CRATER DETECTOR EVALUATION USING ARTEMIS I IMAGERY

James S. McCabe[1*]; Sofia G. Catalan[2]; [1]Spacecraft Navigation Senior Discipline Expert (EG6, Aeroscience and Flight Mechanics Division, NASA Johnson Space Center, 2101 E NASA Parkway, Houston, TX), [2]Graduate Research Assistant (University of Texas at Austin, Austin, TX 78712). [*][james.s.mccabe@nasa.gov]

Abstract. *Artemis I was launched on November 16th, 2022 and the Space Launch System successfully placed the Orion spacecraft on a trans-lunar trajectory. Orion subsequently performed a lunar flyby, entered a distant retrograde orbit, performed another lunar flyby, and then successfully returned home with a Pacific splashdown on December 11th. During these flybys, Orion's optical navigation camera collected imagery of the lunar surface, and this paper conducts a qualitative performance analysis of a NASA-developed crater detector on a subset of the collected images.*

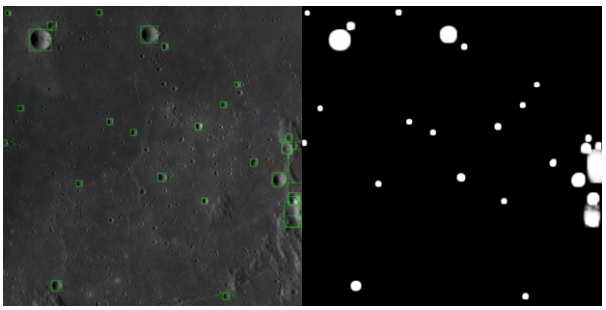

Figure 1. Example detected image with bounding boxes (left) and Mask R-CNN's binary mask (right) for an LROC image.

Background. The abundance and distribution of impact craters throughout the lunar surface enable the development of crater-based terrain relative navigation solutions. These craters are observable targets from a wide range of altitudes as a function of an observing camera's resolution. However, the primary challenge in using craters for TRN lies in the variation in how they appear in imagery, where their shape, size, surrounding terrain are all affected by the local lighting conditions while also being subject to the observer's altitude and attitude relative to the craters, and the camera model and exposure. Because of these data variations in how craters can be captured in images, previous works have provided different approaches on how to build a crater detector suitable for TRN.[1,2]

A robust crater detection capability would enable TRN in various flight phases, from high altitudes that may include highly eccentric, off-nadir images of craters, to lower altitudes that may focus on nearly circular craters for hazard avoidance. To handle the expected variations in what craters can look like, this paper further develops a neural network model that can be iteratively trained with thousands of crater image samples. The goal of this work is to process imagery taken during the Artemis I flight, using the OpNav camera. With the variations in crater shapes, sizes, projections, and lighting conditions, the detector model is able to better handle the real imagery after improvements were made to the neural network training pipeline through the use of simulated images.

This paper describes enhancements to the crater detector described in Ref. 3, primarily consisting on the addition of new training data produced by a ray-traced lunar scene generator. Then, performance of the enhanced detector is qualitatively evaluated on data collected by Orion's OpNav camera during its flybys during Artemis I.

Previous Work Summary. One of the most challenging tasks for neural network-based image processing is to obtain or develop a labeled dataset that contains the images and the associated "true" labels. This labeled dataset is what the neural network iterates on and allows the current model to check itself against the truth, updating the internal weights and biases of the network to more accurately detect the desired objects. Previous work[3]focused on the development of an automated labeling pipeline that overlays the Robbins crater catalog[4] to the LROC Global Morphologic Maps.[5] The previous development effort focused on enabling a user to set configurations for what kinds of craters are included in the labeled dataset through catalog parameters like min/max crater diameter and eccentricity. Through testing, the detection performance significantly improved by computing metrics like Shannon entropy and area in shadow for the crater samples which led to the inclusion of additional parameters that the user can control. By enabling a parametric capability to preprocess the dataset to include "good" craters, the resulting labels used to train the detector can be easily adjusted by a user in an automated manner.

After generating the labeled dataset, a general image detection method can be trained to learn what a "crater" is and how they can appear in images. This work relies on the Mask Region-based Convolutional Neural Network (Mask R-CNN),[6] which provides a set of detection outputs: a bounding box containing the rectangular region where a crater is located and a binary mask containing the "crater" or "not crater" label for each pixel (an example of which is given in Fig. 1).

An advantage of Mask R-CNN is its ability to detect multiple objects of the same type in an image so an image containing multiple craters can be processed. Following iterative training cycles, the resulting detector model was then evaluated for centroiding accuracy and tested for a variety of synthetic and real flight im-

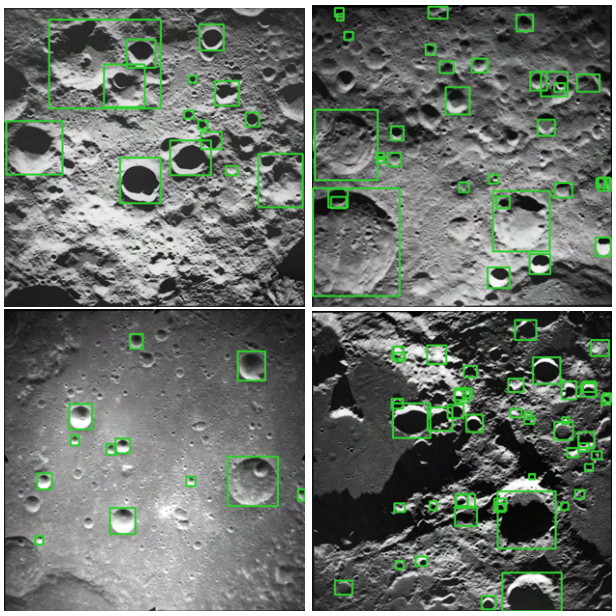

Figure 2. Apollo 16 crater detection samples (produced by the previous 2020 model in Ref. 3).

agery (such as the Apollo 16 images depicted in Figure 2). The crater detctor was subsequently utilized to assess filter performance in an extended Kalman filter (EKF) paired with a crater identification algorithm,[3] an identification-free anonymous feature processing (AFP) navigation problem,[7] and further integrated estimation tests.[8,9] This document serves as a sequel to Ref. 3 to demonstrate model improvements that have been made recently.

