remote sensing

MDPI

Article

ELCD: Efficient Lunar Crater Detection Based on Attention Mechanisms and Multiscale Feature Fusion Networks from Digital Elevation Models

Lili Fan *, Jiabin Yuan, Keke Zha and Xunan Wang

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

* Correspondence: fanlily913@nuaa.edu.cn

Abstract: The detection and counting of lunar impact craters are crucial for the selection of detector landing sites and the estimation of the age of the Moon. However, traditional crater detection methods are based on machine learning and image processing technologies. These are inefficient for situations with different distributions, overlaps, and crater sizes, and most of them mainly focus on the accuracy of detection and ignore the efficiency. In this paper, we propose an efficient lunar crater detection (ELCD) algorithm based on a novel crater edge segmentation network (AFNet) to detect lunar craters from digital elevation model (DEM) data. First, in AFNet, a lightweight attention mechanism module is introduced to enhance the feature extract capabilities of networks, and a new multiscale feature fusion module is designed by fusing different multi-level feature maps to reduce the information loss of the output map. Then, considering the imbalance in the classification and the distributions of the crater data, an efficient crater edge segmentation loss function (CESL) is designed to improve the network optimization performance. Lastly, the crater positions are obtained from the network output map by the crater edge extraction (CEA) algorithm. The experiment was conducted on the PyTorch platform using two lunar crater catalogs to evaluate the ELCD. The experimental results show that ELCD has a superior detection accuracy and inference speed compared with other state-of-the-art crater detection algorithms. As with most crater detection models that use DEM data, some small craters may be considered to be noise that cannot be detected. The proposed algorithm can be used to improve the accuracy and speed of deep space probes in detecting candidate landing sites, and the discovery of new craters can increase the size of the original data set.

Keywords: crater detection; image segmentation; moon; deep learning; remote sensing

check for updates

Citation: Fan, L.; Yuan, J.; Zha, K.; Wang, X. ELCD: Efficient Lunar Crater Detection Based on Attention Mechanisms and Multiscale Feature Fusion Networks from Digital Elevation Models. *Remote Sens.* **2022**, *14*, 5225. https://doi.org/10.3390/rs14205225

Academic Editors: Yue Wu, Kai Qin, Maoguo Gong and Qiguang Miao

Received: 22 September 2022
Accepted: 17 October 2022
Published: 19 October 2022

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

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Impact craters constitute an important property of the lunar surface. Impact craters provide significant information for lunar evolution [1,2]. For example, the distribution and number of craters are often used to estimate the relative age of the Moon [3–5], and craters also provide important landmark information to accurately guide spacecraft to land [6,7]. The discovery of impact craters on the lunar surface is very important for studying the Moon, for example, by using the manual analysis and comparative evaluation of craters' images with different features to identify the permanently shadowed lunar polar regions [8]. In the study of crater counting, some crater catalogs have been formed manually by planetary scientists, such as the crater catalog of the Moon (diameter 5∼20 km [9], diameter $\geq$ 20 km [10]). However, the manual discovery of craters is time-consuming and laborious, and because experts may disagree on the interpretation of image data, the manual marking of craters also faces consistency and repeatability challenges.

Several automatic crater detection algorithms have been proposed to detect craters, and these can be roughly grouped into two categories. The first kind of method is unsupervised-based algorithms, which use digital image processing technology to detect craters, and the second kind of method is supervised-based algorithms, which employ machine learning or deep learning to extract impact craters.

Unsupervised-based automatic crater feature extraction algorithms are mainly based on traditional image processing methods, including Hough transforms [11–13], template matching [14], edge detection, convex grouping [15], and other recognition techniques. For example, the performance of a Hough transform applied to large scale crater counting was evaluated [13] in terms of its ability to automatically detect craters down to sub-km sizes on high-resolution images of the Martian surface. The Canny edge detector is widely used in computer vision to locate sharp intensity changes and to find object boundaries in an image. The combined adaptive Canny algorithm, which uses histograms of images and multi-scale Gaussian filtering, was used in [16] to achieve a crater matching rate of better than 85%. However, for irregular, incomplete shapes and areas with a high degree of overlap, the detection accuracy of such methods is poor. Furthermore, Chen et al. [17] used terrain analysis and mathematical morphology methods to identify different types of impact craters, which fit the crater edge based on the Moon's digital elevation model (DEM) data. In contrast, the mathematical fitting method is more reliable than the Hough ring transform algorithm, but its computational complexity is higher for the identification of large, dense craters.

Automatic crater supervised-based technology has developed rapidly through machine learning and deep learning methods. Machine learning-based methods often involve building a classifier to recognize candidate craters, and common classifiers, such as the principal components analysis [18], decision trees technique [19], support vector machine, and other hybrid methods [20], are used to classify candidate craters. To improve the classification accuracy of small craters, Kang et al. [21] combined a histogram of oriented gradient features and the support vector machine classifier to extract small-scale impact craters from charge-coupled device images. Furthermore, based on the scale of training samples generated from the surface imagery and digital elevation models of the Moon, [22] proposed an active machine learning approach to automatically detect candidate craters by training a classifier with better performance. These methods are able to recognize craters or non-craters with a high classification accuracy. However, they need to extract features manually when training a classifier to detect craters. For large-scale and high-density crater detection, most of them have poor recognition accuracy and robustness. Some of them cannot count craters or locate the positions of craters.

Deep learning, especially when based on convolution neural networks (CNNs), has achieved great success in solving problems with image classification, image segmentation [23,24], and synthetic aperture radar (SAR) automatic object detection [25,26] in the remote sensing fields. The CNN is a key representative network structure in deep learning techniques. Such techniques are different from machine learning techniques, which are more efficient and portable without a set of human-designed features [27]. Impact crater detection based on deep learning is an important method in the vision-based navigation systems and is used to solve the task of pinpoint landing on the Moon. Some works [28,29] have used CNN feature extraction and standard image processing technologies to detect and match the observed craters, which were used as visual landmark measurements by the navigation filter. Moreover, image segmentation [23,30] and object detection methods based on CNNs, e.g., faster region-CNN (R-CNN) [31] and mask R-CNN [32], are used to solve crater detection problems. For example, Tewari et al. [32] utilized the mask R-CNN framework to detect craters from optical images, digital elevation maps, and slope maps by post-processing to eliminate duplicate craters and extract the craters' global locations. Moreover, to improve the detection accuracy of small-impact craters, [33] proposed an end-to-end high-resolution feature pyramid network framework, denoted as HRFPNet. HRFPNet uses a new backbone with a feature aggregation module to enhance the feature extraction capability of small craters from thermal infrared imaging on Mars. However, most object detection-based methods need to consider the generation of the number of candidate boxes. For highly overlapping and dense craters, the quality of the generation of a large number of duplicate bounding boxes may affect the recognition speed and accuracy of crater detection. Therefore, most object detection schemes display relatively poor performances and high levels of computational complexity in crater detection.

Crater detection is also solved as a semantic segmentation problem, in which the rims or edges of craters can be extracted by pixel-level classification, and the crater position and size can be obtained by a post-pipeline method. For example, a semantic segmentation method based on the fully convolutional neural network was proposed [34]. This method uses different feature maps with multi-scale receptive fields to detect multi-scale impact craters from remotely sensed planetary images. Moreover, semantic segmentation models [35–37] based on U-net [38] have been presented to detect craters. Silburtet et al. [35] proposed DeepMoon based on the U-net network structure to recognize lunar craters from DEM data. This method can successfully identify about 45% of newly discovered craters in its validation data. However, the U-net network structure loses large amounts of detailed information in the encoder of the network, which leads to poor crater image contour recovery in the decoder process. To improve the accuracy of crater detection, a new network structure, ERU-Net [36], introduced the deep residual network module to improve the crater feature extraction ability. This successfully achieved a recall rate of 81.2% and a precision rate of 75.4% in lunar crater recognition when training 30,000 DEM data images. Furthermore, to explore craters on Mars, DeLatte et al. [37] employed segmentation convolution neural networks based on U-net for automatic crater detection from Martian daytime infra-red images. This method identified 65–75% of craters in common with a human-annotated dataset, and [39] used the ResUNET [30] model to detect craters with the global maps and infra-red imagery for Mars. However, resources in the deep space environment are limited [40]; thus, automated crater detection methods require a balance between model computational complexity and identification efficiency. Most of the above methods ignore the computational complexity of the model.

The deep learning-based algorithms described above have different improved optimization approaches for different crater tasks. However, the majority of object detection schemes perform relatively poorly as they are constrained by their vanilla network architectures or semantic segmentation. By comparing the network complexity and recognition results, it can be seen that crater detection methods based on the semantic segmentation model are more efficient than the end-to-end object detection model. However, most semantic segmentation-based crater detection methods mainly focus on the accuracy of recognition and neglect the reasoning speed of the network. Moreover, due to crater images having different distributions, degrees of overlap, and sizes on the surface of the Moon, and because the crater data may be imbalanced, crater detection algorithms based on semantic segmentation networks may suffer from significant performance degradation. Therefore, achieving a fast and effective crater detection method with a high level of precision based on a semantic segmentation model represents a challenging scenario.

To address this issue, in this study, we establish an efficient lunar crater detection (ELCD) algorithm that addresses the requirements for accurate and fast crater detection. In the ELCD algorithm, first, the crater edge is segmented by the attention mechanisms and multiscale feature fusion networks (AFNet). Then, the crater position and size are extracted by postprocessing based on the crater extract algorithm (CEA). In AFNet, a light-weight attention mechanism is used to improve the feature extraction ability of the network, and a new multiscale feature fusion (MFF) module is designed in the upsampling process of the network to reduce the loss of detail in the semantic segmentation results. In addition, we consider the crater data imbalance of the classification and distributions and design a new crater edge segmentation loss (CESL) function for network training. The proposed loss function improves the optimization ability and convergence speed of the network through adaptive balance weights.

The main contributions of the paper are as follows:

- We propose an efficient crater detection network based on a new semantic segmentation network architecture, AFNet, which uses the lightweight attention mechanism and multiscale feature fusion module to provide better and faster detection of lunar impact craters.
- To improve the optimization capability of the network, we present the crater edge segmentation loss function, which considers the imbalance of classification and distributions of crater data to calculate the loss value using the different degrees of imbalance in the data.
- The experiment is conducted on the PyTorch platform [41] with lunar DEM data to verify the effectiveness of the ELCD. The results show that the ELCD outperforms the state-of-the-art crater detection models in terms of its detection accuracy and inference speed.

The rest of this paper is organized as follows: Section 2 describes the proposed network architecture, the design of the crater edge segmentation loss function, the crater edge extraction algorithm, and the details of the experiment. Section 3 provides the experimental results, and Section 4 presents our discussion. Eventually, in Section 5, we conclude our work.

