Applying Deep Learning for Optical Navigation in Solar System Exploration Missions

by

Alfredo Escalante Lopez

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Aerospace Engineering

Universidad Carlos III de Madrid

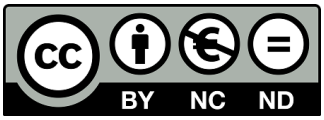
Advisor(s):

Manuel Sanjurjo-Rivo
Pablo Ghiglino

Tutor:

Manuel Sanjurjo-Rivo

December 2025



Dedicado a mi familia.

Acknowledgements

First and foremost, this thesis would not have been possible without the invaluable support of my supervisors, Manuel Sanjurjo and Pablo Ghiglino. Their continuous support, both technical and personal, throughout these years has undoubtedly been the main reason why this project has successfully reached completion. I am equally grateful for the opportunity to have collaborated with and learned from the Klepsydra team, particularly Mandar Harshe and Manuel López.

I also express my gratitude for the resources and funding provided through the R+D+i project TED2021-132099B-C31 from Ministerio de Ciencia, Innovación y Universidades, and Agencia Estatal de Investigación, MCIN/AEI (10.13039/501100011033).

I would also like to thank all those who have influenced and contributed with ideas that have become fundamental pillars of this thesis. My colleagues at ESAC and in the SPICE team—Ricardo, Rafel, Marc, and Boris—along with the entire NAIF team at JPL, inspire me every day to take on the challenges that arise. A special thank you also goes to my fellow PhD students, Pelayo and Thomas. It has been a true pleasure to collaborate and learn with you.

To my family, I owe everything and more. For as long as I can remember, my parents influence has left a profound mark on who I am. But specially, I want to thank my uncles for their support and affection, my grandparents for their unconditional love, and my dear brother for his companionship. Finally, to my wife: I will never be able to thank you enough for having the patience and courage to stand by me every day, for facing all the obstacles of this journey with me, and for being the light that guides and inspires me.

.

Published and submitted content

Papers published in peer-reviewed journals, in ascending chronological order (whenever material from these sources is included in this thesis, it is singled out with typographic means and an explicit reference):

- **A. Escalante**, P. Ghiglino and M. Sanjurjo-Rivo, "Churinet - Applying Deep Learning for Minor Bodies Optical Navigation," in IEEE Transactions on Aerospace and Electronic Systems, Volume 59, Issue 4, Pages 3566-3578, August 2023, doi: doi.org/10.1109/TAES.2022.3227497 (**Paper I**, fully included in Chapter 2).

- **A. Escalante**, P. Ghiglino and M. Sanjurjo-Rivo, "Applying Machine Learning Techniques for Optical Relative Navigation in Planetary Missions," in IEEE Transactions on Geoscience and Remote Sensing, Volume 62, Pages 1-11, 2024, Art no. 4702811, doi: doi.org/10.1109/TGRS.2024.3374454 (**Paper II**, fully included in Chapter 3).

- **A. Escalante**, P. Ghiglino and M. Sanjurjo-Rivo, "Bennunet - An Update on Applying Deep Learning for Minor Bodies Optical Navigation," in IEEE Transactions on Aerospace and Electronic Systems, Volume 61, Issue 3, Pages 7125-7139, June 2025, doi: doi.org/10.1109/TAES.2025.3533471 (**Paper III**, fully included in Chapter 4).

Other research merits

Conference proceedings papers, in ascending chronological order:

- **A. Escalante**, P. Ghiglino and M. Sanjurjo-Rivo, "Churinet - A Deep Learning Approach to Optical Navigation for Minor Bodies," in Proceedings of 72nd International Astronautical Congress, Dubai (2021).
- **A. Escalante**, P. Ghiglino and M. Sanjurjo-Rivo, "KAMnet - A Deep Learning Approach to Optical Navigation for a Dawn-Dusk Earth Observation Mission," in Proceedings of 73rd International Astronautical Congress, Paris (2022).
- **A. Escalante**, P. Ghiglino and M. Sanjurjo-Rivo, "Bennunet - Applying Machine Learning Techniques for Autonomous Optical Relative Navigation of an Asteroid," in ESA GNC and ICATT Conference, Sopot (2023), doi: doi.org/10.5270/esa-gnc-icatt-2023-015.
- **A. Escalante**, P. Ghiglino and M. Sanjurjo-Rivo, "Applying Machine Learning Techniques for Optical Navigation in Lunar Missions," in Proceedings of 75th International Astronautical Congress, Milan (2024).

Papers to which the author of this dissertation provided minor contribution, in chronological order:

- T. Frekhaug, **A. Escalante**, M. Sanjurjo-Rivo, and M. Soler, "Robust Model Predictive Control with Monocular Optical Navigation for Asteroid Circumnavigation," in European Control Conference (ECC), Pages 1938-1941, Bucharest (2023).
- P. Peñarroya, **A. Escalante**, T. Frekhaug, and M. Sanjurjo-Rivo, "A Fully Autonomous On-Board GNC Methodology for Small-Body Environments Based on CNN Image Processing and MPCs," in Aerospace, Volume 11, Issue 11, Art no. 952, 2024, doi: doi.org/10.3390/aerospace11110952.

Abstract

Exploring the Solar System is key to understanding the processes that shape celestial bodies, as well as our past and future place in the universe. Autonomous navigation has become mission-critical for exploring distant worlds, where communication delays and harsh lighting conditions limit the effectiveness of current navigation methods. Although new hardware-based alternatives exist, they are unsuitable for low-resource missions, which are rapidly spreading in the new space era, highlighting the need for lightweight, software-driven solutions that remain robust under variable lighting conditions. This thesis presents a complete framework for developing neural network–based optical navigation algorithms for Solar System exploration missions. It introduces methods to generate accurate synthetic training datasets by leveraging measurements and derived products from past space missions. These synthetic datasets are used to train Convolutional Neural Networks (CNNs), enabling robust, efficient, and lightweight autonomous navigation solutions. To support deep-space missions to small bodies, a synthetic image generation pipeline was developed using shape models and operational data from targets such as comet 67P/Churyumov-Gerasimenko. CNNs trained on these images demonstrated reliable navigation performance in scenarios where traditional techniques struggle. For planetary missions, the synthetic image generation pipeline was extended to incorporate tiled global terrain models combining elevation and spectral data. By fusing high-resolution topography and surface reflectance data from multiple sources, the resulting networks generalize across seasons, orbital regimes, and spacecraft orientations, overcoming challenges such as atmospheric perturbations and limited instrument coverage. The framework was further enhanced by modeling albedo and small-scale terrain variations, applying advanced data augmentation to bridge the gap between synthetic and real images. Custom state-of-the-art CNN architectures were implemented, significantly improving accuracy and demonstrating the benefits of time-distributed neural networks for processing sequences of images. The trained networks were validated using real navigation imagery from the OSIRIS-REx mission to asteroid Bennu, achieving operational performance even under the most restrictive illumination conditions. This work demonstrates that synthetic imagery and deep learning can jointly enable robust and lightweight optical navigation, particularly for missions with limited onboard resources. The framework is adaptable to a wide range of mission targets and provides a scalable path toward autonomous navigation for future space missions.

Keywords: *Optical navigation, machine learning, solar system missions.*

Resumen

Explorar el Sistema Solar es clave para comprender los procesos que modelan los cuerpos celestes, así como nuestro lugar en el universo, tanto en el pasado como en el futuro. La navegación autónoma es esencial para la exploración de mundos distantes, donde los retrasos en las comunicaciones y las condiciones de iluminación limitan la eficacia de los métodos de navegación actuales. Aunque existen nuevas alternativas basadas en hardware, estas no son adecuadas para misiones con recursos limitados, cada vez más relevantes en la nueva era espacial, lo que resalta la necesidad de soluciones ligeras, basadas en software y robustas frente a condiciones de iluminación variables. Esta tesis presenta un marco integral para el desarrollo de algoritmos de navegación óptica basados en redes neuronales para misiones de exploración del Sistema Solar. Se introducen métodos para generar conjuntos de datos sintéticos de entrenamiento precisos, aprovechando mediciones y productos derivados de misiones espaciales anteriores. Estos conjuntos sintéticos se emplean para entrenar Redes Neuronales Convolucionales (CNN), lo que permite soluciones de navegación autónoma robustas, eficientes y de bajo coste computacional. Para apoyar misiones en el espacio profundo a cuerpos menores, se ha desarrollado una herramienta de generación de imágenes sintéticas utilizando modelos 3D y datos operacionales de objetivos como el cometa 67P/Churyumov-Gerasimenko. Las CNN entrenadas con estas imágenes demostraron que pueden facilitar la navegación óptica en escenarios donde los métodos tradicionales resultan ineficaces. Para misiones planetarias, la generación de imágenes sintéticas se amplió incorporando modelos globales de terreno que combinan datos de elevación y espectrales. Al fusionar datos de topografía de alta resolución y reflectancia superficial de múltiples fuentes, las redes resultantes logran invarianza operacional en cuanto a época, régimen orbital y estado del satélite, superando desafíos como las perturbaciones atmosféricas y la limitada cobertura de los instrumentos. El modelo se extendió simulando variaciones de albedo y de topografía a pequeña escala, aplicando técnicas avanzadas de *data augmentation* para reducir la brecha entre imágenes sintéticas y reales. Se implementaron arquitecturas modificadas de CNNs, mejorando significativamente la precisión y demostrando las ventajas de las redes neuronales recurrentes para procesar secuencias de imágenes. Las redes entrenadas se validaron con imágenes reales de navegación de la misión OSIRIS-REx al asteroide Bennu, alcanzando un rendimiento operacional incluso bajo las condiciones de iluminación más restrictivas. Este trabajo demuestra que la combinación de imágenes sintéticas y aprendizaje profundo permiten llevar a cabo navegación óptica robusta y ligera, especialmente para misiones con recursos limitados a bordo. El marco es adaptable a una amplia variedad de misiones espaciales y ofrece una vía escalable hacia la navegación autónoma en misiones futuras.

