IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 3, 2020, accepted January 1, 2021, date of publication January 18, 2021, date of current version February 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052206

Optimizing the Sediment Classification of Small Side-Scan Sonar Images Based on Deep Learning

XIAOMING QIN, XIAOWEN LUO, ZIYIN WU, AND JIHONG SHANG
Second Institute of Oceanography, Ministry of Natural Resources, Hangzhou 310012, China
Key Laboratory of Submarine Geosciences, Ministry of Natural Resources, Hangzhou 310012, China

Corresponding authors: Xiaowen Luo (cdslxw@163.com) and Ziyin Wu (zywu@vip.163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFC1521700 and Grant 2020YFC1521705; and in part by the National Natural Science Foundation of China under Grant 41830540, Grant 41676037, Grant 41906069, and Grant 42006073.

ABSTRACT Acoustic seabed classification (ASC) is a fast and large-scale seabed sediment survey method. In particular, combining it with an automated classifier can theoretically achieve fast automatic seabed sediment classification. However, owing to the cost of sampling, a lack of labeled data for sediment classification based on seabed acoustic images impedes the training and deployment of classifiers. Herein, we use shallow-water, side-scan sonar images collected from the Pearl River Estuary combined with deep learning to study sediment classification and optimization methods for a small dataset of seabed acoustic images. In this paper, we applied different and deeper convolutional neural networks (CNNs) and used grayscale CIFAR-10 for pretraining to achieve large-span parameter migration and improve model performance. The best result in the experiment is a 3.459% error rate achieved by ResNet after fine tuning, verifying the improvement brought by our fine tuning strategy and the deeper models used in such tasks. The results of data enhancement based on generative adversarial networks (GANs) indicated that this method can improve the accuracy of sediment classification; however, the effects of GANs are limited and they are computationally expensive. Overall, our findings resolve, to an extent, the dilemma of using small datasets of seabed acoustic images for sediment classification and provide a framework for future studies on sediment classification, which has a certain significance in helping people better understand the seabed.

INDEX TERMS Acoustic seabed classification, side-scan sonar, deep learning, convolutional neural network, pretraining, generative adversarial network.

I. INTRODUCTION

Inspired by the great success of deep learning (DL) in computer vision and related fields, research on the applications of DL to underwater imaging has also begun. There are two main types of underwater images—optical photographic images and acoustic images—which typically include underwater targets, seafloor topography, and seafloor sediments, among others. Seabed sediment classification is used to investigate the type and distribution of seabed sediments, which is of great significance to marine geology and related research. However, the traditional seabed sampling method is time-consuming and expensive, and it is also difficult to cover a large seabed area. Therefore, an effective-cost method for seabed sediment classification is necessary.

Acoustic seabed classification (ASC) has been studied for a long time [1], and its basic principle is to realize classification through the information contained in back-scattering (BS) intensity. When sound waves are transmitted to the seabed, they will undergo complex reflection and refraction and return in the direction of the incident angle, which is the BS intensity. It is usually related to surficial sediments properties, such as hardness, grain size, and roughness, which is the basis of ASC [2]–[4]. At present, there are many pioneers in the research of BS-based sediment classification, such as the geoacoustic inversion method [5]–[9] and statistical algorithms [10]–[14]. These studies have proved the effectiveness of this method of ASC.

Side scan sonar (SSS), as a large-scale and rapid seabed detection equipment, has been widely used in missions such as seabed topography and geomorphology survey. The principle of SSS involves transmitting dense beams to the seabed, and receiving and recording the BS intensity through the

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

transducer. Some researchers have studied the relationship between SSS data and seabed sediments [15], and the application research of seabed sediment classification based on SSS images shows the effectiveness of this method [16]–[18]. However, relying on manual operation when classifying the sediments based on SSS images is a time-consuming process. Therefore, the automatic seabed sediment classification method based on acoustic images has become a research focus in related fields. Sediment classification based on SSS or other acoustic data usually requires some known samples or ground truthing, but such ground truthing is usually very scarce because seabed sampling is expensive and time-consuming. Therefore, it is difficult to construct an automatic SSS-based sediment classification method with small datasets.

There are many precedents in the field of sediment classification which are based on seabed acoustic images and use classic machine learning algorithms, such as support vector machines (SVMs) [19], decision trees [20], k-means [18], [21], and back-propagation neural networks (BPNNs) [22]–[24]. Paired with a combination of such machine learning algorithms and feature engineering, the end-to-end training method of DL omits complex feature engineering, rendering it more convenient in the application. As a representative algorithm in DL, convolutional neural networks (CNNs) have facilitated considerable achievements in various fields, especially in computer vision, making it the current mainstream method for processing image data.

As convolutional neural networks are excellent algorithms for processing image data, they are suitable for dealing with seabed acoustic images; and there have been multiple studies in which CNNs were applied to sediment classification [25], [26], target detection [27]–[31], and semantic segmentation [32]–[34]. The results yielded have been favorable. Compared with traditional algorithms, CNNs can automatically learn data features without the need for feature engineering and other preprocessing steps required by traditional algorithms, and theoretically have better generalization and application deployment capabilities. At present, CNNs have made considerable development, especially deep CNNs with better high-dimensional feature extraction capabilities have been proposed and applied. However, as mentioned earlier, there is usually a lack of sufficient labeled data in SSS-based sediment classification tasks, a problem faced when applying CNNs to the task of the current study. Compared with the dataset that usually have thousands of samples in the field of computer vision (such as ImageNet), our dataset is actually very small, so this is a typical DL application based on small dataset. Under such data conditions, what kind of CNNs can achieve the best performance is not conclusive, so in the experiment we test CNNs of different depths and complexity. Besides, in this paper, we combine transfer learning and generative adversarial networks (GANs) in DL to optimize CNN-based sediment classification under a small SSS dataset.

To the best of our knowledge, fine tuning has already been applied in sonar image recognition and target detection [35]–[38], but it is rarely used for sediment classification based on SSS data. Therefore, we believe that it is necessary to verify its feasibility for such applications. However, due to the lack of mature relevant datasets, we can only focus on some existing datasets of computer vision, such as CIFAR-10 [39]. The CIFAR-10 dataset contains 10 classes of images, such as cars, airplanes, whereas the content of our SSS image dataset is of three types of sediments, and there is almost no similarity between the two datasets. There is little precedent for the transfer of model parameters between two such extremely different datasets in sediment classification based on a small SSS dataset; thus, we termed it fine tuning based on large-span parameter migration (LSPM) and verified its utility.

Data enhancement is derived from the field of data optimization. In addition to performing a certain geometric transformation on the original data, a generation model such as a GAN [40], is also a means by which simulation data may be generated to expand the dataset. At present, some GANs have been applied in the study of seabed acoustic images for data enhancement, such as the generation of synthetic aperture sonar (SAS) data [36], [41] and sonar wreck images [37]. We believe that generating new SSS data by GANs can enrich the feature expression of a small SSS dataset, which could optimize sediment classification performance.

In this study, we apply deeper CNNs, LSPM-based fine tuning, and GANs to achieve high-accuracy sediment classification based on a small SSS image dataset. We introduce several commonly used CNNs for classification testing and migrate the model parameters, pretrained on the grayscale CIFAR-10 (GCIFAR-10) dataset, to our target dataset to fine tune the CNNs. Our experimental results verify that fine tuning based on LSPM can greatly improve accuracy and training stability. In particular, compared to SVM combined with feature engineering, significant improvement in accuracy is evident. According to the results, the ResNet-4-2 in our experiment was able to achieve a very low error rate of 3.459% after fine tuning, which is suitable for the target mission. We also expanded the training dataset by applying GANs to enrich the small SSS dataset, and our results demonstrate that the enhancement of GANs can improve sediment classification to a certain extent; but it is limited in application and computationally expensive. Overall, the results of this study can be used to alleviate the problem of small datasets in sediment classification to a certain extent. We believe our research has practical meaning in helping us to understand the distribution of seabed sediments, topography, and even the distribution of some biological habitats.

II. DATA

The study aimed to classify sediments from a small SSS image dataset and study the optimization of such classification. The data we used were taken from SSS images obtained in the Pearl River Estuary as part of the "China Offshore and Ocean Comprehensive Survey and Evaluation Special." The micro-geomorphology and sediment types of the Pearl River Estuary are diverse. Based on an analysis of the acquired

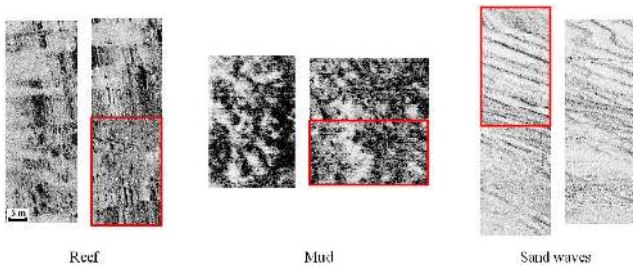


FIGURE 1. SSS image data used in this study.

scanning images, there are large areas of sand waves and reefs distributed in the estuary. Mud is concentrated in the inner Lingtingyang shoal area, where the water depth is relatively shallow.

