

Research on Vector Road Data Matching Method Based on Deep Learning

Lin Zhao^{1,2}, Yanru Liu^{1*}, Yuefeng Lu^{1,3,4*}, Ying Sun¹, Jing Li¹, Kaizhong Yao¹

¹School of Civil and Architectural Engineering, Shandong University of Technology, Zibo, China

²School of Geography Science, Nanjing Normal University, Nanjing, China

³Hunan Provincial Key Laboratory of Geo-Information Engineering in Surveying, Mapping and Remote Sensing, Hunan University of Science and Technology, Xiangtan, China

⁴State Key Laboratory of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, China

Email: *yflu@sdut.edu.cn, *21407010764@stumail.sdut.edu.cn

How to cite this paper: Zhao, L., Liu, Y.R., Lu, Y.F., Sun, Y., Li, J. and Yao, K.Z. (2023) Research on Vector Road Data Matching Method Based on Deep Learning. *Journal of Applied Mathematics and Physics*, 11, 303-315.

<https://doi.org/10.4236/jamp.2023.111017>

Received: December 24, 2022

Accepted: January 28, 2023

Published: January 31, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Most of the existing vector data matching methods use traditional feature geometry attribute features to match, however, many of the similarity indicators are not suitable for cross-scale data, resulting in less accuracy in identifying objects. In order to solve this problem effectively, a deep learning model for vector road data matching is proposed based on siamese neural network and VGG16 convolutional neural network, and matching experiments are carried out. Experimental results show that the proposed vector road data matching model can achieve an accuracy of more than 90% under certain data support and threshold conditions.

Keywords

Deep Learning, Vector Matching, Similarity, VGG16, Siamese Network

1. Introduction

Vector data is an important part of the geographic information industry, and a large amount of vector data is produced based on the different needs of different departments. When facing the update problem of massive vector data, how to integrate and analyze them efficiently, give incremental updates, speed up the update speed of data, improve the update efficiency of data, and increase the reuse rate is one of the hot issues in GIS research field [1]. Since vector data may have a certain scale and timing differences depending on the source, it is necessary to use vector data matching methods for efficient data updates in order to

reduce the update cost of data and improve the reuse rate of data.

Line entity matching is generally achieved by comparing topological relationships between line segments as well as geometric similarity and spatial distance differences between nodes. Filin *et al.* [2] used Hausdorff distance to determine the matching relationship between elements. In the case of large local differences between line element entities, using only the Euclidean distance difference of nodes will produce large errors in the matching results, so Deng *et al.* [3] extended the Hausdorff distance. Huang *et al.* [4] studied surface rivers by extracting the skeleton line of surface rivers and using the skeleton line nodes and radians to achieve matching between linear and surface rivers. Chen *et al.* [5] classified and analyzed the point elements constituting the lineal roads to find the distance difference and angle offset between important point elements to realize the matching of road networks at different scales. Based on the advantages of the ant colony algorithm, Gong *et al.* [6] proposed a constrained mathematical model to find the globally optimal matching scheme for road entities with the same name. Liu [7] studied vector road data into two cases, homologous and non-homologous, and matched homologous data with high similarity by comparing buffers; for non-homologous data with overall convergence, matching was performed by combining data parallelism and distance. Fu *et al.* [8] proposed a road network matching algorithm based on a logistic regression model. Fang [9] *et al.* proposed a model for matching using the topological relationship and spatial location between line elements. Wen [10] proposed three metrics applicable to vector road matching based on the geometric and topological characteristics of roads, which are spatial relationship descriptors based on the summation of the product of direction and distance, shape descriptors based on the vector of feature points, and area shape descriptor based on the minimum convex package. Yang [11] proposed a 2-channels-conv network with deeper layers than 2-channels network for heterogenous remote sensing image matching. Tao [12] *et al.* applied the deep learning approach to the heterogenous SAR image and visible image matching problem by using twin neural networks for similarity metric, and the matching results and accuracy were better than the traditional template matching method.