Palabras clave: *Navegación óptica, aprendizaje automático, misiones del Sistema Solar.*

Contents

List of Figures

List of Tables

List of Acronyms

CCD	Charge-Coupled Device
CNN	Convolutional Neural Network
CNSA	China National Space Administration
CPU	Central Processing Unit
DART	Double Asteroid Redirection Test
DEM	Digital Elevation Model
ESA	European Space Agency
FoV	Field of View
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HL	High Level
ISRO	Indian Space Research Organisation
JAXA	Japan Aerospace Exploration Agency
LiDAR	Light Detection and Ranging
LL	Low Level
MAE	Mean Absolute Error
MSE	Mean Squared Error
MTE	Mean Translation Error
NAC	Narrow Angle Camera
NASA	National Aeronautics and Space Administration
NAVCAM	Navigation Camera
OCAMS	OSIRIS-REx Camera Suite
PBR	Physically Based Rendering
PDS	Planetary Data System
PSA	Planetary Science Archive

RELU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SLAM	Simultaneous Localization and Mapping
SPC	Stereophotoclinometry
SRP	Solar Radiation Pressure
TdCNN	Time-Distributed Convolutional Neural Network
UAESA	United Arab Emirates Space Agency

1. Introduction

1.1. A Retrospective on Solar System Exploration

Exploration of the Solar System has been a cornerstone of human scientific endeavor, driven by the desire to understand our place in the universe and the fundamental processes that shape planetary bodies. Since the launch of the first artificial satellite, Sputnik 1, in 1957, space exploration has rapidly evolved from Earth-orbiting satellites to robotic missions probing the farthest reaches of our Solar System and beyond. Explorer 6 was the first satellite to capture images of the Earth from space in 1959 [1] and on the same year, Luna 1 was the first manmade object to reach escape velocity, performing a fly-by to the Moon and entering heliocentric orbit around the Sun [2]. The following decades saw the first spacecrafts to reach other planets in the Solar System with Mariner 2 flying-by Venus in 1962, Mariner 4 flying-by Mars in 1965, and years later Mariner 10 flying-by Mercury in 1974 [3]. In the 70s and 80s it was the turn for the outer planets, with Pioneer 10 reaching Jupiter in 1973, Pioneer 11 flying-by Saturn in 1979 [4], followed by the famous and still alive Voyager 1 and Voyager 2 probes [5], which are now in the interstellar space, not without first exploring also Uranus and Neptune. However, in order to understand the processes shaping the celestial bodies of the Solar System longer stays at these bodies are required. To do so, a long series of orbiters have been exploring several planets and minor bodies in the last decades, overcoming numerous challenges in terms of technological developments and operations.

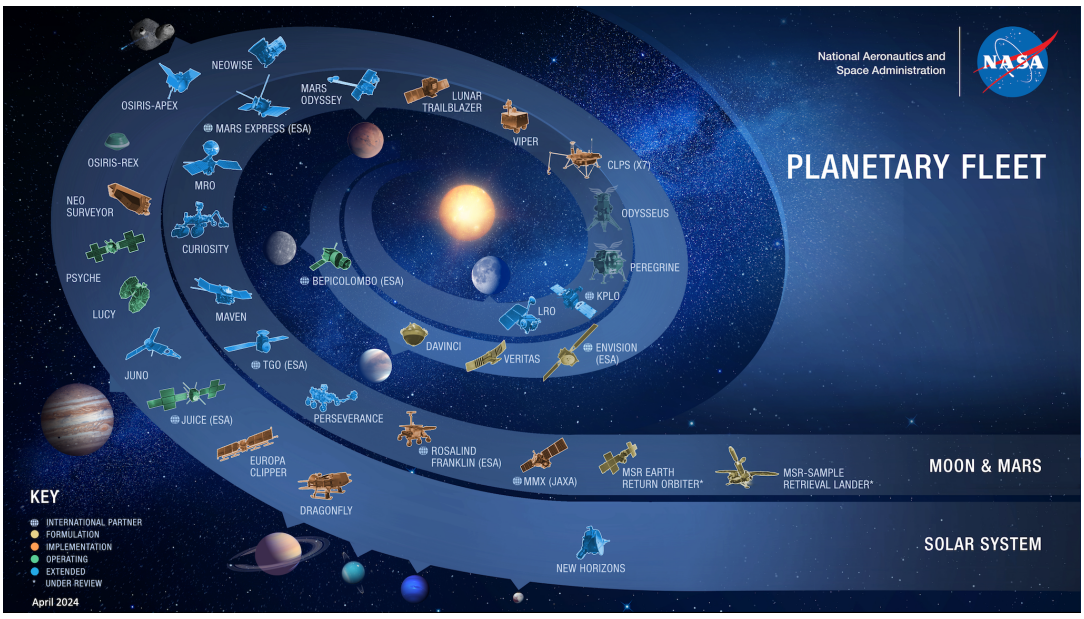


Figure 1.1: Current and in-development Solar System Explorers by NASA, ESA and JAXA (as of April 2024).

Our closest neighbor, the Moon, is the most visited body in the Solar System other than the Earth. Since Luna 10 became the first satellite to reach Lunar orbit in 1966 [2], many satellites operated by multiple space agencies in the world have explored the Moon. Lunar Orbiter 1 was the first NASA mission to orbit the Moon, exposing and sending back to Earth 205 frames with a spatial resolution up to 60 meters (thanks to an astonishing 68-kilogram Eastman Kodak imaging system) which were devoted mainly to study possible Apollo landing sites [6]. SMART-1 was the first ESA mission to reach the Moon and the first satellite to do so by using an Electric Propulsion engine, at the time a technology demonstration which laid the stones to the now flying Mercury Transfer Module of the BepiColombo mission [7]. SELENE, the second Lunar orbiter operated by JAXA, produced detailed altitude and geological data which notably improved the existing Lunar topography maps [8]. Chandrayaan-1 by ISRO produced more than 70000 three-dimensional images of the Lunar surface. Thanks to Chandrayaan's Moon Impact Probe and the Moon Minerology Mapper payload provided by NASA, the presence of water locked in minerals on the Moon was confirmed for the first time, a key element for future human permanent bases [9].

After the Earth, Mars was the first planet to be orbited by a spacecraft and remains the most extensively explored. In 1971, Mariner 9 became the first probe to enter orbit around Mars, arriving during a massive dust storm that had engulfed the entire planet. Due to the dust storm, the main surface imaging did not began until January 1972. However, surface-obscured images did contribute to the collection of Mars science, including breathtaking pictures of Olympus Mons and the three Tharsis volcanoes as seen in Figure 1.2, that gradually became more visible as the dust storm abated. This unexpected event made a strong case for the desirability of studying a planet over longer periods of time from orbit, rather than merely flying past like its predecessors. In order to achieve accurate Mars orbit insertion and operations, optical navigation demonstration in near real time was for the first time applied for an orbiter in another planet. Images from an onboard television camera were downlinked to Earth and used to improve radiometric data affected by Mars ephemeris errors [10]. Optical navigation has since then been used in almost all planetary orbiters and landers. After Mariner 9, another 13 spacecrafts have orbited Mars in the last decades, responsible of discoveries as important as the presence of water ice in the South pole and subglacial lakes below the polar cap [11], the first evidences for a stable body of liquid water on the planet. 7 spacecrafts are still doing science on Mars operated by 5 different space agencies, NASA, ESA, ROSCOSMOS, UAESA and CNSA, and the coming decades promise to see a surge in the number of orbiters in Mars to support the first human crewed missions to the Red Planet.

Venus is the second most visited planet with 8 orbiters, none of which are still active today. However, with 4 planned missions for the next decade, there is a renewed interest in the planet after the possible discovery of phosphine traces in the clouds of Venus [12], which could hint the presence of life in the less extreme environment of the upper atmosphere. The rest of the planets of the Solar System have a significantly smaller

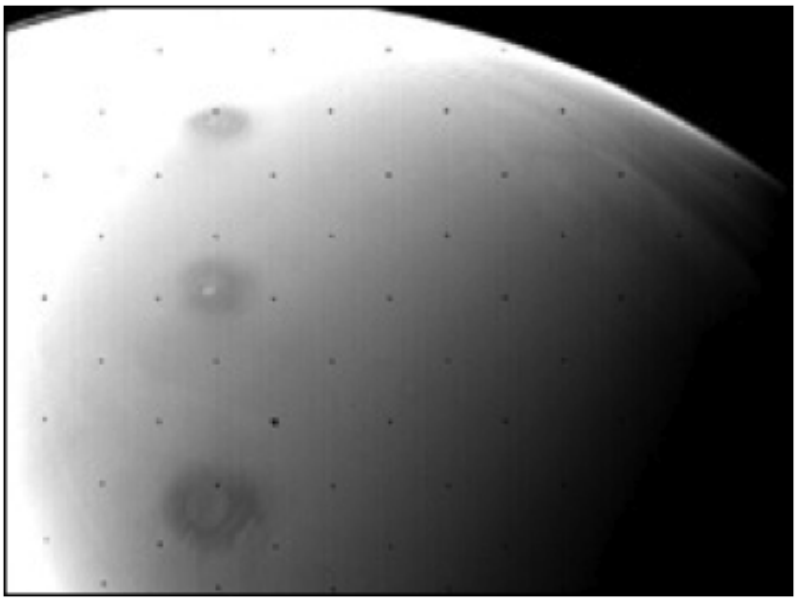


Figure 1.2: Mars Tharsis volcanoes from Mariner 9 orbiter slow scan vidicon camera with 11 by 14 degree field of view at approximately 10,000 kilometer altitude.

number of visitors due to the technical challenges to reach them and survive their hostile environment. The high heliocentric speed of Mercury (the fastest planet) combined with the gravitational pull of the Sun makes it extremely challenging to get to Mercury. MESSENGER mission by NASA, is the only one to have orbited Mercury, doing so between 2011 and 2015, providing evidence of large amounts of water in the exosphere and past volcanic activity on the surface [13]. BepiColombo by ESA and JAXA will be the second spacecraft to orbit the planet, set to arrive to Mercury in 2026. Regarding the outer planets, Jupiter, the largest planet of the Solar System, has only been orbited by two spacecrafts, Galileo in 1995, and JUNO from 2016 and still active. Saturn on the other hand has only been orbited by the Cassini orbiter between 2004 and 2017, performing also close encounters with various Saturn major and minor moons [14].