2. Materials and Methods

The workflow description of two stages of the lunar crater detection method using DEM data is shown in Figure 1. The workflow includes two parts: (i) crater edge prediction by the semantic segmentation network AFNet and (ii) crater edge extraction with the post-pipeline method. The details of the ELCD are as follows. The workflow input is the lunar crater DEM image. The DEM contains abundant 3D morphology and topography morphological characteristics, and it is insensitive to light [27]. The workflow output is the crater's positional information, such as its longitude, latitude, and radius, which is determined by the crater edge extraction algorithm. First, crater images with different degrees of size, overlap, and distribution are transferred to the crater edge segmentation network to undergo crater edge prediction. Then, the network prediction results are processed with a post-processing pipeline based on the match template method to obtain the location information and radial size of craters.

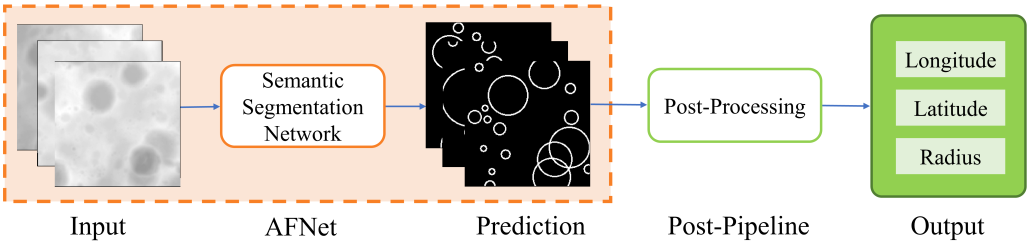


Figure 1. Workflow of two stages used in the crater detection method based on the semantic segmentation network and the crater edge extraction method. The network input is the DEM image. The digital elevation model (DEM) image is first processed by AFNet to recognize crater edges by pixel-level classification. Then, the prediction result of the crater images from network training is processed by a post-processing pipeline based on the match template method to detect the location information and radius size of craters.

2.1. AFNet

To obtain efficient crater edge prediction results, we formally describe the crater edge detection network architecture, as shown in Figure 2. The AFNet includes three parts: the network encoder, feature fusion, and decoder. In Figure 2, the black line is the network encoder, the blue line denotes the process of feature fusion, and the orange line represents the network decoder process. The network input is the gray DEM image, which has a fixed size of 256 × 256 pixels, and the output is the pixel-level classification for the prediction result.

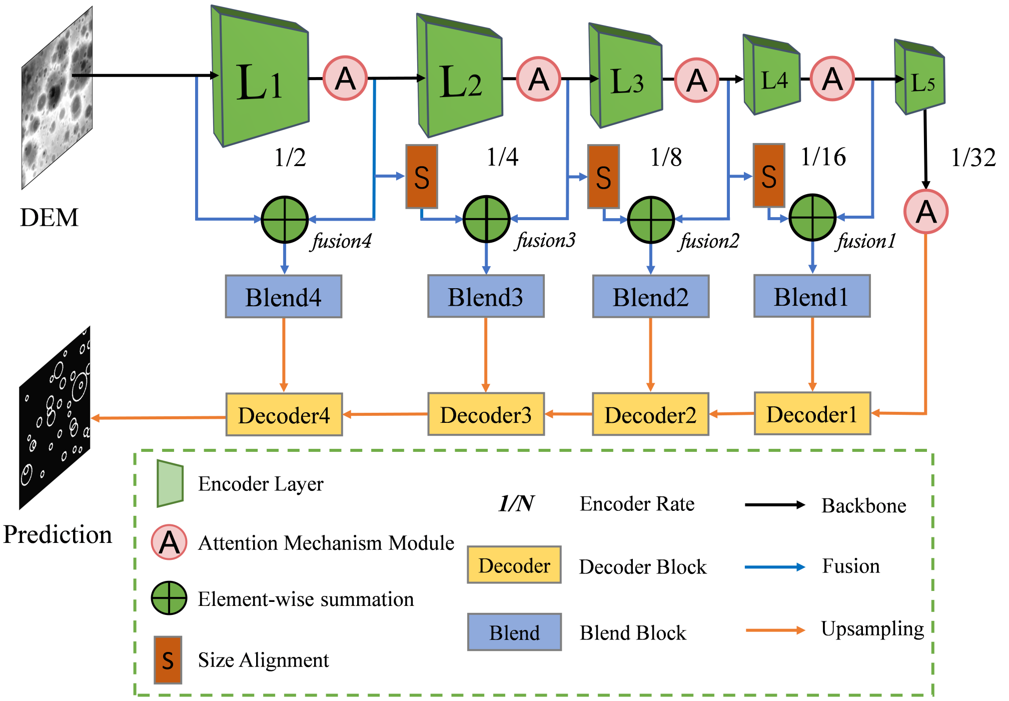


Figure 2. AFNet framework based on the improved VGG-16. The input is the DEM image transferred to the network encoder process (green trapezoid). First, the DEM image is processed with a $1/N$ downsampling rate with an attention mechanism module (pink circle) and five convolution blocks. Then, feature maps with different resolutions are saved and fused by the multiscale feature fusion module (blue line) with element-wise summation (green $\oplus$) and the data blending block (blue squares) through the decoder process (yellow squares) to get a more fine-grained output feature map. The final output result denotes the network prediction results with pixel-level classification.

In encoder processing, we use the visual geometry group-16 (VGG-16) [42] as the backbone to extract the crater features. This allows us to obtain a bigger receptive field using fewer parameters compared with other network structures. The backbone network includes five feature extraction blocks, denoted as $L = \{L_1, L_2, \dots, L_i\}$, where i is the number of feature extraction blocks. At the end of each feature extract block, we introduce the attention mechanism module to extract the important features of the crater. In L_1 and L_2, each feature extraction block contains two convolution layers: an attention machine module and a max-pooling layer. L_3, L_4 and L_5 contains three convolution layers, an attention machine module, and a max-pooling layer, and all convolution layers use a 3 × 3 convolution kernel in each block.

In feature fusion, to obtain a more fine-grained feature map in the network decoder, we designed a simple and efficient MFF to obtain more fine-grained output feature maps. The four fusion modules $fusion_j$ are shown in Figure 2 and $j = \{1, 2, 3, 4\}$. The MFF first uses the element wisdom summation (green in Figure 2) to fuse a low-resolution feature map and a high-resolution feature map in each step of the upsampling process (decoder). Then, the obtained fusion feature map is blended and transferred to the decoder process as an input for the next step (blue squares in Figure 2).

In decoder processing, the bilinear interpolation operation is used to restore the size of the feature maps by four decoder blocks, $Decoder_k$, $k = \{1, 2, 3, 4\}$ (yellow squares in Figure 2). We use 2× upsampling and fuse more rice feature map information in each decoder to restore the feature map to its original size.

2.2. Attention Mechanism Module

The original impact craters have different density distributions, sizes, and degrees of overlap in the different lunar regions. A description of the characteristics of crater DEM data used in network training is given in Figure 3. When the crater DEM images are processed by random clipping, they may have an incomplete shape. These crater characteristics bring performance challenges to the semantic segmentation network.

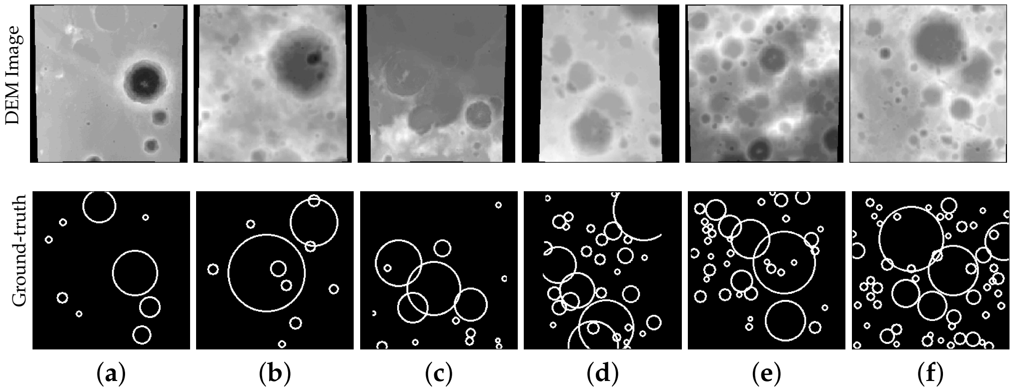
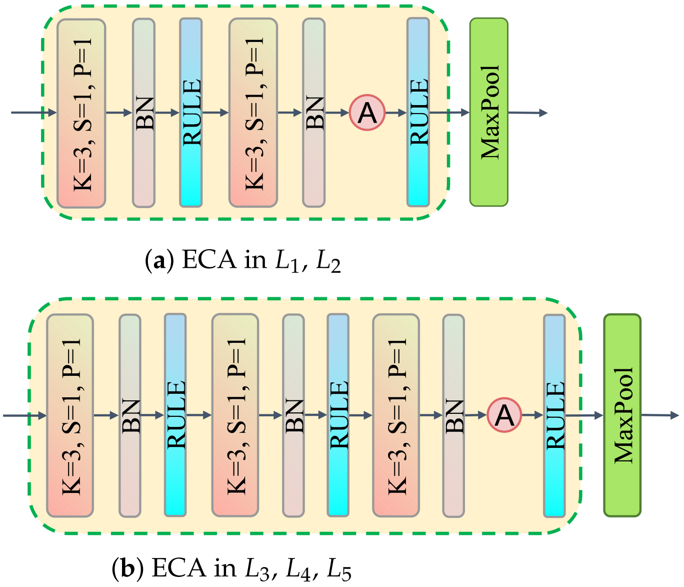


Figure 3. The different characteristics of craters on the surface of the Moon from DEM data. The top figure is the original crater images, and the bottom figure denotes the labeled images denoted as the ground truth. (**a**–**c**) show the distribution of sparse craters, (**d**–**f**) denote the distribution of dense craters, (**b**–**f**) represent the different degrees of crater overlap, (**c**–**f**) show the incomplete craters.

In the encoder, to improve the feature extraction ability of the network, we introduce the attention mechanism through efficient channel attention (ECA) [43], which is attached to the end of each feature extraction block of the proposed network to enhance the extraction of important features. Efficient channel attention with the lightweight module has great potential to produce a trade-off between performance and complexity. This only involves a handful of parameters while bringing a clear performance gain. The ECA block is termed an attention mechanism, as shown in Figure 2 with a pink circle. In the ECA, 1D convolution with a kernel size of 3 was used to achieve information exchange between channels. The details of the ECA block attached to the end of the five feature blocks are given in Figure 4. The ECA module was placed behind the activate function rectified linear unit (ReLU) in each feature extraction block. Figure 4a denotes the location of the ECA in the feature extraction block $\{L_1, L_2\}$, and Figure 4b shows the location of the ECA in the feature extraction block $\{L_3, L_4, L_5\}$ in the decoder process of the network. The ECA can combine the crater channel and spatial attention to enhance crater feature aggregation, which can enhance the extraction of salient crater features.

