Automatic Crater Detection Using Convex Grouping and Convolutional Neural Networks

Ebrahim Emami[1], George Bebis[1], Ara Nefian[2], and Terry Fong[2]

[1]Department of Computer Science and Engineering,
University of Nevada, Reno
ebrahim@nevada.unr.edu, bebis@cse.unr.edu
[2]Intelligent Robotics Group (IRG)
NASA Ames Research Center
ara.nefian@nasa.gov, terry.fong@nasa.gov

Abstract. Craters are some the most important landmarks on the surface of many planets which can be used for autonomous safe landing and spacecraft and rover navigation. Manual detection of craters is laborious and impractical, and many approaches have been proposed in the field to automate this task. However, none of these methods have yet become a standard tool for crater detection due to the challenging nature of this problem. In this paper, we propose a new crater detection algorithm (CDA) which employs a multi-scale candidate region detection step based on convexity cues and candidate region verification based on machine learning. Using an extensive dataset, our method has achieved a 92% detection rate with an 85% precision rate.

1 Introduction

Craters are topographic features on planetary surfaces that result from impacts with meteoroids. They are found on all hard-surface bodies in the solar system, but are most abundant on planets like the Moon or Mars where they can accumulate due to slow surface erosion rates [1]. Craters are important landmarks for autonomous spacecraft and rover navigation and control, which have become key technologies in deep space exploration [2]. Craters can be used for high-precision spacecraft landing missions, and accurate identification of potential hazards [3]. Crater surveys also contain important information about planetary surfaces; for example, crater counting can be used for establishing relative chronology of planetary surfaces.

Currently, all crater databases have been gathered manually via visual inspection of images. However, they are not comprehensive as they mostly contain large craters only. On the other hand, advances in gathering planetary data by space probes has resulted in high resolution images that can show smaller craters on planets like Mars, and Moon. Clearly, manual crater detection is not be appropriate for generating comprehensive catalogues of craters and this task can only be achieved by automating the process of crater surveying [4][1].

There have been numerous efforts to develop CDAs using image processing and machine learning techniques. However, most of the previous approaches are not capable of achieving high performance in real world crater detection applications [1]. Variations in illumination and surface properties as well as variations in shape and size make automatic crater detection a very challenging task. Specifically, crater dimensions in an image might differ by orders of magnitude. Crater shapes may also vary depending on their interior morphologies (central peaks, peak rings, central pits, and wall terraces), level of degradation, and degree of overlap with other craters [3] [4]. These challenges make it hard to design a robust and usable CDA which maintains high accuracy.

In this paper, we propose a new CDA to better deal with these issues. The proposed method consists of two main phases. In the first phase, candidate crater regions are extracted using convexity, an important perceptual organization cue. In the second phase, candidate regions are the classified as crater or non-crater regions using machine learning techniques. In contrast to other CDAs, the proposed method does not rely on strong assumptions about crater shape and properties. Instead of assuming strong circular or elliptical shapes or bright and shadow regions with specific shapes, sizes, and orientations, we only assume that crater regions have a nearly convex shape which is a much weaker assumption. Moreover, we allow crater boundaries to contain gaps which is often the case due to imperfect edge detection results. Using this multiscale scheme based on convexity, we are able to detect almost all craters while rejecting many non-crater regions. The candidate regions are then verified using a Convolutional Neural Network (CNN).

The rest of the paper is organized as follows: Section 2 discussed related work in crater detection. Section 3 presents the proposed approach in detail. Section 4 presents our experimental results and comparisons. Finally, Section 5 presents our conclusions and directions for future research.

2 Background

There has been extensive research on crater detection over the past years. Kamarudin et al. [5] have reviewed several methods on crater detection; they claim that the most accepted method of crater detection is based on edge detection and the Hough Transform, while there exist other techniques based on detection of a bright to dark shading pattern inside the crater due to lighting orientation. Salamuniccar and Loncaric [6] have proposed a framework for evaluating crater detection algorithms, however, the evaluation is not from a machine vision point of view. We classify previous CDAs into two categories: unsupervised and supervised.

