Unsupervised method for detecting surface defects in steel based on joint optimization of pseudo-labeling and clustering

Dongxu Bai, Gongfa Li, Du Jiang, Guozhang Jiang, Zhiqiang Hao, Dalin Zhou, Zhaojie Ju

Abstract

Advances in the field of measurement science and technology have improved the detection of defects in industrial production. One of the key challenges in steel plate surface defect detection is the need to quickly detect a small number of defects in an overwhelmingly defect-free sample. Unlike supervised learning, which relies heavily on precise sample labeling, unsupervised learning leverages its inherent learning capabilities for detection. This paper introduces an innovative method for smart steel diagnosis, integrating joint optimization of feature extraction and clustering. The proposed approach merges mini-batch K-Means clustering with a feature extraction network to acquire pseudo-label information for current images. It employs a multi-view transformation strategy, enabling classification through the optimized feedback from pseudo-labels. This method allows the network to self-optimize the distinction of image features through backpropagation. The method exhibits a mere 4% classification failure rate for steel surface images. This significant reduction in additional data processing requirements enhances the inspection system's efficiency and accuracy. Furthermore, the versatility of this method extends beyond steel defect diagnosis. It holds potential for application in various engineering domains, particularly in scenarios characterized by data imbalance.

Keywords: steel surface defect detection; unsupervised learning; feature extraction; clustering; image processing

1. Introduction

Big data and supervised learning techniques have been shown to surpass some of the perceptions and capabilities of traditional manual recognition. It is also widely used in scenarios such as image processing [1] and measurement science and technology [2]. However, labeling the necessary correlation data is quite expensive in supervised learning.

When inspecting many steel plate surface images, standard supervised learning methods often require costly fine labeling of defect samples. This can hinder the progress of intelligent defect detection in big data [3, 4].

In contrast, contrastive learning and clustering for feature representation learning can examine the data's distribution patterns and intrinsic features by leveraging the data's specificity. Good mining of the data itself through unsupervised learning to achieve better results. Methods such as clustering, semi-supervised and unsupervised learning have strong self-learning capabilities that reduce the dependence on sample labeling during training [5, 6].

In particular, unsupervised learning, which self-learns the intrinsic mapping relationships of the data, achieves good results. And it does not require any labeling. This approach has also achieved impressive results in various computer vision applications [7]. With the increasing sophistication of the steel production process, defective samples represent a minimal portion of the total. Efficient classification and identification of these defective samples can substantially decrease the burden on image data processing and computational resources [8]. It can also empower anomaly detection and smart manufacturing.

A common scenario in high-speed steel plate production lines involves a preponderance of defect-free images, with defective samples possibly constituting less than 1% of the total [9]. The challenge lies in swiftly categorizing these few defective samples in an imbalanced inspection setting. Exploring solutions to this challenge is crucial, and in this context, unsupervised learning emerges as a promising avenue for research.

This study explores joint clustering learning and feature extraction self-learning for defect detection. The key contributions of this paper are:

1) The inclusion of a dynamic optimization step in the clustering method.

2) Feedback optimization by clustering pseudo-labeled information.

3) Multi-view transformation strategy to optimize feature extraction.

4) A joint optimization method combining 2) and 3) for unsupervised learning is proposed.

The rest of this paper is organized as follows. The related work on contrastive learning and clustering for unsupervised feature learning is covered in Section II. Based on CNN feature extraction and clustering methods, a joint optimization classification model with unsupervised learning is constructed in Section III. In addition, a combined optimization strategy with multi-view transformation, cross-entropy loss and clustering pseudo-label is suggested. Section IV experiments on the proposed method for steel surface defect diagnostics, proving the method's stability and advantages. Section V presents the conclusion and prospect.

2. Related work

2.1. Contrastive learning

Leveraging a priori knowledge to create auxiliary tasks is crucial for surface image recognition, as it facilitates understanding the intricate information embedded within the data via these tasks. The essence of contrastive learning lies in its ability to minimize the distance between "similar" samples and maximize it for "non-similar" ones.

Moco [10] describes contrast learning as constructing discrete dictionaries on high-dimensional continuous inputs. Subsequently, Chen *et al* proposed MoCoV2 [11] and MoCoV3 [12] to improve data enhancement methods by adding a nonlinear layer to the encoder representation and applying cosine decay to the learning rate. SimCLR [13] demonstrates the great potential of contrast learning for semantic feature extraction, which outperforms supervised learning methods in the classification of ImageNet datasets.

SwAV [14] achieves data augmentation by mixing views of different resolutions and distinguishing clusters obtained by clustering. BYOL [15] removes the same part from the same batch of samples and retains the distinguished features. By not using negative samples, BYOL is more robust to different data enhancement methods. SimSiam [16] continues the idea of BYOL by further investigating twin networks and proposing a simple representation learning mechanism to avoid the Crash problem in representation learning. In addition, CAE [17], SimMIM [18], and PixMIM [19] constructed contrast learning methods through the potential features of the images, which achieved good self-learning capability.

In defect detection, the application of contrastive learning hinges on enabling the feature extraction network to self-improve through a feedback mechanism based on contrast. The incorporation of clustering methods as a means to facilitate this feedback has proven to be highly effective.

2.2. Clustering for unsupervised feature learning

Feature learning is the dimensional transformation of data in the original space so that it contains only the most essential information and the type to which it belongs [20, 21]. DeepCluster [22] jointly learned the neural network parameters and the clustering assignment of the resulting features. DeepCluster uses the K-Means clustering algorithm for iterative grouping of features and subsequent assignments are used as supervision to update the weights of the network. Cao *et al* [23] used two forms of adversarial-based produced data as well as raw data to train feature extractors to build effective latent representations. Cai *et al* [24] introduced focus loss into clustering in an unsupervised manner. Tan *et al* [25] proposed an iterative unsupervised learning representation model. The model utilizes a deep feature representation learning loss function and embedded fuzzy clustering with weighted adaptive entropy. Thereby, the deep clustering model is progressively improved.

In addition, clustering can be used to find feature information about intra-class tightness and inter-class dispersion to achieve a differentiated representation of the data in feature space. Srivastava *et al* [26] combined SMOTE algorithm and K-Means clustering algorithm sequentially to deal with the class imbalance problem and achieved good results. Zhang *et al* [27] developed a defect identification system based on principal component analysis (PCA) and histogram-based anomaly score (HBOS). The method extracts features using PCA and then fuses them using non-significance suppression. The approach requires a few unlabeled samples and has a low computing complexity. It also demonstrates that not only fast detection but also good results can be achieved by extraction and clustering methods.

Clustering discrimination is a useful interface problem for unsupervised representation learning that typically consists of two phases: clustering and discrimination. Clustering is the process of assigning a pseudo-label to each instance to learn the representation under discrimination [28]. Feature extraction by clustering learning and convolutional neural networks and their joint for label optimization is used to achieve good classification and detection of steel plate surface images.

