Uncertainty-Aware Deep Learning for Autonomous Safe Landing Site Selection *

Kento Tomita †
Georgia Institute of Technology, Atlanta, GA, 30332

Katherine A. Skinner‡
University of Michigan, Ann Arbor, MI, 48109

Koki Ho§
Georgia Institute of Technology, Atlanta, GA, 30332

Hazard detection is critical for enabling autonomous landing on planetary surfaces. Current state-of-the-art methods leverage traditional computer vision approaches to automate identification of safe terrain from input digital elevation models (DEMs). However, performance for these methods can degrade for input DEMs with increased sensor noise. At the same time, deep learning techniques have been developed for various applications. Nevertheless, their applicability to safety-critical space missions has been often limited due to concerns regarding their outputs’ reliability. In response to this background, this paper proposes an uncertainty-aware learning-based method for hazard detection and landing site selection. The developed approach enables reliable safe landing site selection by: (i) generating a safety prediction map and its uncertainty map together via Bayesian deep learning and semantic segmentation; and (ii) using the generated uncertainty map to filter out the uncertain pixels in the prediction map so that the safe landing site selection is performed only based on the certain pixels (i.e., pixels for which the model is certain about its safety prediction). Experiments are presented with simulated data based on a Mars HiRISE digital terrain model and varying noise levels to demonstrate the performance of the proposed approach.

I. Introduction

A utonomous landing is cited as a medium-high priority for future NASA missions [1] [2]. Real-time hazard detection and avoidance (HDA) is a critical component for enabling autonomous landing. The current state-of-the-art method, developed for the Autonomous Landing Hazard Avoidance Technology (ALHAT) program, demonstrated potential for computer vision algorithms to automate online identification of safe or unsafe terrain from input Digital

---

*This paper is a substantially revised version of Paper AAS 21-253 presented at the AAS/AIAA Space Flight Mechanics Meeting, Online, 2021.

†Ph.D. Student, Daniel Guggenheim School of Aerospace Engineering, AIAA Student Member.

‡Assistant Professor, Naval Architecture and Marine Engineering, AIAA Member.

§Assistant Professor, Daniel Guggenheim School of Aerospace Engineering, AIAA Senior Member.
Elevation Models (DEM) [3,14]. However, ALHAT is sensitive to severe sensor noise, navigation error, and missing data, which can lead to error in safe landing site prediction.

In recent years, deep learning has been shown to outperform traditional hand-crafted computer vision algorithms across a wide range of tasks. Recent work has demonstrated the potential for deep neural networks (DNNs) to improve both accuracy and computational cost for onboard hazard detection [15,16]. These prior works take input DEMs to output a binary map labeling each pixel as safe or unsafe for landing. However, one caveat in machine learning methods is their output’s reliability. Most such methods rely on gathering a representative dataset for training, and may not generalize well to test data that is outside of the training distribution. In these cases, a predicted safety map will still be returned without any indication of how reliable that prediction is based on the input DEM.

There has been much recent interest in estimating network uncertainty in deep learning predictions to provide further insight on the reliability of the network prediction [17–21]. However, these works have not been evaluated for autonomous landing on planetary surfaces, and there is a lack of literature on how to translate output uncertainty estimates into actionable decisions for landing site selection.

This work focuses on development and validation of uncertainty-aware deep learning methods for hazard detection for landing on a planetary surface. The specific contributions of this work are (i) to perform semantic segmentation and uncertainty prediction for identifying safe landing sites from input DEMs, (ii) to develop an algorithm that takes in the safety predictions and uncertainty maps to produce an uncertainty-aware (and thus reliable) decision for safe landing site selection, and (iii) to perform experiments for demonstration of developed methods on DEMs with varying sensor noise.

