Article

LaeNet: A Novel Lightweight Multitask CNN for Automatically Extracting Lake Area and Shoreline from Remote Sensing Images

Wei Liu ^{1,2} , Xingyu Chen ¹, Jiangjun Ran ^{1,3,*} , Lin Liu ⁴ , Qiang Wang ⁵ , Linyang Xin ¹ and Gang Li ^{6,7}

¹ Department of Earth and Space Sciences, Southern University of Science and Technology, Shenzhen 518055, China; 3161050@nyist.edu.cn (W.L.); 11930407@mail.sustech.edu.cn (X.C.); 12031229@mail.sustech.edu.cn (L.X.)

² School of Computer and Software, Nanyang Institute of Technology, Nanyang 473004, China

³ Shenzhen Key Laboratory of Deep Offshore Oil and Gas Exploration Technology, Southern University of Science and Technology, Shenzhen 518055, China

⁴ Earth System Science Programme, Faculty of Science, The Chinese University of Hong Kong, Hong Kong, China; liulin@cuhk.edu.hk

⁵ Vultus AB, Lilla Fiskaregatan 19, 22222 Lund, Sweden; qiang.wang@vultus.se

⁶ Guangzhou Marine Geological Survey, Guangzhou 510075, China; ligang@gmgs.cn

⁷ Southern Marine Science and Engineering Guangdong Laboratory, Guangzhou 511458, China

* Correspondence: ranjj@sustech.edu.cn; Tel.: +86-0755-88018644

Abstract: Variations of lake area and shoreline can indicate hydrological and climatic changes effectively. Accordingly, how to automatically and simultaneously extract lake area and shoreline from remote sensing images attracts our attention. In this paper, we formulate lake area and shoreline extraction as a multitask learning problem. Different from existing models that take the deep and complex network architecture as the backbone to extract feature maps, we present LaeNet—a novel end-to-end lightweight multitask fully CNN with no-downsampling to automatically extract lake area and shoreline from remote sensing images. Landsat-8 images over Selenco and the vicinity in the Tibetan Plateau are utilized to train and evaluate our model. Experimental results over the testing image patches achieve an Accuracy of 0.9962, Precision of 0.9912, Recall of 0.9982, F1-score of 0.9941, and mIoU of 0.9879, which align with the mainstream semantic segmentation models (UNet, DeepLabV3+, etc.) or even better. Especially, the running time of each epoch and the size of our model are only 6 s and 0.047 megabytes, which achieve a significant reduction compared to the other models. Finally, we conducted fieldwork to collect the in-situ shoreline position for one typical part of lake Selenco, in order to further evaluate the performance of our model. The validation indicates high accuracy in our results (DRMSE: 30.84 m, DMAE: 22.49 m, DSTD: 21.11 m), only about one pixel deviation for Landsat-8 images. LaeNet can be expanded potentially to the tasks of area segmentation and edge extraction in other application fields.

Keywords: lightweight CNN; lake area segmentation; lake shoreline extraction; spatial gradient map

Citation: Liu, W.; Chen, X.; Ran, J.; Liu, L.; Wang, Q.; Xin, L.; Li, G. LaeNet: A Novel Lightweight Multitask CNN for Automatically Extracting Lake Area and Shoreline from Remote Sensing Images. *Remote Sens.* **2021**, *13*, 56. <http://dx.doi.org/10.3390/rs13010056>

Received: 16 November 2020

Accepted: 22 December 2020

Published: 25 December 2020

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

Copyright: © 2020 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

Lakes are an important component in the global terrestrial ecosystem and represent a key water resource for human beings. The expansion or shrinkage of lakes are affected by regional as well as global conformation and climate changes [1]. Hence, the lake area and shoreline variation can be taken as an indicator to monitor the current climate fluctuation and predict future climate change [2]. For example, since the 1990s, some lakes in the Tibetan Plateau have expanded significantly [3]. This is induced by: (a) increasing precipitation rates [4,5]; (b) melting snows or glaciers [6]; and (c) decreasing lake evaporation [7]. Both expansion and shrinkage of lakes have a great influence on the regional environment and inhabitants. At present, frequent measurements of lake

dynamic changes by using remote sensing images are necessary for the conservation and utilization of the lakes as well as to understand climate change [8]. It is also a practical and effective method for numerous lake-changed applications [9] including lake water storage changes [10], lake level changes [11–13], etc. For the lake change detections, accurate extraction of lake area and shoreline is a crucial step.

Many studies have focused on lake area segmentation or shoreline extraction by using remote sensing images which can observe a wide area periodically in a few days. For example, a scheme to investigate the spatial distribution of the lake area and temporal changes of glacial lakes shoreline using manual digitization of Landsat data by GIS technologies were presented in [14,15]. Although traditional manual digitization methods guarantee consistent examination and quality control to some extent, it requires large amount of domain knowledge, time, as well as cost while the accuracy might be not high. Apart from manual digitization methods, many techniques can automatically or semiautomatically derive the lake area or shoreline from satellite images as technology advances. The dominant methods are threshold approaches [16–23] which are based on the water index, such as Normalized Difference Water Index (NDWI) [24] and Modified Normalized Difference Water Index (MNDWI) [25]. They are captured by the normalized relationship of the appropriate bands. Such threshold methods are easy to calculate and less time consuming, but the magnitude of the errors varies significantly. This is because the appropriate thresholds are difficult to be configured when water and land are mixed in pixels.

To address the variation in the optimal thresholds and reduce errors, many machine learning algorithms [26–34] have been used for lake water body extraction. For example, Random Forest (RF) [26], Support Vector Machine (SVM) [27–29], and Artificial Neural Networks (ANNs) [30–32] were extensively applied in lake level prediction and lake quality mapping. Recently, deep learning [35–41] has attracted great attention in the field of remote sensing image processing, especially deep Convolutional Neural Network (CNN)-based semantic segmentation [42–45]. Such semantic segmentation methods [46–49] aim to assign pixel-level componential labels for remote sensing image, which means each pixel of the image is classified as a componential category. However, such studies apply deep learning algorithms to perform one task first, then perform another task by using postprocessing tools. It is unfavorable to get the mutual feedback which is beneficial information in multitask deep learning models, thereby, this can cause models to anchor in local minima. To overcome this, Liu et al. [50] proposed an end-to-end semantic segmentation network based on UNet [51] which can be reinforced with spatial boundary information for remote sensing images. Waldner et al. [52] have proposed a multitask semantic segmentation with ResUNet [53] which includes the extent of fields, the field boundaries, and the distance to the closest boundary. Despite the success of such end-to-end multitask learning for obtaining global optimum, the backbone is normally a deep and complex network architecture which generates numbers of parameters; this requires larger training time for each epoch and a huge amount of space. Meanwhile, training the deep network on a small remote sensing image dataset tends to result in the overfitting problem. Additionally, some deep learning algorithms (such as DeepLabV3+ [54,55]) are not feasible to be applied for some multispectral (including more than three bands) remote sensing observations since the band information will be lost when reducing the original bands to three bands.

To overcome the aforementioned difficulties, we propose a novel end-to-end lightweight multitask no-downsampling fully convolutional neural Network to segment area and extract edge from remote sensing images simultaneously. We name it the **Lightweight area and edge Network (LaeNet)** and its architecture is illustrated in Figure 1. Specifically, we firstly pack several no-downsampling and multichannel fully convolutional layers with *ReLU* activation function as a feature extractor to learn high-level feature map from multiband remote sensing imagery. Then, another no-downsampling and single-channel convolutional layer with *Sigmoid* activation function is applied to predict lake area and nonarea (land), thereby achieving area segmentation. Based on this, the difference between area segmentation and its spatial gradient is derived as the corresponding predictive

edge. The edge label is derived from the mask label by the Canny edge detection operator in OpenCV. This does not require extra manual labeling, but still can provide accurate spatial edge details and reduce semantic feature ambiguity at area segmentation stage. We train our model on Landsat-8 images near the lake Selenco and validate on the ones from the lake Selenco in the Tibetan Plateau. Extensive experiments demonstrate that our model can achieve comparable or even better result in both the performance and the complexity, compared to mainstream semantic segmentation model (UNet, DeepLabV3+, etc.). Meanwhile, our model can generate lake shoreline very well. In order to further validate the effectiveness, we compare the lake shoreline results of our model with the in-situ measurements over lake Selenco captured by GPS.

