International Journal of Geo-Information

MDPI

Article

An Automatic Extraction Method for Hatched Residential Areas in Raster Maps Based on Multi-Scale Feature Fusion

Jianhua Wu, Jiaqi Xiong *, Yu Zhao and Xiang Hu

School of Geography and Environment, Jiangxi Normal University, Nanchang 330022, China; wujianhua2021@jxnu.edu.cn (J.W.); 201940100068@jxnu.edu.cn (Y.Z.); 202040100120@jxnu.edu.cn (X.H.)
* Correspondence: 201940100061@jxnu.edu.cn; Tel.: +86-189-4235-6689

Abstract: Extracting the residential areas from digital raster maps is beneficial for research on land use change analysis and land quality assessment. In traditional methods for extracting residential areas in raster maps, parameters must be set manually; these methods also suffer from low extraction accuracy and inefficiency. Therefore, we have proposed an automatic method for extracting the hatched residential areas from raster maps based on a multi-scale U-Net and fully connected conditional random fields. The experimental results showed that the model that was based on a multi-scale U-Net with fully connected conditional random fields achieved scores of 97.05% in Dice, 94.26% in Intersection over Union, 94.92% in recall, 93.52% in precision and 99.52% in accuracy. Compared to the FCN-8s, the five metrics increased by 1.47%, 2.72%, 1.07%, 4.56% and 0.26%, respectively and compared to the U-Net, they increased by 0.84%, 1.56%, 3.00%, 0.65% and 0.13%, respectively. Our method also outperformed the Gabor filter-based algorithm in the number of identified objects and the accuracy of object contour locations. Furthermore, we were able to extract all of the hatched residential areas from a sheet of raster map. These results demonstrate that our method has high accuracy in object recognition and contour position, thereby providing a new method with strong potential for the extraction of hatched residential areas.

Keywords: residential area extraction; historical raster map; deep learning; geographical information; image segmentation

Citation: Wu, J.; Xiong, J.; Zhao, Y.; Hu, X. An Automatic Extraction Method for Hatched Residential Areas in Raster Maps Based on Multi-Scale Feature Fusion. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 831. https://doi.org/10.3390/ijgi10120831

Academic Editor: Wolfgang Kainz

Received: 21 October 2021
Accepted: 10 December 2021
Published: 10 December 2021

1. Introduction

Many geographic information mapping departments store their maps in the form of a digital raster graphic (DRG). These historical raster maps contain rich and valuable historical geographic information, such as the locations of residential areas, topography, vegetation, roads and toponyms. This information can be used in many applications and research fields in order to express the spatial structure of a range of geographic information and its interconnections. Because historical raster maps can also show changes over time, which is a mobile history, they play a key role in urban construction, transportation planning, scientific research, resource exploration, national defense and military command. Residential areas are of significance to national economic construction, culture and science and national defense and military and, furthermore, they are one of the most important elements in maps. The shape of the external boundary and the internal structure of the residential areas together form the representation of the residential areas in the map. The method of representation of residential areas varies with the map scale. Usually, the interior of a residential area is represented by filled diagonal hatched lines in topographic maps at 1:50,000 map scale and other map scales. As shown in Figure 1, the hatched lines are oriented at a certain angle with the horizontal direction of the image and the internal hatched lines of the residential area show a periodic texture structure [1]. This paper primarily focuses on the method of extracting residential areas with hatched lines in historical raster maps.

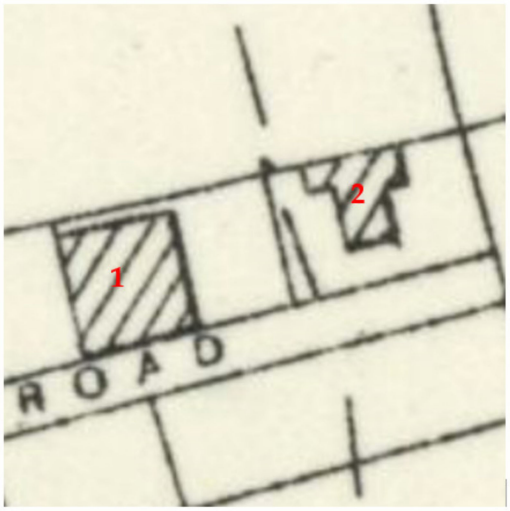


Figure 1. Residential areas with hatched lines (element 1 is a regular quadrilateral residential area and element 2 is an irregular polygonal residential area).

Since the 1990s, there have been many studies on the automatic vectorization of raster map elements. Hu S. et al. [2] proposed a method for applying frequency domain matching in order to identify residential areas by using the characteristics of clump structure and fixed filling patterns; they implemented a system for identifying residential areas in maps accordingly. Liu X. et al. [3] proposed an integrated method for the segmentation and recognition of house targets, which used the directionality of the hatched lines and the fixity of hatched lines' spacing. The prerequisite for this method is being able to correctly detect each hatched line, but the accurate detection of each hatched line in a more complex topographic map remains a challenge. In addition, the assumption that the contours of neighborhood-type residential areas cannot be disconnected is not realizable in the actual topographic map. Fu Z. et al. [4] proposed a method for extracting the target contours of houses by using manual determination of the starting points inside the residential areas. Zheng H. [5] proposed a neighborhood-based residential area recognition algorithm. It extracted the hatched lines according to the mathematical morphological decomposition method and node features, then performed the transformation of specific directions. Although this method reduced the rate of misidentification cases, it only considered the case where one end of the hatched line was disconnected by other map symbols and did not study the recognition of polygons with incomplete hatched lines. As a result, its application in practical engineering is limited.

Yang Y. et al. [6] used vector data compression and the total least squares method in order to fit block boundaries so as to obtain the final vector data of the residential areas. However, this method cannot automatically handle the problems of sticking, deinking and intersection with other elements in some residential blocks and it requires manual data acquisition and processing. Wu J. et al. [7] proposed a Gabor filter-based algorithm of residential areas using the characteristics of hatched lines, which automatically extracted residential areas with hatched lines by Gaussian smoothing, binarization, erosion, image logic operations and tracking the boundary pixel based on the pixel-value relationship of the neighbor pixels. However, this method still required the manual setting of parameters such as the maximum width of feature adhesion, length of bar detector and filling area according to the resolution of raster map; in addition to this, the straightening effect of the extracted residential contour lines needed to be improved.

Much of the research on the automatic identification of residential area elements in raster topographic maps still has the following problems:

- Lack of adaptability: It is difficult to apply the identification method to cases with low image resolution, blurred hatched lines after magnification or irregular structure

of the hatched lines in residential areas (e.g., unequal spacing of the hatched lines or broken hatched lines);

- Poor identification ability: The shapes of the identified residential areas are not ideal, the accuracy of the location of the residential areas contour is not high and it is easy to incorrectly identify map elements that are similar in structure to polygons with hatched lines;
- Impracticality: It is difficult for the parameter-dependent method to perfectly grasp the threshold value and it is also difficult to obtain good results in the segmentation task of complex and large images. Moreover, because most of the literature has involved carrying out experiments on partial regions of the entire map image, it was difficult to process map sheet-level raster images and the performance of the algorithms was also less than ideal, which led to the reduced applicability of the algorithms.

Due to their special structure of local weight sharing and good fault tolerance, parallel processing capability and self-learning ability, convolutional neural networks (CNNs) [8–10] are now widely used in image classification [11,12], image detection [13] and image semantic segmentation [14–16]. For urban remote sensing research, fully convolutional networks (FCNs) can directly classify each pixel and greatly improve the efficiency and accuracy of building extraction [17]. Some studies have adapted deep learning architectures in order to enhance their abilities to solve the discontinuity issue of road extraction [18]. The rapid evolution of deep learning has opened up new possibilities for the efficient identification and extraction of residential areas with hatched lines in raster maps. The classical U-Net is widely used in image segmentation because it can achieve high segmentation accuracy with a small number of samples. Although the classical U-Net network uses skip connections to fuse feature information from the encoding layers and decoding layers, the up-sampling of a single chain cannot fully convey the scale information and make good use of the feature information at other scales of the CNN [19].

Therefore, we investigated an improved semantic segmentation model for U-Net networks—multi-scale U-Net with fully connected conditional random fields (MSU-Net-CRFs)—and proposed a method for the automatic extraction of hatched residential areas (AEHRA) in raster maps. The AEHRA method makes full use of the multi-scale information of the image and the neighborhood information of the pixels in order to improve the accuracy of its residential area segmentation. It extracts the hatched residential areas from a sheet of raster maps by leveraging the chunking and merging strategies. This enhances the practicality of the method.

