

Article

Pattern Recognition and Segmentation of Administrative Boundaries Using a One-Dimensional Convolutional Neural Network and Grid Shape Context Descriptor

Min Yang ¹ , Haoran Huang ¹, Yiqi Zhang ¹ and Xiongfeng Yan ^{2,*} 

¹ School of Resource and Environmental Sciences, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

² College of Surveying and Geo-Informatics, Tongji University, 1239 Siping Road, Shanghai 200092, China

* Correspondence: xiongfengyan@tongji.edu.cn; Tel.: +86-021-65981436

Abstract: Recognizing morphological patterns in lines and segmenting them into homogeneous segments is critical for line generalization and other applications. Due to the excessive dependence on handcrafted features in existing methods and their insufficient consideration of contextual information, we propose a novel pattern recognition and segmentation method for lines, based on deep learning and shape context descriptors. In this method, a line is divided into a series of consecutive linear units of equal length, termed lixels. A grid shape context descriptor (GSCD) was designed to extract the contextual features for each lixel. A one-dimensional convolutional neural network (1D-U-Net) was constructed to classify the pattern type of each lixel, and adjacent lixels with the same pattern types were fused to obtain segmentation results. The proposed method was applied to administrative boundaries, which were segmented into components with three different patterns. The experiments showed that the lixel classification accuracy of the 1D-U-Net reached 90.42%. The consistency ratio was 92.41%, when compared with the manual segmentation results, which was higher than either of the two existing machine learning-based segmentation methods.

Keywords: line segmentation; pattern recognition; one-dimensional convolutional neural network; grid shape context descriptor



Citation: Yang, M.; Huang, H.; Zhang, Y.; Yan, X. Pattern Recognition and Segmentation of Administrative Boundaries Using a One-Dimensional Convolutional Neural Network and Grid Shape Context Descriptor. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 461. <https://doi.org/10.3390/ijgi11090461>

Academic Editors: Maria Antonia Brovelli and Wolfgang Kainz

Received: 16 June 2022

Accepted: 25 August 2022

Published: 28 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In map space, various geographical entities are modeled as points, lines, polygons, or other element types. Lines are one of the most abundant elements, representing rivers, roads, administrative boundaries, and coastlines. Due to the heterogeneity inherent in geographical environments, lines typically differ in terms of their geometric properties and spatial structures. In particular, an individual line may have different shape pattern characteristics, which reveal the spatial distribution characteristics of the associated geographical entity. During the processing of line generalization, which simplifies the shape of a line by removing unwanted small details, it is important to recognize the morphological patterns of the lines and segment them into homogenous segments [1–7]. A simple example is shown in Figure 1, where a line consisting of two parts with different pattern characteristics has been simplified using two popular algorithms (i.e., the bend-based algorithm [8] and the orthogonality-preserving simplification algorithm [9]). The bend-based algorithm was appropriate for simplifying segment I with a complex hierarchy of bends, however, it generated an unsatisfying result for segment II with orthogonal characters. In contrast, the orthogonality-preserving algorithm performed well to maintain the orthogonality of segment II; however, it distorted the main bend structures of segment I. This example indicates that each line generalization algorithm has its own strengths and weaknesses for differing shape characteristics [10–12].

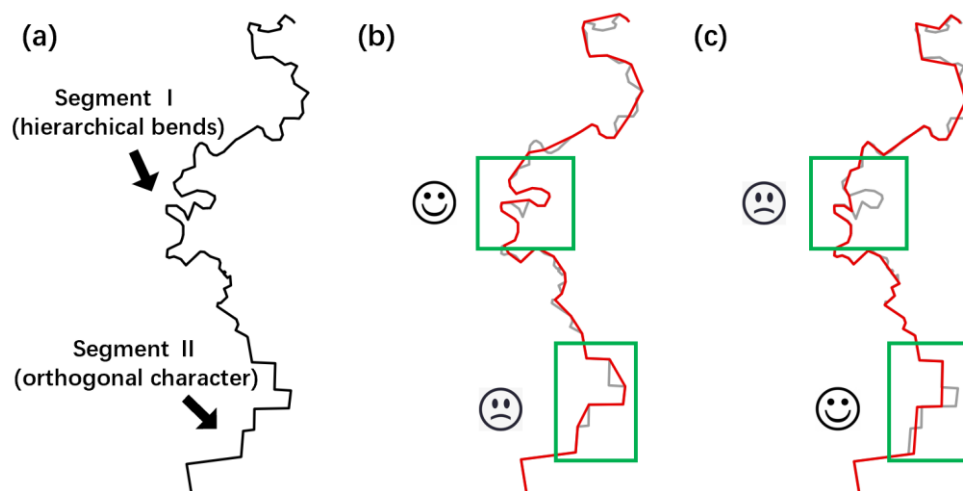


Figure 1. Simplified results (in red) of a line with different shape characteristics using two different algorithms: (a) the original line; (b) result using the bend-based simplification algorithm; (c) result using the orthogonality-preserving simplification algorithm.

To overcome this drawback, one promising solution is to develop hybrid generalization methods which allow each part of the line to be processed using the appropriate algorithm. Specifically, for each input line, the shape characteristics of the line are first recognized and then divided into geometrically homogeneous segments. Next, the appropriate algorithm and parameter settings can be applied to each segment according to its pattern type. Since a large number of line generalization algorithms have been developed, methods for line pattern recognition and segmentation have become a focus of current research. This necessity has been highlighted by researchers during the generalization of various map features, including building outlines [11,13,14], administrative boundaries [12], road lines [15–17], coastlines [18], and land-use boundaries [5].

In the past few decades, some studies have been carried out devoted to line segmentation. Broadly, these methods can be classified into two categories: critical-point- and shape-analysis-based methods. Critical points, such as inflection points [19], are usually detected as break points, thus yielding segmentation results. Additionally, several studies implicitly achieved segmentation by applying point-compression models in which the retained points were used as the critical points [3,20]. Shape-analysis-based methods identify homogeneous segments with geometric measurements. For example, in the approach proposed by Plazanet et al. [4], an input line was recursively divided until all parts became geometrically homogeneous. The pattern type of each segment was then classified based on the shape measurements of the bends. Samsonov and Yakimova [12] proposed several procedures, including filtering, segmentation, and squaring, to decompose a line into homogeneous segments. They identified orthogonal and non-schematic segments based on analysis of the angle and distance measurements.

Intelligent methods have been explored for line segmentation based on shape analysis and geometric measures. For instance, Balboa and López [16,17] attempted to emulate expert human segmentation for road lines via a backpropagation artificial neural network (BANN). A set of geometric features was used to describe the shape characteristics of each road segment. A BANN classification model was then designed to establish mapping between the descriptive features and the pattern types. Finally, the line segmentation results were obtained by applying the classifier to predict the class of each component derived from a window moving along the lines. Liu and Yang [18] developed a similar method for coastline segmentation that processes the geometrical features of a segment using principal component analysis (PCA). They adopted the Bayesian model to construct a classifier to predict the pattern types of segments extracted by a moving window, and output the segmentation results by merging the segments belonging to identical classes.

Although many methods have been developed for the purpose, the effective analysis and segmentation of lines remains challenging. First, it is difficult to comprehensively and objectively reveal the shape characteristics of a line, resulting in a lack of information for distinguishing segments with different shape characteristics. Second, existing methods lack an effective mechanism to consider local contextual information, which limits the performance of pattern recognition and segmentation.

From a visual cognition perspective, the line segmentation problem is similar to the image segmentation problem (i.e., combining units with similar characteristics to form local continuous homogeneous structures). In the image segmentation field, convolutional neural networks (CNNs) have excellent properties, including local perception and multiscale feature characterization, rendering them the most advanced technology for solving this problem [21,22]. Recently, CNNs and their variants have been successfully applied to map data processing, including geographical pattern recognition and classification [23–25], shape representation and classification [26,27], and cartographical generalization [7,28,29]. In this study, we employed CNN technology to construct a method for pattern recognition and segmentation of lines. However, unlike image data with grid-topology structures, each line consisted of a sequence of points with uneven distributions; the distance between two adjacent points was not identical. Thus, the processing units were not stationary. Therefore, we divided the lines into a series of consecutive linear units with equal lengths, also known as *lixels* [30], which serve as the basic units that compose a line. Based on this, each line was organized into a list of *lixels*, sequentially ordered and evenly distributed. In this manner, the line segmentation problem was transformed into a *lixel* classification problem, which could be analyzed and processed using advanced image segmentation techniques, to solve the shortcomings inherent in existing methods that include excessive dependence on handcrafted features and insufficient consideration of contextual information.

