The Visual Computer (2023) 39:5451–5467
https://doi.org/10.1007/s00371-022-02671-3

ORIGINAL ARTICLE

A robust defect detection method for syringe scale without positive samples

Xiaodong Wang[1] · Xianwei Xu[1] · Yanli Wang[1] · Pengtao Wu[1] · Fei Yan[1] · Zhiqiang Zeng[1]

Accepted: 7 September 2022 / Published online: 27 September 2022

Abstract
With the worldwide spread of the COVID-19 pandemic, the demand for medical syringes has increased dramatically. Scale defect, one of the most common defects on syringes, has become a major barrier to boosting syringe production. Existing methods for scale defect detection suffer from large volumes of data requirements and the inability to handle diverse and uncertain defects. In this paper, we propose a robust scale defects detection method with only negative samples and favorable detection performance to solve this problem. Different from conventional methods that work in a batch-mode defects detection manner, we propose to locate the defects on syringes with a two-stage framework, which consists of two components, that is, the scale extraction network and the scale defect discriminator. Concretely, the SeNet is first built to utilize the convolutional neural network to extract the main structure of the scale. After that, the scale defect discriminator is designed to detect and label the scale defects. To evaluate the performance of our method, we conduct experiments on one real-world syringe dataset. The competitive results, that is, 99.7% on F1, prove the effectiveness of our method.

Keywords Deep learning · Defect detection · Image processing · Image segmentation

1 Introduction

With the COVID-19 pandemic rampaging the world, millions of people need medical care desperately. Syringes, as the most critical medical devices, are being manufactured in huge quantities. With the acceleration of syringe production, defects inevitably occur during manufacture. As one of the majority marks of the syringe, scale defects are no exception. These defects undoubtedly affect the usage of health care professionals and even are fatal to patients.

In the syringe scale defect detection, the diversity and uncertainty of scale defects is the first matter that should be considered. For the convenience of description, we divide the defects of syringe scales into six classes, that is, “Broken” (BK), “length abnormal” (LA), “excessive” (EX), “width abnormal” (WA), “missing” (MIS), and “other” (OT). To be more specific, the OT refers to the defects that are not belong to the other five categories. These defects are shown in Fig. 1 clearly. Diverse and uncertain defect types make it difficult for conventional methods (*e.g.*, classical machine vision-based methods) to encode or extract accurate features for each special defect type.

The second matter that should be considered is the lack of scale defect data. To collect and label the defective samples in vast syringe products is time-consuming and laborious. The problem becomes even worse when collecting rare defect samples, which seldom appear on syringe scales. The lack of data makes it hard for the machine learning-based method (*e.g.*, CNN) to learn the feature of defects and easy to fall into overfitting. In summary, due to the abovementioned two matters, how to design a robust and discriminative detection method for syringe products is becoming an impending demand.

Currently, the existing methods to detect scales defects are mainly classified into three types, manual detection, classical machine vision-based (CMV) methods, and deep learning (DL) methods. Empirically, all of these methods can be applied to scale defect detection. However, given the pandemic’s environment, the above method faces more limitations. For instance, manual detection methods are easily influenced by the subjective opinions of the detecting personnel, making it hard to achieve a balance of high efficiency and high accuracy [1,2]. Making up for the manual detection

✉ Xianwei Xu
18665596731@163.com

[1] College of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

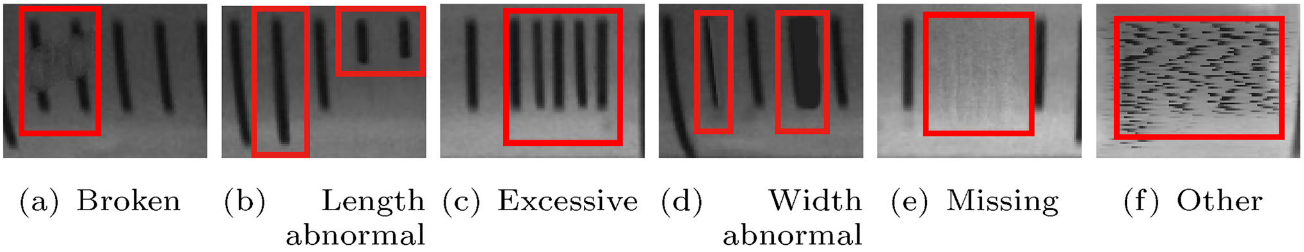


Fig. 1 The demonstration of scale defects. We divide the scale defects into six classes. Sub-image **a** indicates the broken (BK) defect, which may be caused by scale incomplete printing. Sub-image **b** shows the length abnormal (LA) defect, in which the length is not in the range of the standard scale length (*i.e.*, too long or too short). The excessive (EX) defect is shown in sub-image (**c**), meaning excessive scale or spots appear near the normal scales. Similar to LA defects, width abnormal (WA) defects are defined as the width of scales out of the standard width range. Sub-image **d** shows the WA defects clear. Sub-image **e** presents a sample of the missing (MIS) defect, meaning that some scales are absent. The defects which do not belong to the five category defects mentioned above are classified as other defects shown in sub-image (**f**)

method, the CMV method [3,4] can detect defects automatically. It has higher accuracy of detection and is more suitable for large-scale detection. However, most of them depend on handcrafted features, meaning it is challenging to handle the detection task with uncertain defects [5]. With the self-learning mechanism, deep learning detection methods [6–10] are more suitable for the detection of uncertain defects. However, their performance overemphasizes the collection of vast training samples [11,12], which is not suitable for the pandemic's environment.

To solve the above problems, we propose a scale defect detection method without positive samples that can robustly handle uncertain scale defects and requires only small numbers of negative samples. Our method mainly consists of two components:

(i). Scales extraction network (SeNet), a lightweight semantic segmentation network based on the convolution neural network. SeNet is utilized to extract the scale, which can effectively shield the interference of background and noise, and can facilitate the subsequent defect detection.

(ii). Scale defects discriminator (SDD), a scale defect classifier using the scale grouping strategy. Detecting defects on the extracted scale, SDD shows favorable defect detection performance and is suitable for diverse and uncertain detects.

Correctly, our method transforms the defect detection task into two steps, scale structure extraction (*i.e.*, SeNet) and scale defects classification (*i.e.*, SDD). The SeNet employs deep learning technology and training with non-defective samples. This strategy results in the SeNet extracting the non-defective scales and the common scale defects as foreground, while the scales with rare or obvious defects are predicted as background. Utilizing the normal samples for training, our method avoids the difficulty of collecting defect samples as well as the detection accuracy is less affected by defect types. In addition, the SeNet regards each pixel as a training sample, making it require fewer training images (syringe) for training and reducing the risk of model overfitting.

After scale extraction by SeNet, the SDD, which works in the classical machine vision-based manner, is deployed to detect and label the scale defects. Benefitting from SeNet's favorable extraction accuracy, the defect detection on extracted scales requires fewer classification capabilities and computational resources for classifying scales. Therefore, SDD can be designed to work in the classical machine vision-based manner, which accurately classifies defects without positive (defective) samples.

To sum up, the combination of deep learning and classical machine vision-based technologies allows us to detect diverse and uncertain defects without using defect samples. In this paper, we collect and build one real-world dataset including 1205 samples of syringes to evaluate our method. Practical results (*i.e.*, 99.7% on F1) show that our method can meet defect detection requirements of the syringe production.

2 Related work

In the early stage of automatic defect detection, the classical machine vision-based (CMV) methods have made an excellent contribution to the manufacturing industry. In this period, surface defects are generally described as local anomalies inhomogeneous textures [13], in which hand-craft features, such as Local Binary Pattern (LBP), are utilized to determine the existence of defects [3]. Later, traditional machine learning-based methods, such as SVM [14] and fuzzy logic [15], are introduced to the domain of defect detection. Presenting well generalization performance and complex nonlinear boundaries modeling performance, the traditional machine learning-based methods have proven to be powerful for classification tasks. However, it requires complex optimization of the regularization to control the risk of over-fitting [16] and shows poor performance on the defects with complex texture characteristics [17].

In 2012, the proposal of AlexNet [18] triggers a new wave of artificial intelligence. Since then, more and more deep neural networks have been applied to defect detection. Ciregan et al. [19] apply deep neural network technology on image classification and achieve near-human performance on MNIST handwriting benchmark firstly. In the same year, Masci et al. [20] start to apply convolutional neural network (CNN) to defect detection and prove that CNN technology is superior to the CMV method. They build up a five-layer CNN network to classify several types of defects on steel. However, the small network capacity limits its application to complex scenarios. In order to obtain more robust defect detection performance, Weimer et al. [21] build a deep convolutional neural network for defect detection, including the CNN portion and fully connected neural network (FCNN) portion. The CNN portion consists of nine convolutional layers and three max-pooling layers response for generating features, while the FCNN portion consists of three fully-connection layer responses for classification. The method of Weimer et al. enhances the accuracy of defect detection, but the heavy computation burden caused by a large number of parameters and the need for large amounts of training data become new challenges. Based on the idea that different parts of the network are responsible for different tasks, Tabernik et al. [6] propose a two-stage network. The network includes two sub-networks, namely, the segmentation network is responsible for segmenting defects from the raw sample, and the decision network is responsible for deciding whether the defect exists. This method divides the task of defect detection into two parts, which reduces the complexity of each sub-network, alleviating the risk of model over-fitting.