Most of the existing vector data matching methods use the traditional geometric attribute features of elements for matching. With the development of deep learning in the field of computer vision, how to use deep learning to deal with vector data matching problems has become a new direction of current research. Due to the poor applicability of more current similarity metrics in cross-scale data, it leads to low accuracy in identifying objects. In order to effectively solve this problem, this paper considers the application of deep learning methods for vector data matching, explores the neural network model applicable to vector data matching, and through experiments, continuously optimizes the matching method to obtain a higher matching accuracy. The technical route adopted in this paper is shown in **Figure 1**.

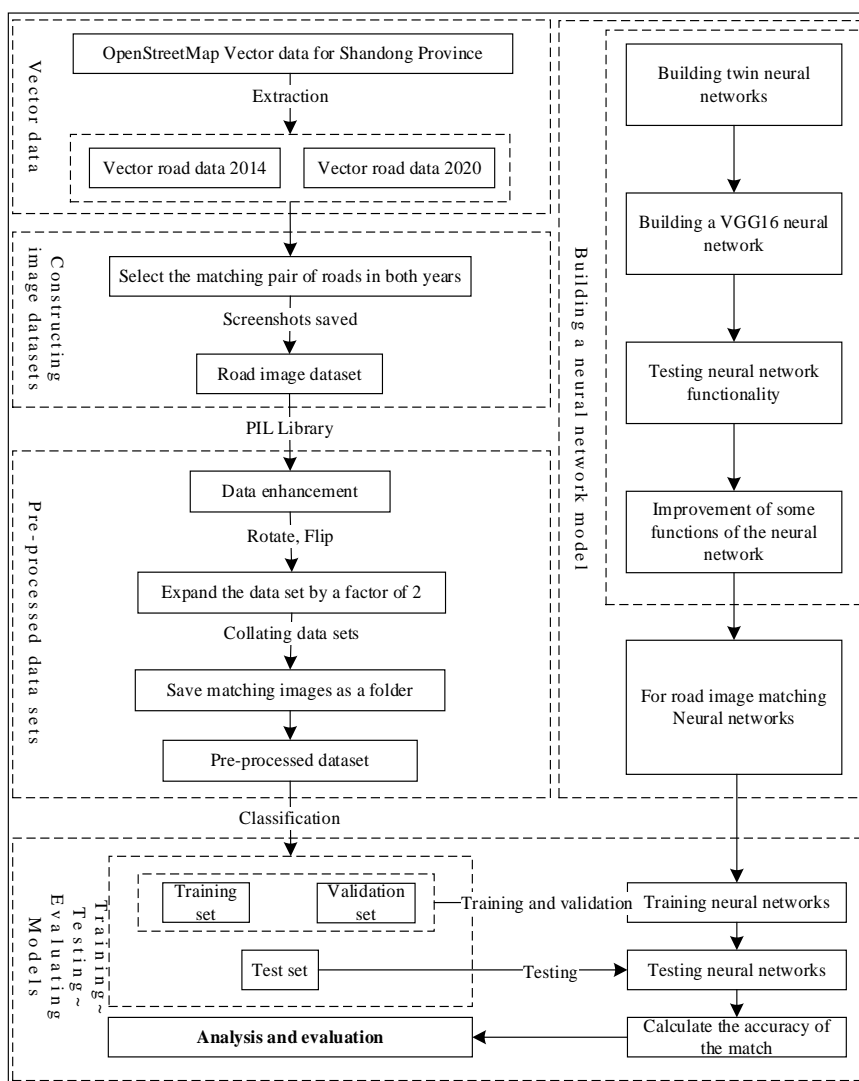


Figure 1. Technical route caption.

2. Data Set

The vector data of 2014 and 2020 provided by OSM (OpenStreetMap) with the coordinate reference system of GCS_WGS_1984 and the data format of shape file are used for the experimental data.

The highways, national roads and provincial roads are screened, and the roads with the same name matching 2014 and 2020 are selected by local zooming, in order to achieve cross-scale road matching so that they are displayed at different scales, and the effect is shown in **Figure 2**. According to the principle of uniform distribution of sampling area and random selection, a number of homonymous roads with typical geometric features are selected among highways, national roads and provincial roads, and ensure that the number of roads in each category is approximately equal, and are collected by the above method. Finally, 108 pairs and 216 road pictures were collected as the original data set for the experiment. Also, 40 pairs and 80 road pictures were collected as the original test set for the experiment.



