Ore Image Segmentation Method Based on U-Net and Watershed

Hui Li[1], Chengwei Pan[2, 3], Ziyi Chen[1], Aziguli Wulamu[2, 3, *] and Alan Yang[4]

Abstract: Ore image segmentation is a key step in an ore grain size analysis based on image processing. The traditional segmentation methods do not deal with ore textures and shadows in ore images well Those methods often suffer from under-segmentation and over-segmentation. In this article, in order to solve the problem, an ore image segmentation method based on U-Net is proposed. We adjust the structure of U-Net to speed up the processing, and we modify the loss function to enhance the generalization of the model. After the collection of the ore image, we design the annotation standard and train the network with the annotated image. Finally, the marked watershed algorithm is used to segment the adhesion area. The experimental results show that the proposed method has the characteristics of fast speed, strong robustness and high precision. It has great practical value to the actual ore grain statistical task.

Keywords: Image segmentation, ore grain size, U-Net, watershed algorithm.

1 Introduction

In the mining industry, ore grain size is an important indicator to measure the quality and efficiency of the entire beneficiation process [Budzan and Marek (2018)].There are many methods for calculating ore grain size, and the measurement based on image segmentation has now become a key method. For the segmentation of ore images, many methods have been proposed, most of which is the combination of basic steps such as grayscale, normalization, filtering and watershed segmentation. Some people have tried to improve the existing algorithms, especially the watershed segmentation algorithm [Dong and Jiang (2014); Mohanapriya and Kalaavathi (2019)].

However, the methods mentioned above have many problems in terms of robustness and generalization. Due to the influence of the actual production environment, the stones in the ore image collected outdoors often overlap each other, and the surface of the stone often

[1] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

[2] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

[3] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, 100083, China.

[4] Amphenol AssembleTech, Houston, 77070, USA.

[*] Corresponding Author: Aziguli Wulamu. Email: aziguli@ustb.edu.cn.

Received: 19 January 2020; Accepted: 04 May 2020.

CMC.doi:10.32604/cmc.2020.09806

has shadows and textures. The conventional methods of image processing often suffer from under-segmentation and over-segmentation when encountering such problems, and these methods rely on fine parameter adjustments, so their generalization is poor.

In this case, we use deep learning to solve these problems. Deep learning has developed rapidly in the field of image processing in recent years. Among them, convolutional neural networks have the characteristics of accurate segmentation and strong robustness in semantic segmentation [Shelhamer, Long and Darrell (2017); Wu, Liu and Liu (2019)]. Therefore, we collect ore images for annotation, then train the modified U-Net [Ronneberger, Fischer and Brox (2015)], and combine the marked watershed algorithm to optimize the segmentation results of neural networks. The experimental results show that the proposed method has the characteristics of fast speed, strong robustness, high precision and the great practical value of the actual ore granularity statistical task.

To sum up, the main contributions of this work can be summarized as follow: (1) In response to the real time and generalization requirements of the ore grain size statistical task, we modify the structure and loss function of U-Net, which increases the processing speed of the network by 70% and enhance the performance while processing different types of ore images. (2) For the segmentation task, we collect ore images from different scenes, design the annotation method, selectively label the ore images and use it to train the improved U-Net. The experimental results show that the obtained model has high segmentation accuracy and robustness. (3) In view of the adhesive area problem of segmentation results, we use a marked watershed algorithm to optimize, which makes the method of this article have higher accuracy and practical value in a statistical sense.

The rest of this article is organized as follows. The related work is summarized in Section 2, and the details of the proposed method are described in Section 3. The experimental results and discussion are presented in Section 4. Finally, the conclusion is drawn in Section 5.

2 Related work

2.1 Ore image segmentation method

In literature, ore image segmentation methods can be summarized into two categories. The first category is based on handmade image features. The ore image is segmented by analyzing shallow features on the pixel level of the ore image, such as thresholds, edges, and regions. The second category is based on deep learning which uses a large number of depth features of images for processing.

The first category of ore image segmentation methods is mostly based on the low order visual information of the image pixel itself. Therefore, these methods are difficult to achieve a satisfactory effect in the complex segmentation task such as ore image segmentation. Dong et al. [Dong and Jiang (2013)] proposed a complex ore image segmentation algorithm combining local adaptive thresholding and improved the watershed transform. The algorithm extracts the ore area by using the integral image and the local threshold, and combines the watershed transformation with the area combination to segment the image. The segmentation rate and the adaptability to illumination are improved, and the over-segmentation problem is effectively avoided, but the accuracy of the segmentation result needs to be improved. Zhang et al. [Zhang and Jiang (2011)] achieve the marker watershed transformation based on distance transformation and morphological reconstruction by smoothing the image using a bilateral filter. This method can effectively reduce the over-allocation rate, but most of the thresholds in the algorithm are selected by manual debugging, and the robustness is poor.

The second category, which is based on deep learning, usually achieves good results compared to traditional methods. Deep learning is a new idea for solving image segmentation in recent years. The image segmentation model with strong robustness can be obtained by studying a large number of samples. [Chen, Papandreou, Kokkinos et al. (2018); Badrinarayanan, Kendall, Cipolla et al. (2017); Zhao, Shi, Qi et al. (2017); Tian, He, Shen et al. (2019); Girshick, Donahue, Darrell et al. (2014)]. Yuan et al. used HED [Xie and Tu (2017)] to perform edge detection on ore images, then extracted the refined edges with a table lookup algorithm, and finally marked the connected regions and obtained segmentation results, which is better robust than traditional methods, but the result is not ideal [Yuan and Duan (2018)].