The supervised learning methods have achieved much success in many domains. Nevertheless, their performance overemphasizes the collection of vast training samples [22], particularly defective samples. In this case, they may show poor defect detection performance with few training samples. To relieve the difficulty of collecting and labeling defective samples, some unsupervised defect detection methods based on image reconstruction [23,24] or image reparation [25,26] are proposed. One of the most representative approaches is Autoencoder (AE) [7]. AE builds an encoder-decoder network, in which the encoder structure is in charge of segmenting meaningful information from the input image, namely encoded feature. The decoder structure reconstructs the input image from the encoded features [27]. According to the theory of deep neural network learning via memorization [28], AE will show high reconstruction error between raw data and the reconstruction for defective sample. This framework is of better practical application value due to its needless of defective samples or manual labels [17]

However, the unsupervised defect detection approaches, such as AE, are easy to fall into the false detection of the defects with complex backgrounds [24]. This drawback weakens the defect detection performance of unsupervised methods and limits their application in industry, where the background of the product is usually complex. In addition, unsupervised methods require training on a vast of non-defective samples [24,29], which is time-consuming and computing-intensive. These two drawbacks above limit the application of unsupervised deep learning methods.

In recent years, the combination of the deep learning (DL) method with the classical machine vision-based (CMV) method has gained increasing attention, which can properly balance the prediction accuracy and the data requirement in industrial applications [30]. In this paper, we work on combining the DL method and CMV method to propose a syringe scale defect detection approach. Our approach obtains excellent performance and requires few negative samples for training. Our approach consists of two components, that is, the scale extraction network (SeNet) and the scale defect discriminator (SDD).

3 Scale extraction network (SeNet)

Facing the limitations and requirements above, the intuition of SeNet is that although the defects are diverse and uncertain, the main structure of scales is relatively unchangeable. Therefore, we can extract the normal scales, based on which scale defects can be further detected. To be more specific, we identify the normal scale pixels as the positive pixels, while the other pixels are classified as negative pixels. By applying this extraction strategy, we can achieve the following two advantages: firstly, SeNet treats each pixel on the scale as the training sample, which reduces the requirement of training images collection [6]. Secondly, it helps to detect various and uncertain defects, less affected by the category of defects. This section first introduces the preprocessing techniques used in this paper, that is, syringe location and rectification, followed by our network architecture and loss function.

3.1 Syringe location and rectification

For the computation reduction and external noise avoidance, we introduce the syringe location and syringe rectification as data preprocessing techniques for our SeNet. The syringe location is responsible for locating the syringe of interest (the topmost syringe) in the raw image, shown in Fig. 3. The syringe rectification corrects the tilted angle of the syringe and makes it horizontal. The flowchart of syringe location and rectification is shown in Fig. 2.

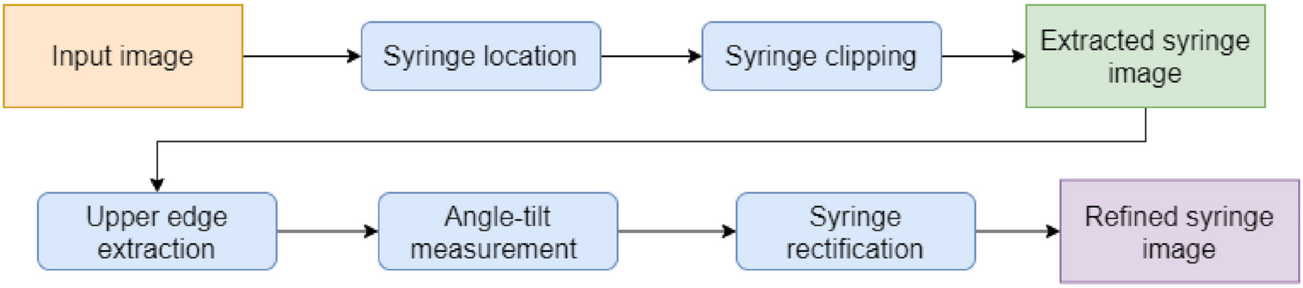


Fig. 2 The flowchart of syringe location and rectification. We first utilize the average gray value changes to locate and extract the syringe. Then, we extract the upper edge of the top syringe to obtain the syringe's tilt angle. Finally, we rectify the extracted syringe horizontally for the subsequent scale extraction task

Syringe location As shown in Fig. 3a, the syringe (green dotted box) has a high pixel value, while the background is represented by relatively low pixel values. Utilizing this characteristic, we can simply construct a sliding window to locate the syringe. For convenience, we denote the input image as I, where $I(x, y)$ indicates the pixel value at the location (x, y) of the raw input image. W_I and L_I refer to the width and the height of the raw image, respectively. Suppose Sw is the sliding window. We use W_S and L_S to represent the width and height of Sw, respectively. To locate the syringe, we initially set the sliding window on the top of the input image. Then, we move it down at a step of ten pixels. For each step, we compute the average pixel value $Sw(k)$ for the sliding windows using Eq. 1, where k ($k \in [1, L_I - L_S]$) is the offset relative to the top of the input image. During the moving process of the sliding window, $Sw(k)$ will stay in relatively low values if the sliding window only covers the background pixels. However, when the sliding window begins to cross the syringe, $Sw(k)$ will gradually increase and reach its largest value when the syringe is entirely covered by the sliding window. Furthermore, after the sliding window leaves the syringe step by step, the $Sw(k)$ also drops. This changing rule of $Sw(k)$ could help us locate the position of the syringe. For better visualization, we illustrate the line chart between $Sw(k)$ and k in Fig. 3b. The corresponding syringe location algorithm is presented in Algorithm 1. It is worth noting that the sliding window only needs to position the topmost syringe and uses a large step size, thus, the sliding window requires a few moving steps and computational resources to locate the syringe.

$$Sw(k) = \frac{1}{W_s L_s} \sum_{x=1}^{W_s} \sum_{y=1}^{L_s} I(x, y + k) \tag{1}$$

Algorithm 1: Syringe Location

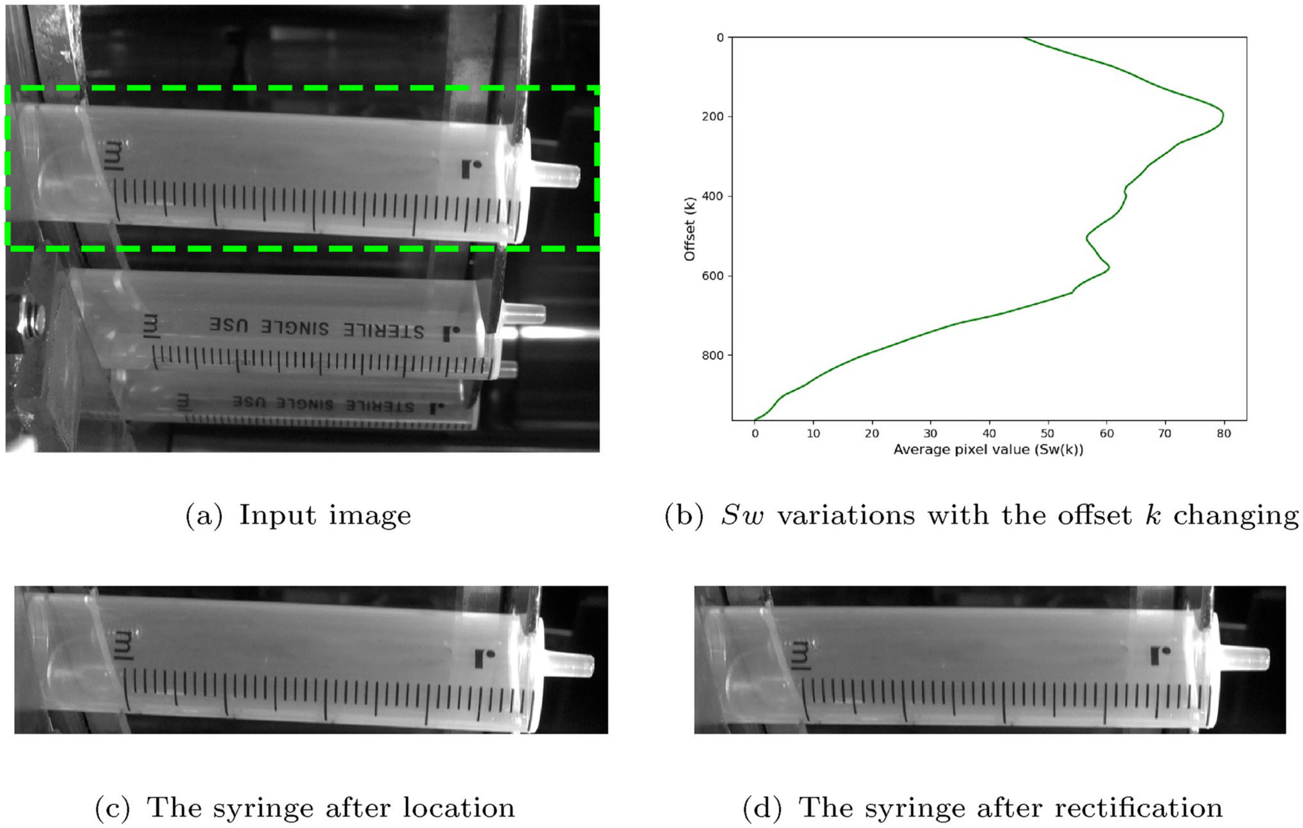
```
Input: The input image $I$, where $I(x, y)$ refers to the pixel value
        of $I$ at the location $(x, y)$
           The width $W_I$ and height $L_I$ of the input image
           The width $W_s$ and height $L_s$ of the sliding window $Sw$
Output: The optimal location of syringe $\hat{k}$
$Sw(0) = 0$ // Initialize the average value in
    $Sw$ to 0
for $k = 1$ to $L_I - L_S$ do
    $Sw(k) = \frac{1}{W_s L_s} \sum_{x=1}^{W_s} \sum_{y=1}^{L_s} I(x, y + k)$ // Calculate
        average pixel value of current sliding
        window
    if $Sw(k) < Sw(k-1)$ then // When the value of
      $Sw(k)$ begins to drop
        $\hat{k} = k$
        break
    end
end
return $\hat{k}$
```

Syringe rectification Owing to the external interference, for example, the vibration of the production equipment, the syringes are frequently out-of-level, like the top syringe in Fig. 3a. To keep extracted scale vertical, for detection convenience, rectifying the syringe is necessary. After obtaining the syringe's location, we clip the syringe from the input image, namely the syringe image. Next, we utilize the Sobel operator to detect the edge of the syringe and then employ Hough transform [31] to obtain the line of the syringe's upper edge. By measuring this line, we can obtain the tilted angle of the syringe. Finally, we rotate the syringe image with the tilted angle and finish the syringe rectification. The result of syringe location and rectification is shown in Fig. 3c, d.

