Article

U-Net-LSTM: Time Series-Enhanced Lake Boundary Prediction Model

Lirong Yin [1], Lei Wang [1], Tingqiao Li [2], Siyu Lu [2], Jiawei Tian [2], Zhengtong Yin [3], Xiaolu Li [4] and Wenfeng Zheng [2,*]

[1] Department of Geography & Anthropology, Louisiana State University, Baton Rouge, LA 70803, USA
[2] School of Automation, University of Electronic Science and Technology of China, Chengdu 610054, China
[3] College of Resource and Environment Engineering, Guizhou University, Guiyang 550025, China
[4] School of Geographical Sciences, Southwest University, Chongqing 400715, China
[*] Correspondence: winfirms@uestc.edu.cn

Abstract: Change detection of natural lake boundaries is one of the important tasks in remote sensing image interpretation. In an ordinary fully connected network, or CNN, the signal of neurons in each layer can only be propagated to the upper layer, and the processing of samples is independent at each moment. However, for time-series data with transferability, the learned change information needs to be recorded and utilized. To solve the above problems, we propose a lake boundary change prediction model combining U-Net and LSTM. The ensemble of LSTMs helps to improve the overall accuracy and robustness of the model by capturing the spatial and temporal nuances in the data, resulting in more precise predictions. This study selected Lake Urmia as the research area and used the annual panoramic remote sensing images from 1996 to 2014 (Lat: 37°00′ N to 38°15′ N, Lon: 46°10′ E to 44°50′ E) obtained by Google Earth Professional Edition 7.3 software as the research data set. This model uses the U-Net network to extract multi-level change features and analyze the change trend of lake boundaries. The LSTM module is introduced after U-Net to optimize the predictive model using historical data storage and forgetting as well as current input data. This method enables the model to automatically fit the trend of time series data and mine the deep information of lake boundary changes. Through experimental verification, the model's prediction accuracy for lake boundary changes after training can reach 89.43%. Comparative experiments with the existing U-Net-STN model show that the U-Net-LSTM model used in this study has higher prediction accuracy and lower mean square error.

Keywords: lake boundary prediction; U-Net; CNN; long-short time memory; remote sensing; deep learning

check for updates

Citation: Yin, L.; Wang, L.; Li, T.; Lu, S.; Tian, J.; Yin, Z.; Li, X.; Zheng, W. U-Net-LSTM: Time Series-Enhanced Lake Boundary Prediction Model. *Land* **2023**, *12*, 1859. https://doi.org/10.3390/land12101859

Academic Editors: Ebrahim Ghaderpour and Paolo Mazzanti

Received: 22 August 2023
Revised: 21 September 2023
Accepted: 27 September 2023
Published: 29 September 2023

1. Introduction

With the deepening of environmental science research [1–3], the research on the trend prediction of natural lake boundaries has become a research direction with potential value [4–6]. The combination of optical remote sensing images and deep learning technology provides a favorable tool for the research and development of lake boundary change prediction research [4,7,8]. In addition, to better predict lake boundary changes, change detection is one of the most important prerequisites for predicting lake boundaries based on remote sensing images [9–11]. For change detection, it is necessary to learn the time series data of optical remote sensing images and record the learned change information, and then use it for time series data with transferability. Traditional models, such as manual and satellite-based mapping, rely heavily on human input and thus are only suitable for certain steps in the prediction task [12]. In the final decision stage [13,14], an integrated and independent change detection method is needed to detect changes more extensively and conveniently without auxiliary tasks, such as threshold selection or classification. In order to solve the above problems, it is necessary to reduce the dependence on humans in the learning process. In recent years, with the development of deep learning and remote sensing technology, many researchers have applied these strategies to land cover prediction, greatly reducing the manual workload in the model.

In 2018, Liu et al. [15] applied a convolutional neural network (CNN) to multi-temporal polarimetric synthetic Aperture radar (SAR) images to detect and predict land cover change trends. This model first employs a CNN with only convolutional layers for change detection and then forms an LRCNN by imposing spatial constraints called local restrictions on the output layer of the CNN. LRCNN can not only identify different types of changed and unchanged data, but can also ensure noise insensitivity without losing details of changed regions. In 2020, Li et al. [16] designed a land cover classification model by using a CNN classifier to extract high-resolution features of land from Google Earth images and combining traditional machine learning classifiers, random forest (RF) and support vector machine (SVM), to extract spectral features from 30-m Landsat data. In the same year, Giang et al. [17] used the U-Net convolutional network to design a land cover classification model based on multispectral unmanned aerial vehicle (UAV) images. A total of 7 optimizer function types were used, and an area of 0.5 km $\times$ 0.8 km was used to train and test seven U-Net models. The experimental results show that the two models using the Nadam and Adadelta optimizer functions achieved higher than 83% accuracy in classifying the six land cover types, especially in open-pit mining areas and polluted streams that flow from mining areas.

However, current land cover prediction models based on CNN has great limitations when detecting time-series data such as land cover change [18–21]. However, we expect that the land cover prediction model can detect changes in all available information using complete and independent change rules and can transform the change rules into new target multi-temporal images. Simply stated, an effective change rule should have the ability to represent the change information reliably, and the learned change rule can be transferred to the new target image without any additional learning process, which shows its transferability [22].

Recently, recurrent neural network (RNN) has shown its ability to help solve the problem above [23]. In RNN, the output of neurons can act directly on itself in the next timestamp. For change detection, RNN uses hidden layers or storage units to learn the time evolution characteristics of changes and model the dynamic changes of input continuous time series data [24]. RNN, especially the long-term–short-term memory (LSTM) model, has shown its ability to solve many challenging problems involving time series data [25–27]. In the RNN learning framework, learning the appropriate representations of sequence data is essential to realizing artificial intelligence. It perfectly matches the requirement of change detection, i.e., the model must be able to extract and detect changes from the change information representation reliably. Therefore, the RNN model is a representative method for learning reliable difference information and providing memorability of change detection in temporal remote sensing data [28–30]. It might be possible to directly detect the change information with migration ability of land cover by designing a dynamic change rule with LSTM and extracting the change information characteristics of time series data.

In this study, a lake boundary change prediction model based on U-Net-LSTM is proposed. The model analyzes lake boundary change trends and extracts multi-level change characteristics using the U-Net network. After U-Net, the LSTM module is added to enhance the prediction model utilizing both recent input data and previous data storage and forgetting. The ability of LSTMs to capture temporal patterns enables combined networks to adapt to changing conditions or learn time-varying interactions. This work discusses the fundamental procedure for predicting lake boundary in remote sensing images using U-Net-LSTM. The ensemble of LSTMs helps to improve the overall accuracy and robustness of the model by capturing the spatial and temporal nuances in the data, resulting in more precise predictions. After that, a comparison with a model that just uses U-Net [31] is completed to confirm the usefulness of the LSTM module. Multiple sets of control experiments were conducted to compare and analyze the impact of the timing length of a single input image on the prediction results on two data sets, verifying the effectiveness of the proposed method.

2. Dataset

2.1. Data Source

Lake Urmia is a salty lake in northwestern Iran, close to Turkey, between East Azerbaijan Province and West Azerbaijan Province. The coordinates of Lake Urmia are located at 37°42′ north latitude and 45°22′ east longitude. Figure 1 shows the geographical location of Lake Urmia in Iran (red box).

