Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3510313, 10 pages
https://doi.org/10.1155/2020/3510313

Research Article

A Weakly Supervised Method for Mud Detection in Ores Based on Deep Active Learning

Zhijian Huang,[1] **Fangmin Li**,[1] **Xidao Luan**,[1] **and Zuowei Cai**[2]

[1]*School of Computer Engineering and Applied Mathematics, Changsha University, Changsha 410003, China*
[2]*School of Information Science and Engineering, Hunan Women's University, Changsha 410004, China*

Correspondence should be addressed to Fangmin Li; lifangmin@whut.edu.cn

Received 1 February 2020; Revised 12 April 2020; Accepted 20 April 2020; Published 30 May 2020

Guest Editor: Chunjia Han

Automatically detecting mud in bauxite ores is important and valuable, with which we can improve productivity and reduce pollution. However, distinguishing mud and ores in a real scene is challenging for their similarity in shape, color, and texture. Moreover, training a deep learning model needs a large amount of exactly labeled samples, which is expensive and time consuming. Aiming at the challenging problem, this paper proposed a novel weakly supervised method based on deep active learning (AL), named YOLO-AL. The method uses the YOLO-v3 model as the basic detector, which is initialized with the pretrained weights on the MS COCO dataset. Then, an AL framework-embedded YOLO-v3 model is constructed. In the AL process, it iteratively fine-tunes the last few layers of the YOLO-v3 model with the most valuable samples, which is selected by a Less Confident (LC) strategy. Experimental results show that the proposed method can effectively detect mud in ores. More importantly, the proposed method can obviously reduce the labeled samples without decreasing the detection accuracy.

1. Introduction

Bauxite is usually mixed with a large amount of mud lump, which is the main impurity in alumina ore. It requires a large dose of chemical reagents (such as alkali) for removal of the mud, which increases the production cost and environmental pollution. More seriously, the mud is highly viscous, which likely blocks production equipment and affects the stability of production. At present, mud removal still relies on traditional manual operating. So, automatically detecting and removing mud from ores with AI technology is important and valuable for production cost reduction and environmental pollution.

However, it is challenging to distinguish the mud and the ore in a real scene. The reason lies in several aspects. (1) Since the mud and the ore are both in the form of lumps, the shape difference is not obvious. (2) Since ore usually cannot be cleaned thoroughly, there is little difference between the mud and the ore in color and texture (see Figure 1). Even experienced experts need careful identification to distinguish. (3) One image often contains multiple pieces of mud whose sizes vary significantly (diameter from 50 mm to 500 mm). (4) More seriously, since the ores from different mines have different compositions and contents, their color and texture have obvious differences.

Currently, there is no special method for mud detection in ores. But we can benefit from the common object detection method which is usually based on deep neural networks and trained with an amount of exactly labeled samples. There are two typical methods: the region-proposal-based method and the regression-based method. The former is also called the two-stage method. A region proposal algorithm finds candidate object regions in the first stage, and then a CNN network extracts the features and classifies candidate objects in the second stage. These methods include the R-CNN [1], the Fast R-CNN [2], the Faster R-CNN [3], the SPP-NET [4], the SSD [5], the R-FCN [6] and the newest Cascaded RCNN [7]. The latter treats object detection as a regression problem and predicts the location and category at the same time. The most representative ones are the YOLO deep neural networks, including the YOLO [8], the YOLO9000 [9], and the YOLO-v3

(a) (b)

Figure 1: The mud in ores. There are only slight differences between mud and ore in shape, color, and texture, and their scales vary significantly. The red squares with confidences are mud.

[10]. Compared with these two typical methods which emerged at the same time, the former is more accurate while the latter is faster overall.

There are several problems in directly using the aforementioned methods for mud detection. Firstly, since the mud and the ore are difficult to distinguish, the common object detection method cannot give a highly accurate result. It needs a special and finer model to give higher accuracy. Secondly, the aforementioned methods are strongly supervised, which need a large number of exactly labeled samples to train a model. Since there is similarity between mud and ores in the real scene, even experienced experts need careful identification to distinguish. So, it is expensive and time consuming to exactly label a large number of samples. Last but not least, since the ores from different mines have different colors and textures, it needs a model that can be transferred easily from a mine to another, which is important for mud detection.

To solve the challenging problem, this paper proposed a weakly supervised method based on deep active learning (AL), named YOLO-AL. The method uses the YOLO-v3 model as the basic detector, which is initialized with the pretrained weights on the MS COCO dataset. Then, an AL framework-embedded YOLO-v3 model is constructed. In the AL framework, it iteratively fine-tunes the last few layers of the YOLO-v3 model with the most important samples.

This paper focuses on the important problem of automatically detecting mud in ores, which is rarely studied. The contributions are summarized in three aspects. (1) We propose a weakly supervised method based on deep active learning for detecting mud in ores, which extensively reduces human labor for annotating training data while achieving performance comparable with the fully supervised learning approaches. (2) We propose a sample selection method based on Less Confident (LC) strategy, which selects the most valuable samples according to the confidences. The confidence of an object is calculated with the scores predicted by the YOLO-v3 detector. (3) Since the proposed method only fine-tunes the last few layers of the YOLO-v3 model with the most valuable samples, it can easily be transferred from one mine to another.