Unsupervised methods mainly employ basic image processing and pattern recognition techniques such as thresholding [7], circle detection, and ellipse detection [7], [8]. Specifically, Troglio et al. [8] perform crater detection by extracting elliptical regions using watershed segmentation and the Generalized Hough transform. Smirnov [7] performs crater detection by detecting shadow regions; he assumes very specific geometric properties for craters and limits the geometry of shadow regions to three main shapes which are detected using thresholding, pixel clustering and circle fitting. In [9], Kim et al. propose a CDA based on edge detection and ellipse fitting followed by template matching to discriminate between crater and non-crater regions.

Supervised methods use machine learning techniques to learn how to distinguish between crater and non-crater regions. These methods rely on a large number of labeled data for training. Meng et al. [10] perform candidate crater region selection using the Kanade–Lucas–Tomasi (KLT) detector, while MatLSSVM is employed for verifying the candidate crater regions. They claim that their method detects 88% of craters on their dataset which consists of 160 preprocessed image patches from Google Mars. In [11], Martins el al. use the popular Adaboost algorithm [12] for crater detection using 3216 Haar-like features.

Wetzler et al. [13] have employed several supervised machine learning approaches for small size crater detection. According to their experiments, Support Vector Machines (SVMs) outperform Feed-Forward Neural Networks, AdaBoost (with feed-forward neural networks as base learners) and Continuously Scalable Template Models (CSTM) for crater detection. The classifiers were trained on normalized size image blocks and applied on test images using a sliding window approach. To detect different size craters, image pyramids were utilized. In a similar approach, Palafox et al. [14] evaluated the performance of SVMs and CNNs for the detection of craters and volcanic rootless cones. Although quantitative results were not presented in that work, CNNs were reported to perform better than SVMs in classifying randomly extracted patches from HiRISE images.

In general, unsupervised approaches are fast and more appropriate for the detection of relatively large craters; however, their performance degrades when dealing with smaller craters or more challenging terrains. Therefore, these techniques cannot be used as a general purpose crater detection tool [10] [13]. Supervised methods, on the other hand, are more robust but usually slower and their performance depends on the quality and number of training data.

It should be mentioned that although the majority of methods in the literature employ image data for crater detection, other types of data, such as Digital Elevation Map (DEM) data, have also been used for crater detection [15] [16]. Our interest in this study is on using image data for crater detection.

3 Proposed Method

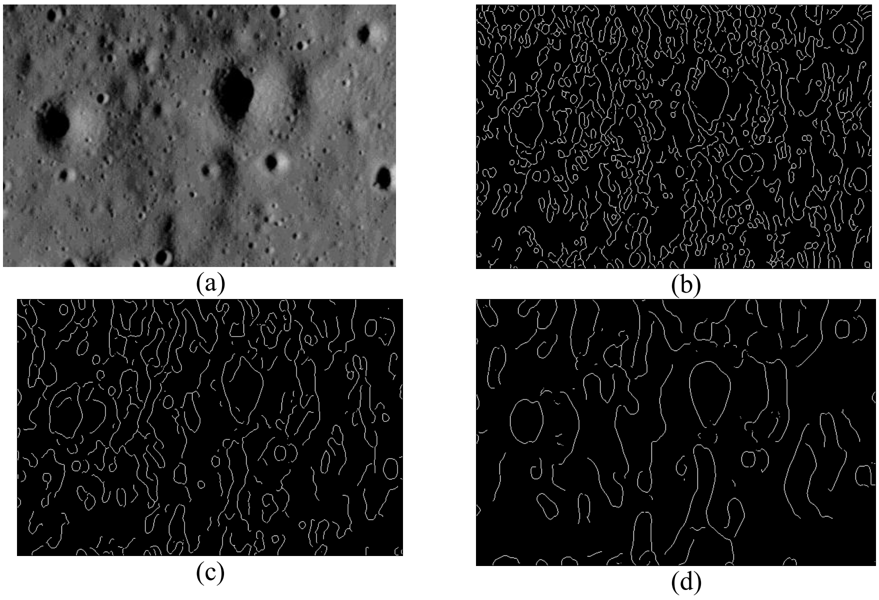
In this paper, we propose a supervised CDA which consists of two main phases: (i) multi-scale candidate crater region detection, and (ii) candidate crater region verification. In the first phase, we extract candidate crater regions by applying multi-scale edge detection and convex grouping. Candidate crater regions are then classified into crater and non-crater regions using machine learning techniques. A set of discriminative features that can accurately separate craters from non-craters should be chosen in this phase. For training, we use a representative set of crater and non-crater training examples. After verification, a post-processing step is applied to combine detections corresponding to the same crater regions.

