

Article

Single-Tree Detection in High-Resolution Remote-Sensing Images Based on a Cascade Neural Network

Dong Tianyang, Zhang Jian, Gao Sibin, Shen Ying and Fan Jing *

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China; dty@zjut.edu.cn (D.T.); workmail_jason@163.com (Z.J.); zjutgsb@163.com (G.S.); shenyings@zjut.edu.cn (S.Y.)

* Correspondence: fanjing@zjut.edu.cn; Tel.: +86-571-85290116

Received: 17 July 2018; Accepted: 31 August 2018; Published: 6 September 2018



Abstract: Traditional single-tree detection methods usually need to set different thresholds and parameters manually according to different forest conditions. As a solution to the complicated detection process for non-professionals, this paper presents a single-tree detection method for high-resolution remote-sensing images based on a cascade neural network. In this method, we firstly calibrated the tree and non-tree samples in high-resolution remote-sensing images to train a classifier with the backpropagation (BP) neural network. Then, we analyzed the differences in the first-order statistic features, such as energy, entropy, mean, skewness, and kurtosis of the tree and non-tree samples. Finally, we used these features to correct the BP neural network model and build a cascade neural network classifier to detect a single tree. To verify the validity and practicability of the proposed method, six forestlands including two areas of oil palm in Thailand, and four areas of small seedlings, red maples, or longan trees in China were selected as test areas. The results from different methods, such as the region-growing method, template-matching method, BP neural network, and proposed cascade-neural-network method were compared considering these test areas. The experimental results show that the single-tree detection method based on the cascade neural network exhibited the highest root mean square of the matching rate ($RMS_{R_{mat}} = 90\%$) and matching score ($RMS_M = 68$) in all the considered test areas.

Keywords: single-tree detection; high-resolution; remote-sensing images; backpropagation network; cascade neural network

1. Introduction

Reliable information concerning a forest is required to perform extensive forest management, as well as for planning purposes to maintain sustainable forestry. With the increasing availability of high-spatial-resolution data and computational power, a growing amount of remote-sensing research on forestry focused on detecting and measuring individual trees as opposed to obtaining stand-level statistics. High-resolution satellite remote-sensing imagery is currently one of the most widely used types of data in forestry applications [1]. Today, many remote-sensing satellites can obtain sub-meter remote-sensing images; these satellites include Orbview5, WorldView, and QuickBird-2 of the United States, EROS-B and EROS-C of Israel, and Gaofen-2 of China. The color and contour features of trees, which cannot be observed in low-resolution remote-sensing images, can be observed in the high-resolution remote-sensing images.

Currently, there is widespread interest among many researchers regarding the detection of individual trees and gathering forest information from digital aerial photographs or high-resolution remote-sensing images, and several researchers proposed automatic or semi-automatic single-tree

detection methods. The conventional methods of tree detection can be mainly divided into two categories.

One method involves tree detection based on pixels. For example, the local-maximum method [2–4] extracts the maximum value of a local area as the center point of a tree. In addition, the algorithm combines region growing, watershed segmentation, and other methods to detect a single tree. Novotný and Hanuš et al. [5] proposed a method of local maxima with variable window sizes, and they used seed region growing methods to detect individual trees. Hirschmugl et al. [6] firstly compared the different methods of obtaining the coronal center, and then proposed a deformation algorithm to determine coronal centers. However, the methods based on the local maximum cannot make full use of the overall characteristics of the tree. Therefore, the selection of seed points in a complex background significantly affects the accuracy. The template-matching algorithm [7–9] uses the template of the tree to compare the square error of all the pixels with the same size of the image in sequence. However, the template-matching method is not suitable for areas in which the trees are located in a crowded manner and the canopies often overlap, because many trees cannot be detected.

The other category corresponds to object-based tree-detection methods, which gradually incorporate machine-learning algorithms. For example, Salim Malek et al. [10] extracted a set of candidate key points of a palm farm using the scale-invariant feature transform (SIFT) and analyzed these key points with a recent kernel-based classification method termed as extreme learning machine (ELM). However, SIFT selects the characteristics of several key points in the sample, and it is less competitive than the method based on the global features of red, green, and blue channels. In addition, Lin Yang et al. [11] trained a pixel-level classifier for each pixel in the aerial image based on a set of visual features, and introduced methods for model and data selection based on two-level clustering. However, these methods require artificially setting a large number of parameters for different scenes, which is extremely difficult without prior knowledge.

Overall, there are some problems with the existing single-tree detection methods. Firstly, these methods have a strong parameter dependency. They usually need experts to set various parameters in advance according to the forestlands. Secondly, the detection performance differs greatly across different forest types, and the generalization ability of these methods is weak. For example, the region-growing method can obtain the best detection results in the case of mixed and dense forests, but the detection results in the case of an isolated forest are much worse than those by other detection methods.

The detection of individual trees in high-resolution remote-sensing images is typically a target-recognition problem. Owing to the advantages of the cascade neural networks [12,13], such as a strong ability in nonlinear mapping, fast convergence, and good fault tolerance, these networks achieved great success in dealing with image-identification problems. To reduce the dependence of the detection method on prior knowledge and improve the generalization performance of the classification model for different scenes, this paper presents a single-tree detection method for high-resolution remote-sensing images based on a cascade neural network. Unlike the methods based on a pixel analysis, the proposed method based on a set of pixels can take the overall characteristics of trees into account. Firstly, this model calibrated many tree and non-tree samples in high-resolution remote-sensing images and used these samples to train the classic backpropagation (BP) neural network model [14–16]. The neural network at the first stage can perform the nonlinear characterization of tree features and provide a preliminary classifier. To further improve the accuracy of the classifier, we analyzed the statistical characteristics of trees and designed a BP neural network in the second stage. The second network input layer includes both the output of the first BP network and the statistical characteristics of the tree samples on the three RGB channels.

2. Materials

Google Earth is a virtual-earth software application that renders a simulacrum of the Earth based on satellite imagery. In addition, Google Earth images are easily available, and this can

have large implications for forest management and land applications. The remote-sensing images used in this study are WorldView pan-sharpened imagery. They involve red, green, and blue channels, with a resolution of 0.31 m. Because of concerns regarding transferability, we processed the satellite-derived images directly from Google Earth without radiometric (radiance and reflectance) or sun-glint corrections.

For conducting the comparative experiment to prove the effectiveness of the proposed method on single-tree detection, we chose six uncalibrated forest areas as the test areas. Because field measurement data for these test areas cannot be obtained, we considered the artificial visual interpretation of six volunteers to calibrate the trees as the ground truth.

The forestlands considered in this study are located in China and Thailand. Figure 1 shows the satellite imagery and corresponding reference data of all six test areas; here, the test area in every image was defined using a yellow line, as the trees outside the yellow line are difficult to be assessed by the human eye. The latitude and longitude coordinates of these six test areas are shown in Figure 1. The left column of Figure 1 is the RGB image, and the right column of Figure 1 is the corresponding reference data.

Test areas 1 and 2 are located in Thailand, and the main tree species in these two areas is oil palm, an important economic crop in Thailand. There are 801 and 179 reference trees in test areas 1 and 2, respectively. Test area 3 is located in Hangzhou, China. This area is relatively complicated. There are not only different trees, but also rivers, buildings, and lawns in this area. There are 312 reference trees in test area 3. Test area 4 is located in Shaoxing, China. This area mainly consists of some small seedlings, and the trees are common varieties. There are 338 reference trees in this area. Test area 5 is located in Hangzhou, China, and mainly consists of red maples. The photo was taken in autumn; thus, the leaves were red. There are 341 reference trees in test area 5. Test area 6 is located in Dongguan, China, and the main tree species of this area is longan. The forest density in this area is relatively high; thus, the tree crowns overlap each other. There are 521 reference trees in test area 6.

