Article

Intelligent Extraction of Terracing Using the ASPP ArrU-Net Deep Learning Model for Soil and Water Conservation on the Loess Plateau

Yinan Wang, Xiangbing Kong *, Kai Guo, Chunjing Zhao and Jintao Zhao

Key Laboratory for Soil and Water Conservation on Loess Plateau, Yellow River Institute of Hydraulic Research, Yellow River Conservation Commission of the Ministry of Water Resources, Zhengzhou 450003, China; 20220121@hhu.edu.cn (Y.W.); 181309010004@hhu.edu.cn (K.G.); zhaochunjing@hky.yrcc.gov.cn (C.Z.); zhaojintao@hky.yrcc.gov.cn (J.Z.)

* Correspondence: kongxb@foxmail.com; Tel.: +86-13939019189

Abstract: The prevention and control of soil erosion through soil and water conservation measures is crucial. It is imperative to accurately and quickly extract information on these measures in order to understand how their configuration affects the runoff and sediment yield process. In this investigation, intelligent interpretation algorithms and deep learning semantic segmentation models pertinent to remote sensing imagery were examined and scrutinized. Our objective was to enhance interpretation accuracy and automation by employing an advanced deep learning-based semantic segmentation model for the astute interpretation of high-resolution remote sensing images. Subsequently, an intelligent interpretation algorithm model tailored was developed for terracing measures in high-resolution remote sensing imagery. Focusing on Fenxi County in Shanxi Province as the experimental target, in this research we conducted a comparative analysis between our proposed model and alternative models. The outcomes demonstrated that our refined algorithm model exhibited superior precision. Additionally, in this research we assessed the model's generalization capability by utilizing Wafangdian City in Liaoning Province as another experimental target and performed a comparative analysis with human interpretation. The findings revealed that our model possesses enhanced generalization ability and can substantially augment interpretation efficiency.

Keywords: loess plateau; soil and water conservation; terracing measures; deep learning; convolutional neural network; U-Net

check for updates

Citation: Wang, Y.; Kong, X.; Guo, K.; Zhao, C.; Zhao, J. Intelligent Extraction of Terracing Using the ASPP ArrU-Net Deep Learning Model for Soil and Water Conservation on the Loess Plateau. *Agriculture* **2023**, *13*, 1283. https://doi.org/10.3390/agriculture13071283

Academic Editors: Liujun Zhu, Jeffrey Walker and Carsten Montzka

Received: 7 June 2023
Revised: 18 June 2023
Accepted: 20 June 2023
Published: 22 June 2023

1. Introduction

Soil and water conservation measures are important means to curb soil erosion by changing the water flow path, runoff velocity, and ground morphology further to alter the movement pattern of water and sand and abate runoffs and sediment loss [1]. From the perspective of the affected areas, soil and water conservation measures can be categorized into slope measures that mainly include terracing and afforestation and gully measures that mainly involve the building of soil-retaining dams [2]. Since the middle of the last century, large-scale soil and water conservation measures have been taken in China's loess plateau areas, and the long-term treatment practice has seen significant effects in soil and water conservation [3]. Nevertheless, limited by the extraction precision of existing soil and water conservation measures, different problems still exist, including the lack of a sound and accurate quantitative and parametric index system for soil and water conservation, the inferior structure of soil and water conservation measures, and the unclear identification of how the configuration of these measures affects the runoff and sediment yield process [4]. Therefore, developing remote sensing technologies for real-time high-frequency and rapid multi-target identification and high-precision intelligent extraction of different soil and water conservation measures, such as vegetations, terraces, and soil-retaining dams, and clarifying the relevant parameter information such as the distribution and magnitude of soil and water conservation measures is of great importance for the quantitative research on how each measure contributes to the conservation of soil and water and for river basin management [5].

The measurement techniques for soil and water conservation mainly include traditional techniques based on manual ground measurements, remote sensing techniques based on visual interpretation, and intelligent techniques based on automatic extraction [6]. Specifically, most traditional measurement approaches are carried out manually, where hand-held GPS is used for field measurement and topographic mapping. These approaches are applicable to areas of small size and regular topography [7]. With the various types of soil and water conservation measures available, major measures such as vegetation and terrace are scattered and take up areas of large size. As a result, traditional measurement approaches typically require substantial manpower and material resources, along with high time and economic costs.

In the visual interpretation of terrace extraction, integrated inferences are made mainly according to the spectral, texture, and morphological features of terrace imaging, as well as professional background knowledge [8]. Terraces can be quantitatively expressed through a series of geometric measurement parameters, such as location, area, length (width), ridge height and slope, etc. Scholars have used remote sensing images to extract indicators such as terrace texture, field surface and ridge, and edge line, which has laid a solid foundation for building numerical simulation models for terraces [9]. However, the width of a single terrace is mostly less than 20 m, and under the constraints of image quality and information processing measures, visual interpretation has various shortcomings in extracting terraces, such as substantial precision variations, poor repeatability, and low precision in terrace positioning and boundary determination, which result in the unsatisfactory effect of terrace information extraction [10]. With the rapid development of computer science and high-resolution remote sensing technologies, as well as the emergence of massive remote sensing data, computer-automated extraction has been the focus of attention, and important progress has been made in the research of extracting information on soil and water conservation measures in large regional areas through high-resolution remote sensing images [11,12]. Scholars in China and abroad have been able to extract terrace information through image unit-based extraction techniques and object-oriented methods [13,14], the latter of which has overcome the constraints in image unit-based approaches and shown improvement in the precision and reliability of classification and extraction of high-resolution remote sensing images [15,16]. However, as the topography of loess plateau areas is fragmented with complicated spectral features, an object-oriented approach will be affected by the phenomena of "same spectrum for different objects" and "different spectrums for the same object". Additionally, most object-oriented methods only extract terraced areas on large scales without a thorough investigation of the relevant information, such as the extraction precision of the field surfaces and the size statistics. Therefore, such approaches lack a quantitative evaluation of the extraction indicators and their precision.

Deep learning is a development from artificial neural networks and is an important branch in the machine learning field. Machine learning derives generalized laws from a limited amount of observed data through learning and extends the summarized laws to unobserved data [17]. Unlike traditional machine learning, deep learning uses more complex models. The so-called "deep" means the number of non-linear feature transformations performed on the original data, with the main objective of automatically learning valid feature representations from the input data [18]. With the development of computer hardware and the emergence of graphics processing unit (GPU) devices, computer performance has been improving, ushering in the peak of deep learning research with the emergence of various deep learning network models, including AlexNet, VGGNet, and GoogLeNet. All these models are highly powerful in feature extraction and can independently learn high-level abstract features of data without having to manually extract these complex features [19,20]. Presently, deep learning is still rapidly developing with constant updates, and its application has been widely studied in various industries and fields.

With the rapid emergence of artificial intelligence technology, machine learning methods based on deep learning have been emerging gradually in different research fields, and the interpretation of remote sensing images has also been going "smart". Particularly, extensive research and applications have been made in relevant fields, such as remote sensing image classification, semantic segmentation, and target detection based on deep learning methods [21,22]. Deep learning can discriminate images, texts, sounds, and other factors by simulating the mechanism of a human brain, with the most prominent feature being its ability to automatically extract deep semantic features from massive data and address complex classification issues by fitting multi-layer non-linear networks. Unlike backpropagation (BP) neural networks, support vector machines, and other traditional machine learning algorithms, deep learning approaches address the challenge of classification by fitting complex mathematical models [23]. As deep learning theory and computer hardware and software keep improving, deep learning technologies are applied to the intelligent interpretation of targets in high-resolution remote sensing images, with their powerful capacity for feature extraction being fully utilized to improve the automation of interpretation and the precision of results.

This research was based on high-spatial-resolution remote sensing images, where an improved U-type neural network (U-Net) model of deep learning artificial neural network algorithms was used to develop intelligent extraction technologies for terracing measures for soil and water conservation. Furthermore, this research established an identification library of algorithm models and evaluated the adaptability of and optimized the models to improve the intelligent extraction ability and efficiency of terracing measures for soil and water conservation.

2. Study Area and Data Source

As shown in Figure 1, this research took Fenxi County in Linfen City of Shanxi Province as an example. Fenxi County is situated in northern Linfen City and the southeastern foot of the Luliang Mountains, with a total land area of 880 km^2. Specifically, Fenxi County is seated on the east side of the great anticline of the Luliang Mountains, which is a transition zone of a complex geological structure between the central uplift part of the anticline and the Linfen Faulted Basin. The western part of the county is composed of bedrock hills with an average elevation of 1200–1300 m and exposed bedrock in some areas, and the stratigraphy mainly consists of rocks, such as Ordovician limestones and Permian sandstones. Under the effect of faulted rivers, the place has formed a natural state where sheer cliffs are alongside deep river valleys, and hills and valley streams coexist, with latent subterranean caves, a lack of water resources, and poor vegetation coverage [24]. In terms of topography, Fenxi County is higher in the northwest and lower in the southeast, and the valleys in the eastern part are deeply cut with wide and open floors and an average elevation of 800–1000 m. The county's highest point is the Gushe Mountain in the west, with an elevation of 1890.8 m, while the lowest point is the Tuanbai River in the east, with an elevation of 550 m, putting the relative elevation difference at 1340.8 m. The rivers in Fenxi County are all part of the Fen River system in the Yellow River Basin, with an annual average runoff depth of 35.4 mm and runoff volume of 60 million m^3, except for the 4 km^2 river on the western slope of the Gushe Mountain, which is part of the Xinshui River Basin. Inside the county, there are a total of 670 rivers and ditches that are over 0.5 km long, with a density of 4.4 km/km^2, and the main tributaries include the Tuanbai River, the Duizhu River, the Dianping River, the Gouxi River, and the Junyang River, etc. [25].