Since the data we obtained is post-processing data, we only have partial experimental parameters. In this survey, we used a DF1000/560D digital dual-frequency SSS system (EdgeTech, USA), a differential global positioning satellite system, an SDH-13D depth sounder (South Surveying & Mapping Instrument Co., Ltd., China), and a CAP-6600 Chirp II shallow profiler system (Datasonics, Inc., Australia). The SSS operating frequency was 500 KHz, and the analog-to-digital (A/D) resolution of the SSS system was 12 bits/sample. We estimate that its working depth is about 10 meters from the water surface. The width and height of a pixel in the SSS images correspond to 0.2 m each. The specific image data are shown in Fig. 1.

SSS data labeling is based on actual seafloor grab samples. The SSS dataset we used in this research was very small and, as shown in Fig. 1, the SSS images of each sediment type were evidently separable; this was suitable for our research. In practice, grab sampling is used to obtain ground truth in a small area, and then applied it to train the classifier. Specifically, the very few seabed samples resulted in a very small training set, and it is impossible to sample from the entire dataset like random sampling. Therefore, to get closer to the actual situation, we cut out about a quarter of the original data as a training set (the area in the red rectangle in Fig. 1), and use the rest as training set. We believe that this division is better aligned with actual application scenarios.

The original data were cut using a window with a core size of 20 × 20 pixels to avoid losing feature information, and 2 pixels were filled around the window to appropriate account for the influence of the surrounding content of the core, so each sample after cutting is a picture with a size of 24 × 24 pixels. The training dataset adopted sliding-window sampling, with a spacing of 6 pixels. Such a continuous sampling method can theoretically ensure the feature richness, and can also expand the training set as much as possible. For the testing set, the continuous sampling method obviously has no meaning, so there is no intersection between the sliding windows when cutting. After cutting with different sampling methods, the total number of the training set was ∼900, and the number of testing set was ∼300. It is worth noting here that the reason why the training set, which accounts for about a quarter of the original data, can be obtained with more samples is that we adopt the small spacing (6 pixels) sliding window sampling method.

As the original datasets were very small, we divided the cutted images into training set and testing set, without setting a verification dataset. In our experiments, we use the training set to train the model, and the testing set to measure the performance of the algorithm. However, the division of data will inevitably affect the results, which can easily lead to unreliable results. To weaken the influence of dataset division, we combined different parts of the original data as a training dataset in turn and finally divided the data into 64 subdatasets (SCH-0, 1, 2, . . . , 63). This segmentation method is very similar to cross validation, in which a quarter of each type of sediment samples are selected in turn to form a training set (as shown in the red rectangle area in Fig. 1). Our model was then applied to each of these 64 datasets, which weakened the impact of division as much as possible and made the experimental results more reliable.

III. METHOD

A. CONVOLUTIONAL NEURAL NETWORKS

The basic purpose of a CNN is to extract high-dimensional features through the combination of a convolution kernel, nonlinear activation function, and pooling layer, whose structure is shown in Fig. 2. During forward propagation, the convolution kernel automatically extracts the features of the input. After convolution, features are generally mapped by a nonlinear activation function, such as the widely used rectified linear unit (ReLU) function [42], given by

$$ReLU\ (x) = max(x, 0). \tag{1}$$

Down sampling underlies the nonlinear activation function, which usually employs maximum or mean pooling. The CNNs typically output prediction values through the softmax function, and the loss function is calculated according to the output of the CNNs and the real label. The loss function is the key to training CNNs through a back-propagation (BP) algorithm, and CNNs rely on BP algorithms for training to update the network parameters. The cross-entropy formula is generally used in CNNs to calculate the loss value (LOSS), given by

$$LOSS = J\ (W, b) = \frac{1}{N} \sum_{i}^{N} \sum_{j}^{M} y_{ij}\, log(p_{ij}), \tag{2}$$

where W is the convolution kernel parameter, b is the bias, N is the number of samples in the batch, M is the number of classifications, y_{ij} is the label value of the i-th sample on class j, and p_{ij} is the predicted value of the i-th sample output by the CNNs on class j. These operations are described in detail by Goodfellow *et al.* [43].

Since their introduction, the depth and width of CNNs have increased, and various structures have been introduced to optimize their performance. As a pioneer in CNNs, LeNet-5 (hereafter, "LeNet") [44] has a small number of parameters

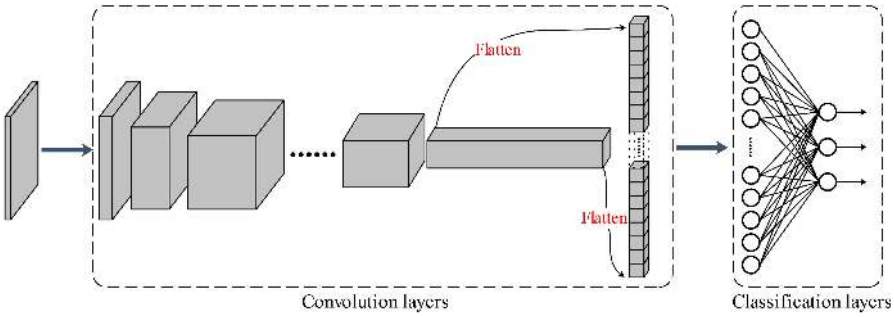


FIGURE 2. Basic structure of CNNs, wherein the convolution layer is used to extract features to form a high-dimensional feature vector, and the classifier establishes the mapping relationship between the feature vector and the output.

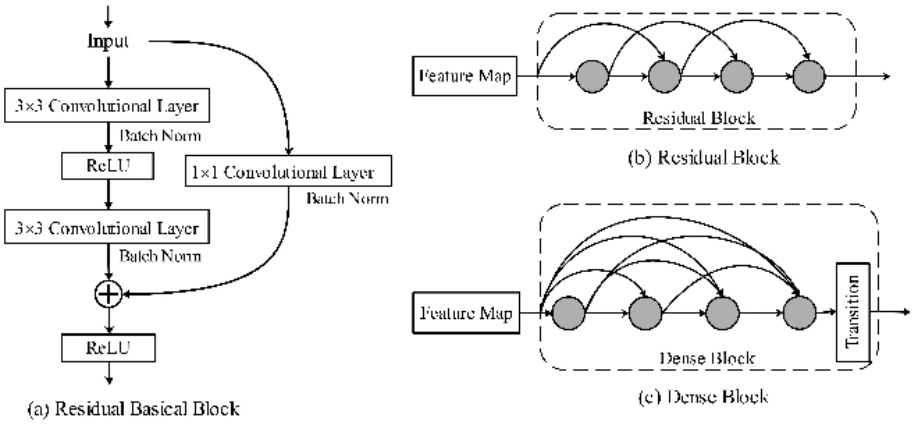


FIGURE 3. Basic structure of a (a) basic residual block, (b) residual block, and (c) dense block.

and a simple structure, which is designed to recognize handwritten digits. Meanwhile, AlexNet [45] achieved a breakthrough on ImageNet in 2012. As an exploratory means of developing CNNs, VGG [46] has achieved the same receptive field as large convolution kernels with fewer parameters by combining convolution kernels, and thereby increasing the depth of CNNs and improving performance. A milestone was achieved for CNNs with the development of ResNet [47], which connects input and output by introducing a residual module design, as shown in Fig. 3a. The mathematical expression of the residual is

$$Output = F(x) + x \tag{3}$$

where x is the input and $F(x)$ is the convolution operation. The residual structure utilizes multiscale feature information and alleviates network degradation. Compared to the ResNet (Fig. 3b), DenseNet [48] concatenates all of the output from the previous layer in the depth dimension as input (Fig. 3c). In theory, DenseNet makes better use of multiscale feature information and reduces the number of parameters; however, the calculation speed is also reduced due to excessive concatenation and memory operation.

