

Retraction

Retracted: A Segmentation Algorithm of Image Semantic Sequence Data Based on Graph Convolution Network

Security and Communication Networks

Received 26 December 2023; Accepted 26 December 2023; Published 29 December 2023

Copyright © 2023 Security and Communication Networks. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] Z. Jia and D. Chen, "A Segmentation Algorithm of Image Semantic Sequence Data Based on Graph Convolution Network," *Security and Communication Networks*, vol. 2021, Article ID 5596820, 11 pages, 2021.

Research Article

A Segmentation Algorithm of Image Semantic Sequence Data Based on Graph Convolution Network

Zheshu Jia and Deyun Chen 

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

Correspondence should be addressed to Deyun Chen; yangbo@hrbust.edu.cn

Received 24 February 2021; Revised 7 April 2021; Accepted 16 April 2021; Published 22 April 2021

Academic Editor: Chi-Hua Chen

Copyright © 2021 Zheshu Jia and Deyun Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image semantic data have multilevel feature information. In the actual segmentation, the existing segmentation algorithms have some limitations, resulting in the fact that the final segmentation accuracy is too small. To solve this problem, a segmentation algorithm of image semantic sequence data based on graph convolution network is constructed. The graph convolution network is used to construct the image search process. The semantic sequence data are extracted. After the qualified data points are accumulated, the gradient amplitude forms complete rotation field and no scatter field in the diffusion process, which enhances the application scope of the algorithm, controls the accuracy of the segmentation algorithm, and completes the construction of the data segmentation algorithm. After the experimental dataset is prepared and the semantic segmentation direction is defined, we compare our method with four methods. The results show that the segmentation algorithm designed in this paper has the highest accuracy.

1. Introduction

In recent years, deep learning has achieved revolutionary changes and success in the field of machine learning and data mining, especially for the processing of unstructured data, such as image recognition, natural language processing, and machine translation, which has made full progress and development, making artificial intelligence technology to a new level. Deep learning has powerful ability of feature extraction and expression. It can overcome the complicated preprocessing process of artificial feature engineering in traditional machine learning methods, automatically identify and extract the potential feature law behind the training data, and learn the feature law for data modeling. For image, speech, and other unstructured data, the processing method of the deep learning algorithm is to simplify it into grid data forms, such as multidimensional array or multidimensional vector (tensor), which is used for data modeling, recognition, and processing. For this kind of data which can be simplified into tensor form, it is called “Euclidean domain” data, and the strong point of deep learning is that it can be

used for efficient processing of Euclidean data [1]. However, in reality, there are many data that do not conform to the Euclidean domain, such as e-commerce data, social network data, and protein molecular spatial structure. These data have complex spatial topological relationships, which cannot be simply treated as tensor data such as multidimensional array or multidimensional vector. Therefore, similar data with spatial structure and spatial connection is called “non-Euclidean” data. For non-Euclidean domain data, the expression ability of the deep learning algorithm has certain limitations, and the processing effect is poor. Therefore, a new method that can efficiently process non-Euclidean domain data needs to be explored and proposed.

Image semantic segmentation is to cluster pixels belonging to the same category in digital image into a region. Generally speaking, scene annotation and human interpretation belong to the category of pixel-level classification. Its common point is that each pixel of image is classified into predefined specific category and background category. The image semantic segmentation algorithm has been applied to road sign detection and recognition, brain tumor

segmentation, medical instrument tracking, colon recess segmentation, land use, and cover classification, and other specific fields. Semantic segmentation has been applied to 2D image, video, and 3D data and has become one of the key questions in the field of computer vision. Traditional machine learning usually needs a delicate structure and professional knowledge to design a feature extractor when dealing with unprocessed data. It transforms the original data into an appropriate internal feature representation or feature vector and then constructs a classifier to detect and classify the input samples [2]. Deep learning is a feature learning method, which transforms the original data into a higher level and more abstract expression through some simple but nonlinear models. Compared with traditional pattern recognition, the core of deep learning is that different levels of features are not designed by artificial engineering but learned from data by a common learning process. Deep learning has also achieved the best results in other visual tasks such as object detection and recognition, action recognition, face verification and recognition, image segmentation, and so on.

Image object detection algorithms can be used to extract semantic concepts in a single video frame. Image object detection algorithms are mainly divided into two categories: traditional algorithms based on artificial design features and object detection models based on deep convolution neural network. The traditional object detection algorithm follows the framework of artificial design features and shallow classifier. This method extracts image features from the image content in the window by sliding the detection window on the image and then trains the traditional machine learning classifier based on these features. The effect of this algorithm depends heavily on the design of features and the performance of the classifier. At the same time, the mechanism of sliding window also makes the time cost of the algorithm very large. With the development of image feature extraction and machine learning theory, different methods are proposed to improve the effect and speed of object detection algorithm. These improvements mainly focus on the detection window, feature design, and classifier performance optimization. However, traditional feature extraction methods are difficult to mine deep feature values, which affects the subsequent image segmentation and object detection performance.

In this paper, we propose a novel segmentation algorithm of image semantic sequence data based on graph convolution network (GCN). First, we construct image search process based on GCN to update edge weights according to accuracy automatically, thereby reducing the randomness caused by sampling. Then, using the edge weights of images, the multiple-level features of semantic sequence data are extracted. Afterwards, due to the gradient amplitude factor calculated by continuous diffusion, the accuracy of the segmentation algorithm is improved. Finally, the results of simulation experiment indicate that the proposed method achieved better accuracy in shorter execution time, which outperformed the compared methods. In short, we present a new image segmentation algorithm and get promising results.

Our contribution is threefold as follows:

- (1) In view of the limitations of existing segmentation algorithms, this paper uses graph convolutional network to construct the image search process and constructs a segmentation algorithm of image semantic sequence data based on graph convolutional network.
- (2) After the accumulation of qualified data points, the gradient amplitude formed a complete rotation field, and there was no scattered point field in the diffusion process, which enhanced the application range of the algorithm, controlled the precision of the segmentation algorithm, and completed the construction of the data segmentation algorithm.
- (3) Experimental results show that the proposed segmentation algorithm has the highest segmentation accuracy.

The paper is organized as follows. In Section 2, we discuss related work, and in Section 3, we elaborate the proposed image segmentation algorithm. Then, we design a simulation experiment to validate the effectiveness of our method in Section 4. In Section 5, we summarize and review the algorithm proposed in this paper.

