A combined method of crater detection and recognition based on deep learning

Haibo Li, Bei Jiang, Yuyuan Li and Le Cao

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, People’s Republic of China

ABSTRACT
The crater is one of the main obstacles that need to be avoided when Mars probe lands. In order to further improve the accuracy of crater detection, this paper proposes a combined detection method based on deep learning. Firstly, the random structured forest is trained offline to detect the edge information of craters. Secondly, according to the detected edge information of the crater, the candidate areas of the crater are determined with the morphological method. For the identified candidate areas of the crater, Alexnet network trained by transfer learning was used to identify crater areas. Compared with other methods, the proposed method has relatively good effect.

ARTICLE HISTORY
Received 3 August 2020
Accepted 16 November 2020

KEYWORDS
Crater detection; crater recognition; deep learning

1. Introductions
The craters distributed on the surface of Mars are main obstacles for the spacecraft to avoid when landing. In order to avoid the probe falling into the crater during landing, effective methods are needed to detect and identify the crater areas during the landing process. For craters with a diameter of 1–512 km, they can be manually marked by images taken by orbiters (Wang et al., 2015). For the crater whose diameter is < 1 km, it is necessary for the detector to detect and evade independently when landing. Generally, optical image has a higher resolution than digital elevation model data. There are three kinds of crater detection methods based on optical image: unsupervised detection method, supervised detection method and combined detection method.

The unsupervised detection method determines the edges of the crater through the circular or elliptical features in the image, and detects craters using the related theories of image processing and target detection. The main methods include: Hough transform and its improved algorithm (Emami et al., 2019), genetic algorithm (Hong et al., 2012), radial consistency method (Earl et al., 2005) and template matching method (Cadogan, 2020; Lee et al., 2020). For example, one crater detection method based on morphological image processing method was proposed by Pedrosa et al. (2017). This method mainly includes three steps: firstly, the candidate regions were determined by the morphological method, and then the possible crater areas were found by removing noise; secondly, the template matching method based on fast Fourier transform was used to establish the association between candidate regions and templates; finally, probability analysis was carried out to determine crater areas. The advantage of unsupervised method is that it does not need to label a large number of samples to learn the accurate classifier. This kind of method can be used when the computing power of autonomous navigation system is limited. However, it is not robust in complex terrain detection.

The supervised detection methods use the marked training data set in knowledge domain and machine learning methods to construct classifiers, such as neural network (Li & Hsu, 2020); support vector machine (Kang et al., 2019); AdaBoost method (Xin et al., 2017). These methods mainly focus on the classification part of crater detection, and do not take the confirmation of candidate crater areas into account. Urbach and Stepinski (2009) proposed the idea of searching candidate regions of the crater. The candidate regions of the crater are the regions that may contain craters in an image, which can be regarded as the combination of a pair of crescent shaped highlight regions and shadow parts. In the work of Urbach and Stepinski, a small feature set was used to describe the highlight shape and shadow areas of the candidate craters. When there are non-crater landforms with similar shapes in the detection area, this method is not suitable for detection. In reference (Ding et al., 2011), an integrated framework for crater detection based on boosting and transfer learning was proposed. In this framework, they use the idea of searching candidate regions of the crater proposed by Urbach and Stepinski to extract image texture features from the candidate areas of the crater.
Then, several supervised learning algorithms were used to select a group of features to confirm whether the candidate region is a crater or not.

The combined detection method uses a variety of methods to detect craters, including supervised detection method and unsupervised detection method. For example, a combined detection method proposed by Yan et al. (2019), KLT detector was used to extract candidate regions of craters. In this method, image blocks were used as the input of the supervised detection method, and the detection accuracy was greatly affected by the parameter $\lambda$ of the KLT detector. In Li and Hsu’s (2020) work, a method combining template matching and neural network was used in the process of crater detection. The problem of this method is that the recognition effect of craters in rugged area was not good, because the recognition method based on neural network cannot remove a large number of false alarms. In addition, there are a lot of research on crater detection using combination methods, such as Sawabe et al. (2006).