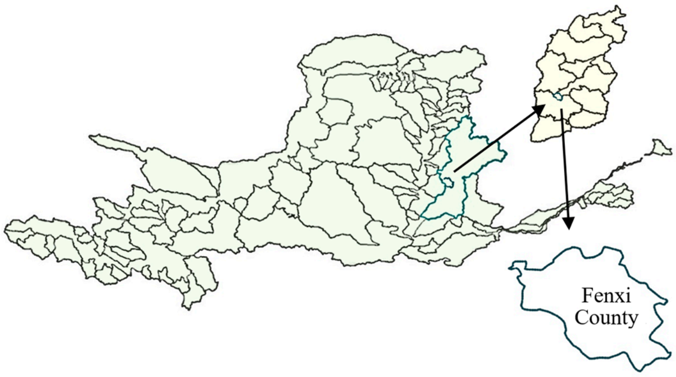


Figure 1. Schematic diagram of the study area.

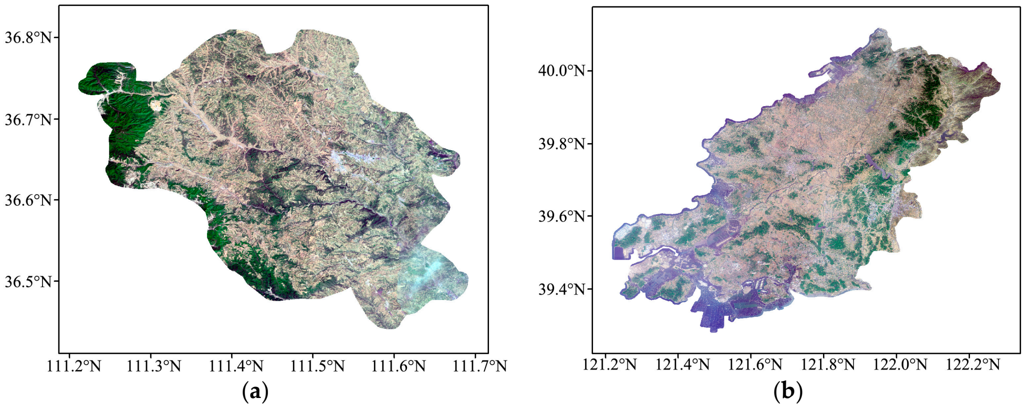
The high-resolution remote sensing image data used in this research are from the “Gaofen-1” satellite (hereinafter referred to as “GF-1”), which has a designed service life of 5 to 8 years and is the first satellite in China’s “Gaofen series” of high-resolution earth observation systems with key technological breakthroughs in high spatial resolution and multi-spectral and wide-coverage optical remote sensing. The GF-1 satellite features a combination of high and medium spatial resolution for earth observation and high-bandwidth imaging, with the width of 2 m resolution panchromatic and 8 m resolution multi-spectral images being larger than 60 km and that of 16 m resolution multi-spectral images being larger than 800 km, indicating its greatly improved observation capability with unique advantages for large-scale ground observation and environmental monitoring [26]. Furthermore, in addition to having a similar spatial resolution, GF-1 can also take repeated pictures of an area within a shorter period of time with a repetition cycle of only four days. Therefore, GF-1 has achieved the integration of high spatial and high temporal resolution. Presently, GF-1 data have been widely used for soil and water conservation and research in related fields [27] because it is easier and less expensive to acquire data with GF-1 than with foreign commercial satellites; also, its high spatial and temporal resolution can meet the requirements of most domestic soil and water conservation projects, which allows it to quickly position and obtain results for land disturbance, vegetation restoration, and the status of soil and water conservation measures.

This research carried out corresponding field surveys and monitoring to train, validate and analyze the relevant remote sensing inversion parameters to verify the precision of the image data and the inversion calculation results. Wafangdian City (in Dalian City, Liaoning Province) in the Liaohe River Basin was chosen as the verification area for experiments to verify the generalization ability of the model algorithms. Wafangdian City is located in the middle and west of Liaodong Peninsula, 39°20′~40°07′ N, and 121°13′~122°16′ E. It is a national development zone and also an important economic zone connecting Shenyang and Dalian. The total land area of Wafangdian City is 4176 km^2, and it has numerous and widely distributed terraces, which are conducive to the experimental study and analysis of model generalization ability.

3. Methods

3.1. Processing of Remote Sensing Image Data and Establishment of Model Identification Library

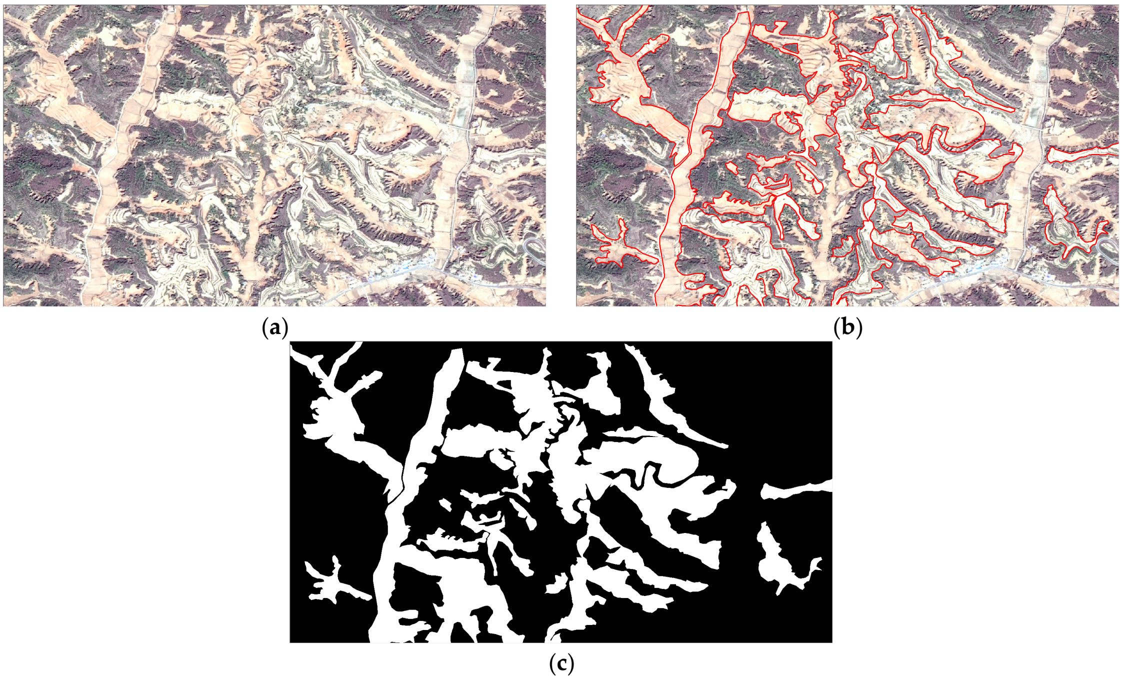
The processing of remote sensing images in this research mainly included orthorectification, image fusion, true color output, the dodging, mosaicking, and cropping of the images, and format conversion, etc., through which the result data of the images were eventually obtained, as shown in Figure 2.



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

Figure 2. Processing results of remote sensing images. (**a**) Fenxi County; (**b**) Wafangdian City.

During the training process, a large amount of data are required to support the deep learning neural network model. Insufficient data can lead to overfitting, which will further affect the generalization ability of the model and result in the poor precision of the final results. Likewise, sample data are indispensable to the training of neural network models, thus necessitating the labeling of the images in the training datasets [28]. To make sure that the training samples were accurate, in this research we carried out human visual interpretation of the images of the study area and manually labeled them, and outlined and labeled the interpreted terraces in the form of polygons. A single-band black-and-white binary map was eventually produced, with the white indicating the terraces and the black indicating the other areas, as shown in Figure 3.



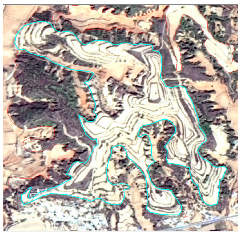
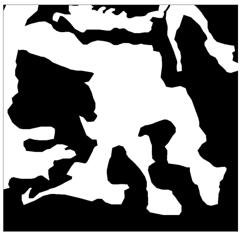
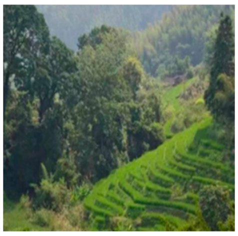
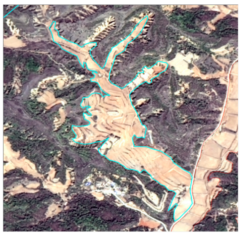
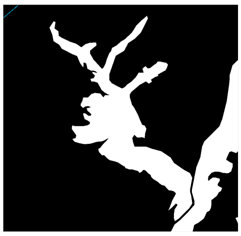
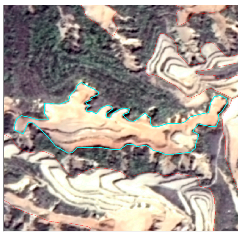
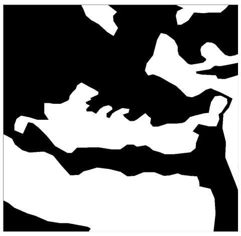
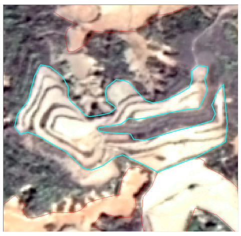
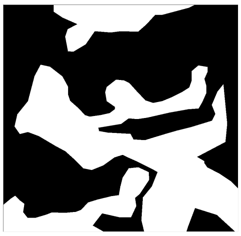
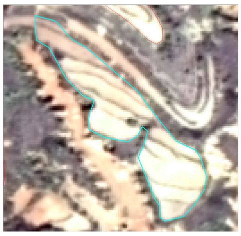
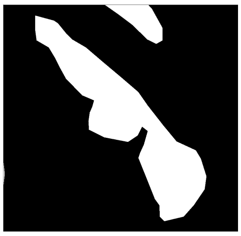
(**a**) (**b**)

(**c**)

Figure 3. Images and sample identification map of the study area. (**a**) Satellite image map; (**b**) sample identification map; (**c**) sample binary map.