2. Related Work

Active learning [11, 12] assumes that the ground-truth labels of unlabeled instances can be queried from a database [13]. For simplicity, it assumes the labeling cost only depends on the number of queries. Thus, the goal of active learning is to minimize the number of queries. Such that, the labeling cost for training a good model can be minimized. Given a small set of labeled data and abundant unlabeled data, active learning attempts to select the most valuable unlabeled instance to query [13].

Active learning is always used in scenes where data collection is convenient while sample labeling is expensive. Kapoor et al. [14] combined active learning with Gaussian stochastic processes for object categorization. Yang et al. [15] used the AL to train a group of fully convolutional networks (FCN) for biomedical image segmentation. Sun et al. [16] proposed an AL framework based on MRF model for the spectral-spatial classification of hyperspectral imagery. Yang et al. [17] proposed a semisupervised batch mode multiclass active learning algorithm for visual concept recognition, which selects uncertainty sampling with diversity maximization. Dutt Jain and Grauman [18] proposed an active learning method for natural scene image segmentation, which achieves state-of-the-art level performance using significantly less training data.

Recently, weakly supervised learning, in which training sets require only binary labels indicating whether an image contains the object or not, has attracted considerable attention. Han et al. proposed a novel object detection framework by combining the weakly supervised learning and high-level feature learning [19]. Zhou et al. developed a transferred deep model to extract high-level features for object detection from remote sensing images by pretraining a convolutional neural network model on a large-scale annotated dataset and then transferring it by domain-specific fine-tuning [20]. Cheng. et al. trained rotation-invariant and Fisher discrimination CNN models for rotational object detection by imposing a rotation-invariant regularizer and a Fisher discrimination regularizer to the objective function [21]. Cheng. et al. introduced a new rotation-invariant layer on the basis of the existing CNN architectures and learned a

rotation-invariant CNN for object detection from remote sensing image [22].

However, there are few papers that use active learning for object detection, especially for low-distinguishable objects (such as mud and ore). This paper proposed an AL method-integrated YOLO-v3 model for mud detection in ores. The YOLO-v3 model detects the mud and predicts its class confidence and box bound confidence. Based on these confidences, the AL selects the most valuable samples to be labeled. With the gradually increasing labeled samples, a more accurate YOLO-v3 model is trained. The method will bring at least two benefits. (1) Only the most valuable samples are selected to the expert for labeling, which will reduce the number of training samples. (2) Since the expert only needs to check and modify the labels instead of relabeling, the work of labeling is reduced further.

3. Method

3.1. The Overall Framework. The overall framework of the proposed deep active learning method, named the YOLO-AL, is shown in Figure 2, which contains four basic modules: YOLO-v3 model fine-tuning, object detection, sample selection, and expert verification.

As shown in Figure 2, the proposed method is actually an iterative training process. At the beginning, we initiate a YOLO-v3 model with the weight pretrained on the MS COCO dataset. Then, it fine-tunes the last few layers with a little of labeled mud and ore samples to get a new mud detector. With the mud detector, all the unlabeled samples are tested and given confidence to each object. Based on the confidence, a sample selection method selects the most valuable samples which are sent to the expert. The expert checks and modifies the labels of these samples and adds these samples to the labeled training set. With the updated training set, the YOLO-v3 model will be fine-tuned again. The process iterates with gradually increasing labeled samples until reaching the termination condition.

3.2. YOLO-v3 Model Fine-Tuning. The YOLO-v3 model was proposed in [10] for general object detection in the nature scenes. Due to its excellent performance on the speed and accuracy, we use the YOLO-v3 as the detector. We initialize the YOLO-v3 model with the pretrained weights on the MS COCO dataset (http://images.cocodataset.org). Then, we fine-tune the last layers of the YOLO-v3 model iteratively in the AL framework.

As Yosinski et al. [23] pointed out, fine-tuning a deep neural network can preserve the general feature and overcome the difference between datasets to extract special high level features, which help us to quickly construct a new model on a new dataset. In this paper, we froze the Darknet-53 of the YOLO-v3 and fine-tuned the last layers shown in Figure 3. It is worth noting that the Darknet-53 has far more layers and weight parameters than the layers in the dashed box.

In Figure 3, the DBL unit consists of three layers: convolutional (Conv), Batch Normalization (BN), and Leaky ReLU Activation Layer. The ResUnit is a unit with the residual structure. The Resn is composed by a zero-padding, a DBL, and n ResUnit. The DBLU unit consists of a DBL layer and an upsampling layer, while the DBLC consists of a DBL layer and a convolutional layer. The Concat layer combines features at different scales.