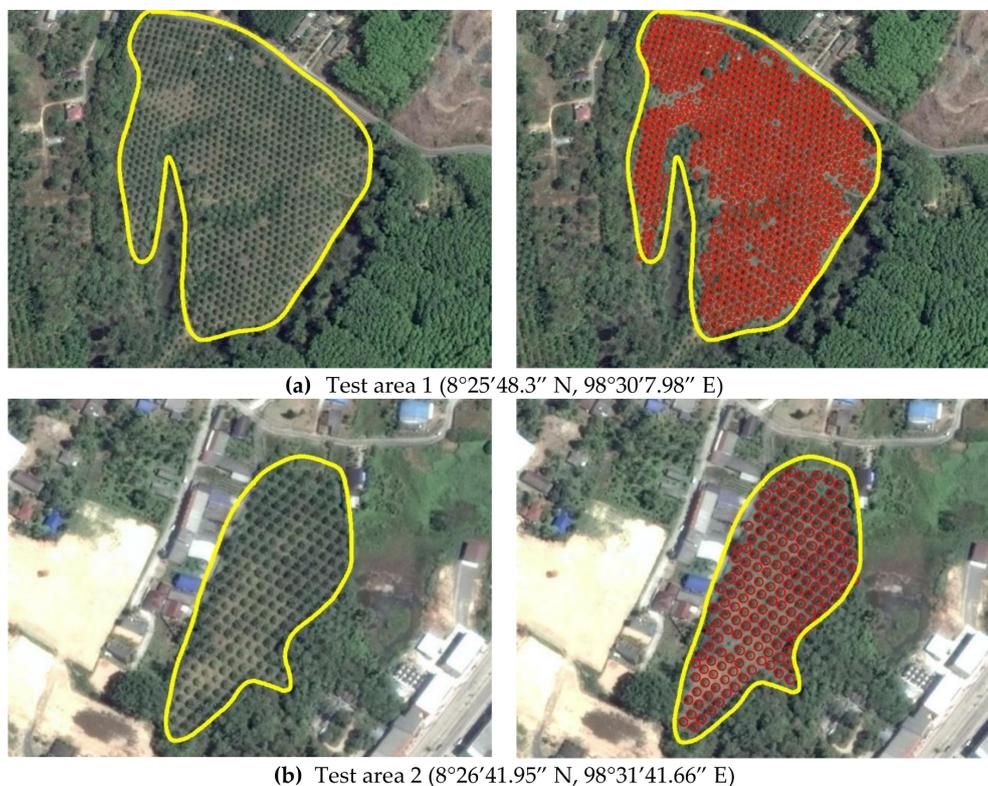


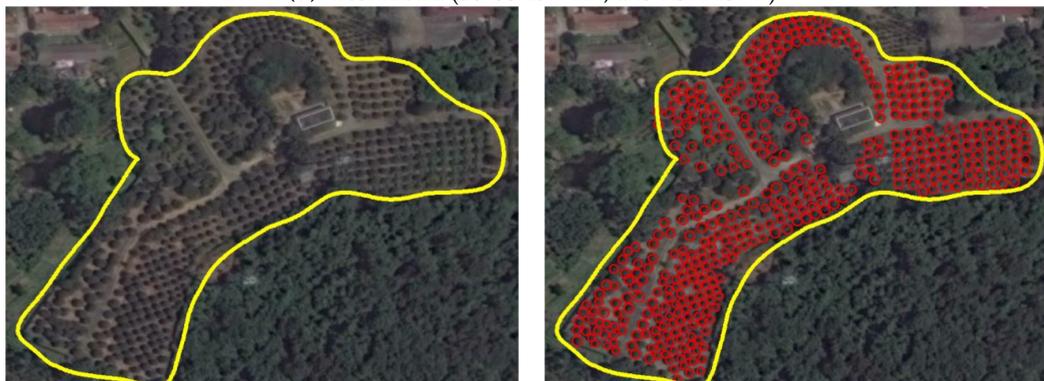
Figure 1. Cont.



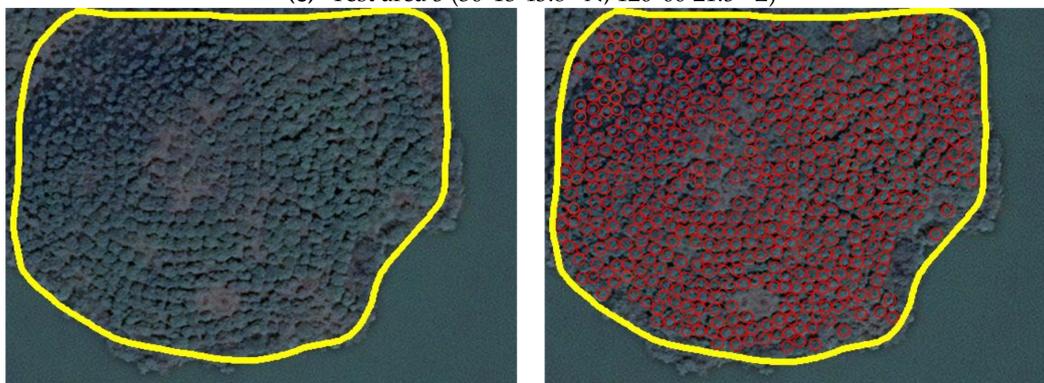
(c) Test area 3 ($30^{\circ}13'41.8''$ N, $120^{\circ}01'55.8''$ E)



(d) Test area 4 ($30^{\circ}06'43.7''$ N, $120^{\circ}20'27.8''$ E)



(e) Test area 5 ($30^{\circ}13'43.8''$ N, $120^{\circ}00'21.5''$ E)



(f) Test area 6 ($22^{\circ}50'25.3''$ N, $114^{\circ}12'15.8''$ E)

Figure 1. Six different test areas and the corresponding reference data.

3. Method

The flowchart of the single-tree detection method for high-resolution remote-sensing images based on a cascade neural network is shown in Figure 2. Firstly, we selected different types of forestlands from the high-resolution remote-sensing images and calibrated the representative tree and non-tree samples for different forest types. Secondly, we normalized the samples to the same size to ensure a uniform input layer size of the neural network, and we calculated the first-order statistical features of the samples, such as the energy, entropy, mean, skewness, and kurtosis. Finally, the neural-network model was trained with these samples and features, until the errors in the desired output and the actual output met the requirements. After neural-network learning, the trained neural-network model could be adopted as a classifier to detect single trees for different forests.

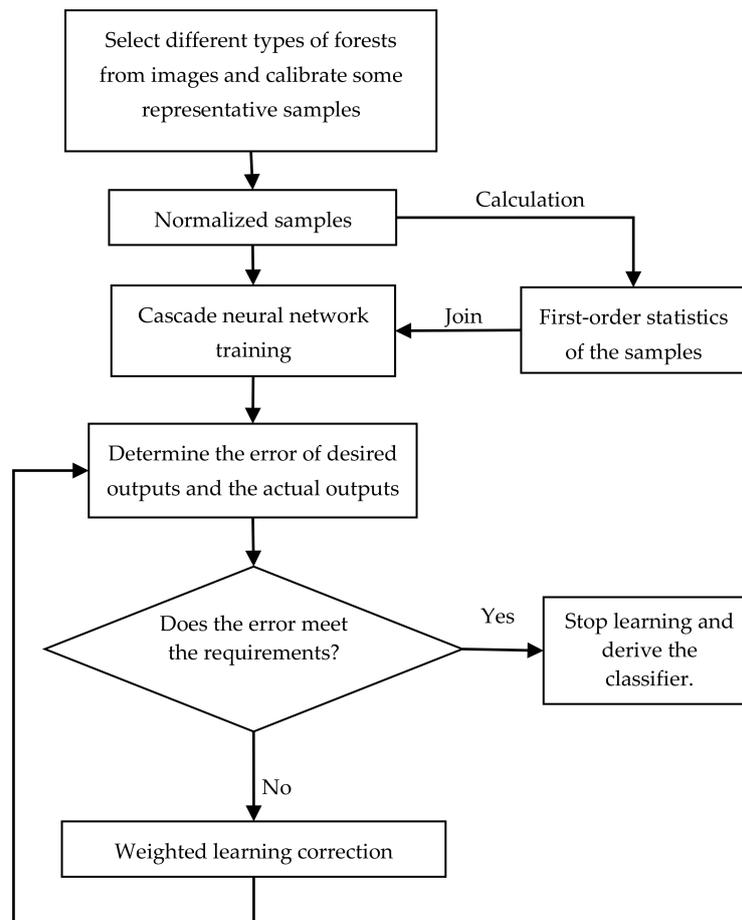


Figure 2. Flowchart for the single-tree detection method.