II. Background

A. Autonomous Hazard Detection

The ALHAT program was launched by NASA in 2005 to develop advanced capabilities of spacecraft for autonomous hazard detection and precise landing on lunar and planetary surfaces [4,6,7]. The work developed during the ALHAT program is considered state-of-the-art in autonomous hazard detection for planetary landing. In particular, Ivanov et al. developed a probabilistic method for autonomous hazard detection that considers the vehicle size, the vehicle-surface configuration during landing, and navigation error [5]. The algorithm searches over the potential footpad positions and lander orientations during landing to determine the slope of the lander based on where the footpads fall on the terrain, as well as the roughness of hazards underneath the lander based on the final lander pose. If there are unsafe configurations for the lander at a given point on the DEM, this point is labeled as unsafe for landing. This provides a pixel-wise labeling of safe and unsafe landing points. The final output of the algorithm is a list of proposed safe landing sites, along with their coordinates and safety probability. This work was integrated onboard the rocket-powered Morpheus vehicle and
has been extensively tested in real flight scenarios. Flight tests demonstrated successful autonomous hazard detection and hazard relative navigation capabilities for the Morpheus vehicle over realistic lunar-like terrain [8][9][14]. With simplifying assumptions, the ALHAT algorithm developed by Ivanov et al. is capable of operating online for efficient and effective hazard detection during landing. However, performance of the algorithm can degrade with increased sensor noise, navigation error, and missing data. Building off of the success of the ALHAT project and Morpheus experiments, next generation hazard systems are currently under development through NASA’s Safe & Precise Landing - Integrated Capabilities Evolution (SPLICE) program [22].

B. Machine Learning for Safe Landing on Planetary Surfaces

Deep learning has demonstrated great performance improvements from traditional approaches across a range of tasks. Recent work has focused on developing machine learning-based approaches to aid in guidance, navigation, and control for spacecraft landing on planetary surfaces [23][30]. For scenarios where high resolution maps are available prior to landing, these methods can learn guidance and control policies for navigating to a target landing site. Our work focuses on the perceptual task of hazard detection to determine safe and unsafe landing sites during the landing phase to provide updates to the guidance system. This capability is critical for applications where high resolution maps are not available prior to landing, and for detection of small hazards such as rocks.

There are several prior works that leverage machine learning for detecting craters on the surface of the moon [31][36]. Detected craters can be used for terrain relative navigation (TRN). Our proposed work instead focuses specifically on detecting general hazardous terrain conditions to label each pixel as safe or unsafe for landing. Recently, Moghe and Zanetti developed a DNN for detecting safe and unsafe areas in a DEM for automated hazard detection [16]. Their work develops a novel loss function specifically designed to decrease the false positive (safe) rate, since false positive cases indicate that an area is safe when it is actually unsafe, which can lead to catastrophic situations for autonomous landing operations. Tomita et al. also explored several DNN architectures for real-time hazard detection. In particular, their results demonstrated that a semantic segmentation DNN can outperform the current state-of-the-art ALHAT algorithm for prediction of safe pixels when the input DEM has sensor noise [15]. Our work builds on these prior works to include estimation of network uncertainty for the output binary safety map. Our proposed approach leverages output uncertainty maps to enable uncertainty-aware landing site selection. Furthermore, we provide experiments to evaluate our approach with varying noise parameters for training and testing the network.

C. Uncertainty Estimation

There are two main types of uncertainty that can be modeled: aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty is uncertainty due to sensor noise or motion noise. This type of uncertainty is the main source of uncertainty if the test distribution is well-aligned with the training distribution. Aleatoric uncertainty is efficient to
compute, but predictions with low aleatoric uncertainty may still be incorrect, especially if the test distribution contains samples that fall outside of the training distribution. Epistemic uncertainty is model-based uncertainty in the DNN parameters. This can be estimated through approximate Bayesian inference techniques, such as ensembling [37] or Monte Carlo (MC) dropout [38]. Epistemic uncertainty can be decreased with increasing size and variability of the training dataset. However, it is an important contributor to overall uncertainty when training sets are small or limited in variability, or for safety critical applications. Thus, for the application of autonomous landing on planetary surfaces, it is critical to account for both aleatoric and epistemic uncertainty to ensure that the estimate considers uncertainty due to sensor noise or unseen test cases. Prior work has focused on developing uncertainty-aware semantic segmentation networks [38]. This paper leverages this prior work to enable uncertainty-aware semantic segmentation and safe landing site selection for a planetary lander.

III. Technical Approach

![Proposed approach for uncertainty-aware learning-based landing site selection.](image)

Figure 1 shows an overview of our developed pipeline for learning-based landing site selection. The pipeline takes a noisy DEM as input to the semantic segmentation network. The base network is the Bayesian semantic segmentation network, Bayesian SegNet [38]. Bayesian SegNet leverages MC-dropout in which dropout is included during both training and testing. During inference, this allows for output of both a per-pixel safety prediction (safe/unsafe) and an aligned uncertainty map that indicates how certain the network is in the prediction. The aligned safety and uncertainty
maps are input to the next stage, where our proposed algorithm outputs an actionable decision for safe landing site selection based on both the network prediction and uncertainty map.