Figure 2. Example of a matching road picture.

3. Matching Models

3.1. VGG16 Convolutional Neural Network

In early convolutional neural networks, such as AlexNet, LeNet, the features of input samples are extracted by larger convolutional kernels. Although the larger the size of the convolutional kernel, the greater the ability to extract spatial information, it also increases the number of parameters and computational effort.

Unlike AlexNet, VGGNet uses several consecutive 3×3 convolutional kernels instead of larger convolutional kernels such as 11×11 , 7×7 , and 5×5 . This way VGGNet has less number of parameters while having more nonlinear transformations. The depth of VGGNet is increased under the condition that it is guaranteed to have the same perceptual field, which improves the effectiveness of the neural network to some extent [13]. The more widely used structure in VGGNet is the VGG16 convolutional neural network, whose structure is shown in **Figure 3**.

3.2. Twin Neural Networks

A twin neural network is a neural network model that can weaken the existence of intensity differences between heterogeneous images by extracting features from the input samples and then measuring their similarity [12]. A twin neural network in the narrow sense consists of two identical neural networks [14] and is mainly used to deal with the case where two input samples are more similar, and the twin neural network used in the experiments of this paper is of this type; a twin neural network in the broad sense, which can consist of any two artificial neural networks [15], is mainly used to deal with the case where two input samples are more different.

The input of twin neural networks can be numerical data, image data or sequence data, and they have good scalability to classify untrained categories. Since twin neural networks have two inputs as well as two sub-neural networks, their training is more computationally intensive and takes longer compared to conventional networks, and their output is not the probability but the distance between the two inputs.

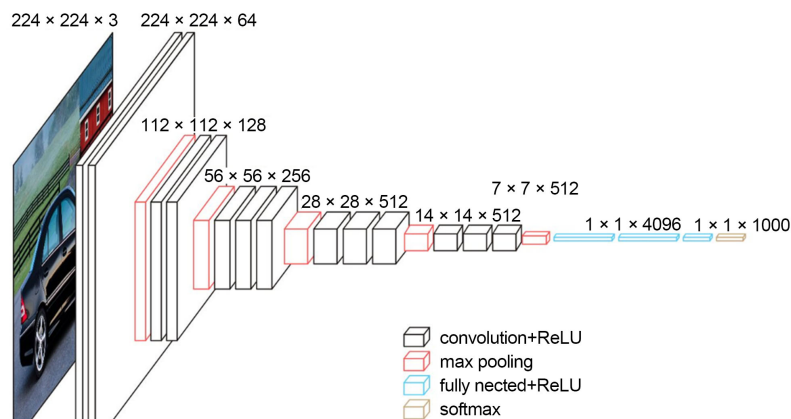


Figure 3. Structure of VGG16.

The structure of the twin neural network is shown in **Figure 4**, *i.e.*, the twin neural network is used for two inputs, and its *loss* is calculated by the network on the two branches, which can evaluate the similarity of the one-dimensional feature vector obtained after the processing of the two branches, *i.e.*, the similarity of the two inputs.

Since the dataset is a series of road pictures, for the twin neural network used in the experiment, its input is two pictures. After inputting the images and the corresponding labels into the neural network model, the output of the neural network and the real labels of the images are subjected to Cross-Entropy operation to obtain the final loss. The loss function used in the experiment is the Binary Cross-Entropy *loss* function (Binary Cross-Entropy, the expression is shown in Equation (1)), which returns a higher loss value for wrong predictions and a lower *loss* value for good predictions.

$$loss = -\frac{\sum_{i=1}^N y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))}{N} \quad (1)$$

In Equation (1): N is the number of samples; y is the label representing whether two samples match, $y = 1$ represents a match and $y = 0$ represents a mismatch; $p(y)$ is the probability of correct match for each sample.

