SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I3P105

Volume 10 Issue 3, 44-51, March 2023

Original Article

Prototype Deep Learning System for Terrain Relative Navigation Missions

B. Saravana Kumar[1], Nagendra Gajjar[2]

[1] *Space Applications Centre, ISRO, Gujarat, India*
[2] *Institute of Technology, Nirma University, Gujarat, India*

[1]*Corresponding Author : saravana@sac.isro.gov.in*

Received: 15 January 2023 Revised: 01 March 2023 Accepted: 13 March 2023 Published: 29 March 2023

Abstract *- Navigation of a lander craft to accurately soft land in a predetermined spot on an extra-terrestrial body is a challenging task. Due to the absence of GPS and real-time ground communication links, the lander craft has to navigate autonomously to the targeted landing site solely with the help of its onboard sensors. One of the major challenges in position estimation of the lander craft is the measurement inaccuracies associated with the inertial navigation systems. Optical landmark detection-based techniques are employed to aid the lander navigation and guidance system. In this technique, craters and their relative positions are used as landmarks for position estimation. Typical crater detection algorithms employ handcrafted feature extractors for crater identification. Recently deep learning-based approaches have been employed for crater detection, mainly for geomorphological studies and cataloguing. This paper proposes using deep learning-based object detection techniques applied to autonomous landing missions and their implementation on a Xilinx MPSoC FPGA device. Prototype hardware has been developed, and the YOLOv4-tiny algorithm has been implemented on it with an inference time of 471 ms per image. The results are presented, and their performance is evaluated.*

Keywords *- Crater detection, CNN, Deep learning, FPGA, Object detection, YOLO.*

1. Introduction

Autonomous landing missions are undertaken to carry out in situ measurements on extra-terrestrial bodies. Currently, a considerable number of interplanetary missions have been accomplished, and even many more are being planned shortly. The landing site for a mission is chosen based on various considerations, viz., terrain with minimal hazards, an area rich in geological features etc. To accurately soft-land on a particular area of interest, estimates of the lander craft's positional parameters are essential. In addition, GPS-based navigation and real-time communication with ground stations are not available in extra-terrestrial landing missions. Navigation based on inertial sensors is prone to error accumulation in the long term (due to the integration of biases and noise). It needs periodic corrections from other sensors for accurate estimates of the positional parameters. Typically, vision-based sensors are employed for this purpose. Based on these inputs, the guidance system activates the on board thrusters/actuators for necessary course correction and subsequent landing of the lander craft.

2. Lander Mission Scenario and Requirements

Landing missions aim to soft land safely within a predetermined distance (typically within 100m radii) of the planned landing spot. Lunar landing is accomplished in various progressive phases, namely the De-orbit phase; powered descent phase; approach phase; and terminal descent phase. The navigation and guidance system driving the lander craft must have an accurate estimate of its position in the descent trajectory to control the lander craft and transit from the current operational phase to the next operational phase. In the absence of GPS and real-time ground communication, the lander craft has to rely only on board sensors for the required estimates. Measurements only from inertial systems lack the accuracy based on which reliable on board decisions can be made. These inertial systems must be corrected periodically with inputs (Position, velocity, altitude etc.) from additional sensors.

The lander craft's position needs to be estimated at a larger scale (i.e. concerning a global coordinate system during powered descent and approach phase at a higher altitude) and also at a local scale (i.e. within an image at higher resolution during the terminal descent phase, just before touch down). For position estimation, Terrain Relative Navigation (TRN) scheme is employed. In TRN, the instantaneous position of the lander craft is estimated by comparing terrain features (sensed on board using the camera) with a reference map generated a priori.