**A. Bayesian Deep Learning for Semantic Segmentation**

Figure 2 shows the network architecture used for the semantic segmentation stage. This network architecture is based on the Bayesian SegNet architecture, which leverages MC-dropout to enable uncertainty prediction [38]. SegNet, the base architecture of Bayesian SegNet, is a state-of-the-art network for semantic segmentation [39]. Our prior work demonstrated that SegNet performs well for hazard detection on planetary surfaces [15]. The base structure of SegNet is a convolutional encoder-decoder, which successively downsamples the input until a bottleneck layer, at which point successive upsampling stages are used to output the desired resolution. Each convolutional layer includes convolution, batch normalization, and a rectified linear unit (ReLU) activation function. Convolutional blocks are followed by pooling/upsampling layers to achieve the encoder-decoder architecture. The network output is input to a softmax layer to produce the final output. The network is trained with cross-entropy loss.

![Fig. 2 A schematic of the network architecture for semantic segmentation, which is based on Bayesian SegNet [38].](image)

For MC-dropout, network dropout layers are activated for both training and testing. During test time, each test DEM is input to the trained network $M$ times, where $M$ is the number of MC samples. The final prediction is given by the mean over the softmax output across the $M$ stochastic samples:

$$
\hat{p}(y = c|x, D_{train}) = \frac{1}{M} \sum_{m} p(y = c|x, \hat{w}_m)
$$

(1)

where for output $y$, $p(y = c|x, \hat{w}_m)$ is the softmax probability that input $x$ has class label $c$, given the model parameters of the $m^{th}$ sample, $\hat{w}_m$. To output the final segmentation map, we take the argmax of the mean of the softmax probability
across the $M$ samples.

**B. Uncertainty Estimation**

The uncertainty map can be estimated during test time. There are several methods for computing the uncertainty map given the network output to capture aleatoric and/or epistemic uncertainty. These methods include computing the variance of the output samples, predictive entropy, or mutual information [20]. Reference [20] notes that predictive entropy models both epistemic and aleatoric uncertainty, which are both important for the application of safe planetary landing. Thus, for this work we use predictive entropy to compute uncertainty maps. Given a test sample, $x$, and the training set, $D_{train}$, the predictive entropy, $\hat{H}[y|x, D_{train}]$, is given by [20]:

$$\hat{H}[y|x, D_{train}] = - \sum_c \left( \frac{1}{M} \sum_m p(y = c|x, \hat{w}_m) \right) \log \left( \frac{1}{M} \sum_m p(y = c|x, \hat{w}_m) \right)$$

Figure 3 shows sample output.

![Figure 3](image-url) **Fig. 3** Sample results for safety map prediction and uncertainty estimation.

From left to right, the figure shows sample input noisy DEMs generated from a Mars HiRISE digital terrain model (DTM), the ground truth label produced by running a replicated version of the ALHAT algorithm on an input true DEM without noise, the output network prediction, and the output uncertainty map. The network prediction shows the network output labeling each pixel as safe (blue) or unsafe (yellow). Note that border pixels are considered invalid due to the label generation process. These pixels are ignored during training and evaluation. The uncertainty map is produced through calculating predictive entropy. Low uncertainty (black) indicates that that network prediction can be trusted. High uncertainty (yellow) indicates that the network output may be inaccurate.

It is necessary to select an uncertainty threshold for determining whether the uncertainty value should be labeled as certain or uncertain [20]. Predictions determined to be uncertain should not be trusted. Figure 4 provides an illustrative example to demonstrate the effect of varying the uncertainty threshold to determine whether pixels are certain or uncertain. The best case scenario is to have certain pixels corresponding to accurate predictions. The worst case
scenario is to have certain pixels corresponding to inaccurate predictions. Selecting a conservative threshold leads to many accurate pixels being labeled uncertain, potentially discarding accurate predictions. With a less conservative threshold, more predictions pass through as certain and accurate. However, some predictions appear as certain although they are inaccurate, which could be dangerous for a safety critical application. Selecting the proper uncertainty threshold is a critical consideration for uncertainty-aware safety prediction.

![Conservative and Non-conservative Uncertainty Maps]

**Fig. 4** Effect of varying the uncertainty threshold to determine if prediction is certain or uncertain.

### C. Uncertainty-aware Landing Site Selection

A main contribution of this work is to integrate the output uncertainty map and network prediction to enable uncertainty-aware selection of safe landing sites. The developed pipeline is shown in Fig. 5.