3. Optimization of joint feature extraction and clustering

Learning high-quality features helps to improve the effectiveness of classification methods, while learning better features is oriented toward classification results [29]. The process is similar to supervised deep learning in that both feature learning tasks and subsequent tasks (classification, regression, etc.) are done simultaneously [30].

Based on this, the joint optimization method of feature extraction and clustering proposed in this research is shown in figure 1. Specifically, it includes data transformation, feature extraction, clustering, clustering pseudo-label information and classification loss joint optimization.

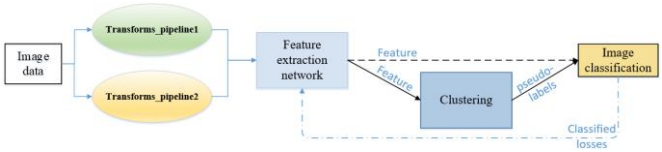

Figure 1. Flowchart of steel surface defect image classification method.

The data modification allows the feature extraction network to collect images with a variety of fields of view, improving resilience. The key idea behind the feature extraction link is to use a deep neural network with a strong feature extraction ability to extract as many information features from an image. To achieve image clustering differentiation, the clustering brings characteristics from the same category as near together as possible and features from different categories as far apart as possible. The feedback optimization session optimizes network performance by providing appropriate feedback for classification pseudo-labels and backpropagating model losses. This session enhances class distinction of image feature vectors to accomplish speedy and accurate categorization of data imbalance.

3.1. Feature extraction and clustering joint model

3.1.1 Feature extraction based on CNN. In recent years, the field of visual recognition has seen rapid modernization and performance improvements, driven by ever-improving architectures and better representation learning frameworks [31]. The current backbone network is almost the era of Transformers, with modern ConvNets represented by ResNet [32] and ConvNeXt [33] showing strong performance in a variety of scenarios. Although these models were originally designed for supervised learning using ImageNet tags, they may also benefit from self-supervised learning techniques.

ConvNeXt V2 [34], a new family of ConvNet models, covers a wider range of complexity. The structure of ConvNeXt V2 is shown in figure 2, and although there are few changes compared to the ConvNeXt architecture, through the introduction of Global Response Normalization (GRN), ConvNeXt V2 effectively solves the problems of feature collapse, feature cosine distance analysis, etc., and can preserve feature diversity in multi-scale feature layers.

ConvNeXt and ResNet have similar series structures, and both ConvNeXt and ConvNeXt V2 have structures such as Tiny, Base, Large, etc. In addition, ConvNeXt V2 has Atto, Femto, Pico, Nano and other small model structures. In this paper, the main focus is on using ConvNeXt V2-T(iny) as the backbone of the feature extraction module and using it for comparison experiments.

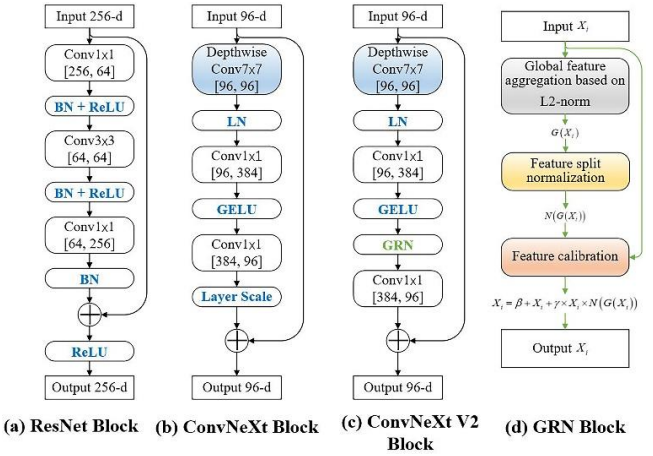

Figure 2. Comparison of ResNet, ConvNeXt, ConvNeXt V2 and GRN.

LayerScale multiplies the input feature layer by a trainable parameter (which is a vector) with the same number of elements as the feature layer channel, i.e., LayerScale scales the data of each channel. When GRN is applied, LayerScale

becomes unnecessary and can be removed. In figure 2, Depthwise conv represents the deep convolution using the inverse bottleneck structure, with [256, 64] representing the 256-dimensional input channel and 64-dimensional output channel.

The structure of GRN is shown in figure 2(d), which mainly consists of global feature aggregation based on L2-norm, feature spilt normalization and feature calibration. GRN uses L2 parametrization to aggregate the input features $X_i \in R^{H\times W\times C}$ into the global vector of $G(X_i)$:

$$G(X_i) = \|X_i\| \in R \tag{1}$$

where $\|X_i\|$ is the scalar that aggregates the feature information of the *i*-th channel. Next, using the response normalization function, the global features are normalized to obtain $N(G(X_i))$:

$$N(G(X_i)) = N(\|X_i\|) = \frac{\|X_i\|}{\sum_{j=1,\ldots,C}\|X_j\|} \in R \tag{2}$$

Feature normalization creates competition between features by comparing each *i*-th channel with other channels *j*. Finally, the features are calibrated to obtain the new X_i:

$$X_i = \beta + X_i + \gamma \times X_i \times N(G(X_i)) \tag{3}$$

GRN aims to achieve better results by increasing the contrast and selectivity of the channels. γ and β are two learnable parameters and are initialized to 0.

The excellent backbone network is a prerequisite for all feature extraction efforts. In this study, the ConvNeXt V2-T is proposed to be used as the main backbone of joint learning for feature learning.

3.1.2 Feature extraction and clustering joint model construction. The flowchart of the joint classification method based on feature extraction and clustering is shown in figure 3. First, the robustness of the model is improved by the fusion of multiple field-of-view transformations. The extraction of defective features is achieved using an improved feature self-learning network. After that, the joint optimization of feature learning and clustering is achieved through two branches.

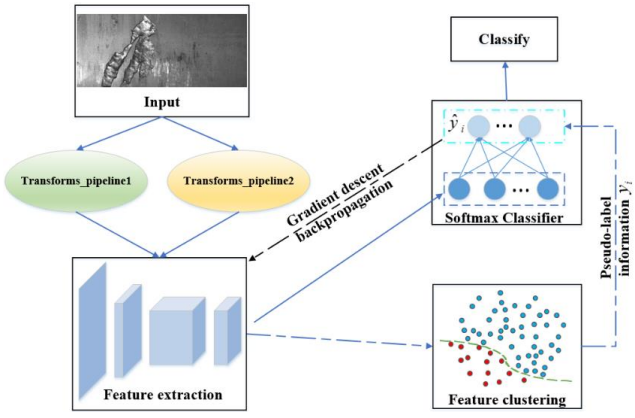

Figure 3. Joint feature extraction and clustering model.