3.1 Candidate crater region extraction

The primary goal in this step is to detect all true crater regions in order to avoid searching the whole image and speed-up the verification step. Our method is based on a perceptual organization approach which is a bottom up process that clusters image features into higher level organizations, each likely to come from a single object. Many different cues have been proposed in the literature for extracting perceptually salient structures including continuity, parallelism, and proximity; here, we employ convexity [17]. It has been demonstrated that groups of edges forming convex polygons rarely occur at random and are very likely to have resulted from the same convex object. Our method consists of the following steps: (1) multi-scale edge detection, (2) extraction of convex groups, (3) combination of convex groups, and (4) expansion of candidate crater regions.

Multi-scale edge detection

The first step of our method is extracting crater edges using the Canny edge detector. However, using a single scale to extract crater edges would be insufficient since craters typically appear at different sizes in an image.



(a)

(b)

(c)

(d)

Fig 1. Multi-scale Canny edge detection: (a) input image, (b) Canny edges at scale 3 using a threshold of 25 (c) Canny edges at scale 5 using a threshold of 25, and (d) Canny edges at scale 9 using a threshold of 25.

To deal with this issue, we perform edge detection at multiple scales by varying the scale parameter of the Canny edge detector. Figure 1 shows an example using three different scales. As it can be observed, larger craters are more prominent at higher scales while smaller craters are more prominent at lower scales.

Extraction of convex groups

Many methods for crater detection in the literature assume that craters have a circular or elliptical shape. By observing many craters in our data set, however, we have concluded that this is not always the case. Other methods assume that craters consist of a pair of dark and bright regions with specific (relative) sizes, orientations and distances from each other. However, this assumption can be violated depending on the position of the sun.

In this paper, we make a weaker assumption about the shape of craters; specifically, we assume that the shape of craters is nearly convex and we use an efficient convex grouping algorithm [17] to extract candidate crater regions. This algorithm is simple, efficient, and robust to noise, occlusion (i.e., gaps), and clutter. Initially, the image is processed to find line segments by performing edge detection followed by line approximation. Here, we use the split-and-merge algorithm [18] which approximates curves with lines, such that the curve points are no more than a fixed threshold from the line segments. Figure 2 shows an example.

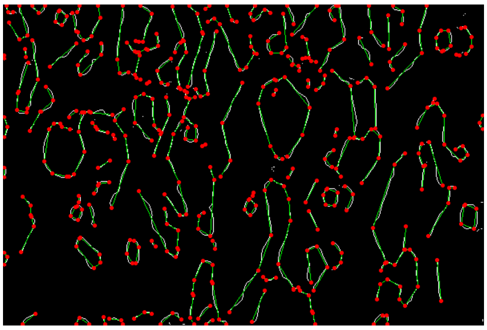


Fig 2. Line fitting results on a sample edge image. The line segments are shown in green and their end points are shown in red.

Each line segment is characterized by its length, orientation and direction (by distinguishing one endpoint as the first endpoint). One way to define convexity is by considering the sum of absolute values of the angles turned as we traverse the line segments of the group. In the case of convex groups, the sum should be 360 degrees. Alternatively, a group of line segments is convex if for each -directed- line segment of the group, all other line segments are on the same side as its normal (it points to the right of the line segment when we traverse it from the first endpoint to the second).

Since the number of convex groups in an image can be very large, the algorithm considers only finding the most *salient* convex groups. A group is considered to be salient, if the sum of gap lengths between line segments is smaller than some fixed proportion of the sum of line lengths in the group. Let us assume that a convex group S_n contains line segments line segments $(l_1, l_2 \ldots l_n)$. We define L_i to be the length of line segment l_i and G_i to be the length of the gap between l_i and l_{i+1}, then the sum of line lengths $L_{1,n}$ and the sum of gap lengths $G_{1,n}$ are defined as follows:

$$L_{1,n} = \sum_{i=1}^{n} L_i \quad \text{and} \quad G_{1,n} = \sum_{i=1}^{n} G_i \tag{1}$$

Then, S_n is called a salient convex group if:

$$\frac{L_{1,n}}{L_{1,n} + G_{1,n}} > k \tag{2}$$

where k is a fixed threshold. Figure 3 shows an example.

