AIVIONIC - Artificial intelligence techniques in on-board avionics and software

Robert Hinz[(2)], Baltazar Parreira[(1)], João Belfo[(1)], Marcos Quintana[(2)], Paulo Rosa[(1)], Jesus Balsalobre[(2)], Francisco Membibre[(2)], Antonio Latorre[(2)], Fintan Buckley[(3)], Jose Luis Espinosa-Aranda[(3)], Elena Hervas-Martin[(3)], Jesús Gamero-Tello[(3)], Ilaria Bloise[(4)], Lorenzo Feruglio[(4)], Mattia Varile[(4)], Stefano Silvestrini[(5)], Margherita Piccinin[(5)], Andrea Brandonisio[(5)], Paolo Lunghi[(5)], Michèle Lavagna[(5)], Amit Sahu[(6)], Michell Guzman[(6)], Holger Pfeifer[(6)], Luis Mansilla[(7)], Christophe Honvault[(7)]

[(1)]Deimos Engenharia, Av. Columbano Bordalo Pinheiro 75, 9A1, 1070-061 Lisboa, Portugal. Phone +351 21893 3010. {baltasar.parreira, joao.belfo, paulo.rosa}@deimos.com.pt
[(2)]Deimos Space, Ronda de Poniente 19, 28760 Tres Cantos, Spain. Phone (+34) 91 806 34 50. {robert.hinz, marcos.quintana}@deimos-space.com
[(3)]Ubotica Technologies, Camino de Moledores s/n, Incubadora de Empresas, local 0.32, 13005, Ciudad Real, Spain. Phone +353 878229594. {fintan.buckley, elena.hervas, josel.espinosa, jesus.gamero} @ubotica.com
[(4)]AIKO, Via dei Mille 22, 10123, Torino, Italy. Phone +39 3291127762. {ilaria, mattia}@aikospace.com
[(5]Politecnico di Milano, Department of Aerospace Science and Technology, Via G. La Masa, 34, 20156, Milan, Italy. Phone +39-02-2399-8364. {michelle.lavagna, margherita.piccinin, stefano.silvestrini, andrea.brandonisio, paolo.lunghi}@polimi.it
[(6)]fortiss GmbH, Guerickestraße 25, 80805 München, Germany, {pfeifer, guzman}@fortiss.org
[(7)]ESA-ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands. Phone: +31 71 565 6565. {christophe.honvault, Luis.Mansilla}@esa.int

ABSTRACT

Artificial Intelligence (AI) has revolutionized industries by disrupting traditional methods and enabling ground-breaking applications. Its potential has extended to the space domain, particularly in Earth Observation. Enhanced AI algorithms and powerful processing units facilitate new applications, especially for onboard implementations.

The AIVIONIC project aims to implement a dependable **HW/SW demonstrator of an AI-based Visual Navigation System**, showcasing AI in space critical systems. Neural Network (NN) algorithms were carefully chosen based on on-board implementability and adaptability. Lightweight, modular AI pipelines with Object Detection and Keypoint Regression Networks met processing and latency constraints. Rigorous validation encompassed design, prototyping, training, and implementation on target hardware. Synthetic and laboratory image datasets and extensive Monte Carlo campaigns assessed navigation performance and robustness. AI runtime monitoring supported algorithm validation during operation.

AIVIONIC successfully developed a dependable HW/SW demonstrator reaching Technology Readiness Level (TRL) 4. The AI techniques meet mission requirements in accuracy, latency, flexibility, and reusability. AIVIONIC addresses challenges of data availability and AI dependability in space critical systems. This paper provides insights into challenges, system design, validation strategies and Image Processing (IP) and Navigation results.

1 INTRODUCTION

The use of AI has been recognized as a major advance in several industries including Automotive and Agriculture, disrupting several traditional approaches and leading to a myriad of novel applications. The space domain has also been reached by the innovation potential of AI, with ESA promoting AI for space in activities such as AI4EO. Modern AI methods show a series of remarkable advantages:

- The ability to explicitly model **nonlinear environments**.
- The **robustness to uncertainty**, especially because AI methods are typically trained with sets obtained from dispersed configurations of parameters, which inherently account for such uncertainties. This is especially true for Deep Learning algorithms, which can process previously unseen inputs thanks to the extraction of deep features.
- The **superior performance levels** when compared to classical approaches, with AI methods unleashing new applications that would otherwise be infeasible, such as face recognition, tagging, etc.

An application that could greatly benefit from these characteristics is the autonomous vision-based spacecraft navigation, as there is a demand of highly accurate relative navigation in non-linear and unpredictable harsh environments. However, the computational complexity of AI methods has long prevented or limited their use for onboard applications. Thankfully, the **increasingly more powerful processing units**, together with **enhanced and less computationally intensive AI algorithms**, make it possible to explore new AI applications. Such approaches can now efficiently be implemented in resource restricted embedded hardware, which renders the use of AI onboard a spacecraft and within the avionics system onboard spacecraft a remarkable target to be pursued.

Nevertheless, a significant number of **challenges** still exist regarding the use of AI techniques in on-board avionics. While challenges such as the computational burden of AI algorithms, as mentioned above, are well addressed, both from the HW and the SW side, solutions for the validation of AI implementations that can be conveyed to space applications are a rather new field of research. This challenge is primarily motivated by the gap between the superior performance of AI and the lack of a complete theoretical understanding of the behaviour of those algorithms. Moreover, due to the intrinsic nonlinear structure of the AI algorithms, the generalization of the learned features to different scenarios is not straightforward. While linear approaches guarantee that the results obtained at a given linearization point are satisfied within a region around that point (in the space of parameters), AI algorithms typically do not provide such certainty. Therefore, Monte-Carlo (MC) simulations are not guaranteed to cover the whole space of potential configurations of parameters, as minor changes in the inputs of the AI can lead to completely different results in the output.
When it comes to the Space sector, this challenge is further amplified by the lack of existing large real-world data sets which are a prerequisite for the training and validation of AI algorithms.

The goal of the AIVIONIC technology development project is to implement a HW/SW demonstrator of an AI-based Visual Navigation System that can be applied to mission scenarios of *Lunar landing* and *Rendezvous and Capture of a Non-Cooperative Target*. This follows a novel development line towards demonstrating the use of AI in space critical systems, in a dependable manner.

The AIVIONIC project aims to achieve this goal performing the following tasks:

1) Description of the field of avionics problems that AI can solve and the related technology solutions. This is fundamental to lay the basis for a clear understanding of the potential of

most recent AI algorithms, and their applicability to the most challenging vision-based navigation scenarios defined in the proposed activity.