The loss function of the YOLO-AL is defined as follows:

$$\begin{aligned} f_{\text{loss}} = {} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{obj}} \left[\left(b_x - \hat{b}_x\right)^2 + \left(b_y - \hat{b}_y\right)^2 + \left(b_w - \hat{b}_w\right)^2 \right. \\ & \left. + \left(b_h - \hat{b}_h\right)^2 \right] \\ & + \lambda_{\text{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{obj}} \left[-\log\left(p_c\right) + \sum_{k=1}^{n} \text{BCE}\left(\hat{c}_k, c_k\right) \right] \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{\text{noobj}} \left[-\log\left(1 - p_c\right) \right], \end{aligned} \tag{1}$$

where b_x and b_y are the location, while b_w and b_h are the width and the height of the predicted box. S is the number of the grid, namely, S^2 is usually set as 13×13, 26×26, and 52×52 for scales from coarse to fine. B is the predicted box number. $1_{i,j}^{\text{obj}}$ is defined as

$$1_{i,j}^{\text{obj}} = \begin{cases} 1, & \text{if there is a real object in the box,} \\ 0, & \text{else.} \end{cases} \tag{2}$$

and $1_{i,j}^{\text{noobj}}$ is the opposite of $1_{i,j}^{\text{obj}}$ which is defined as

$$1_{i,j}^{\text{noobj}} = \begin{cases} 1, & \text{else,} \\ 0, & \text{if there is a real object in the box.} \end{cases} \tag{3}$$

BCE is binary cross entropy:

$$\text{BCE}\left(\hat{c}_k, c_k\right) = -\hat{c}_k \times \log\left(c_k\right) - \left(1 - \hat{c}_k\right) \times \log\left(1 - c_k\right). \tag{4}$$

p_c is the object class probability. λ_{coord}, λ_{obj}, and the λ_{noobj} are the proportions of the three parties.

3.3. The Sample Selection. The sample selection strategy is the core of the AL [24]. Since the proposed AL method shown in Figure 2 is used for object detection, the sample selection strategy is defined based on the predicted results of the YOLO-v3 detector. It predicts the class probability of each object, based on which we can calculate the confidence. As shown in Figure 4, the YOLO-v3 predicts a vector containing 3 boxes for each grid in the feature map. Each box is a prediction of an object where p_0 is the objectness score and $p_1 \sim p_n$ are the class scores for n-classes. Here, we only consider 3 classes, namely, the mud, the ore, and others. The objectness score p_0 indicates the possibility of whether the box contains an object, namely, $p(object)$, while the class score is the posterior probability $p(\text{class} \mid \text{object})$. So, the confidence of a box can be calculated as follows:

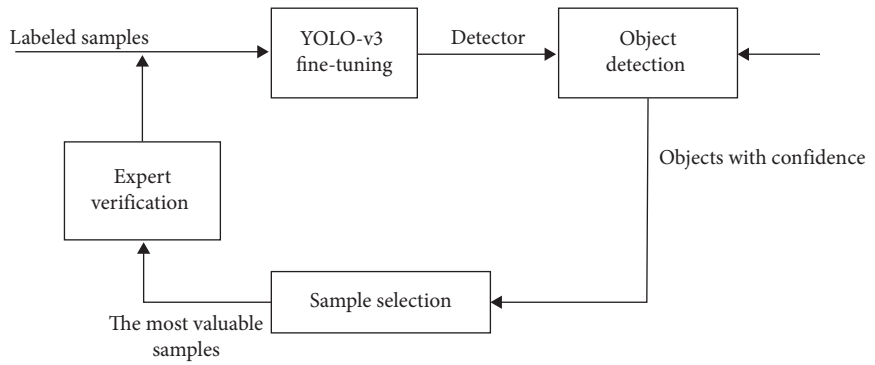


Figure 2: The flowchart of the proposed YOLO-AL method.

