Automatic Lunar Crater Detection from Optical Images and Elevation Maps

Mara Finkelstein
Stanford University
mfinkels@stanford.edu

Susanna Maria Baby
Stanford University
susmaria@stanford.edu

Hugo Kitano
Stanford University
hkitano@stanford.edu

Abstract

Our project approaches the problem of automatic crater detection using a variety of models and data sources. We adapted Faster R-CNN, an object detection model, and DeepMoon, a segmentation model, to identify the centers and radii of craters given ground-truth crater catalogs. We trained our models on both optical images and digital elevation models of the Moon's surface. Lastly, we saw that these models were able to detect craters on other planetary bodies (Mercury and Mars) as well.

1 Introduction/Related Work

Mapping of the Moon's surface is important for rover traverse and human landing site planning, both of which are priorities for all of the space agencies involved in upcoming lunar missions. Planetary scientists are also interested in astronomical crater statistics, which can reveal crucial information about planet formation and geology. In fact, NASA launched a "crater detection challenge" in 2011, declaring crater detection algorithms as "one of the most challenging and important data mining problems in space exploration" [10]. This competition only lead to limited improvements in existing algorithms, and robust, automatic crater detection remains an open problem.

A wide variety of machine learning algorithms have historically been applied to the problem of crater detection. Initial attempts involved various feature extractors, including the Hough transform [13], to detect craters as circular features, as well as other unsupervised techniques such as detecting shadow regions [25], edge detection, and template matching [15]. Following these initial attempts, it became clear that unsupervised approaches alone do not suffice for complete and accurate crater detection, which led to the advent of various multi-step processes combining unsupervised algorithms for hypothesis generation and supervised algorithms for hypothesis verification [7], [8]. The unsupervised component has remained largely unchanged, while the supervised techniques have spanned the gamut from support vector machines [6] to boosting algorithms (such as Adaboost [16]), polynomical classifiers [4], feed-forward neural networks [12] and, most recently, convolutional neural networks (CNNs) [7], [8], [24]. CNNs have proven particularly effective for the crater detection task, and their hierarchical architecture makes them well-suited for classifying objects, such as craters, that appear at multiple scales. DeepMoon [24], the current state-of-the-art, forgoes unsupervised hypothesis generation altogether and simply uses a CNN-based architecture (combined with post-processing) for crater localization.

In addition to model architecture, data representation is also an important design choice when developing crater detection algorithms. While optical images have been the traditional inputs, digital elevation models (DEMs) have become more popular in recent years since they are not subject to varying sun angles, which can confound detection. DEMs are used as inputs in the DeepMoon model [24], amongst others [19], [27].

Our project primarily investigates supervised machine learning approaches, and we experiment with both optical image and DEM data representations. Before training supervised models, however, we built a template-matching pipeline that augments our crater catalogues with missing labels. We then experimented with CNNs both for object detection and segmentation. First, we trained two baseline models: an adapted version of Faster R-CNN [18] and DeepMoon [24]. In addition to running optical images through our template-matching pipeline before inputting them into our Faster R-CNN model, we also ran DeepMoon on optical images instead of DEMs, and added dilated convolutions to DeepMoon to improve detection of large craters. Lastly, we investigated whether our DeepMoon model could detect craters on other planetary bodies, and found that it could identify craters on Mercury and Mars with reasonable accuracy.

2 Datasets

We compared the performance of crater detection algorithms on two different representations of lunar topography: optical images and digital elevation models (DEMs) of the Moon. We used various crater catalogs compiled by experts to train and evaluate our models.

2.1 Optical Images

Our optical dataset, obtained from the Lunar Reconnaissance Orbiter Narrow Angle Cameras [1], consists of 570 images from the Moon's equatorial region. Each image has size 600×400 pixels with a resolution of 1 meter per pixel. Dr. Ebrahim Emami from the University of Nevada-Reno [8] provided us with two raw labels files, one containing crater annotations that have been confirmed by NASA scientists, and the other containing craters labeled by Emami's students. All labeled craters are in the range of $10 - 100$ m, but there are significant missing labels. We wrote scripts to combine the scientist and student labels, to convert them into the appropriate format (with top left and bottom right bounding box coordinates), and to organize them into a file structure readable by our crater detection algorithms. We also randomly divided the data into train, validation, and test splits of sizes 342 images, 114 images, and 114 images, respectively, and applied horizontal and vertical flips to augment the dataset.