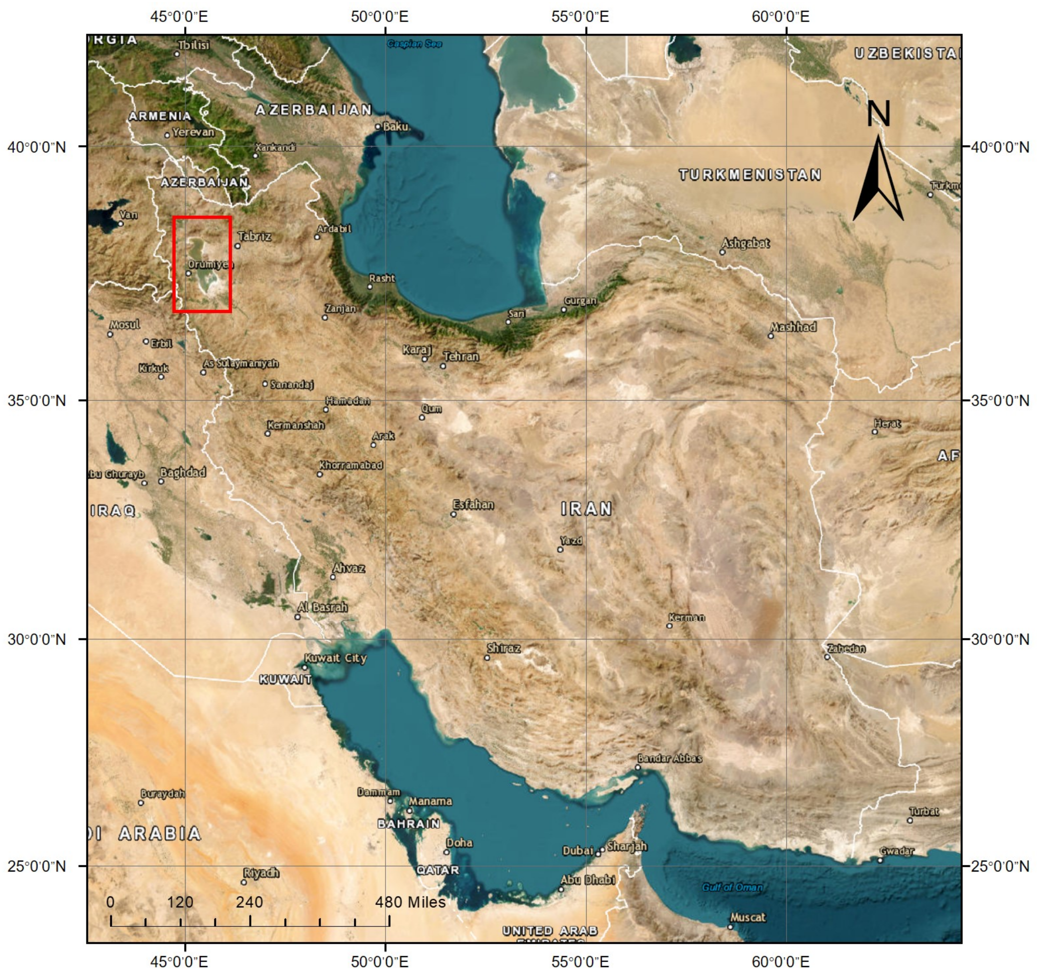
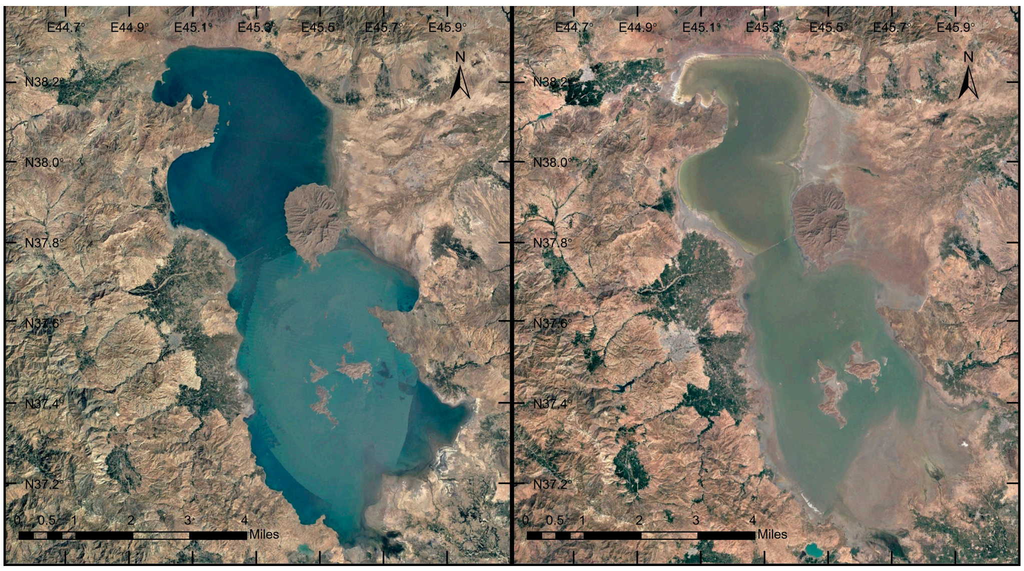


Figure 1. The location of Lake Urmia in Iran (red box).

Lake Urmia is the largest lake in Iran and the second largest saltwater lake on the Earth. Its surface area was about 5180 square kilometers, its longest point was about 140.01 km, and its widest was about 54.72 km. However, in recent years, the lake has faced significant environmental challenges, including shrinking water levels and increased salinity, making its conservation an important concern. This study selects Lake Urmia as the research area and uses its historical remote sensing images to study the rules of its boundary changes. Figure 2 shows satellite images from 1984 and 2020.

By observing historical data from 1984 to 2020, the boundaries of the lake show a trend of shrinking year by year [32]. Figure 3 presents comparative diagram of boundary reduction using images from 1984, 2000, and 2020.



(**a**) (**b**)

Figure 2. Remote sensing satellite images of Lake Urmia in different periods. (**a**) 1984; (**b**) 2020 (from Google Earth Pro Version 7.3).

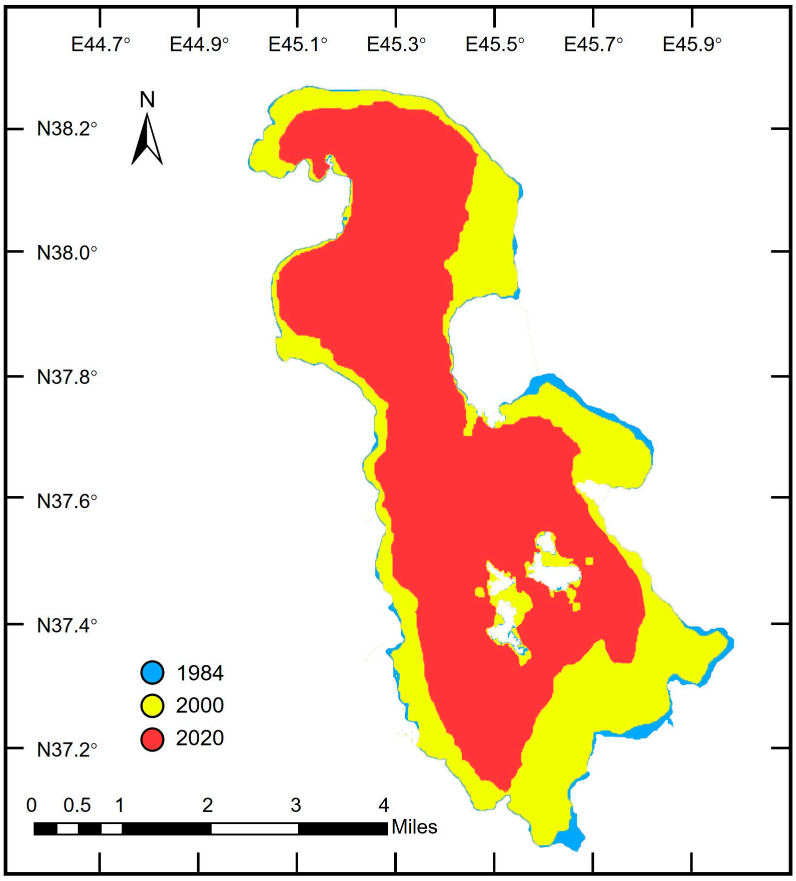


Figure 3. Lake Urmia boundary reduction diagram, 1984–2000–2020.

This article obtained yearly images of the selected area from Google Earth Pro Version 7.3 from the years 1996 to 2014 and used them as analysis data. Each image was taken on December 30 of that year.

This study constructed two data sets for experiments. Dataset 1 is a panoramic remote sensing image of Lake Urmia to verify the effectiveness of this model at predicting the overall evolution trend of cover. Dataset 2 is a remote sensing image of part of Lake Urmia, used to test the ability of this model to capture changes in regional details. The data set information is shown in Table 1 [31].

Table 1. Dataset information.

Name	Time Period	Spatial Coverage	Image Size	Number
Dataset 1	1996–2014	Lat: 37°00′ N to 38°15′ N Lon: 46°10′ E to 44°50′ E	560 × 640	19
Dataset 2	2000–2014	Lat: 38°05′ N to 38°15′ N Lon: 45°30′ E to 44°20′ E	1280 × 560	15

2.2. Data Preprocessing

The data preprocessing process is shown in Figure 4.

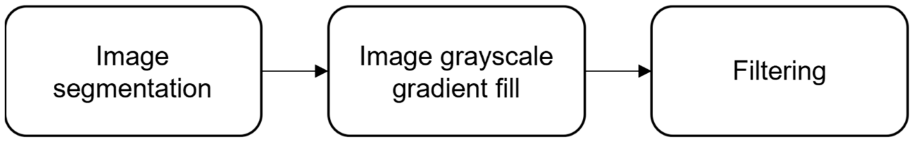


Figure 4. Data preprocessing process.

2.2.1. Image Segmentation

Image segmentation and binarization are the key steps in image preprocessing [33]. The main purpose of image segmentation is to divide an image into several segments, making it easier to extract information from the segmentation and reducing the computational cost of processing large images. This technique is conducive to displaying the evolution process of lake boundaries and is of great help to the subsequent model in learning the evolution process of coverage [34]. Binarization reduces the complexity of an image by converting it into a binary form, making it easier to analyze and process.

In this study, taking dataset 2 as an example, 15 remote sensing images were segmented, and the lake was successfully distinguished from other areas, which provided help for the visualization of land coverage and the subsequent extraction and prediction of land coverage change features. Figure 5 shows the result of the remote sensing images from 2000 after segmentation and binarization.