In order to further improve the detection accuracy, this paper adopts the method of combination detection. In the framework of structure learning, the method of structural random forest was used to detect the edges of craters, and then the candidate areas of craters were determined by the morphological method. For the identification of candidate areas of craters, the recognition method based on deep learning Alexnet model was studied to confirm crater areas.

2. Edges detection of craters

Craters usually appear in the shape of a circle or an ellipse in images. For the detection of craters, the edge structure features can be fully used to determine the candidate regions. Therefore, edge detection is an important part of crater detection. Traditional edge detection methods are mostly based on colour gradient information (Canny, 1986; Freeman & Adelson, 1991; Ziou & Tabbone, 1998), but many edges do not correspond to colour gradient. For example, texture edge (Martin et al., 2004) and illusion contour (Ren et al., 2005). In this paper, the structure learning method was used to detect the edges of craters, and the problem of edges detection was transformed into predicting the local segmentation mask and corresponding edge map of a given input image patch. The overall scheme flow is shown in Figure 1.

It can be seen from Figure 1 that the image containing craters was firstly divided into image patches, then the image patches were classified and recognized by trained structured random forest, and finally, the edge information of craters was output.

2.1. Structured random forest

Structured random forest is an extension of random decision forest (Dollár & Zitnick, 2013; Geurts et al., 2006; Speybroeck, 2012). The method proposed by Dollár and Zitnick (2013) can only classifies labels, while structured random forest can classify 2D image patches. First, we review random decision forest. Decision tree $f(x)$ divides samples by recursive left and right branches $x \in X$. Each node $j$ in the tree corresponds to a binary partition function related to the parameter $\theta_j$

$$h(x, \theta_j) \in \{0, 1\}$$

Figure 1. Overall scheme for the edge detection of the crater.
If \( h(x, \theta_j) = 0 \), the sample \( x \) belongs to the left, otherwise it belongs to the right. This process continues to reach the leaf node. Corresponding to sample \( x \), the output of the tree is the prediction stored at the leaf node that sample \( x \) reaches. It can be a label \( y \in Y \). For example, \( \theta = (k, \tau) \), \( H(x, \theta) = [x (k) < \tau] \), \([ \cdot ] \) is an indicator function. Another common choice is \( \theta = (K_1, K_2, \tau) \), \( H(x, \theta) = [x (K_1) - x (K_2) < \tau] \), and the two methods have been applied in practice (Criminisi et al., 2011).

Decision forest is a set of single decision tree \( f_t(x) \). For a single sample \( x \), the output of all decision trees were given by a set model. The selection of set model depends on the output. The usual choice is voting and averaging. There are more complex set models (Criminisi et al., 2011). Leaf nodes of decision tree can store arbitrary information. For input sample \( x \), the output of all decision trees was given by a set model. The prediction of multiple trees was combined in some way to give the final output.

In structured random forests, \( x \in X \) represents an image patch in the image to be detected, \( y \in Y \) represents the segmentation mask corresponding to the image patch.

### 2.2. Decision tree training

The training of each tree is independent. For the known node \( j \) and the corresponding training set \( S_j \subset X \times Y \). The purpose of training is to determine the parameter \( \theta_j \) in the function \( H(x, \theta_j) \). So, we need to define an information gain standard

\[
I_j = I(S_j, S_j^L, S_j^R)
\]

(2)

where \( S_j^L = \{(x, y) \in S_j | h(x, \theta_j) = 0 \}, S_j^R = S_j \setminus S_j^L \). The selection of \( \theta_j \) is to maximize the information gain \( I_j \) and the data on the left \( S_j^L \) continue to recursively train the nodes on the left and the data on the right \( S_j^R \) continue to recursively train the nodes on the right. The training stop criterion is to reach the maximum depth or information gain, and the training set size reaches the set threshold.

For multi class \( (Y \subset Z) \), training, information gain can be expressed in the following form

\[
I_j = H(S_j) - \sum_{k \in \{L, R\}} (S_j^k / S_j) H(S_j^k)
\]

(3)

where \( H(S) = - \sum_y p_y \log(p_y) \), it denotes Shannon entropy, and \( p_y \) denotes the proportion labelled \( y \) in \( S \).