2.1.1 Generating Missing Labels through Template Matching

The most significant issue with this project, and one that separates it from other object detection tasks, is incomplete labeling of the ground-truth data. All crater annotations must be manually recorded by an expert, and a significant portion of true craters are unlabeled in our optical image dataset (Figure 3 (a)). Moreover, Greeley and Gault [11] estimate that manual crater counting by experts can yield disagreements as high as 40%, and Silburt et al. [24] suggest that further work using a "more complete ground-truth crater distribution would presumably yield better results." Previous crater detection algorithms have generated "crater hypotheses" using such methods as Hough transforms, highlight-shadow region detection, and convex grouping. The hypotheses are then verified in a second step [7].

Here, we investigate a novel automated labeling approach, in which we generate high-confidence missing labels via multi-scale template matching and add these to our ground-truth dataset before training our crater detection algorithms. While template matching in isolation is not robust enough to yield optimal crater detection results, we found that given a high enough match threshold, it generates very few false positives while notably increasing the total number of ground-truth labels and, hence, the quality of the training set.

We built the template-matching pipeline from scratch using OpenCV. To generate template-based labels, we selected one ground-truth crater annotation per image (given the variation in sun angle across images, a per-image template was necessary), then identified all matches using the normed correlation coefficient as the similarity metric and a match threshold of 0.9. (We also experimented with thresholds less than 0.9, but found that these yielded too many false positive crater detections.) Our pipeline identified multi-scale matches at 10 scales between 0.5 and 2.5, by scaling the template relative to the image. We then used non-maximum suppression (NMS) to remove duplicate (overlapping) proposals for the same crater, with an intersection-over-union (IoU) threshold of 0.5.

2.2 DEMs

We obtained our DEMs from the Kaguya Merged Digital Elevation Model [26]. Using DEMs instead of optical images functions as a type of feature extraction since DEMs are not affected by changes in lighting, which can cause shadows that interfere with detection. However, DEMs are expensive to compute, and so are often downsampled, in this case to 118 meters per pixel. Hence, the DEMs used here have a lower resolution than the optical images, which can be problematic when the crater detection algorithm is being used for fine-scale mission planning. The DeepMoon pre-processing pipeline augments the DEMs with shifts and rotations. The main catalog we used for training with our DeepMoon-based models was combined from two sources: the LROC crater dataset assembled by Povilaitis et al [17], which contains craters of diameter 5-20 km, and the crater dataset assembled by Head et al [9], which contains craters larger than 20 km in diameter.

To see if our models could detect craters on other planetary bodies, we used the Mercury MESSENGER Global DEM [3] and the Mars MGS MOLA - MEX HRSC Blended Global DEM [2]. There is no ground-truth crater catalog for Mercury, but there is for Mars: the MA132843 dataset assembled by Salmuniccar et al [20]. We used this catalog to obtain accuracy metrics for predictions on Mars.

3 Methods

We experiment with applying two distinct computer vision frameworks to the crater detection task, comparing performance both across frameworks and across datasets. The first framework is an object detection model, and the second, an object segmentation model.

3.1 Faster-RCNN

As our baseline, we trained Faster R-CNN [18] (using the implementation from [14]), a state-of-the-art neural network for object detection, on the optical crater dataset. In contrast to Fast R-CNN, which consists of an isolated (and computationally intensive) region proposal network (RPN), followed by a deep convolutional network for object detection, the proposal and detection networks in Faster R-CNN are combined into a single deep convolutional neural network, sharing features. Faster R-CNN consists of a stack of convolutional layers which function as feature extractors, followed by the RPN, which functions as an attention mechanism and outputs hypotheses (i.e."region proposals"), parameterized relative to reference boxes called "anchors", for the positive ("object") and negative ("no object") classes, along with "objectness scores" (the likelihood that the proposal is an object). Lastly, the hypotheses are fed into the detector, which converts all of the proposals to a fixed shape and produces the final predictions. The RPN performs a sliding window pass over the feature map to generate lower-dimensional feature representations, then branches into two fully-connected layers: a bounding-box regression layer, which outputs the coordinates $\{t_i\}$ of the proposals relative to several pre-defined anchors i centered at each pixel location, and a bounding-box classification layer, which outputs “objectness scores” $\{p_i\}$ for each anchor. (The final output of the detector also has a similar two-branch structure, with a softmax classifier and a bounding box regressor).