Updates to Crater Detector. If the input dataset used for training contains all of the expected image variations that the model is expected to handle, the developed detector can provide high confidence detections when given an image it has never seen before. To robustly handle the variations in data that is expected of real mission imagery, the detector must be trained to handle these different conditions. The previous work relied on image crops of the LROC maps and the Robbins catalog was projected onto the flat images to generate the automated labels. This process did not account for the approximately spherical projection of the image and the catalog onto the Moon and assumed that the camera would always be nadir pointing to the surface. While the previous work focused on near-equatorial regions of the Moon, where the inaccurate projections were assumed to minimally cause problems for the following filter tests, they instead resulted in issues resulting in filter bias and poor identification performance.[8,9]

Limitations in image generation capabilities and issues with correctly handling the map projections resulted in focusing on handling these errors in post-processing. However, a better approach is to use an accurate image rendering tool, supplemented with the correct projection of the Robbins catalog, with a configurable camera model that allows a user to set the position and attitude of the observer. With the use of a Blender-based image rendering tool for lunar imagery (soon to be released for open-source use, hopefully early 2025), the Robbins crater catalog is projected onto the spherical Moon surface, and the automated labeling pipeline is continued from previous work. The updated, automated label generation pipeline is preserved to generate large image batches for training, again continuing the use of user-based parametric control for crater sizes, eccentricity, shadow, and entropy.

The Blender-based image generation pipeline comprises of the key update to the Mask-R CNN detector, and it massively improved the crater detector's performance via the ability to generate off-nadir imagery at varying lighting conditions. Examples of the images used to train the network can be seen for varied camera poses in Fig. 3 and for varied lighting conditions in Fig. 4. So, images were generated at varied poses, labeled/annotated utilizing the aforementioned automation tool, and the labeled results were paired with the images to improve the network training.

Instead of training a new model entirely from scratch to focus only on the Blender-based images, the updated model builds from the previous 2020 model through PyTorch's interface to Mask R-CNN. PyTorch tools enable further training from an existing model such that the detector is updated to include the Blender dataset while retaining the weights and biases computed from the LROC images. Through the use of this interface, the detector development pipeline is configured to train with any input dataset to support including other datasources in the future, from simulated imagery using Blender, Unreal Engine, or other rending tools, images from derived output products from LRO, or real imagery from future lunar missions. This input-independent property of neural network-based tools is another beneficial element that can enable the generalized development of a crater detection model.

Performance Results. The performance of the detector models on the Artemis I OpNav images were used as the primary comparison benchmark, since the models were not trained on these images at all. Craters can be seen in the Artemis I images from Flight Day 06 and Flight Day 20, which correspond to the approximately closest approaches of the Orion spacecraft to the lunar surface, however, the performance of the detector models depend on how well their respective input data captures the crater variations seen in these image sets. For clarity, the detector model trained only with the LROC images will be referred to as the "2020 model" while the model trained with LROC and the Blender images will be called the "2024 model." Note that the Mask R-CNN framework outputs a confidence metric which is between zero and one, with increasing confidence reflecting the network's internal belief in a positive classification. It is