The seabed sediment classification based on SSS images is actually a computer vision task—tasks for which CNNs are theoretically suitable. In particular, the end-to-end structure of CNNs makes feature engineering and other preprocessing work on the original data unnecessary. At present, the application of CNNs to sediment classification based on small SSS datasets lacks theoretical guidance. Moreover, the model choice needs to meet both performance and speed requirements. In this study, various types of CNNs were used to measure and evaluate their performances, and we aimed to determine the optimal solution in the target scenario.

B. FINE TUNING IN TRANSFER LEARNING

Solving the difficulties caused by small datasets is one of the most pressing challenges in DL. In this section, we present methods of optimizing sediment classifications based on small SSS datasets by fine tuning according to transfer learning. The key contribution of fine tuning in transfer learning is that it provides a priori knowledge to CNNs, so that they have better initialization parameters and generalization capabilities. The fine tuning of CNNs can be regarded as a method of obtaining better initialization parameters for convolution kernels, which also appears to provide CNNs with a better starting point. A pretrained model already can extract shallow basic features and deep high-dimensional features. Therefore, when fine tuning on a target dataset, the updating

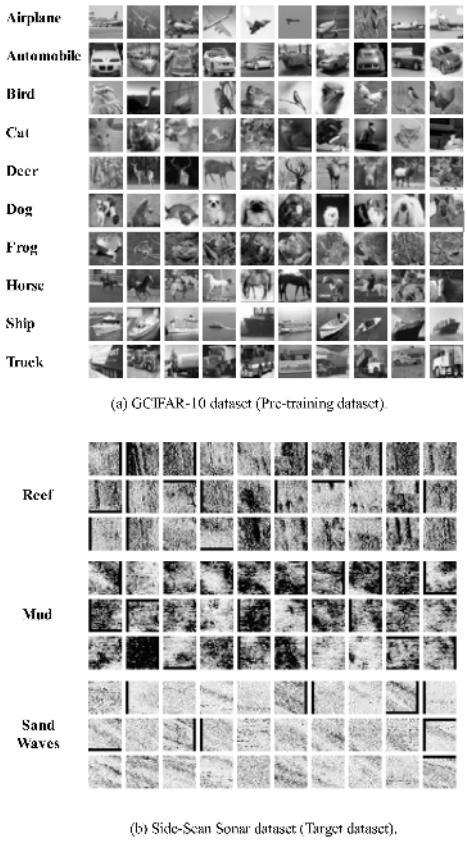


FIGURE 4. (a) Grayscale CIFAR-10 (GCIFAR-10) dataset for pretraining and (b) SSS image dataset for sediment classification.

of the parameters of the convolutional layer during training generally uses a small learning rate or is frozen, as it will otherwise cause losses to the existing feature extraction capabilities. The subsequent classification layer (usually a fully connected layer) is then redesigned and retrained because it establishes a mapping relationship between high-dimensional input features and output results.

Sediment classification is mainly based on textures and other features in sonar images. In previous studies, seabed acoustic image-based fine tuning has been applied to the detection and classification of underwater targets [37], [38], as well as to tasks like the classification of sonar images [35] and synthetic aperture sonar (SAS) image data [36]. In our experiments, we used the GCIFAR-10 as our pretraining dataset, and our sediment classification dataset was based on the division of SSS image data, as shown in Fig. 4. From the comparison of these two datasets, it can be seen that there is a large difference between them. Our assumption is to pretrain the model on GCIFAR-10 and then migrate the parameters of the convolution layers to our target mission, which we term LSPM. To the best of our knowledge, there is currently no precedent for the application of such fine tuning methods to the classification of sediments based on small SSS image datasets. If our assumptions are correct, it means that, given the current lack of relevant available seabed acoustic image datasets, fine tuning based on LSPM is useful, and it will provide insights in related future studies.

C. DATA ENFORCEMENT BASED ON GANs

Transfer learning is the optimization of model parameters, whereas data enhancement is the optimization of data. In this paper, we included the geometric deformation data enhancement method (i.e., random flip, random mirror transformation) by default, but the effect is still limited. To enhance the data used in this study, we expanded small datasets to enrich the expression features in the training data. The GANs were first proposed by Goodfellow [40] and have been widely used in generation tasks. The theory of GANs is based on the adversarial relationship between a "generator" and a "discriminator," and its key contribution is to reduce the distance between the distributions of the model and real data, so that the generated image of the generator is as close as possible to the real image.

Considering that our training dataset included three sediments, training a GAN model for a single type of sediment would waste time, and the size of the training data would be very small. Therefore, we used conditional batch normalization (CBN) [49], which is an improved form of batch normalization (BN) [50]. The basic principle of BN is shown in (4), where $B = [F_1, F_2, \ldots]$ is a batch, E stands for the exception, c represents the channel, ϵ is a constant damping factor with numerical stability, and γ and β are the learned BN parameters. The improvement of CBN over BN is the introduction of conditional labels, which can change γ and β according to the input labels, as shown in (5).

$$BN\left(F_{i,c} \mid \gamma_c, \beta_c\right) = \frac{F_{i,c} - E_B(F_{\cdot,c})}{\sqrt{Var_B\left[F_{\cdot,c}\right] + \epsilon}}\gamma_c + \beta_c \tag{4}$$

$$\begin{cases} \gamma_{CBN} = \gamma + \Delta\gamma \\ \beta_{CBN} = \beta + \Delta\beta \end{cases} \tag{5}$$

Therefore, it can generate different classes of data. By adding CBN, the algorithm can be directly deployed on the entire training dataset. As a branch of GAN development, the Wasserstein GAN-gradient penalty (WGAN-GP) [51], [52] improves the stability, as well as the performance of GANs by adding a GP and modifying the LOSS function. The LOSS function of the WGAN-GP is shown in (6), which includes a GP term $\lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\parallel \nabla_{\hat{x}} D\left(\hat{x}\right) \parallel -1)^2]$, so that the weight parameters satisfy the 1-Lipschitz limit. In particular,

$$\begin{aligned} LOSS_{WGAN-GP} = {} & \mathbb{E}_{\tilde{x}\sim\mathbb{P}_x}\left[D\left(\tilde{x}\right)\right] - \mathbb{E}_{x\sim\mathbb{P}_r}\left[D\left(x\right)\right] \\ & + \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\parallel \nabla_{\hat{x}} D\left(\hat{x}\right) \parallel -1)^2] \end{aligned} \tag{6}$$

where D represents the discriminator, λ represents the weight parameter, and $\hat{x} = \epsilon x - (1-\epsilon)\tilde{x}$.

To combine with CBN, the class information needs to be outputted at the end. Additionally, as the LOSS value of the classification ($LOSS_{CLASS}$) uses the cross-entropy function mentioned earlier, the LOSS of the algorithm is combined

TABLE 1. Parameters and FLOPs of different CNNs and their training time and testing time.

CNNs	Parameters (Million)	FLOPs (Million)	Training time (s/batch)	Testing time (s/batch)
AlexNet-BN	4.595	40.056	0.011	0.003
LeNet-BN	2.221	10.207	0.009	0.003
DenseNet-52	0.263	111.442	0.080	0.019
DenseNet-100	0.766	304.704	0.162	0.034
DenseNet-151	1.511	580.443	0.250	0.051
ResNet-3-2	2.775	422.332	0.024	0.005
ResNet-3-3	4.325	649.742	0.032	0.007
ResNet-4-2	11.169	556.707	0.030	0.007
VGG13	36.697	282.894	0.025	0.005
VGG16	42.157	405.874	0.031	0.006

with $LOSS_{\text{WGAN-GP}}$ and $LOSS_{\text{CLASS}}$, as shown in (7), where α represents the weight parameter.

$$LOSS = LOSS_{WGAN-GP} + \alpha LOSS_{CLASS} \tag{7}$$

The specific structural flow of the entire algorithm is shown in Fig. 5, and is actually a generation algorithm based on the fusion of ACGAN [53] and WGAN-GP. Such a structure can generate specified types of image data according to the conditional label input.

IV. RESULTS

The results of our experiments can be divided into two parts, based upon the goals: (1) optimization of CNNs; (2) data enhancement based on DL.

A. COMPUTATIONAL SETUP

The configuration of the workstation we used in the experiments was as follows: the central processing unit (CPU) was an Intel Core-i9-9820X with a C422 motherboard (Intel Corp., USA), the memory was 32 GB, and the graphics processing unit (GPU) was a single GeForce RTX2080Ti (Nvidia Corp., USA). We used Python as the programming language. The architecture used in the CNNs was PyTorch v. 1.2.0, and for GANs, it was TensorFlow v. 1.17.0.

B. OPTIMIZATION OF CNNs

In our experiments, we tested multiple groups of CNNs and verified the impact of LSPM-based fine tuning on their classification accuracy and also added the traditional algorithm of SVM combined with feature engineering as a comparison.

