International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Real-Time Hazard-Free Planetary Landing using Deep Recurrent Neural Network

Janhavi Hemant Borse*[1], Dipti Durgesh Patil[2], Vinod Kumar[3]

Submitted: 09/05/2023 **Revised**: 17/07/2023 **Accepted**: 08/08/2023

Abstract*:* Accurate understanding and interpretation of the underlying field of view (FoV) are paramount in a real-time planetary landing. This understanding helps detect hazardous bodies and bypasses unfavourable situations well in advance. Existing planetary landing missions rely on the 3-dimensional digital elevation models (DEM) and achieve terrain-relative navigation. These DEMs are used as a reference for spotting the hazards on the pre-defined landing site. These are computationally intensive and time-consuming pattern-matching tasks. The primary concern is the existence of such DEMs before the missions and high storage requirements. This study aims to tackle the abovementioned drawbacks and build a robust intelligent system for autonomous hazard-free planetary landing. This paper utilizes the advanced deep learning approach for accurately detecting and positioning the hazards in the current FoVs using vision sensors of the spacecraft. Deep convolutional neural networks are utilized for feature extraction purposes. These features are further utilized by the recurrent neural network's region proposal algorithm to spot the distinct regions inside the current FoV. These proposals are the keys to detecting the hazards like craters and boulders. The detection results are interpreted by classifying the hazards into craters and boulders. It also classifies the safe landing region as a plain surface. Further, the classes are positioned accurately using the bounding boxes of the coco model. Transfer learning is used to build and train the network. The work also includes the creation of a valid planetary dataset required for generating a ground truth. The overall results are validated through comparative judgments and exhaustive analysis. Experimental results show that the transfer learning approach for hazard detection and localization achieved excellent results.

Keywords: *Convolutional Neural Network, Deep Neural Network, Hazard Detection, Terrain, Space Navigation, Space Guidance*

1. Introduction

On-earth navigation guidance system like the Global Positioning System (GPS) allows aircraft to land safely and accurately at desired locations. But spacecraft exploring other bodies in space does not have a system to guide them to land at hazard-free locations. It needs to use different methods to determine where they are. During the Apollo moon landings, astronauts sighted landmarks for landing. They looked out the window during the final descent to avoid craters and boulder fields and land safely. But this approach must depend on the onboard astronaut, and it also involves human error and perception of humans. Since then, sensors, algorithms, and onboard computing can replace and surpass human capability to navigate and enable safe landings in space. NASA's Mars 2020 spacecraft has used such a system to land the Perseverance rover at the Jezero Crater. Here by matching onboard sensor data to a map of the landing area, Terrain Relative Navigation (TRN) provides a map-relative position fix that can be used to accurately target specific landing points on the surface of a celestial body and avoid hazards. TRN allows for mitigating landing hazards in the regions and facilitates hazard-free landings. TRN's effort relied heavily on the design of the Entry, Descent, and Landing (EDL) a system where the spacecraft descends toward the surface on a parachute, and then, during the final seconds before landing, it lowers the upright spacecraft on a tether to the surface, much like a crane. But This system uses ground-based navigation from Earth to determine the position of the spacecraft just before atmospheric entry. But the position error is too large at 3km above the surface to enable precise navigation. To reduce this error below 40m relative to a landing site map, the spacecraft is augmented with Lander Vision System (LVS). The onboard guidance system selects a reachable landing point given the fuel onboard. It avoids hazards identified a priori in the map. This effort required flight software development for landmark matching, state estimation, and safe site selection; a high-performance compute element for vision processing; a high frame-rate camera; and high-precision mapping techniques.

These DEMs are the 3-dimensional representation of the pre-known FoVs captured by the orbital missions. It provides elevation information about all possible viewpoints of the planet. The time required to process this data is a hindrance to the navigation. The high response time of the DEM approach is due to computer vision tasks such as pattern matching.

[1] *Department of Computer Engineering, SKNCOE, Savitribai Phule Pune University, Pune-India*
ORCID ID : 0000-0003-0761-7428
[2] *Department of Information Technology, MKSSS's Cummins College of Engineering for women, Savitribai Phule Pune University, Pune-India*
ORCID ID : 0000-0001-7379-863X
[3] *Director, Promotion Directorate, IN-SPACe, Executive Secretary, ASI Department of Space, Govt. of India, Bengaluru, INDIA – 560094*
ORCID ID : 0000-0002-4468-4083
** Corresponding Author Email: borse.janhavi@gmail.com*

Our prior work [1], [18]-[20] discusses the methods of mathematical approximations, feature extraction methods [2], [3], vision-based navigation methods using LiDAR or SAR [4]–[7], and methods based on DEMs [8]–[11]. Our previous work proposed a combined approach with machine learning to avoid the need for all the facilities. It uses deep neural networks, classifies onboard camera images into categories, and shows hazards by coloured segments. After the segmentation map is generated, a graded membership of a particular hazardous/hazard-free area is also shown by embedding a fuzzy membership function as a non-linear functionality. The method is proposed with a clear objective of detecting the terrain objects like boulders using graded classification. With known landing site coordinates, this method can be used to map the given coordinates onto the classified image to check where exactly the landing spot lies, and then retargeting decision can be made. The combined approach succeeds in the hazard detection, and graded classification shows the logical way of representing the possible hazards of the terrain and provides visual completeness to the solution. Hence the dependency of hazard detection system on heavy pre-processing and other needs of DEM models can be avoided.