First, a global uncertainty threshold is selected. The uncertainty threshold, $H_t$, is selected to be the mean uncertainty value across the validation set, as suggested in Reference[20] This threshold worked well in practice for our application. Given this uncertainty threshold, $H_t$, labels are overwritten as invalid if the uncertainty value $\hat{H}$ is larger than the threshold $H_t$, or their labels are retained if $\hat{H} \leq H_t$. Selecting a global uncertainty value determined from a set of

![Pipeline Diagram]

**Fig. 5** Pipeline for integrating uncertainty map and network prediction to enable uncertainty-aware selection of safe landing sites.
inputs ensures that input DEMs with low variation of uncertainty values, but relatively high uncertainty, will be labeled as invalid, with no viable safe landing sites.

Once each safety map is thresholded by the global uncertainty threshold, the pipeline continues to select a safe landing site. First, we create a binary safety map that contains the remaining sites that are predicted to be safe and not invalid, as these are the sites that will be considered for safe landing site selection. For these remaining sites, we compute the distance to the closest unsafe or invalid site through the distance transform. The proposed safe landing site is selected as the site with the greatest distance to an unsafe or invalid site to accommodate navigation uncertainty.

IV. Experiments & Results

A. Datasets

Network training and validation is achieved using simulated terrain data for input DEMs and a replicated version of the ALHAT’s hazard detection (HD) algorithm for output safety maps.

1. Simulated terrain data:

Improving from our previous work [40], this work leverages an open-source sensor simulation toolbox, BlenSor, to simulate data collection of realistic terrain with a time-of-flight sensor [41,42]. BlenSor is built upon Blender, an open-source 3D simulation engine. To create the simulation environment, we first load a Mars HiRISE digital terrain model (DTM) into BlenSor using the Blender HiRISE plug-in [43]. This provides a realistic base terrain collected during a real mission for generating simulated data. The original Mars HiRISE DTM is 1 m/pixel resolution. Due to memory limitations, we load the DTM at 25% of the full resolution. Next we configure a time-of-flight sensor to simulate parameters of a real landing scenario. Table 1 provides the sensor parameters of the simulated sensor. These parameters are selected to obtain DEMs with similar dimensions and characteristics of DEMs obtained through real and simulated experiments with FLASH LiDAR surveys for imaging planetary surfaces [22,44–46]. For the simulation, the sensor is moved across the terrain at a fixed altitude of 500 m to collect incremental scans of the terrain. The output from BlenSor is a point cloud. The point cloud is converted to a DEM with a resolution of 1 m/pixel using bilinear interpolation. The final DEM is cropped to a resolution of 100 x 100. BlenSor allows for output of the true (noise-free) sensor scan, or a noisy scan. We generated a low-noise dataset with a 1-\(\sigma\) noise level of 1.67 cm (S-167), a moderate-noise dataset with a 1-\(\sigma\) noise level of 3 cm (S-300), and a high-noise dataset with a 1-\(\sigma\) noise level of 7 cm (S-700). Each dataset uses the same terrain and the same sensor trajectory to ensure that the main difference in the datasets is the noise level. Figure 6 shows the simulation environment during the data generation process, and Fig. 7 shows the sample output terrain obtained through our simulation pipeline.
Table 1  Sensor parameters for generating simulated data in BlenSor \cite{41,42}.

| Parameter                  | Value       |
|----------------------------|-------------|
| Detector size              | 128x128     |
| Horizontal field-of-view   | 12°         |
| Vertical field-of-view     | 12°         |
| Focal length               | 25 cm       |

Fig. 6  Simulation environment for data generation

Fig. 7  Example of generated terrain

2. Label generation:

To prepare ground truth labels of safety maps, we replicated ALHAT’s HD algorithm \cite{5}. ALHAT’s HD algorithm calculates maximum slope and roughness the lander would experience at touchdown by explicitly evaluating the lander geometry and the landing pad contacts to the given terrain. We replicated deterministic slope evaluation and probabilistic roughness evaluation, resulting in the algorithm that returns pixel-wise safety probability. When we need to make safety determination, we adopt a safety threshold of 50% to assign a label of safe or unsafe to each pixel. Note that this process requires evaluation of all the surrounding pixels for every aiming point, so it may not be used online for high-resolution DEM inputs without approximation.

B. Implementation Details

We generated simulated datasets of 1000 DEMs. From this data, 800 DEMs are used for training, 100 DEMs are used for validation, and 100 DEMs are used for testing, selected through random selection. Each training set contains a noised DEM and a label generated with the replicated ALHAT algorithm from the noise-free DEM. The input data is
100x100 resolution. The data is upsampled with bilinear interpolation to 512x512 before being input to the network. The data is also normalized between 0 and 1 based on the minimum and maximum values across the training set.