As shown in Table 1, to measure its influence, we varied the depth of the models for the ResNet, DenseNet, and VGG architectures. That of ResNet-3-2 revealed that there were 3 residual blocks in the model, with each residual block having 2 basic residual blocks; the numbers behind DenseNet and VGG represent the depth of the network. The terms AlexNet-BN and LeNet-BN imply that we added BN layers to improve these classical CNNs. For fine tuning, we modified the sample size in our SSS dataset to the same format as GCIFAR-10, namely, (32,32,1). Regarding the specific parameter settings, the batch size was 16, the optimizer used stochastic gradient descent (SGD), and the epoch was set to 200. For the learning rate, the initial learning rate of AlexNet, LeNet and VGG was 0.01, whereas that of ResNet and DenseNet was set to 0.1. However, when fine tuning, we set the learning rate of the convolutional layer part to be the learning rate multiplied by 0.1 (compared to frozen parameters of convolutional layers, we found this method to be more effective). It is worth noting that we used geometric transformations to enrich the training set, including random mirroring, random resized cropping, and random flipping.

TABLE 2. Experimental results of fine tuning with the lowest error rates indicated in bold.

Model	Test accuracy w/o fine tuning (error rate)	Test accuracy w/ fine tuning (error rate)
AlexNet-BN	**4.672 %**	4.769 %
LeNet-BN	5.325 %	**4.664 %**
DenseNet-52	11.402 %	**4.689 %**
DenseNet-100	13.906 %	**4.653 %**
DenseNet-151	17.688 %	**3.629 %**
ResNet-3-2	5.949 %	**4.279 %**
ResNet-3-3	6.439 %	**4.040 %**
ResNet-4-2	6.501 %	**3.459 %**
VGG13	6.791 %	**6.414 %**
VGG16	**6.798 %**	6.831 %

TABLE 3. Experimental results of SVM.

Model	Test accuracy (error rate)
SVM	13.101 %
SVM+PCA	10.510 %

To better compare the accuracy of CNNs to existing traditional algorithms, we considered the traditional algorithm based on an SVM [54] and feature engineering. As a well-known algorithm, SVM occupies an important position in the field of classification or other missions. In our experiment, the parameter C of the SVM was set to 1, the kernel function was the radial basis function (RBF), and the degree of the kernel function was set to 3. In terms of feature engineering, we extracted a total of 12 features, including the gray level co-occurrence matrix features [55], [56] and the mean, variance, and standard deviation of the grayscale. In order to optimize the performance of SVM, we introduce principal component analysis (PCA), a commonly used data compression method in feature engineering. Thus, we set two groups of experiments: the first group used the extracted original features; the second group used PCA to compress features into 6 dimensions. Besides, the features were normalized before being inputted into the SVM. The results are shown in Table 3.

Table 2 shows the averaged classification results of the application of CNNs to all subdatasets before and after fine tuning. Furthermore, Fig. 6 provides an illustration of these results before and after LSPM-based fine tuning under

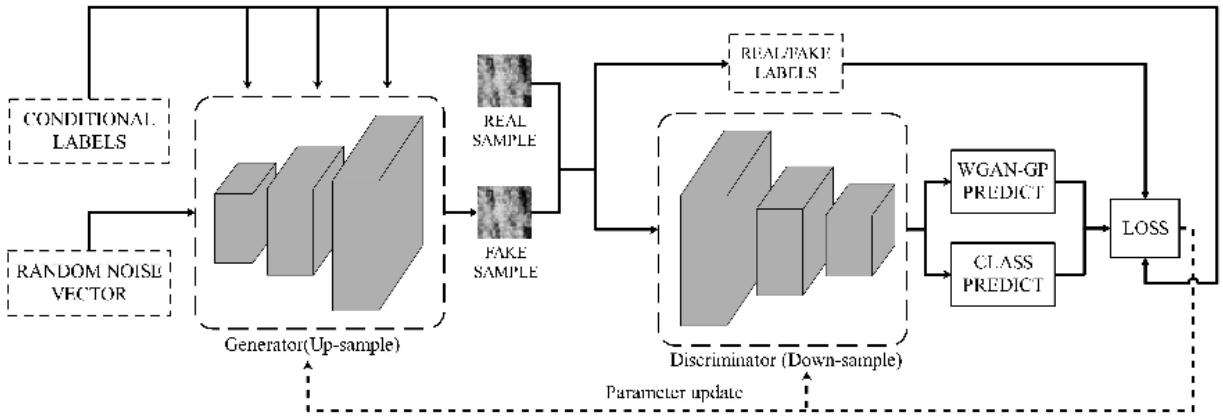


FIGURE 5. Structural flow of generative algorithm.

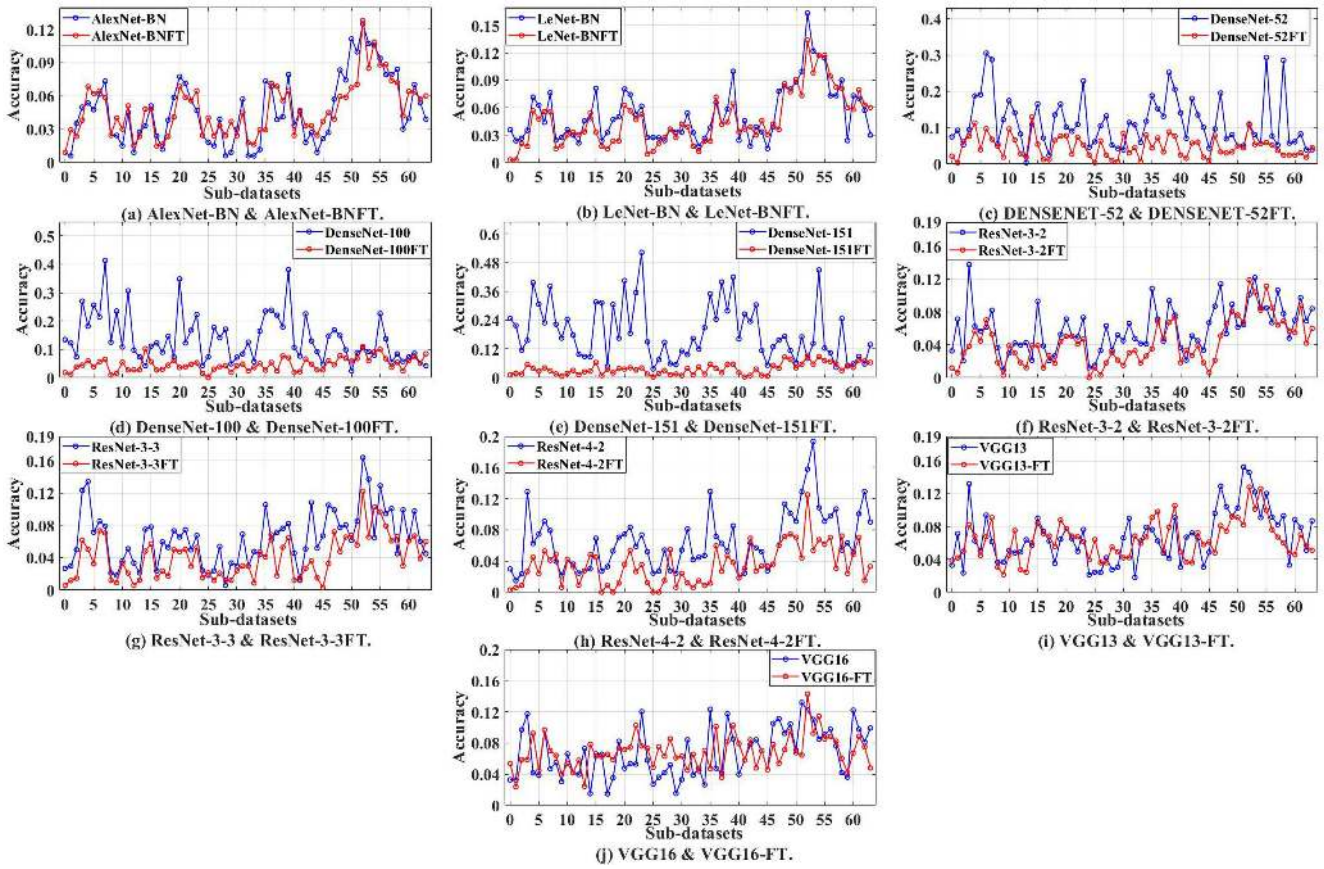


FIGURE 6. Comparison of the classification error rates of different CNNs on all datasets before and after fine tuning.

all subdatasets. Notably, AlexNet-BN and LeNet-BN could achieve high accuracies and performed much better than other deeper models, even though they were reasonably simple and small. Meanwhile, other models, such as ResNet and DenseNet, showed a clear trend of decreasing performance as complexity and depth increased. However, after fine tuning, the deeper networks showed better results. Through comparison, it is obvious that fine tuning can allow for significant improvements to some of the models. Table 3 presents the classification results of SVM as the control group, which also achieved classification, but the effect was poor. Although PCA helped its error rate decrease from 13.101% to 10.510%, there was still a performance gap between the SVM and the CNNs. Ultimately, this showed that CNNs performed better on this mission than traditional algorithms.