In recent decades, attention has shifted towards exploring small Solar System bodies, including asteroids and comets. The low-temperature and low-gravity environments of comets and asteroids help preserve their volatile-rich composition, making them invaluable as the most accessible samples of the primitive material from which the Solar System formed [15], [16]. The need for in-situ measurements has driven the exploration of various comets and asteroids over the past few decades. The International Cometary Explorer (ICE), became in 1985 the first spacecraft to encounter a comet when it crossed the plasma tail of Comet Giacobini-Zinner [17]. Notable missions to comets include Vega 1 and 2, which intercepted Comet Halley in March 1986 [18]; the Stardust Sample Return mission, which collected particles from the coma of Comet 81P/Wild 2 [19]; and the Rosetta mission, which became the first spacecraft to rendezvous with, and land on a comet, 67P/Churyumov-Gerasimenko, following flybys of asteroids Steins and Lutetia during its cruise phase [20], [21]. In total there have been 33 missions to asteroids

and dwarf planets, although just a few have entered orbit around its target. NEAR Shoemaker was the first mission to orbit and soft-land on an asteroid successfully in 2001 [22]; Hayabusa 1 and 2 explored asteroids Itokawa and Ryugu respectively, the later bringing samples from Ryugu back to the Earth in 2020 [23], [24]; and more recently, OSIRIS-REx mission [25] visited asteroid Bennu and collected samples that returned to the Earth in September 2023 before continuing its extended mission across the Solar System, set to visit asteroid Apophis in 2029 [26].

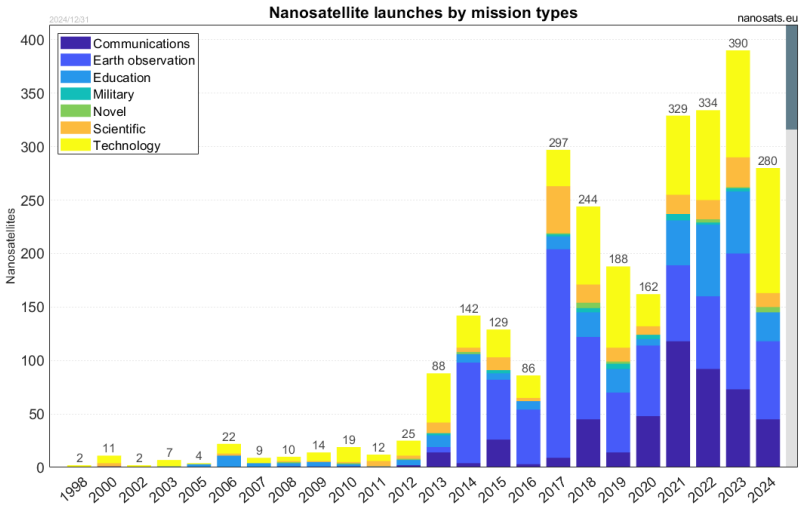


Figure 1.3: Nanosats launched per year and type of mission [27].

The operations in the vicinity of small bodies for these missions were achieved by a combination of radio navigation ground support and autonomous optical navigation involving the use of navigation cameras or NAVCAMs [29], [30]. En route and future missions will also make use of autonomous relative navigation to achieve its science goals. ESA Hera mission launched in 2024, will rendezvous and orbit Didymos binary system, where it will deploy two cubesats, Milani and Juventas, that will orbit and soft land in Didymos while performing accurate navigation and joint observations along the three spacecrafts [31]. RAMSES is a planned mission to Apophis that will explore the asteroid before its close fly-by to Earth, carrying also two cubesats and payload inherited from Hera mission [32]. Also this decade, M-ARGO will be the first nanosatellite ever to standalone rendezvous with an asteroid and perform close proximity operations over an extended period for identification of in-situ resources [33]. Figures 1.3 and 1.4 show a clear growth in the employment of cubesats in science and deep space missions, making clear the need to adapt traditional operations techniques to more lightweight and low-resources platforms. These compact satellites capitalize on miniaturization trends and Commercial-Off-The-Shelf components, fostering cost-effective and streamlined spacecraft development [34], [35]. Equipped with scientific payloads, cubesats are capable of competing with traditional large-scale science satellites for a fraction of the cost. However, the reduced hardware complexity trades-off with a need of better onboard algorithms, enabling autonomous navigation and operations that do not require ground-based interactions.

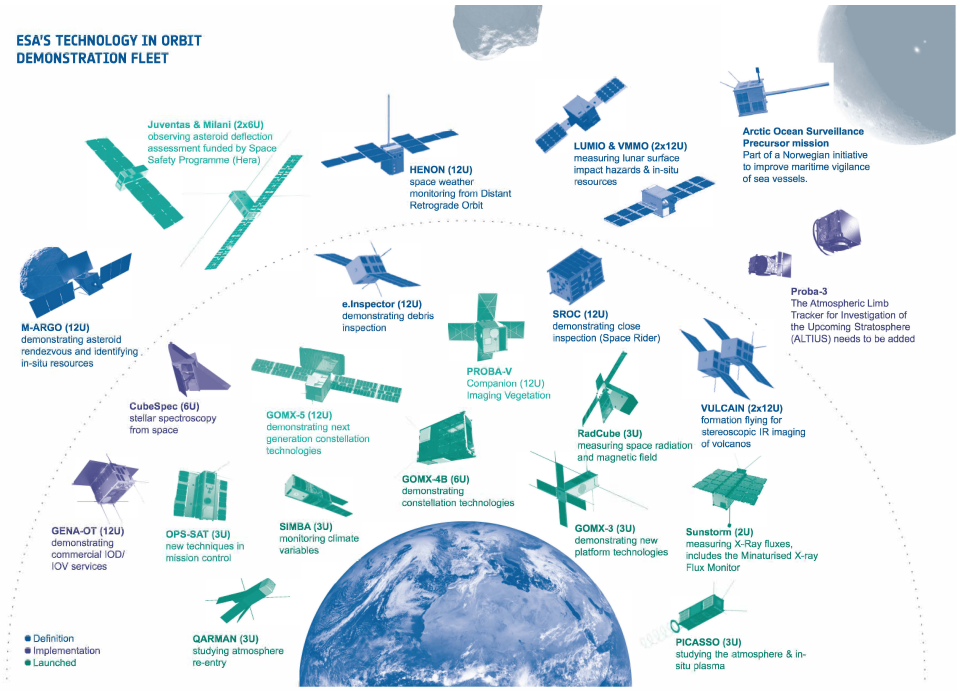


Figure 1.4: Nanosats Science and Technology Demonstration ESA Missions [28].

1.2. Autonomous Navigation

Autonomous navigation is essential in space exploration due to the significant communication delays between Earth and distant spacecraft, which make real-time control impractical. By enabling onboard navigation, autonomous systems allow missions to operate efficiently and adapt to unforeseen conditions without on-ground commanding. Autonomous navigation took a long way before being feasible, mainly due to the computational resources required. The navigation algorithm has to perform the resource-consuming task of resolving the spacecraft state based on a given sensor input, typically cameras or other optical payloads. The first attempts at applying optical navigation in deep space were carried out in 1969 for Mariner 6 and 7 missions at Mars. The pictures from the television cameras onboard the Mariner probes were downlinked to Earth to be printed, then an operator manually overlaid a transparent plastic on the printed image to match a circle corresponding to the limb of Mars to finally read off the coordinates [36]. These coordinates were then merged by a navigation filter reading the previous orbit determination solution derived from radiometric data and stored in magnetic tapes. This manual yet effective approach, reduced by half the B-plane error (position in the plane normal to the incoming asymptote of the hyperbolic flyby trajectory) of the radio-only solution. The process was then automatized on ground with one of the firsts image centerfinding computer programs in preparation for the Mariner 9 Mars orbiter. During the orbital phase, the improvements in the optical navigation and the enhancements on the satellite ephemeris, allowed for the first time ever to perform close-up imagery of the Martian moons, Phobos and Deimos.

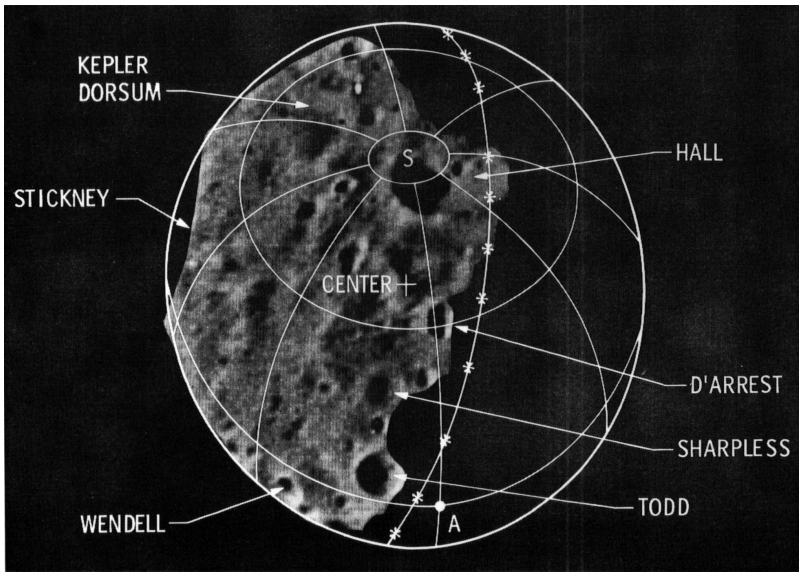


Figure 1.5: Mariner 9 center-of-figure image coordinates determination using optical navigation computer drawn overlay [37].

The lessons learned from the Mariner and Viking missions to Mars were crucial for the successful navigation of the Voyager missions during their tour of the outer planets. The vast distances in the outer Solar System meant that radiometric orbit determination alone could not meet the mission requirements. The radio-only solution for Voyager 2 encounter with Uranus would have resulted in B-plane uncertainties of several thousand kilometers. However, the integration of optical navigation data reduced this error to less than 100 kilometers [36]. Thanks to the efforts of the optical navigation team, who determined the orbits of many minor moons of the gas giants as part of the orbit determination loop, high-resolution images of these moons were captured for the first and only time. Years later, in 1995, it was the first time that at least part of the optical navigation was done onboard. When the high-gain antenna of Galileo spacecraft failed to deploy, the considerably smaller data rate provided by low-gain antennas meant that the planned optical navigation data could not be downlinked to Earth from Jupiter. The navigation team workaround it by combining image compression techniques with programming of the flight software for the onboard computer to do real-time processing, using ancillary data uplinked to the spacecraft, at the same time the picture was being read-off the charge-coupled device detector (CCD).