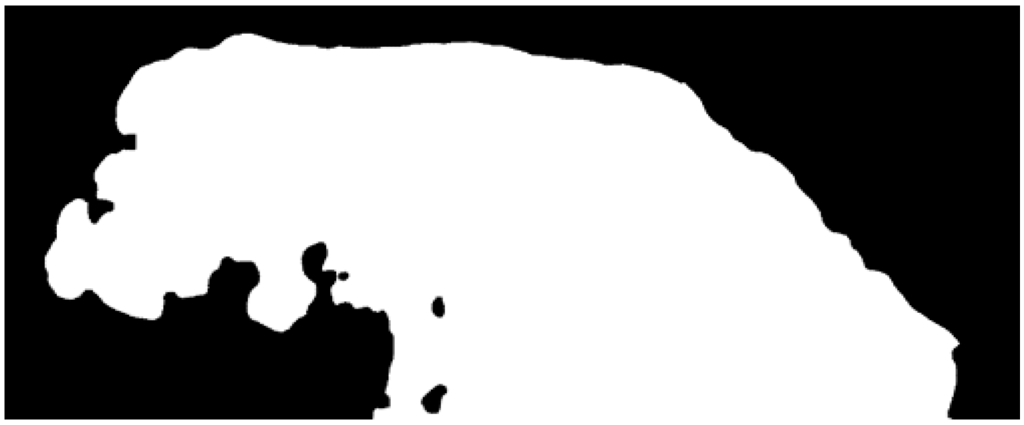


Figure 5. Remote sensing image of Lake Umir in 2000 after image segmentation.

2.2.2. Image Grayscale Gradient Filling

To facilitate the spatial transformation module in extracting the evolution characteristics of the cover according to the movement of the image pixels, this paper proposes a novel method of grayscale gradual filling based on the above binarized image [35]. The steps of this method are as follows:

Step one: The cover boundary in the image is gradually and smoothly shrunk to the center of the cover by multiple consecutive erosion operations, and the intermediate results after each contraction are saved. The formula for the erosion of image A using the erosion operator B is given in Equation (1).

$$A \ominus B = \{z|(B)_z \subset A\}, \tag{1}$$

where $\ominus$ is the erosion operation, and the erosion operator B is also called the structuring element, which can generally be identified with a cross, rectangle, circle, or oval. In this section, the images from the year 2000 in the second dataset segmented by the above image are taken as an example, and the circular operator with radius 3 is used to erode the images 50 times in turn.

Step 2: According to the order of erosion, the difference image of two adjacent shrinkage images is obtained by subtracting sequentially, that is, the pixel value corresponding to the result of the ith corrosion and the result of the i + 1th corrosion is subtracted. The point set with the difference value of 255 is the pixel set deleted in the i + 1 corrosion, and the result is displayed as the lake boundary.

Step 3: The obtained difference image set is sorted in turn, and the gray level is filled according to the linear change from front to back, changing the pixel value of the point shown as white in the difference image. The set of pixels in the first difference image has the highest gray value and is close to white, while the set of pixels in the last difference image has the lowest gray value and is close to black.

Figure 6 shows the result of Figure 5 after grayscale gradient filling.

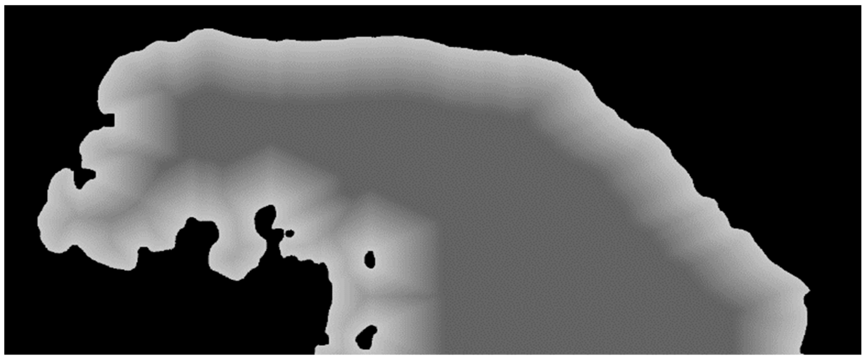


Figure 6. Remote sensing image of Lake Umir in 2000 after grayscale gradient filling.

2.2.3. Filtering

The filtering operation can smooth the gray level change of the image after the grayscale gradients filling. Such a trend is beneficial to the model to obtain the optimal solution in the gradient descent algorithm. Common filtering algorithms are Gaussian filtering, mean filtering, and median filtering. Grayscale images often contain continuous variations in intensity, and Gaussian filtering can preserve these variations while reducing noise and unwanted details. This can result in a smoother and more natural appearance [36]. To better smooth the gray level change of the image, the two-dimensional Gaussian filter is selected for filtering in this paper. The formula for calculating the weight of the neighboring pixels in the two-dimensional Gaussian filter is shown in Equation (2) [37]:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}, \tag{2}$$

The size of the Gaussian kernel is 5 × 5. Figure 7 shows the results of Figure 6 after filtering.

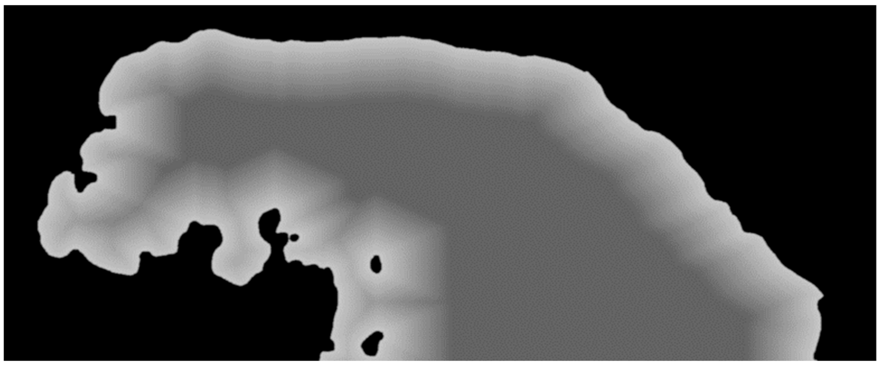


Figure 7. Remote sensing image of Lake Umir in 2000 after filtering.

3. Method

This study proposes a dynamic evolution prediction model for land cover in remote sensing images based on the U-Net model [31] with an added LSTM module to capture the multi-dimensional change characteristics of time-series remote sensing data. The model is depicted in Figure 8 and consists of sequence image preprocessing, sequence image evolution feature extraction, evolution field fitting, spatial variation, and coverage prediction.

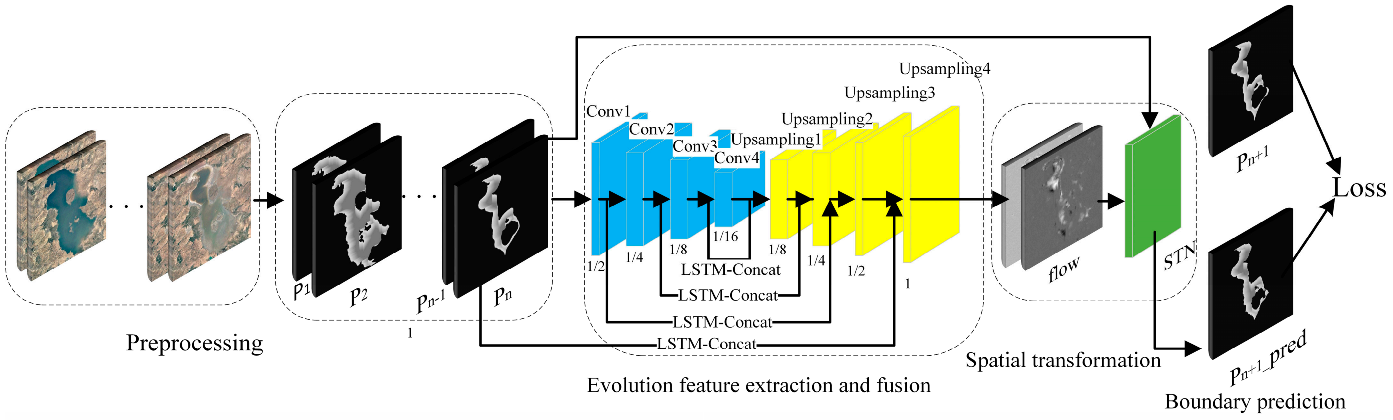


Figure 8. Schematic diagram of U-Net-LSTM model prediction process.

The specific steps of this model are as follows:

(1) Sequence image preprocessing. In this section, N consecutive time series images $[P_1, P_2 \ldots\ldots P_{n-1}, P_n]$ are preprocessed, filled with gray levels in sequence, and input into the evolution feature extraction module in chronological order.

(2) Evolutionary feature extraction. After each convolutional layer, an LSTM module is added, and the input is the preprocessed N consecutive time series images in step (1). For example, suppose the image width is W and the height is H, and the time-series images are spliced into a tensor of size $N \times W \times H$ according to the time dimension. LSTM saves the intermediate output results of N time steps, and it is possible to return the output results of all time steps or the last time step.