The model is trained by minimizing the gap between the identified labels and their true labels, while the loss function is used as a measure of the label gap. The feature vector containing the clustered pseudo-label information is solved for the loss through the first branch, and the feature vector containing the classifier features is solved for through the second branch. The feature extraction network and Softmax classifier are trained and optimized in alternating iterations. The cross-entropy loss function is optimized for gradient backpropagation until the model loss converges or the termination condition is satisfied to achieve the classification of steel surface defects.

Their losses are back-propagated and passed to the feature extraction network for feedback optimization to update the weights of the network [35]. If the number of iterations of the model is not satisfied, the optimization is continued, and if the number of iterations is satisfied, the classification diagnostics result is output.

The first branch uses a feature extraction network to extract features from multi-view transformed images. The features are then clustered to generate pseudo labels. And output the clustering result containing the pseudo label information and pass it to the classifier. And the second branch uses the classifier directly for classification based on the extracted features. In this paper, Softmax is used as the activation function of the classifier to give the probability of different labels corresponding to the current input. Softmax is defined as follows

$$\hat{y}_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{4}$$

where $\hat{y}_i$ is the probability that the current input belongs to class *i*. For the whole convolutional neural network model, the model is trained by minimizing the gap between the recognized label $\hat{y}_i$ and its true label y_i, while the loss function is used as a measure of the label gap.

A feature vector with clustered pseudo-label information is obtained through the first branch, and a feature vector with classifier classification features is obtained through the other. The two perform a joint computation of the classification loss. This loss is back propagated to the feature extraction network for feedback optimization to update the weights and bias parameters of the network convolution. The convolutional operation can be represented by

$$F_{conv} = \omega_{conv} \cdot F_{prev} + b_{conv} \tag{5}$$

where F_{conv} and F_{prev} represents the output and input of convolution, ω_{conv} and b_{conv} represents the weight and bias parameters of convolution, respectively. To achieve good classification performance, the classification method based on a feature extraction network extracts image features

through the convolutional operation of a convolutional neural network, updates the weight and bias parameters of convolution, and updates the parameters using the pseudo-label.

The method proposed in this paper alternates between iterative optimization during training of the network, pseudo-label information generation, assignment, and classification loss supervised by using these pseudo-labels. The specific iteration phases are as follows: The feature extraction network optimizes the Softmax classifier once every *k* iteration, ensuring that the entire model has a steady optimization process. If *k* is too little, the model's features will soon cluster, resulting in a local optimum and rendering the classifier ineffective. If *k* is too large, the number of repetitions for feature clustering is insufficient to serve as a feature clustering process. Through experimental comparison and verification, the *k* is set to 10, i.e., for every ten iterations of the classifier, the feature extractor iterates once.

3.2. Joint optimization of clustering pseudo-label and cross-entropy loss

3.2.1 Different data transforms pipeline enhancement strategies.

This paper adopts a multi-crop method similar to SwAV, and different Transforms pipeline data enhancement is shown in figure 4. Transforms_pipeline 1 and 2 represent different data enhancement strategies, and View1 and View2 represent images after different random cropping. Transforms represent different data enhancement means, including random color dithering, random flipping, random grayscale transformation, random Gaussian filtering, etc.

Different random cropping views with various transform strategies motivate the network to learn the contextual information between the parts of the image. It enables ConvNets to learn more adequate and perfect features.

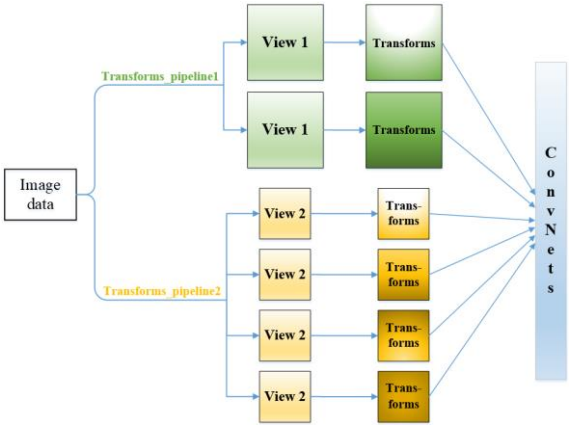

Figure 4. Different data transforms pipeline enhancement strategies.

The Transforms_pipeline1 and Transforms_pipeline2 of the example in figure 4 contain 2 views1 and 4 views2, respectively. transforms_pipeline1 and transforms_pipeline2 can contain different sub-views, taking into account the surface defects of the steel plate. Considering the factors of the data set, the random cropping size of View1 and View2 are set to 200 and 96, and the number of View1 and View2 are 2 and 6, respectively.

3.2.2 Joint optimization of clustering and cross-entropy loss.

In this paper, the feature space is divided into different sub-clustering spaces according to the classes to which the samples belong. The probability of a correct label corresponding to a feature is considered as its confidence level. By selecting the center through clustering algorithms, the distance between features and their corresponding centers is reduced so that similar features are brought closer to each other. This improves the distinguishability between the features, which in turn improves the effectiveness of the whole model in image classification [36].

Given a training set with N images: $X = \{x_1, x_2, \ldots, x_N\}$, one wants to find a parameter θ^* such that the mapping f_{θ^*} yields good generic features.

These parameters are traditionally learned by supervised learning, i.e., each image x_n is associated with a label y_n in $\{0,1\}^k$ which indicates that the image belongs to one of the *k* possible predefined classes. The parametric classifier g_W predicts the correct label over the features $f_\theta(x_n)$ and then learns the parameters *W* of the classifier and θ of the mapping jointly by optimizing the following problem.

$$\min_{\theta,W} \frac{1}{N} \sum_{n=1}^{N} \ell\left(g_W\left(f_\theta(x_n)\right), y_n\right) \tag{6}$$

The polynomial log-loss function ℓ uses small-batch stochastic gradient descent and backpropagation to compute gradient minimization.

When θ is sampled from a Gaussian distribution, f_θ does not produce good features without any learning. The convolutional structure has a strong prior on the input signal, and this weak prior is used to guide the discriminative power of the convolutional network, i.e., to add pseudo-label information. The output of the convolutional network is classified and the feature clustering information is used as its pseudo-label information to optimize equation (6).

The goal of the classification model is to get a good neural network $f_W(\cdot)$ as the feature extractor, which makes the transformed sample $\{z_i\}_{i=1}^n$ more suitable for classification, and this paper proposes to a use split attention network as the feature extractor of the classification model.