2) Development of a vision-based relative navigation system using AI, in particular under low-illumination conditions. Traditional vision-based relative navigation algorithms suffer from non-ideal conditions, especially in terms of illumination level and relevant uncertainties. The outstanding generalization capability and flexibility of deep neural networks offer a promising solution for such shortcomings. The definition of a flexible architecture for AI implementation is important for the sake of space application, where the architecture can be employed in several scenarios.
3) Identification and analysis of the capabilities of AI-based navigation solutions given the mission requirements, especially in comparison with traditional approaches considered for benchmarking purposes, for the selected scenarios Pinpoint Landing on the Moon and Rendezvous and Capture of an Uncooperative Target.
4) Implementation of a HW and SW demonstrator: Design and prototype a flexible architecture of hardware and software elements which demonstrate the execution of onboard AI vision-based navigation for the use case scenarios, using an open architecture capable of being repurposed for other navigation scenarios with little change. This includes the design, training and validation of the appropriate AI networks and the development of the software algorithms for the selected navigation scenarios.
5) Identification of validation and verification methods able to assess the quality of the results. AI algorithms follow a model-based, data-driven approach. For space applications which are often highly safety critical and expensive, fail-safe capabilities and robustness guarantees are required. Hence, a critical evaluation of the approach is paramount, especially in AI applications for GNC and on-board autonomy. Validation methods analyse the model response to input variations. Verification approaches check for input domain coverage, verify risk properties, and provide runtime monitoring.
6) Validation of the overall AI-based navigation solution for the use case scenarios and verification of the behaviour of the system against the mission requirements. Monte-Carlo simulation campaigns, with images generated by software (PANGU), are used to demonstrate the performance and robustness achieved by the overall navigation solution. In addition, a set of tests will be performed by using real images, acquired by a camera in a representative scenario, making use of an established visual robotics lab from POLIMI.

2 SCENARIOS AND AI MODEL DEVELOPMENT

A literature review of state-of-the-art AI-based algorithms for the two different mission scenarios was performed, focusing on on-board implementability. Subsequently, the identified techniques were scored on a number of criteria and traded-off in order to find the most promising solutions. The considered criteria were:

- Expected performances
- Training and data needs
- Interpretability and explainability
- Embedded systems and resources
- Adaptability and Flexibility

The baseline solutions were then implemented, trained, and their performances were assessed. The following subsections present additional details on the mission scenarios, implemented solutions and obtained performances.

2.1 Rendezvous and Capture of an Uncooperative Target

The reference mission scenario is based on a deorbiting mission with ENVISAT as a target. The focus lies on the final capture phase (operational range between 50m and 0.5m), with the chaser starting in a station keeping location. At this point, a synchronous fly-around motion is acquired, and the forced approach to reach capture distance is performed while the chaser remains co-rotating with the target. The sensors available for navigation are: Camera, Inertial Measurement Unit, and Star tracker. The AI-based Image Processing (IP) function will then provide to Navigation the target pose estimation obtained from monocular camera images.
The selected AI technique follows the modular architecture presented in [1], and shown in Figure 1.

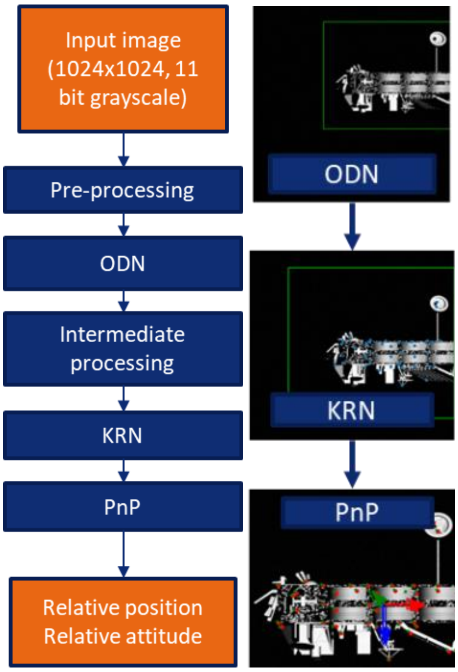

Figure 1: AI architecture for the rendezvous scenario.

This technique relies on defining a set of keypoints in the target S/C geometry, which are then identified by the AI model. The Object Detection Network (ODN) is tasked with finding the Region of Interest (ROI) in the image where the target S/C is visible. The Keypoint Regression Network (KRN) then estimates the position of the predefined keypoints in the cropped image. Finally, PnP provides the pose estimation based on correspondences between the observed 2D keypoint positions and 3D-wireframe positions.
The models are trained using synthetic images of the ENVISAT satellite that covers the space of different relative positions, attitudes and illumination conditions. This is complemented with the ground truth metadata corresponding to the coordinates of the 2D bounding box containing the target S/C, as well as the projected 3D keypoints positions in the cropped image.

2.2 Lunar landing scenario

The reference mission scenario considers a S/C descent on the Lunar South Pole from an altitude of 100 km, down to 3 km. There are two parts of the scenario that use vision-based AI-aided algorithms for:

- Absolute Navigation, during Parking Orbit and Coasting Phase;
- Relative Navigation, during Main Brake and Final Approach Phase.

The sensors available for navigation are camera and altimeter. The selected AI technique is a modular architecture inspired by the work from [2]. An ODN based on a Single-Shot Detector (SSD) is at the core of the AI-based Image Processor, identifying craters visible in the terrain.

Although the same observables are provided by the AI solution in absolute and relative navigation, they are treated differently depending on the sub-phase of the landing mission:

- Absolute Navigation: Detected craters are matched to a known landmark database (see Figure 2, left);
- Relative Navigation: Detected craters are treated as features to be matched between consecutive images (see Figure 2, right).