Figure 1. The architecture of our proposed LaeNet model for automatically and simultaneously extracting lake area and shoreline. The wheat boxes represent multichannel convolutional layers for semantic feature extraction. The khaki box is a single-channel convolutional layer for area segmentation. The red box indicates the gradient of the segmented area by using the max pooling operation.

Our main contributions can be summarized as follows:

- (1) An end-to-end lightweight multitask no-downsampling fully convolutional neural network (LaeNet) is proposed to extract lake area and shoreline automatically from multiband remote sensing images simultaneously.
- (2) The edge is extracted by computing the difference between the area segmentation map and its spatial gradient, where the spatial gradient is produced by commonly used max-pooling operation. This does not increase the complexity of our LaeNet model.
- (3) We assess the capability of the proposed LaeNet model on a real-world multiband remote sensing images. Extensive experimental results demonstrate the superiority of our model in extracting lake area compared with mainstream deep image semantic segmentation models (UNet, DeepLabV3+, etc.), especially in the cost time of each epoch and the model size. Moreover, the in-situ observed data collected from GPS further validate the effectiveness of the model.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the study area and data. Then, we elaborate our proposed lightweight end-to-end multitask no-downsampling fully CNN (LaeNet) in Section 3. The experimental settings, evaluation criteria, analysis, and assessment of the model are reported in Section 4.

In Section 5, we discuss the application of the model on different attention mechanisms as well as various satellite sensors. Finally, conclusion is summarized in Section 6.

2. Study Area and Data

2.1. Study Area

The Selinco region in the Tibetan Plateau (Figure 2) is selected as the study area since it is sensitive to climate change with numbers of lakes scattered over this area. It covers an area of 113,781 km² with the longitude ranges from 87°13'19"E to 91°00'16"E and latitude ranges from 30°08'04"N to 32°48'19"N. The average altitude of this area is 4542 m a.s.l. and numerous lakes are scattered in the surrounding area. The annual mean precipitation of this area is about 315 mm with the monsoon season covers from May to September [56]. The mean annual temperature, the average annual sunshine duration and annual panevaporation are 0.7 °C, 2950 h and 2080 mm, respectively [57].

Over this region, the 18 lake regions (indicated by the white boxes in Figure 2) around lake Selinco are chosen as training data. The Selinco lake region (indicated by the red box in Figure 2) is selected as testing data to evaluate the effectiveness and practicability of the extractor since it covers a large area and sensitive to climate change, this is proved by 26% expansion over the past 40 years [58].

Figure 2. Study area in central Tibet by using Landsat-8 RGB composition with the training and test areas are outlined by the white and red boxes, respectively.

2.2. Landsat Images

The Landsat-8 OLI/TIRS images are acquired from the USGS (<https://glovis.usgs.gov>). In total, 7 scenes of cloudless and clear images are downloaded. Each scene has blue, green, red, near-infrared, and short-wave infrared-1 bands. The wavelength and spatial resolution of each band are shown in the Landsat-8 part of Table 1. After atmospheric correction, water reflectivity image is produced. The bit depth is unified into 8 bits. The binary label is derived from the specific band of the corresponding Landsat-8 image by using the single-band threshold method, as shown in Figure 3. According to the input requirement of the LaeNet model, we need to further subdivide the lake-images and the corresponded binary labels into patches by setting a patch size as 512 pixels with an overlap of 300 pixels. The overlap ensures training data augmentation and consistent results among the adjacent patches. Ultimately, 121 subdivided testing image patches are obtained by using the image

of Selneco lake region. Similarly, we have 542 subdivided training image patches by utilizing the images of 18 lake regions near Selneco lake region.

Table 1. Band parameters of Landsat-5 and Landsat-8 used in this study.

Landsat-5			Landsat-8		
BandName	WaveLength (μm)	Resolution (m)	BandName	WaveLength (μm)	Resolution (m)
B1(Blue)	0.45–0.52	30	B2(Blue)	0.45–0.51	30
B2(Green)	0.52–0.60	30	B3(Green)	0.53–0.59	30
B3(Red)	0.63–0.69	30	B4(Red)	0.64–0.67	30
B4(NIR)	0.76–0.90	30	B5(NIR)	0.85–0.88	30
B5(SWIR)	1.55–1.75	30	B6(SWIR)	1.57–1.65	30

Figure 3. Example of the subsets used for training and testing data. (a,b) An example of training subset and its corresponding binary label image. (c,d) An example of testing subset and its corresponding binary label image.

2.3. Field-Measured Lakeshore from GPS

To further validate the LaeNet model more accurately, we conducted fieldwork to collect the in-situ observation of one typical part of the lake Selinco (indicated as the yellow box of Figure 4) by handheld GPS. We used SOUTH S86 GPS RTK Surveying instrument for GPS data collection and set it to automatically record positions at equal intervals of 5 s. Before surveying, the RTK was kept still to obtain the static accuracy (STD = E: 1.5 cm, N: 0.5 cm, U: 4.8 cm). Then, we walked along the lake Silingco from 88°36'13"E, 31°44'12"N to 88°37'13"E, 31°42'50"N on 20 August 2020. Since we only focused on the 2-D horizontal results, the antenna height of RTK was ignored in our measurement. During the data postprocess stage, the modified RTKLIB software [59] was utilized. We chose positioning mode as PPP-Kinematic for calculation after adding the final precise ephemeris, which were downloaded from IGS Analysis Center of the Helmholtz-Centre Potsdam-GFZ German Research Centre for Geosciences (<ftp://ftp.gfz-potsdam.de/GNSS/products/final/>). Finally, we obtained approximately 5 km long vector line data with geographic coordinates, shown as the orange line in Figure 4.

Figure 4. Lake shoreline measured with GPS (orange line) and the measured area (yellow box).

3. LaeNet Model

The LaeNet model feeds with a multiband input image, then generates two binary grayscale segmentation area and edge probability maps of the corresponding input image. It mainly consists of three components: (1) extracting semantic feature maps from multiband remote sensing images via several multichannel no-downsampling fully CNN layers with *ReLU* activation function; (2) segmenting area by a single-channel no-downsampling fully CNN layer with *Sigmoid* activation function; (3) computing edge using the difference between the area and the corresponding gradient map. The detailed procedure is illustrated as follows.

3.1. Semantic Feature Extraction

It is known that the convolutional layers can extract different kinds of semantic features from images. In order to extract semantic features of multiband remote sensing image and keep their subtle structures, we also employ a multichannel fully convolutional network including several no-downsampling convolutional layers in this work, which is inspired by matting refinement used three no-downsampling fully convolutional layers

in [60]. Following this, a rectified linear unit (*ReLU*) is added, shown as the wheat boxes in Figure 1. Each convolutional layer can be formulated as:

$$\mathbf{X}^{l+1} = \max(0, \mathbf{W}_{conv}^l * \mathbf{X}^l + \mathbf{b}_{conv}^l) \quad (1)$$

where $\mathbf{W}_{conv}^l$ and $\mathbf{b}_{conv}^l$ represent the filters and bias of the l -th convolutional layer; $*$ denotes the convolution operation, and $\max(0, \cdot)$ implements *ReLU* activation function. Wherein, when $l=0$, $\mathbf{X}^0$ refers to multiband remote sensing image; or $\mathbf{X}^l$ represents the l -th semantic feature map. Specifically, in each convolutional layer, the kernel size is set as 3×3 , the number of convolutional filters is 64, the stride is 1 and the padding is *SAME* type which implements the no-downsampling function. Regarding the number of convolutional layers to be used, it can be derived based on the experimental performance.

