

Side Scan Sonar Shipwreck Image Recognition Algorithm based on Transfer Learning and Deep Learning

Boyu Zhang¹, Xiao Wang^{1,2,*}, Shudong Li¹, Jinghua Yang¹

¹School of Marine Technology and Geomatics, Jiangsu Ocean University, Lian yungang 222000 China;

²Jiangsu Institute of Marine Resources, Lian yungang 222000 China;

Abstract. Current underwater shipwreck side scan sonar samples are few and difficult to label. With small sample sizes, their image recognition accuracy with a convolutional neural network model is low. In this study, we proposed an image recognition method for shipwreck side scan sonar that combines transfer learning with deep learning. In the non-transfer learning, shipwreck sonar sample data were used to train the network, and the results were saved as the control group. The weakly correlated data were applied to train the network, then the network parameters were transferred to the new network, and then the shipwreck sonar data was used for training. These steps were repeated using strongly correlated data. Experiments were carried out on Lenet-5, AlexNet, GoogLeNet, ResNet and VGG networks. Without transfer learning, the highest accuracy was obtained on the ResNet network (86.27%). Using weakly correlated data for transfer training, the highest accuracy was on the VGG network (92.16%). Using strongly correlated data for transfer training, the highest accuracy was also on the VGG network (98.04%). In all network architectures, transfer learning improved the correct recognition rate of convolutional neural network models. Experiments show that transfer learning combined with deep learning improves the accuracy and generalization of the convolutional neural network in the case of small sample sizes.

1. Introduction

Marine surveying and mapping is the basis of all underwater activities and the premise of seabed surveys, seabed mineral explorations, and marine economic development. Side Scan Sonar (SSS), a high resolution, multi-purpose and low-cost marine survey equipment, is widely used for surveying oceans, ports, rivers, lakes, and other waters. The main applications of SSS are hydrographic surveys, underwater engineering site selection, detection of underwater targets such as aircraft wreckage, shipwrecks, torpedoes, mines, containers, chemical barrels, anchor chains, sunken rocks, buoys, anchorage obstructions, and seabed bottom material classification, marine survey and marine scientific research [1]. Currently, commercial SSS data processing software for underwater target detection and recognition relies on manual interpretation, which is inefficient and does not guarantee accuracy. Traditional target recognition and detection methods of SSS images mainly include recognition methods based on Markov Random Fields (MRF) [2], gray histogram and geometric features [3], extracting features in the frequency domain [4], fuzzy clustering [5], detection methods based on template

matching [6], threshold [7], sparse representation [8], and classification models based on Support Vector Machine (SVM) [9], among others. The general workflow of such methods includes: feature extraction, model construction, and model application. Subject to the complex marine environment and the presence of a large number of noises in SSS images, the model constructed in one sea area may not be applicable to another. Moreover, due to the difficulty in obtaining typical features and distinguishing them, the recognition efficiency is low.

Convolutional Neural Network (CNN)[10] is a widely used artificial intelligence algorithm for image classification, image recognition, image segmentation and other fields [11-14] because of its advantages such as not requiring manually selected features and having excellent feature extraction skill. There are many architectures of convolutional neural networks [15-20]. However, an excellent convolutional neural network can only be obtained with a large amount of appropriate training data. For common study objects, such as cars, plants and animals, it is easy to collect a large number of images [21]. However, in special cases, such as in the case of underwater shipwrecks, obtaining a large number of samples is very difficult.

To overcome the problem of insufficient samples in deep learning, Matias et al. [22] proposed a method of training convolutional neural networks with small samples, which can achieve an accuracy of 80% when the

Correspondence author: *wxsdau2005@163.com;

Tel: +86-137-7548-9830

samples are less than 20. Liang et al. [23] proposed a small sample underwater target recognition method based on a convolutional neural network, and validated it with fish image data. The experiment achieved an accuracy recognition rate of 85.08%. Mehrotra et al. [24] combined transfer learning with deep learning methods to train the network with a small sample data of brain tumors, successfully distinguishing malignant from benign tumors. Zhen et al. [25] studied how to identify power towers in remote sensing images based on photos of natural power towers. In the experiment, the photos of natural power towers were used to train the network architecture, and then the training parameters were extracted and imported into the new network, and then the power towers in remote sensing images were used for network training. The research results show that transfer learning is helpful for network training. Zhu et al. [26] built a submarine model at a scale of 100:1 and performed simulation experiments on underwater submarine recognition. During the research, the parameters of the sonar were adjusted to the corresponding intensity, and the simulation data were collected in an underwater acoustic pool. Multiple pre-training networks can be used to perform transfer learning training on the data. Transfer learning can achieve higher recognition rates and faster convergence rates in the case of little data. These manuscripts provide a theoretical solution on how to apply CNN model well for SSS shipwreck that has small sample image.

The marine environment is complex and noisy, and sample image characteristics are not obvious. With very few marked samples, deep learning applications are limited. Transfer learning provides a good solution, but what kinds of data can be used for transferring the SSS shipwreck image and which CNN models are suitable for underwater shipwreck target detection and recognition have not yet been identified. We propose a method of combining transfer learning with a convolutional neural network. Through design experiments, the SSS data were transferred to a network trained with public data from different sources and different characteristics, and to a network trained with ship data from different sources but the same characteristics. The aim of this study was to obtain a convolutional neural network model suitable for underwater shipwreck recognition, and solve the problems related to the difficulty in obtaining large SSS image samples under complex marine environment, and in typical feature extraction.