We propose a grid shape context descriptor (GSCD) to describe the contextual characteristics of each *lixel* and its neighbors. This descriptor provides a standardized computation for the contextual features with respect to each *lixel*, thereby reducing the subjective impact of manually defined features encountered in previous methods. Subsequently, with reference to the classic architecture [31], a novel one-dimensional convolutional neural network (1D-U-Net) that combines operations such as one-dimensional convolution, pooling, and transposed convolution, was constructed to analyze the extracted contextual features of the *lixels* and assign a pattern type to each. Finally, adjacent *lixels* with the same pattern types were fused to output the final segmentation results. To verify the proposed method, administrative boundaries were chosen as the experimental data. This is because human and natural factors both influence the geometrical morphologies of administrative boundaries, thus presenting a variety of pattern types conducive to effectively testing the proposed method.

The remainder of this paper is organized as follows. Section 2 introduces the experimental datasets and pattern types for the administrative boundaries. Section 3 details the proposed segmentation method using GSCD and 1D-U-Net. Section 4 presents the experimental design and results, as well as a detailed analysis, comparison, and discussion. Finally, Section 5 concludes this study.

2. Experimental Datasets and Shape Patterns

2.1. Experimental Datasets

In this study, two sets of administrative boundary data for southern China extracted from a 1:50,000 land-use database were used as the training and test datasets, respectively (Figure 2). The training dataset contained 102 administrative boundaries, with a total length of 927.781 km and an average length of 9.096 km. The test dataset contained 52 administrative boundaries, with a total length of 783.857 km and an average length of 15.074 km. The boundaries of the two datasets had complex and diversified morphological characteristics, suitable for confirming the effectiveness of the proposed method.

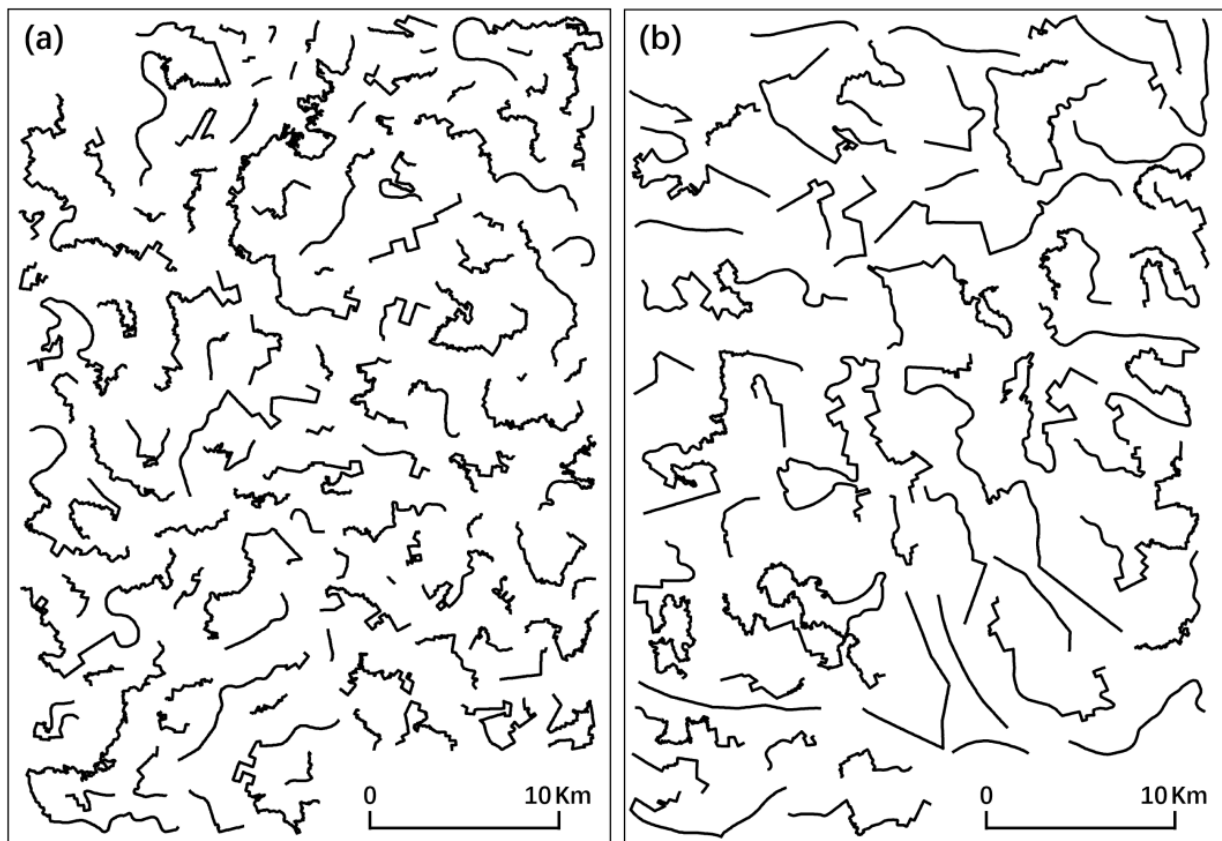





Figure 2. Experimental data: (a) administrative boundaries in the training dataset, and (b) administrative boundaries in the testing dataset.

2.2. Administrative Boundary Shape Pattern Types

Numerous variants and combinations of shape patterns for lines were included in this study. For administrative boundary data, components may derive from natural objects, such as rivers and coastlines, as well as anthropogenic objects, such as roads and built-up areas. Natural objects are usually characterized by a hierarchical bend structure, whereas anthropogenic objects tend to have artificially sharp or right-angled features. Samsonov and Yakimova [12] demonstrated that these differences in line shape patterns can be conceptualized in three aspects: smoothness, schematism, and regularity. The degree of smoothness indicates whether the shape change along a line is smooth (i.e., whether it has a gradually changing tangent direction or contains a large deviation angle). Schematism indicates whether the shape composition of a line is simple or if it contains complex hierarchical curved structures of various sizes. Regularity refers to whether a line shape has repetitive characteristics.

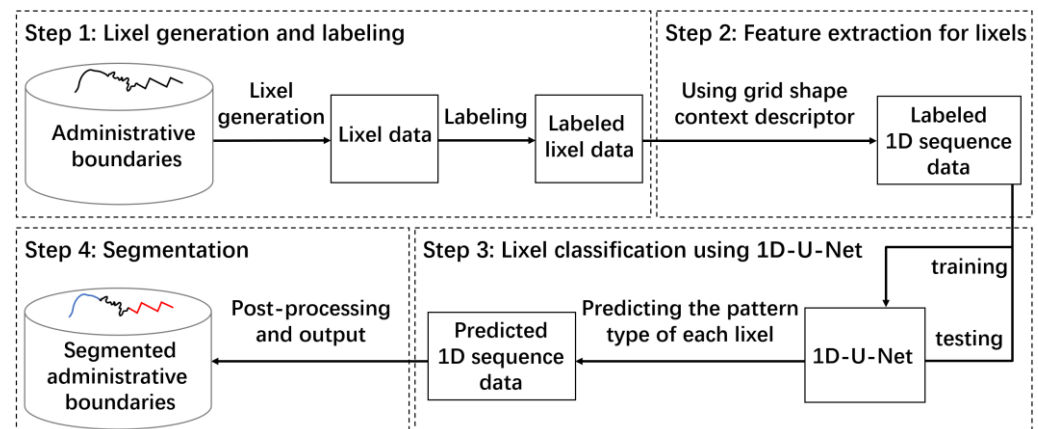
Based on the shape pattern space constructed from these three dimensions, as well as the characteristics of the experimental data, we classified the shape patterns of the boundaries into three pattern types: smooth irregular schematic (SIS), sharp regular schematic (SRS), and sharp irregular non-schematic (SIN), as listed in Table 1. SIS segments were characterized as smooth, with no repetitive characteristics and uncomplicated curved structures, and were derived from natural and artificial environments. SRS segments were characterized by sharp orthogonal regularities similar to building outlines, mainly derived from artificial environments. SIN segments were characterized as sharp, with no repetitive characteristics and complex hierarchical bends of various sizes, mainly derived from the natural environment.

Table 1. Line characteristics of pattern types for administrative boundary segments.

Pattern Type	Example	Smoothness	Regularity	Schematism
Smooth irregular schematic (SIS)		Smooth (gradually changing tangent directions)	Irregular (no repetitive characteristics)	Schematic (simple shape)
Sharp regular schematic (SRS)		Sharp (angles with large deviations)	Regular (repetitive right angles)	Schematic (simple shape)
Sharp irregular non-schematic (SIN)		Sharp (angles with large deviations)	Irregular (no repetitive characteristics)	Non-schematic (complex hierarchical bends with various sizes)

3. Methodology

Figure 3 shows the overall framework of the proposed method, including four main steps: lixel generation and labeling, lixel feature extraction, classification, and segmentation.