2. Methods

2.1. The AEHRA Approach

The AEHRA workflow, as shown in Figure 2, was divided into three stages: data input and processing, MSU-Net-CRFs model building of hatched residential area segmentation and result output and processing.

In the first stage, there were two types of data that were input: the sample that was input for model training and the input of the tile image set from a sheet of raster maps that was used when the model was applied. The method that was utilized for the sample data set production will be described in Section 2.2.

In the second stage, the AEHRA trained the MSU-Net network model. The training set was put into the network model and the model then predicted the input raster images and compared the predicted values with the true values, calculated the loss values and updated the weights of the network through the optimizer. When all of the training samples had been used for model training, one iteration was considered to have been completed. By analyzing the results of the evaluation metric that was output by the validation set, a determination was made as to whether the model itself would be output. If the evaluation metrics met the requirements, then the model was output; if they did not meet the requirements, then the iterative training process was continued.

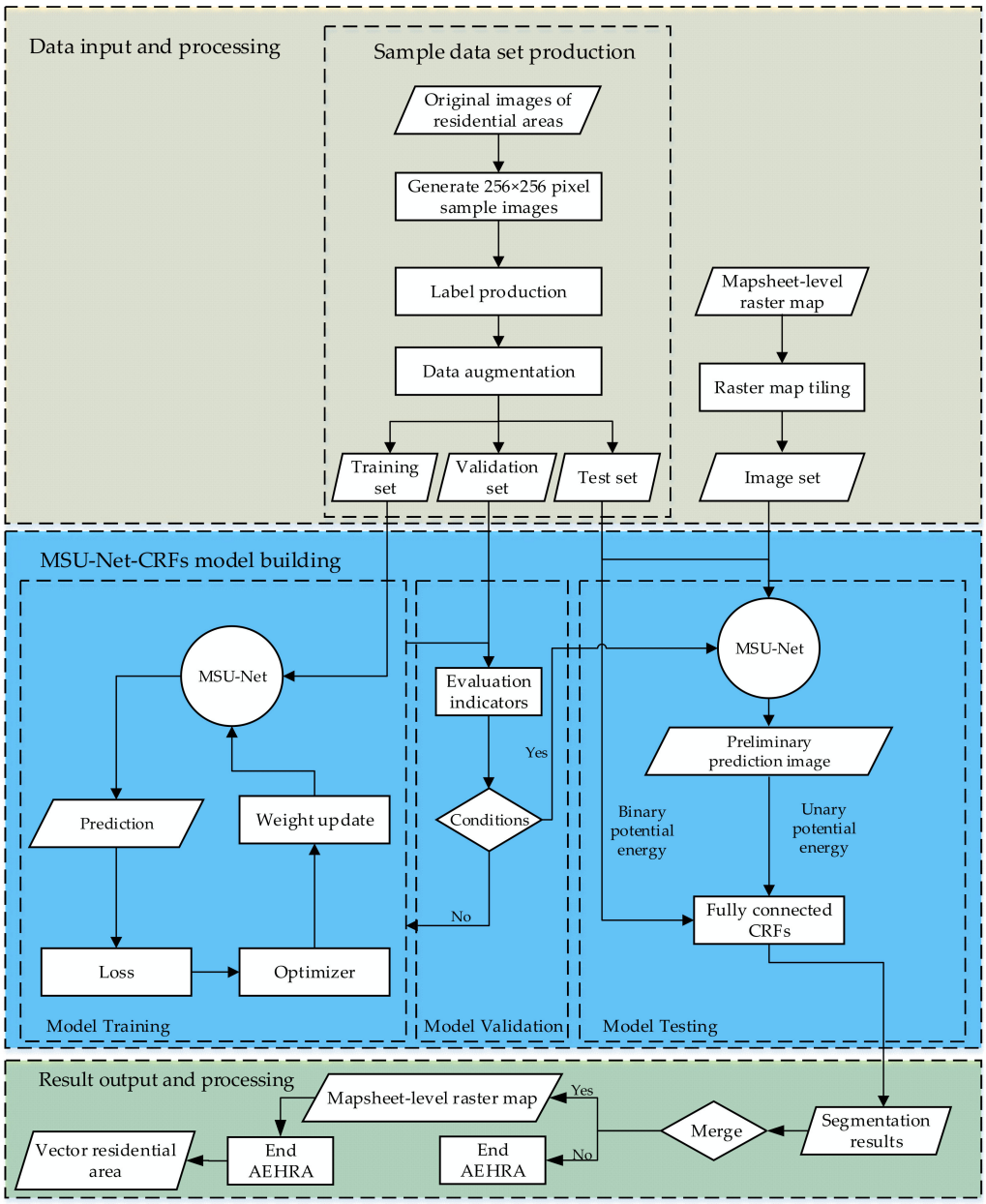


Figure 2. Workflow of AEHRA.

In the testing of the AEHRA model, the test set was input to the constructed MSU-Net network in order to obtain the preliminary prediction image and the preliminary prediction image was then used as the input of the unary potential in the fully connected CRFs. The position and color information in the binary potential energy were provided by the original image and the final segmentation results were output after the output of the fully connected CRFs.

In the third stage, there were two types of exporting prediction result images: the single raster image with a small size of 256 × 256 pixels, which the model directly output, and the merged prediction result image of a sheet of map. Finally, the identified raster residential areas were vectorized and simplified.

2.2. Sample Data Set Production Method

The sample data set needed to be created before the sample input. First, due to computer memory limitations, it was necessary to slice the raster maps into residential area sample images with a size of 256 × 256 pixels at different scales and to manually label each

image after the slice by using the LabelMe tool. In CNNs, insufficient training data makes the network difficult to train and prone to overfitting, i.e., if the model over-fits the data of the training set, it will decrease the accuracy of the prediction on the test set. Performing random data enhancement operations on both the original image and the corresponding labeled image, including performing random rotation, mirroring and random zooming on both the original images and the corresponding labeling images, reduced the sensitivity of the training dataset. Figure 3a shows the unprocessed original image, Figure 3b shows the image after vertical mirror flipping, Figure 3c,d show the images after rotation and Figure 3e shows the image after random zooming.

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

Figure 3. Training data set. (**a**) Original image; (**b**) image after flipping vertically; (**c**) image after rotating 270°; (**d**) image after rotating 90°; (**e**) image after random zooming.

2.3. Architecture of the MSU-Net

The classical U-Net network is a semantic segmentation network based on FCN [20] with a U-shaped network structure, which was proposed by Ronneberger et al. [21] in 2015 and initially used in medical image segmentation due to its ability to attain high segmentation accuracy with a small number of samples. The U-Net network structure that was utilized in this study consisted of three parts: the encoder, decoder and skip connection. A feature map represented the detection of a feature and the strength of a value in a feature map represented a response to the strength of the current feature. In the encoding part, the number of feature maps was doubled through a max pooling down-sampling operation and, at the same time, the image size was compressed twice as much as it previously was. Then, the feature extraction was performed. In the decoding process, the abstract feature map in the down-sampling process was expanded twice by deconvolution and was concatenated with the feature map of the equivalent stage in the encoder stage to achieve feature fusion. In the final output layer, the low-resolution feature map was restored to the input image resolution through multiple up-sampling operations and the feature map was convolved by a 1×1 convolution kernel in order to generate the same number of channels as the classification category. Because of the shallow localization and deep segmentation, the feature FCN could ensure that the pixel position information was not lost by the skipping connection and could use feature fusion to improve the segmentation accuracy. However, the classical U-Net only classified the pixel semantics in the last output layer. Although the U-Net used the skip connection to fuse the feature information of the previous layers, the single-chain up-sampling could not fully transfer the scale information and make good use of the feature information at the other scales of the CNN.

Based on this, we designed an MSU-Net model that extracted, saved and combined the final classification feature maps at each resolution scale. Through the superimposition of the feature maps from the different scales, the final prediction image incorporated information from multiple scales. Because the feature maps at each scale play an active role in back propagation and network parameter updating, the MSU-Net network could extract the residential areas more accurately. The MSU-Net model structure is shown in Figure 4.