(3) In the feature fusion stage, the Nth or last time step of the multi-scale time series characteristics passed through the LSTM module is selected, and it is combined with the output of the symmetrically upsampled layer. The resulting output is then forwarded to the next upsampled layer. The width and height of the output image of the final upsampled layer are the same as that of a single input image. Additionally, the first M sequential images are selected and sent to step (4).

(4) Evolution field. The evolution field estimates the displacement in two directions from the image at the Nth time step to the image at the $N+1$-th time step, and the size is $W \times H \times 2$.

(5) Spatial transformation. Each pixel of the N-th time series image is spatially transformed based on the evolving field of step (4) to fit the time series image of the $N+1$th time step, namely P_{n+1}_pred.

(6) Cover prediction. Calculate the similarity loss between the P_{n+1}_pred output in step (5) and the actual image P_{n+1}, and add it to the regular term of the evolution field gradient to form the loss function, which is based on backpropagation Gradient descent loss and update network parameters.

Step (1) of the above was described in detail in the data preprocessing section, so it will not be addressed here. Steps (2), (3), and (4) are the evolution feature extraction and fusion module. Step (5) is the space transformation module. Step (6) is the loss function. This article will introduce these three modules in detail.

3.1. Evolutionary Feature Extraction and Fusion Module

This section presents the use of LSTM to build a multi-layer U-Net-LSTM network model for remote sensing image time series coverage prediction. Figure 9 illustrates the schematic diagram of the deep stack U-Net-LSTM model, which can be divided into the following components: input, encoder, LSTM timing processing, decoder, and evolution field.

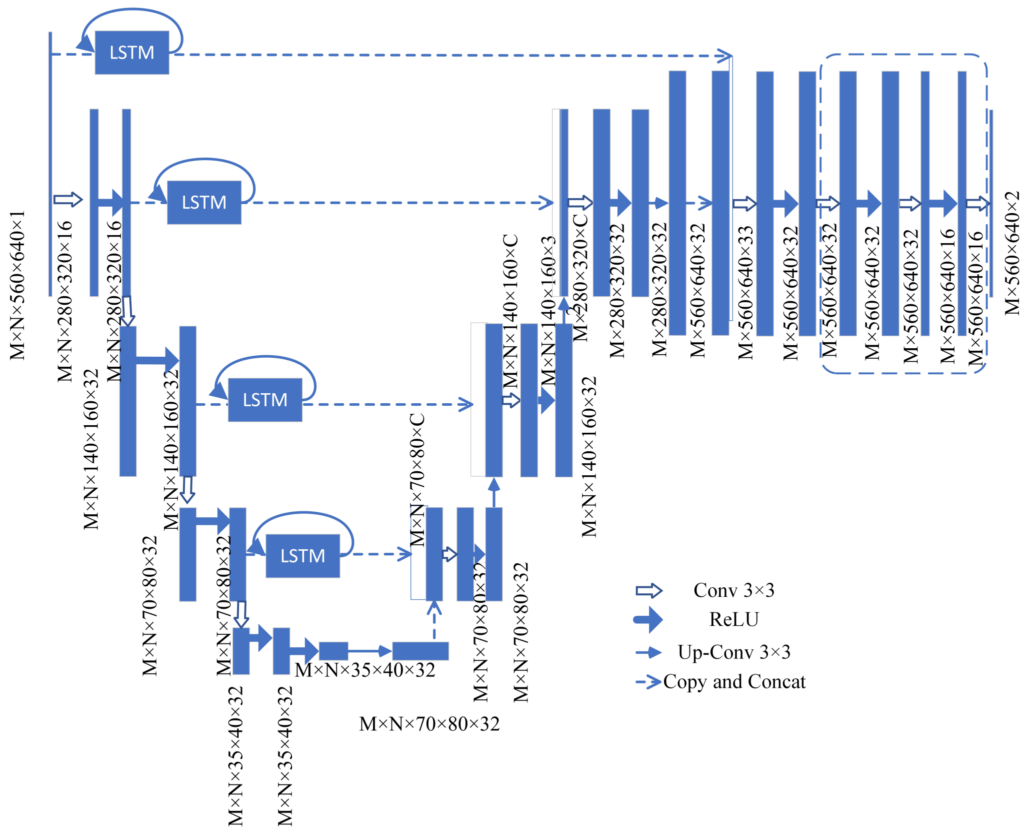


Figure 9. Structural diagram of deep trestle U-Net-LSTM model.

(1) Input. Suppose there are N+M+1 images $[P_1, P_2 \ldots\ldots P_{n+m}, P_{n+m+1}]$ in the dataset. Take the image in size $W \times H$; for example, N sequential images of the same size are spliced in chronological order $N \times W \times H \times 1$. The N + 1 image is taken as the ground truth. According to this, the data set can be divided into m + 1 samples, i.e., $[P_1, P_2 \ldots\ldots P_n/P_{n+1}]$, $[P_2, P_3 \ldots\ldots P_{n+1}/P_{n+2}]$, ……, $[P_m, P_{m+1} \ldots\ldots P_{m+n-1}/P_{m+n}]$, $[P_{m+1}, P_{m+2} \ldots\ldots P_{m+n}/P_{m+n+1}]$.

The first M samples are training sets, which can be spliced into $M \times N \times W \times H \times 1$ tensor direct input model.

(2) Encoder. Taking M = 1 as an example, this section adds the LSTM module behind each convolution layer and the activation function of the encoder. The size of the convolution core is 3 × 3, and the number of channels is 16, 32, 32, and 32 in turn. The input of each LSTM is the multi-scale feature of N time steps, and the output can be selected as the output sequence of N time steps, with the size of $N \times W \times H \times 1$, or as only the output of the last time step, size $1 \times W \times H \times 1$. The jump connection's output of LSTM is connected with the upper sampling layer's output with the same U-Net dimension.

(3) Decoder. Taking LSTM only outputs the result of the last time step as an example; the decoding stage includes four upper sampling layers, convolution layers, and activation functions, the convolution kernel size is 3 × 3, and the number of channels is 32, 32, 32, and 32 in turn. Before the first convolution operation of the encoder, the input image is connected to the upper sampling layer of the last layer through LSTM. After up sampling, the output of the last convolution layer of the encoder passes through LSTM as the input of the subsequent up sampling layer. The number of hidden layer nodes and activation functions can be adjusted according to the dimension and size of input features, and the time series length determines the time step parameters of LSTM.

(4) Evolution field. Before evolving the field, multiple convolutional layers and activation function layers can be appropriately added to extract the timing sequence information further. The evolution field predicts the displacement of the image from the $N - th$ time step to the $N + 1 - th$ time step in the two directions by learning the evolution information of the previous N time steps. The size is $W \times H \times 2$.

(5) Initialization mode. To improve accuracy, the general batch normalization (BN) layer is added between the convolution layer and the activation function layer of the encoder and decoder. This scheme optimizes the adverse effects caused by the initialization mode. The method performs Gaussian normalization and linear transformation on the output value of the convolution layer before activating the function. The execution flow of the BN algorithm is shown in Algorithm 1.

Algorithm 1: BN algorithm

```
Input: Sample values of a mini-batch: $X = \{x_1, \ldots, x_m\}$,
        BN layer parameters to be learned: β, γ.
Output: $y_i = \{BN_{\gamma,\beta}(x_i)\}$.
Calculate the sample mean $\mu_X$ according to Equation (3);
Calculate the sample variance $\sigma_X^2$ according to Equation (4);
The sample values are normalized according to Equation (5);
$\gamma$ and $\beta$ are introduced to translate and scale the sample values, and the sample values are
updated according to Equation (6) during backpropagation.
```

$$\mu_X = \frac{1}{m}\sum\nolimits_{i=1}^{m} x_i, \tag{3}$$

$$\sigma_X^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_X)^2 \tag{4}$$

$$\hat{x}_i = \frac{x_i - \mu_X}{\sqrt{\sigma_X^2}} \tag{5}$$

$$y_i = \gamma \hat{x}_i + \beta = BN_{\gamma,\beta}(x_i) \tag{6}$$

During training, the mean and variance are calculated from the current batch. When testing, the mean and variance of the previous training should be used instead of the new batch of data during the test. Figure 10 shows the output value distribution diagram of the random normal distribution initialization combined with the BN algorithm.

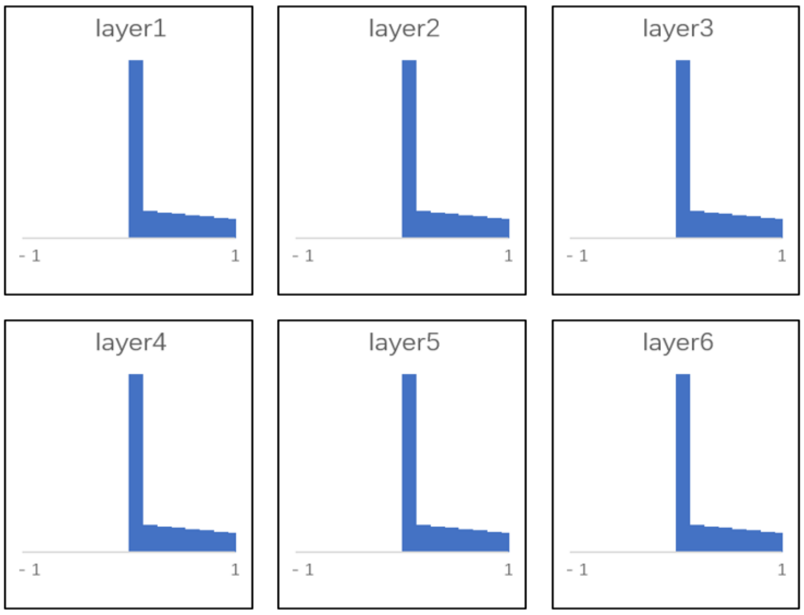


Figure 10. Random normal distribution initialization combined with BN output value.