2.3. Multiscale Feature Fusion Module

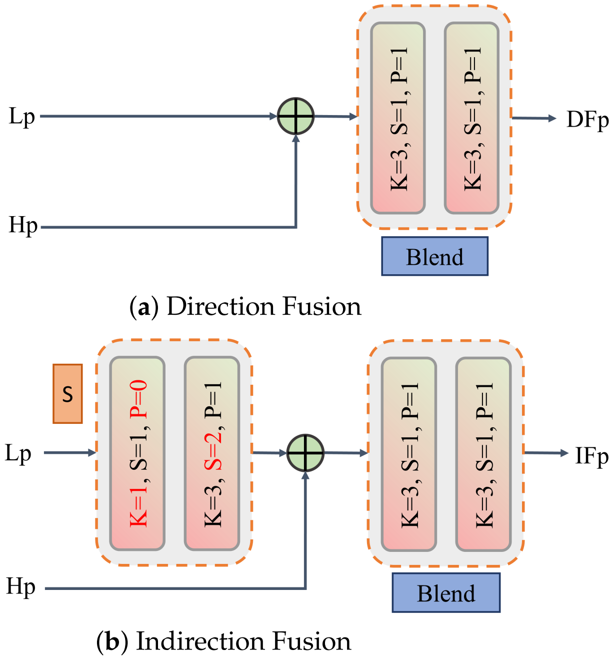
Visual features with a coarse spatial resolution can be obtained by the encoder process. During the network encoder process, shallow crater networks can learn some local features because of the low perception threshold, and the deeper convolution layer can obtain more abstract features. With the deepening of the network, the receptive field of the network becomes larger, but because of the down-sampling operation, a great deal of detailed information may be lost. The purpose of the decoder process is to obtain a segmented prediction image with the same input size through the upsampling operation. Traditional segmentation networks use the simple upsampling module with skipped lateral connections to restore the feature map, which may cause the restored feature map to lack detailed features. To overcome the problem of poor image contour recovery in the decoder process, we designed a simple and efficient multiscale feature fusion module to fuse more low-layer features in each decoder block. The four multiscale feature fusion modules $fusion_j$, $j = \{1, 2, 3, 4\}$, are shown in Figure 5. We first obtained feature maps of different resolutions from the network encoder process. Then, we fused two close feature maps as the upsampling input for the MFF to obtain an output feature map with more fine-grained information.



(**a**) ECA in L_1, L_2

(**b**) ECA in L_3, L_4, L_5

Figure 4. ECA module. (**a**) ECA module in L_1 and L_2, (**b**) ECA module in L_3, L_4, and L_5.



(**a**) Direction Fusion

(**b**) Indirection Fusion

Figure 5. Multiscale feature fusion module. (**a**) is the direct fusion with the same feature map resolution and size. (**b**) denotes indirect fusion with feature maps of two different resolutions.

The MFF included two cases, direct fusion and indirect fusion, as shown in Figure 5. Figure 5a shows direct fusion for two feature maps of the same resolution: $fusion_1 - fusion_3$. Figure 5b denotes indirect fusion with two different resolution feature maps: $fusion_4$. In direct fusion, a low-resolution feature map denoted as Lp and high-resolution feature map represented as Hp have the same resolution. They use direct fusion by the element summation operation to obtain the fusion feature map. However, the size of the feature map is often different and usually has a two-fold difference in size after the encoder process. Therefore, processing is done through the indirect fusion module. In the indirect fusion module, the low-resolution feature map is not the same as the high-resolution feature map. Lp is first processed with the size alignment module to obtain the same resolution as Hp. Then, the two maps are fused by element summation to obtain the fusion feature map. Finally, the fusion feature map is transferred to the blending block (blue squares in Figure 2) to obtain the final fusion output, the indirect fusion feature map denoted as IFp or the direct fusion feature map indicated as DFp, as the branch input for upsampling processing.

The size alignment module includes a 1 × 1 convolution kernel to reduce the dimensions and a 3 × 3 convolution kernel. This stride is set to 2 to adjust the map to the same size as Hp. The blend module contains the simple two 3 × 3 convolution kernel network to blend the fusion results. The final fusion feature maps have richer low-layer features, which could help us to obtain high-quality output prediction results in the encoder process.

2.4. Crater Edge Segmentation Loss Function

In the crater prediction network, crater images can be divided into foreground images and background images by pixel-level segmentation. In a crater image, the foreground image is the segmented object (crater edge), and the background image represents everything but the object. However, most crater detection methods based on segmentation networks use traditional loss functions, such as the cross-entropy (CE) loss function [35–37], to train the network, and they cannot overcome the variation in size and the serious crater data imbalance problem, resulting in a performance decrease. The CE can be computed as

$$CE(p_i, y_i) = \begin{cases} -log(p_i), & y_i = 1 \\ -log(1 - p_i), & otherwize \end{cases} \tag{1}$$

where p_i is prediction value of the network, y_i is the ground-truth, and $p_i \in$ [0,1], y$\in \{0,1\}$.

However, in cross-entropy loss, the weight of each sample is the same, and the CE loss is overwhelmed when facing the data classification imbalance. Later, the focal loss (FL) function [44] considering the classification imbalance in dense object detection was proposed to improve the network performance. The FL is defined as

$$FL(p_i) = -\alpha(1 - p_i)^{\gamma} log(p_i) \tag{2}$$

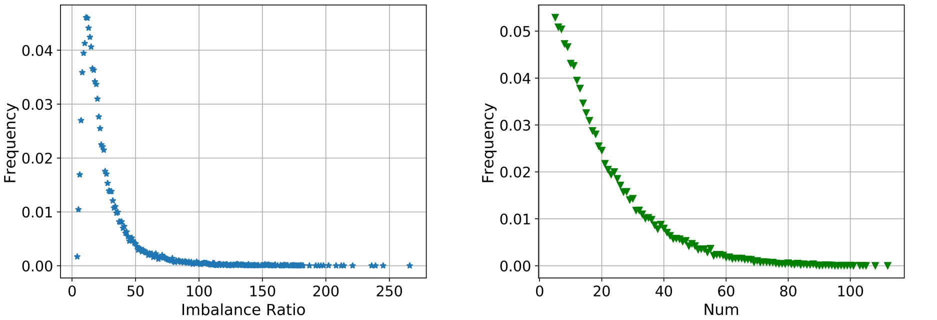
where α is a weighting factor, $\alpha \in$ [0,1] for class 1 and $1 - \alpha$ for another class; $(1 - p_i)^{\gamma}$ denotes the modulating factor; and γ denotes the tunable focusing parameter. The FL can balance the importance of positive and negative examples and differentiate between easy and hard examples by modulating the two factors α, and γ.

Inspired by the FL [44], we propose a novel crater edge segmentation loss function to optimize the proposed network. In contrast to FL, only the classification imbalance of data was considered when designing the loss function. In this paper, two data imbalance factors were considered, including the classification imbalance and the distribution imbalance of crater data, and the modulating factor of the loss function was set adaptively. We first calculated the imbalance characteristics of the crater data, which are shown in Figure 6. We used the data imbalance ratio (IR) to represent the crater data classification imbalance. This is the ratio between the numbers of majority class samples (background) and the minority class samples (object). The crater classification imbalance is shown in Figure 6a. Moreover, we counted the distribution imbalance ratio (DR) as the number of craters in each label's image, as shown in Figure 6b.

We set the parameters α' and γ' adaptively based on the DR and IR of the crater data for the CESL. The proposed craters edge segmentation loss function can be computed as

$$CESL(p_i) = -\alpha'(1 - p_i)^{\gamma'} log(p_i) \tag{3}$$

where α' is used to adjust the weights of different categories, and γ' is employed to differentiate between easy and hard examples. In this work, our goal was to accurately detect lunar craters. Some crater images are easy to distinguish, while others are difficult to distinguish. Simple examples show that the distribution of some craters is sparse and complete with little overlap, making these craters easy to detect. These crater images are shown in Figure 3a,b. The hard example shows that, in the lunar crater image, the distribution of craters is dense with high overlap, and the shape is incomplete. These crater images are shown in Figure 3b,c. α' and γ' were calculated based on the average value in each trained batch. IR_b and DR_b are the classification imbalance ratio and the distribution imbalance ratio in the b-batch of network training.



(**a**) Crater Classification Imbalance (**b**) Crater Distribution Imbalance

Figure 6. Data distribution statistics of impact craters. We randomly generated 30,000 crater training images to show the imbalanced distribution. (**a**) is the crater image classification imbalance, the x-axis is the data imbalance ratio (IR) [45], and the y-axis denotes the frequency distribution of the IR. (**b**) shows the distribution imbalance ratio (DR) of the craters, the x-axis denotes the number of craters in each training image, and the y-axis represents the frequency distribution of the crater number in the DEM images.

In our crater data, we found that classification imbalance was common in the training data of each DEM image, and we calculated the max IR to be about 266 times and the average IR to be about 26 times, as shown in Figure 6a. In the crater training image, the densest crater image has 112 craters, and the average crater number is 20, as shown in Figure 6b. We defined the data imbalance degree in three cases based on the imbalance characteristics of craters, namely, low classification imbalance, median classification imbalance, and high classification imbalance. To balance the proportions of the data distribution, we calculated the ratio of three imbalance degree cases, which are more balanced when the ratio is about 3:2:1 in the crater training data, and the range of the corresponding IR is $IR > 40$, $20 < IR <= 40$, and $IR > 40$. The value of α' was set by the degree of imbalance, where IR_b was used to adjust the data imbalance with different weights. α' was set as

$$\alpha' = \begin{cases} 0.2, & IR_b < 20 \\ 0.3, & 20 < IR_b <= 40 \\ 0.4, & IR_b > 40 \end{cases} \tag{4}$$

Moreover, in general, highly overlapping, dense data may have a bad effect on crater classification. Thus, we also considered the craters' sparse distribution characteristics to improve the crater classification accuracy by setting the different values of γ'. The craters' sparse distribution characteristics DR were represented by the crater number in the DEM images. We defined DR_b by the crater number in the DEM images to set γ'. The parameter γ' is defined as

$$\gamma' = \begin{cases} 2, & DR_b < 20 \\ 1, & 20 < DR_b <= 100 \\ 1.5, & DR_b > 100. \end{cases} \tag{5}$$

2.5. Crater Extraction Algorithm

The crater image segmentation results were obtained by AFNet. The results included activated pixels corresponding to the locations of the crater rims. We were able to extract crater positions and sizes from the crater image segmentation results through the post-pipeline method with the crater extraction algorithm based on the template matching method. Most impact craters are circular on the lunar surface. The craters are detected by the ring feature in the extraction algorithm. However, for overlapping craters, traditional methods (such as Hough transform, Candy) [46] cannot detect rings in the segmentation results efficiently. We used the more efficient match template algorithm in scikit-image [47] (an image processing library implemented in Python programming language) to extract crater positions. This method was used in [36,37] for crater edge extraction.