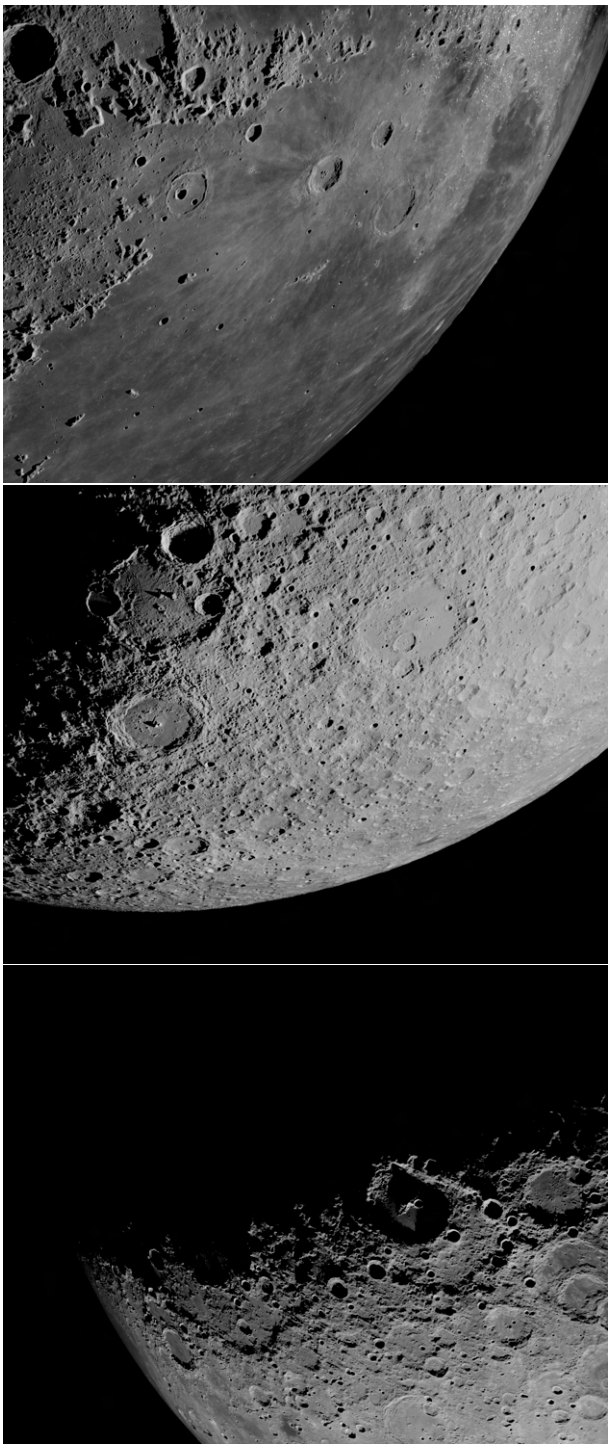

Figure 3. Sample images of varying pose within the Blender-based image pipeline used to update the crater detector.

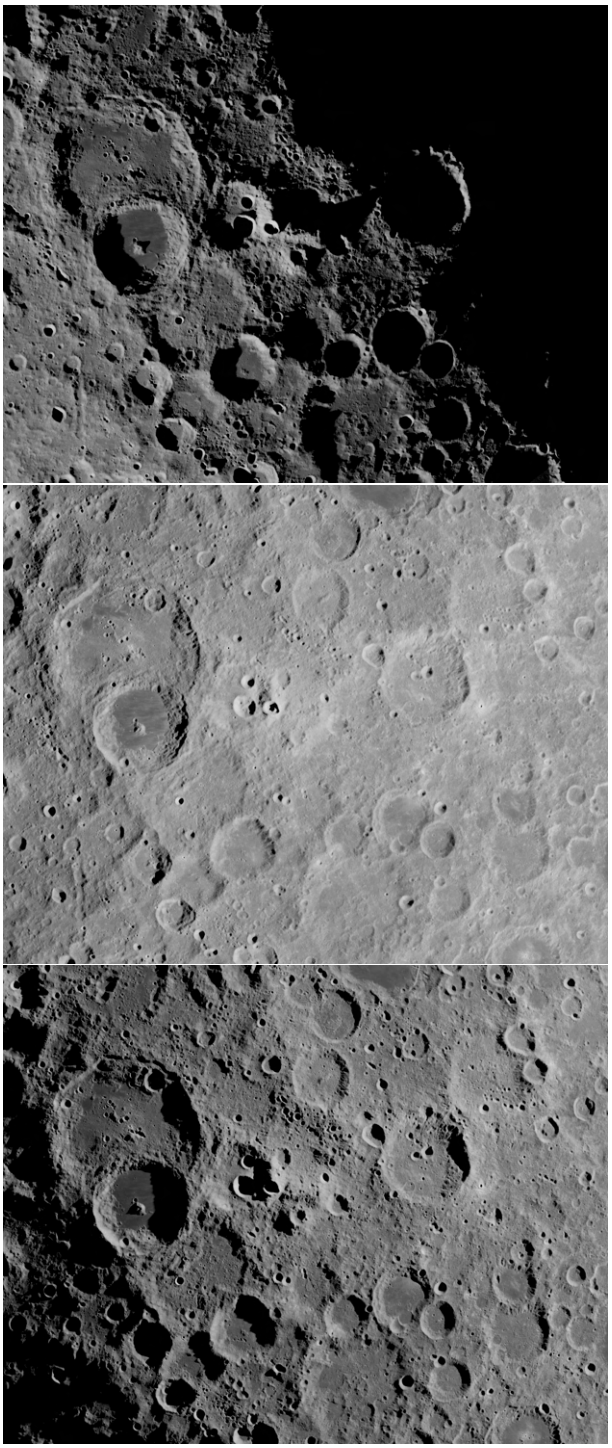

Figure 4. Sample images of varying lighting conditions within the Blender-based image pipeline used to update the crater detector.

logical, and useful, to truncate detections below a given threshold before providing them to a navigation filter.

Flight Day 20. Images from Flight Day 20 were exposed well such that the bright and dark regions of the lunar surface are captured. The detection results for these images, comparing the 2020 and 2024 model, can be seen in the following figures:

- Figure 5: detection rate versus confidence.
- Figure 6: detection rate versus image index.
- Figure 7: sample image for all detections with a confidence exceeding 0.0.
- Figure 8: sample image for all detections with a confidence exceeding 0.5.
- Figure 9: sample image for all detections with a confidence exceeding 0.8.

The detection count versus network confidence can be seen in Fig. 5. Note that, as is obvious, both trends for the 2020 and 2024 model (in this figure called "Batch 3+4 model") monotonically decrease. While at first it seems that the 2020 model is outperforming the 2024 model due to having a higher detection rate, it will be demonstrated in later results that the higher detection rate of the 2020 model is due to false detections rather than reliable crater detections. Additionally, note that the 2020 model's loss of detection rate is approximately linear across the confidence level, whereas the 2024 model exhibits a "shelf" when confidence exceeds 0.4. Our findings reflect that the improved training has drastically improved detection reliability, and the overall detection performance is very consistent on these images.

