On-line hazard detection on lunar digital elevation maps using semantic segmentation

Rahul Moghe* and Renato Zanetti†
The University of Texas at Austin, Austin, Texas 78712

NASA's Autonomous Landing and Hazard Avoidance Technology project has three core functionalities, namely, Terrain Relative Navigation, Hazard Detection and Avoidance and Hazard Relative Navigation. In this paper, we present a real time machine learning based algorithm for Hazard Detection to be deployed during the landing phase. A computer vision technique called Semantic Segmentation is used to classify safe and hazardous landing spots for the spacecraft. Randomly sampled Lunar DEMs from the Lunar Reconnaissance Orbiter mission of 2009 are used to train the convolutional neural network (CNN). The ground truth is calculated according the mission requirements and use existing techniques to calculate slope and roughness. Data augmentation techniques are then used to artificially create additional DEMs by transforming the existing data set. The CNN is validated and testing using similarly sampled DEMs. The results show that CNNs perform well for real time processing of spatially correlated data and hence can be useful for performing Hazard Detection and autonomous landing in future missions.

I. Introduction

The Hazard Detection System (HDS) is a primary component of the cross-NASA developed Autonomous Landing and Hazard Avoidance Technology (ALHAT) sensor suite [1–4]. The objective of the HDS is to provide guidance, navigation and control capabilities for the spacecraft to perform autonomous landing under robust lighting conditions. Using active range sensors, the HDS generates a Digital Elevation Map (DEM) of the terrain which is then processed to detect hazards on the landing area. In order to determine safe landing sites, the DEM is analyzed for candidate locations that satisfy mission specifications such as geometric constraints, slope requirements, terrain roughness limits, etc. Since the mission specifications are spatially correlated, convolutional neural networks (CNN) are effective in finding landing sites which satisfy them. As we further complicate the mission specifications, evaluating each landing spot to satisfy these specifications in real time becomes a computationally intensive task. Hence, CNNs can speed up the process significantly by aggregating all the specifications into a neural network and evaluating the DEM as a whole. In this paper, we present a robust learning algorithm to detect hazards and safe landing sites using CNNs. In particular, we use

*Ph.D. Student, Aerospace Engineering and Engineering Mechanics Department; rahul.moghe@utexas.edu, Student Member AIAA.
†Assistant Professor, Aerospace Engineering and Engineering Mechanics Department; renato@utexas.edu, Senior Member AIAA.

a machine learning technique called Semantic Segmentation to determine safe landing options by analyzing the DEM of the Lunar surface.

Previous studies on Hazard Detection and Avoidance (HDA) have used passive optical sensors like cameras as well as active sensors like LIDAR sensors. LIDAR based methods have become popular because they are robust to different lighting conditions. The ALHAT sensor suite primarily uses LIDAR and other range sensors for the same reason. In order to lower the computational costs, flash LIDAR sensors are used because of their ability to instantly measure the surface heights rather than scan every position of the surface. Studies have been conducted to construct sensors to obtain real time reliable DEMs [5–7].

Machine learning methods have been used in crater identification on the Lunar surface in the past [8–12]. CNNs were used for learning the position and radius of the crater on the surface by analyzing the digital elevation map (DEM) of the surface. The detected craters are used for Terrain Relative Navigation (TRN) wherein a comparison with an existing map provides a position measurement used for navigation. In this paper, we present a CNN based method for detecting hazardous areas for autonomous landing. Since, the HDA phase typically takes place between $0.5 - 2$ km. A number of tests for LIDAR based Hazard Detection have been performed in the past. For example, the ALHAT sensor suite was fully integrated with the Morpheus vehicle and Hazard detection and precision landing was tested [13–16].

Semantic Segmentation is a method used in the computer vision community wherein a CNN is trained to identify parts of the image and classify them into a set of predetermined classes. This technique excels at tasks like HDA because of the strong spatial correlation between the pixels of the DEM used to determine the parameters of the surface. Safety of the landing spot depends on parameters such as slope of the surface, geometry of the vehicle, surface roughness and proximity to potential hazards. The CNN learns a probability of a position being hazardous and thresholds this probability to obtain an inference.

