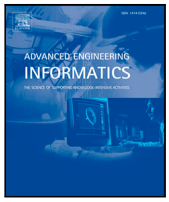
Advanced Engineering Informatics 56 (2023) 101933

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Full length article

# Selecting informative data for defect segmentation from imbalanced datasets via active learning

Weifeng Li [a], Bin Li [a,b,*], Shuanlong Niu [a], Zhenrong Wang [a], Baohui Liu [a], Tongzhi Niu [a]

[a] School of Mechanical Science and Engineering, HUST, 1037 Luoyu Road, Wuhan, 430074, Hubei, China
[b] Wuhan Intelligent Equipment Industrial Institute Co., Ltd., 8 Ligou South Road, Wuhan, 430074, Hubei, China

ARTICLE INFO

*Keywords:*
Deep learning
Neural networks
Active learning
Cold start
Defect segmentation

ABSTRACT

The storage and labeling of industrial data incur significant costs during the development of defect detection algorithms. Active learning solves these problems by selecting the most informative data among the given unlabeled data. The existing active learning methods for image segmentation focus on studying natural images and medical images, with less attention given to industrial images, and little research has been performed on imbalanced data. To solve these problems, we propose an active learning framework to selecting informative data for defect segmentation under imbalanced data. In the initialization stage, the framework uses self-supervised learning to initialize the data so that the initialization data contain more defect data, thereby solving the cold-start problem. During the iterative stage, we design the main body of the active learning framework, which is composed of a segmentation learner and a reconstruction learner. These learners use supervised learning to further improve the framework's ability to select informative data. The experimental results obtained on public and self-owned datasets show that the framework can save 70% of the required storage space and greatly reduce the cost of labeling. The intersection over union value proves that the designed framework can achieve the equivalent effect of labeling the whole dataset by labeling partial data.

## 1. Introduction

Defect segmentation plays a vital role in quality control in industrial production. In recent years, neural networks have been widely used in defect detection [1–3]. However, due to the wear of cutting tools or grinding tools in the production line, the characteristics of new data change, which makes it difficult for an existing model to adapt to the new data. Labeling all new data and using it to update existing models is a tedious and time-consuming process. In particular, due to the control of industrial product yields, the distributions of industrial data is imbalanced; that is, the number of information-rich defect samples is far less than the number of normal samples, as shown in Fig. 1, which indicates that labeling all data is not necessary. Therefore, how to select the most informative data from an imbalanced dataset for storage and labeling is critical to reducing costs and improving product quality.

By selecting the most informative data from the entire data for labeling, better performance can be achieved under a limited labeling budget; this strategy belongs to the research content of active learning. In the existing active learning methods, these data are selected based on the distribution of the data or the uncertainty of the predicted result. [4–6] et al. focused on active learning methods for natural image classification problems. [7,8] proposed a task-independent active learning method and claimed that it can be applied to image segmentation tasks. [9,10] designed active learning methods for medical image segmentation and achieved excellent results. However, due to the imbalanced and highly similar characteristics of industrial data, these methods face many challenges when selecting the most informative data. Among them, distribution-based methods aim to select the data that cover the feature distribution space so that the selected data are still imbalanced data. Uncertainty-based methods are affected by the initialization data. If the initialization data do not contain defect data, such an approach is not able to select information-rich defect data.

Among the previously developed active learning methods for images, only pretext tasks for active learning (PT4AL) [7] utilizes data initialization; the remaining studies adopted the method of random initialization [7], that is, the cold-start approach. Random initialization means that the initialization data may not contain defect data. PT4AL solves the cold-start problem in natural image classification tasks by rotation [11], the jigsaw_puzzle strategy [12] and other self-supervision methods [13,14]. However, when applied to industrial image segmentation tasks, these self-supervision methods tend to learn background

* Corresponding author at: School of Mechanical Science and Engineering, HUST, 1037 Luoyu Road, Wuhan, 430074, Hubei, China.
*E-mail addresses:* li_weifeng@hust.edu.cn (W. Li), libin999@hust.edu.cn (B. Li), slniu@hust.edu.cn (S. Niu), zora_wang@hust.edu.cn (Z. Wang), lbhui@hust.edu.cn (B. Liu), tzniu@hust.edu.cn (T. Niu).

https://doi.org/10.1016/j.aei.2023.101933
Received 19 November 2022; Received in revised form 2 February 2023; Accepted 4 March 2023
Available online 29 April 2023
1474-0346/

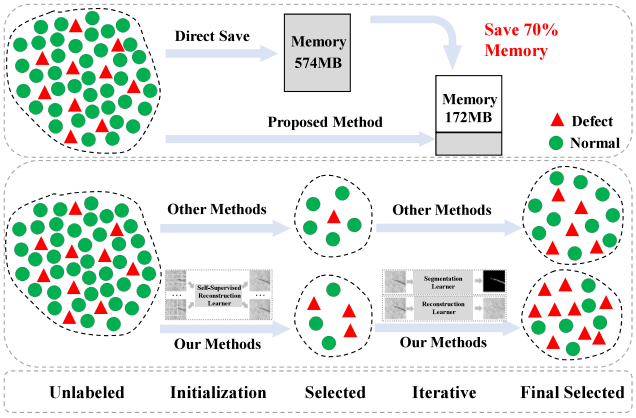



**Fig. 1.** We propose a new active learning framework for industrial scenarios to reduce the incurred storage and labeling costs. In the initialization stage, more informative data (defect data) can be selected by using the proposed method than by utilizing other methods. After several iterations, when 30% of the total data are selected, the data selected by combined segmentation and reconstruction method proposed in this paper contain almost all defect data, which contrasts with the data selected by other methods. By using the proposed active learning method, 70% of the storage and labeling costs can be saved on the test dataset.

information rather than defects because there are far more normal images than defective images and because the defective region in a single image is usually smaller than the normal region. As a result, these self-supervision techniques are irrelevant to the image segmentation task, as shown in Figs. 2(a),(b), and both normal images and defective images have small loss values. Therefore, selecting informative data according to these self-supervised methods is not suitable for solving the cold-start problem presented by imbalanced industrial image data. Applying active learning to select the most informative data for imbalanced image segmentation in the industrial field still faces the data imbalance problem and the cold-start problem caused by data imbalances, which urgently need to be solved.

To solve these problems, we propose an active learning framework implemented by a combination of self-supervised learning and supervised learning, which includes an initialization stage and an iteration stage. We find that for the task of segmenting defects in imbalanced industrial images, the segmentation losses of defective images are generally greater than those of normal images, as shown in Fig. 2. This proves that in the defect segmentation task involving imbalanced data, defective images contain more information than normal images. Therefore, we hope to acquire more defect data in active learning tasks. Specifically, during the initialization stage, to solve the cold-start problem in active learning, we design a self-supervised reconstruction learner with predictive preferences based on the characteristics of data imbalances. The learner tends to reconstruct defective images as normal images, while the normal images remain unchanged. When selecting informative data, we preferentially select data with large reconstruction errors for manual annotation by evaluating the reconstruction differences between the input images and the reconstructed images. Fig. 2(c) shows that the larger the reconstruction error is, the more likely it is to be a defective image, but when the reconstruction error is small, it is difficult to distinguish a defective image from a normal image. To address this problem, in the iterative stage, we design an image segmentation learner and a new reconstruction learner to further enhance the framework's ability to select defect data. The new reconstruction learner and segmentation learner are trained using the data obtained in the initialization stage. This new reconstruction learner has the same prediction preferences as the reconstruction learner in the initialization stage. The segmentation learner predicts suspicious defect regions from unlabeled data. When selecting informative data, we also preferentially select data with large reconstruction errors for manual annotation but only consider the reconstruction error of the suspected defect region. Fig. 2(d) shows the relationship between the suspected defect region reconstruction error and the segmentation loss in the iterative stage. Similar to Fig. 2(c), the larger the reconstruction error of the suspected defect region is, the higher the probability that the image is a defect. However, the reconstruction error in the iterative stage further magnifies the differences between normal and defective images, making it simpler to select more defective data from the unlabeled data. Finally, the image segmentation learner and the reconstruction learner iterate in a loop to label all data. Experimental results show that the proposed method can save storage space, reduce the cost of labeling by approximately 70%, and greatly reduce the required manpower and material resources.

Briefly, the contributions of our method can be summarized as follows.