Figure 6 provides the detection count against image index for these images, and this makes the performance contrasts even more stark. Note that the 2020 model fails to detect anything in the first 60 or so images, but then quickly ramps up in detection rate. As later images will demonstrate, this is actually due to the presence of many false detection and multiple detections (i.e., assigning multiple detections to different regions of the same crater).

Figures 7, 8, and 9 depict comparison frames for 0.0, 0.5, and 0.8 confidence level thresholds, respectively. In each of these figures, identical images are compared, with the 2020 model on the left and the 2024 model on the right. The performance enhancements afforded via the improvements described previously are immediately apparent, with the 2024 model virtually eliminating the false/multiple detection artefacts exhibited by the 2020 model, even at very low confidence thresholds. Note that in previous work, an assessment of the relationship between confidence and centroiding accuracy was performed to show that with craters with high confidence had lower centroiding errors. For navigation filters relying on the centroids only, using the high confidence craters reduces the measurement error associated with

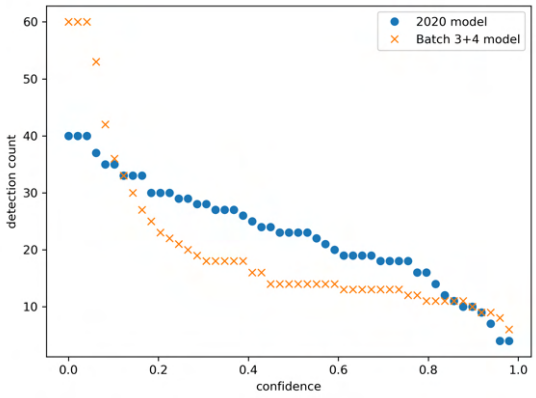

Figure 5. Flight Day 20 detection count versus confidence for the 2020 model and the 2024 model (here called "Batch 3+4 model").

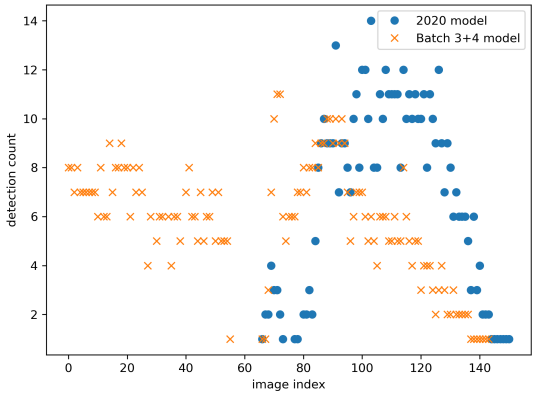

Figure 6. Flight Day 20 detection count versus image index for the 2020 model and the 2024 model (here called "Batch 3+4 model").

the centroid accuracy which would also reduce the resulting state error. As such, the consistent detection performance of the 2024 model suggests it is providing very valuable centroids for navigation processing.

Flight Day 6. Unfortunately, the outbound flyby images collected on Flight Day 6 were overexposed. As described in Ref. 10, the camera settings were adjusted mid-flight to provide the excellent image qualities obtained on Flight Day 20. However, these overexposed images were an interesting stressing case for the 2024 detector model. The compared detection results for a select frame are given in Figs. 10, 11, and 12 for confidence thresholds of 0.0, 0.5, and 0.8, respectively. This is a significantly more challenging set of images to evaluate the detector models with because of the (lack of) contrast between the crater rims, shadows, and the local terrain. Since the LROC images did not include samples that look like these images, the 2020 model did not produce high confidence detections. However, because the Blender images contained samples with different lighting conditions, the 2024 model is able to detect craters

with sufficiently high confidence. Given more samples with difficult lighting that emulate poor exposure, additional training can further improve the performance of the detector model for these kinds of surface images.

Discussion and Future Work. It is apparent from the analysis performed on the Artemis I images collected by Orion's OpNav camera that the updated detector model exhibits drastically improved performance despite all the complications induced by real flight imagery. These results suggest that the detector's output on these flight images provide consistent and desirable centroiding performance that is suitable for navigation operations. In fact, this is demonstrated in Ref. 11, wherein the Flight Day 20 images are utilized to stimulate a navigation filter.

In both image sets, the OpNav camera included horizontal stripes throughout the image that is likely a result of the thermal conditions of the camera. Neither the LROC nor Blender images included these kinds of image artifacts, however, the resulting models entirely disregard these issues and are able to provide detections because of their robust handling for image variations. The detector models expect that the images they are evaluated with are not perfect, however, if the input datasets used to train the models include these image artifacts, it is expected that the detector performance can even more robust. As can be expected for network-driven detectors of this kind, additional training data, if responsibly curated, can be expected to only further improve detection performance.

Additionally, the Blender-based image generation pipeline is altitude-constrained (as is any lunar scene generator), meaning the additional network training described in this paper captures high-altitude crater data. To further improve this network, high-resolution, low altitude imagery will be needed, and this well depend on inclusion of local high-resolution DEMs as they are/as they become available. Fortunately for the evaluation of the Artemis I OpNav imagery, their relative altitudes were sufficiently high to provide promising performance.

The current detector development effort focuses on a generalized capability to detect craters throughout the lunar surface, however the software can be used to generate custom detectors that cater to specific camera models or trajectories. Instead of a general detector, a mission-specific model can limit the set of what craters it expects to observe, however, it would be the user's task to include the expected variations in lighting conditions, camera pose, and exposure in the training dataset. By generating a custom detector model, the performance may be improved since there would be fewer crater shapes and sizes that the model would need to include in its internal weights and biases.