In addition to visual interpretation, in this research we collected information from field surveys and summarized it to build an identification library of deep learning neural network models, as shown in Table 1.

Table 1. Sample identification of terracing measures.

SN	Satellite Image	Sample Identification	Field Photo
1			
2			
3			
4			
5			

3.2. Comparison and Selection of Deep Learning Frameworks

Deep learning is based on BP algorithms to update the model parameters, while human calculation of the gradient leads to a lower efficiency of computation [29,30]. This led to the emergence of some deep learning frameworks that can automatically calculate the gradient with added support for GPU computing, which is aimed at the more efficient and accurate completion of the model design of deep learning networks. Some typical deep learning frameworks include TensorFlow, Caffe, PyTorch, etc. [31,32].

TensorFlow is a deep learning framework that was first launched by Google, Inc. in November 2015 and supports different programming languages, including C++, Python, Java, and R. Under the TensorFlow framework, a data flow diagram is used to build network models. Specifically, in the diagram, the nodes represent specific mathematical operations, and the lines represent the inter-node multidimensional data, which are called tensors. The TensorFlow framework has the following key advantages: (1) Higher flexibility. The framework comes with various underlying algorithms that can be used to customize the desired operations without the user worrying about the loss of computational performance. (2) Automatic differentiation; that is, as a network model is being built, TensorFlow only needs the forward propagation to be written before it automatically customizes the back-propagation. (3) Good compatibility, which can reduce extra workload caused by platform migration. These advantages allow TensorFlow frameworks to reduce the development cost of deep learning network models and become widely used in industry and academia.

Caffe is a deep learning framework that emerged early with a kernel written in C++, and its underlying operation is layer-based, including convolutional layers, pooling layers, fully connected layers, etc. It supports Matlab and Python interfaces with various advantages, including strong readability and fast realization, and greatly facilitates the training of network models. Some early typical deep learning network models, such as VGG and DenseNet, have all been developed based on the Caffe framework. Plus, in Caffe's model library, there is a large number of pre-trained models that can facilitate migration and learning between different tasks and have been widely used in the computer vision field.

PyTorch is a deep learning framework launched by Facebook in 2017 and is written in Python. As a later-emerging deep learning framework, PyTorch draws on the advantages of other deep learning frameworks and uses dynamic flow diagrams to build network models. It is more intuitive than the static flow diagrams adopted by the TensorFlow framework and also has the function of automatic differentiation. Presently, PyTorch has been increasingly used and is the framework under which the deep learning neural network model in this research was constructed.

3.3. Selection of Deep Learning Algorithm Models and Determination of Indicators for Precision Evaluation

3.3.1. Comparison of Deep Learning Models

Deep-learning-based image segmentation models are mainly categorized into two types: instance segmentation and semantic segmentation. The former only infers the class of specific objects in an image, while the task of the latter is the classification of each pixel in an image according to the semantic class to which the pixel belongs. In the computer vision field, the semantic segmentation of images plays a crucial role. Like any other images, remote sensing images also have the relevant features of semantic information, such as color, texture, and shape, a set of features based on which semantic segmentation divides an image into several disjoint regions according to the class each pixel belongs to. All the pixels in each of these regions have the same semantic information, but those in different regions are of different semantic information [33]. Commonly used semantic segmentation models include the fully convolutional neural network (FCN), semantic segmentation network (SegNet), U-Net, etc.

(1) FCN model

The FCN network was first proposed by Long et al. in 2015 [34]. It was the first neural network that achieved pixel-level classification in deep learning, as shown in Figure 4 [29], and has overcome the shortcomings of CNNs in the earlier stage with no limitation on the size of the input image. It can input images of any size, the feature vectors of which, after several convolution and pooling operations, are no longer constructed by the fully connected layers. Instead, the feature map is deconvoluted for upsampling, so that the output and input images are of the same size, and the upsampled feature map decides the class each pixel belongs to through the softmax classifier, thus realizing the end-to-end semantic segmentation of the images. To avoid the feature map becoming coarse during the upsampling process, FCN adopts the shortcut connection approach to fuse the locational features extracted from the shallow layers with the semantic features extracted from the deep layers by adding them together, thus optimizing the prediction results.

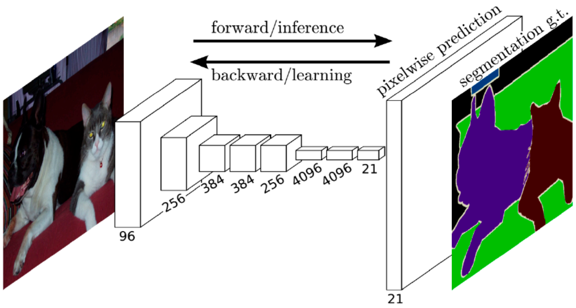


Figure 4. Structure of the FCN model.

In the FCN model, deconvolutional layers are used instead of fully connected layers in CNN to realize the one-dimensional to the two-dimensional transition of the results outputted by the neural network model, and the shortcut connection method is used to improve the segmentation result of the model significantly. The FCN model has been dubbed the pioneer of semantic segmentation models in deep learning. However, experimental studies have shown that although the FCN model uses shortcut connections to achieve multi-feature fusion, it does not take into sufficient account the spatial consistency between the pixels at different levels and neglects the contextual information of the images, which leads to blurred edge information in the final prediction results.

(2) SegNet model

The SegNet model is a semantic segmentation model of an encoder–decoder structure, as shown in Figure 5 [19]. The model was initially applied in the scene identification of autonomous driving, with the encoding and decoding ends being in a symmetric relationship. The encoding end is composed of the first 13 convolutional network layers of the VGG-16 model for feature extraction of images. Pooling is used to reduce the dimensionality of the extracted feature map and record the spatial position of each pooling result on the original feature map, i.e., the max pooling index value, which is then used for upsampling the input feature map with the boundary information being thus preserved. The main role of the decoding end is to recover the size of the images by upsampling the feature map of low-resolution images and then perform a non-linear unpooling operation based on the recorded max pooling index value. The unpooling operation can optimize the boundary division of the segmented images and reduce the number of training parameters in the model.

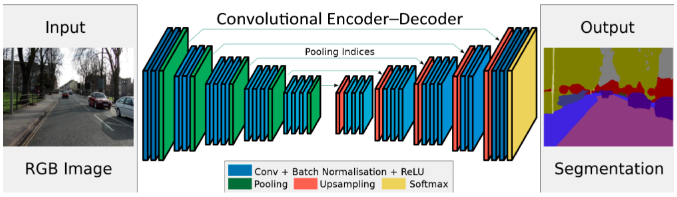


Figure 5. Structure of the SegNet model.

(3) U-type neural network model

The U-type neural network (U-Net) model is a segmentation network that was first proposed by Ronneberge et al. in 2015. It is of a symmetric U-shaped structure [35] with its network structure, as shown in Figure 6. The contracting path on the left side of the network is the encoding end comprising four convolutional layers, each of which is a stack of two 3 × 3 convolutional kernels, and each time, the neighboring convolutional layers use a 2 × 2 max pooling operation to downsample the feature map. The key function of the contracting path is to extract the features of the input image after several convolution and pooling operations and eventually form a high-dimensional feature map. The expanding path on the right end is a decoding structure composed of four upsampling layers. Identically to the upsampling process of the FCN model, it upsamples through deconvolution to avoid the loss of features during the feature transfer process. The U-Net model is an improvement of the FCN model in the following senses. Firstly, during the upsampling process, the FCN model directly deconvolves small-sized feature maps to expand the size of the images, while the U-Net model, on the other hand, deepens the depth of the network model by performing multiple convolution operations on small-sized feature maps, so that the model can learn more complex features. Secondly, the FCN model only fuses the features on the shallow and deep layers by summing up the corresponding pixels, while the U-Net model, on the other hand, uses the shortcut connection to splice the features in the band dimension, which improves the problem of blurred edge details and preserves the information of latitudinal position.

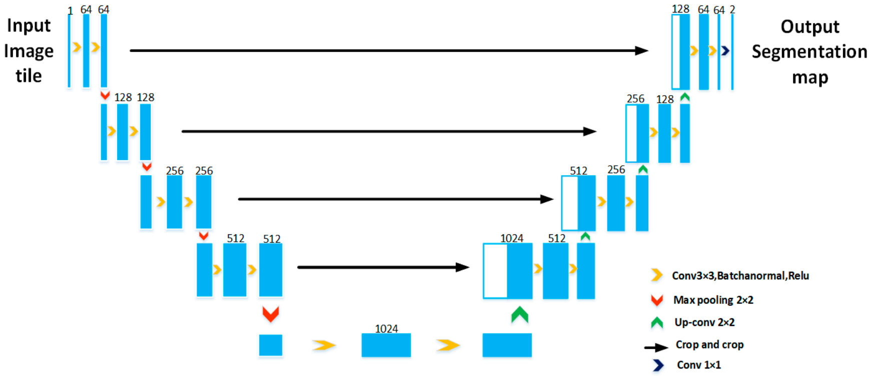


Figure 6. Structure of the U-Net model.

