DETECTION AND INITIAL ASSESSMENT OF LUNAR LANDING SITES USING NEURAL NETWORKS

Daniel Posada, Jarred Jordan, Angelica Radulovic, Lillian Hong, Aryslan Malik, and Troy Henderson

Robotic and human lunar landings are a focus of future NASA missions. Precision landing capabilities are vital to guarantee the success of the mission, and the safety of the lander and crew. During the approach to the surface there are multiple challenges associated with Hazard Relative Navigation to ensure safe landings. This paper will focus on a passive autonomous hazard detection and avoidance sub-system to generate an initial assessment of possible landing regions for the guidance system. The system uses a single camera and the MobileNetV2 neural network architecture to detect and discern between safe landing sites and hazards such as rocks, shadows, and craters. Then a monocular structure from motion will recreate the surface to provide slope and roughness analysis.

INTRODUCTION

With NASA’s recent emphasis on lunar missions, both private companies and space agencies are planning missions to go back to the Moon. There is a growing need for space hardware and software capable of guaranteeing mission success. The first missions to return to the Moon since Apollo are part of the Commercial Lunar Payload Services (CLPS) awards by the National Aeronautics and Space Administration (NASA). Then, following the Artemis program, astronauts will land with the recently awarded Human Landing System (HLS) spacecraft. There are an expected 90 missions to the Moon by the year 2030. These missions undergo many challenges for landing; to illustrate the difficulty, some recent attempts that failed are the Indian Space Research Organisation Vikram and the Israel Aerospace Industries Beresheet lunar lander. Landing a spacecraft on an unknown surface with unknown hazards is risky; therefore, the terminal phase of any space exploration mission is a critical moment. This stage will likely be the closest the lander will be to experience any physical hazard as it is performing a touchdown maneuver.

There are multiple active and passive sensors that can provide awareness for this purpose. For example Light Detection and Ranging (LiDARs) which are active sensors provide high fidelity mapping however they are still being designed for space, are expensive, power-hungry, and bulky.

---

*Ph.D. Candidate, Space Technologies Lab, Embry-Riddle Aeronautical University, 1 Aerospace Blvd., Daytona Beach, FL, 32114.
†Undergraduate, Space Technologies Lab, Embry-Riddle Aeronautical University, 1 Aerospace Blvd., Daytona Beach, FL, 32114.
‡Graphics Programmer, Intuitive Machines, 3700 Bay Area Blvd, Houston, TX, 77058.
§Visiting Assistant Professor, Space Technologies Lab, Embry-Riddle Aeronautical University, 1 Aerospace Blvd., Daytona Beach, FL, 32114.
¶Associate Professor, Space Technologies Lab, Embry-Riddle Aeronautical University, 1 Aerospace Blvd., Daytona Beach, FL, 32114.
Cameras on the contrary, are great passive sensors due to the amount of information they provide, and their footprint on a vehicle is smaller. However, the downfall of these two sensors is the amount of data to be processed, and interpretation is as good as the processing algorithms. Additionally, cameras require light (either natural or provided) in order to operate, potentially limiting future landing site areas.

An image is composed of millions of pixels that translates into processing very large arrays, and performing any computer vision task implies operating over this amount of data. To provide awareness of the desired landing sites, a deep learning architecture was selected due to its benefits in processing data sets like images in a fast fashion by reducing dimensions. MobileNetV2 is a well-known deep learning architecture implemented for multiple classification, detection, and segmentation tasks. This network was designed and optimized to be used in a mobile environment providing a useful portable method without the need of powerful computation. This pipeline provides an easy solution to be implement in robotic spacecraft lander due to cost, optimization, and size. This paper will assume a small lunar lander with a Vehicle Footprint Dispersion Ellipse (VFDE) of 10 meters. In other words, the size of the spacecraft and the GNC error is accounted for within this footprint. The VFDE also provides valuable information as it will provide the minimum area to be analyzed that it is considered safe for the landing.