3.2. Area Segmentation

The objective of our first task is to segment area from multiband remote sensing images. Based on the obtained high-level semantic feature, we perform a mask predictive layer to predict the lake areas or nonarea (land), resulting in image area segmentation. Specifically, as the khaki box is shown in Figure 1, we apply a single-channel no-downsampling convolutional layer with *Sigmoid* activation function as the mask predictive layer. It can be formulated as follows:

$$\mathbf{M}^{pt} = \sigma(\mathbf{W}_{conv} * \mathbf{X} + \mathbf{b}_{conv}) \quad (2)$$

where $\mathbf{W}_{conv}$, $\mathbf{b}_{conv}$ and $*$ denote the filters, bias and convolution operation, respectively; $\sigma(\cdot)$ implements *sigmoid* activation function; $\mathbf{X}$ is the high-level semantic feature extracted from the last layer of the multiple convolutional layers indicated in Section 3.1; and $\mathbf{M}^{pt}$ is the predictive area. Here, for the convolutional layer, the kernel size is set as 3×3 , the stride is 1, the padding is *SAME* type, and the number of convolutional filters is configured as 1 which produces a binary area segmentation map.

Cross-entropy loss is commonly used in the field of semantic segmentation. For area segmentation, each value of the final predictive layer is corresponding to a binary value (such as lake or land). Hereby, binary cross-entropy loss is introduced for each pixel during the training process in this paper. The formulation of binary cross-entropy loss for two classes (such as lake or land) is as follows:

$$Loss_{area} = -\frac{1}{N} \sum_{i=1}^N (\mathbf{M}_i^{gt} \log(\mathbf{M}_i^{pt}) + (1 - \mathbf{M}_i^{gt}) \log(1 - \mathbf{M}_i^{pt})) \quad (3)$$

where $\mathbf{M}_i^{gt}$ is the pixel value of the mask label (1 for lake and 0 for land) and $\mathbf{M}_i^{pt}$ is the probabilistic estimate value of the predictive layer.

3.3. Edge Extraction

The second task is to identify and extract edge from multiband remote sensing images. When area segmentation is finished by following the procedure in Section 3.2, we can obtain the probability map of the area from predictive mask layer. Motivated by [61], edge can be derived by spatial gradient from the area segmentation. In particular, max-pooling layer is employed to derive spatial gradient with the formulation is indicated as follows:

$$\nabla \mathbf{M} = \maxpooling(\mathbf{M}^{pt}) \quad (4)$$

where ∇ is gradient calculation; $\nabla \mathbf{M}$ is the spatial gradient map of the $\mathbf{M}^{pt}$, shown as the red box in Figure 1; *maxpooling* is a max-pooling layer used to calculate the gradient map of the predictive area $\mathbf{M}^{pt}$. Here, for the max-pooling layer, the kernel size and stride are set to 3 and 1, respectively. The type of padding is *SAME* which implements the no-downsampling function. The simple and general max-pooling layer operation is implemented here because it is available in any deep learning framework (such as Keras,

Tensorflow, Pytorch, etc.). Based on this, we can obtain the edge by computing difference between the probability map of segmentation area and its spatial gradient. In order to make the difference value amenable to train via backpropagation and preserve semantic information of the edge to maximize, *Leaky ReLU* activation function is applied. The computational procedure of the edge can be formulated as follows:

$$E^{pt} = \max(0, M^{pt} - \nabla M) + \alpha \times \min(0, M^{pt} - \nabla M) \quad (5)$$

where E^{pt} is the predictive edge; $\max(0, \cdot) + \alpha \times \min(0, \cdot)$ implements *Leaky ReLU* activation function. Here, the α value is set as 0.2. For clarity, an example of extracting the edge of a 7×7 binary image is given as shown in Figure 5. The red "0" represents the edge of the binary image in Figure 5a. A 3×3 maxpooling operation with the *SAME* type of padding and the stride of 1 (the red rectangle of Figure 5a) is performed on the binary image to derive the spatial gradient map (Figure 5b). Then the edge can be obtained by computing difference value between the binary image (Figure 5a) and its spatial gradient map (Figure 5b). The remaining red "1" denotes the edge of the binary image in Figure 5c.

Figure 5. An example of extracting edge from binary image. (a) A 7×7 binary image using a 3×3 maxpooling with the *SAME* type of padding and the stride of 1. (b) The spatial gradient map of the 7×7 binary image. (c) The 7×7 edge image.

In order to better train the edge extraction task, the edge label is derived from the mask label generated by Canny edge detection operator in OpenCV without extra manual labeling effort. Following prior work proposed by Zhen et al. [61], we also apply Mean Absolute Error Loss to measure inconsistency between predictive edge map and edge label. The loss can be computed as follows:

$$Loss_{edge} = \frac{1}{N} \sum_{i=1}^N |E_i^{pt} - E_i^{gt}| \quad (6)$$

where E_i^{gt} is the pixel value of the edge label (1 for edge and 0 for nonedge) and E_i^{pt} is the probabilistic estimate value of the predictive edge map.

We put the area segmentation loss and edge extraction loss together to train our network via back-propagation. Thus, the total loss function can be formulated as follows:

$$Loss_{total} = Loss_{area} + \lambda Loss_{edge} \quad (7)$$

where λ is hyperparameter to balance the area segmentation loss and edge extraction loss. Here, it is set to 1 in our experiments.

4. Results

In this section, we firstly describe our experimental settings and evaluation criteria. Then, extensive experiments are carried out by various networks to extract lake

area and shoreline. Moreover, the results are analyzed with quantitative and qualitative comparisons.

4.1. Experimental Settings

In the training phase, we randomly shuffled the training data and employed data augmentation for the training subdivided image patches and the corresponding label patches. The data augmentation includes flipping, rotating, and random cropping. All the experiments were implemented with the help of Keras framework (<https://keras.io/>) and conducted on a 64-bit Ubuntu 18.04 Server with Intel(R) Core(TM) i7-6700HQ CPU at 2.60 GHz $\times$ 8 on 16 GB RAM and NVIDIA GeForce GTX 1070 GPU support. The adaptive moment estimation (ADAM) [62] was selected as the optimizer to train the networks. The initial learning rate for the training was set to 10^{-2} . The reduction factor was set to 0.7 and the patience number was set as 15 in the learning rate reduction policy. Training stops if the learning rate reaches the value of 10^{-5} or there is no significant improvement after 50 epochs. The batch size was set as 4. All the trainable parameters of the networks in the kernel of convolutional layers were initialized with a uniform random distribution between $[-1, 1]$. In order to remove the randomness, each experiment was repeated ten times and the average result was recorded.

The testing phase was employed under the same experimental environment with the training phase. Furthermore, the postprocessing (comparing with in-situ results, etc.) was implemented by Python 2.7, ArcGIS 10.5 and ArcGIS Pro 2.5.

4.2. Evaluation Criteria

We mainly evaluate the performance of lake area segmentation task. The main reasons are: (1) lake area segmentation task is the prerequisite of lake shoreline extraction task. (2) lake area segmentation task is an application in semantic segmentation field, which can be compared with other mainstream deep learning models (UNet, DeepLabV3+, etc.). In the testing dataset, the predictive mask is compared with the corresponding mask label to perform assessments at the pixel level. If a pixel correctly detected as lake water, we refer to it as a True Positive. Otherwise, it is a False Positive, which is misclassified as lake water. True Negative denotes the nonlake pixel that is detected. Nevertheless, False Negative means that the nonlake pixel is misidentified as lake water. The metrics can be derived as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (11)$$

$$mIoU = \frac{1}{2} \times \left(\frac{TN}{FP + TN + FN} + \frac{TP}{FP + TP + FN} \right) \quad (12)$$

where TP , TN , FP , and FN are the number of True Positives, True Negatives, False Positives, and False Negatives, respectively. $mIoU$ is the mean intersection over union for two categories generally. Here, it is used to represent the shared regions.