In structured random forest, in order to calculate the information gain, all labels \( Y \) are mapped to an intermediate space \( Z \) which is easy to measure the distance. Thus, the difference of \( y \in Y \) is transformed into the Euclidean space distance in \( Z \). Specifically, \( y \in Y \) is mapped to a discrete label set \( c \in C \), where \( C = 1, \ldots, k \). Thus, the information gain can be approximately calculated. For example, the crater edge detection label \( y \in Y \) is a 16 × 16 segmentation mask, and \( z = \Pi(y) \) is defined as a long binary vector to determine whether each pair of pixels in \( Y \) belongs to the same segmentation. In space \( Z \), the distance is easy to measure. In the 16 × 16 split mask, the dimension of \( Z \) is \( C_{16}^{2 \times 16} = 32640 \) dimensions. In order to reduce the dimension, random extraction method was used to get \( m \)-dimension data in application, which can not only improve the operation speed, but also ensure the difference of trees. Then, the \( m \)-dimensional data were divided into \( k \) classes by \( K \)-means clustering method, and mapped to the discrete label set \( C \). On this basis, Shannon entropy can be calculated, and then information gain can be calculated by formula (3). In this paper, \( m = 256 \), \( k = 2 \). Because of the generality of edge detection, this paper uses the method of transfer learning to train decision tree on bsds500 dataset (Martin et al., 2001).

### 2.3. Edge detection of crater

For the input image, the first step is to fill the image so that it can be extracted 32 × 32 patch continuously. For each image patch, the multi-channel information were expanded to form a feature vector \( x \in R^{32 \times 32 \times k} \), and \( K \) is the channel. In this paper, 14 channel information is extracted, including 3 colour channels, 3 gradient amplitude channels and 8 directional gradient channels. The calculation of three colour channels was based on CIELUV colour space; the calculation of three gradient amplitude channels was based on Gaussian blur with \( \sigma = 0, 1.5, 5 \). For each gradient amplitude channel with \( \sigma = 0 \) and \( \sigma = 1.5 \), four directional gradient channels were divided based on the direction, and eight directional gradient channels were formed. In this paper, two kinds of features were used: pixel feature \( x(i, j, k) \) and pixel pair difference feature \( x(i_1, j_1, k) - x(i_2, j_2, k) \). By down-sampling the pixel features with factor 2, we get \((32 \times 32 \times 14) / 4 = 3584 \) candidate features \( x(i, j, k) \). The triangle blur of 8 pixel radius is applied to each channel, and then the sampling resolution was reduced to 5 × 5, and the difference feature of pixel pair was calculated to form \( C_{5 \times 5}^{2 \times 300} = 300 \) features, 14 channels, a total of 4200 features. Therefore, each image patch has 3584 + 4200 = 7784 candidate features. For the candidate features, partial sub-set were sent to the trained structured forest classifier, which can predict a 16 × 16 segmentation mask. For an image patch, the corresponding binary edge map can be obtained by dividing the mask (Dollár & Zitnick, 2013). As the decision forest leaf node can store any information, the corresponding binary edge image was stored in the leaf node. For the prediction results of multiple trees, the ensemble model of reference (Dollár & Zitnick, 2013) was
used to merge, and the final edge map was obtained by combining the binary edge graphs to get the average.

3. Determination of candidate areas of crater

After the crater edge map was obtained by the method described in Section 2, candidate areas of the crater can also be obtained by morphological methods, such as randomly selecting a Google Mars surface image, as shown in Figure 2(a). The edge detection method described in Section 2 was used for image Figure 2(a), and the results are shown in Figure 2(b).

For the edge detection result shown in Figure 2, the structure element matrix $B$ was used to expand, where

$$B = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}$$

The results of the expansion are shown in Figure 3(a). The adaptive optimal threshold contrast adjustment was used to make the edge form a connected region. The results are shown in Figure 3(b).

The connected regions in Figure 3(b) were searched according to 8 connectivity. Each connected region was marked, and the small connected areas less than a certain threshold were removed. The boundary box of the marked areas was calculated. Generally, the edge of the crater is the periphery of crater, so the region in the bounding box can be used as the candidate region of the crater. Marking the boundary box at the same position in Figure 2(a), the result is shown in Figure 4.