2. Related Work

2.1. Image Semantic Segmentation. Guo et al. provided an overview of image semantic segmentation based on deep learning and divided the literature into three categories: region-based, fully convolutional network (FCN)-based, and weakly supervised segmentation methods [3]. Hu et al. summarized the most commonly used RGB-D datasets for semantic segmentation, as well as traditional machine learning-based methods and deep learning-based network architectures for RGB-D segmentation [4]. Le et al. conducted extensive investigations on deep learning architectures, datasets, and evaluation methods to use deep neural networks for semantic segmentation of natural images [5]. Similarly, for medical imaging, Goceri and Goceri outlined the medical image analysis technology and application fields based on deep learning [6]. Hesamian et al. outlined the latest methods of medical image segmentation using deep learning by covering the literature related to network structure and model training techniques [7]. Karimi et al. reviewed the literature on techniques for processing label noise in deep learning-based medical image analysis and evaluated existing methods on three medical imaging datasets for segmentation and classification tasks [8]. Zhou et al. presented an overview of the medical image proposed technologies that integrate multiple image medical image segmentation methods [9]. Goceri discussed fully supervised, weak-supervised, and transfer learning techniques for training deep neural networks for medical image segmentation [10]. He also discussed existing methods to solve the problems of lack of data and class imbalance. Zhang et al. reviewed the methods for solving small sample problems in medical image analysis and divided the literature into five

categories, including interpretation, weak supervision, storage, transfer learning, and active learning technology [11]. Tajbakhsh et al. proposed a review of the relevant literature to deal with scarce annotations and weak annotations [12].

2.2. Graph Neural Network. Graph neural network includes graph convolution network (GCN) [13, 14], graph attention network (GAT) [15], graph autoencoders [16–18], and graph generation network [19–21]. Graph convolutional networks extend convolution operations from traditional data (such as images) to graph data. The core idea is to learn a functional map. Attention mechanisms are now widely used in sequence-based tasks and have the advantage of amplifying the impact of the most important parts of the data. This feature has proven useful for many tasks, such as machine translation and natural language understanding. Nowadays, the number of models incorporating attention mechanism continues to increase, and graph neural network also benefits from this. It uses attention in the process of aggregation to integrate the output of multiple models. Graph autoencoder is a kind of graph embedding method, whose purpose is to use neural network structure to represent the vertices of a graph as low-dimensional vectors. A typical solution is to obtain node embedding using MLP as encoders, in which the decoder reconstructs the neighborhood statistics of the node. The goal of a graph generation network is to generate new graphs given a set of observed graphs. Many of the ways in which graphs generate networks are domain specific.

3. A Segmentation Algorithm of Image Semantic Sequence Data Based on Graph Convolution Network

3.1. Graph Convolution Network Used to Construct Image Search Process. When using graph convolution network to construct image search process, a chain structure framework is constructed to continuously update graph convolution network. The chain structure is shown in Figure 1.

Using the chain frame structure shown in Figure 1, a subnetwork structure is obtained in the search space through a controller RNN, and then the network structure is trained on the dataset, and the accuracy R is tested on the verification set. The accuracy R is sent back to the RNN for the next step adjustment, and the network structure is continuously optimized. This process is repeated until the stable convergence result is obtained, and the neural network structure search is completed in the whole process [22]. In this version of neural network structure search, RNN only generates information about the size, number, and sliding step size of convolution kernel, not about the connection mode of network. The parameter θ in the controller is the optimized variable of RNN, which is passed to θ through R , so as to maximize the expectation of this accuracy. Here, the calculation method is as follows:

$$J(\theta) = \frac{E_{P(a_1: T_\theta)}}{[R]}, \quad (1)$$

where $[R]$ is the convolution sum of graph convolution, a_1 is the optimized variable, T_θ is the action period of graph convolution network, $p(a_1: T_\theta)$ is the sliding step length of convolution sum, and E is the expectation function. However, after each update, the accuracy rate is a discrete variable, which is not continuous, so it cannot be differentiated like conventional CNN. Therefore, the author uses the first-order approximation to train the network. The approximate expression of this method has a more convenient solution mode:

$$\nabla J(\theta) = \sum_{k=1}^m \sum_{t=1}^T \nabla_{\theta} \log p(a_1: T_\theta), \quad (2)$$

where m is the number of different structures in a training batch, T is the number of super parameters, ∇_{θ} is the dataset training process, and the meaning of other parameters remains unchanged. For operations involving parameter nonlinearity, such as using sigmoid to calculate input data, through the above calculation process, we can see that the RNN and reinforcement learning structure are used to make the final network give a serialized neural network structure. After splitting the programmed structure into multiple cells, each cell is defined as the directed acyclic edge of N nodes. Each node is a network layer, denoted as $x^{(i)}$, and the edge between nodes is represented as $o^{(i,j)}$. In this case, the operation from node i to new feature in the network can be expressed as follows:

$$x^{(i)} = \sum_{i < j} o^{(i,j)}(x^{(i)}). \quad (3)$$

In the above feature processing process, a DAG operation similar to convolution is obtained. The discretization structure is the same, and there is still no continuous value and differential expression. Therefore, the above relationship needs to be remapped to other operations to complete the continuity.

Assuming that φ is a subsidiary set of operations $o^{(i,j)}$, the edge weights and operation vectors of the neural network are transformed into a continuous variable α , and the discretization process can be expressed as follows:

$$o^{-(i,j)}(x) = \sum_{\alpha \in \varphi} \frac{\exp(\alpha_0^{(i,j)})}{\sum_{\alpha \in \varphi} \exp(\alpha_0)}. \quad (4)$$

In formula (4), α_0 represents the initial variable of graph convolution network, and the meaning of other parameters remains unchanged. If the operation and weight are assigned to a number at the same time, the essence of this number is a network structure [23]. Discretization is that nodes and weights exist separately. If continuous operations put these two values on the same number, then the structure that integrates these two points must be a network. With this network, it can really move towards the goal. By deriving the continuous numerical value, it can find the optimal value based on this result, that is, the current optimal structure. The gradient descent method is used to solve the above networks to get the following:

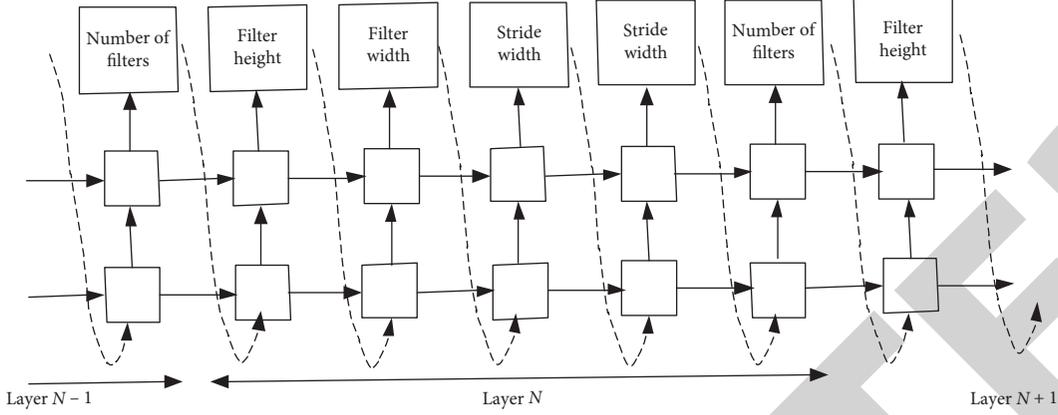


FIGURE 1: The constructed chain frame structure.

$$\nabla_{\alpha} = \frac{w - \varepsilon \nabla_w(w, \alpha)}{L(w^*)}, \quad (5)$$

where w is the gradient value of the network, w^* is the gradient optimal solution, ε is the structural parameter, and L is the continuous function. In the above process, the edge standardization algorithm is introduced to weaken the edge weight, and an additional set of edge selection super parameters is added. Thanks to the partial join strategy, the algorithm can increase the batch size. Selecting $1/k$ channel can reduce memory by k times [24]. It can increase the batch size by k times and finally accelerate the whole network by k times, which obviously improve the training speed. In the specific implementation level, the algorithm proposes to use mask to directly set the channels that are directly transmitted to the output part. In this way, formula (4) can be rewritten as follows:

$$f_{i,j}^{\text{PC}}(x^{(i)}, S_{i,j}) = \sum_{\sigma \in \varphi} \frac{\exp(\alpha_0^{(i,j)})}{\sum_{\sigma \in \varphi} \exp(\alpha_0)} + (1 - S_{i,j}), \quad (6)$$

where $S_{i,j}$ is the mask tag, so the last part of the formula is 0. At the same time, it is obvious that $f_{i,j}^{\text{PC}}(x^{(i)}, S_{i,j})$ of the formula only takes part in the calculation of the previous $1/k$ and then increases the speed by k times directly. However, there are still some problems in practice. Random sampling will make the network results unstable, and there will be a large range of shocks; that is, the results are good and bad. On this point, there are a lot of problems in practice. If it is analyzed in theory, readers can understand that under the independent and identically distributed data, there will be periodic oscillation. If the sampling becomes more, the instability will be reduced. This point corresponds to the fact that in information theory, entropy will increase when the information is reduced, and this entropy corresponds to a large shock in the network [25]. This situation is not conducive to the network in a larger local search for the optimal solution but will fall into some nonoptimal solution. Therefore, this paper proposes its own solution, which is to add an edge regularization to reduce the randomness. It has been stated before that the edge weight and the operation on the node are vectorized into a numerical sequence, that is, a

At this time, if it wants to add the regularization of the edge, it needs to display the weight of the edge. In PC-DARTS, the redefined edge (i, j) of the display is expressed as follows:

$$x_j^{\text{PC}} = \sum_{i < j} \frac{o^{(i,j)}(x^{(i)})}{\exp(\alpha_0^{(i,j)})}. \quad (7)$$

In the above calculation formula, the meaning of each parameter remains unchanged. After such regularization, the edge weights will automatically form a search process. On the basis of the above search process, the semantic sequence data will be extracted to form the basis for the construction of the segmentation algorithm.

3.2. Extraction of Semantic Sequence Data. In the above formed search process, the image obtained from the search is randomly selected as the cascade feature extraction target, the number of Gabor filter transform features is set as 64 dimensions, the frequency of dimension is set as four frequencies, and the four frequencies constitute eight directions [26]. The direction value F is set as follows:

$$F = \left(0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{4\pi}{8}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\right). \quad (8)$$

Under the above direction value, the characteristic value of the direction is extracted, respectively, and the image can be expressed by setting the processed image of the direction value as $I(x, y)$ and the complex characteristic value of images as follows:

$$g(x, y) = \exp\left(-\frac{x^2 + \gamma^2 y^2}{2\sigma^2}\right) \left(2\pi \frac{x}{\lambda} + \psi\right), \quad (9)$$

where λ is the wavelength of sine function, σ is the standard deviation of Gaussian function, γ is the aspect ratio of space, and ψ is the phase shift. The changes of image complex features in the extraction period are shown in Figure 2.

In the period variation range shown in Figure 2, the numerical period of the fixed cascade feature is between 30 and 100. When the complex feature is cascade, the real part and imaginary part of the complex feature are filtered [27]. The filtered complex feature is as follows:

$$\begin{cases} S = \cos i \left(2\pi \frac{x}{\lambda} + \psi \right), \\ X = \sin i \left(2\pi \frac{x}{\lambda} + \psi \right), \end{cases} \quad (10)$$

where i is the direction of the feature parameter, and the meaning of other parameters remains unchanged. The real part and the imaginary part of the complex feature of an image are regarded as cascading objects [28, 29]. The cascade object is represented as follows:

$$tx = \begin{bmatrix} SX_1(1) & SX_1(2) & \cdots & SX_1(64) \\ SX_2(1) & SX_2(2) & \cdots & SX_2(64) \\ \vdots & \vdots & \ddots & \vdots \\ SX_n(1) & SX_n(2) & \cdots & SX_n(64) \end{bmatrix}. \quad (11)$$

In formula (11), tx represents the cascade sample set of complex features and n represents the cascade feature dimension. In the cascade set constructed by the above calculation formula, the cascade process as shown in Figure 3 is constructed. Feature cascade process is shown in Figure 3.