3.2. Space Transformation Module

The spatial transformation module receives the output from the evolution feature module, which is the evolution field, and outputs the predicted image *z_pred* after spatial transformation of the second image y in the time series image pair x, y. The model adjusts the network parameters by minimizing $y \cdot \varphi$, that is, the difference between *z_pred* and z. In order to use the gradient descent method to minimize the loss function, this paper constructs a differentiable operation based on a spatial transformation network to calculate it.

For each pixel p of *z_pred*, the sub-pixel position of the pixel before the evolution field transformation is calculated as $p' = p + \varphi(p)$. Because pixels are only defined in integer, the model uses the values of eight neighboring pixels for linear interpolation:

$$y\cdot\varphi = \sum_{q\in Z(p')} Z(q)\prod_{d\in\{m,n\}}\left(1 - \left|p'_d - q_d\right|\right), \tag{7}$$

where $Z(p')$ is the set of domain pixels of pixel $p\prime$, d is the dimension of the image, including the m and n dimensions, and $\cdot$ represents the spatial transformation operation.

3.3. Loss Function

The model is trained with a differentiable loss function. The loss function of this model is composed of the following parts:

(1) Similarity loss l_{sim} is shown in Equation (8):

$$MSE(P_n, \widetilde{P}_n) = \frac{1}{|\omega|}\sum_{(u,v)\in\omega}\left\|P_n(u,v) - \widetilde{P}_n(u,v)\right\|^2, \tag{8}$$

(2) Term of penalty $l_{smooth}(\varnothing)$ is shown in Equation (9):

$$l_{smooth}(\varnothing) = \sum_{(u,v)\in\omega}\|\nabla_2\varnothing(u,v)\|^2 + \|\nabla_2\varnothing(u,v)\|^2, \tag{9}$$

where $\widetilde{P}_n = P_n \cdot \varnothing$ is the nth image predicted by the model, where $\cdot$ represents the spatial transformation operation, P_n is the real nth image, w is the set of image pixels, and (u, v) is the pixel coordinate of any point. l_{sim} is the mean square value of pixel values at all pixels of the predicted image and the real image, and $l_{smooth}(\varphi)$ is used to smooth the cover evolution field.

The final loss function is shown in Equation (10):

$$loss = l_{sim} + \alpha l_{smooth}(\varnothing), \tag{10}$$

where α is the regularization parameter.

4. Experiment and Results

4.1. Experiment Setting

In this section, Dataset 1, a total of 19 images $[P_1, P_2 \ldots\ldots P_{18}, P_{19}]$ are divided into 20-T samples according to time step T, as the experiment 1 and experiment 3 data sets Col_1.

For example, when T = 10, there are a total of 10 samples, which are $[P_1, P_2 \cdots P_8, P_9/P_{10}]$, $[P_2, P_3 \cdots P_9, P_{10}/P_{11}] \ldots\ldots [P_{10}, P_{11} \cdots P_{17}, P_{18}/P_{19}]$. The first nine images in each sample are input data, and the last image is ground truth. The first nine samples are used as the training set, and the last group is used as the test set to evaluate their performance.

Dataset 2 has a total of 15 images, which are similarly split into 15-T samples as the data sets Col_2 of Experiment 2 and Experiment 4.

Experiment 1 used the U-Net-LSTM model proposed in this article to predict the time series images of the first 4 years, 7 years, 10 years, 14 years, and 18 years in Col_1.

Experiment 2 used the U-Net-LSTM model proposed in this article to predict the time series images of the first 5–7 years in Col_2.

Experiments 3 and 4 compared and analyzed the influence of the time series length of a single input image on the prediction results of the data sets of Col_1 and Col_2.

The experimental environment and parameter settings are shown in Table 2.

Table 2. Experimental environment parameters.

Component	Specification
CPU	Intel® Core™ i9-10900K
RAM	32 G
GPU	GeForce RTX 2080 $*$ 2
Operating System	Ubuntu 18.04 LTS
Development Language	Python 3.6
Framework	PyTorch 1.5.0
Training Method	Batch Gradient Descent (BGD)
Learning Rate	1×10^{-4} (0.0001)
Training Epochs	10,000

4.2. Evaluating Indicators

This study uses ACC, MSE, MAE, and DICE as the experimental evaluation indicators. ACC is the accuracy of coverage prediction, MSE is the mean square error, and MAE is the average absolute error. DICE is an ensemble similarity measurement index, usually used to calculate the similarity of two samples. The image difference between the predicted and actual results is evaluated to visually show the gap.

MSE is calculated as shown in Equation (8). ACC, MAE, and $DICE$ are calculated as shown in Equations (11)–(13).

$$ACC = \frac{N_{correct}}{N_{total}}, \tag{11}$$

$$MAE(P_n, \widetilde{P}_n) = \sqrt{\frac{1}{|\omega|}\sum\nolimits_{(u,v)\in\omega} \left\|P_n(u,v) - \widetilde{P}_n(u,v)\right\|^2}, \tag{12}$$

$$DICE(T, P) = \frac{|T_1 \cap P_1|}{(|P_1| + |T_2|)/2} = \frac{2N_{TP}}{N_{FP} + 2N_{TP} + 2N_{FN}}, \tag{13}$$

4.3. LSTM Effectiveness Analysis in Experiments 1 and 2

The model proposed in this section adds an LSTM module. Compared with a separate U-net model, the U-Net-LSTM model can input time series data at multiple time steps simultaneously. In order to verify the effectiveness of the LSTM module, the following is a comparison of the prediction results of the U-Net model and the U-Net-LSTM model after training using T time series data. The U-Net model inputs two images each time, and the U-Net-LSTM model can input image time steps each time. In Experiments 1 and 2, to verify the LSTM module's effectiveness, the control variable input step length is the same.

4.3.1. Experiment 1

In experiment 1, the training set is 20-T samples of T time steps before 2014 in Col_1, and the test set is [2012, 2013 | 2014]. Figure 11 and Table 3 show the comparison of experimental results for T = 8, 12, and 18.

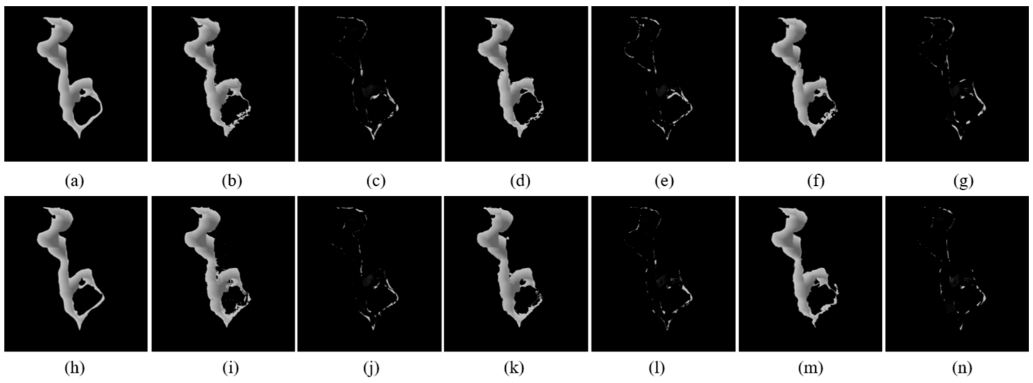


Figure 11. Compares the prediction results of the U-Net model and U-Net-LSTM model. (**a**) Image from 2013; (**h**) Image from 2014; (**b**) Predictions by U-Net model at T = 8; (**i**) Predictions by U-Net-LSTM model at T = 8; (**d**) Predictions by U-Net model at T = 12; (**k**) Predictions by U-Net-LSTM model at T = 12; (**f**) Predictions by U-Net model at T = 18; (**m**) Predictions by U-Net-LSTM model at T = 18; (**c**) Difference image between (**h**,**b**); (**j**) Difference image between (**h**,**i**); (**e**) Difference image between (**h**,**d**); (**l**) Difference image between (**h**,**k**); (**g**) Difference image between (**h**,**f**); (**n**) Difference image between (**h**,**m**).

Table 3. U-Net and U-Net-LSTM different time step prediction results in Col_1.