After being proposed, the U-Net model was widely used in medical imaging, and with the development of intelligent interpretation in remote sensing images, extensive studies have been gradually carried out to investigate various tasks performed by the U-Net model, such as the classification and the building and road extraction of remote sensing images [36,37].

3.3.2. Precision Evaluation Indicators

The information extraction of remote sensing images can be regarded as a matter of binary classification. In this research, to accurately evaluate the extraction precision of each model, the precision evaluation method is used in machine learning binary classification to construct the confusion matrix with the extracted and unextracted image units corresponding to two different categories, as shown in Table 2.

Table 2. Confusion matrix for precision verification.

Image Map	**Identification Map**	
	To Be Extracted	**Not To Be Extracted**
Extracted	TP	FP
Unextracted	FN	TN

In the table, TP (true positives) indicates the number of image units that should be and have been correctly extracted; FN (false positives) indicates the number of image units that should be but have not been extracted; FP (false positives) indicates the number of image units that should not be but have been incorrectly extracted; and TN (true positives) indicates the number of image units that should not be and have not been extracted.

Based on the statistics of the confusion matrix in the above table, the calculation formulas for the precision evaluation of the extraction results are as follows:

(1) Precision (P): the ratio of the number of correctly extracted image units to the number of all extracted image units.

$$P = TP/(TP + FP) \quad (1)$$

(2) Recall (R): the ratio of the number of correctly extracted image units to the number of all image units to be extracted.

$$R = TP/(TP + FN) \quad (2)$$

(3) Omission (O): the ratio of incorrectly extracted image units to the number of all image units to be extracted.

$$O = FN/(TP + FN) \quad (3)$$

(4) F1-Score (F1): the precision indicator for binary classification in statistics, which is the weighted average of the precision and recall.

$$F1 = 2PR/(P + R) \quad (4)$$

3.4. Optimization of Model Algorithms and Improvement of Model Precision

3.4.1. Phenomena and Avoidance Methods of Model Overfitting and Underfitting

In the training process of a neural network model, two types of typical problems are often encountered. Firstly, the model fails to achieve a low level of training errors during training, a phenomenon called model underfitting; secondly, the training errors of the neural network model are much fewer than those in the validation data set. In other words, the model achieve high precision on the training data set but not so much on the validation data set, a phenomenon called model overfitting. Figure 7 is an example of the overfitting and underfitting phenomenon of a regression problem, where the left figure shows the poor result of using a simple linear function to classify the triangles and circles, and the right figure displays the successful classification of all the circles and triangles. However, this classification function is too complex; that is, the loss function converges completely in the current training set with high prediction precision, but the validation set shows almost no precision with the occurrence of overfitting.

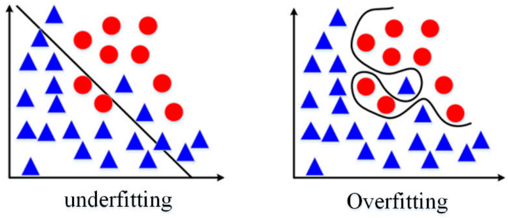


Figure 7. Schematic diagram of underfitting and overfitting.

The deep learning artificial neural network model, per se, predicts the unknown data by fitting the data on the training set and obtaining the fitted model after the training is completed. Therefore, overfitting of the model inevitably occurs during the training process and is usually avoided through the following methods.

(1) Terminating the training of the neural network model early. During the training process of the neural network model, the loss value of the current model output is recorded. On the training set, the loss value will show a significant decrease at the beginning of training, followed by a gradual decrease after several iterations, until it converges to a specific value. In the validation set, the loss value will first decrease, and when it drops to a specific value before rising, it means that the model has been overfitted. Therefore, in model training, overfitting can be avoided by terminating the model early when the overfitting phenomenon appears, a method called the early termination method.

(2) Data augmentation. The most effective way to address overfitting is to increase the amount of data in the training set because when the amount of data in the training set is large enough, the training process will repeatedly update the parameters to increase the generalization ability of the model, with better prediction results being eventually achieved. However, in the actual application, the labeling of samples is carried out manually, making it hard to obtain a large number of training samples due to the constraint of manpower and financial resources. Presently, a common practice is to geometrically transform existing training samples and achieve data augmentation by panning, rotating, randomly cropping, and adding noise to the images to expand the training samples.

(3) Dropout. In addition to the above methods, other common methods are used to address overfitting encountered with the deepening of the number of the network layers, including regularization, dropout, adding noise, model improvement, etc. Of these approaches, dropout is one of the most commonly used methods to control overfitting in neural networks [38], and its principle is to randomly select, subject to a certain probability, some of the neurons to be included in the training. This is equivalent to training subnetworks composed of these neurons, with one subnetwork with different neurons being trained at each time, and a total neural network is eventually formed by putting all the trained subnetworks together. As some neurons are not involved in each training, the number of parameters of the network is greatly reduced, which reduces the complexity of the model and effectively avoids overfitting.

The intrinsic principle of dropout is to reduce the interactions between the neurons, and as neurons are chosen randomly for isolation each time, the blocked targets are different for each training, making the update of weights (parameters) independent from the inter-node interactions. On the other hand, the subnetworks eventually have to be assembled into a total neural network, which is an averaging process. Since the weights are different for each training, performing multiple averaging will stabilize the network and avoid overfitting.

3.4.2. Strategies of Model Algorithm Optimization

The purpose of optimization algorithms in deep learning is to find the optimal weights and bias values to minimize, to the greatest extent possible, the difference between the predicted and target values of the model, and finding the optimal parameters is the process of neural network model training. Optimization algorithms are the strategies to update the parameters throughout the whole training process of the neural network, with the two following commonly used algorithms:

(1) Stochastic gradient descent (SGD)

The gradient is, per se, a vector, which means the maximum value of the directional derivative at a certain point of the function will be obtained along this direction; that is, the function has the maximum rate of changes at this point. Identifying the derivative of a multivariate function will help to obtain multiple partial derivative functions which form the vector, i.e., the gradient. Stochastic gradient descent (SGD) involves taking one sample in each iteration and calculate the gradient of the loss function of the sample for the update of the parameters. Using a SGD algorithm is equivalent to the introduction of random noise to the process of gradient descent, which can keep away from the local optimal point when the objective function is non-convex, thus preventing the loss function from falling into a local optimal solution. The SGD algorithm can calculate the gradient value faster and accelerate the convergence of the neural network.

(2) Adam optimization algorithm

The Adam (adaptive moment estimation) optimization algorithm is an extension of the SGD algorithm and an adaptive neural network optimization algorithm with a learning rate that not only takes into account the first-order moment (mean) to calculate the learning rate of the adaptive parameters but also makes full use of the second-order moment mean (variance) of the gradient. The Adam algorithm uses the prior i-1st iteration gradient to update the parameters of the *i*-th iteration; that is, to estimate the first-order moment, while calculating the second-order moment to update the parameters, thus solving the problem of being unable to update the parameters due to a small learning rate. It also introduces correction factors to the calculation of the first-order and second-order moments. Presently, it is one of the most commonly used optimization algorithms in deep learning.

3.4.3. Methods of Model Precision Improvement

High-resolution remote sensing images contain complex scene and feature information, with greatly varying sizes of different targets, which can lead to the absence of spatial information and thus result in omissions and errors if existing deep learning neural network models are used directly. In addition, during certain operations, such as convolution and pooling, a lot of location information is lost, leading to the loss of details in the extracted objects, other problems and the further problem of under-segmentation. This research improved the U-Net model to train a higher-precision measure extraction model for high-resolution remote sensing images.

(1) Residual learning

Scholars have proposed various methods to improve the prediction precision of models, such as reducing the size of the convolution kernels and increasing the width and the number of layers of the network models, etc. Theoretically, when the number of layers in a neural network increases, the model can extract deep and complex features from the input image to improve its prediction ability. CNN has also been developed from the 5-layer LeNet-5 model and the 22-layer GoogleNet model, and its prediction effect has been constantly improved. Based on this characteristic, neural network models have been developing toward deep network structures. However, numerous studies have shown that when the number of layers of a neural network model reaches a certain depth, the predictive ability of the model becomes worse rather than better, indicating the occurrence of model degradation. Model degradation refers to the phenomenon where, as the number of layers of the training network increases, the loss function of the training set gradually decreases before it saturates, and when the depth of the network is further increased, the loss function of the training set rebounds, leading to worse training effects. The difference with overfitting is that the loss function of overfitting decreases throughout the whole process.

This phenomenon is mainly caused by two reasons. (1) As the network parameters are initialized close to 0, the gradient will disappear with the deepening of the model layers when the parameters of the shallow layers are being updated, resulting in the update failure of the parameters on the shallow layers, which will further lead to the problem of gradient disappearance. (2) The feature extraction ability of a neural network does not increase with the increasing number of layers in the network. Instead, each neural network model has an optimal value for the number of layers, which when exceeded in the design of the network structure will lead to model degradation.

Given the above reasons, this research introduced the residual learning method [39], where the input feature of a layer was set as x, and the expected output feature was set as (H)x, in the case of which the residual mapping fitted by this network would be (H)x = x + F(x). The key idea of residual learning is to use multiple convolutional layers to keep fitting the residual function by accumulating it with the input feature x and comparing it with the expected output feature (H)x. When the value of the residual function becomes close to 0, the expected output feature value will reach the optimal solution. The residual network contains a stacked series of residual blocks, can effectively avoid gradient disappearance and realize the training of deeper network models. Contrast with the traditional CNN, the key idea of a residual network is to add a shortcut connection between the convolutional layers, an approach that can help the gradient to bypass the weight layer and thus form a shortcut connection between the residual units, which can reduce the computation of the model and transfer the input information directly to the input end through the shortcut connection, so that the whole network only needs to learn the differences between input and output, making it possible to extract new features further. The introduction of residual networks can effectively prevent model degradation from occurring with the deepening of the network and largely avoid gradient disappearance or explosion.