In the cascade process as shown in Figure 3, firstly, the color feature data in the image are extracted, and the RGB color space method is used to represent the color components in the image in the cascade process [30]. The calculation formula can be expressed as follows:

$$\begin{cases} R = \int_{\lambda} E(\lambda) S_R(\lambda) d\lambda \\ G = \int_{\lambda} E(\lambda) S_G(\lambda) d\lambda, \\ B = \int_{\lambda} E(\lambda) S_B(\lambda) d\lambda \end{cases} \quad (12)$$

where S_R , S_G , and S_B represent the filters of different colors, $E(\lambda)$ represents the light ray entering the filter, and λ represents the wavelength of the light ray. The extraction of semantic sequence is a dynamic process, and the color component forms a certain image color conversion in the process of semantic sequence transformation [31]. The color conversion process can be expressed as follows:

$$\begin{cases} H = \arctan \left(\frac{\sqrt{3}(G-B)}{(R-G) + (R-B)} \right), \\ I = \frac{R+G+B}{3}, \\ S = 1 - \frac{\min(R, G, B)}{I}, \end{cases} \quad (13)$$

where H , I , and S , respectively, represent the color space after color conversion, and the meaning of other parameters remains unchanged. Given formula (13), an interval $[0, 1]$ forms a region to be extracted, and the whole image region to be extracted can be divided into three fan-shaped regions [32]. The area of the region can be expressed as follows:

$$A = r \frac{S \cos H}{2 \cos(60^\circ - H)}, \quad (14)$$

where A is the area of the region, r is the radius of the sector, and the meaning of other parameters remains unchanged. A segmentation algorithm is constructed by using the fan area formed by the cascade features as the construction object of the segmentation algorithm.

3.3. Completing the Construction of Segmentation Algorithm.

Taking the above process as the basis of algorithm construction, when accumulating qualified data points [33], it is assumed that the fan area formed by cascade is R with a size of $N \times N$, R^{ij} is the corresponding block search subimage of reconstruction, and the pixel gray values of reconstruction image and original image are expressed as $R^{ij}(m, n)$ and $T(m, n)$. The average value of pixel gray values of volume image is calculated.

$$\left\{ \bar{R}^{ij}(i, j) = \frac{1}{M^2} \sum_{m=1}^M \sum_{n=1}^N R^{ij}(m, n) \bar{T}(i, j) = \frac{1}{M^2} \sum_{m=1}^M \sum_{n=1}^N T(m, n). \right. \quad (15)$$

In formula (15), (m, n) is any imaging pixel data point, and the error of formula (7) is calculated.

$$\kappa(i, j, m_k, n_k) = |R^{ij}(m, n) - \bar{R}^{ij}(i, j) - T(m_k, n_k) - \bar{T}(i, j)|. \quad (16)$$

The constant threshold (m, n) calculated by formula (16) is selected, and a pixel data point is randomly selected from $R^{ij}(m, n)$. The data point error [34] calculated by formulae (15) and (16) is used, to continue to accumulate until the accumulated value is greater than the set constant threshold Q_C , then the accumulation processing is stopped, and the times of current accumulated r is recorded. At this time, the accumulated degree surface is expressed as follows:

$$L(i, j) = \{r \mid \min[\kappa(i, j, m_k, n_k) \geq Q_C]\}. \quad (17)$$

$L(i, j)$ calculated in formula (17) is the matching point, and the maximum point (i, j) in the point $L(i, j)$ with the least accumulated error times is selected as the final reconstructed data point. Within the scope of the surface, the reconstructed data point area is shown in Figure 4.

In the data region points reconstructed in Figure 4, the reconstruction degree of qualified data points is measured by the cross-correlation value. Assuming that the size of the image G_r before segmentation is $M_r \times N_r$, and the size of the semantic sequence image containing the maximum value of segmentation data (i, j) is $M_s \times N_s$, the measure of normalization of segmentation semantics is as follows:

$$\chi_g(u, v) = \frac{\sum_{i=1}^{M_s} \sum_{j=1}^{N_s} [G_r(i+u, j+v) - \bar{G}_r(u, v)]}{\left\{ \sum_{i=1}^{M_x} \sum_{j=1}^{N_x} [G_r(i+u, j+v) - \bar{G}_r(u, v)]^2 \right\}}. \quad (18)$$

The measurement calculated by formula (18) uses the deaveraging operation to eliminate the deviation of the

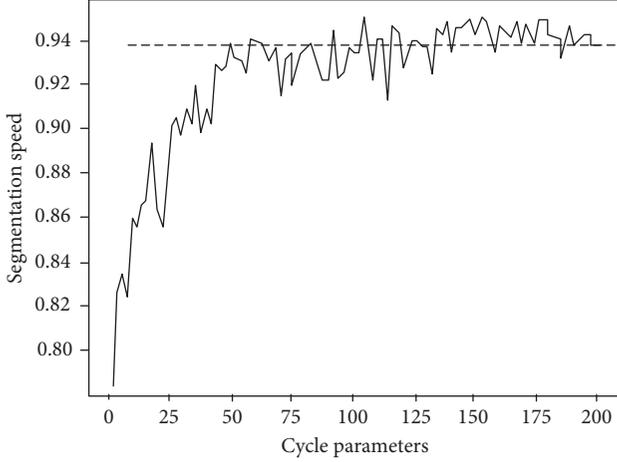


FIGURE 2: Variation of characteristic value in a period.

amount of data in the algorithm and improve the anti-interference ability of the algorithm in dealing with unqualified data [35].

Using the reconstructed data points calculated from the first-order derivative Bezier curve fitting, it is assumed that there are $n + 1$ data nodes in the reconstruction and the node position can be expressed as follows:

$$T_k (k = 0, 1, \dots, n). \quad (19)$$

In this case, the Bezier curve reconstructed by n times can be expressed as follows:

$$T(i) = T_0(1-i)^n + \binom{n}{1}T_1(1-i)^{n-1}t + \dots + T_n i^n, i \in [0, 1]. \quad (20)$$

By simplifying formula (12), the final calculation is as follows:

$$\{ T(i) = \sum_{k=0}^n T_k B_{k,n}(i), i \in [0, 1] \} B_{k,n}(i) = C_n^k (1-i)^{n-k} i^k. \quad (21)$$