3.2 Our network architecture

As shown in Fig. 1, the shape of the syringe scale is relatively slender as well as the defective scale is similar to the normal one. Therefore, to better detect scale defects, precisely extracting the scale is necessary. To be specific, precise extraction of scale can be divided into two aspects, scale recognition and scale edge extraction. Take a broken scale as an example, scale recognition helps to correctly judge multiple fragments from one scale, while scale edge extraction makes the defects more obvious.

After syringe location and rectification, we build the SeNet to extract the main structure of the scale from the syringe



(a) Input image

(b) Sw variations with the offset k changing

(c) The syringe after location

(d) The syringe after rectification

Fig. 3 The input image and the extracted syringe after location and rectification. In subfigure (**b**), we can find the optimal offset k by locating the largest value of $Sw(k)$, where the sliding window entirely covers the syringe. Then, we can accurately extract the syringe shown in subfigure (**c**). After we extract the syringe, we rectify it horizontally to correct the tilted angle, shown in subfigure (**d**)

image shown in Fig. 3d. Enlighten by Tabernik et al. [6] and Xie et al. [32], the network includes two main compositions, backbone structure and skip connection structure. The network architecture is presented in Fig. 4. The backbone is in charge of extracting the multi-level features from the input image, while the skip connection is responsible for fusing these features. The combination of backbone and skip connection structure ensures the extracted scale contains the structural information from the deep layer of the backbone as well as the details information from the shallow layer of the backbone.

The backbone structure consists of nine convolutional layers with the 3×3 kernel, and two max-pooling layers with the kernel and stride both of 2. Each convolutional layer follows a batch normalization and a ReLU activation function. The batch normalization normalizes each channel to a zero-mean distribution with unit variance, while the ReLU activation function is a nonlinear activation function helping the network learn complex patterns in the data. The join of the convolutional layer, batch normalization, and a ReLU activation function, constitutes a convolutional unit. Totally, there are nine convolutional units in our network, which can be divided into three convolutional groups. Besides, one max-pooling layer is inserted between every two convolutional groups. By selecting the max value in the kernel, the max-pooling layer extracts the noteworthy feature and reduces the feature map resolution at the same time. The segmentation network proposed by Tabernik et al. [6] enlightens the design of our backbone structure, but our backbone with more concise network architecture, less convolutional layers, and fewer numbers of parameters.

The skip connection structures are marked with the red lines and the brown cuboids in Fig. 4. There are three deconvolutional layers plugged in after each convolutional group, and then each result of deconvolutional layers is concatenated to a three-channels feature map. Finally, the three-channel feature map is fused by a convolutional layer with the 1×1 kernel as well as a binarization operation, and outputs a single-channel extracted scale. This special structure enables SeNet to take advantage of both abstract and specialized semantic features from the deep layer and fine-grained features from the shallow layer. These features from each layer are integrated, and various information from different recep-

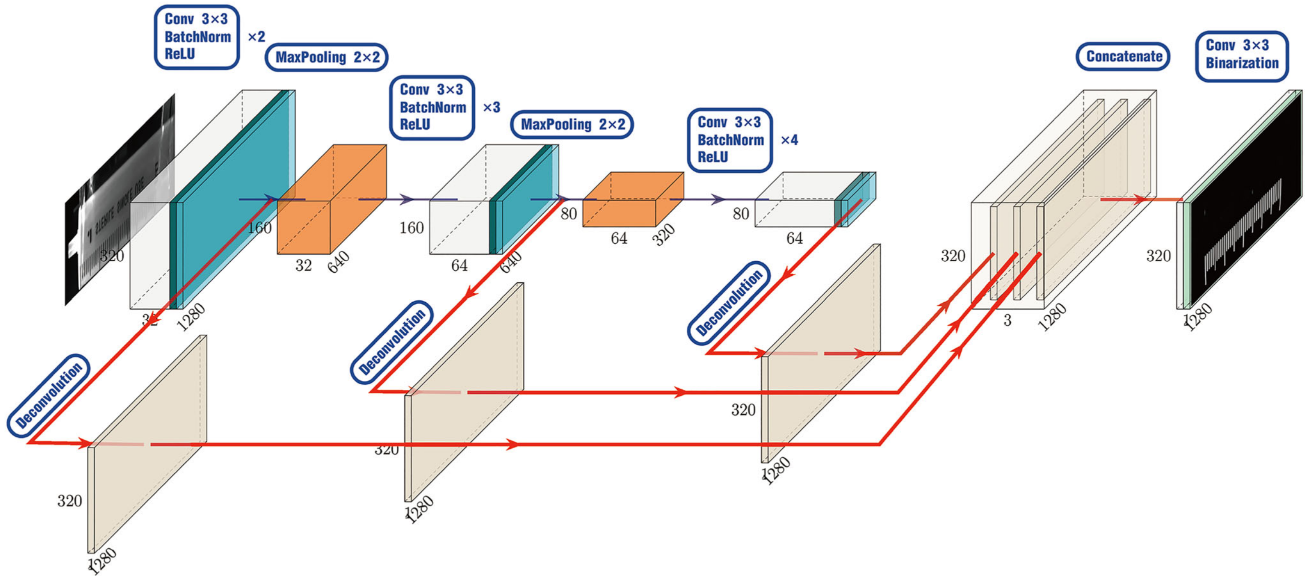


Fig. 4 The structure of SeNet. The red lines and brown cuboids indicate the skip connection structure. Through this structure, we can integrate various features from different receptive fields

tive fields are merged, which helps SeNet to generate an accurate extraction result.

3.3 Loss function

In semantic segmentation, the class imbalance is a common problem [33], where the number of negative (background) pixels is extremely larger than that of positive (foreground) pixels. In this situation, the predictions of pixels are easily biased towards the background and tend to give a false prediction. To address this problem, we adopt the Balanced Cross Entropy (BCE) loss function. By introducing a weighting factor $\alpha \in [0, 1]$ for positive class and $1 - \alpha$ for negative class, the BCE loss function balances the contributions of positive pixels and negative pixels for network training. Formally, BCE is defined as Eq. 2.

$$\mathrm{BCE}(P(x, y), Gt(x, y)) = \begin{cases} -\alpha \log(P(x, y)) & \text{if } Gt(x, y) = 1 \\ -(1 - \alpha)\log(1 - P(x, y)) & \text{otherwise} \end{cases} \tag{2}$$

In the Eq. 2, Gt and P specify the ground-truth label and model's prediction, respectively. $Gt(x, y) \in \{0, 1\}$ and $P(x, y) \in [0, 1]$ indicate the pixel value at the location (x, y) of Gt and P, respectively. In this paper, we use BCE with $\alpha = 0.6$ as the loss function. We analyze the parameter selection on α in detail in Sect. 5.

4 Scale defect discriminator (SDD)

Based on the main structure of the scales (white pixels) shown in Fig. 5, the scale defect discriminator (SDD) is responsible for detecting the defects. Firstly, the extracted scales are grouped to reduce the influence of noise and broken scales. Then, SDD discriminates the defective scale on the scale grouping results.

Notations After extracting the main structure of scales, we select each scale block and use the center coordinates on the x-axis of the scale block to represent its position. Then the scale blocks are sorted in ascending order according to their positions. We use s_0 to represent the scale block with the smallest position value, s_1 to represent the scale block with the second smallest position value, and so on. The set of scale blocks is denoted as $S = \{s_0, s_1, \cdots, s_m\}$, where m refers to the number of scale blocks. Besides, $\mathcal{C}_s(s_i)$ is defined as the position of the scale block s_i. $\mathcal{D}_s(i, j)$ refers to the distance between two scale blocks s_i and s_j, which is formulated as Eq. 3. In addition, we define the average distance of all the scale blocks as $\overline{Ds}$.

$$\mathcal{D}_s(i, j) = |\mathcal{C}_s(s_i) - \mathcal{C}_s(s_j)|. \tag{3}$$

4.1 Scale grouping

In practice, the $\mathcal{D}_s(i, j)$ in Eq. 3 is easy to be affected by noise and broken scales. For example, the broken defect causes the extracted scale to be divided into several parts, making the measurement of the distance biased from the truth values. To solve this problem, we propose a scale grouping strategy to

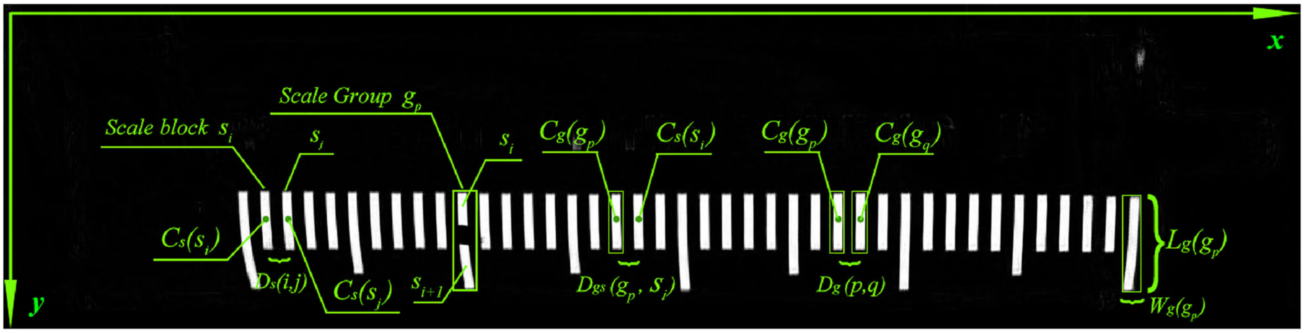


Fig. 5 The scale extraction result of our SeNet. In this figure, we add x-axis (horizontal axis) and y-axis(vertical axis) for better visualization

group the scale blocks, where the scale blocks are put into the same group if they are close to each other.