As mentioned earlier, there are significant differences between the results of the CNNs before and after LSPM-based fine tuning, especially the correlation between the depth and performance of the model. Before fine tuning, small and shallow models such as LeNet-BN and AlexNet-BN performed better than a series of deep and complex models such as DenseNet and, generally, showed an inverse proportion between depth and performance. This overall trend is also in line with the result of our previous experiments [24]. Of course, we noticed that AlexNet-BN and LeNet-BN are contrary to the overall trend. This may have been caused by the shallow structure of LeNet-BN; in the absence of a large enough training set, a small model with shallow feature extraction ability can only capture basic features, lacking the ability to acquire high-dimensional abstract features. The results of ResNet and DenseNet before fine tuning reflected a poor performance of complex and deeper models on small datasets, which may have been due to the small datasets causing the models to learn all the features of the training sets, resulting in weak inference capabilities.

However, the deeper models after LSPM-based fine tuning showed better results. As shown in Table 2 and Fig. 6, the ResNet and DenseNet improved greatly, with DenseNet, improving from an error rate $> 11.4\%$ to $\sim 4\%$, and ResNet-4-2 reaching a minimum error of 3.459%. Additionally, the relationship between the depth and complexity of the model and its performance showed a generally proportional trend. Although some models had no substantial improvement before and after fine tuning, the performance of ResNet-4-2FT (fine-tuned ResNet-4-2; for simplicity, the abbreviation FT is added to all fine-tuned models to indicate fine tuning) and DenseNet-151FT is enough to illustrate the significance of LSPM-based fine tuning.

Against our expectation, AlexNet-BN performed well before fine tuning. Therefore, we compared AlexNet-BN with ResNet-4-2FT and DenseNet-151FT, and plotted its test accuracy on all subdatasets in Fig. 7. According to Fig. 7, AlexNet-BN seemed to perform well, but the performance of ResNet-4-2FT and DenseNet-151FT on more than half of the subdatasets was better than that of AlexNet-BN by one or two percentage points. Ultimately, the fine tuning based on LSPM is alleviation, to a certain extent, of the problem of poor performance of complex and huge models on small datasets.

To analyze the relationship between performance and depth more intuitively, the averaged testing accuracy for various CNNs with different depths before and after LSPM based fine tuning are presented in Fig. 8. ResNet (Fig. 8a) and DenseNet (Fig. 8b) both showed the same trend as previously discussed, i.e., the depth and classification accuracy were inversely proportional after fine tuning, whereas the opposite was true before fine tuning. However, VGG (Fig. 8c) did not show the same trend; this may have been caused by the small dataset or the absence of residual connections. The improvements brought by LSPM-based fine tuning went against the supposition that deep CNNs perform poorly on small datasets, thereby allowing them to exert their high-dimensional feature extraction capabilities.

Finally, fine tuning based on LSPM accelerated the training speed and convergence of the models, resulting in a lower amount of time required to train the model, which is significant in some applications. The better starting point for parameter initialization allowed by fine tuning can make CNNs converge more stably and rapidly. Figure 9 shows the comparison of ResNet-4-2 and AlexNet-BN before and after fine tuning on the SCH-52 dataset. Notably, both AlexNet-BN and ResNet-4-2 exhibited faster convergence speeds after fine tuning, meaning that fine tuning can help CNNs converge quickly. Moreover, the curves for training loss and testing accuracy of the model trained from scratch were chaotic, which may have been caused by random initialization that let the model fall into some local optimums, resulting in numerous oscillations before it could smooth out. In summary, fine tuning based on LSPM allows the models to reach a usable level with a lower training time, being of great significance for certain application scenarios in which a long-term training process needs to be avoided.

According to the results, although small CNNs performed well, especially when they were trained from scratch, they still had a comparatively performance and convergence speed. In general, the fine tuning of models based on LSPM is practical, effective, and can improve the performance of CNNs, especially for deeper CNNs, such as DenseNet and ResNet. The LSPM-based fine tuning allowed for the application of deep CNNs' high-dimensional extraction abilities and, notably, resulted in ResNet-4-2FT achieving a very low error rate of 3.459%, highlighting the general applicability of ResNet to this mission. These results may also allow for theoretically better deep CNNs to be applied to similar missions. Although different missions should select models based on their characteristics, our research shows that when performing sediment classification under a small SSS dataset, it is useful to use a deeper model and pretrain on other irrelevant datasets. Based on the discussion above, we believe that if the needed computing power is available, it is feasible to use a large model and pretrain it by LSPM based fine tuning.

C. DATA ENHANCEMENT BASED ON DL

In this study, we used GANs for DL-based data enhancement to improve the richness of the training dataset. This approach differs from data enhancement based on geometric transforms, and generates new data by inputting random noise into the generator after training the GANs. We applied a CBN-based WGAN-GP algorithm to generate new SSS data. In contrast to training a GAN on a limited single-sediment SSS image dataset, after adding the CBN, the algorithm can train on the entire training set and generate corresponding SSS images under the control of conditional labels. In the experiment, a total of 100,000 iterations (not epochs) were set; the initial learning rate was 0.0002, and the batch size was 64. All convolutional parts in the algorithm were residual

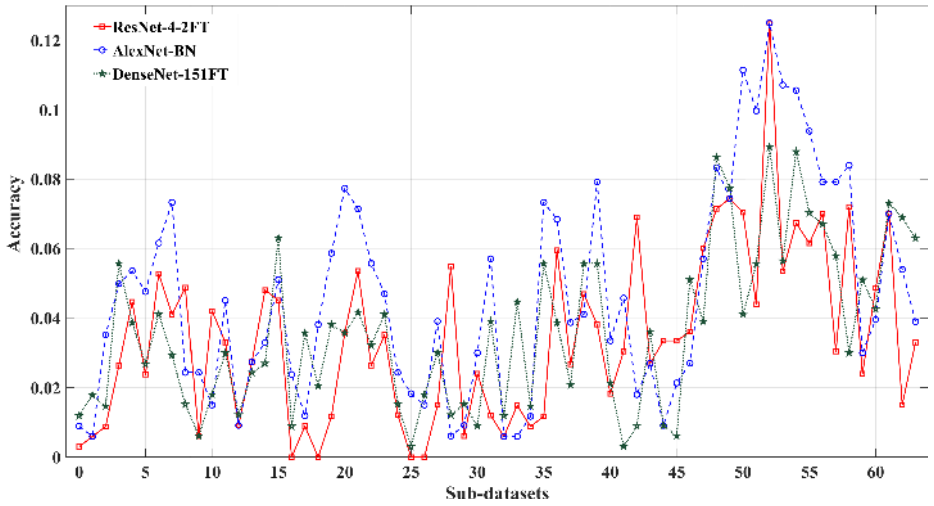


FIGURE 7. Comparison of the classification error rates of AlexNet-BN before fine tuning, and ResNet-4-2 and DenseNet-151 after fine tuning based on LSPM.

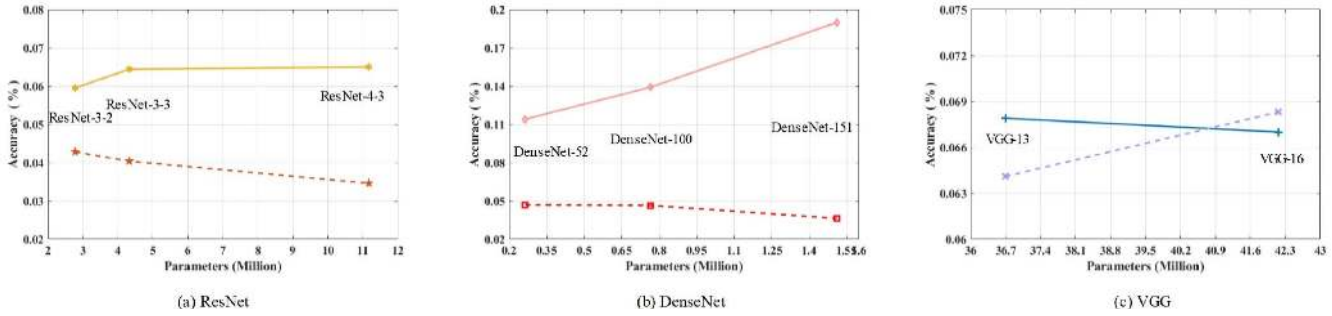


FIGURE 8. Comparison of the classification error rates of (a) ResNet, (b) DenseNet, and (c) VGG before and after fine tuning.