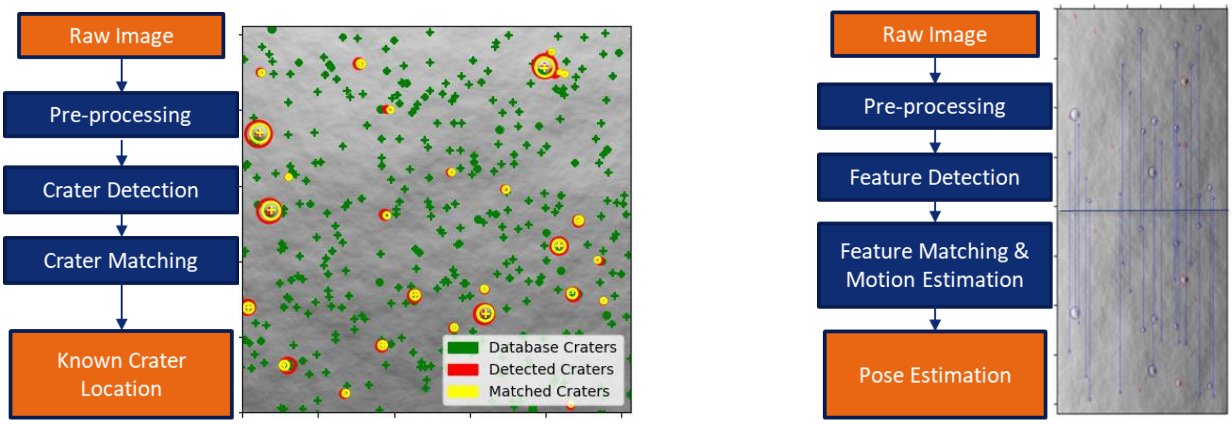

Figure 2: AI architecture for landing scenario. Left: Absolute Navigation. Right: Relative Navigation

The models are trained with synthetic images of the lunar terrain, covering different areas and illumination conditions. The ground truth metadata is the list of craters (radius, center coordinates) visible in the images.

3 SYSTEM DESIGN AND IMPLEMENTATION

3.1 HW and SW demonstrator for Lunar Landing and Rendezvous and Capture of a Non-Cooperative Target.

The HW/SW demonstrator is composed of the following main elements:

- **FLIGHT Element**: All SW/HW comprising the Visual Navigation System (VNS): AI-based IP, Navigation, HW representative of future space components.
- **PREP Element**: All SW/HW required to perform the development and preparation of the VNS: development environment, training/test harness, scripts for training/test data gathering.
- **EGSE Element**: All SW/HW elements required to perform the formal verification and validation testing of the VNS. This allows providing the test vectors, control the execution of the system, and record the results.

The HW/SW architecture of the system is shown in the following figure.

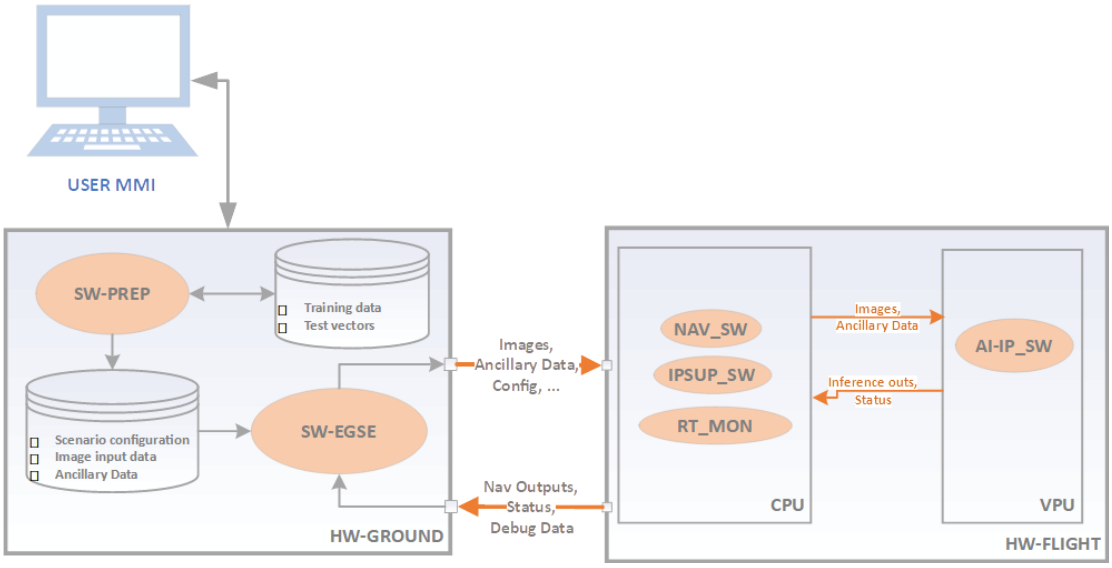

Figure 3: HW/SW Architecture.

The HW platform for the VNS system is composed of:

- A general-purpose CPU, providing the platform for the execution of the Navigation, and the supporting and monitoring functions for the AI-based IP.
- A VPU, dedicated for the implementation of the AI part of the image processing function.

The VPU selection was performed considering a number of criteria such as: power consumption, performance, and configurability for different applications. The final selection converged on the CogniSat™ Accelerator, which uses a Myriad 2 VPU (Lunar Landing Scenario) or Myriad X (Rendezvous Scenario) for both AI inferencing and hardware accelerated computer vision algorithms. The Myriad 2 has flight heritage from the PhiSat-1 mission [4].

The onboard system environment is shown in Figure 4. The selected OBC is the FZ3 Card, based on Xilinx Zynq UltraScale+. The highly performant device features a 1.2 GHz quad-core ARM Cortex-A53 64-bit application processor, a 600MHz dual-core real-time ARM Cortex-R5 processor, a Mali400 embedded GPU and rich FPGA fabric. Ubotica's CogniSat™ platforms are connected to the OBC over Ethernet (Myriad 2) and USB (Myriad X).

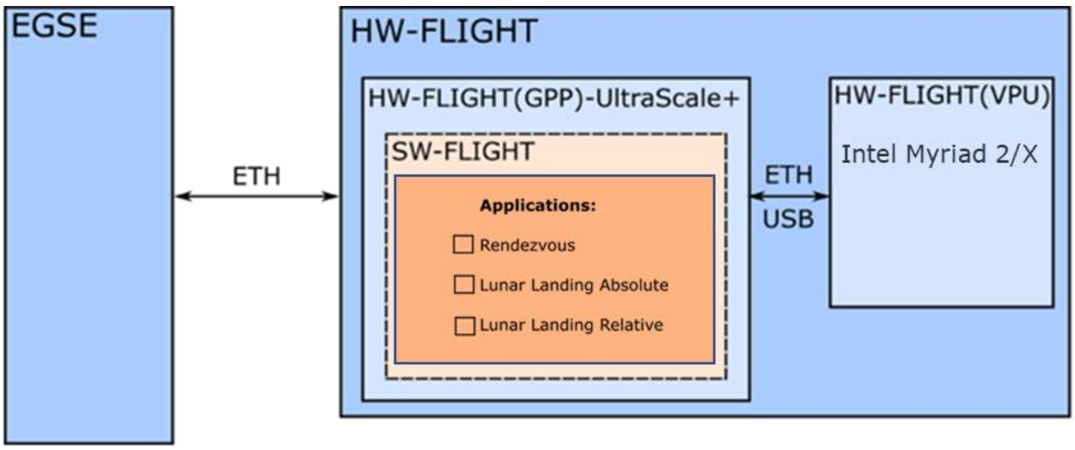

Figure 4: Onboard system environment

The physical HW setup for the FZ3 board and Myriad 2/X is shown in Figure 5.