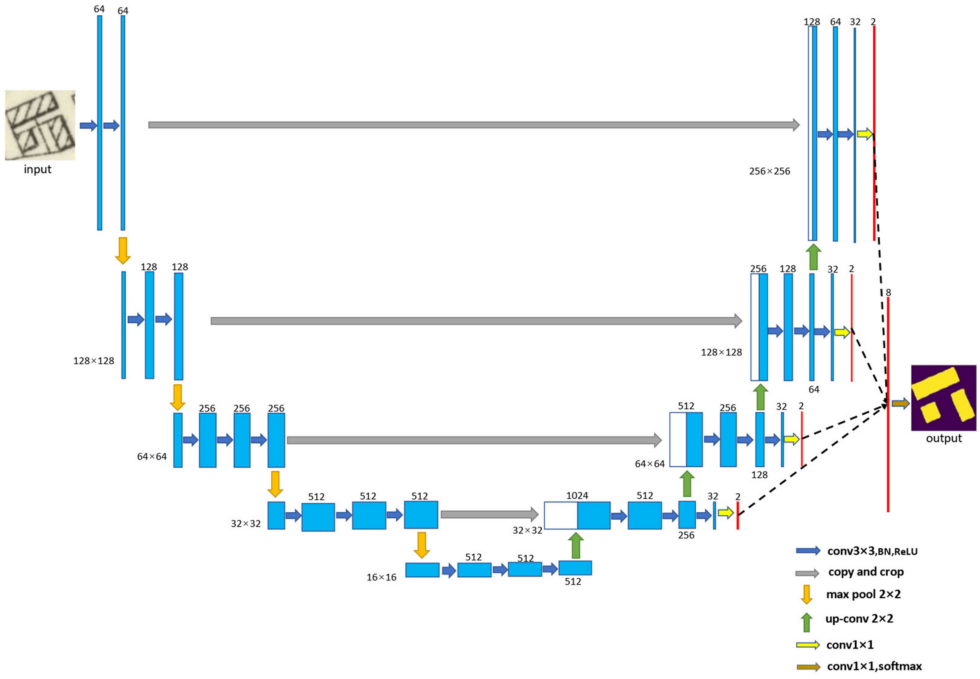


Figure 4. Structure of MSU-Net.

The MSU-Net network used the same encoder-decoder paradigm as the classical U-Net network. In the encoding part, the original VGG16 [22] network had five pooling layers. The MSU-Net removed the last pooling layer of the VGG16 network as the encoding layer of the network and the VGG16 network as the feature extractor of the encoding stage was compared to the feature extraction part of the classical U-Net. The VGG network was deeper and could extract high-dimensional feature information from the hatched residential areas. This part consisted of four down-sampling layers and 13 convolutional layers. The convolutional kernel size was set to 3×3. The small sized convolutional kernel had a small receptive field, which could better extract the boundary feature information. The use of the rectified non-linear unit (ReLU) [23] function for nonlinear activation was able to prevent gradient disappearance and gradient explosion by introducing a batch normalization (BN) layer after each convolution operation. Then, after multiple down-sampling operations, as the image size decreased, the receptive field increased and the MSU-Net network could learn deeper and more abstract semantic information of the image. In the decoding process, the image size could be restored to the original input image size by four up-sampling operations and the feature maps of the decoding and coding layers at the same level were fused by skip connection after each up-sampling operation. Two feature maps at this scale were output after three convolution operations and a 1×1 convolution operation.

Because there were four up-sampling operations, we could obtain four groups of two feature maps at different resolution scales. The four groups of feature maps were restored to their original image size by a single or multiple up-sampling operation(s) and these four groups of original-sized feature maps were summed up, forming eight feature maps. The final prediction image was obtained by 1×1 convolution and the softmax activation function. After this series of operations, the final prediction image aggregated the information from each scale, which improved the accuracy of the image segmentation.

2.4. Fully Connected CRFs Post-Processing

The U-Net performed up-sampling on the feature map several times during the decoding process, which restored the feature map to the input image size, but it made the segmentation result too smooth and generated blurred object boundaries. In addition, the convolution process could only provide the association information of the pixels in a certain region because of the restricted size of the convolution kernel. Because the size of

the convolution kernel was unchanged and the image size grew smaller with each pooling operation, the convolution kernel receptive field became larger. Although the receptive field increased with each pooling layer during the down-sampling, it was not possible to associate a pixel with all pixels, even at the last convolution layer. Therefore, based on the MSU-Net model, we introduced the fully connected CRFs [24–27] in order to improve the accuracy of the image segmentation by calculating the distance similarity and color similarity between two neighboring pixels in order to determine whether they belonged to the same class. The unary potential energy of the fully connected CRFs was the preliminary prediction map that was output by MSU-Net and the binary potential energy was sourced from the original image. The energy function of the fully connected CRFs was:

$$E(x) = \sum_{i \in V} \psi_U(x_i) \sum_{i \in V, j \in N_i} \psi_p(x_i, x_j). \quad (1)$$

V is the set of all of the pixels and N_i is the set of the pixels in the domain of pixel i in Equation (1). The i and j variables represent two pixels. x_i is the category label of pixel i. The energy function was determined by the unary and binary potential energy. The unary potential energy function $\psi_U(x_i)$ was used to measure the probability that a pixel belonged to the category label x_i when the color of pixel i was y_i. The binary potential energy function $\psi_p(x_i,x_j)$ describes the relationship between two pixels. It was suggested that similar pixels be assigned the same label while pixels with significant difference be assigned to different labels. The calculation formula was:

$$\psi_p(x_i, x_j) = \frac{U\left(x_i, y_j\right) \sum_{m=1}^{M} \omega^m K_G^m}{K\left(f_i, f_j\right)}(f_i, f_j). \quad (2)$$

U in Equation (2) is the probability function that was calculating the probability that pixel i and pixel j belonged to the same probability. If $x_i \neq y_j$, then $U(x_i,y_j) = 1$, otherwise 0. M denotes the number of kernel functions. K_G^m is the mth Gaussian kernel function and the expression of $K_G^m(f_i, f_j)$ is:

$$K_G^m(f_i, f_j) = \exp\left(-\frac{1}{2}(f_i, f_j)^T \Lambda^{(m)}(f_i - f_j)\right). \quad (3)$$

The f_i and f_j in Equation (3) represent the feature vectors of pixel i and pixel j. Each Gaussian kernel K_G^m is characterized by a symmetric positive precision matrix $\Lambda^{(m)}$. For the multiclassification problem of image segmentation, $K(f_i,f_j)$ in Equation (2) is typically a dual kernel potential. The formula is:

$$K(f_i, f_j) = w_1 \exp\left(-\frac{\left|p_i - p_j\right|^2}{2\theta_\alpha^2} - \frac{\left|I_i - I_j\right|^2}{2\theta_\beta^2}\right) + w_2 \exp\left(-\frac{\left|p_i - p_j\right|^2}{2\theta_\gamma^2}\right). \quad (4)$$

I_i and I_j in Equation (4) represent the color vectors on pixels p_i and p_j, w_1 and w_2 are the weights of the two kernel functions. $k(f_i,f_j)$ was obtained by adding the appearance kernel and the smoothing kernel. The appearance kernel assumed that neighboring pixels with the same color belong to the same class and the smoothing kernel may have eliminated some isolated small regions.

The function of Equation (4) was to determine whether the pixels belong to the same class by their color and distance similarity. If the energy function value was relatively small, the pixels were thought to belong to the same class; if the energy function value was relatively large, the pixels were thought not to belong to the same class. In this way, the location and color information between each pixel in the image could be used in order to refine the extraction results of the residential areas with hatched lines.

2.5. Evaluation Metrics

The segmentation results were evaluated by five metrics: Dice similarity coefficient, Intersection over Union (IoU), recall, precision and accuracy. The Dice similarity coefficient is a function of the aggregation similarity measure, which was used to calculate the similarity between the algorithm segmentation result and the manual labeling result. IoU represents the degree of intersection or overlap between the algorithm results and the manual labeling results, which is the ratio of the intersection to the union. The closer the ratio is to 1, the higher the correlation. Recall refers to the ratio of the number of correctly predicted positive samples to the total number of true positive samples. Precision refers to the ratio of the number of correctly predicted positive samples to the total number of samples that were predicted to be positive. Accuracy refers to the ratio of all correctly predicted positive and negative samples to all samples. The calculation equations for these scores are:

$$Dice = \frac{2|X \cap Y|}{|X|+|Y|}, \tag{5}$$

$$IoU = \frac{X \cap Y}{X \cup Y}, \tag{6}$$

$$Recall = \frac{TP}{TP + FN}, \tag{7}$$

$$Precision = \frac{TP}{TP + FP}, \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

In Equations (5) and (6), X indicates the total area of the residential areas that were identified by the algorithm and Y indicates the total area of the residential areas that were identified by the manual marking result. In Equations (7)–(9), TP indicates the number of correctly extracted pixels that were sourced from residential areas, TN indicates the number of pixels correctly identified as background, FP indicates the number of incorrectly extracted pixels that were sourced from residential areas and FN indicates the missed pixels from residential areas.