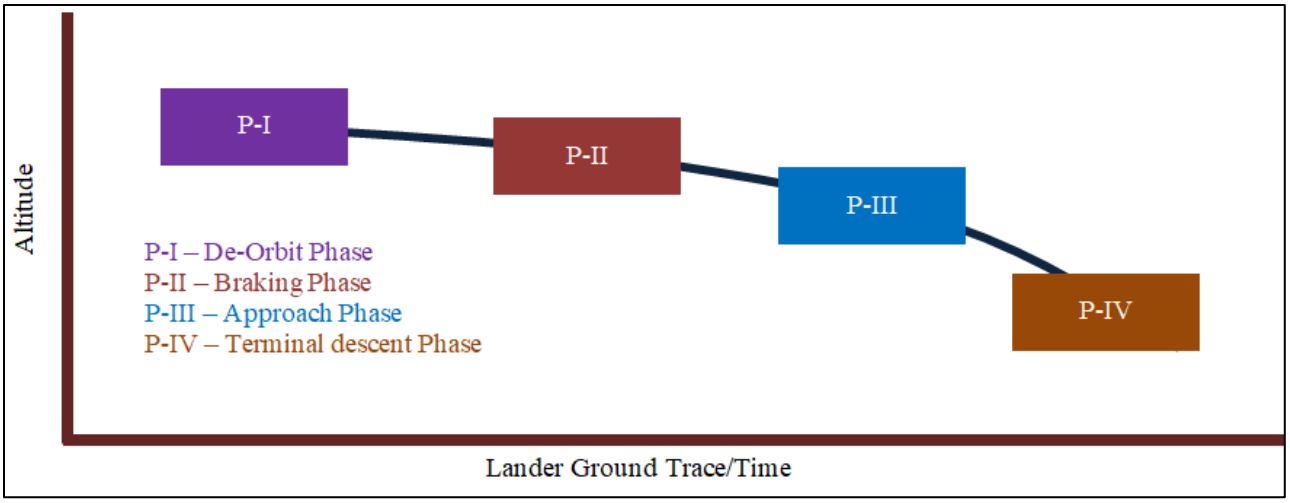

Fig. 1 Lander descent profile

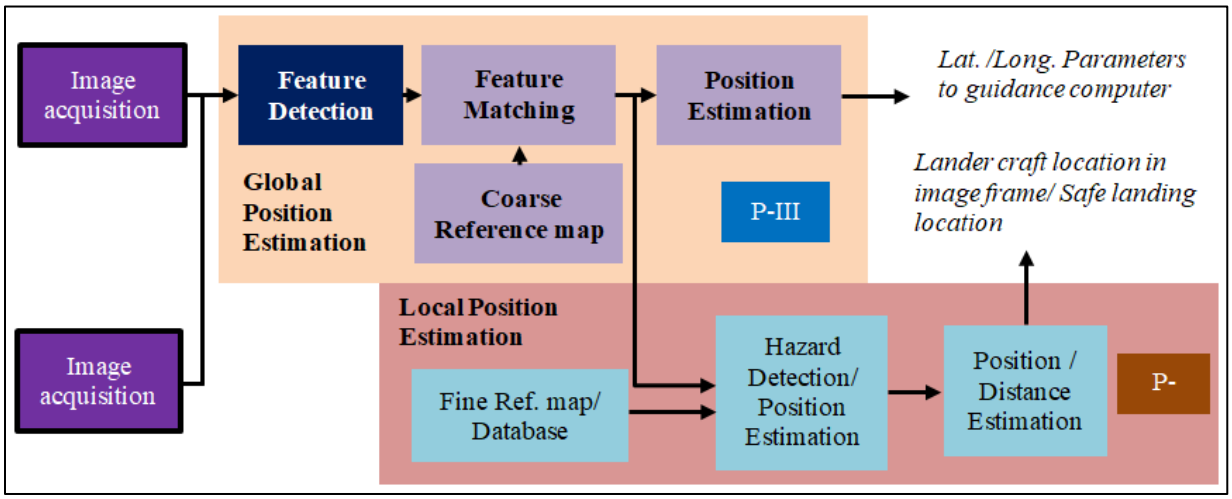

Fig. 2 Onboard processing scheme for a typical soft landing mission

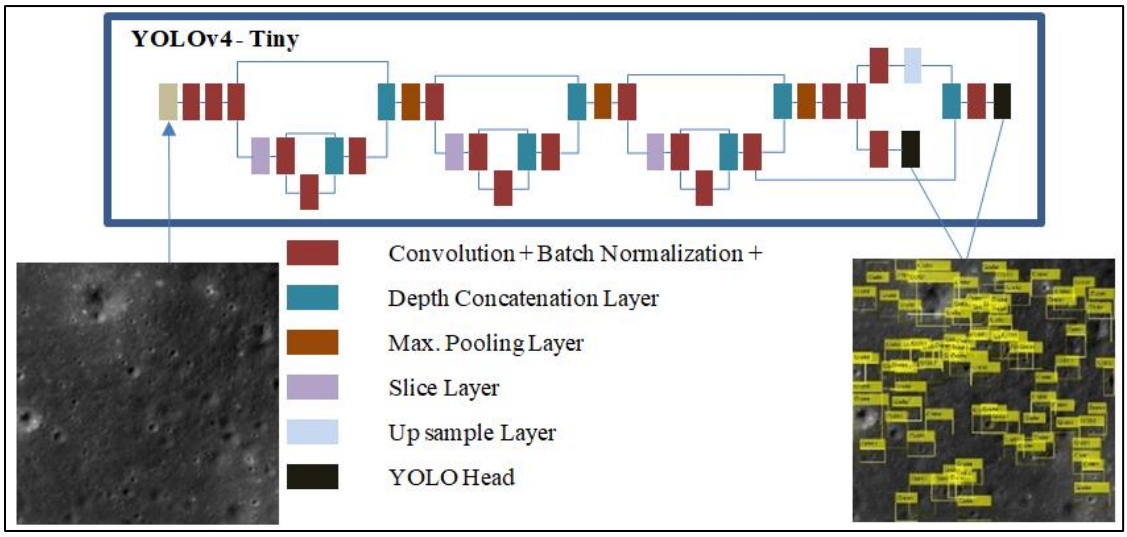

Fig. 3 Block diagram of Yolov4-tiny model

The lander descent profile and mission scenario are depicted in Figure 1 and Figure 2. This work focuses on using deep learning-based models for real-time feature detection (highlighted box in Figure 2). Once the features are detected, they are matched with the catalogue/reference database for position estimation. The reference database/catalogue of the landing area of interest is generated using a database of images acquired during previous orbital missions. The metric distance between the identified features is used as a signature for the matching operation. The catalogue stored on board consists of feature maps covering a larger swath around the area of interest.

3. Landmarks for Vision-based Position Estimation

Craters are the most abundant landmarks are lunar surfaces. Craters are bowl-shaped depressions on the surface of extra-terrestrial bodies, which are formed by the frequent bombardment of meteorites. The life span of medium to large-sized craters is quite large since their degradation or deformation is slow, particularly in atmosphere-less bodies similar to our moon. Due to their longevity and uniqueness in distribution, craters are ideal candidates to be used as landmarks for the geolocation of lander craft on alien bodies. [1] Apart from craters, image-based landmarks have also been used for position estimation. [2-7] for the current work, craters have been used as landmarks for TRN.

4. Crater Detection Techniques

Studies of crater densities and their distribution are carried out to estimate the age of the extra-terrestrial body. Manual counting of craters has been traditionally used, which is a tedious process and restricted to cataloguing larger craters. Researchers have employed various automatic techniques for crater detection. Both image-based (Panchromatic) and elevation map-based (DEM/DTM) data [5-9] are utilized for Crater Detection Algorithm (CDA). Specifically, elevation-based datasets are used for offline identification and classification of craters and subsequent cataloguing in a database. Traditionally handcrafted methods were devised for automatic crater detection. This involved identification and extraction of crater feature like rims, shapes, unsupervised learning, template matching, SIFT etc. [2-3]. However, these methods are not robust enough to varying illumination conditions, scale changes, object rotation and imaging geometry.