A popular technique called data augmentation is used to prevent the network from overfitting the training data, a popular technique used is data augmentation. The DEM collected from the Lunar Reconnaissance Orbiter (LRO) mission is used in this paper. Existing training data is transformed using geometric transformations like rotation by a multiple of 90°, vertical and horizontal flipping, transposing operation, and addition of Gaussian noise, to artificially construct additional training data. This artificial data is augmented to the training set to train a network which is robust to sensor noises and different topographies that may appear in the DEM.

This paper presents a machine learning methodology for Hazard Detection and Avoidance for precision landing on the Lunar surface. The mission specifications for a typical Lunar landing mission is described in Section II. The data preparation methods for creating the training, testing and validation data sets from the LRO mission is described in Section III. The data augmentation methods are described in Section III.B. The neural network structure and the training methodology is detailed in Section IV. We finally showcase our results on the testing data set in Section V.

Parameters	Requirements
Heading of the spacecraft	$< \pm 10^\circ$ from the vertical
Slope	$< 5^\circ$
Proximity to hazards	> footprint size and system uncertainties
Landing position error	$< 100m$
DEM Processing time	$<$ 5 seconds
Surface feature size	$< 20cm$

Table 1 Mission specifications for a safe landing spot on the Lunar surface.

II. Specifications

Table 1 presents the mission specifications given in the ALHAT project for HDA. The main parameters considered when deciding safety precautions are the slope, geometry of the spacecraft and its landing pads, and the size of the features on the surface. Computational complexity is also a constrained since the DEMs have to be processed and used for navigation. The intended or the preplanned landing site must not be too far from the calculated landing site. Hence, safe landing spots closer to the intended landing site will be preferred.

In this paper, proximity to the intended landing site is considered. The aim is to find all the safe landing spots and then post-processing may be used to determine proximity to the planned landing spot.

III. Data Preparation

A. Ground Truth

For this study we use the DEM data collected during the LRO mission which started in 2009 [17]. A large DEM of the Lunar surface spanning between the equator and 15° north latitude and longitude between 30° and 60° is used for creating the training set. Smaller areas of size $(400, 400)$ pixels are then randomly sampled from this dense DEM. Each pixel represents the height of that position on the surface. The sampled DEM simulates a scan generated during the HDA phase of the spacecraft descent phase. Existing algorithms from the literature are then applied which calculate the slope of the terrain at a particular spatial position. In our case, a third order difference method is used to calculate the partial derivatives in a global x and y direction [18]. This is given by

$$f_x = \frac{(z_{SE} - z_{SW} + \sqrt{(2)}(z_E - z_W) + z_{NE} - z_{NW})}{(4 + 2\sqrt{2})g} \tag{1}$$

$$f_y = \frac{(z_{NW} - z_{SW} + \sqrt{(2)}(z_N - z_S) + z_{NE} - z_{SE})}{(4 + 2\sqrt{2})g} \tag{2}$$

wherein, z_A denotes the elevation of the neighboring position on the map in the A direction and g is the spatial resolution of the DEM. The slope of the surface at a spatial position is calculated according to

$$\theta = \tan^{-1}(f_x^2 + f_y^2) \tag{3}$$

The slope is thresholded according to the specifications and a binary map is generated with the flat and rough patches labeled as 1's and 0's respectively. An imbalanced or sparse binary map populated primarily with 0's or 1's is an outlier for training and hence, the training set was constrained to have between $25 - 75\%$ flat areas. The obtained map represents a high resolution slope map of the terrain. A median filter is then applied of appropriate size to remove obstacles from the map which are smaller than the size given in the mission specifications. A minimum filter of an appropriate size is then applied which ensures that the landing spot is away from the hazardous areas on the surface. Figure 1 shows two examples of the sampled DEM and the binary map showing safe and hazardous areas.