Similar to scale blocks, we use the center coordinates on the x-axis of the scale group to represent its position. The position of the scale group g_p is set as the center of scale blocks within the scale group. The calculating formula of the scale group position is provided as follows

$$\mathcal{C}_g(g_p) = \frac{1}{\mathcal{N}_g(g_p)} \sum_{s \in g_p} \mathcal{C}_s(s), \tag{4}$$

where $\mathcal{N}_g(g_p)$ is the number of scale blocks in the scale group g_p. $\mathcal{D}_{gs}(g_p, s_i)$ refers to the distance between a scale group g_p and a scale block s_i, formulated as Eq. 5.

$$\mathcal{D}_{gs}(g_p, s_i) = |\mathcal{C}_g(g_p) - \mathcal{C}_s(s_i)|. \tag{5}$$

To group the scale blocks, we first construct an empty scale group g_0 and put the nearest (numbered smallest) non-grouped scale block s_0 to g_0. At that time the position of g_0 same as s_0. Then we measure the distance between the scale group g_0 and the next non-grouped scale block s_1. If $\mathcal{D}_{gs}(g_0, s_1) < \beta \widehat{Ds}$, we add the s_1 to g_0, where $\widehat{Ds}$ is the standard distance between two adjacent scale blocks. Next, we update the position of g_0 using Eq. 4 and add another nearest non-grouped scale block to g_0 until there are no scale blocks close to g_0 (*i.e.*, with distances less than $\beta \widehat{Ds}$). Similar to g_0, we construct and fill the next scale group g_1. This grouping process ends till all the scale blocks are grouped. According to this grouping rule, the procedure of scale grouping is summarized in Algorithm 2.

Algorithm 2: Scales Grouping

```
Input: The set of scale blocks $S = \{s_0, s_1 \cdots s_m\}$
       The standard distance $\widehat{Ds}$
       A hyper-parameter $\beta$
Output: The set of scale group $G$
$p = 0$ // Initialize the index of scale group as 0
$g_p = \{\}$ // Initialize the scale group as an empty group
$G = \{\}$ // Initialize the set of scale group as an empty set
for $s_i \in S$ do // Iterate each scale blocks
    if $s_i$ is $s_0$ then // If it is processing the first scale block
        $g_p = g_p \cup \{s_i\}$ // Add the first scale block to the first scale group
        $G = G \cup \{g_p\}$ // Add the first scale group to the set of the scale group
    else
        $\mathcal{D}_{gs}(p, i) = |\mathcal{C}_g(g_p) - \mathcal{C}_s(s_i)|$ // Calculate the distance between the scale group with the next scale block
        if $\mathcal{D}_{gs}(p, i) > \beta \widehat{Ds}$ then
            $g_{p+1} = \{s_i\}$ // Create a new scale group to save the scale blocks
            $G = G \cup \{g_{p+1}\}$ // Add the new scale group to the scale group set
            $p = p + 1$ // Update $p$
        else
            $g_p = g_p \cup \{s_i\}$ // Add to the last scale group
            $\mathcal{C}_g(g_p) = \frac{1}{\mathcal{N}_g(g_p)} \sum_{s \in g_p} \mathcal{C}_s(s)$ // Update the position of $g_p$
        end
    end
end
return $G$ // Return the scale group set
```

4.2 Scale defect discriminator

Based on the result of scale grouping, we design a scale defect discriminator to perform the defect inspection. Our discriminator aims at distinguishing six defects discussed in Sect. 1. In the subsequent defect detection, we need to compare the input scale group set with a standard scale group set, which is defined as G_T for description convenience. G_T contains N_T standard scale groups which can be divided as c classes according to the scale group length. We denote the standard length of different classes scale group as $L_T^{(t)}$ $(t \in [1, c])$, where t means the class of scale group. In a non-defective syringe, the width of each scale group and the separation between each two scale groups should be basically the same.

We denote the standard width of the scale group as W_T, while D_T refers to the standard distance between two adjacent scale groups. Suppose we obtain a set $G = \{g_1, g_2, \cdots, g_m\}$ containing m scale groups from an input image. Here, our discriminator individually separates each kind of defect as follows.

Firstly, to identify the excessive (EX) defect, missing (MIS) defect, we can simply compare the number of scale groups to that of the standard scale group set. If the number of scale groups N_G is larger than that of the standard scale group set (*i.e.*, $N_G > N_T$), there is an EX defect in the input image. In contrast, $N_G < N_T$ indicates the MIS defect. Further, to detect the broken (BK) defect, we deeply analyze the number of scale blocks in each scale group of G. If there exist some scale groups each containing more than one block, then, the number of scale blocks in G is larger than m, namely, $\sum_{p=1}^{m} N_g(g_p) > m$. In this case, we assume that the input image can be classified as the BK defect.

Secondly, we inspect the separation based on the grouping result. For a non-defective sample, the separation between any two adjacent scale groups should be approximately the same. If the separation between two adjacent scale groups is out of the normal range, we assume that the input image exists OT defects. To be more specific, we define the separation between any two adjacent scale groups g_p and g_{p+1} as $\mathcal{D}_g(p, p+1)$. Besides, the standard separation between g_p and g_{p+1} is defined as D_T and the average separation of all adjacent scales groups is defined as $\bar{D}_g$. When $\mathcal{D}_g(p, p+1)$ is larger than δD_T and $\theta \bar{D}_g$ or $\mathcal{D}_g(p, p+1)$ is smaller than γD_T and $\eta \bar{D}_g$, SDD assumes the separation between two adjacent scale groups g_p and g_{p+1} are out of the normal scope. D_T keeps consistent for each test sample, while $\bar{D}_g$ may change with the different syringe images. Utilizing $\bar{D}_g$ as one of the criteria to determine abnormal separation is equivalent to providing an implicit calibration algorithm for syringes, in which the reasonable range of scale separation change follows with the different test syringes. Using D_T and $\bar{D}_g$ promotes SDD to achieve a more robust performance to the small position shift of the injector. Many defects can cause scale group separation abnormal, such as rare scale defects between two scale groups (*e.g.*, Fig. 1f). In practice, it is difficult to specify all kinds of these defects. Therefore, we classify the defect, leading to the scale group separation abnormal, as other (OT) defect. Besides, the detection abnormality which is not belong to the other five categories, such as failure to detect syringe in the input image or the inexistence of scale (*e.g.*, Fig. 6g), are also classified as other (OT) defect. For description convenience, we define a mapping function $\mathcal{S}(g_p, g_{p+1})$ as follows

$$\mathcal{S}(g_p, g_{p+1}) = \begin{cases} 1 & \text{if } Max(\gamma D_T, \eta \bar{D}_g) < \mathcal{D}_g(p, p+1) < Min(\delta D_T, \theta \bar{D}_g) \\ 0 & otherwise \end{cases} . \tag{6}$$

where γ, δ, η, and θ are the coefficients that determine whether the separation is abnormal, which depend on the syringe type. It should be noticed, if some separations are out of the normal range, the total number of $\mathcal{S}(g_p, g_{p+1})$ with non-zero values is larger than zero, that is, $\sum_{p=1}^{m} \mathcal{S}(g_p, g_{p+1}) > 0$. In this case, we assume that there are uncertain or other (OT) defects between the adjacent scales.

Thirdly, length abnormal (LA) defect and width abnormal (WA) defect are common defects in the syringe. To capture these defects, we compare the length and width of the scale group to that of the standard scale group. In practice, there are multiple scale group lengths in a syringe, similar to the standard scale group lengths $L_T^{(t)}$ $(t \in [1, c])$, we denote the average scale group lengths as $\overline{L}_g^{(t)}$ $(t \in [1, c])$, where t refers to the type of scale group length. To evaluate if the scale group lengths are in the normal range, we use the standard length ratio $\mathcal{R}_{ls}(p) = \mathcal{L}_g(g_p)/L_T^{(t)}$ and average length ratio $\mathcal{R}_{la}(p) = \mathcal{L}_g(g_p)/\overline{L}_g^{(t)}$ as the evaluation indicators. Obviously, $\mathcal{R}_{ls}(p)$ is the ratio between the scale group length and the corresponding standard scale group length, while $\mathcal{R}_{la}(p)$ is the ratio between the scale group length and the average scale group length. Similarly, we define the standard width ratio as $\mathcal{R}_{ws}(p) = \mathcal{W}_g(g_p)/W_T$ and the average width ratio is defined as $\mathcal{R}_{wa}(p) = \mathcal{W}_g(g_p)/\overline{W}_g^{(t)}$.

In practice, the $\mathcal{R}_{ls}(p)$ and $\mathcal{R}_{ws}(p)$ should be within the range of **M**, while the $\mathcal{R}_{la}(p)$ and $\mathcal{R}_{wa}(p)$ should be within the scope of **N**; otherwise, we assume the ratio is out of scope, that is, there are defective scales with LA defects or WA defects.