Substantial improvements towards autonomous optical navigation were achieved in consecutive years thanks to various missions to minor bodies. The main challenge of navigating around bodies with very small size and mass is that the ephemeris and physical properties of the target are typically not known with enough accuracy for using radio-only orbit determination [22]. The small size of asteroids and comets means that non-conservative and perturbing forces acting on the spacecraft, including solar radiation pressure (SRP) and spacecraft thermal re-radiation, could contribute significantly to the dynamics. Depending on the orbit, these perturbations could be of equivalent order of

magnitude as the target gravitational acceleration [38]. Therefore, navigation autonomy and robustness are paramount for these missions. Due to the lack of accuracy derived from ground observations of these objects, the only solution then is to rely on in-situ measurements for determining the relative position of the spacecraft with respect to the target. For NEAR Shoemaker in 2001, the centering techniques used since Mariner could not be applied due to the irregular shape of asteroid Eros and its high surface brightness, moreover, during low altitude phases, the limb of the asteroid would not be visible. This lead to the introduction of landmarks, still used today in modern optical navigation algorithms. A set of identifiable surface features or *landmarks*, constant in target body-fixed frame, were identified in the images, then an operator manually fitted an ellipse over the visible craters and the navigation algorithm used the position and size of these landmarks as reference points.

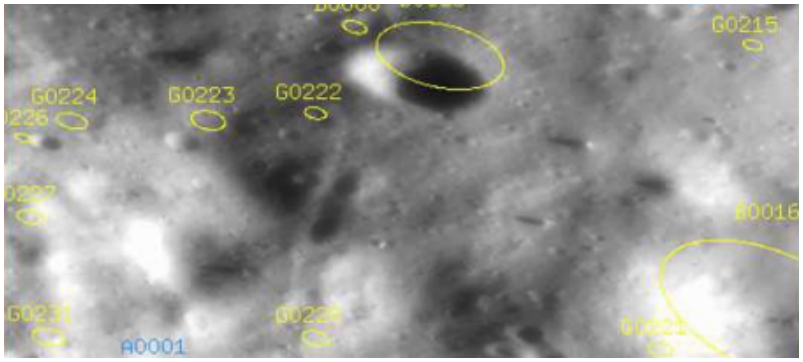


Figure 1.6: Landmarks used for optical navigation by NEAR Shoemaker mission [36].

Following missions with the goal of performing close encounters with comets and ultimately impacting on one, lead to the introduction of another major milestone, the deployment of onboard autonomous navigation. For Deep Space 1, the optical navigation took place onboard and it was fully autonomous, including an orbit determination filter and even being able to directly command the main engine. The fast encounter could not have been commanded from Earth due to the round-trip light time to the spacecraft introducing long delays compared to the duration of the fly-by. This technology has since then been used in most NASA missions to minor bodies [36].

On the hardware side, cameras have long been the primary devices for providing relative navigation input. Although detector technologies have advanced over the years, from the television cameras onboard Mariner spacecraft to the CCDs now standard in navigation cameras and optical payloads on deep space missions, the fundamental algorithms for determining spacecraft position using monocular sensors have remained largely consistent. In recent years other devices have been used to complement optical sensors, including thermal cameras [39], Light Detection and Ranging (LiDAR), and more recently, Flash LiDAR [40], which provides 3-dimensional information instead of the single range 1-dimension information of standard LiDARs. While LiDARs can provide very accurate distance information, monocular vision cameras are able to estimate the relative position of the spacecraft with lower hardware complexity, mass, size, and power requirements.

On the downside, standalone monocular vision requires highly complex solutions for relative position estimation as monocular cameras are not able to directly resolve the distance to the target. Once this algorithmic complexity is overcome, monocular vision sensors are suitable for a full pose estimation solution for low-resources missions with highly restrictive operational requirements.

The state-of-the-art monocular pose determination methods for optical relative navigation in deep space missions rely on Stereophotoclinometry (SPC) algorithms [41]. SPC was first developed for Dawn mission, which orbited asteroid Vesta and later on dwarf planet Ceres, and replaced the manual intervention required for landmark processing used in NEAR Shoemaker. Since then it has been used operationally in the latest missions to minor bodies like the International Rosetta Mission by ESA, OSIRIS-REx by NASA, and Hayabusa 2 by JAXA. SPC works by combining stereo and photoclinometry techniques to form the backbone of the terrain-modeling and landmark-navigation software [42]. A synthetic image is produced using the shape model of the target, and it is cross-correlated with the ground-truth image to obtain the target shift between both, allowing the individual landmark matching. For the Rosetta mission, automatic landmark tracking was applied using the image database, the landmark coordinates and the shape model as input parameters, with the major disadvantage that the image had to be downlinked to Earth for manual visual inspection and final orbit determination [43]. However, the main drawback of SPC is its dependency on the illumination conditions, specially to the phase angle (angle between the illumination source vector and sub-spacecraft point position vector). At very low phase angles the Sun is right at the zenith of the sub-spacecraft point, resulting on no shadows and surface features being washed-out. At high phase angles, extended shadows, specially in irregularly shaped bodies, cover many surface features. Both situations result in less robust landmark identification and degradation of accuracy [44]. Some missions complemented the navigation cameras with additional payloads, as the Hayabusa and OSIRIS-REx spacecrafts which also counted with LiDAR for spacecraft-to-target range determination and accurate shape modelling [45], [42], or the Hayabusa 2, which utilized retroreflective artificial landmarks carried by the spacecraft and deployed to the surface of asteroid Ryugu, such that they could be tracked by the on-board autonomous navigation system at variable illumination conditions [46]. Nevertheless, improved optical navigation algorithms can avoid the increased hardware complexity, cost and weight budget associated to auxiliary navigation payloads, while retaining operational requirements.

1.3. Motivation

The increasing involvement of commercial companies and low-resources missions in deep space exploration highlights the need for more efficient and adaptable navigation techniques. Despite significant advancements in technology, many recent missions continue to rely on optical navigation algorithms that have remained largely unchanged for over

two decades. These traditional methods require of substantial computational power or auxiliary infrastructure, which might not be available for low-resources missions. Recent studies also propose the application of Simultaneous Localization and Mapping (SLAM) techniques using optical sensors to build a map of the environment while navigating with respect to it, enabling the exploration of previously uncharacterized minor bodies [47] or feature-based autonomous approach [48]. However, SLAM methods usually need to be complemented by other sensors as Accelerometers, Gyroscopes or Star Trackers, which increases the hardware complexity. In addition, the required image processing and optimization algorithms can be computationally expensive.

The efficiency and robustness of machine learning algorithms stands out over other methods, and could be the key enabler to autonomous navigation in low-resources deep space exploration missions. Consequently, it is crucial to understand why these have not been adopted yet in operational missions. Recent research has explored the use of machine learning for feature extraction in terrain-relative optical navigation for celestial bodies [49], [50]. However, these approaches still depend on classical feature-matching algorithms to estimate relative position, limiting their potential to fully replace legacy systems. Machine learning algorithms, by contrast, could be trained to directly learn the nonlinear transformation from the 2D input image space to the 6D pose vector space, bypassing the need for explicit feature detection and matching.

Despite this potential, a significant barrier to adopting machine learning for direct pose estimation is the inherent lack of human interpretability in neural networks. The apparent random adjustment of neurons, makes it challenging to trace decision-making processes, raising concerns about reliability and robustness in critical mission scenarios. As a result, extensive testing and validation are essential to identify potential failure cases and ensure system safety and accuracy across a wide range of operational conditions.

Furthermore, the scarcity of diverse and representative datasets for various operational scenarios only exacerbates this issue. The production of data tailored to these scenarios could bridge this gap, and enable the successful training and deployment of a Convolutional Neural Network (CNN) for fully autonomous navigation. With enough high-quality data, more advanced and complex CNN architectures could be developed and trained, potentially exceeding the performance of traditional methods while offering greater efficiency and reduced operational costs.

1.4. Convolutional Neural Networks

The utilization of Convolutional Neural Networks is spreading in many industries as the main computer vision solution due to their lightweight architecture, high precision, robustness, and efficient performance in changing scenarios. CNNs are a type of deep learning model specifically designed to process data with a grid-like topology such as images. The general architecture of a CNN consists of multiple layers, including: convo-

lutional layers that apply filters to detect local patterns; pooling layers to downsample data and reduce dimensionality; and fully connected layers to make the final predictions. The convolutional layers enable the network to learn spatial hierarchies of features, starting from simple edges and textures in top layers to more abstract and complex shapes, like clusters of craters or body limb, in deeper layers. This hierarchical learning, combined with weight sharing and perturbation invariance, makes CNNs exceptionally effective for image analysis tasks. CNNs have revolutionized computer vision by automating feature extraction, allowing models to learn complex patterns directly from raw data. This capability has proven invaluable in domains where manual feature engineering would be infeasible or suboptimal. For instance, CNNs excel in identifying subtle patterns and structures in vast datasets, making them well-suited for space applications where rapid, autonomous decision-making is essential.

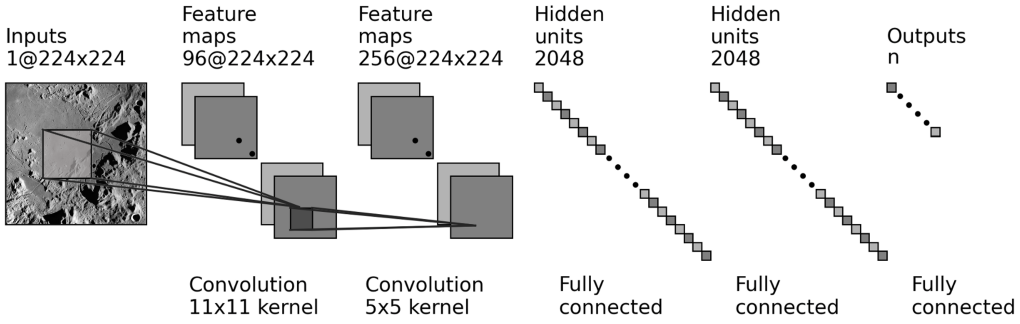


Figure 1.7: Example CNN architecture consisting of two convolutional layers and 3 fully connected layers.