The network is trained via alternating minimization, and the RPN is trained first, using a multi-task loss function that seeks to simultaneously minimize the classification loss from labeling a bounding box incorrectly and the regression loss of the predicted bounding box coordinates relative to the ground truth (with the regression term limited to bounding boxes which have positive ground-truth labels). Given p_i and t_i, let p_i^* and t_i^* be the corresponding ground-truth labels, and let L_{cls} and L_{reg} represent the classification loss (log loss) and regression loss (smooth L_1), respectively. Then the multi-task loss for an image can be written as

$$L(\{p_i\},\{t_i\}) = \sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda \sum_i p_i^* L_{\text{reg}}(t_i, t_i^*),$$

where $\lambda \approx 1$, so that the two terms are approximately equally weighted.

Training Faster R-CNN on the optical crater dataset required significant pre-processing (see Section 2.1) in order to format the data and labels into a structure readable by the model. After establishing our baseline, we also trained Faster R-CNN on the optical images with labels augmented by template matching (see Section 2.1.1).

3.2 DeepMoon

In contrast to Faster R-CNN, which detects objects by outputting bounding boxes, DeepMoon [24] (using the implementation from [23]) is a segmentation model, which classifies individual pixels via image-to-image mapping. It consists of a convolutional neural network adapted from the UNET architecture, and is customized for lunar crater detection. UNET consists of a contraction path followed by an expansion path: for the first half of its layers, it downsamples, and then upsamples for the second half, all while maintaining short-cut connections between the layers. The small 3×3 convolutional filters used in DeepMoon limit the model’s ability to detect large craters, so in order to increase the receptive field size without inflating the number of parameters, we also modified DeepMoon by dilating its inner convolutional layers.

DeepMoon is designed to take DEMs as input, but we also modified the input pre-processing pipeline to accept optical images to investigate the variation in performance across these two different feature representations. The ground truth for DeepMoon consists of binary ring masks outlining crater circumferences. The model outputs arrays of pixel intensities, which are converted to binary masks via thresholding, and a circular template is applied to extract the craters. Lastly, there is a post-processing step which aggregates images across targets (which overlap in space) and removes duplicate craters. DeepMoon trains on pixel-wise binary cross-entropy as its loss: The loss l_i at pixel i is given by

$$l_i = x_i - x_i z_i + \log(1 + \exp(-x_i)),$$

where z_i is the ground-truth target value of pixel i and x_i is the corresponding DeepMoon prediction.

3.2.1 Model Predictions on the Craters of Mercury and Mars

Given that craters on different planets have divergent properties, due to varying gravitational acceleration, surface composition, terrain, and impact history across astronomical bodies [24], we investigated DeepMoon’s sensitivity to domain shift by evaluating it on Mercury and Mars. For Mercury, we were able to locate a DEM but no ground truth catalog, so we had to verify the results manually. For Mars, we adapted a DEM and a crater catalog to the DeepMoon infrastructure, and predicted craters from our lunar model. Since we had a ground-truth, we could directly compare the accuracy metrics from Mars predictions to Moon predictions.

4 Experiments/Results/Discussion

We used recall and precision as our evaluation metrics across all models, and also computed the mean average precision (mAP) for models trained using Faster R-CNN. The mAP, a standard metric used in object detection competitions, calculates the mean area under the precision-recall curve across intersection-over-union (IoU) values ranging from 0.5 to 0.95 with a step size of 0.05. The IoU threshold is used to distinguish true positives from false positives, based on the amount of overlap between the predicted bounding box and the ground truth. Figure 1 summarizes our results across all models and datasets:

Summary of test set results	No filtering		Filtering for craters with diam. $>$ 20 pixels	
	Precision	Recall	Precision	Recall
Faster R-CNN baseline	0.36	0.19	0.41	0.48
Faster R-CNN with template-generated labels	0.58	0.22	0.61	0.62
DeepMoon baseline	0.84	0.51	0.33	0.66
DeepMoon with dilated conv	0.85	0.50	0.34	0.64
DeepMoon on optical images	0.64	0.36	0.46	0.59
DeepMoon with dilated conv on optical images	0.67	0.32	0.50	0.55
DeepMoon baseline on Mars DEMs	0.93	0.36	0.34	0.62