5. Deep Learning based Techniques for Object Detection

Deep learning has become the latest tool for solving problems which are/were difficult to counter with rule-based/handcrafted techniques. Convolutional Neural Networks (CNN) has been employed in a variety of computer vision-based applications.

Object detection [11-12] involves identifying the exact location of the objects in an image or video apart from its classification. Two broad categories of techniques are applied for object detection, namely multi-stage and single-stage detector. In the multi-stage detection technique, the input image is segmented into various regions using algorithms like selective search etc. Subsequently, these segments of images are passed on to a Convolutional Neural Network (CNN) to extract various features. The extracted features are used for classification using a Support Vector Machine (SVM) followed by bounding box regression. The multi-stage methods are accurate, albeit at the expense of increased computation. Some of the widely used multi-stage algorithms are the R-CNN, Fast R-CNN, Faster R-CNN etc. [13-17] Single-stage object detectors apply a single neural network to obtain both position and classification in a single pass. The most widely used single-stage object detection models are YOLO (You Only Look Once), YOLOv2, YOLOv3, YOLOv4, SSD etc. [18,30] These single-stage detectors are fast and with low latency, making them ideal candidates for real-time applications.

6. YOLO-based Crater Detection

YOLO (You Look Only Once) is a popular framework used for object detection. Other object detection networks employ multiple stages to accomplish object detection tasks. YOLO, as the name suggests, employs a single stage to detect the objects in an image along with the probability score conveying the confidence of its predictions. YOLOv4-tiny is a trimmed-down version of YOLOv4. [30] YOLOv4-tiny has fewer layers than the original YOLOv4 model, and only two detection heads exist. YOLOv4-tiny [20-21] is much simpler and faster to compute when compared to a full-fledged YOLOv4 model. Considering the real-time processing requirements for crater detection, the YOLOv4-tiny model has been selected for prototype implementation. YOLOv4-tiny uses CSPDarknet-53 as the backbone for feature extraction. Figure 3 gives the block diagram of the YOLOv4-tiny algorithm.