The proposed CEA received the prediction map I of the crater segmentation network and output the crater evaluation results. The crater extraction pipeline process is as follows. First, a prediction result is filtered by the binary threshold β, described as

$$p_i = \begin{cases} 1, & p_i \geq \beta \\ 0, & p_i < \beta \end{cases} \tag{6}$$

where p_i is the pixel intensity. p_i is set to 1 when p_i is greater than β; otherwise, p_i is set to 0. Then, the match template algorithm is applied to match the crater over a radius range with a maximum radius r_{max} and minimum radius r_{min}. The match template threshold P_m is used to choose the high confidence target. Lastly, an evaluation of whether the crater is correctly identified is carried out.

We detected the minimum radius r_{min} of the craters as 5 km and the maximum radius r_{max} as 40 km from the network prediction result by the CEA. This algorithm iteratively slides generated rings through the target, and it calculates the match threshold at each (x, y, r) coordinate to eliminate false target results, where (x, y) is the centralization of the generated ring, and r is the radius. Any (x, y, r) ring with a match probability greater than P_m is classified by the coordinate and radius constraints to get the correct crater, expressed as

$$[(x_i - \tilde{x}_j)^2 + (y_i - \tilde{y}_j)^2]/min(r_i, \tilde{r}_j)^2 < D_{x,y} \tag{7}$$

$$|r_i - \tilde{r}_j|/min(r_i, \tilde{r}_j) < D_r \tag{8}$$

where (x_i, y_i) is the position of the crater c_i extracted from the prediction image I, x_i, y_i are the latitude and longitude of I, respectively, and r_i is the radius of the crater c_i. For the ground-truth image $\tilde{I}$, $(\tilde{x}_i, \tilde{y}_i)$ presents the position corresponding to the crater c_i, $\tilde{x}_i$ is the latitude of the crater, $\tilde{y}_i$ is the longitude of the crater, and the radius of crater c_i is $\tilde{r}_i$. $D_{x,y}$ is the error threshold of the longitude and latitude, and D_r is the radius error threshold. When the detection crater meets these limits, it is regarded as the correct crater; otherwise, it is considered a false crater.

The pseudo-code of the efficient lunar crater detection ELCD algorithm includes crater edge prediction by the semantic segmentation network AFNet and the post-pipeline method with CEA, as described in Algorithm 1.

The input of the network contains the test DEM data Y with a pixel size of 256 × 256 for the DEM image, the number of batch image processes $|Z(\widetilde{k})|$, the crater classification number N_{class}, the trained network model $\overline{M}$, and the ground-truth of the crater image $\widetilde{Y}$. The outputs are the position and size of the crater and the evaluation of the crater detection results. First, the batch data $Y(i)$ of crater images in test set Y are transferred to the trained model $\overline{M}$ by the AFNet to obtain the prediction results $pred_{dem}$ of the network. Then, the prediction feature map $pred_{dem}$ is processed by binary threshold processing β, using the match template threshold P_m to filter out matching craters. The correctly identified craters are evaluated by the error constraints shown in Equations (7) and (8), and the results of the evaluation are counted using statistical functions $Count()$. Finally, the position and size of the crater *Pos* and the evaluation results *Det* of the correctly identified craters are obtained using the mean results for the test crater DEM data Y.

Algorithm 1: Efficient Lunar Crater Detection Algorithm

```
Input :
A set of DEM images, test dataset $Y$;
Each batch test data $Y(i)$ has $|Z(\widetilde{k})|$DEM images;
Each DEM image $y_i \in Y(i)$ has a size of 256 × 256 pixels;
The category of pixel segmentation is $N_{class}$ ;
The trained network model $\overline{M}$;
The ground-truth in the test dataset $\widetilde{Y}$;
Output:
 the information about the position and size Pos, and the crater detection results Det;
begin
    $\widehat{Z}$ = [];
    Load test dataset $Y$;
    $Pre - processing$ test dataset $Y$ by normalization;
    model.eval();
    for Each batch $Y(j)$ in $Y(i)$ do
        for Each image $y$ in $Y(j)$ do
            $orig_{dem}$ = $y[0]$;
            $true_{dem}$ = $y[1]$;
            model.cuda(), $orig_{dem}$.cuda(), $true_{dem}$.cuda();
            $pred_{dem}$ = $\overline{M}$ ($orig_{dem}$);
            $\widehat{Z}$.add ($pred_{dem}$);
        end
    end
    Match = [];    // the results of the crater match template
    Pos = [];      // the information about the crater's position and size
    Det = [];      // the crater detection results
    Match-template(); // calculate template matching
    Count();       // statistical crater detection performance
    for Each DEM $z_k$ in $\widehat{Z}$ do
        if $N_{class} = 2$ and $z_k > \beta$ then
            $z_k \leftarrow 1.0$;
        else
            $z_k \leftarrow 0.0$;
        end
        if Match-template($z_k$) $> P_m$ then
            Match.add ($longitude, latitude, radius$);
        end
        Choose the correctly identified craters by Equations (7) and (8);
        Output result $Ps, Dt \leftarrow Count\ (Match, \widetilde{y}_k)$, $\widetilde{y}_k \in \widetilde{Y}$;
        Pos.add(Ps);
        Det.add (Dt);
    end
    Pos = mean(Pos);
    Det = mean(Det);
end
Output: Pos, Det
```

2.6. Experiments

In this section, we describe the experiments conducted to verify the performance of the proposed algorithm. The experiments involved the experimental setup, experimental datasets, evaluation metrics, and comparison algorithms. The details are given below.

2.6.1. Experimental Setup

The experiment was performed on a single GPU (NVIDIA GeForce RTX 3060, 64GB RAM, 8 core CPU) with CUDA 11.0 and PyTorch 1.7.1. The CE-Adam [48] optimizer was used to improve the capability of the network model, and the learning rate was set to 1×10^{-4}. The number of iterations in the network was set to epoch 100, and the batch size was set to 32. We conducted crater detection experiments on the lunar DEM datasets, where the input DEM image was 256 × 256 pixels in size. The crater edge semantic segmentation network AFNet and crater edge extract results were evaluated using relevant evaluation criteria, as detailed in Section 2.6.3.

2.6.2. Datasets

In our experiment, we used lunar DEM data from the Lunar Reconnaissance Orbiter (LRO) and the Kaguya merged digital elevation model. The resolution of the DEM was about 59 m/pixel [49], and it spanned 180° W to 180° E and 60° S to 60° N . The global DEM map was downsampled to 118 m/pixel with a size of 92,160 × 30,720 pixels. This was used to randomly generate crater images that were 256 × 256 pixels in size.

Two lunar crater catalogs were used for the ground truth. The first catalog was termed Head [10], where the size of the crater was larger than 20 km in diameter. The other catalog was taken from Povilaitis [9], and the crater diameter size was 5–20 km. We used the combined catalog, termed Head-LROC, to train our model in this paper. The total numbers of Head and Povilaitis craters were 5186, and 19,337, respectively. The different distributions and diameter sizes of craters based on the Head-LROC catalog are shown in Figure 7. We can see that around 51.5% of craters had a diameter of less than 10 km, which accounts for more than half of all data. Moreover, around 78.8% of craters had a radius of less than 20 km, representing about three-quarters of all crater data. Only 1.3% of craters had a radius of greater than 100 km.

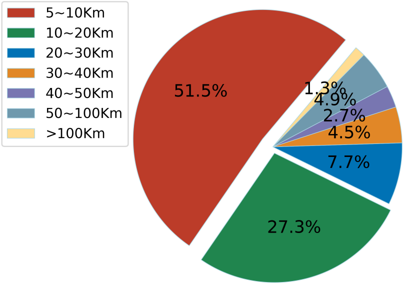


Figure 7. The distribution proportions of the different radius craters in the Head-LROC catalog [9,10].

In the experiment, the original crater images and ground-truth images were generated by the global DEM map and two lunar crater catalogs. The numbers of generated training sets, validation sets, and test sets were 30,000 DEM images, 3000, and 3000, respectively. The training set was processed by the random invert method. We randomly inverted θ to the DEM image using random number probability p, $p \in [0,1]$, where θ is defined as

$$\theta = \begin{cases} 0^{\circ}, & 0 <= p < 0.25 \\ 90^{\circ}, & 0.25 < p <= 0.5 \\ 180^{\circ}, & 0.50 < p <= 0.75 \\ 270^{\circ}, & 0.75 < p <= 1. \end{cases} \tag{9}$$

2.6.3. Evaluation Criteria

In two-stage crater detection algorithms, the performance of the prediction network may affect the final crater edge extraction result. When other parameters were fixed, the clearer the crater edge was segmented, the better the crater edge extraction result was. Thus, we first evaluated the performance of the proposed crater edge segmentation network, AFNet. The four metrics from common semantic segmentation criteria [23,24] were used to evaluate the proposed network model. We computed four metrics, the pixel accuracy (PA), mean pixel accuracy (MPA), mean intersection over union (MIoU), and frequency weighted intersection over union (FWIoU), to evaluate the performance of AFNet. Via an ablation study, we can prove the validity of our proposed model and the improved crater edge segmentation loss function.

After obtaining the crater image segmentation results, the crater positions and sizes can be obtained through the crater extraction algorithm. To evaluate the crater detection performance of the proposed ECLD algorithm, we used an evaluation method that is commonly used in machine learning to evaluate the precision (P), recall (R), and F_λ-score (F_1 or F_2) for each identified crater basis. The detection precision is the ratio of matching numbers N_{match} to detection numbers N_{detect} of craters. The recall was computed by the ratio of matching numbers N_{match} to the number of human-annotated N_{csv}, and the F_λ-score was used to balance the precision and recall. For the F_λ-score, λ denotes the tune parameter. When $\lambda > 1$, the recall is more important; otherwise, when $\lambda < 1$, the precision is more important for the model's evaluation. The detailed calculation process is described in [35,36].