- We design a new active learning framework for industrial image segmentation, which can effectively reduce the costs of storage and annotation. In particular, the utilized framework is strongly related to the segmentation task. In the testing dataset, the effect of labeling all of the data can be achieved by labeling part of the data.
- Based on the characteristics of data imbalances, we design a self-supervised reconstruction learner with predictive preferences, which tends to predict defective images as normal images while normal images remain unchanged. On this basis, a maximum sequential mask reconstruction error is designed to evaluate the difference between the input image and the reconstructed image, and the larger the error is, the more likely it is to be a defective image. By preferentially selecting the data corresponding to the largest error values, the cold-start problem in active learning under data imbalance settings is solved.
- Based on the characteristics of high image similarity, we use supervised learning to build a new reconstruction learner with predictive preferences. On this basis, a suspicious defect region reconstruction error is designed to evaluate the difference between the input image and the reconstructed image, which amplifies the difference between the defect image and the normal image. This further enhances the framework's ability to select information-rich data.

The remainder of this article is organized as follows. Section 2 introduces the related work on active learning in image classification and segmentation. Next, the specific structure of the proposed method for selecting informative data for defect segmentation (SID4DS) is described in detail in Section 3. Section 4 provides the experimental data and parameter settings. In Section 5, we describe the experiments designed to verify the effectiveness of the proposed framework and each learner. Finally, Section 6 concludes this article.

## 2. Related work

Image-related research contents in active learning applications include image classification and image segmentation. This section reviews typical applications of active learning in these domains.

### 2.1. Image classification

In image classification, Senser et al. [4]. proposed an active learning method based on core set selection, which utilizes batch processing to select data for labeling to better adapt to an active learning method based on a convolutional neural network. Additionally, Senser et al. [4]. used methods such as K-means to select data with high uncertainty and selected data covering the entire feature space by evaluating the uncertainty of the input data. Wang et al. [5]. not only considered the most uncertain and representative data but also selected unlabeled data with high confidence for pseudolabeling to make full

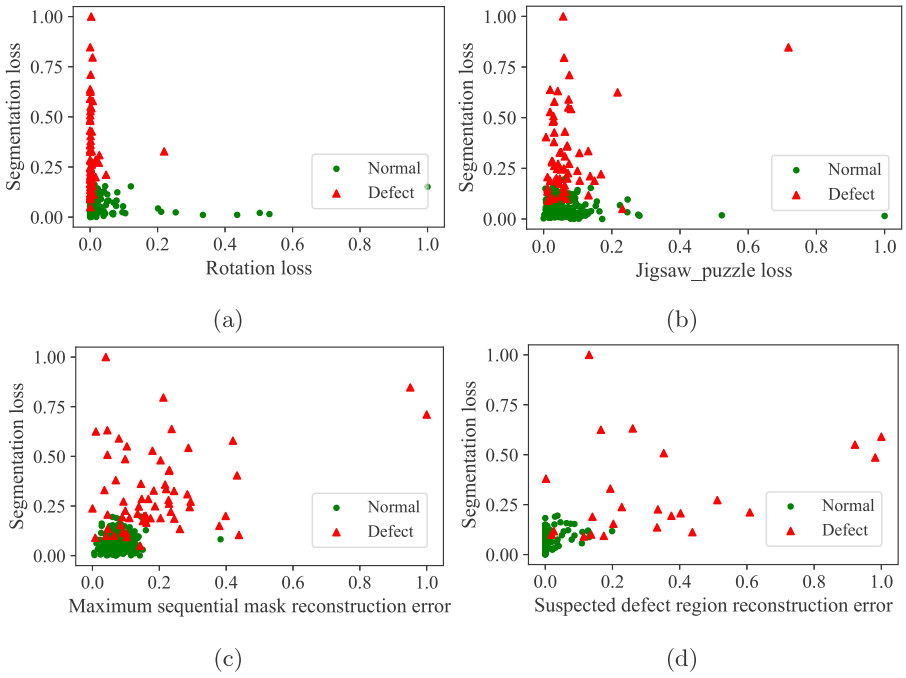



**Fig. 2.** The relationships between the different methods' evaluation values and the segmentation losses incurred on the KOS dataset. (a) (b) Neither the rotation loss nor jigsaw_puzzle loss is significantly related to the segmentation loss. (c) (d) The reconstruction error of the defective image is larger than that of the normal image. The reconstruction error includes the maximum sequential mask reconstruction error and the suspected defect region reconstruction error. The x- and *y*-axis data are normalized.

use of the information contained in the unlabeled data. Yoo et al. [15]. proposed a core set selection method based on an additional module called the "loss prediction module". The design of this module makes it unnecessary to design corresponding sample selection strategies when facing different tasks. Drawing on the learning loss method, Yi et al. [7]. directly used a pretext module to replace the additional module, evaluated unlabeled data using the pretext model, and selected the core data according to the loss value of the pretext model. In addition to the above examples, [16] also used active learning to solve image classification problems. virtual adversarial active learning(VAAL) [17], task-aware variational adversarial active learning (TA-VAAL) [8], etc., learn from generative adversarial networks (GANs) [18] to realize active learning.

### 2.2. Image segmentation

To design an active learning method for image segmentation tasks, Lin Yang [10] et al. used the bootstrapping [19] method to estimate the uncertainty and similarity of the top features of a segmentation model encoder. After selecting the sample with the highest uncertainty, they used a greedy algorithm to select the sample with the lowest similarity for manual labeling. Zhao et al. [20]. designed a core set selection method for medical image segmentation based on the differences in the segmentation results produced by different decoder layers. In particular, the papers that proposed VAAL [17], PT4AL [7] and other approaches [21] pointed out that their active learning methods can also be applied to image segmentation.

In addition to the above studies, researchers have also studied the applications of active learning methods to object detection [22,23], video processing [24], and other fields [25–27]. For example, in object detection, Hamed H [28] et al. proposed sorting the given unlabeled images in descending order according to the prediction results of the detection network and selecting the core data for labeling according to the sorting results, thereby reducing the cost of data labeling.

Overall, these methods design different active learning strategies for different problems in multiple visual tasks, but little relevant research has been conducted on how to deal with the data collected in industrial scenes. According to the characteristics of industrial data, we design an active learning method that includes a self-supervised reconstruction learner, a segmentation learner, and a reconstruction learner. Finally, our proposed method addresses the active learning problem in industrial image segmentation under the imbalanced data, ultimately reducing the cost of labeling and storage.

## 3. Methods

In this section, we give a detailed introduction to our method. The method is divided into two stages and consists of 3 learners, as shown in Fig. 3. (i) Initialization is performed based on a self-supervised reconstruction learner. This learner divides the input unlabeled data with the reconstruction model and selects informative data for labeling. (ii) Pseudolabel prediction is implemented based on a segmentation learner. This learner trains a segmentation model with labeled data and assigned fake labels and predicts pseudolabels for the unlabeled data using a segmentation model. (iii) The informative data selection process is based on the reconstruction learner in the iteration stage. This learner trains the reconstruction model on labeled data and the data with assigned fake labels. According to the pseudolabels predicted by the segmentation model, the reconstruction error is calculated for the unlabeled data through the reconstruction model. The informative data are selected for labeling after sorting the reconstruction errors.

### 3.1. Self-supervised reconstruction learner

The self-supervised reconstruction learner is used to solve the cold-start problem. For datasets with more normal images and fewer defective images, the key is how to make the initialization data contain more defect data. We use the difference between the reconstructed image and the original image to achieve this goal.