For training, we use a batch size of 8. The dropout rate for dropout layers is set to 0.5 following Reference [38]. The learning rate is 0.0001 and the momentum is 0.9. The network is trained for 10k epochs. This training takes approximately 2 days on an NVIDIA GeForce RTX 2070 SUPER GPU.

For testing, we use $M = 8$ samples for MC-dropout.

C. Results & Discussion

Figure 8 shows qualitative results of the network prediction together with the baseline performance for a sample DEM with varying levels of noise: low noise of 1.67 cm in 1-$\sigma$ (S-167), moderate-noise of 3 cm in 1-$\sigma$ (S-300), and high-noise of 7 cm in 1-$\sigma$ (S-700). On the top, we shows the true, noise-free DEM and corresponding safety label. We can observe that safe-unsafe border pixels have higher uncertainty, and their area increases as the noise level increases. As our method incorporates the uncertainty map in our final prediction, our uncertainty-aware safety map has decreased error for valid pixels, at the expense of increasing the number of invalid pixels. In other words, our method can cope with the increased sensor noise with the aid of uncertainty map. On the other hand, the baseline prediction from the replicated ALHAT algorithm, which is a state-of-the-art method, degrades with increased sensor noise. The baseline method is conservative, especially for the high noise input, where all pixels are labeled as unsafe and there are no viable safe landing sites. The learning-based approach can still return viable safe landing site candidates, even when the noise of the test sample is outside the distribution of the training set.

Figure 9 shows qualitative results of the uncertainty pipeline for selecting the final safe landing site for a sample input DEM with varying levels of noise: low noise of 1.67 cm in 1-$\sigma$ (S-167), moderate-noise of 3 cm in 1-$\sigma$ (S-300), and high-noise of 7 cm in 1-$\sigma$ (S-700). On the top, we shows the true, noise-free DEM and corresponding safety label. As described in subsection III.C, we propose the landing site with the largest distance to the closest unsafe or invalid site, through the distance transform. Figure 9 shows that the proposed landing site is safe to land given the ground truth of the safety map, even with the high noise level, which is outside the noise distribution at training.

The following section presents quantitative metrics to evaluate the results: pixel accuracy, mean intersection over union, true positive rate, false positive rate, true negative rate, and false negative rate. Each metric is described below.

Pixel accuracy, PA, is computed by

$$PA = \frac{\sum_{i=1}^{k} N_{ii}}{\sum_{i=1}^{k} \sum_{j=1}^{k} N_{ij}}$$

where $k$ is the number of classes, and $N_{ij}$ is the number of pixels of class $i$ denoted as class $j$ [47]. We compute
Fig. 8 Qualitative sample results compared with baseline method for varying levels of input noise.
Fig. 9 Qualitative sample results for uncertainty-aware safe landing site selection for varying levels of input noise.
per-pixel pixel accuracy across all of the pixels in the test set.

Mean intersection over union is a common metric for evaluating semantic segmentation. Intersection over union is computed by

\[
\text{IoU} = \frac{A \cap B}{A \cup B} \tag{4}
\]

where \( A \) denotes the ground truth label and \( B \) denotes the output prediction. The mean intersection over union (mIoU) is defined as the mean IoU across all classes.

Lastly, we provide the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), and false negative rate (FNR). These values are calculated as following:

\[
\text{TPR} = \frac{TP}{TP + FN} \tag{5}
\]

\[
\text{FPR} = \frac{FP}{FP + TN} \tag{6}
\]

\[
\text{TNR} = \frac{TN}{FP + TN} \tag{7}
\]

\[
\text{FNR} = \frac{FN}{TP + FN} \tag{8}
\]

where TP is true positive (safe) count, FP is false positive (safe) count, TN is true negative (unsafe) count, and FN is false negative (unsafe) count. These counts are accumulated across pixels for the full test set and the rates are computed at the end.