HAZARD DETECTION AND AVOIDANCE

There has been some previous Hazard Detection and Avoidance (HDA) development efforts, but only a few have really been implemented taking into account that any HDA implementation is unique to the landing requirements, available sensors, and computational requirements of the mission. The only current active implementation of an HDA development was used for the landing of Mars 2020 Perseverance Rover. This landing combined their Terrain Relative Navigation (TRN) technology with a camera that resized the imagery to process it in time against a pre-loaded map that required time for development using the Mars Reconnaissance Orbiter. These systems require more testing but they will become more common with the increase in missions. Other newer approaches take use the footprint of the hazards of landing site and the geometry of the spacecraft, but require significant computational capabilities. Previous research and current state-of-the-art mostly assume that LiDAR technology is ready for spaceflight; however, this active sensor is still being developed for this application (power consumption, mass, and volume) and has a long way to be tested. Therefore, passive sensors such as cameras are still one the most viable sensors for this application with the known disadvantage of the sensitivity to the light conditions.

One of the biggest challenges of performing Hazard Relative Navigation (HRN) is to be able to perform HDA in almost real-time. Being able to provide the Guidance, Navigation, and Control (GNC) system with a waypoint for landing requires multiple steps. Classical computer vision methods require significant computational power in order to process the amount of data and the operations performed in these arrays require time, for this reason images are down-sampled (resized) to improve speed at the expense of accuracy. An example is detection and segmentation of rocks, shadows, and craters. These task requires three independent operations tailored to the properties of each hazard which requires intensive computations. However, the use of MobileNetV2 brings a direct operation on the image that can detect all of these hazards at the same time. This is done in an efficient way with great accuracy by mapping the features detected from the image into different higher dimensions to extract their unique features making it more robust to different lighting con-
ditions, as it employs less hyper-parameters and can be run on computationally-constrained mobile hardware providing almost real-time inference which makes it an effective auxiliary tool for safe landing. Other approaches to this problem using semantic segmentation and reinforced learning have been proposed recently. These papers illustrate the challenges of a mobile environments and provides an alternative on the use of NNs to solve the problem.21,22

Once the Neural Network (NN) detects the hazards, a mask can be created to map areas that are good or bad. Good area selection can be augmented by subdividing the mask using a Quadtree algorithm to quickly identify areas away from the hazards. The approximate size of the pixels can be determined using the camera intrinsics to verify that area of those regions of interest is greater than or equal to the VFDE. Then, a classical structure from motion pipeline using the attitude and the camera parameters can be used to build a point cloud. This point cloud can be segmented using the hazard mask, to verify slope and roughness in the areas denominated good by fitting a plane using a simple least squares approach. An example of the hazard mask can be seen in Figure 1.

![Figure 1. Example of hazard mask (rocks and craters) to filter safe landing sites.](image)

The sparse point cloud is generated by matching and triangulating features.23 These features are obtained by finding unique identifiers in the image using a feature detection algorithm. Multiple known methods such as SIFT, SURF, Fast, are used in robotics; however, for this approach ORB was used due to its feature detection, descriptor resilience, and computation speed.24 Once these features are identified in each region of interest, they can be triangulated using epipolar geometry mapping the pixel space into the 3-D space. Figure 2 shows an example of the point cloud reconstruction on right side and the lunar surface from the imagery acquired. The reconstruction is trapezoidal due to the camera geometry and it can be projected from the 3-D ILS frame into the camera frame and then into the camera frame using the projection matrix and the camera intrinsics. This is a well known problem in multiple view geometry and many examples can be found in the literature.25

Next step is to fit plane to measure the slope of the desired region. A simple least squares can be used to find the plane coefficients and therefore the vector that can provide the slope w.r.t the local gravity vector.10

$$G\tilde{p} = \tilde{d}$$
$$\tilde{p} = (G^T G)^{-1} G^T \tilde{d}$$
$$[a \ b \ c]^T = (G^T G)^{-1} G^T \tilde{d}$$

where $G$ is the point cloud matrix, and $d$ is a unitary column vector.
Once the coefficients are obtained, the angle $\beta$ between the local gravity vector $\left( g = [0 \ 0 \ 1]^T \right)$ and the normal vector $\vec{p}$ is obtained using the dot product rule:

$$\beta = \cos^{-1} \left( \frac{\vec{p}^T \cdot \vec{g}}{||\vec{p}|| \cdot ||\vec{g}||} \right)$$

$$\beta = \cos^{-1} \left( \frac{|c|}{\sqrt{a^2 + b^2 + c^2}} \right)$$

(2)

