Generated Pseudo-labels Guided by Background Skeletons for Overcoming Under-segmentation in Overlapping Particle Objects

Jinshi Liu, *Graduate Student Member, IEEE*, Zhaohui Jiang, Ting Cao, *Member, IEEE*, Zhiwen Chen, *Member, IEEE*, Chaobo Zhang, *Member, IEEE*, and Weihua Gui, *Member, IEEE*

***Abstract*—Unlike general image segmentation, highly complex particle images have significant challenges in labeling and segmentation due to the information occlusion and texture disturbance. Aiming at the highly under-segmentation problem caused by complex particle image segmentation, this paper proposes a Semi-supervised Hybrid-training Particle Segmentation framework (SHPS) based on skeleton-guided pseudo-labels. First, a pre-trained model is obtained by training a popular segmentation algorithm on partially labeled data. Then, a Background Skeleton-guided Pseudo-label generation algorithm (BSP) is proposed to generate pseudo-labels closer to the ground truth in terms of structural integrity based on coarse segmentation. The final segmentation model is obtained by training a mixed dataset consisting of labeled data and pseudo-labels from another partition on the pre-trained model. The skeleton differences of pseudo-labels and coarse segmentation are added to the loss function. Experimental results show that our method achieves 84.4% accuracy on mIoU with uniform label data distribution, which is 2.1% higher than the accuracy of UNet and reduces the degree of under-segmentation.**

***Index Terms*—Image segmentation, semi-supervised, overlapping particle, pseudo-label, skeleton, leaf vertices.**

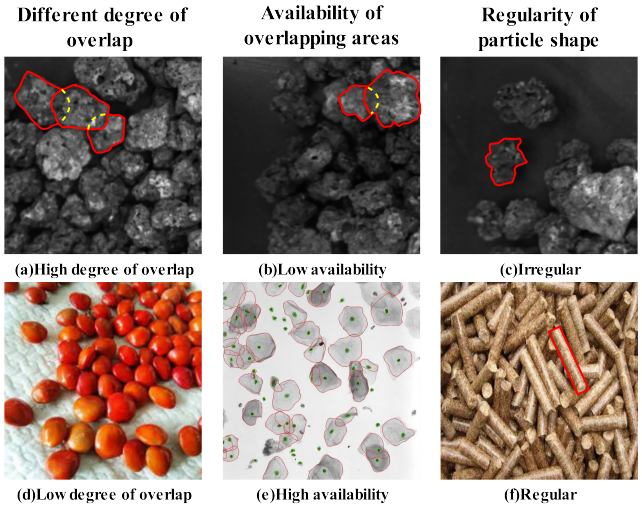


Fig. 1. The main factors affecting the difficulty of particle image segmentation. The dashed yellow line represents the true occluded edge.

I. INTRODUCTION

A particle is a kind of small object in nature that appears as a group. In current research applications, particles generally refer to solid particles, droplets, or bubbles with a specific shape, such as seeds, ores, and cells. Particles' size, quantity, and shape play an essential role in some practical application studies. Many scholars from different fields have conducted massive research on particles, such as ore particle size detection in industry, cell image segmentation in medicine, and leaf segmentation in agronomy [**?**], [**?**], [**?**]. Notably, image segmentation is one of the most widespread methods in particle analysis.

In computer vision, particle segmentation is a challenging and characteristic applied task. Unlike the generic semantic segmentation task, particle segmentation is a series of segmentation tasks for particle objects [**?**], [**?**]. Different particle objects bring different difficulties and challenges to the segmentation task. The more complex the structure of particles, the worse the segmentation effect is. In addition to the algorithm, the main reason for a lousy segmentation is the high complexity of the particles themselves, including their complex spatial distribution and surface texture. The complexity of particles mainly includes shape complexity, particle occlusion ratio (POR), and the availability of occlusion area information, as shown in Fig. 1. Therefore, this article focuses on the most complicated situation, i.e., irregular overlapping occlusion particles (IOOP).

When a generic segmentation algorithm deals with the most complex particles, the segmentation accuracy is low because there is massive under-segmentation [**?**], as shown in Fig. 2. Although under-segmentation and over-segmentation are both evaluation indices describing the difference in the label, the former has a more significant impact on the segmentation results for particle images than the latter because of its powerful destructive effect on particle structure. Specifically, the extra pixels generated in over-segmentation only form erroneous branches of different lengths instead of

This work was supported by the National Major Scientific Research Equipment of China (No. 61927803), National Natural Science Foundation of China (No. 62103206) and the China Postdoctoral Science Foundation (No. 2021M701804). (Corresponding author: Zhaohui Jiang).

Jinshi Liu, Zhaohui Jiang, Zhiwen Chen, and Weihua Gui are with the School of Automation, Central South University, Changsha 410083, China. Zhaohui Jiang, Ting Cao, Zhiwen Chen, Chaobo Zhang and Weihua Gui are with the Pengcheng Laboratory, Shenzhen 518055, China. (e-mail: ljs11528@csu.edu.cn; jzh0903@csu.edu.cn; caot@pcl.ac.cn; zhiwen.chen@csu.edu.cn; zhangchb@pcl.ac.cn; gwh@csu.edu.cn.)

an additional closed connection domain, which has an almost negligible impact on the initially closed particle structure. However, the lack of some critical pixels resulting from under-segmentation has a fatal impact on the structure of a closed-connected domain, which will directly merge multiple initially closed-connected domains into one, as shown in Fig. 2. In addition, high-complexity particles also bring great difficulties to labeling, resulting in the amount of labeling data not being enough to support the complete training of a model. Compared to generic image segmentation, there are two main challenges with particle image segmentation (PIS).

Difficult to maintain structural integrity. Compared with the overall segmentation accuracy, PIS often requires more attention to the particle structural integrity. Massive under-segmentation in the segmentation results obtained with generic algorithms can cause devastating damage to the particle structural integrity.

Insufficient label data. Due to particle texture and distribution complexity, the cost and difficulty of annotation for PIS are high. Thus there is often a shortage of labeled data.

Motivated by this, this paper proposes a semi-supervised segmentation framework based on pseudo-labels guided by skeletons to reduce under-segmentation in particle images in the case of limited label data. The main contributions of our work are as follows:

(1) A hybrid-training framework for semi-supervised particle segmentation based on pseudo-labels is proposed. The proposed framework consists of a pre-trained segmentation model, hybrid-training with pseudo-labels, and fine-tuning. The optimization of pseudo-labels is guided first by learning the performance of labeled data in the pre-trained process. Then the final model is obtained by mixing training and fine-tuning the pre-trained model. Our framework can obtain better application results on the particle segmentation problem than existing self-training frameworks.

(2) A background skeleton-guided pseudo-label generation algorithm is proposed. The optimization tendency of pseudo-labels is first implicitly modeled based on the background skeleton and foreground target's correlation and geometric features. And then, the critical parameters learned from the pre-training process with labels are used to generate pseudo-labels with better performance in particle structural integrity and higher overall accuracy by segmenting again.

(3) A dual-segmentation framework to overcome under-segmentation is proposed for particle objects. We designed a re-segmentation algorithm workflow framework with skeleton extraction, feature matching, and cluster analysis as the core. The framework performs secondary segmentation based on coarse segmentation to obtain pseudo-labels and segmentation results with higher accuracy and less under-segmentation.

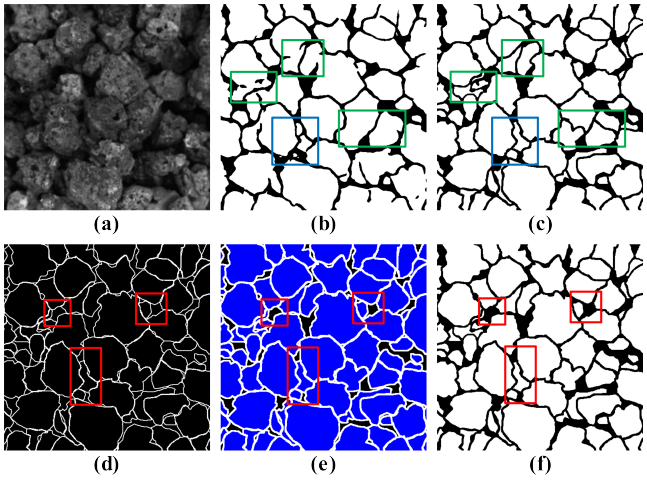


Fig. 2. Different performance of under-segmentation and over-segmentation with different classification targets in particle image segmentation. (a)RGB image; (b)two-classification(particle,non-particle); (c)Ground truth; (d)two-classification(edge,background); (e)three-classification; (f)Ground truth. The blue boxes represent over-segmentation, and the green boxes represent under-segmentation. The red boxes indicate the position that the binary classification cannot distinguish whether it is a particle or gap.