But this approach does not guarantee real-time processing capability or help position the terrain hazards. Hence in the current work, we introduced a new approach for real-time hazard detection using the on-the-spot captured camera images. Due to the scarcity of landing videos, a new synthetic dataset of landing videos is generated through the Unreal Engine [12] simulator. The ground truth images containing bounding boxes are generated using Roboflow [13] software to position the hazards. A deep recurrent neural network model is trained using this synthetic dataset and then validated using pre-known object detection base models: You Look Only Once (YOLO), and MobileNet. The deep learning architecture is designed and respective hazard detection models are built and validated using comparative analysis. The pixel positions obtained through the trained models can be easily converted to the 3D world coordinates with the help of known camera calibrations and geometry.

Significant contributions of this work are:

1. Synthetic Video Data Generation with labels for landing space vehicle

2. Video Data Annotation for hazard detection

3. Real-time hazard detection system for planetary landing

1.1. Abbreviations and Acronyms

Global Positioning System (GPS), Field of View (FoV), Region of Interest (RoI), Convolutional Neural Network (CNN), Terrain Relative Navigation (TRN), Entry Descent and Landing (EDL), Lander Vision System (LVS), Digital Elevation Models (DEM), Recurrent convolutional neural network (RCNN), Batch Normalization (BN), Rectifier Linear Unit (ReLU), Graphics Processing Unit (GPU).

1.2. Organization of the paper

Section 2 discusses the data generation methodology and the proposed real-time hazard detection and localization methodology. Section 3 discusses the results of the proposed approach and two more approaches using MobileNet and YOLO for comparative analysis. At last, section 4 concludes the paper.

2. Material and Proposed Method

The proposed method involves data generation in the first step and designing a deep recurrent neural network architecture that utilizes the faster RCNN [1] as the base model in the next step. A typical RCNN network is so designed that it utilizes some edge detection algorithm in the first step to divide the input images into several semantically different segments. Later in the second step, these segments extract the different region proposals using a Region Proposal algorithm [2]. Such regions are called Region of Interest (RoI). While the region proposals are extracted, deep convolutional neural networks (CNN) are employed for the feature extraction process. The results of both steps are logically combined and fed to the output layer for classification and regression tasks. The following subsections explain the complete data generation process and the proposed deep neural network architecture based on the RCNN object detection model. It describes the planetary hazard detection classification and hazard localization processes using regression.

2.1. Synthetic Data Generation

Hazard-free planetary landing using supervised machine learning methods requires much training data. This data must incorporate the different landing scenarios. More precisely, the real-time videos or images captured while descending a spacecraft to the desired location are required. The solution to the problem is using a software simulator named Unreal Engine-4. Generally, this simulator is intended for gaming and has been proven to create real-world gaming environments. It can also be helpful for AI, Computer Vision tasks such as training agents; generating and simulating prototypes is only possible because several reusable APIs support multiple programming languages.

For the same reason, the unreal simulator is employed for generating the synthetic data in this work. A lunar environment is used as an underlying Terrain. A spacecraft agent is created with a camera to capture the underlying terrain. Multiple simulations are run for the safe landing of the agent, and thus descent videos are captured. A glimpse of the data generated through an Unreal environment is shown in Fig. 1. These images are extracted at a 30 fps frame

rate. Images are shown with the altitude information in Fig. 1 for better understanding.

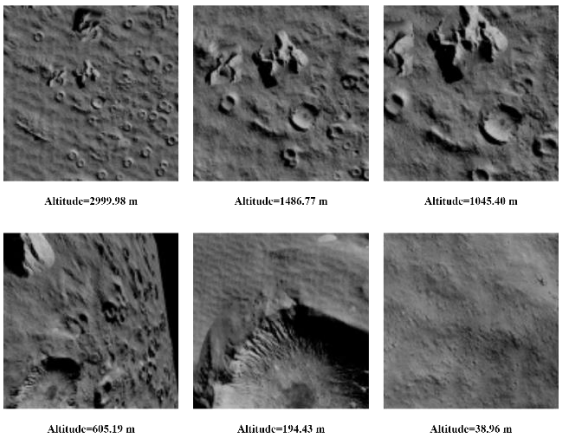

Fig. 1. Database samples generated through simulated lunar environment

2.2. System Architecture for Real-Time Hazard Detection using Recurrent Neural Network

The real-time hazard detection and localization system is depicted in Fig. 2. The system has three major processing parts: region proposal algorithm (RPN), convolutional neural network (CNN), and classifier/regression layer. The region proposal algorithm is a network proposed by Shaoqing Ren et al. [14], [15]. It takes the input image and segments it into distinct regions after pre-processing. These are the possible RoIs, as suggested by the network. It has an anchor point concept. The anchor point is the centre of each proposed region. A deep network consists of multiple back-to-back convolutions, inner activations, and pulling layers intended to generate the feature maps from an input image. Each region of interest is a sliding window over the feature maps