If the the angle is larger than $90^\circ$ it can be corrected as:

$$\beta = 180^\circ - \beta$$

The least squares fitting error that represents the roughness of the surface $d_e$ can be calculated as:

$$d_e = \frac{|ax_k + by_k + cz_k - 1|}{\sqrt{a^2 + b^2 + c^2}}, \quad k = 1, 2, \ldots, n$$

(3)

Figure 2. Left: Lunar surface. Right: Point cloud reconstructed using structure from motion.

Figure 3. Examples of plane fit on a region of interest.
Finally, a cost function can be minimized to rank the candidate sites based on the area, roughness, and slope. This can be augmented with the fuel ellipse depending on how much fuel the spacecraft will have when the algorithm is used. For a safety factor it is also assumed that the spacecraft is able to withstand a slope of $10^\circ$ before tilting and hazards up to 30 cm. The values in the cost function are normalized between 0 and 1, where closer to 0 is ideal and 3 is the worst.

$$J = \frac{10m^2}{\text{Area}} + \frac{\text{Slope}}{10^\circ} + \frac{\text{Roughness}}{0.3m}$$

(4)

DATABASE

To generate the imagery of the Database a trajectory for landing on the Moon was assumed with a specific path for performing site detection. Figure 4 illustrates an operational trajectory for HDA, which begins at 400 meters altitude and 400 meters downrange from the intended landing site (ILS). This trajectory allows approximately 5 seconds for image acquisition, and an additional 10 seconds for computation and relaying sites to the GNC system. Therefore the need for fast mobile implementation.

![Figure 4. Ideal landing trajectory with respect to the intended landing site. The green highlight represents image collection for HDA. White represents the time HDA has to process and provide landing sites, as well as contingency for more image acquisition if needed.](image)

**Camera Parameters**

A pinhole camera is assumed for the purpose of this imagery as none of the images use a wide field of view (FOV) lens to limit the amount of radial and tangential distortion. The sensor is assumed to be a 5 MP camera with a FOV of 45°, resolution of 2592x1944 pix, and it is canted 45° from the Nadir axis of the spacecraft looking directly at ILS in the lunar surface. The camera
parameters are:

\[
K = \begin{bmatrix}
7363.60 & 0 & 1295.5 \\
0 & 7363.60 & 971.5 \\
0 & 0 & 1
\end{bmatrix}
\] (5)

This information can be used to estimate a scale that relates the meters per pixel to identify areas greater than or equal based on the pixel location on the image.

\[
\text{angLocation} = \frac{\text{pixLocation}}{\text{numPixels} \times FOV}
\]

\[
\rho = \frac{\text{altitude} \times \cos(\text{camAngle} - \frac{\text{FOV}}{2} + \text{angLocation})}{\text{FOV}}
\]

\[
res = \frac{2\sqrt{\rho^2 - \text{altitude}^2 \tan(\frac{\text{FOV}}{2})}}{\text{numPixels}}
\] (6)

where \text{altitude} is the spacecraft z component w.r.t the ILS, \text{camAngle} is the canted angle, \text{angLocation} is the angular location of the pixel within the FOV, \text{pixLocation} is the desired pixel location, \text{numPixels} is the number of pixels along the image direction, \rho is used to compute distance from Nadir to \text{pixLocation}, and \text{res} is the final scaling resolution in meters per pixel to convert the side of the region of interest. For the intent it is assumed that the scale of the region of interest is the same and not going to vary much from the actual centroid.

**Synthetic Images**

Training and validation of the model depends on the database. Therefore, the synthetic imagery for testing HDA algorithms is built using high resolution synthetic digital elevation maps (DEMs), physically-based materials, artificial hazards, and dynamic lighting conditions. A 40 x 40 km height map with 0.5 mpp (meters per pixel) resolution is imported into Unreal Engine 5 (UE5) and placed in the defined lunar-centered lunar-fixed (LCLF) location and orientation for rendering. Physically based materials are applied to the model using UE5’s material blueprint shader graph to mimic a realistic lunar regolith. The lunar surface material is composed of an RGB base color and a roughness parameter that controls light reflectivity of the surface. These DEMs are enhanced for the analysis and hazards such as rocks and craters are added and modeled using known mathematical distribution. Some examples of this database can be seen in Figure 5.