II. RELATED WORK

A. Semantic Segmentation

Some semantic segmentation methods were proposed in early research to solve PIS. Ning et al. proposed an image segmentation method of adhesion rock particles based on Harris corner detection and concave point matching [**?**], which is only suitable for low POR. In addition, Lu et al. separated the overlapping cells by using the joint optimization of multiple level set functions and constraints based on the overlapping area [**?**]. Zafari et al. estimated the contour of objects by extracting seed points and ellipse fitting, but only for regular particles [**?**]. These methods are suitable for particular particle images with either information on an available overlap area or regular shapes.

Singh et al. used UNet [**?**] and SegNet [**?**] to segment the river ice images in their experiments [**?**]. In addition, Tao et al. proposed an entire network structure that consists of a basic U-Net module and multi-frequency module (MFM), which enhances multi-frequency feature fusion of conductive particles and accelerates network training [**?**]. Furthermore, Hou et al. established a particle inspection model based on self-supervised convolutional neural networks (CNNs) and transfer learning [**?**]. The above methods inevitably produce some overlapping particles that have not been separated.

In addition to semantic segmentation, some scholars have also proposed some instance segmentation methods, which perform well in specific scenarios. Because most instance segmentation methods include object detection that generates independent particles, most methods based on instance segmentation generate fewer overlapping particles than those based on semantic segmentation [**?**], [**?**], [**?**]. However, instance segmentation relies heavily on the training data quality, which can only detect fewer particles, increasing the number of targets and POR increase in the sample.

In conclusion, these methods that work well in other domains are unsuitable for complex objects such as IOOPs. This paper focuses on accurately segmenting IOOP images with limited training data while reducing under-

segmentation.

B. Semi-supervised Semantic Segmentation

Some semi-supervised semantic segmentation methods were proposed with limited training data. Lee et al. trained the network by constructing pseudo-labels, which were used as if they were actual labels [**?**]. Later, some methods generated additional training data via GAN-based models, but these data had difficulty meeting the demand for accuracy [**?**], [**?**], [**?**]. In addition, Zou et al. obtained well-calibrated soft pseudo labels via intelligent fusion of predictions from diverse sources [**?**]. Some iterative self-training approaches were proposed to train a good teacher model with the labeled data that generates pseudo labels for a student model [**?**], [**?**], [**?**], [**?**]. However, the pseudo-labels of particle images generated by these methods are incomplete in the particle structure, resulting in massive under-segmentation. Recently, some scholars have used modeling methods to find the structural relationship between samples and labels [**?**], [**?**], [**?**]. However, the difference between samples and labels of particle images is often insignificant, and it is not easy to describe via modeling.

In addition, some semi-supervised approaches have been proposed in video processing to improve the results. Zhu et al. facilitated the classification of a few photo videos by integrating unsupervised information into the memory [**?**]. [**?**] iteratively learned from a few tokens and tagged many instances to localize multiple unknown object instances in long videos. These ideas of unsupervised learning using unlabeled and missing labeled images/videos have inspired subsequent video segmentation methods [**?**], [**?**], [**?**]. [**?**] used a ConvGRU to combine the outputs of pre-trained appearance and motion networks and generate a final segmentation. Similarly, Tan et al. supplement the missing information by adding a supplementary module in semi-supervised real-time video segmentation [**?**]. However, due to the particularity of particle images, these methods are difficult to apply in segmentation tasks. Therefore, motivated by the above studies, this study proposes a Semi-supervised Hybrid-training Particle Segmentation framework (SHPS) for IOOP, which supplements the missing structural information of particle objects by adding a pseudo-label generation module guided by the background skeleton.

III. METHODOLOGY

Aiming at the massive under-segmentation problem caused by complex PIS, this paper proposes a SHPS based on skeleton-guided pseudo-labels, as shown in Alg. 1. In addition, a Background Skeleton-guided Pseudo-label generation algorithm (BSP) is proposed to generate pseudo-labels closer to the ground truth in structural integrity based on coarse segmentation, as shown in Alg. 2. First, the labeled data T is randomly divided into T_1, T_2, and T_3 by a ratio r.

$$T_1, T_2, T_3 = f_{split}(r, T) \tag{1}$$

Algorithm 1 Semi-supervised hybrid-training Algorithm

Input: A network model Net, labeled samples T and unlabeled samples U

Output: The pseudo-labels P corresponding to U and the final model f_{ult}

$T_1, T_2, T_3 = f_{split}(r, T)$;//Divide the labeled dataset
$f_{pre} = f_{sup}(T_1, Net)$;//Get the pre-trained model
for each $i \in [0, len(T_2 + U)]$ **do**
  if the sample $s_i \in U$ **then**
    $C_i = f_{pre}(s_i)$;//Get the coarse segmentation
    $P_i = C_i + f_{pseudo}(C_i)$;//Get the pseudo-labels
    $S_i = f_{MAT}(P_i)$;//Update the skeleton
    Update $Loss = L_{CE}(P_i) + \lambda L_S(S_i)$ and f_{pre};
  else
    $C_i = f_{pre}(s_i)$;//Get the coarse segmentation
    $S_i = f_{MAT}(C_i)$;//Get the background skeleton
    Update $Loss = L_{CE}(C_i) + \lambda L_S(S_i)$ and f_{pre};
  end if
end for
Fine-tuning the f_{pre} with T_3 to get the final model f_{ult}.

where f_{split} refers to a method of randomly partitioning a dataset according to a certain ratio. T_1 is input into the pre-trained model f_{pre}.

$$f_{pre} = f_{sup}(T_1, Net) \tag{2}$$

where f_{sup} refers to the process of supervised learning, and Net refers to a supervised learning-based network model. Then, T_2 and unlabeled data U are randomly mixed to form a mixed dataset for the next training stage. Samples of the mixed dataset are input into f_{pre} to obtain a binary rough segmentation image C.

$$C = f_{pre}(U) \tag{3}$$

Next, C, obtained by unlabeled samples, is input into the pseudo-label generation model f_{pseudo} guided by the background skeleton to obtain pseudo-labels P.

$$P = C + f_{pseudo}(C) \tag{4}$$

where f_{pseudo} refers to the proposed BSP. All pseudo-labels P are used as actual labels in training. After training the mixed dataset, T_3 is used to fine-tune the model to obtain the final model f_{ult}, as shown in Fig. 3.

$$f_{ult} = f_{tune}(T_3, f_{sup}(T_2 + P, f_{pre})) \tag{5}$$

where f_{tune} refers to the fine-tuning.

A. Pre-trained Model

Constructing a pre-trained model is to obtain a rough binary segmentation image using a small amount of labeled data. The segmentation results obtained by this pre-trained model contain massive under-segmentation areas and overlapping unseparated connected domains. Typically, training

