446.
53. Tian Y, Krishnan D, Isola P. Contrastive Multiview Coding. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, 2020. vol 12356. Springer, Cham. https://doi.org/10.1007/978-3-030-58621-8_45
54. Oord AV, Li Y, Vinyals O. Representation learning with contrastive predictive coding. arXiv preprint. 2018. arXiv:1807.03748.
55. He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. pp. 9729–9738. arXiv:1911.05722
56. Hadsell R, Chopra S, LeCun Y. Dimensionality Reduction by Learning an Invariant Mapping, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006. pp. 1735–1742, https://doi.org/10.1109/CVPR.2006.100.
57. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In International conference on machine learning. 2020. pp. 1597–1607. PMLR. arXiv:2002.05709
58. Dwibedi D, Aytar Y, Tompson J, Sermanet P, Zisserman A. With a little help from my friends: nearest-neighbor contrastive learning of visual representations. 2021. arXiv:2104.14548
59. Gutmann M, Hyvärinen A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, In Proceedings of Machine Learning Research. 2010: 9; 297–304 https://proceedings.mlr.press/v9/gutmann10a.html.
60. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

61. Zhang X, Zhou X, Lin M, Sun J. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. 2018 IEEE/CVF conference on computer vision and pattern recognition, 2018, pp. 6848–6856, https://doi.org/10.1109/CVPR.2018.00716.
62. Ma N, Zhang X, Zheng HT, Sun J. Shufflenet v2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European conference on computer vision (ECCV) 2018. pp. 116–131.
63. Qayyum W, Ehtisham R, Bahrami A, Camp C, Mir J, Ahmad A. Assessment of convolutional neural network pre-trained models for detection and orientation of cracks. Materials. 2023;16(2):826.
64. Konovalenko I, Maruschak P, Brevus V. Steel surface defect detection using an ensemble of deep residual neural networks. J Comput Inf Sci Eng. 2022;22(1): 014501.
65. Zeeshan M, Adnan SM, Ahmad W, Khan FZ. Structural crack detection and classification using deep convolutional neural network. Pakistan J Eng Technol. 2021;4(4):50–6.
66. Singh SA, Desai KA. Automated surface defect detection framework using machine vision and convolutional neural networks. J Intell Manuf. 2023;34(4):1995–2011.
67. Lee Y, Kang P. AnoViT: Unsupervised anomaly detection and localization with vision transformer-based encoder-decoder. IEEE Access. 2022;10:46717–24.
68. Smith AD, Du S, Kurien A. Vision transformers for anomaly detection and localisation in leather surface defect classification based on low-resolution images and a small dataset. Appl Sci. 2023;13(15):8716.
69. Hu C, Wang Y. An efficient convolutional neural network model based on object-level attention mechanism for casting defect detection on radiography images. IEEE Trans Industr Electron. 2020;67(12):10922–30.
70. Luo Q, Su J, Yang C, Gui W, Silven O, Liu L. CAT-EDNet: Cross-attention transformer-based encoder–decoder network for salient defect detection of strip steel surface. IEEE Trans Instrum Meas. 2022;71:1–13.
71. Hu C, Dong B, Shao H, Zhang J, Wang Y. Toward purifying defect feature for multilabel sewer defect classification. IEEE Trans Instrum Meas. 2023;72:1–11.

Publisher's Note