#### 3.1.1. Analysis of the self-supervised reconstruction learner

Following the vision transformer (ViT) [29] and masked autoencoder (MAE) [30], we divide the input image $I$ into regular nonoverlapping patches, and the set composed of all nonoverlapping patches

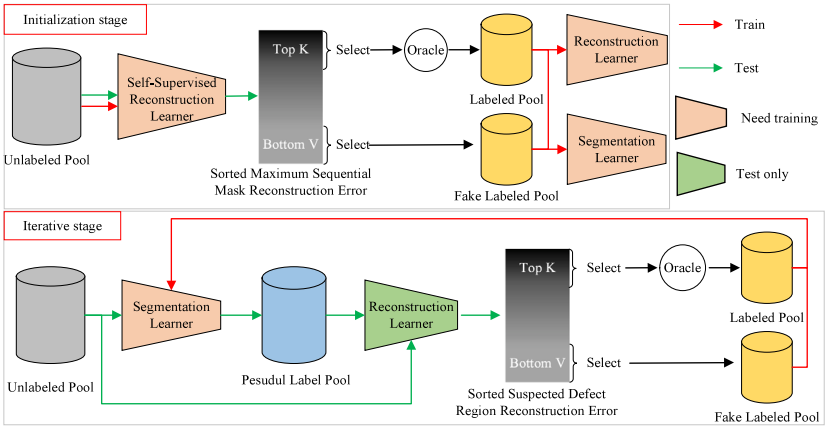



**Fig. 3.** The overall framework of the proposed method. In the initialization stage, we sort the unlabeled data according to their sequential mask reconstruction errors in descending order; then, we select the top $K$ data for manual labeling and the bottom $V$ data to assign fake labels. Segmentation and reconstruction learners are trained with labeled data and fake labeled data. In the iterative stage, we feed the unlabeled data into the segmentation learner to predict pseudolabels. The reconstruction learner calculates the reconstruction error of the suspected defect region according to the pseudolabels. Similar to the initialization stage, after sorting, the top $K$ data are selected for manual labeling, and the bottom $V$ data are assigned fake labels.

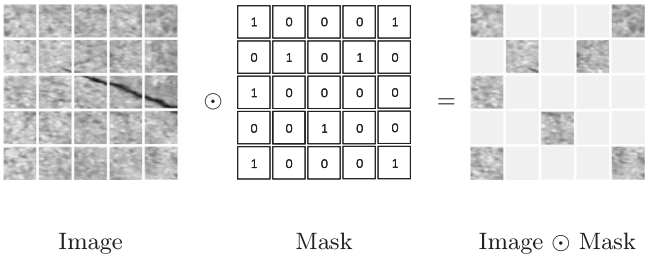



**Fig. 4.** Random mask image creation method.

is denoted as the full set $\mathbb{U}$. Partial nonoverlapping patches are randomly selected from $\mathbb{U}$ to form a new subset $B$. The selection of the subset $B$ is achieved by constructing a mask to mask out some of the nonoverlapping patches in the image. The detailed process is shown in Fig. 4.

During the construction of $B$, performing random sampling with high masking rates can largely eliminate redundancy while also creating a task where the original image content cannot be easily reconstructed via extrapolation from visible adjacent blocks. Following the practice in the MAE, combined with the high similarity of the background in industrial images, we choose a high masking ratio to remove some patches. The selected mask removes no less than 70% of the image information, which yields a higher probability of removing defective regions from the image, and the reconstruction result is more inclined to the image background (normal region). This phenomenon is verified in the reconstruction results presented in Fig. 5. It can be seen from Fig. 5 that for the masked region, the reconstructed image is more similar to the normal region. For our method, it is very important that the reconstruction results have prediction preference. The generation of this prediction preference is closely related to the characteristics of the training data.

For industrial images, the requirements for the product qualification ratio are higher, so the images that do not contain defective areas ($I_D$) in the entire dataset are much more numerous than the images that contain defective areas ($I_N$).

For a single industrial image $I$, we assume that an image consists of two parts, a defect region $D$ and a normal region $N$. The range of $N$ and $D$ is $[\emptyset, I]$; that is, an image has no defect regions or all defect regions. Influenced by the principles of defect generation in industrial images, such as cutting angles and cutting forces, the area of $D$ is usually smaller than that of $N$.

$$I = N + D \qquad N, D \in [\emptyset, I] \tag{1}$$

For the above reasons, when the reconstruction network is trained with a high masking rate, the probability of the input and reconstruction target of the network being a normal image is much higher than that of being a defective image. This makes the trained reconstruction network more inclined to reconstruct normal images.

### 3.1.2. Structure of the self-supervised reconstruction learner

Self-supervised reconstruction learners need to restore images to their original image size. Various structures can be employed to achieve this, such as UNet [31] and ViT. The symmetric encoder–decoder structure and skip connections in UNet enable Unet to achieve better convergence with less data. Therefore, it is widely used in the industrial field. For example, [32–36] applied the Unet structure to the image reconstruction task and achieved good results. Following [32–36], the self-supervised reconstruction learner adopts the UNet structure, as shown in Fig. 6. For the Unet structure, various implementation methods can be used, such as the encoder adopts the VGG structure or the ResNet structure, and the VGG encoder is used in this paper. However, the difference between our learner and UNet is that we replace the convolution kernel in UNet with a dilated convolution kernel to increase the receptive field. In addition, atrous convolution can reduce the influence of adjacent pixels on the pixel to be reconstructed. In particular, we make the skip connection structure optional. Skip connections impact the visualization of the reconstruction results but have no significant impact on the selection of informative data. In this paper, we use skip connections to better visualize the reconstruction results. Fig. 5 shows the partial reconstruction results.

### 3.1.3. Self-supervised reconstruction learner for batch splitting

In this subsection, we explain the method for dividing the unlabeled dataset $X_U$ into different batches using a random mask reconstruction algorithm. We use $\mathcal{L}_{rmse}$ to minimize the difference between the predictions and labels to train a self-supervised reconstruction model.

$$\mathcal{L}_{rmse} = \frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{M}(R(y_i);\theta_m) - y_i\right)^2 \tag{2}$$

where $R(\cdot)$ represents the random masking operation. $\mathbf{M}$ stands for the self-supervised reconstruction model. $\theta_m$ represents the weight parameters of $\mathbf{M}$. $y_i$ is the input and learning target, and $m$ is the batch size.

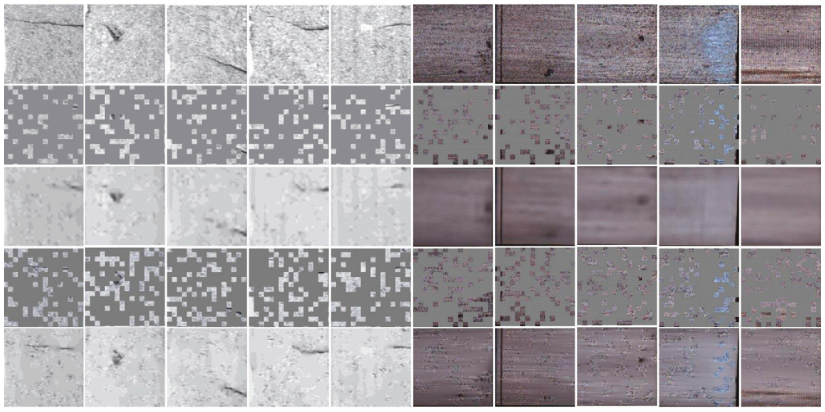


**Fig. 5.** The input and output of the self-supervised reconstruction learner. The second and fourth rows are the mask images, the third row contains the reconstructed images without skip connections, the fifth row includes the reconstructed images with skip connections, and the first row is the ground truth. The mask ratio is not less than 70%. The mask region in the reconstruction result is more inclined to the normal image.

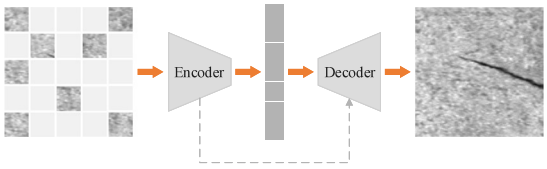



**Fig. 6.** The structure of the self-supervised reconstruction learner.

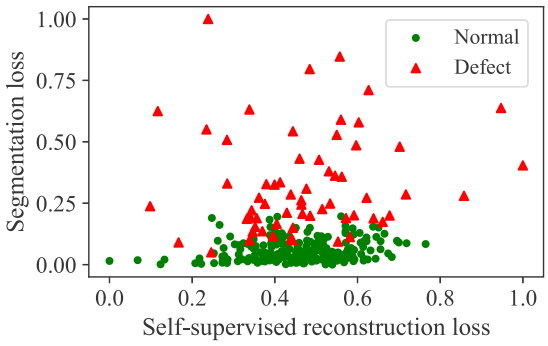



**Fig. 7.** The relationship between the self-supervised reconstruction loss ($\mathcal{L}_{rmse}$) and the segmentation loss on the KOS dataset. There is no obvious relationship between the two losses. The x- and $y$-axis data are normalized.