Finally, the scale defect discriminator is defined as Eq. 7.

$$\mathrm{SDD}(G) = \begin{cases} \text{EX}, & \text{if } N_G > N_T \\ \text{MIS}, & \text{if } N_G < N_T \\ \text{BK}, & \text{if } \sum_{p=1}^{m} \mathcal{N}_g(g_p) > m \\ \text{OT}, & \text{if } \sum_{p=1}^{m} \mathcal{S}(g_p, g_{p+1}) > 0 \text{ or} \\ & \text{other detection abnormality} \\ \text{LA}, & \text{if } \mathcal{R}_{ls} \text{ or } \mathcal{R}_{la} \text{ is out of the normal range} \\ \text{WA}, & \text{if } \mathcal{R}_{ws} \text{ or } \mathcal{R}_{wa} \text{ is out of the normal range} \\ \text{Non-defective}, & \text{otherwise} \end{cases} \tag{7}$$

5 Experiments

5.1 Dataset

In order to verify the performance of our approach, we build a dataset of syringe images, which are captured in a real production environment. We illustrate one sample image in Fig. 3. The resolution of raw input images is 1292 × 964. There are 1205 images in this dataset. Since it is hard to collect the defective samples, we randomly select 590 images from this dataset to artificially generate defective samples. These defective samples can be classified into six categories, shown in Fig. 1. In total, we get 615 non-defective samples and 590 defective samples. In this experiment, we utilize 20 non-defective images for training, five defective samples and five non-defective samples for validation, and the remaining samples (*i.e.*, 585 defective samples and 590 non-defective samples) for testing.

5.2 Experimental setting

The experiment is implemented using the Pytorch framework which is run the hardware of CPU Intel Xeon E5-2680 v3 @ 2.50GHz and GPU Nvidia GTX TITAN X. Two data augmentation strategies, namely, horizontally flipping and vertically flipping, are used in this experiment [34]. The scale extraction network is trained 300 epochs with the mini-batch size of 2. During the training, the learning rate is initialized as 0.01 and decreased by $1/5$ at the epoch 50, 110, 180, and 260, respectively. Adam algorithm [35] is utilized to optimize the parameters.

In the part of SDD, based on the size of the standard syringe, we set the size of sliding windows, namely, W_s and L_s to 1280 and 320, respectively. According to the distance between two standard adjacent scale blocks, we set the ratio β, used to determine whether group the scales blocks, to $1/3$. Similarly, γ, δ, η, and θ, the coefficients to judge separation abnormal, are set as 0.6, 1.7, 0.5, and 2, respectively. Besides, based on the standard scales length and width, the normal range of scale length and width $\mathbf{M}$ and $\mathbf{N}$ is set as [0.8, 1.2] and [0.9, 1.1]. The parameters are shown in Table 1 intuitively.

5.3 Evaluation metrics

To evaluate the performance of our method, four different metrics are considered, that is, Precision, Recall, F1, and Intersection over Union (IOU). Precision, also called positive predictive value, is the fraction of correctly predicted positive samples among all predicted positive samples. The Recall is the ratio between correctly predicted positive samples and ground-truth positive samples. Both Precision and Recall can reflect the defect detection performance of our method. They are defined as Eqs. 8 and 9, respectively. What is more, we also introduce F1, the harmonic mean of Precision and Recall, to take both the precision and recall into account. The special form of F1 is shown in Eq. 10.

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

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

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

TP and FN are defined as the number of correctly predicted positive samples, incorrectly predicted negative samples, respectively. FP is the number of incorrectly classified samples.

IOU is the ratio of intersection and union of prediction and ground truth. It is employed to measure the performance of SeNet, which special form is shown as Eq. 11. In Eq. 11 we treat a pixel as a sample for TP, FP, and FN.

$$\text{IOU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \tag{11}$$

5.4 Defect detection results

Our method contains two components, namely, SeNet and SDD. SeNet extracts the main structure of scale from the raw input image, while SDD detects and labels defects by analyzing the extraction result, that is, the output of SeNet. In this section, we show the defect detection results of our method. The results are shown in Table 2, where the first six rows show the performance of our approach for different defects, while the overall performance is placed at the bottom row.

As shown in the Table 2, our method obtains the overall F1 of 99.7%, which misclassifies four samples out of 585 samples. It should be noticed that we achieve zero misclassification for OT defect, demonstrating the favorable performance of our method for uncertain defects. Besides, the F1 scores of our method on all types of defects are over 99%, showing that our method is robust for syringe defect detection. The overall precision achieved by our approach is 99.5%, while the Recall is 99.8%. This balanced performance in terms of precision and recall demonstrates that our method takes both “find-correct-defects” and “find-all-defects” into account. We show the intuitive result in Sect. 5.6.

5.5 Experiment on SeNet

SeNet is the key to detecting uncertain defects. The concise network structure allows us to use a small number of samples for training model. The introduction of BCE enables us

Table 1 The parameters of training

Item	Value
The standard number of scales	41
The width of sliding window W_s	1280
The length of sliding window L_s	320
The ratio of determine group β	1/3
The coefficients determining separation abnormal γ, δ, η, and $theta$	0.6, 1.7, 0.5, 2
The coefficients judging length abnormal η, θ, λ, μ	0.8, 1.2, 0.9, 1.1

Table 2 The result of defect detection on syringe

Defects	Numbers	TP	FN	FP	Precision (%)	Recall (%)	F1 (%)
BK	100	99	1	1	99.0	99.0	99.0
LA	91	90	1	0	100.0	98.9	99.4
WA	85	85	0	0	100.0	100.0	100.0
EX	88	87	1	0	100.0	98.9	99.4
MIS	94	94	0	0	100.0	100.0	100.0
OT	127	127	0	0	100.0	100.0	100.0
Overall	585	582	3	1	99.8	99.5	99.7

to achieve powerful extraction effects in case of the class imbalance. In this section, we first show the performance of the SeNet and compare it with four state-of-the-art segmentation methods. The compared methods include the U-Net [36], SN [6], SCUNet [37], and FSDNet [38]. Secondly, we explore the number of samples needed for training the network. Finally, by adjusting the α on BCE and replacing BCE with other loss functions, we further analyze the effect of different loss functions on the SeNet.

5.5.1 Scale extraction result

In this section we first present the extracted result of our SeNet in Fig. 6. To clearly compare the extraction results, we artificially generate two representative defects (indicated by the red box and yellow box) for each special defect type on the normal syringe. According to whether the defective scale is similar to the normal one, the defect can be divided into similar scale defects and distinct scale defects. For the former, the extracted results of SeNet are also similar to the normal scales, for example, the results in Fig. 6c, which need to be further processed by subsequent SDD. For the latter, the SeNet extracts the defective scales as background, which can be easily detected, for example, the results in Fig. 6d, g. From the results can see that SeNet can shield the influence of background and noise, and highlights the differences between non-defective and defective scales, thus reducing the difficulty of subsequent detection.

5.5.2 Comparison with other segmentation network

Segmentation performance comparison of different networks To evaluate SeNet objectively, we compare SeNet with four commonly used segmentation networks as follows. U-Net [36], which has been widely applied in various fields. SN [6], which enlightens us to propose SeNet according to the characteristics of the scale. SCUNet [37], a U-Net like segmentation network with depthwise convolution, which slashes the complexity and the size of U-Net sharply. FDSNet [38], which is a novel segmentation network based on a two-stage defect detection framework. The experimental settings of each segmentation network are shown in Table 3. We utilize IOU as the primary evaluation metric, and then we discuss the complexity and size of each network. We run the experiment three times and report the average results. The comparison of extraction accuracy is shown in the first row of Table 4, while the comparison of model complexity and size of all networks are illustrated in the second and third rows of Table 4 respectively.

As shown in the first row of Table 4, SeNet obtains the best IOU of 88.0%, which is significantly superior to SN by 21.2 percentage points and also has a slight advantage (*i.e.*, 1.1 percentage points) over U-Net. From another perspective, the advantages of SeNet are more obvious. Along with better segmentation performance, the network complexity and the size of SeNet are only 1/5 and 1/25 of SCUNet, respectively. SCUNet can be regarded as a variation of U-Net by replacing the convolutional layers with depthwise convolutional layers, which slashes the complexity and the size of U-Net sharply. SCUNet can be divided into two parts: encoder and decoder. SCUNet first encodes the input image

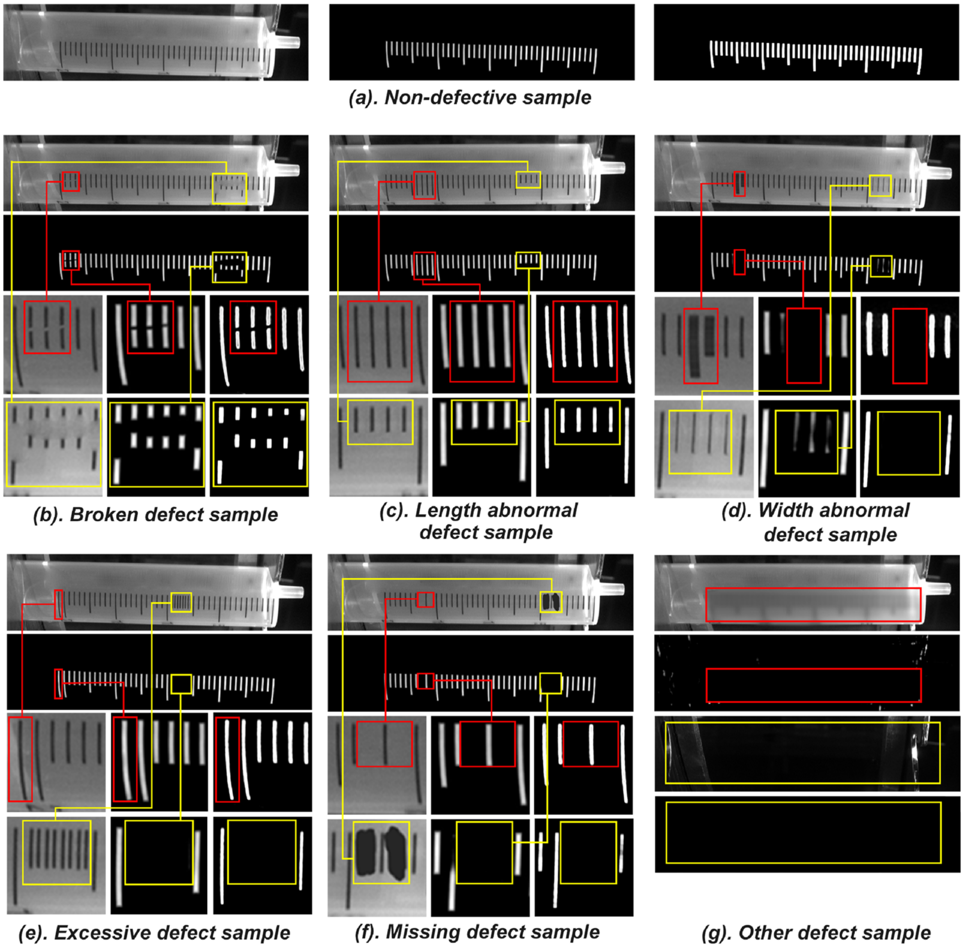

(a). Non-defective sample
(b). Broken defect sample
(c). Length abnormal defect sample
(d). Width abnormal defect sample
(e). Excessive defect sample
(f). Missing defect sample
(g). Other defect sample