7. Data Curation and Training

We use lunar images from NASA's Lunar Reconnaissance Orbiter (LRO). A Lunar Reconnaissance camera (LROC) is a system of three cameras mounted on the LRO. LROC consists of two NACs (Narrow-angle camera with 0.5m resolution and 5 km swath) and a WAC (Wide Angle Camera with 100m resolution and a 60 Km swath). Specifically, images from Narrow Angle Camera (NAC) were used for training (Figure 4). Figure 5 gives the photograph of the LROC NAC camera. As no labelled crater data sets were available, data curation and manual data labelling were carried out. About 500 images were manually labelled using the Yolomark tool [22-23].

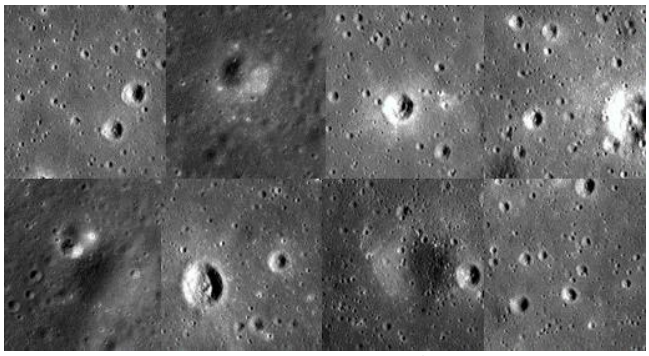

Fig. 4 LROC NAC images used for training

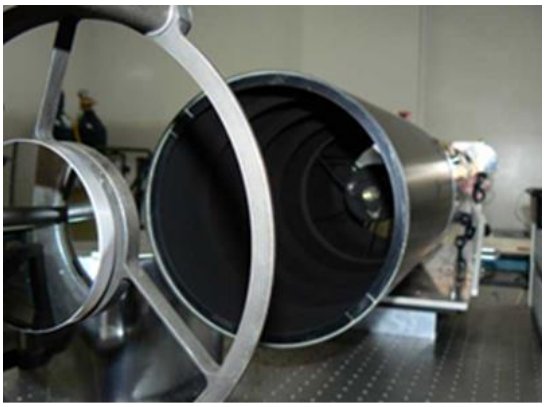

Fig. 5 LROC NAC camera [25]

A total of about 10000 labelled craters were used for training. Google Colab platform [24] was used for training the YOLOv4-tiny model, and it took 14 hours for training. The training configuration is given in Table 1. Figure 6 gives the training loss curve.

8. Prototype Hardware Configuration and Implementation

Onboard digital processing hardware for space missions is constrained by stringent SWaP (Size, Weight and Power) requirements. Apart from the SWaP requirements, all space missions typically employ radiation-hardened devices to withstand harsh radiation environments. The functional requirements of onboard space-borne digital hardware are classified into two categories, viz. control and data/signal processing.

Control functionalities typically involve coordinating a sequence of events (like powering on the other subsystems), monitoring the health of multiple systems, telemetry etc. These are realized using radiation-hardened processors/ASICs based on LEON/PowerPC/SPARCv7 cores. Signal processing functionalities encompass a wide range of tasks like filtering, data compression, parameter estimation etc. FPGAs, with their concurrent processing capability, are the ideal candidates to implement signal/data processing algorithms requiring high processing throughput.