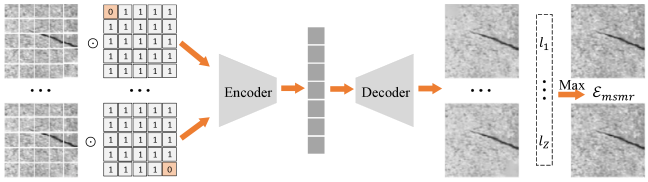



**Fig. 8.** $\mathcal{E}_{msmr}$ calculation method for a single unlabeled image.

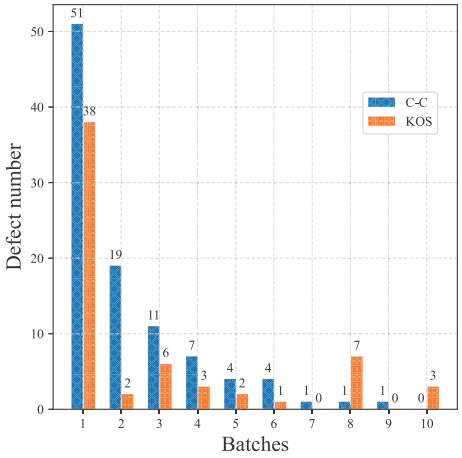



**Fig. 9.** Defect data distributions of the C–C dataset and KOS dataset divided into 10 groups in descending order based on $\mathcal{E}_{msmr}$. $\mathcal{E}_{msmr}$ gradually decreases from left to right, the number of defective samples in each group also decreases gradually, and the corresponding number of normal samples gradually increases.

In inference, this approach is chosen due to the weak correlation between $\mathcal{L}_{rmse}$ and the segmentation loss, as shown in Fig. 7. Therefore, we propose a method for calculating $\mathcal{L}_{rmse}$ in blocks and then calculate the maximum value, which is denoted as the maximum sequential mask reconstruction error ($\mathcal{E}_{msmr}$), as shown in Fig. 8. The relationship between $\mathcal{E}_{msmr}$ and the segmentation loss is shown in Fig. 2(c), from which it can be seen that the proposed method can better separate the defect data from the normal data.

Specifically, we divide each unlabeled data point into $Z$ equal blocks and generate $Z$ groups of masks to mask the $Z$ image blocks individually. The masked image is input into $\mathbf{M}$ to predict its result, and $\mathcal{E}_{msmr}$ is calculated, as shown in Fig. 8.

$$\begin{aligned} \mathcal{E}_{msmr} &= max\{l_1, \ldots, l_Z\}^T \\ J &= \left(Q(\mathbf{M}(Q(y_z); \theta_m) - y_z)\right)^2 \\ l_z &= \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} J_{ij}}{HW}. \end{aligned} \tag{3}$$

In inference, the batch size is set to 1. We replace $R(\cdot)$ with $Q(\cdot)$, which represents the sequential masking operation. $l_z$ represents the $\mathcal{L}_{rmse}$ value calculated for the $zth$ masked image, and there are $Z$ values in total. $\mathcal{E}_{msmr}$ is the maximum value of $l_z$. $H$ and $W$ are the height and width of the image $J$, respectively.

For all unlabeled data, a set $L$ consisting of $\mathcal{E}_{msmr}$ values is obtained by calculating the $\mathcal{E}_{msmr}$ corresponding to each image. Due to the preferences of our reconstruction method, the loss value corresponding to a defective image is larger, and the loss value corresponding to a normal image is smaller, so the data corresponding to the loss values at the bottom are closer to the normal data, as shown in Fig. 9. We sort the set $L$ in descending order. The top $K$ data are manually labeled, and normal labels are assigned to the bottom $V$ data. The proportion of $K$ and $V$ is related to the product qualification rate. Finally, the reconstruction model and segmentation model are trained using the top $K$ data and the bottom $V$ data.

### 3.2. Supervised segmentation learner

Due to the prediction bias of the reconstruction network, the selected $K$ samples contain more defective samples than the samples selected by other methods. First, we manually label the top $K$ samples to obtain a labeled dataset $X_L$ and assign normal labels to the bottom $V$ data to form $X_{FN}$. Second, we divide $X_L$ into a labeled defect dataset $X_{LD}$ and a labeled normal dataset $X_{LN}$. $X_{LN}$ and $X_{FN}$ are combined into $X_N$. Finally, we train the segmentation model using datasets $X_L$ and $X_N$.

Many segmentation networks are available, such as fully convolutional network (FCN) [37] and pyramid scene parsing network (PSPNet) [38], etc [39,40]. Due to the excellent results obtained by UNet on medical and industrial images, our segmentation network adopts UNet. For the design of the loss function, we first choose the popular binary cross-entropy (BCE) loss. However, BCE loss focuses on pixel-level differences, measuring the sum of the distances between predictions and labels for each corresponding pixel pair. Thus, the BCE loss focuses on the overall distribution of the data. In industrial data, BCE loss pays more attention to normal data with more quantity. To give more attention to defects, the Dice loss is chosen. Dice loss is considered to measure image-level differences and focuses on measuring the overlap between predicted maps and ground truth. In industrial images, Dice loss pays more attention to defect regions. The weighted loss function based on BCE and Dice can combine the advantages of the two loss functions. When this weighted loss is applied to industrial data, it can focus not only on the overall distribution of the image but also on defect regions. At the same time, BCE loss and Dice's loss function have been applied to image segmentation tasks by [41–45] and have achieved excellent results. Following [41–45], the segmentation loss is a weighted sum of the BCE and Dice:

$$\mathcal{L}_{seg} = \beta * Bce + \gamma * Dice \tag{4}$$

Similar to [41], we set the weight parameter $\beta$ to 0.5 and $\gamma$ to 1.

The training set is $X_L$, and the test set is $X_U$. Since $X_U$ does not have labels, it is difficult to judge the quality of the prediction results produced by the trained model for unlabeled data through evaluation indicators, such as the intersection over union (IoU). We select the segmentation model (**S**) corresponding to the minimum loss value over the last 20 training epochs to predict $X_U$ for generating pseudolabels ($L_p$). The $L_p$ are used in the iterative stage to compute the reconstruction loss.

### 3.3. Supervised reconstruction learner

The self-supervised reconstruction learner makes the reconstruction results of defect regions tend to the normal region, while the normal region is maintained. This is achieved by using random masks to remove defective regions with a certain probability. However, when the labeled image is obtained, we can use the labels to remove the defective regions from the image, and the reconstruction target (normal image) can also be constructed from the labels. When calculating the reconstruction error, the result predicted by the segmentation network is used as the calculation region. In this way, the situation in which some defective images and normal images are indistinguishable due to their small reconstruction errors in the self-supervision stage is alleviated. Therefore, we design a new informative data selection algorithm to replace the self-supervised reconstruction model-based data selection algorithm. We use the labeled data to train a new image reconstruction model whose structure is exactly the same as that of the self-supervised reconstruction model, so we use the self-supervised reconstruction model as a pretrained model. When we load the data, we first select an image $I$ and its corresponding label $I_L$ from the dataset $X_{LD}$. Due to the high similarity between industrial images, we randomly select $B$ images from $X_{LN}$ with a weighted average to obtain a denoised normal image $I_0$.

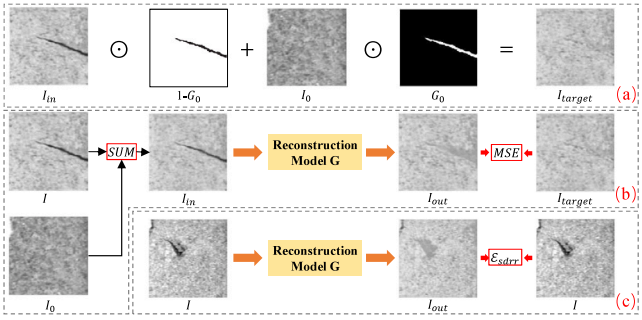



**Fig. 10.** The overall process of the pseudolabel reconstruction method. **(a)** is the construction method for the training target. **(b)** is the input image construction method and the loss calculation method used during training. **(c)** is the error calculation method in the inference process.

$$I_0 = \frac{1}{B}\sum_{i=1}^{b} X_i \tag{5}$$

To input more information into the neural network, we add $I$ and $I_0$ with weighting to obtain the input image $I_{in}$:

$$I_{in} = \begin{cases} \alpha_0 I + (1-\alpha_0) I_0, & id\%3 == 0 \\ \alpha_1 I + (1-\alpha_1) I_0, & id\%2 == 0 \\ I, & \text{else} \end{cases} \tag{6}$$

Among them, $id$ is a random integer generated during addition, which is used to control the addition method. $\alpha_0$ and $\alpha_1$ are weight coefficients, which are set to 0.8 and 0.7, respectively. We hope to recover a defect-free image from a defective image, so the reconstruction target ($I_{target}$) for removing the defect region is obtained by the following formula:

$$I_{target} = I_{in} \odot (1 - G_0) + I_0 \odot G_0 \tag{7}$$

where $G_0$ is the mask generated by the label $I_L$, the background is marked as 0, and the defect region is marked as 1. With this equation, we remove the defect regions from the image and fill the original defect regions with the normal image, as shown in Fig. 10(a).