**Figure 3.** Framework of the segmentation method for administrative boundaries using the one-dimensional convolutional neural network (1D-U-Net).

- Lixel generation and labeling: Each administrative boundary was converted into a series of lixels via equidistant subdivision; the pattern type of each lixel was labeled;
- Feature extraction for lixels: Automatic extraction of the contextual features for each lixel, using GSCD;
- Lixel classification using 1D-U-Net: Construction of a 1D-U-Net to classify the pattern type of each lixel based on the extracted features;
- Segmentation: Obtaining the segmentation results by fusing adjacent lixels with the same pattern type.

3.1. Lixel Generation and Labeling

Figure 4 illustrates the lixel generation and labeling processes. First, each administrative boundary was manually segmented according to its morphological characteristics. The pattern type of each segment was classified according to the pattern category described in Section 2.2, and the visual classification was performed at a fixed scale to eliminate the influence of the data display scale on pattern discrimination. The boundary was then divided into a series of lixels with a fixed length. Note that the last divided part of the boundary that did not satisfy the predefined length was considered a lixel.

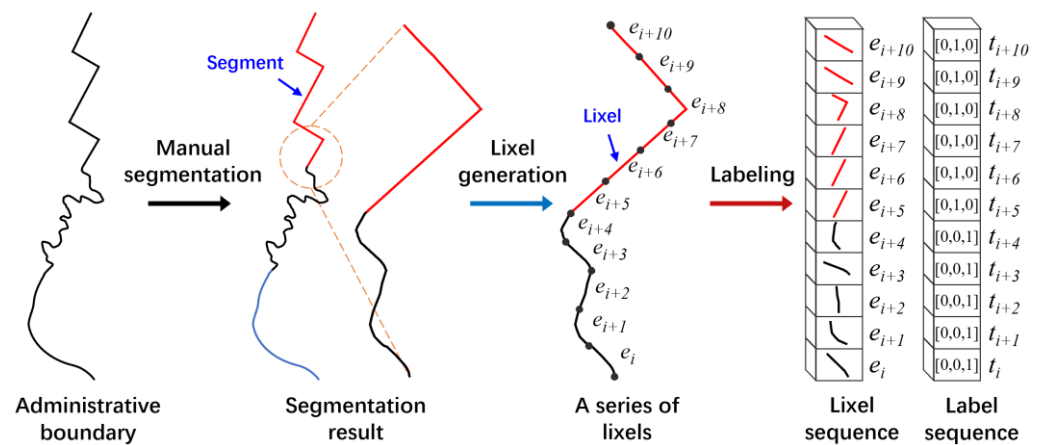


Figure 4. Lixel generation and labeling for each administrative boundary. Blue, red, and black colors of the divided segments indicate SIS, SRS, and SIN patterns, respectively.

The pattern type of each lixel was determined based on the pattern type of the segment to which it belonged. If a lixel was associated with multiple segments, it was labeled with the pattern type of the segment where it had the longest overlap length. Let $E_{n \times 1} = \langle e_1, e_2, \dots, e_n \rangle$ ($n > 1$) denote the sequence of lixels of an administrative boundary. The label information was organized as sequence data: $T_{n \times 1} = \langle t_1, t_2, \dots, t_n \rangle$, where $t_i \in \{[1, 0, 0], [0, 1, 0], [0, 0, 1]\}$ ($i = 1, 2, \dots, n$) is a three-dimensional one-hot vector that represents the pattern type of lixel e_i .

3.2. Extracting Lixel Features Using GSCD

The morphological characteristics of a lixel were determined by its neighbors within a certain range (i.e., the contextual information). In this study, we employed a GSCD commonly used for shape analysis and pattern recognition [32] to extract the contextual features of the lixels. For each lixel on an administrative boundary, the GSCD was computed as follows:

1. A regular grid centered on the midpoint of the lixel was created. The grid contained $p \times p$ cells and the cell edges were always horizontal and vertical. The length of the cell edges was set to the fixed length of the lixels.
2. The length of the boundary located in each cell was counted and normalized by dividing the total length of the boundary within all cells.
3. The normalized values of all cells were arranged from left to right and from bottom to top into a feature vector that was used to describe the contextual features of the lixel.

Figure 5 shows the GSCDs with 5×5 cells for three different lixels. The grayscale value of each grid cell represents the normalized feature value. The GSCDs of the three lixels in different contexts also significantly varied, indicating that the GSCD method had good feature characterization ability.

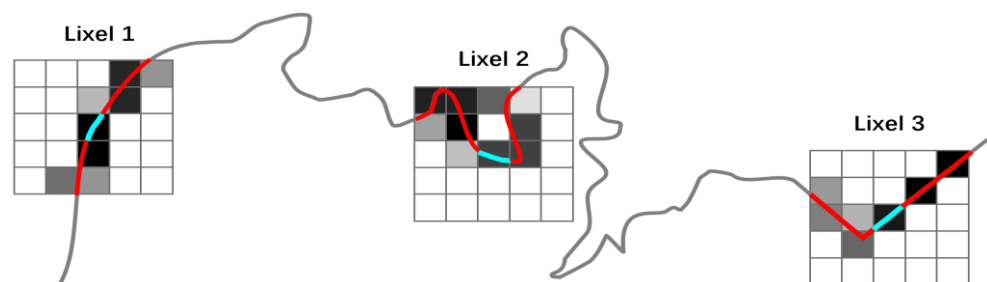


Figure 5. Grid shape context descriptors (GSCDs) with 5×5 cells for describing the contextual characteristics of three different lixels (in green) on an administrative boundary.

After computing the GSCDs of all lixels along an administrative boundary, we obtained sequence data, $F_{n \times (p \times p)}$, with n lixels and $(p \times p)$ -dimensional features, which served as input for the 1D-U-Net for subsequent pattern classification and segmentation.

3.3. Classifying Lixels Using a 1D-U-Net

This study referred to the classic U-Net [21] to construct the 1D-U-Net for classifying lixel patterns. The basic idea was to map the input features, $F_{n \times (p \times p)}$, into labels, $T_{n \times 1}$, through multiple 1-D convolution, pooling, and upsampling operations. Figure 6 shows the detailed architecture of the 1D-U-Net, which consisted of four downsampling blocks and four upsampling blocks. Each downsampling block contained two 1-D convolutions and one max pooling; each upsampling block contained one skip connection, one 1-D transposed convolution, and two 1-D convolutions. The last layer consisted of one 1-D convolution and one softmax activation function that mapped each feature vector into a label vector.

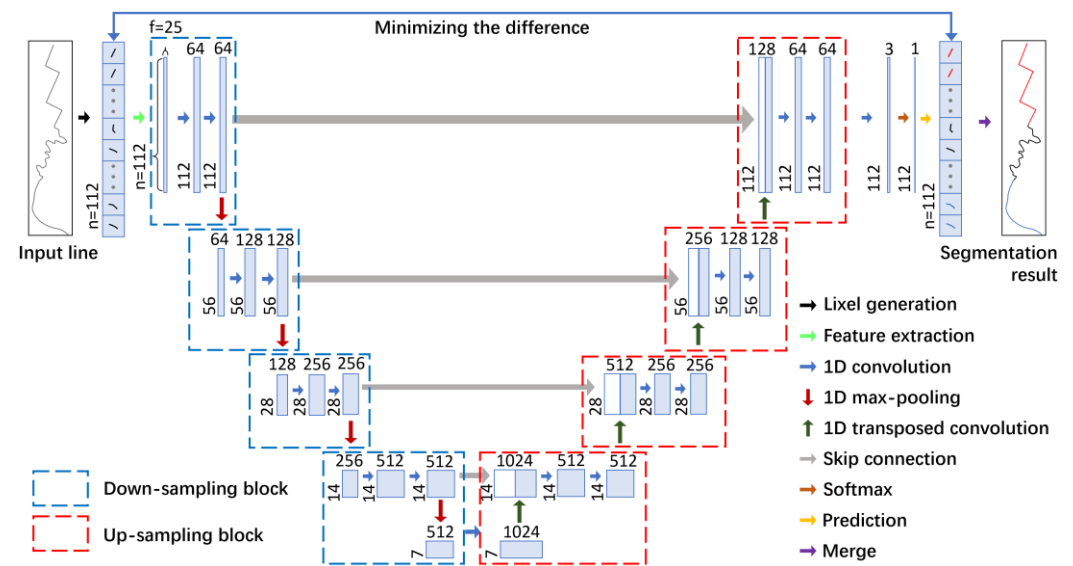


Figure 6. Architecture of the one-dimensional convolutional neural network (1D-U-Net) for classifying lixel pattern types.