The method in this article is also based on deep learning. We use the modified U-Net to segment the ore image and then optimize it using a marked watershed algorithm. Not only has it achieved high segmentation accuracy, but it also improved the segmentation rate and adaptability to illumination, and it also has good generalization in ore images of different scenes.

2.2 U-Net model

U-Net is a segmentation network built on the FCN [Shelhamer, Long and Darrell (2017)] architecture, which is a typical encoder-decoder convolutional network. The author adds a number of feature channels to combine shallow features with deep features. Deep features are used for positioning and shallow features are used for precise segmentation. Its loss function is computed by a pixelwise softmax over the final feature map combined with the cross-entropy loss function. In order to reflect the importance of different pixels, the closer the distance is to the boundary of the target object, the greater the weight.

U-Net is widely used in the field of medical imaging, thanks to its unique data augmentation. In the scenario of the actual application, the organ tissue in the human body will be in a different state of rotation at any time, and sometimes the shape will become distorted when it is squeezed by other organ tissues. Therefore, the author uses excess data augmentation by applying elastic deformations. In each round of training, the original image is stretched, scaled, and translated to greatly improve the accuracy of the model.

The tasks we are dealing with have a lot of similarities to the segmentation tasks in the field of medical images. Firstly, the stacking of stones is very similar to the extrusion deformation of the tissue. Its data enhancement method is also applicable to the segmentation of ore images. Secondly, the gaps between the tissue and the gaps between the stones are closed and connected, belonging to a connected domain. They have a lot of semantic information similar to the edges inside the edges. For instance, in a cell image, there are some internal cell structure edges in the cell. And in the ore image, the ore surface also has some texture and shadow.

Therefore, making training set and training on U-Net can achieve good results in our task.

Moreover, since the ore image segmentation task has high requirements for realtime and generalization in practice, we modify the structure of U-Net to speed up its processing without reducing the accuracy. We also do the loss function. The modification makes the model based on the same training set to achieve better generalization performance on the rock image of different scenes.

2.3 Statistics of ore grain size

Ore grain size is an important indicator to measure the quality and efficiency of the entire ore beneficiation process. First, in order to measure the size of the ore we have to choose a metric, such as the length and width of the circumscribed rectangle, and so on. Then the ore grain size statistical task is to count the number of ores in different size intervals. Through it, we can also detect whether there is ore beyond the alarm threshold.

3 Proposed method

The process of our proposed method is shown in Fig. 1. The whole process can be divided into four steps: (1) Image acquisition, (2) Image preprocessing, (3) Image segmentation, (4) Grain size analysis. Through step (1), we get the ore-image for preprocessing. After step (2), we use the trained U-Net for image segmentation and the watershed algorithm for optimization. The grain size can be analyzed by traversing the closed region of the obtained binary image.

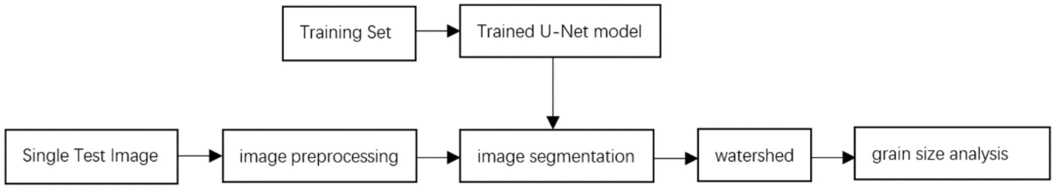


Figure 1: The process

3.1 Image acquisition

In order to solve the problem of inaccuracy caused by texture and shadow on the ore, we collect images under different lighting conditions from the production site for a total of 1500. As shown in Figs. 2(a) and 2(b) are obtained under direct sunlight at different angles. (c) is obtained during the day without sunlight, and (d) is obtained under the light of lamps at night. After that, we convert the captured image to grayscale and intercept the region of interest. Then, we use bilateral filtering to denoise the image. Finally, we equalize the image histogram and complete the image preprocessing stage.

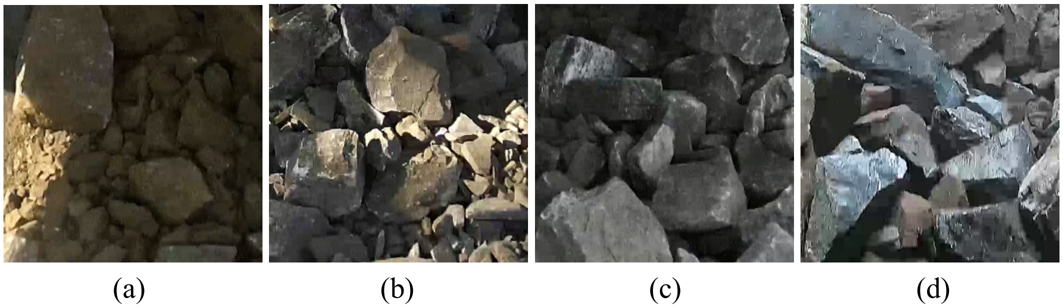


Figure 2: Image acquisition

3.2 Image annotation

In the traditional segmentation method, the problem of under-segmentation and over-segmentation is often encountered. This is because the ore has a roundness. In the pictures taken by the camera, the ore is often stacked together, so the segmentation result of different stones may adhere too, which called the under-segmentation. And for the texture and shadows on the surface of the stone are very similar to the edges of the stone, they may be treated as gaps between the stones during the segmentation, causing over-segmentation.