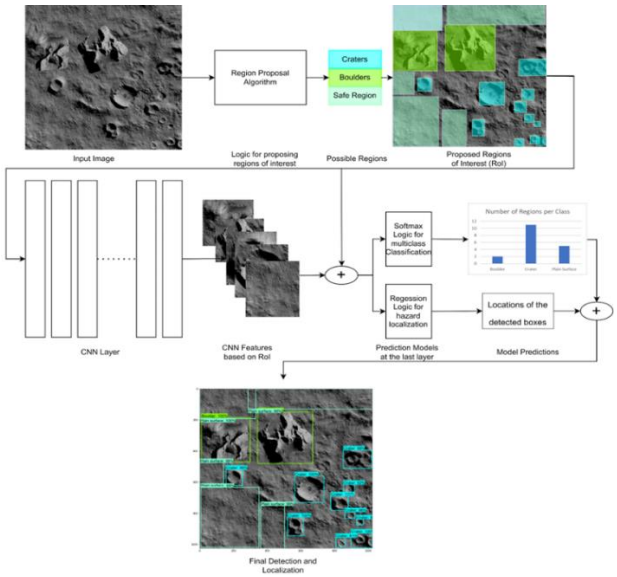

Fig. 2. Network architecture for real-time hazard detection and localization

obtained through the convolution operation. The adder symbol in the network shows this operation. A softmax activation function is used for multi-class classification. The classification result is further revised using SVM class confidences. Hazards are classified into three categories: Crater, Boulder, and Plain surface. To localize the hazardous region, simple regression logic is implemented. For each input image captured through a real-time camera, the image spotted with hazards and their corresponding localization is the outcome of the entire process. A pre-trained RCNN base model is utilized as a starting point for training the network.

2.3. Hazard Detection & Localization using MobileNet

The basic steps followed in the MobileNet [16] architecture (Fig. 3) are:

1. Apply depthwise convolution (channel-wise separate)

2. Apply point convolution (to combine depth convolutions)

In MobileNetV2, there are two types of blocks. One is a residual block with a stride of 1. Another one is a block with a stride of 2 for downsizing. There are 3 layers for both types of blocks. This time, the first layer is 1×1 convolution with ReLU6. The second layer is the depthwise convolution. The third layer is another 1×1 convolution but without any non-linearity. It is claimed that if ReLU is used again, the deep networks only have the power of a linear classifier on the non-zero volume part of the output domain.

2.4. Hazard Detection & Localization using YOLO

YOLO [17] first resizes the images to "448 $pixels$ × 448 $pixels$" and then feds to the convolution layers. All 24 convolution layers have preceded max pool layers for extracting vital features. In the end, two fully connected layers are attached to the softmax activation function for classification. A typical YOLO architecture is trained using an Image net dataset freely available online. This work utilizes this pre-trained YOLO model and rebuilds the network for current application use using transfer learning.

It divides an input image into an m x m grid. Each grid tends to have n bounding boxes. The network outputs a class probability and extracts offset for each bounding box. The bounding boxes with the class probability above a threshold value are selected and used to locate the object within the image. It is found to be the fastest object detection algorithm, such that it can process around 45 images per second. Although it has some spatial constraints due to grids and hence fails to detect smaller-sized objects.

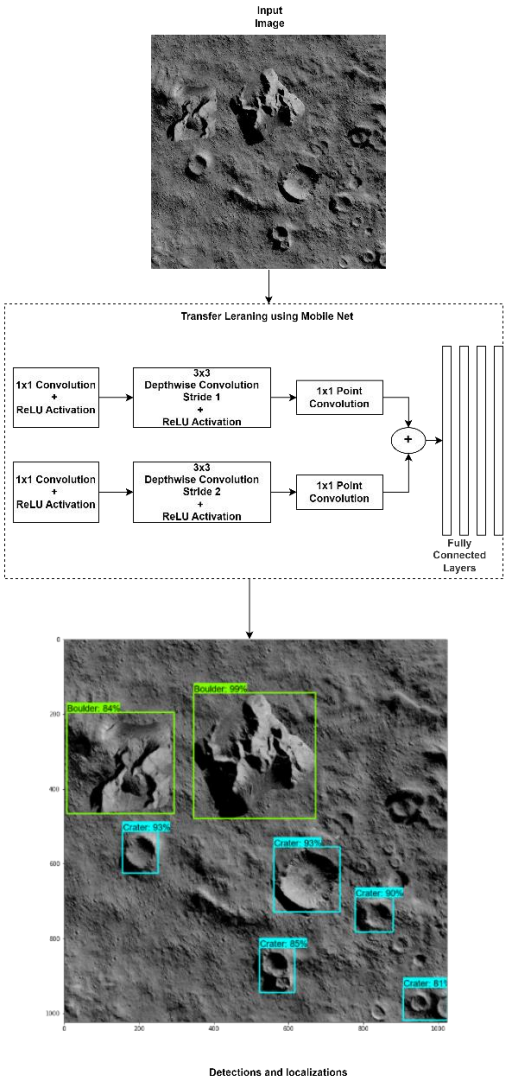

Fig. 3. Real-time hazard detection and localization using MobileNet architecture.