Figure 1: Summary of results across all models and datasets

4.1 Faster R-CNN

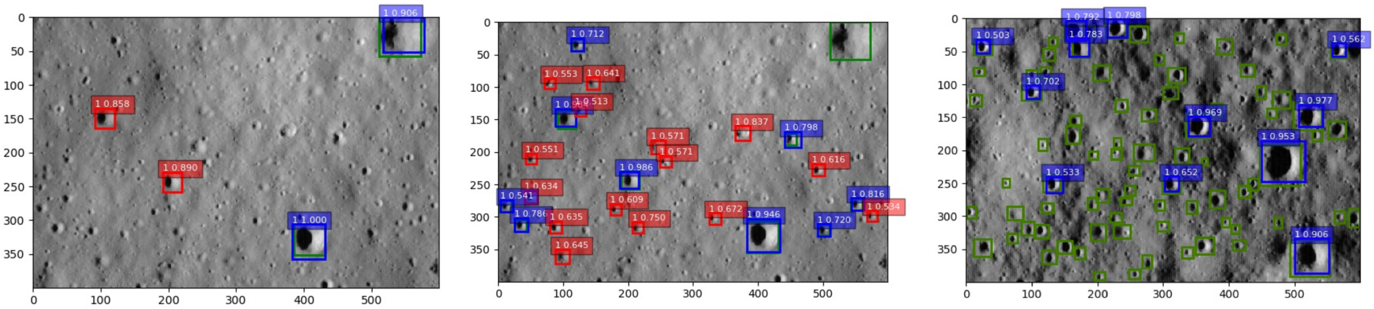
We experimented with various hyperparameter settings for Faster R-CNN, and settled on the hyperparameters that yielded the highest mAP on the validation set. These included an initial learning rate of 0.001 with a decay factor of 0.5 every 10 epochs, RPN anchor boxes scales of $[8, 16, 24] *$ aspect ratios, where we used aspect ratios of $1:1$, $1:2$, and $2:1$. For the RPN, we used an IoU minimum threshold of 0.5 for classifying positive examples, and an IoU maximum threshold of 0.3 for classifying negative examples (with examples having IoU between 0.3 and 0.5 not considered during training). We also used a NMS threshold of 0.7 for filtering RPN proposals to feed into the detector. We used one image per batch, with 128 regions of interest per image, as in the default Faster R-CNN implementation.

Figure 1 and Figure 2 show that the Faster R-CNN trained on the augmented labels performed significantly better than the baseline, with especially notable increases in precision, recall, and mAP when filtering for craters with diameters greater than 20 pixels. Even without filtering, the precision increased by 0.22, accompanied by a small increase in recall. Also, note that without filtering, precision is consistently higher than recall, indicating more false negatives than false positives (Figure 3 (c)), likely because the negative examples sampled from the background actually contained unlabeled craters, and so with more training, the model learned to classify true craters as part of the negative class. With filtering, the recall increases dramatically across both Faster R-CNN-based models, suggesting that most of the false negatives are smaller craters.

Test set mAP	No filtering	Filtering for craters with diam. $>$ 20 pixels
Faster R-CNN baseline	0.12	0.37
Faster R-CNN with template-generated labels	0.17	0.52

Figure 2: mAP of Faster R-CNN improved when trained on crater catalog with template-augmented labels

Figure 3 qualitatively compares the performance of Faster R-CNN when trained on datasets with and without augmented labels. Green bounding boxes indicate ground truth labels, blue boxes indicate true positive predictions, and red boxes indicate false positive predictions. The numbers on the blue and red boxes indicate the "objectness scores," which can be interpreted as the confidence that the detection is an object. For the Faster R-CNN baseline (Figure 3 (a)), both "false positive" predictions in the image are actually true craters not labeled in the ground truth. These false positives from part (a) are labeled during template matching, and are (still) successfully predicted by the model trained on template-augmented labels, but this time are classified as true positives (Figure 3 (b)). Note that this model also discovers several new craters (red boxes), even though these are classified as false positives during the model's evaluation. Hence, the true precision is likely higher than indicated by the evaluation metrics, given incomplete ground-truth labels. Finally, note that the higher incidence of false negatives relative to false positives (Figure 3 (c)) is preferable, since we can always augment this algorithm with additional methods to detect more craters.