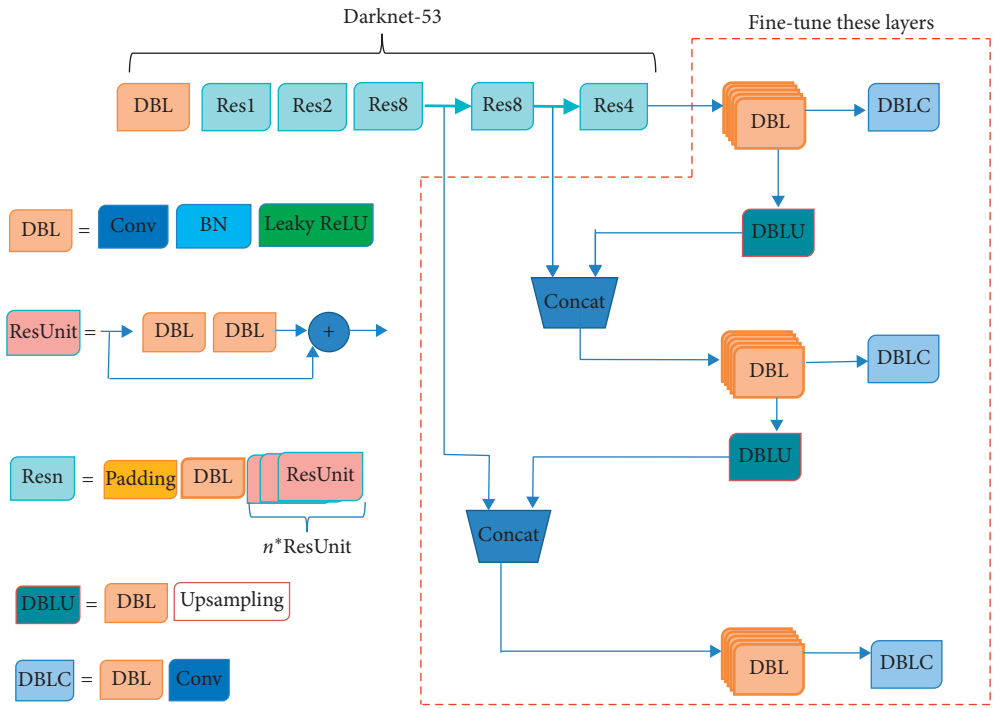


Figure 3: The structure of the YOLO-AL where the Darknet-53 is trained with COCO dataset and the layers in the dashed box are fine-tuned iteratively in an AL process.

$$p(\text{class}) = p(\text{object}) \times p(\text{class} \mid \text{object}). \tag{5}$$

In AL, the strategy of sample selection decides which sample to query or to be labeled by experts. In this paper, we consider two sample selection strategies, the random selection (RS) and the less confident (LC).

The RS method is referred to as passive selection method in contrast to the active selection methods. In the RS method, unlabeled candidates are selected randomly without any active criterion. The RS method is often served as the baseline to be compared with the active selection methods.

The LC method selects the samples with less confident based on the posterior probabilities of all the classes. When using a probabilistic model for binary classification, the LC method selects the sample whose posterior probability is near 0.5.

$$p_{\text{uncertain}} = \left| \max_{m \in L} p(y_i = m \mid \mathbf{x}_i) - 0.5 \right|, \tag{6}$$

$$\hat{\mathbf{x}}_{LC} = \arg\min_{x_i \in D_U} (p_{\text{uncertain}}), \tag{7}$$

where $\max_{m \in L} p(y_i = m \mid \mathbf{x}_i)$ means the most possible label of sample $\mathbf{x}_i$ is m and D_u is the unlabeled dataset. Specific to the problem of mud detection based on YOLO-v3, $p(y_i = m \mid \mathbf{x}_i)$ is actually $p(\text{class})$ in formula (5).

If a candidate object satisfies condition (7), it is considered to be an object containing the most useful information and should be labeled as a training sample. Considering the efficiency of the model training, we sort the mud objects in ascending order according to $p_{\text{uncertain}}$ and take the first n mud objects to be labeled. If there is an

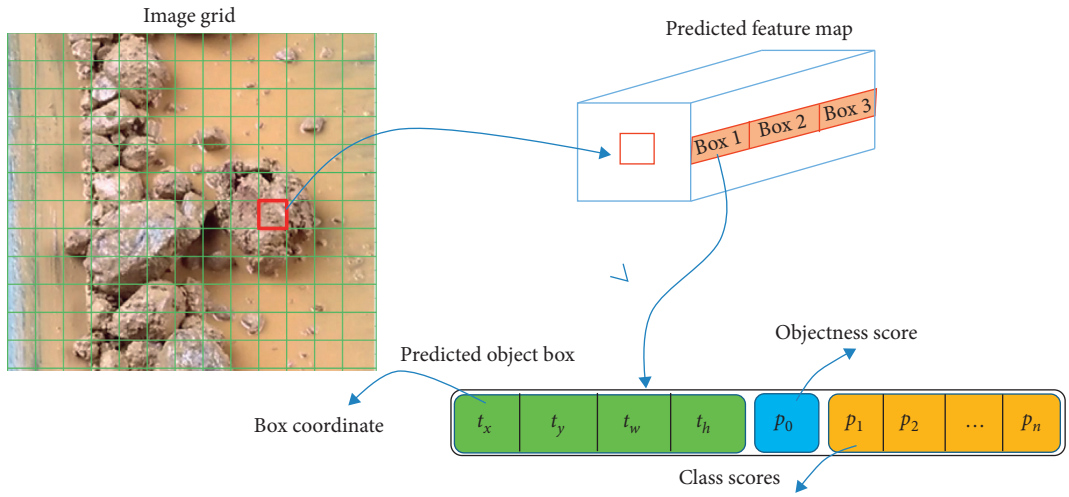


Figure 4: The object with confidence predicted by YOLO-v3. The confidence is the product of the class score and the objectness score.

uncertain object in an image, the image with its labels is put into a dataset $I_{\text{uncertain}}$ which will be recommended to expert to check and modify labels.

Since the images recommended to the expert have predicted objects with labels, the experts only need to modify the class labels or box bounds instead of relabeling them, which will also reduce the work of sample labeling.