3.1. Sample Calibration

To obtain a classifier that can accurately distinguish between a tree and non-tree, the neural network must be trained using manually calibrated tree and non-tree samples. Because there are many types of trees in the remote-sensing images, such as isolated trees, sticky trees, larger trees, smaller trees, and trees under shadows, each tree type must be considered. Therefore, sample calibration requires a large number of positive (tree) and negative (non-tree) samples. In our study, a total of 849 positive samples and 848 negative samples were calibrated. An example of positive and negative sample calibrations is shown in Figure 3.

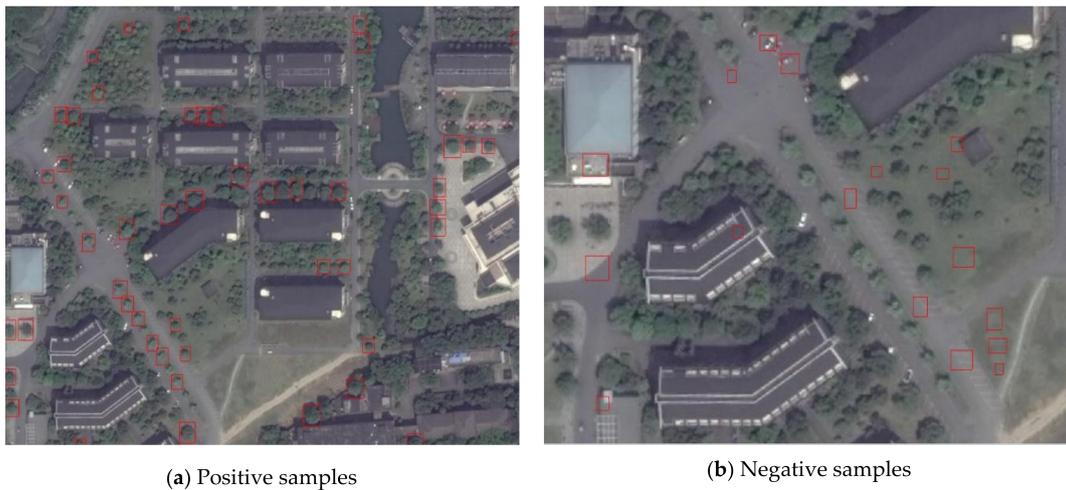


Figure 3. Calibration of samples.

Manual calibration sampling takes a large amount of time and effort. However, more samples can achieve better training results and stronger generalization ability of the neural-network model; thus, it was necessary to expand the limited number of manually calibrated samples. Therefore, we used the data-argument technology [17,18] to increase the number of samples. The calibration samples were manipulated using left and right mapping, with a left and right rotation of 15 degrees each; thus, the number of positive and negative samples was four times the number of original samples. Finally, we obtained 3396 positive and 3392 negative samples. An example of the sample extension is shown in Figure 4.

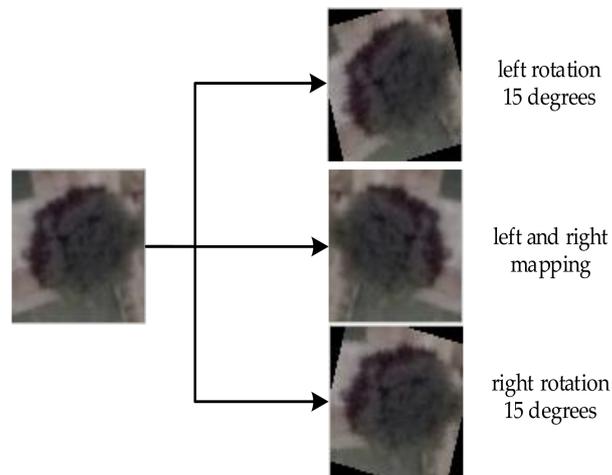


Figure 4. Sample-extension demonstration.

To unify the number of neurons in the input layer, all positive and negative samples were required to be resized to the same size. The specified size is usually the average size of the individual trees in the forestlands; for example, the value was 25×25 pixels for our experiment.

After normalizing the individual tree samples, we divided the samples into three separate sets for the neural-network training: 50% of the samples were the training set, 25% were the validation set, and the remaining 25% were the test set. The training set was used to train the model, the validation set was used to determine the final parameters of the control network, and the test set was used to evaluate the single-tree detection method.

3.2. Training Samples Using the BP Neural Network at the First Stage

The BP neural network is generally a three-layer network, involving the input layer, hidden layer, and output layer. In our method, the sigmoid function, $f(x) = 1/(1 + e^{-x})$, was chosen as the activation function of the neural network. The activation function maps the combination of neurons and bias non-linearly to enhance the expressiveness of the network. Firstly, the feed-forward transmission is adopted to reconstruct the network and update the parameters. The aim of feed-forward transmission is to achieve the representation of the original data of the input layer as much as possible. The error of the output's direct front layer can be estimated using a backpropagation algorithm in the process. Then, the error can be used to estimate the further layer and achieve the error estimation for other layers by sequentially performing backpropagation. The feedback error is used to update weights.

In the training of a BP neural network, it is necessary to firstly determine the number of neurons in each layer of the neural network, as shown in Figure 5. The sample image in this method is a single patch of 25×25 pixels of grayscale image, and, since the input layer needs a bias, the number of neurons in the input layer is 626. Since the identification of a tree is a binary classification problem, the number of neurons in the output layer is only one. If the output has a value of zero, a tree does not exist in the input image; otherwise, a tree is present in the input image. After determining the number of neurons for the input and output layers, the number of neurons in the hidden layer must be determined.

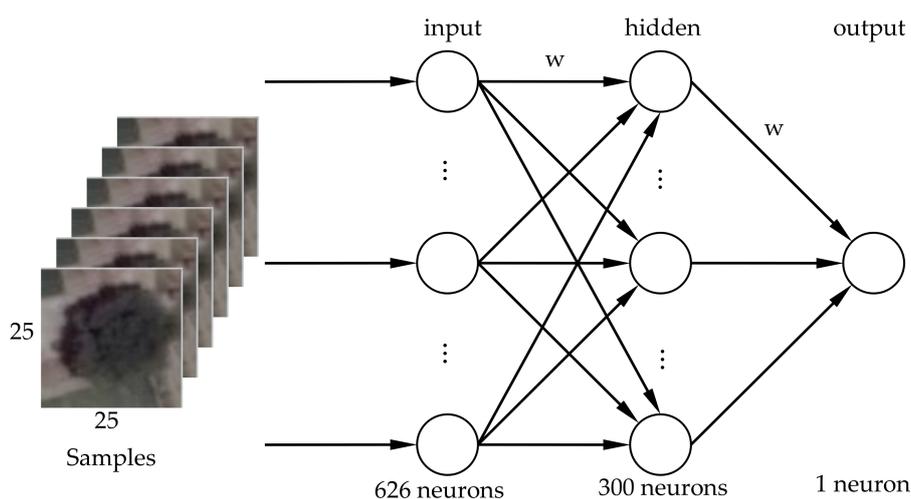


Figure 5. Structural design of a backpropagation (BP) neural network.