During training, the reconstruction loss adopts the mean squared error ($MSE$) loss, and the reconstruction model **G** is obtained via training (Fig. 10(b)). In the inference phase, we input unlabeled data and compute the suspected defect region reconstruction error ($\mathcal{E}_{sdrr}$) with the following equation:

$$\begin{aligned} P &= (G_1 \odot \mathbf{G}(I;\theta_g) - G_1 \odot I)^2 \\ \mathcal{E}_{sdrr} &= \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} P_{ij}}{HW} \end{aligned} \tag{8}$$

$I_{pre}$ is the prediction results of the segmentation model for unlabeled images, which are selected from $L_p$. $G_1$ is calculated by $I_{pre}$:

$$G_1 = \begin{cases} 1, & I_{pre}[i][j] \geq 0.5, i \in [0,H], j \in [0,W] \\ 0, & I_{pre}[i][j] < 0.5, i \in [0,H], j \in [0,W] \end{cases} \tag{9}$$

where $H$ and $W$ are the height and width of the image $P$, respectively. $\theta_g$ represents the weight parameters of **G**. When calculating $\mathcal{E}_{sdrr}$, we only calculate the loss value of the mask region; that is, only the suspected defect region is considered. The larger $\mathcal{E}_{sdrr}$ is, the greater the difference between the defect region and the normal region.

For all unlabeled data, after calculating the corresponding $\mathcal{E}_{sdrr}$ for each image, a set $C$ composed of $\mathcal{E}_{sdrr}$ is obtained. $C$ is arranged in descending order, the top $K$ data are taken for manual labeling, and these data are put into $X_L$. The bottom $V$ data are assigned normal

**Algorithm 1** Selecting Informative Data for Defect Segmentation from Imbalanced Data via Active Learning.

**Input:** Unlabeled dataset $X_U$, number of iterations $T$, self-supervised reconstruction model training loss $\mathcal{L}_{rmse}$, maximum sequential mask reconstruction error $\mathcal{E}_{msmr}$, segmentation model training loss $\mathcal{L}_{seg}$, suspected defect region reconstruction error $\mathcal{E}_{sdrr}$.

**Output:** $X_L$

*/* **Initialization stage***/*

Use $X_U$ and $\mathcal{L}_{rmse}$ to train the self-supervised reconstruction model **M**

Sequential masks calculate $\mathcal{E}_{msmr}$ for $X_U$ by **M**

$X_L \leftarrow \{X_U | X_U \in \arg\underset{X_U}{\text{top K}}\ (\mathcal{E}_{msmr})\}$

$X_{FN} \leftarrow \{X_U | X_U \in \arg\underset{X_U}{\text{btm V}}\ (\mathcal{E}_{msmr})\}$

Use $X_L$, $X_{FN}$, and $\mathcal{L}_{seg}$ to train the segmentation model **S**

Use $X_L$, $X_{FN}$, and the $MSE$ to train the reconstruction model **G**

$X_U \leftarrow X_U - X_L, X_{FN} \leftarrow \emptyset$

*/* **Iterative stage***/*

**while** maximum number of iterations $T$ not reached **do**

  **S** generates the $L_P$ of $X_U$ to calculate $\mathcal{E}_{sdrr}$ by **G**

  $X_L \leftarrow \{X_U | X_U \in \arg\underset{X_U}{\text{top K}}\ (\mathcal{E}_{sdrr})\}$

  $X_{FN} \leftarrow \{X_U | X_U \in \arg\underset{X_U}{\text{btm V}}\ (\mathcal{E}_{sdrr})\}$

  Use $X_L$, $X_{FN}$, and $\mathcal{L}_{seg}$ to train **S**

  $X_U \leftarrow X_U - X_L,\ X_{FN} \leftarrow \emptyset$

**endwhile**

**return** $X_L$

**Table 1**
The numbers of normal and defective data in the datasets.

| Dataset | Image_size | Defect | Normal | Defect/Normal |
|---|---|---|---|---|
| CCD | 256 × 256 | 99 | 2187 | 0.045 |
| KOS | 500 × 500 | 62 | 1138 | 0.054 |
| RSDDs | 55 × 55 | 189 | 2671 | 0.071 |

labels and put into $X_{FN}$. These data are used to train the segmentation model.

On the basis of introducing the three proposed modules in detail, we organize the operation process of our method into pseudocode, as shown in Algorithm 1.

## 4. Experimental settings

First, we select state-of-the-art active learning methods for image segmentation to conduct comparative experiments. Random sampling is a method that randomly selects data from unlabeled samples $X_U$ for labeling, which is the lower limit of the active learning method comparison. PT4AL [7], LearningLoss [15] and TA-VAAL [17] are task-independent active learning methods that perform best on multiple datasets. Deeply supervised active learning (DSAL) [20] and suggestive annotation (SA) [10] are active learning methods that have been proposed for image segmentation tasks.

### 4.1. Datasets

We evaluate the proposed active learning method for semantic segmentation on three datasets. The relevant information about the three datasets is shown in Table 1.

- C–C dataset. The commutator cylinder dataset (C–C dataset) is a machining dataset that was collected by a self-designed arc light and a CCD high-speed camera in a commutator production line. As shown in the first row of Fig. 11, the dataset contains 96 workpieces, each with 12 images, and after data augmentation, 2187 normal images and 99 abnormal images are obtained. The data augmentation method scales each surface image to 512 × 256 and then divides each image into 4 blocks, resulting in four images of size 256 × 256. The dataset contains a variety of defects, such as inclusions, surface dirt, and tin color, as shown in the second row of Fig. 11.
- KOS dataset. The KOS dataset [46] is a public commutator cylinder dataset, which is also a machining dataset. The dataset consists of 50 artefacts, each of which contains up to 8 images with an image size of 500 × 1265. For the convenience of training, we use data augmentation to obtain 1138 normal images and 62 defect images. The data augmentation method divides the original images and labels according to a step size of 500 × 500. The last image is divided from the bottom to the top, so each image can be divided into three images. These data contain only one defect type, as shown in the third row of Fig. 11.
- RSDDs dataset. The RSDDs is a publicly available normal/heavy transport track surface defect dataset. In this dataset, the background difference between different images is obvious, and the noise is larger, which makes the detection more difficult. The RSDDs dataset contains 128 images with an image size of 55 × 1250. For the convenience of training, we cropped the original image with a step size of 55 in the height direction and obtained 2671 normal images and 189 defective images. Some normal images and defect images in this dataset are shown in Fig. 11.

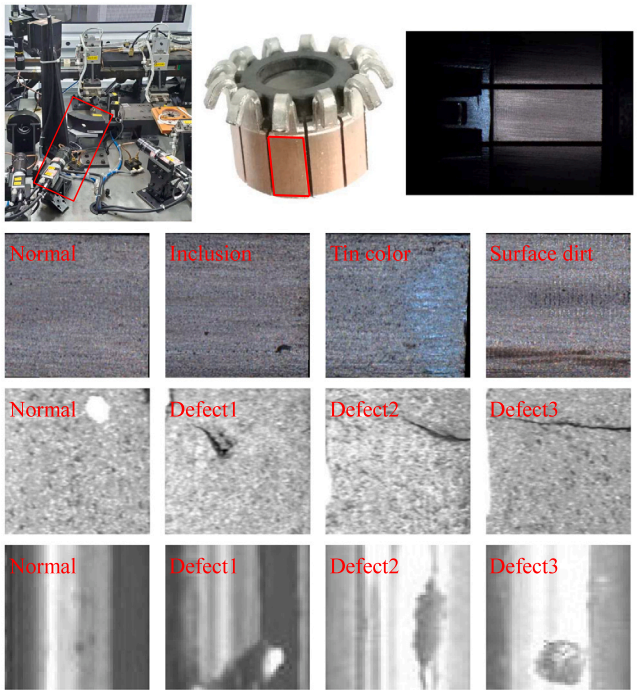



**Fig. 11.** The first row presents the collection and detection locations of the C–C dataset. The second row is a partial sample from the C–C dataset. The third row is a partial sample from the KOS dataset. The fourth row is a partial sample from the RSDDs dataset.