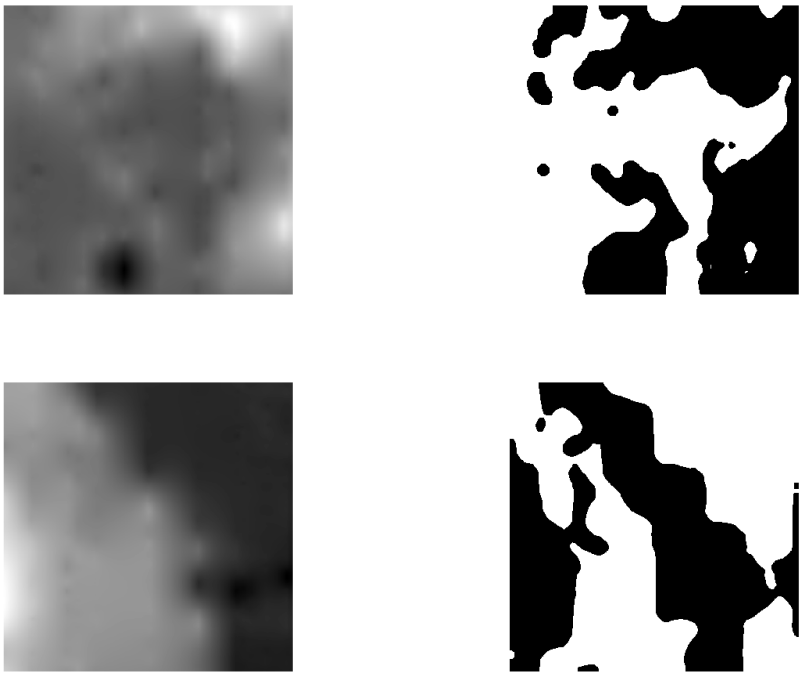

Fig. 1 The above figures exemplifies the training data set containing the sampled DEM on the left as the input and the processed DEM mask on the right as the output. The safe areas are in white and hazardous ones are in black.

B. Data Augmentation

A data augmentation technique is applied for increasing the size and diversity of the training dataset. An output preserving input transformation is applied to the input data and the transformed input data is augmented to the training set. Another type of data augmentation used applies the same geometric transformation to the inputs and outputs and augments the data to the training dataset. We use the fast image augmentation technique called Albumentations [19] in

our method. Using data augmentation allows the trained network to be robust to errors in the DEM and allows the network to learn from a rich dataset.

IV. Learning Framework

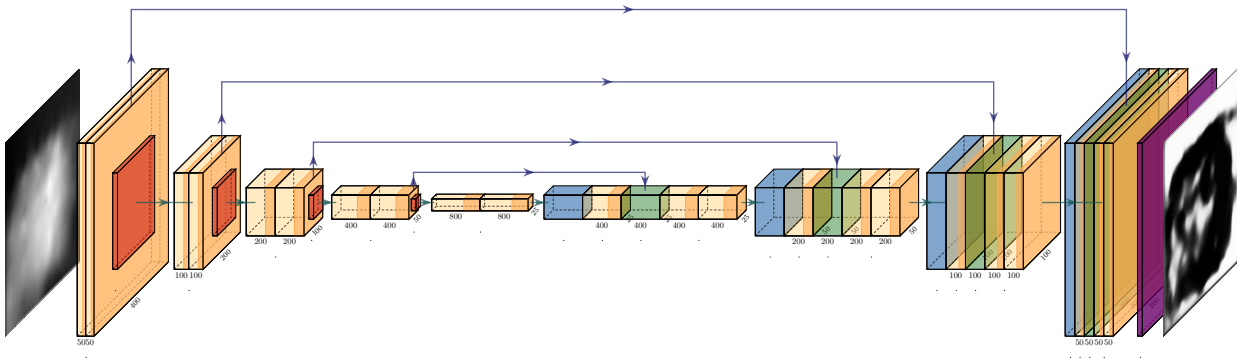

Fig. 2 The figure above describes the topology of the network. Yellow layers: convolutional layers with ReLU activation, Red layers: Max Pooling layers, Blue layers: Upsampling layers, Green layers with connections: Concatenations or skip connections, and Purple Layer: Sigmoid Layer.

A fully connected convolutional neural network is constructed. The architecture used in this framework is similar to the UNet Neural Networks used for biomedical representation [20]. The idea behind using the UNet architecture is to reduce the dimensionality of the input DEM and retain the essential data from the DEM using the first half of the UNet, and then projecting this essential information back onto the DEM. The size of sampled DEMs, $(400, 400)$, is the input and the binary map generated above is the output to the neural network. The input DEM is first centered and normalized before training since only the relative values of the elevations affect the measure of safety of a landing spot. All the layers expect for the final layer use the Rectified Linear Unit (ReLU) as the activation function [5]. Since, the output is to constrained between 0 and 1, a Sigmoid activation layer is applied at a last layer to that effect. The network also includes residual or skip connections between lower and higher layers wherein a lower layer is concatenated as is to a higher layer of the network [21]. This facilitates free flow of gradients without passing through the nonlinear layers. Dropout layers are included within the model to prevent overfitting [22]. The loss function used for training is the Dice loss $\mathcal{L}(\cdot, \cdot)$ [23] defined as