The following sections described the operations performed on the sequence data for the lixels used in this model.

3.3.1. One-Dimensional Convolution and Pooling Operations

One-dimensional convolution and pooling operations were used to extract high-level multi-scale features for lixel pattern classification from the shallow GSCD features of the lixels. The convolution operation processes local data in a window through a sliding kernel and generates new features using a nonlinear activation function. A sequence, $X = \{X_1, X_2, \dots, X_n\}$, was convolved using a kernel, $k = \{w_1, w_2, \dots, w_{l_1}\}$, with a window size of l_1 to generate a new sequence, C . Each feature, c_i , in C was computed as follows:

$$c_i = f\left(k \cdot X_{i - \frac{(l_1-1)}{2} : i + \frac{(l_1-1)}{2}} + b\right) \quad (1)$$

where $f(\cdot)$ denotes the nonlinear Rectified Linear Unit (ReLU) activation function and b denotes the bias. The generated sequence is denoted as $C = \{c_1, c_2, \dots, c_{\lfloor \frac{n+2 \cdot d_1 - l_1}{s_1} \rfloor + 1}\}$, where $\lfloor \cdot \rfloor$ is the largest integer function, d_1 is the width of the padding, and s_1 is the stride length of the window.

The pooling operation obtained coarser-grained features by representing multiple features in a local window as one feature, which was conducive to reducing parameters, accelerating computations, and preventing overfitting, thereby improving the characterization and generalization ability of the model. Max pooling was used in this study. For multiple features, $C_{(i-1)*s_2+1:(i-1)*s_2+l_2}$, with a window size of l_2 , the maximum value was applied as the output feature, expressed as follows:

$$p_i = \max_{j=0}^{l_2-1} (C_{(i-1)*s_2+1+j}) \quad (2)$$

where s_2 is the stride length of the window. The generated sequence can be expressed as $P = \{p_1, p_2, \dots, p_{\lfloor \frac{n-l_2}{s_2} \rfloor + 1}\}$.

Figure 7 provides an example of the convolution and pooling operations, where the window size l_1 of the kernel was three and the window size l_2 of the pooling was two. A sequence C with a size of n and a sequence P with a size of $n/2$ were obtained after the convolution and max pooling operations, respectively.

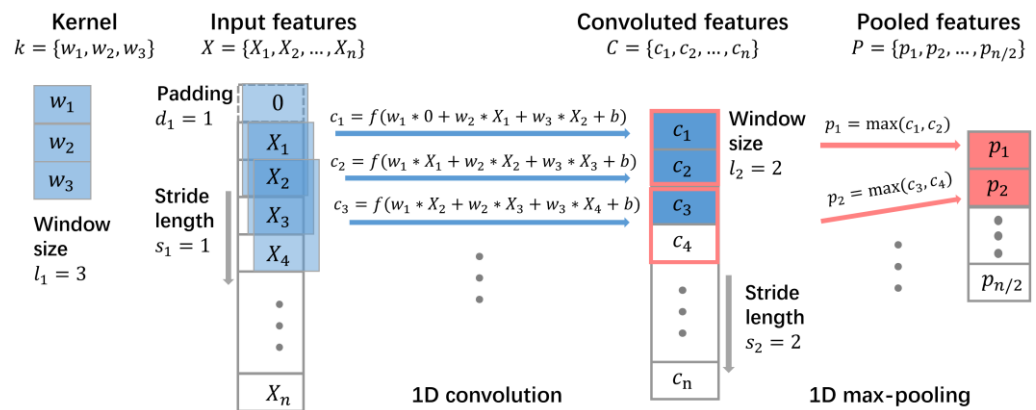


Figure 7. Illustration of the one-dimensional convolution and pooling operations.

3.3.2. One-Dimensional Upsampling Operation and Skip Connection

The pooling operation changed the size of the input features. The upsampling operation, implemented during the transposed convolution, was used to restore the feature size to the original input size.

The key step in the transposed convolution was to construct a transposed matrix, W , and multiply it with the feature vector, P . For the input features, P , with size $n \times 1$, the transposed matrix, W , was obtained by sliding a convolution kernel, $k = \{w_1, w_2, \dots, w_{l_3}\}$, with a window size of l_3 n times, where the vertical stride length of the sliding was s_3 and the horizontal stride length s_4 was 1. The transposed matrix W was computed:

$$W_{ij} = \begin{cases} w_{i-s_3*(j-1)}, & s_3 * (j-1) + 1 \leq i \leq s_3 * (j-1) + l_3 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $i = \{1, 2, \dots, s_3 * (n-1) + l_3\}$ and $j = \{1, 2, \dots, n\}$ denotes the number of rows and columns, respectively. The transposed convolution operation can be expressed:

$$V_{(s_3*(n-1)+l_3) \times 1} = W_{(s_3*(n-1)+l_3) \times n} \times P_{n \times 1} \quad (4)$$

where $W_{(s_3*(n-1)+l_3) \times n}$ denotes the transposed matrix and $V_{(s_3*(n-1)+l_3) \times 1}$ denotes the output features.

Figure 8 presents an example of a transposed convolution operation, where the vertical stride length s_3 was two and the convolution kernel width l_3 was two. The size of the transposed convolution matrix, W , is $2n \times n$, and thus a new feature vector, V , with a size of $2n \times 1$ is obtained.

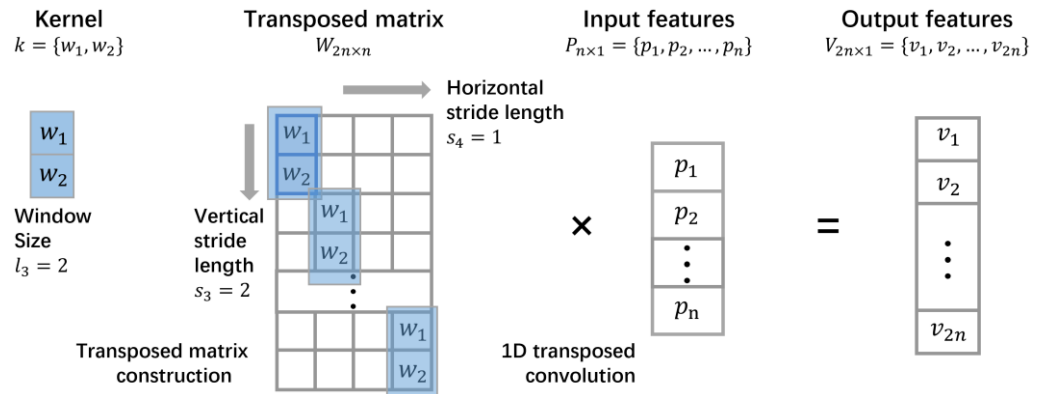


Figure 8. Illustration of the one-dimensional transposed convolution operation.

We note that the pooling and upsampling operations may omit important spatial location information from the original input features. To alleviate this problem, a skip connection operation was put in place between the 1-D convolution and 1-D transposed convolution. As illustrated in Figure 6, the output feature vectors after the 1-D transposed convolution were connected to the feature vectors after the 1-D convolution with the same number of channels. Then, the values of each element in the feature vector were corrected using a convolution operation to restore the number of channels.

3.3.3. Definition of Loss Function

After convolution processing in the final layer, a sequence the same length as the input sequence was output; its number of channels was three, corresponding to the three pattern types. Finally, the softmax function was employed to activate the output features to obtain the predicted probabilities. For the i -th ($i = 1, 2, \dots, n$) lixel of the output sequence, the probability $(a_i)_j$ ($j = 1, 2, 3$) that it belongs to the j -th pattern type was computed as follows:

$$(a_i)_j = \frac{e^{z_j}}{\sum_{k=1}^3 e^{z_k}} \quad (5)$$

where z_j denotes the feature of the j -th channel of the lixel. The output probability vector, $a_i = \langle (a_i)_1, (a_i)_2, (a_i)_3 \rangle$, for the i -th lixel satisfied $\sum_{j=1}^3 (a_i)_j = 1$. The pattern type with the highest probability was considered the predicted pattern for this lixel.

The training process allowed the differences (i.e., the loss value, E) to be minimized between the predicted probability vectors, a_1, a_2, \dots, a_n , and the labeling one-hot vectors, t_1, t_2, \dots, t_n , where n was the number of lixels. In this study, the cross-entropy function was applied to measure the difference, expressed as follows:

$$E = -\frac{1}{n} \sum_{i=1}^n t_i \log(a_i) \quad (6)$$

The smaller the loss value, E , the closer the predictions were to the labels. Here, 1D-U-Net was trained using a backpropagation algorithm. During training, the predicted value for each mini-batch of data was obtained via forward propagation and the loss value was calculated. The trainable parameters were then updated gradually according to the learning rate and partial derivatives of each parameter relative to the loss value.