To further demonstrate the prediction ability of the model, Mean Square Error (MSE) and Mean Absolute Error (MAE) are used. They can be formulated as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^N (p_i^{mask} - p_i^{pt})^2 \quad (13)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |P_i^{mask} - P_i^{pt}| \quad (14)$$

where P_i^{mask} is the pixel value of the mask label and P_i^{pt} is the predicted value of the model.

Similarly, the model size, which is mainly decided by the model parameter number, is also a metric for the network model. In our proposed LaeNet model, the parameter number is only generated by the convolutional layers. Specifically, the parameter number of each convolutional layer by calculating weights and biases can be derived as follows:

$$Num_p = C_o \times (K_w \times K_h \times C_i + 1) \quad (15)$$

where C_i and C_o are input and output channel numbers, K_w and K_h are width and height of kernel in the convolutional layer. In addition, the time expenditure of each epoch in the training process is also considered as a metric.

The field observation dataset captured by GPS is recorded as a vector format. However, the edge binary image predicted by our LaeNet model is a raster. To make them comparable, the center coordinate of the predictive edge pixel and the coordinate of the shortest distance point from it to the measured line are converted from raster to vector using the API in ArcGIS Pro 2.5. In order to assess the deviation between the center coordinate and the corresponding coordinate measured by GPS, Distance Root Mean Square Error (*DRMSE*), Distance Mean Absolute Error (*DMAE*), and Distance Standard Deviation (*DSTD*) are adopted. The formulations are as follows:

$$d_i = \sqrt{(x_i^{GPS} - x_i^{pt})^2 + (y_i^{GPS} - y_i^{pt})^2} \quad (16)$$

$$DRMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N d_i^2} \quad (17)$$

$$DMAE = \bar{d} = \frac{1}{N} \sum_{i=1}^N d_i \quad (18)$$

$$DSTD = \sqrt{\frac{1}{N} \sum_{i=1}^N (d_i - \bar{d})^2} \quad (19)$$

where (x_i^{pt}, y_i^{pt}) is the center coordinate of i -th edge pixel in binary image obtained by edge extraction; (x_i^{GPS}, y_i^{GPS}) is the coordinate of the shortest distance point from the center coordinate value of i -th edge pixel to the measured line by GPS; d_i is a distance from (x_i^{pt}, y_i^{pt}) to (x_i^{GPS}, y_i^{GPS}) .

4.3. Performance Comparison on Band Combination and Different CNN Layers

We aim to explore the effects of the multiband information and the semantic feature extraction with the CNN for lake area segmentation. The single-band of Landsat-8 images, different band combination, and the CNN of the various layer numbers for semantic feature extraction are considered. In the Landsat-8 image, B2 as Blue, B3 as Green, B4 as Red, B5 as NIR, and B6 as SWIR-1 were used in our experiments. We summarized the quantitative performance results of utilizing different band combinations and CNN layers in terms of mIoU in Table 2. From this table, we can conclude as follows: (1) In the case of single band, blue and green show the worst performance, while NIR and SWIR-1 perform best. This is caused by the spectral absorption characteristics of lake water. With the increase of spectral wavelength, the stronger the spectral absorption of lake water, this enables a better performance for LaeNet model to segment lake area. (2) Under different band combination situations, bands combination outperforms the single-band due to taking full advantage

of complementary information among different bands. Nevertheless, combination of B5 and B6 outperforms the other bands. One possible explanation is that the worst spectral information limits the overall performance of the LaeNet model. This is a consensus with the famous theory saying “cannikin law”. (3) There is no clear relationship that can be obtained between the performance and number of CNN layers at the semantic feature extraction stage. This implies that number of CNN layers should be selected carefully in multispectral semantic feature for different tasks. With this experiment, we decided to choose B5 and B6 as spectral information and set 1 CNN layer to extract semantic feature of the lake and land in Landsat-8 images in the next experiments, because it performs better than other combinations.

Table 2. Performance based on mIoU in combination of different band and various numbers for semantic feature extraction CNN layers.

Band	1 Layer	2 Layers	3 Layers
B2	0.2331	0.2331	0.2331
B3	0.2331	0.2331	0.2331
B4	0.8971	0.8471	0.8520
B5	0.9832	0.9826	0.9823
B6	0.9855	0.9855	0.9854
B2 + B3	0.8202	0.8224	0.8396
B2 + B4	0.9606	0.9574	0.9636
B2 + B5	0.9837	0.9831	0.9837
B2 + B6	0.9850	0.9768	0.9827
B3 + B4	0.9756	0.9756	0.9755
B3 + B5	0.9838	0.9843	0.9841
B3 + B6	0.9844	0.9838	0.9829
B4 + B5	0.9848	0.9839	0.9845
B4 + B6	0.9848	0.9851	0.9843
B5 + B6	0.9879	0.9875	0.9867
B2 + B3 + B4	0.9747	0.9752	0.9755
B2 + B3 + B5	0.9853	0.9836	0.9820
B2 + B3 + B6	0.9841	0.9846	0.9840
B2 + B4 + B5	0.9853	0.9841	0.9840
B2 + B4 + B6	0.9847	0.9844	0.9840
B2 + B5 + B6	0.9865	0.9862	0.9857
B3 + B4 + B5	0.9849	0.9838	0.9847
B3 + B4 + B6	0.9846	0.9843	0.9828
B3 + B5 + B6	0.9869	0.9854	0.9856
B4 + B5 + B6	0.9873	0.9860	0.9853
B2 + B3 + B4 + B5	0.9847	0.9836	0.9838
B2 + B3 + B4 + B6	0.9838	0.9839	0.9833
B2 + B3 + B5 + B6	0.9858	0.9858	0.9846
B2 + B4 + B5 + B6	0.9866	0.9860	0.9850
B3 + B4 + B5 + B6	0.9861	0.9855	0.9854
ALL	0.9852	0.9857	0.9856

4.4. Performance Comparison with Different Semantic Segmentation Models

To verify the effectiveness and superiority of our proposed LaeNet model, we have compared our results with mainstream semantic segmentation model of DeepLabV3+ [54], AttUNet [63], CloudNet [64], DeepUNet [65], UNet++ [66,67], UNet [51]. The numerical and visual results of the testing on Selinco lake region are demonstrated in Table 3 and Figure 6.

We summarized the quantitative results of the above models and the LaeNet model in Table 3 based on nine quantitative evaluation metrics - Accuracy, Precision, Recall, F1-score, mIoU, MSE, MAE, Time, and Size. We have the following findings from Table 3: (1) DeepLabV3+ only accepts three image bands as input, hence we have chosen the best performance when Band B5, B5, and B6 are combined. However, it has achieved the worst performance as compared to other learning models. This problem may come from Xception65 [68] architecture we used for DeeplabV3+. The Xception65 is pretrained on ImageNet [69] which is a complicated dataset under nature scenes. This may lead an oversmoothing and loss of detail in our task of lake-land segmentation. (2) Similar performance is found among family of UNet variants including the original UNet, AttUNet, CloudNet, DeepUNet, and UNet++. Although AttUNet adds attention mechanism, CloudNet increases Convolution-Identity-Concatenation mapping [70], DeepUNet applies much deeper network architecture and UNet++ is based on nested and dense skip connections, original UNet still outperforms them. This demonstrates that complex network architecture cannot show excellent performance in small remote sensing image dataset and simple application scenarios. (3) Our proposed LaeNet model outperforms all other learning models we tested in terms of Accuracy, Precision, Recall, F1-score, and mIoU, while both the MSE and MAE between labeled pixel value and the corresponding LaeNet model prediction are also the smallest among all algorithms. Although the LaeNet model has only two CNN layers, one for semantic feature extraction of the lake-land and the other for lake area segmentation, it can achieve comparable or even better accuracy. It demonstrates that the lightweight model is more suitable for small datasets and simple pixel-level binary classification problems, such as lake-land segmentation. (4) Regarding the efficiency over the training set, the time usage of each epoch during training of the LaeNet model is only 6 s. It is vastly superior to the other models. Meanwhile, the size of the LaeNet model is only 0.047 megabytes (MB), which is approximately 1/10,000 of the CloudNet model. This is because there are only several CNN layers to extract high-level semantic features in the LaeNet model, resulting in faster convergence ability and smaller model size. This further indicates that the lightweight LaeNet model is superior.