The remainder of this paper is organized as follows: In Section 2, the source of the data set used in the experiment is introduced, the classification of the data, and the relevance of the data are described. In Section 3, briefly describes the experimental principle, experimental method, experimental procedure and accuracy evaluation method. Section 4 describes the experimental results. Discussion and the conclusion of the paper is given in Section 5.

2. Data

2.1 SeaShips-3

Shao et al. [21] created SeaShips, a ship dataset, by placing fixed cameras in river courses and estuaries to record a large number of samples through video screenshots. This dataset, with a total of 31455 pictures and 6 classes, has been used for research on target recognition algorithms for ships on water. Since the number of samples in each class is inconsistent in this dataset, three classes were randomly selected from SeaShips (Figure 1), each with 800 pictures, and the selected dataset was named SeaShips-3 for this experiment.

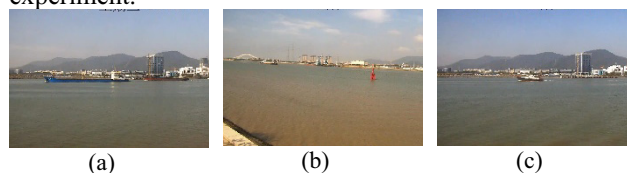


Figure 1. Examples of (a) Bulk cargo carrier samples; (b) General cargo ship samples; (c) Fishing boat samples.

2.2 CIFAR-3

Cifar-10 is a commonly used public dataset, which is composed of 10 classes and 60,000 color images with a size of 32 x 32 pixels. Three classes were randomly selected from CIFAR-10, and 1000 images were randomly selected for each class as the experimental data and named Cifar-3. Example images are shown in Figure 2.

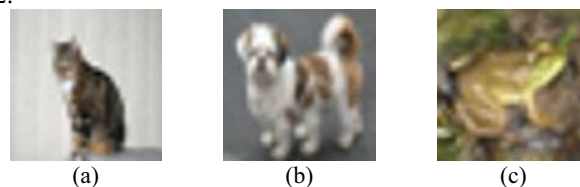


Figure 2. Examples of (a) Cat samples; (b) Dog samples; (c) Frog samples.

2.3 SSS-3

The SSS-3 is a small dataset of SSS images (see examples in Figure 3). It consists of three classes: Underwater shipwreck, pure seabed background, and marine fish. Each class contains 55 sample pictures.

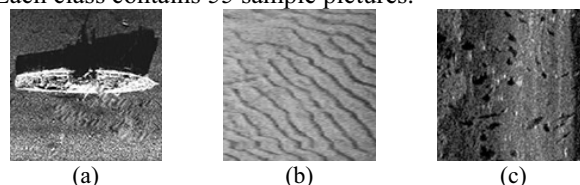


Figure 3. Examples of (a) Underwater shipwreck samples; (b) Pure seabed background samples; (c) Marine fish samples.

The above three datasets were divided into training data and testing data according to a ratio of 7:3. The training data was used for training the convolutional neural network, and the testing data was used for testing the trained network.

2.4 Data Correlation

Data correlation refers to the relationship between samples. In general, as long as the data are of the same source and type they are considered to be correlated. For example, when cats, leopards, and lions in the cat family appear in the data at the same time, they are considered to be strongly correlated data. Bicycles, electric bicycles, motorcycles, etc. are also considered to be strongly correlated data because they all have common characteristics. In contrast, cars, dogs, and apples are considered to be weakly correlated or irrelevant data.

In addition to subjective correlation, the degree of similarity between two images can be calculated using a hash algorithm. A number of sample images are randomly selected from the data, hash values are calculated between different data, and the final result is processed into a heat map that visualizes the correlation between the data. Three datasets were used in the experiment, namely SeaShip-3, CIFAR-3 and SSS-3. We used the following methods to identify strong and weak correlations between the datasets:

(1) Randomly select 100 pictures from SeaShip-3 to form subset A. $A \subset \text{SeaShip-3}, a_i \in A$. Similarly, 100 pictures were randomly selected from SSS-3 to form subset B. $B \subset \text{SSS-3}, b_j \in B$.

(2) Use the Hash algorithm to calculate the similarity between a_i and b_j , and record the data in the matrix $m[i, j]$:

$$m[i, j] = \sum_{i,j}^{100} \text{Hash}(a_i, b_j) \quad (1)$$

(3) Calculate the average value of the matrix elements and convert the matrix $m[i, j]$ into a grayscale image.

(4) Repeat steps (1) to (3) for datasets CIFAR-3 and SSS-3.

Through the above steps, the correlation between SeaShip-3 and CIFAR-3 with SSS-3 can be calculated (Figure 4). The more similar the two pictures are, the greater the hash value; a greater hash value is indicated with a lighter shade of grayscale in the image. The average hash value of SeaShip-3 and SSS-3 is 129.1188 while the average hash value of CIFAR-3 and SSS-3 is 127.7184. This indicates that SeaShip-3 and SSS-3 are more similar than CIFAR-3. From Figure 4, it can also be seen that the color of the correlation heat map of SeaShip-3 and SSS-3 is brighter than that of CIFAR-3 and SSS-3. indicating that SeaShip-3 is strong correlate to SSS-3, and CIFAR-3 is a weak correlate SSS-3.