3. Results and Discussion

3.1. Metrics for results validation

The results of the hazard detection are validated using precision and recall metrics and then compared among three implementations. The mean average precision and mean average recall are computed as in equations (1) and (2), respectively. TP is true positive, FP is false positive, and FN is false negative for the class category C.

$$Mean\ Average\ Precision = \frac{1}{|C|}\sum_{c\in C}\frac{TP(c)}{(TP(c)+FP(c))} \quad (1)$$

$$Average\ Recall = \frac{1}{|C|}\sum_{c\in C}\frac{TP(c)}{(TP(c)+FN(c))} \quad (2)$$

The hazard categories are further separated into large, medium, and small regions based on the total area that region is spanning. The large area spans regions greater than "$96\ pixels \times 96\ pixels$", and small area spans regions lesser than "$32\ pixels \times 32\ pixels$". The medium area holds regions between "$32\ pixels \times 32\ pixels$" and "$96\ pixels \times 96\ pixels$".

The confidence of precision and recall values is also taken into consideration using Intersection over Union values as given by equation (3). It tells how much overlapping the predictions have with the ground truth values.

$$IoU(Predictions\ ,\ Truth\ Values) = \frac{(Predictions \cap Truth\ Values)}{(Predictions \cup Truth\ Values)} \quad (3)$$

3.2. Hazard Detection Accuracy using Precision, Recall metrics

Table 1 enlists the detection accuracy of transfer learning architectures using YOLO, RCNN, and MobileNet models through precision-recall metrics.

3.2.1 Detection Accuracy using RCNN

The overall mean average precision (mAP) of RCNN detections is 0.74 in validation, achieved at 20k steps per epoch. On the other hand, a precision value of 0.73 is achieved at 15k steps per epoch in training. For large areas, validation mAP is 0.77 at 20k steps per epoch. For medium-sized regions, mAP is 0.37, while for smaller regions, it is 0.39. Overall detection box precision with constrained 0.5 IoU is 0.96, which is very positive for both the training and validation set, while precision with constrained 0.75 IoU is 0.86, which is also a good value at such a high threshold.

The recall is measured over detections per image and the size of regions per image. Three thresholds maintained are 1, 10, and 100, which tell the number of output detections. Per epoch (20k steps in validation and 15k steps in training), with 1 max detection, the recall is 0.18; with 10 max detections, it is 0.60; and with 100 max detections, it is 0.70, which is not bad. For the large proposed region, recall is 0.72; the medium region is 0.56, while for small regions, it is significantly less equal to 0.30.

Fig. 11 depicts classification, regression (RPN localization), regularization, and total loss throughout 20k steps. The regularization loss can be seen as nullable throughout the training and validation process. However, the other losses are relatively steady till 10k steps and then sharply decline towards 0 in 20k steps.

3.2.2 Detection Accuracy using MobileNet

The overall mean average precision of detections using MobileNet is 0.68 in validation, achieved at 35k steps per epoch. On the other hand, a precision value of 0.68 is achieved at 35k steps per epoch in training. For large proposed regions, training & validation mean average precision-mAP is 0.72 at 35k steps per epoch, respectively. For medium-sized regions, mAP is 0.26, while for smaller regions, it is 0.07. Overall detection box precision with constrained 0.5 IoU is 0.94, which is very positive for both training and validation set but at the cost of 45k steps per

epoch, while precision with constrained 0.75 IoU is 0.76 which is also a good value at such a high threshold.

The recall is measured over detections per image and the size of regions per image. Three thresholds maintained are 1, 10, and 100, which tell the number of output detections. Per epoch (45k steps in validation and 45k steps in training), with 1 max detection, the recall is 0.02; with 10 max detections, it is 0.68; and with 100 max detections, it is 0.77, which is not bad. For the large proposed region, recall is 0.79; medium region, it is 0.44; while for small regions, it is significantly less equal to 0.26.

Fig. 10 depicts classification, localization, regularization, and total loss for 45k steps. The training and validation process losses sharply decline towards 0 at 45k steps. It tells the progressive nature of the model in learning.

3.2.3 Detection Accuracy using YOLO

The average precision and recall using YOLO are around 0.9, which is an excellent value. Mean average precision is also computed over variable constraints of confidence level ranging from 0.5 to 0.95. It is found to be 0.75 with such a high-quality threshold. The loss function decreases steadily until it reaches 0 at 50k steps per epoch. YOLO, MobileNet, and RCNN models show comparable results for all combined regions. But our design with RCNN base architecture outperforms the YOLO and MobileNet models in detection accuracy (precision and recall) for smaller and medium areas. And it achieves that in only 20k steps per epoch compared to the 50k steps for YOLO and 35k for MobileNet.

3.3. Comparative Discussion

The colour theme differentiates three hazard kinds: Crater, Boulder, and Plain-Surface. The localization of hazards is interpreted through bounding boxes. Each bounding box shows the confidence with which a particular hazard is detected.

Fig.s 4, 5, and 6 show detection and localization using RCNN and MobileNet models for comparative analysis. The colour theme used for hazards is cyan for Crater, parrot green for the boulder, and sea blue for the Plain-Surface category. The test image in Fig. 4 shows the image captured at an altitude of around 3km. The results show average 95.33 % accuracy of detection using RCNN. MobileNet failed to detect the hazards. The test image in Fig. 5 shows the image captured at an altitude of around 1.5km. The results show average 95% accuracy of detection using RCNN. The hazard detection average accuracy using MobileNet is 85%. The test image in Fig. 6 shows the image captured at an altitude of around 1km. The results show average 97 % accuracy of detection using RCNN. MobileNet results show 95% accuracy for 3 detections.