For correctly matched samples, $y = 1$ and $loss = -\log(p(y))$, when $p(y)$ is larger, *loss* is smaller, and ideally $p(y) = 1$, at this time $loss = 0$. For incorrectly matched samples, $y = 0$ and $loss = -\log(1 - p(y))$, when $p(y)$ is smaller, *loss* is smaller, and ideally $p(y) = 0$, at this time $loss = 0$. Therefore, Binary Cross-Entropy as the loss function of vector road data matching network can measure the accuracy of classification.

3.3. Combination of VGG16 and Twin Neural Network

The twin neural network is chosen as the backbone framework of the road picture matching network for inputting two pictures; the VGG16 convolutional neural network is chosen as the backbone feature extraction network of the twin neural network for feature extraction. The twin neural network uses the feature

vectors of the two road images extracted by VGG16 to perform matching by calculating the similarity between them.

The basic framework of the vector road data matching network constructed by using the respective features of the twin neural network and the VGG16 convolutional neural network is shown in **Figure 5**.

Figure 6 illustrates the whole process of the network implementation from the input samples, to the output image similarity. Where X_1 and X_2 represent a pair of input data, G_w represents the neural network model with parameter w , $G_w(X)$ serves to convert the input data X into feature vectors, and E_w is used to describe the L1 distance between the feature vectors of the two input data. The specific implementation process is described below.

First, the features of the input images are extracted using the VGG16 convolutional neural network, which mainly involves the convolutional operations of the convolutional layers. The convolutional operations are performed to extract different features of the input images, and these features become more and more complex as the number of layers of the neural network increases. After obtaining the multidimensional feature vectors, the functions in the NumPy library are used to convert them into one-dimensional vectors, and finally the one-dimensional feature vectors of the two input images are obtained.

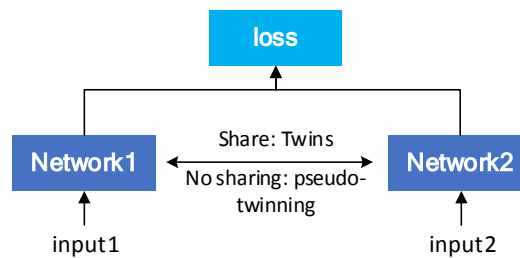


Figure 4. Structure of siamese network.

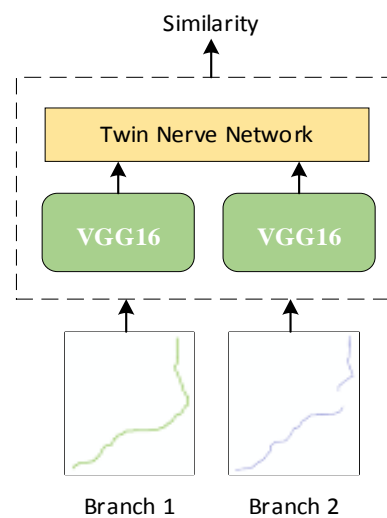


Figure 5. Structure of the vector road data matching network.

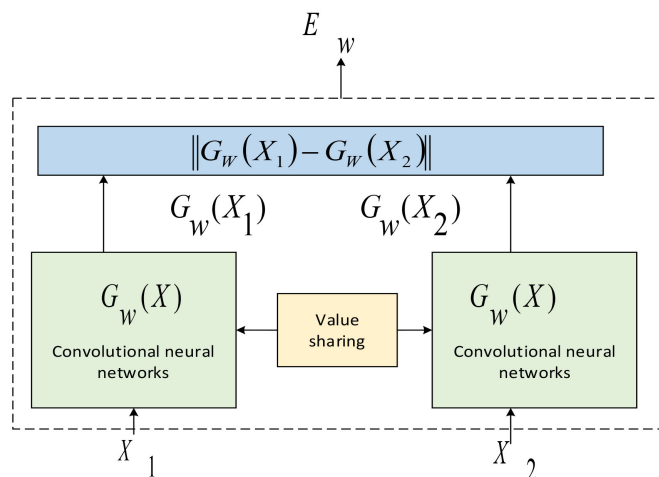


Figure 6. The implementation process of the vector road data matching network.