Many truly existing craters were not marked in the ground truth; they were regarded as false negatives. In addition, in this paper, we used the combined lunar crater catalog Head-LROC [9,10]. The label of the training dataset was incomplete in the crater catalog, and some newly discovered craters were identified through network prediction. We calculated the discovery rate, that is, the false-positive rate for crater recognition. We used two methods to evaluate newly discovered craters. R^1_{new}, R^2_{new} was computed as

$$R^1_{new} = \frac{FP}{FP + TP} \tag{10}$$

$$R^2_{new} = \frac{FP}{FP + TP + FN} \tag{11}$$

where R^1_{new} denotes the ratio between the newly discovered craters and all recognized craters. TP denotes true positives and FP denotes false positives. The second evaluated method used was R^2_{new}, which shows the proportion of newly discovered craters to all impact craters, and FN indicates false negatives.

In the process of lunar crater recognition, the performance of the model was evaluated from the accuracy computation by the positions and sizes of the recognized craters. We calculated the latitude error (E^{lo}), longitude error (E^{la}), and radius error (E^r) to evaluate the network model using

$$E^{lo} = \frac{abs(lo^p - lo^t)}{2 \times (r^p + r^t)} \tag{12}$$

$$E^{la} = \frac{abs(la^p - la^t)}{2 \times (r^p + r^t)} \tag{13}$$

where lo^p denotes the predicted longitude value, and lo^t is the corresponding true longitude value of the crater. In Equation (13), la^p is the latitude value of the predicted crater, and the latitude value of the corresponding true crater is denoted as la^t.

The radius error (E^r) was calculated as follows:

$$E^r = \frac{abs(r^p - r^t)}{2 \times (r^p + r^t)} \tag{14}$$

where r^p denotes the radius of the predicted crater, and the corresponding true radius of the crater is indicated as r^t.

2.6.4. Compared Algorithms

The proposed algorithm ELCD was compared with five different crater detection algorithms using image segmentation technology that contained DeepMoon [35], ERU-Net [36], D-LinkNet [23], and SwiftNet [24]. The general procedure used for each algorithm was as follows:

- DeepMoon [35]: The basic idea of this algorithm is that deep learning based on the U-net network architecture is used to train the lunar crater DEM data to discover lunar craters.
- ERU-Net [36]: To improve the detection accuracy of lunar craters, ERU-Net introduced the residual network module to the U-Net network architecture to enhance the crater feature extraction ability.
- D-LinkNet [23]: D-LinkNet with high efficiency is often used for comparisons in crater detection. D-LinkNet is a semantic segmentation neural network that combines the encoder–decoder structure, dilated convolution, and a pre-trained encoder to carry out road extraction tasks.
- SwiftNet [24]: To verify the inference speed of the proposed model, we added SwiftNet to compare the network models. SwiftNet is a real-time semantic segmentation method based on residual network frameworks, which can achieve real-time detection for road-driving images.

3. Results

3.1. Ablation Study

The ablation study on the AFNet explored the influences of different network structures and loss functions on the crater recognition accuracy. The proposed modules and three loss functions (LFs), CE, FL, and the proposed loss function CESL, were compared in the ablation study. The comparison network was initialized by using VGG-16 pre-training weights and normal initialization, where the ✓ denotes the use of the module, and VGG-16 denotes the basic network structure to give a better comparison. The results of the ablation study were obtained by evaluating PA, MPA, MIoU, and FWIoU in the crater validation data. The results are shown in Table 1, and the values in bold are the best values in each compared column.

In Table 1, we can see that the VGG-16-ECA increased by 0.1 and 0.2 MIoU in the CE and FL loss functions, and the MIoU increased by 0.1 and 0.2 MIoU compared with VGG-16 in VGG-16-MFF. When adding the attention machine module VGG-16-ECA and the efficient multiscale feature fusion module MFF, the MIoU obtained values of 73.0%, 74.4%, and 75.3% for the CE, FL, and CESL loss functions in AFNet. The AFNet network under the CESL achieved the best performance of 96.8%, 82.8%, 75.2%, and 94.3% for PA, MPA, MIoU, and FWIoU, respectively. The CESL considers crater data imbalance in classification and distributions and can balance the importance of positive and negative examples by adaptively setting the loss function weights. The proposed CESL loss function obtained a better performance in the compared network structures relative to CE and FL.

We also show several feature maps of a crater image sample at *decoder4* with the VGG-16, VGG-16-ECA, and AFNet network structures in Figure 8. We found that the output features had a clear distinction in AFNet and VGG-16-ECA compared with VGG-16. Some chance information was strengthened, while other chance information was weakened. AFNet and VGG-16-ECA included the attention mechanism ECA, which strengthens some important features to quickly distinguish the edges of craters from their backgrounds.

Table 1. Ablation experiment of the proposed modules on the DEM data.

Network Structures	LFs	ECA	MFF	PA (%)	MPA (%)	MIoU (%)	FWIoU (%)
VGG-16	CE			96.3%	80.0%	72.1%	93.5%
VGG-16-ECA	CE	✓		96.4%	80.6%	72.9%	93.7%
VGG-16-ECA	FL	✓		96.6%	81.6%	73.9%	94.0%
VGG-16-MFF	CE		✓	96.5%	81.1%	73.5%	93.9%
VGG-16-MFF	FL		✓	96.6%	81.8%	74.1%	94.0%
VGG-16-ECA-MFF (AFNet)	CE	✓	✓	96.5%	80.7%	73.0%	93.8%
VGG-16-ECA-MFF (AFNet)	FL	✓	✓	96.7%	82.0%	74.4%	94.1%
VGG-16-ECA-MFF (AFNet)	CESL	✓	✓	**96.8%**	**82.8%**	**75.2%**	**94.3%**

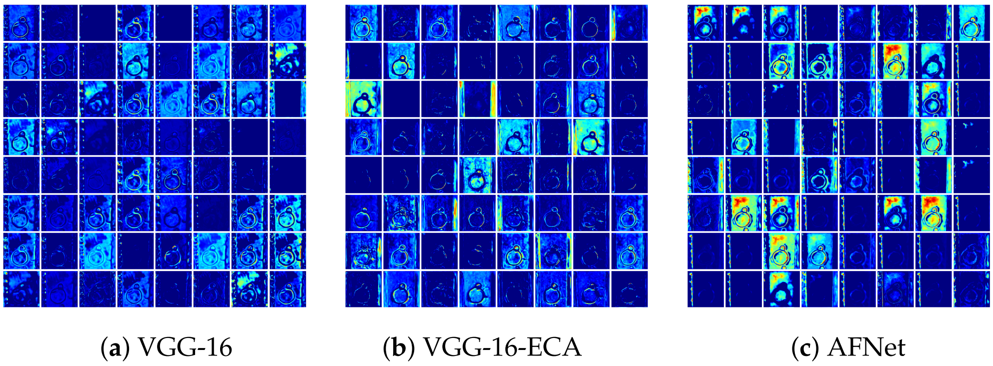
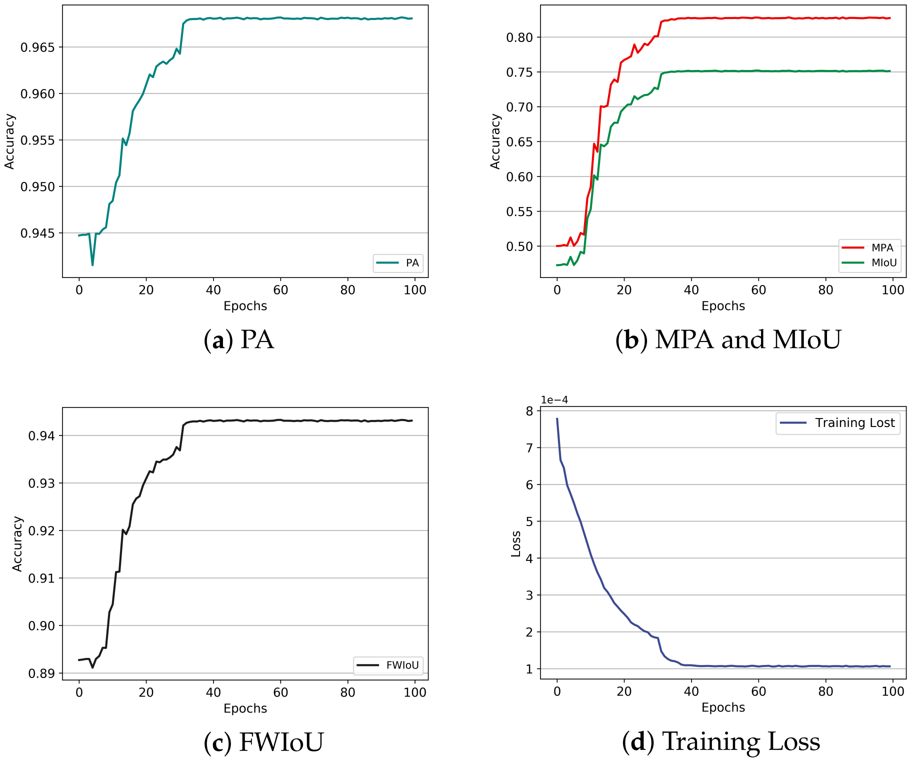

(**a**) VGG-16 (**b**) VGG-16-ECA (**c**) AFNet

Figure 8. Comparison of partial output results of different network structures in decoder4. (**a**) Partial output feature maps with the basic model VGG-16, (**b**) partial output feature maps with VGG-16-ECA, (**c**) partial output feature maps with AFNet.

3.2. *The Evaluation Results for AFNet*

In the iterative process of network training, the values of PA, MPA, MIoU, and FWIoU for AFNet in the validation set are shown in Figure 9. The accuracy of all evaluation criteria increased with the epoch. When the network was in about epoch 35 of network training, the network began to converge. The proposed model achieved a pixel accuracy of 96.8%, as shown in Figure 9a; the mean pixel accuracy was 82.8%, and the MIoU was 75.2%, as shown in Figure 9b. The FWIoU was 94.3%, as shown in Figure 9c. The training loss of the AFNet is shown in Figure 9d. We can see that the initial loss function was very small under the VGG-16 pre-training weight initialization, and the network had a faster convergence speed to allow it to obtain the best performance.



(**a**) PA (**b**) MPA and MIoU

(**c**) FWIoU (**d**) Training Loss

Figure 9. Semantic segmentation results on the validation set and training set of DEM data, (**a**–**c**) show the results of the validation set and c denotes the results of the training set. (**a**) is PA, (**b**) represents the MAP and MIoU, (**c**) denotes the FWIoU, and (**d**) is the training loss of the training set.

The network prediction results with AFNet are shown in Figure 10. The top figure denotes the ground truth of the DEM images, and the bottom figure shows the edge segmentation results. In lunar catalogs, some crater labeling is incomplete with small and shallow craters missing, and some obvious craters are not labeled, which may affect the crater detection accuracy. However, AFNet was used to recognize the crater edges through the classification of each pixel. We can see that the proposed AFNet network was able to segment crater edges with different characteristics.