The feature extraction network maps the posterior point $z_i = f_W(\cdot)$ in a lower dimensional space than x_i. Therefore, the network can be used as a dimensionality

reduction method and is more powerful than other methods such as PCA and SVD. However, the network is not trained for a specific classification task, and for a dataset $X=\{x_i \in R^D\}_{i=1}^n$ with n samples, it needs to be fine-tuned using:

$$\min_{W,s} \frac{1}{n}\sum_{i=1}^{n} \| f_W(x_i) - \mathrm{M}s_i \|_2^2, \quad s.t.\ s_i \in \{0,1\}^K, 1^{\acute{u}} s_i = 1 \tag{7}$$

The feature extraction network maps each sample $x_i \in R^D$ to $z_i \in R^d$ · $s_i \in \{0,1\}^K$ represents the classification index assigned to the sample z_i (to discriminate whether it belongs to the class, and K is the number of classifications). $y_i = \mathrm{M}s_i$ represents the classification center of the class to which the sample z_i belongs. $\mathrm{M}=[m_1, m_2, \ldots, m_K] \in R^{d\times K}$ represents the matrix containing all classification centers; s_i represents the classification index of the *i*-th sample. $1^{\acute{u}}$ represents the column with 1 for each element of the vector is transposed. $\|\square\|_2^2$ represents the square of the 2-parametric number. s.t. is the abbreviation of subject to, which represents the objective function that needs to be constrained conditions.

In this study, data points are separated for better classification by alternately updating the classification center *M* and the assignment *s*. *M* is fixed in equation (8) after initialization to avoid degenerate solutions where all samples converge to a single point with zero objective. Meanwhile, the optimization of *W* and *s* can be performed alternatively. When s is fixed, equation (4) degenerates to:

$$\min_{W} \frac{1}{n}\sum_{i=1}^{n} \| f_W(x_i) - y_i \|_2^2 \tag{8}$$

From equation (8), it is clear that the target label of the sample x_i is its classification center y_i . $y_i = \mathrm{M}s_i$ is the center of the clustering to which the sample x_i belongs in the feature space. Therefore, the minimization of equation (8) forces the samples in the feature space to be dispersed in groups. Squeezing samples in the same classification does not guarantee that they will always be in the same category, so using a good clustering algorithm is key.

As an example, K-Means takes as input a set of vectors, this input in the classification is the features $f_\theta(x_n)$ generated by the convolutional network and classifies them into k different groups according to geometric criteria, and it jointly learns a d × k prime matrix C and a classification assignment y_n for each image n by solving the following problems.

$$\min_{C\in\square^{d\times k}} \frac{1}{N}\sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \| f_\theta(x_n) - Cy_n \|_2^2 \tag{9}$$

Equation (9) is subject to condition $y_n^{\acute{u}} 1_k = 1$. It provides a set of optimal assignments (y_n^*) and a center-of-mass matrix C^* with optimal assignments satisfying $n \le N$, which are then used as pseudo-label information. and the pseudo-label q is obtained by minimizing equation (7).

$$\min_{C\in R^{d\times k}} \frac{1}{N}\sum_{n=1}^{N} \min_{q} - z_n^{\acute{u}} C q \tag{10}$$

In the training phase, the columns of $z_n^{\acute{u}}$ and C are normalized, representing *K* features. The method minimizes the multinomial logical loss of the pseudo-label *q* in the classification problem.

$$\min_{z} \ell(z,c,q) = -\sum_{k} q^{(k)} \log p^{(k)}, \quad p^{(k)} = \frac{\exp\left(\frac{1}{\tau} z^{\acute{u}} c_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau} z^{\acute{u}} c_{k'}\right)} \tag{11}$$

The pseudo-label information is fixed during the training phase and updated once for the entire model in each pseudo-label generation allocation phase. The allocation phase's goal is to produce an allocation label *q* for each sample of the dataset, implying that the pseudo-label can represent a feature representation of the complete dataset *z*. If the allocation is modified at each epoch, the allocation phase accounts for one-third of the overall training time. To accelerate training, features computed in the previous epoch are chosen to be employed rather than being sent only to allocation.

To improve classification accuracy, the technique suggested in this study organically combines the allocation loss and training loss. Furthermore, the loss function in convolutional neural networks is an important parameter for determining model correctness and can provide evidence for the network model's high accuracy [37]. As the model's Loss value is decreased, the model's accuracy and robustness are both improved. The model loss function generally uses the Binary Cross Entropy (BCE) loss function, which is

$$L_{BCE}(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n} (y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)) \tag{12}$$

where *n* denotes the total number of samples in the training set, y_i denotes the true label of the *i*-th sample, and $\hat{y}_i$ denotes the model prediction of the *i*-th sample.

Sigmoid can convert linear inputs to nonlinear outputs and is also commonly used in binary classification problems, as well as in the activation function of neural networks. BCEWithLogitsLoss is a fusion improvement of BCELoss and Sigmoid, which can be expressed as

$$\ell(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n} \left(y_i \log \sigma(\hat{y}_i) + (1-y_i)\log\left(1-\sigma(\hat{y}_i)\right)\right) \tag{13}$$

$$\sigma(\hat{y}_i) = sigmoid(\hat{y}_i) = \frac{1}{1+\exp(-\hat{y}_i)} \tag{14}$$

BCEWithLogitsLoss is more stable than using BCELoss and Sigmoid alone in the calculation of the loss function [38]. Therefore, BCEWithLogitsLoss is used as the loss function of the steel surface defect classification model.

3.2.3 Learning rate optimization. The entire loss function of the defect detection model proposed in this paper is optimized by gradient backpropagation, while the gradient backpropagation of the entire CNN model is usually optimized by a gradient descent-based learning rate optimization algorithm.

There are numerous learning rate optimization algorithms based on gradient descent: Momentum, SGD (Stochastic Gradient Descent), SGDM (SGD with Momentum), RMSProp (Root Mean Square Propagation), NAG (Nesterov-accelerated gradient), Adam (Adaptive Moment Estimation), AdamW (Fixing weight decay regularization in Adam), RAdam (Rectified Adam), NAdam (Nesterov-accelerated Adaptive Moment Estimation), etc. The above algorithms are shown in figure 5.

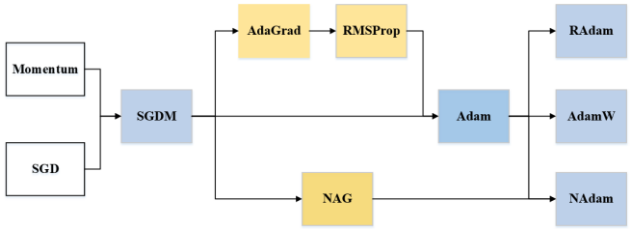

Figure 5. The evolution of the gradient descent-based learning rate optimization algorithm.

Radam [39] combines the advantages of Adam and SGD, with faster convergence without falling into local optima, and is insensitive to the learning rate of the initial values and has better stability. AdamW [40] is a regularization improvement of weight decay in ADAM, while NAdam [41] combines the advantages of Adam and NAG, which can make the optimization algorithm with better performance.

Therefore, RAdam, AdamW, and NAdam are proposed to be used for the experiments of the classification detection model in this paper. In addition, a strategy shared by LinearLR and CosineAnnealingLR is used for learning rate parameter optimization so that the learning rate can find the optimal value for gradient descent.