Fig.s 7 and 8 show detection and localization results using RCNN and YOLO base models for comparative analysis. Blue is for the Crater category, cyan is for the Boulder category, and orange is for the Plain-Surface category. The above nomenclature is followed for RCNN, while red for boulder, orange for plain surface, and light pink is used for the YOLO model. Say a crater is detected with 1.0 confidence. It means that there is 100% confidence in that detection. At an altitude of 1486.77 m, the detections from RCNN and YOLO are depicted in Fig. 1. For this sample image, the number of detections is almost equal through both models' predictions. But the confidence level is much higher through RCNN than through the YOLO model for each category. Sample Fig. 8 depicts the predictions at an altitude of around 1 km for both models. In this case, RCNN could logically find a small shadowed crater behind a large boulder. But the YOLO model failed to do so. Rest all predictions almost match, although the confidence level is much lower in YOLO's case.

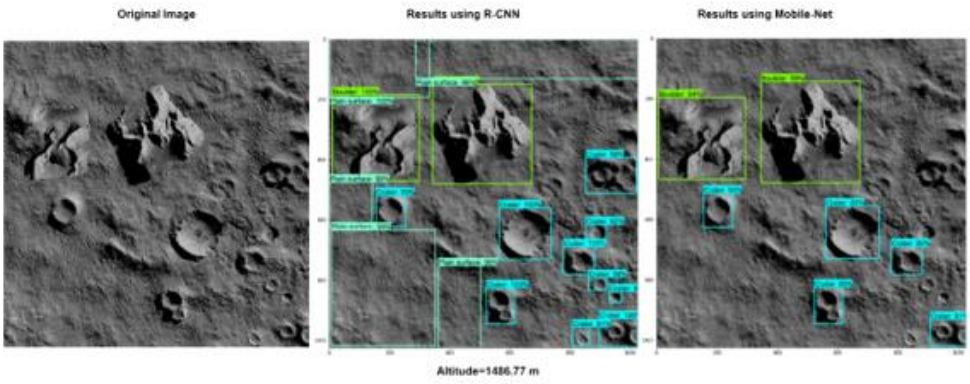

Fig. 1 Comparative analysis between RCNN and Mobile Net based models at varying altitude levels at altitude=1486.77 m

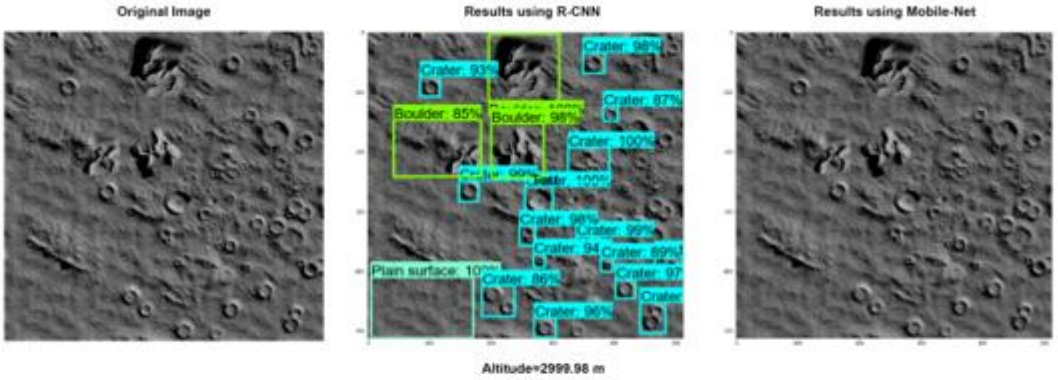

Fig. 2 Comparative analysis between RCNN and Mobile Net based models at varying altitude levels at altitude=2999.98 m

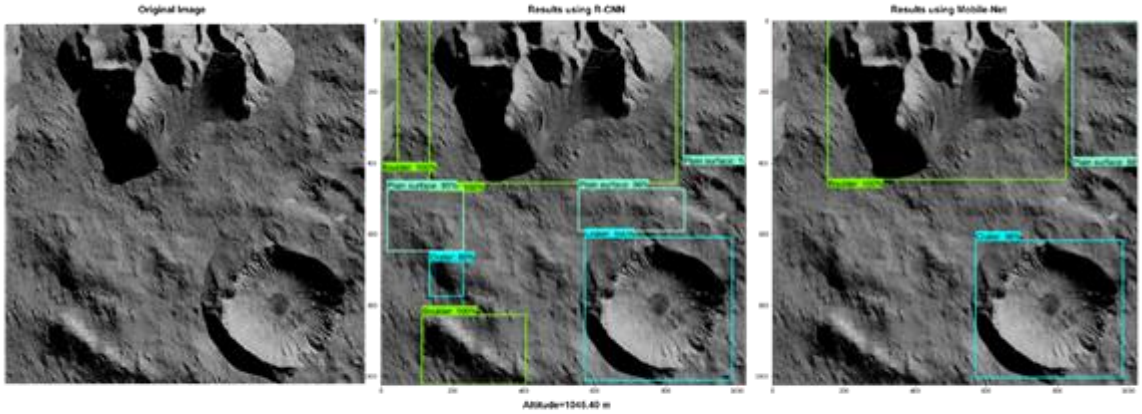

Fig. 6 Comparative analysis between RCNN and Mobile Net based models at varying altitude levels at altitude=1045.40 m