Second, the L1 parametric of the interpolation of these two feature vectors is obtained, which is also equivalent to deriving the L1 distance between them. Where, the L1 parametric number represents the sum of the absolute values of the non-zero elements in vector x . The expression is shown in Equation (2), which serves to allow feature selection and can directly set the coefficients of features of lower importance to zero, thus improving the sparsity of the neural network model parameters; the L1 distance, also called the Manhattan distance, is the projection of the axis generated by the line segment formed by two points on a fixed right-angle coordinate system in Euclidean space the sum of the distances [16], and for the n -dimensional vectors $a(x_{11}, x_{12}, \dots, x_{1n})$ and $b(x_{21}, x_{22}, \dots, x_{2n})$, the Manhattan distance is as in Equation (3).

$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad (2)$$

In Equation (2): $\|x\|_1$ is the L1 parametrization; x_i is the coordinate of each dimension of the vector x .

$$d_{12} = \sum_{k=1}^n |x_{1k} - x_{2k}| \quad (3)$$

In Equation (3): d_{12} is the Manhattan distance between the two vectors; x_{1k} and x_{2k} are the coordinates of each dimension of the two vectors, respectively.

Then, the L1 distance is input to the excitation layer using two fully connected layers, and the result is transformed to the (0, 1) interval, representing the similarity of the two input road images, by the Sigmoid function.

Among them, the function of the excitation layer is to make a nonlinear mapping to the output of the convolutional layer, introducing nonlinear factors to the neurons. The pooling layer, which is connected afterwards, functions as a pooling (also known as downsampling) operation for the feature matrix, mainly through feature dimensionality reduction, data compression and reduction of the number of parameters, to sparse the feature matrix, reduce the size of the

feature matrix, and prevent the model from overfitting. The pooling layer is introduced into the convolutional neural network to ensure the translation invariance of the input image, *i.e.*, the same image is input again after flipping and deforming, and similar results can be obtained after processing by the pooling layer. The fully connected layer, located at the end of the neural network, is used to refit the features downsampled by the pooling layer, *i.e.*, the feature matrix obtained by the pooling layer processing is expanded into a one-dimensional vector to integrate the features together to reduce the loss of feature information.

Finally, using the above method, the original test set is traversed and the one with the highest similarity to each image is used as the road image with which it matches, completing the automatic matching of road images.

4. Experiment and Analysis

The training is divided into two stages: forward propagation and backward propagation, each with a different learning rate. The first stage only makes relatively rough predictions, just to try to maintain the recall rate; the second stage makes further adjustments and predictions based on the first stage predictions combined with the features extracted by the convolutional neural network, making the final predictions more accurate.

The task of the prediction stage is: traverse all the road pictures in the original test set as one input *input1* of the twin neural network, and then traverse all the road pictures to be matched in it as another input *input2* of the twin neural network; then use the networks on the two branches to extract the features of *input1* and *input2* and calculate their similarity; finally take the road picture with the highest similarity *input2* as the road picture matching with road picture *input1* and judge whether the match is a correct match according to the number of the two images, if it is consistent, it is a correct match, otherwise it is a wrong match. Also in the prediction process, the number of correct matches is counted, and finally the matching accuracy is calculated.

4.1. Preliminary Experiments

The experiments are conducted using the original training set (hereinafter referred to as “Experiment 1”), and the parameters are set as shown in **Table 1**.

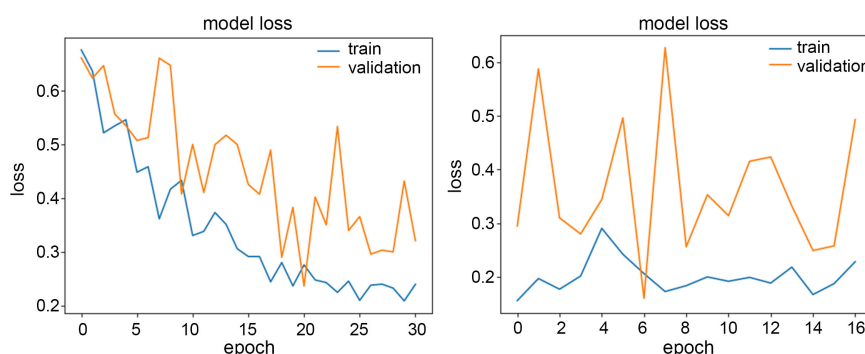
Training was obtained for the two phases with the relationship shown in **Figure 7**.