To determine the number of neurons in the network hidden layer, we performed an experiment concerning the different numbers of neurons. In the field of machine learning, and specifically, in statistical-classification problems, a confusion matrix [19], also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm. We split all samples into a training set, validation set, and test set. Accordingly, for the different network structures, we could obtain the confusion matrices of the training set, validation set, test set, and all sample datasets to evaluate the classification capabilities of the network. We selected different numbers of neurons for the hidden layer, such as 150, 300, and 450 neurons, which represent intermediate values between the number of neurons in the input and output layers.

3.3. Calculating First-Order Statistics to Train Samples at the Second Stage

In the first step of training, we used the grayscale image to quickly distinguish the trees from the background without much RGB information. To further improve the recognition rate and reduce the omission rate, the proposed approach combines the training results of a previous BP neural network

with the first-order statistics of individual trees as the input for another BP neural network. Since the grayscale image is synthesized by the three RGB channels, some information is lost. The first-order statistics from each color band can improve the recognition rate and reduce the omission rate. The features of first-order statistics, including energy, entropy, mean, skewness, and kurtosis of the trees in samples help determine whether the object is a tree. In this study, we adopted the following first-order statistics as features, where N is the largest grayscale, n is one of the grayscales, and $H(n)$ is a normalized histogram:

- Energy: $e = \sum_{n=0}^{N-1} H(n)^2$;
- Entropy: $s = - \sum_{n=0}^{N-1} H(n) \log H(n)$;
- Mean: $\mu = \frac{1}{N} \sum_{n=0}^N nH(n)$;
- Skewness: $\gamma_1 = \frac{1}{\sigma^3} \sum_{n=0}^{N-1} (n - \mu)^3 H(n)$;
- Kurtosis: $\gamma_2 = \frac{1}{\sigma^4} \sum_{n=0}^{N-1} (n - \mu)^4 H(n)$.

3.4. Training Samples Using a Cascade Neural Network

This constitutes a two-level cascade neural network to improve the results in single-tree detection for different scenarios.

The number of neurons in the input layer of the second neural network is 17. There are 15 neurons for the five features of energy, entropy, mean value, skewness, and kurtosis in the three RGB bands in a remote-sensing image, one is for the output value of the previous BP neural network, and another is for the bias. The output layer is also used for determining whether the input image involves a tree; thus, only one output neuron is required. The number of neurons in the hidden layer is half of the number of input neurons, specifically, eight neurons. The structure of the cascaded neural network is shown in Figure 6. The gray box in the middle dotted-line box indicates that the first BP network uses the gray information of the samples, and the RGB boxes corresponding to the first-order statistical features indicate that the three channels of RGB information were extracted. In the second level of the network, we mixed the grayscale and RGB information. The cascade-neural-network model is composed of two three-layer BP neural networks; thus, it is a 3–3-layer cascade-neural-network model. In this way, we can further improve the ability of the network to classify trees. In addition, the numbers of neurons and layers in the network are determined through grid search.

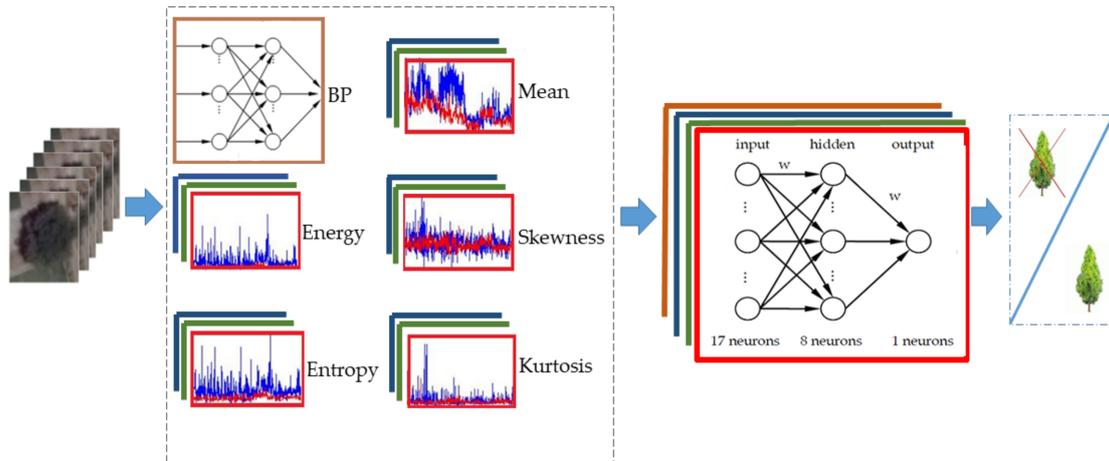


Figure 6. 3–3-layer cascade-neural-network model.

To compare the accuracy of detection results using network models with different depths, this paper presents another network structure, shown in Figure 7, in which the second level of the neural network involves the addition of a hidden layer of four neurons. In a similar manner to a 3–3-layer network model, this model is called a 3–4-layer cascade-neural-network model. The level of the 3–4-layer cascade-neural-network model in Figure 7 is deeper than that of a 3–3-layer cascade-neural-network model.

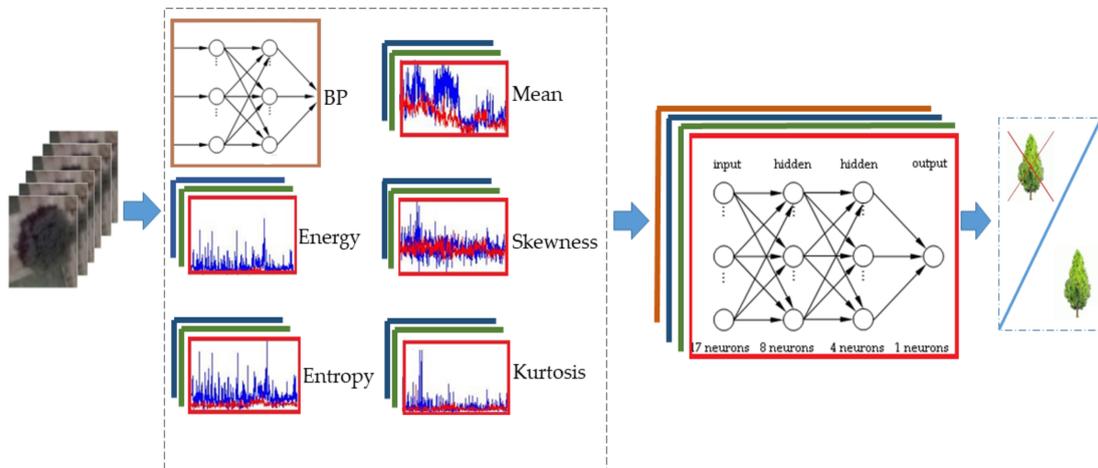


Figure 7. 3–4-layer cascade-neural-network model.

3.5. Sliding Window and Redundant Sample Removal

To verify the validity of the proposed method, we used the trained classifier to traverse each image through a sliding window to find single trees, where the sliding window size varied from 17×17 to 33×33 , according to the tree size in the study areas. If the image in the window was a tree, we marked the window with a label (saving the location and size) to represent tree detection. After traversing the entire image, several labels indicating potential trees are obtained; however, many of them represent the same tree; thus, non-maximal suppression (NMS) technology is used to remove the redundant labels. Firstly, all labels are sorted by their classification probability from high to low. Subsequently, we extract the top label, $label_{now}$, in the sequence as a detected tree, and remove all the labels in the sequence for which the overlap areas with $label_{now}$ are greater than a threshold. Next,

we repeatedly extract the top label, $label_{now}$, in the sequence and repeat the above process until the sequence is empty. An example of NMS is shown in Figure 8.

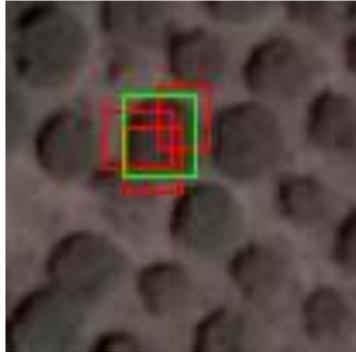


Figure 8. An example of non-maximal suppression (NMS). The rectangles represent a detected tree, and the green rectangle indicates the biggest classification probability.