In formula (21), i is the parameter, T_0 is the starting point, T_i is the control point, T_n is the end point, and $B_{k,n}(i)$ is the Bezier basis function, which consisted of the starting point T_0 to the end point T_n . Connecting different control points from the starting point, Bezier curve control polygon can be connected and fit. In order to overcome the edge ambiguity of the tensor after curve fitting [36], the edge ambiguity of the linear tensor is calculated by simultaneous formulae (20) and (21). After transformation, the representation parameters of the reconstruction algorithm expression are obtained, which are integrated into the matrix field representing the smoothing strength and direction of the curve. The calculation formula is as follows:

$$V(x, y) = [u(x, y), v(x, y)]. \quad (22)$$

In formula (22), $u(x, y)$ is the matrix field of smoothing force and $v(x, y)$ is the matrix field of smoothing direction [37]. In order to keep the curve of the reconstructed

algorithm after fitting smooth and enhance the data processing ability of the fitting algorithm [38], formula (18) is minimized to obtain the edge processing formula at this time.

$$\rho = \int \int (\mu(u_x^2 + u_y^2 + v_x^2 + v_y^2)) dx dy. \quad (23)$$

In formula (23), μ is the weight factor of gradient amplitude. The final calculation expression of the reconstruction algorithm is as follows:

$$\begin{cases} \mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0, \\ \mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0, \end{cases} \quad (24)$$

where $\nabla^2 u$ is Laplacian operator and $f(x, y)$ is gradient amplitude factor in semantic sequence. The gradient amplitude factor calculated by continuous diffusion ensures the existence of amplitude value in the segmented image semantic sequence data so that the gradient amplitude is not completely rotating field and no scattering field in the diffusion process [39], which enhances the segmentation accuracy of the algorithm. Based on the above processing, the research on the segmentation algorithm of image semantic sequence data based on graph convolution network is completed.

4. Simulation Experiment

4.1. Experimental Preparation. The dataset used in the experiment is ShapeNet part. This dataset contains 16 categories of 16881 image models. These image models are segmented and prelabeled into 50 different categories of components, and each independent model is labeled with no more than 6 components. Different components in each model are labeled with different semantic information, that is, different category information labels, for the verification of point cloud segmentation experiments basis. The ShapeNet part datasets of 16881 graph convolution networks are divided into three different data subsets: training set, cross validation set, and test set. There are 12137 data models in 16 categories in training set, 1870 data models in 16 categories in cross validation set, and 2874 data models in 16 categories in test set [40]. 2048 data points of each graph convolution network are uniformly sampled to generate a point cloud dataset of a single object, which is used as the original input of the separation model, and each data point has a unique prelabeled information label, which is used as the test basis of the experiment on the segmentation algorithm of image semantic sequence data [41]. The performance evaluation index of the model used in the experiment is the intersection and union ratio commonly used in semantic segmentation or target detection tasks, as shown in Figure 5.

Under the semantic sequence segmentation shown in Figure 5, controlling the ratio of intersection and union of "Predicted border" and "Real border" calculated by IoU is a performance quantitative index at pixel level. The larger the value of IoU is, the higher the overlap between predicted

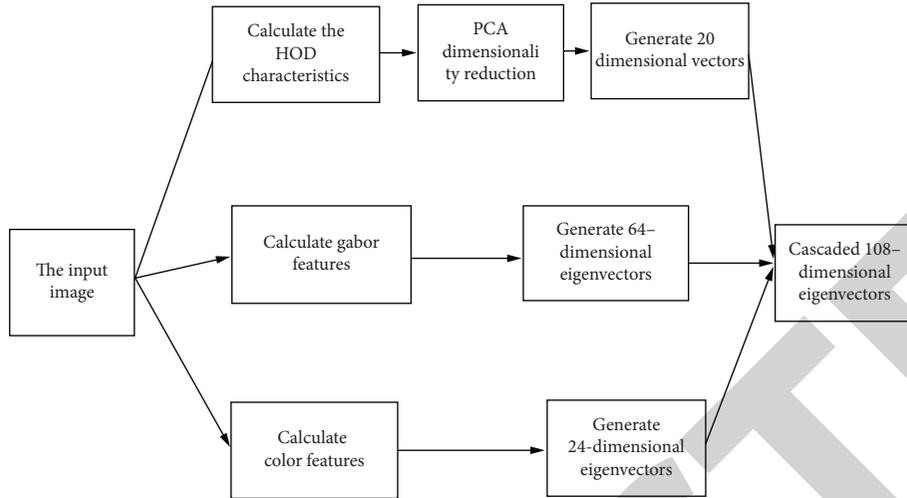


FIGURE 3: Feature cascade process.

value and real value is and the more accurate the actual segmentation result is. With the help of deep learning framework, a NVIDIA GTX 1080Ti GPU is adopted to train and test the model, and the results of independent segmentation of experimental dataset are obtained. The set segmentation speed is as shown in Figure 6.

Under the segmentation speed set in Figure 6, the experimental datasets are kept running normally under different algorithms. The data to be segmented in the image semantics of the datasets are collected, and the collected data are shown in Table 1.

In the experimental dataset shown in Table 1, the segmentation algorithm in references [4, 6, 10, 12], which are named as Seg-MF, Seg-U-Net, Seg-U-Like-Net, Seg-GMM, respectively, and the segmentation algorithm designed in this paper, Seg-GCN, are used to carry out experiments, and the performance of the three segmentation algorithms are compared.

4.2. Results and Analysis. Based on the above experimental preparation, the unprocessed data in each data group of the experiment are defined as the residual redundant data, and the residual redundant data of the three algorithms are taken as the comparison index. The final residual redundant data results of the three data segmentation algorithms are shown in Figure 7.

From the results of residual redundant data shown in Figure 7, taking the sample data collected in the preparation stage of the experiment as the experimental object, the three segmentation algorithms show different amounts of residual data. According to the above results, when the number of redundant data is 200, Seg-MF and Seg-U-Net have more residual redundant data, with the value of about 25. The redundant data segmentation algorithm designed in this paper has the least number of redundant data, and the number of redundant data is about 10. With the increasing amount of experimental redundant data, Seg-MF, Seg-U-Net, Seg-U-Like-Net, and Seg-GMM have more redundant data to be segmented, while the redundant data

segmentation algorithm designed in this paper has less redundant data, which can basically segment all the collected sample data without corresponding data redundancy.

In the above experimental environment, it takes the beginning of the algorithm as the time record point and the complete image of the computer running the algorithm as the time statistics cut-off point, to count the execution time results of the three segmentation algorithms. The results are shown in Figure 8.