Based on the characteristics of this specific segmentation task, we design the annotation strategy. We define the gaps and the edges between different stones as the background class because their diversity at the pixel level is generally small. And the texture and shadow of the surface are usually inside the stone area, so we combine them with the stones as the foreground class. The semantic information of the stone itself allows the network to learn to distinguish between texture and edges. For the very thin edges between the stacked stones, we also annotate them as the background class and thicken them appropriately.

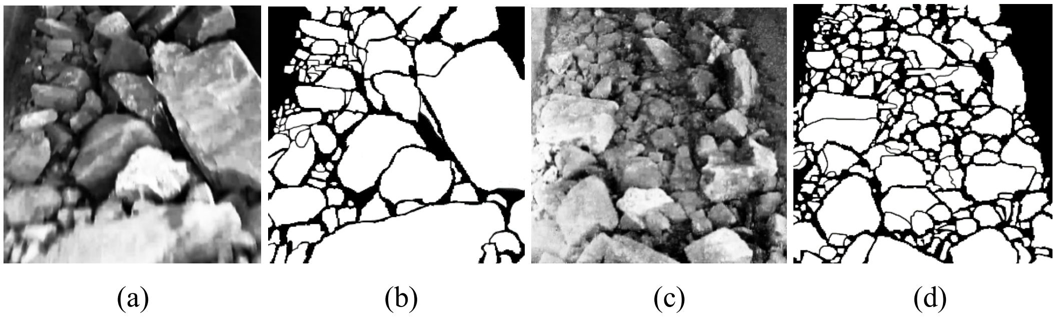


Figure 3: Annotation result

As shown in Fig. 3, no matter how small the pixel value on the stone surface is, as long as these pixel points belong to stone, they are marked as a whole; no matter how small the gap between the stones is, as long as it is completely separated, mark it as two stones. In other words, the labeling strategy is based on whether the edge of the stone is closed or not. The annotation process is done manually, (b) is the annotating result of (a) and (d) is the annotating result of (c) in Fig. 3.

The entire annotation process is iterative, which can reduce the workload and make the model very general and robust. We first select a small part of the ore image from the collected ore images for annotating, and make a training set to train the model. Then, using the trained model to process the remaining ore images, and a part of the images with results of poor prediction is selected for the next round of annotating. After repeating the iteration several times, we randomly crop each annotated image and resize the result to the same size, which enriches the multiscale information and increase the training set.

In the experiment, we select 25 images for annotating and conduct the first round of training. Then we use it to process the remaining images, select out 10 to 20 images with the effects of the worst prediction, and carry out the second round of annotating and training. The whole process has gone through six rounds, and a total of 78 images have been selected. The training set finally obtained after random crops with a total of 258 annotated images.

3.3 Modification on U-Net

In response to the realtime and generalization requirements of the ore grain size statistical task, we modify the structure and loss function of U-Net.

In the original U-Net structure, the resolution of the input image is 572×572, while the output is 388×388 because of the crop operation. However, in actual scenes, the size of the ore is usually no more than 1 meter. Using 1 pixel to represent 1 cm in the actual scene can meet the basic needs of statistical tasks, which does not pay attention to the number of tiny stones. Therefore, we reduce the resolution of the input image to 256×256 and also reduce the inference time of the network. We also delete the crop operation in the original network. This is because the crop operation will reduce the resolution of the segmentation result, resulting in a loss of semantic information and affecting the accuracy of statistical tasks. Under the modifications mentioned above, the classic U-Net structure is shown in Fig. 4.

Moreover, in order to improve the processing speed without reducing the accuracy of the model, we also adjust the convolution at every stage, using deeper features to reduce the total amount of parameters. The most intuitive way is to halve the number of convolution kernels in each layer of the network, as shown in Fig. 5(a). We define it as operation A. However, this operation may reduce the accuracy of the network. To compensate for this reduction, we can increase the number of convolution operations at each stage, making the network deeper. We define it as operation B, as shown in Fig. 5(b). Operation B adds 1 convolution operation to each stage.

We use c to indicate the number of U-Net convolution kernels in the first stage. If operation A is performed such that c is reduced, each layer of the network will reduce the corresponding ratio. The number of convolution operations at each stage is indicated by **d**. Then the classic U-Net structure can be represented by 64**c**×2**d**. By performing different combinations of operations A and B, we have designed several sets of experiments. After the experiments, we obtain the structure with the best performance of 16**c**×4**d**, and set it as our proposed model. The number of convolution operations between every stage is changed from 2 to 4, and the feature maps generated per convolution are decreasing to a quarter.