$$\mathcal{L}(Y_{pred}, Y_{true}) = 1 - \frac{2\ sum(Y_{pred} \& Y_{true}) + s}{sum(Y_{pred}^2 + Y_{true}^2) + s} \tag{4}$$

wherein $s = 1$ in our case, allows for unstable data which are mostly made up of zeros. Here, $sum(\cdot)$ represents the sum of all the elements of the matrix. Note that this function counts the number of matching hazardous pixels from the predicted output to the ones from the ground truth. The value of the Dice loss lies between 0 and 1 for all values of the inputs. This is a popular choice of loss function for training semantic segmentation models.

We use the Adam optimizer [24] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for training the network. The validation set is created in a similar way as the training set and the network is trained till the validation loss stops improving. Early stopping was used to stop the training when the loss on the validation set stopped improving for 7 straight epochs. The network is evaluated on the testing set sampled randomly from the large DEM and the Dice coefficient metric is used for evaluating the performance. The Dice coefficient, which is 1 − Dice loss, determines the degree of similarity between the predicted and the ground truth.

V. Results

The accuracy and loss history for the training and validation set is given in the Figure 3. The dice loss and coefficient were used to evaluate the loss and accuracy respectively. A few example outputs from the testing phase are given in Fig. 4. The dice coefficient for the testing set was evaluated with the ground truth for 100 sampled DEMs. The mean dice coefficient was found to be 0.83 ± 0.05, where a value of 1 denotes perfect matching of the ground truth and the predicted.

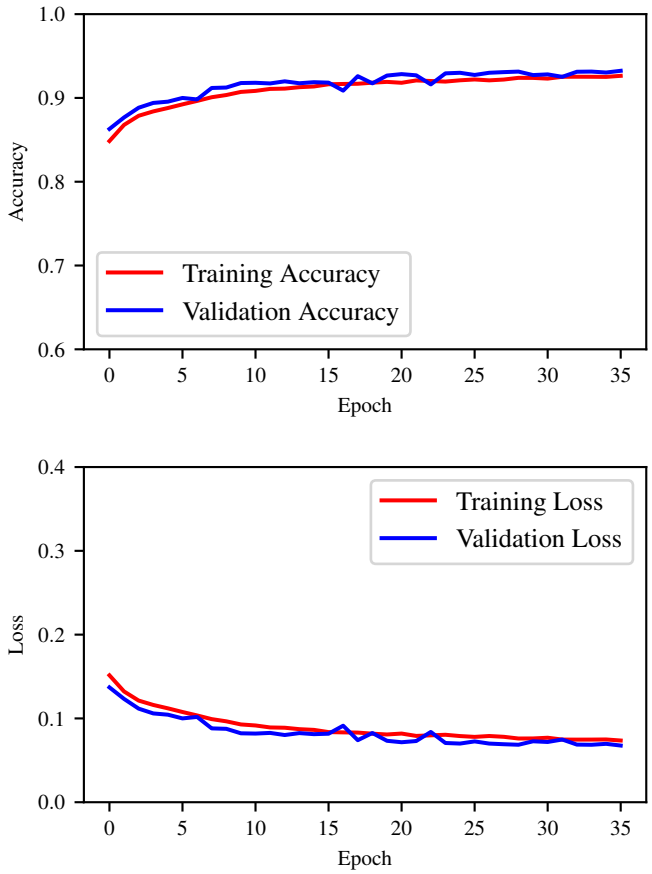

Fig. 3 The dice loss coefficient and the dice loss respectively on the training and validation set.