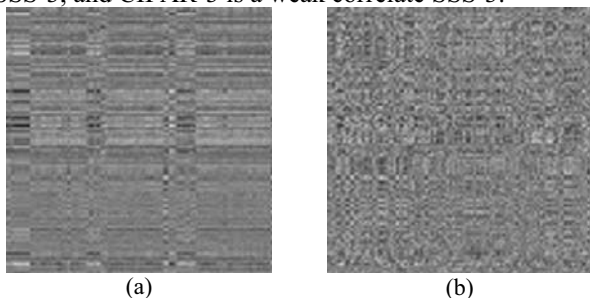


Figure 4. (a) The result of the hash algorithm between SeaShip-3 and SSS-3; (b) The result of the hash algorithm between CIFAR-3 and SSS-3.

3. Model and Experiment Method

3.1 Transfer Learning

Applications of transfer learning mainly include computer vision, text classification, behavior recognition, natural language processing, indoor positioning, video surveillance, public opinion analysis, and human-computer interaction [27-32]. The principle of transfer learning is to start with the model developed by task A and reuse it in the process of developing a model for task B. A complete learning task is composed of domains and tasks. A domain D consists of two components: a feature space x and a marginal probability distribution $P(x)$, so that $D = \{x, P(x)\}$. Defining the tag space Y and the model prediction function $f(\cdot)$, Task $T = \{y, f(\cdot)\}$. The definition of transfer learning is: Given a source domain D_s and a learning task T_s , a target domain D_T and a learning task T_T , transfer learning is dedicated to using the knowledge of D_s and T_s to help improve the learning of the target prediction function $f(\cdot)$ in T_T . D_s and D_T can be equal or unequal, and T_s and T_T can also be equal or unequal. The specific classification is shown in Table 1.

Unsupervised transfer learning was used in this paper.

Table 1. Classification of types of transfer learning.

Domain	Task	Learning setting	
$D_s = D_T$	$T_s = T_T$	Traditional transfer learning	
$D_s = D_T$	$T_s \neq T_T$	Inductive transfer learning	
$D_s \neq D_T$	$T_s \neq T_T$	Unsupervised transfer learning	Transfer learning
$D_s \neq D_T$	$T_s = T_T$	Transductive transfer learning	

3.2 Convolutional Neural Network

3.2.1 Principle

A convolutional neural network consists of an input layer, a hidden layer, and a fully connected layer. The main task of the input layer is to input data. The input layer of the one-dimensional network can process one-dimensional or two-dimensional data, while the input layer of the two-dimensional network can process two-dimensional or three-dimensional data. In addition, the size of the input layer is fixed and should be consistent with the network. The fully connected layer is the output layer of the network. In the hidden layer of the convolutional neural network, a large number of parameters are accumulated through the calculation of the convolutional layer and the pooling layer. A large number of features are extracted from the fully connected layer for nonlinear combination, and the results are finally output. The hidden layer of a convolutional neural network is the main computing part

of the network and is composed of multiple parts. Its main architecture can be divided into the following types:

(1) Convolutional Layer

The convolutional layer is composed of a convolution kernel and an activation function (Figure 5). The convolution kernel is mainly used to extract features. The convolution kernel has no size limit and is mostly a regular rectangle. A convolution layer contains multiple convolution kernels, and each convolution kernel provides a set of parameters for passing to the next layer. The function of the activation function is to assist the convolution kernel in expressing complex features. Commonly used activation functions include ReLU, sigmoid, dropout, among others. A large number of experiments have shown that the existence of the activation function can improve the classification accuracy [16].

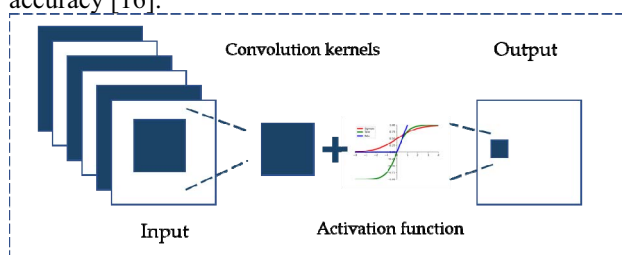


Figure 5. Convolution layer schematic.

(2) Pooling Layer

The function of the pooling layer (Figure 6) is to reduce the dimensionality of the data, remove redundant information, and, at the same time, expand the receptive field and achieve invariance, including translation invariance, rotation invariance, and scale invariance. Pooling also has the effect of simplifying the complexity of the network, reducing the amount of calculation, and reducing memory consumption [16].

The convolutional neural network finally realizes the classification of the image through the calculation of the above layers.