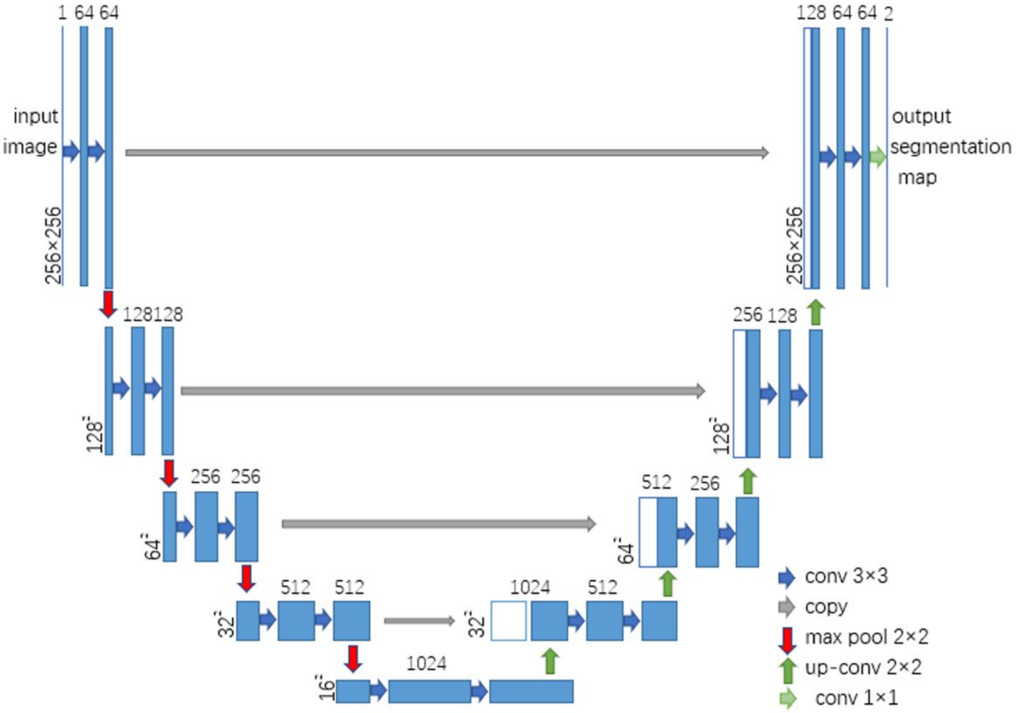
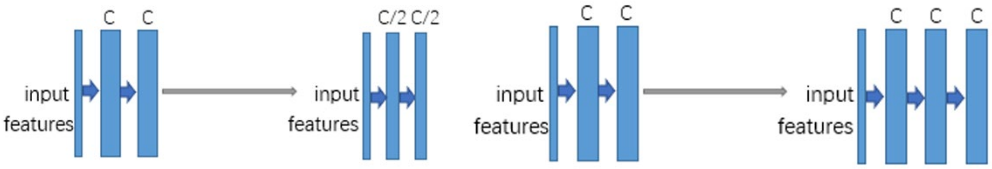


Figure 4: Classic U-Net structure



(a) (b)

Figure 5: Operation A and B

As for the loss function, we modify it from two aspects. Firstly, in the actual scene, there is a large amount of texture and shadows on the surface of stones, especially in the case of uneven illumination. The neural network may separate such stones into several regions because of the overfitting, causing over-segmentation. To avoid it, we add the influence of uniform distribution on the loss function. Secondly, the sample capacity of the foreground in the annotated image is much larger than the background, and the training of the background target is the key to solving the under-segmentation problem. In order to equalize the foreground and background in sample capacity, we increase the weight of the background. The details of our loss function are described below and we will show the benefits of this modification through a contract experiment in Section 4.

Let $a_k(x)$ denotes the activation in feature channel k at the pixel position $x \in \Omega$ with $\Omega \subset Z^2$and the number of classes is K. The softmax is defined as follow:

$$p_k(x) = exp(\, a_k(x))/(\textstyle\sum_{k'=1}^{K} exp(\, a_{k'}(x))) \tag{1}$$

Then we use cross entropy to calculate E_L and E_{Avg}. E_L denotes the cross entropy between $p_k(x)$ and true label distribution at every pixel position of Ωwhile E_{Avg} is the cross entropy of $p_k(x)$ and uniform distribution. Their expressions are shown as follow:

$$E_L = \textstyle\sum_{x\in\Omega} log(\, p_{l(x)}(x)) \tag{2}$$

$$E_{Avg} = \frac{1}{K}\textstyle\sum_{x\in\Omega}\sum_{k'=1}^{K} log(\, p_{k'}(x)) \tag{3}$$

And $l{:}\, \Omega \to \{1,\dots,K\}$is the true label of each pixel. We definef_k as the frequency of pixels belong to class k , which is obtained by calculating average on the entire annotated training set. Thus, our loss function is:

$$E = \frac{1}{f_{l(x)}}(E_L + bE_{Avg}) = \textstyle\sum_{x\in\Omega}\frac{1}{f_{l(x)}}(log(\, p_{l(x)}(x)) + \frac{b}{K}\sum_{k'=1}^{K} log(\, p_{k'}(x))) \tag{4}$$

where b is a hyperparameter and we set it to 0.1 during training.

3.4 Marked watershed algorithm

We use a watershed algorithm for binary images to further deal with under-segmentation problems. Refer to the method in Zhang et al. [Zhang and Jiang (2011)], which achieves the marker watershed transformation based on morphological reconstruction. In this article, we perform morphological reconstruction on the obtained binary image and use it as a mark image to optimize the segmentation result. The small connections between different regions can be cut off through this, which has a great improvement in the accuracy of the statistical results.