As shown in Figure 8, under the given basic structure of the residual blocks, information x was inputted and processed in the first layer (convolution operation) to obtain F(x), before the first round of F(x) was fed to the network in the second layer (convolution operation) and updated to obtain the new F(x) with the rectified linear unit (ReLU) activation function also being adopted. At this point, the updated F(x) was added to the input x to obtain the next round of input x. The completion of such a cycle is called a residual block, and the residual network is formed by the stacking of multiple residual blocks as described above. However, it should be noted that the dimensionality of the input information (image information) constantly changes during the operation process of convolution and pooling, with the possibility that the dimensionality of x does not match that of F(x). To address this situation, when F(x) + x is carried out, the dimensionality of the information should be gauged; when inconsistency arises in its dimensions, a 1 × 1 convolution operation will be needed to achieve dimensional unification.

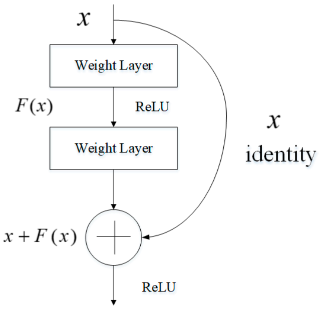


Figure 8. Schematic diagram of the structure of the residual blocks.

(2) Recurrent residual convolutional unit

The function of a recurrent residual convolutional unit (RRCU) is to accumulate the results of the convolution operation on the prior layer with those on the current layer as the input of the next layer. The feature mapping of x_i is calculated by a linear transformation and non-linear activation function in turn on the $i-1$ network layer to obtain the final output pixel value, and if the input passes through an RRCU, its output, y_i, can be expressed as follows:

$$y_i = H_i(W_i^T \times x_i + W_{i-1}^T \times x_{i-1}, b) \tag{5}$$

where, H(·) represents the non-linear network mapping of the *i*-th layer; W_i and W_{i-1} represent the weight of the input features in the $i-1$ convolution layer and that in the i-th layer, respectively; and b is the bias term. Such a RRCU can extract multi-scale features of different perceptual fields, combine the features in the front and back layers, and reuse the feature maps, which makes it conducive for the feature extraction of small data sample sets. As shown in Figure 9, adding residual learning to the RRCU can further improve the efficiency of using the features, solve the degradation problem caused by the presence of too many network layers, and accelerate the convergence of the network model.

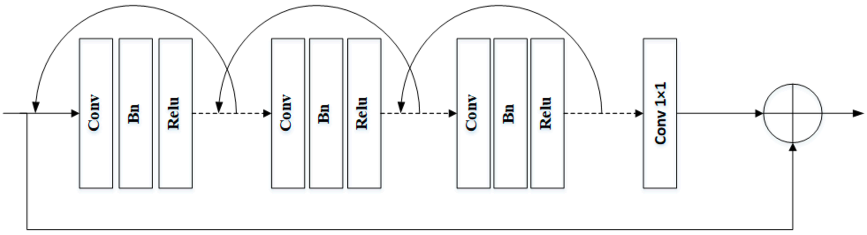


Figure 9. Schematic diagram of the structure of the RRCU.

(3) Atrous spatial pyramid pooling unit

Atrous convolution was first proposed in the Deeplab v1 model [40]. Differently from the conventional convolution operation, which extracts the features of closely-aligned pixel points in the image, atrous convolution introduces paddings by adding a 0 value in the standard convolution unit. There is an atrous convolution unit with a dilation rate of r, where the convolution kernel of size m × m is dilated to $N_m \times N_m$, and the receptive field is RF, and the formulas for the calculation of N_m and RF are shown in Figures 6 and 7, where RF_k represents the size of the receptive field and S_k represents the step size of the convolution kernel in the k-th layer; $RF_{k\text{-}1}$ represents the size of the receptive field and m_k represents the size of the convolution kernel in the *k*-th layer. This approach can expand the receptive fields in the context without increasing the computation volume, allowing the feature map output from the convolution layer to contain a larger range of feature information. In the subsequent Deeplab v2 model [41], atrous spatial pyramid pooling (ASPP) was proposed, and its structure is shown in Figure 10. Given the multi-scale features of the objects in the image, ASPP uses atrous convolution modules with a dilation rate of 6, 12, 18, and 24 to obtain the information of receptive fields at different scales, as well as to obtain the contextual information at different scales in the feature extraction of the image.

$$N_m = m + (m - 1) \times (r - 1) \tag{6}$$

$$RF_k = RF_{k-1} + (m_k - 1) \times S_k \tag{7}$$

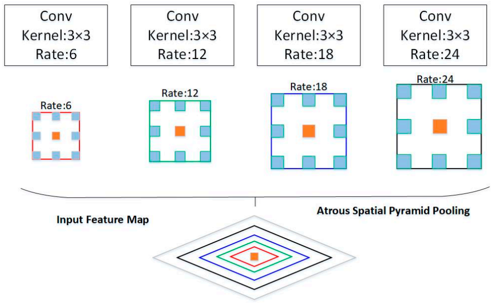


Figure 10. Structure of the ASPP unit.

(4) Attention unit

The attention unit [42], inspired by the attention mechanism of the human brain, refers to the weighted reinforcement of the target data. Over a long period of evolution, the human visual system has developed a selective mechanism; i.e., when humans see a scene, they will first make a preliminary screening of each target in the scene instead of focusing on all the targets, and when they spot salient targets, they will focus their attention on the information of these targets and thus ignore other irrelevant and secondary information. This shows that the human visual mechanism has a different distribution of attention for each target appearing in the scene, and such an attention mechanism allows the human visual system to quickly select the key information that attracts its attention from a large amount of information in a complex scene and further process such information with limited resources. Additionally, this human attention mechanism is acquired through continuous learning, a process that is extremely similar to the learning process of neural networks, the nature of which is the simulation of how a human brain works. Therefore, the attention unit can be introduced in CNN to improve the prediction precision of the model.

In principle, the attention unit in deep learning is similar to humans' selective visual attention mechanism, which selects the more important information for the current task from a large amount of information. The attention unit's identification of critical information is mainly determined by the attention weight $0 <<< \alpha << 1$, which is obtained from high-level feature maps that contain semantic information and low-level feature maps that include contextual information, with the following calculation formulas:

$$L_1 = \sigma_1 \left(W_X^T X_I^1 + W_g^t g_i^1 + b_1\right) \tag{8}$$

$$\alpha_i = \sigma^2 \left(W_T L_1 + b_2\right) \tag{9}$$

where X_i^l and g_i^l are the low-level and high-level feature maps, respectively; W_X, W_g, and W_T are linear transformation parameters; and b_1 and b_2 are bias terms. Completing the linear transformation using a 1 × 1 convolution operation can effectively reduce the number of parameters. σ_1 and σ_2 are the ReLU activation function and Sigmoid activation function, respectively, and their attention weight α is normalized to (0, 1) before it is multiplied pixel by pixel with the low-level feature map, and the results will be the output activation features.

$$L_I^1 = X_{i,c}^1 \times \alpha_i \tag{10}$$

The attention unit can automatically learn the structure of the target during model training and generates soft regions during the testing of the dataset to increase the weight of the change regions. As shown in Figure 11, the attention unit uses the semantic information of the high-level feature map to increase the weight of the features of the change regions. The unit uses a 1 × 1 convolutional layer with the introduction of very minimal parameters, which can significantly improve the sensitivity of the model.

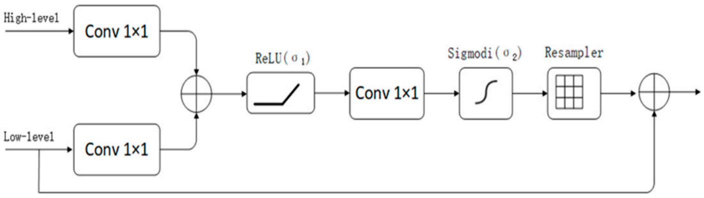


Figure 11. Schematic diagram of the attention unit.

3.5. ASPP ArrU-Net Model

Based on the model algorithm optimization and precision improvement methods mentioned above, this research optimized and improved the U-Net model by introducing a RRCU in the upsampling and downsampling stages to enhance the propagation of the features, reuse the feature maps, and improve their utilization efficiency. The ASPP unit was introduced to obtain the multi-scale contextual information of the feature maps before upsampling, and the attention unit was brought into the original shortcut connection to increase the weight of information in the change region while reducing the weight of the feature information in the non-change region to make the results more accurate.

As shown in Figure 12, the structure of the ASPP ArrU-Net model is similar to that of the U-Net model, as both consist of three components: an encoder, decoder, and shortcut connection. In the encoder, the ASPP ArrU-Net model is composed of four convolutional layers and downsampling layers, where the convolutional layers use the RRCU mentioned above to improve the problem of model degradation caused by the deepening of the network layers, reuse the extracted feature maps, and enhance the propagation of the features, allowing them to extract more complex features. In this paper, the ASPP unit was introduced at the bottom of the encoder, and four atrous convolutions with different dilation rates were used to expand the receptive field and extract the multi-scale information of the image target. The decoder was composed of four sets of convolutional layers and upsampling layers with the same convolutional layers as the encoder. A shortcut connection to merge the attention units was used between each layer of the four layers in the encoder and decoder. The shortcut connection could splice the shallow and deep features in the band dimension, and the added attention unit could increase the feature weight of the change information, thus improving the model's noise immunity. The entire model adopted the Pytorch deep learning framework, was written in Python language, and GPU calls were made during operation to improve computational efficiency.