Table 2 shows the results for these metrics compared across different methods. The baseline method results are generated from our replicated version of the probabilistic ALHAT algorithm. The initial network prediction obtained through MC-sampling with Eq. 1 is the output safety map without incorporating uncertainty, and the uncertainty-aware prediction is the result of our proposed method to integrate estimated uncertainty and the network prediction to provide a final uncertainty-aware safety map. Our method for uncertainty-aware safety map prediction shows the best performance for pixel accuracy and mIoU across all cases, as invalid (uncertain) pixels that are filtered out tend to be inaccurate. Note that evaluation only considers the valid pixels that are considered certain, so evaluation of our method is based on
Table 2  Quantitative results comparison between the baseline replicated ALHAT method (Baseline), Bayesian deep learning prediction before uncertainty thresholding (Base Net.), and uncertainty-aware Bayesian deep learning prediction (Uncertainty-Aware). V/C pixels stand for valid and certain pixels.

| Method                          | Train Noise 1−σ | Test Noise 1−σ | V/C Pix. % | Pix. Acc. | mIoU | TPR | FPR | TNR | FNR |
|--------------------------------|-----------------|----------------|------------|-----------|------|-----|-----|-----|-----|
| Baseline (Replicated ALHAT)     | –               | 1.67cm         | 100%       | 0.9345    | 0.8294 | 0.7506 | 0.0051 | 0.9949 | 0.2494 |
| Base Net.                       | 1.67cm          | 1.67cm         | 100%       | 0.8672    | 0.7320 | 0.9435 | 0.1579 | 0.8421 | 0.0565 |
| Uncertainty-Aware               | 1.67cm          | 1.67cm         | 64%        | 0.9828*   | 0.9589* | 0.9953* | 0.0221* | 0.9779* | 0.0047* |
| Baseline (Replicated ALHAT)     | –               | 3cm            | 100%       | 0.8702    | 0.6663 | 0.4849 | 0.0033 | 0.9967 | 0.5151 |
| Base Net.                       | 1.67cm          | 3cm            | 100%       | 0.8507    | 0.7070 | 0.9383 | 0.1781 | 0.8219 | 0.0617 |
| Uncertainty-Aware               | 1.67cm          | 3cm            | 63%        | 0.9744*   | 0.9413* | 0.9958* | 0.0344* | 0.9656* | 0.0042* |
| Baseline (Replicated ALHAT)     | –               | 7cm            | 100%       | 0.7539    | 0.3790 | 0.0044 | 0.0000 | 1.0000 | 0.9956 |
| Base Net.                       | 1.67cm          | 7cm            | 100%       | 0.8465    | 0.6996 | 0.9244 | 0.1790 | 0.8209 | 0.0756 |
| Uncertainty-Aware               | 1.67cm          | 7cm            | 58%        | 0.9684*   | 0.9239* | 0.9807* | 0.0359* | 0.9641* | 0.0193* |

* The accuracy for our method is calculated out of the valid pixels on the uncertain-aware safety map, which are the pixels with less uncertainty than the threshold. The threshold is pre-calculated independently of the test set. The V/C Pix.% column shows the percentage of valid pixels used for evaluation.
Table 3 shows the safe rate of the final safe sites selected across 100 images. Note that for some DEMs, there are no viable safe sites remaining after the uncertainty threshold. Out of 100 input test DEMs, 3 DEMs are determined to have no viable safe landing sites for $1-\sigma = 1.67\text{cm}$ input noise case, and 4 DEMs for $1-\sigma = 3\text{cm}$ and $7\text{cm}$ input noise cases. Out of those DEMs predicted to have safe landing sites, 5, 6, and 3 DEMs have the sites selected that are unsafe in the ground truth for $1-\sigma = 1.67\text{cm}$, $3\text{cm}$, and $7\text{cm}$ input noise cases respectively.

V. Conclusion

This paper proposed a method for uncertainty-aware deep learning for safe landing site selection on planetary surfaces. Despite the recent advances of artificial intelligence techniques, their applicability to safety-critical space missions has often been limited due to concerns regarding their outputs’ reliability. In this paper, we demonstrated that incorporating network uncertainty into the final safety map can improve its accuracy of safe landing site selection. More importantly, we also demonstrated that safe sites can be reliably selected from DEMs with high noise, and from DEMs where the sensor noise of the test samples is outside the distribution of the training samples. The proposed method can be an important step towards a reliable and robust machine-learning method for realistic spacecraft applications.

Acknowledgments

This work is supported by the National Aeronautics and Space Administration under Grant No.80NSSC20K0064 through the NASA Early Career Faculty Program.

References

[1] Mustard, J., Adler, M., Allwood, A., Bass, D., Beaty, D., Bell, J., Brinckerhoff, W., Carr, M., Des Marais, D., Drake, B., Edgett, K., Eigenbrode, J., Elkins-Tanton, L., Grant, J., Milkovich, S., Ming, D., Moore, C., Murchie, S., Onstott, T., and Treiman, A., “Report of the Mars 2020 Science Definition Team,” 2013, pp. 1–154.