Further, some crater-based navigation tactics that have been appearing in literature intend to exploit not just the crater centers but the shape of the crater themselves. Most commonly, this employs elliptical approximations to the craters in an image to utilize that as a valuable navigation resource. Since the present detector provides an output mask, it is straightforward to provide ellipse approximations for each crater detector. This is currently considered forward work, but it seems worth mentioning as it is a current, though unused, capability of the detector we have described here.

Additional efforts in deploying these detection models in flight software-like environments are required for future use. The current models used GPU-enabled training with CPU-based evaluation on desktop-grade hardware, and while neural networks are often deployed to smaller computing devices, the computational load of this Mask R-CNN model may require efforts toward model optimization to reduce runtime. The authors would like to warn those considering adopting neural-network-based detectors like this away from utilizing the training toolbox's evaluation routines (such as those shipped with PyTorch) in CPU-limited environments. Experience has indicated that these routines induce a tremendous amount of overhead for user-friendliness and input sanitation that is, generally, unnecessary in an onboard context. If one simply employs these, they may deem evaluating the network more CPU intensive than actually required by a hand-built evaluation routine.

Acknowledgments. The authors would like to sincerely thank Mr. Max Marshall for his tremendous assistance with Blender-based lunar scene generation.

References.

[1] L. M. Downes, T. J. Steiner, and J. P. How, "Neural network approach to crater detection for lunar terrain relative navigation," *Journal of Aerospace Information Systems*, vol. 18, no. 7, pp. 391–403, 2021.

[2] E. Emami, G. Bebis, A. Nefian, and T. Fong, "Automatic crater detection using convex grouping and convolutional neural networks," in *Advances in Visual Computing: 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14-16, 2015, Proceedings, Part II 11*, pp. 213–224, Springer, 2015.

[3] S. G. Catalan, J. McCabe, and B. A. Jones, "Implementation of machine learning methods for crater-based navigation," AAS/AIAA Astrodynamics Specialist Conference, AAS AAS 21-519, Aug. 2021.

[4] S. J. Robbins and B. M. Hynek, "A new global database of Mars impact craters $\geq$ 1 km: 1. database creation, properties, and parameters," *Journal of Geophysical Research: Planets*, vol. 117, no. E5, 2012.

[5] E. J. Speyerer, M. S. Robinson, B. W. Denevi2, and the LROC Science Team, "Lunar Reconnaissance Orbiter Camera global morphological map of the moon," 42nd Lunar and Planetary Science Conference, 2011.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[7] J. S. McCabe, "On a higher order method for anonymous feature processing," in *AIAA SciTech 2022 Forum*, 2022.

[8] Z. R. McLaughlin, R. E. Gold, S. G. Catalan, B. A. Jones, and R. Zanetti, "Crater navigation and timing for autonomous lunar orbital operations in small satellites," in *Proceedings of the 44th Annual American Astronautical*

Society Guidance, Navigation, and Control Conference, 2022, 2022.

[9] R. E. Gold, S. G. Catalan, B. Jones, and R. Zanetti, "Extending capabilities of crater navigation and timing for autonomous lunar orbital operations (paper siw22-30)," 2022 Space Imaging Workshop, Oct. 2022.

[10] K. R. Kobylka, "Camera exposure time determination for Artemis I lunar flyby," in *Proceedings of the 4th Space Imaging Workshop*, 2024.

[11] J. S. McCabe, "On non-Gaussian noise in angles-only navigation," in *Proceedings of the 4th Space Imaging Workshop*, 2024.