(a) Faster-RCNN test set predictions on model trained without template-generated labels.

(b) Faster-RCNN test set predictions on model trained with template-generated labels.

(c) Same model as part (b). Precision is higher than recall, as the false negative rate is greater than the false positive rate.

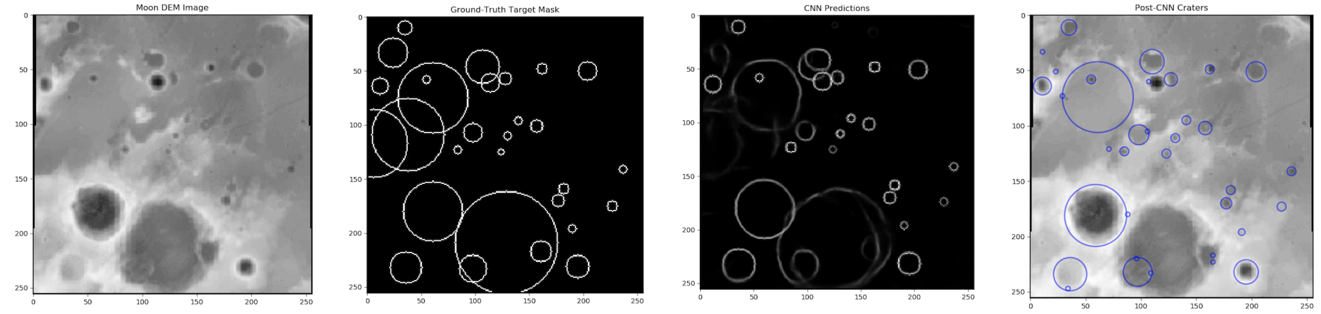
Figure 3: Comparison of Faster R-CNN test set performance with and without augmented labels

4.2 DeepMoon

We used the original DeepMoon architecture, with 112 convolutional filters in the outer convolutional layers, and cropped our input images to size 256×256 pixels. We set a learning rate of 0.0001, a batch size of 8, and used a lambda regularization coefficient of 1e-6. We also used a dropout rate of 0.15.

When we trained DeepMoon on the Kaguya DEM and the combined LROC and Head crater catalogues, the model identified craters with smaller radius very well, even finding some craters clearly visible in the images though absent in the ground-truth dataset, but sometimes missed larger craters, when the craters were larger than the model's receptive field size. In an effort to improve detection on large craters, we tweaked DeepMoon to use dilated convolutions. Surprisingly, our dilated convolutions did not result in improved detection of larger craters ($>$ 20 pixels in general). We saw marginally better precision and marginally worse recall on both optical images and DEMs, which could be useful in cases where we seek to reduce the number of false positives as much as possible.

We also trained DeepMoon on the optical images instead of the DEMs, and saw worse performance. This is likely because DeepMoon's preprocessing works better with DEMs and, despite the data augmentation (including rotations), the optical images' lighting variability interfered with detection.