For assessing lunar landing site several classes were identified: rocks, shadows, craters, and safe spots. Images were picked random with different contrasts of a database of more than 300 images. Images were then labeled using free software labelImg using the selected classes “rocks” and “crater”. where these features were outlined by bounding boxes. An example of a tagged image from the database is demonstrated in the Appendix in Figure 14.

**EMBEDDED SYSTEMS NEURAL NETWORKS**

One of the main challenges of choosing or designing a NN for a space application is the size of the network, the number of hyper-parameters, portability, and the compute capability for a computationally-constrained mobile hardware. Currently available radiation hardened hardware is still many generations back having sometimes just one core and no parallel processing. Some of the state-of-the-art mobile NN for single image detection are YOLOV5 “You-Only-Look-Once” (Recently V6 and V7 were released), Single Shot Detectors (SSD) based, and Feature Pyramid
Network (FPN) based NNs.\textsuperscript{28} Each NN has a trade-off between accuracy and latency. To achieve accuracy, portability, and speed MobileNetV2 NN was chosen as the core network (Figure 6) to decompose the image.\textsuperscript{29} This architecture can be utilized for a variety of tasks, including classification, segmentation, and detection; however, the latter is the emphasis of this study. Then an SSD and FPN NN architecture is used to augment MobileNetV2 to identify the hazards and extract their features at different scales using a pyramid levels. This network has been tested before effectively in embedded systems with an effective inference time and a simple deployment.\textsuperscript{30,31}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{mobilenet.png}
\caption{MobileNetV2 NN architecture before SSD and FPN.}
\end{figure}

\section*{TRAINING AND VALIDATION}

MobileNetV2 network was training is done using the custom database described in the previous section. The NN training is tweaked to take the image at full-resolution and resize it to 1024 but retain the aspect ratio of the original image. Then a batch size of 2 was chosen in order to not rush the convergence of the learning and improve accuracy on detection over 50000 steps. The training was conducted using a TensorFlow-Python workflow on a workstation with two Intel Xeon Silver 4214 processors, an NVIDIA Quadro K6000, and 96 GB of system RAM. Figure 7 shows the decrease in the different cost functions proving that the model prediction becomes better to detect the hazards. For validation, the network was evaluated in the computer it was trained as well on a Raspberry Pi.
4B+ using the augmented dataset. The scripts are written in python using TensorFlow’s version 2.4 and OpenCV’s version 4.5.

Figure 7. Training output of cost functions. Total Loss, Classification Loss, Regularization Loss, and Learning Rate.

Table 1. Approximate average time for detection inference for a 100 runs.

| Validation Hardware                  | Average Inference Time (sec) |
|--------------------------------------|------------------------------|
| Desktop PC                           | 3.223                        |
| Intel Xeon + Quadro K6000            | 3.223                        |
| Raspberry Pi 4B+                     | 2.867                        |
RESULTS

After training Figure 8 and Figure 9 display the results of the NN running inference on the desktop computer and the Raspberry Pi 4B+ respectively. In most of the detection cases the network was able to identify almost every hazard, some small rocks can be passed as noise depending on the texture of the surface. However, this could be addressed by increasing the dataset with more contrast enhancements and more brightness conditions. A stress case with more than 250 hazards was tested on the Raspberry Pi to test inference time. The right image from Figure 9 illustrates this stress case where in the far FOV the rocks are not detected, but inference happened in $\approx 2.9$ secs. Once the detections are obtained, a binary mask is generated as seen in Figure 10, which is then augmented using the Quadtree algorithm (Figure 11). This mask and augmentation runs in $\approx 100$ msecs as the mask is binary and easier to process. When the mapping is done, ideal candidates based on the area are selected. To obtain the approximate size the of the area, Equation (6) is used and yields an approximate value of 128 pix for a 10 m side VFDE.
These areas are used to delimit the point cloud to analyze for slope locally on each region of interest. Figure 12 illustrates a plane fitted to the point cloud from the top left corner with a slope of $\approx 2.3$ deg.
Figure 13 illustrates all the safe sites in the location in close and mid FOV. The green circles are the VFDE that can be fitted into the desired regions; however, the real area identified in the Quadtree is the one used in a cost function that minimizes and ranks the sites based on area, roughness, and slope. The larger and cleaner the area, the safer it is for the spacecraft to land.