3.6. Accuracy Evaluation Method

To verify the effectiveness of the proposed single-tree detection method, it is necessary to evaluate the accuracy of the detection results according to the reference data. When the spatial-position difference between the detected tree in the detection result and a ground-truth tree is within a certain range, we can say that the detected tree matches the ground-truth tree, that is, the detected tree is a correct result. A detailed evaluation of the accuracy involves three steps [20]:

- (1) Candidate tree selection: For a detection tree, the reference trees are added to the candidate set if the horizontal difference ΔD_{2D} is within a certain threshold range. We set $\Delta D_{2D} < 3m$ in test areas 1, 2, 3, and 4, and $\Delta D_{2D} < 4m$ in test areas 5 and 6.
- (2) Selection of the best candidate tree: We determine the nearest reference tree to the test tree from the candidate set as the best candidate tree.
- (3) Candidate testing: The matching problem is not a one-way problem. A test tree needs to find the best-matching reference tree. The reference tree also needs to find the best-matching test tree. These two trees are considered as a successful match only when the candidate tree of the test results and the candidate tree of reference data are candidate trees for each other. The accuracy evaluation parameters and calculation method are defined as follows:

- N_{test} : Number of extracted trees.
- N_{ref} : Number of reference trees.
- N_{match} : Number of matched trees.
- A_{mean} : Mean area difference between matching trees.
- R_{extr} : Extraction rate, $R_{extr} = N_{test} / N_{ref}$.
- R_{mat} : Matching rate, $R_{mat} = N_{match} / N_{ref}$.
- R_{com} : Commission rate, $R_{com} = (N_{test} - N_{match}) / N_{test}$.
- R_{om} : Omission rate, $R_{om} = (N_{ref} - N_{match}) / N_{ref}$.
- M [21]: Matching score, $M = 100 \times \frac{N_{mat}}{N_{mat} + N_{com} + N_{om}}$.

The above parameters are used to measure the detection results of one image, and the following parameters are used to evaluate the accuracy of the entire dataset:

- RMS_A : Root mean square of all the A_{mean} .
- RMS_R_{extr} : Root mean square of all the R_{extr} .

- $RMS_{R_{mat}}$: Root mean square of all the R_{mat} .
- $RMS_{R_{com}}$: Root mean square of all the R_{com} .
- $RMS_{R_{om}}$: Root mean square of all the R_{om} .
- RMS_M : Root mean square of all the M .

The root mean square (RMS) of X is defined as the formula below, where X can be A_{mean} , R_{extr} , R_{mat} , R_{com} , R_{om} , and M ; i denotes the i -th test area.

$$RMS_X = \sqrt{\frac{1}{n} \sum_1^n X_i^2}$$

4. Experiment Results and Discussion

4.1. The Comparison Results of Different Neurons

The test set is used to measure the generalization performance and classification ability of the optimal model. Therefore, we usually focus on the blue cells in the test set for each neural-network model. According to the training results, when the number of hidden neurons was 150, 300, and 450, the overall accuracy rate of the trained model for the test set was 90%, 94%, and 90.5%, respectively (Figures 9–11). Thus, the model of 300 hidden neurons can achieve the best results. Therefore, our neural network model has 300 hidden neurons. Specifically, the accuracy rate of the model for the training set is 95.2%, and the accuracy rate for the testing set is 94%. In other words, the results for the training set and the testing set exhibit no significant difference, which shows that our BP neural-network model can be applied to more occasions.

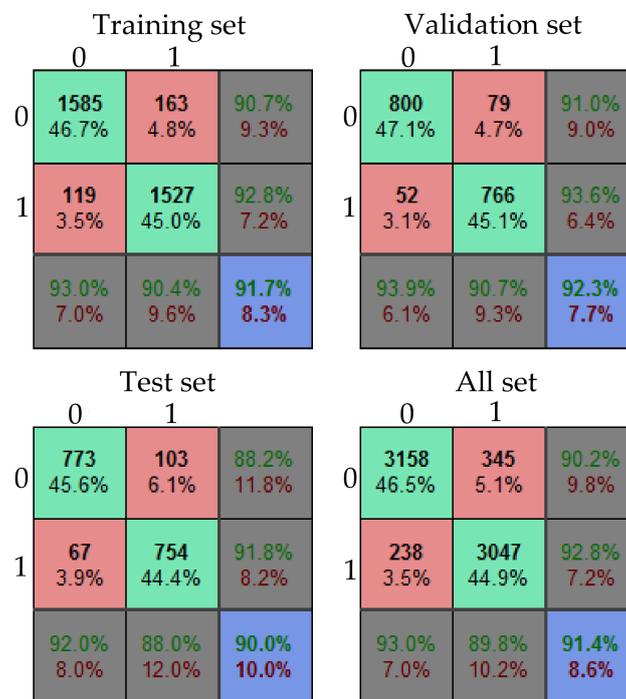


Figure 9. Training results using the BP neural-network model with 150 neurons in the hidden layer.

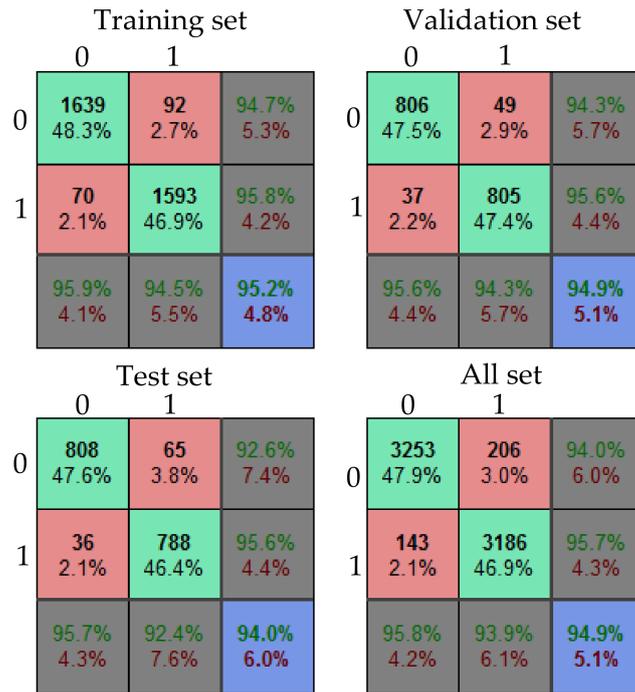


Figure 10. Training results using the BP neural-network model with 300 neurons in the hidden layer.

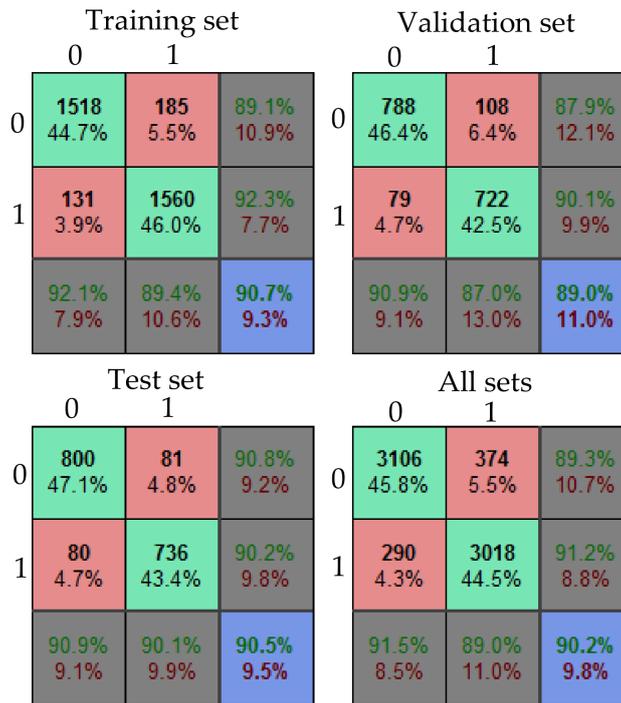


Figure 11. Training results using the BP neural-network model with 450 neurons in the hidden layer.