4. Classification experiments and results analysis

4.1. Classification experiment evaluation metrics

An unsupervised learning-based categorization algorithm should provide high similarity within the same category and low similarity between distinct categories as its optimal outcome [42].

Accuracy (ACC), Mutual Information (MI), and Normalized Mutual Information (NMI) are the primary evaluation criteria. NMI can be applied to any assignment of classification or true labels. If X and Y are independent, NMI is equal to 0. If one can accurately predict the other, NMI is equal to 1.

To comprehensively measure the effectiveness of the steel surface defect classification model, the evaluation indexes of Missing detection rate (MDR), False detection rate (FDR), and Effective Accuracy (EACC) are proposed by combining the confusion matrix.

The steel surface images without defects are set as positive class and the images with defects are set as negative class, and the confusion matrix of the image classification experiment is shown in table 1.

Table 1. Confusion matrix for image classification.

Predict \ Real	No defects	Defects
No defects	TP	FP
Defects	FN	TN

TP (True Positive) represents that the model predicts correctly, and the prediction and the real results are positive classes. FN (False Negative) represents that the model predicts incorrectly, and the prediction result is negative class, and the real is positive class. FP (False Positive) represents that the model predicts incorrectly, the prediction result is positive class, and the real is negative class. TN (True Negative) means the model predicts correctly, and the prediction and the real result are negative classes.

The accuracy rate is computed as the total number of correctly categorized samples without flaws divided by the total number of correctly classified samples with defects.

$$\mathrm{ACC} = \frac{TP + TN}{TP + FN + FP + TN} \tag{15}$$

The miss-detection rate is defined as the number of defective samples that are incorrectly identified as defect-free samples divided by the total number of samples, and is calculated as

$$\mathrm{MDR} = \frac{FP}{TP + FN + FP + TN} \tag{16}$$

False detection rate is defined as the number of non-defective samples wrongly identified as defective samples divided by the total number of samples, and is calculated as

$$\mathrm{FDR} = \frac{FN}{TP + FN + FP + TN} \tag{17}$$

In defect image classification, what cannot be tolerated is the misclassification of an otherwise defective image as a defect-free image, i.e., the lower the leakage rate, the better. In the subsequent defect recognition, the misclassified defective image will be recognized. Therefore, it can be assumed that this does not affect the accuracy of the classification. Therefore, the sum of the accuracy rate and the false detection rate is defined as the effective accuracy rate of

steel surface defect image classification, which is calculated as

$$\mathrm{EACC} = \frac{TP + TN + FN}{TP + FN + FP + TN} \tag{18}$$

Therefore, NMI, ACC, MDR, FDR and EACC are selected as the classification effect evaluation indexes of the experiment.

4.2. Experimental environment and hyperparameter settings

4.2.1 Experimental environment and dataset. The experiments are performed under the same conditions and are based on the Apex and Faiss with CUDA extension. The system environment of this experiment is Ubantu20.04, and the software environment is Python3.6.12, PyTorch1.10.1, CUDA10.2, cuDNN7.6.5, etc. The hardware configuration is: the CPU is Intel(R) Core(TM) i7-9700k @3.60GHz, the GPU is NVIDIA GeForce RTX2080 Ti, and the running memory (RAM) is DDR5 16GBx4.

The experimental mainly uses the open-source Severstal steel surface defect image dataset (https://www.kaggle.com/c/severstal-steel-defect-detection/), NEU surface defect dataset (http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html) and DAGM 2007 dataset (https://conferences.mpi-inf.mpg.de/dagm/2007/prizes.html).

Severstal dataset contains 12568 train and 1801 test data, and all images have vertical and horizontal resolutions of 256 and 1600, respectively. The dataset has 5 types of images: defect-free, pits, edge cracks, scratches, and iron oxide.

NEU dataset contains a total of 1800 images, and all images have a vertical and horizontal resolution of 200. NEU dataset has 6 types of defect images: crazing, inclusion, patches, pitted, rolled-in scale, and scratches.

4.2.2 Selection of feature clustering method and hyperparameter setting. The hyperparameter setting of the experiment is mainly to set the parameters of the clustering methods. The clustering effect of the four feature clustering methods, K-Means, Mini-Batch K-Means, BIRCH and spectral clustering, can be compared by a small ablation experiment, and the comparison results are shown in Fig. 6. The purpose of the ablation experiments with hyperparameter settings is to quickly screen out good clustering methods, optimizers, etc., which are not very relevant to the backbone network structure of feature extraction. Therefore, the simplest ResNet-34 is chosen as the experimental network. The bolded in the figure represent the optimal values. It can be seen that with the same feature extraction network, the method using Mini-Batch K-Means has the best effect. Therefore, this paper uses the feature clustering method of Mini-Batch K-Means for steel surface image classification.

Three main hyperparameters need to be set for the Mini-Batch K-Means feature clustering method.

1) n_clusters (K value, i.e., the number of categories to be classified), which is set according to the number of categories to be classified, and is set to 2 or 5 in this paper.

2) batch_size (the size of the sample set for running the Mini-Batch K-Means algorithm), comparing 64, 128, 256, and 512, respectively, to choose the best batch_size setting value.

3) init_size (set the number of samples that are candidates for the initial value of the center of mass, the value is usually set to 3 times the value of batch_size).

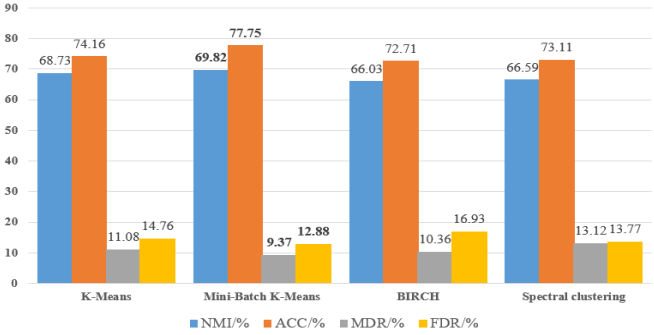

Figure 6. Comparison of feature clustering detection based on ResNet-34.

The Mini-Batch K-Means feature clustering method's batch size hyperparameters are compared and analyzed, and five different batch sizes, including 64, 128, 256 and 512, are used for comparison experiments. The experimental results are shown in figure 7. It can be seen that when the batch size is 256, the comprehensive effect is the best.

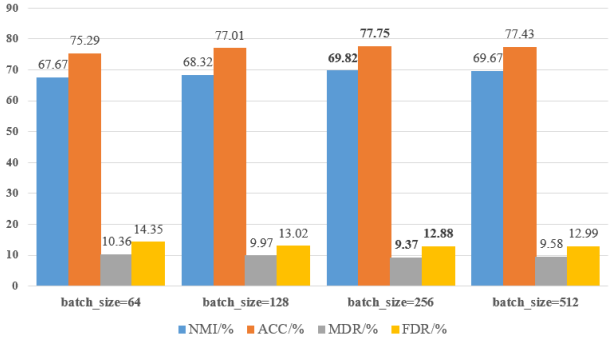

Figure 7. Comparison of batch_size of Mini-Batch K-Means.