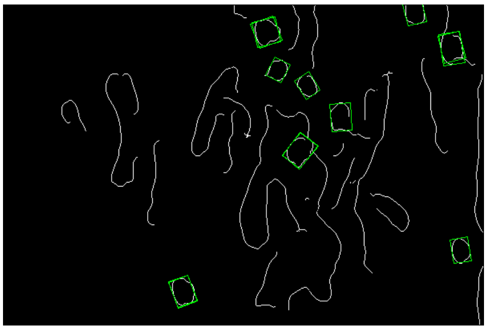


Fig 3. Convex grouping results on a sample image using k=0.85. The detected convex groups are represented with their bounding boxes.

Initially, the algorithm considers every line segment as defining a new convex group. Each group is then grown by adding more segments to it using backtracking. To avoid considering every possible case, several constraints are imposed based on distance (i.e., only segments within a certain distance are considered), convexity (i.e., only segments that preserve convexity are considered) and saliency (i.e., only segments that do not degrade saliency are considered). To facilitate efficient implementation of these constraints, information about the line segments is precomputed and pre-stored in appropriate data structures. To increase system's robustness, several heuristics were used (e.g., the convexity criterion was relaxed to accept not perfectly convex groups). It should be mentioned that the salience criterion is rotation and scale independent. Finding the *m* most salient groups in an image containing *n* segments has $O(n^2 log(n) + mn)$ complexity [17].

Combination of convex groups

Since the same crater might be detected multiple times at different scales or even within the same scale (i.e., by using slightly different line segments each time), it is desirable to combine multiple detections to reduce verification cost but also to improve the extraction of candidate crater regions. We apply the following two steps in order to combine multiple detections: (1) cluster convex groups based on the overlap of their bounding boxes and (2) represent each cluster by the average of bounding boxes. Specifically, two convex groups are clustered together if their intersection to union ratio is above a threshold:

$$\frac{Area(b1 \cap b2)}{Area(b1 \cup b2)} > T \tag{3}$$

where b_1 and b_2 are their bounding boxes. A low threshold results in less candidate crater regions while a high threshold results in more accurate candidate crater regions.

Expansion of candidate crater regions

The convex groups resulting from the previous step might not be perfectly localized around craters. This can affect verification performance especially since the training data comes from manually extracted craters which are typically well localized using a square window. To address this issue, candidate regions are expanded to become square in size (i.e., by making the shorter side of tits bounding box equal to its longer side). In particular, each region is expanded in three different ways: (i) from its left side, (ii) from its right side, and (iii) both from its left and right sides; Figure 4 shows an example. It should be mentioned that although this step increases the number of candidate crater regions, our experimental results show that expanding the candidate regions improves accuracy considerably.

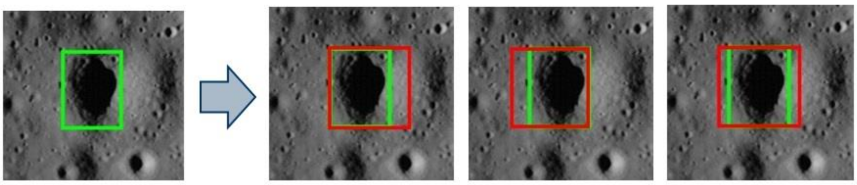


Fig 4. Expanding a candidate region (green box) in three different ways (red boxes).

3.2 Candidate Crater Region Verification

Once the candidate crater regions have been detected, they need to be verified in order to reject non-crater regions. We have opted for a machine learning approach since crater appearance might vary a lot. We have experimented with different features and classifiers including raw pixels, Haar features, Histograms of Oriented Gradient (HOG) features, SVMs, and CNNs. The best performance was obtained using raw features and CNNs which has been often the case in many classification applications. Next, we provide a brief overview CNNs.