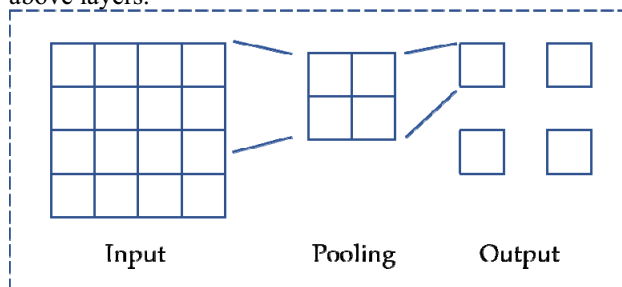


Figure 6. Pooling layer schematic

3.2.2 Architecture

There are many architectures of convolutional neural networks. LeNet-5 [15] is the first CNN, with excellent performance in handwriting recognition but, limited by the hardware conditions at the time it was devised, the network cannot be applied well to all types of images. AlexNet [16] introduced the ReLU function in the network architecture, which, to a certain extent, solved the problem of gradient disappearance. GoogLeNet [17] integrates multi-scale convolutional transformation through the idea of splitting, transforming, and merging.

The traditional convolutional layer is replaced with small blocks. The concept of auxiliary learner is also introduced to accelerate the network convergence speed. ResNet [18] introduced the concept of residual learning in the network architecture, and performs well in image recognition and localization tasks. VGG [20], with a depth of 19 layers, is, at present, the network architecture with the most parameters. The small-scale filter used in the network replaces the large-scale filter, so that it achieves good image classification results and spatial positioning.

3.3 Method

The data were used for training the combined transfer learning and convolutional neural network. The goal was to train a convolutional neural network with strong generalization and high accuracy, which can be used for unfamiliar sonar band data, and overcome the limitations of training a convolutional neural network with small samples of sonar images. We first used the SSS-3 dataset for network training, and saved the results as a control group for subsequent experiments. We then used the weakly correlated data to train the network, extracted and saved the network parameters, used them on the new network, and then used the SSS-3 data to train the network. Finally, we used the strongly correlated data to train the network, and saved the network parameters, put them on the new network, and then trained with the SSS-3 data. Under normal circumstances, the training process of the convolutional neural network is shown in Figure 7. The training process of combined transfer learning and convolutional neural network used in the experiment is shown in Figure 8.

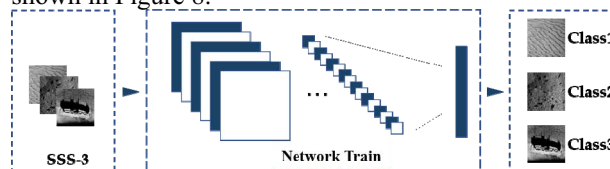


Figure 7. Convolutional neural network training process.

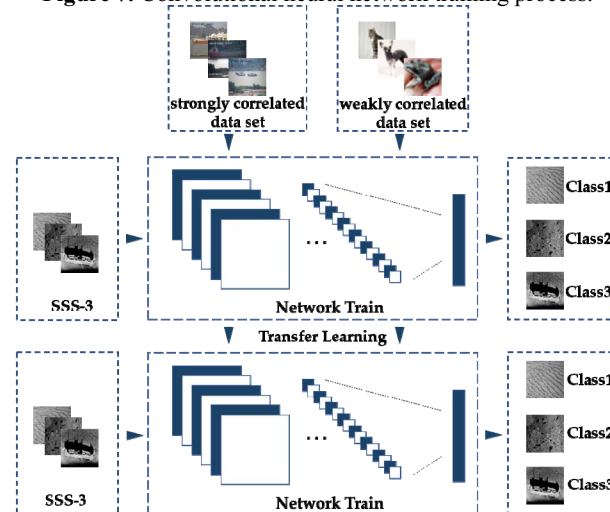


Figure 8. Combined transfer learning and convolutional neural network training process.

3.4 Experiment

The experiments were all run on OMEN on an HP Laptop 15-dc1061TX equipment. The hardware configuration includes an Intel i7-9750H CPU and NVIDIA GTX 1650 GPU. The software configuration includes Pytorch 1.4.0 and Python 3.7.6. The following three experiments were performed on five network models: LeNet-5, AlexNet, GoogLeNet, ResNet, and VGG:

- (1) Non-transfer: Train the network with SSS-3.
- (2) Weakly correlated data transfer: Train the network with weakly correlated data, extract the parameters into the new network, and then train the network with SSS-3.
- (3) Strongly correlated data transfer: Train the network with strongly correlated data, extract the parameters into the new network, and then train the network with SSS-3.

There were fifteen groups in all. The network model was trained with training data. After the training was completed, it was tested with the test data. The accuracy of the network on the test data was used as the final evaluation index.

3.5 Precision Evaluation

A confusion matrix is an important tool for model accuracy evaluation. It is represented by a matrix with N rows and N columns. The number of correctly classified examples, that is, the number of instances that are actually positive and classified as positive by the classifier is TP (True Positives). The number of incorrectly classified as positive examples, that is, the number of instances that are actually negative but classified as positive by the classifier is FP (False Positives). The number of incorrectly classified as negative examples, that is, the number of instances that are actually positive but classified as negative by the classifier is FN (False Negatives). The number of negative examples that are correctly classified, that is, the number of examples that are actually negative and classified as negative by the classifier is TN (True Negatives). The correct rate is a commonly used evaluation index. A higher correct rate corresponds to a better model effect. The calculation formula is as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \quad (2)$$

4 Results

4.1 LeNet-5

The accuracy of the experiments on LeNet-5 was 78.43% without transfer (Figure 9a). The accuracy was 80.39% when the weakly correlated data was transferred (Figure 9b). The accuracy was 84.31% when the strongly correlated data was transferred (Figure 9c). The performance of the three data of experiments on the test data is summarized in Figure 9d and shows that the transfer of the strongly correlated data is higher than the

transfer of the weakly correlated data, and it is higher than for the non-transferred.