Predict 2014	ACC	MSE	DICE
8 years U-Net [31]	0.8855	410.01	0.9537
8 years U-Net-LSTM	0.8876	392.37	0.9581
12 years U-Net [31]	0.8879	402.41	0.9574
12 years U-Net-LSTM	0.8929	373.38	0.9599
18 years U-Net [31]	0.8900	377.20	0.9576
18 years U-Net-LSTM	0.8943	234.79	0.9608

These images and difference images are used to compare the predictions made by the U-Net and U-Net-LSTM models at different time steps with the actual images from 2013 and 2014, as well as to assess the accuracy of the predictions. The results show that, after adding LSTM, the model predicts fewer difference image areas and has higher accuracy.

4.3.2. Experiment 2

In Experiment 2, the training set consists of T time steps before 2014 in Col_2, while the test set is [2012, 2013 | 2014]. Figures 12 and 13 show the comparison of experimental results for T = 6, 7, and 8. The first row (10a, 10b, 10c, 11a, 11b, and 11c) of Figures 10 and 11 uses the remote sensing image from 2003 to 2008 as the training set and the remote sensing image from 2009 as the ground truth for comparison. The second row (10d, 10e, 10f, 11d, 11e, and 11f) compares the prediction results using the first seven time steps, with the training set consisting of the remote sensing image from 2003 to 2009, and the ground truth being the remote sensing image from 2010. The third row (10g, 10h, 10i, 11g, 11h, and 11i) compares the prediction results using the first eight time steps, with the training set consisting of the remote sensing image from 2003 to 2010 and the ground truth being the remote sensing image from 2011.

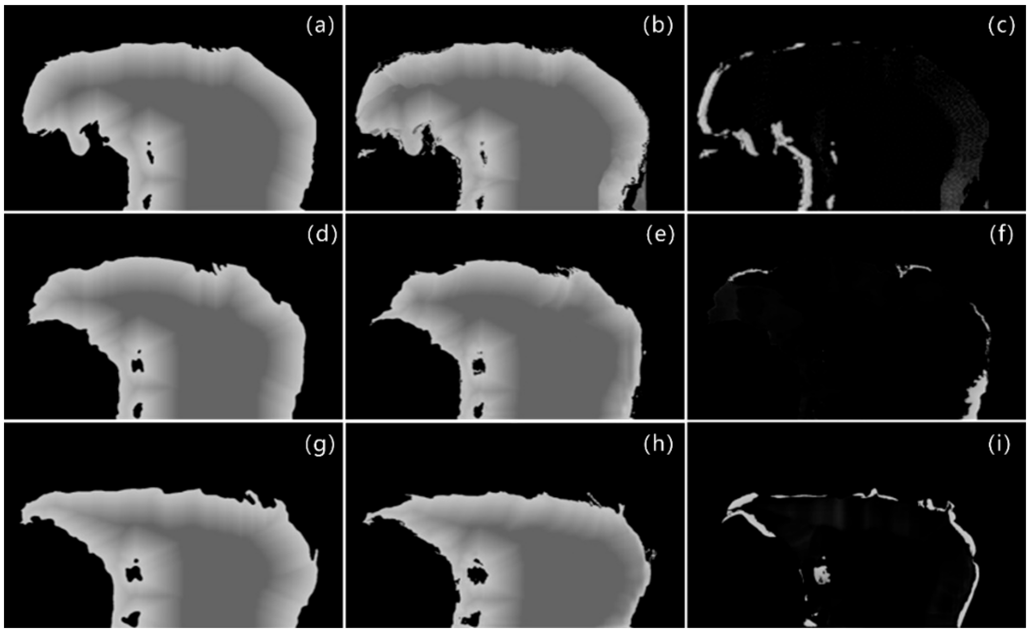


Figure 12. The comparison of prediction results of the U-Net-LSTM Model. (**a**) 2009 original image; (**b**) Prediction result for 2009; (**c**) Difference image of (**a**,**b**); (**d**) 2010 original image; (**e**) Prediction result for 2010; (**f**) Difference image of (**d**,**e**); (**g**) 2011 original image; (**h**) Prediction result for 2011; (**i**) Difference image of (**g**,**h**).

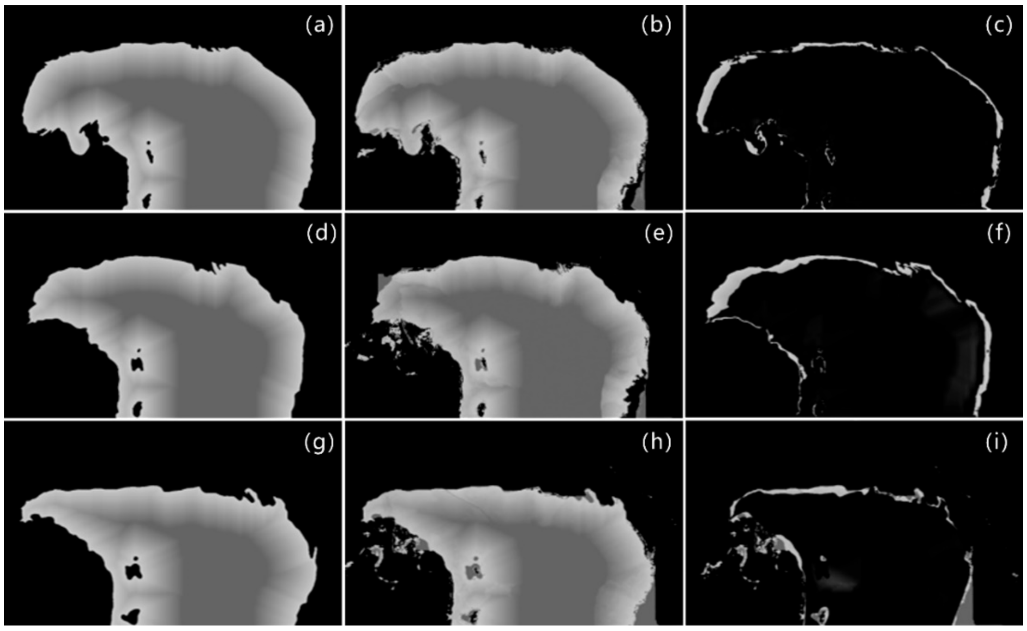


Figure 13. The comparison of prediction results of the U-Net Model. (**a**) 2009 original image; (**b**) Prediction result for 2009; (**c**) Difference image of (**a**,**b**); (**d**) 2010 original image; (**e**) Prediction result for 2010; (**f**) Difference image of (**d**,**e**); (**g**) 2011 original image; (**h**) Prediction result for 2011; (**i**) Difference image of (**g**,**h**).

Table 4 shows the prediction evaluation indicators for time steps T = 6, 7, and 8.

Table 4. U-Net and U-Net-LSTM different time step prediction results in Col_2.

Predict 2014	ACC	MSE	DICE
6 years U-Net [31]	0.7425	1487.35	0.9268
6 years U-Net-LSTM	0.7646	1405.79	0.9297
7 years U-Net [31]	0.7810	1475.24	0.9574
7 years U-Net-LSTM	0.7893	1555.08	0.9673
8 years U-Net [31]	0.7873	1169.98	0.9680
8 years U-Net-LSTM	0.8022	918.82	0.9743

4.4. Time Series Length and Sample Number Analysis

The number of input samples of the U-Net-LSTM model is M, and the time step of each sample is N. To test the impact of the number of samples and the length of the training data on the prediction results. In Experiments 3 and 4, T is unchanged. In such a case, by changing the input sample time step N, the number of samples M will also change.

4.4.1. Experiment 3

Experiment 3 uses the first 18 time steps of 2014 in Col_1, and the test set is [2012, 2013 | 2014]. Figure 14 and Table 5 compare the experimental results when N = 18, 13, 8, 2, and the corresponding M = 1, 5, 10, and 16, respectively.

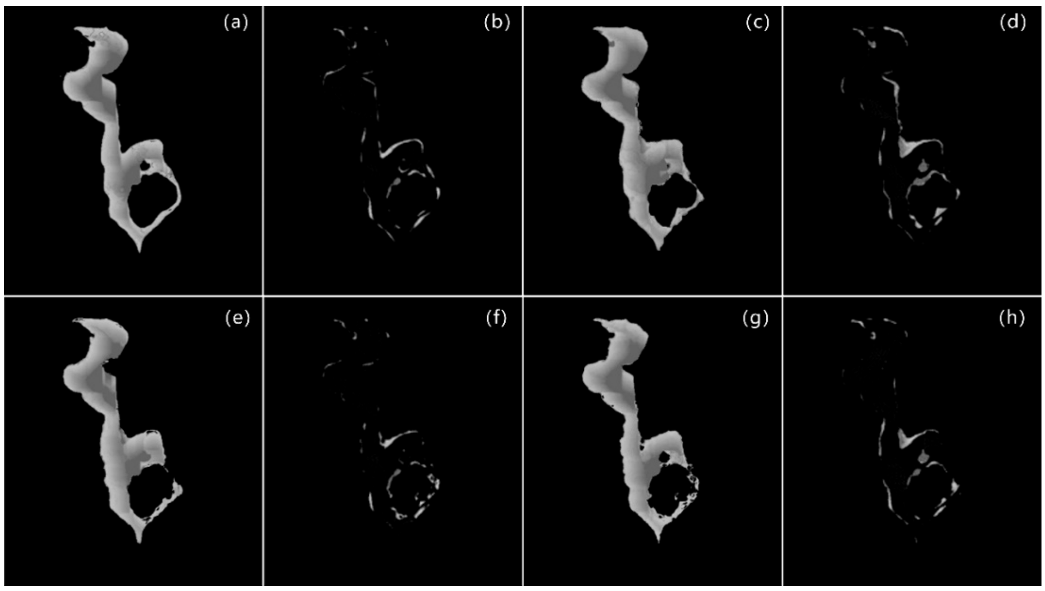


Figure 14. The comparison of U-Net-LSTM results using different sample length and sample. (**a**) 2014 prediction result when N = 18; (**b**) Difference image between (**a**) and original image; (**c**) 2014 prediction result when N = 13; (**d**) Difference image between (**c**) and original image; (**e**) 2014 prediction result when N = 8; (**f**) Difference image between (**e**) and original image; (**g**) 2014 prediction result when N = 2; (**h**) Difference image between (**g**) and original image.