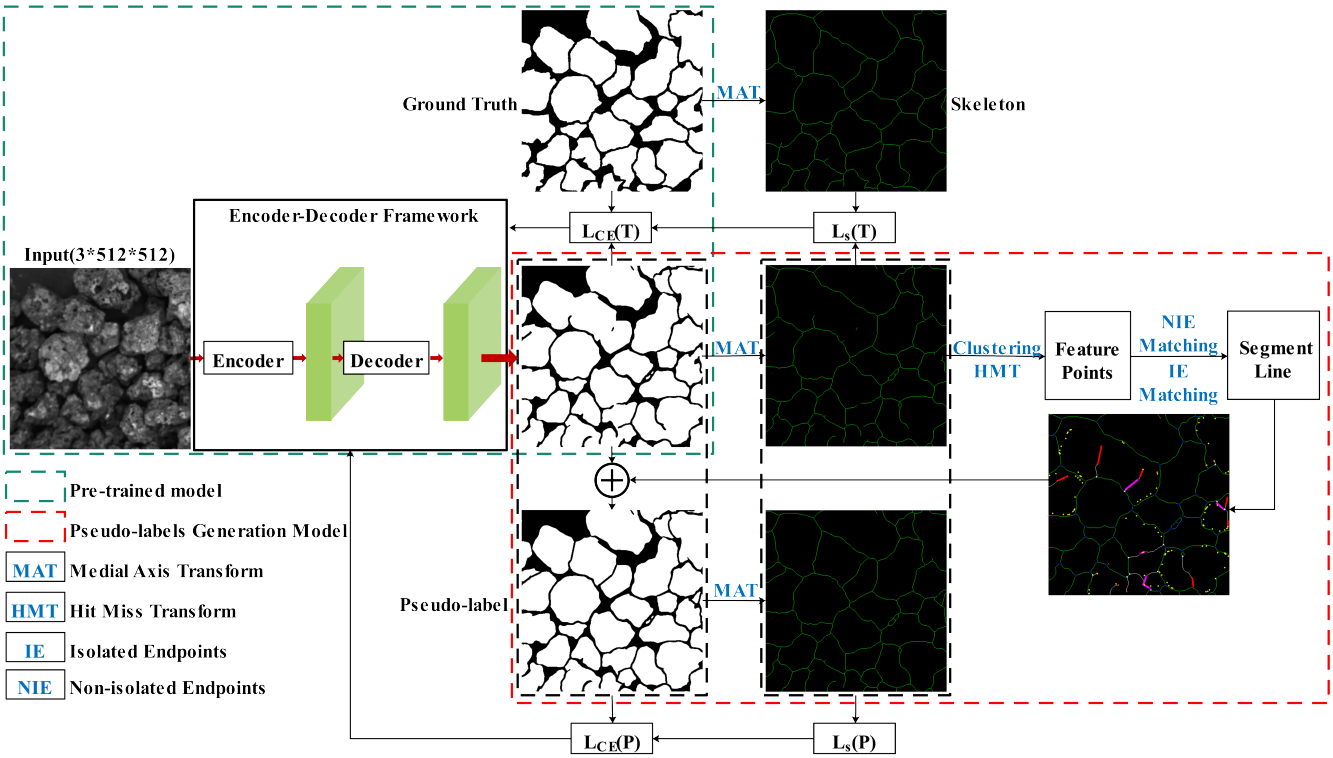


Fig. 3. The structure of network. $L_s(P)$ represents the skeleton difference between pseudo-labels and the coarse segmentation. $L_{CE}(P)$ represents the cross entropy between pseudo-labels and the coarse segmentation.

Algorithm 2 Pseudo Label Generation Algorithm f_{pseudo}

Input: The pre-trained model f_{pre}, labeled samples T_1 and the unlabeled sample U

Output: The pseudo-labels P corresponding to U

1: //Initialize α, β, γ, t in T_1.
2: $C = f_{pre}(U)$;//Get the coarse segmentation.
3: $S = f_{MAT}(C)$;//Get the background skeleton.
4: $O = f_{clu}(C)$;//Get the overlapping domains.
5: $V_l, V_c = f_{HMT}(S)$;//Get leaf and connection vertices.
6: $t_i = t_{i-1} - \eta R$;//Update the parameter t.
7: $p_{nie} = \arg X(V_l) \geq t_i$;
8: $p_{ie} = \arg\min SDD(\arg X(V_l) < t)$;
9: //Get the key points for generating the re-divided line.
10: $P = C + f_{gen}(p_{nie} \cup p_{ie})$;//Generate pseudo-labels

a binary classification model that discriminates between the edge and non-edge pixels of a particle does not discriminate between the internal pixels of the particle and the background pixels among the non-edge pixels. Therefore, we combine edge and background pixels into one category in the annotation file, and particle interior pixels is another category, as shown in Fig. 1.

B. Pseudo-label Generation Guided by Background Skeleton

We propose BSP, which mainly includes connected domain clustering f_{clu}, feature point detection f_{detect}, and feature point matching f_{match}, as shown in Alg. 2.

$$f_{pseudo}(C) = f_{match}(f_{detect}(-C)) \cap f_{clu}(C) \quad (6)$$

*1) **Extraction of Overlapping Connected Domains by Clustering**:* The connected domain clustering algorithm f_{clu} extracts the overlapping connected domains O in coarse segmentation images C. C can be divided into several connected domains and background regions b, denoted as $C = \{c_0, c_1, \cdots, c_k, o_0, o_1, \cdots, o_j, b\}$. To improve computational efficiency, only those connected domains O where the particles overlap needs to be identified. A clustering algorithm based on K-means has been applied to extract overlapping connected domains O_i [**?**].

Each extracted particle connected domain as a sample. The eigenvector of the sample is defined as $F = (\frac{p_h}{p_c}, \frac{a_h}{a_c}, \frac{d_f}{d_s})$, where p_h, p_c and a_h, a_c represent the perimeter and area of the connected hull and connected domains, respectively. d_f indicates the farthest distance from the convex defect to the convex hull. d_s denotes the equivalent diameter of the connected domain area.

Step 1: a sample point z from the dataset Z is randomly selected as the first initial cluster center μ_1.

Step 2: the shortest distance $D(z) = \arg\min \|z - \mu_r\|_2^2$ between each sample and the existing cluster center can be calculated, where $r = 1, 2$. The sample point corresponding to the maximum probability $P(z) = \frac{D(z)^2}{\sum_{z \in Z} D(z)^2}$ is selected as the next cluster center. Moreover, step 2 is repeated until 2 cluster centers are selected.

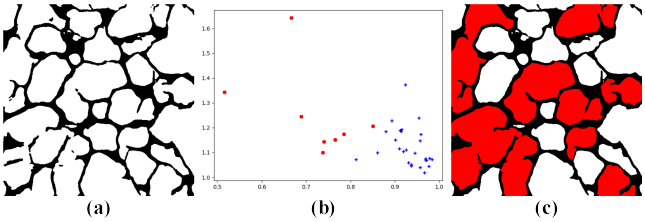


Fig. 4. Connected Domain Clustering. (a)Coarse segmentation image; (b)Results of the clustering algorithm; (c)Extracting overlapping connected domains based on the results of clustering. The number of cluster centers k is 2, where one is the connected domain that needs to be separated, and the other is the connected domain that does not need to be separated. So, the cluster division is defined as $H = (h_1, h_2)$, and the maximum number of iterations is denoted as N.

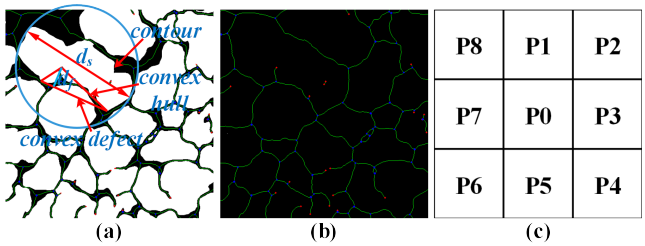


Fig. 5. Extraction of background skeleton in the coarse segmentation image. (a)Geometric features associated with the skeleton; (b)Green pixels represent the skeletons; red pixels represent the leaf vertices $V_l = (v_l^0, v_l^1, \dots)$, and blue pixels represent the connection points $V_c = (v_c^0, v_c^1, \dots)$. And leaf vertices refer to the endpoints of the skeleton branches, and connection points refer to the intersection points between the skeleton branches. (c)8 Neighborhoods of Pixels. The current pixel being processed is noted as p_0.

Step 3: The distance $d_{i,r}$ between the sample z_i and each centroid vector μ_1, μ_2 is calculated. Furthermore, z_i is marked as the category corresponding to $\min d_{i,r}$. Then h_r is updated at this time.

Next, new centroids $\mu_r = \frac{1}{|H_r|}\sum_{z \in H_r} z$ are recomputed for all sample points in H_r. If all centroid vectors have stayed the same or the maximum number of iterations is reached, the cluster division $H = (h_1, h_2)$ is output. Otherwise, $d_{i,r}, H_r, \mu_r$ need to be recomputed.

*2) **Detection of Feature Points on Background Skeleton**:* In overlapping connected domains, some implicit modeling is needed to find feature points to generate segmentation lines to separate overlapping particles. The background has always been regarded as insignificant information in image processing, and the skeleton extraction is also focused on the extraction of the internal part of interest objects. However, the background skeleton provides some valuable information for segmentation in the task of PIS [**?**]. With this motivation, we found that the leaf vertices of the skeleton branches become possible segmentation points after extracting the background skeleton branches that neighbor these connected domains, as shown in Fig. 5.

a) Extraction of Background Skeleton: A skeleton extraction algorithm f_{MAT} is introduced to extract the background skeleton of segmentation images, which can guide our optimization decision for pseudo-labels [**?**]. This algorithm can refine a connected region to one-pixel width

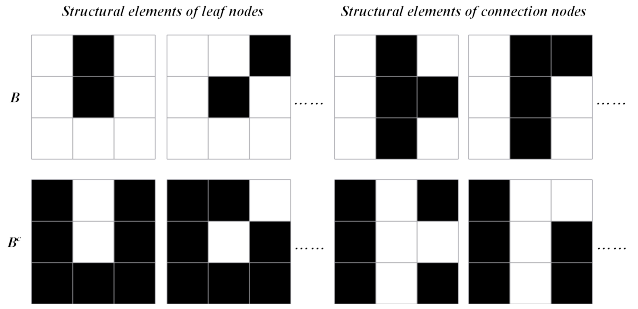


Fig. 6. Some structural elements and their complements are used in HMT [**?**]. The template in the first row is the structural element, while the template in the second row is its complement. In the four-pixel structure element, only three positions in the 8-neighborhood represent 0, for a total of C_8^3 kinds of templates. In the two-pixel structure element, only one position in the 8-neighborhood represents 0, for a total of C_8^1 templates.

for feature extraction and object topology representation.