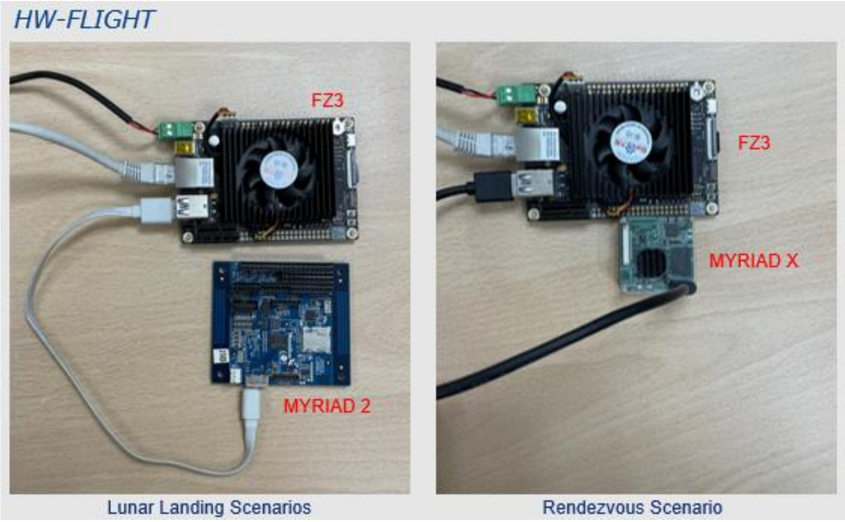

Figure 5: HW-FLIGHT. Left: Configuration with FZ3 and Myriad 2, used in lunar landing scenario. Right: FZ3 and Myriad X, used in the Rendezvous Scenario**.**

3.2 Development of a vision-based relative navigation system using AI

The overall architecture of the software components for the Rendezvous Scenario and for the absolute and relative Lunar Landing scenario are shown in Figure 6, Figure **7** and Figure **8**, respectively.

For the Rendezvous Scenario, shown in Figure 6, the camera module is providing the image to the AI-based Image Processing Module. Then the estimated pose is used by the navigation system for estimating the chaser spacecraft state with respect to the target spacecraft.

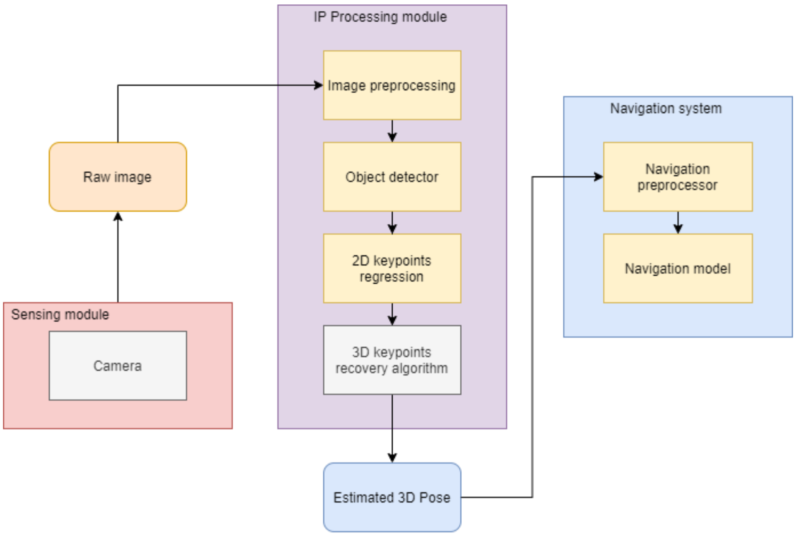

Figure 6: Overall architecture of the system components for the Uncooperative target scenario.

In Figure 7 the Absolute navigation diagram is depicted. Absolute navigation relies on detection of the known craters (using the labelled crates from available dataset) by AI-model. The AI outputs are then fed in the navigation block to have the control action to perform the landing. In the case of Absolute Navigation, the AI outputs will be related to craters detection.

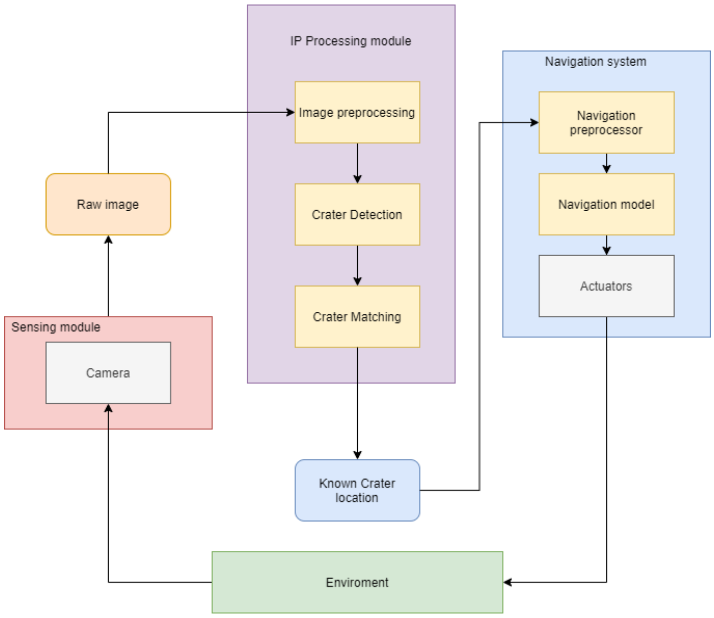

Figure 7: Overall architecture of the software component for the Lunar Landing scenario in Absolute Navigation.

For Relative Navigation shown in Figure 8, the pipeline differs mainly in the outputs. In this case the information fed to the navigation block consists of the estimated pose of the spacecraft.