For the candidate areas of the crater marked by Figure 4, it is necessary to further identify and eliminate the non-crater areas. The related contents are described in Section 4.

![Figure 2](image2.png) Google Mars surface image and edge detection result.

![Figure 3](image3.png) Expanded result and edge connected region.

![Figure 4](image4.png) Candidate areas of the crater.
4. Crater recognition based on deep learning

For the candidate areas of craters determined in Section 3, a method for crater recognition by deep learning of alexnet model (Krizhevsky, 2012) was also studied. The modified linear units (Rectified Linear Units, ReLU) were introduced into the alexnet model as the activation function, which has a faster convergence rate than sigmoid function and tanh function. In addition, the alexnet model has fewer connections and parameters and it is relatively easy to train. The alexnet model has eight layers in total, including five convolution layers and three full connection layers, as shown in Figure 5.

Before identifying the candidate areas of the crater, the image needs to be preprocessed, and the size specification is converted to $227 \times 227 \times 3$. Then, the craters were identified by using alexnet model. The identification process is mainly carried out according to the following steps.

Step 1. Convoluting the image of candidate crater region ($227 \times 227 \times 3$) and the convolution kernel ($11 \times 11 \times 3$), the convolution kernel performs the calculation in the $X$ and $Y$ directions of the input image. Each operation obtains a new pixel with a sliding interval of 4 pixels. After sliding, a $55 \times 55$ pixel layer was formed. Ninety-six convolution kernels were set in the first layer, and every 48 convolution kernels are a group. Therefore, two sets of $55 \times 55 \times 48$ pixel layers are generated. The obtained pixel layer data is input to the correction linear unit, and the output is two groups of $55 \times 55 \times 48$ active pixel layers. The active pixel layer is further pooled. The scale of calculation is $3 \times 3$ and the interval is 2 pixels. After calculation, two groups of $13 \times 13 \times 128$ pixel layers are obtained. The pixel layer is standardized with a scale of $5 \times 5$. After processing, two groups of $13 \times 13 \times 128$ pixel layers are obtained.

Step 2. The $27 \times 27 \times 96$ pixel layer obtained by step 1 was divided into two groups of $27 \times 27 \times 48$ pixel layers.

The pixel layer is convoluted with a convolution kernel of $5 \times 5 \times 48$. The convolution kernel slides in the $X$ and $Y$ directions of the image to perform the calculation. In order to facilitate the calculation, two pixels are expanded around the image, and a new pixel is obtained each time. The sliding interval is 1 pixel. After sliding, a $27 \times 27$ pixel layer is formed. In the second layer, 256 convolution kernels are set, and each 128 convolution cores are a group, so two sets of $27 \times 27 \times 128$ pixel layers are generated. The obtained pixel layer data are input to the convolution layer, and the output is two groups of $27 \times 27 \times 128$ active pixel layers. The active pixel layer is further pooled. The scale of calculation is $3 \times 3$ and the interval is 2 pixels. After calculation, two groups of $13 \times 13 \times 128$ pixel layers are obtained. The pixel layer is standardized with a scale of $5 \times 5$. After processing, two groups of $13 \times 13 \times 128$ pixel layers are obtained.

Step 3. Two groups of $13 \times 13 \times 128$ pixel layers obtained by step 2 were convoluted with convolution kernel with a size of $3 \times 3 \times 256$. Convolution kernel slides in $X$ and $Y$ directions to perform calculation. In order to facilitate the calculation, the image was expanded by 1 pixel each time, and a new pixel was obtained each time. The sliding interval is 1 pixel. After sliding, a $13 \times 13$ pixel layer was formed. In the third layer, 384 convolution kernels were set, and every 192 convolution kernels are a group, so two sets of $13 \times 13 \times 192$ pixel layers are generated. The obtained pixel layer data are input to the correction linear unit, and the output is two groups of $13 \times 13 \times 192$ active pixel layers.