Step 1: all foreground pixels are traversed, and the pixels p_0 need to be deleted if they meet these conditions: (1) $2 \leq N(p_0) \leq 6$; (2) $Z_0(p_0) = 1$; (3) $p_1 * p_3 * p_5 = 0$; (4) $p_3 * p_5 * p_7 = 0$, i.e., let $p_0 = 0$. Among them, $N(p_0)$ is the number of foreground pixels in the eight neighborhoods of p_0. $Z_0(p_0)$ represents the number of occurrences of $0- > 1$ in two adjacent pixels in the clockwise direction in the 8-neighborhood pixels, as shown in Fig. 5. Points judged to be deleted are not deleted immediately but wait until all points have been traversed, and then all points marked as deleted are deleted together.

Step 2: Similarly, another deletion is performed, but the judgment condition (3) and (4) becomes: (3) $p_1 * p_3 * p_7 = 0$; (4) $p_1 * p_5 * p_7 = 0$. Steps 1 and 2 are repeated until there are no more points to delete.

b) Detection of leaf nodes and connection points: HMT is the hit-miss transform algorithm, which is a morphological operator [**?**]. HMT is mainly used to find some structures existing in binary images.

$$V_l, V_c = f_{HMT}(S) = (S \ominus E) \cap (S^c \ominus E^c) \tag{7}$$

where $\ominus$ denotes the erosion process, E represents the structural element, and S^c represents the complementary set of the background skeleton S. V_l refers to all leaf vertices, and V_c denotes all connection points, as shown in Fig. 5(b). The structural element E is designed as two types, one for detecting leaf nodes and the other for detecting connection points, as shown in Fig. 6.

*3) **Matching Feature Points based on Geometric Features**:* In overlapping connected domains, there is at least one endpoint. If there is only a single endpoint or no other matching endpoint, this endpoint is called an isolated endpoint (IE). Otherwise, if an endpoint has matched at least one other endpoint, it is called a non-isolated endpoint (NIE).

a) NIE matching: NIE matching is proposed to match the candidate feature points, and its matching effect significantly impacts on the experimental results. Therefore, a series of ablation experiments are needed to analyze and verify

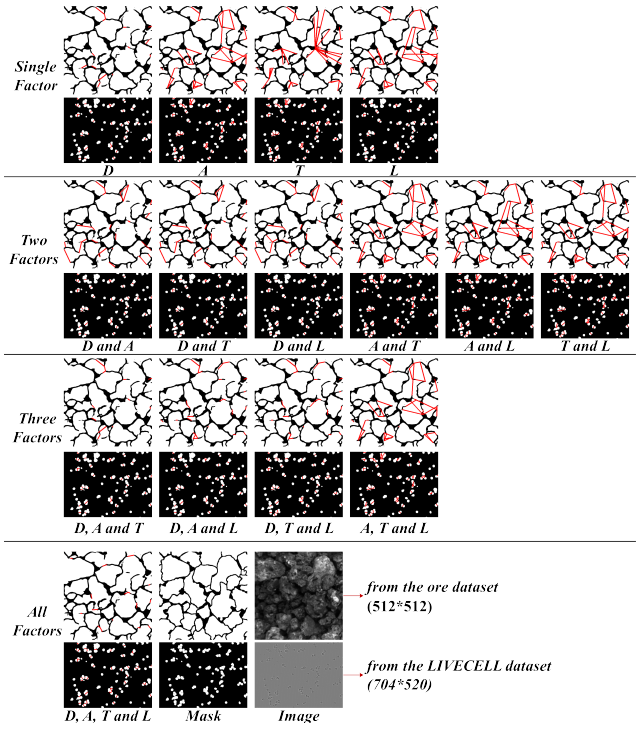


Fig. 7. Ablation experiments exploring the factors in NIE matching. The best results are obtained when the four influencing factors are considered simultaneously.

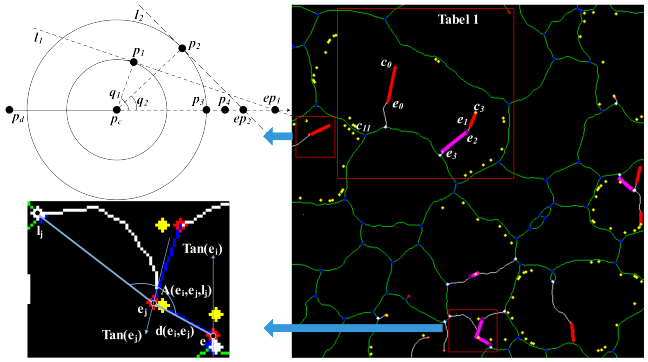


Fig. 8. The matching process of leaf vertices and the generation of dividing lines. The purple line represents the dividing line generated after the two leaf nodes are successfully matched. The red line represents the dividing line formed by connecting a leaf vertex to the nearest FPCD when it cannot match other leaf vertices. p_1, p_2, p_3, and p_4 represent the sample points, and l_1 and l_2 represent the tangent of p_1 and p_2 on the circle, respectively. The line between p_d and p_c determines the direction of p_c. θ_1, θ_2 represent the angles $\pi - \angle p_1 p_c p_d$ and $\pi - \angle p_2 p_c p_d$, respectively. ep_1 and ep_2 represent the equivalent distance between p_1, p_2 and p_c on the line. Two points initially equal in the measurement of Euclidean distance have changed under this equivalent rule. The distance sort $p_3 < p_4 < p_2 < p_1$ is obtained from the above formula.

the rationality of our proposed matching index. Specifically, we assumed some descriptive variables based on geometric features: pinch angle (A), tangent line pinch angle (T), branch length (L), and distance (D) by comparing observations in the background skeleton of labeled and segmented images. The closer A and T are to $180°$ for two candidate feature points, the higher the plausibility of matching the two points. Moreover, the longer L and the smaller D is, the more they may also form a match. Therefore, to find patterns in the effects of these descriptive variables on matching, we conducted quantitative comparison experiments on the ore dataset and the LIVECELL dataset [**?**] with different combinations to design reasonable matching metrics, mainly consisting of considering only a single factor, two factors, three factors, and all factors, as shown in Fig. 7.

A metric X describes the rationality of a segmentation line formed between different endpoints. This metric determines whether the endpoints have enough possibility to match others. The larger the value is, the more reasonable the segmentation line formed by two endpoints will be.

$$X(v_l^i, v_l^j) = \frac{\alpha A(v_l^i, v_l^j) + \beta T(v_l^i, v_l^j) + \gamma L(v_l^i)}{D(v_l^i, v_l^j)} \tag{8}$$

where $A(v_l^i, v_l^j)$ represents the angle formed by the endpoints v_l^i, v_l^j and connection point v_c^i (v_c^i is the intersection point). $T(v_l^i, v_l^j)$ represents the angle formed by the tangents $tan(v_l^i)$ and $tan(v_l^j)$. Tangents are obtained by fitting the pixels on the skeleton branch within their 7×7 neighborhood. $L(v_l^i) = \ln len(v_l^i)$ represents the length of the skeleton branch $B(v_l^i, v_c^i)$. $D(v_l^i, v_l^j)$ represents the Euclidean distance between e_i and l_i. Moreover, α, β, γ are coefficients determined automatically during initialization. Finally, if $X(v_l^i, v_l^j)$ is greater than the threshold t for automatic updates, then v_l^i successfully matches the endpoint v_l^j, which are connected to form a line, as shown in Tab. I.

$$p_{nie} = \arg X(V_l) \geq t \tag{9}$$

where p_{nie} denotes the set of NIE point pairs used to generate the line segments.

b) IE matching: IEs need to find other feature points because no other leaf nodes can match. The feature point to select is the farthest point of the convex defect (FPCD) in a connected domain. A novel distance metric, the Same Direction Distance (SDD), is proposed to match the best feature point for each IE. The distance between the two points and the continuity and smoothness between the two points and the skeleton branches need to be considered. Euclidean distance cannot reflect the direction deviation of two points on the curve, impacting the matching. Therefore, the two-point distance in different directions is converted to a two-point distance on a straight line by a circle, reflecting the distance deviation caused by angular deviation. SDD is defined by considering the non-negativity and smoothness.

$$SDD(p_s, p_c, p_d) = \frac{d(p_s, p_c)}{e^{\cos(\pi - \angle p_s p_c p_d)}} \tag{10}$$

where $d(p_s, p_c)$ represents the Euclidean distance between p_s and p_c. An IE is connected with the nearest FPCD to form a segment line via the SDD, as shown in Fig. 8.

$$p_{ie} = \arg\min SDD(\arg X(V_l) < t) \tag{11}$$

where p_{ie} denotes the set of IE point pairs. Finally, the segmentation line of each connected domain O is obtained by cyclic calculation, which is superimposed on C to form the pseudo label P.

$$P = C + f_{gen}(p_{nie} \cup p_{ie}) \tag{12}$$

where f_{gen} denotes the function that generates line segments based on point pairs.