Classification using CNNs

Deep learning systems have recently achieved state-of-the-art performance on many classification tasks. CNNs are among the most prominent deep learning techniques [19]. CNNs are feedforward neural networks with a special architecture inspired from the human visual system. They consist of alternating convolution and sub-sampling/pooling layers and work directly on 2D data (maps). The convolution layers compose feature maps by convolving kernels over feature maps in layers below them while the sub-sampling layers, down-sample the feature maps by a constant factor. The activation of a single map *j* in convolution layer *l* is given by:

$$a_j^l = f\left(b_j^l + \sum_{i\in M_j^l} a_i^{l-1} * k_{ij}^l\right) \tag{4}$$

where f is a non-linear function (e.g., *tanh*), and b is a scalar bias. M_j^l is a vector of indices of feature maps in layer *l-1* which feature map *j* in layer *l* should sum over, " * " is the 2D convolution operator and k is the kernel used on feature map *i* in layer *l-1*. For a single feature map *j* in sub-sampling layer *l*:

$$a_j^l = down\left(a_j^{l-1}, N^l\right) \tag{5}$$

where *down* means down-sampling by a factor *N*.

To discriminate between C classes, a fully connected output layer with *C* neurons is added. The output layer takes as input the concatenated feature maps of the layer below it, denoted by the feature vector *fv*:

$$o = f(b^o + W^o fv) \tag{6}$$

where b^o is a bias vector and W^o is a weight matrix which can be determined using the back-propagation learning algorithm [19].

Combination of verified regions

As mentioned in Section 3.1 we combine candidate crater regions by thresholding the ratio of their area of overlap over the union of their areas. Using a fairly high threshold in that step but also adding extra regions by expanding the candidate regions allow us to improve verification performance, however, we might end up with verifying the same crater multiple times. To eliminate multiple detections, we apply the same methodology described in Section 3.1 to the verified regions, however, using a lower threshold this time.

4. Experimental Results and Comparisons

4.1 Data Set

Our data set consists of 448 images, each having a size of 600×400, obtained from the Lunar Reconnaissance Orbiter (LRO) [21]. Craters with a size between 20x20 and 200x200 have been partially labeled by NASA scientists in this dataset. We have used 428 for training and 20 images for testing. A total of 1830 craters (i.e., ground truth) exist in the training images. To increase the variability of crater appearance in the training set, we generate more training samples by randomly shifting the original ones. Specifically, we generate 3 samples for each ground truth crater by slightly changing its position and size. The new samples are still well localized and have more 90% overlap with the ground truth. Combining these samples with the original ground truth craters make up our training set of 7320 samples. All training samples are then normalized to size 24×24.

It should be mentioned that the original partially labeled images are not suitable for testing and fully labeled images are needed for this purpose. Therefore, we have manually labeled all craters larger than 20x20 in the 20 test images. We use 7320 non-crater training examples which are chosen randomly; to reduce the number of false positives, we use bootstrapping [20] to augment the non-crater training samples.

4.2 Performance evaluation measures

Standard recall and precision rates are used to evaluate the performance of our CDA. These measures are defined as follows:

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

where TP, FN, and FP are the number of true positives, false negatives, and false positives respectively. A verified region is a TP if it has more than 40% overlap with a ground truth crater; otherwise, it is a false positive. The overlap between a candidate region and a ground truth crater is calculated using Equation 3. It should be mentioned that our algorithm is designed to detect craters bigger than 20x20, but it is common that smaller craters are also detected. These craters are not considered as true or false detections in our performance evaluation.

4.3 Performance analysis of candidate crater region detection

To evaluate the performance of the proposed candidate region detection method discussed in Section 3.1, we have applied it on all 448 images which include 2480 labeled craters.

Table 1. Statistical performance analysis of the proposed candidate crater region detection.

Total Number of ground truth craters	2480
Number of detected ground truth craters	2464 (99.4%)
Average number of detections per ground truth crater	18.22 (std : 12.27)
Average overlap between true detections and the coresponding ground truth crater	52.70 (std: 11.70)
Average overlap between the best candidate crater regions and the corresponding ground truth crater	75.70 (std: 10.69)
Average number of candidate crater regions per image	7889 (std: 1675)

For edge detection, we used the Canny edge detector at scales 3, 5, 7, and 9. A low threshold of 25 was used to keep most of the detected edges (the high threshold was twice the low threshold). Convex grouping was then applied using a gap tolerance parameter k = 0.51. As it can be inferred from our parameter selection, our main goal is detecting all true craters. To combine the detected convex groups, we used a 70% overlap threshold. While this threshold allows for combining many convex groups, it still allows multiple detections of the same crater region which lead to better verification performance as discussed in Section 3.1.