3.4. The Algorithm. The YOLO-v3-based AL framework is provided in Algorithm 1.

4. Datasets and Experiment Design

4.1. Dataset Description. The open MS COCO [25] dataset is used to pretrain a YOLO-v3 model. It is downloaded from http://images.cocodataset.org, which contains 12 major classes and 80 subclasses. To compare with the mud and the ore in the Ore Dataset, we only focus on the dog and the cat in the MS COCO2017 dataset. The dog and the cat belong to the same major class and have high similarity to each other, which is like the mud and ore. The training set of the MS COCO contains 4385 images with 5508 label boxes for the dog and 4114 images with 4768 label boxes for the cat. Each image is 608×608 pixels.

The Ore Dataset is used to fine-tune the YOLO-v3model. It is collected from a real mine and labeled by the experienced workers. Since the actual production is more focused on a larger object, there is no label for objects with diameters less than 50 mm. It contains 5683 images, and each image is 720×640 pixels. The Ore Dataset is also organized with the format of the MS COCO. Different from the MS COCO, each image in the Ore Dataset contains ore object, but partly contains mud object. The detail is shown in Table 1.

As shown in Figure 5, the scene of the Ore Dataset is more complex than the MS COCO. One image often contains multiple pieces of mud and ores with large-scale change. The inhomogeneous slurry makes the background more complex. Nonuniform illumination and the occlusion between ore and mud further complicate the scene. So, detecting mud from the ores is more challenging than dog or cat detection.

4.2. Experiment Design. In order to verify the effectiveness of the proposed method, this paper designs comparison experiments with YOLO-v3 method, YOLO-v3 (RS) method, and YOLO-v3 (LC).

(1) *YOLO-v3.* We pretrain a YOLO-v3 model with MS COCO Dataset without cat and dog samples and fine-tune the model with the dog and cat dataset and the Ore Dataset, respectively. Different from the YOLO-AL method that iteratively increases labeled samples, it uses all samples at one time for training a fine YOLO-v3 model. Then, we observe the detection performance and compare with the YOLO-AL method.

To train the YOLO-v3 model, we take the default hyperparameters. The ratios of training and testing samples are 0.7 and 0.3, respectively. The experimental results are shown in Table 2 and are marked with a red* in Figure 6. It is worth noting that all samples were used for model training and testing at one time, which is significantly different from the increasing training samples in the YOLO-AL method.

(2) *YOLO-AL (RS).* The YOLO-AL (RS) method with the RS strategy randomly selects samples for training. It treats the training sample indiscriminately, which is the same as that in experiment 1. So, they are essentially the same with each other. The only difference is that the RS increases the labeled samples gradually, while the YOLO-v3 uses all samples at one time. However, the RS can still find out exactly how many training samples are enough for training a model.

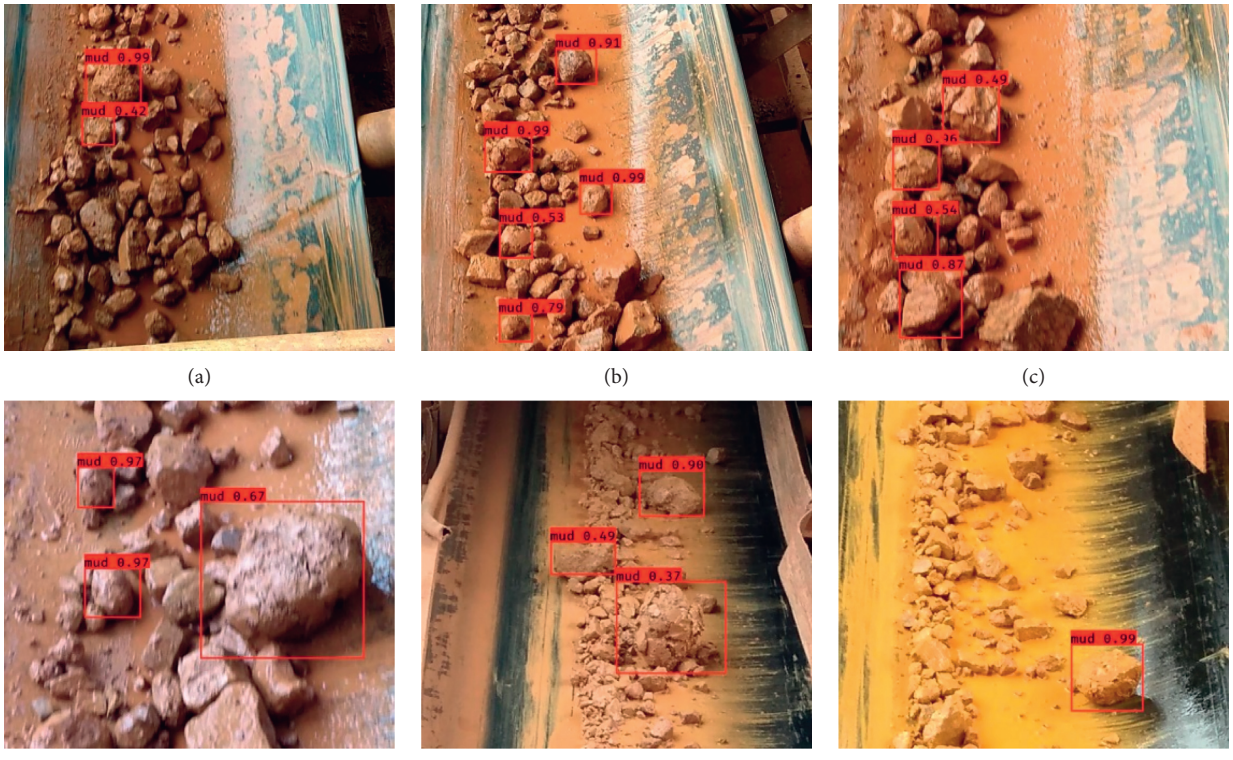
On the cat and dog dataset and the Ore Dataset, the YOLO-AL (RS) was trained with the sample increment $h = 50$. For simplicity, the expert verification is