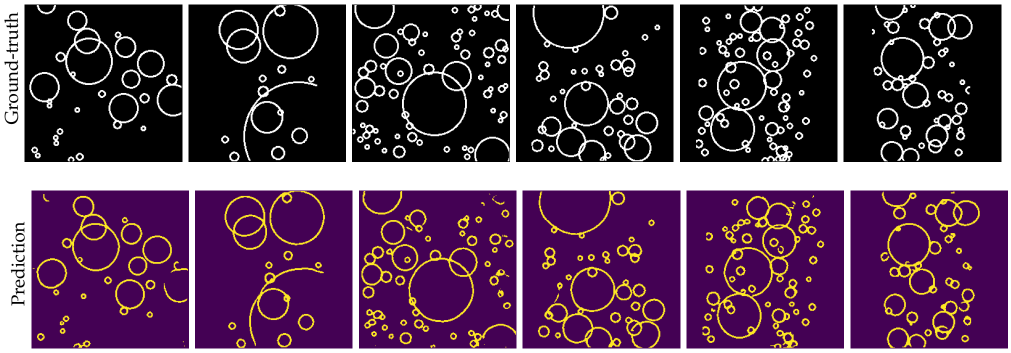


Figure 10. Crater edge segmentation prediction results based on the AFNet for the DEM data. The figure shows the ground truth of the DEM images, while the bottom figure denotes the crater edge segmentation prediction results.

3.3. The Evaluation Results for the ELCD

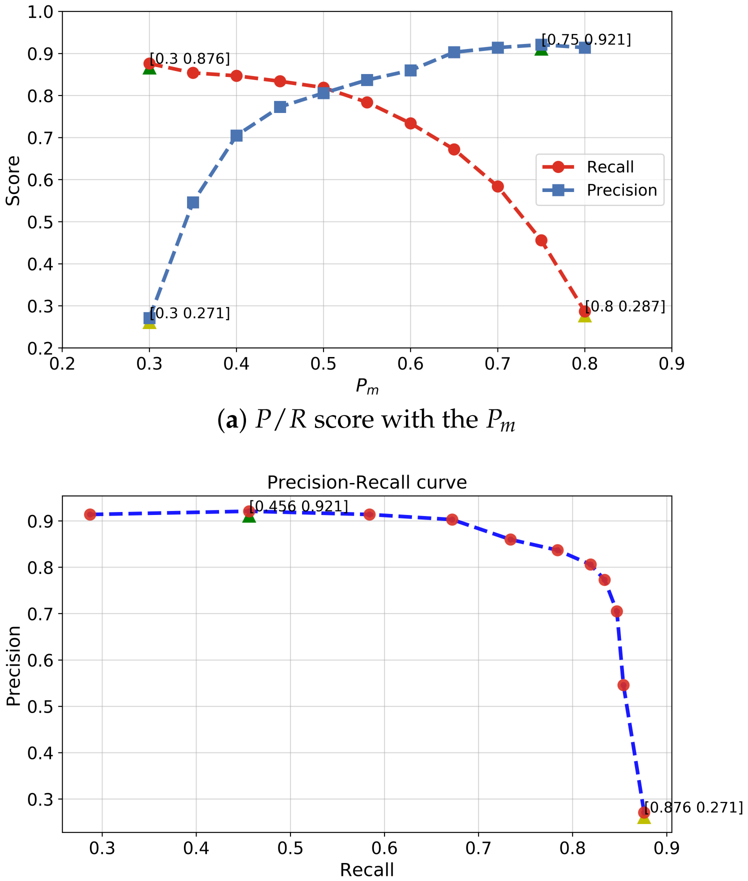
We evaluated the performance of the ELCD based on the edge segmentation network and crater extraction algorithm by detecting the crater radius, latitude, and longitude. Moreover, we computed the precision, recall, F_1, F_2, and the errors in the latitude, longitude, and radius of the crater for the match template method. In order to compare with other crater methods, we calculated the detection results of craters with a radius of 5–40 km. The error threshold of the longitude and latitude $D_{x,y}$ was set to 1.8, the radius error threshold D_r was set to 0.1, and the binary threshold β was set to 0.1. We tuned the match threshold P_m of the match template. For further details about the parameter setting process, refer to [35]. We evaluated the various metrics when the parameter of the match template threshold P_m ranged from 0.3 to 0.8 with an interval of 0.05. The average crater edge extraction resulted in different match threshold values P_m, as shown in Table 2. The best value in each compared row is presented in bold, and the gray column indicates the best tuning parameters.

In Table 2, we can see that the values of precision, F_1, and F_2 increased as P_m increased, while the values of recall and other metrics decreased as P_m increased. A high precision rate of 92.1% was obtained when P_m was 0.75 and the error values of E^{lo}, E^{la}, and E_r were also minimal. When γ was set to 0.3, the value of recall was maximal and more new craters were obtained under the maximum error values of E^{lo}, E^{la}, and E_r. New craters accounted for 41.9% and 70.2%, as shown by R^1_{new} and R^2_{new}. F_1 can balance the value of precision and recall. The best F_1 was 79.4% when P_m was set to 0.5, where the precision was 80.6%, the recall was 81.9%, and the error values of E^{lo}, E^{la}, and E^r were relatively small, at 12.0%, 9.8%, and 6.6%, respectively. F_2 pays more attention to the recall evaluation. When P_m was 0.45, F_2 obtained the best value of 80.9%. In this paper, in accordance with [35,36], we used F_1 and F_2 to evaluate the ELCD algorithm.

The precision and recall curves of the ELCD algorithm are shown in Figure 11, where the upper green triangle represents the maximal point, and the yellow triangle denotes the minimal value point. Figure 11a is the score of precision and recall with the different match thresholds P_m. The focus of these two lines is that P_m is about equal to 0.5, which is a balance point between precision and recall. The relation curve of the precision and recall curves is shown in Figure 11b.

Table 2. Crater edge extraction results of test sets in terms of various match thresholds P_m.

Metrics	**0.3**	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**
Precision	27.1%	54.6%	70.5%	77.3%	80.6%	83.7%	86.0%	90.3%	91.4%	**92.1%**	91.4%
Recall	**87.6%**	85.4%	84.7%	83.4%	81.9%	78.4%	73.4%	67.2%	58.4%	45.6%	28.7%
F_1	40.0%	64.7%	74.9%	78.3%	**79.4%**	79.0%	77.2%	74.9%	69.2%	58.5%	41.6%
F_2	58.4%	74.9%	80.0%	**80.9%**	80.6%	78.3%	74.6%	69.9%	62.0%	49.8%	32.6%
R^1_{new}	**41.9%**	29.9%	21.4%	17.1%	14.9%	12.7%	11.1%	8.1%	7.2%	6.6%	6.9%
R^2_{new}	**70.2%**	41.6%	26.4%	19.9%	16.7%	13.5%	11.0%	7.2%	5.5%	4.0%	2.9%
E^{lo}	17.5%	13.9%	12.7%	11.3%	12.0%	10.7%	9.3%	9.4%	8.8%	9.6%	**8.2%**
E^{la}	17.0%	13.7%	11.3%	10.6%	9.8%	9.2%	8.3%	7.7%	7.4%	7.0%	**6.8%**
E^{r}	13.0%	9.2%	8.0%	7.3%	6.6%	5.7%	4.8%	4.6%	4.2%	4.1%	**3.7%**



(**a**) P/R score with the P_m

(**b**) $P-R$ curve

Figure 11. Precision/recall curve for the crater detection results.

3.4. Comparison of Multiple Crater Detection Methods

In this section, we present an evaluation of the comparison results with ELCD under different crater detection methods using the test set. $P_m = 0.5$ is balance point between precision and recall. As shown in Figure 11a, we used the result where P_m was 0.5 as a comparison of ELCD. We also measured the computation complexity with different network architectures. In this paper, the billions of floating-point operations (FLOPs), network parameters (Params), and the number of processed frames per second (FPS) were used to evaluate the computational complexity of the trained networks. In the FPS computation, in accordance with [24], we set the test batch size as 1.

The average crater extraction results under various crater detection algorithms are shown in Table 3. In Table 3, we can see that the DeepMoon increased the recall and the proportion of newly discovered craters, and ERU-Net obtained a low detection error for the crater radius, respectively. SwiftNet and D-linkNet had relatively poor detection accuracy levels, but they had the lowest FLOPs and network parameters. The crater detection algorithm required not only a high detection accuracy due to autonomous landing requirements for deep space probes in the deep space environment, but the crater detection algorithm should have a fast detection speed. The SwiftNet and D-linkNet network structures were designed for the real-time target detection of road-driving images. They have fewer parameters, low FLOPs, and high FPS during the running of the network to meet the needs of real-time detection. However, as the SwiftNet and D-linkNet network structures are simple network structures, they are inefficient for complex crater detection problems, and they perform poorly in lunar crater detection compared with other networks such as DeepMoon, ERU-Net, and the proposed algorithm. DeepMoon and ERU-Net achieved good crater detection results compared with the SwiftNet and D-linkNet network structures, but they require more computational resources, and the network computation speed of FPS is also lower.

Table 3. Comparison of the detection results of test sets under various crater detection algorithms.

Algorithms	P	R	F_1	F_2	R^1_{new}	R^2_{new}	E^{lo}	E^{la}	E^r	FLOPs (G)	Params (M)	FPS (HZ)
DeepMoon [35]	56.0%	**92.0%**	66.2%	72.9%	**40.0%**	**42.0%**	14.0%	11.0%	8.0%	74.3	10.28	8.7
ERU-Net [36]	75.4%	81.2%	78.1%	78.5%	18.3%	21.5%	**9.9%**	10.0%	7.8%	183.3	23.7	4.3
D-LinkNet [23]	77.2%	68.3%	61.2%	55.1%	17.3%	17.1%	10.1%	10.0%	7.3%	6.0	21.0	46.4
SwiftNet [24]	77.1%	52.6%	61.4%	56.1%	17.0%	13.3%	22.9%	19.9%	13.2%	**3.2**	**11.8**	60.2
ELCD (**our**)	**80.6%**	81.9%	**79.4%**	**80.6%**	14.9%	16.7%	12.0%	**9.8%**	**6.6%**	43.7	21.8	**73.2**

In Table 3, the proposed algorithm is shown to achieve better crater detection precision (P) and F_1, F_2 scores than the DeepMoon, SwiftNet, D-linkNet, and ERU-Net network structures with minimal E^{la} and E^r errors. Moreover, ELCD has a faster inference speed than the other algorithms. The proposed model combines the encoder, feature fusion, and decoder processes to achieve good network parallelism to speed up the network inference speed. The proposed ELCD has lower FLOPs than the DeepMoon and ERU-Net methods, and the total FLOPs in ELCD were shown to be about 1.7 times and 4.1 times lower than the values of DeepMoon and ERU-Net, respectively. For the FPS measure, although the parameters of the ELCD were not lower than those of DeepMoon and ERU-Net, the total FPS of the ELCD was about 8 times and 17 times higher than the values of DeepMoon and ERU-Net, respectively. Thus, the proposed ELCD algorithm achieved the best crater detection results with relatively few parameters and a low network complexity. It can achieve a balance between crater detection precision and network computation efficiency.