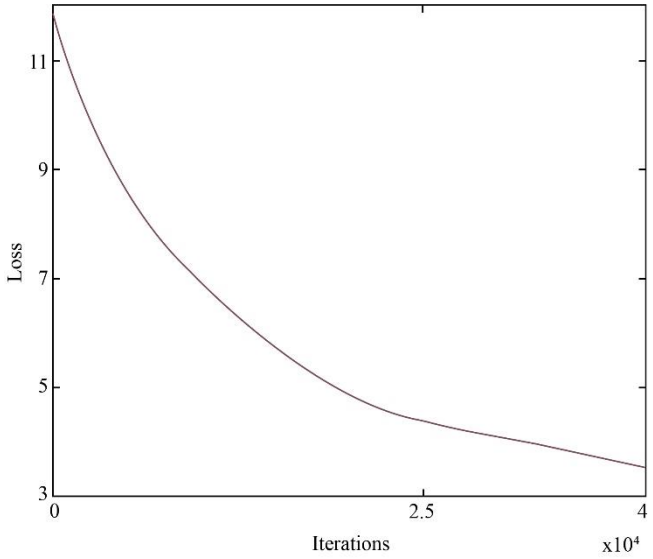

Fig. 6 Training-loss curve

Table 1. Training parameters and configuration

Parameter	Specification
Image type	Panchromatic
Source	LROC, NAC
Resolution and Swath	0.5m and 5 Km
Input image dimensions	608 x 608 x 3
Total images for training and validation	500
Total number of craters	~10000
Number of training iterations	40000
Anchor boxes used for training	[116 90; 156 198; 373 326] [30 61; 62 45; 59 119] [10 13; 16 30; 33 23]
Mini batch size	64
Learning rate	0.0026
Hardware platform for training	NVIDIA A100-SXM4-40GB
Train test validation	80%:10%:10%

Various radiation-hardened FPGAs are employed in various missions with the different underlying technology, namely Xilinx Virtex-5QV FPGA [19] (SRAM-based technology), RTG4 (Flash-based technology), RTAX2000 (Anti-fuse) etc. These devices do not have sufficient logic resources to implement a DNN-based model for inference.

For EDGE-based inference, usually, GPUs (NVIDIA Jetson Nano, NVIDIA T4 etc.) are the first choice. GPUs and other commercial embedded processors are unsuitable for reliable operation in the harsh space environment. Also, purely processor-based architecture lacks the parallel processing power of FPGAs. On the other hand, implementing deep learning models on FPGA is tedious and time-consuming.

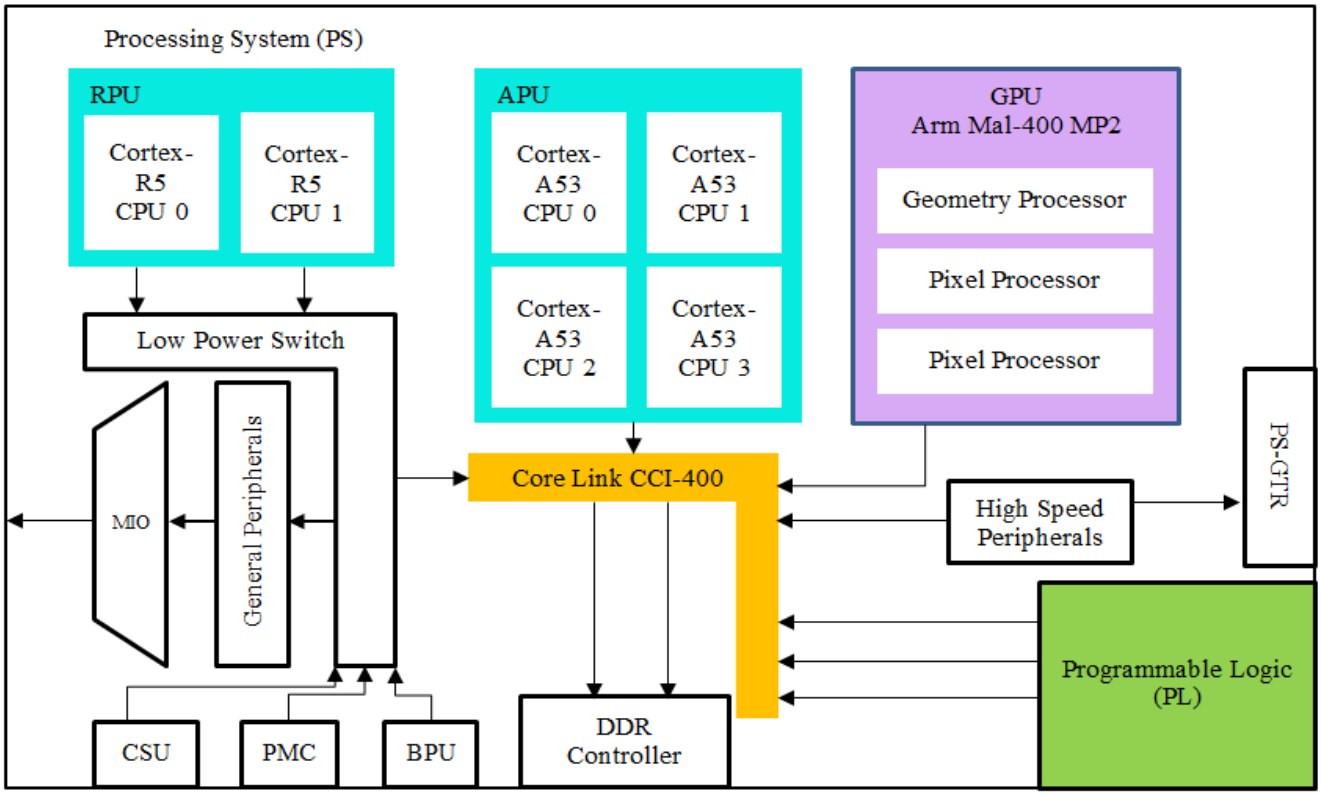

Fig. 7 MPSoC processor architecture [26]