From the above experimental results, it can be seen that the three segmentation algorithms show different execution time in actual application. The Seg-MF has the longest actual execution time, with an average execution time of 14 ms. The Seg-U-Net has an execution time of 10 ms, with a shorter actual execution time. The final execution time of Seg-GCN is 2 ms. Compared with other two segmentation algorithms in the reference, the algorithm in this paper has the shortest execution time and the best running timeliness.

Keeping the above experimental environment unchanged and defining the segmentation accuracy of the segmentation algorithm, the calculation formula can be expressed as follows:

$$P = \frac{1 - P_1}{R}, \quad (25)$$

where P is the segmentation degree of the algorithm, P_1 is the semantic distance of the image, and R is the confidence degree of the semantic sequence data. Under the control of the above calculation formula, the final segmentation accuracy results of the three data segmentation algorithms are shown in Figure 9.

It can be seen from the experimental results shown in Figure 9 that, according to the performance of different segmentation algorithms, after setting the semantic distance that can be run, under the control of three segmentation algorithms and defining six data segmentation points, the actual maximum segmentation accuracy of Seg-MF is about 0.6, and the accuracy value is the smallest. The maximum segmentation accuracy of Seg-U-Net is about 0.8, and the

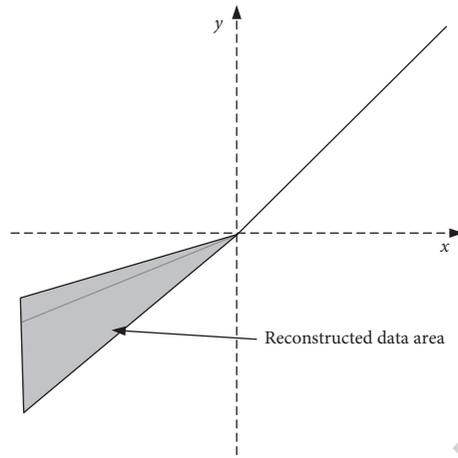


FIGURE 4: Reconstruction of data point area.

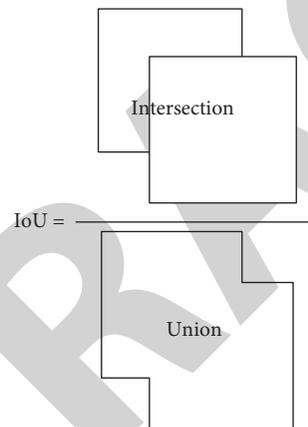


FIGURE 5: Semantic sequence segmentation process.

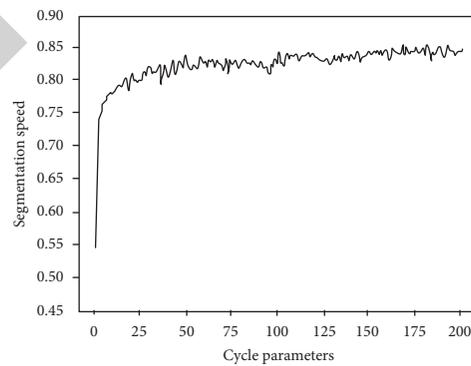


FIGURE 6: Set segmentation speed.

TABLE 1: Collected image semantic sequence dataset.

Name	IoU value	Ours	PCNN	KD-Net
ARFO	82.3	81.7	76.4	90.3
BAG	82.4	85.4	74.3	90.4
CAP	84.0	78.5	71.2	91.1
CAR	81.2	83.1	79.9	89.9
CHAIR	82.7	87.6	79.5	89.7
EAR PHONE	82.5	87.1	75.2	91.2
GUITAR	83.4	85.6	78.9	91.1
KNIFE	81.2	82.8	73.6	91.7
LAMP	82.6	80.8	79.3	89.5
LAP	82.8	85.4	77.1	90.4
MOTOR	82.4	85.1	76.5	90.5
MUG	82.6	85.6	80.6	89.9
PISTOL	82.1	86.8	76.5	91.3
TABLE	82.3	81.3	79.1	90.2

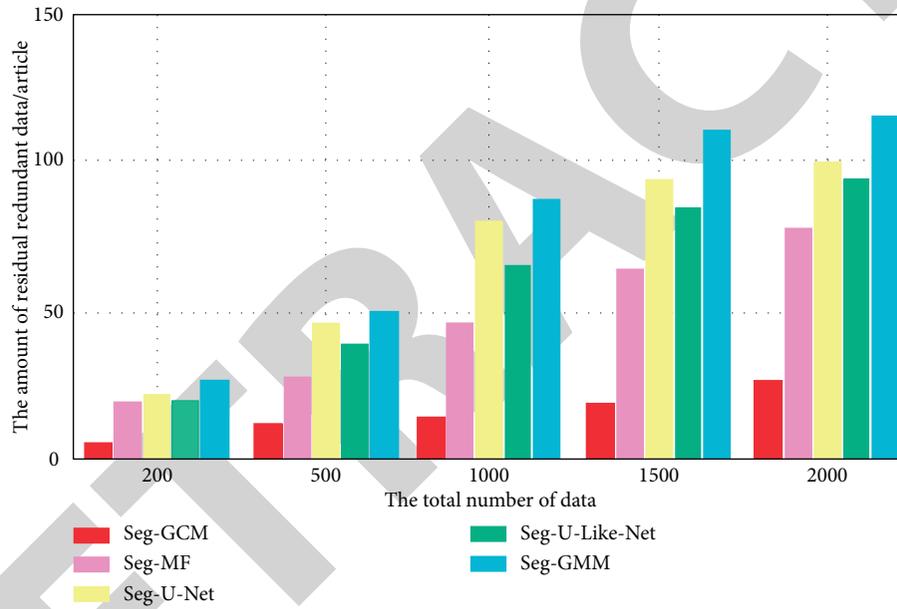


FIGURE 7: Residual redundant data results of five data segmentation algorithms.