As it is well known, training CNNs requires a large amount of data in order to successfully generalize the solution to be robust to multiple orbital regimes, spacecraft pointing profiles, instrument intrinsic characteristics and target variations. For optical navigation, data products from former missions could be used at some bodies in the Solar System, however in most cases the existing data is inherently limited to the geometry, specially, orbital ranges, image dimensions and illumination conditions, of the mission generating the data. Therefore, it is hardly possible to train CNNs based only on datasets from past missions, even applying data augmentation techniques to prevent overfitting, the trained network would not work for different scenarios. Nevertheless, in applications for which enough data is available and where the spacecraft pose does not impact the characteristics of such data, CNNs are thriving. Their strengths in pattern recognition and feature extraction have led to significant advancements in tasks like: terrain classification of high-resolution satellite images on Earth [51] and other planets [52]; despeckling of SAR images [53]; on-board image processing for coverage estimation and detection of clouds [54]; and others.

The training process of a CNN consists of feeding batches of images through the network over multiple iterations, also known as epochs, until going over all the training set (with full sets being of the order of tens or hundreds of thousands of images). Each batch

passes through the multiple layers, producing predictions that are compared to the ground truth labels by means of a loss function that quantifies the error at the current epoch. This error is then back-propagated through the network, adjusting the weights and biases of the neurons using an optimization algorithm like Stochastic Gradient Descent (SGD) or Adam [55]. The goal is to minimize the target loss function over a number of epochs, allowing the CNN to learn increasingly accurate representations of the data. For the optical navigation case, this means learning the non-linear transformation from the 2-D input image space (for a gray-scale image) to the 6-D pose vector space (3 position coordinates and 3 Euler angles). To further enhance generalization, techniques like dropout, batch normalization, and early stopping are employed, preventing the network from overfitting to the training set and improving its performance on unseen data.

Assuming enough data is available to cover most geometric combinations for a given mission to a known target, the CNN estimation of the spacecraft pose would be feasible, and it could be approached through two primary methods: discrete or continuous variable estimation. In the discrete variable estimation method, known as multiclass classification, the pose space is discretized into distinct bins, each representing a specific pose state. The CNN is trained to compute a probability distribution of an input image to be associated to each of the independent predefined states. In turn, the element of the output distribution with the highest probability represents the estimated pose state by the classification CNN [56], [57]. While this approach simplifies the problem to a classification task, the achievable accuracy is inherently limited by the resolution of the discretization, finer granularity increases precision but exponentially expands the number of classes, complicating training and inference. Alternatively, the continuous method involves regression, where the Neural Network directly outputs the coordinates of the pose vector as continuous values [58], [59]. This approach avoids the limitations of discretization and can provide highly accurate pose estimates. However, it may be more sensitive to noise and outliers, requiring careful design of the network architecture and loss functions to balance precision and robustness. Hybrid approaches that combine classification and regression leverage the strengths of both methods to enhance performance and reliability for variable mission profiles, such as orbiters and planetary landers. Regardless of the chosen estimation method, both rely heavily on the availability of diverse and representative datasets to ensure generalization and robustness, making the quality and scope of the input data a critical factor for successful training and deployment.

1.5. Data Available and Rendering

Numerous datasets derived from past space missions exist for various celestial bodies in the Solar System. For the inner planets, some dwarf planets, large moons, and a few asteroids and comets, topographic and spectral maps are typically available. While each dataset is inherently constrained by the geometry and instrumentation of the mission that acquired it, combining multiple datasets enables the creation of more generalized models.

With proper processing, different types of geospatial data can be integrated into a rendering engine to generate synthetic products. These synthetic products can help address the current lack of high-quality and generic datasets required for training neural networks for spacecraft pose estimation. However, in order to render or simulate the observations of optical remote sensing instruments of space missions, multiple data sources are necessary to construct the scene accurately.

To accurately reproduce target models, the shape and surface properties must be integrated. The process begins with an ellipsoidal mesh, which is refined by adjusting vertex positions based on topographic data. A detailed shape model is crucial for simulating self-cast shadows on the surface, especially for complex, irregularly shaped bodies like comet 67P/C-G or asteroid Itokawa. Even for larger celestial bodies, shadows play a significant role in dawn-dusk orbits or in the Moon polar regions, where shadows can extend for hundreds of kilometers. After shape modeling, albedo or surface reflectance data is extracted from imagery and projected onto the target surface. This texture projection is essential for accurately rendering low-phase-angle illumination, resulting in topographic details being washed out. For non-active bodies, the combination of topography and albedo is typically sufficient to generate realistic images. However, active bodies, especially those with atmospheres, require additional modeling to account for light-scattering effects, cloud cover, and other phenomena that significantly influence image characteristics and pixel intensity distributions.

A vast number of past missions have produced high-resolution datasets for planetary bodies, with particularly extensive coverage for the Moon and Mars. Even for less frequently visited targets, such as asteroids, comets, or dwarf planets, a single explorer mission often provides sufficient global coverage at high resolution. Many of these global products are publicly available through planetary archives, including NASA Planetary Data System (PDS), ESA Planetary Science Archive (PSA), and independent archives maintained by other space agencies. For the Earth, platforms like Google Earth Engine [60] facilitate access to diverse Earth observation datasets, allowing users to specify parameters such as area of interest, spatial resolution, temporal range, cloud coverage, and output format. Altogether, these data sources enable the creation of high accuracy digital twins of the most visited bodies in the Solar System.

1.6. Gaps and Research Questions

Multiple challenges have been identified with the motivations introduced in section 1.3, specially related to the application of machine learning for autonomous optical navigation and the lack of data suitable for training machine learning models, transferable to multiple orbital regimes and mission classes. Overcoming these gaps is essential to enable autonomous optical navigation solutions based on neural networks, the main high-level points to address are:

G.1 Limited Availability of Training Data: A significant obstacle is the lack of high-quality datasets from past and current missions. Moreover, existing software for synthetic data generation is often unsuitable for machine learning applications, being too slow to produce large sets, resource-intensive, and lacking the flexibility to simulate diverse scenarios. Addressing this gap requires the development of efficient, customizable rendering pipelines tailored for neural network training.

G.2 Sensitivity to Illumination Conditions: Current optical navigation methods struggle with adverse illumination conditions, which affect feature visibility and detection accuracy. Machine learning models must be adapted to generalize across a wide range of illumination scenarios, enabling to extend operational limits.

G.3 Lack of Generalized Navigation Solutions: The literature tends to focus on specific mission profiles, such as minor body encounters, planetary orbiters, or descent modules, without providing a unified approach adaptable to varying spatial scales and resolutions. Developing a generalizable navigation framework would enable broader applicability across diverse mission types.

G.4 Partial Adoption of Machine Learning: While some publications explore neural networks for feature extraction or crater detection, none extend this approach to the core pose estimation, which is still performed through standard feature matching. There is a need for end-to-end learning frameworks that directly predict spacecraft position and orientation without further computational burden.

G.5 Resources Constraints in Small Spacecraft: Current optical navigation methods are computationally demanding, making them unsuitable for resource-limited platforms like nanosatellites, typically lacking also ground support due to hardware limitations. A line of research is dedicated to developing lightweight neural network architectures, optimized for deployment on low-power, edge-computing devices.

1.7. Objectives

Building on the limitations and research gaps of state-of-the-art optical navigation methods discussed in Section 1.6, this thesis seeks to advance the research lines by addressing the challenges outlined in the previous section. The first research objective is to develop a complete framework for generating synthetic datasets tailored for training neural networks devoted to spacecraft optical navigation. This framework needs to be highly adaptable, capable of simulating diverse geometric configurations, including spacecraft pose, lighting conditions, instrument parameters, and target characteristics. Furthermore, to enable the creation of sufficiently large image datasets, the rendering pipeline must leverage optimized rendering techniques to ensure high-speed data generation. To build the previous framework, the following intermediate objectives need to be met:

O.1.1 High-fidelity models: The synthetic images shall resemble the physical and spectral characteristics of the real targets, as the trained CNNs must be validated and operationally work with real images. Consequently, multiple types of data such as, altimetry, surface reflectance, albedo, or composition, should be combined to achieve a digital twin of the target to be loaded in the rendering pipeline. This objective in combination with **O.1.2** is paramount to bridge the lack of quality data highlighted by **G.1**.

O.1.2 Tailorable rendering pipeline: An adaptable pipeline to produce large sets of images intended for training CNNs shall be developed, aiming to tackle **G.1**. Taking as input a target model, and a set of geometric and lighting constrains, the pipeline should produce labeled images containing all the associated parameters to be used for training. The flexibility in the scene variation will allow to train for any illumination condition and target, addressing **G.2**. GPU-accelerated and Physically Based Rendering engines must be implemented for fast and accurate rendering.

Leveraging the outcomes of the previous objectives and relying on the produced training sets, this research aims to explore novel CNN architectures suitable to be deployed onboard for fully autonomous optical navigation. The proposed CNN solution must be capable of adapting to changing scenarios and resilient to varying operational conditions, consistently providing accurate spacecraft pose estimates. The key steps for developing this CNN are outlined below:

O.2.1 Operational invariance: By carefully managing the training process and applying data augmentation techniques, this objective aims to enhance the model robustness to scene variations. In addition to those inherent to the training images, additional random variations introduced during runtime such as, pixel shifts, intensity histogram perturbations, and random erase, help tackle gaps **G.2** and especially **G.3**. These techniques enable the trained CNN to adapt to challenging conditions, including rapid altitude changes or atmospheric phenomena.

O.2.2 Full CNN approach: The proposed CNN model must provide the spacecraft pose directly, without the need for any post-processing, to address **G.4**. To achieve a complete pose estimation solution, a combination of various CNN types working sequentially will be designed, relying solely on efficient, lightweight models with fast inference times.

O.2.3 Light-weight architectures: State-of-the-art and customized architectures will be explored, focusing on efficiency-oriented models with reduced size and fast inference times, capable of running on small platforms. Combined with optimization techniques like quantization, this approach addresses **G.5**.