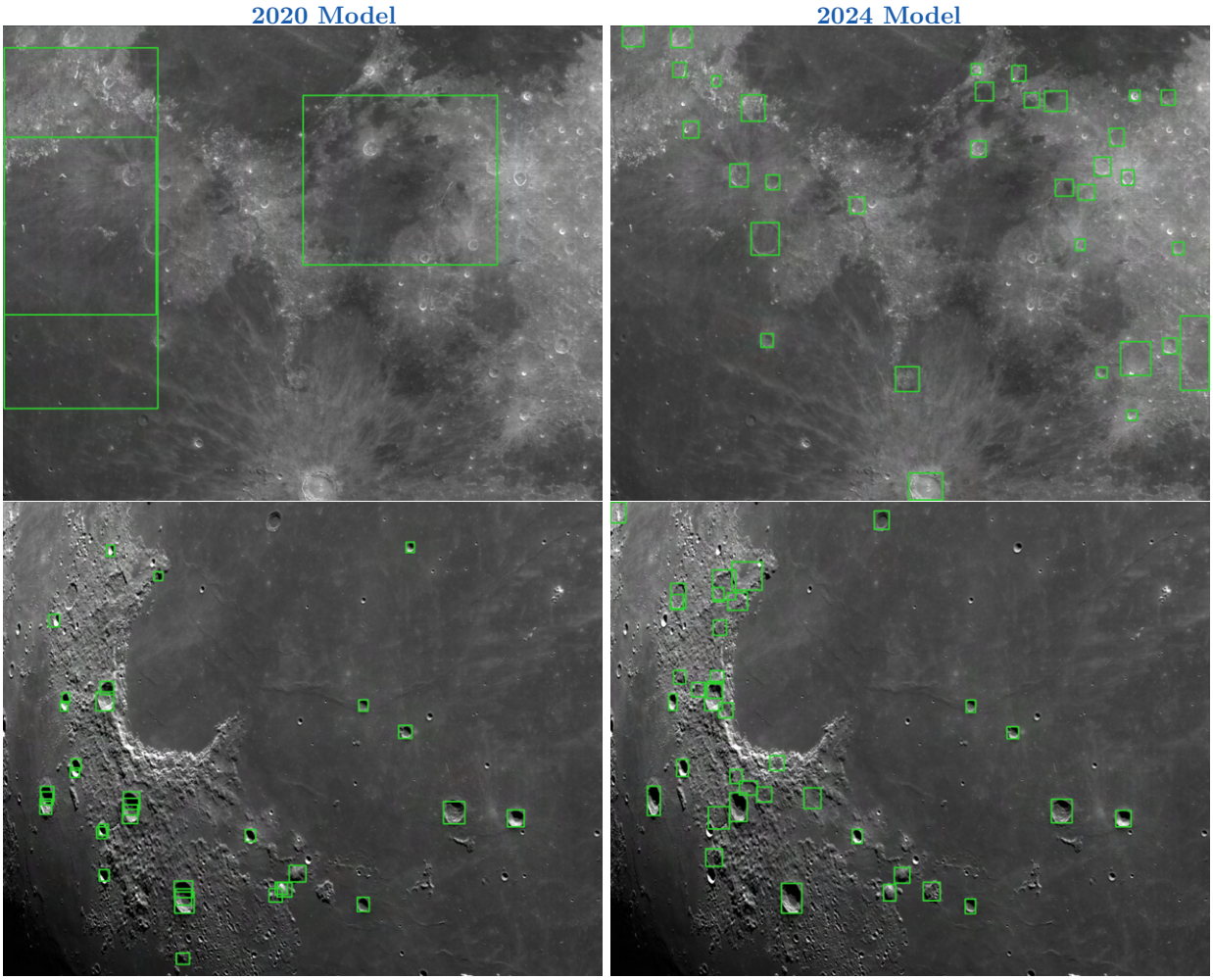

Figure 7. Flight Day 20 detection results, displaying all craters with confidence greater than 0.0 – (left) 2020 model, (right) 2024 model.

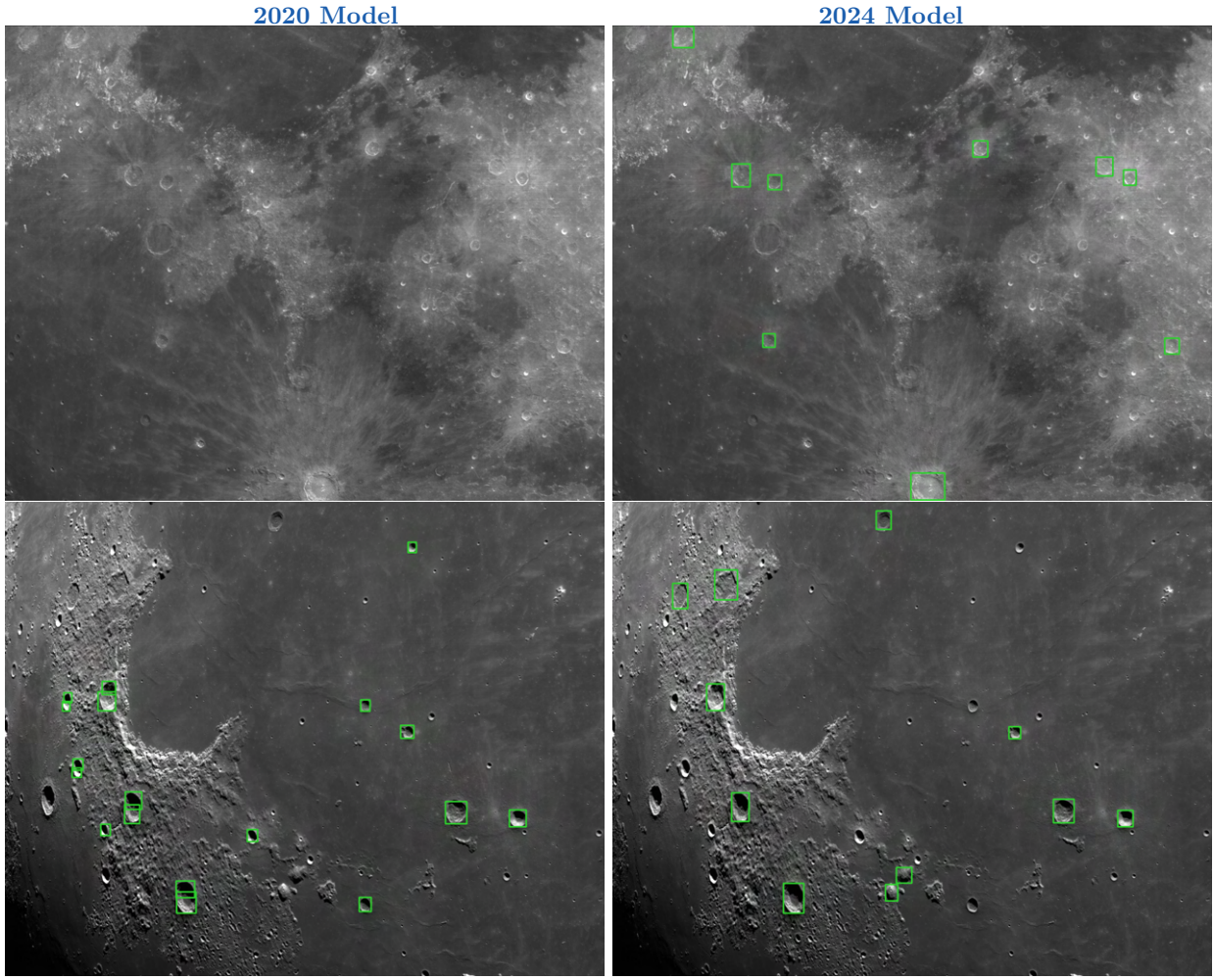

Figure 8. Flight Day 20 detection results, displaying all craters with confidence greater than 0.5 – (left) 2020 model, (right) 2024 model.