CONCLUSIONS AND FUTURE WORK

Using NN for detecting hazards is useful in constrained embedded environments. MobileNetV2 particularly is a good architecture for this type of devices as it can process almost full-resolution imagery. A Raspberry Pi which is a low-cost, low-power embedded device is able to generate a complete solution for safe landing in the lunar surface. The network training can be improved with an increased database spanning a set that is able to generalize landing conditions not only on the lunar environment, but also in asteroids, and other planets such as Mars or Venus. The current implementation relies on python and could be faster if properly optimized in C++. This system can be easily deployed and used with a simple of-the-shelf-component to augment the safety of a spacecraft when landing. The authors encourage the space community to be more open to test and use mobile embedded systems for this type of applications. Future work includes the generation of a larger dataset, optimization of code in lower level programming language such as C++, use GPUs to decrease processing times, and testing of more embedded hardware such as NVIDIA portfolio of mobile platforms.
APPENDIX: ADDITIONAL FIGURES

Figure 14. labelImg is used for labeling the database.

Figure 15. Additional output of the NN for different imagery running inference on the desktop computer. Green boxes are detections of rocks and shadows, Yellow boxes are detections of craters.
REFERENCES

[1] G. Daines, “Commercial Lunar Payload Services,” Mar 2019.
[2] A. Balossino and F. Davarian, “The Plan To Give The Moon Decent Wireless Coverage,” Jan 2022.
[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
[4] C. D. Epp and T. B. Smith, “Autonomous precision landing and hazard detection and avoidance technology (ALHAT),” 2007 IEEE Aerospace Conference, IEEE, 2007, pp. 1–7.
[5] E. S. Crane, Vision-Based Hazard Estimation during Autonomous Lunar Landing. PhD thesis, Stanford University, 2014.
[6] M. Yu, H. Cui, and Y. Tian, “A new approach based on crater detection and matching for visual navigation in planetary landing,” Advances in Space Research, Vol. 53, No. 12, 2014, pp. 1810–1821.
[7] X. Jiang, S. Li, and T. Tao, “Innovative hazard detection and avoidance strategy for autonomous safe planetary landing,” Acta Astronautica, Vol. 126, 2016, pp. 66–76.
[8] K. Iiyama, K. Tomita, B. A. Jagatia, T. Nakagawa, and K. Ho, “Deep reinforcement learning for safe landing site selection with concurrent consideration of divert maneuvers,” arXiv preprint arXiv:2102.12432, 2021.
[9] K. Tomita and K. Ho, “Sequential Terrain Mapping and Processing for Online Hazard Detection via Deep Generative Models,” AIAA SCITECH 2022 Forum, 2022, p. 2207.
[10] D. Posada, “An Open Source, Autonomous, Vision-Based Algorithm for Hazard Detection and Avoidance for Celestial Body Landing,” Master’s thesis, Embry–Riddle Aeronautical University, 2020. Master’s Thesis. 534.
[11] J. Getchius, D. Renshaw, D. Posada, T. Henderson, S. Ge, and G. Molina, “Hazard Detection And Avoidance For The Nova-C Lander,” 2021 AAS/AIAA Astrodynamics Specialist Conference, 2021, pp. August 8–12, Big Sky, USA.
[12] C. Y. Villalpando, A. E. Johnson, R. Some, J. Oberlin, and S. Goldberg, “Investigation of the tilera processor for real time hazard detection and avoidance on the altair lunar lander,” 2010 IEEE Aerospace Conference, IEEE, 2010, pp. 1–9.
[13] A. E. Johnson, J. A. Keim, and T. Ivanov, “Analysis of flash lidar field test data for safe lunar landing,” 2010 IEEE Aerospace Conference, IEEE, 2010, pp. 1–11.