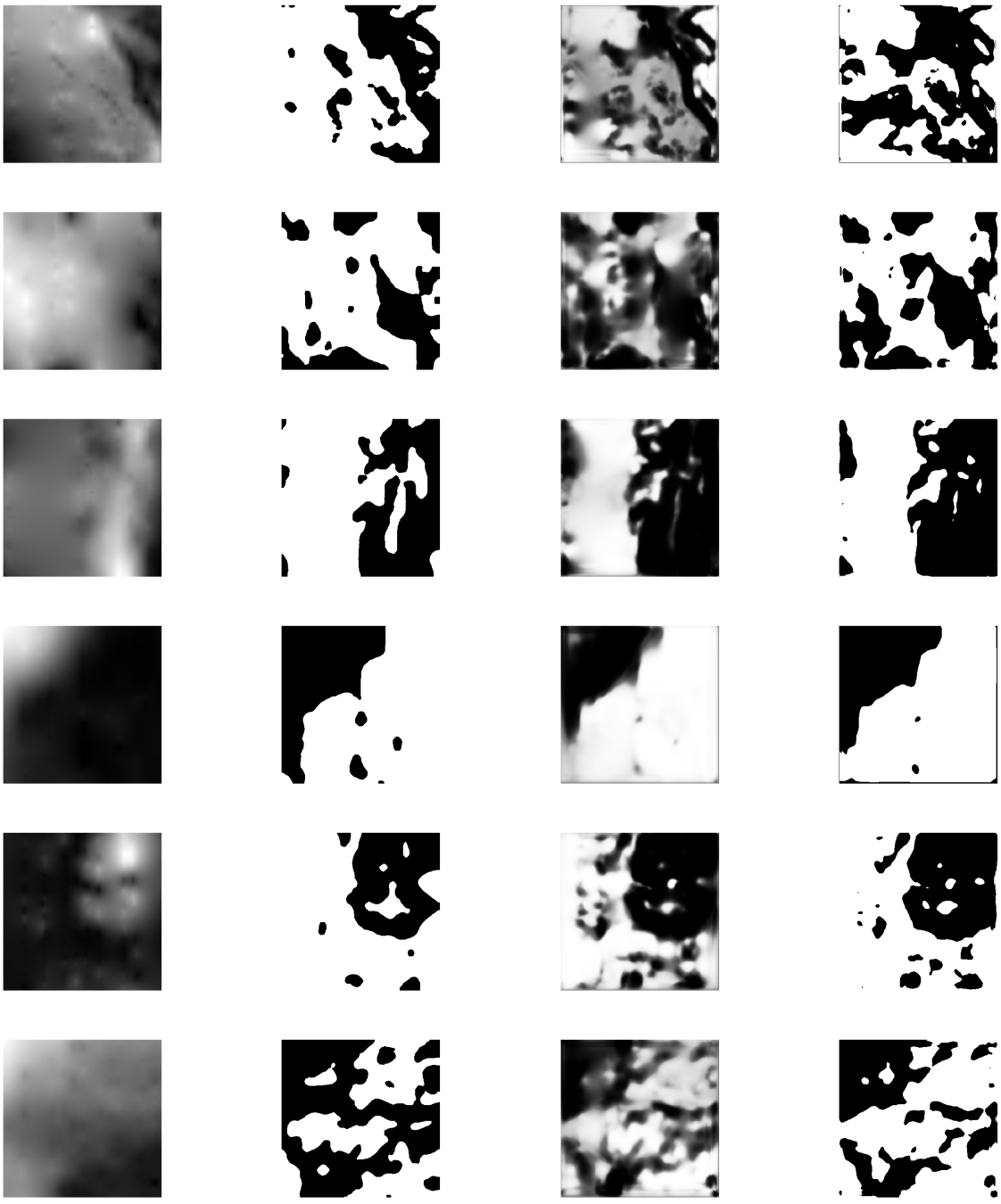

Fig. 4 The columns above denote the sampled DEM, the processed ground truth, the network output and the thresholded output are given in the figure. The third column generated by the learnt neural network is thresholded to give the fourth column. It is important to note that the values at which the network outputs are thresholded maybe different in each case.

VI. Conclusion

A deep learning algorithm called semantic segmentation popularized in computer vision applications is used for identifying safe landing spots on the Lunar surface. The digital elevation map (DEM) from the Lunar Reconnaissance Orbiter (LRO) is used to train a convolutional neural network (CNN) to learn the probability of landing spot being hazardous. Existing algorithms are used to analyze the DEM for find safe landing spots which satisfy the mission specifications. A UNet-like neural network architecture is trained to learn the safe landing spots given the DEM as the input. After testing the trained CNN on the randomly sample DEMs from the Lunar surface, it was found that the CNN gives an average accuracy of 83%. The Dice coefficient metric was used for comparison between the predicted and the ground truth. Future work includes choosing a single best safe landing spot which is close to the intended landing spot. This may be done by either by clustering the CNN output or by including the intended landing spot in the training itself.