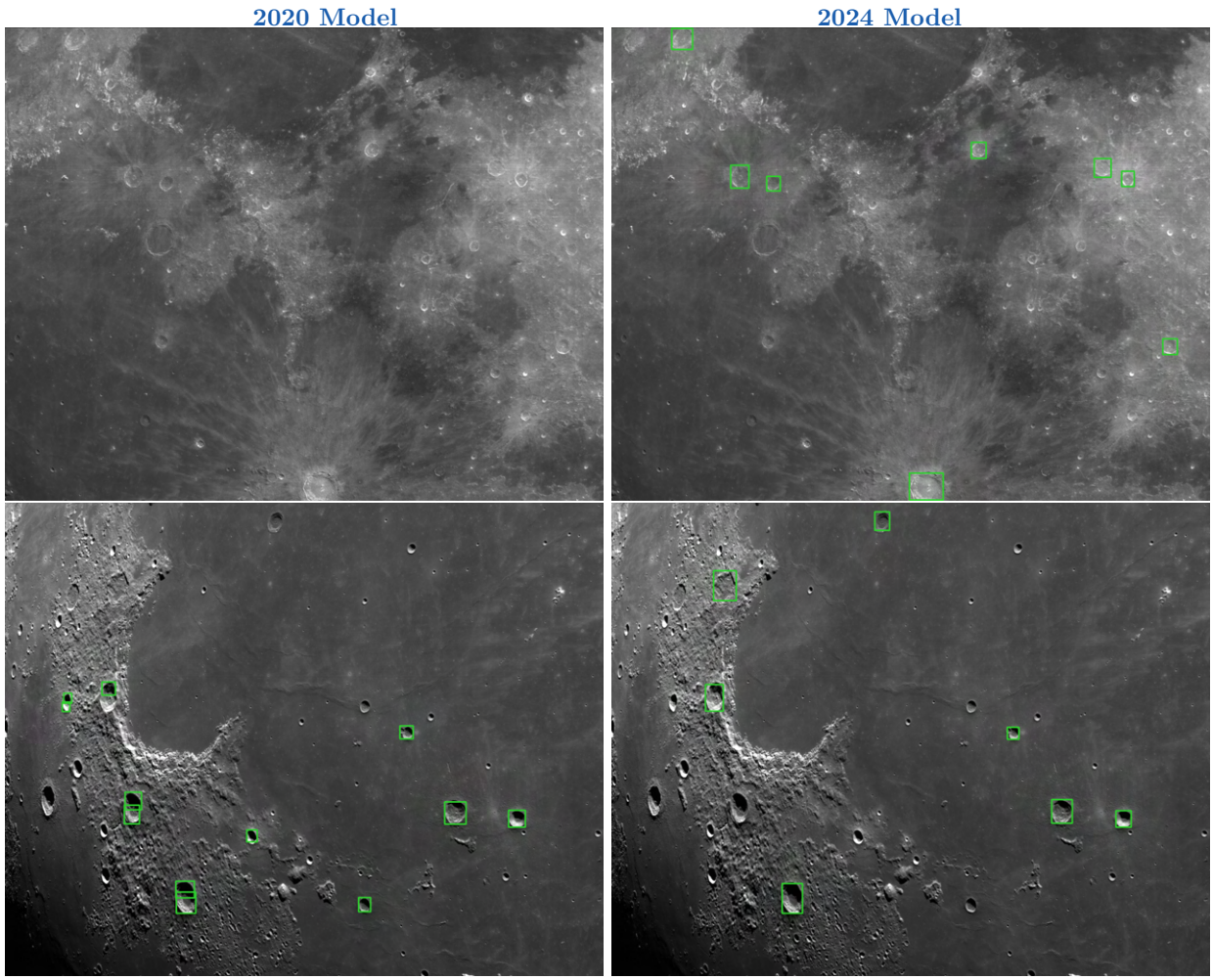

Figure 9. Flight Day 20 detection results, displaying all craters with confidence greater than 0.8 – (left) 2020 model, (right) 2024 model.