Fig. 6 Extracted result of SeNet. For sub-image (**a**), the normal original syringe, the corresponding extraction results of our method, and the manual label results are arranged from left to right. For sub-image (**b**–**f**), the first row of each sub-image shows the defective origin syringe, and the second row presents the scale extraction results. The enlarged defective parts are put in the third and fourth rows, where the origin syringe is on the left, the extraction results of our method are placed in the middle, and the rightmost column is the ground truth mask. Sub-image **g** shows the extraction result for "other" or "uncertain" defect samples. From top to bottom, there are origin syringe, extraction result of our method, and manual label result, respectively

Table 3 The experimental settings of different segmentation methods

Network	Loss	Optimizer	Learning rate	Batch size	Epochs
SN [6]	Cross Entropy	Adam	0.001	2	300
U-Net [36]	Cross Entropy	Adam	0.001	2	300
SCUNet [37]	IOU Loss	Adam	0.01	2	300
FSDNet [38]	Cross Entropy	Adam	0.001	2	300
Ours	BCE	Adam	0.001	2	300

Table 4 The comparison of different segmentation methods

Metrics	SN [6]	SCUNet [37]	FDSNet [38]	U-Net [36]	Ours
IOU	66.8%	82.2%	83.6%	86.9%	88.0%
GFLOPs	142.07	90.40	1284.34	408.70	17.26
Parameters (M)	15.44	6.36	17.54	34.53	0.25

into a small-size feature map through 14 convolutional layers and then decodes the feature map to segmentation result with the same size of the input image through nine convolutional layers and four deconvolutional layers. This symmetric encoder and decoder framework generates a large number of parameters and raises the computational requirements. In contrast, SeNet utilizes an asymmetric encoder and decoder framework, which first encodes the input image into multiscale feature maps through ten convolutional layers, and then decodes these feature maps through one convolutional layer and three deconvolutional layers. This architecture of SeNet contributes to better segmentation accuracy with lower network complexity. The result demonstrates that SeNet is a more excellent method for scale extraction.

Table 5 The detection accuracy comparison of different segmentation networks combined with SDD

Method	IOU (%)	Precison (%)	Recall (%)	F1 (%)
U-Net+SDD	86.9	96.1	96.6	96.3
SeNet+SDD	88.0	99.8	99.5	99.7

Impact of different segmentation networks on SDD In this section, we further discuss the impact of different segmentation networks on SDD. First, we compare the defect detection accuracy of our method (SeNet + SDD) with U-Net [36] combined with SDD (U-Net + SDD). Then we discuss the reasons for the comparison results by showing the extracted scales segment by different networks. The comparison results are shown in Table 5.

As shown in Table 5, higher extraction precision promotes higher detection accuracy, our method exceeds "U-Net + SDD" by 3.4 percentage points in F1. A more intuitive result is shown in Fig. 7.

As shown in Fig. 7a, U-Net can not recognize the small scratches whose results in the SDD identifying the broken scale as the normal one. In contrast, SeNet shows excellent extraction performance for small scratches in Fig. 7b, avoiding the misclassification of SDD. The intuitive results also demonstrate that SeNet is more suitable for scale extraction.

5.5.3 Sensitivity to the number of training samples

In the industrial environment, small training sample size requirements are conducive to reducing the cost of data collection and beneficial for improving the model extensibility for industrial products. In this section, we evaluate our method for scenarios with small-size training sets. Concretely, we randomly select 5, 10, 15 negative samples from our training set in Sect. 5.1 to construct three new training sets TR-5, TR-10, and TR-15, respectively, while the test set and validation set remain unchanged. For clarity, we name our original training set TR-20. The experiment adopts the same training setting as the previous experiment and repeats three times. The average IOU is reported to evaluate the performance of the SeNet. The results are presented in Table 6.

As Table 6 shows, it is no doubt that using the training set with more samples brings better extraction results, that is, SeNet obtains 88.0% IOU when using the training set of TR-20. However, a deserve-attention tendency is that, as the number of training samples decreases, the extraction results still maintain a favorable IOU. For example, when reducing half of the training samples (TR-10), the IOU only decreases by 2.2 percentage points. Even in the extreme case, namely, using five training samples (TR-5), SeNet only decreases 3.3 percentage points on IOU. In summary, the experiment results present that the SeNet maintains superior and stable performance even when a small-scale training set is available. We attribute this favorable characteristic to the non-defective extraction strategy of SeNet, which identifies the defects by extracting the normal scale instead of directly detecting defects. Under this detection framework of SeNet, for any syringe sample containing multiple scales, each of these scales is treated as a training sample. Therefore, SeNet could obtain sufficient training data and accurately explore the main structure of the scale.

Table 6 The comparison of the extraction results with different sizes of the training sets

Training set	Number of training samples	IOU (%)
TR-5	5	84.7
TR-10	10	85.8
TR-15	15	86.7
TR-20 (Origin)	20	88.0

We randomly select 5, 10, 15 negative samples from the original training set to construct training set TR-5, TR-10, TR-15. Especially, for clear demonstration, we denote the original training set as TR-20

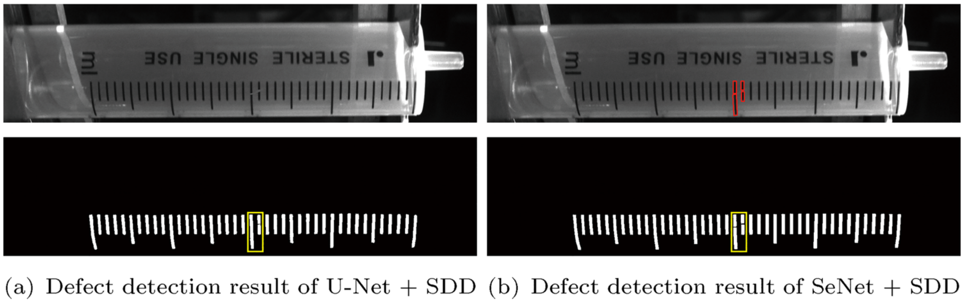
5.5.4 The influence of different loss function

To solve the class imbalance problem in scale extraction, we utilize the Balanced Cross Entropy (BCE) loss function to balance the contributions of positive pixels and negative pixels for network optimization and prevent the model from tending to produce the biased predictions toward the majority class in the training set. In this section, we first discuss the effect of different $\boldsymbol{\alpha}$ on scale extraction accuracy. Then, different loss functions are adopted to compare with BCE.

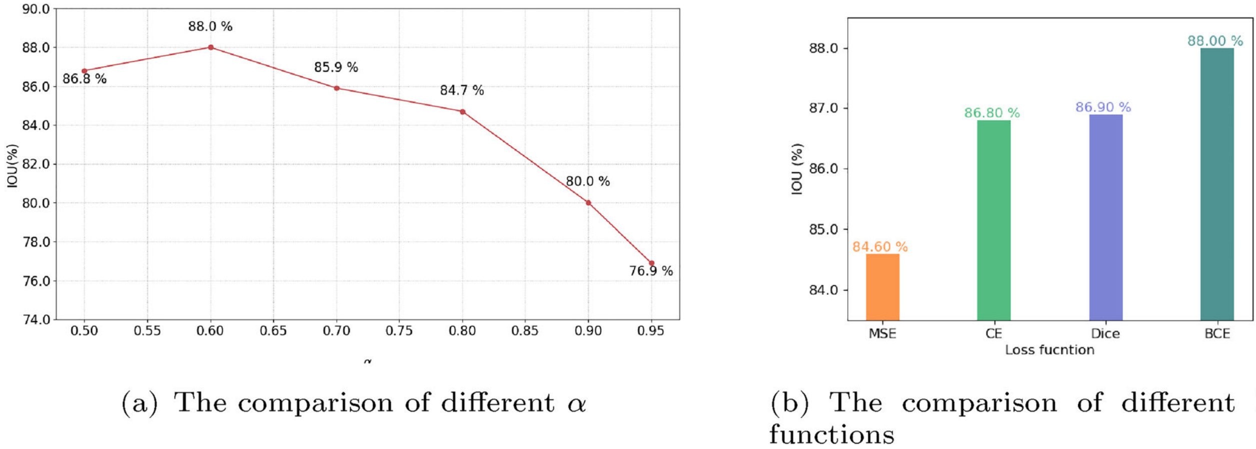
Effect of different $\boldsymbol{\alpha}$ *on extraction accuracy* Conventional BCE loss function uses a weighting factor $\alpha \in [0, 1]$ to balance the contributions of negative pixels and positive pixels. An unsuitable α may lead to an overemphasis on one particular class of pixel and degraded model performance. For example, when α is set to 1, the model will completely ignore the loss from the negative pixels and tend to predict all pixels as positive. In contrast, when α is set to 0, the model tends to predict all pixels as background. To find an appropriate α for the BCE loss function, we tune it from $\{0.6, 0.7, 0.8, 0.9, 0.95\}$ on the training set and validation set. Following the previous experiment, we repeat the experiment three times and report the average IOU. The results are shown in Fig. 8a.