Table 2. Specifications and performance of MPSoC-based hardware

Parameter	Specification
FPGA	XCZU7EV
PL Operating Frequency	200 MHz
Memory	DDR 4GB + 1 GB
Control Interface	MIL 1553B
Camera Interface	LVDS / RS422 / Ethernet
Programming/ Debug interface	USB
Total Inference time with overheads	558.79 ms.
Effective inference time without overheads	470.44 ms.

An ideal processing platform would be a combination of both CPU and FPGA functionalities on a single chip. Xilinx MPSoC FPGA [27] (Figure 7) combines the best of both worlds, as it is a combination of both CPU and FPGA. It consists of Quad-core ARM cortex A53 processor, dual-core Arm R5F processor, and programmable FPGA fabric.

A hardware platform (Figure 8) based on Xilinx MPSoC FPGA to carry out crater detection based on YOLOv4-tiny has been developed. It is based on Xilinx MPSoC SOM (System On Module) mounted on a customized daughter card which has various interfaces such as MIL1553, Ethernet, LVDS, RS422, USB etc.

Fig. 8 Photograph of prototype hardware

Inbuilt LVDS/RS422/Ethernet interfaces are available for ingesting data from the optical camera. The hardware has MIL 1553 interface to communicate the results to the mission/navigation computer. The specifications of the hardware are given in Table 2.