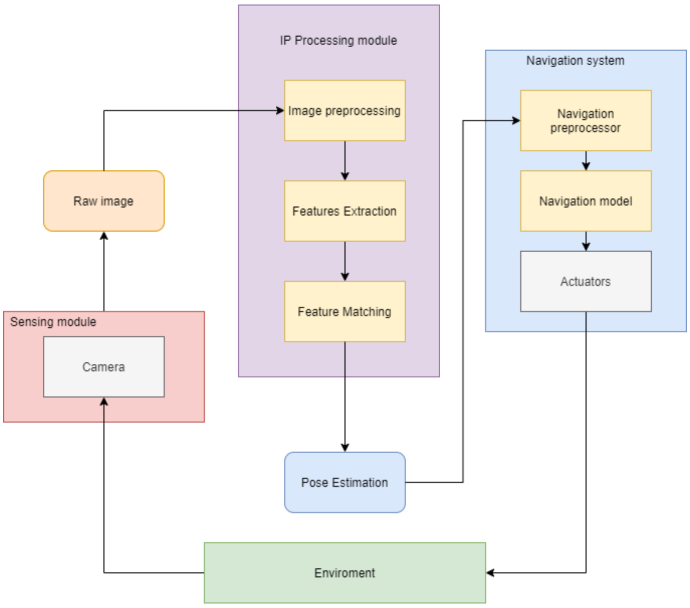

Figure 8: Overall architecture of the software component for the Lunar Landing scenario in Relative Navigation.

3.3 Validation and Verification

Validation plays a major role for safety- and mission-critical elements, such as those related to GNC for space vehicles. Therefore, the use of AI as part of the Navigation or any decision-taking element of the GNC requires proper validation. The proposed AI validation logic follows two complementary approaches:

- **AI development process**, referring to all the steps to be followed in the AI development, starting from the design, and ending in the implementation and validation in the target HW.
- **AI runtime monitoring**, referring to the active monitoring of the AI algorithms while in operation.

The considered **development process** is composed of the following steps:

- **AI Design, Training and Prototyping:** Starting from the AI Design, a first prototype of the AI solution is implemented and then used for training. AI training is an iterative process where the weights of the AI model are adjusted to improve performances. The process continues until the AI is shown to be providing the required performances on the validation dataset. The selection of the layer to be monitored at runtime can be done following the conclusion of the AI training & validation.
- **Dataset preparation:** The datasets required for training, validation and testing are defined and subsequently generated. The datasets consist of multiple images (synthetic and laboratory), together with the corresponding ground truth meta-data required for AI training and validation. Laboratory images are used for validating the synthetically generated images, as well as for testing of the complete system.
- **Implementation and Testing:** Once the AI model training is complete, it can be used for testing of the complete AI-Navigation chain, but also for implementation in the target HW. The AI validation in the VPU aims to verify if the AI operation matches results obtained previously at prototype level. At this stage it is also necessary to generate the final NAP database and check the operation of the AI runtime monitor.

For the online verification of AI results in both scenarios the **runtime monitor** approach based on FORTISS' Neural Network Dependability Kit [3] was selected. The concept of runtime monitoring is based on the fact that one can only expect reliable performance from the AI model when it is being applied to a datapoint with similarities to the training data. Fundamentally, the AI behaviour during training and operation could be different if the two datasets do not have an equivalent data distribution.
The runtime monitoring is put in place at different stages:

- During training, datapoints can be tracked by recording their Neural Activation Pattern (NAP) on a pre-selected layer, and stored in a database.
- During runtime, the NAP of the datapoint provided to the AI during operations is compared with the NAP database using the hamming distance. If the pattern is not found in the database, then a warning is generated, indicating that the model's output is not supported by the training data.

For the landing scenario, the Neural Network (NN) architecture consists of an SSD network built using MobileNetv2 features. The object detection happens at multiple resolutions. Hence, multiple layers are used as features. Since the most high-level features are in the penultimate layer, these layers were used for neural activation pattern (NAP) generation.
To further identify the most important neurons, the contribution of each neuron for the classifications was calculated using weights from feature to the classification layer.
For the rendezvous scenario, out the two AI models used (ODN and KRN), the KRN was selected for NAP generation as it extracts features specific to the uncooperative target. Due to computation limitations, only limited neurons based on their importance are selected for each keypoint and concatenated to form the NAP.

4 RESULTS

4.1 Rendezvous and Capture of an Uncooperative Target

As illustrated in Figure 9, the **verification of the output of the PC and Myriad implementations** of the ODN and KRN (ROI and detected keypoints, respectively) show good agreement.

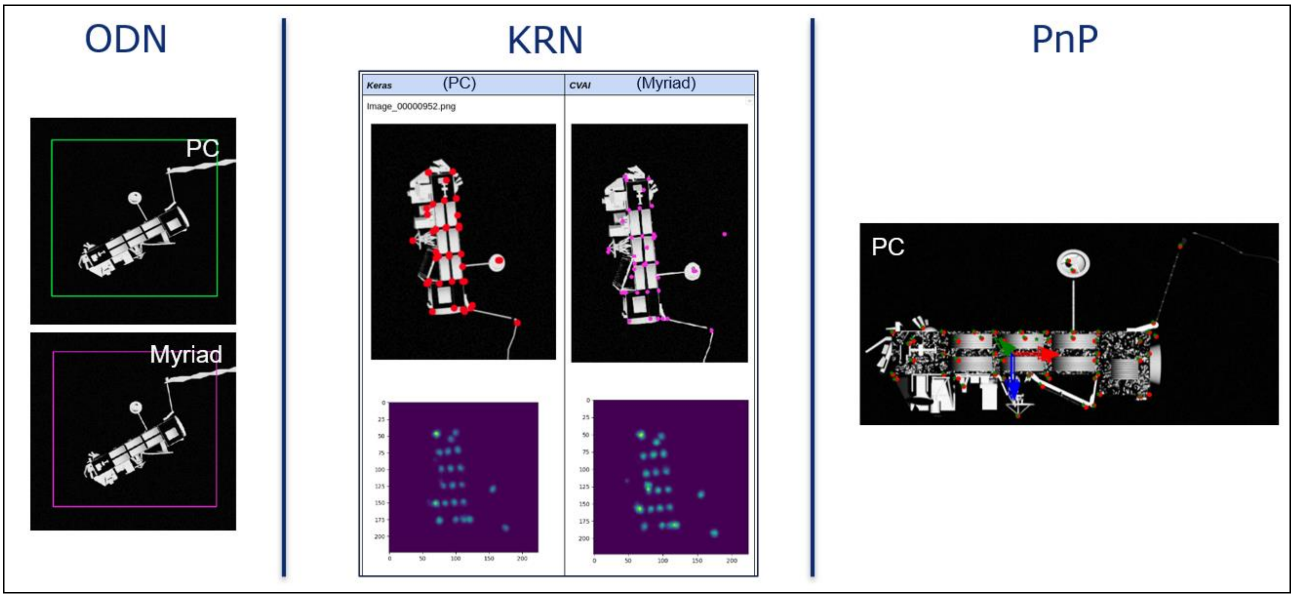

Figure 9: Comparison between PC and Myriad implementations of the ODN (ROI, left) and KRN (detected keypoints, centre), and between the detected keypoints and estimated pose with the respective ground truth (right)

The agreement with the ground truth pose (Figure 9, right) is given by the **pose estimation accuracies** presented in Figure 10.