[14] S. B. Aaron, Y. Cheng, N. Trawny, S. Mohan, J. Montgomery, H. Ansari, K. Smith, A. E. Johnson, J. Goguen, and J. Zheng, “Camera Simulation For Perseverance Rover’s Lander Vision System,” AIAA SCITECH 2022 Forum, 2022, p. 0746.
[15] Y. Cheng, A. Ansar, and A. Johnson, “Making an Onboard Reference Map From MRO/CTX Imagery for Mars 2020 Lander Vision System,” Earth and Space Science, Vol. 8, No. 8, 2021, p. e2020EA001560.
[16] J. D. Nelson and H. Schaub, “Landing Site Selection Using a Geometrically Conforming Footprint on Hazardous Small Bodies,” Journal of Spacecraft and Rockets, 2021, pp. 1–11.
[17] A. E. Johnson, A. R. Klumpp, J. B. Collier, and A. A. Wolf, “Lidar-based hazard avoidance for safe landing on Mars,” Journal of guidance, control, and dynamics, Vol. 25, No. 6, 2002, pp. 1091–1099.
[18] R. Moghe and R. Zanetti, “A Deep learning approach to Hazard detection for Autonomous Lunar landing,” The Journal of the Astronautical Sciences, Vol. 67, No. 4, 2020, pp. 1811–1830.
[19] F. Amzajerdian, D. Pierrottet, L. B. Petway, G. D. Hines, V. E. Roback, and R. A. Reisse, “Lidar sensors for autonomous landing and hazard avoidance,” AIAA Space 2013 Conference and Exposition, 2013, p. 5312.
[20] V. Roback, A. Bulyshev, F. Amzajerdian, and R. Reisse, “Helicopter flight test of 3D imaging flash LIDAR technology for safe, autonomous, and precise planetary landing,” Laser Radar Technology and Applications XVIII, Vol. 8731, International Society for Optics and Photonics, 2013, p. 87310H.
[21] K. Tomita, K. Skinner, K. Iiyama, B. Jagatia, T. Nakagawa, and K. Ho, “Hazard detection algorithm for planetary landing using semantic segmentation,” ASCEND 2020, 2020, p. 4150.
[22] K. Tomita and K. Ho, “Adaptive Hazard Detection and Avoidance for Planetary Landing via Bayesian Semantic Segmentation,” AIAA SCITECH 2022 Forum, 2022, p. 2207.
[23] P. Lindstrom, “Triangulation made easy,” 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 1554–1561.
[24] E. Karami, S. Prasad, and M. Shehata, “Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images,” arXiv preprint arXiv:1710.02726, 2017.
[25] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[26] B. Li, Z. Ling, J. Zhang, and J. Chen, “Rock size-frequency distributions analysis at lunar landing sites based on remote sensing and in-situ imagery,” *Planetary and Space Science*, Vol. 146, 2017, pp. 30–39.

[27] R. Ginosar, “Survey of processors for space,” *Data Systems in Aerospace (DASIA). Eurospace*, 2012, pp. 1–5.

[28] F. Sultana, A. Sufian, and P. Dutta, “A review of object detection models based on convolutional neural network,” *Intelligent Computing: Image Processing Based Applications*, 2020, pp. 1–16.

[29] Y.-C. Chiu, C.-Y. Tsai, M.-D. Ruan, G.-Y. Shen, and T.-T. Lee, “Mobilenet-SSDv2: An improved object detection model for embedded systems,” 2020 *International conference on system science and engineering (ICSSE)*, IEEE, 2020, pp. 1–5.

[30] D. Posada, J. A. Jordan, A. Malik, A. Radulovic, J. J. Wang, S. Rhodes, C. Buhler, and T. Henderson, “Autonomous dust detection and removal for space surface vehicles navigation and exploration awareness,” *Sensors and Systems for Space Applications XV*, Vol. 12121, SPIE, 2022, pp. 87–97.

[31] J. Jordan, D. Posada, D. Zuehlke, A. Radulovic, A. Malik, and T. Henderson, “Satellite Detection in Unresolved Space Imagery for Space Domain Awareness Using Neural Networks,” *arXiv preprint*, 2022.