C. Initialization and Update of Key Parameters

a) Parameter initialization: Parameter initialization is possible as a preprocessing and does not directly affect the model's training. First, the coarse segmentation results are obtained with the labeled data T_1 input to f_{pre}. Then we use MAT to extract its background skeleton with the corresponding label and calculate the branch difference B_{dif} between these two.

$$B_{dif} = f_{MAT}(f_{pre}(s)) - f_{MAT}(Label(s)) \tag{13}$$

where $Label(s)$ denotes the label of s. Next, all candidate feature points are extracted according to the similar steps. Moreover, the feature points are selected as the initial points according to the position of the branch difference nearby.

$$V_b = \underset{b_i, b_j \in B_{dif}}{\arg\min}\, d(f_{gen}(V_l), f_{gen}(b_i, b_j)) \tag{14}$$

where b_i, b_j denote the endpoints of the branch difference B_{dif}. $V_b = (v_b^0, v_b^1, \dots)$ denotes the endpoint pairs closest to the split line in the pseudo-label in the branch difference B_{dif}. d denotes the Euclidean distance. A system of equations is then constructed based on all $X(V_b)$:

$$X(V_b) = \begin{bmatrix} \frac{A_0}{D_0} & \frac{T_0}{D_0} & \frac{L_0}{D_0} \\ \cdots & \cdots & \cdots \\ \frac{A_m}{D_m} & \frac{T_m}{D_m} & \frac{L_m}{D_m} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x_0 \\ \cdots \\ x_m \end{bmatrix} \tag{15}$$

Those other feature points that are not selected to generate line segments also construct a system of equations $X'(\tilde{V}_b)$, where $\tilde{V}_b$ denotes the point pairs in V_l that does not correspond to V_b.

TABLE I
VALUE CHANGES OF EACH DESCRIPTIVE VARIABLE DURING THE MATCHING PROCESS OF CANDIDATE FEATURE POINTS

Endpoints	X or SDD	A	T	L	D	label	prediction
e_0	0.61	125.69	128.38	3.56	75.99	$e_0 \to c_0$	$e_0 \to c_0$
e_1	0.63	125.69	128.38	1.95	79.13	$e_1 \to c_3$	$e_1 \to c_3$
e_2	3.52	162.54	172.27	1.95	46.23	$e_2 \to e_3$	$e_2 \to e_3$
e_3	3.37	155.42	130.44	1.94	46.23	$e_3 \to e_2$	$e_3 \to e_2$
c_0	18.61	——	——	——	——	——	——
c_3	15.14	——	——	——	——	——	——
c_{11}	20.17	——	——	——	——	——	——
...	...	...	...	...	...	...	...
t	X_3-0.001						

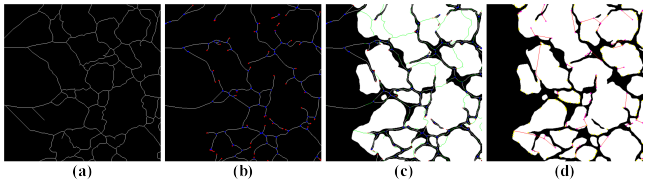


Fig. 9. Initialize parameters by learning how labeled data behaves in a pre-trained model. (a)Background skeleton of the label; (b)Background skeleton of coarse segmentation results. Among them, red points refer to leaf vertices V_l, and blue points denote the connection point V_c. (c)Green line refers to the branch difference B_{dif} between the background skeleton of the coarse segmentation result and the background skeleton of the label. (d)The red line segments are obtained by fitting the branch differences B_{dif} to the label skeleton, and these will determine the initialization of α, β, γ, t.

$$\begin{cases} \max X'(\alpha, \beta, \gamma) < t_0 \le \min X(\alpha, \beta, \gamma) \\ \alpha + \beta + \gamma = 1 \\ \alpha, \beta, \gamma > 0 \end{cases} \tag{16}$$

where X' denotes X for unmatched point pairs. Finally, the above system of inequality equations is solved to obtain the range of values of $\alpha, \beta, \gamma, t_0$, which is used as the initial parameter of this data set for the subsequent calculation.

b) Parameter Update: Threshold t is one of the main factors affecting the matching results of feature points, which mainly determines the quality of pseudo-labels. First, we obtain P by inputting the output C of the T_1 dataset in f_{pre} into the BSP and making the difference between P and the label minimal. t in these pseudo-labels is the initial threshold for subsequent hybrid-training.

$$t_i = t_{i-1} - \eta \left[ma(s_{i-1}) - \frac{1}{N(i-1)} \sum_{j=0}^{N(i-1)-1} ma(s_j) \right] \tag{17}$$

where $ma(s) = \frac{1}{n} \sum_{i=0}^{n-1} a_i$ represents the average area of all connected domains a_i in the foreground of a sample s. $N(i-1)$ represents the number of all labeled samples before the $i-1$ sample. s_{i-1} represents the $i-1$ sample. η represents the step size. When $ma(s_{i-1})$ is greater than the average area mA of all previously labeled samples, the generated pseudo-label has under-segmentation, so t needs to be decreased to add more segmentation lines. Conversely, t needs to be increased to reduce the degree of over-segmentation. This mechanism is similar to training network parameters by a loss function, which is not the same as the existing learning mechanism. Still, the learning idea is essentially the same, i.e., adjusting the propensity of the generated pseudo-labeled data by learning the features of the labeled data.

D. Loss Function

Focal Loss has been applied in the pre-training [**?**]. In the hybrid-training, even when mIoU performs well, there may still be many overlapping particles that are not separated because the difference between the overlapping particles

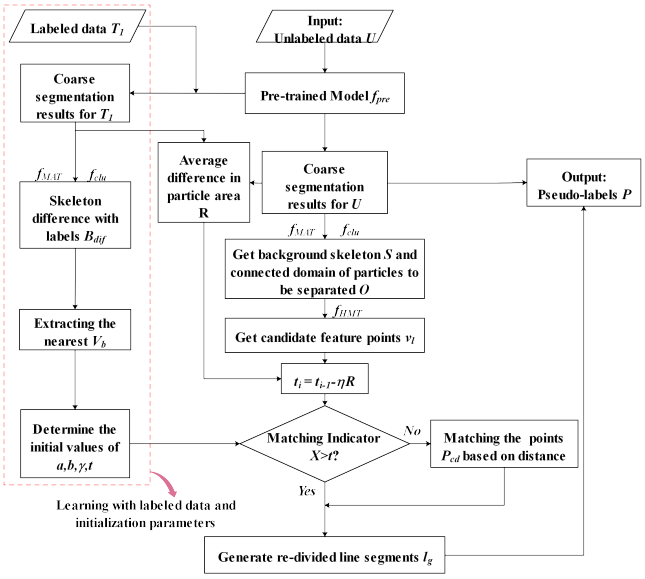


Fig. 10. Flow chart of parameter learning and updating in the process of pseudo label generation. R indicates the content of square brackets in Formula. 17.

and the ground truth is just a part of edge pixels, which have little impact on mIoU. However, these pixels has destroyed overlapping particles' basic structure and contour. The network may mistakenly believe that the training needs to be improved. Most of the background skeleton has almost no change during the training process compared to the label. Suppose a segmentation result has massive under-segmentation or over-segmentation. In that case, the skeleton change is mainly reflected in the presence of more leaf vertices in the graph structure. Therefore, the loss function is adjusted by using the skeleton difference to improve the training.

$$Loss(Y) = L_{CE}(T) + L_{CE}(P) + \lambda \cdot (L_S(T) + L_S(P)) \quad (18)$$

where $L_S = \sum_i \sum_j d(v_{i,j}, \min v_i^t)$ denotes the skeleton loss. $v_{i,j}$ represents the j-th vertex position in the background skeleton of sample i. v_i^t denotes all the vertex positions in the background skeleton of label or pseudo-label i. $d(v_{i,j}, v_i^t)$ indicates the Euclidean distance between $v_{i,j}$ and v_i^t on the image. $\lambda = \frac{n}{k}$ implies the ratio of the overlapping connected domain number n to the total connected domain number k in the coarse segmentation result.