Inputs:
I_L (labeled training image sample set)
I_U (unlabeled image set)
$M_{\text{yolo}-v3}$ (the pretrained YOLO-v3 model)
h (the sample increment)
Outputs:
$M_{\text{yolo}-v3}$ (fine-tuned by AL method)
(1) Train the YOLO-v3 model $M_{\text{yolo}-v3}$ with the labeled training image sample set I_L and update $M_{\text{yolo}-v3}$ and YOLO-v3 detector.
(2) Detect objects for each image in I_U and calculate confidence for each object.
(3) Sort the mud objects in ascending order according to $p_{\text{uncertain}}$ and take the first n mud objects contained in h images which is composed of sample set $I_{\text{uncertain}}$.
(4) For each image in $I_{\text{uncertain}}$, the expert checks the object label or box bound and makes appropriate modification. The images with verified labels form sample set I_v.
(5) Add verified samples I_v to the current training set I_L and remove them from I_U.
(6) Continue step 1 to step 5 till the set I_U is null or reaches the specified number of iterations.

Algorithm 1: The YOLO-v3-based AL framework.

Table 1: The samples of the datasets.

	MS COCO dog and cat		Ore Dataset	
	Dog	Cat	Mud	Ore
Images	4385	4114	4375	5683
Label boxes	5508	4768	14345	45742



(a) (b) (c)

(d) (e) (f)

Figure 5: Continued.

(g) (h) (i)

Figure 5: Results of the mud detection.

Table 2: The detection accuracy (AP) of the methods.

	Dog and cat/AP (%)		Ore/AP (%)	
	Dog	Cat	Mud	Ore
YOLO-v3	72.5	72.5	73.8	66.4
YOLO-AL (RS)	**72.9**	72.4	73.7	66.5
YOLO-AL (LC)	72.8	**72.6**	**75.1**	**68.3**

done instead by sample querying during the experiment. Namely, the labels of the unlabeled samples are hidden first, and when the selected samples are sent to the expert for verification, the corresponding labels are queried from the label set.

(3) *YOLO-AL (LC)*. The YOLO-AL (LC) method with the LC strategy selects samples. The other setting is the same as the YOLO-AL (RS).

4.3. Results and Analysis. To assess the effectiveness of the proposed YOLO-AL model, the average precision (AP) and mean average precision (mAP) are adopted. The AP measures the quality of bounding box prediction in the test set. If the IoU of a predicted box with the ground truth is larger than 0.5, the prediction is considered as true positive [26].

In order to avoid the randomness of detection performance, we performed five experiments for each method. Then, we calculated the mean and standard deviation of the mAP to form Figures 6 and 7, where the colored background areas represent the standard deviation floating range.

In Figures 6 and 7, the x coordinate is sample number with the increment $h = 50$, while the y coordinate is the mAP. As shown in Figure 6, the converged mAPs of the three methods have no obvious difference on the dog and cat dataset, which can also be seen from Table 2. However, the required training samples in the proposed methods are much less than those in the YOLO, as shown in Table 3. The required samples of the three methods are 2350, 3100, and 4400, respectively. The YOLO-AL (LC) is about 53.4% of the YOLO and 70.5% of the YOLO (RS).

On the Ore Dataset, the YOLO-AL (RS) is not obviously different from the YOLO-AL (LC). However, the accuracy of the YOLO-AL (LC) is about 1.5% higher than the other. The result is amazing due to the complex scene and the low discrimination between ore and mud. The required samples of the three methods are 1950, 2650, and 4400, respectively. The YOLO-AL (LC) is about 44.3% of the YOLO and 73.6% of the YOLO (RS).

The following conclusions can be drawn. (1) The detection of accuracy of the proposed method is no less than that of the YOLO-v3. (2) The required training samples of the proposed method are obviously less than those of the YOLO-v3. (3) The proposed method can be easily transferred from one mine to another. On the one hand, the proposed method uses the most valuable samples to fine-tune a model which needs less labeled samples. On the other hand, the difference between mines is less than that between Ore Dataset and COCO dataset, so it needs fewer labeled samples to transfer the model from one mine to another.

The main reason maybe lies in the training process. The AL (LC) can pick the most uncertain samples, which are most valuable for model training. In other words, the sample that cannot be accurately "understood" by the current model can provide meaningful information for improving model accuracy. The samples that cannot be accurately "understood" by the current model provide only a little meaningful information and can even be ignored.