Table 1 shows the performance of our candidate crater region detection step along with some useful statistics; as it can be seen, we can detect almost all ground truth craters (99.4%). On average, 18 candidate crater regions are detected for each ground truth crater. The detected regions corresponding to a ground truth crater have more than 50% average overlap with it which is higher than our desired 40% overlap. More interestingly, the best candidate regions (i.e., the regions with highest overlap) have more 75% average overlap with their ground truth craters.

4.4 Performance analysis of candidate crater region verification

The performance of the complete crater detection algorithm has been evaluated on the 20 test images. There is a total of 251 ground truth craters in these images which were all detected in the candidate crater region detection step. These regions along with other detections were passed to the verification step. We have performed several different experiments using the CNN and SVM classifiers. The CNN classifier is trained using raw pixel intensities since it extracts its own features. The SVM classifiers was trained using raw pixel intensities, Haar features, and HOG features. Table 2 shows our experimental results without using bootstrapping.

Table 2. Experimental results using different clssifiers and features witout bootstrapping.

Type of classifier	Recall (%)	Precision (%)
SVM using raw pixel intensities	84	24.11
SVM using Haar features	82.21	27.9
SVM using HoG features	93.67	27.31
CNNs using raw pixel intensities	94.46	58.9

As it can be observed from Table 2, the CNN classifier outperforms the SVM classifier both in terms of recall and precision. However, both classifiers have low precision which is mostly due to the lack of challenging non-crater samples in the training set. We have tried to improve the quality of the training set using bootstrapping. Table 3 shows our verification results for the CNN classifier using bootstrapping. By performing two iterations of bootstrapping, we have added around 3000 false positive samples to the training set. This has increased the precision of the CNN classifier from 58.29% to 85.66% while its recall rate has slightly dropped from 94.46% to 92.09%. Figure 5 shows the verified crater regions for a sample test image.

Table 3. CNN classifier`s performance improvement on test set using bootstapping

Bootstrapping round	Number of samples added to the data set	Recall	Precision
#0	---	94.46	58.29
#1	1920	91.30	74.51
#2	1140	92.09	85.66

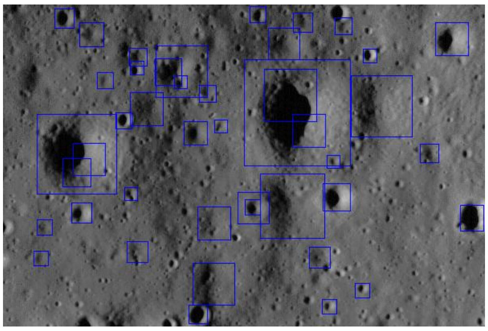


Fig. 5 Verified regions (blue boxes) for a sample test image.

It should be mentioned that most of the false detections are regions which look very similar to eroded craters in our training set. Since it is not clear whether these regions are true craters, we have considered them as false positives. Many of the false negatives are also craters which have very low contrast, lack of edges, and overlap with other surface features. Since these properties are not abundant in our training set, they could not be learnt effectively.

Conclusions

In this paper, we proposed CDA based on perceptual organization and machine learning. Using a multi-scale candidate crater region detection method, we were able to include almost all ground truth crater regions in our candidate list. Using a CNN classifier, we were able to verify 92% of ground truth craters with an 85% precision rate. The proposed approach can be enhanced in several ways. First of all, using a more informative training set with more diverse crater and non-crater samples would improve performance. Second, combining the responses of several classifiers trained on different features is also expected to improve verification performance. Finally, fusing crater detection results from images and DEMs is expected to improve overall performance.

Acknowledgements

This material is based upon work supported by NASA EPSCoR under cooperative agreement No. NNX11AM09A.