4.2. The Comparison Results of Different Layers

The training results of the 3–3-layer cascaded network are shown in Figure 12. From the training results, the accuracy of all the samples using the model was 97%. The accuracy rate of the training set was 97%, and the accuracy rate of the testing set was 97.2%. The results exhibit no significant difference; thus, our training model has better generality.

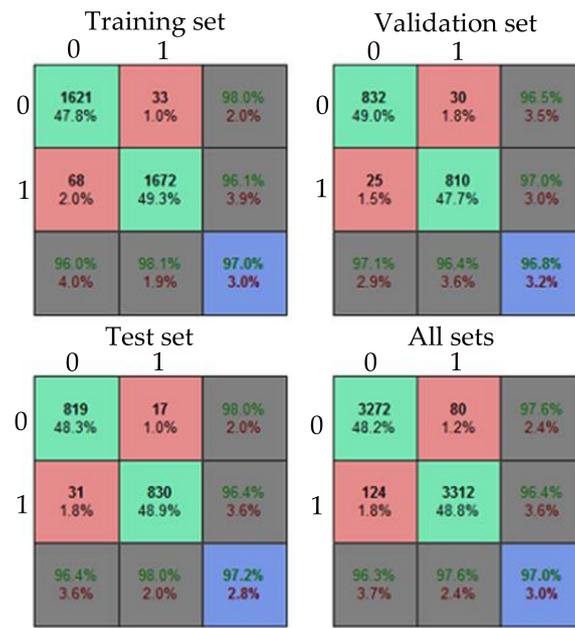


Figure 12. Training results of the 3–3-layer model.

The training results of the 3–4-layer cascade-neural-network model (Figure 13) showed that, in each of the three considered scenarios for the training set, validation set, and testing set, the 3–3-layer cascade-neural-network model achieved better results than the 3–4-layer cascade-neural-network model does, but the difference was very small. This indicates that these two network models achieved nearly comparable results. Therefore, the 3–3-layer cascade-neural-network model (shown in Figure 6) was adopted to detect individual trees for different forests. In our single-tree detection method, a list of rectangular areas is provided, and the rectangular area in the list represents an individual tree.



Figure 13. Training results of the 3–4-layer network model.

4.3. The Comparison Results of First-Order Statistical Features of the Samples

Figure 14 shows the eigenvalues of the energy, entropy, mean, skewness, and kurtosis for the positive and negative samples in the red band of the images. It can be seen from the figure that the values of energy, entropy, and kurtosis of positive samples were more stable than those of negative samples, and the values were generally smaller than those of negative samples. The fluctuation ranges of the mean and skew of the positive samples were smaller than those of the negative samples.

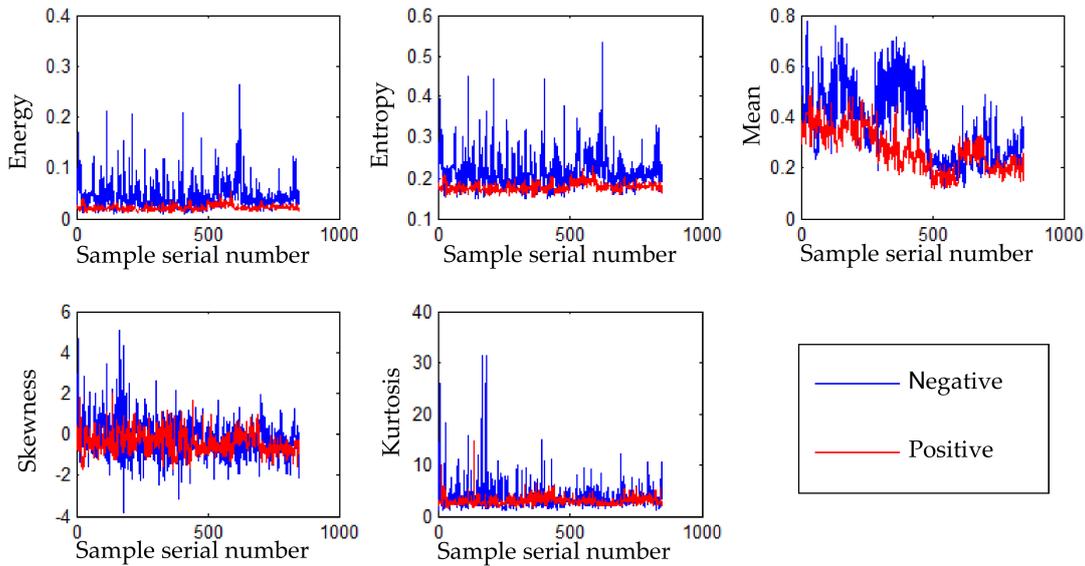


Figure 14. The first-order statistical features of the samples.

4.4. The Detection Results of Each Test Area

Figure 15 shows the detection results of the six test areas, where the red rectangle represents a detected tree. To measure the difference between detection results and the ground truth, the area of the tree crown must be estimated. Since the tree crown is approximately round in the remote-sensing images, it is easy to calculate the diameter of the crown. We took the smaller values of length and width of the rectangle for each detected tree as its crown diameter.

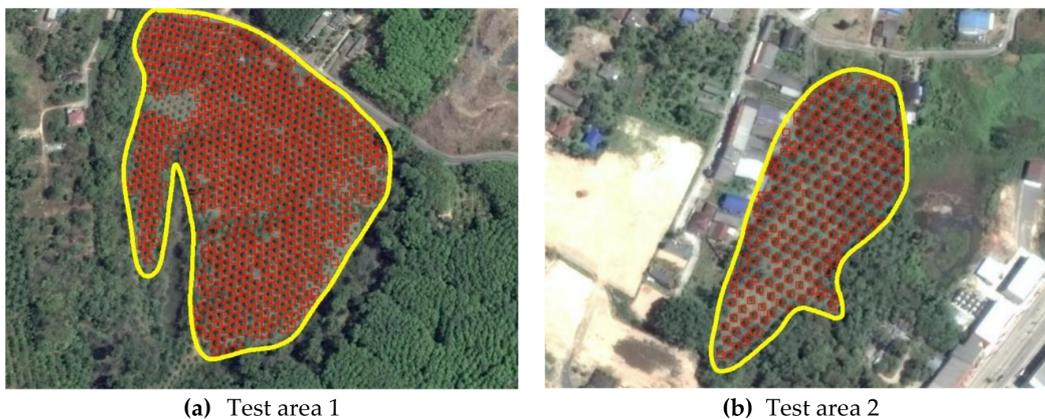


Figure 15. Cont.