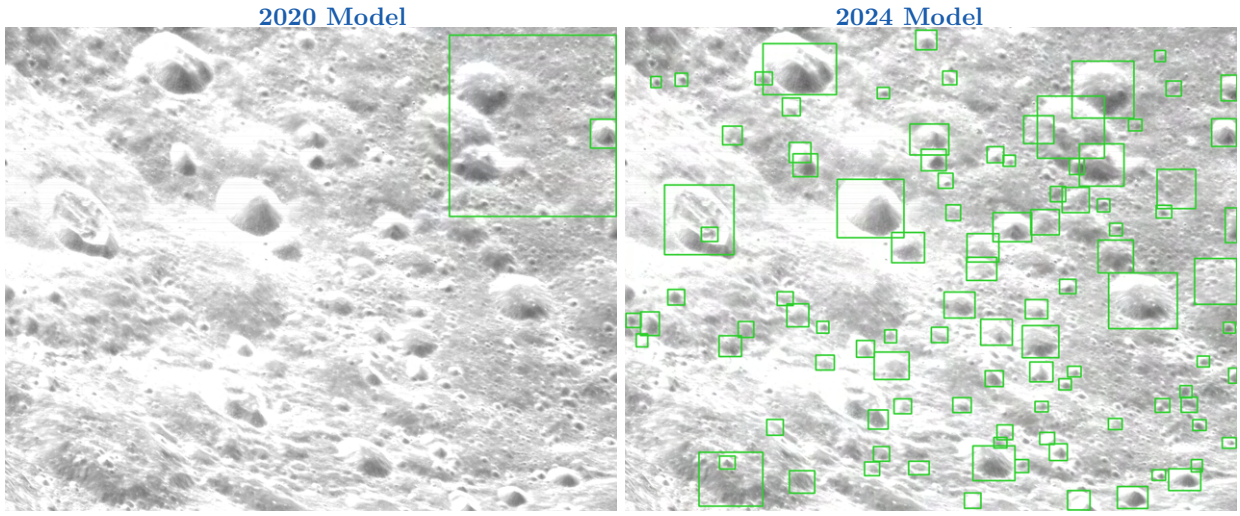

Figure 10. Flight Day 6 detection results, displaying all craters with confidence greater than 0.0 – (left) 2020 model, (right) 2024 model.

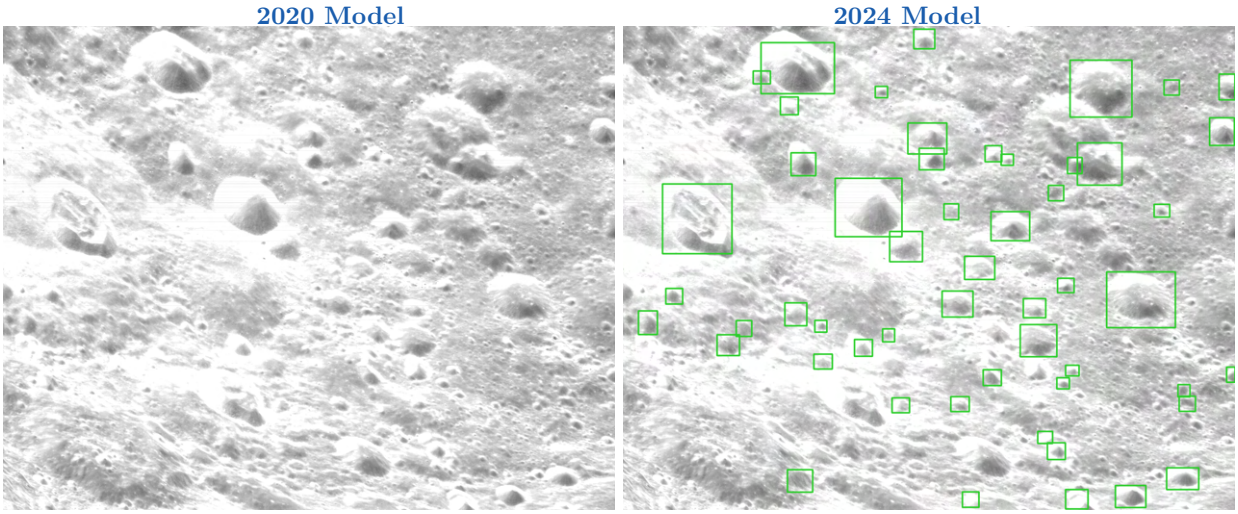

Figure 11. Flight Day 6 detection results, displaying all craters with confidence greater than 0.5 – (left) 2020 model, (right) 2024 model.

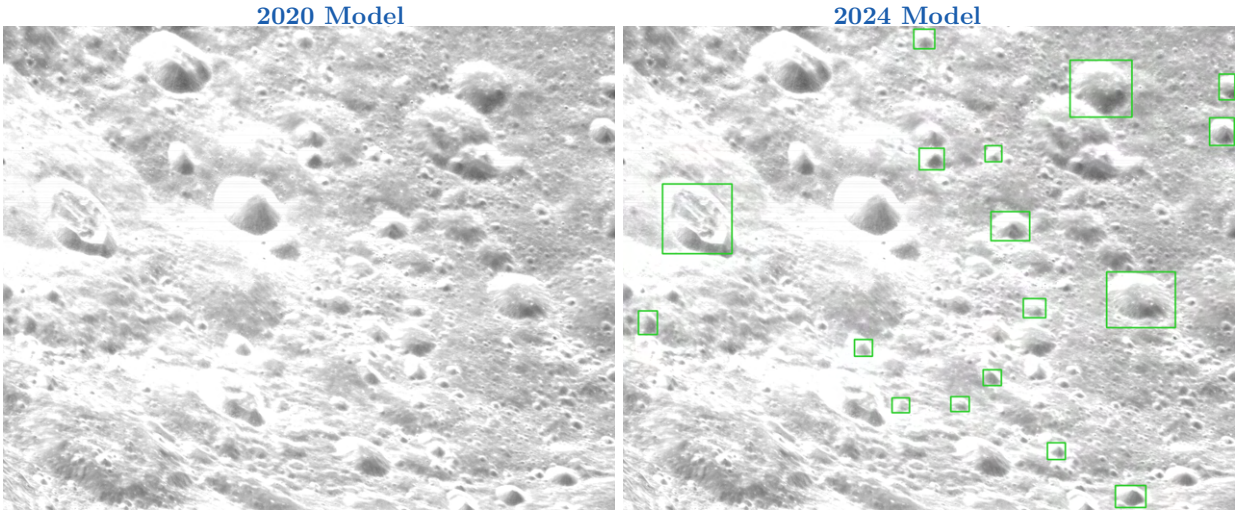

Figure 12. Flight Day 6 detection results, displaying all craters with confidence greater than 0.8 – (left) 2020 model, (right) 2024 model.