	Er	Et	Et(x)	Et(y)	Et(z)
mean	0.67	0.05	0.01	0.00	0.00
std	0.97	0.08	0.02	0.03	0.08
min	0.01	0.00	-1.20	-1.63	-4.44
25%	0.28	0.02	0.00	-0.00	-0.03
50%	0.46	0.04	0.01	0.00	0.00
75%	0.77	0.06	0.02	0.01	0.03
99%	3.26	0.18	0.05	0.05	0.14
99.7%	4.55	0.24	0.08	0.07	0.20
max	50.44	4.87	0.14	0.13	0.32

Figure 10: Pose estimation performances. Er – Rotation error [deg], Et – Absolute translation error [m], Et(x), Et(y), Et(z) – Translation error in the x-, y- and z-component [m]

The **Navigation errors** with these IP performances which were obtained in a MC (Monte-Carlo) campaign are shown in Figure 11 for the approach phase of the scenario. The results confirm the robustness of the system to the uncertainties considered in the MC campaign, since the MC shots do not deviate much from the nominal scenario. While the relative attitude results are well within the requirement level (5 deg), the translation error is high in general, but decreasing with decreasing distance chaser-target, meeting the requirements (25cm) towards the end of the approach phase.

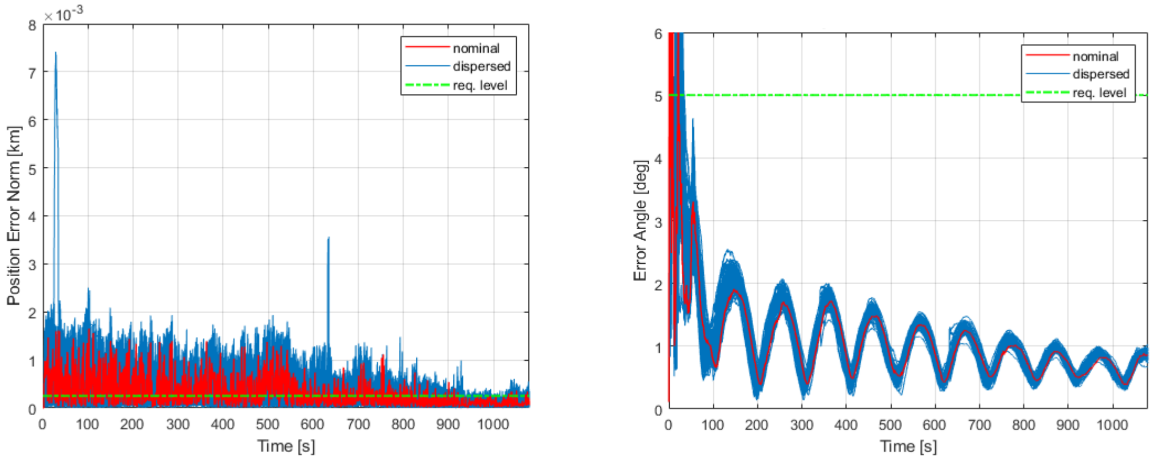

Figure 11: Navigation results with IP inputs for approach phase. Left: Translation error. Right: Attitude error.

Runtime Monitor performance was assessed by performing trigger tests with four different sequences containing OoD (out-of-domain) images. Each test vector consisted of 30 images, with 10 images from the nominal test vector, followed by 10 OoD images, followed again by 10 nominal images. Images were rejected if not found in the NAP database or less than 4 Keypoints were detected. As an example, for the glint effect OoD image shown in Figure 12, 65% (13 of 20) of nominal images are correctly found in the NAP database, and 100% (10 of 10) OoD images are correctly not found in the NAP database. While this is presented here for one specific test vector, the outcome is representative of the overall result.

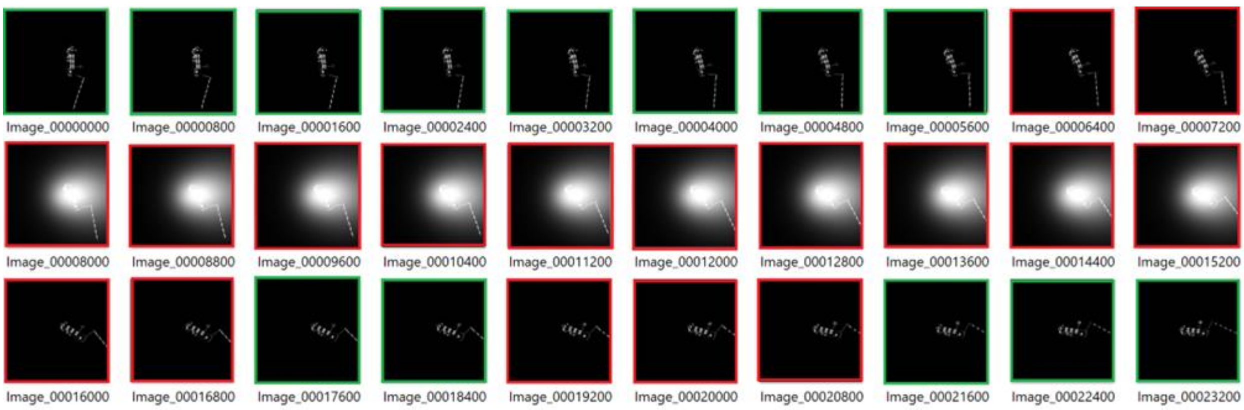

Figure 12: Glint OoD image trigger tests for the Rendezvous Scenario. Red: Images not found in the NAP database or with key points < 4; Green: accepted images

The **latency** between image input and Navigation output was measured to be below the requirement level of 800 ms.

4.2 Lunar landing scenario

In terms of the IP verification of the output of the PC and Myriad implementations, there is a good agreement between the PC and Myriad implementations (Figure 13).