1.8. Contribution of this research

Provided the background, the current gaps highlighted in section 1.6, and the set of objectives outlined in section 1.7, this section summarizes the original contributions of this thesis. Each contribution covers multiple research objectives, and each of them is linked to the publications previously listed in chapter *Published and Submitted Content*. The three aforementioned publications (Paper I, Paper II and Paper III) are peer-reviewed articles published in international indexed scientific journals. The detailed contributions are described below:

C.1 ChuriNet: Development of a rendering pipeline implementing GPU-accelerated physically-based rendering, devoted to generating large sets of synthetic images. Aimed for asteroids and comets, the pipeline generated sets are suitable for the training and testing of the designed CNNs. A hybrid CNN sequential model combining multi-class classification and regression approaches is trained for estimating the spacecraft pose (**Paper I**).

C.2 EarthNet: Development of very high-resolution and fidelity models devoted to planetary missions simulations. Focusing on the Earth, the rendering pipeline is extended to use shape, albedo and atmospheric models to produce synthetic training sets, bridging the gap between synthetic and real images. Efficiency oriented CNN architectures are implemented and trained, tested on real hardware for cubesat missions (**Paper II**).

C.3 BennuNet: Design and validation of modified state-of-the-art CNN architectures, achieving operational performance and accuracy. Time Distributed CNNs are implemented to ingest sequences of images and take advantage of contextual information. The trained models are integrated with a standard navigation filter, and its readiness, validated with real images from OSIRIS-REx mission (**Paper III**).

The first contribution, **C.1**, focuses on the development of the rendering pipeline, addressing objective **O.1.2** and laying the foundation for the complete CNN solution outlined in **O.2.2**. The second contribution, **C.2**, extends the rendering pipeline to incorporate high-fidelity models that combine multiple data types, meeting objective **O.1.1**. It also adapts the pipeline from minor body scenarios to planetary orbiters and landers, enabling full operational variations as described in **O.2.1**. Finally, **C.3** consolidates the rendering pipeline with additional geometric and image variations, supporting objective **O.1.2**, improves the blocks of the hybrid CNN solution with enhanced architectures for each coordinate of the pose vector, **O.2.2**, and explores the implementation of lightweight architectures capable of executing the complete CNN solution on small platforms, targeting **O.2.3**.

With the purpose of summarizing the conceptual blocks of the thesis, the research gaps, objectives, contributions, and their interconnections, are illustrated in Figure 1.8.

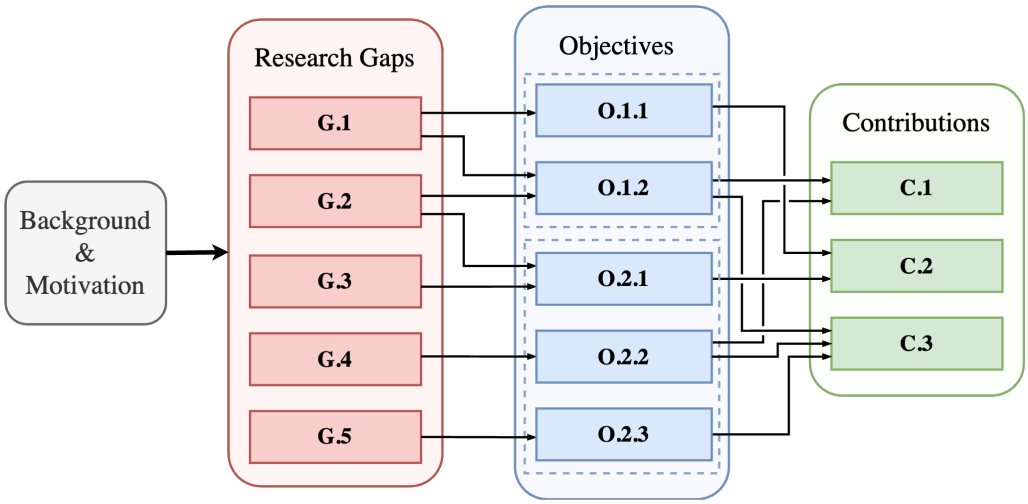


Figure 1.8: Diagram of the thesis research gaps, objectives, contributions and their connections.

1.9. Thesis Structure

This thesis is presented in the compendium of publications format, with each chapter containing a published research work that addresses the research gaps outlined in Section 1.6 and meets the objectives described in Section 1.7. The articles are included in their entirety and remain unchanged, except for visual adjustments to figures and tables to align with the formatting of this thesis.

The rest of this document is organized as follows: Chapter 2 presents **Paper I**, which details the development of a customizable rendering pipeline for generating synthetic training datasets, laying the groundwork for a complete CNN-based pose estimation solution. Chapter 3 features **Paper II**, which extends the rendering pipeline with high-fidelity models of celestial bodies, adapting it for planetary orbiters and achieving operational invariance. Chapter 4 contains **Paper III**, completing the CNN pose estimation with optimized lightweight architectures, validated on real hardware for small platforms. Finally, Chapter 5 summarizes the conclusions of this research, highlighting the capabilities, limitations, and open points for further work that could be explored.

2. ChuriNet – Applying Deep Learning for Minor Bodies Optical Navigation

The content of the current chapter coincides with the following journal publication:

A. Escalante, P. Ghiglino and M. Sanjurjo-Rivo, "Churinet - Applying Deep Learning for Minor Bodies Optical Navigation," in IEEE Transactions on Aerospace and Electronic Systems, Volume 59, Issue 4, Pages 3566-3578, August 2023, doi: doi.org/10.1109/TAES.2022.3227497 (**Paper I**).

2.1. Paper content and author contribution

This article presents the development of a GPU-accelerated rendering pipeline designed to generate large sets of synthetic images. These image sets comprehensively cover a wide range of geometric and illumination conditions, addressing the lack of high-quality data from existing missions identified in research gap **G.1** and contributing to objective **O.1.2**.

The synthetic training sets produced by the rendering pipeline are used to train neural networks for spacecraft pose estimation. To mitigate the challenges associated with a fully CNN-based solution, as outlined in objective **O.2.2**, the author developed a hybrid sequential model. This model combines a high-level classification CNN with a set of low-level regression CNNs for precise pose prediction.

The author formulated the problem, developed the rendering pipeline, and generated multiple training sets to assess the impact of geometric variations on CNN performance. Furthermore, the Ph.D. candidate implemented the training framework, trained and tested various CNN architectures, and evaluated their ability to estimate the components of the pose vector. The author also conducted the analysis and interpretation of the test results and prepared the manuscript for submission to the indexed journal *IEEE Transactions on Aerospace and Electronic Systems*.

2.2. Abstract

This article presents Churinet, a hybrid neural network-based method, devoted to on-board spacecraft relative position and attitude estimation in the vicinity of minor bodies like asteroids, comets or small moons, using monocular vision. In the context of navigating such minor bodies, traditional heuristic methods for spacecraft position and attitude determination encounter limitations in robustness and precision in the presence of adverse illumination conditions. Moreover, its performance is limited due to the computational cost resulting from the evaluation of a large number of possible pose hypotheses. In comparison, Churinet solves the relative pose estimation problem by directly learning the nonlinear transformation from a 2-D grayscale image to the 6-D pose vector space. Churinet is conformed by a set of sequential convolutional neural networks (CNNs) organised in two levels. The high-level multiclass-classification CNN is in charge of determining the sector of the discretized 3D space. Then, based on the sector estimation, the image is ingested by a low-level regression CNN, trained specifically for that sector, which estimates the pose of the camera. The secondary contribution of this research is the development of SPyRender, a tool for the generation of large sets of synthetic images, suitable for the training and testing of the designed CNNs. SPyRender implements GPU-accelerated physically-based rendering, enabling the efficient generation of photorealistic images. SPyRender has been used with the 3-D model of comet 67P/C-G for producing multiple image sets covering the whole range of camera position, attitude, and illumination conditions, allowing to study the impact of different geometries and image effects in the network performance.

2.3. Introduction

Small Solar System bodies have been the target of space missions since the ICE (International Cometary Explorer) crossed the plasma tail of Comet Giacobini-Zinner on September 11, 1985, and became the first spacecraft visitor of a comet [17]. The low temperature and low gravity environment existing on comets and asteroids preserve its high volatile content, making them the most accessible samples to the primitive material from which the Solar System formed [15], [16]. The need of taking in-situ measurements, has driven the exploration of various comets and asteroids in the last decades, like the Vega 1 and 2 spacecrafts which intercepted Comet Halley in March 1986 [18]; Stardust Sample Return mission which collected samples from the coma of Comet 81P/Wild 2 [19]; and Rosetta International Mission which was the first to rendezvous with a comet, 67P/Churyumov-Gerasimenko, and to land on it [20] after performing fly-bys to asteroids Steins and Lutetia [21]. The operations in the vicinity of small bodies for these missions was achieved by a combination of radio navigation ground support and autonomous optical navigation involving the use of navigation cameras or NAVCAMs [29], [30]. Future missions as the Martian Moons Explorer (MMX) [61] by JAXA, Comet Interceptor [62]

and Hera [31] by ESA, or DART [63] by NASA, will also make use of autonomous relative navigation. These missions will respectively visit the Martian Moons, Phobos and Deimos, investigate small worlds of a kind never characterized before, and test for the first time the redirection of an asteroid.

The main challenge of navigating such small bodies is that the ephemeris and physical properties of the target are typically not known with enough accuracy for orbit determination [22]. The large distances involved in the ground-observations devoted to improve the ephemeris of the target body reach accuracies of the order of hundreds of kilometers [29]. The only solution then is to rely on in-situ measurements for determining the relative position of the spacecraft with respect to the target. Moreover, low-gravity field and non-uniform target shape means that orbit and attitude estimation must be calculated with the spacecraft on-board computer.

The sensors used for on-board pose estimation and relative navigation in small bodies missions include monocular vision cameras, thermal cameras [39], Light Detection and Ranging (LiDAR), and more recently, Flash LiDAR [40], which provides 3-dimensional information instead of the single range 1-dimension information of standard LiDARs. Compared to LiDAR technologies, monocular vision cameras are able to estimate the relative position of the spacecraft with lower hardware complexity, mass, size, and power requirements. Nevertheless, monocular vision requires highly complex solutions for relative position estimation as monocular cameras are not able to directly resolve the distance to the target. Once this algorithmic complexity is overcome, monocular vision sensors are suitable for a full pose estimation solution for low-resources missions with highly restrictive operational requirements.