Shown in Fig. 8a, SeNet obtains the best IOU of 88.0% when $\alpha = 0.6$. This demonstrates that slightly increasing the weight of positive pixel loss is beneficial to enhance the accuracy of the SeNet. However, overemphasis on the loss from positive pixels will weaken the model. For example,

Fig. 7 Defect detection results of different segmentation networks combined with SDD. In each subfigure, we show the defect detection results of SDD on the first row, while the extracted scales are shown in the second row. In the first row, the red boxes are marked by the algorithm, and we mark the defect manually with yellow boxes in the second row



(a) Defect detection result of U-Net + SDD (b) Defect detection result of SeNet + SDD



(a) The comparison of different α

(b) The comparison of different loss functions

Fig. 8 Performance comparison of different α and loss functions

when α is gradually increased from 0.6 to 0.95, SeNet shows a degradation tendency. This result indicates that α is important to SeNet, and an appropriate α could properly improve the accuracy of SeNet.

Performance comparison of different loss function To address the class imbalance problem, BCE is adopted to balance the positive and negative pixels. In this section, to verify the effect of BCE, we replace loss function with Cross Entropy loss function (CE), Mean Square Error loss function (MSE), and Dice loss function (Dice) for the experiment. The comparison results of different loss functions are shown in Fig. 8b.

From Fig. 8b, BCE loss could improve the extraction accuracy of the SeNet indeed. Compared with the imbalance-ignorant loss function, for example, MSE loss, our approach obtains an elevation of 3.4 percentage points. In comparison with MSE loss, the imbalance-aware loss functions, for example, Dice loss and BCE, are less affected by the class imbalance problem [39], achieving superior performance. Furthermore, when compared with the Dice loss, our approach has an enhancement of nearly 1.1 percentage points. The reason may lie in that Dice loss suffers from the gradient unstable problem, which may cause difficulty for model training [40]. These results demonstrate that using a loss function that can refine the class imbalance problem is meaningful to the syringe defect detection task, and BCE is more effective than the other compared loss functions.

Table 7 The detection accuracy comparison of different methods

Method	Precison (%)	Recall (%)	F1 (%)
SCCAE [41]	69.90	62.70	166.1
MSCDAE [25]	94.1	65.1	77.0
SDD	99.8	99.5	99.7

5.6 Experiment on SDD

Defect discriminator is responsible for detecting and labeling defects on the extracted scale. In this section, we first show the labeled result of the defect discriminator. Then, an ablation experiment is implemented to quantify the effect of scale grouping on the accuracy of defect detection.

5.6.1 Comparison with other defect detection methods

Syringe scale defect detection is a special task in machine vision applications. First, there are vastly diverse and uncertain defect types of scale defects ("other defect"), such as Fig. 1f. Secondly, the defective scale is similar to the normal one as well as position relevant. For example, in Fig. 1b the defective scale (surrounded by the red box) is almost the same as the non-defective one on the left. Therefore, it is difficult for us to find a suitable classical machine vision-based method to compare with our method. In addition, limited by the small training set, which contains only 20 non-defective samples, a lot of machine learning methods are unsuitable for detecting scale defects. In this section, we compare our method with MSCDAE proposed by Mei et al. [25] and SCCAE proposed by Xu et al. [41]. MSCDAE utilizes a denoising autoencoder and the Gaussian pyramid to reconstruct image patches, considering images with poor reconstruction as defects. With a similar architecture to MSCDAE, SCCAE constructs a convolutional auto-encoder with skip connections to rebuild the input image and considers the incorrect rebuilding regions as defects. We utilize the MSCDAE and SCCAE to reconstruct and detect defects on the extracted scale, which reduces the difficulty of reconstruction and releases interferences from the background. Benefiting from the Gaussian pyramid, MSCDAE is an unsupervised method suitable for the small training set [25]. However, due to the limitation of training samples, SCCAE obtains an inferior result. The results are shown in Table 7.

As shown in the third column of Table 7, our method achieves F1 of 99.7%, far ahead of MSCDAE and SCCAE by 12.7 and 33.6 percentage points. This result illustrates that our method is more suitable for defect detection on syringe scales.

5.6.2 The visual result of SDD

After the process of SeNet, the distinct scale defects, such as the first row of Fig. 9f, are predicted as background, while the similar scale defects, such as the first row of Fig. 9b, are extracted as foreground. Scale defect discriminator (SDD) is responsible for further processing of the extracted result and detecting the scales defect. We show diverse kinds of scale defects (the first row) and the detected result (the last row)of our method in Fig. 9.

Shown in Fig. 9, SDD presents a favorable ability to detect both distinct scale defects and similar scale defects. Concretely, for the distinct scale defect, such as the OT defect, shown in Fig. 9f, our method can mark the defect target accurately and is less affected by the category of defects. For the similar scale defect, like the broken defect, shown in Fig. 9a, SDD can identify all the parts of defected scales by scale grouping. These results demonstrate that our method not only presents robust to diverse and uncertain defects but also has the ability to accurately mark defect.

5.6.3 The effect of scale grouping

Scale grouping is used to group scale blocks and detect broken defects. To verify the effectiveness of scale grouping, we implement the ablation experiment on it. The result is shown in Table 8. Clearly shown in the third and fourth rows of the table when we adopt scale grouping in SDD, our method obtains a better F1, that is, 6.7 percentage points improvement over the result without scale grouping (the fifth and

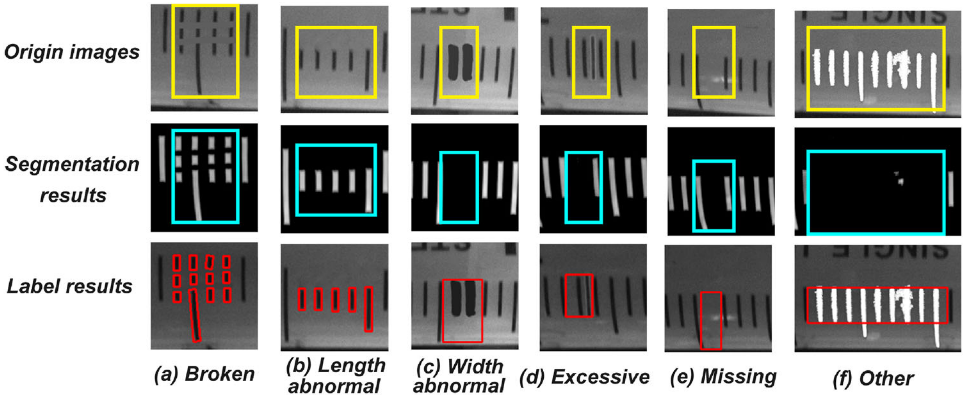


Fig. 9 The detecting results by our method. In the first row, the yellow box indicates the defect on the original images, while the blue box in the middle row marks the corresponding extracted defects or segmentation results. The red box, in the bottom row, shows the mark detected by our SDD

Table 8 Result of scale grouping ablation experiment. To verify the effect of scale grouping in SDD, we implement an ablation experiment on the test set with 585 defective samples

Grouping		Error items						Recall (%)	Precision (%)	F1 (%)
		BK	LA	WA	EX	MIS	OT			
✓	FN	1	1	1	0	0	0	99.5	99.8	99.7
	FP	1	0	0	0	0	0			
×	FN	71	1	0	2	3	0	100	86.8	93.0
	FP	0	0	0	0	0	0			

sixth rows). Concretely, adopting scale grouping drastically reduces the number of broken defect error items from 71 to 2. Besides, scale grouping also promotes the detection accuracy for EX and MIS defects. These improvements illustrate the importance of scale grouping for SDD.

6 Conclusion

In this paper, we propose a scale extraction neural network with the BCE loss function (SeNet) and a scale defect discriminator (SDD) to detect the defects for syringes. Utilizing a small number of negative samples to train the model, our SeNet can extract the main structure of scales effectively. After dividing the extracted scales into groups, SDD inspects these scale groups by analyzing the correlations among different groups to identify the defects. To verify our method, we build a real-world syringe scale dataset with 1205 samples and apply our method to it. We reach the F1 of 99.7% for image-level prediction, while the IOU of 88.0% for pixel-level prediction. In summary, our method shows the combination of deep learning (our deep neural network, *i.e.*, SeNet) and the conventional machine vision-based method (our scale defect discriminator, *i.e.*, SDD) is important for syringe defects detection, especially in the scenarios with many defect uncertainty and limited negative samples.

Acknowledgements This paper was supported by National Natural Science Foundation of China (Grant Nos. 61871464, U1805264), National Natural Science Foundation of Fujian Province (Grant Nos. 2020J01266, 2021J011186), the "Climbing" Program of XMUT (Grant No. XPDKT20031), Scientific Research Fund of Fujian Provincial Education Department (Grant No. JAT200486), the University Industry Research Fund of Xiamen (Grant No. 2022CXY0416).