4 Experiments

4.1 Implementation details

The experiment of U-Net image segmentation is conducted under the deep learning framework TensorFlow. The network model is trained and tested by GPU with the model GeForce GTX 1080Ti. In Python3.5 environment, combined with image processing libraries such as OpenCV and Skimage, image processing, watershed optimizing and connected region markers are programmed. All our network models are trained with the annotation data set mentioned above with the same parameter settings, and the training parameters are shown in Tab. 1. We use AMSGrad as the optimize algorithm. The 'lr' represents the basic learning rate; 'beta_1' and 'beta_2' represent the attenuation coefficient of AMSGrad; 'steps_per_epoch' represents the number of iterations per epoch, and the maximum number of iterations is 50000 since epochs=50. Each iteration takes approximately 1 s. When the training iteration has reached 40000, the convergence is basically completed.

We also apply data augmentation of U-Net during training. The 'rotation_range' represents the angle of random rotation; 'width_shift_range' represents the amplitude of the random horizontal offset; 'height_shift_range' represents the amplitude of the random vertical offset; 'shear_range' represents the degree of random shear transformation; 'zoom_range' represents the magnitude of the random scaling of the image.

Table 1: Training parameters

Parameter name	Value
lr	0.0001
beta_1	0.9
beta_2	0.9
steps_per_epoch	1000
epochs	50
rotation_range	0.2
width_shift_range	0.05
height_shift_range	0.05
shear_range	0.05
zoom_range	0.05

4.2 Evaluation metrics

To prove the generalization of our model, we annotate the ore image download from the web and make it our test set. For the several modified and classic U-Net, we train them all and apply them on the test set.

First of all, we define the standard for calculating the Statistics of ore grain size. When using the segmented image for grain size statistics, it is necessary to traverse the closed region in the image to obtain the area of them. Using the obtained region area S, the number of pixels belonging to a closed region in the segmented image, we can calculate the grain size D by Eq. (5):

$$D = 2\left(\frac{S}{\pi}\right)^{\frac{1}{2}} \tag{5}$$

After getting the grain size, we need to define its statistical interval. We use 1 pixel to represent 1 cm. Since the grain size of a piece of ore is usually less than 160 cm, 160-256 of D is defined as a statistical interval. Setting a statistical interval every 20 between 0 and 160 can basically meet the needs of the statistical tasks. Finally, the total amount of ore needs to be counted as a statistical interval as well.

Since the ultimate goal of ore grain size statistics is to count the cumulative size distribution, we calculate the average of all the test results, and the result of annotation images is used as a reference.

In order to evaluate the performance of the proposed method, we have to define several metrics. Accuracy and Recall are common metrics in traditional segmentation networks. Accuracy is defined as the number of correct pixels in the segmentation result divided by the total number of pixels. And we calculate the Recall from the point of view of the foreground class, and it is defined as the number of correct pixels belonging to the foreground in the segmentation results divided by the number of pixels in the annotation images that belong to the foreground. This metric reflects the correctness of the model for foreground detection.

Then we define the error of average statistical of the ore grain size statistics, denoted by ASE. In each interval, we count the absolute value of the difference between the statistical quantity of ore and the actual quantity and divide it by the latter. ASE is obtained by calculating the mean of each interval. This metric reflects the deviation of the analyzed grain size distribution from the actual distribution. The smaller the value, the better the result. It is the core metric of the ore grain size statistical task.

4.3 Results and evaluation

Then we compare the images generated in every experiment with the annotation images using the metrics defined before. In experiment 1, we gradually perform operation A on the classical U-Net, and obtain four model structures of 64**c**×2**d**, 32**c**×2**d**, 32**c**×2**d** and 32**c**×2**d** according to the expression defined in section 3.2. For models that use the loss function defined in Section 3.3, add -l to the end of the expression. All models are trained using the same parameters and training sets. The training accuracy of each model in experiment 1 is shown in Fig. 6, which gradually decreases due to operation A, and the loss function in our article can slightly improve the convergence speed of the model.