3.4. Obtaining Segmentation Results

The trained 1D-U-Net was used to predict the pattern types of all lixels along an administrative boundary. Homogeneous segments were then obtained by merging adjacent lixels with the same pattern types. However, there were some predictions for a few lixels that were incorrect, resulting in extremely short segments after the merging operation. As

these short segments interfered with the segmentation results, post-processing was required to eliminate them. In this study, an iterative fusion method was used. As illustrated in Figure 9, post-processing was implemented as follows:

1. The segmentation results of the administrative boundary were traversed and the segment z_i with the smallest length was identified;
2. If the length of z_i was smaller than the predefined threshold S , z_i was merged with its neighbor with a longer length;
3. Steps (1) and (2) were repeated until there were no segments smaller than S .

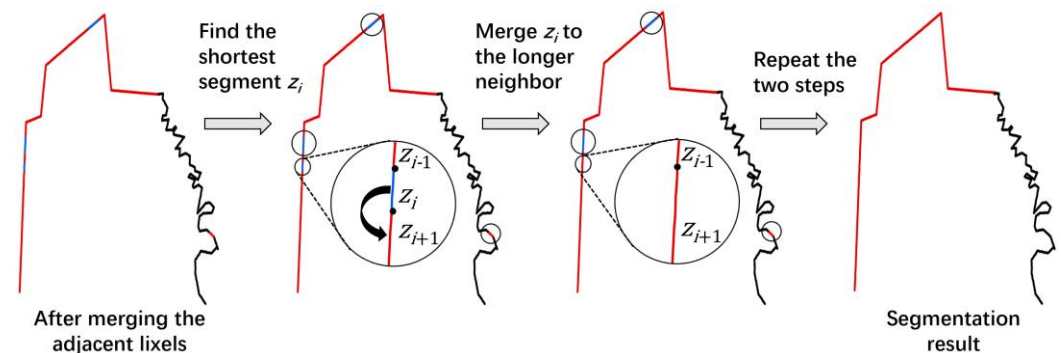


Figure 9. Post-processing for short segments to obtain fine segmentation results.

4. Experiments

The GSCD to automatically extract the features for pixels was implemented as an ArcMap add-in (Environmental Systems Research Institute, Redlands, CA, USA) with C#, and the constructed 1D-U-Net to classify pixels was implemented with Python in TensorFlow. This section presents the experimental design, pixel classification performance, line segmentation results and analysis, and a discussion on the parameter sensitivities.

4.1. Experimental Design

4.1.1. Sample Dataset Generation

To generate the samples, two volunteers with specialized knowledge independently segmented the administrative boundaries in the training and test datasets according to the criteria listed in Table 1. If the segmentation results from the two volunteers were different, a third volunteer with extensive cartographic experience rechecked and made the final decision. To ensure the consistency of cartographical details, the display scale when segmenting the boundaries was fixed at the data scale (i.e., 1:50,000). Next, each boundary was divided into a series of pixels, as described in Section 3.1. The setting of pixel size is critical for perceiving line patterns [33,34]. In this current study, by referring to the concept of the smallest visual object (SVO) discussed in the work of Li and Openshaw [33], and their comparative studies on the size settings of this parameter, the length of each pixel was set to 0.5 mm (map distance), corresponding to a ground distance of 25 m at a scale of 1:50,000. Consequently, there were 37,235 and 31,425 pixels for the training and test datasets, respectively.

To fully train the pixel classification model, two data augmentation methods were used to increase the sample size of the training dataset. As illustrated in Figure 10a, each administrative boundary was first rotated every 30° ; the sample size was increased 11 times. Next, we implemented the sliding window method, sliding a fixed window with a size of 112 pixels along each boundary at steps of 30 pixels. As shown in Figure 10b, each window slide yielded a sample; thus, multiple samples with partial overlaps were obtained from one administrative boundary. Through data augmentation, 9432 training samples were obtained.

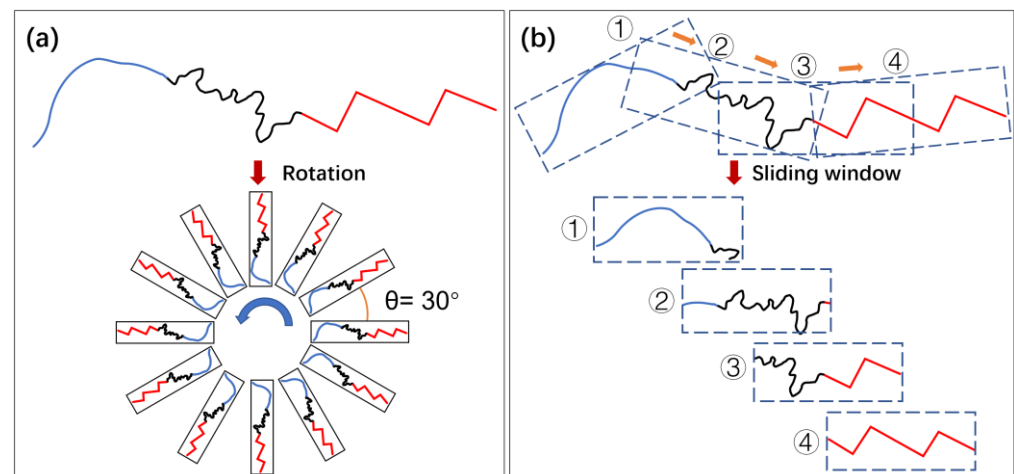


Figure 10. Two data augmentation methods for the training dataset: (a) rotation transformation method and (b) sliding window method.

For the test dataset, data augmentation was not applied and the lixel sequence of each administrative boundary was divided into samples with a length of 112 lixels. If the last sample did not reach the required length, zeros were added. Finally, 308 test samples were obtained.

4.1.2. Parameter Settings

After preparing all the samples, the contextual features of each lixel were computed using the GSCD; a grid of 5×5 cells was used to construct the GSCD in the experiments. Therefore, each lixel was described using 25 dimensional features. The model was trained using the Adam optimizer for 50 epochs, with a learning rate of 0.0001.

Two existing segmentation methods based on machine learning were implemented for comparison; the backpropagation artificial neural network (BANN) [16,17] and naïve Bayesian (NB) [18] methods. Both methods used the sliding window method to generate samples from the administrative boundaries. The key step was to cut the administrative boundaries according to the fixed-size window which moved with an increment of a certain length. Referring to the experimental parameter settings discussed in the literature [17,18], the window size and length of the increment were set to 1500 and 150 m, respectively. Ten features were extracted to describe the morphological structures of the samples, including the segment length and baseline length ratio, mean of the bend lengths, median vertical distance from each point to the baseline, coefficient of variance of the bend baseline length, coefficient of variance of the bend length and baseline length ratio, median distance between two continuous points, coefficient of variance of the bend area and squared baseline length ratio, median ratio of the bend length to baseline length, fractal dimension, and mean of the turning angles. For more information on the definitions and computations of these features, please refer to the works of Ariza López and Balboa [16,17] and Liu et al. [18]. Based on principal component analysis (PCA), seven features with a sum of information greater than 90% were used as the inputs for the BANN and NB models. The number of neurons in the hidden layer of BANN was set to 15 and the ReLU activation function was used. A Gaussian model was used for the NB method.

4.2. Lixel Classification Performance Using 1D-U-Net

Figure 11 presents the accuracy and loss values of 1D-U-Net for lixel classification during the training phase. The classification accuracy and training loss changed rapidly during the first five rounds, gradually stabilized after ten epochs, and reached a peak after 50 epochs. After the training converged, the classification accuracy of the model on the training set reached 99.04%. Subsequently, the trained model was used to classify the

lixels in the test samples, with a classification accuracy of 90.42%. This result indicates that 1D-U-Net can classify the pattern types of lixels with high accuracy.