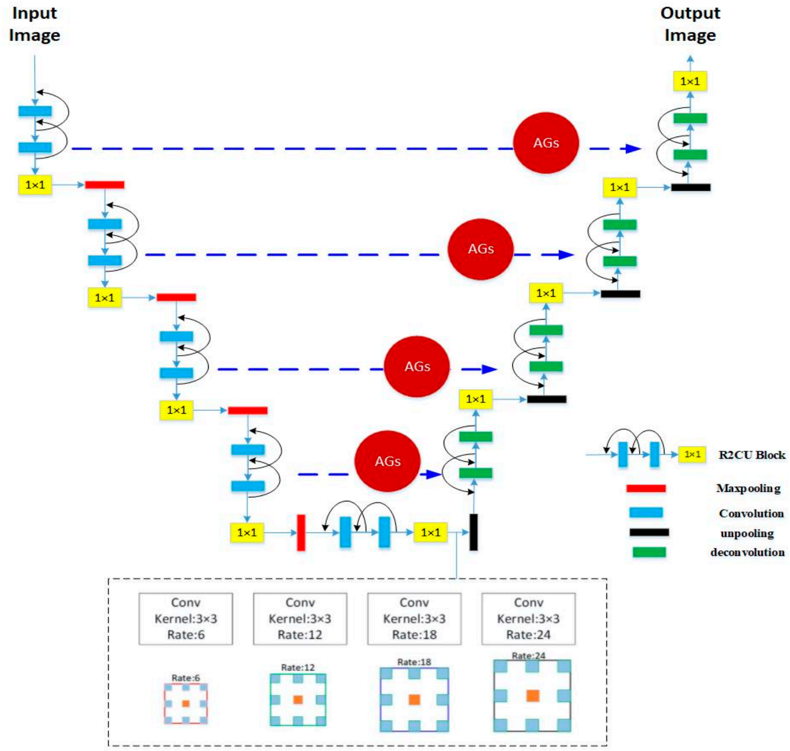


Figure 12. Structure of the ASPP ArrU-Net model.

4. Results and Discussion

4.1. Results and Analysis of Model Tentative Calculation

To verify the precision of the ASPP ArrU-Net model for terracing measure extraction in high-resolution remote sensing images, in this research we carried out experiments on images with the five models of FCN, SegNet, FC-Siam-diff, DeepLabv3+, and U-Net and compared the results for verification [43,44]. By comparing and analyzing the identification maps and extraction result maps, the study allowed the obtention of the relevant quantitative evaluation metrics from the experimental results of the five models, such as the precision, recall, omission, and F1-Score, and compared them, point by point, with those of five other commonly used studies, including the FCN model, the SegNet model, the FC-Siam-diff model, the DeepLabv3+ model, and U-Net, with the GF-1 images of Fenxi County as the experimental dataset.

In the model training process, the categorical cross-entropy loss function was used, with the Adam function chosen as the parameter optimizer. Additionally, the initial learning rate was set at 1×10^{-4}, with the change index of the learning rate [45] (gamma) being introduced. If the learning rate is too large, it will lead to difficult convergence and the possible improbability of finding the optimal solution; on the contrary, if the learning rate is too small, it will lead to slow learning and waste a great deal of arithmetic power. Therefore, this research introduced a learning rate change indicator to achieve the dynamic change in the learning rate. In the early stage, a larger learning rate was used to improve the speed of convergence, and when the loss value of the verification dataset no longer dropped, the learning rate was decayed to reduce the change magnitude to identify the optimal solution and improve the decoding precision. The decayed learning rate was equal to the product of the initial learning rate and the learning rate change indicator. Finally, when the precision of the verification dataset no longer improved, the early stopping approach was used to end the training. Due to the small amount of data in the experimental area, the running time of the entire model was about 16 h, including the total running time of the ASPP

ArrU-Net model and other comparative models. The individual running time of the ASPP ArrU-Net model was approximately 4–5 h. Because of the fact that the operation of the model is not only related to the complexity of the model, but also to the computer hardware configuration and runtime computer state, the time required for each test was different.

In the image verification region, all six models could extract the rough contours of the terrace polygons from the image identification map as shown in Figure 13. Particularly, the polygons extracted by the FCN model showed apparent jagged splicing traces; the results of the SegNet model displayed large missing areas; the results of the U-Net method and those of the FC-Siam-diff model showed large false areas; and the DeepLabv3+ model demonstrated blurred object boundaries. Comparatively, the results of the ASPP ArrU-Net model were closer to those of the change reference map and generated the extraction results with fewer omission and false areas, despite a few problems including blurred boundaries and the adhesion of change areas. From the perspective of quantitative analysis, the proposed model in this research outperformed the other five models in terms of precision, recall, and F1-Score. As shown in Table 3, the F1-Score was calculated by combining precision and recall, and the result was 12.80%, 6.17%, 9.73%, 7.76%, and 5.27% higher than the other five models, in corresponding order, while the omission rate was also the lowest compared to that of the other five models.

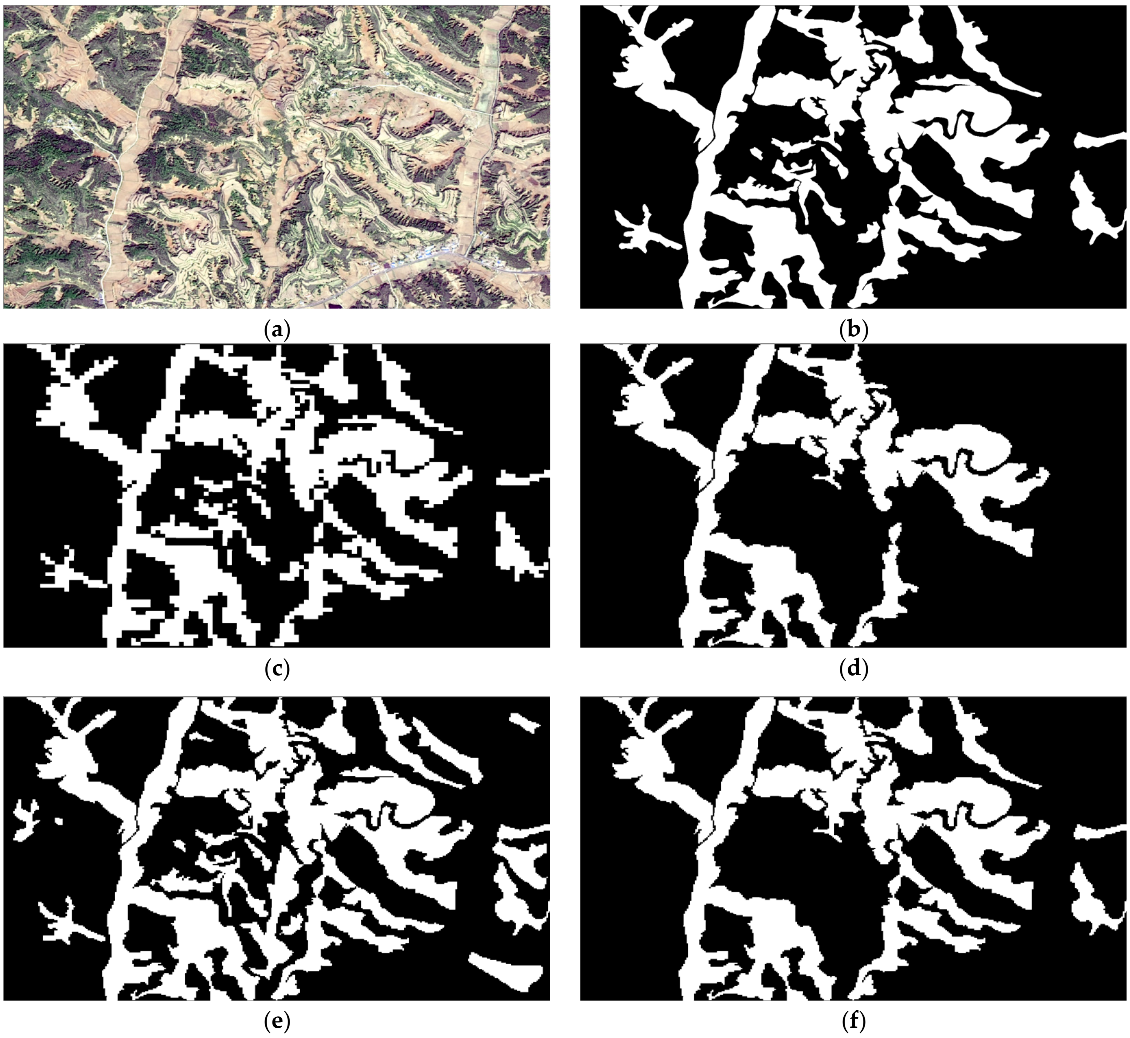

(**a**) (**b**)
(**c**) (**d**)
(**e**) (**f**)

Figure 13. *Cont.*

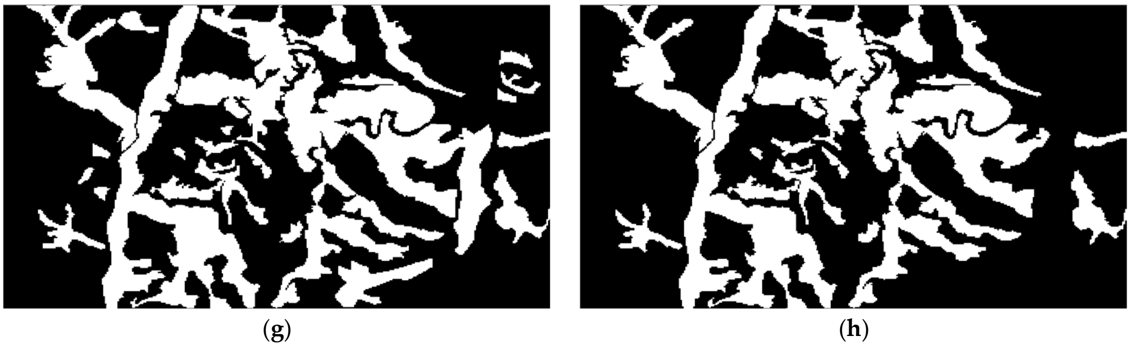


Figure 13. Binary map of the image extraction results of the verification area. (**a**) Satellite image; (**b**) image identification map; (**c**) FCN; (**d**) SegNet; (**e**) FC-Siam-diff; (**f**) DeepLabv3+; (**g**) U-Net; (**h**) ASPP ArrU-Net.