Table 3. Performance on different semantic segmentation models.

Model	Accuracy	Precision	Recall	F1-Score	mIoU	MSE	MAE	Time	Size
DeepLabV3+	0.9711	0.9440	0.9976	0.9701	0.9474	0.0488	0.0536	3'09"	329.2 MB
AttUNet	0.9960	0.9581	0.9628	0.9604	0.9650	0.0033	0.0048	1'55"	95.1 MB
CloudNet	0.9962	0.9579	0.9644	0.9611	0.9652	0.0035	0.0038	2'46"	438.1 MB
DeepUNet	0.9958	0.9898	0.9955	0.9926	0.9857	0.0038	0.0087	0'28"	416.1 MB
UNet++	0.9960	0.9883	0.9986	0.9933	0.9864	0.0035	0.0058	0'53"	27.4 MB
UNet	0.9961	0.9907	0.9975	0.9939	0.9875	0.0036	0.0057	1'52"	372.6 MB
LaeNet	0.9962	0.9912	0.9982	0.9941	0.9879	0.0033	0.0046	0'06"	0.047 MB

Figure 6 shows the visual results of a qualitative and intuitive evaluation for the performance of our proposed LaeNet model. The first row shows the pseudo color remote sensing images combined with Band 5, 5, and 6. The second row shows the corresponding semantic label. Focusing on the 3-th row, it can be seen that the junction of lake and land obtained from DeepLabV3+ are the smoothest, which misses many details and some small areas. In the rows 4–8, it can be observed that the visual appearances are mostly similar and some local results in detail are much better than the one of the DeepLabV3+. The 9-th row shows our model is able to achieve comparable or even better results than the family of UNet Variations, especially for the small land area. In addition, we obtained the lake shoreline while producing the lake segmentation result, as shown in the last row. We have used red rectangle boxes to highlight the shortcomings of different models reflected in Figure 6. These qualitative results were consistent with the quantitative results in Table 3.

Figure 6. Lake segmentation visualization from DeepLabV3+ [54], AttUNet [63], CloudNet [64], DeepUNet [65], UNet++ [66,67], UNet [51] and lake segmentation and shoreline results from our proposed LaeNet model.

4.5. Performance Comparison with Situ Observed Results

In order to further evaluate the practicability and accuracy of the proposed LaeNet model, we collected in-situ GPS trajectory data along the one typical part of lake Selinco on 20 August 2020, as illustrated in Figure 7. The orange line represents the in-situ GPS trajectory, which has irregular edges. This is due to complex terrains and intricate structures of the region. The trajectory line has characteristics of continuous linearity and is slightly close to the inside of the land. This is because there are shoals or marshlands at the shoreline of lake Selinco, so that our surveyors cannot get too close to it. The black raster line represents the lake Selinco shoreline, which is the sawtooth-shaped edge extracted by our LaeNet model from the Landsat-8 image on 19 August 2020. The main principle is to classify each pixel of the remote sensing image as shoreline or nonshoreline. The predictive shoreline (black) is very close to the measured line (orange), which indicates the effectiveness of the LaeNet model. Furthermore, quantitative results were computed between predictive shoreline and measured line in terms of DRMSE, DMAE and DSTD. The center coordinate of predictive lake shoreline pixel can be derived by converting raster to vector with the help of the API in ArcGIS Pro 2.5. Then, the coordinates of the shortest distance can be found from the center coordinate to the measured line. Following Equations (17)–(19), the DRMSE, DMAE, and DSTD are derived as 30.84 m, 22.49 m, and 21.11 m which are about one pixel as the spatial resolution of Landsat-8 is 30 m. The quantitative analysis further indicates that our LaeNet model has high performance and strong practicability.

Figure 7. In-situ GPS trajectory (orange line) and the predictive edge pixels (black line) in the comparison region (yellow box).

5. Discussion

5.1. Applications on Different Attention Mechanisms

The remote sensing images have multiple spectral bands and complex spatial structures, which are capable of providing rich spectral and spatial information. The experimental results in Section 4.1 demonstrate that not all spectral bands are equally informative and predictive. We have tried to introduce spectral attention [71] and channel attention [72,73] under the all bands combination situations to emphasize useful bands and suppress less useful ones, respectively. The spectral attention extracted global spatial information and sent it into the spectral gates (i.e., the *sigmoid* function), then, adaptively recalibrated

spectral bands by applying a global convolutional layer. The number and sizes of the convolutional filters are equal to the number of the spectral bands and size of the input images. For channel attention, it extracted semantic attributes from average-pooled and max-pooled features of the input images by employing a shared two-layer perception firstly. Then, the sigmoid value of the semantic attributes is multiplied with band maps to produce spectral bands extraction. To be noticed, within the two-layer perception, the number of neurons in the first layer should be greater than the number of the spectral bands and it is set to 16 in our experiment, while the number of neurons in the second layer is equal to the number of the spectral bands. Both the attention mechanisms were placed before the input of the LaeNet model. Table 4 implies that the two attention mechanisms can enhance the properties of all bands of the remote sensing images, but the LaeNet model performance is not significantly improved compared with the combination of B5 and B6. This indicates that even attention mechanisms have been adopted, less useful spectral bands in CNNs also may introduce noises and weaken the performance of the LaeNet model.

Table 4. Performance on all bands with different attention mechanisms.

Band	Accuracy	Precision	Recall	F1-Score	mIoU
ALL	0.9959	0.9901	0.9961	0.9930	0.9863
ALL + spectral attention	0.9960	0.9894	0.9971	0.9932	0.9867
ALL + channel attention	0.9960	0.9911	0.9960	0.9935	0.9869
B5+B6	0.9962	0.9912	0.9982	0.9941	0.9879

5.2. Effect of Pixel Tolerance on Different Semantic Segmentation Models

Apart from pixelwise evaluation that we described in Section 4.4, the patchwise evaluation [74–76] is also being introduced generally to overcome data-labeling uncertainty for semantic segmentation tasks. Hereby, we applied a certain pixel tolerance to evaluate our label generated by using the single-band threshold method in Section 2.2. In this experiment, the tolerance margins are set as 0, 1, 2, 3, and 4 pixels away from the label pixels as the True Positive (TP), respectively. The corresponding results are summarized in Table 5 and seen from Table 5, we have the following findings. Firstly, when the margin of the pixel tolerance is 0 pixel, the performance of all model decreased compared with the pixelwise evaluation result as shown in Table 3. The is because the result images predicted by using these models are the grayscale images which can lose some information during manual binary converting postprocessing in the patchwise evaluation. Secondly, in the case of same semantic segmentation model, the model performance decreases with an increasing tolerance margin size and this indicates that utilizing pixel tolerance will not improve the model performance. This further implies that our label produced by using the single-band threshold is quite high. This is because the multispectral image of Landsat-8 has a strong spectral absorption to the lake water; it is conducive to utilize the manual threshold method to generate the corresponding label for scenes one by one in lake-land segmentation task.