Step 4. Two groups of $13 \times 13 \times 192$ pixel layers obtained by step 3 were convoluted with convolution kernels of $3 \times 3 \times 192$, and convolution kernels slide in $X$ and $Y$ directions to perform calculation. In order to facilitate the calculation, the image is expanded by 1 pixel each time, and a new pixel is obtained each time. The sliding interval is 1 pixel. After sliding, a $13 \times 13 \times 96$ pixel layer was formed. In the fourth layer, 512 convolution kernels were set, and every 256 convolution kernels are a group, so two sets of $13 \times 13 \times 256$ pixel layers are generated. The obtained pixel layer data are input to the correction linear unit, and the output is two groups of $13 \times 13 \times 256$ active pixel layers.

The obtained pixel layer data are input to the classification layer, and the final output is the classification result of the image. The classification result can be used to determine whether the image contains a crater or not.

![Figure 5. Structure diagram of alexnet model.](image-url)
interval is 1 pixel. After sliding, a $13 \times 13$ pixel layer is formed. In the fourth layer, 384 convolution kernels are set, and every 192 convolution kernels are a group, so two sets of $3 \times 3 \times 192$ pixel layers are generated. The obtained pixel layer data are input to the correction linear unit, and the output is two groups of $3 \times 3 \times 192$ active pixel layers.

Step 5. Convoluting two groups of $3 \times 3 \times 192$ pixel layers obtained by step 4 with convolution kernel with a size of $3 \times 3 \times 192$, convolution kernel slides in $X$ and $Y$ directions to perform calculation. In order to facilitate the calculation, the image is expanded by 1 pixel each time, and a new pixel is obtained each time. The sliding interval is 1 pixel. After sliding, a $13 \times 13$ pixel layer is formed. In the fifth layer, 256 convolution cores are set, and each 128 convolution cores are a group, so two sets of $13 \times 13 \times 128$ pixel layers are generated. The obtained pixel layer data are input to the correction linear unit, and the output is two groups of $13 \times 13 \times 128$ active pixel layers. The active pixel layer is further pooled. The scale of calculation is $3 \times 3$ and the interval is 2 pixels. After calculation, two groups of $6 \times 6 \times 128$ pixel layers are obtained, and the total is $6 \times 6 \times 256$ pixel layers.

Step 6. Convoluting the $6 \times 6 \times 256$ pixel layer obtained by step 5 with the filter with the size of $6 \times 6 \times 256$, and the calculation result is output through the neuron. In the sixth layer, 4096 filters and 4096 neurons are set up, and the filter convolution calculation results are output by these neurons. Four thousand ninety-six output values are generated by modifying the linear element, and 4096 calculated results are obtained by dropout calculation.

Step 7. Four thousand ninety-six neurons were set in the seventh layer and connected with 4906 output results of the sixth layer. After that, 4096 output values were generated by the modified linear unit, and 4096 calculated results were obtained by dropout calculation.

Step 8. One thousand neurons were set in the eighth layer and connected with 4906 output results of the seventh layer, and then the results of crater recognition were output.

In this paper, we train the alexnet model on the platform of MATLAB R2018a. The main configuration of the computer is Intel Core i7-4790 processor, NVIDIA geforce GT 705 graphics card. The training samples are obtained from Google Mars surface image, and the Mars surface image was used to make the sample library. The images containing different scale craters were taken as positive samples, and the surface images without craters were selected to make negative samples. One thousand positive and negative samples were selected for image training, and the learning rate is 0.01. Image enhancement strategies such as mirror image, flip and random clipping were used to verify the number of images in the verification set. The training process is shown in Figure 6.

As can be seen from Figure 6, after 20 iterations of training, the recognition accuracy (Validation) is 95.83%, which has high recognition accuracy.

5. Analysis of experimental results

5.1. Performance analysis of edge detection

The edge detection algorithm proposed in this paper is compared with the method proposed by Zhang et al. (2010), and a Mars surface image is randomly selected from the sample database, as shown in Figure 7.

Zhang’s method (2010) and the method of this paper were used to detect the edge of Figure 7, and the results are shown in Figure 8.