3. Experiments, Result Analysis and Applications

The hardware configuration of the computer for this experiment included an Intel Core i7-9750H CPU, 16 GB RAM and an NVIDIA GeForce GTX 1660Ti graphics card. For the software, Jupyter Notebook 6.3.0 was used for program development in Python 3.7.4, Snipaste 2.2.4 was used for capturing the sample images. LabelMe 4.6.0 was used for labeling the samples, Augmentor 0.2.9 was used for sample enhancement and Pillow 8.2.0 was used for the cutting and integration of the map sheet-level raster images. TensorFlow 2.3.0 was used for network building and Pydensecrf 1.0rc2, OpenCV4.5.1 and Matplotlib 3.2.2 were used for CRF building.

3.1. Experimental Data Description

The raster map data that were studied in this paper were obtained from the National Museum of Scotland's website (http://maps.nls.uk/geo, accessed on 25 September 2021). The website integrates several types of map data, including Google Maps and raster maps, and includes high-resolution real surface topography maps and their corresponding raster maps. In this study, 32 sheets of historical raster maps from the period from 1894–1950 that recorded five regions of England were selected: namely those from Anglesey, Bedfordshire, Cambridgeshire, Cardiganshire, and London. One of these maps is shown in Figure 5. Each had a resolution of approximately 9000 × 6800 pixels. These maps cover both rural and urban areas with a large number of residential areas, woodlands, streets, rivers and annotations.

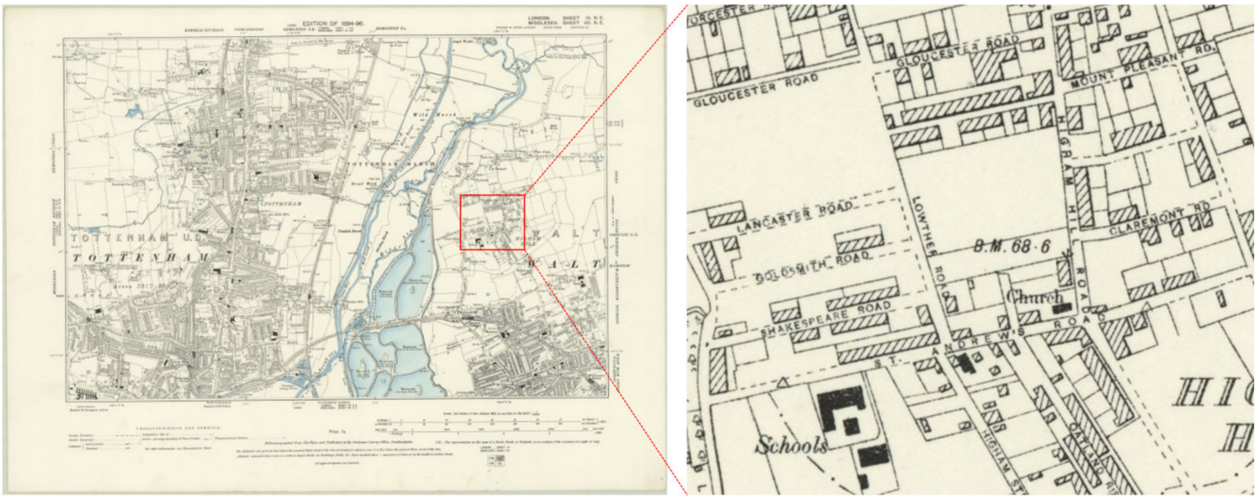


Figure 5. Raster map of northeast London published from 1894 to 1896. The map sheet size is 31 × 46 cm (ca. 12 × 18 inches).

Initially, we manually produced 1000 residential area sample images with a size of 256 × 256 pixels at different scales from these 32 sheets of raster maps, this resulted in 4000 sample images being obtained after the sample enhancement process.

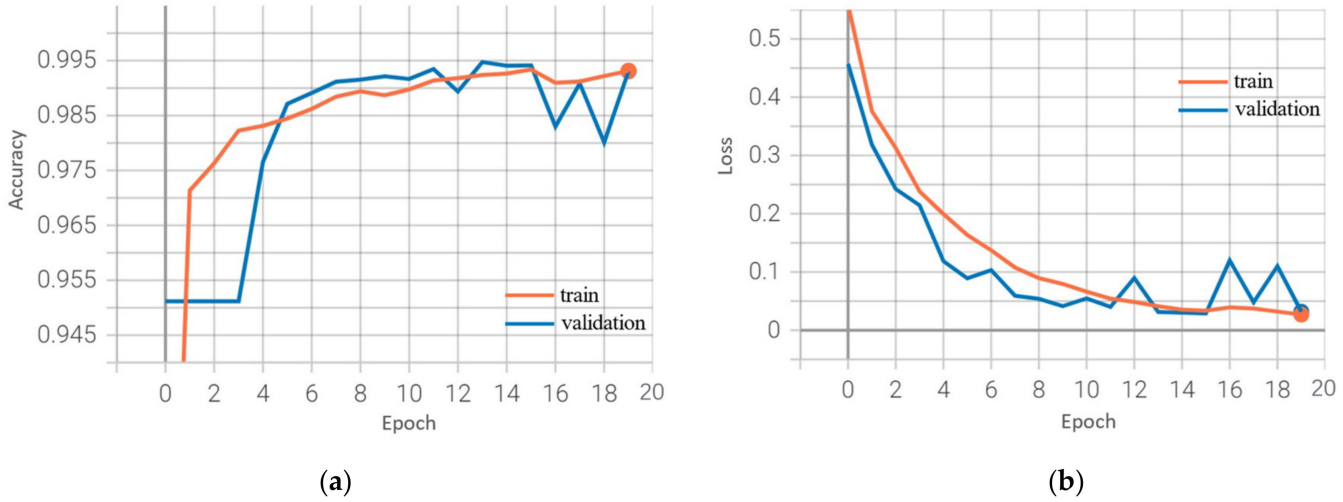
3.2. Model Training and Parameter Setting

In the model training process, 70% of the sample images were used as training samples and 20% as validation samples. The training samples were used to train the MSU-Net model and the validation samples were used to determine whether the trained model met the requirements. The AEHRA put the assigned 2800 hatched residential area training samples into the MSU-Net network for training and tested the image segmentation ability of the model after each training round by using the assigned 800 hatched residential area validation samples. From the accuracy change curve of the validation set in Figure 6a, it can be seen that the model had been in an underfitting convergence state until the 11th training round and that the accuracy of the model increased with each training round. After the 15th round of training, the accuracy of the model vacillated with the continuation of the rounds, which indicates that the model had been overfitted. From the loss value change curve of the validation set that is shown in Figure 6b, we can see that the model was also in a converged state before the 11th training round and that the loss value of the model decreased with each training round. After the 15th training round, the loss value of the model vacillated as the number of rounds increased, which indicates that the model appeared to be overfitting. Because both accuracy and loss value appeared to be overfitting after the 15th round, the number of iterations of the model was finally set to 20 in order to have an iterative training buffer.

The initial learning rate of the model was set to 1×10^{-4} in order to prevent the model from falling into a local minimum during the gradient descent and failing to find the optimal solution. At the same time, this reduced the fitting speed of the model so as to prevent the model from overfitting prematurely. We chose the Adam algorithm as the optimization algorithm and set the final experimental parameters, as shown in Table 1, which took into consideration the model's computational efficiency, result accuracy and hardware.

To fully understand the generalization capabilities of the models, the study used K-fold cross-validation in order to validate the model. The value of K was set to 5 and the 3600 samples were randomly divided (this pool included both training samples and validation samples) into 5 groups. The groups were taken, one at a time, to be used as the test data set and the remaining 4 groups were used as the training set. This process was repeated 5 times and the mean of the 5 results was taken as the final error estimation result. According to Table 2, it can be calculated that the mean value of Dice was 95.39%, IoU was

92.21%, recall was 90.36%, precision was 93.67% and accuracy was 99.27%. These results indicate that MSU-Net had a good generalization ability.



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

Figure 6. Indicator change line chart. (**a**) Line chart of accuracy value changes of 20 epochs. (**b**) line chart of loss value changes of 20 epochs.

Table 1. Training parameters of model.

Parameter	Value
Training samples	2800
Validation samples	800
Batch size	8
Weight decay	5×10^{-4}
Learning rate	1×10^{-4}
Optimizer	Adam
Epoch	20

Table 2. 5-fold cross-validation of MSU-Net.