(a) An example of an original DEM image

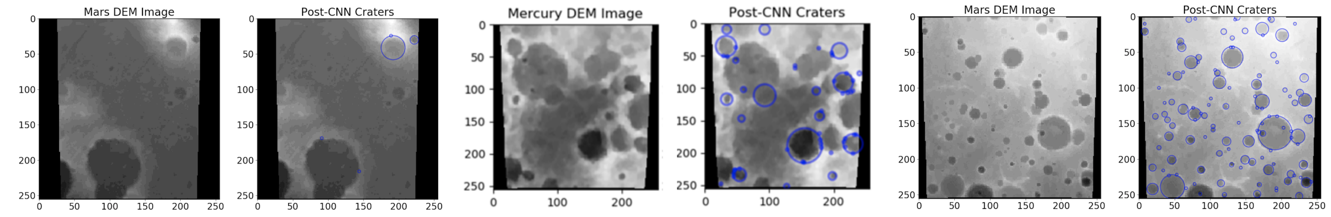
(b) Ground-truth crater visualization

(c) Craters predicted by pre-trained DeepMoon model

(d) Predicted craters overlaid on DEM image

4.3 Model performance on the Craters of Mercury and Mars

We input DEMs of the surface of Mercury and Mars into our DeepMoon model (trained on lunar craters) to see if our model could identify their craters as well. For Mercury, we could not find an accompanying ground-truth catalog, but from the eye-test, our model seems to do well. For Mars, we did have a ground-truth catalog, and thus were able to obtain evaluation metrics: our precision was very high (0.93), even better than on our lunar data, though our recall was about the same (0.36). However, as in the baseline, our model struggled to detect larger craters for similar reasons. The fact that our model is able to detect craters from multiple planetary bodies implies that our model has learned about the features of craters themselves, rather than specifically attributes of lunar craters.



(a) prediction failure, Mars

(b) prediction partial success, Mercury

(c) prediction success, Mars

5 Conclusion/Future Work

Even in the presence of complete ground truths, designing robust crater detection models is difficult: craters vary greatly in shape, illumination, and size. They also overlap and suffer degradation over time, which gradually changes their morphology. The incomplete crater catalogs further confounded both model training and evaluation. Precision is affected since our models are penalized for detecting true craters that are not recorded in our catalogs, and recall is affected since the models are trained on background distributions that contain unlabeled craters, capping their possible performance. Thus, the effectiveness of our models is likely underestimated. Going forward, complete crater catalogs, though time-consuming to prepare, are necessary for more powerful models.

We showed that automatically generating some missing labels in the crater catalogs through template matching can measurably improve model performance. We also showed that the UNET-based segmentation model (DeepMoon) performs better than the Faster R-CNN-based object detection model, and that model performance transfers to crater detection on other astronomical bodies. While Faster-RCNN and DeepMoon both perform reasonably well on mid-sized craters, we would need higher-resolution DEMs or better techniques for detecting craters in optical images, as well as more complete labels, to detect craters across different size scales. Optimization of the dilated convolutions in DeepMoon may also help with large crater detection. Finally, combining high-resolution DEMs and optical images as model inputs would likely be a fruitful path for future work.

6 Acknowledgements

We thank Dr. Ebrahim Emami for providing the crater catalogs for the optical images, Dr. Ara Nefian for putting us in touch with Dr. Emami, and Dr. Mark Shirley for suggesting this project.

7 Contributions

- Mara completed all work related to Faster R-CNN, including configuring the model to run on crater data, training, hyperparameter tuning, and evaluation across different radius thresholds. She also conceived of and built the template-matching label augmentation pipeline, built the pipeline for evaluating DeepMoon on custom DEM datasets with and without ground-truth labels, and ran DeepMoon predictions on the Mercury DEMs. She performed the literature review, conceived of training DeepMoon on optical images, and of evaluating its performance on Mars DEMs. She wrote all of sections 1, 2.1 (including 2.1.1), 3.1, 3.2, 4.1, and 5 of this paper, and wrote parts of sections 2.2, 3.2, 4.2, and 4.3.

- Susanna worked on reproducing the original DeepMoon model, improving it with dilated convolutions and hyperparameter tuning, extending and retraining DeepMoon to work on optical images, testing all of these with different radius threshold, and built the pipeline for evaluating any DeepMoon model on any custom DEM dataset like Mars. She also worked on PyCDA:a python based crater detection algorithm as a baseline model.

- Hugo investigated the entire DeepMoon pipeline, wrote scripts for the DeepMoon image visualizations, prepared and adapted the Mars images and crater catalog, and obtained the Mars evaluation results. He wrote sections 1.1, 2.2, 3.2, 4,2, 4.3 and 5 of this paper.

Our code can be found at `https://github.com/hugokitano/cs229_project` and `https://github.com/maraf10/crater_detection`.

References

[1] Lunar reconnaissance orbiter (lro) online data volumes.

[2] Mars mgs mola - mex hrsc blended dem global 200m v2.

[3] Mercury messenger global dem 665m v2.

[4] A. Boukercha, A. Al-Tameemi, A. Grumpe, and C. Wöhler. Automatic crater recognition using machine learning with different features and their combination. In *Lunar and Planetary Science Conference*, volume 45, page 2842, 2014.