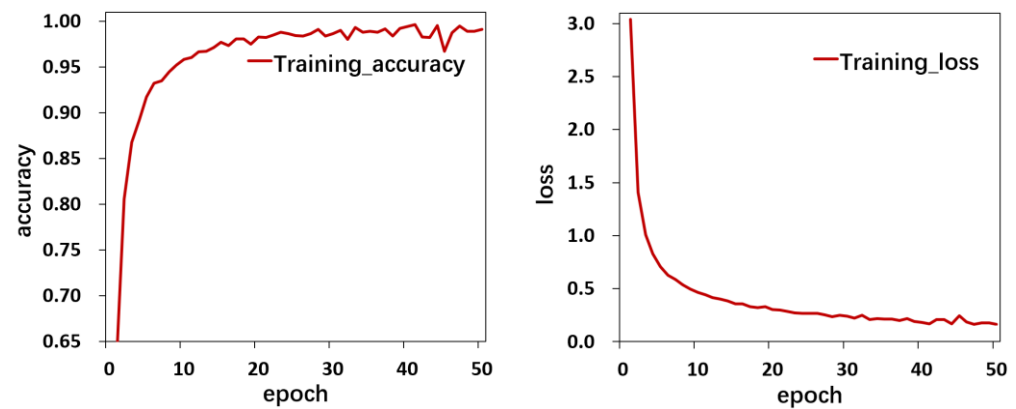


Figure 11. Training accuracy and loss in 1D-U-Net for lixel classification.

The *Precision*, *Recall*, and *F₁-score* were used to quantitatively evaluate the classification results. For each pattern type, the three metrics are defined as follows:

$$Precision = \frac{TP_{num}}{TP_{num} + FP_{num}} \times 100\% \quad (7)$$

$$Recall = \frac{TP_{num}}{TP_{num} + FN_{num}} \times 100\% \quad (8)$$

$$F_1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

where TP_{num} denotes the number of lixels that were both automatically predicted and manually labeled with the same pattern type; FP_{num} is the number of lixels that were automatically predicted to be this pattern type but were manually identified as a different pattern type; and FN_{num} denotes the number of lixels that were manually identified as this pattern type but were incorrectly predicted as a different pattern types.

Table 2 lists the confusion matrix of the lixel classification results for the test samples using the trained model, and the three metrics. The *F₁-score* value for the SIN pattern reached 0.96 while the *F₁-score* values for the SIS and SRS patterns were slightly lower, at 0.90 and 0.88, respectively. This is because misclassification occurred mainly between the SIS and SRS lixels. Overall, the proposed model could extract the shapes' contextual features and achieve lixel classification for the test boundaries with relatively high accuracy.

Table 2. Confusion matrix and three evaluation metrics for lixel classification of the test samples using 1D-U-Net.

Manually Labeled \ Automatically Predicted				<i>Precision</i> (%)	<i>Recall</i> (%)	<i>F₁-Score</i>
	SIS Pattern	SRS Pattern	SIN Pattern			
SIS pattern	11,642	1277	184	90.76	88.85	0.90
SRS pattern	1083	9851	157	86.92	88.82	0.88
SIN pattern	102	206	6923	95.31	95.74	0.96

4.3. Segmentation Result Evaluation

4.3.1. Qualitative Evaluation

Based on the classification results, homogeneous segments were obtained by merging adjacent lixels with the same pattern types. To make the comparison fairer, the iterative fusion method was also employed in the BANN and NB methods after classifying each

increment. In all three methods, the length threshold S for the short segment needing to be fused was set to 900 m (i.e., 36 lixels in the proposed method and six increments in the BANN and NB methods). Figure 12 shows the segmentation results for the test boundaries using different methods.

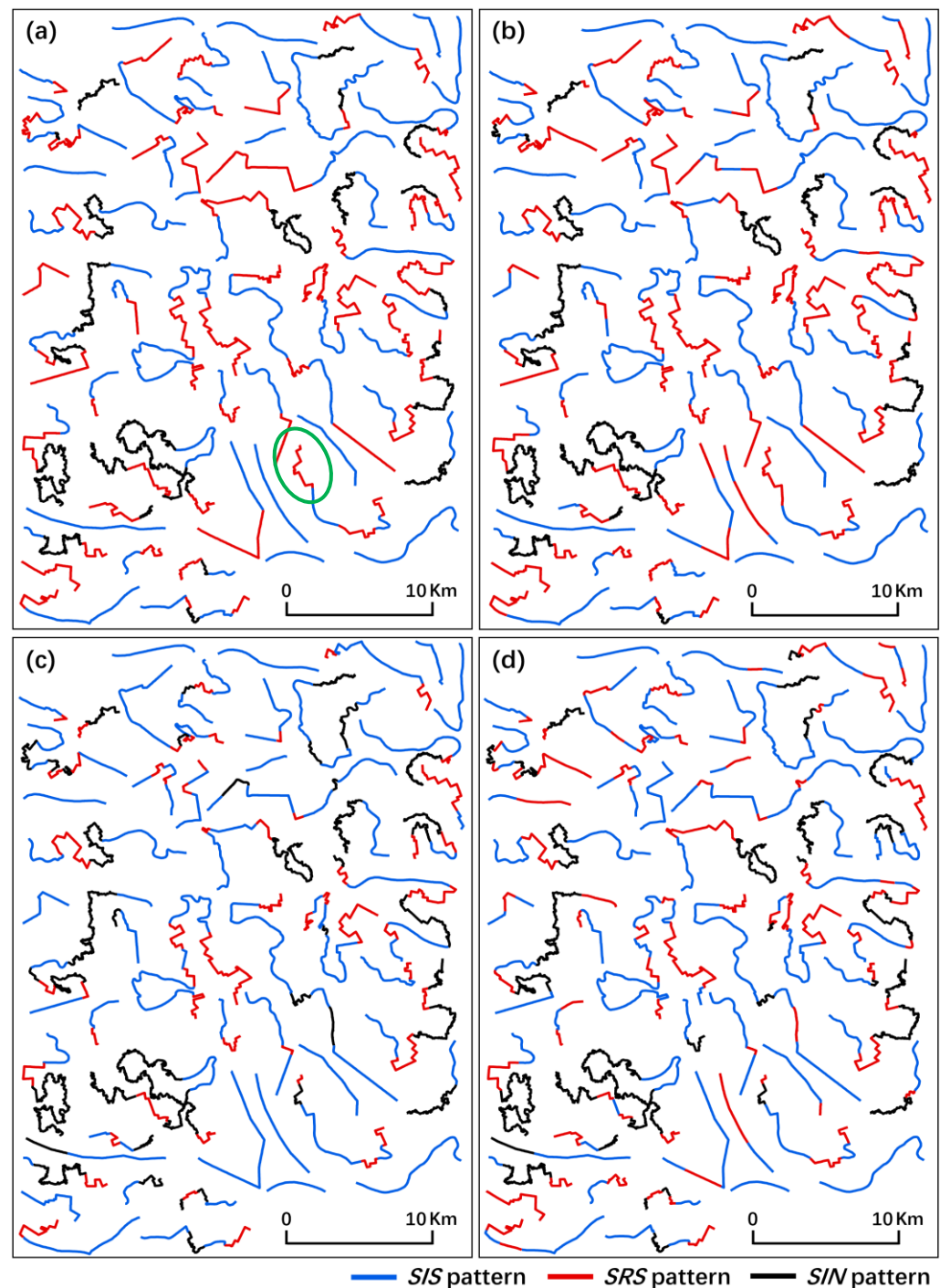


Figure 12. Segmentation results for the test administrative boundaries: (a) manual segmentation, (b) 1D-U-Net method, (c) BANN method, and (d) NB method.

All three methods performed well in the segmentation of the test boundaries. By using the iterative fusion method, many short segments were removed and the segmentation results became more coherent. Careful comparison revealed that the proposed method had fewer segmentation errors than the BANN and NB methods. For example, the segment

marked by the green ellipse in Figure 12a was manually labeled as an SRS pattern, but both the BANN and NB methods incorrectly predicted it as SIS or SIN; only the results of the proposed method were consistent with the manual labeling.

Figure 13 shows the differences between the manual and predicted results based on different methods, where predictions that coincided with manual results are rendered in gray while colored results indicate inconsistent results. There were significantly fewer inconsistent segments in the results of the proposed method than in the other two methods. The length of the inconsistent segments in the results of the proposed method was 59.5 km while the BANN and NB methods produced totals of 167.64 and 184.18 km of inconsistent segments, respectively.