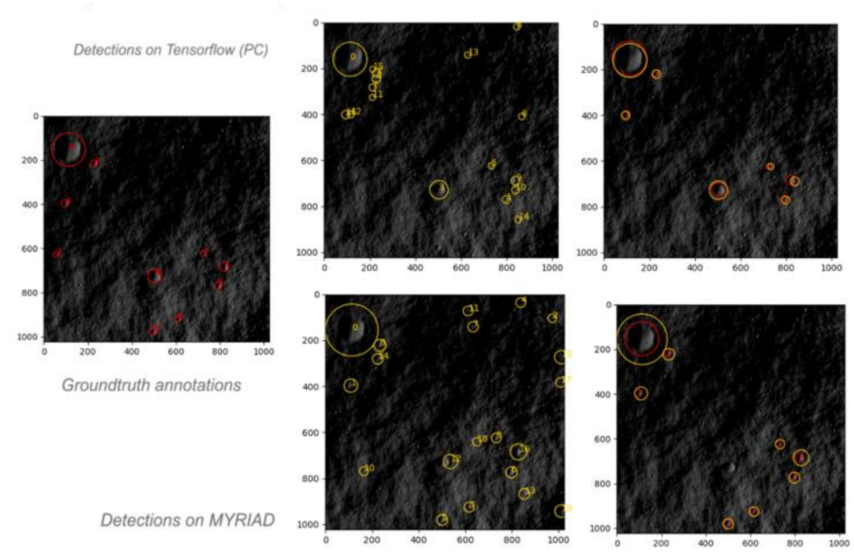

Figure 13: Comparison between PC (top) and Myriad implementations (bottom) of the crater detection network. The ground truth annotations are also shown (left)

The system is able to detect the set number of craters (60) in all of the test images. The crater detection accuracies that can be obtained are shown in Figure 14. The selected AI model results were found to be sufficient for Navigation to meet the performance requirements, while minimizing the inference time. The default maximum value of 60 craters is detected in all images.

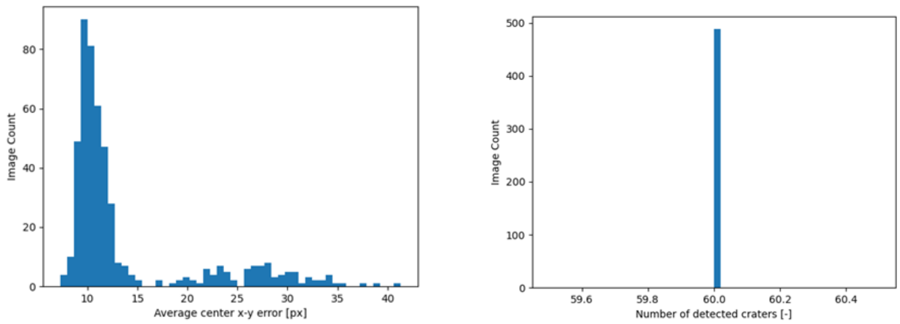

Figure 14: Crater detection performance.

The **Navigation errors** which can be obtained with these crater detection performances are shown in Figure 15 and Figure 16, for Absolute and Relative Navigation, respectively.

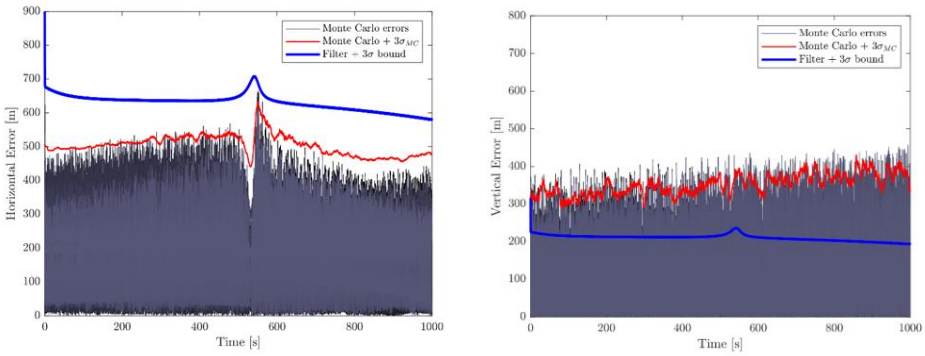

Figure 15: Navigation performances in landing scenario, absolute navigation. Left: Horizontal error, right: Vertical error.

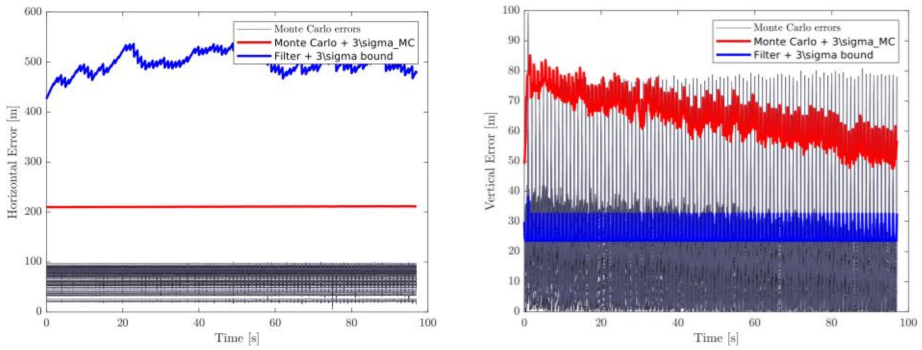

Figure 16: Navigation performances in landing scenario, relative navigation. Left: Horizontal error, right: Vertical error.

The results follow closely what is expected from the nominal, undispersed trajectory, which meets the requirements (Table 1). The requirements are met in the great majority of the MC shots, with only few outlier cases: For Absolute Navigation, 15 % of the samples exceed the horizontal error requirement at least at one instant in time of which on average 28 instants in the trajectory out of requirements (~3% of the time). For Relative Navigation, 5 % of the samples fail and diverge, and 15 % of the samples exceed the horizontal error requirement at least at one instant in time.

Table 1: Requirements on the horizontal and vertical error for Absolute and Relative Navigation.

Requirements	**Distance to surface**	**Horizontal Error**	**Vertical Error**
Absolute Navigation	**80 km altitude:**	230 m	430 m
	50 km altitude:	240 m	610 m
	15 km altitude	90 m	230 m
	10 km altitude:	60 m	90 m
Relative Navigation	**3 km altitude:**	30 m	50 m

Runtime Monitor performance was assessed by trigger tests as described above for the Rendezvous Scenario. For pixelated OoD images, 100% (20 of 20) of nominal images were correctly found in the NAP database, and 80% (8 of 10) were correctly not found .