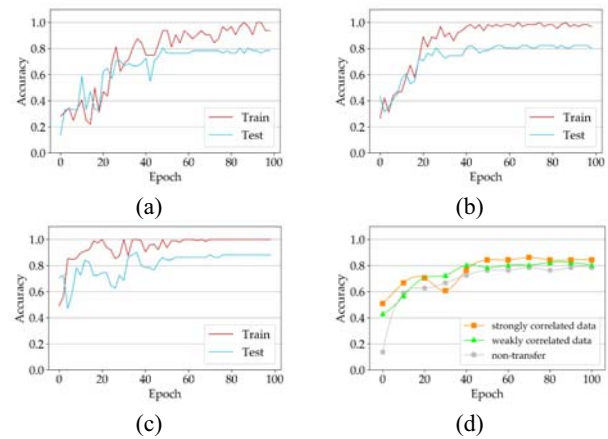


Figure 9. (a) Non-transfer; (b) Transfer of the weakly correlated data; (c) Transferred of the strongly correlated data; (d) The performance of the three data of experiments on the test data.

4.2 AlexNet

The accuracy of the experiments on AlexNet was 76.47% without transfer (Figure 10a). The accuracy was 82.35% when the weakly correlated data was transferred (Figure 10b). The accuracy was 88.24% when the strongly correlated data was transferred (Figure 10c). The performance of the three data experiments on the test data is summarized in Figure 10d, showing that the transfer of the strongly correlated data is higher than the transfer of the weakly correlated data, and it is higher than the for the non-transferred.

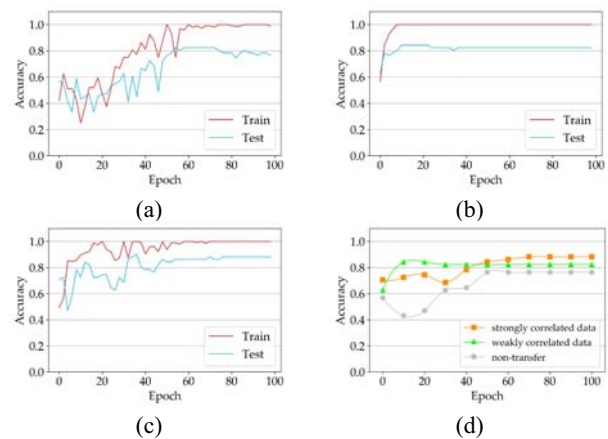


Figure 10. (a) Non-transfer; (b) Transfer of the weakly correlated data; (c) Transferred of the strongly correlated data; (d) The performance of the three data of experiments on the test data.

4.3 GoogLeNet

The accuracy of the experiments on GoogLeNet was 82.35% without transfer (Figure 11a) while it was 86.27% when the weakly correlated data was transferred (Figure 11b). The accuracy was 88.24% when the strongly correlated data was transferred (Figure 11c). The performance of the three data of experiments on the test data is summarized in Figure 11d which shows that the transfer of the strongly correlated data is higher than the

transfer of the weakly correlated data, and it is higher than the non-transferred.

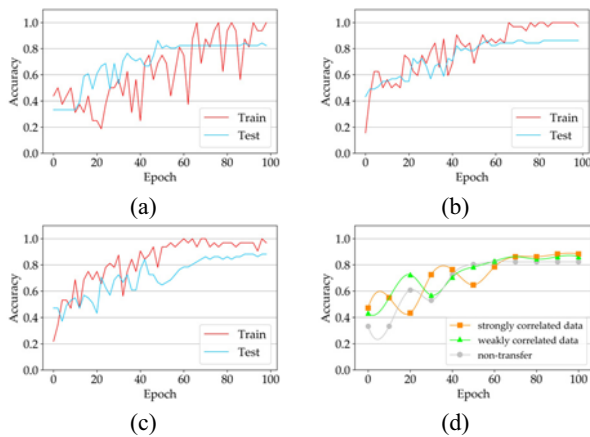


Figure 11. (a) Non-transfer; (b) Transfer of the weakly correlated data; (c) Transferred of the strongly correlated data; (d) The performance of the three data of experiments on the test data.

4.4 ResNet

The accuracy of the experiments on ResNet was 86.27% without transfer (Figure 12a). The accuracy was 86.27% when the weakly correlated data is transferred (Figure 12b). The accuracy was 90.20% when the strongly correlated data is transferred (Figure 12c). The performance of the three data of experiments on the test data is summarized in Figure 12d, which shows that the transfer of the strongly correlated data is higher than the transfer of the weakly correlated data, and it is higher than the non-transferred.