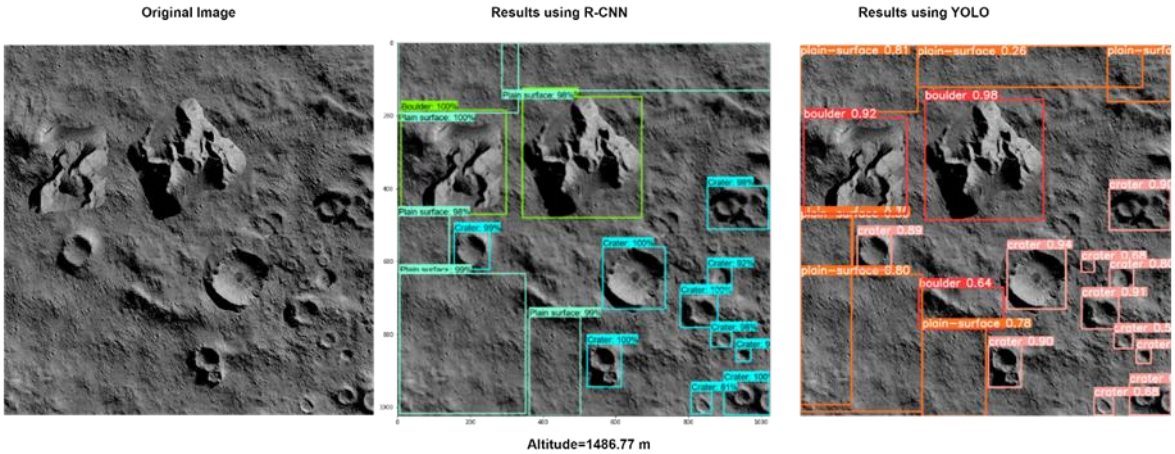

Fig. 7 Comparative analysis between RCNN and YOLO models at varying altitude levels at altitude=1486.77 m

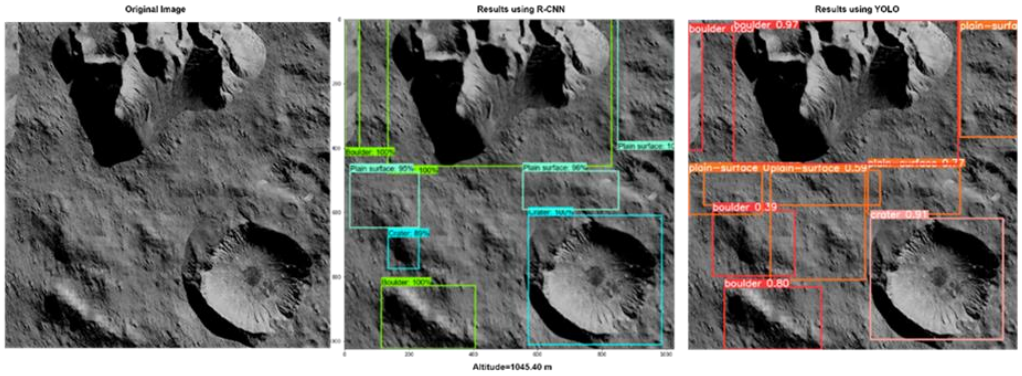

Fig. 8 Comparative analysis between RCNN and YOLO models at varying altitude levels at altitude=1045.40 m

Fig.s 9, 10, and 11 show detection losses using YOLO, MobileNet, and RCNN models. YOLO takes up to 50k steps to generalize, Mobilenet takes upto 40k steps, while RCNN generalize very quickly in 20k steps per epoch.