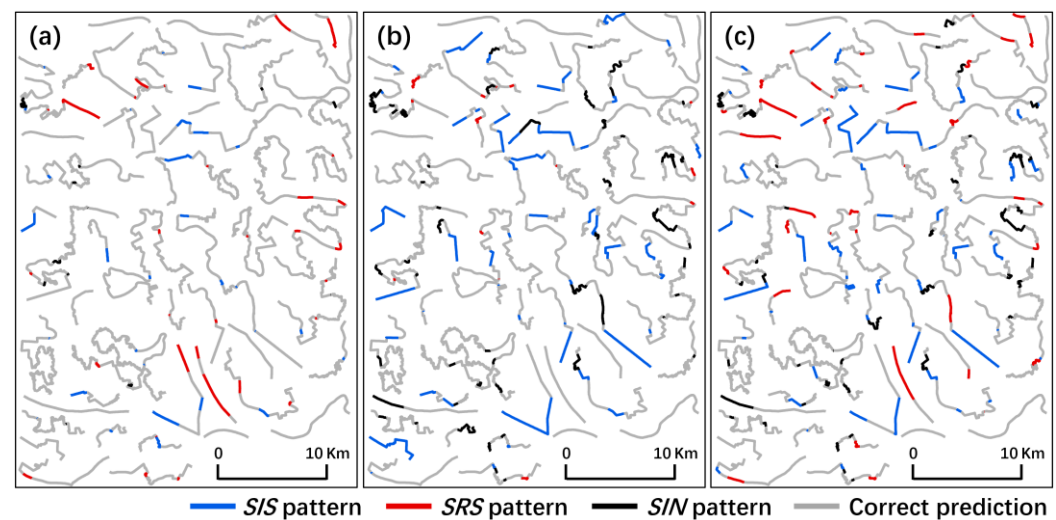


Figure 13. Differences in the manual segmentation results and predicted results of different methods for the test administrative boundaries: (a) 1D-U-Net, (b) BANN method, and (c) NB method. Gray indicates that the segmentation results are consistent; blue, red, and black denote the segments with inconsistent predictions as SIS, SRS, and SIN patterns, respectively.

Inconsistent segmentation can be divided into two classes. The first class is characterized by inconsistent segments that are short. This type of inconsistency was produced by deviations in the segment points; deviations within a certain range had a negligible impact on subsequent analyses. Instances in the second class included long inconsistent segments, produced by the incorrect classification of pattern types. As shown in Figure 13, both classes of inconsistencies in the results obtained by the BANN and NB methods were significantly greater than those of the proposed method.

To further investigate the two classes of inconsistencies that occurred in the segmentation results using different methods, we analyzed the inconsistencies in four typical boundaries, as shown in Figure 14. The deviations in the segmentation points occurred mainly at the junctions of the SIS and SRS segments. In terms of incorrect classification, the identification of SRS segments generated the worst performances, particularly in the results of the BANN and NB methods. Lines with SRS patterns may have long straight segments on both sides or may exhibit consecutive bends. In the former case, the local characteristics are similar to those of SIS lines in gently sloping areas, whereas in the latter case, SRS and SIN lines exhibit overall similarity. Overall, accurately identifying the SRS segments requires understanding the local and global characteristics of the lines.

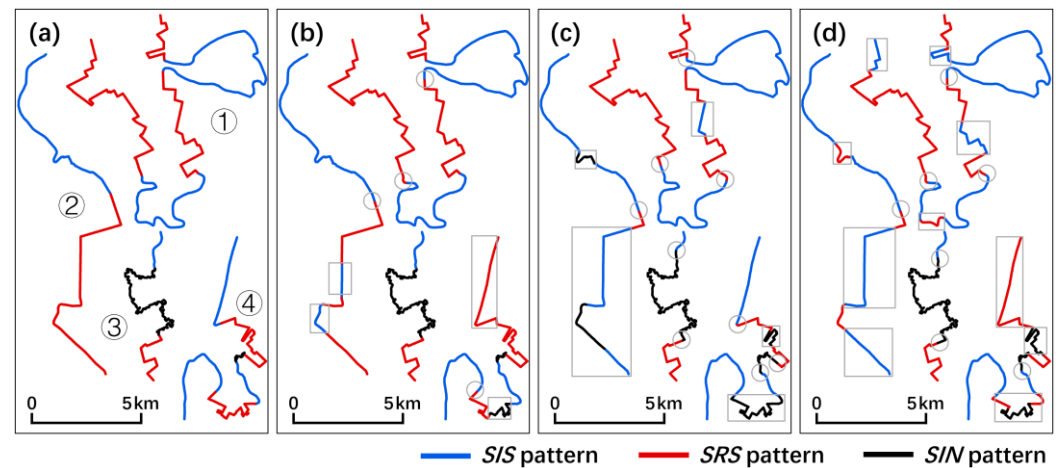


Figure 14. Segmentation results for four typical test boundaries (numbered 1–4) using different methods: (a) manual identification, (b) 1D-U-Net, (c) BANN method, and (d) NB method. Boxes highlight incorrect classifications and circles mark deviations in the segmentation points.

4.3.2. Quantitative Evaluation

A consistency ratio (CR) metric was defined to quantitatively analyze the segmentation performance of the different methods. For the i -th pattern type, the CR_i metric was computed as follows:

$$CR_i = \frac{Lc_i}{Lt_i} \times 100\% \quad (10)$$

where Lc_i and Lt_i denote the total length of the segments correctly identified as the i -th pattern type and the total length of the segments labeled as the i -th pattern type, respectively. The overall CR (OCR) was calculated as follows:

$$OCR = \frac{\sum_i Lc_i}{Lt} \times 100\% \quad (11)$$

where Lt denotes the total length of all test boundaries. Table 3 lists the CR and OCR metrics for the segmentation results obtained using different methods. The OCR of the 1D-U-Net method reached 92.41%, which was higher than for either of the two existing methods. Moreover, the identification of SIN segments had the highest CR, reaching 97% using all three methods. In contrast, the three methods each had the lowest CR for SRS segments, with CRs for the two existing methods that were less than 60%; the CR of the proposed method approached 90%. Overall, the proposed method demonstrated the best segmentation performance in terms of CR and OCR metrics; it was advantageous in the identification of SRS segments, compared with the traditional BANN and NB methods.

Table 3. Consistence ratios (CRs) and overall CRs (OCRs) of the segmentation results for the test boundaries using different methods.

Method	CR (%)			OCR (%)
	SIS Pattern	SRS Pattern	SIN Pattern	
BANN	91.59	50.28	98.51	78.61
NB	83.51	54.26	97.87	76.50
1D-U-Net	91.23	90.54	97.40	92.41

4.4. Discussion

As previously mentioned, the GSCD can capture the local contextual features of each lixel and alleviate the subjective influence of artificially defined features. However, the grid

size (i.e., the number of cells) had a significant impact on the lixel feature extraction and may have affected the segmentation performance. For descriptors with different grid sizes, the extracted features differed (Figure 15).

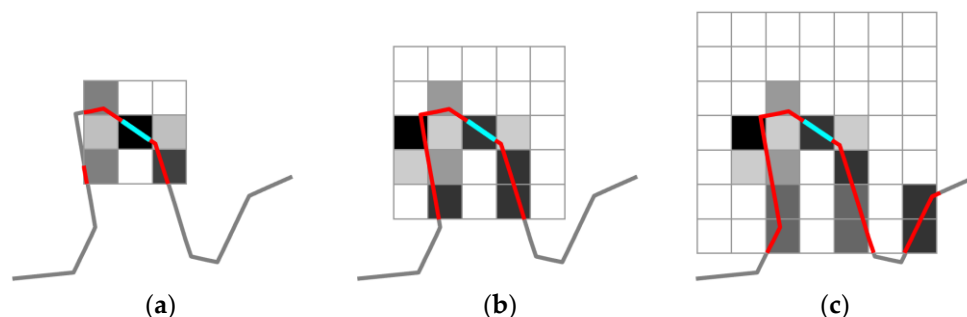


Figure 15. Grid shape context descriptors (GSCDs) with different sized grids for each lixel: (a) grid size of 3×3 ; (b) grid size of 5×5 ; (c) grid size of 7×7 .

To discuss the sensitivity of the proposed method to the size of the GSCD, an experiment was conducted by varying the grid sizes from 3×3 to 7×7 . Table 4 lists the lixel classification accuracies. With an increase in grid size, the lixel classification accuracy of the model initially increased and then decreased. The model performed best when the grid cell size was 5×5 . A possible reason for this result is that as the number of cells increased from 3×3 , the receptive field gradually increased; therefore, the contextual features of the lixels were enriched, thus improving the classification performance. However, when the grid size was greater than 5×5 , many features were zero (Figure 15c) and the contextual information may have been disturbed by long-distance boundaries, thus leading to the degradation of classification performance.

Table 4. Classification accuracies of the proposed model using GSCDs with different grid sizes.