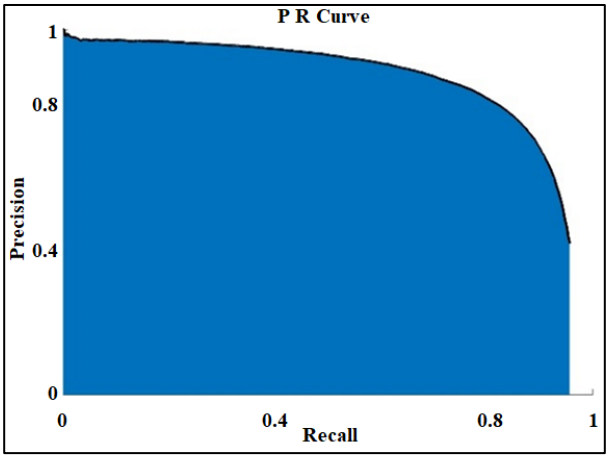

Fig. 9 Precision-recall curve

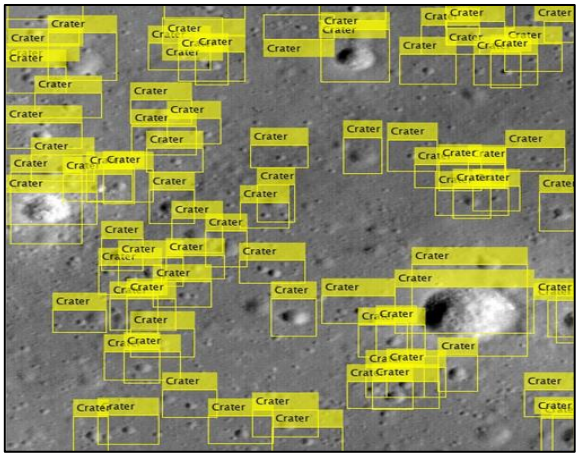

Fig. 10 Crater detection output using MPSoC hardware

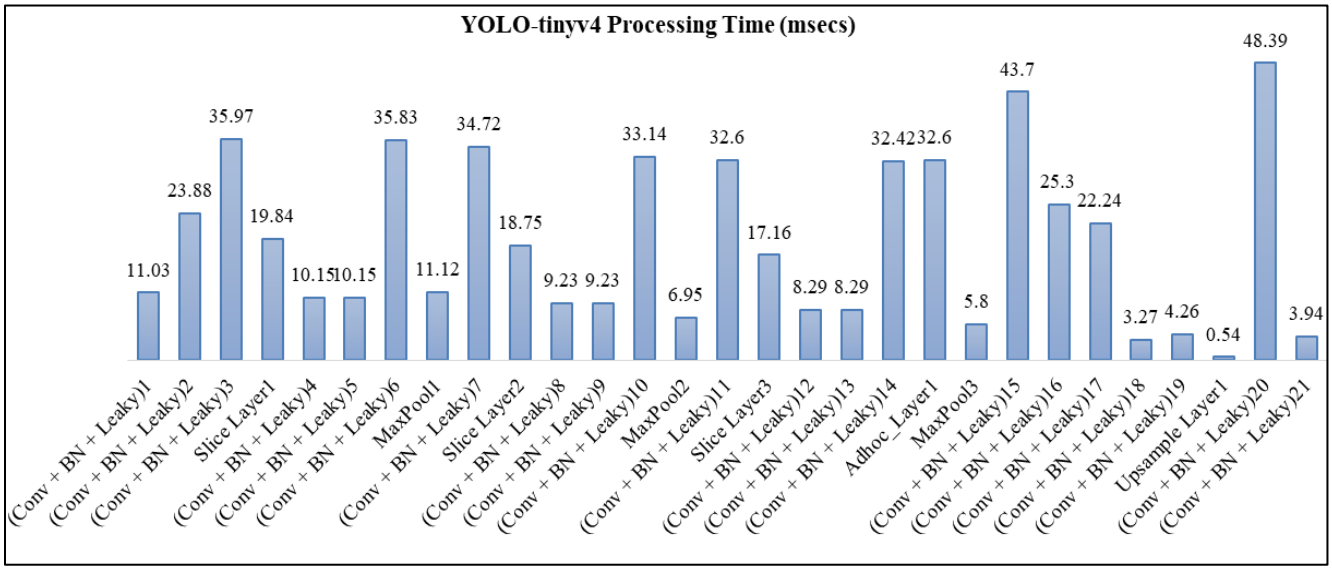

Fig. 11 Layer-wise inference time breakup

9. Results and Performance Evaluation

To evaluate the performance of the YOLOv4-tiny detector, the following metrics, namely, precision, recall, and mean Average Precision (mAP), were employed.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

The trained model was evaluated on 50 (~1000 craters) images, and a mean average precision (mAP) of 75% @ IoU = 0.5 was achieved. Figure 9 gives the Precision-Recall curve, and Figure 10 gives the YOLOv4-tiny predicted input from the prototype hardware.

A high-level synthesis (HLS) approach was adopted for YOLOv4-tiny detector implementation on the MPSoC processor. MATLAB Deep Learning HDL toolbox was extensively used to implement the various layers of the object detector. MATLAB does not provide a YOLOv4-tiny detector, which can be directly synthesized for hardware implementation. The YOLOv4-tiny model trained in the Colab environment was imported to MATLAB. The Deep Learning HDL library did not support a few layers. For example, the slice layer (For channel slicing) is not a supported layer and was replaced with a 2D convolution layer with filters tuned to perform the function of channel slicing.

With this configuration, the total processing time per frame comes out to be 558.79 ms. The effective processing, not considering the slice layers, comes out to be 470.44 ms. Figure 11 gives the layer-wise processing time breakup.

10. Conclusion and Future Work

This paper investigated the application of deep learning-based techniques for use in autonomous soft landing missions. We employed an off-the-shelf YOLOv4-tiny object detection algorithm in the position estimation pipeline for the feature detection process.

- A hardware test bed based on Xilinx MPSoC FPGA was developed to prototype the YOLOv4-tiny model
- About 10000 manually labelled craters were used to train the YOLOv4-tiny model in the google Colab environment, and a mAP of 75% was achieved. The mAP can be further improved by increasing the datasets, training iterations and fine-tuning the hyper-parameters parameters.
- The object detector was prototyped on Xilinx MPSoC FPGA-based hardware, and an effective inference time of 471 ms (with floating-point weights and biases) was achieved. This inference time may be sufficient for the global position estimation phase of a soft landing. However, for the local position estimation/hazard detection phase, the inference time has to be kept as low as possible owing to onboard fuel constraints of the lander craft
- Further reduction in inference time is possible by weight-bias quantization, network pruning, paralleling the implementation using hand-coded RTL etc.
- In future, we plan to extend this work to improve the mAP of detection and implement the entire position estimation pipeline, including feature matching, feature database generation etc. and evaluate the same on the prototype hardware.

Acknowledgements

The authors would like to thank Dr C V N Rao [DD-MRSA] and Shri N M Desai [Director SAC] for their guidance and encouragement.