Data availability The datasets generated during and analyzed during the current study are not publicly available due to the data also forming part of an ongoing study, but are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Chen, J., Liu, Z., Wang, H., Núñez, A., Han, Z.: Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network. IEEE Trans. Instrum. Meas. **67**(2), 257–269 (2018). https://doi.org/10.1109/TIM.2017.2775345
2. Liu, G., Zheng, X.: Fabric defect detection based on information entropy and frequency domain saliency. Vis. Comput. (2021). https://doi.org/10.1007/s00371-020-01820-w
3. Bulnes, F.G., Usamentiaga, R., Garcia, D.F., Molleda, J.: An efficient method for defect detection during the manufacturing of web materials. J. Intell. Manuf. **27**(2), 431–445 (2016). https://doi.org/10.1007/s10845-014-0876-9
4. Zhang, X., Chen, Y., Hong, H.: Pavement crack detection based on texture feature. Proc. SPIE Int. Soc. Opt. Eng. **8003**, 11 (2011)
5. Wang, Q., Meng, X., Sun, T., Zhang, X.: A light iris segmentation network. Vis. Comput. (2021). https://doi.org/10.1007/s00371-021-02134-1
6. Tabernik, D., Šela, S., Skvarč, J., Skočaj, D.: Segmentation-based deep-learning approach for surface-defect detection. J. Intell. Manuf. **31**(3), 759–776 (2020). https://doi.org/10.1007/s10845-019-01476-x
7. Tian, H., Li, F.: Autoencoder-based fabric defect detection with cross-patch similarity. In: Proceedings of the 16th International Conference on Machine Vision Applications, MVA 2019 (2019). https://doi.org/10.23919/MVA.2019.8758051
8. Napoletano, P., Piccoli, F., Schettini, R.: Anomaly detection in nanofibrous materials by CNN-based self-similarity. Sensors (Switzerland) (2018). https://doi.org/10.3390/s18010209
9. Hu, W., Wang, T., Wang, Y., Chen, Z., Huang, G.: Le-msfe-ddnet: a defect detection network based on low-light enhancement and multi-scale feature extraction. Vis. Comput. (2021). https://doi.org/10.1007/s00371-021-02210-6
10. Huang, Y., Qiu, C., Guo, Y., Wang, X., Yuan, K.: Surface defect saliency of magnetic tile. IEEE Trans. Autom. Sci. Eng. (2020). https://doi.org/10.1007/s00371-018-1588-5
11. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. Int. J. Comput. Vis. (2021). https://doi.org/10.1007/s11263-020-01395-y
12. Wu, W., Zhang, S., Tian, M., Tan, D., Wu, X., Wan, Y.: Learning to detect soft shadow from limited data. Vis. Comput. (2021). https://doi.org/10.1007/s00371-021-02095-5
13. Xu, L., Lv, S., Deng, Y., Li, X.: A weakly supervised surface defect detection based on convolutional neural network. IEEE Access **8**, 42285–42296 (2020). https://doi.org/10.1109/ACCESS.2020.2977821
14. Fern, C., Garc, I.: Surface Classification for Road Distress, pp. 600–607 (2012)
15. Xu, L., He, X., Li, X., Pan, M.: A machine-vision inspection system for conveying attitudes of columnar objects in packing processes. Meas. J. Int. Meas. Confed. **87**, 255–273 (2016). https://doi.org/10.1016/j.measurement.2016.02.048
16. Devos, O., Ruckebusch, C., Durand, A., Duponchel, L., Huvenne, J.-P.: Support vector machines (SVM) in near infrared (NIR) spectroscopy: focus on parameters optimization and model interpretation. Chemom. Intell. Lab. Syst. **96**(1), 27–33 (2009). https://doi.org/10.1016/j.chemolab.2008.11.005
17. Luo, Q., Fang, X., Liu, L., Yang, C., Sun, Y.: Automated visual defect detection for flat steel surface: a survey. IEEE Trans. Instrum. Meas. (2020). https://doi.org/10.1109/TIM.2019.2963555
18. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inform. Process. Syst. **25**(2) (2012)
19. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3642–3649 (2012). https://doi.org/10.1109/CVPR.2012.6248110
20. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., Fricout, G.: Steel defect classification with max-pooling convolutional neural networks. In: Proceedings of the International Joint Conference on Neural Networks, pp. 10–15 (2012). https://doi.org/10.1109/IJCNN.2012.6252468
21. Weimer, D., Scholz-Reiter, B., Shpitalni, M.: Design of deep convolutional neural network architectures for automated feature

extraction in industrial inspection. CIRP Ann. Manuf. Technol. **65**(1), 417–420 (2016). https://doi.org/10.1016/j.cirp.2016.04.072

22. Li, K., Jin, Y., Akram, M., Han, R., Chen, J.: Facial expression recognition with convolutional neural networks via a new face cropping and rotation strategy. Vis. Comput. (2020). https://doi.org/10.1007/s00371-019-01627-4
23. Dai, W., Erdt, M., Sourin, A.: Self-supervised pairing image clustering for automated quality control. Vis. Comput. (2022). https://doi.org/10.1007/s00371-021-02137-y
24. Zhao, Z., Li, B., Dong, R., Zhao, P.: A surface defect detection method based on positive samples. In: Geng, X., Kang, B.-H. (eds.) PRICAI 2018: Trends in Artificial Intelligence, pp. 473–481. Springer, Cham (2018)
25. Mei, S., Yang, H., Yin, Z.: An unsupervised-learning-based approach for automated defect inspection on textured surfaces. IEEE Trans. Instrum. Meas. (2018). https://doi.org/10.1109/TIM.2018.2795178
26. Dai, W., Erdt, M., Sourin, A.: Detection and segmentation of image anomalies based on unsupervised defect reparation. Vis. Comput. (2021). https://doi.org/10.1007/s00371-021-02257-5
27. Wang, D., Hu, G., Lyu, C.: Frnet: an end-to-end feature refinement neural network for medical image segmentation. Vis. Comput. (2021). https://doi.org/10.1007/s00371-020-01855-z
28. Zhang, C., BeNgio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: International Conference on Learning Representation (2016)
29. Zheng, Z., Zheng, L., Yang, Y.: Unlabeled samples generated by gan improve the person re-identification baseline in vitro (2017)
30. Tu, K., Wen, S., Cheng, Y., Zhang, T., Pan, T., Wang, J., Wang, J., Sun, Q.: A non-destructive and highly efficient model for detecting the genuineness of maize variety 'JINGKE 968' using machine vision combined with deep learning. Comput. Electron. Agric. **182**(February), 106002 (2021). https://doi.org/10.1016/j.compag.2021.106002
31. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1), 11–15 (1972)
32. Xie, S., Tu, Z.: Holistically-nested edge detection. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1395–1403 (2015). https://doi.org/10.1109/ICCV.2015.164
33. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. **42**(2), 318–327 (2020). https://doi.org/10.1109/TPAMI.2018.2858826
34. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34 (2017). https://doi.org/10.1609/aaai.v34i07.7000
35. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. Comput. Sci. (2014)
36. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional Networks for Biomedical Image Segmentation. Springer (2015)
37. Cheng, L., Yi, J., Chen, A., Zhang, Y.: Fabric defect detection based on separate convolutional UNet. Multimed. Tools Appl. (2022). https://doi.org/10.1007/s11042-022-13568-7
38. Huang, Y., Jing, J., Wang, Z.: Fabric defect segmentation method based on deep learning. IEEE Trans. Instrum. Meas. (2021). https://doi.org/10.1109/TIM.2020.3047190
39. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: fully convolutional neural networks for volumetric medical image segmentation (2016)
40. Yeung, M., Sala, E., Schönlieb, C.-B., Rundo, L.: Unified focal loss: generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. Comput. Med. Imaging Graph. **95**, 102026 (2022). https://doi.org/10.1016/j.compmedimag.2021.102026
41. Xu, H., Huang, Z.: Annotation-free defect detection for glasses based on convolutional auto-encoder with skip connections. Mater. Lett. **299**, 130065 (2021). https://doi.org/10.1016/j.matlet.2021.130065

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

Xiaodong Wang received his B.E and M.E. degree from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China in 2007 and 2010, respectively. He received his Ph.D. degree in Informatics from Chaoyang University of Technology, Taiwan, in 2019. He is currently an associate professor with the College of Computer and Information Engineering, Xiamen University of Technology. His research interests cover pattern recognition, image processing, and deep learning.

Xianwei Xu received his B.E. degree in Electronic and Information Engineering from Guangzhou University in 2018. He is currently pursuing the M.E in Electronic and Information with the College of Computer and Information Engineering, Xiamen University of Technology. His research interests include computer vision and deep learning, in particular, defect detection and semantic segmentation.

Yanli Wang received her B.E. degree in Computer Science and Technology from Zhengzhou University of Aeronautics, Zhengzhou, China, in 2019. She is currently pursuing the M.E in Electronic and Information with the College of Computer and Information Engineering, Xiamen University of Technology. Her research interests cover pattern recognition, computer vision, and deep learning.

Pengtao Wu received the B.E. degree from the College of Computer and Information Engineering, Xiamen University of Technology, in 2021. He is currently pursuing the master degree in Electronic and Information with the School of Xiamen University of Technology. His research interests include machine learning, pattern recognition, and deep learning.

Fei Yan received her B.E. degree from the College of Technology from Hunan Normal University, Changsha, China, in 2007 and received her M.E. degree from the College of Computer Science and Electronic Engineering from Hunan University, China, in 2010. She is currently a senior lab master in the College of Computer and Information Engineering, Xiamen University of Technology. Her research interests include pattern recognition and data hiding.

Zhiqiang Zeng received the M.Sc. and Ph.D. degrees in Computer Science from the Xi'an Jiaotong University and Zhejiang University, China, in 2004 and 2007, respectively. He holds a Bachelor's degree in Automation from Sichuan University, Chengdu, China, in 1994. In 2008, he joined the Computer Science Department of the Xiamen University of Technology as a research associate. His interests include pattern recognition and machine learning, in particular, support vector machines and general kernel methods.