As can be seen from the above figure, a total of 31 *epochs* were performed in the first stage and a total of 17 *epochs* were performed in the second stage.

In the 57th *epoch*, the *loss* of the training and validation sets are 0.2060 and 0.1599, respectively, and the *accuracy* is 0.9187 and 0.9375, respectively, with the best overall performance. The weights file generated by this *epoch* was selected for matching prediction on the original test set, and the number of successful matches was counted as 21 and the number of failed matches was 19, with a matching accuracy of 52.500%.

Table 1. Parameter settings for Experiment 1.

Parameter Name	Parameter Value
Total number of images in the training set and validation set	108 pairs, 216 sheets
Ratio of training set, validation set and test set	172:44:40 (4:1:1)
Batch size	32
Learning rate (I)	1e-4
Epoch (I)	50
Learning rate (II)	1e-5
Epoch (II)	50

**Figure 7.** Training results of Experiment 1.

4.2. Data Augmentation Experiments

Data augmentation is a regularization technique that primarily manipulates the training data by randomly altering it through a series of digital image processing methods. Using data augmentation techniques can improve the accuracy of neural network models while helping to avoid overfitting of the training. In addition, the amount of data in the dataset can be increased by data augmentation, thus reducing the need for large manually labeled datasets for deep learning. In general, the more training samples collected, the better, but in cases where it is not possible to increase the real training samples, data augmentation can be a good way to overcome the limitations of small datasets.

Based on the results of Experiment 1, this experiment (hereinafter referred to as “Experiment 2”) was conducted based on the data enhanced dataset, considering that the model was not sufficiently trained due to the small size of the dataset, resulting in poor matching accuracy.

The data enhancement was performed by first doubling the original dataset (108 pairs of road images) by flipping it up and down and randomly rotating it clockwise and counterclockwise by 5° , and then doubling the original dataset by flipping it left and right and randomly rotating it clockwise and counterclockwise by 5° . The test set is also processed in the same way. Thus, the data set used in Experiment 2 is 324 pairs and 648 road images, and the test set is 120 pairs and 240 road images.

The parameters of Experiment 2 were set as shown in **Table 2**, and the relationship between *loss* and *epoch* for the two phases of training was obtained as shown in **Figure 8**.

As can be seen from the above figure, a total of 20 *epochs* were performed in the first stage and a total of 16 *epochs* in the second stage.

In the 65th *epoch*, the best overall performance was achieved with *loss* of 0.1300 and 0.2800 and *accuracy* of 0.9766 and 0.9153 for the training and validation sets, respectively. The weights file generated by this *epoch* was selected for matching prediction on the test set after data enhancement, and the number of successful matches was counted as 68 and the number of failed matches was 52, with a matching accuracy of 56.667%.

4.3. Expanded Threshold Experiments

Based on the above two experiments, considering that in the vector road data matching problem, in addition to using deep learning to match the geometric features of road images, many other matching indicators, such as coordinate position, should be linked to achieve a comprehensive and integrated fine matching.

Therefore, when using deep learning for coarse matching of road images, the threshold value of matching can be appropriately reduced (the threshold value of both Experiment 1 and Experiment 2 is 1:1), so that the matching accuracy can be improved, and the precise matching results can be achieved by combining other indicators.

The parameters of this experiment (hereinafter referred to as “Experiment 3”) were set as in Experiment 2, and the relationship between *loss* and *epoch* for the two phases of training was obtained as shown in **Figure 9**.

As can be seen from the above figure, a total of 33 *epochs* were performed in the first stage and a total of 25 *epochs* in the second stage.

In the 65th *epoch*, the best overall performance was achieved with *loss* of 0.0833 and 0.1555 and *accuracy* of 0.9785 and 0.9453 for the training and validation sets, respectively. The weights file generated by this *epoch* was selected for matching prediction on the data-enhanced test set, and the matching thresholds were set to 1: 3, 1: 4, and 1: 5, and it was statistically found that.

Table 2. Parameter settings for Experiment 2.