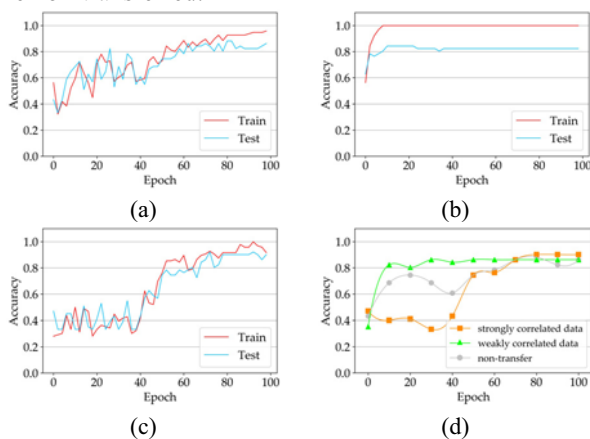


Figure 12. (a) Non-transfer; (b) Transfer of the weakly correlated data; (c) Transferred of the strongly correlated data; (d) The performance of the three data of experiments on the test data.

4.5 VGG

The accuracy of the experiments on VGG was 80.39% without transfer (Figure 13a). The accuracy was 92.16% when the weakly correlated data was transferred (Figure 13b). The accuracy was 98.04% when the strongly correlated data was transferred (Figure 13c). The performance of the three data experiments on the test data is summarized in Figure 13d, which shows that the transfer of the strongly correlated data is higher than the

transfer of the weakly correlated data, and it is higher than the non-transferred.

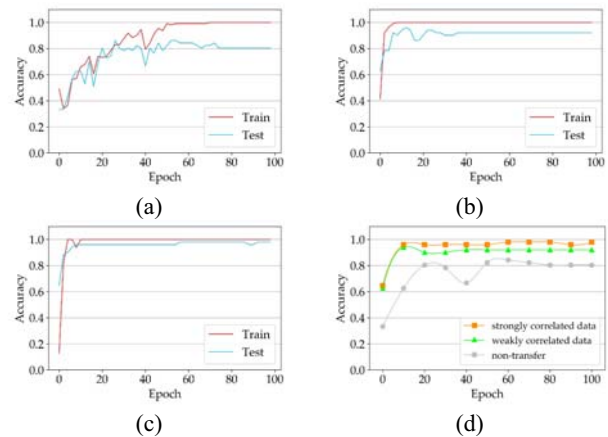


Figure 13. (a) Non-transfer; (b) Transfer of the weakly correlated data; (c) Transferred of the strongly correlated data; (d) The performance of the three data of experiments on the test data.

The accuracy of the above-mentioned experiments on the test set is summarized in Table 2.

Table 2 Test data accuracy

	Non-transfer	Weakly correlated data set	Strongly correlated data set
LeNet-5	78.43%	80.39%	84.31%
AlexNet	76.47%	82.35%	88.24%
GoogLeNet	82.35%	86.27%	88.24%
ResNet	86.27%	86.27%	90.20%
VGG	80.93%	92.16%	98.04%

In LeNet-5, the classification accuracy of weakly correlated data increased by 1.96%; the classification accuracy of strongly correlated data is increased by 5.88%; At the same time, the classification accuracy of strongly correlated data is 3.92% higher than that of weakly correlated data.

In the AlexNet architecture, compared with the non-transferred experimental group, the classification accuracy of weakly correlated data was improved by 5.88%. The classification accuracy of strongly correlated data increased by 11.77%. At the same time, the classification accuracy of strongly correlated data is 5.89% higher than that of weakly correlated data.

In the GoogLeNet architecture, compared with the non-transferred experimental group, the classification accuracy of weakly correlated data was improved by 3.92%. The classification accuracy of strongly correlated data increased by 5.89%. At the same time, the classification accuracy of strongly correlated data is 1.97% higher than that of weakly correlated data.

In the ResNet architecture, compared with the non-transferred experimental group, the classification accuracy of weakly correlated data was improved by 0%. The classification accuracy of strongly correlated data increased by 3.93%. At the same time, the classification accuracy of strongly correlated data is 3.93% higher than that of weakly correlated data.

In the VGG architecture, compared with the non-transferred experimental group, the classification accuracy of weakly correlated data was improved by 11.77%. The classification accuracy of strongly

correlated data increased by 17.65%. At the same time, the classification accuracy of strongly correlated data is 5.88% higher than that of weakly correlated data.

The comparisons of difference value between various data sets were shown in Table 3.

Table 3 difference value between various data sets

	W-N	S-N	S-W
LeNet-5	1.96%	5.88%	3.92%
AlexNet	5.88%	11.77%	5.89%
GoogLeNet	3.92%	5.89%	1.97%
ResNet	0%	3.93%	3.93%
VGG	11.77%	5.88%	5.88%

Remark: N=Non-transfer, W=Weakly correlated data, S=Strongly correlated data.

5 Discussion

5.1 Performance Comparison of Transfer Learning Methods and Non-transfer Learning Methods

Comparing the three data experiments under different architectures, the accuracy after using transfer learning is higher than that without using transfer learning (Figure 14).