Length of Cell Edge (m)	Grid Size (Lixel \times Lixel)	Classification Accuracy (%)
25	3×3	89.17
	4×4	89.58
	5×5	90.42
	6×6	89.68
	7×7	88.53

5. Conclusions and Outlook

This study proposed a novel deep-learning approach for pattern recognition and segmentation of administrative boundaries, based on 1D-U-Net. In the model, a lixel was used as the basic processing unit and a GSCD was employed to extract the descriptive features of each lixel. Subsequently, a 1D-U-Net architecture was constructed to predict the lixels' pattern types. Finally, the predicted results were iteratively fused to obtain the final segmentation results for the administrative boundaries. The experimental results showed that the lixel classification accuracy of 1D-U-Net reached 90.42% for the test administrative boundaries; the OCR of the segmentation results for the test samples was 92.41%, which was higher than that of the BANN- and NB-based segmentation methods.

Unlike existing methods, the proposed method is derived from image segmentation. It transforms the line segmentation problem into lixel classification by representing unstructured vector-based data as regular lixel-based sequence data. Additionally, the GSCD provides a promising means for characterizing local contextual information in lixels. These two advantages allow the proposed method to effectively identify the shape characteristics of a line, which significantly improves segmentation performance. Follow-up studies should focus on certain aspects. The method should be applied to other geographical lines,

such as coastlines, rivers, and roads. Model optimization and parameter settings for the method, such as lixel size, should be further investigated. In addition, high-quality sample libraries should be constructed to further improve the model.

Author Contributions: Conceptualization, Min Yang and Xiongfeng Yan; methodology, Min Yang and Xiongfeng Yan; software, Haoran Huang and Yiqi Zhang; validation, Min Yang and Xiongfeng Yan; formal analysis, Min Yang, Haoran Huang, Yiqi Zhang and Xiongfeng Yan; investigation, Min Yang, Haoran Huang, Yiqi Zhang and Xiongfeng Yan; writing—original draft preparation, Min Yang, Haoran Huang, Yiqi Zhang and Xiongfeng Yan; writing—review and editing, Min Yang and Xiongfeng Yan; supervision, Min Yang and Xiongfeng Yan; funding acquisition, Min Yang and Xiongfeng Yan. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant numbers 42071450, 42001415.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Buttenfield, B.P. *A Rule for Describing Line Feature Geometry*; Longman: London, UK; Wiley: New York, NY, USA, 1991; pp. 150–171.
2. Dutton, G. Scale, sinuosity, and point selection in digital line generalization. *Cartogr. Geogr. Inf. Sci.* **1999**, *26*, 33–54. [\[CrossRef\]](#)
3. Visvalingam, M.; Williamson, P.J. Simplification and generalization of large scale data for roads: A comparison of two filtering algorithms. *Cartogr. Geogr. Inf. Syst.* **1995**, *22*, 264–275. [\[CrossRef\]](#)
4. Plazanet, C. Modelling geometry for linear feature generalization. In *Geographic Information Research: Bridging the Atlantic*; Craglia, M., Coucleis, H., Eds.; Taylor & Francis: London, UK, 1997; pp. 264–279.
5. Ai, T.; Ke, S.; Yang, M.; Li, J. Envelope generation and simplification of polylines using Delaunay triangulation. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 297–319. [\[CrossRef\]](#)
6. Samsonov, T.; Yakimova, O. Regression modeling of reduction in spatial accuracy and detail for multiple geometric line simplification procedures. *Int. J. Cartogr.* **2020**, *6*, 47–70. [\[CrossRef\]](#)
7. Du, J.; Wu, F.; Xing, R.; Gong, X.; Yu, L. Segmentation and sampling method for complex polyline generalization based on a generative adversarial network. *Geocarto Int.* **2021**, *37*, 4158–4180. [\[CrossRef\]](#)
8. Wang, Z.; Müller, J.C. Line generalization based on analysis of shape characteristics. *Cartogr. Geogr. Inf. Sci.* **1998**, *25*, 3–15. [\[CrossRef\]](#)
9. Wang, Z.; Lee, D. Building simplification based on pattern recognition and shape analysis. In Proceedings of the 9th International Symposium on Spatial Data Handling, Beijing, China, 10–12 August 2000; pp. 58–72.
10. García Balboa, J.L.; Ariza López, F.J. Sinuosity pattern recognition of road features for segmentation purposes in cartographic generalization. *Pattern Recognit.* **2009**, *42*, 2150–2159. [\[CrossRef\]](#)
11. Park, W.; Yu, K. Hybrid line simplification for cartographic generalization. *Pattern Recognit. Lett.* **2011**, *32*, 1267–1273. [\[CrossRef\]](#)
12. Samsonov, T.E.; Yakimova, O.P. Shape-adaptive geometric simplification of heterogeneous line datasets. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1485–1520. [\[CrossRef\]](#)
13. Wei, Z.; Liu, Y.; Cheng, L.; Ding, S. A progressive and combined building simplification approach with local structure classification and backtracking strategy. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 302. [\[CrossRef\]](#)
14. Yang, M.; Yuan, T.; Yan, X.; Ai, T.; Jiang, C. A hybrid approach to building simplification with an evaluator from a backpropagation neural network. *Int. J. Geogr. Inf. Sci.* **2022**, *36*, 280–309. [\[CrossRef\]](#)
15. Mustiere, S. Cartographic generalization of roads in a local and adaptive approach: A knowledge acquisition problem. *Int. J. Geogr. Inf. Sci.* **2005**, *19*, 937–955. [\[CrossRef\]](#)
16. García Balboa, J.L.; Ariza López, F.J. Generalization-oriented road line classification by means of an artificial neural network. *Geoinformatica* **2008**, *12*, 289–312. [\[CrossRef\]](#)
17. Ariza López, F.J.; García Balboa, J.L. Generalization-oriented road line segmentation by means of an artificial neural network applied over a moving window. *Pattern Recognit.* **2008**, *41*, 1593–1609. [\[CrossRef\]](#)
18. Liu, P.; Yang, Q. Coastline segment model research for map generalization based on Bayesian method. *Comput. Eng. Appl.* **2016**, *52*, 174–179. [\[CrossRef\]](#)
19. Plazanet, C.; Affholder, J.G.; Fritsch, E. The importance of geometric modeling in linear feature generalization. *Cartogr. Geogr. Inf. Syst.* **1995**, *22*, 291–305. [\[CrossRef\]](#)

20. Douglas, D.H.; Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartogr. Int. J. Geogr. Inf. Geovis.* **1973**, *10*, 112–122. [[CrossRef](#)]
21. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [[CrossRef](#)] [[PubMed](#)]
22. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI), Cham, Germany, 5–9 October 2015; pp. 234–241.
23. Yan, X.; Ai, T.; Yang, M.; Yin, H. A graph convolutional neural network for classification of building patterns using spatial vector data. *ISPRS J. Photogramm. Remote Sens.* **2019**, *150*, 259–273. [[CrossRef](#)]
24. Yang, M.; Jiang, C.; Yan, X.; Ai, T.; Cao, M.; Chen, W. Detecting interchanges in road networks using a graph convolutional network approach. *Int. J. Geogr. Inf. Sci.* **2022**, *36*, 1119–1139. [[CrossRef](#)]
25. Yu, H.; Ai, T.; Yang, M.; Huang, L.; Yuan, J. A recognition method for drainage patterns using a graph convolutional network. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *107*, 102696. [[CrossRef](#)]
26. Yan, X.; Ai, T.; Yang, M.; Tong, X. Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *Int. J. Geogr. Inf. Sci.* **2021**, *35*, 490–512. [[CrossRef](#)]
27. Liu, C.; Hu, Y.; Li, Z.; Xu, J.; Han, Z.; Guo, J. TriangleConv: A deep point convolutional network for recognizing building shapes in map space. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 687. [[CrossRef](#)]
28. Feng, Y.; Thiemann, F.; Sester, M. Learning cartographic building generalization with deep convolutional neural networks. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 258. [[CrossRef](#)]
29. Courtial, A.; El Ayedi, A.; Touya, G.; Zhang, X. Exploring the potential of deep learning segmentation for mountain roads generalisation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 338. [[CrossRef](#)]
30. He, Y.; Ai, T.; Yu, W.; Zhang, X. A linear tessellation model to identify spatial pattern in urban street networks. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1541–1561. [[CrossRef](#)]
31. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751. [[CrossRef](#)]
32. Fan, H.; Zhao, Z.; Li, W. Towards measuring shape similarity of polygons based on multiscale features and grid context descriptors. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 279. [[CrossRef](#)]
33. Li, Z.; Openshaw, S. Algorithms for automated line generalization based on a natural principle of objective generalization. *Int. J. Geogr. Inf. Sci.* **1992**, *6*, 373–389. [[CrossRef](#)]
34. Karsznia, I.; Gołębiowska, I.M.; Korycka-Skorupa, J.; Nowacki, T. Searching for an optimal hexagonal shaped enumeration unit size for effective spatial pattern recognition in choropleth maps. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 576. [[CrossRef](#)]