Parameter Name	Parameter Value
Total number of images in the training set and validation set	324 pairs, 648 sheets
Ratio of training set, validation set and test set	518:130:120 (4:1:1)
Batch size	32
Learning rate (I)	1e-4
Epoch (I)	50
Learning rate (II)	1e-5
Epoch (II)	50

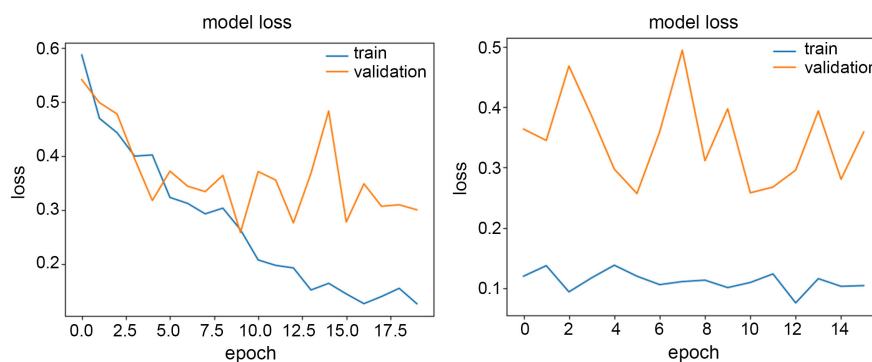


Figure 8. Training results of Experiment 2.

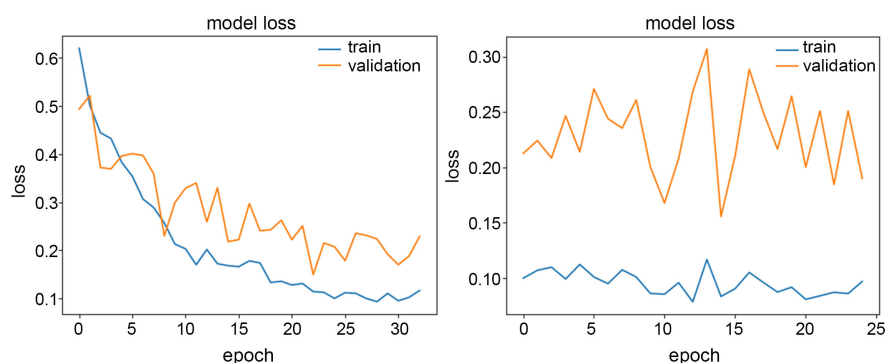


Figure 9. Training results of Experiment 3.

1) When the threshold value is 1:3, the number of successful matches is 99, the number of failed matches is 21, and the matching accuracy is 82.500%.

2) When the threshold is 1:4, the number of successful matches is 106, the number of failed matches is 14, and the matching accuracy rate is 88.333%.

3) When the threshold value is 1:5, the number of successful matches is 109, the number of failed matches is 11, and the matching accuracy rate is 90.833%.

5. Conclusions

Several vector data images of highways, national roads, and provincial roads in 2014 and 2020 were selected as the experimental objects, and for the vector road data matching problem, a deep learning approach was used to investigate.

1) The matching problem of road images is transformed into a problem of calculating similarity using a neural network model and considering the image with the highest similarity as the matching image.

2) During the experiment, the matching accuracy is improved by using data enhancement and expanding the threshold, and the feasibility of this deep learning-based vector data matching method is demonstrated by three comparison experiments.

For the research results obtained in this paper, there are still many problems to be solved, which need to be continuously accumulated and improved in practical applications. In the following aspects, further research and improvement are needed.

1) The input of the neural network proposed in this paper can only be road pictures, and it can not directly input and process the coordinate files, shape files, and other vector formats of roads.

2) The accuracy of the neural network proposed in this paper is not high enough in the vector path data matching problem, and subsequent studies will consider the key modification of the neural network to make it more applicable to the treatment of vector road data matching problems.