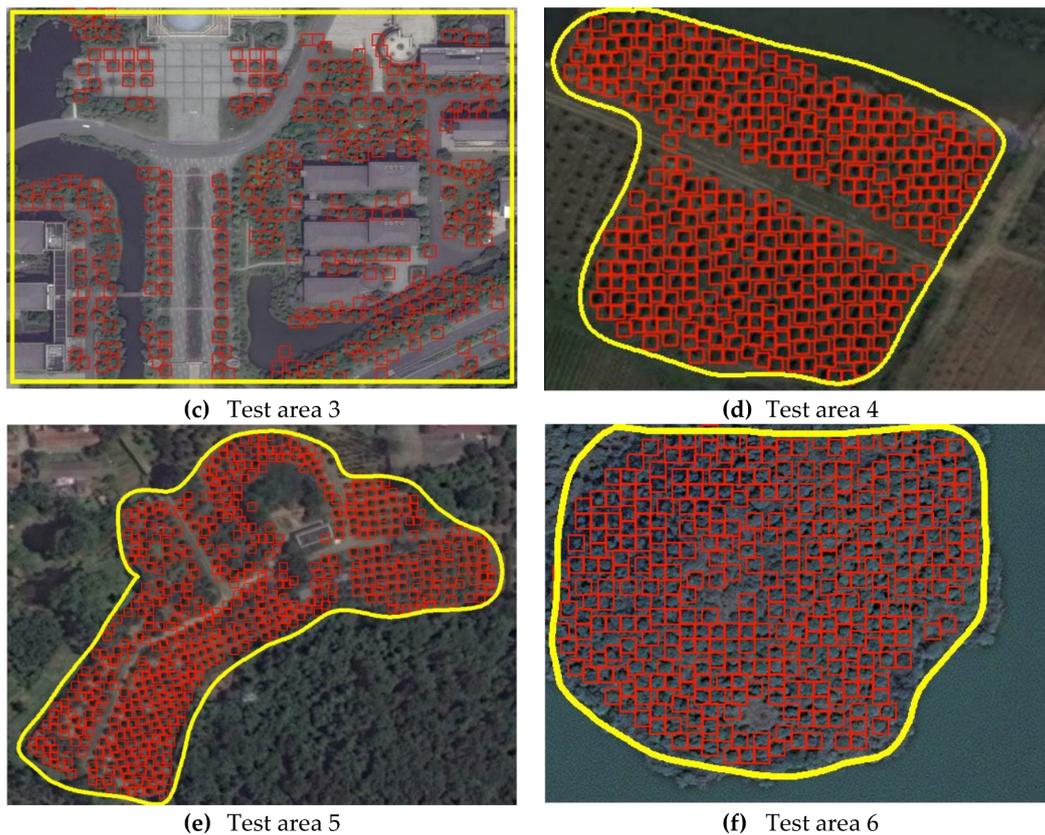


Figure 15. The detection result of each test area. The red rectangle represents a detected tree.

4.5. Detection Results of Each Test Area

Tables 1–6 present the comparison of detection results using different methods, where the region-growing method, template-matching method, BP neural network, and the cascade neural networks were adopted to detect individual trees in six different test areas. This section compares the results of N_{test} , N_{match} , R_{extr} , R_{mat} , R_{com} , R_{om} , A_{mean} , and M , which have been defined in Section 3.6.

Table 1. Detection results for test area 1.

Method	N_{test}	N_{match}	R_{extr} (%)	R_{mat} (%)	R_{com} (%)	R_{om} (%)	A_{mean} (m ²)	M
Region growing ($W = 3$)	792	703	99	88	11	12	1.71	79
Template matching ($T = 0.79$)	757	678	94	85	10	15	1.19	77
BP neural network	833	742	104	93	11	7	1.72	84
Cascade neural network	819	753	102	94	8	6	1.81	87

Table 2. Detection results for test area 2

Method	N_{test}	N_{match}	R_{extr} (%)	R_{mat} (%)	R_{com} (%)	R_{om} (%)	A_{mean} (m ²)	M
Region growing ($W = 3$)	292	254	95	82	13	17	1.87	73
Template matching ($T = 0.75$)	273	243	89	79	11	21	2.33	71
BP neural network	353	280	114	91	21	9	1.72	75
Cascade neural network	360	289	117	94	20	6	2.12	78

Table 3. Detection results for test area 3

Method	N_{test}	N_{match}	R_{extr} (%)	R_{mat} (%)	R_{com} (%)	R_{om} (%)	A_{mean} (m ²)	M
Region growing ($W = 5$)	777	231	249	74	70	26	3.24	44
Template matching ($T = 0.86$)	416	251	133	80	40	20	2.75	58
BP neural network	621	249	199	80	60	20	3.85	50
Cascade neural network	607	257	195	82	58	18	3.31	52

Table 4. Detection results for test area 4.

Method	N_{test}	N_{match}	R_{extr} (%)	R_{mat} (%)	R_{com} (%)	R_{om} (%)	A_{mean} (m ²)	M
Region growing ($W = 5$)	211	144	127	87	32	13	3.50	66
Template matching ($T = 0.68$)	185	137	111	83	26	17	1.99	66
BP neural network	222	154	133	93	31	7	2.31	71
Cascade neural network	228	165	137	99	28	1	2.13	77

Table 5. Detection results for test area 5.

Method	N_{test}	N_{match}	R_{extr} (%)	R_{mat} (%)	R_{com} (%)	R_{om} (%)	A_{mean} (m ²)	M
Region growing ($W = 5$)	515	282	152	83	45	17	2.91	57
Template matching ($T = 0.66$)	433	266	128	79	39	21	2.25	56
BP neural network	508	288	150	85	43	15	2.51	59
Cascade neural network	512	294	151	87	43	13	2.04	61

Table 6. Detection results for test area 6.

Method	N_{test}	N_{match}	R_{extr} (%)	R_{mat} (%)	R_{com} (%)	R_{om} (%)	A_{mean} (m ²)	M
Region growing ($W = 5$)	866	405	173	81	53	19	3.86	53
Template matching ($T = 0.65$)	311	255	62	51	18	49	3.71	43
BP neural network	600	397	120	79	34	21	5.06	59
Cascade neural network	611	417	122	83	32	17	5.11	63

In the region-growing method, W represents the size of the sliding window. In the template-matching method, T represents the similarity threshold between the template and the detection tree. These two parameters are very important because W and T significantly influence the detection results of the region-growing and template-matching methods, respectively. If W is too large, then only local maximum points are selected as seed points in the region-growing method; thus, a large number of little trees will be left out, and the leakage rate will be very high. In contrast, if W is too small, the same tree may be repeatedly detected and the computation cost may be very high.

In test areas 1, 2, and 4, the forestland was a large area of planted forest with similar and widely spaced tree species. The data in Tables 1, 2 and 4 show that the proposed method had a better detection score than the other three methods, especially in Table 4 ($M = 77$). In Table 3, the considered forestland had a complex area, involving trees, buildings, water, and other objects; thus, the detection results of these methods were not satisfactory in test area 3. The region-growing method achieved its best detection result when the size of the sliding window W was 5. The detection result shows that the commission rate of the region-growing algorithm is high. The main reason is that the region-growing algorithm cannot effectively avoid the interference of complex background and several incorrect seed points are selected, thus resulting in low precision. Even in such a scenario, the cascade-neural-network method can still get the highest matching rate ($R_{mat} = 82\%$) and score ($M = 52$). Therefore, the proposed method has a strong anti-interference ability. However, in Tables 5 and 6, the difference in the detection results of the four methods is not obvious because the trees were similar, and the background was singular in the scenarios of test areas 5 and 6. Even in such a relatively simple scenario, the proposed method had better tree-detection performance.

4.6. Overall Test Results

Table 7 presents the overall detection results in the six considered areas. For the region-growing method, the extraction rate was 149%, but the matching rate was low at only 83%. This led to the highest commission rate among these four methods. The template-matching method achieved the lowest extraction rate, matching rate, and commission rate. Its low matching rate led to the highest omission rate of 24%. The cascade neural network could achieve higher detection scores ($RMS_M = 68$) than the other three methods. Thus, the cascade-neural-network method can achieve the best tree-detection result for different types of forests.

Table 7. Detection results for the six test areas.

Method	RMS_R_{extr} (%)	RMS_R_{mat} (%)	RMS_R_{com} (%)	RMS_R_{om} (%)	RMS_A (m ²)	RMS_M
Region growing	149	83	37	17	2.66	61
Template matching	103	76	24	24	2.57	61
BP neural network	137	87	33	13	3.22	65
Cascade neural network	137	90	32	10	3.27	68

According to the above detection results in the six test forestlands, the region-growing method is generally suitable for trees that have clear boundaries. The system is not only able to determine the location and the size of a single tree, but it is also able to depict the outline of the tree crown. The template-matching method is more suitable for applications in a complex forest, because it is less affected by the surrounding environment. Overall, the region-growing method achieved a good matching rate, but the detection score of this method was much lower than those of the other methods. The reason for this lies in the difficulty in selecting the local maximum as the center of trees in the test area. The template-matching method is highly dependent on the quality of the template; thus, the detection effect of different regions may vary greatly. For example, the score of test area 3 was high, but the scores of other regions were very low. The BP neural-network method and cascade neural network exhibit a better performance in all six regions. In particular, in a dense forest, the cascade-neural-network method demonstrated the best performance and achieved the best detection score among all the test areas. In addition, for scenarios with different levels of complexity, our method can have better detection results, which indicates that our method has better generalization ability.