Another reason may be that the AL with LC strategy can prevent samples from being overconcentrated in a certain area of the feature space, which may lead to biased estimates.

With the model trained by the proposed method, the Ore Dataset was tested. Partial results of the mud detection are shown in Figure 5. For the sake of clarity, the bounding boxes of the ore object are hidden here. Although the scene is complex and the mud and ore are only slightly different in color, texture, and shape, the proposed method can effectively distinguish ore and mud.

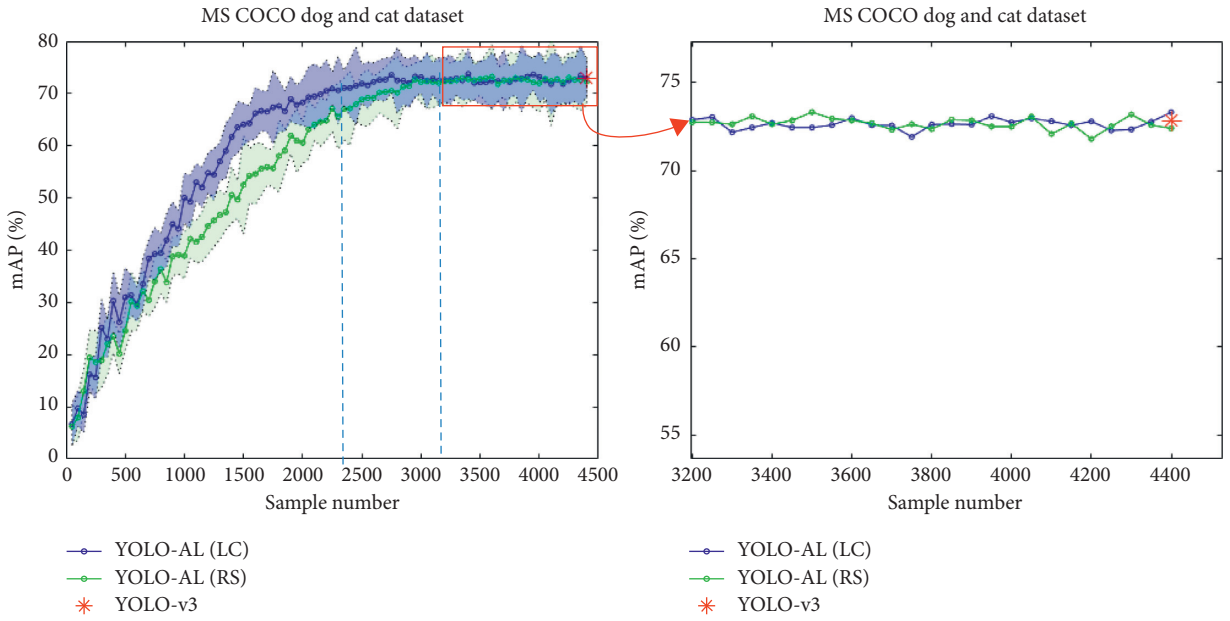


Figure 6: The mAP-sample number curves of the MS COCO dog and cat dataset.

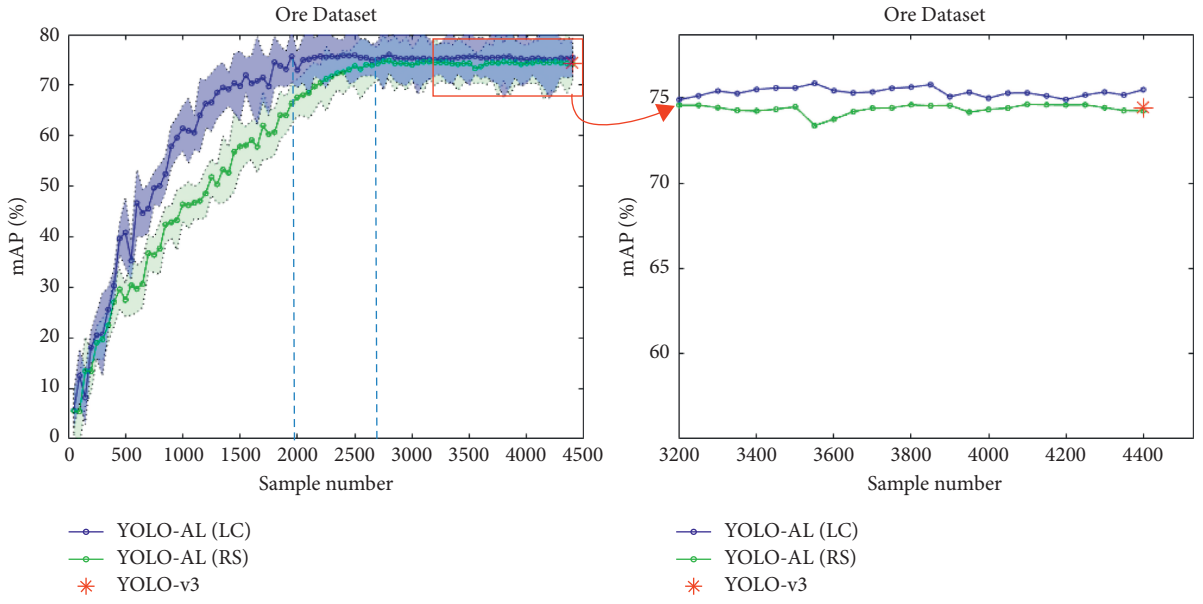


Figure 7: The mAP-sample number curves of the Ore Dataset.