5.3. Applications on Images from Different Satellite Sensors

Various satellites sensors can collect different remote sensing images in the same geographical area at different times. Besides the landsat-8 images we described in Section 2.2, we also downloaded Landsat-5 images for the same area and applied our trained LaeNet model to extract the lake area and shoreline accordingly. It turns out that the basic information of lake area and shoreline can be extracted, but some details were missing—the wetlands and the mountain shadows were recognized as lake area, as shown in Figure 8b,c. This is because remote sensing images from Landsat-8 near the lake Selenco were only considered as training samples in LaeNet model, yet the remote sensing images to be segmented and extracted were from Landsat-5. The two kinds of images (Landsat-5 and Landsat-8) have the same resolution and similar wavelength, but there are still slight

differences in the corresponding wavelength, as shown in Table 1. In order to segment lake area and extract shoreline more accurately from Landsat-5 images, we have rebuilt a new training data from Landsat-5 for the study area by utilizing the method provided in Section 2 and retrained the LaeNet model. In this way, we can apply the model to extract lake area and shoreline from the images of the Landsat-5. The results are shown in Figure 8d,e, which has similar performance as our proposed LaeNet model for Landsat-8.

Table 5. Performance evaluation with the 0-, 1-, 2-, 3-, and 4-pixel tolerance.

Model	Tolerance in Pixels	Accuracy	Precision	Recall	F1-Score	mIoU
DeepLabV3+	0	0.9531	0.9901	0.9181	0.9340	0.9166
	1	0.9529	0.9944	0.9183	0.9357	0.9051
	2	0.9526	0.9978	0.9183	0.9374	0.9016
	3	0.9513	0.9989	0.9155	0.9366	0.8958
	4	0.9498	0.9992	0.9118	0.9349	0.8899
AttUNet	0	0.9938	0.9981	0.9531	0.9668	0.9384
	1	0.9931	0.9990	0.9533	0.9672	0.9248
	2	0.9921	0.9996	0.9519	0.9668	0.9194
	3	0.9904	0.9997	0.9482	0.9649	0.9128
	4	0.9885	0.9998	0.9439	0.9626	0.9064
CloudNet	0	0.9933	0.9968	0.9516	0.9622	0.9346
	1	0.9934	0.9983	0.9513	0.9627	0.9204
	2	0.9921	0.9990	0.9493	0.9621	0.9138
	3	0.9903	0.9993	0.9453	0.9602	0.9129
	4	0.9884	0.9993	0.9407	0.9578	0.9052
DeepUNet	0	0.9933	0.9979	0.9518	0.9663	0.9361
	1	0.9928	0.9991	0.9526	0.9672	0.9234
	2	0.9922	0.9996	0.9529	0.9677	0.9195
	3	0.9906	0.9996	0.9494	0.9639	0.9070
	4	0.9888	0.9996	0.9453	0.9639	0.9070
UNet++	0	0.9939	0.9985	0.9533	0.9664	0.9393
	1	0.9930	0.9993	0.9523	0.9662	0.9243
	2	0.9918	0.9996	0.9503	0.9653	0.9184
	3	0.9901	0.9997	0.9464	0.9633	0.9116
	4	0.9883	0.9997	0.9422	0.9610	0.9053
UNet	0	0.9941	0.9990	0.9530	0.9658	0.9397
	1	0.9931	0.9997	0.9520	0.9656	0.9245
	2	0.9918	0.9998	0.9499	0.9646	0.9183
	3	0.9901	0.9999	0.9460	0.9625	0.9115
	4	0.9882	0.9999	0.9416	0.9601	0.9050
LaeNet	0	0.9943	0.9987	0.9549	0.9676	0.9406
	1	0.9935	0.9994	0.9546	0.9676	0.9258
	2	0.9924	0.9996	0.9532	0.9670	0.9202
	3	0.9907	0.9996	0.9495	0.9651	0.9136
	4	0.9888	0.9996	0.9452	0.9628	0.9072

Figure 8. Four examples (row 1–4) of lake area and shoreline of lake Selenco from Landsat-5 images by using LaeNet model with different training images as input. (a) indicate the pseudo color images by using band 4, 4, 5 as the RGB channel; (b,c) are test results of lake area and shoreline by using Landsat-8 for training; test results of lake area and shoreline by using Landsat-5 images for training is indicated in (d,e), respectively.

6. Conclusions

Lake area segmentation and shoreline extraction are crucial steps in lake monitoring. In this paper, we proposed a lightweight, but still effective end-to-end multitask no-downsampling fully CNN (LaeNet) to segment lake area and extract shoreline simultaneously from remote sensing images. Firstly, several no-downsampling CNN layers with *ReLU* activation function are applied to extract semantic features. Then another no-downsampling CNN layer with *Sigmoid* activation function is utilized to segment lake area. Finally, the difference between lake area segmentation and its spatial gradient is derived as the lake shoreline. Extensive experimental results showed that our LaeNet model outperforms mainstream deep semantic segmentation approaches (i.e., UNet, DeepLabV3+ and etc.) in terms of both the performance and the simplicity of model, which can be indicated by time and space usage. Furthermore, the in-situ GPS measurements from one typical part of lake Selenco was used to validate the effectiveness of the LaeNet model. Moreover, we also discussed the applications of the LaeNet model on multibands with different attention mechanisms and images from different remote sensing sensors.

In the future, we will expand the study area from Selenco region to the whole Tibet Plateau and enrich data sources from Landsat series to more sensors (for example, Sentinel-2) to improve the robustness and generality of the LaeNet model. Moreover, powerful deep

network models for area segmentation and edge extraction will be redesigned for tackling more complex application scenarios.

Author Contributions: W.L. designed and developed the model of this study, conducted the experiments and analysis, and wrote the manuscript. J.R. and L.L. supervised W.L. and codesigned the model. X.C. provided the training dataset, assessed the error, and also designed use case diagrams. L.X. helped with GPS data process. Q.W., L.X., and G.L. revised the manuscript. All coauthors helped write the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (grant nos. 41974094 and 41874004) and Key Special Project for Introduced Talents Team of Southern Marine Science and Engineering Guangdong Laboratory (Guangzhou) (GML2019ZD0209).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: Great thanks to the editors and all the anonymous reviewers for their constructive and insightful comments on the earlier drafts of this manuscript. We would like to acknowledge USGS for providing the Landsat-8 and Landsat-5 images for our study and we also thank IGS Analysis Center of the Helmholtz-Centre Potsdam-GFZ German Research Centre for Geoscience for providing the final precise ephemeris.