Indicator	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Dice%	97.47	94.46	96.09	94.72	96.92
IoU%	95.07	89.51	92.48	89.97	94.03
Recall%	95.08	84.06	89.99	88.70	93.97
Precision%	95.33	94.50	94.85	89.66	94.03
Accuracy%	99.34	98.93	99.19	99.49	99.41

3.3. Experimental Results and Comparative Analysis

Figure 7 shows the residential area extraction results of seven sample images in the test set, based on the AEHRA. It can be seen that the AEHRA fully and accurately extracted the hatched residential areas in the raster images and that, in addition to the common rectangular residential area in Area 2 and Area 6, other more complex polygonal residential areas (such as those in Area 1, Area 3, Area 4, Area 5, and Area 7) could also be easily identified.

To demonstrate the validity and capability of the AEHRA model qualitatively and quantitatively, we compared the AEHRA with the FCN-8s and classical U-Net networks by using the same training set, validation set and test set. The predictions were performed on 400 samples in the test set and, to observe the comparison results, four typical areas were selected for display, as shown in Figure 8. The comparison of the experimental results showed that the FCN-8s suffered from blurred boundaries when recognizing the residential areas with hatched lines and that some places had convexities and others had concavities (Area 1). In the FCN-8s, the cavity phenomenon occurred during the identification of the

inner regions of the residential areas (Area 2 and Area 4). The FCN-8s also identified the region where some of the smaller black lines intersect with white areas as residential areas with hatched lines, forming isolated regions (Area 2 and Area 3). However, the FCN-8s could not fully extract the details of the residential areas because it had difficulty in using the features of shallow network localization. The classical U-Net network was prone to omission or breakage when extracting smaller and finer parts of the residential areas with hatched lines (Area 1, Area 2, Area 4). In some smaller areas where the black lines intersect with white areas, the misidentification phenomenon occurred (Area 3). The classical U-Net network also showed the same classification blur at the residential area boundaries. The MSU-Net used a feature map overlay in order to integrate the multiple scales of information into the final prediction map. Compared with the first two networks, the segmentation integrity and boundary extraction were significantly improved, but there were still a few cavities (Area 2) and isolated islands (Area 4). The MSU-Net-CRFs in the AEHRA added fully connected CRFs to MSU-Net, which resolved the phenomenon of cavities and isolated islands existing in MSU-Net-based recognition and improve segmentation accuracy.

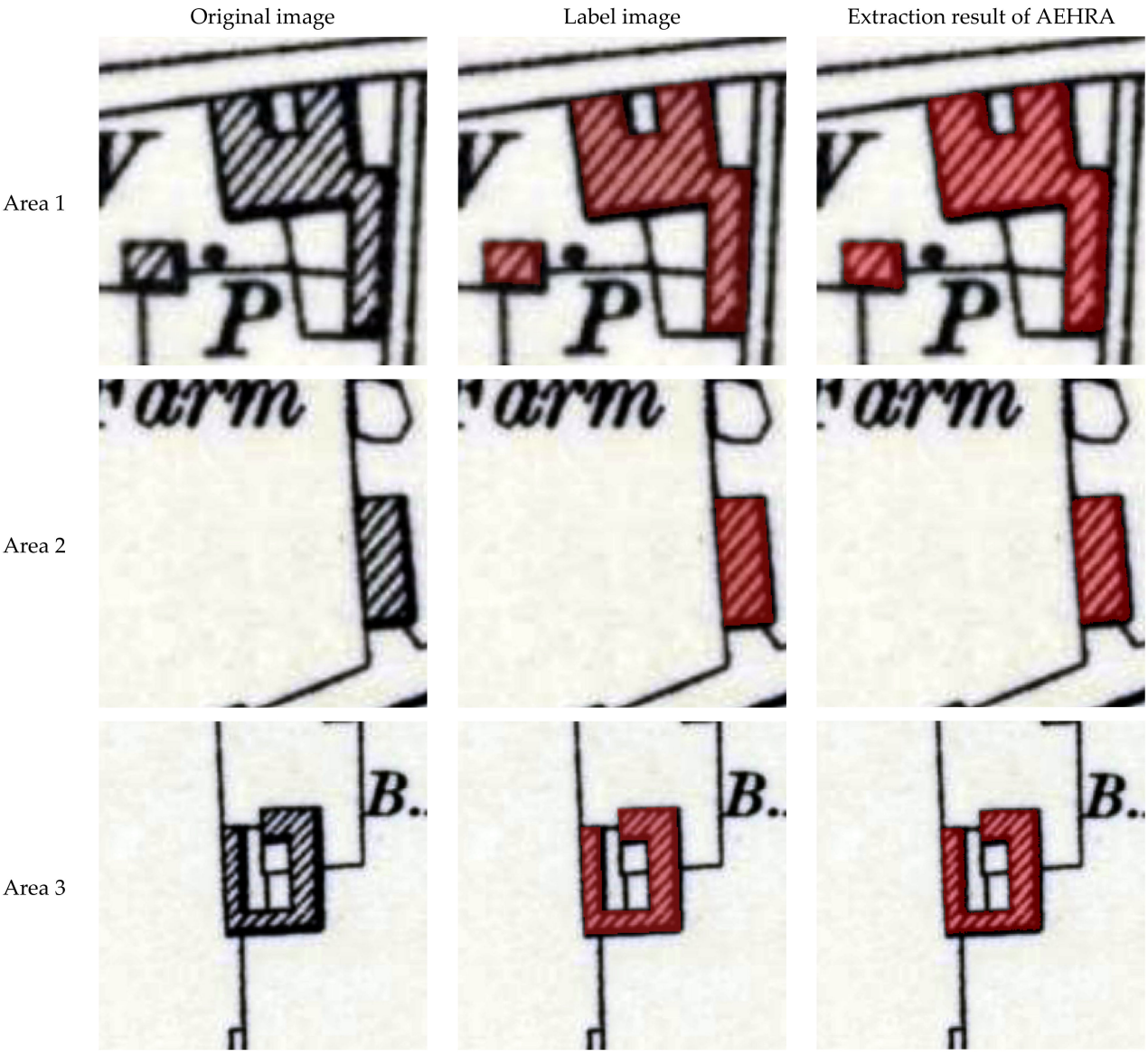


Figure 7. *Cont.*

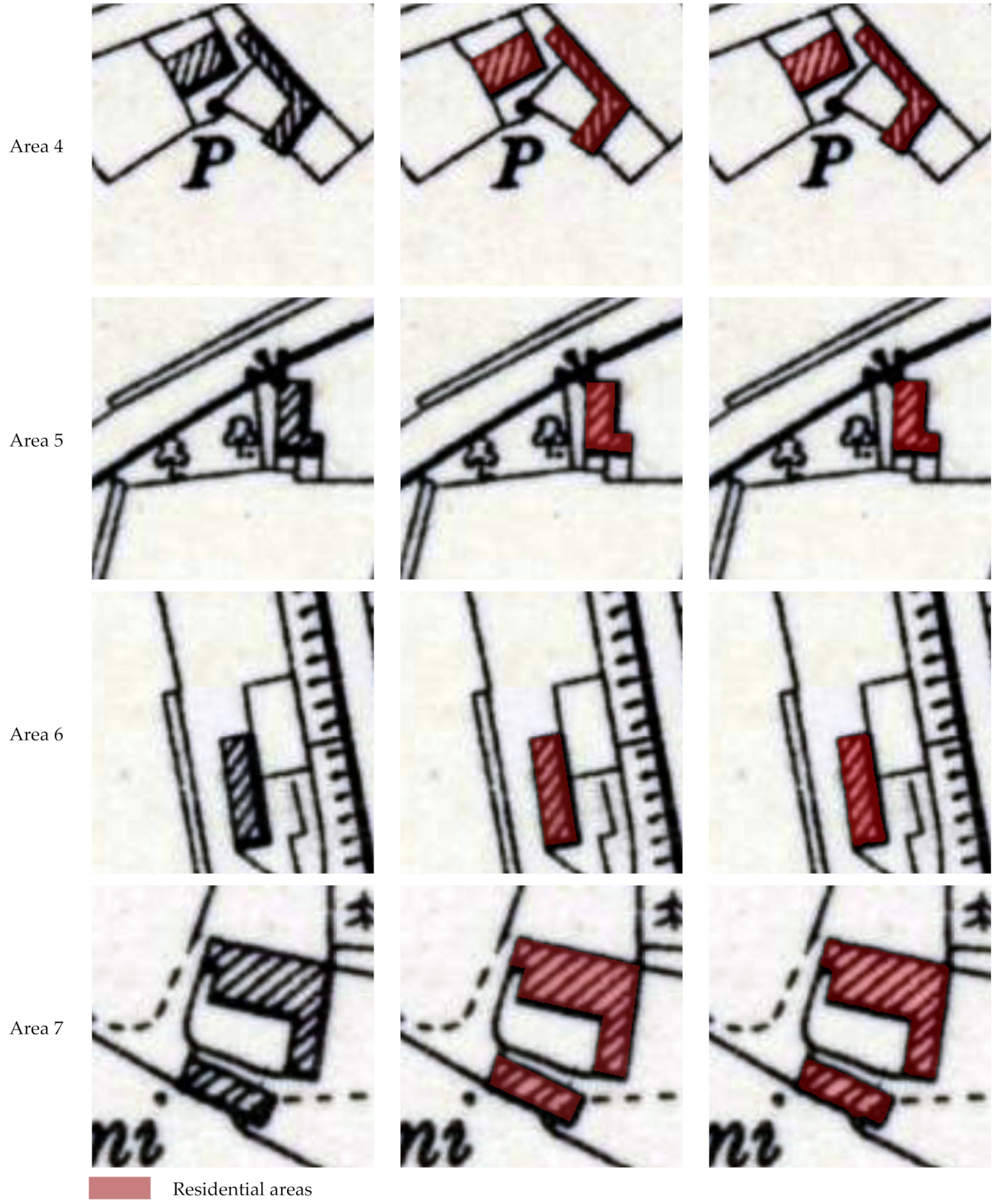


Figure 7. AEHRA-based extraction results of hatched residential areas.