Table 3: The required samples when the model converges.

	Dog and cat mAP = 73.1%	Ore mAP = 71.8%
YOLO-v3	4400	4400
YOLO-AL (RS)	3100	2650
YOLO-AL (LC)	2350	1950

The detection speed of this method is close to that of YOLO-v3. Our personal computer is 64 bit Windows 8.1 system, with Intel Core i5 CPU, 2.60 Hz, and 8 GB RAM. On the COCO dataset, the detection speed is about 30 fps, while on the Ore Dataset, the detection speed is about 28 fps. It is because the image of the Ore Dataset is a little larger than that from the COCO dataset.

5. Conclusions and Future Work

Automatically detecting mud in bauxite ores is valuable and challenging. This paper proposed a novel weakly supervised method which combines the deep active learning and the YOLO-v3 model. To select the most valuable samples, it adopts the Less Confident (LC) strategy according to the confidences of objects predicted by the YOLO-v3 detector. Then, it fine-tunes the model in the AL process with the

valuable samples every time. Experimental results show that the proposed method can effectively detect mud in ores. More importantly, the proposed method needs much fewer labeled samples than YOLO-v3 without decreasing the detection accuracy, which extensively reduces human labor for annotating training data. Also, the proposed method can be easily transferred from one mine to another, which is important for the practical application of mud detection.

In future work, we will study more appropriate sample selection strategies to further reduce the labeling cost. In addition, ores containing more types of impurities will be considered.

Data Availability

The ore images and labeled data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Scientific Research Fund of Hunan Provincial Education Department (nos. 18A376 and XJK17BXX010) and National Natural Science Foundation of China (no. 11701172).

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2015.

[2] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 91–99, Montreal, Canada, December 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2014.

[5] W. Liu, D. Anguelov, and D. Erhan, *SSD: Single Shot Multibox Detector. European Conference on Computer Vision*, Springer, Cham, Switzerland, 2016.

[6] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 379–387, Vancouver, Canada, December 2016.

[7] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, USA, June 2018.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[9] Redmon J., Farhadi A., YOLO9000: Better, Faster, Stronger, arXiv preprint, 2017.

[10] Redmon J., Farhadi A., Yolov3: An Incremental Improvement, arXiv preprint arXiv:1804.02767, 2018.

[11] M. M. Crawford, D. Tuia, and H. L. Yang, "Active learning: any value for classification of remotely sensed data?" *Proceedings of the IEEE*, vol. 101, no. 3, pp. 593–608, 2013.

[12] B. Settles, "Active learning literature survey," Technical Report 1648, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, USA, 2010.

[13] S. J. Huang, R. Jin, and Z. H. Zhou, "Active learning by querying informative and representative examples," in *Proceedings of the International Conference on Neural Information Processing Systems*, Vancouver, Canada, December 2010.

[14] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with gaussian processes for object categorization," in *Proceedings of the 2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2015.

[15] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: a deep active learning framework for biomedical image segmentation," in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2017*, pp. 399–407, Springer, Berlin, Germany, 2017.

[16] S. Sun, Z. Ping, H. Xiao, and R. Wang, "A MRF model-based active learning framework for the spectral-spatial classification of hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 6, pp. 1074–1088, 2015.

[17] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class Active learning by uncertainty sampling with diversity maximization," *International Journal of Computer Vision*, vol. 113, no. 2, pp. 113–127, 2015.

[18] S. Dutt Jain and K. Grauman, "Active image segmentation propagation," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2864–2873, Las Vegas, NV, USA, June 2016.

[19] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3325–3337, 2015.

[20] P. Zhou, G. Cheng, Z. Liu, S. Bu, and X. Hu, "Weakly supervised target detection in remote sensing images based on transferred deep features and negative bootstrapping," *Multidimensional Systems and Signal Processing*, vol. 27, no. 4, pp. 925–944, 2016.

[21] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 265–278, 2018.

[22] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.

[23] J. Yosinski, J. Clune, Y. Bengio et al., "How transferable are features in deep neural networks?" in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3320–3328, Montreal, Canada, 2014.

[24] B. Settles, "Active learning literature survey. computer sciences," Technical Report 1648, University of Wisconsin–Madison, Madison, WI, USA, 2009.

[25] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, Berlin, Germany, 2014.

[26] T. Deselaers, B. Alexe, and V. Ferrari, "Weakly supervised localization and learning with generic knowledge," *International Journal of Computer Vision*, vol. 100, no. 3, pp. 275–293, 2012.