References

1. Bandeiraa, L., Ding, W., Tomasz, F.: Detection of Sub-kilometer Craters in High Resolution Planetary Images Using Shape and Texture Features, Advances in Space Research, Vol. 49. Issue 1. (2012) 64–74
2. Yu, Z. , Zhu, S., Cui, P.: Sequence Detection of Planetary Surface Craters From DEM Data, World Congress on Intelligent Control and Automation, (2012)
3. Maoyin A., Pan, W. : Crater Detection Algorithm With Part PHOG Features For Safe Landing, International Conference on Systems and Informatics, (2012) 103-106
4. Salamunićcara, G. , Lončarićb, S., Mazarico, E. : LU60645GT and MA132843GT Catalogues of Lunar and Martian Impact Craters Developed Using a Crater Shape-based Interpolation Crater Detection Algorithm For Topography Data, Planetary and Space Science, Vol. 60. Issue 1. (2012) 236-247
5. Kamarudin, N., Ghani, N., Mustapha, M., Ismail, A., Daud, N. : An Overview of Crater Analyses, Tests and Various Methods of Crater Detection Algorithm, Frontiers in Environmental Engineering, Vol. 1. Issue 1. (2012) 1-7
6. Salamunićcara, G., Lončarić, S. : Open Framework For Objective Evaluation of Crater Detection Algorithms With First Test-field Subsystem Based on MOLA Data, Advances in Space Research, Vol. 42. Issue 1. (2008) 6–19.
7. Smirnov, A. : Exploratory Study of Automated Crater Detection (2012)

8. Troglio, G., Le Moigne, J., Benediktsson, A., Moser, G., Serpico, S. : Automatic Extraction of Ellipsoidal Features for Planetary Image Registration, Geoscience and Remote Sensing Letters, Vol. 9. Issue 1. (2012) 95-99
9. Kim, J., Muller, J. : Impact Crater Detection on Optical Images and DEMS, International Society for Photogrammetry and Remote Sensing, Working Group IV/9: Extraterrestrial Mapping Workshop, Advances in Planetary Mapping (2003)
10. Ding, M,. Caob, Y., Wub, Q. : Novel Approach of Crater Detection by Crater Candidate Region Selection and Matrix-pattern-oriented Least Squares Support Vector Machine, Chinese Journal of Aeronautics, Vol. 26. Issue 2. (2013) 385–39.
11. Martins, R., Pina, P., Marques, J., Silveira, M., Silveira, M. : Crater Detection by a Boosting Approach, Geoscience and Remote Sensing Letters, Vol. 6. Issue 1. (2009) 127-131
12. Viola, P., Jones, M. : Robust Real-Time Face Detection, International Journal of Computer Vision, Vol. 57. Issue 2. (2004) 137-154
13. Wetzler, P., Honda, R., Enke, B., Merline, W., Burl, C. : Learning to Detect Small Impact Craters, 7th IEEE Workshop on Application of Computer Vision, Vol. 1. (2005) 178-184
14. Palafox, L., Alvarez, A., Hamilton, C. : Automated Detection of Impact Craters and Volcanic Rootless Cones in Mars Satellite Imagery Using Convolutional Neural Networks and Support Vector Machines, 46th Lunar and Planetary Science Conference (2015)
15. Salamuniccar, G., Loncaric, S., : Method for Crater Detection From Martian Digital Topography Data Using Gradient Value/Orientation, Morphometry, Vote Analysis, Slip Tuning, and Calibration, IEEE Transactions on Geoscience and Remote Sensing, Vol. 48. Issue 5. (2010) 2317-2329
16. Xie, Y., Tang, G., Yan, S., Hui, L. : Crater Detection Using the Morphological Characteristics of Chang'E-1 Digital Elevation Models, Geoscience and Remote Sensing Letters, IEEE , Vol.10. Issue 4. (2013) 885-889
17. Jacobs, D. : Robust and Efficient Detection of Convex Groups, IEEE Transactions on Patern Analysis and Machine Inteligence, Vol. 18. Issue 1. (1996) 23-37
18. Pavlidis, T., Horowitz, S. : Segmentation of Plane Curves, IEEE Transactions on Computers, Vol. C-23, Issue 8. (1974) 860-870
19. Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence,, 35(8):1798-1828, 2013.
20. Sung, K., Poggio, T. : Example-based Learning For View-based Human Face Detection, IEEE Transactions on Pattern Analysis and Machine Intelligenc, Vol. 20. Issue 1. (1998) 39-51
21. http://www.nasa.gov/mission_pages/LRO/main/index.html.