The Mini-Batch K-Means feature clustering method was used to conduct comparison experiments with different gradient descent-based learning rate optimizers for Adam, AdamW, Radam and NAdam, with the same settings except for the different optimizers, and the results are shown in figure 8.

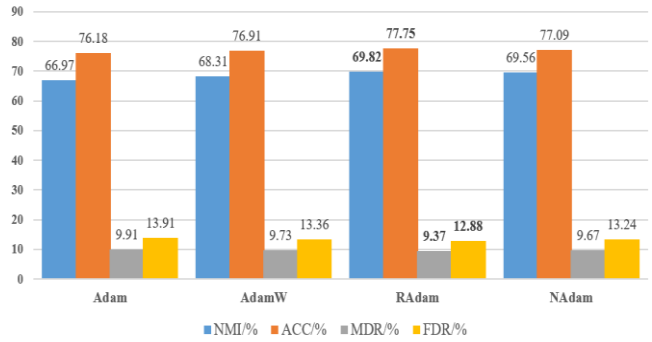

Figure 8. Effect of different optimizers on the classification model.

According to the experimental findings, RAdam performs optimization at its finest. The benefits of Adam and SGD are combined in RAdam, which also converges more quickly without hitting a local optimum, is less sensitive to the initial value learning rate, and has superior stability. Therefore, this paper uses the RAdam optimizer to optimize the model.

Through the above comparative experimental analysis, the feature clustering method of the joint optimization method of feature extraction and clustering in this paper selects the Mini-Batch K-Means clustering method, and its batch_size is set to 256, the reassignment_ratio value is set to 0.01, and the model is optimized by using the RAdam optimizer.

4.3. Steel surface image classification experiment

The experiments are divided into four parts.

(1) Selected convolutional neural networks of the ConvNets family are integrated with Mini-Batch K-Means clustering to validate the effectiveness of the proposed method in this paper.

(2) On the surface defect dataset (500 randomly selected images, 300 without defects and 200 with defects, of which there are 50 samples for each defect type), five comparative experiments with four 2-class clusters and one 5-class cluster using the method proposed in this paper are conducted to verify the stability and robustness of the proposed method.

(3) Classification experiments on the NEU dataset to verify the stability and robustness of the proposed method.

(4) State-of-the-art comparative experiments on the DAGM 2007 dataset are conducted to validate the advantages of the proposed method.

4.3.1 Comparative validity experiments. The processing comparison table for the dataset classified using the method in this paper and clustered into 2 classes (without defects and with defects) based on the Mini-Batch K-Means clustering method is shown in table 2. It compares different feature extraction methods: ResNet, ResNeXt [43], SE-ResNet [44], ResNeSt [45], ConvNeXt and ConvNeXt V2 networks, and the bolded values in the table represent the optimal values.

Table 2. Comparative experiments are clustered into 2 classes.

Networks	NMI/ %	ACC/ %	MDR/ %	FDR/ %	Test time/ms
ResNet-50	70.77	80.32	8.47	11.21	144.67
ResNeXt-50	71.94	82.06	7.33	10.61	147.45
SE-ResNet-50	73.21	84.16	5.98	9.86	152.44
ResNeSt-50	74.51	88.19	3.86	7.95	170.32
ConvNeXt-T	75.59	89.07	4.25	6.68	155.84
ConvNeXt V2-T	**76.61**	**90.04**	**3.94**	**6.02**	**166.95**

Table 3 displays a comparison for processing using the Mini-Batch K-Means clustering method, which is clustered into 5 classes. In the table, the EACC of the ConvNeXt V2-T network's steel surface defect detection is 94.97%, the NMI is 77.03%, the MDR is 5.03%, and the FDR is 6.01%.

Combined with the aforementioned data analysis, the ConvNeXt V2-T network is selected as the feature extractor for the method.

Table 3. Comparative experiments are clustered into 5 classes.

Networks	NMI/ %	ACC/ %	MDR/ %	FDR/ %	Test time/ms
ResNet-50	70.16	80.11	8.11	11.78	153.25
ResNeXt-50	71.63	81.33	7.69	10.98	154.98
SE-ResNet-50	73.11	82.97	6.97	10.06	161.76
ResNeSt-50	74.76	85.32	5.23	9.45	179.43
ConvNeXt-T	75.87	87.17	5.61	7.22	160.34
ConvNeXt V2-T	**77.03**	**88.96**	**5.03**	**6.01**	**169.75**

Through experimental analysis and comparison, the EACC of steel surface defect detection using the method clustered into 2 classes and clustered into 5 classes reaches 96.06% and 94.97% respectively, which verifies the effectiveness of the method.

4.3.2 Stability and Robustness Comparison Experiment. In order to verify the stability and robustness of the proposed method, 300 defect-free images and 200 single-class defective images in the dataset are randomly selected from Experiment I to IV, and 300 defect-free images and 50 defective images in the dataset are selected as Experiment V at the same time. The above experiments were done 10 times randomly and compared the effect of clustering the proposed method into 2 and 5 classes respectively. The average experimental results are shown in figure 9.

When clustered into 2 classes, the EACC for image classification of pits, edge cracks, scratches, and iron oxide defects was 92.4%, 84.8%, 95.8% and 95.4%, respectively. The EACC of experiment V is 94.4%, which further proves the robustness and stability of the proposed method.

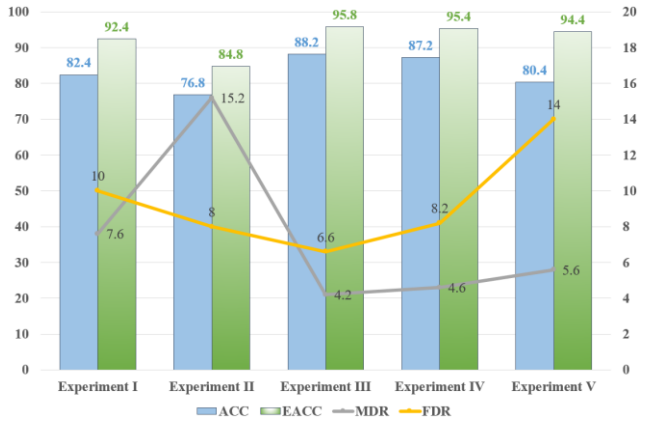

Figure 9. Data analysis of five comparison experiments.

Due to the small number of edge cracks and the fact that they will be cut in the subsequent steel production process, they have little impact on the overall detection results. Therefore, the method is still superior in image classification with data imbalance.

These five comparative experiments prove that the overall ACC of the proposed method is effective and stable. The experimental results show that the method in this paper can effectively classify steel surface defects, which not only has high classification efficiency of data imbalance but also can effectively reduce the FDR and MDR of surface defect images.