References

[1] Epp, C., Robertson, E., and Carson, J. M., "Developing Autonomous Precision Landing and Hazard Avoidance Technology from Concepts through Terrestrially Flight-Tested Prototypes," *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 0324.

[2] Brady, T., and Schwartz, J., "ALHAT system architecture and operational concept," *2007 IEEE Aerospace Conference*, IEEE, 2007, pp. 1–13.

[3] Johnson, A. E., and Montgomery, J. F., "Overview of terrain relative navigation approaches for precise lunar landing," *2008 IEEE Aerospace Conference*, IEEE, 2008, pp. 1–10.

[4] Amzajerdian, F., Pierrottet, D., Petway, L., and Vanek, M., "Development of lidar sensor systems for autonomous safe landing on planetary bodies," *International Conference on Space Optics—ICSO 2010*, Vol. 10565, International Society for Optics and Photonics, 2017, p. 105650M.

[5] Nair, V., and Hinton, G. E., "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[6] Bulyshev, A., Pierrottet, D., Amzajerdian, F., Busch, G., Vanek, M., and Reisse, R., "Processing of three-dimensional flash lidar terrain images generating from an airborne platform," *Three-Dimensional Imaging, Visualization, and Display 2009*, Vol. 7329, International Society for Optics and Photonics, 2009, p. 73290I.

[7] Yan, B., Wang, Y., Feng, L., Zhou, H., and Jiang, Z., "Terrain Matching Based on Adaptive Digital Elevation Map," *2018 International Conference on Advanced Control, Automation and Artificial Intelligence (ACAAI 2018)*, Atlantis Press, 2018.

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

[9] Cohen, J. P., Lo, H. Z., Lu, T., and Ding, W., "Crater detection via convolutional neural networks," *arXiv preprint arXiv:1601.00978*, 2016.

[10] Di, K., Li, W., Yue, Z., Sun, Y., and Liu, Y., "A machine learning approach to crater detection from topographic data," *Advances in Space Research*, Vol. 54, No. 11, 2014, pp. 2419–2429.

[11] Wang, Y., and Wu, B., "Active Machine Learning Approach for Crater Detection From Planetary Imagery and Digital Elevation Models," *IEEE Transactions on Geoscience and Remote Sensing*, 2019, pp. 1–13. doi:10.1109/TGRS.2019.2902198.

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

[13] Johnson, A., and Ivanov, T., "Analysis and testing of a lidar-based approach to terrain relative navigation for precise lunar landing," *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6578.

[14] Cheng, Y., Clouse, D., Johnson, A., Owen, W., and Vaughan, A., "Evaluation and improvement of passive optical terrain relative navigation algorithms for pinpoint landing," *Spaceflight Mechanics*, Vol. 140, 2011.

[15] Trawny, N., Huertas, A., Luna, M. E., Villalpando, C. Y., Martin, K., Carson, J. M., Johnson, A. E., Restrepo, C., and Roback, V. E., "Flight testing a real-time hazard detection system for safe lunar landing on the rocket-powered morpheus vehicle," *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 0326.

[16] Steiner, T. J., and Brady, T. M., "Vision-based navigation and hazard detection for terrestrial rocket approach and landing," *2014 IEEE Aerospace Conference*, IEEE, 2014, pp. 1–8.

[17] Riris, H., Sun, X., Cavanaugh, J. F., Ramos-Izquierdo, L., Liiva, P., Jackson, G. B., Schmidt, S., McGarry, J., and Smith, D. E., "The lunar orbiter laser altimeter (LOLA) on NASA's lunar reconnaissance orbiter (LRO) mission," *Conference on lasers and electro-optics*, Optical Society of America, 2008, p. CMQ1.

[18] Zhou, Q., and Liu, X., "Error analysis on grid-based slope and aspect algorithms," *Photogrammetric Engineering & Remote Sensing*, Vol. 70, No. 8, 2004, pp. 957–962.

[19] A. Buslaev, E. K. V. I. I., A. Parinov, and Kalinin, A. A., "Albumentations: fast and flexible image augmentations," *ArXiv e-prints*, 2018.

[20] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[21] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition. eprint," *arXiv preprint arXiv:0706.1234*, 2015.

[22] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, Vol. 15, No. 1, 2014, pp. 1929–1958.

[23] Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., and Cardoso, M. J., "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *Deep learning in medical image analysis and multimodal learning for clinical decision support*, Springer, 2017, pp. 240–248.

[24] Kingma, D. P., and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.