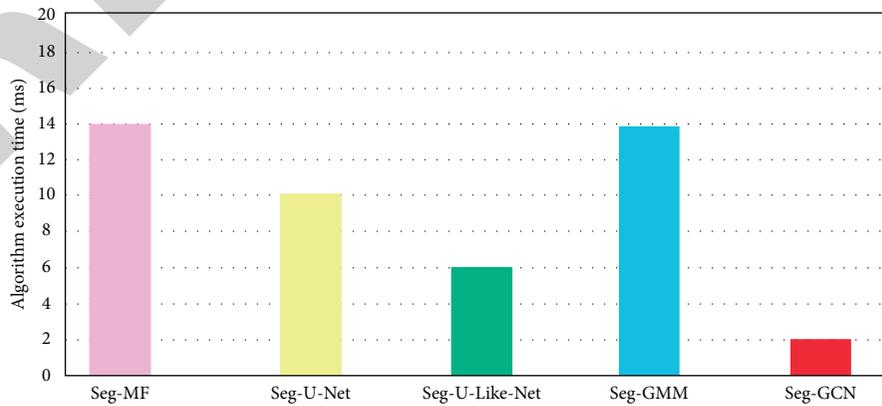


FIGURE 8: Execution time results of five segmentation algorithms.

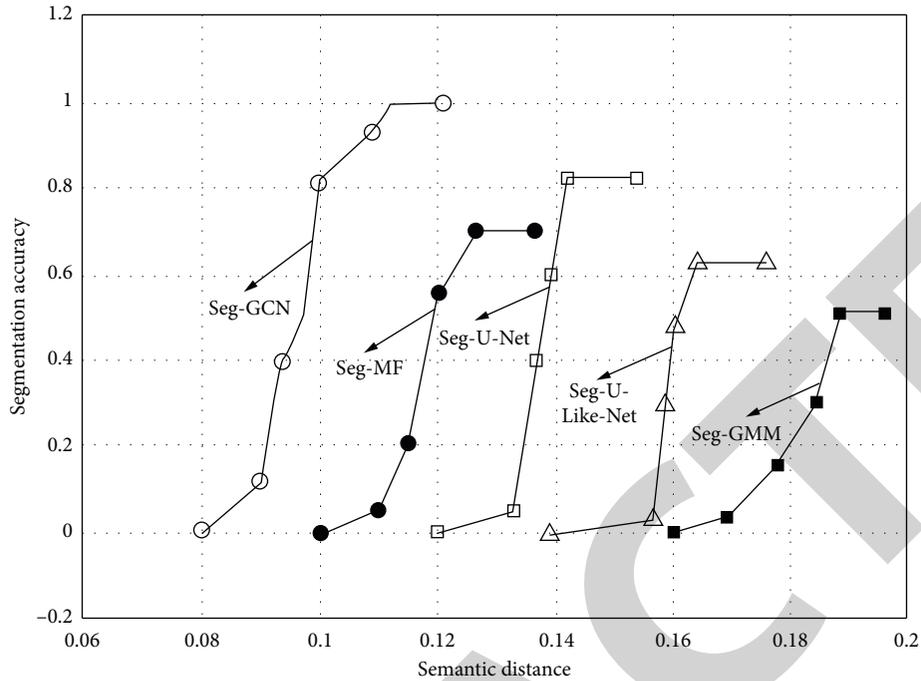


FIGURE 9: Segmentation accuracy results of five segmentation algorithms.

segmentation accuracy is high. Compared with the two kinds of segmentation algorithms in the reference, the segmentation accuracy of the segmentation algorithm designed in this paper is the highest, which is suitable for practical use.

5. Conclusion

This paper studies the image semantic segmentation algorithm based on deep learning. The current framework of the algorithm is deep network feature extraction, feature graph up sampling, and pixel Softmax loss sum, which is used as the process to achieve end-to-end semantic segmentation. The segmentation algorithm designed in this paper focuses on improving the depth network to extract effective features, so there are a lot of improvements in resolution sampling and loss function design for segmentation. On the other hand, the content focuses on image semantic segmentation, which can provide a certain research direction for the future research of the sequential data segmentation algorithm.

A limitation to address in the future is to optimize the parameters of the proposed segmentation algorithm and further improve the segmentation performance. Another direction for improvement is to expand the scale of the used experimental data and ameliorate the generalization ability of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This paper was funded by the National Natural Science Foundation of China (project nos. 60572153 and 60972127).

References

- [1] Q.-C. Tian and Y. Meng, "Image semantic segmentation based on convolutional neural network," *Journal of Chinese Computer Systems*, vol. 41, no. 6, pp. 1302–1313, 2020.
- [2] Z. Zhang, W. Fang, and L. Du, "Semantic segmentation of remote sensing image based on encoder-decoder convolutional neural network," *Acta Optica Sinica*, vol. 40, no. 3, pp. 46–55, 2020.
- [3] Y. Guo, Y. Liu, and T. Georgiou, "A review of semantic segmentation using deep neural networks," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 87–93, 2018.
- [4] Y. Hu, Z. Chen, and W. Lin, "RGB-D semantic segmentation: a review," in *Proceedings of the 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6, IEEE, San Diego, CA, USA, July 2018.
- [5] T. L. T. Le, N. Thome, and S. Bernard, "Multitask classification and segmentation for cancer diagnosis in mammography," 2019, <https://arxiv.org/abs/1909.05397>.
- [6] E. Goceri and N. Goceri, "Deep learning in medical image analysis: recent advances and future trends," 2017.
- [7] M. H. Hesamian, W. Jia, and X. He, "Deep learning techniques for medical image segmentation: achievements and