Table 5. Results for different combinations of sample length and number of samples in Col_1.

Sample Length/Number of Samples	ACC	MSE	DICE
18/1	0.8855	377.20	0.9595
13/5	0.8906	369.67	0.9627
8/10	0.8922	271.11	0.9709
2/16	0.8913	323.35	0.9630

When N = 18, the prediction result of the model is almost the same as that of the covering image in 2013, indicating that the migration ability of the model is not strong at this time. As N decreases, the prediction result of the model gradually changes and approaches the actual image of 2014, indicating that the model's prediction ability has been improved. However, when N is reduced to a certain value, the model's prediction accuracy gradually decreases, and some situations, such as the covering boundary, cannot be closed, and too much noise occurs.

4.4.2. Experiment 4

Experiment 4 uses the data from 2004 to 2013 in Col_2 as the training set, and the data in 2014 are ground truth. Figure 15 lists the prediction results when the sample length is 5, 6, 7, and 8. Table 6 shows their respective evaluation index values.

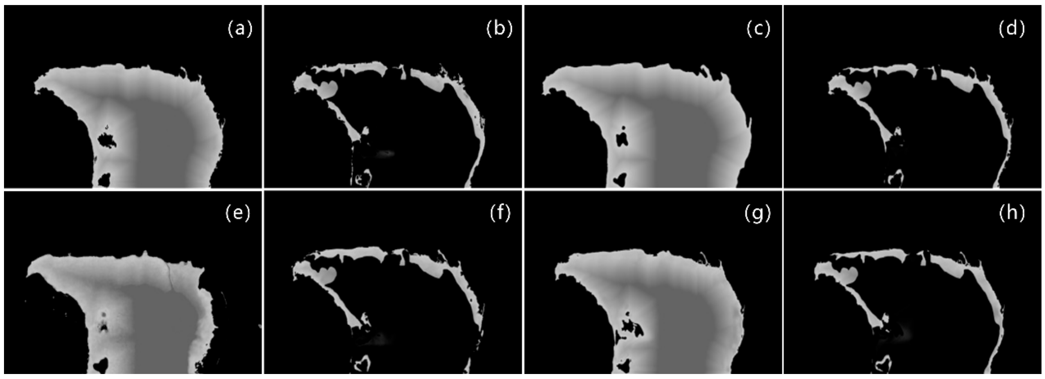


Figure 15. The comparison of U-Net-LSTM results using different sample lengths and samples. (**a**) 2014 prediction result when N = 5; (**b**) Difference image between (**a**) and original image; (**c**) 2014 prediction result when N = 6; (**d**) Difference image between (**c**) and original image; (**e**) 2014 prediction result when N = 7; (**f**) Difference image between (**e**) and original image; (**g**) 2014 prediction result when N = 8; (**h**) Difference image between (**g**) and original image.

Table 6. Results for different combinations of sample length and number of samples in Col_2.

Sample Length/Number of Samples	ACC	MSE	DICE
5/5	0.7914	1645.64	0.9329
6/4	0.7930	1323.35	0.9360
7/3	0.7948	1293.69	0.9407
8/2	0.7930	1411.99	0.9347

As N increases, the prediction accuracy is improved. However, when N increases to a certain extent, the number of samples is too small, and the model is easily over-fitted, making it difficult for the learned change rules to be applied to brand-new data, resulting in reduced prediction accuracy.

The results show that the number and length of training samples will have a greater impact on the prediction results, which is consistent with the results of Experiment 3. As the sample length N increases, the accuracy of predicting coverage details is improved. However, when N increases to a certain extent, the number of samples is too small, which leads to insufficient mobility of the model. As a result, the prediction accuracy is reduced when the model is applied to brand-new data. Comparing Experiments 2 and 4 with Experiments 1 and 3, the model's accuracy in predicting changes in detail coverage is slightly lower than that of predicting overall coverage.

5. Discussion

The results of Experiments 1 and 2, which verified the effectiveness of the model, showed that, after adding LSTM, the model has higher prediction accuracy than the U-Net model. When t is consistent, the contours predicted by the U-Net-LSTM model are closer to the coverage of the original image compared to the U-Net model. Both the accuracy index of the prediction results and the similarity measurement index verify the effectiveness of the introduced LSTM module.

Experiments 3 and 4 show that the choice of N and M greatly affects the prediction of coverage changes by the model. However, there is no uniform standard or obvious rule for the selected scheme, which must be tested repeatedly according to the actual situation. When T is the same, the control variable method cannot be used to control the consistency of any variable in M and N. As the length N of each sample increases, the number of samples M must decrease accordingly. From the experimental results, we can see that the prediction accuracy improves initially as N decreases. However, when N is reduced to a certain level, the model cannot extract more temporal information, the role of the LSTM module is not fully utilized, and the prediction accuracy is reduced. As the sample length N increases, the accuracy of the predicted coverage details improves. However, when N increases to a certain extent, the number of samples is too small, resulting in insufficient mobility for the model. As a result, the prediction accuracy decreases when the model is applied to completely new data. Therefore, it is necessary to choose the appropriate M and N for experiments to achieve the best prediction.

Overall, the U-Net-LSTM model proposed in this study can realize differential memory and forgetting of historical data and different levels of utilization of current input data. This method can customize a prediction model of boundary changes for natural areas, such as lakes, with a small data set and provide valuable reference results. Using LSTM for prediction can extract the state changes of land cover from remote sensing image time series and generate migration change rules, which is of great significance for natural lake boundary prediction. However, this study also has notable limitations. It solely relies on remote sensing imagery as the sole basis for predicting changes in lake boundaries and the results lack interpretability. In future research, we aspire to follow the lead of previous scholars [32] by incorporating additional factors, such as temperature, precipitation, population changes, and industrial shifts, to explore a more comprehensive set of variables related to lake changes. By leveraging multimodal data, we aim to construct more precise and real-time prediction models that can provide superior solutions to water body protection issues.

6. Conclusions

U-Net performs well in an encoder–decoder architecture by capturing spatial features and hierarchical relationships and is able to detect changes in lake boundaries, but it lacks sensitivity to time-series data. To address this limitation, this paper proposes a novel method combining U-Net and LSTM in an end-to-end lake boundary prediction model. The U-Net network is used to extract multi-level change features and analyze land cover change trends. The LSTM module is introduced after U-Net, using historical data storage and forgetting and current input data to optimize the prediction model. This method enables the model to automatically fit the trend of time series data and mine the spatio-temporal fusion information of lake boundary changes. To further optimize the prediction model, various model optimization algorithms are used in this paper. These methods help to ensure the validity of the model and prevent overfitting, allowing continuous optimization through experimental methods. Compared with traditional prediction models such as Markov and cellular automata, this model can simultaneously detect and predict lake boundary rate changes, provide more comprehensive and accurate predictions, and provide a valuable tool for natural lake management planning and decision-making. The introduction of the LSTM module allows the extraction of state changes in land cover from a time series of remote sensing images. LSTM introduces memory cells that enable the network to retain and selectively update information over long sequences. New movement rules can then be generated, which is crucial for accurate lake boundary predictions. By integrating LSTM with U-Net, the fusion of spatio-temporal information can be achieved. This is useful in tasks that involve both spatial structure and temporal dependencies. This paper demonstrates in detail the feasibility and advantages of introducing the LSTM module h by comparing it with the existing U-net-STN model.

Author Contributions: Conceptualization, L.Y., T.L., W.Z. and L.W.; methodology, T.L. and L.Y.; software, T.L., X.L. and Z.Y.; validation, J.T. and S.L.; formal analysis, Z.Y., S.L. and X.L.; investigation, J.T. and Z.Y.; resources, X.L. and Z.Y.; data curation, T.L. and S.L.; writing—original draft preparation, L.Y., S.L. and W.Z.; writing—review and editing, L.Y., L.W. and W.Z.; visualization, L.Y. and J.T.; supervision, L.W.; project administration, W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Supported by Sichuan Science and Technology Program (2023YFSY0026, 2023YFH0004).