4.3.3 NEU comparison experiment. In addition to the experiments on the Severstal dataset, this paper also conducts comparative experiments to verify the robustness of the method on the NEU dataset, which has 1800 images in 6 classes. In this paper, the dataset is divided into 1500 training sets (250 per class) and 300 validation sets (50 per class).

To ensure the robustness and reliability of the data, the trained model was tested 10 times using 300 validation set images, and the test results after summing the 10 times results are shown in table 4.

Table 4. Confusion matrix for NEU dataset.

Predicted \ Real	Real	Crazing 500	Inclusion 500	Patches 500	Pitted 500	Rolled 500	Scratches 500
Crazing	498	**423**	14	9	20	17	15
Inclusion	495	11	**445**	5	8	6	20
Patches	501	16	6	**457**	7	9	6
Pitted	505	21	10	12	**428**	24	10
Rolled	503	18	8	10	25	**433**	9
Scratches	498	11	17	7	12	11	**440**

The overall classification accuracy was 87.53%, and the MDR was 5.97% when tested on the NEU dataset using the proposed method. Crazing, Pitted and Rolled were below the overall average ACC of 87.53%, while Inclusion, Patches and Scratches were above the average ACC. Crazing was the lowest at 84.6% and Patches was the highest at 91.4%, as shown in figure 10.

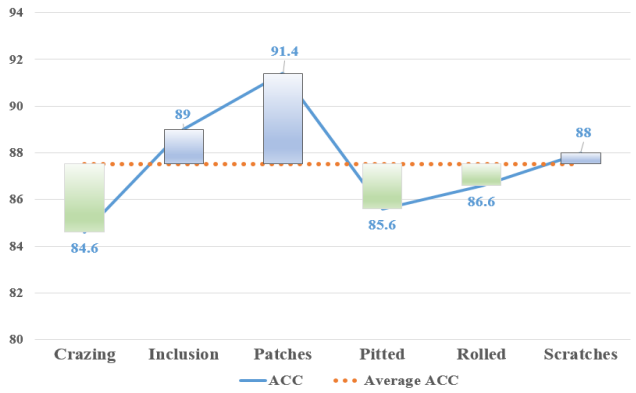

Figure 10. Accuracy of NEU dataset comparison for each category.

Typically, NEU datasets have defective feature diversity, and the classification accuracy of some categories will be lower than the overall average accuracy. Data enhancement strategies and corresponding loss functions for data under sample imbalance are also worthy of subsequent research work.

4.3.4 Advanced comparison experiment. To validate the effectiveness and sophistication of the proposed method, comparative experiments were conducted on the DAGM 2007 dataset using the proposed method. The comparison results of different methods are shown in table 5.

PatchCore [46] uses the most representative repository of nominal patch features to enable detection. DRGS-Net (Dual Path Reconstruction Guided Segmentation Network) [47] is made up of a sub-network that uses dual path reconstruction and anomaly segmentation. The former generates fine-grained reconstruction findings, and the latter uses the results to determine the decision border between normal and abnormal regions. CSKD (Cosine Similarity Knowledge Distillation) [48] reduces the impact of the teacher-student model on response similarity in anomalous regions by distilling losses. FCGAN (Fuzzy Clustering based Generative Adversarial Network) [49] implements a unique method for imbalance defect diagnosis. It improves algorithmic performance by including fuzzy clustering into the discriminator's unsupervised learning training process.

Table 5. Comparison of different methods.

Methods	PatchCore	HBOS	DRGS-Net	CSKD	FCGAN	**Ours**
Ref.	[46]	[27]	[47]	[48]	[49]	-
Years	2022	2023	2023	2024	2024	**2024**
ACC	92.1	93.8	95.2	94.8	95.21	**95.74**

Although our method improves less than the former method, the GAN network used in the former is very time-consuming. The method in this paper achieves anomaly detection only through the simplest joint feature extraction network and clustering method. The effectiveness, advancement, stability and robustness of the proposed method are verified through four parts of experiments.

When the steel surface images are clustered into 2 classes by the method in this paper, the MDR of the images is only 4.03%, which meets the requirements of the industrial field. The method also provides help for the applied intelligence of industrial intelligent manufacturing.

5. Conclusion

Steel surface image classification model with joint optimization of feature extraction and clustering is constructed based on unsupervised learning. The model is based on ConvNeXt V2 and the Mini-Batch K-Means clustering method. Combined with the Softmax classifier to obtain the pseudo-label information of the current image, the network performance is jointly optimized by the feedback of the classified pseudo-labels and the back-propagation of the model loss. In turn, category distinction of different image feature vectors is improved, and fast and accurate classification of steel surface images is achieved. The experimental results verify the effectiveness, accuracy and advancement of the proposed method. The method greatly reduces the amount of data processing required for subsequent defect image identification in the inspection system. It also saves computational resources by quickly classifying and identifying a large number of steel surface images. This research is also expected to be applied to the advanced manufacturing of other industrial products.

Acknowledgements

This work was supported by the grants of the National Natural Science Foundation of China (Grant Nos. 51575407, 51505349); "The 14th Five Year Plan" Hubei Provincial advantaged characteristic disciplines (groups) project of Wuhan University of Science and Technology (2023C0401); the China Scholarship Council (No. 202308420243).

ORCID iDs

Dongxu Bai https://orcid.org/0000-0002-1137-2737
Gongfa Li https://orcid.org/0000-0002-2695-2742
Dalin Zhou https://orcid.org/0000-0003-2363-9125
Zhaojie Ju https://orcid.org/0000-0002-9524-7609

References

[1] Tang, J., Wang, Z., Zhang, H., Li, H., Wu, P., & Zeng, N. (2024). A lightweight surface defect detection framework combined with dual-domain attention mechanism. *Expert Systems with Applications*, **238**, 121726.

[2] Liu, M., Kong, X., Luo, J., & Yang, L. (2024). Incipient fault detection based on ensemble learning and distribution dissimilarity analysis in multi-feature processes. *Measurement Science and Technology*, **35(4)**, 045905.

[3] Singh, S. A., & Desai, K. A. (2023). Automated surface defect detection framework using machine vision and convolutional neural networks. *Journal of Intelligent Manufacturing*, **34(4)**, 1995-2011.

[4] Leberruyer, N., Bruch, J., Ahlskog, M., & Afshar, S. (2023). Toward zero defect manufacturing with the support of artificial intelligence—Insights from an industrial application. *Computers in Industry*, **147**, 103877.

[5] Li, D., Zhang, Z., Wen, G. (2023). Classifier subset selection based on classifier representation and clustering ensemble. *Applied Intelligence*, **53**, 20730–20752.

[6] Mai, H. T., Lieu, Q. X., Kang, J., & Lee, J. (2023). A novel deep unsupervised learning-based framework for optimization of truss structures. *Engineering with Computers*, **39(4)**, 2585-2608.