### 4.2. Training details

Our method includes three models: the self-supervised reconstruction model (**M**), the supervised segmentation model (**S**) and the supervised reconstruction model (**G**). In these models, the parameter settings are the same except for the learning rate and number of epochs. The detailed settings are shown in Table 2. Especially for **S**, we augment the data by randomly changing the image color and contrast to prevent

**Table 2**
Parameters settings in different models.

| Models | M | S | G |
|---|---|---|---|
| Batch size | 32 | 32 | 32 |
| Optimization method | SGD | SGD | SGD |
| Momentum | 0.9 | 0.9 | 0.9 |
| Weight_decay | 5e−4 | 5e−4 | 5e−4 |
| Learning rate | MultiStepLR[a] | 0.001 | MultiStepLR[a] |
| Number of epochs | 120 | 150 | 120 |

[a]The basic learning rate is set to 0.1, and the update step size is set to 30, 60, and 90.

segmentation model overfitting or training failures due to the use of too few defective data.

For the C–C, KOS, and RSDDs datasets, the number of labeled samples in each iteration is set to 10% of the total number of samples, i.e., 228, 120, and 286 images, respectively. We iterate 6 times and select 60% of the total data for labeling. During this process, the self-supervised reconstruction model is only trained and used in the initialization stage. A supervised reconstruction model can be trained once and called repeatedly. The supervised segmentation model needs to be trained in the initialization stage and the iteration stage, while incremental training is performed according to the data selected after each iteration. In the initialization stage, we do not use pretrained weights to ensure that the improvement in the initial segmentation results is due to the inclusion of more informative data in the training set rather than the influence of pretrained weights. In the iterative stage, the training weights obtained in the previous round can reflect the effect of the proposed method in the previous round. Therefore, we load the training weights of the previous round as pretraining weights to speed up the training of the model and make the model training process more stable.

During the experiment, we did not adjust the data mask type and the architecture of each learner when testing the three datasets. For example, for the random mask in the self-supervised reconstruction learner, since the design of the random mask is based on the characteristics of the high product qualification rate of industrial data, it is less affected by the difference in the dataset. Therefore, when we conduct experiments on three different datasets, we only need to ensure that the mask rate is not lower than 70%. For the mask type in the supervised reconstruction learner, it is completely generated by the segmentation learner, so manual adjustment is not needed. The structure of each learner has not been modified in the experiments of the three datasets because each learner structure is designed with full consideration of industrial application scenarios and the universality of these structures in industry. At the same time, the similarity of tasks is also the reason why no modification of the learner structure is needed. For example, the input of the reconstruction learner is the image with some regions removed (random mask or pseudolabel), and the reconstruction target is the original image, so the structure of the reconstruction learner was not modified in the experiments of the three datasets.

The experimental system is a CentOS 8 Linux platform, which is designed and implemented in the Pycharm software through the open source PyTorch library, and the training and testing procedures of the model are implemented in an NVIDIA Tesla A100 40-GB GPU.

## 5. Experimental results

### 5.1. Study on the cold-start problem

To address the cold-start problem, we propose a method to initialize the data with a self-supervised reconstruction model. We apply a random mask to the active learning initialization process for the first time. Our method is also obtained by optimizing the random mask. Therefore, we also compare the random mask as an initialization method. The trained self-supervised models are tested with all unlabeled images and sorted in descending order according to their test loss values or reconstruction errors. We split the sorted data into 10 batches and select the unlabeled images corresponding to the first batch as initialization data.

**Table 3**
Times required by different active learning methods.

| Methods | TA-VAAL | LearningLoss | SA | DSAL | PT4AL | Ours |
|---|---|---|---|---|---|---|
| Time (min) | 145 | 88 | 33 | 30 | 45 | 30 |

The results of different cold-start initialization methods are shown in Fig. 12. From the experimental results, it can be seen that compared with other methods, our method selects more defect samples (22 → 51) with rich information in the first batch. At the same time, the defect images in the data selected by our method are more concentrated in the first few batches (e.g., 82% of the defect data are contained in the first three batches). Other methods have higher randomness and are similar to randomly initialized distributions. The data distribution characteristics selected by our method are more in line with the requirement that an active learning method should mark as many defect images as possible at a lower cost. This proves that our method has a good effect on solving the cold-start problem in industrial data segmentation.

### 5.2. Study on informative data screening

The experimental results obtained by our method and other methods on the C–C dataset, KOS dataset, and RSDDs dataset are shown in Fig. 13. Compared with other methods, our proposed method finds more defects. The proportions of defects selected in the first iteration are 51.5% for the C–C dataset, 61.3% for the KOS dataset, and 92.1% for the RSDDs dataset, which are approximately 240%, 375%, and 657% higher than those of other methods, respectively. It is worth noting that due to the obvious structural characteristics of the RSDDs dataset, the rotation method adopted in PT4AL pays more attention to the texture than the defect, so that the amount of defect data selected in the RSDDs dataset is even weaker than the random initialization method. In the second iteration, the numbers of selected defective images (51.5% → 92.9% from the C–C dataset, 61.3% → 98.4% from the KOS dataset, and 92.1% → 98.9% from the RSDDs dataset) are rapidly increased again by the reconstruction method, which indicates that this method is beneficial for selecting more informative defective images from an unlabeled dataset. In the RSDDs dataset, since the proportion of defective data selected in the first iteration is more than 92.1%, in the second iteration, the proportion of defective data selected is relatively small. However, in the second iteration, the number of selected defects accounted for 86.7% of the remaining unfound defects. After three iterations, the number of selected defect images accounts for more than 95% of all defects. This proves that compared with other methods, our method can greatly reduce the cost of data screening. We observe that SA and DSAL, which are normally applied to medical segmentation tasks, have better performance than the task-independent LearningLoss, TA-VAAL, and PT4AL methods in industrial data segmentation tasks, which indicates that it is necessary to explore task-oriented active learning methods. In conclusion, our method has distinct advantages over other methods and has great adaptability in different datasets. This proves that the proposed framework is beneficial for segmentation tasks because it selects the most informative samples from an unlabeled pool.

### 5.3. Algorithmic running time analysis

Table 3 shows the running times of the tested models in the stable iteration stage (the fourth to fifth iterations) for different methods. As seen from the table, except for those of the TA-VAAL and LearningLoss methods, the running times of the other methods are similar. The reason for the long running time of TA-VAAL is that in each iteration, both

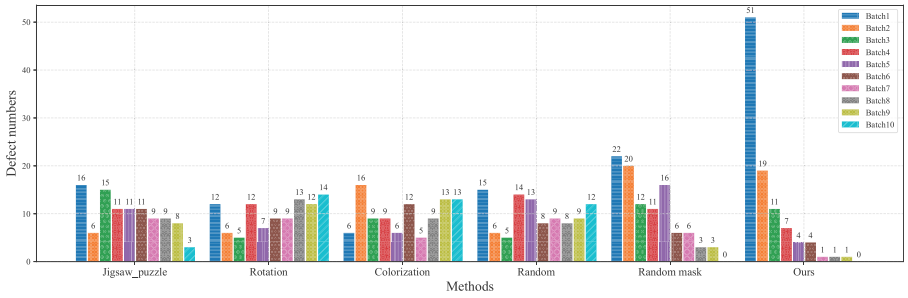
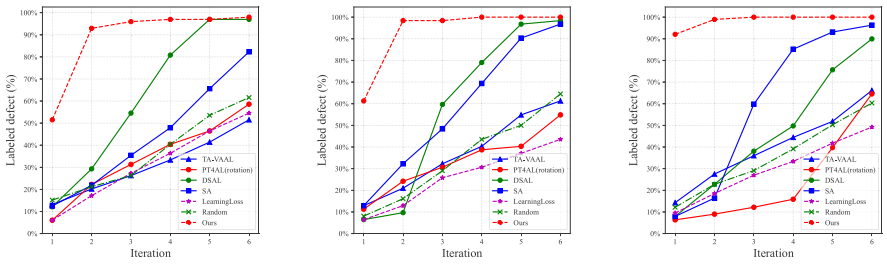