A comparison of the results obtained with different crater detection methods is shown in Figure 12. Each row represents the detection result of all compared crater methods for the same types of crater data. Each column represents the performance of the same detection method in different types of craters with varying degrees of classification and distribution imbalance. IR is the classification imbalance ratio, and DR denotes the distribution imbalance ratio, which was computed by the number of craters in each image label. The details are presented in Section 2.4. The greater the DR is, the denser the crater images are, and relatively speaking, the smaller the IR is. The original DEM image shown in Figure 12a,b is the ground truth, and Figure 12a–g denotes the compared algorithms. The blue circle denotes the correctly detected craters, the green circle is the newly detected craters, and the red circle is unrecognized craters. We can see that D-LinkNet and SwiftNet performed poorly for crater detection, especially for dense crater data. There are many incorrectly detected craters marked as red circles in Figure 12f,g. DeepMoon and ERU-Net could detect most of the labeled craters in contrast to D-LinkNet and SwiftNet, but they performed poorly for large craters. For example, in IR = 9.3, DR = 49 and IR = 7.5, DR = 49, DeepMoon could not detect the large crater that is represented by the red circle in Figure 12d,e. In the third column, we can see that the proposed model increased the accuracy of crater detection compared with the other models for craters of different densities and sizes, as shown in Figure 12c. Moreover, the proposed model was able to detect some new unlabeled craters. However, small craters with a high degree of overlap in the DEM data were difficult to identify with high precision using DEM data for all compared algorithms. The proposed model regarded such craters as noise and could not detect them well.

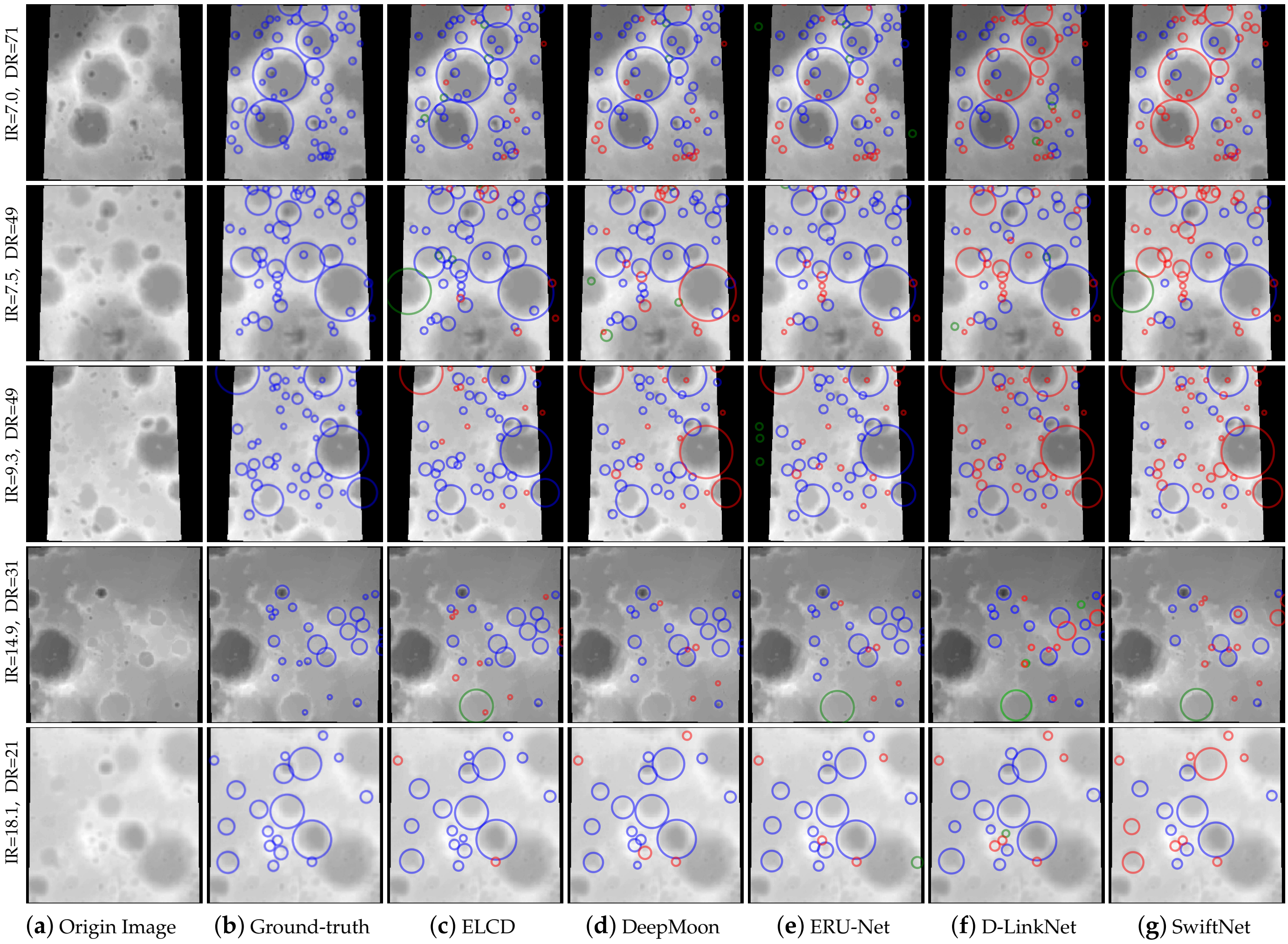


Figure 12. Comparison of the results of test sets using different crater detection methods for DEM data. (**a**) The original lunar DEM images in the test set. (**b**) The ground-truth DEM image. (**c**) The detection results obtained with DeepMoon based on U-net [35]. (**d**) The recognition results obtained with the ERU-Net network [36]. (**e**) The detection results obtained with D-LinkNet with the ResNet-18 network [23]. (**f**) The detection results obtained with SwiftNet (**g**), designed by the paper [24]. In the figure, the blue circles represent correctly recognized crates, the green circles denote new craters discovered by compared methods, and the red circle indicates unrecognized craters.

4. Discussion

With the application of deep learning techniques, great progress has been made in automated impact crater detection. The proposed method builds an efficient crater edge prediction network with a lightweight attention mechanism module and a multiscale feature fusion module to recognize crater edges from digital elevation models. The experimental results show that the presented method achieves high precision and recall rates and a fast detection speed when undergoing lunar crater detection, mainly due to the following reasons: (1) we used the digital elevation model as the crater data, which contain abundant 3D morphology and topography morphological characteristics and are insensitive to light; (2) the proposed crater edge segmentation network is an efficient model to improve the accuracy of crater detection. The proposed network uses a lightweight attention mechanism module to enhance the feature extraction capability of the network encoder and designs a multiscale feature fusion module that fuses multi-level different resolution feature maps to reduce information loss in the network encoder; and (3) considering the imbalance of classification and different density distributions of craters, we proposed an efficient crater edge segmentation loss function to optimize the network performance.

In the experimental results, Table 1 shows that the multiscale feature fusion module can increase the crater detection accuracy, and it shows that the proposed crater loss function can achieve the best crater edge segmentation results. Figure 8 shows that the attention mechanism module can strengthen some chance information about craters and weaken other chance information, which can strengthen the importance of crater features to allow the edges of craters to be quickly distinguished from their backgrounds. Figure 9 shows that the CESL can improve the ability of the network to obtain optimal solutions and can speed up the convergence of the improved model. The final crater detection results show that the proposed model, which includes the attention mechanism module and the multiscale feature fusion module, can achieve more fine-grained segmentation for crater edges with different characteristics, as shown in Figure 10. In Table 3 and Figure 12, which shows a comparison of the different crater detection methods, the proposed model is shown to achieve the best detection performance with minimal errors in E^{la} and E^{r}. Compared with other real-time target detection methods, this method has a faster reasoning speed. Compared with the survey of the global lunar orbiter laser altimeter (LOLA) dataset of the Moon, the algorithm can detect the marked craters on the lunar surface more accurately and can detect some undiscovered craters. There are some false and ambiguous markers in the global LOLA dataset, and the proposed algorithm can correct false positives in the original data. Moreover, the newly discovered craters can increase the size of the original data set.

The discovery of impact craters is important for studying the evolution of the Moon. There are many small craters on the Moon's surface, and they influence the estimation of the Moon's age. However, the study still has some limitations with regard to small crater detection. Most crater digital elevation models have a lower resolution than the optical image and other higher-resolution images. Some craters that are too small appear as points in DEM images, and they are likely to be ignored or considered to be noise and thus cannot be detected successfully using a digital elevation model. The optical image has a high resolution, but it is sensitive to illumination. Thus, determining how to avoid the impact of light on impact craters in optical images or fusing the optical image and the digital elevation model to improve the small crater detection accuracy deserve further attention in the future.

5. Conclusions

In this paper, an efficient lunar crater detection algorithm, AFNet, based on the segmentation convolutional neural network was proposed to improve the crater detection accuracy and speed. Based on the VGG-16 network architecture, a lightweight attention mechanism module was introduced to enhance the extraction of important crater features in the network encoder. The proposed model uses a new feature fusion method that fuses multi-level different feature maps obtained from the network encoder to reduce the information loss of the output map in the network decoder. Then, considering the classification and distribution imbalance of the crater data, the crater edge segmentation loss function was used to improve the optimization performance of the proposed model. Last, the crater positions were extracted by the crater edge extract algorithm based on the match template method. The proposed model was applied to two crater catalogs and compared with four state-of-the-art crater detection algorithms. The results demonstrate that the ELCD achieved an inference speed of about 73 HZ and a precision of 80.6% for lunar crater detection in a DEM image with 256 × 256 pixels on *GeForce RTX 3060*, and it obtained the best accuracy of 79.4% for F_1 and 80.6% for F_2 compared with the other crater detection models. Moreover, the ELCD can be used to discover new craters and expand the size of the original data set. It is hoped that this algorithm will further improve the accuracy of lunar age estimation and the positioning accuracy of spacecraft landing. For future work, the network structure should be further optimized so that the model can improve its real-time detection speed and achieve a high crater detection accuracy in the detection of impact craters of different sizes.

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

Funding: This work was funded by the National Natural Science Foundation of China (Grant No. 62076127).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank R. Povilaitis and J. Head for providing the 5–20 km and >20 km dataset of lunar craters. In addition, the authors would like to thank the reviewers for their valuable comments and suggestions.