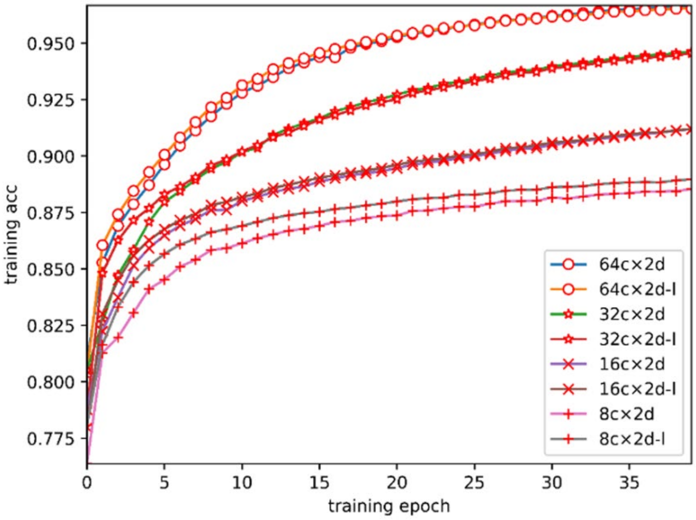


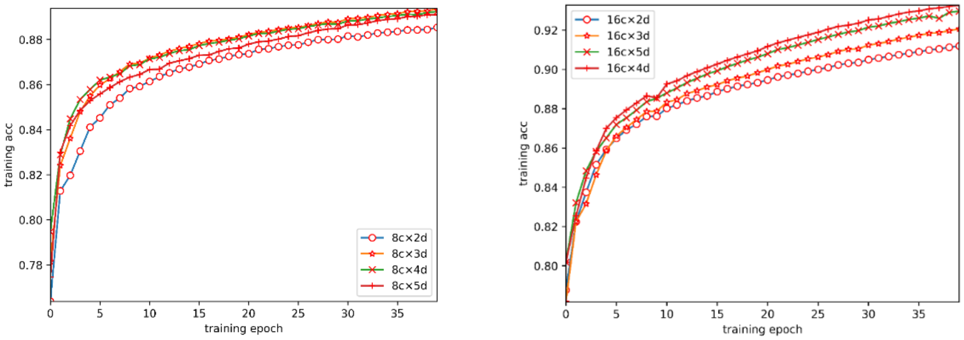
Figure 6: Training accuracy of the experiment

Tab. 2 shows the statistical results with ASE of different models on the test set. With the increasing use of operation A, the statistical results are also gradually getting worse, while using the loss in this article has improved the statistical results under various conditions. This proves the effectiveness of our proposed loss function.

Table 2: Statistics of ore grain size in experiment 1

Models	0-20	20-40	40-60	60-80	80-100	100-120	120-140	140-160	160+	Total	ASE
label	660	304	94	38	20	18	4	8	2	1148	0
64**c** × 2**d**	500	263	75	40	25	19	10	6	4	942	0.2981
64**c** × 2**d**-l	484	255	76	44	20	17	9	6	5	916	**0.2739**
32**c** × 2**d**	428	238	86	43	24	14	8	8	6	855	**0.2826**
32**c** × 2**d**-l	424	252	73	46	19	15	8	6	8	851	0.3165
16**c** × 2**d**	262	180	63	28	30	16	7	6	11	603	0.4440
16**c** × 2**d**-l	373	256	64	51	24	13	7	5	9	802	**0.3692**
8**c** × 2**d**	259	155	37	32	29	13	8	2	16	551	0.5928
8**c** × 2**d-l**	302	219	73	43	25	15	8	6	9	700	**0.3802**

In experiment 2, we gradually perform operation B on different structures in experiment 1 with the proposed loss function. Fig. 7 shows the training accuracy varies with different parameter d in experiment 2. It can be seen from Figs. 7(a) and (b) that operation B can increase the accuracy and make up for the decline caused by operation A. But in Fig. 7 (a), the accuracy of the model decreases when **d**=4 is increased to **d**=5, which indicates that operation B does not lead to continuous improvement.



(a) (b)

Figure 7: Training accuracy of experiment 2

Tab. 3 shows the statistical results with ASE in experiment 2. Under the condition of **c**=32, the effect of the model gradually deteriorates with the increase of **d**, which confirms that the effect of operation B is related to the parameter c of the network. In the case of **c**=16 and **c**=8, **d**=4 achieves the best ASE respectively.

In experiment 3, we apply the watershed operation mentioned in Section 3.4 to a different model, and use -w to represent the model using watershed operation. The effect of watershed optimization in this article is shown in Fig. 8, many small connections are cut off, and the several regions will not be considered as a whole area in statistics. This operation has a great effect on improving the performance of our methods on ore grain size statistical tasks.

Table 3: Statistics of ore grain size in experiment 2

Models	0-20	20-40	40-60	60-80	80-100	100-120	120-140	140-160	160+	Total	ASE
label	660	304	94	38	20	18	4	8	2	1148	0
32**c**×2**d-**l	424	252	73	46	19	15	8	6	8	851	**0.3165**
32**c**×3**d-**l	505	272	83	42	24	15	12	5	4	962	0.3546
32**c**×4**d-**l	511	230	78	39	23	10	10	2	10	913	0.4238
16**c**×2**d-**l	262	180	63	28	30	16	7	6	11	603	0.4440
16**c**×3**d-**l	462	267	63	40	23	16	11	4	4	890	0.3585
16**c**×4**d-**l	460	265	90	37	22	16	11	5	5	911	**0.3213**
16**c**×5**d-**l	442	224	82	44	20	14	12	3	7	848	0.4272
8**c**×2**d-**l	302	219	73	43	25	15	10	6	9	700	0.4300
8**c**×3**d-**l	437	254	87	30	27	19	9	2	9	869	0.3863
8**c**×4**d-**l	418	216	84	27	22	20	9	1	6	803	**0.3814**
8**c**×5**d-**l	446	208	67	35	24	16	10	3	5	814	0.3836

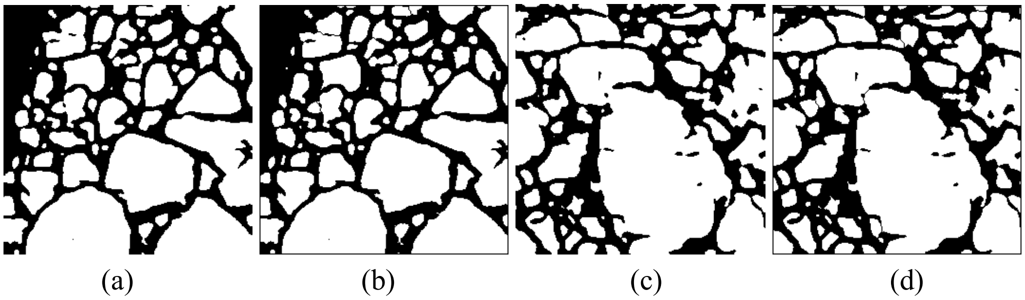

(a) (b) (c) (d)