The current state-of-the-art monocular pose determination methods for spaceborne applications rely on classical image processing algorithms that identify pre-defined visible target features or landmarks [64]. The expected image is produced using a database of maplets (local models of the target) or the global shape model of the target. Then the ground-truth and synthetic images are cross-correlated for obtaining the target shift between both, allowing the individual landmark matching. For the Rosetta mission, automatic landmark tracking was applied using the image database, the landmark coordinates and the shape model as input parameters, with the caveat that the orbit determination team operators had to visually confirm them, meaning that the images had to be downlinked for position estimation [43]. Moreover, the robustness of this method is strongly dependant on the illumination conditions. At the expense of increased hardware complexity, other missions complemented the navigation cameras with other elements, as the Hayabusa spacecraft which also counted with LiDAR for relative position computation [45], or The Hayabusa 2, which utilized retroreflective artificial landmarks carried by the spacecraft and deployed to the surface of asteroid Ryugu, such that they could be tracked by the on-board autonomous navigation system [46]. Recent studies also propose the application of Simultaneous Localization and Mapping (SLAM) techniques using optical sensors to build a map of the environment while navigating with respect to it, enabling the ex-

ploration of previously uncharacterized minor bodies [47] or feature-based autonomous approach [48]. However, SLAM methods usually need to be complemented by other sensors as Accelerometers, Gyroscopes or Star Trackers, and the image processing and optimization algorithms might be computationally expensive.

Convolutional Neural Networks (CNNs) applied to computer vision are becoming the common ground solution for many industries, mainly because of their precision and robust performance in changing scenarios. Instead of relaying on landmarks matching as feature-based methods, deep learning algorithms are trained to learn the nonlinear transformation from the 2-D input image space to the 6-D pose vector space (3 position coordinates plus 3 Euler angles). The main drawback of implementing this direct nonlinear transformation, is that the CNN behaves like a black box, so they should be extensively tested and validated in order to identify possible failure cases, otherwise very difficult to detect. Two deep learning approaches can be utilized to estimate the pose vector. The first, called multiclass-classification, consists on solving a classification problem after discretizing and labelling the pose space [56], [57], meaning that the maximum achievable accuracy depends on the level of discretization. The second method applies regression such that the input images are directly mapped to the continuous 6-D output pose space [58], [59].

The main contribution of this paper is Churinet, a Convolutional Neural Network organised in two levels, high-level multiclass-classification and low-level regression, capable of estimating the relative pose of a camera with respect to a target body. This novel two-levels approach combines the advantages of both, low discretization classification for global position estimation, and regression for local accurate relative pose estimation. Data augmentation techniques applied during training have been investigated in order to generalise the CNNs estimating capabilities as much as possible, accounting for image shift, rotation, and distortions, while maintaining a reasonable accuracy in the pose estimation. Moreover, the layers and architecture configuration, applied regularization techniques, and training metrics, have been assessed focusing on optimizing accuracy. For the case presented in this work, namely, navigation for space missions to minor bodies, the amount of data for training is extremely limited. There exist some data sets of images from real missions such as Rosetta for comet 67P/Churyumov-Gerasimenko [65], Hayabusa 1 [66] and Hayabusa 2 [67] for asteroids Itokawa and Ryugu respectively, or the recent OSIRIS-REx of the asteroid Bennu [68], nevertheless those sets are restricted by the spacecraft orbit and illumination conditions, limiting the cases for which a Neural Network could be successfully trained.

The secondary contribution of this paper is SPyRender, a Python package implementing a GPU-accelerated renderer aimed at increasing the amount of available training data by systematically generating large sets of photo-realistic synthetic images suitable for training the designed CNN for a wide range of camera pose, illumination conditions, and camera extrinsic. The illumination source, camera model, and target shape can be configured and modified during runtime when rendering the scene, allowing for the efficient

production of different combinations of geometric conditions. Furthermore, Physically-Based Rendering (PBR) [69] materials can be utilized with SPyRender, enabling the use of multiple texture maps for achieving increased photo-realism. The fact of counting with synthetically generated sets covering any possible mission scenario, enables the proper training and testing of CNNs capable of providing a robust and efficient pose estimation solution.

The rest of this paper is organized as follows: Section 2 describes the methodology used for the synthetic image sets generation; Section 3 explains the CNNs architecture and training methods; Section 4 introduces the concept of Hybrid Neural Network Solution; Section 5 presents the results of the trained networks and its application to pose estimation; and Section 6 summarises the conclusions from the current study and the basis for further work and developments.

2.4. Data Generation Methods - SPyRender

The main goal of this work is training a CNN (or set of CNNs) capable of estimating the relative position and attitude of a camera with respect to a target body, in this case a comet, expressed in the target body-fixed frame. This pose estimation problem is depicted in Fig. 2.1, where $\vec{r}_{AB}$ represents the position vector of the camera focal point with respect to the comet centered body-fixed frame to be estimated by the CNNs. In addition, the rotation transformation in the form of Euler Angles corresponding to the transformation from comet frame ref_A to camera frame ref_B is estimated as well.

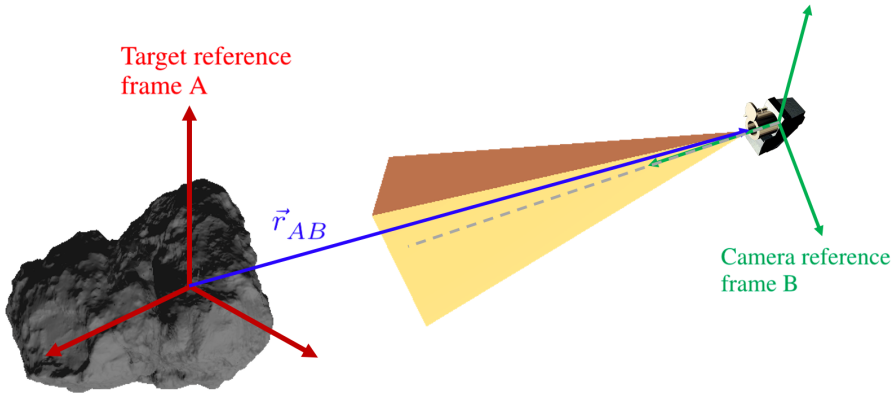


Figure 2.1: Depiction of camera pose estimation with respect to target

With the purpose of studying the impact of the different geometric configurations of the depicted scenario on the CNN accuracy, multiple synthetic image data sets have been generated. Therefore, the first part of this work describes the development of a pipeline capable of generating large sets of labelled synthetic images required for the training of CNNs. These sets account for different image effects via configuration like illumination

source position, type and intensity, camera and target position and rotation, camera field-of-view aperture angles, and image resolution.

SPyRender is a Python package developed within this work for the systematic generation of synthetic images focused on the analysis of space-borne instrument observations. The graphic capabilities of SPyRender are based on Pyrender [70], a pure Python library for physically-based rendering and visualization. This package implements a lightweight offscreen renderer with support for GPU-accelerated rendering, substantially alleviating the time consuming task of generating large sets of images, sometimes larger than 50,000 images. The geometric information determining the position and orientation of objects in the scene can be defined either of two ways. The first method allows the ingestion of a user-defined list of random poses within a cloud of points, most suitable for generating training sets including randomly shifted (offset of the target from nadir pointing) and rotated (around boresight axis) images, with random positions within a specified range of latitude, longitude and range, aiding generalization in pose estimation. In addition, the direction vector and intensity defining the Pyrender built-in Directional light are randomized within a user-specified range with the purpose of achieving illumination invariance. The second method relies on the computation of the state and attitude of the objects in the scene via SPICE Toolkit [71] taking advantage of the integration of SpiceyPy [72] functions with SPyRender. SPICE is comprised of a set of high level functions which provide the position and attitude of a given body defined in the SPICE Kernel Dataset (SKD) as a function of time. The data included in the Rosetta SKD comes directly from the orbit determination group and the telemetry of the spacecraft, such that it has been applied to reproduce the real geometry of the Rosetta mission. This second method, devoted to the generation of validation sets, has been used for simulating actual OSIRIS NAC (Narrow Angle Camera) images of comet 67P/C-G.

The shape model of the target is loaded in the scene via Trimesh [73], an open-source Python package for loading and manipulating triangular meshes in multiple file formats and fully compatible with Pyrender, permitting the photo-realistic rendering of textured 3D models. For the use case of comet 67P/C-G, multiple models are available at the Planetary Science Archive (PSA) [74] being classified in terms of producer, generation technique and number of polygons. The selected digital shape model implemented in the scene, contains the SHAP5 version of shape models for the Rosetta target 67P/C-G [75], produced at the Planetary Science Institute (PSI) and the Laboratoire d'Astrophysique de Marseille (LAM) using OSIRIS data obtained at the comet between July 11, 2014 and Feb 16, 2016 and applying the stereophotoclinometry (SPC) technique [41], a method used to generate ultra-high resolution digital elevation model (DEM) from images. High-detail shape models of future missions targets, specially in the case of comets, may not be available ahead of the spacecraft arrival and in-situ measurements have taken place. However, there are some cases of shape models based on punctual observations of past and current missions, like for Phobos based on Mars Express HRSC observations [76], for Comet Halley based on Vega-1/2 and Giotto observations [77], or in the near future for

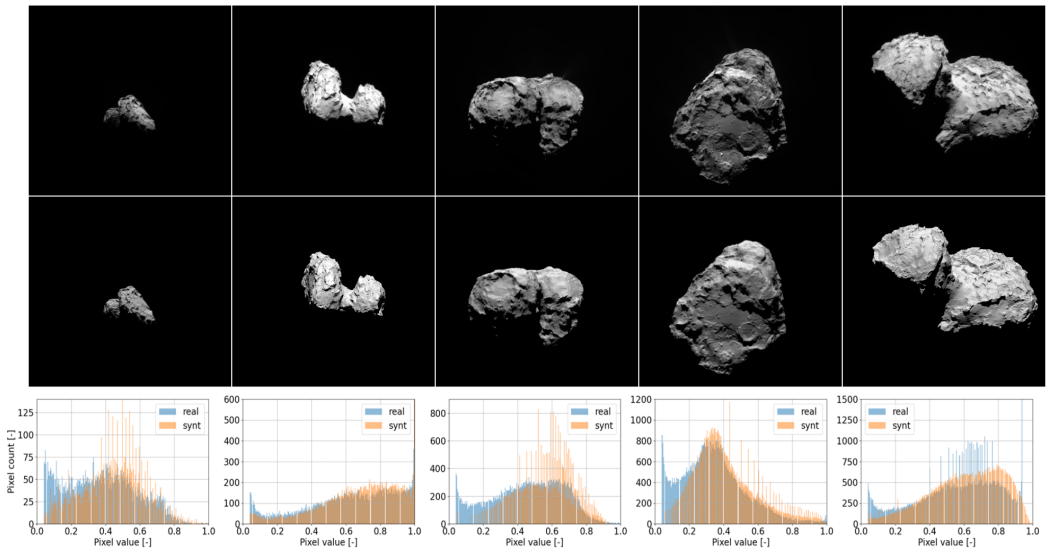


Figure 2.2: Comparison of real OSIRIS NAC images of 67P/C-G (first row) vs synthetic ones (second row). The grayscale intensity histogram for both images is displayed below

Didymos binary asteroid by LICIA [78] which will be later on visited by Hera spacecraft. It is also possible that some features of the surface of these minor bodies may be affected by shape changing effects. These changes may be considered by generating train and test sets introducing small scale variations in the target shape model so to encourage the network to give priority to large scale features which are less prone to change and analysing the impact on the network accuracy.