[2] San Martin, M., Mendeck, G. F., Brugarolas, P. B., Singh, G., Serricchio, F., Lee, S. W., Wong, E. C., and Essmiller, J. C., “In-flight experience of the mars science laboratory guidance, navigation, and control system for entry, descent, and landing,” *CEAS Space Journal*, Vol. 7, No. 2, 2015, pp. 119–142.
[3] Brady, T., Robertson, E., C. App, S. P., and Zimpfer, D., “Hazard detection methods for lunar landing,” 2009 IEEE Aerospace Conference, 2009, pp. 1–18.

[4] App, C. D., and Smith, T. B., “Autonomous precision landing and hazard detection and avoidance technology (ALHAT),” 2007 IEEE Aerospace Conference, 2007, pp. 1–7.

[5] Ivanov, T., Huertas, A., and Carson, J. M., “Probabilistic Hazard Detection for Autonomous Safe Landing,” AIAA Guidance, Navigation, and Control (GNC) Conference, 2013, pp. 1–12.

[6] Brady, T., and Schwartz, J., “ALHAT system architecture and operational concept,” 2007 IEEE Aerospace Conference, 2007, pp. 1–13.

[7] Striepe, S. A., Epp, C. D., and Robertson, E. A., “Autonomous precision landing and hazard avoidance technology (ALHAT) project status as of May 2010,” International Planetary Probe Workshop 2010 (IPPW-7), 2010.

[8] Rutishauser, D., Epp, C., and Robertson, E., “Free-Flight Terrestrial Rocket Lander Demonstration for NASA’s Autonomous Landing and Hazard Avoidance Technology (ALHAT) System,” AIAA SPACE 2012 Conference and Exposition, 2012, pp. 1–12.

[9] Epp, C., Robertson, E., and Carson, J. M., “Real-time hazard detection and avoidance demonstration for a planetary lander,” AIAA SPACE 2014 Conference and Exposition, 2014, pp. 1–23.

[10] Johnson, A. E., Huertas, A., Werner, R. A., and Montgomery, J. F., “Analysis of on-board hazard detection and avoidance for safe lunar landing,” 2008 IEEE Aerospace Conference, 2008, pp. 1–9.

[11] Huertas, A., Johnson, A. E., Werner, R. A., and Maddock, R. A., “Performance evaluation of hazard detection and avoidance algorithms for safe Lunar landings,” 2010 IEEE Aerospace Conference, 2010, pp. 1–20.

[12] Cohanim, B. E., and Collins, B. K., “Landing point designation algorithm for lunar landing,” Journal of Spacecraft and Rockets, Vol. 46, No. 4, 2009, pp. 858–864.

[13] Paschall, S., and Brady, T., “Demonstration of a safe and precise planetary landing system on-board a terrestrial rocket,” 2012 IEEE Aerospace Conference, 2012, pp. 1–8.

[14] 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 (GNC) Conference, 2015.

[15] Tomita, K., Skinner, K. A., Iiyama, K., Jagatia, B. A., Nakagawa, T., and Ho, K., “Hazard Detection Algorithm for Planetary Landing Using Semantic Segmentation,” In Proceedings of AIAA ASCEND, 2020, pp. 1–11.

[16] Moghe, R., and Zanetti, R., “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.
[17] Gustafsson, F. K., Danelljan, M., and Schon, T. B., “Evaluating scalable bayesian deep learning methods for robust computer vision,” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 318–319.

[18] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q., “On calibration of modern neural networks,” arXiv preprint arXiv:1706.04599, 2017.

[19] Kendall, A., and Gal, Y., “What uncertainties do we need in bayesian deep learning for computer vision?” Advances in neural information processing systems, 2017, pp. 5574–5584.

[20] Mukhoti, J., and Gal, Y., “Evaluating bayesian deep learning methods for semantic segmentation,” arXiv preprint arXiv:1811.12709, 2018.

[21] Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P., and Green, W. H., “Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction,” Journal of Chemical Information and Modeling, 2020, pp. 2697–2717.

[22] Restrepo, C. I., Chen, P.-T., Sostaric, R. R., and Carson, J. M., “Next-Generation NASA Hazard Detection System Development,” AIAA Scitech 2020 Forum, 2020, pp. 1–10.

[23] Furfaro, R., Simo, J., Gaudet, B., and Wibben, D., “Neural-based trajectory shaping approach for terminal planetary pinpoint guidance,” AAS/AIAA Astrodynamics Specialist Conference 2013, 2013, pp. Paper–AAS.