Acknowledgements

The authors thank the providers of the data used in this article, including the Major Project of High-Resolution Earth Observation System of China (No. GFZX0404130304); the Open Fund of Hunan Provincial Key Laboratory of Geo-Information Engineering in Surveying, Mapping and Remote Sensing, Hunan University of Science and Technology (No. E22201); the Agricultural Science and Technology Innovation Program (ASTIP No. CAAS-ZDRW202201); a grant from State Key Laboratory of Resources and Environmental Information System; the Innovation Capability Improvement Project of Scientific and Technological Small and Medium-sized Enterprises in Shandong Province of China (No. 2021TSGC1056).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Devogele, T., Parent, C. and Spaccapietra, S. (1998) On Spatial Database Integration. *International Journal of Geographical Information Science*, **12**, 335-352. <https://doi.org/10.1080/136588198241824>
- [2] Filin, S. and Doytsher, Y. (2000) The Detection of Corresponding Objects In A Linear-Based Map Conflation. *Surveying and Land Information Systems*, **60**, 101-108.
- [3] Deng, M., Li, Z. and Chen, X.Y. (2007) Extended Hausdorff Distance for Spatial Objects in GIS. *International Journal of Geographical Information Science*, **21**, 459-475. <https://doi.org/10.1080/13658810601073315>
- [4] Huang, W. and Jiang, J. (2011) Simple Geometry Matching of Multi-Scales Spatial Data. *Remote Sensing Information*, No. 1, 27-31. (In Chinese)
- [5] Chen, Y.M., Gong, J.Y. and Shi, W.Z. (2007) A Distance-Based Matching Algorithm for Multi-Scale Road Networks. *Acta Geodaetica et Cartographica Sinica*, **36**, 84-90. (In Chinese)
- [6] Gong, X.Y., Wu, F., Ji, C.W. and Zhai, R.J. (2014) Ant Colony Optimization Approach to Road Network Matching. *Geomatics and Information Science of Wuhan University*, **39**, 191-195. (In Chinese)
- [7] Liu, Y.N. (2014) Research on Multi-Source Vector Road Data Matching. *Modern Surveying and Mapping*, **37**, 16-18. (In Chinese)
- [8] Fu, Z.L., Yang, Y.W., Gao, X.J., et al. (2016) Road Networks Matching Using Multiple Logistic Regression. *Geomatics and Information Science of Wuhan University*,

- 41**, 171-177. (In Chinese)
- [9] Fang, M., Huo, L., Song, L., et al. (2018) Design of Linear Feature Matching Method Based on Node Similarity. *Bulletin of Surveying and Mapping*, No. 3, 66-70. (In Chinese)
- [10] Wen, Q. (2021) Research on Multi-scale Road Network Vector Data Matching Method. Shandong University of Technology, Zibo, 1-64. (In Chinese)
- [11] Yang, X.Y. (2021) Research on 2-Channels-Conv Network Algorithm of Remote Sensing Image Matching. Master's Thesis, Liaoning Technical University, Fuxin, 2-95. (In Chinese)
- [12] Tao, K., Wu, L.L., Han, P.L. and Wang, Z.P. (2022) Heterogeneous Image Matching Based on Siamese Neural Network. *Journal of Detection & Control*, **44**, 41-45. (In Chinese)
- [13] Chen, X., Xu, M.J. and Wang, W.H. (2020) Intelligent Parking System Based on Internet of Things and Image Recognition. *Computer Knowledge and Technology*, **16**, 187-189. (In Chinese)
- [14] Bromley, J., Guyon, I., LeCun, Y., et al. (1993) Signature Verification Using a "Siamese" Time Delay Neural Network. *International Journal of Pattern Recognition and Artificial Intelligence*, **7**, 669-688. <https://doi.org/10.1142/S0218001493000339>
- [15] Hughes, L.H., Schmitt, M., Mou, L., Wang, Y. and Zhu, X.X. (2018) Identifying Corresponding Patches in SAR and Optical Images with a Pseudo-Siamese CNN. *IEEE Geoscience and Remote Sensing Letters*, **15**, 784-788. <https://doi.org/10.1109/LGRS.2018.2799232>
- [16] Ji, B., Ye, Y.D. and Lu, H.X. (2013) Taxi Gathering Area Recognition Algorithm Based on Sample Weight. *Journal of Computer Applications*, **33**, 1338-1342. (In Chinese)