IV. Experiment

A. Dataset and Experimental Platform

A blast furnace sinter is a kind of IOOP with a complicated label. To obtain the sinter images, we installed industrial cameras and other equipment on the blast furnace feeding belt in a steel plant. 56 non-duplicate pictures are selected as labeled datasets from the collected images, while 84 other pictures are chosen as the unlabeled dataset from the collected images. The label drawing tool is employed to label the 56 pictures manually, which labels the pixels as background and sinter internal area. The 56 images were divided into 42 training sets, 7 validation sets, and 7 test sets. 2100 512×512 labeled images and 4200 512×512 unlabeled images were obtained by data enhancement. 2100 labeled data were divided into three parts during the experiment according to a preset ratio.

In addition, cells are also representative particle objects in the biomedical field. Therefore, we have tested on LIVECELL, a currently open-source biomedical cell segmentation dataset [**?**]. This paper configures the experimental hardware environment as Intel core i9-9900K 3.6 GHz 32 GB and NVIDIA GeForce RTX2080TI 11 GB. The software environment is configured as Python 3.7.6, PyTorch 1.4.0, CUDA 10.1, and CUDNN 7.6.5. Adam is employed for optimization with the learning rate set to 1×10^{-4}, where a cosine annealing learning rate scheduling policy is adopted to adjust the learning rate with a minimum value of 1×10^{-6} in the last epoch. The weight decay is set to 1×10^{-8}.

B. Evaluation Metrics

$OSR = \frac{FP}{TP+FP+FN}$ (over-segmentation) and $USR = \frac{FN}{TP+FP+FN}$ (under-segmentation) are applied to evaluate the segmentation results for labeled data. However, mIoU only pays attention to the accuracy of pixels and edges but not the semantic information of objects, i.e., the structural integrity of connected domains. The key to quantitatively evaluating the quality of PIS is to evaluate whether the edge is correct and to ensure the particle numbers' accuracy and structural integrity. Therefore, a mean particle quality difference ($mPQD$) describes the connected domain difference between the sample and label. The smaller the $mPQD$ is, the better the particle structural integrity.

$$mPQD = 1 - \varepsilon \cdot \frac{mA - mA_T + vA - vA_T}{mA_T \cdot vA_T} \quad (19)$$

where ε represents a parameter to balance the size. mA, vA, mA_T and vA_T represent the mean and standard deviation of the connected domain area in samples and labels, respectively.

C. Comparative analysis of experimental results

1) Overall Accuracy and Efficiency Comparison: Because UNet, Segmenter [**?**], DANet [**?**], Mask R-CNN [**?**] and ALSS [**?**] have better effects and wider applications in supervised semantic segmentation, instance segmentation and semi-supervised semantic segmentation, they are used for comparative analysis. Some test results are shown in Fig. 11 and Tab. II selected from the ore and LIVECELL datasets. First, our model has been trained on half of the unlabeled data, and it achieves very close results on mIoU compared to fully supervised algorithms such as UNet, which is only 2.1% less than UNet. However, our model even outperformed UNet by 3.4% on mPQD, which indicates that our model performs better on the particle structure integrity.

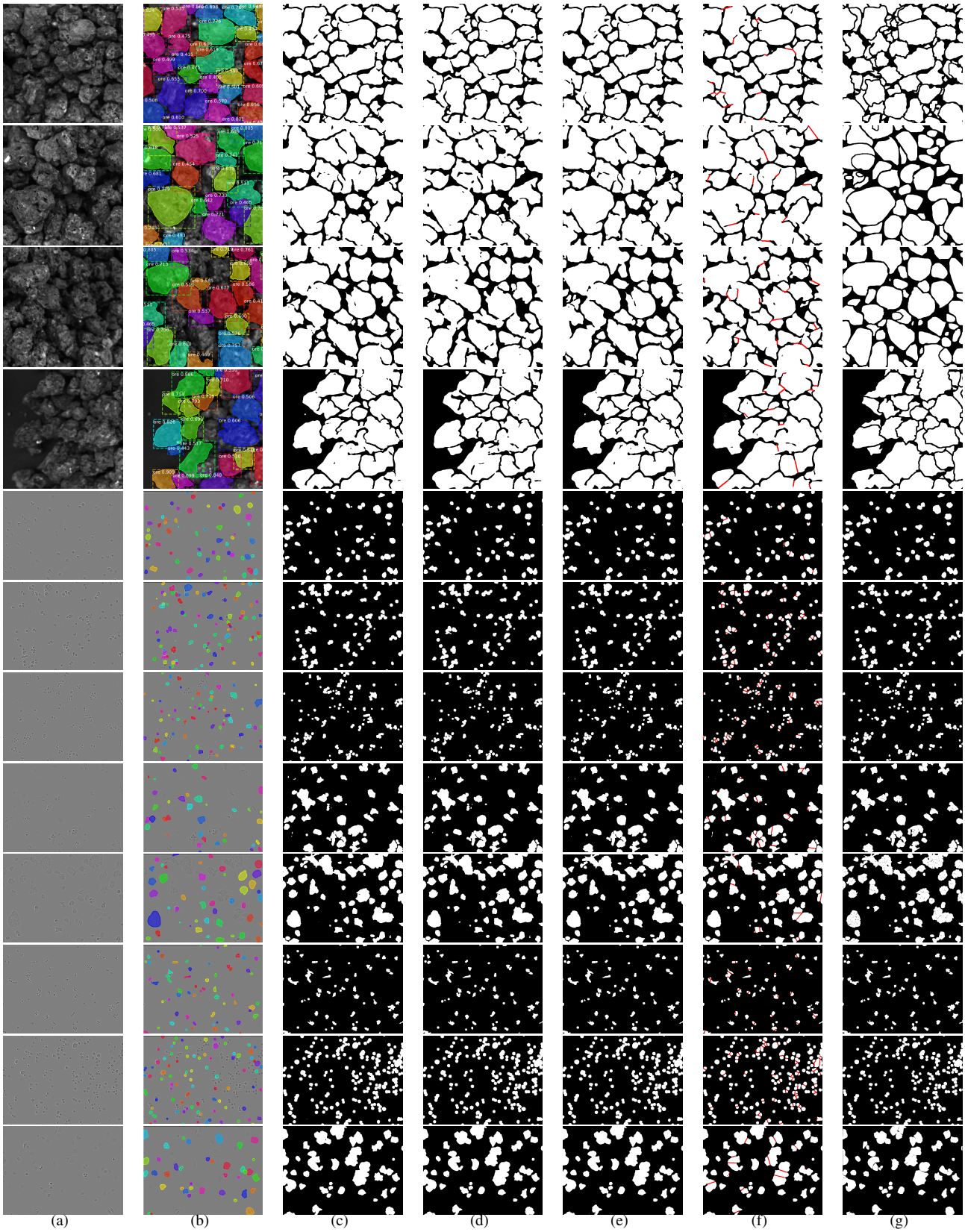


Fig. 11. The performance of different methods in some test sets. (a)RGB image; (b)Mask R-CNN; (c)UNet; (d)Segmenter; (e)DANet; (f)SHPS+BSP(Ours); (g)Ground truth. The first 4 samples are from the ore dataset, and the latter 8 samples were from the cell segmentation dataset LIVECELL with A172, BT474, BV2, Huh7, MCF7, SHSY5Y, SkBr3 and SKOV3 class cells [**?**].

TABLE II
THE COMPARISON OF DEPTH MODEL SEGMENTATION QUALITY WITH DIFFERENT TRAINING DATASET

Method	pre-trained	Ore Dataset			Background IoU	Foreground IoU	mIoU	mPQD	OSR	USR	Testtime
		Labeled T_1	T_2+Unlabeled	T_3							
SegNet [**?**]	——	2100	0 + 0	0	0.745	0.875	0.810	0.842	0.102	0.088	0.403
UNet [**?**]	——	2100	0 + 0	0	0.756	0.890	0.823	0.843	0.103	0.074	1.087
Segmenter [**?**]	——	2100	0 + 0	0	0.751	0.871	0.811	0.841	0.109	0.080	1.212
DANet [**?**]	——	2100	0 + 0	0	0.778	0.872	0.825	0.856	0.104	0.071	3.814
MaskFormer [**?**]	——	2100	0 + 0	0	0.755	0.873	0.814	0.842	0.108	0.078	0.878
SegFormer [**?**]	——	2100	0 + 0	0	0.746	0.868	0.807	0.834	0.110	0.083	1.180
SETR [**?**]	——	2100	0 + 0	0	0.774	0.884	0.829	0.858	0.101	0.070	4.436
Mask R-CNN [**?**]	——	2100	0 + 0	0	0.652	0.856	0.754	0.881	0.104	0.142	3.392
ALSS [**?**]	——	1050	0 + 1050	0	0.651	0.833	0.742	0.818	0.142	0.116	1.301
SHPS+BSP(Ours)	SegNet	1050	0 + 1050	0	0.705	0.839	0.772	0.849	0.132	0.096	2.055
SHPS+BSP(Ours)	UNet	1050	0 + 1050	0	0.719	0.845	0.782	0.857	0.128	0.090	2.645
SHPS+BSP(Ours)	UNet	2100	0 + 4200	0	0.765	0.887	0.826	0.898	0.103	0.071	2.760
SHPS+BSP(Ours)	UNet	1575	0 + 4200	525	0.766	0.892	0.829	0.903	0.102	0.069	2.823
SHPS+BSP(Ours)	UNet	1575	525 + 4200	0	0.768	0.896	0.832	0.905	0.101	0.067	2.798
SHPS+BSP(Ours)	UNet	1575	263 + 4200	262	0.778	0.904	0.841	0.918	0.098	0.061	2.701
SHPS+BSP(Ours)	UNet	1050	0 + 4200	1050	0.771	0.903	0.837	0.913	0.100	0.063	2.873
SHPS+BSP(Ours)	UNet	1050	1050 + 4200	0	0.769	0.901	0.835	0.909	0.099	0.066	2.629
SHPS+BSP(Ours)	UNet	1050	525 + 4200	525	0.781	0.908	**0.844**	**0.926**	0.097	**0.059**	**2.739**
Ground truth	——	——	——	——	1	1	1	1	0	0	0