[24] Gaudet, B., and Furfaro, R., “Adaptive pinpoint and fuel efficient mars landing using reinforcement learning,” IEEE/CAA Journal of Automatica Sinica, Vol. 1, No. 4, 2014, pp. 397–411.

[25] Sánchez-Sánchez, C., and Izzo, D., “Real-time optimal control via deep neural networks: study on landing problems,” Journal of Guidance, Control, and Dynamics, Vol. 41, No. 5, 2018, pp. 1122–1135.

[26] Furfaro, R., Bloise, I., Orlandelli, M., Di Lizia, P., Topputo, F., Linares, R., et al., “A recurrent deep architecture for quasi-optimal feedback guidance in planetary landing,” IAA SciTech Forum on Space Flight Mechanics and Space Structures and Materials, 2018, pp. 1–24.

[27] Furfaro, R., Bloise, I., Orlandelli, M., Di Lizia, P., Topputo, F., Linares, R., et al., “Deep learning for autonomous lunar landing,” 2018 AAS/AIAA Astrodynamics Specialist Conference, Vol. 167, Univelt, 2018, pp. 3285–3306.

[28] Gaudet, B., Linares, R., and Furfaro, R., “Deep reinforcement learning for six degree-of-freedom planetary landing,” Advances in Space Research, Vol. 65, No. 7, 2020, pp. 1723–1741.

[29] Scorsoglio, A., Furfaro, R., Linares, R., and Gaudet, B., “Image-based Deep Reinforcement Learning for Autonomous Lunar Landing,” AIAA Scitech 2020 Forum, 2020, p. 1910.

[30] Iiyama, K., Tomita, K., Jagatia, B. A., Nakagawa, T., and Ho, K., “Deep reinforcement learning for safe landing site selection with concurrent consideration of divert maneuvers,” AAS/AIAA Astrodynamics Specialist Conference 2020, 2020, pp. Paper–AAS.
[31] 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.

[32] 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.

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

[34] 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*, Vol. 57, No. 8, 2019, pp. 5777–5789.

[35] 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.

[36] Downes, L., Steiner, T. J., and How, J. P., “Deep learning crater detection for lunar terrain relative navigation,” *AIAA SciTech 2020 Forum*, 2020, p. 1838.

[37] Lakshminarayanan, B., Pritzel, A., and Blundell, C., “Simple and scalable predictive uncertainty estimation using deep ensembles,” *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[38] Kendall, A., Badrinarayanan, V., and Cipolla, R., “Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding,” *arXiv preprint arXiv:1511.02680*, 2015.

[39] Badrinarayanan, V., Kendall, A., and Cipolla, R., “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 12, 2017, pp. 2481–2495.

[40] Skinner, K. A., Tomita, K., and Ho, K., “Uncertainty-aware deep learning for safe landing site selection,” *AAS/AIAA Space Flight Mechanics Meeting 2021*, 2021, pp. Paper–AAS.

[41] “BlenSor: Blender Sensor Simulation,” [https://www.blensor.org/](https://www.blensor.org/) ??. Accessed: 2020-10-02.

[42] Gschwandtner, M., Kwitt, R., Uhl, A., and Pree, W., “BlenSor: Blender sensor simulation toolbox,” *International Symposium on Visual Computing*, Springer, 2011, pp. 199–208.

[43] University of Arizona, “HiRISE DTMs, University of Arizona,” [https://www.uahirise.org/dtm/](https://www.uahirise.org/dtm/) 2020. Accessed: 2020-10-02.

[44] Bulyshev, A., Vanek, M., Amzajerdian, F., Pierrottet, D., Hines, G., and Reisse, R., “A super-resolution algorithm for enhancement of FLASH LIDAR data,” *Computational Imaging IX*, Vol. 7873, International Society for Optics and Photonics, 2011, p. 78730F.

[45] Keim, J. A., Mobasser, S., Kuang, D., Cheng, Y., Ivanov, T., Johnson, A. E., Goldberg, H. R., Khanoyan, G., and Natzie, D. B., “Field test implementation to evaluate a flash LIDAR as a primary sensor for safe lunar landing,” *2010 IEEE Aerospace Conference*, IEEE, 2010, pp. 1–14.
[46] Amzajerdian, F., Roback, V. E., Bulyshev, A., Brewster, P. F., and Hines, G. D., “Imaging flash lidar for autonomous safe landing and spacecraft proximity operation,” *AIAA SPACE 2016*, 2016, p. 5591.

[47] Long, J., Shelhamer, E., and Darrell, T., “Fully convolutional networks for semantic segmentation,” *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.