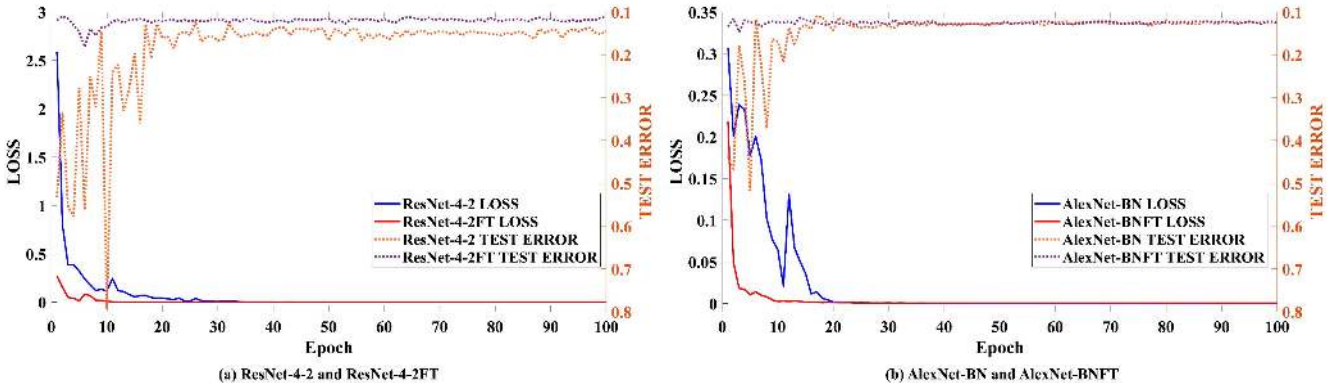
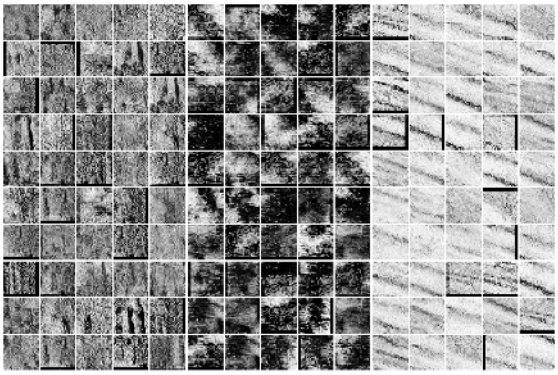


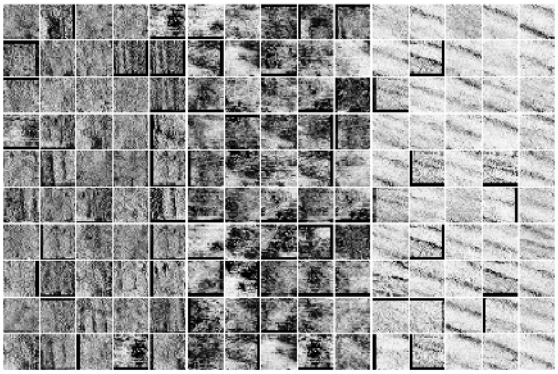
FIGURE 9. Comparison of the training losses (left y-axis) and test errors (right y-axis) of the (a) ResNet-4-2 and (b) AlexNet-BN models before and after fine tuning with the SCH-52 dataset. The solid line represents the loss function and the dashed line the test accuracy.

networks. The dimension of the feature maps in the discriminator and generator were both 128, and the discriminator was trained 5 times per generator update (to ensure the stability of the algorithm). However, considering the large computational resources and time consumed by GANs, it is impractical to apply GAN-based data enhancement to all subdatasets. Therefore, we selected the following subdatasets for our tests: SCH-36, SCH-52, SCH-56, and SCH-60. Some generated images are shown in Fig. 10.

In our experiments, we generated 300 images of each sediment and put them into the original dataset to expand the training dataset, which was approximately doubled in size. For each subdataset, we randomly generated six sets of enhanced datasets to measure data enhancement performance



(a) Generated images based on SCH_52.



(b) Generated images based on SCH_60.

FIGURE 10. Images generated by GANs based on the SCH-52 and SCH-60 datasets.

TABLE 4. Experimental results of GAN-based data enhancement.

Sub-dataset	CNNs	Test accuracy w/o GAN enforcement (error rate)	Test accuracy w/ GAN enforcement (error rate)
SCH-36	AlexNet-BN	7.202 %	**5.843 %**
	AlexNet-BNFT	6.190 %	**4.970 %**
	ResNet-4-2FT	5.179 %	**3.591 %**
SCH-52	AlexNet-BN	13.690 %	**13.028 %**
	AlexNet-BNFT	13.333 %	**6.0112 %**
	ResNet-4-2FT	12.560 %	**5.407 %**
SCH-56	AlexNet-BN	7.683 %	**5.955 %**
	AlexNet-BNFT	9.207 %	**6.057 %**
	ResNet-4-2FT	7.012 %	**3.638 %**
SCH-60	AlexNet-BN	**5.427 %**	5.884 %
	AlexNet-BNFT	5.976 %	**4.776 %**
	ResNet-42FT	**5.000 %**	5.813 %

as well as to weaken the impact of random processes (i.e., stochasticity). To further weaken the impact of random processes in CNNs, the model performed five repeated experiments on each enhanced dataset, and the mean of all results was taken as the final result. We selected AlexNet-BN, AlexNet-BNFT, and ResNet-4-2FT for evaluating the effects of data enhancement, using the same training configurations as described in the previous section.

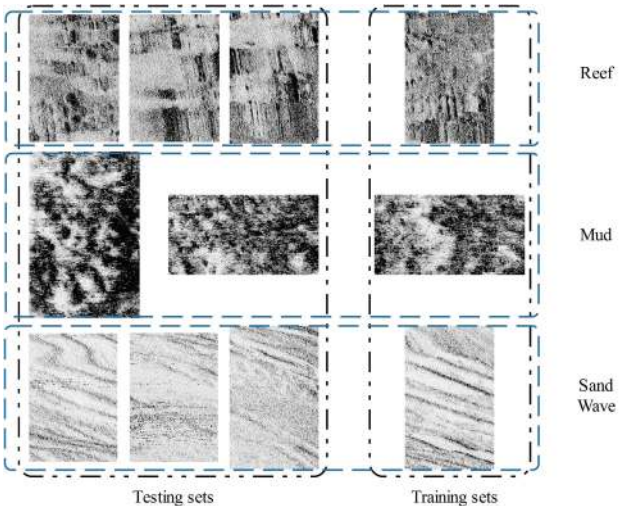


FIGURE 11. Training set and testing set in the SCH-60 dataset.

Our experimental results are shown in Table 4. For the SCH-52 and SCH-56 datasets, the application of CNNs to the enhanced dataset brought significant improvements, whereas for the SCH-36, only minor improvements were achieved. Among them, the degree of improvement for SCH-52 and SCH-56, facilitated using GANs, was greater. This was especially the case for SCH-52, with which AlexNet-BNFT and ResNet-4-2FT achieved substantial improvements in accuracy of ∼7%; this confirmed the important role of generating data for improving training datasets. It is worth noting that there was a slight decline in accuracy (∼0.4–0.8%) in the SCH-60 dataset.

However, we are still unsure about the unexpected performance under the SCH-60 dataset, which shows that the data enhancement based on GANs did not allow for improvement, or even a decline. For this reason, we analyzed what caused the data enhancement to be unsuccessful or unstable. Figure 1 shows the data division in the SCH-60, where the red box is the divided training set, and the specific comparison of the training set and testing set in the SCH-60 sub-dataset is shown in Fig. 11. A large deviation between the training and test set for reefs were seen, with a specific lack of flat and strong reflection areas in the training set (but a large number in the testing set). Additionally, strong reflection areas existed in the training set for mud, which had high similarity with some areas in the testing set for reefs and sand waves. Thus, in the SCH-60 subdataset, data quality caused a similar situation after cutting, which led to misclassification. In this case, generating a large amount of similar data through GANs will not significantly improve the situation and may even cause the classifier to be confused, resulting in a slight decrease in accuracy. In general, data enhancement based on GANs does not bring any improvement when data quality is insufficient and discrimination is low, which may even result in a slight decrease in the performance of the CNN classifier. Finally, GANs are used to imitate the original information

distribution and generate similar information and not to manufacture information.