5. Conclusions

High-resolution remote-sensing images are widely used in high-precision forest resource surveys, forest management, timber production estimations, and other applications. Currently, researchers have developed various methods to extract individual trees and their characteristics in digital aerial photographs of various types. However, the existing single-tree detection methods are heavily dependent on certain features determined by prior knowledge. Furthermore, these methods cannot be applied in forest scenes with different complexities. The model needs to learn the overall characteristics of the trees in the training process, so that the trees of interest are best isolated.

To automatically and effectively identify individual trees, this paper presents a single-tree detection method for high-resolution remote-sensing images based on a cascade BP neural network. To improve the recognition rate and reduce the omission rate of the single-tree detection method, we introduced first-order statistical features of samples as supplementary features of individual trees and combined these with the BP neural network model to build a cascade-neural-network model. The experimental results show that the single-tree detection method for high-resolution remote-sensing images based on the BP cascade neural network proposed in this paper can achieve better detection results than the existing methods that obtain a highest matching rate and detection score. A BP cascade neural network does not need to extract different tree features artificially for different scenes, and the neural network automatically learns how to represent the most essential features of trees. Although the detection results of various methods are not ideal in complex scenarios, our method still maintains a good detection effect. Therefore, our method has better generalization performance for different scenarios.

The BP neural network has superiority in many aspects compared with the methods based on artificial rules. However, it is still a kind of shallow-learning model containing only a hidden layer of nodes, which requires a large number of neurons in the hidden layer. In recent years, object-detection methods based on convolutional neural networks (CNN) gradually became hotspots, mainly because this method demonstrated excellent accuracy in the field of object recognition and image classification [22–26]. The method is also used in the field of remote sensing. CNN introduces receptive fields and weight-sharing mechanisms to reduce the number of parameters that the neural network is required to train. These methods based on deep learning will be the focus of future work.

Author Contributions: Formal analysis, Z.J. and G.S. Methodology, D.T. and F.J. Project administration, D.T. and F.J. Validation, S.Y. Visualization, G.S. and S.Y. Writing—original draft, D.T. and G.S. Writing—review and editing, Z.J.

Funding: This work was supported by the following foundations: the National Natural Science Foundation of China (No.61572437, No.61672464), and the Key Research and Development Project of Zhejiang Province (No.2017C01013).

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

References

1. Lin, H. A Review on remote sensing's application, puzzle and prospect in forestry. *Remote. Sens. Inf.* **2002**, *1*, 39–43.
2. Pouliot, D.; King, D.; Bell, F.; Pitt, D. Automated tree crown detection and delineation in high-resolution digital camera imagery of coniferous forest regeneration. *Remote. Sens. Environ.* **2002**, *82*, 322–334. [[CrossRef](#)]
3. Walsworth, N.; King, D. Image modelling of forest changes associated with acid mine drainage. *Comput. Geosci.* **1999**, *25*, 567–580. [[CrossRef](#)]
4. Culvenor, D.S. Tida: An algorithm for the delineation of tree crowns in high spatial resolution remotely sensed imagery. *Comput. Geosci.* **2002**, *28*, 33–44. [[CrossRef](#)]
5. Novotný, J.; Hanuš, J.; Lukeš, P.; Kaplan, V. Individual tree crowns delineation using Local Maxima approach and seeded region-growing technique. In Proceedings of the GIS Ostrava 2011, Eight International Symposium, Ostrava, Czech Republic, 24–26 January 2011; pp. 27–39.
6. Hirschmugl, M.; Ofner, M.; Raggam, J.; Schardt, M. Single tree detection in very high resolution remote sensing data. *Remote. Sens. Environ.* **2007**, *110*, 533–544. [[CrossRef](#)]
7. Pollock, R.J. *The Automatic Recognition of Individual Trees in Aerial Images of Forests Based on A Synthetic Tree Crown Image Model*; The University of British Columbia: Vancouver, BC, Canada, 1996; p. 172.
8. Tarp-Johansen, M.J. Automatic stem mapping in three dimensions by template matching from aerial photographs. *Scand. J. For. Res.* **2002**, *17*, 359–368. [[CrossRef](#)]
9. Warner, T.A.; Lee, J.Y.; McGraw, J.B. Delineation and identification of individual trees in the eastern deciduous forest. In Proceedings of the 1998 International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry, Victoria, BC, Canada, 10–12 February 1998; pp. 81–91.
10. Malek, S.; Bazi, Y.; Alajlan, N.; AlHichri, H.; Melgani, F. Efficient framework for palm tree detection in uav images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2014**, *7*, 4692–4703. [[CrossRef](#)]
11. Lin, Y.; Wu, X.; Emil, P.; Ma, X. Tree detection from aerial imagery. In Proceedings of the 17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009; pp. 131–137.
12. Yegnanarayana, B. *Artificial Neural Networks*; Prentice-Hall of India Private Ltd.: New Delhi, India, 2009.
13. Demuth, H.B.; Beale, M.H.; De Jess, O. *Neural Network Design*, 2nd ed.; Martin Hagan: Stillwater, OK, USA; Washington, DC, USA, 2014; ISBN 0971732116 9780971732117.
14. Hecht-Nielsen, R. Theory of the backpropagation neural network. *Neural Netw.* **1988**, *1*, 445–448. [[CrossRef](#)]
15. Dreyfus, S.E. Artificial neural networks, back propagation, and the kelley-bryson gradient procedure. *J. Guid. Control. Dyn.* **1990**, *13*, 926–928. [[CrossRef](#)]
16. Li, Y.; Fu, Y.; Li, H.; Zhang, S. The improved training algorithm of back propagation neural network with self-adaptive learning rate. In Proceedings of the 2009 International Conference on Computational Intelligence and Natural Computing, Wuhan, China, 6–7 June 2009; pp. 73–76.
17. Guirado, E.; Tabik, S.; Alcaraz-Segura, D.; Cabello, J.; Herrera, F. Deep-learning versus OBIA for scattered shrub detection with google earth imagery: *Ziziphus lotus* as case study. *Remote. Sens.* **2017**, *9*, 1220. [[CrossRef](#)]
18. Chen, Z.; Zhang, T.; Ouyang, C. End-to-End airplane detection using transfer learning in remote sensing images. *Remote. Sens.* **2018**, *10*, 139. [[CrossRef](#)]
19. Larsen, M.; Rudemo, M. Optimizing templates for finding trees in aerial photographs. *Pattern Recognit. Lett.* **1998**, *19*, 1153–1162. [[CrossRef](#)]
20. Eysn, L.; Hollaus, M.; Lindberg, E.; Berger, F.; Monnet, J.-M.; Dalponte, M.; Kobal, M.; Pellegrini, M.; Lingua, E.; Mongus, D.; Pfeifer, N. A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the alpine space. *Forests* **2015**, *6*, 1721–1747. [[CrossRef](#)]
21. Larsen, M.; Eriksson, M.; Descombes, X.; Perrin, G.; Brandtberg, T.; Gougeon, F.A. Comparison of six individual tree crown detection algorithms evaluated under varying forest conditions. *Int. J. Remote.* **2011**, *32*, 5827–5852. [[CrossRef](#)]
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.

23. Le, Q.V. Building high-level features using large scale unsupervised learning. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8595–8598.
24. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; Kingsbury, B. 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](#)]
25. Sainath, T.N.; Mohamed, A.; Kingsbury, B.; Ramabhadran, B. Deep convolutional neural networks for LVCSR. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8614–8618.
26. Zhu, X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote. Sens. Mag.* **2017**, *5*, 8–36. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).