Figure 8: Effect of watershed

Tab. 4 shows the statistical results before and after the watershed in experiment 3. We use ASE as the final evaluation metric. “Time” is the time required for the method to process a picture. Because of the running time of the watershed algorithm on our platform is 16ms, so the method using the watershed algorithm increased the time by 16ms. “Complexity” represents the number of parameters of the model, which is proportional to the inference time. In the model that does not use watershed algorithm, the classic U-Net structure achieves the best result, with the inference time of 46ms and ASE of 0.2739. The 8c×4d-l-w model (using watershed algorithm) has a running time of 32ms and ASE of 0.2016. The running time of 32c×2d-l-w model is 30ms and ASE is 0.1920. The running time of 16c×4d-l-w model is 34ms and ASE is 0.1612. It can be seen that the 16c×4d-l-w model achieves the best performance.

We can see here a promising improvement by modification. Compared with the classic u-net structure without watershed, the proposed method has improved ASE metric by 11.27% and running speed by 26.09%. Even with watershed processing, the classical U-Net structure is only 0.2009 on ASE, but the processing time is added to 62ms, which confirms the efficiency of our modifications.

Moreover, we also compare the average Recall and average Accuracy of different models on the test set. As shown in Tab. 5, it can be seen that operation A can decrease the ASE performance of the model. But the average Accuracy and average Recall on the test set of the several models are very close, which indicates that the key to the ore grain statistics task is to solve the over-segmentation and under-segmentation problems in the segmentation process, rather than the accuracy of segmentation.

Table 4: Statistics of ore grain size in experiment 3

Models	Complexity	Time	0-20	20-40	40-60	60-80	80-100	100-120	120-140	140-160	160+	Total	ASE
label	/	/	600	304	94	38	20	18	4	8	2	1148	0
64**c**×2**d**-l	31M	46ms	484	255	76	44	20	17	9	6	5	916	**0.2739**
64**c**×2**d**-l-w	31M	62ms	577	303	96	32	21	17	9	6	3	1064	0.2009
32**c**×2**d**-l	7.7M	14ms	424	252	73	46	19	15	8	6	8	851	0.3165
32**c**×2**d**-l-w	7.7M	**30ms**	518	300	87	39	22	18	8	6	4	1002	**0.1920**
16**c**×2**d**-l	1.9M	9ms	373	256	64	51	24	13	7	5	9	802	0.3692
16**c**×2**d**-l-w	1.9M	25ms	465	287	80	40	22	16	8	7	5	930	0.2318
8**c**×2**d**-l	0.48M	4ms	302	219	73	43	25	15	10	6	9	702	0.4300
8**c**×2**d**-l-w	0.48M	20ms	396	296	85	39	25	15	7	6	7	876	0.2588
32**c**×3**d**-l	11.7M	18ms	505	272	83	42	24	15	12	5	4	962	0.3546
32**c**×3**d**-l-w	11.7M	34ms	618	327	78	44	23	16	7	4	4	1121	0.2047
16**c**×3**d**-l	2.9M	15ms	462	267	63	40	23	16	11	4	4	890	0.3585
16**c**×3**d**-l-w	2.9M	31ms	550	286	73	34	22	17	9	2	4	997	0.2819
8**c**×3**d**-l	0.73M	8ms	437	254	87	30	27	19	9	2	9	869	0.3863
8**c**×3**d**-l-w	0.73M	24ms	503	303	78	40	24	17	8	4	3	979	0.2320
32**c**×4**d**-l	15.6M	27ms	511	230	78	39	23	10	10	2	10	913	0.4238
32**c**×4**d**-l-w	15.6M	43ms	618	283	92	35	21	15	10	6	5	1085	0.2459
16**c**×4**d**-l	3.9M	18ms	460	265	90	37	22	16	11	5	5	911	0.3213
16**c**×4**d**-l-w	3.9M	**34ms**	512	305	91	38	20	20	7	5	3	1001	**0.1612**
8**c**×4**d**-l	0.97M	16ms	418	216	84	27	22	20	9	1	6	803	0.3814
8**c**×4**d**-l-w	0.97M	**32ms**	492	257	90	25	22	17	6	4	2	915	**0.2016**
16**c**×5**d**-l	4.8M	25ms	442	224	82	44	20	14	12	3	7	848	0.4272
16**c**×5**d**-l-w	4.8M	41ms	527	269	83	39	23	17	9	6	4	977	0.2428
8**c**×5**d**-l	1.2M	23ms	446	208	67	35	24	16	10	3	5	814	0.3836
8**c**×5**d**-l-w	1.2M	39ms	498	245	63	39	24	17	7	4	3	900	0.2472

Table 5: Recall and Accuracy on test set

Models	64c×2d	32c×2d	16c×2d	8c×2d
Recall (%)	0.8787	0.8687	0.8569	0.8503
Accuracy (%)	0.8382	0.8351	0.8452	0.8419
ASE (%)	0.2981	0.2826	0.4440	0.5928

Finally, in order to prove the superiority of our method, we complete a comparative experiment. As shown in Fig. 9, segmentation result of our proposed method is compared with the result of the watershed transform based on the gradient correction method.