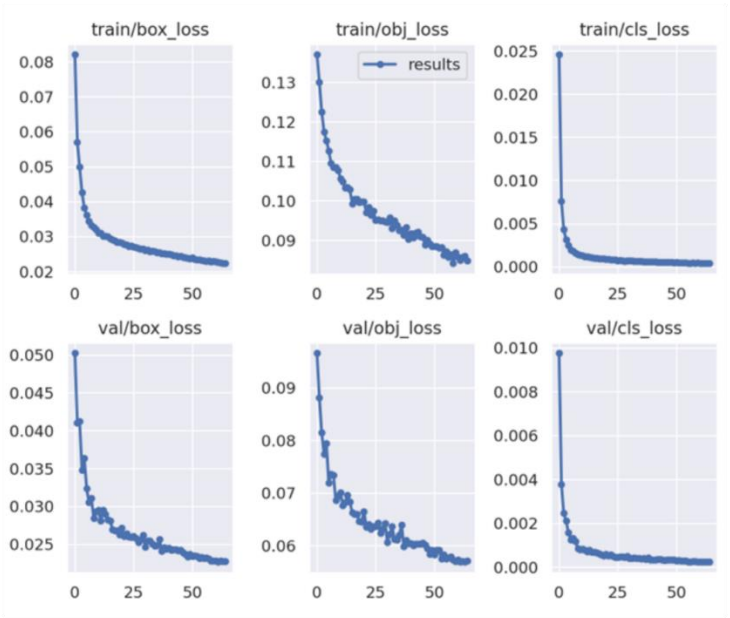

Fig. 9 YOLO Losses in 50k steps per epoch

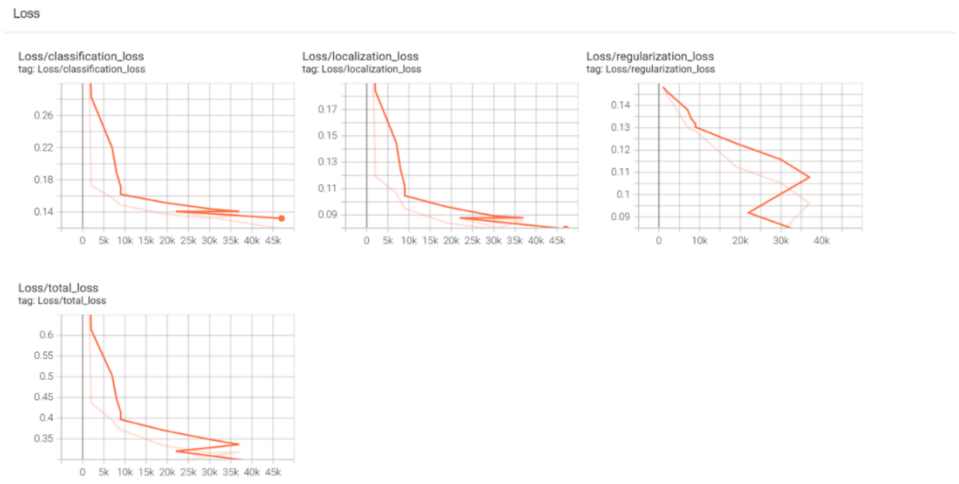

Fig. 10 Mobile Net Losses in 40k steps per epoch

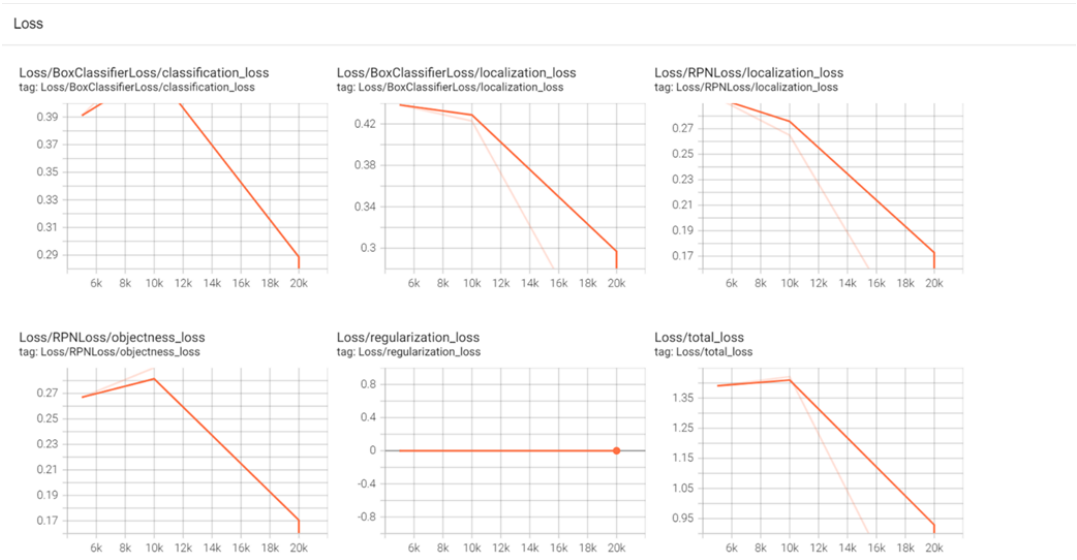

Fig. 11 RCNN detection losses in 20k steps per epoch

4. Conclusion

This paper describes a hazard detection and localization methodology for hazard-free planetary landing. The system first detects the hazards, then classifies them into three categories Craters, Boulders, and Plain Surfaces. For localization, bounding boxes are populated over the detected hazards. Due to the scarcity of planetary data, a new synthetic dataset is generated using simulation software Unreal Engine and Roboflow with ground truth.

The training and validation of the system are performed on the artificial lunar images. The overall results of the deep learning approach using recurrent neural network is promising, with an overall accuracy of 90-95%. The results show the righteousness of the system for real-time terrain hazard detection and localization. The real-time processing speed and accuracy of RCNN are much better than the other two models trained. The RCNN model is more efficient in detecting hazards for smaller regions than other models. It even achieves that with significantly fewer equal to 20k steps per epoch. It means that it is a much faster-converging model.

On the other hand, the MobileNet model still fails to detect all possible regions of interest in many cases. The YOLO model is relatively stable in detections as compared to the others. When tested with real-time video input, it is found that the recurrent neural network model detects the highest number of hazards with accurate localizations.