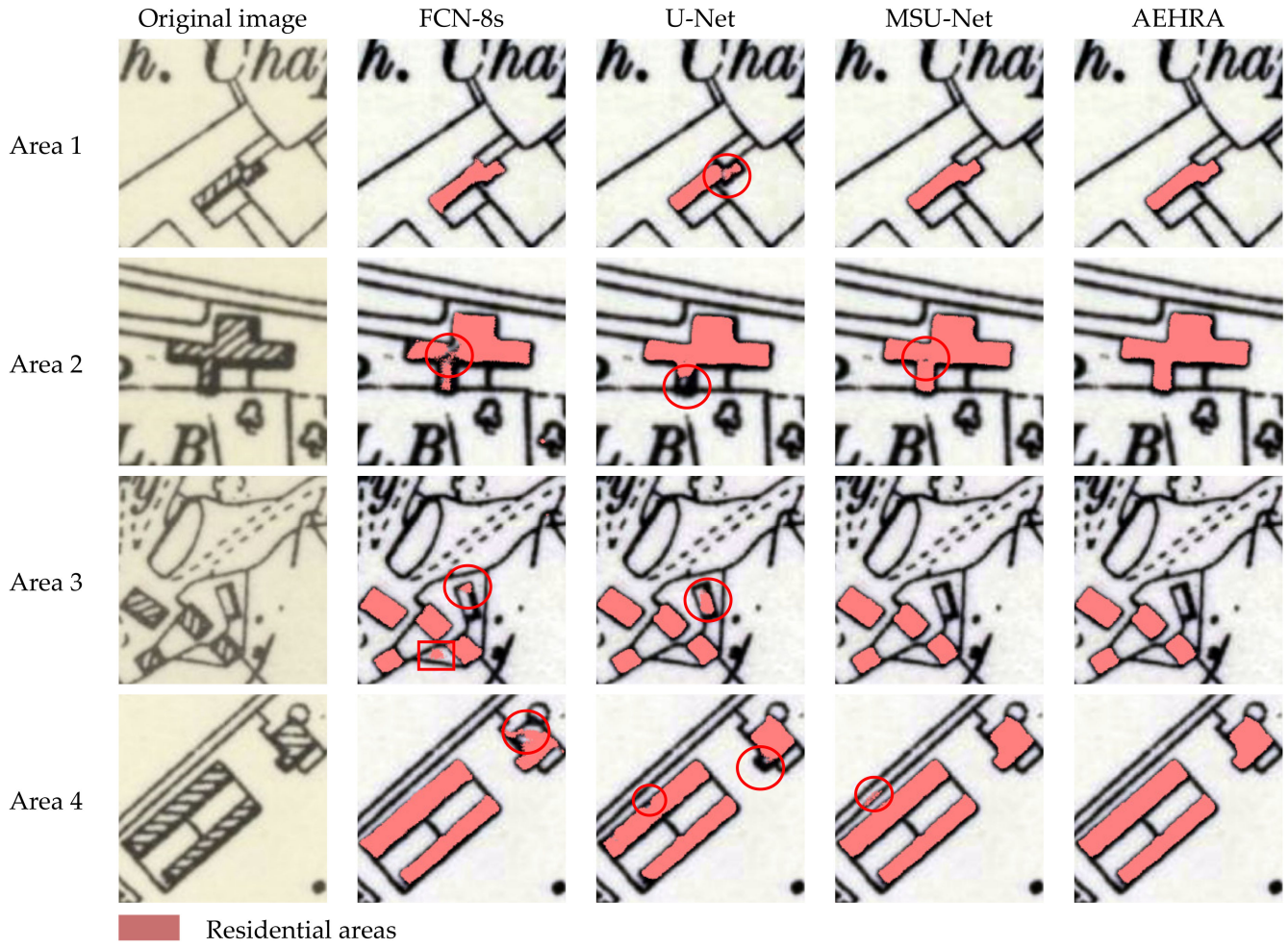


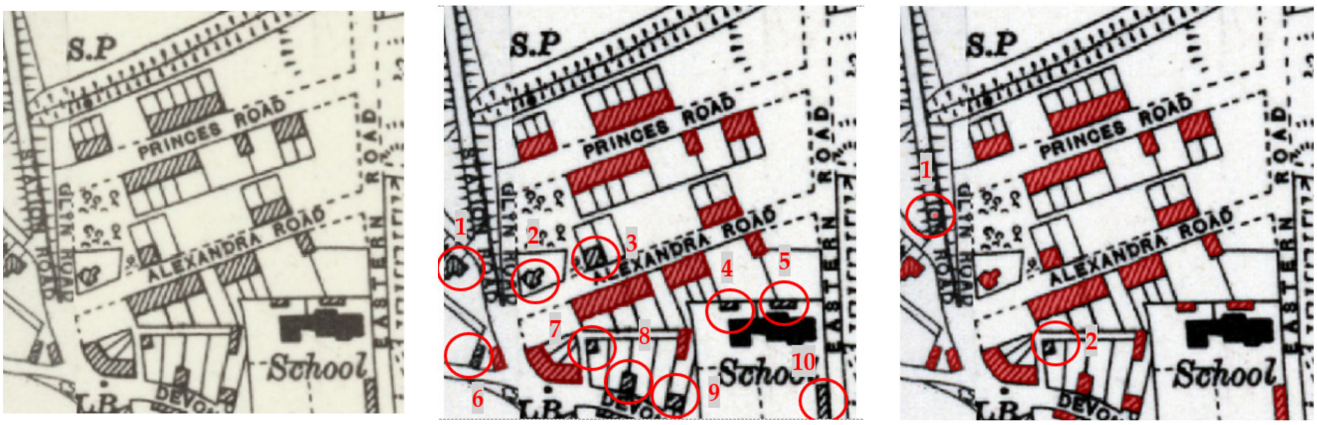
Figure 8. Segmentation results of hatched residential areas based on different algorithms.

The quantitative evaluation results are shown in Table 3. The AEHRA scored higher than the other three algorithms in terms of Dice, IoU, recall, precision and accuracy. Compared with the FCN-8s, these five metrics were improved by 1.47%, 2.72%, 1.07%, 4.56% and 0.26%, respectively. Compared with the U-Net, the five metrics were improved by 0.84%, 1.56%, 3.00%, 0.65% and 0.13%, respectively. Compared with the MSU-Net, the five metrics were improved by 0.03%, 0.03%, 0.02%, 0.07% and 0.02%, respectively.

Table 3. Comparison of segmentation algorithms.

Indicator	FCN-8s	U-Net	MSU-Net	AEHRA
Dice %	95.58	96.21	97.02	97.05
IoU %	91.54	92.70	94.23	94.26
Recall %	93.85	91.92	94.90	94.92
Precision %	88.96	92.87	93.45	93.52
Accuracy %	99.26	99.39	99.50	99.52

The AEHRA was also compared with the Gabor filter-based method [7]. As shown in Figure 9b, the identification results of the Gabor filter-based algorithm missed many of the hatched residential areas (circles 1–10). In contrast, as shown in Figure 9c, the identification results of the AEHRA only missed one hatched residential area (circle 2) and a smaller isolated island (circle 1) appeared at a similar hatch location in the figure.



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

Figure 9. Segmentation results of the Gabor filter-based algorithm and the AEHRA-based method. (**a**) Original raster image; (**b**) results of Gabor filter-based method; (**c**) results of AEHRA-based method.