B. Saravana Kumar and Dr Nagendra Gajjar contributed equally to this work.

References

[1] Yang Cheng et al., “Optical Landmark Detection for Spacecraft Navigation,” *AAS/AIAA, Astrodynamics Specialist Conference*, 2003. [Google Scholar] [Publisher Link]

[2] Adnan Ansar, and Yang Chen, “Vision Technologies for Small Body Proximity Operations,” *i-SAIRAS 2010*, 2010. [Google Scholar] [Publisher Link]

[3] Nikolas Trawny et al., “Visual Aided Inertial Navigation for Pin Point Navigation using Observations of Mapped Landmarks,” *Special Issue on Space Robotics PART-III*, pp. 357-378, 2004. [Google Scholar] [Publisher Link]

[4] Bach Van Pham, Simon Lacroix, and Michel Devy, “Visual Landmark Constellation matching for Spacecraft Pinpoint Landing,” *AIAA Guidance, Navigation and Control Conference*, 2009. [Google Scholar] [Publisher Link]

[5] Min Chen et al., “Lunar Crater Detection Based on Terrain Analysis and Mathematical Morphology Methods Using Digital Elevation Models,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no.7, pp. 3681-3692, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Andrew E. Johnson, and James F. Montgomery, "Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing," *2008 IEEE Aerospace Conference*, Big Sky, MT, USA, pp. 1-10, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[7] Yunfeng Yan, Donglian Qi, and Chaoyong Li, “Vision-Based Crater and Rock Detection Using a Cascade Decision Forest,” *IET Computer Vision*, vol. 13, no. 6, pp. 549-555, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Ari Silburt et al., “Lunar Crater Identification via Deep Learning,” *Icarus*, vol. 317, pp. 27-38, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Christopher Lee, and James Hogan, “Automated Crater Detection with Human-Level Performance,” *Computers & Geosciences*, vol. 147, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] D. Sheema et al., "An Algorithm for Detection and Identification of Infestation Density of Pest-Fall Armyworm in Maize Plants using Deep Learning based on IoT," *International Journal of Engineering Trends and Technology*, vol. 70, no. 9, pp. 240-251, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Zhengxia Zou et al., “Object Detection in 20 Years: A Survey,” *Proceedings of IEEE*, vol. 111, no. 3, pp. 257-276, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Licheng Jiao et al., “A Survey of Deep Learning-Based Object Detection,” *IEEE Access*, vol. 7, pp. 128837–128868, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[14] Ross Girshick et al., “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ross Girshick et al., "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440-1448, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[16] Shaoqing Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kaiming He et al., "Mask R-CNN," *IEEE International Conference on Computer Vision*, pp. 2980-2988, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[18] Joseph Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," *CVPR*, 2016. [Google Scholar] [Publisher Link]

[19] Ramya R, and Madhura R, "FPGA Implementation of Optimized BIST Architecture for Testing of Logic Circuits," *SSRG International Journal of VLSI & Signal Processing*, vol. 7, no. 2, pp. 36-42, 2020. [CrossRef] [Publisher Link]

[20] Shanyong Xu et al., "YOLOv4-Tiny-Based Coal Gangue Image Recognition and FPGA Implementation," *Micromachines*, vol. 13, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Sergio Saponara1 et al., "Developing a Real-Time Social Distancing Detection System Based on Yolov4-Tiny and Bird-Eye View for COVID-19," *Journal of Real-Time Image Processing*, vol. 19, pp. 551–563, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] [Online]. Available: https://github.com/AlexeyAB/Yolo_mark

[23] [Online]. Available: https://github.com/AlexeyAB/darknet

[24] [Online]. Available: https://colab.research.google.com

[25] [Online]. Available: https://lroc.asu.edu/about

[26] L. H. Crockett et al., *Exploring Zynq MPSoC: With PYNQ and Machine Learning Applications*, Strathclyde Academic Media, 2019. [Google Scholar] [Publisher Link]

[27] [Online]. Available: https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

[28] S. Bhuvaneswari et al., "Disease Detection in Plant Leaf using LNet Based on Deep Learning," *International Journal of Engineering Trends and Technology*, vol. 70, no. 9, pp. 64-75, 2022. [CrossRef] [Publisher Link]

[29] Y. Sawabe, T. Matsunaga, and S. Rokugawa, "Automated Detection and Classification of Lunar Craters Using Multiple Approaches," *Advances in Space Research*, vol. 37, no. 1, pp. 21-27, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[30] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv.2004.10934, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[31] Ms.Manjula B.M, and Dr.Chirag Sharma, "FPGA Implementation of BCG Signal Filtering Scheme by using Weight Update Process," *SSRG International Journal of VLSI & Signal Processing*, vol. 3, no. 3, pp. 1-7, 2016. [CrossRef] [Google Scholar] [Publisher Link]