Acknowledgements

This research was supported by ISRO-SPPU reasearch Grant. We thank Research Centre Smt. Kashibai Navale College of Engineering Pune and Savitribai Phule Pune University for valuable inputs and guidance.

Author contributions

Janhavi Borse: Conceptualization, Design, Methodology, Data Preparation, Implementation, Drafting **Dipti Patil:** Design , Data Curation, Validation, Proof Reviewing **Vinod Kumar:** Visualization, Investigation, Writing-Reviewing

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] V. K. Janhavi Borse, Dipti Patil, "DEEP SEMANTIC CLASSIFICATION OF VISUAL INPUTS FOR HAZARD-FREE LUNAR LANDING," vol. 3, no. June, pp. 14–18, 2021.

[2] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *J. Geophys. Res.*, vol. 76, no. 8, pp. 1905–1915, Mar. 1971, doi: 10.1029/jb076i008p01905.

[3] K. S. P. Lancaster, "Surface generated by moving least squares methods," *Math Comp*, vol. 37, pp. 141–158, 1981.

[4] J. Song, D. Rondao, and N. Aouf, "Deep learning-based spacecraft relative navigation methods: A survey," *Acta Astronaut.*, vol. 191, pp. 22–40, Feb. 2022, doi: 10.1016/J.ACTAASTRO.2021.10.025.

[5] D. E. Smith, M. T. Zuber, G. A. Neumann, and F. G. Lemoine, "Topography of the Moon from the Clementine lidar," *J. Geophys. Res. E Planets*, vol. 102, no. E1, pp. 1591–1611, 1997, doi: 10.1029/96JE02940.

[6] "A tutorial on synthetic aperture radar," *ieeexplore.ieee.org*, doi: 10.1109/MGRS.2013.2248301.

[7] M. Hidaka, M. Takahashi, T. Ishida, and S. Fukuda,

"Terrain relative navigation enhanced with sar for moon's shadowed regions," *AIAA Scitech 2020 Forum*, vol. 1 PartF, no. January, pp. 3–14, 2020, doi: 10.2514/6.2020-0946.

[8] M. S. R. A. C. Cook, P. D. Spudis, "Lunar topography and basins mapped using a Clementine stereo digital elevation model," *Lunar Planet Sci*, vol. 1, p. 1281, 2002.

[9] J. Ping, Q. Huang, J. Yan, J. Cao, G. Tang, and R. Shu, "Lunar topographic model CLTM-s01 from Chang'E-1 laser altimeter," *Sci. China, Ser. G Physics, Mech. Astron.*, vol. 52, no. 7, pp. 1105–1114, Jul. 2009, doi: 10.1007/s11433-009-0144-8.

[10] H. Araki *et al.*, "Lunar global shape and polar topography derived from Kaguya-LALT laser altimetry," *Science (80-.).*, vol. 323, no. 5916, pp. 897–900, Feb. 2009, doi: 10.1126/science.1164146.

[11] Z. Cai, C. Zheng, Z. Tang, and D. Qi, "Lunar digital elevation model and elevation distribution model based on Chang'E-1 LAM data," *Sci. China Technol. Sci.*, vol. 53, no. 9, pp. 2558–2568, Aug. 2010, doi: 10.1007/s11431-010-3180-8.

[12] Epic Games, "Unreal Engine," 2019. https://www.unrealengine.com.

[13] "Roboflow Software." Roboflow Software, [Online]. Available: https://roboflow.com/.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, 2017, doi: 10.1109/TPAMI.2016.2577031.

[15] R. Girshick, J. Donahue, T. Darrell, J. Malik, U. C. Berkeley, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, p. 5000, 2014, doi: 10.1109/CVPR.2014.81.

[16] M. Sandler, A. Howard, M. Zhu, and A. Zhmoginov, "Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.pdf," pp. 4510–4520, 2018.

[17] S. D. Joseph Redmon and A. F. , Ross Girshick, "You Only Look Once: Unified, Real-Time Object Detection," *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3243394.3243692.

[18] J. H. Borse, D. D. Patil, V. Kumar, and S. Kumar, "Soft Landing Parameter Measurements for Candidate Navigation Trajectories Using Deep Learning and AI-Enabled Planetary Descent," *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/2886312.

[19] J. Borse*, D. Patil and V. Kumar, "Tracking Keypoints from Consecutive Video Frames Using CNN Features for Space Applications", *Tehnički glasnik*, vol.15, no. 1, pp. 11-17, 2021. [Online]. doi: 10.31803/tg-20210204161210.

[20] Janhavi H. Borse and Dipti D. Patil, “Empirical Analysis of Feature Points Extraction Techniques for Space Applications” International Journal of Advanced Computer Science and Applications (IJACSA), 12(9), 2021. doi: 10.14569/IJACSA.2021.0120910.

[21] Revathy, S. ., & Priya, S. S. . (2023). Enhancing the Efficiency of Attack Detection System Using Feature selection and Feature Discretization Methods. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 156–160. https://doi.org/10.17762/ijritcc.v11i4s.6322

[22] Basaligheh, P. (2021). A Novel Multi-Class Technique for Suicide Detection in Twitter Dataset. Machine Learning Applications in Engineering Education and Management, 1(2), 13–20. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/14