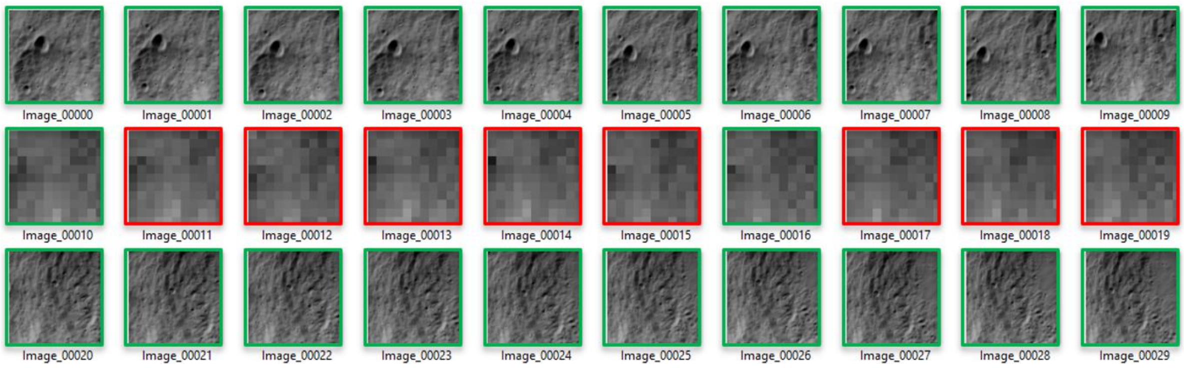

Figure 17: Pixelation OoD image trigger tests for the Lunar Landing Scenario. Red: Images not found in the NAP database; Green: accepted images

The **latency** between image input and navigation output was measured to be below the requirement level of 1s for both Absolute and Relative Navigation.

5 CONCLUSION AND OUTLOOK

The objective of **developing a HW/SW demonstrator for a vision-based relative navigation system using AI in a dependable manner** has been achieved by the AIVIONIC study. It shows that AI techniques reach the accuracies and latencies to meet the requirements, and provide advantages in terms of flexibility and reusability. Data availability and AI dependability methods play a key role for the development and use of AI in space critical systems, and AIVIONIC provides successful solutions for both. Accordingly, as detailed in this paper, the development entails a series of achievements:

- **Development of a vision-based relative navigation system using AI**. Suitable AI techniques for the two use case scenarios Lunar Landing and Rendezvous and Capture of a Non-Cooperative Target were identified and selected by performing a trade-off analysis of state-of-the-art techniques considering the specific mission characteristics. Lightweight, modular AI processing pipelines were selected to account for the onboard processing resource restrictions and latency requirements. For the Rendezvous and Capture of a Non-Cooperative Target scenario, the onboard algorithm implementation needs further development to replicate state-of-the-art AI performances in order to meet the IP requirements of the relative navigation system. The Lunar Landing scenario demonstrates that the system can meet the required latency, IP detection and navigation performances.
- **HW and SW demonstrator for Lunar Landing and Rendezvous and Capture of a Non-Cooperative Target.** The consortium has designed and prototyped a flexible architecture of hardware and software elements which demonstrate the execution of onboard AI vision-based navigation for two use-case scenarios. The open architecture can be repurposed for other navigation scenarios with little change. Appropriate AI models for the identified scenarios were trained and validated, and the software algorithms to implement the selected navigation scenarios using these models was developed. Intel's Myriad2/MyriadX Vision Processing Unit together with the Zynq UltraScale+ from Xilinx was used, for both image pre-processing and AI inference for the Flight elements of the architecture. The architecture can support multiple VPUs for both redundancy and performance. A Ground support element (EGSE) was implemented to configure the Flight Elements and to analyse the navigation output results. A full system integration and validation of the HW/SW demonstrator was performed for the Lunar Landing scenario, reaching TRL 4. For the Rendezvous and Capture of a Non-Cooperative Target scenario, system elements were integrated in the target platforms and validated separately in the form of unit tests; full systems tests were also performed at prototype level
- **Identification of validation and verification methods, and validation of the overall AI-based navigation system, considering the specifics of AI validation and using supporting AI validation tools.** A validation process was identified and defined, using both synthetic and laboratory image data sets for AI-IP verification and validation. Extensive Monte Carlo campaigns were designed and executed for both use cases, allowing to measure the impact of input data variations, including such in the image data, on the overall navigation performance and to assess the robustness of the AI-IP – Navigation pipeline. The runtime monitor approach based on FORTISS' Neural Network Dependability Kit [3] was identified as a tool to support AI algorithm V&V, allowing online validation of AI-IP results, and the runtime monitor performance was analysed. The neural activation pattern runtime monitor was implemented in both use case scenarios as a verification method for input domain coverage and it was shown that it can successfully filter out-of-domain input images to the neural network.

The following activities are proposed to consolidate the use of AI in vision-based relative navigation. **Closed-loop validation of IP – GNC** in a laboratory facility (camera + robotic arm). This is considered the next logical step to increase the TRL of the developed system. **An in-orbit demonstration mission**, for example as part of an existing close proximity operations mission (piggyback payload) in order to obtain realistic measurements relative to another spacecraft and test GNC capabilities in an open-loop manner. The **Validation in real-world environment**, by flying the integrated system in an adapted/scaled real-world scenario, e.g., on a helicopter or drone, to validate it in a realistic closed-loop manner. The **further study of online runtime monitoring and offline AI validation tools** to increase their computational efficiency and monitoring performance, and to obtain robustness guarantees. An **extended campaign for laboratory image generation** using a robotic facility for the validation of synthetic images and training and testing of AI-based image processing, building on the experience obtained during this study. **AI improvements** in terms of accuracy and computational complexity, by keeping track of the quickly changing availability of DL architectures for onboard implementation, and using further data augmentation and domain adaptation techniques to narrow the gap between the synthetic training and laboratory test domains. **Investigation of additional sensors** (TIR, LiDAR) to overcome illumination limitations inherent in the camera use in this study, and to provide additional measurements to improve the AI and navigation estimation performance.

AKNOWLEDGEMENTS

This work has been funded by ESA, contract number 4000129466/19/NL/CRS: “Artificial intelligence tech-niques in on-board avionics and software”.

REFERENCES

[1] Tae Ha Park, Sumant Sharma, Simone D'Amico. Towards Robust Learning-Based Pose Estimation of Noncooperative Spacecraft. arXiv:1909.00392.2019

[2] Silburt A, Ali-Dib M, Zhu C, et al. Lunar crater identification via deep learning. Icarus. 2019;317:27-38. doi:10.1016/j.icarus.2018.06.022

[3] Cheng, Chih-Hong & Huang, Chung-Hao & Nuhrenberg, Georg. (2019). nn-dependability-kit: Engineering Neural Networks for Safety-Critical Autonomous Driving Systems. 1-6. 10.1109/ICCAD45719.2019.8942153

[4] G. Giuffrida et al., "The Φ-Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-14, 2022, Art no. 5517414, doi: 10.1109/TGRS.2021.3125567.