Compared to ALSS, our method significantly improves the mPQD and USR by 5.9% and 4.6%, respectively. In addition, the results of the comparison experiments also show that the better performing pre-trained models also lead to better overall performance of our models.

We elaborate more on the differences between our and existing algorithms in accuracy. Better pseudo-labels in the particle segmentation task refers to pseudo-labels with higher accuracy (with label) and complete particle structure (without label). The existing pseudo-labels are mainly obtained by predicting unlabeled data from models trained with labeled data and then selecting samples with higher confidence as pseudo-labels. However, these pseudo-labels are greatly limited by the accuracy of pre-trained models and cannot guarantee the structural integrity. To address these two most essential requirements, our proposed BSP differs from existing approaches in two ways: (a) Instead of selecting samples with higher confidence, we optimize each sample predicted by the pre-training model. We learn geometric features such as background skeleton from the pre-training model prediction with labeled data to pseudo-labels in the form of physical labels to obtain pseudo-labels with better performance in particle structural integrity. (b) Instead of training a new model, we continue to train and fine-tune the pre-trained model by mixing the optimized pseudo-label with another portion of the labeled data.

We use the information learned from the pre-trained process with labels to optimize the pseudo-labels by physical labels. This approach is equivalent to implicitly modeling some prior information, which can reduce the dependence of pseudo-labels on the pre-trained model and improve the robustness of pseudo-labels. Our pseudo-labels are closer to the actual labels in the particle structural integrity, with less under-segmentation and higher accuracy.

TABLE III
COMPARISON OF SEGMENTATION ACCURACY IN LIVECELL

A172	BT474	BV2	Huh7	MCF7	SHSY5Y	SkBr3	SKOV3	mIoU
0.865	0.866	0.859	0.851	0.853	0.867	0.862	0.850	0.859

TABLE IV
TIME SPENT IN DIFFERENT DATASETS

	pre-trained	MAT	Clustering	HMT	Matching	Total
Ore (ms)	1086.71	146.71	7.915	1429	69.03	2739
LIVECELL	1579.36	179.36	8.128	1523	73.04	3362
TC	Flops=15.5M	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	-
SC	Para=7.77M	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	-

In the efficiency, we counted the time consumed by each step in testing the ore and LIVECELL datasets, as shown in Tab. III. In addition, we also did a complexity analysis. The pre-trained model in this paper is a convolutional neural network, which its time complexity (TC) is $O(\sum_{d=1}^{D} M_d^2 \cdot K_d^2 \cdot C_{d-1} \cdot C_d)$ and space complexity (SC) is $O(\sum_{d=1}^{D} K_d^2 \cdot C_{d-1} \cdot C_d + \sum_{d=1}^{D} M^2 \cdot C_d)$. D denotes the number of convolution layers, i.e., the depth of the network. d denotes the d-th convolution layer. C_d denotes the output channel of the d-th convolution layer. M denotes the edge length of the output feature map. K denotes the convolutional kernel size. After calculation, we use FLOPs and Para to denote the number of floating point operations and the number of parameters of the model. According to the execution cycle and nesting of the algorithm, the complexity of our other algorithms is shown in Table. IV.

2) Effectiveness and Generalization Ability of HMT and NIE matching: HMT has been applied to image matching in various fields, and a large amount of literature demonstrates the effectiveness of HMT as a template matching algorithm [**?**], [**?**], [**?**]. In the present, because the existence of leaf nodes and connection points is considered in enumerable

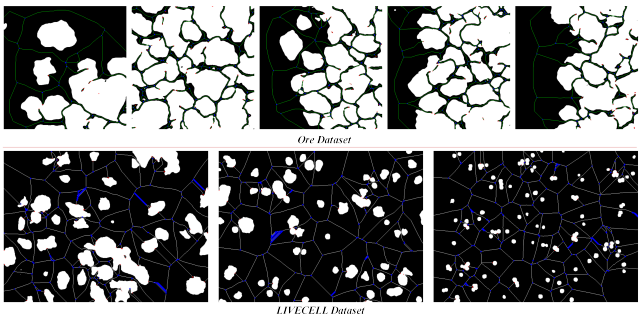


Fig. 12. Test results of HMT on different samples from different datasets. Red pixels represent the leaf vertices, and blue pixels represent the connection points.

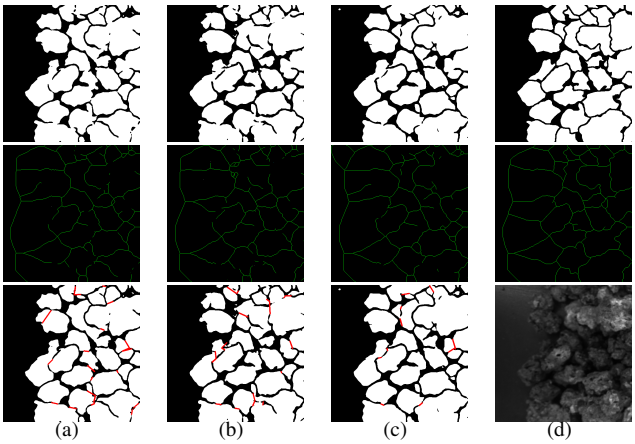


Fig. 13. The performance of our model with different label distributions. (a)1575, (0+4200), 525; (b)1575, (263+4200), 262; (c)1050, (525+4200), 525; (d)Ground Truth

forms, we wish to detect leaf nodes and connection points by template matching. Compared with the template matching problem in astronomical images, our difficulty in detecting leaf nodes and connection points in particle images is lower, so HMT possesses sufficient validity in the study of this paper. We conducted comparison experiments on the ore and LIVECELL datasets to demonstrate the generalization ability of HMT, as shown in Fig. 12.

NIE matching is essentially a feature point matching method based on the proposed matching metric, and its key point is the design of the matching metric X. We give the plausibility analysis experiments of its design to demonstrate the effectiveness and generalization ability of NIE matching on page 6, as shown in Fig. 7.

3) Robustness Analysis under Different Label Divisions: The previous section mentioned that dividing the labeled data into three parts is required. There are two issues to

TABLE V
ACCURACY VARIATION UNDER THREE RANDOM DIVISIONS

Ore Dataset	First	Second	Third	mean
mIoU	0.844	0.838	0.841	0.841

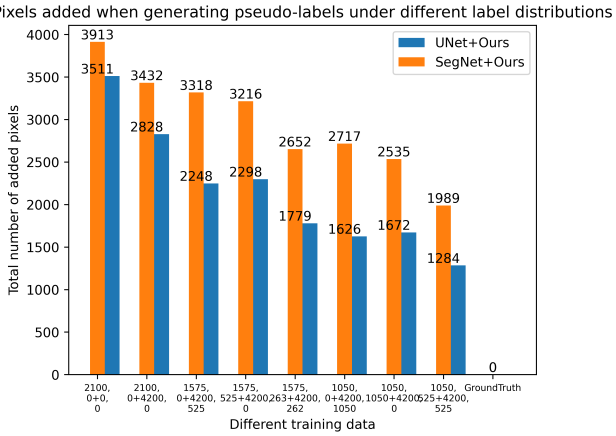


Fig. 14. Distribution of the number of background pixels added by our method during generating pseudolabels for different pre-training models. The training results generate pseudo-labels that are closer to the real labels when the label data are more evenly distributed, and fewer pixels need to be added to optimize the pseudo-labels. In addition, the effect of pseudolabel generated by different baselines is different, and the pseudolabel generated by a good baseline is of better quality.

be discussed, one is how to determine the proportion of the division, and the other is whether it will significantly impact on the results since each sample is divided randomly. To address the first question, we conduct separate ablation experiments in label distributions with different division ratios to find patterns in a limited time cost, as shown in Tab. II. To evaluate the good and bad effects under different divisions, in addition to mIoU, we count the total number of added pixels (corresponding to the red pixels in Fig. 13) when pseudo-labels are generated by the proposed method, as shown in Fig. 14. These increased pixels are generated by our algorithm when the under-segmented particles are re-segmented. Under the same conditions, adding more pixels indicates that the results need more re-segmentation and the model performs terribly. On the other hand, fewer added pixels mean that the results are already very close to the labels, the particles have strong structural integrity, and the model performs better. As the training progresses, the segmentation results obtained by the model gradually do not need to generate a large number of segmentation lines to make up for the under-segmentation, and only a small number of pixels are needed to make the pseudo-labels closer to the actual labels. Experimental results show that the group with more evenly divided labels obtained the best results, reaching 84.4% on mIoU, demonstrating the necessity and effectiveness of our proposed hybrid-training.