3.4. Identifying Residential Areas from a Sheet of Raster Maps

Many historians understand the pattern of human economic activities through the number and density of the residential areas that are shown on historical raster maps. Insurance companies may need to know how the land has historically been used. In this regard, it is particularly important to be able to extract the hatched residential areas from the map sheet-level raster maps. To study the ability of the AEHRA method to identify map sheet-level raster images, we selected the historical raster map of Northeast London from 1894 to 1896 for the experiment. The longitude of the central point of the historical map that was used is 0°5′0″ E and the latitude is 51°32′15″ N. Due to the limitations of the computer's performance, the raster map with a resolution of 8924 × 6521 pixels was cut into 450 small raster images with a resolution of 356 × 362 pixels. The predicted results of the AEHRA were re-stitched into the original resolution in the corresponding order and overlaid with the original raster map image in order to obtain the image in Figure 10a. The raster map contains a total of 1006 hatched residential areas and the AEHRA recognized 1012 elements, of which 14 non-hatched residential areas were incorrectly predicted as hatched residential area elements and eight residential areas were missed. In general, the AEHRA was highly accurate in identifying the hatched residential areas, but there were several cases of incorrect identification. Figure 10b shows an enlarged view of the corresponding Area 1 in Figure 10a, while Figure 10c shows one of the cut images with a size of 356 × 362 pixels in the area of Figure 10b. Figure 10b shows that if the hatched residential area is at the cut edge of the image and the part that was cut out is also the edge part of the hatched residential area, then the phenomenon of incomplete object recognition will occur.

The extraction accuracy for the hatched residential areas was inversely proportional to the number of segmented tile images. As far as the experimental environment allowed, the fewer the number of segmented tile images, the higher the extraction accuracy. The experimental area in Figure 11a is Area 2 in Figures 10a and 11b is the extraction result corresponding to the red-framed area in Figure 11a after dividing the experimental area into 16 tiles. Figure 11c is the extraction result corresponding to the red-framed area in Figure 11a after dividing the experimental area into 100 tiles. In Figure 11b, it can be seen that there is only one incomplete object due to the identified object being at the tile image edge, while in Figure 11c multiple incomplete objects are identified because there are more tile images.

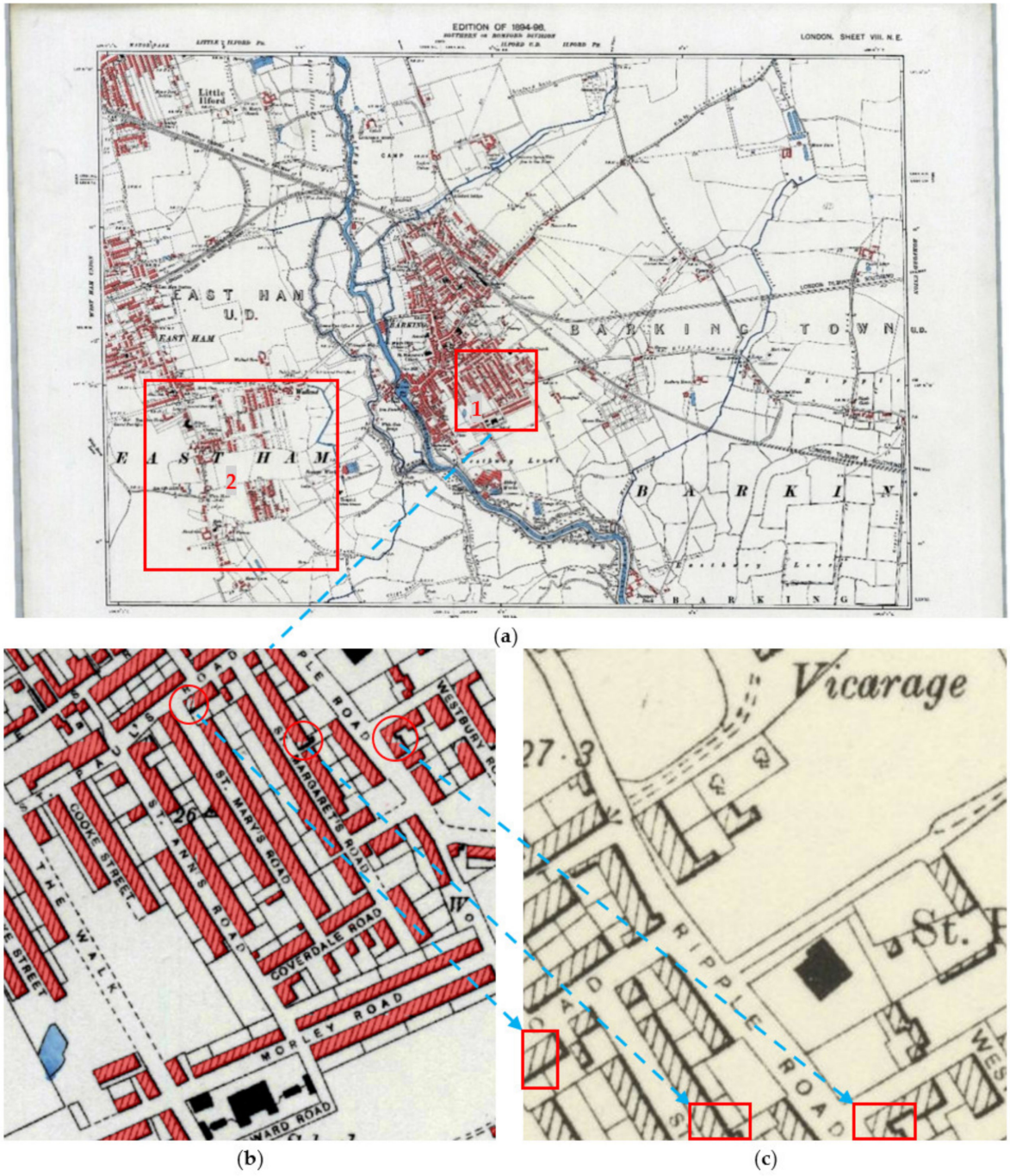


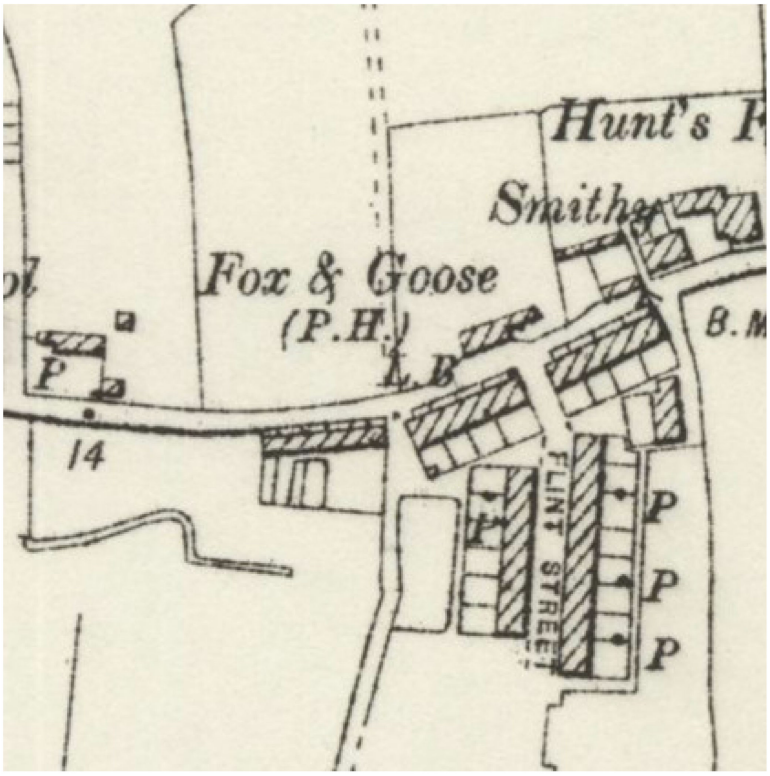
Figure 10. Extraction results of hatched residential areas from a raster map. (**a**) 1894–1896 historical raster map of Northeast London; (**b**) enlarged view of area 1; (**c**) small cut image with a size of 356 × 362 pixels.