**Fig. 12.** Batch division results obtained for unlabeled data by different cold-start methods on the C–C dataset.




(a) C-C dataset (b) KOS dataset (c) RSDDs dataset

**Fig. 13.** Selection results for informative data by different active learning methods on different datasets.

**Table 4**
Pseudo-label reconstruction model selects the number of defect images by different initialization methods on the C–C dataset.

| Methods | Initialization | Second iteration | | |
|---|---|---|---|---|
| | | PT4AL | Random | Ours |
| Jigsaw_puzzle | 16 | 6 | – | 38 |
| Rotation | 6 | 16 | – | 60 |
| Colorization | 12 | 6 | – | 30 |
| Random mask | 22 | 20 | – | 57 |
| Random | 15 | – | 25 | 69 |

**Table 5**
IoU values under different proportions of data selected by different active learning methods from the C–C dataset.

| Labels | TA-VAAL | PT4AL | DSAL | SA | LearningLoss | Random | Ours |
|---|---|---|---|---|---|---|---|
| 10% | 0.0000 | 0.0111 | 0.0290 | 0.008 | 0.2288 | 0.0291 | **0.5533** |
| 20% | 0.0401 | 0.2911 | 0.5424 | 0.5583 | 0.5085 | 0.3318 | **0.5666** |
| 30% | 0.3373 | 0.3312 | 0.5432 | 0.5590 | 0.5330 | 0.3215 | **0.5701** |
| 40% | 0.5608 | 0.3272 | 0.5667 | 0.5582 | 0.5592 | 0.3368 | **0.5716** |
| 50% | 0.5469 | 0.4678 | 0.5299 | 0.5617 | 0.5609 | 0.5495 | **0.5671** |
| 60% | 0.5044 | 0.5194 | 0.5707 | 0.5581 | 0.5241 | 0.5440 | **0.5649** |
| 70% | 0.5443 | 0.5291 | 0.5544 | 0.5639 | 0.5534 | 0.5454 | **0.5669** |
| 80% | 0.5630 | 0.5564 | 0.5526 | 0.5623 | 0.5432 | 0.5564 | **0.5542** |
| 90% | 0.5394 | 0.5402 | 0.5426 | 0.5612 | 0.5646 | 0.5207 | **0.5554** |
| 100% | 0.5431 | | | | | | |

a variational autoencoder (VAE) network and a segmentation network need to be trained. LearningLoss requires training to be conducted in both the segmentation network and the loss prediction network. Both SA and DSAL only need to train the segmentation network. In addition, both PT4AL and our method need to train a pretext model, but the pretext model only needs to be trained in the initialization stage. Therefore, in the stable iteration stage, PT4AL and our method only need to train the segmentation model. Overall, our method is basically the same as the other methods in terms of the required runtime and achieves better query results.

### 5.4. Ablation study

In the ablation study, we explore the effectiveness of different data selection methods in the iterative stage and the performance of the selected data. The cold-start experiment has been studied in Section 5.1 and is not discussed here.

#### 5.4.1. Data selection methods in the iterative stage

To verify that the data selection method in the iteration stage is independent of the initialization method, we design the following experiments. First, the data are initialized using different cold-start methods, such as Jigsaw_puzzle, Colorization, the Rotation technique in PT4AL and Random initialization. Then, the segmentation model and reconstruction model are trained according to the initialization data, and pseudolabels are obtained by predicting all unlabeled data using the trained segmentation model. Finally, $\mathcal{E}_{sdrr}$ is calculated by using the reconstruction model and pseudolabels, and the first 228 images are selected as informative data after sorting them in descending order. The number of defective images among the selected images is shown in Table 4.

It can be seen from the data in Table 4 that no matter what method is used for initialization, after the second iteration, the number of defect samples found significantly increases. Compared with the PT4AL method, after the second iteration, the number of defective samples included in the informative samples selected by our method is 5.3 times, 2.75 times, and 4 times higher than those of the Jigsaw_puzzle, Rotation, and Colorization methods, respectively. Compared with methods that select informative samples by random initialization, our method selects 1.76 times as many defective samples and is the best-performing method among these tested approaches. These experiments fully demonstrate the effectiveness of each method proposed in this paper.

#### 5.4.2. The performance of the selected data

To further verify that data screening is necessary, we use the data selected by various methods to train the UNet model and test it on the same batch of datasets. The experimental results are shown in Table 5. It can be seen from the experimental results that our method

significantly outperforms other methods when the data labeling ratios are 10% and 30%. This is closely related to the fact that we pick almost all of the defect data in the first three rounds. When the number of markers gradually increases and reaches 40%, the performance of other methods also improves, but it is still lower than that of our method. It is worth noting that the IoU value obtained from the partial labeling of the data is higher than that of the whole labeled data. Additionally, Table 5 shows that for the C–C dataset, the best performance of our method occurs when the number of markers is 40%, and when the number of markers is above 40%, the segmentation IoU gradually decreases. The optimal ratio of other methods is above 60%, such as 90% for learning loss and 70% for SA. Especially, the maximum IoU value of other methods is lower than ours. The reason for this result is that we preferentially select more defective data, avoiding the influence of a large number of normal images with low information on the neural network training process, thereby preventing the network from overfitting the normal images. This justifies the necessity of preferentially selecting defective data from imbalanced datasets.

## 6. Conclusion

In this paper, we propose an active learning framework for defect segmentation, aiming to preferentially select defect data from imbalanced industrial datasets for labeling and storage, thereby reducing data storage and labeling costs. The active learning framework is divided into an initialization stage and an iterative stage. In the initial stage, to solve the cold start problem, a random mask self-supervised reconstruction learner and a maximum sequential mask reconstruction error are proposed. The data selection method in the initialization stage makes full use of the characteristics of less defect data and more normal data in industrial data. For solving the cold start problem, the proposed method is more suitable for industrial scenarios that require a higher product qualification rate than other methods, such as Rotation, Colorization, and Jigsaw_puzzle in PT4AL. In practical applications, the defect data selected by the proposed method accounted for 51.5%, 61.3%, and 92.1% of the total defect data in the two commutator datasets (C–C dataset and KOS dataset) and one rail dataset (RSDDs dataset), respectively, which is significantly better than other methods. In the iterative stage, the supervised reconstruction learner based on the label mask and the suspected defect region reconstruction error are further designed. Data selection methods in the iterative stage focus on the defect region, which further amplifies the difference between the defect image and the normal image. A more accurate and efficient selection of informative defect data from unlabeled data is realized. This method associates the image segmentation problem with the image reconstruction problem, which is more suitable for image segmentation tasks compared with LearningLoss, SA and other methods. Finally, on the three test datasets, after two rounds of iterations on the basis of the initialization data, the proportion of the selected defective data is further increased to 96% (C–C dataset), 98% (KOS dataset), and 100% (RSDDs dataset). This demonstrates that the proposed framework achieves approximately 70% storage and labeling cost savings. In particular, the proposed method can achieve the same effect of labeling the whole dataset by labeling a partial dataset.

In the industrial field, defect data are more important than normal data. Therefore, the defect data preferentially selected by our method form the more valuable part of the overall dataset collected in an industrial production line. This plays an important role in reducing the cost of labeling and quickly iterating the model. However, for the selected datasets, which data from the same category are more valuable remains a task worth studying. This task becomes increasingly important as the number of defective samples collected in the production line increases. Therefore, in the future, we will explore core set selection methods for different classes of defect data.

## Ethical approval

We confirm that the content of the manuscript has not been published or submitted for publication elsewhere. All authors have seen the manuscript and approved it for submission to the journal.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The authors do not have permission to share data.

## Acknowledgments

This work was supported by the the Key R&D Program of Hubei China under Grant 2020BAB106. (Corresponding author: Bin Li.).