[5] M. Chen, D. Liu, K. Qian, J. Li, M. Lei, and Y. Zhou. Lunar crater detection based on terrain analysis and mathematical morphology methods using digital elevation models. *IEEE Transactions on Geoscience and Remote Sensing*, 56(7):3681–3692, 2018.

[6] M. Ding, Y. Cao, and Q. Wu. Novel approach of crater detection by crater candidate region selection and matrix-pattern-oriented least squares support vector machine. *Chinese Journal of Aeronautics*, 26(2):385–393, 2013.

[7] E. Emami, T. Ahmad, G. Bebis, A. Nefian, and T. Fong. Crater detection using unsupervised algorithms and convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[8] E. Emami, G. Bebis, A. Nefian, and T. Fong. Automatic crater detection using convex grouping and convolutional neural networks. In *International Symposium on Visual Computing*, pages 213–224. Springer, 2015.

[9] C. I. Fassett, S. J. Kadish, J. W. Head, S. C. Solomon, and R. G. Strom. The global population of large craters on mercury and comparison with the moon. *Geophysical Research Letters*, 38(10), 2011.

[10] T. Fong. Irg "crater detection" challenge for nasa tournament lab.

[11] R. Greeley and D. E. Gault. Precision size-frequency distributions of craters for 12 selected areas of the lunar surface. *The Moon*, 2(1):10–77, 1970.

[12] B. Industry. By application. *IEEE Guide for Performing Arc-Flash Hazard Calculations Amendment 2: Changes to Clause 4 IEEE Industry Applications Society*, 2011.

[13] N. D. Kamarudin, K. A. Ghani, M. Mustapha, A. Ismail, and N. Daud. An overview of crater analyses, tests and various methods of crater detection algorithm. *Frontiers in Environmental Engineering*, 1(1):1–7, 2012.

[14] Kevinjliang. kevinjliang/tf-faster-rcnn.

[15] J. Kim and J. Muller. Impact crater detection on optical images and dems. 2003.

[16] R. Martins, P. Pina, J. S. Marques, and M. Silveira. Crater detection by a boosting approach. *IEEE Geoscience and Remote Sensing Letters*, 6(1):127–131, 2008.

[17] R. Povilaitis, M. Robinson, C. Van der Bogert, H. Hiesinger, H. Meyer, and L. Ostrach. Crater density differences: Exploring regional resurfacing, secondary crater populations, and crater saturation equilibrium on the moon. *Planetary and Space Science*, 162:41–51, 2018.

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[19] G. Salamuniccar and S. Loncaric. 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*, 48(5):2317–2329, 2010.

[20] G. Salamunićcar, S. Lončarić, and E. Mazarico. 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*, 60(1):236–247, 2012.

[21] A. L. Salih, A. Boukercha, A. Grumpe, C. Wöhler, and H. Hiesinger. Automatic crater detection and implications for surface age estimation. In *European Planetary Science Congress*, volume 9, 2014.

[22] Y. Sawabe, T. Matsunaga, and S. Rokugawa. Automated detection and classification of lunar craters using multiple approaches. *Advances in Space Research*, 37(1):21–27, 2006.

[23] Silburt. silburt/deepmoon.

[24] A. Silburt, M. Ali-Dib, C. Zhu, A. Jackson, D. Valencia, Y. Kissin, D. Tamayo, and K. Menou. Lunar crater identification via deep learning. *Icarus*, 317:27–38, 2019.

[25] A. A. Smirnov. Exploratory study of automated crater detection algorithm. *Boulder, CO*, 2002.

[26] L. Team et al. Kaguya team, 2015. lro lola and kaguya terrain camera dem merge 60n60s 512ppd (59m).

[27] Y. Xie, G. Tang, S. Yan, and H. Lin. Crater detection using the morphological characteristics of chang'e-1 digital elevation models. *IEEE Geoscience and Remote Sensing Letters*, 10(4):885–889, 2013.

[28] Y. Zhou, H. Zhao, M. Chen, J. Tu, and L. Yan. Automatic detection of lunar craters based on dem data with the terrain analysis method. *Planetary and Space Science*, 160:1–11, 2018.