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

References

1. Yang, K.; Ye, B.; Zhou, D.; Wu, B.; Foken, T.; Qin, J.; Zhou, Z. Response of hydrological cycle to recent climate changes in the Tibetan Plateau. *Clim. Chang.* **2011**, *109*, 517–534. [\[CrossRef\]](#)
2. Zhu, L.; Wang, J.; Ju, J.; Ma, N.; Zhang, Y.; Liu, C.; Han, B.; Liu, L.; Wang, M.; Ma, Q. Climatic and lake environmental changes in the Serling Co region of Tibet over a variety of timescales. *Sci. Bull.* **2019**, *64*, 422–424. [\[CrossRef\]](#)
3. Zhang, G.; Luo, W.; Chen, W.; Zheng, G. A robust but variable lake expansion on the Tibetan Plateau. *Sci. Bull.* **2019**, *64*, 1306–1309. [\[CrossRef\]](#)
4. Zhang, G.; Yao, T.; Xie, H.; Yang, K.; Zhu, L.; Shum, C.; Bolch, T.; Yi, S.; Allen, S.; Jiang, L. Response of Tibetan Plateau's lakes to climate changes: Trend, pattern, and mechanisms. *Earth-Sci. Rev.* **2020**, *208*, 103269. [\[CrossRef\]](#)
5. Zhang, G.; Yao, T.; Shum, C.; Yi, S.; Yang, K.; Xie, H.; Feng, W.; Bolch, T.; Wang, L.; Behrangi, A. Lake volume and groundwater storage variations in Tibetan Plateau's endorheic basin. *Geophys. Res. Lett.* **2017**, *44*, 5550–5560. [\[CrossRef\]](#)
6. Song, C.; Huang, B.; Richards, K.; Ke, L.; Hien Phan, V. Accelerated lake expansion on the Tibetan Plateau in the 2000s: Induced by glacial melting or other processes? *Water Resour. Res.* **2014**, *50*, 3170–3186. [\[CrossRef\]](#)
7. Ma, N.; Szilagyi, J.; Niu, G.Y.; Zhang, Y.; Zhang, T.; Wang, B.; Wu, Y. Evaporation variability of Nam Co Lake in the Tibetan Plateau and its role in recent rapid lake expansion. *J. Hydrol.* **2016**, *537*, 27–35. [\[CrossRef\]](#)
8. Duru, U. Shoreline change assessment using multi-temporal satellite images: A case study of Lake Sapanca, NW Turkey. *Environ. Monit. Assess.* **2017**, *189*, 385. [\[CrossRef\]](#)
9. Li, X.; Long, D.; Huang, Q.; Han, P.; Zhao, F.; Wada, Y. High-temporal-resolution water level and storage change data sets for lakes on the Tibetan Plateau during 2000–2017 using multiple altimetric missions and Landsat-derived lake shoreline positions. *Earth Syst. Sci. Data Discuss.* **2019**, *11*, 1603–1627. [\[CrossRef\]](#)
10. 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. [\[CrossRef\]](#)
11. Zhang, G.; Xie, H.; Kang, S.; Yi, D.; Ackley, S.F. Monitoring lake level changes on the Tibetan Plateau using ICESat altimetry data (2003–2009). *Remote Sens. Environ.* **2011**, *115*, 1733–1742. [\[CrossRef\]](#)
12. Lei, Y.; Yao, T.; Yang, K.; Sheng, Y.; Kleinherenbrink, M.; Yi, S.; Bird, B.W.; Zhang, X.; Zhu, L.; Zhang, G. Lake seasonality across the Tibetan Plateau and their varying relationship with regional mass changes and local hydrology. *Geophys. Res. Lett.* **2017**, *44*, 892–900. [\[CrossRef\]](#)
13. Wang, H.; Chu, Y.; Huang, Z.; Hwang, C.; Chao, N. Robust, long-term lake level change from multiple satellite altimeters in Tibet: Observing the rapid rise of Ngangzi Co over a new wetland. *Remote Sens.* **2019**, *11*, 558. [\[CrossRef\]](#)
14. Zhang, G.; Yao, T.; Xie, H.; Wang, W.; Yang, W. An inventory of glacial lakes in the Third Pole region and their changes in response to global warming. *Glob. Planet. Chang.* **2015**, *131*, 148–157. [\[CrossRef\]](#)
15. Ye, Q.; Zhu, L.; Zheng, H.; Naruse, R.; Zhang, X.; Kang, S. Glacier and lake variations in the Yamzhog Yumco basin, southern Tibetan Plateau, from 1980 to 2000 using remote-sensing and GIS technologies. *J. Glaciol.* **2007**, *53*, 673–676. [\[CrossRef\]](#)
16. El-Asmar, H.M.; Hereher, M.E.; El Kafrawy, S.B. Surface area change detection of the Burullus Lagoon, North of the Nile Delta, Egypt, using water indices: A remote sensing approach. *Egypt. J. Remote Sens. Space Sci.* **2013**, *16*, 119–123. [\[CrossRef\]](#)

17. Lu, S.; Ouyang, N.; Wu, B.; Wei, Y.; Tesemma, Z. Lake water volume calculation with time series remote-sensing images. *Int. J. Remote Sens.* **2013**, *34*, 7962–7973. [\[CrossRef\]](#)
18. Bolch, T.; Buchroithner, M.F.; Peters, J.; Baessler, M.; Bajracharya, S. Identification of glacier motion and potentially dangerous glacial lakes in the Mt. Everest region/Nepal using spaceborne imagery. *Nat. Hazards Earth Syst. Sci.* **2008**, *8*, 1329–1340. [\[CrossRef\]](#)
19. Salerno, F.; Thakuri, S.; D’Agata, C.; Smiraglia, C.; Manfredi, E.C.; Viviano, G.; Tartari, G. Glacial lake distribution in the Mount Everest region: Uncertainty of measurement and conditions of formation. *Glob. Planet. Chang.* **2012**, *92*, 30–39. [\[CrossRef\]](#)
20. Wang, X.; Ding, Y.; Liu, S.; Jiang, L.; Wu, K.; Jiang, Z.; Guo, W. Changes of glacial lakes and implications in Tian Shan, central Asia, based on remote sensing data from 1990 to 2010. *Environ. Res. Lett.* **2013**, *8*, 044052. [\[CrossRef\]](#)
21. Incekara, A.H.; Seker, D.Z.; Bayram, B. Qualifying the LIDAR-Derived Intensity Image as an Infrared Band in NDWI-Based Shoreline Extraction. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 5053–5062. [\[CrossRef\]](#)
22. Ding, X.; Li, X. Shoreline movement monitoring based on SAR images in Shanghai, China. *Int. J. Remote Sens.* **2014**, *35*, 3994–4008. [\[CrossRef\]](#)
23. Shandi, Z.; Helali, H. Investigation of 2019 Rainfall Effects on Urmia Lake Surface and Extraction of Lake Shoreline Changes and Comparison with the Previous Decade Using Remote Sensing Images and GIS. *Isprs J. Photogramm. Remote Sens.* **2020**, *43*, 759–766. [\[CrossRef\]](#)
24. McFeeters, S.K. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *Int. J. Remote Sens.* **1996**, *17*, 1425–1432. [\[CrossRef\]](#)
25. Xu, H. Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery. *Int. J. Remote Sens.* **2006**, *27*, 3025–3033. [\[CrossRef\]](#)
26. Li, B.; Yang, G.; Wan, R.; Dai, X.; Zhang, Y. Comparison of random forests and other statistical methods for the prediction of lake water level: A case study of the Poyang Lake in China. *Nord. Hydrol.* **2016**, *47*, 69–83. [\[CrossRef\]](#)
27. Khan, M.S.; Coulibaly, P. Application of support vector machine in lake water level prediction. *J. Hydrol. Eng.* **2006**, *11*, 199–205. [\[CrossRef\]](#)
28. Yadav, B.; Eliza, K. A hybrid wavelet-support vector machine model for prediction of lake water level fluctuations using hydro-meteorological data. *Measurement* **2017**, *103*, 294–301. [\[CrossRef\]](#)
29. Minghelli, A.; Spagnoli, J.; Lei, M.; Chami, M.; Charmasson, S. Shoreline Extraction from WorldView2 Satellite Data in the Presence of Foam Pixels Using Multispectral Classification Method. *Remote Sens.* **2020**, *12*, 2664. [\[CrossRef\]](#)
30. Altunkaynak, A. Forecasting surface water level fluctuations of Lake Van by artificial neural networks. *Water Resour. Manag.* **2007**, *21*, 399–408. [\[CrossRef\]](#)
31. Kisi, O.; Shiri, J.; Nikoofar, B. Forecasting daily lake levels using artificial intelligence approaches. *Comput. Geosci.* **2012**, *41*, 169–180. [\[CrossRef\]](#)
32. Young, C.C.; Liu, W.C.; Hsieh, W.L. Predicting the water level fluctuation in an alpine lake using physically based, artificial neural network, and time series forecasting models. *Math. Probl. Eng.* **2015**, *2015*. [\[CrossRef\]](#)
33. Feng, Z.; Huang, G.; Chi, D. Classification of the Complex Agricultural Planting Structure with a Semi-Supervised Extreme Learning Machine Framework. *Remote Sens.* **2020**, *12*, 3708. [\[CrossRef\]](#)
34. Demir, N.; Bayram, B.; Şeker, D.Z.; Oy, S.; İnce, A.; Bozkurt, S. Advanced lake shoreline extraction approach by integration of SAR image and LIDAR data. *Mar. Geodesy* **2019**, *42*, 166–185. [\[CrossRef\]](#)
35. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
36. Reichstein, M.; Camps-Valls, G.; Stevens, B.; Jung, M.; Denzler, J.; Carvalhais, N. Deep learning and process understanding for data-driven Earth system science. *Nature* **2019**, *566*, 195–204. [\[CrossRef\]](#)
37. Yuan, J.; Chi, Z.; Cheng, X.; Zhang, T.; Li, T.; Chen, Z. Automatic Extraction of Supraglacial Lakes in Southwest Greenland during the 2014–2018 Melt Seasons Based on Convolutional Neural Network. *Water* **2020**, *12*, 891. [\[CrossRef\]](#)
38. Liang, C.; Li, H.; Lei, M.; Du, Q. Dongting lake water level forecast and its relationship with the three gorges dam based on a long short-term memory network. *Water* **2018**, *10*, 1389. [\[CrossRef\]](#)
39. Pu, F.; Ding, C.; Chao, Z.; Yu, Y.; Xu, X. Water-quality classification of inland lakes using landsat8 images by convolutional neural networks. *Remote Sens.* **2019**, *11*, 1674. [\[CrossRef\]](#)
40. Rostami, M.; Kolouri, S.; Eaton, E.; Kim, K. Deep transfer learning for few-shot sar image classification. *Remote Sens.* **2019**, *11*, 1374. [\[CrossRef\]](#)
41. Ren, Y.; Zhang, X.; Ma, Y.; Yang, Q.; Wang, C.; Liu, H.; Qi, Q. Full Convolutional Neural Network Based on Multi-Scale Feature Fusion for the Class Imbalance Remote Sensing Image Classification. *Remote Sens.* **2020**, *12*, 3547. [\[CrossRef\]](#)
42. Mostajabi, M.; Yadollahpour, P.; Shakhnarovich, G. Feedforward semantic segmentation with zoom-out features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3376–3385.
43. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2650–2658.
44. Li, L. Deep Residual Autoencoder with Multiscaling for Semantic Segmentation of Land-Use Images. *Remote Sens.* **2019**, *11*, 2142. [\[CrossRef\]](#)