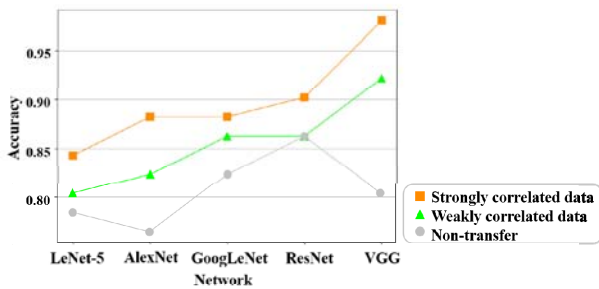
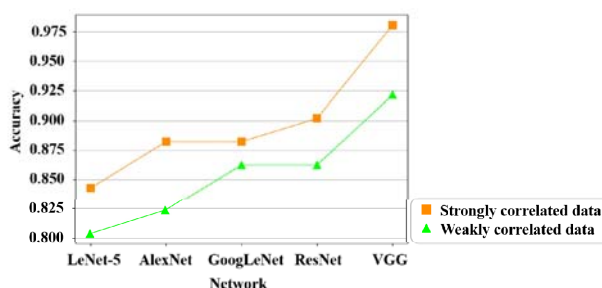


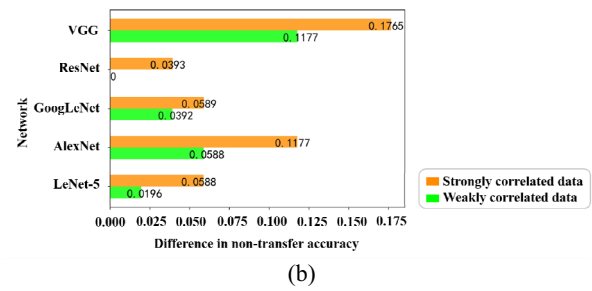
Figure 14. Classification accuracy of LeNet-5, AlexNet, GoogLeNet, ResNet, and VGG with strongly correlated and weakly correlated data.

5.2 Performance comparison between strongly correlated data transfer and weakly correlated data transfer

It is not difficult to conclude that, in all experiments, the accuracy of the strongly correlation data is better than that of the weakly correlation data (Figure 15). This shows that the method of using strongly correlation data transfer is greatly beneficial to training the network with small samples.



(a)



(b)

Figure 15. (a) Transfer learning accuracy; (b) Difference between the accuracy of strongly and weakly correlated data and the accuracy of non-transfer.

5.3 Initial Accuracy Analysis

At the core of transfer learning in deep learning is the transfer of network parameters. Generally, the initial parameters of the network are random. As the network training algorithm continues to modify the network parameters, the goal of accurate image classification and recognition is finally achieved. The initial parameters of each network are shown in Figure 16. In most cases, the initial accuracy of transfer learning is higher than that of the non-transferred experiment. In all cases, the initial accuracy of the strongly correlated data after transfer was higher than that of the experiment without transfer. This shows that transfer learning has a positive effect on network training.

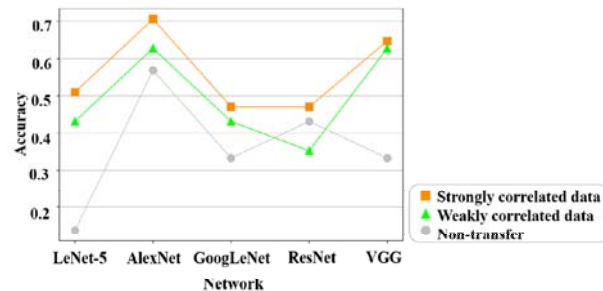


Figure 16. Initial accuracy of different networks with strongly and weakly correlated data and with non-learning transfer.

6 Conclusions

It is very difficult to obtain and label shipwreck side scan sonar images, and the small sample of data used for convolutional neural network training has additional limitations. Aiming at overcoming the classification problems of shipwreck sonar images, we proposed a solution combining transfer learning with a convolutional neural network under small samples. Comparing the experimental results, the following conclusions can be drawn:

(1) Compared with the case of non-transfer learning, the transfer of strongly correlated data and the transfer of weakly correlated data can effectively improve the accuracy of training convolutional neural networks with small sample data.

(2) In all experiments, the transfer accuracy of strongly correlated data is higher than that of weakly correlated data. In the VGG network, the classification

accuracy is as high as 98.20% after the transfer of the strongly correlated data.

(3) The initial accuracy of the strongly correlated data transfer is obviously better than that of the non-transferred case, indicating that the strongly correlated data transfer effectively transfers the features from the data to the network.

(4) The VGG network after the transfer of the strongly correlated data performs better than other methods in terms of accuracy and generalization. It is the most suitable convolutional neural network for small-sample shipwreck sonar image data.

The method of combining transfer learning with a convolutional neural network not only overcomes the difficulty of training for shipwreck sonar image recognition, but also provides an effective solution for other small sample classification problems. In the future, the method of combining transfer learning and convolutional neural network will be used in side scan sonar image mapping and real-time image recognition of unfamiliar shipwreck strips.