In view of the fact that the data enhancement of GANs brought improvements for the SCH-36, SCH-52, and SCH-56 and a slight decline for the SCH-60, we believe that this kind of data enhancement is effective to a certain extent, but it is not universally applicable. In case of good data quality, it can bring a certain extent of improvement, but this improvement depends on the quality of the original data. In addition to the improvements brought by GANs, we identified two problems in the experiments.

(1) The GANs input includes random noise, which results in differences in the generated images, so that the accuracy of the CNNs classifier differs slightly between repeated experiments.
(2) The training of GANs is resource intensive, occupying approximately 9 GB of GPU memory, and each iteration takes 0.212 s on average when the number of iterations is set to 100,000. It is difficult to accept such costs in general application scenarios.

Through our analysis, we determined that the GANs in the experiment could improve model accuracy for some inferior datasets, but it was not suitable for all application scenarios (or data). Additionally, GANs demand a substantial amount of computing resources and time. In general, they can help to solve the poor performance of classification for small datasets; however, they are resource-intensive and time-consuming, and improvement depends on the quality of the original data. Therefore, we consider data enhancement using GANs to be a possible effective solution only in certain cases.

V. DISCUSSION AND CONCLUSION

In this study, we optimized the sediment classification of a small SSS dataset based on DL using deeper CNNs, fine tuning based on LSPM and data enhancement employing new data generated by GANs to expand the original dataset. The experimental results showed that fine tuning allowed for further breakthroughs in the classification accuracy of CNN classifiers, whereas the use of GANs for data enhancement was unstable but could generally improve classification accuracy. The small dataset is a bottleneck to sediment classification based on seabed acoustic image data and, thus, our findings have practical implications for related studies.

A. FINE TUNING BASED ON LSPM

Notably, after fine tuning, the CNNs had more rapid and stable performances, achieving meaningful improvements in classification accuracy, especially for ResNet-4-2. We observed that, even if the pretraining and target data were essentially irrelevant, such LSPM-based fine tuning was of great value in optimizing the performance of CNNs in small SSS datasets. Compared with existing studies on the application of fine tuning to seabed acoustic image data, the key contribution of our research is the broadening of its application, especially when there is a lack of mature and relevant datasets.

More importantly, the deep CNNs (e.g., ResNet and DenseNet) that performed poorly in previous experiments displayed improved accuracies after fine tuning and achieved relatively promising results. Contrary to the general recommendation of not applying deep complex models to small datasets, we applied deep CNNs to our target dataset and achieved favorable results through fine tuning, thereby broadening the application of CNNs for similar tasks. Considering the importance of parameter weight initialization, we believe that training a CNN with random initialization on limited data will cause problems in model training, which can be greatly improved by fine tuning. In general, the fine tuning of LSPM can optimize the performance of CNNs applied to sediment classification based on small seabed acoustic images dataset, and aid in the application of theoretically better deep CNNs to similar tasks. We suggest, if the computing resources are sufficient, using a larger model and reasonably pretraining the CNNs. For example, the use of fine-tuned ResNet is an effective path.

B. DATA ENHANCEMENT BASED ON GANs

The deployment of our enhanced dataset revealed that the new data generated by GANs can enhance the feature richness of a training dataset. GANs generate simulation images by antagonizing the distance between the distribution of the model and the real data. However, because sediment images are composed of textures and gray scales, they often lack a sufficient number of expressive features and enough labeled data; therefore, data enhancement based on GANs seems to be an effective solution.

In contrast to the prevailing research on the application of GANs to seabed acoustic images [36], [37], [41], we applied GANs to generate SSS image data with sediment contents and then added them into the original dataset to enrich its features, finally verifying the resulting improvements. However, whether our approach works depends largely on the quality of the original data. That is, GANs is impractical for be in common use in optimizing the processing of small seabed acoustic image datasets. However, GANs still have considerable potential in the field of data generation and, in theory, their application can optimize seabed acoustic image data processing tasks. Therefore, in future research, we will combine the latest results of current related studies in an attempt to optimize the data enhancement effects of GANs and apply them to a wider range of applications for seabed acoustic image data.

To a certain extent, our methods allowed us to overcome the lack of labeled data for sediment classification based on seabed acoustic image data and provided a feasible way to apply CNNs to such small datasets. However, our methods also had several limitations, including the unstable data generation quality and the existence of random processes in GANs. We will conduct in-depth analyses of these problems in the future. Finally, DL achieved excellent performance in

fields such as style transfer, target detection, and few-shot learning, and still has great development potential. Therefore, we believe that DL has great application potential for seabed acoustic image processing tasks, which is the main direction of our future research.

REFERENCES

[1] J. T. Anderson, D. Van Holliday, R. Kloser, D. G. Reid, and Y. Simard, "Acoustic seabed classification: Current practice and future directions," *ICES J. Mar. Sci.*, vol. 65, no. 6, pp. 1004–1011, Apr. 2008.

[2] G. Shumway, "Sound speed and absorption studies of marine sediments by a resonance method," *Geophysics*, vol. 25, no. 2, pp. 451–467, Apr. 1960.

[3] D. R. Jackson, D. P. Winebrenner, and A. Ishimaru, "Application of the composite roughness model to high-frequency bottom backscattering," *J. Acoust. Soc. Amer.*, vol. 86, no. 5, pp. 2029–2032, 1986.

[4] D. R. Parrott, D. J. Dodds, L. H. King, and P. G. Simpkin, "Measurement and evaluation of the acoustic reflectivity of the seafloor," *Can J. Earth Sci.*, vol. 17, pp. 722–737, Dec. 1980.

[5] A. Tolstoy, "Applications of matched-field processing to inverse problems in underwater acoustics," *Inverse Problems*, vol. 16, no. 6, pp. 1655–1666, Dec. 2000.

[6] D. P. Knobles, R. A. Koch, L. A. Thompson, K. C. Focke, and P. E. Eisman, "Broadband sound propagation in shallow water and geoacoustic inversion," *J. Acoust. Soc. Amer.*, vol. 113, no. 1, pp. 205–222, Jan. 2003.

[7] M. D. Collins, W. A. Kuperman, and H. Schmidt, "Nonlinear inversion for ocean-bottom properties," *J. Acoust. Soc. Amer.*, vol. 92, no. 5, pp. 2770–2783, Nov. 1992.

[8] Z.-H. Michalopoulou and U. Ghosh-Dastidar, "Tabu for matched-field source localization and geoacoustic inversion," *J. Acoust. Soc. Amer.*, vol. 115, no. 1, pp. 135–145, Jan. 2004.

[9] J. A. Shorey, L. W. Nolte, and J. L. Krolik, "Computationally efficient Monte Carlo estimation algorithms for matched field processing in uncertain ocean environments," *J. Comput. Acoust.*, vol. 02, no. 03, pp. 285–314, Sep. 1994.

[10] Z. Michalopoulou and D. Alexandrou, "Bayesian modeling of acoustic signals for seafloor identification," *J. Acoust. Soc. Amer.*, vol. 99, no. 1, pp. 223–233, Jan. 1996.

[11] Z. Michalopoulou, D. Alexandrou, and C. de Moustier, "Application of a maximum likelihood processor to acoustic backscatter for the estimation of seafloor roughness parameters," *J. Acoust. Soc. Amer.*, vol. 95, no. 5, pp. 2467–2477, May 1994.

[12] B. Zou, J. Zhai, Z. Qi, and Z. Li, "A comparison of three sediment acoustic models using Bayesian inversion and model selection techniques," *Remote Sens.*, vol. 11, pp. 1–30, Mar. 2019.

[13] A. Caiti and S. M. Jesus, "Acoustic estimation of seafloor parameters: A radial basis functions approach," *J. Acoust. Soc. Amer.*, vol. 100, no. 3, pp. 1473–1481, Sep. 1996.

[14] C. Frederick, S. Villar, and Z.-H. Michalopoulou, "Seabed classification using physics-based modeling and machine learning," 2020, *arXiv:2003.11156*. [Online]. Available: http://arxiv.org/abs/2003.11156

[15] J. S. Collier and C. J. Brown, "Correlation of sidescan backscatter with grain size distribution of surficial seabed sediments," *Mar. Geol.*, vol. 214, no. 4, pp. 431–449, Feb. 2005.