Data Availability Statement: The data and source code are available at https://github.com/Alienwei/Lake-Urmia-boundary-prediction (accessed on 20 September 2023).

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

References

1. Alam, A.; Bhat, M.S.; Maheen, M. Using Landsat satellite data for assessing the land use and land cover change in Kashmir valley. *GeoJournal* **2020**, *85*, 1529–1543. [CrossRef]
2. Arsanjani, J.J. Characterizing, monitoring, and simulating land cover dynamics using GlobeLand30: A case study from 2000 to 2030. *J. Environ. Manag.* **2018**, *214*, 66–75. [CrossRef] [PubMed]
3. Zhang, X.; Han, L.; Han, L.; Zhu, L. How well do deep learning-based methods for land cover classification and object detection perform on high resolution remote sensing imagery? *Remote Sens.* **2020**, *12*, 417. [CrossRef]
4. Qiao, B.; Zhu, L.; Yang, R. Temporal-spatial differences in lake water storage changes and their links to climate change throughout the Tibetan Plateau. *Remote Sens. Environ.* **2019**, *222*, 232–243. [CrossRef]
5. Zhang, G.; Yao, T.; Chen, W.; Zheng, G.; Shum, C.; Yang, K.; Piao, S.; Sheng, Y.; Yi, S.; Li, J. Regional differences of lake evolution across China during 1960s–2015 and its natural and anthropogenic causes. *Remote Sens. Environ.* **2019**, *221*, 386–404. [CrossRef]
6. Hui, F.; Xu, B.; Huang, H.; Yu, Q.; Gong, P. Modelling spatial-temporal change of Poyang Lake using multitemporal Landsat imagery. *Int. J. Remote Sens.* **2008**, *29*, 5767–5784. [CrossRef]
7. Pooja, M.; Thomas, S.; Udayasurya, U.; Praveej, P.; Minu, S. Assessment of Soil Erosion in Karamana Watershed by RUSLE Model Using Remote Sensing and GIS. In *Innovative Trends in Hydrological and Environmental Systems: Select Proceedings of ITHES 2021*; Springer: Berlin/Heidelberg, Germany, 2022; p. 219.
8. Wan, W.; Xiao, P.; Feng, X.; Li, H.; Ma, R.; Duan, H.; Zhao, L. Monitoring lake changes of Qinghai-Tibetan Plateau over the past 30 years using satellite remote sensing data. *Chin. Sci. Bull.* **2014**, *59*, 1021–1035. [CrossRef]
9. Symeonakis, E. Modelling land cover change in a Mediterranean environment using Random Forests and a multi-layer neural network model. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 5464–5466.
10. Zhang, C.; Wei, S.; Ji, S.; Lu, M. Detecting large-scale urban land cover changes from very high resolution remote sensing images using CNN-based classification. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 189. [CrossRef]
11. Chowdhury, M.; Hasan, M.E.; Abdullah-Al-Mamun, M. Land use/land cover change assessment of Halda watershed using remote sensing and GIS. *Egypt. J. Remote Sens. Space Sci.* **2020**, *23*, 63–75. [CrossRef]
12. Talukdar, S.; Singha, P.; Mahato, S.; Pal, S.; Liou, Y.-A.; Rahman, A. Land-use land-cover classification by machine learning classifiers for satellite observations—A review. *Remote Sens.* **2020**, *12*, 1135. [CrossRef]
13. Wang, W.; Chen, Y.; Ghamisi, P. Transferring CNN with adaptive learning for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5533918. [CrossRef]
14. Zhao, Z.; Li, J.; Luo, Z.; Li, J.; Chen, C. Remote sensing image scene classification based on an enhanced attention module. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 1926–1930. [CrossRef]
15. Liu, F.; Jiao, L.; Tang, X.; Yang, S.; Ma, W.; Hou, B. Local restricted convolutional neural network for change detection in polarimetric SAR images. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 818–833. [CrossRef] [PubMed]
16. Li, W.; Dong, R.; Fu, H.; Wang, J.; Yu, L.; Gong, P. Integrating Google Earth imagery with Landsat data to improve 30-m resolution land cover mapping. *Remote Sens. Environ.* **2020**, *237*, 111563. [CrossRef]
17. Giang, T.L.; Dang, K.B.; Le, Q.T.; Nguyen, V.G.; Tong, S.S.; Pham, V.-M. U-Net convolutional networks for mining land cover classification based on high-resolution UAV imagery. *IEEE Access* **2020**, *8*, 186257–186273. [CrossRef]

18. Zhang, C.; Yue, P.; Tapete, D.; Shangguan, B.; Wang, M.; Wu, Z. A multi-level context-guided classification method with object-based convolutional neural network for land cover classification using very high resolution remote sensing images. *Int. J. Appl. Earth Obs. Geoinf. IJAEO* **2020**, *88*, 102086. [CrossRef]
19. Wang, D.; Chen, X.; Jiang, M.; Du, S.; Xu, B.; Wang, J. ADS-Net: An Attention-Based deeply supervised network for remote sensing image change detection. *Int. J. Appl. Earth Obs. Geoinf. IJAEO* **2021**, *101*, 102348.
20. Shi, W.; Zhang, M.; Zhang, R.; Chen, S.; Zhan, Z. Change detection based on artificial intelligence: State-of-the-art and challenges. *Remote Sens.* **2020**, *12*, 1688. [CrossRef]
21. Law, S.; Seresinhe, C.I.; Shen, Y.; Gutierrez-Roig, M. Street-Frontage-Net: Urban image classification using deep convolutional neural networks. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 681–707. [CrossRef]
22. Tao, C.; Qi, J.; Lu, W.; Wang, H.; Li, H. Remote sensing image scene classification with self-supervised paradigm under limited labeled samples. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 8004005. [CrossRef]
23. Xu, X.; Chen, Y.; Zhang, J.; Chen, Y.; Anandhan, P.; Manickam, A. A novel approach for scene classification from remote sensing images using deep learning methods. *Eur. J. Remote Sens.* **2021**, *54*, 383–395. [CrossRef]
24. Chen, Z.; Wang, C.; Li, J.; Xie, N.; Han, Y.; Du, J. Reconstruction bias U-Net for road extraction from optical remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2284–2294. [CrossRef]
25. Zhou, Y.N.; Wang, S.; Wu, T.; Feng, L.; Wu, W.; Luo, J.; Zhang, X.; Yan, N.N. For-backward LSTM-based missing data reconstruction for time-series Landsat images. *GISci. Remote Sens.* **2022**, *59*, 410–430. [CrossRef]
26. Zhou, Y.n.; Luo, J.; Feng, L.; Yang, Y.; Chen, Y.; Wu, W. Long-short-term-memory-based crop classification using high-resolution optical images and multi-temporal SAR data. *GISci. Remote Sens.* **2019**, *56*, 1170–1191. [CrossRef]
27. You, H.; Tian, S.; Yu, L.; Lv, Y. Pixel-level remote sensing image recognition based on bidirectional word vectors. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 1281–1293. [CrossRef]
28. Zhao, R.; Shi, Z.; Zou, Z. High-resolution remote sensing image captioning based on structured attention. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5603814. [CrossRef]
29. Mao, W.; Jiao, L.; Wang, W.; Wang, J.; Tong, X.; Zhao, S. A hybrid integrated deep learning model for predicting various air pollutants. *GISci. Remote Sens.* **2021**, *58*, 1395–1412. [CrossRef]
30. Lobry, S.; Marcos, D.; Murray, J.; Tuia, D. RSVQA: Visual question answering for remote sensing data. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8555–8566. [CrossRef]
31. Yin, L.; Wang, L.; Li, T.; Lu, S.; Yin, Z.; Liu, X.; Li, X.; Zheng, W. U-Net-STN: A Novel End-to-End Lake Boundary Prediction Model. *Land* **2023**, *12*, 1602. [CrossRef]
32. Taheri Dehkordi, A.; Valadan Zoej, M.J.; Ghasemi, H.; Ghaderpour, E.; Hassan, Q.K. A New Clustering Method to Generate Training Samples for Supervised Monitoring of Long-Term Water Surface Dynamics Using Landsat Data through Google Earth Engine. *Sustainability* **2022**, *14*, 8046. [CrossRef]
33. Kotaridis, I.; Lazaridou, M. Remote sensing image segmentation advances: A meta-analysis. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 309–322. [CrossRef]
34. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [CrossRef]
35. Wu, Y.; Li, Q. The algorithm of watershed color image segmentation based on morphological gradient. *Sensors* **2022**, *22*, 8202. [CrossRef]
36. Ding, F.; Shi, Y.; Zhu, G.; Shi, Y.-Q. Real-time estimation for the parameters of Gaussian filtering via deep learning. *J. Real-Time Image Process.* **2020**, *17*, 17–27. [CrossRef]
37. Chen, G.; Pei, Q.; Kamruzzaman, M.M. Remote sensing image quality evaluation based on deep support value learning networks. *Signal Process. Image Commun.* **2020**, *83*, 115783. [CrossRef]