- challenges,” *Journal of Digital Imaging*, vol. 32, no. 4, pp. 582–596, 2019.
- [8] D. Karimi, H. Dou, and S. K. Warfield, “Deep learning with noisy labels: exploring techniques and remedies in medical image analysis,” *Medical Image Analysis*, vol. 65, Article ID 101759, 2020.
- [9] T. Zhou, S. Ruan, and S. Canu, “A review: deep learning for medical image segmentation using multi-modality fusion,” *Array*, vol. 3, Article ID 100004, 2019.
- [10] E. Goceri, “Challenges and recent solutions for image segmentation in the era of deep learning,” in *Proceedings of the 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6, IEEE, Istanbul, Turkey, November 2019.
- [11] P. Zhang, Y. Zhong, and Y. Deng, “A survey on deep learning of small sample in biomedical image analysis,” 2019, <https://arxiv.org/abs/1908.00473>.
- [12] N. Tajbakhsh, L. Jeyaseelan, and Q. Li, “Embracing imperfect datasets: a review of deep learning solutions for medical image segmentation,” *Medical Image Analysis*, vol. 63, Article ID 101693, 2020.
- [13] Z. Li, Q. Chen, and V. Koltun, “Combinatorial optimization with graph convolutional networks and guided tree search,” in *Proceedings of the Neural Information Processing Systems*, pp. 536–545, Montreal, Canada, December 2018.
- [14] T. H. Nguyen and R. Grishman, “Graph convolutional networks with argument-aware pooling for event detection,” in *Proceedings of the AAAI*, pp. 5900–5907, New Orleans, LA, USA, February 2018.
- [15] X. Wang, H. Ji, C. Shi et al., “Heterogeneous graph attention network,” in *Proceedings of the WWW*, Minneapolis, MN, USA, June 2019.
- [16] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, “Adversarially regularized graph autoencoder for graph embedding,” in *Proceedings of the IJCAI*, pp. 2609–2615, Stockholm, Sweden, July 2018.
- [17] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, “Deep recursive network embedding with regular equivalence,” in *Proceedings of the KDD*, pp. 2357–2366, ACM, London, UK, August 2018.
- [18] W. Yu, C. Zheng, W. Cheng et al., “Learning deep network representations with adversarially regularized autoencoders,” in *Proceedings of the AAAI*, pp. 2663–2671, ACM, New Orleans, LA, USA, February 2018.
- [19] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, “Scene graph generation by iterative message passing,” in *Proceedings of the CVPR*, Honolulu, Hawaii, July 2017.
- [20] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, “Graph r-cnn for scene graph generation,” in *Proceedings of the ECCV*, pp. 690–706, Springer, Munich, Germany, September 2018.
- [21] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, “Factorizable net: an efficient subgraph-based framework for scene graph generation,” in *Proceedings of the ECCV*, pp. 346–363, Springer, Munich, Germany, September 2018.
- [22] Cai Shuo, H. Hu, and W. Wang, “Semantic segmentation of high-resolution remote sensing image based on deep convolutional network,” *Journal of Signal Processing*, vol. 35, no. 12, pp. 2010–2016, 2019.
- [23] N. Xiao and P. Zhao, “Segmentation algorithm of annual ring image based on U-Net convolution network,” *Chinese Journal of Ecology*, vol. 38, no. 5, pp. 1580–1588, 2019.
- [24] J. Yao, J. Wu, and Y. Yang, “Segmentation in multi-spectral remote sensing images using the fully convolutional neural network,” *Journal of Image and Graphics*, vol. 25, no. 1, pp. 180–192, 2020.
- [25] P. Cao, Q. Sheng, and Q. Pan, “Combining sequence learning and u-like-net for hippocampus segmentation,” *Journal of Computer-Aided Design & Computer Graphics*, vol. 31, no. 8, pp. 1382–1390, 2019.
- [26] X. Cheng, L. Zhao, and Q. Hu, “Real-time semantic segmentation based on dilated convolution smoothing and lightweight up-sampling,” *Laser & Optoelectronics Progress*, vol. 57, no. 2, pp. 185–192, 2020.
- [27] H. Yang and F. Wang, “Meteorological radar noise image semantic segmentation method based on deep convolutional neural network,” *Journal of Electronics & Information Technology*, vol. 41, no. 10, pp. 2373–2381, 2019.
- [28] D.-W. Zhou, T. Jin-Yue, and Ma Lu-yao, “Lightweight image semantic segmentation based on multi-level feature cascaded network,” *Journal of Zhejiang University(Engineering Science)*, vol. 54, no. 8, pp. 1516–1524, 2020.
- [29] P.-C. Zhou, S.-R. Gong, and S. Zhong, “Image semantic segmentation based on deep feature fusion,” *Computer Science*, vol. 47, no. 2, pp. 126–134, 2020.
- [30] C. Xiao, Li Yu, and H. Zhang, “Semantic segmentation of remote sensing image based on deep fusion networks and conditional random field,” *Journal of Remote Sensing*, vol. 24, no. 3, pp. 254–264, 2020.
- [31] Y.-X. Liu, B. Zhang, and B. Wang, “Semi-supervised semantic segmentation based on Generative Adversarial Networks for remote sensing images,” *Journal of Infrared and Millimeter Waves*, vol. 39, no. 4, pp. 473–482, 2020.
- [32] Z.-Y. Wang, N. Xian-yang, and Z.-D. Shang, “Autonomous driving semantic segmentation with convolution neural networks,” *Optics and Precision Engineering*, vol. 27, no. 11, pp. 2429–2438, 2019.
- [33] A. Zhang, L. Liu, and X. Zhang, “Multi-feature 3D road point cloud semantic segmentation method based on convolutional neural network,” *Chinese Journal of Lasers*, vol. 47, no. 04, pp. 269–277, 2020.
- [34] Y.-A. Ruxianguli-Abudurexiti and G. Sun, “Sensitive image detection method based on deep neural network and sparse semantics,” *Application Research of Computers*, vol. 47, no. 04, pp. 269–277, 2020.
- [35] S. Ji and W. Shiqing, “Building extraction via convolutional neural networks from an open remote sensing building dataset,” *Acta Geodaetica et Cartographica Sinica*, vol. 48, no. 4, pp. 448–459, 2019.
- [36] D. Ma, S. He, and C. Yang, “Semantic segmentation based on convolutional neural networks with feature fusion,” *Computer Engineering and Applications*, vol. 56, no. 10, pp. 193–198, 2020.
- [37] B. Rui-jun, L. Zhong, and Q.-Y. Zhang, “An improved deep convolution neural network image denoising algorithm,” *Science Technology and Engineering*, vol. 19, no. 36, pp. 247–252, 2019.
- [38] W. Zhi-qiang, Bi Hai-xia, and X. Liu, “A graph-based semi-supervised polsar image classification method using deep convolutional neural networks,” *Acta Electronica Sinica*, vol. 48, no. 1, pp. 66–74, 2020.
- [39] Li Zhu, J. Zhao, and Y. Fu, “Deep learning algorithm for the segmentation of the interested region of an infrared thermal image,” *Journal of Xidian University (Natural Science)*, vol. 46, no. 4, pp. 107–121, 2019.
- [40] J. Cao and F. Wen, “The impact of the cross-shareholding network on extreme price movements: evidence from China,” *Journal of Risk*, vol. 22, no. 2, pp. 79–102, 2019.
- [41] F. Wen, F. Min, and Y. J. Zhang, “Crude oil price shocks, monetary policy, and China’s economy,” *International Journal of Finance & Economics*, vol. 24, no. 2, pp. 812–827, 2019.