[16] D. R. Carmichael, L. M. Linnett, S. J. Clarke, and B. R. Calder, "Seabed classification through multifractal analysis of sidescan sonar imagery," *IEE Proc.—Radar, Sonar Navigat.*, vol. 143, no. 3, pp. 140–148, Jun. 1996.

[17] A. Bartholomä, "Acoustic bottom detection and seabed classification in the german bight, southern north sea," *Geo-Marine Lett.*, vol. 26, no. 3, pp. 177–184, Sep. 2006.

[18] L. Atallah and P. J. P. Smith, "Automatic seabed classification by the analysis of sidescan sonar and bathymetric imagery," *IEE Proc.—Radar, Sonar Navigat.*, vol. 151, no. 5, pp. 327–336, Oct. 2004.

[19] M. Wang, Z. Wu, F. Yang, Y. Ma, X. Wang, and D. Zhao, "Multifeature extraction and seafloor classification combining LiDAR and MBES data around Yuanzhi Island in the south China sea," *Sensors*, vol. 18, no. 11, p. 3828, Nov. 2018.

[20] H. Liu, K. Xu, B. Li, Y. Han, and G. Li, "Sediment identification using machine learning classifiers in a mixed-texture dredge pit of Louisiana shelf for coastal restoration," *Water*, vol. 11, no. 6, p. 1257, Jun. 2019.

[21] K. I. Ahmed, H. Caughey, P. Hung, P. Harris, U, Demsar, S, McLoone, S. Fotheringham, X. Monteys, and R. O'Toole, "Classification and mapping of seabed type from deep water multibeam echosounder (MBES) data," in *Proc. 6th Int. Conf. Geography Inf. Sci.*, Zürich, Switzerland, 2010, pp. 1–5.

[22] D. Vray, P. Delachartre, N. Andrieux, and G. Gimenez, "Bottom classification using information in the spectral domain and time-frequency domain," in *Proc. OCEANS*, vol. 2, Brest, France, 1994, pp. 659–664.

[23] J. Tgowski, J. Nowak, M. Moskalik, and K. Szefler, "Seabed classification from multibeam echosounder backscatter data using wavelet transformation and neural network approach," in *Proc. 4th Int. Conf. Exhib. Underwater Acoustic Meas., Technol. Results*, Kosisland, Greece, 2011, pp. 1257–1264.

[24] W. K. Stewart, M. Jiang, and M. Marra, "A neural network approach to classification of sidescan sonar imagery from a midocean ridge area," *IEEE J. Ocean. Eng.*, vol. 19, no. 2, pp. 214–224, Apr. 1994.

[25] X. Luo, X. Qin, Z. Wu, F. Yang, M. Wang, and J. Shang, "Sediment classification of small-size seabed acoustic images using convolutional neural networks," *IEEE Access*, vol. 7, pp. 98331–98339, 2019.

[26] T. Berthold, A. Leichter, B. Rosenhahn, V. Berkhahn, and J. Valerius, "Seabed sediment classification of side-scan sonar data using convolutional neural networks," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Honolulu, HI, USA, Nov. 2017, pp. 1–8.

[27] I. Kvasic, N. Miskovic, and Z. Vukic, "Convolutional neural network architectures for sonar-based diver detection and tracking," in *Proc. OCEANS*, Marseille, France, Jun. 2019, pp. 1–6.

[28] X. Wang, J. Jiao, J. Yin, W. Zhao, X. Han, and B. Sun, "Underwater sonar image classification using adaptive weights convolutional neural network," *Appl. Acoust.*, vol. 146, pp. 145–154, Mar. 2019.

[29] H. Henley, A. Berard, E. Lapisky, and M. Zimmerman, "Deep learning in shallow water: CNN-based 3D-FLS target recognition," in *Proc. OCEANS MTS/IEEE Charleston*, Charleston, SC, USA, Oct. 2018, pp. 1–7.

[30] P. Zhu, J. Isaacs, B. Fu, and S. Ferrari, "Deep learning feature extraction for target recognition and classification in underwater sonar images," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Melbourne, VIC, Australia, Dec. 2017, pp. 2724–2731.

[31] M. Valdenegro-Toro, "End-to-end object detection and recognition in forward-looking sonar images with convolutional neural networks," in *Proc. IEEE/OES Auto. Underwater Vehicles (AUV)*, Tokyo, Jpn, Nov. 2016, pp. 144–150.

[32] Wu, Wang, Rigall, Li, Zhu, He, and Yan, "ECNet: Efficient convolutional networks for side scan sonar image segmentation," *Sensors*, vol. 19, no. 9, p. 2009, Apr. 2019.

[33] P. Liu and Y. Song, "Segmentation of sonar imagery using convolutional neural networks and Markov random field," *Multidimens. Syst. Signal Process.*, vol. 31, no. 1, pp. 4–21, Jan. 2020.

[34] M. Rahnemoonfar and D. Dobbs, "Semantic segmentation of underwater sonar imagery with deep learning," in *Proc. IEEE Sci., Geosci. Remote Sens. Symp.*, Yokohama, Japan, Dec. 2019, pp. 9455–9458.

[35] Z. Zhu and Y. Hu, "Sonar image recognition based on fine-tuned convolutional neural network," *MATEC Web Conf.*, vol. 283, Dec. 2019, Art. no. 04012.

[36] J. L. Chen and J. E. Summers, "Deep neural networks for learning classification features and generative models from synthetic aperture sonar big data," *Proc. Meet. Acoust.*, vol. 29, no. 1, 2016, Art. no. 03200.

[37] L. Xu, X. Wang, and X. Wang, "Shipwrecks detection based on deep generation network and transfer learning with small amount of sonar images," in *Proc. IEEE 8th Data Driven Control Learn. Syst. Conf. (DDCLS)*, Dali, China, May 2019, pp. 638–643.

[38] G. Huo, Z. Wu, and J. Li, "Underwater object classification in sidescan sonar images using deep transfer learning and semisynthetic training data," *IEEE Access*, vol. 8, pp. 47407–47418, 2020.

[39] A. Krizhevsky. (2009). *Learning Multiple Layers of Features From Tiny Images*. [Online]. Available: http://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf

[40] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, and, "Generative adversarial networks," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 27, 2014, pp. 2672–2680.

[41] A. Reed, I. Gerg, J. McKay, D. Brown, D. Williams, and S. Jayasuriya, "Coupling rendering and generative adversarial networks for artificial SAS image generation," 2019, *arXiv:1909.06436*. [Online]. Available: http://arxiv.org/abs/1909.06436

[42] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, vol. 15, G. Gordon, D. Dunson, M. Dudík, Eds., 2010, pp. 315–323.
[43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[44] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, eds., 2012, pp. 1097–1105.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
[48] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2261–2269.
[49] T. Miyato and M. Koyama, "CGANs with projection discriminator," 2018, *arXiv:1802.05637*. [Online]. Available: http://arxiv.org/abs/1802.05637
[50] S. Ioffe and C. Szegedy. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. [Online]. Available: https://arxiv.org/abs/1502.03167
[51] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875
[52] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.* vol. 30, 2017, pp. 5767–5777.
[53] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, pp. 2642–2651, 2017.
[54] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[55] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vols. SMC–3, no. 6, pp. 610–621, Nov. 1973.
[56] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

XIAOMING QIN received the B.E. degree from the College of Geomatics, Shandong University of Science and Technology, Qingdao, China, in 2018. He is currently with the Key Laboratory of Submarine Geosciences, Second Institute of Oceanography, Ministry of Natural Resources. His current research interests include underwater target detection and sediment classification based on deep learning.

XIAOWEN LUO received the Ph.D. degree in geodesy and geomatics from the Institute of Geodesy and Geophysics, Chinese Academy of Sciences, Hubei, China, in 2007. He is currently an Associate Professor with the Second Institute of Oceanography, Ministry of Natural Resources. His primary research interests include seabed survey and information systems.

ZIYIN WU received the Ph.D. degree in marine geology from Zhejiang University, Zhejiang, China, in 2008. He is currently a Professor with the Second Institute of Oceanography, Ministry of Natural Resources, and with the School of Oceanography, Shanghai Jiao Tong University. His primary research interests include seabed survey and information systems.

JIHONG SHANG received the Ph.D. degree in marine geology from the Institute of Oceanology, Chinese Academy of Sciences, China, in 2008. He is currently an Associate Research with the Second Institute of Oceanography, Ministry of Natural Resources, China. His research interests include submarine topographic survey and morphotectonics study.