It can be seen from Figure 8 that Zhang’s method (2010) is not as effective as the method proposed in this

![Figure 6. Training process.](image-url)
paper, with more discontinuities and poor coherence. In order to further evaluate the detection results, the evaluation index of reference (Lin, 2003) was used for objective evaluation. The ratio of the number of eight connected components \( C \) to the number of edge points \( A \) was used for evaluation. The smaller the ratio, the better the detection effect is. The comparison results are shown in Table 1.

It can be seen from Table 1 that the \( C/A \) value of the method in this paper is 0.0061, which is smaller than the value of 0.0584 in Zhang’s method (2010), which indicates that the edge detection method proposed in this paper has a better effect.

### 5.2. Comparison with other methods

Comparing the method in this paper with Yan’s method (2019), Yan’s method is also a combined detection method. A test picture is randomly selected from the sample library as shown in Figure 9.

For the sample to be tested in Figure 9, Yan’s method and the method of this paper were used to detect and identify crater in Figure 9, and the results are shown in Figure 10.

It can be seen from Figure 10 that both methods have detected three craters and identified them. In order to further evaluate the detection accuracy, the detected common crater areas are numbered 1, 2 and 3 respectively, as shown in Figure 11.

In order to further evaluate the detection accuracy, the IOU indicators were used to evaluate the areas of craters detected and identified by the two methods, and the results are shown in Figure 12.

It can be seen from Figure 12 that for the detected crater areas 1, 2 and 3, the IOU values of the areas detected by the proposed method are higher than those detected by Yan’s method and the average IOU values detected by the proposed method are also higher than the average IOU values detected by Yan’s method, which shows that the proposed method has relatively good effect.
In addition, 20 samples were randomly selected for detection and identification, and the correct detection rate proposed by Ali-Dib et al. (2020) was used as the evaluation index to compare the two methods. The comparison results are shown in Table 2.

### Table 2. Comparison of correct detection rate.

| Evaluating indicator | Yan’s method | The method of this paper |
|----------------------|--------------|--------------------------|
| Correct detection rate | 86%         | 93%                      |

## 6. Conclusion

In this paper, the detection of craters is mainly divided into three parts: crater edge extraction, candidate area determination and crater recognition. In the edge extraction part of the crater, the method based on structure random forest was used to extract the edge of the crater. Based on edge detection, a candidate region extraction method was studied by morphological methods. For the recognition of candidate regions, the recognition method was studied based on deep learning AlexNet model. Experimental results show that the edge detection method of the crater proposed in this paper is better than other edge detection methods; compared with other crater detection methods, the proposed method has relatively high detection accuracy and correct detection rate.

### Disclosure statement

No potential conflict of interest was reported by the author(s).

### Funding

The work described in this paper was supported by the National Natural Science Foundation of China [Grants Number 61703270].
References

Ali-Dib, M., Menou, K., Jackson, A. P., Zhu, C., & Hammond, N. (2020). Automated crater shape retrieval using weakly-supervised deep learning. *Icarus*, 345, 113749. https://doi.org/10.1016/j.icarus.2020.113749

Cadogan, P. H. (2020). Automated precision counting of very small craters at lunar landing sites. *Icarus*, 348, 113822. https://doi.org/10.1016/j.icarus.2020.113822

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Criminisi, A., Shotton, J., & Konukoglu, E. (2011). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2–3), 81–227. https://doi.org/10.1561/0600000035

Ding, W., Stepinski, T. F., Mu, Y., Marchetti, P. G., & Milnes, M. (2011). Subkilometer crater discovery with boosting and transfer learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4), 39–42.

Dollár, P., & Zitnick, C. L. (2013). *Structured forests for fast edge detection*. IEEE International Conference on computer Vision, Piscataway, USA: IEEE, 1841–1848.

Earl, J., Chicarro, A. F., Koebel, C., et al. (2005). *Automatic recognition of crater-like structures in terrestrial and planetary images*. 36th Annual Lunar and Planetary Science Conference, Texas, USA: LPI, 36.