For the second question, random splitting refers to the operation of randomly dividing the labeled data into three parts according to a certain ratio. The random splitting of labeled data does introduce some fluctuations. However, these fluctuations are tolerable within a single dataset because our parameter initialization and each update and decision step are based on the information learned from the pre-training process. When the pre-trained model changes, our parameter

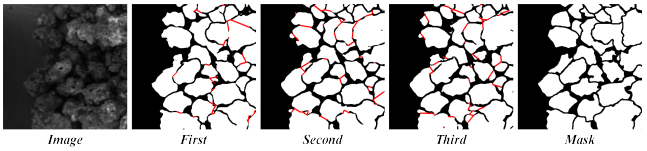


Fig. 15. Test results under three random divisions of the dataset.

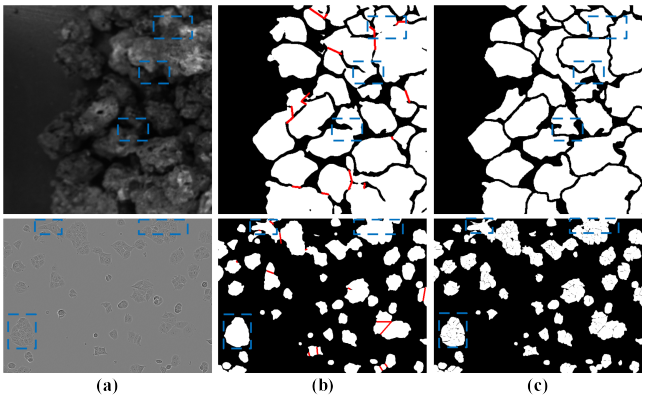


Fig. 16. Some test samples with less-than-satisfactory segmentation results. The blue boxes mark the locations that clearly differ from the labels. (a)Images; (b)Test results; (c)Ground truth.

initialization and each step of update and decision will change accordingly. However, the time and computational cost of finding the best division is too high, so we choose to use random division as a more compromised method, which reduces the experimental cost and avoids the chance situation. Moreover, finding the optimal partitioning and training pattern is still crucial to our algorithm and will be one of the main directions of our subsequent research. Finally, the comparative experimental results show that random splitting does not significantly impact on the accuracy, as shown in Fig. 15 and Tab. V.

4) Analysis of Advantages and Limitations: Our method focuses on solving a class of image segmentation problems with complex overlapping particles. However it still has some limitations, as shown in Fig. 16. Our algorithm has specific requirements for particle shape, and although it can tolerate minor random variations in edges, its robustness is low when facing extremely irregular objects. BSP cannot guarantee the complete elimination of under-segmentation in samples. It may also add a small amount of over-segmentation, although this does not significantly affect the particle structural integrity.

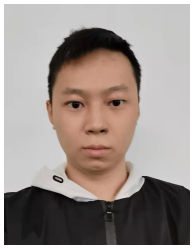
For the first limitation, since the geometric features of particles have high similarity, then if the shape of particles and overlapping shapes can be learned and modeled by some learning mechanism, it can improve the generalization performance of complex particles identified and segmented to some extent. Under-segmentation is still the main challenge problem in PIS for the second limitation. However, the small amount of added over-segmentation still affects the overall accuracy, so the segmentation model with higher accuracy can be pre-trained to improve the overall performance. These limitations are also problems we are prepared to tackle in the future. Their solutions will also bring more valuable ideas to general-purpose image/video segmentation field.

V. CONCLUSION

This paper proposes a pseudo-label-based semi-supervised hybrid-training particle segmentation framework to overcome the under-segmentation problem caused by complex particle image segmentation. Our hybrid-training framework uses the valuable information contained in the labeled data. It achieves results that are close to those of fully supervised training when the labels are evenly distributed, overcoming, to some extent, the problem of too few labels. At the same time, we also propose a pseudo-label generation algorithm guided by the background skeleton that uses the connection between the information of the background skeleton and segmentation to find feature points. The pseudo-labels generated by our method are closer to labels in terms of their structural integrity, and they reduce the degree of under-segmentation to a certain extent. Furthermore, we construct an industrial particle image dataset and conduct ablation experiments in the LIVECELL dataset. Our method achieves 84.4% accuracy on mIoU with uniform label data distribution, which is 2.1% higher than the accuracy of UNet. These results successfully demonstrate that our method performs well on particle image segmentation.

APPENDIX

Notations	Explanation
f_{split}	Randomly dividing the dataset
f_{sup}	Supervised training
f_{pre}	Pre-trained model
f_{pseudo}	Pseudo-labels generation model
f_{tune}	Fine-tuning
f_{ult}	The final model
f_{detect}	Feature point detection process
f_{match}	Feature point matching process
f_{clu}	Clustering process
f_{MAT}	Skeleton extraction algorithm
f_{HMT}	Hit-miss transform algorithm
f_{gen}	Generate line segments from point pairs
T, T_i	Labeled data
r	A ratio for dividing the dataset
Net	A segmentation network model
U	Unlabeled data
C	Rough segmentation results
P	Pseudo-labels
S	Skeltons
s	Samples
O	Overlapping connected domains
X	Matching metric
A, T, D, L	Pinch angle;tangent pinch angle;distance;branch length
a, b, γ	Coefficients in X
SDD	Same Direction Distance
d	Euclidean Distance
p_{nie}	NIE point pairs
p_{ie}	IE point pairs
t	A threshold for determining the matching
B_{dif}	Skelton difference
V_b	The endpoint pairs closest to B_{dif}
$\tilde{V}_b$	The point pairs in V_l that does not correspond to V_b.
V_l	Leaf vertices
V_c	Connection vertices



Jinshi Liu received a bachelor's degree and a master's degree in automation from Central South University, China in 2017 and 2020, respectively. He is currently pursuing a Ph.D. in Control Engineering at Central South University. His research interests include deep learning, detection technology, image segmentation, industrial VR and digital twins.

Zhaohui Jiang received a M. Eng. degree in Automatic Control Engineering and a Ph.D. degree in Control Science and Engineering from Central South University, China in 2006 and 2011, respectively. He is currently a professor at Central South University. His research interests include Detection Technology and Automatic Equipment, image processing, industrial VR, modeling and optimal control of complex industrial processes.

Ting Cao received her B.Eng. degree in Geographic information system and and her Ph.D. degree in Structural Geology from Zhejiang University, Hangzhou, China, in 2013 and 2018, respectively. She is currently a full-time Post-doc researcher with Pengcheng Laboratory, Shenzhen, China. Her research interests are image processing, modeling and optimal control of complex industrial processes.

Zhiwen Chen (M'17) received his B.S. degree in electronic information science and technology and M.S. degree in electronic information and technology from CSU, China, in 2008 and in 2012, respectively, Ph.D. degree in electrical engineering and information tech nology from University of Duisburg-Essen, Germany in 2016. He is currently an associated pro fessor at CSU. His research interests are model-based and data-driven fault diagnosis and health monitoring, data analytics.

Chaobo Zhang (Member, IEEE) received the master's and Ph.D. degrees in civil engineering from Hong Kong University of Science and Technology, Hong Kong, in 2017 and 2020, respectively. He is currently a post-doctoral fellow in the Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen, China. His current research interests include intelligent sensing, big data analysis, deep learning and computer vision for industrial applications.

Weihua Gui received the degree of the B.Eng. in Electrical Engineering and the M.Eng. in Automatic Control Engineering from Central South University, China in 1976 and 1981 respectively. From 1986 to 1988 he was a visiting scholar at Universität-GH-Duisburg, Germany. He has been a full professor in Central South University since 1991. His main research interests are in modeling and optimal control of complex industrial process, distributed robust control, and fault diagnoses.