## References

[1] Qinghua Lu, Junmeng Lin, Lufeng Luo, Yunzhi Zhang, Wenbo Zhu, A cost-effective and automatic surface defect inspection system for hot-rolled flat steel, Adv. Eng. Inform. 53 (2022) 101692.
[2] Rongge Xu, Ruiyang Hao, Biqing Huang, Efficient surface defect detection using self-supervised learning strategy and segmentation network, Adv. Eng. Inform. 52 (2022) 101566.
[3] Du-Ming Tsai, Po-Hao Jen, Autoencoder-based anomaly detection for surface defect inspection, Adv. Eng. Inform. 48 (2021) 101272.
[4] Sener Ozan, Savarese Silvio, Active learning for convolutional neural networks: a core-set approach, 2017, http://dx.doi.org/10.48550/arXiv.1708.00489.
[5] Wang Keze, Zhang Dongyu, Li Ya, Zhang Ruimao, Lin Liang, Cost-effective active learning for deep image classification, IEEE Trans. Circuits Syst. Video Technol. 27 (12) (2016) 2591–2600.
[6] Lin Jianzhe, Zhao Liang, Li Shuying, Ward Rabab, Wang Z Jane, Active-learning-incorporated deep transfer learning for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (11) (2018) 4048–4062.
[7] Yi John Seon Keun, Seo Minseok, Park Jongchan, Choi Dong-Geol, Using self-supervised pretext tasks for active learning, in: 2022 European Conference on Computer Vision (ECCV), 2022, pp. 596–612, http://dx.doi.org/10.1007/978-3-031-19809-0_34.
[8] Kim Kwanyoung, Park Dongwon, Kim Kwang In, Chun Se Young, Task-aware variational adversarial active learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8166–8175.
[9] Zhao Ziyuan, Zeng Zeng, Xu Kaixin, Chen Cen, Guan Cuntai, Dsal: Deeply supervised active learning from strong and weak labelers for biomedical image segmentation, IEEE J. Biomed. Health Inf. 25 (10) (2021) 3744–3751.
[10] Yang Lin, Zhang Yizhe, Chen Jianxu, Zhang Siyuan, Chen Danny Z, Suggestive annotation: a deep active learning framework for biomedical image segmentation, in: International conference on medical image computing and computer-assisted intervention, 2017, pp. 399–407.
[11] Gidaris Spyros, Singh Praveer, Komodakis Nikos, Unsupervised representation learning by predicting image rotations, in: 2018 International Conference on Pattern Recognition (ICLR), 2018, http://dx.doi.org/10.48550/arXiv.1803.07728.
[12] Noroozi Mehdi, Favaro Paolo, Unsupervised learning of visual representations by solving jigsaw puzzles, in: European conference on computer vision, 2016, pp. 69–84.
[13] Chen Xinlei, He Kaiming, Exploring simple siamese representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15750–15758.
[14] Zhang Richard, Isola Phillip, Efros Alexei A, Colorful image colorizationn, in: European conference on computer vision, 2016, pp. 649–666.
[15] Yoo Donggeun, Kweon In So, Learning loss for active learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 93–102.
[16] Ranganathan Hiranmayi, Venkateswara Hemanth, Chakraborty Shayok, Panchanathan Sethuraman, Deep active learning for image classification, in: 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3934–3938.
[17] Sinha Samarth, Ebrahimi Sayna, Darrell Trevor, Variational adversarial active learning, in: Proceedings of the IEEE/CVF International Conference on Computer Visio, 2019, pp. 5972–5981.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[19] Tibshirani Robert J, Efron Bradley, An introduction to the bootstrap, Monographs Stat. Appl. Probab. 57 (1993) 1–436.
[20] Zhao Ziyuan, Yang Xiaoyan, Veeravalli Bharadwaj, Zeng Zeng, Deeply supervised active learning for finger bones segmentationn, in: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020, pp. 1620–1623.
[21] Gaur Utkarsh, Kourakis Matthew, Newman-Smith Erin, Smith William, B.S. Manjunath, Membrane segmentation via active learning with deep networks, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 1943–1947.
[22] Qu Zhenshen, Du Jingda, Cao Yong, Guan Qiuyu, Zhao Pengbo, Deep active learning for remote sensing object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, http://dx.doi.org/10.48550/arXiv.2003.08793.
[23] Roy Soumya, Unmeshv Asim, Namboodiri Vinay P, Deep active learning for object detection, in: BMVC, 2018, p. 91.
[24] Hossain HM Sajjad, Al Haiz Khan MD Abdullah, Roy Nirmalya, Deactive: scaling activity recognition with active deep learning, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2 (2) (2018) 1–23.
[25] An Bang, Wu Wenjun, Han Huimin, Deep active learning for text classification, in: Proceedings of the 2nd International Conference on Vision, Image and Signal Processingio, 2018, pp. 1–6.
[26] Shim Jaewoong, Kang Seokho, Domain-adaptive active learning for cost-effective virtual metrology modelingn, Comput. Ind. 135 (2022) 103572.
[27] Qi Zhou, Xinyu Shao, Ping Jiang, Zhongmei Gao, Chaochao Wang, Leshi Shu, An active learning metamodeling approach by sequentially exploiting difference information from variable-fidelity models, Adv. Eng. Inform. 30 (3) (2016) 283–297.
[28] Aghdam Hamed H, Gonzalez-Garcia Abel, Weijer Joost van de, López Antonio M, Active learning for deep detection neural network, Comput. Ind. (2019) 3672–3680.
[29] Dosovitskiy Alexey, Beyer Lucas, Kolesnikov Alexander, Weissenborn Dirk, Zhai Xiaohua, Unterthiner Thomas, Dehghani Mostafa, Minderer Matthias, Heigold Georg, Gelly Sylvain, et al., An image is worth 16x16 words: transformers for image recognition at scaleg, in: Proceedings of the IEEE/CVF International Conference on Computer Visio, 2020, http://dx.doi.org/10.48550/arXiv.2010.11929.
[30] He Kaiming, Chen Xinlei, Xie Saining, Li Yanghao, Dollár Piotr, Girshick Ross, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, p. 16000.
[31] Ronneberger Olaf, Fischer Philipp, Brox Thomas, U-net: convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, 2015, pp. 234–241.
[32] He Penghao, Yu Ying, Xu Chaoyue, Yang Hao, RAIDU-Net: image inpainting via residual attention fusion and gated information distillation, in: International Conference on Neural Information Processing, 2021, pp. 141–151.
[33] Yang Hao, Yu Ying, Res2u-net: image inpainting via multi-scale backbone and channel attention, in: International Conference on Neural Information Processing, 2020, pp. 498–508.
[34] Zhu Jun-Yan, Park Taesung, Isola Phillip, Efros Alexei A, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.
[35] Liu Guilin, Reda Fitsum A, Shih Kevin J, Wang Ting-Chun, Tao Andrew, Catanzaro Bryan, Image inpainting for irregular holes using partial convolutions, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 85–100.
[36] Yan Zhaoy, Li Xiaoming, Li Mu, Zuo Wangmeng, Shan Shiguang, Shift-net: image inpainting via deep feature rearrangement, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 1–17.
[37] Long Jonathan, Shelhamer Evan, Darrell Trevor, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
[38] Zhao Hengshuang, Shi Jianping, Qi Xiaojuan, Wang Xiaogang, Jia Jiaya, Pyramid scene parsing network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890.
[39] Hou Qibin, Zhang Li, Cheng Ming-Ming, Feng Jiashi, Strip pooling: rethinking spatial pooling for scene parsing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4003–4012.
[40] Chen Liang-Chieh, Papandreou George, Schroff Florian, Adam Hartwig, Rethinking atrous convolution for semantic image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, http://dx.doi.org/10.48550/arXiv.1706.05587.
[41] Zhou Zongwei, Siddiquee Md Mahfuzur Rahman, Tajbakhsh Nima, Liang Jianming, UNet++: A nested U-net architecture for medical image segmentation, in: Deep learning in medical image analysis and multimodal learning for clinical decision support, 2018, pp. 3–11.
[42] Fan Mingyuan, Lai Shenqi, Huan Junshi, Wei Xiaoming, Chai Zhenhua, Luo Junfeng, Wei Xiaolin, Rethinking bisenet for real-time semantic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 9716–9725.
[43] Yang Lei, Fan Junfeng, Huo Benyan, Li En, Liu Yanhong, A nondestructive automatic defect detection method with pixelwise segmentation, Knowl.-Based Syst. 242 (2022) 108338.
[44] Deng Ruoxi, Shen Chunhua, Liu Shengjun, Wang Huibing, Liu Xinru, Learning to predict crisp boundaries, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 562–578.
[45] Fu Xiao, Li Kenli, Liu Jing, Li Keqin, Zeng Zeng, Chen Cen, A two-stage attention aware method for train bearing shed oil inspection based on convolutional neural networks, Neurocomputing 380 (2020) 212–224.
[46] Božič Jakob, Tabernik Domen, Skočaj Danijel, Mixed supervision for surface-defect detection: From weakly to fully supervised learningn, Comput. Ind. 129 (2021) 103459.