45. Wang, J.; HQ Ding, C.; Chen, S.; He, C.; Luo, B. Semi-Supervised Remote Sensing Image Semantic Segmentation via Consistency Regularization and Average Update of Pseudo-Label. *Remote Sens.* **2020**, *12*, 3603. [\[CrossRef\]](#)
46. Zhang, E.; Liu, L.; Huang, L. Automatically delineating the calving front of Jakobshavn Isbræ from multitemporal TerraSAR-X images: A deep learning approach. *Cryosphere* **2019**, *13*, 1729–1741. [\[CrossRef\]](#)
47. Huang, L.; Liu, L.; Jiang, L.; Zhang, T. Automatic mapping of thermokarst landforms from remote sensing images using deep learning: A case study in the Northeastern Tibetan Plateau. *Remote Sens.* **2018**, *10*, 2067. [\[CrossRef\]](#)
48. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
49. Qin, M.; Hu, L.; Du, Z.; Gao, Y.; Qin, L.; Zhang, F.; Liu, R. Achieving Higher Resolution Lake Area from Remote Sensing Images Through an Unsupervised Deep Learning Super-Resolution Method. *Remote Sens.* **2020**, *12*, 1937. [\[CrossRef\]](#)
50. Liu, S.; Ding, W.; Liu, C.; Liu, Y.; Wang, Y.; Li, H. ERN: Edge loss reinforced semantic segmentation network for remote sensing images. *Remote Sens.* **2018**, *10*, 1339. [\[CrossRef\]](#)
51. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
52. Waldner, F.; Diakogiannis, F.I. Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network. *Remote Sens. Environ.* **2020**, *245*, 111741. [\[CrossRef\]](#)
53. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *162*, 94–114. [\[CrossRef\]](#)
54. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
55. Huang, L.; Luo, J.; Lin, Z.; Niu, F.; Liu, L. Using deep learning to map retrogressive thaw slumps in the Beiluhe region (Tibetan Plateau) from CubeSat images. *Remote Sens. Environ.* **2020**, *237*, 111534. [\[CrossRef\]](#)
56. Liu, X.; Chen, B. Climatic warming in the Tibetan Plateau during recent decades. *Int. J. Climatol.* **2000**, *20*, 1729–1742. [\[CrossRef\]](#)
57. Zhang, Y.; Yao, T.; Ma, Y. Climatic changes have led to significant expansion of endorheic lakes in Xizang (Tibet) since 1995. *Sci. Cold Arid Reg.* **2011**, *3*, 0463–0467.
58. Zhou, J.; Wang, L.; Zhang, Y.; Guo, Y.; Li, X.; Liu, W. Exploring the water storage changes in the largest lake (Selin Co) over the Tibetan Plateau during 2003–2012 from a basin-wide hydrological modeling. *Water Resour. Res.* **2015**, *51*, 8060–8086. [\[CrossRef\]](#)
59. Takasu, T.; Yasuda, A. Kalman-filter-based integer ambiguity resolution strategy for long-baseline RTK with ionosphere and troposphere estimation. In Proceedings of the ION GNSS, Portland, OR, USA, 21–24 September 2010; pp. 161–171.
60. Xu, N.; Price, B.; Cohen, S.; Huang, T. Deep image matting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2970–2979.
61. Zhen, M.; Wang, J.; Zhou, L.; Li, S.; Shen, T.; Shang, J.; Fang, T.; Quan, L. Joint Semantic Segmentation and Boundary Detection using Iterative Pyramid Contexts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 13666–13675.
62. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
63. Ozan, O.; Jo, S.; Loic, L.F.; Matthew, L.; Mattias, H.; Kazunari, M.; Kensaku, M.; Steven, M.; Nils Y, H.; Bernhard, K.; Ben, G.; Daniel, R. Attention U-Net: Learning Where to Look for the Pancreas. *arXiv* **2018**, arXiv:1810.03999.
64. Mohajerani, S.; Saeedi, P. Cloud-Net: An end-to-end cloud detection algorithm for Landsat 8 imagery. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 1029–1032.
65. Li, R.; Liu, W.; Yang, L.; Sun, S.; Hu, W.; Zhang, F.; Li, W. Deepunet: A deep fully convolutional network for pixel-level sea-land segmentation. *IEEE J. Sel. Top Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3954–3962. [\[CrossRef\]](#)
66. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. *IEEE Trans. Med. Imaging* **2019**, *6*, 1856–1867. [\[CrossRef\]](#)
67. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: A Nested U-Net Architecture for Medical Image Segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–11.
68. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
69. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [\[CrossRef\]](#)
70. Mohajerani, S.; Saeedi, P. Shadow Detection in Single RGB Images Using a Context Preserver Convolutional Neural Network Trained by Multiple Adversarial Examples. *IEEE Trans. Image Process* **2019**, *28*, 4117–4129. [\[CrossRef\]](#)
71. Mou, L.; Zhu, X.X. Learning to pay attention on spectral domain: A spectral attention module-based convolutional network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 110–122. [\[CrossRef\]](#)
72. Woo, S.; Park, J.; Lee, J.Y.; So Kweon, I. Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.

-
73. Qiao, Y.; Liu, Y.; Yang, X.; Zhou, D.; Xu, M.; Zhang, Q.; Wei, X. Attention-Guided Hierarchical Structure Aggregation for Image Matting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 13676–13685.
 74. Augustauskas, R.; Lipnickas, A. Improved Pixel-Level Pavement-Defect Segmentation Using a Deep Autoencoder. *Sensors* **2020**, *20*, 2557. [[CrossRef](#)] [[PubMed](#)]
 75. Lau, S.L.; Wang, X.; Xu, Y.; Chong, E.K. Automated Pavement Crack Segmentation Using Fully Convolutional U-Net with a Pretrained ResNet-34 Encoder. *arXiv* **2020**, arXiv:2001.01912.
 76. Escalona, U.; Arce, F.; Zamora, E.; Sossa Azuela, J.H. Fully convolutional networks for automatic pavement crack segmentation. *Comput. Sist.* **2019**, *23*, 451–460. [[CrossRef](#)]