Emami, E., Ahmad, T., Bebis, G., Nefian, A., & Fong, T. (2019). Crater detection using unsupervised algorithms and convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8), 5373–5383. https://doi.org/10.1109/TGRS.2019.2899122

Freeman, W. T. & Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9), 891–906. https://doi.org/10.1109/34.93808

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42. https://doi.org/10.1007/s10994-006-6225-1

Hong, X., Chen, S., & Harris, C. J. (2012). Using zero-norm constraint for sparse probability density function estimation. *International Journal of Systems Science*, 43(11), 2107–2113. https://doi.org/10.1080/00207712.2011.564673

Kang, Z., Wang, X., Hu, T., & Yang, J. (2019). Coarse-to-fine extraction of small-scale lunar impact craters from the CCD images of the Chang‘E Lunar orbiters. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1), 181–193. https://doi.org/10.1109/TGRS.2018.2852717

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks*. Advances in Neural Information Processing Systems, Maine, USA: Curran Associates Inc., 1097–1105.

Lee, H., Choi, H. L., Jung, D., & Choi, S. (2020). Deep neural network-based Landmark selection method for optical navigation on lunar highlands. *IEEE Access*, 2020(4), 1–14.

Li, W., & Hsu, C. Y. (2020). Automated terrain feature identification from remote sensing imagery: A deep learning approach. *International Journal of Geographical Information Science*, 34(4), 637–660. https://doi.org/10.1080/13658816.2018.1542697

Lin, H., Zhao, C., & Shu, N. (2003). Edge detection based on Canny operator and evaluation. *Journal of Heilongjiang Institute of Technology*, 17(2), 3–6.

Martin, D. R., Fowlkes, C. C., & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5), 530–549. https://doi.org/10.1109/TPAMI.2004.1273918

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Proceedings of Computer Vision, Piscataway, USA: IEEE, 2, 416–423.

Pedrosa, M. M., de Azevedo, S. C., da Silva, E. A., & Dias, M. A. (2017). Improved automatic impact crater detection on Mars based on morphological image processing and template matching. *Geomatics, Natural Hazards and Risk*, 8(2), 1306–1319. https://doi.org/10.1080/19475705.2017.1327463

Ren, X., Fowlkes, C. C., & Malik, J. (2005). *Scale-invariant contour completion using conditional random fields*. IEEE International Conference on Computer Vision, Piscataway, USA: IEEE, 2, 1214–1221.

Sawabe, Y., Matsunaga, T., & Rokugawa, S. (2006). Automated detection and classification of lunar craters using multiple approaches. *Advances in Space Research*, 37(1), 21–27. https://doi.org/10.1016/j.asr.2005.08.022

Speybroeck, N. (2012). Classification and regression trees. *International Journal of Public Health*, 57(1), 243–246. https://doi.org/10.1007/s00038-011-0315-z

Urbaeh R. E., & Stepinski, T. F. (2009). Automatic detection of sub-km craters in high resolution planetary images. *Planetary and Space Science*, 57(7), 880–887. https://doi.org/10.1016/j.pss.2009.03.009

Wang, Y., Yang, G., & Guo, L. (2015). A novel sparse boosting method for crater detection in the high resolution planetary image. *Advances in Space Research*, 56(5), 982–991. https://doi.org/10.1016/j.asr.2015.05.014

Xin, X., Di, K., Wang, Y., Wan, W., & Yue, Z. (2017). Automated detection of new impact sites on Martian surface from HiRISE images. *Advances in Space Research*, 60(7), 1557–1569. https://doi.org/10.1016/j.asr.2017.06.044

Yan, Y., Qi, D., & Li, C. (2019). Vision-based crater and rock detection using a cascade decision forest. *Iet Computer Vision*, 13(6), 549–555. https://doi.org/10.1049/iet-cvi.2018.5600

Zhang, J., He, K., Zheng, X., & Zhou, J. (2010). *An ant colony optimization algorithm for image edge detection*. International Conference on Artificial Intelligence and Computational Intelligence, Piscataway, USA: IEEE, 215–219.

Ziou, D., & Tabbone, S. (1998). Edge detection techniques—an overview. *Pattern Recognition and Image Analysis*, 8, 537–559.