3.5. Land Change Monitoring Application

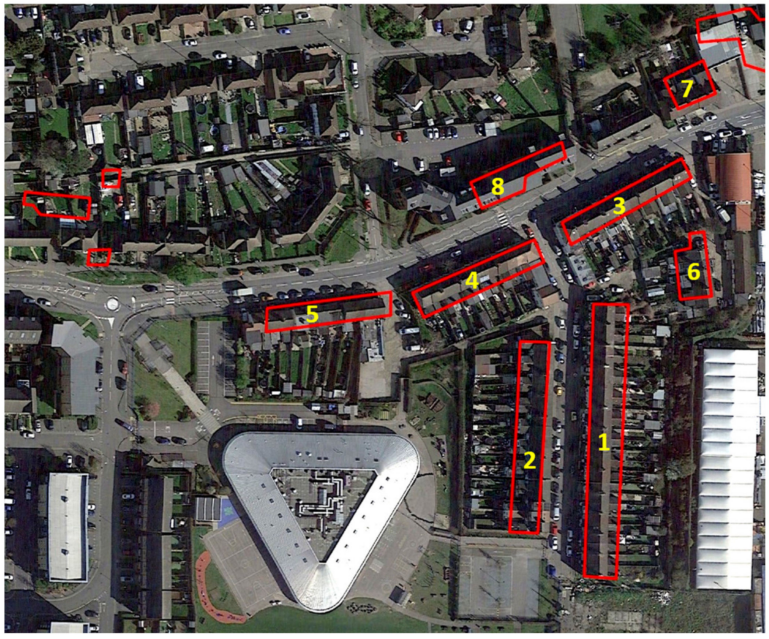
Land change monitoring was applied in order to test the AEHRA. As shown in Figure 12a, the 1898 raster map of the junction of London Road and Hilltop Road in Grays, UK, was selected for the study. The longitude and latitude of the map's center are 0°16′57.31″ E and 51°28′31.34″ N, respectively. The base map that is shown in Figure 12b represents the satellite remote sensing image of the area in March 2021 from Google Maps. The AEHRA method's extraction of the hatched residential areas from the raster map is shown in Figure 12a. The red polygons in Figure 12b are the result of vectorizing the raster hatched residential areas and simplifying the vector data; that is, reducing the number of polygon nodes. Interestingly, polygons 1–8 have not changed since 1898. The other

extracted residential areas can no longer be found on the existing map and it can be seen that most of the formerly vacant lands have now become residential areas.

Figure 11. Comparison of extraction results with different numbers of segmentation tiles. (**a**) Experimental area; (**b**) identification result when the map image was cut into 16 tiles; (**c**) identification result when the map image was cut into 100 tiles.



(**a**)



(**b**)

Figure 12. Identified hatched residential areas used for land change monitoring. (**a**) Historic raster map in 1898; (**b**) vector polygons of the extracted hatched residential areas overlap the satellite image from Google Maps.

4. Conclusions

The AEHRA method that has been proposed in this paper can be used to extract the residential areas that are denoted by hatched lines in historical raster maps. The three main contributions of this paper are as follows:

- We developed a novel deep learning model, called MSU-Net, which combines multi-scale features, outputs a feature image at each scale and fuses the feature images. This solved the problem that the up-sampling of a single chain of classical U-Net network cannot fully transfer the scale information;
- We combined the post-processing method of fully connected CRFs, used the probability distribution map output by the MSU-Net model as a unary potential and used the original map as a binary potential. The prediction results can be optimized by a calculation of the similarity of two pixels nodes in color and position, which can eliminate the isolated island and cavity phenomena;
- The research results of this paper can help historians to understand the relationship between human activities and the physical geographical environment, support land change monitoring and benefit the vectorization of raster maps. In short, it has significant application value.

Admittedly, there are some shortcomings to the AEHRA. Due to the limitations of the image semantic segmentation algorithm in deep learning, there are many nodes in the extracted contour of the residential areas, there is no obvious straight edge of the residential areas and the corners of the residential areas are curved rather than right angles. In addition, some incomplete objects may be generated at the edge of the dividing image. In the future, we will carry out optimization work for the extracted residential areas.

Author Contributions: Conceptualization, Jianhua Wu and Jiaqi Xiong; methodology, Jianhua Wu; validation, Jiaqi Xiong; formal analysis, Jiaqi Xiong and Yu Zhao; investigation, Xiang Hu; resources, Jianhua Wu; data curation, Jiaqi Xiong; writing—original draft preparation, Jianhua Wu; writing—review and editing, Jianhua Wu, Jiaqi Xiong, Yu Zhao and Xiang Hu. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Grant Nos. 41201409, 41561084).

Institutional Review Board Statement: We choose to exclude this statement because our study did not involve humans or animals.

Informed Consent Statement: We choose to exclude this statement because our study did not involve humans.

Data Availability Statement: We choose to exclude this statement because study did not report any data.

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

References

1. Zhigang, S. *Automatic Recognition of Hatched Residential Areas in Topographic Maps*; Jiangxi Normal University: Nanchang, China, 2017.
2. Shi-Ying, H.; Yuan-Hua, Z. Kohonen Clustering Networks with Fuzzy Selective Multiresolution for Intensity Image Segmentation. *Acta Electron. Sin.* **1999**, *27*, 34–37.
3. Xue, L.; Jiana, H. Algorithems for Recognition of Urban Roads in Large scaled Topographical Maps Based on Integration of Segmentation and Recognition. *Signal. Process.* **1998**, *14*, 80–83.
4. Zhongliang, F. *Ditu Saomiao Yingxiang Zidong Shibie Jishu*; The Mapping Publishing Group: Beijing, China, 2003.
5. Huali, Z.; Xianzhong, Z.; Jianyu, W. Research and Implementation of Automatic Color Segmentation Algorithm for Scanned Color Maps. *J. Comput. Aided Des. Comput. Graph.* **2003**, *15*, 29–33.
6. Yun, Y.; Changqing, Z.; Qun, S. Extraction and vectorization of street blocks on scanned topographic maps. *Sci. Surv. Mapp.* **2007**, *32*, 88–90.
7. Wu, J.; Wei, P.; Yuan, X.; Shu, Z.; Chiang, Y.; Fu, Z.; Deng, M. A New Gabor Filter-Based Method for Automatic Recognition of Hatched Residential Areas. *IEEE Access* **2019**, *7*, 40649–40662. [CrossRef]

8. Lecun, Y.; Bottou, L. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
9. Krizhevsky, A.; Sutskever, I.; Hinton, G. *ImageNet Classification with Deep Convolutional Neural Networks*; ACM: New York, NY, USA, 2017; Volume 60, pp. 84–90.
10. Kim, Y. Convolutional Neural Networks for Sentence Classification. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2015.
11. Zhang, W.; Tang, P.; Zhao, L. Fast and accurate land-cover classification on medium-resolution remote-sensing images using segmentation models. *Int. J. Remote Sens.* **2021**, *42*, 3277–3301. [CrossRef]
12. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Manhattan, NY, USA, 16–20 June 2019; pp. 558–567.
13. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; Lecun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arxiv* **2013**, arXiv:1312.6229.
14. Xi, W.; Ming, Y.; Hong, E.R. Remote sensing image semantic segmentation combining UNET and FPN. *Chin. J. Liq. Cryst. Disp.* **2021**, *36*, 475–483.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
16. Manzo, M.; Pellino, S. Fighting together against the pandemic: Learning multiple models on tomography images for COVID-19 diagnosis. *AI* **2021**, *2*, 16. [CrossRef]
17. Luo, H.; He, B.; Guo, R.; Wang, W.; Kuai, X.; Xia, B.; Wan, Y.; Ma, D.; Xie, L. Urban Building Extraction and Modeling Using GF-7 DLC and MUX Images. *Remote Sens.* **2021**, *13*, 3414. [CrossRef]
18. Jiao, C.; Heitzler, M.; Hurni, L. A survey of road feature extraction methods from raster maps. *Trans. GIS* **2021**, 1–30. [CrossRef]
19. Shunping, J.; Shiqing, W. Buidling extraction via convolutional neural networks from an open remote sensing building dataset. *Acta Geod. Cartogr. Sin.* **2019**, *48*, 448–459.
20. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 640–651.
21. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
22. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
23. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. *J. Mach. Learn. Res.* **2011**, *15*, 315–323.
24. 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]
25. Zhang, C.S.; Jia, X.; Wu, R.R.; Cui, W.H.; Shi, S.; Guo, B.X. Object oriented semi-automatic extraction of typical natural areas from high-resolution remote sensing image. *J. Geo-Inf. Sci.* **2021**, *23*, 1050–1062.
26. Wang, Z.H.; Zhong, Y.F.; He, W.W.; Qu, N.Y.; Xu, L.Z.; Zhang, W.Z.; Liu, Z.X. Island shoreline segmentation in remote sensing image based on improved Deeplab network. *J. Image Graph.* **2020**, *25*, 768–778.
27. Huang, X.X.; Li, H.G. Water body extraction of high resolution remote sensing image based on improved U-Net network. *Int. J. Geo-Inf. Sci.* **2020**, *22*, 2010–2022.