Acknowledgement:

This research was supported by the National Natural Science Foundation of China (41806117), and A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

1. X. Wang, A. Wang, T. Jiang, Y. Yang, and B. Zhang.(2019) Review of application areas for side scan sonar image. *Bulletin of Surveying and Mapping*, 01: 1-4.
2. S. Reed, Y. Petillot, and J. Bell.(2003) An automatic approach to the detection and extraction of mine features in sidescan sonar. *IEEE Journal of Oceanic Engineering*, 28: 90-105.
3. F. Yang, Z. DU, Z. Wu, J. Li, and F. Chu.(2006) Object Recognizing on Sonar Image Based on Histogram and Geometric Feature. *Marine Science Bulletin*, 05: 64-69.
4. S. Reed, Y. Petillot, and J. Bell.(2004) Model-based approach to the detection and classification of mines in sidescan sonar. *Applied Optics*, 43:237-246.
5. H. Guo, T. Tian, G. Liu, and F. Lu.(2002) A Recognizing Method Based on Fuzzy Clustering on the Sonar Image of a Small Target on the Sea Bed. *Journal of Chinese Computer Systems*, 02: 139-141.
6. G. J. Dobeck, J. C. Hyland, and L. Smedley.(1997) Automated detection and classification of sea mines in sonar imagery. in: *Spie - Int Soc Optical Engineering*. Bellingham. pp. 90-110.
7. L. Zheng and K. Tian.(2018) Detection of small objects in sidescan sonar images based on POHMT and Tsallis entropy. *Signal Processing*, 142: 168-177.
8. E. Dura, Z. Yan, L. Xuejun, G. J. Dobeck, and L. Carin.(2005) Active learning for detection of mine-like objects in side-scan sonar imagery. *IEEE Journal of Oceanic Engineering*, 30: 360-371.
9. F. Liu, S. Wang, and Y. Zhang.(2018) Overview on Models and Applications of Support Vector Machine. *Computer Systems & Applications*,27: 1-9.
10. A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi.(2020) A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53: 5455-5516.
11. S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, and X. Zhou.(2020) Deep Snake for Real-Time Instance Segmentation. in: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 8530-8539.
12. M. Ding.(2020) Learning Depth-Guided Convolutions for Monocular 3D Object Detection. in: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 11669-11678.
13. E. Shelhamer, J. Long, and T. Darrell.(2017) Fully Convolutional Networks for Semantic Segmentation. *Ieee Transactions on Pattern Analysis And Machine Intelligence*,39: 640-651.
14. Y. Taigman, M. Yang, M. Ranzato, and L. Wolf.(2014) DeepFace: Closing the Gap to Human-Level Performance in Face Verification. in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus. pp. 1701-1708.
15. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner.(1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86: 2278-2324.
16. A. Krizhevsky, I. Sutskever, and G. E. Hinton.(2017) ImageNet Classification with Deep Convolutional Neural Networks. *Communications Of the Acm*, 60: 84-90.
17. C. Szegedy. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston. pp. 1-9.
18. K. He, X. Zhang, S. Ren, and J. Sun.(2016) Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas. pp. 770-778.
19. V. Badrinarayanan, A. Kendall, and R. Cipolla.(2017) SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *Ieee Transactions on Pattern Analysis And Machine Intelligence*, 39: 2481-2495.
20. K. Simonyan and A. Zisserman.(2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science*.
21. Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li.(2018) SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. *IEEE Transactions on Multimedia*, 20: 2593-2604.
22. M. Valdenegro-Toro.(2017) Best practices in convolutional networks for forward-looking sonar image recognition. in: *OCEANS 2017*. Aberdeen. pp. 1-9.
23. H. Liang, L. Jin, and C. Yang.(2019) Research on Underwater Target Recognition Based on Depth Learning with Small Sample. *journal of Wuhan University of Technology (Transportation Science & Engineering)*, 43: 6-10.
24. R. Mehrotra, M. A. Ansari, R. Agrawal, and R. S. Anand.(2020) A Transfer Learning approach for AI-based classification of brain tumors. *Machine Learning with Applications*, 2.
25. X. Zhen, B. Pan, and J. Zhang.(2020) Power tower detection in remote sensing imagery based on deformable network and transfer learning. *Acta Geodaetica et Cartographica Sinica*, 49: 1042-1050.
26. Z. Zhu, X. Fu, and Y. Hu.(2020) A sonar Image Recognition Method Based on Convolutional Neural

- Network Trained through Transfer Learning," *Journal of Unmanned Undersea Systems*, 28: 89-96.
27. S. J. Pan and Q. A. Yang.(2010) A Survey on Transfer Learning. *Ieee Transactions on Knowledge And Data Engineering*,22: 1345-1359.
 28. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson.(2014) How transferable are features in deep neural networks ?. *Advances In Neural Information Processing Systems*, vol. 27.
 29. A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson.(2014) CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. in: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. Columbus. pp. 512-519.
 30. J. Howard and S. Ruder.(2018) Universal Language Model Fine-tuning for Text Classification. in: *Proceedings Of the 56th Annual Meeting Of the Association for Computational Linguistics*. Melbourne. pp. 328-339.
 31. Z. Wang, X. XU, and L. Zhang.(2020) Study of deep transfer learning for SAR ATR based on simulated SAR images. *ournal of University of Chinese Academy of Sciences*, 37: 516-524.
 32. J. Liu, N. Liu, Z. Pan, and X. You.(2019) Domain adaptive methods for device diversity in indoor localization. *Journal of Southeast University(English Edition)*, 35: 424-430.