Table 3. Precision comparison of the extraction results in Fenxi County.

Method	Precision	Recall	Omission	F1-Score
FCN	0.568	0.778	0.222	0.657
SegNet	0.621	0.865	0.135	0.723
FC-Siam-diff	0.576	0.852	0.148	0.687
DeepLabv3+	0.568	0.936	0.064	0.707
U-Net	0.627	0.879	0.121	0.732
ASPP ArrU-Net	0.656	0.976	0.024	0.785

The main reason behind the above situation is that the original models mainly extract the features of the image through convolution layers and sample the extracted feature map through pooling layers. After multiple groups of convolution and pooling operations, multi-scale features of the image can be obtained. However, when the feature map is upsampled and downsampled by simply using the pooling operation, it only restores the feature map to its size when inputted into the model, which will lead to the loss of information. Especially for high-resolution remote sensing images, which contain complex object information, and because the objects they contain, such as terraces, some cultivated land plots, garden areas, and even forest and grass plots, have similar color and texture features, the error of model prediction increases.

To address the above situation, we made structural improvements to the original U-Net model, mainly in the following three areas. (1) The convolution layers of the model in the upsampling and downsampling process were changed into recurrent residual convolutional layers, and the features extracted from the convolution layers were reused to fully learn the features of the image and improve the efficiency of using the feature map. The introduction of the residual unit allowed the model to effectively train deeper network structures and prevent certain relative problems, such as model degradation due to deeper networks. (2) Without reducing the spatial dimension, this research introduced the ASPP unit to expand the receptive field by using different dilation rates, thus acquiring multi-scale image features and improving the segmentation performance of the network model. (3) The purpose of the original shortcut connection of the U-Net model is to splice the low-level and high-level features in the band dimension to achieve feature fusion. However, it does not take into sufficient account the spatial consistency, which will lead to certain problems, such as loss of edge details in the predicted images. Therefore, this paper changed the original shortcut connection of the U-Net model and introduced an attention unit in the shortcut connection process, where the attention unit could adjust the weight of each component in the feature map, suppressing the learning of task-irrelevant features by decreasing their weights and enhancing the learning of task-relevant features to increase their learning. To sum up, the model proposed in this paper has a higher precision, recall, omission, and F1-Score than the other five models, which verifies its effectiveness and superiority.

4.2. Analysis of Model Generalization Ability

To verify the generalization ability of the ASPP ArrU-Net model for high-resolution remote sensing images, this research used terracing measures as the extraction target and the SegNet and U-Net models with a precision higher than 60% as the comparison to conduct a terracing measure extraction experiment and comparison verification on the high-resolution image of Wafangdian City in Liaoning Province. All the terracing measures were manually mapped in the area while 40% of them were taken as the training samples and the remaining 60% were taken as the validation samples, establishing a sample library of deep learning models. Then, the models were imported for calculation and obtained the relevant quantitative evaluation indicators, such as the precision, recall, omission, and F1-Score of the extraction results of the three models. Some of the images and identification maps of the verification areas are shown in Figure 14, and the comparison results of the extraction precision of the models are shown in Table 4.

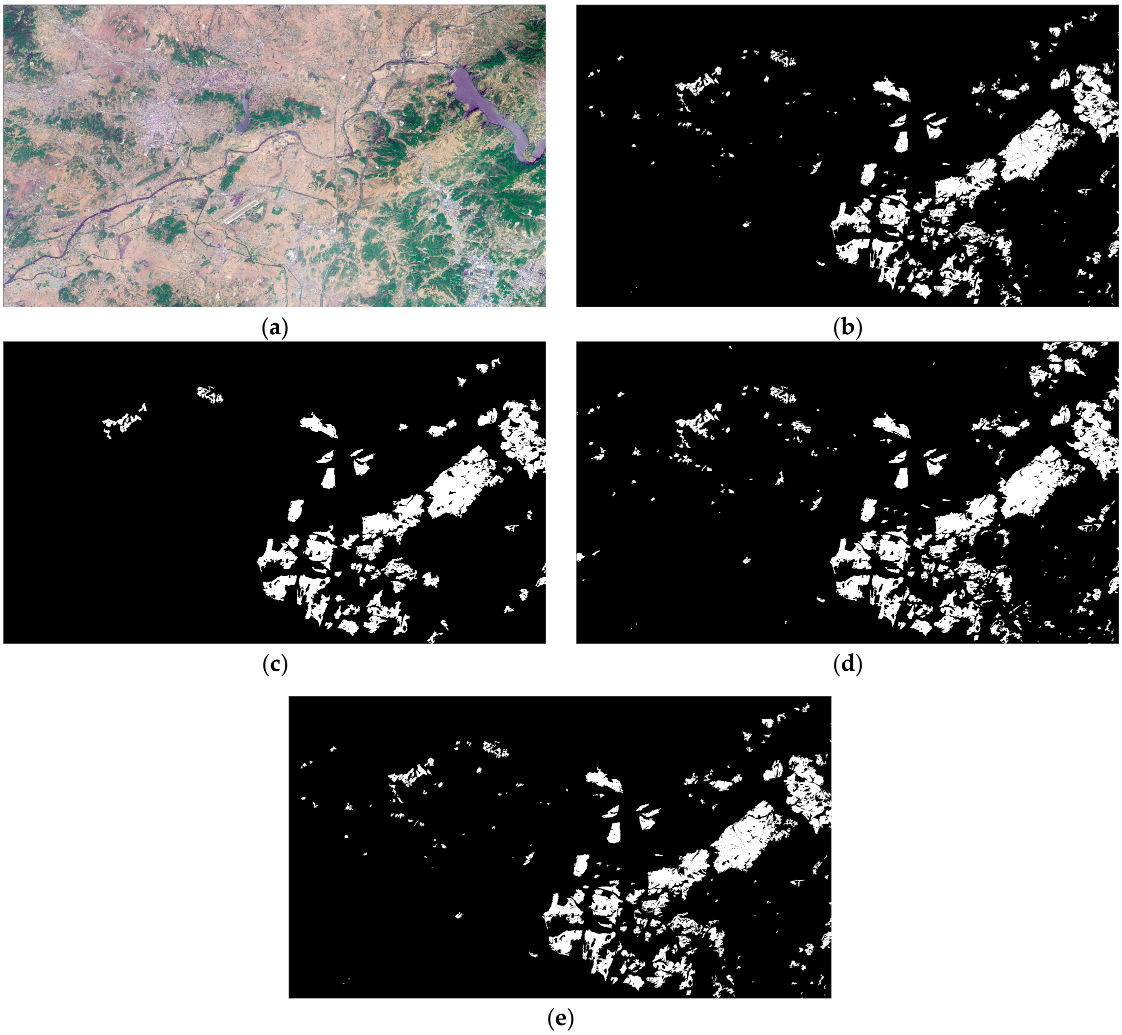


Figure 14. Binary image of image extraction results. (**a**) Satellite image; (**b**) image identification map; (**c**) SegNet; (**d**) U-Net; (**e**) ASPP ArrU-Net.

Table 4. Precision comparison of the extraction results in Wafangdian City.

Method	Precision	Recall	Omission	F1-Score
SegNet	0.594	0.847	0.153	0.698
UNet	0.612	0.856	0.144	0.714
ASPP ArrU-Net	0.635	0.903	0.097	0.746

From the extraction results, the SegNet model still displays some omissions and the UNet model show excessive detections, while the results of the ASPP ArrU-Net model are closest to those of the identification map, indicating that its results are better than those of the other three models. From the precision statistics, the precision of the ASPP ArrU-Net model is 4.74% and 3.19% higher than that of the SegNet and UNet models, respectively. The extraction results from the ASPP ArrU-Net model were compared with the human interpretation results, and the data analysis is shown in Table 5.

Table 5. Result comparison of the ASPP ArrU-Net model and human visual interpretation.

Evaluation Indicator	Interpretation Precision	Interpretation Time
Human interpretation	95%	96 h
ASPP ArrU-Net	89.5%	14 h

Due to the complex features of measure images, human interpretation requires remote sensing interpretation based on one's experience, with errors and omissions here and there. According to the field verification results, the precision of human visual interpretation was about 95%, and the interpretation time was 96 h. Comparatively, the extraction precision of the ASPP ArrU-Net model was 89.5%, and the time was 14 h (the time may vary subject to different hardware conditions). The precision of the latter results met the requirement for the extraction of soil and water conservation measures from high-definition remote sensing images, allowing the achievement of a certain level of precision and significantly shortening the interpretation time, indicating improved interpretation efficiency.