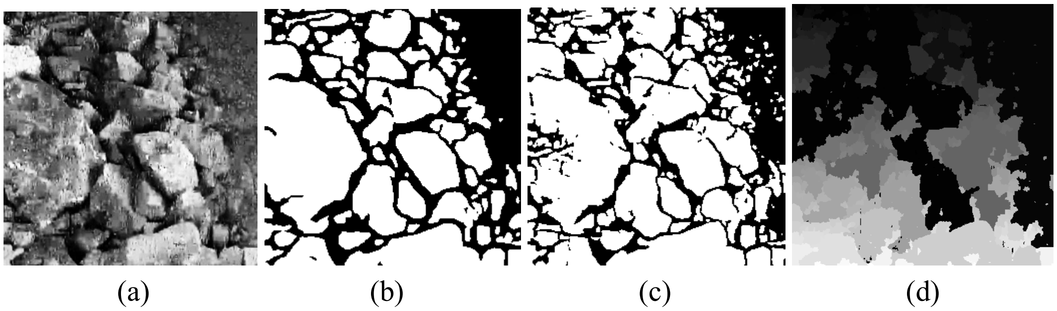

(a) (b) (c) (d)

Figure 9: Segmentation results

In Fig. 9, (a) is the preprocessed image, (b) is the segmentation result of our modified U-Net (16c×4d-l), and compared with the unmodified result (c) (64c×2d), it can be seen that the improvement of this article has achieved good results on the over-segmentation problem. The model does not separate the texture and shadow of the stone surface as edges. And (d) is the result of the watershed transformation based on the gradient correction method (WGC). Intuitively, the proposed method has a better segmentation effect. The statistical results of the experiment are shown in Tab. 6. The ASE of our method is 0.1612, while the traditional method is 0.5073, which strongly proves the superiority of the method in this article.

Table 6: Comparison with traditional methods

Models	0-20	20-40	40-60	60-80	80-100	100-120	120-140	140-160	160+	Total	ASE
Label	660	304	94	38	20	18	4	8	2	1148	0
WGC	725	427	123	54	32	24	14	8	0	1407	0.5073
64**c**×2**d**	500	263	75	40	25	19	10	6	4	942	0.2981
64**c**×2**d**-l-w	577	303	96	32	21	17	9	6	3	1064	0.2009
16**c**×4**d**-l-w	512	305	91	38	20	20	7	5	3	1001	0.1612

5 Conclusion

This article collects a variety of ore images, creates an annotated dataset, and uses the neural network-based U-Net model as well as marked watershed to address the problem of over-segmentation and under-segmentation in the task of ore image segmentation. This process achieves the purpose of image segmentation of ore images in outdoor environments. The experimental results show that the proposed method has the characteristics of fast speed, strong robustness and high precision. It has great practical value for the actual ore grain statistical task.

Funding Statement: This work was supported by The National Natural Science Foundation of China (Grant 61801019).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Badrinarayanan, V.; Kendall, A.; Cipolla, R. (2017): SegNet: a deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 12, pp. 2481-2495.

Budzan, S.; Marek, P. (2018): Grain size determination and classification using adaptive image segmentation with shape information for milling quality evaluation. *Diagnostyka*, vol. 19, no. 1, pp. 41-48.

Chen, L. C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A. L. (2018): DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 40, no. 4, pp. 834-848.

Dong, K.; Jiang, D. (2014): Automated estimation of ore size distributions based on machine vision. *Unifying Electrical Engineering and Electronics Engineering*, vol. 238, pp. 1125-1131.

Dong, K.; Jiang, D. L. (2013): Ore image segmentation algorithm based on improved watershed transform. *Computer Engineering and Design*, vol. 34, no. 3, pp. 899-903.

Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. (2014): Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, pp. 580-587.

Mohanapriya, N.; Kalaavathi, B. (2019): Adaptive image enhancement using hybrid particle swarm optimization and watershed segmentation. *Intelligent Automation and Soft Computing*, vol. 25, no. 4, pp. 663-672.

Ronneberger, O.; Fischer, P.; Brox, T. (2015): U-net: convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing &Computer-Assisted Intervention*, pp. 234-241.

Shelhamer, E.; Long, J.; Darrell, T. (2017): Fully convolutional networks for semantic segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640-651.

Tian, Z.; He, T.; Shen, C.; Yan, Y. (2019): Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, pp. 3126-3135.

Wu, H.; Liu, Q.; Liu, X. D. (2019): A review on deep learning approaches to image classification and object segmentation. *Computers, Materials & Continua*, vol. 60, no. 2, pp. 575-597.

Xie, S.; Tu, Z. (2017): Holistically-nested edge detection. *International Journal of Computer Vision*, vol. 125, no. 1-3, pp. 3-18.

Yuan, L.; Duan, Y. (2018): A method of ore image segmentation based on deep learning. *International Conference on Intelligent Computing, ICIC: Intelligent Computing Methodologies*, pp. 508-519.

Zhang, W.; Jiang, D. L. (2011): The marker-based watershed segmentation algorithm of ore image. *IEEE, International Conference on Communication Software and Networks*, pp. 472-474.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. (2017): Pyramid scene parsing network. *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, pp. 2881-2890.