[7] Xu, X., Du, J., Song, J., Xue, Z., Li, A., & Guan, Z. (2023). Cluster-aware multiplex InfoMax for unsupervised graph representation learning. *Neurocomputing*, **532**, 94-105.

[8] Mordia, R., & Verma, A. K. (2022). Visual techniques for defects detection in steel products: A comparative study. *Engineering Failure Analysis*, **134**, 106047. https://doi.org/10.1016/j.engfailanal.2022.106047

[9] Bai, D., Li, G., Jiang, D., Yun, J., Tao, B., Jiang, G., et al. (2024). Surface defect detection methods for industrial products with imbalanced samples: A review of progress in the 2020s. *Engineering Applications of Artificial Intelligence*, **130**, 107697.

[10] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729-9738.

[11] Chen, X., Fan, H., Girshick, R., He, K. (2020). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv*:2003.04297.

[12] Chen, X., Xie, S., He, K. (2021). An empirical study of training self-supervised vision transformers. *IEEE/CVF International Conference on Computer Vision*, 9640-9649.

[13] Chen, T., Kornblith, S., Norouzi, M., Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning*, **119**, 1597-1607.

[14] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, **33**, 9912-9924.

[15] Grill, J. B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., ... & Valko, M. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, **33**, 21271-21284.

[16] Chen, X., He, K. (2021). Exploring simple siamese representation learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15750-15758. https://doi.org/10.1109/ CVPR46437.2021.01549

[17] Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., ... & Wang, J. (2023). Context autoencoder for self-supervised

representation learning. *International Journal of Computer Vision*, 1-16. https://doi.org/10.1007/s11263-023-01852-4

[18] Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., ... & Hu, H. (2022). Simmim: A simple framework for masked image modeling. I*n Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9653-9663. https://doi.org/10.1109/CVPR52688.2022.00943

[19] Liu, Y., Zhang, S., Chen, J., Chen, K., & Lin, D. (2023). PixMIM: Rethinking Pixel Reconstruction in Masked Image Modeling. *arXiv preprint arXiv*:2303.02416. https://doi.org/10.48550/arXiv.2303.02416

[20] Zhao, C., Fan, Y., Tan, J., Li, Q., Lin, Z., Luo, S., et al. (2024). FCS-YOLO: An efficient algorithm for detecting steel surface defects. *Measurement Science and Technology*, **35(8)**, 086004.

[21] Yun, J., Jiang, D., Huang, L., Tao, B., Liao, S., Liu, Y., et al. (2024). Grasping detection of dual manipulators based on Markov decision process with neural network. *Neural Networks*, **169**, 778-792.

[22] Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. *European Conference on Computer Vision*, **11218**, 139–156.

[23] Cao, W., Zhang, Z., Liu, C., Li, R., Jiao, Q., Yu, Z., & Wong, H. S. (2022). Unsupervised discriminative feature learning via finding a clustering-friendly embedding space. *Pattern Recognition*, **129**, 108768.

[24] Cai, J., Wang, S., Xu, C., & Guo, W. (2022). Unsupervised deep clustering via contractive feature representation and focal loss. *Pattern Recognition*, **123**, 108386.

[25] Tan, D., Huang, Z., Peng, X., Zhong, W., & Mahalec, V. (2023). Deep adaptive fuzzy clustering for evolutionary unsupervised representation learning. *IEEE Transactions on Neural Networks and Learning Systems*.

[26] Srivastava, R. & Jain, A. (2022). Feature clustering and ensemble learning based approach for software defect prediction. *Recent Advances in Computer Science and Communications*, **15(6)**, 868-882.

[27] Zhang, N., Zhong, Y., & Dian, S. (2023). Rethinking unsupervised texture defect detection using PCA. *Optics and Lasers in Engineering*, **163**, 107470.

[28] Qian, Q., Xu, Y., Hu, J., Li, H., & Jin, R. (2022). Unsupervised visual representation learning by online constrained k-means. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16640-16649.

[29] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, **33**, 9912-9924.

[30] Guo, X., Liu, X., Zhu, E., Zhu, X., Li, M., Xu, X., & Yin, J. (2019). Adaptive self-paced deep clustering with data augmentation. *IEEE Transactions on Knowledge and Data Engineering*, **32(9)**, 1680-1693.

[31] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *In Proceedings of the IEEE/CVF international conference on computer vision*, 10012-10022.

[32] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.

[33] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11976-11986.

[34] Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., & Xie, S. (2023). ConvNeXt V2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv*:2301.00808.

[35] Bai, D., Li, G., Jiang, D., Tao, B., Yun, J., Hao, Z., et al. (2024). Depth feature fusion based surface defect region identification method for steel plate manufacturing. *Computers and Electrical Engineering*, **116**, 109166.

[36] Cao, W., Zhang, Z., Liu, C., Li, R., Jiao, Q., Yu, Z., & Wong, H. S. (2022). Unsupervised discriminative feature learning via finding a clustering-friendly embedding space. *Pattern Recognition*, **129**, 108768.

[37] Li, F., Qiao, H., & Zhang, B. (2018). Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognition*, **83**, 161-173.

[38] Qu, H., Zheng, J., & Tang, X. (2022). Effects of loss function and data sparsity on smooth manifold extraction with deep model. *Expert Systems with Applications*, **198**, 116851.

[39] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2019). On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv*:1908.03265.

[40] Loshchilov, I., & Hutter, F. (2019) Fixing weight decay regularization in Adam. *International Conference on Learning Representation*.

[41] Nguyen, T., Baraniuk, R., Bertozzi, A., Osher, S., & Wang, B. (2020). Momentumrnn: Integrating momentum into recurrent neural networks. *Advances in Neural Information Processing Systems*, **33**, 1924-1936.

[42] Sun, Y., Liu, X., Zhai, X., Sun, K., Zhao, M., Chang, Y., et al. (2023). Automatic pixel-level detection of tire defects based on a lightweight Transformer architecture. *Measurement Science and Technology*, **34(8)**, 085405.

[43] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 5987-5995.

[44] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 7132-7141.

[45] Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., ... & Smola, A. (2022). Resnest: Split-attention networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 2736-2746.

[46] Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., & Gehler, P. (2022). Towards total recall in industrial anomaly detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 14318-14328.

[47] Xiao, J., Deng, L., Chen, Z., Li, X., Chen, B., & Yin, H. (2023). Dual-path reconstruction guided segmentation network for unsupervised anomaly detection and localization. *International Joint Conference on Neural Networks*, 1-8.

[48] Sheng, S., Jing, J., Wang, Z., & Zhang, H. (2024). Cosine similarity knowledge distillation for surface anomaly detection. Scientific Reports, 14(1), 8150.

[49] Wang, Y., & Xue, Q. (2024). Fault identification of product design using fuzzy clustering generative adversarial network (FCGAN) model. Soft Computing, 28(4), 3725-3742.