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

Abbreviations

The following abbreviations are used in this manuscript:

ELCD	Efficient lunar crater detection
DEM	Digital elevation model
CNNs	Convolution neural networks
SAR	Synthetic aperture radar
AFNet	Attention mechanisms and multiscale feature fusion networks
CESL	Crater edge segmentation loss
CEA	Crater extraction algorithm
MFF	Multiscale feature fusion
VGG-16	Visual geometry group-16
ECA	Efficient channel attention
IR	Data imbalance ratio
DR	Distribution imbalance ratio
CE	Cross-entropy
FL	Focal loss
LRO	Lunar reconnaissance orbiter
PA	Pixel accuracy
MPA	Mean pixel accuracy
MIoU	Mean intersection over union
FWIoU	Frequency weighted intersection over union
FLOPs	Floating-point operations
FPS	Frames per second
P	Precision
R	Recall

References

1. Strom, R.; Malhotra, R.; Ito, T.; Yoshida, F.; Kring, D. Origin of Impacting Objects in the Inner Solar System. In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 5–9 December 2005; Volume 2005, p. P42A-01.
2. Schmidt, M.W.; Kraettli, G. Experimental Crystallization of the Lunar Magma Ocean, Initial Selenotherm and Density Stratification, and Implications for Crust Formation, Overturn and the Bulk Silicate Moon Composition. *J. Geophys. Res. (Planets)* **2022**, *127*, e07187. [CrossRef]
3. Bottke, W.F.; Norman, M.D. The late heavy bombardment. *Annu. Rev. Earth Planet. Sci.* **2017**, *45*, 619–647. [CrossRef]
4. Kereszturi, A.; Steinmann, V. Terra-mare comparison of small young craters on the Moon. *Icarus* **2019**, *322*, 54–68. [CrossRef]
5. Xu, L.; Qiao, L.; Xie, M.; Wu, Y. Formation age of lunar Lalande crater and its implications for the source region of the KREEP-rich meteorite Sayh al Uhaymir 169. *Icarus* **2022**, *386*, 115166. [CrossRef]
6. Martin, I.; Parkes, S.; Dunstan, M. Modeling cratered surfaces with real and synthetic terrain for testing planetary landers. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 2916–2928. [CrossRef]
7. Song, J.; Rondao, D.; Aouf, N. Deep learning-based spacecraft relative navigation methods: A survey. *Acta Astronaut.* **2022**, *191*, 22–40. [CrossRef]
8. Kereszturi, A.; Tomka, R.; Steinmann, V. Testing statistical impact crater analysis in permanently shadowed lunar polar regions. *Icarus* **2022**, *376*, 114879. [CrossRef]
9. Povilaitis, R.; Robinson, M.; Van der Bogert, C.; Hiesinger, H.; Meyer, H.; Ostrach, L. Crater density differences: Exploring regional resurfacing, secondary crater populations, and crater saturation equilibrium on the moon. *Planet. Space Sci.* **2018**, *162*, 41–51. [CrossRef]
10. Head III, J.W.; Fassett, C.I.; Kadish, S.J.; Smith, D.E.; Zuber, M.T.; Neumann, G.A.; Mazarico, E. Global distribution of large lunar craters: Implications for resurfacing and impactor populations. *Science* **2010**, *329*, 1504–1507. [CrossRef]
11. Troglio, G.; Le Moigne, J.; Benediktsson, J.A.; Moser, G.; Serpico, S.B. Automatic extraction of ellipsoidal features for planetary image registration. *IEEE Geosci. Remote Sens. Lett.* **2011**, *9*, 95–99. [CrossRef]
12. Galloway, M.J.; Benedix, G.K.; Bland, P.A.; Paxman, J.; Towner, M.C.; Tan, T. Automated crater detection and counting using the Hough transform. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1579–1583.
13. Di, K.; Li, W.; Yue, Z.; Sun, Y.; Liu, Y. A machine learning approach to crater detection from topographic data. *Adv. Space Res.* **2014**, *54*, 2419–2429. [CrossRef]
14. Bandeira, L.; Saraiva, J.; Pina, P. Impact crater recognition on Mars based on a probability volume created by template matching. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 4008–4015. [CrossRef]
15. Emami, E.; Bebis, G.; Nefian, A.; Fong, T. Automatic crater detection using convex grouping and convolutional neural networks. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 14–16 December 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 213–224.
16. He, J.; Cui, H.; Feng, J. Edge information based crater detection and matching for lunar exploration. In Proceedings of the 2010 International Conference on Intelligent Control and Information Processing, Dalian, China, 13–15 August 2010; IEEE: Hoboken, NJ, USA, 2010; pp. 302–307.
17. Chen, M.; Liu, D.; Qian, K.; Li, J.; Lei, M.; Zhou, Y. Lunar crater detection based on terrain analysis and mathematical morphology methods using digital elevation models. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3681–3692. [CrossRef]
18. Barata, T.; Alves, E.I.; Saraiva, J.; Pina, P. Automatic recognition of impact craters on the surface of Mars. In Proceedings of the International Conference Image Analysis and Recognition, Porto, Portugal, 29 September–1 October 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 489–496.
19. Urbach, E.R.; Stepinski, T.F. Automatic detection of sub-km craters in high resolution planetary images. *Planet. Space Sci.* **2009**, *57*, 880–887. [CrossRef]
20. Salamunićcar, G.; Lončarić, S. Application of machine learning using support vector machines for crater detection from Martian digital topography data. *38th COSPAR Sci. Assem.* **2010**, *38*, 3.
21. Kang, Z.; Wang, X.; Hu, T.; Yang, J. Coarse-to-fine extraction of small-scale lunar impact craters from the CCD images of the Chang'E lunar orbiters. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 181–193. [CrossRef]
22. Wang, Y.; Wu, B. Active machine learning approach for crater detection from planetary imagery and digital elevation models. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5777–5789. [CrossRef]
23. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 182–186.
24. Orsic, M.; Kreso, I.; Bevandic, P.; Segvic, S. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 12607–12616.
25. Li, Y.; Du, L.; Wei, D. Multiscale CNN based on component analysis for SAR ATR. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–12. [CrossRef]
26. Liao, L.; Du, L.; Guo, Y. Semi-supervised SAR target detection based on an improved faster R-CNN. *Remote Sens.* **2021**, *14*, 143. [CrossRef]

27. DeLatte, D.; Crites, S.T.; Guttenberg, N.; Yairi, T. Automated crater detection algorithms from a machine learning perspective in the convolutional neural network era. *Adv. Space Res.* **2019**, *64*, 1615–1628. [CrossRef]
28. Downes, L.M.; Steiner, T.J.; How, J.P. Neural Network Approach to Crater Detection for Lunar Terrain Relative Navigation. *J. Aerosp. Inf. Syst.* **2021**, *18*, 391–403. [CrossRef]
29. Silvestrini, S.; Piccinin, M.; Zanotti, G.; Brandonisio, A.; Bloise, I.; Feruglio, L.; Lunghi, P.; Lavagna, M.; Varile, M. Optical navigation for Lunar landing based on Convolutional Neural Network crater detector. *Aerosp. Sci. Technol.* **2022**, *123*, 107503. [CrossRef]
30. Zhang, Z.; Liu, Q.; Wang, Y. Road extraction by deep residual u-net. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753. [CrossRef]
31. Li, W.; Zhou, B.; Hsu, C.Y.; Li, Y.; Ren, F. Recognizing terrain features on terrestrial surface using a deep learning model: An example with crater detection. In Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery, Los Angeles, CA, USA, 7 November 2017; pp. 33–36.
32. Tewari, A.; Verma, V.; Srivastava, P.; Jain, V.; Khanna, N. Automated crater detection from co-registered optical images, elevation maps and slope maps using deep learning. *Planet. Space Sci.* **2022**, *218*, 105500. [CrossRef]
33. Yang, S.; Cai, Z. High-Resolution Feature Pyramid Network for Automatic Crater Detection on Mars. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–12. [CrossRef]
34. Wang, H.; Jiang, J.; Zhang, G. CraterIDNet: An end-to-end fully convolutional neural network for crater detection and identification in remotely sensed planetary images. *Remote Sens.* **2018**, *10*, 1067. [CrossRef]
35. Silburt, A.; Ali-Dib, M.; Zhu, C.; Jackson, A.; Valencia, D.; Kissin, Y.; Tamayo, D.; Menou, K. Lunar crater identification via deep learning. *Icarus* **2019**, *317*, 27–38. [CrossRef]
36. Wang, S.; Fan, Z.; Li, Z.; Zhang, H.; Wei, C. An effective lunar crater recognition algorithm based on convolutional neural network. *Remote Sens.* **2020**, *12*, 2694. [CrossRef]
37. DeLatte, D.M.; Crites, S.T.; Guttenberg, N.; Tasker, E.J.; Yairi, T. Segmentation convolutional neural networks for automatic crater detection on mars. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2944–2957. [CrossRef]
38. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
39. Lee, C.; Hogan, J. Automated crater detection with human level performance. *Comput. Geosci.* **2021**, *147*, 104645. [CrossRef]
40. Hu, Z.; Shi, T.; Cen, M.; Wang, J.; Zhao, X.; Zeng, C.; Zhou, Y.; Fan, Y.; Liu, Y.; Zhao, Z. Research progress on lunar and Martian concrete. *Constr. Build. Mater.* **2022**, *343*, 128117. [CrossRef]
41. Ketkar, N.; Moolayil, J. Introduction to pytorch. In *Deep Learning with Python*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 27–91.
42. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
43. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
44. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
45. Van Hulse, J.; Khoshgoftaar, T.M.; Napolitano, A. Experimental perspectives on learning from imbalanced data. In Proceedings of the 24th International Conference on Machine Learning, Corvalis, OR, USA, 20–24 June 2007; pp. 935–942.
46. Bue, B.D.; Stepinski, T.F. Machine detection of Martian impact craters from digital topography data. *IEEE Trans. Geosci. Remote Sens.* **2006**, *45*, 265–274. [CrossRef]
47. Van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager, N.; Gouillart, E.; Yu, T. Scikit-image: Image processing in Python. *PeerJ* **2014**, *2*, e453. [CrossRef]
48. Yong, H.; Huang, J.; Hua, X.; Zhang, L. Gradient centralization: A new optimization technique for deep neural networks. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 635–652.
49. Barker, M.; Mazarico, E.; Neumann, G.; Zuber, M.; Haruyama, J.; Smith, D. A new lunar digital elevation model from the Lunar Orbiter Laser Altimeter and SELENE Terrain Camera. *Icarus* **2016**, *273*, 346–355. [CrossRef]