4.3. Discussion

The interpretation results of high-resolution images are not only related to the choice of the model but also influenced by the quality of the data. In addition, the preparation of the samples also has a direct impact on the precision of the final model prediction. Although the model constructed in this research has shown better results in measure extraction for high-resolution remote sensing images, it cannot be concluded that this model will necessarily perform better than other models in measure extraction. The precision of the same model can vary with different data sets and in different regions; i.e., the robustness and generalization ability of the model should be improved, which is a key issue for future research.

For studies on the intelligent interpretation of remote sensing images, the preparation of the training samples is a tedious and time-consuming issue, and future research must investigate how small samples can be used to train models with higher precision, which is also a popular research topic in the current deep learning field. In addition, scholars can also consider establishing remote sensing image databases of different object types to improve the generalization ability of the training models and provide a more reliable data basis for the intelligent interpretation of remote sensing images.

5. Conclusions

This research reviews and analyzes intelligent interpretation algorithms and deep learning semantic segmentation models for remote sensing images. To improve the interpretation precision and automation, an improved semantic segmentation model of deep learning was applied in the intelligent interpretation of high-resolution remote sensing images and an intelligent interpretation algorithm model for terracing measures in high-resolution remote sensing images was constructed. Additionally, the study was based on Fenxi County of Shanxi Province as the target for conducting the experiments and for comparing the model and other models. The results showed that the improved algorithm model had a higher precision. Furthermore, Wafangdian City of Liaoning Province was used as the experimental target to investigate the generalization ability of the model and a comparative analysis was carried out with human interpretation. The results showed that the model had a better generalization ability and could significantly improve the interpretation efficiency.

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

Funding: This research was funded by the Basic R&D Special Fund of the Central Government for Non-profit Research Institutes, grant/award no. HKY-JBYW-2023-08, Excellent Young Talents Project of Yellow River Conservancy Commission (no. HQK-202308), and High-Resolution Earth Observation System Major Special Project (no. 00-Y30B01-9001-22/23-CY-06).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: All data are available within this article.

Acknowledgments: The authors would like to acknowledge the developers of PyTorch for their open-source projects.

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

References

1. Jiang, E.H.; Cao, Y.T.; Dong, Q.H.; Gao, G.M.; Li, J.H.; Jiang, S.Q. Long-Term Effects of the Yellow River Sediment Resources Utilization. *Yellow River* **2015**, *37*, 1–5+12.
2. Jiang, E.H.; Wang, Y.J.; Zhang, Y.F.; Cao, Y.T. New Advances in Yellow River Sediment Research. *Yellow River* **2016**, *38*, 24–31.
3. Chen, Y.P.; Fu, B.J. Reflections on the Construction of Ecological Civilization in the Yellow River Basin. *China Sci. Dly.* **2019**, *6*, 1–7.
4. Xiao, P.Q.; Wang, L.L.; Yang, J.S.; Jiao, P.; Wang, Z.H. Study on Sediment Reduction Benefits of Soil and Water Conservation Measures in Typical Watersheds in the Loess Plateau under the Heavy Rainfall. *J. Hydraul. Eng.* **2020**, *51*, 1149–1156.
5. Xiao, P.Q.; Lv, X.Z.; Zhang, P. Progress and Results of Scientific Research on Soil and Water Conservation in the Yellow River Basin. *Soil Water Conserv. China* **2020**, *10*, 6–9+82.
6. Li, Z.B.; Wei, X. Prediction of Environmental Evolution Trends of Soil Erosion in Loess Plateau Areas. The Interaction and Role of Water and Socio-economic Development. In Proceedings of the Third National Symposium on Water Research, Chinese Academy of Sciences, Beijing, China, October 2005.
7. Jiang, D.S.; Fan, X.K.; Huang, G.J. Experiment of the Evaluation on Benefits of Soil and Water Conservation Measures for Slope Land I. The Effect of Soil and Water Conservation Measures on Rainfall Infiltration in Slope Land. *J. Soil Water Conserv.* **1990**, *2*, 1–10.
8. Li, D.; Tong, Q.; Li, R.; Gong, J.; Zhang, L. Current Issues in High-resolution Earth Observation Technology. *Sci. Sin. Terrae* **2012**, *42*, 805–813. [CrossRef]
9. Dang, T.M.; Mu, X.M.; Sun, W.Y. Review of Quickly Discriminating Approaches of Terrace Information Based on High-Resolution Remote Sensing Images. *Yellow River* **2017**, *39*, 85–89+94.
10. Yu, H.; Liu, Z.H.; Zhang, X.P.; Li, R. Extraction of Terraced Field Texture Features Based on Fourier Transformation. *Remote Sens. Land Resour.* **2008**, *2*, 39–42.
11. Xue, M.D. Research on Terraced Field Surface Extraction Based on Object Base Image Analysis of UAV Images. Master's Thesis, Northwest A & F University, Xi'an, China, 2018.
12. Hu, Y. Research on Terraced Field Extraction Method for UAV Images and Slope Data. Master's Thesis, Northwest A & F University, Xi'an China, 2018.
13. Eckert, S.; Ghebremicael, S.T.; Hurni, H.; Kohler, T. Identification and classification of structural soil conservation measures based on very high-resolution stereo satellite data. *J. Environ. Manag.* **2017**, *193*, 592–606. [CrossRef]
14. Zhang, Y.G.; Wang, F.; Sun, W.Y.; An, C.C. Terrace Information Extraction from SPOT Remote Sensing Image Based on Object-oriented Classification Method. *Res. Soil Water Conserv.* **2016**, *23*, 345–351.
15. Zhao, W.D.; Tang, G.A.; Xu, Y.; Zhou, C.Y.; Qian, J.Z.; Ma, L. Terrace Morphological Characteristics and Its Comprehensive Digital Classification. *Bull. Soil Water Conserv.* **2013**, *33*, 295–300.

16. Yang, Y.N. Research on UAV Remote Sensing Terraced Field Identification Methods Based on Semantic Segmentation. Master's Thesis, Northwest A & F University, Xi'an China, 2020.
17. Wang, J.; Zhou, Z.H.; Zhou, A.Y. *Applied Machine Learning*; Tsinghua University Press: Beijing, China, 2009.
18. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
20. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
21. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.-R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
22. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv* **2017**, arXiv:1704.06857.
23. Liu, Y.; Fu, Z.Y.; Zheng, F.B. Scene Classification of High-resolution Remote Sensing Image Based on Multimedia Neural Cognitive Computing. *J. Syst. Eng. Electron.* **2010**, *37*, 2623–2633.
24. Guo, W.Y.; Wei, Y.T.; Liu, Z.K.; Xu, G.P.; Nie, X.S.; Zhao, H.X.; Yao, R.T. Analysis of the Ability of Flood Prevention of the Farmland Behind Silt-Arrester in Fenxi County and Investigation of the Experience of Securing Harvest. *Soil Water Conserv. China* **1993**, *8*, 28–31+66.
25. Wang, J.; Ding, K.; Li, S.; Gao, J. Construction of Ecological Security Pattern of Village Landscape in Loess Plateau Based on Geological Sensitivity Evaluation—A Case Study of Houjialou Village of Shanxi Province. *Landsc. Archit. Acad. J.* **2021**, *38*, 54–59.
26. Zhang, Y.; Shi, M.; Zhao, X.; Wang, X.; Luo, Z. Methods for automatic identification and extraction of terraces from high spatial resolution satellite data (China-GF-1). *Int. Soil Water Conserv. Res.* **2017**, *5*, 17–25. [CrossRef]
27. Li, Z.F. Application of GF-1 Satellite in Land-use Remote Sensing Monitoring. *Land Resour. Her.* **2015**, *12*, 85–88.
28. Ji, S.P.; Wei, S.Q. Building Extraction Via Convolutional Neural Networks from An Open Remote Sensing Building Dataset. *Acta Geod. Et Cartogr. Sin.* **2019**, *48*, 448–459.
29. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
30. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
31. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), Savannah, GA, USA, 2–4 November 2016; USENIX Association: Berkeley, CA, USA, 2016.
32. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G. PyTorch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration. Personal Communication. 2017. Available online: https://gitpiper.com/resources/python/deeplearning/pytorch-pytorch (accessed on 7 April 2023).
33. Mi, L.; Chen, Z. Superpixel-enhanced deep neural forest for remote sensing image semantic segmentation. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 140–152. [CrossRef]
34. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 640–651.
35. 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.
36. Lei, T.; Zhang, Y.; Lv, Z.; Li, S.; Liu, S.; Nandi, A.K. Landslide inventory mapping from bitemporal images using deep convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2019**, *16*, 982–986. [CrossRef]
37. Lyu, H.; Lu, H.; Mou, L.; Li, W.; Wright, J.; Li, X.; Li, X.; Zhu, X.X.; Wang, J.; Yu, L.; et al. Long-term annual mapping of four cities on different continents by applying a deep information learning method to Landsat data. *Remote Sens.* **2018**, *10*, 471. [CrossRef]
38. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
40. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *Comput. Sci.* **2014**, *4*, 357–361.
41. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
42. Qiu, J.; Wang, B.; Zhou, C. Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE* **2020**, *15*, e0227222. [CrossRef] [PubMed]
43. Daudt, R.C.; Le Saux, B.; Boulch, A. Fully convolutional siamese networks for change detection. In Proceedings of the 2018 25th IEEE International Conference on Image Processing, Athens, Greece, 7–10 October 2018.

44. 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, Munich, Germany, 8–14 September 2018.
45. Maxwell, A.E.; Warner, T.A.; Fang, F. Implementation of machine-learning classification in remote sensing: An applied review. *Int. J. Remote Sens.* **2018**, *39*, 2784–2817. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.