Besides camera position and orientation, other camera parameters should be tuned in the scene for replicating the OSIRIS NAC images. The aperture of the field of view (FoV) of the camera is defined in terms of reference and cross angles for a rectangular shape, in this case both being equal to 2.208 degrees, providing the same squared FoV as OSIRIS NAC Geometric Distortion Corrected FoV defined in the Rosetta OSIRIS Instrument Kernel [79]. In addition, the resolution in terms of pixel lines and pixels samples can be adjusted for the .png images output by the offscreen renderer. The image resolution has been directly adjusted to the 224x224 input image size expected by the CNN in order to avoid downsizing the images at the pre-processing stage during training. In Fig. 2.2, some examples of real OSIRIS images (first row) compared to the corresponding synthetic images generated with SPyRender (second row) are displayed. The third row depicts the grayscale intensity histogram for each pair of real and synthetic images, showing the similarity between both.

For this work, two main types of image data sets have been produced. The first group of sets contains images generated from the whole range of camera positions around the target body and is mainly devoted to analyse the training of CNNs capable of global position estimation. The second group consists of regional sets of images containing images of one single sector of the target body. For the latter group, the space around the target body has been divided in sectors of 45 degrees longitude and latitude yielding a

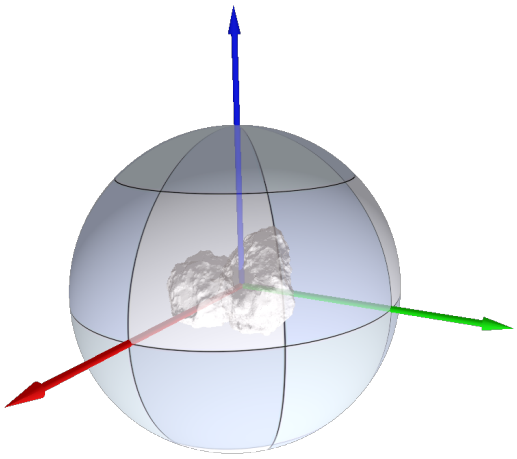


Figure 2.3: Target space division in 32 sectors, 45 deg longitude/lattiude each

total of 32 sectors. For each of these sectors, regional train and test data sets have been produced for analysing the training of CNNs which could estimate position and attitude of the camera with higher accuracy. The discretization in sectors of the space around the target body is depicted in Fig. 2.3.

Data augmentation techniques are applied to the data sets, either during production or directly during the training process, in order to extend the features of the images used for training, seeking rotational invariance, translational invariance and noise invariance. The effect of applying these data augmentation techniques on a single image can be appreciated in Fig. 2.4. In addition, a sequence of synthetic OSIRIS NAC images based on actual geometry of Rosetta observations has been produced to validate the trained models and compare the performance when ingesting real images instead of synthetic ones. While the different camera orientations can be extended during training by rotating and applying a shift to the images (as deviating from nadir pointing), illumination conditions have to be generalised during the generation of the data sets. Illumination conditions play a key role in computer vision and optical navigation, so it should be properly configured to achieve illumination invariance in terms of intensity and direction. Note that keeping the camera position and pointing unchanged, having two different orientations defining the directional light of the scene may result in two completely different images, specially for complex shape bodies as 67P/C-G, as it can be observed in Fig. 2.5.

The produced image data sets intended for global pose estimation are listed in Table 2.1, including the main features of the data sets. Each data set is composed by 50000 images, and its corresponding label files, for which 80% correspond to train set and 20% to validation set. The first and most simple set "simTrain224CG_sr" is composed of 224x224 pixels images covering the whole range of camera positions around the target, with attitude fixed to Nadir pointing, meaning the target is centered in the images and there is no image rotation around the boresight direction. In addition, random illumination source direction has been introduced during image rendering. Note that shift and

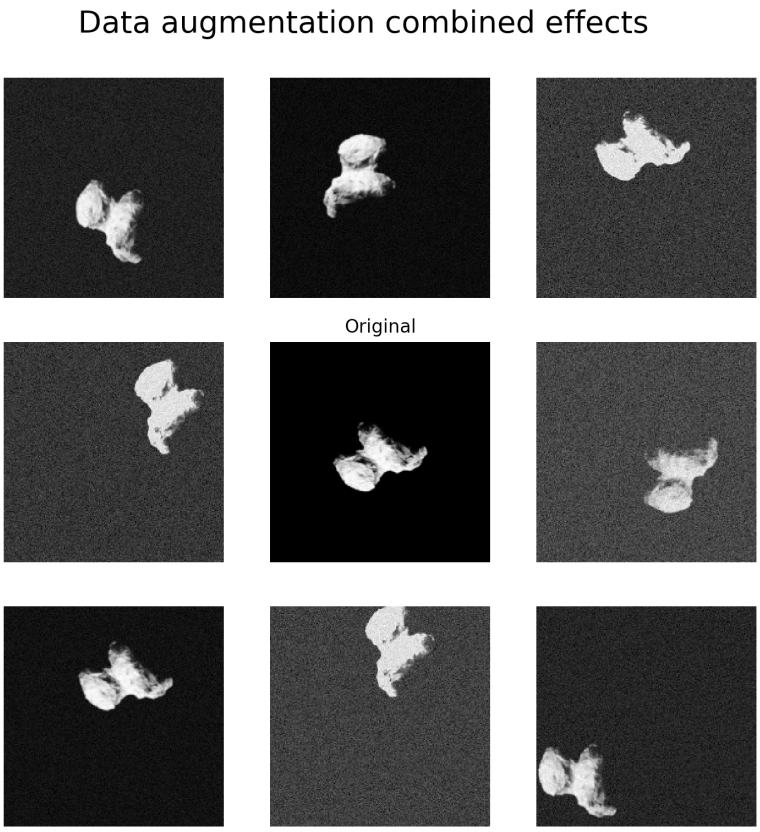


Figure 2.4: Example of combined data augmentation effects (random horizontal and vertical shift, random rotation, random brightness, and random Gaussian noise)

in-detector plane rotation could be artificially added during training via data augmentation techniques. For the rest of the image sets, the different scenarios of around boresight rotation, target shift, illumination intensity, and Gaussian noise, have been introduced producing multiple combinations of these effects so their impact in training and pose estimation performance could be analysed. Similarly, 32 regional image sets implementing the same image variations as the global image set simTrain224CG_sr_rr_br_o10_ng have been produced, but in this case, camera positions for each of them cover just one sector of the discretized space. As for the global sets, each regional set is composed by 50000 images for which 80% correspond to train set and 20% to validation set. The purpose of these regional sets is to improve the global pose estimation accuracy by following a local pose estimation approach in the vicinity of each of these regions.

2.5. Deep Convolutional Neural Network Architecture and Training Methods

The CNNs developed in this contribution are based on the structure of the AlexNet architecture [80], which has been proven to perform adequately in similar neural-network based applications for relative navigation like noncooperative spacecraft rendezvous [81] or asteroid centroiding for autonomous attitude navigation [82]. Other architectures as

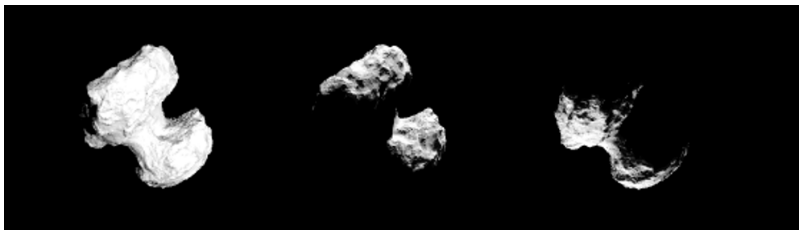


Figure 2.5: Example of different illumination directions keeping geometry of the scenario.

Table 2.1: Description of the global synthetic image datasets generated for this work

Dataset	Description	Images
simTrain224CG_sr	Centered, No boresight rotation, Fixed brightness	50000
simTrain224CG_sr_rr	Centered, Random boresight rotation [0, 360], Fixed brightness	50000
simTrain224CG_sr_br	Centered, No boresight rotation, Random brightness [-98%, 260%]	50000
simTrain224CG_sr_rr_br	Centered, Random boresight rotation [0, 360], Random brightness [-98%, 260%]	50000
simTrain224CG_sr_rr_br_o10	Random shift, Random boresight rotation [0, 360], Random brightness [-98%, 260%]	50000
simTrain224CG_sr_rr_br_o10_ng	Random shift, Random boresight rotation [0, 360], Random brightness [-98%, 260%], Gaussian noise	50000
simTrain224CG_sr_rr_br_o50_ng	Random shift, Random boresight rotation [0, 360], Random brightness [-98%, 260%], Gaussian noise	50000
simTrain224CG_osiris	Synthetic OSIRIS NAC images	50000

Inception [83] or ResNet [84] were considered at the early stages of the development, but AlexNet-based architecture was selected due to better estimation performance and the reduced operations required for a forward pass (inference time) [85]. Moreover, the reduced complexity and training times [86], compared to more recent models like DenseNet [87] or EfficientNet [88], allow a better assessment of the impact on performance when dealing with variable image effects specific for the produced image sets. The final architecture of the designed nets is shown in Fig. 2.6. The network is mainly composed by two convolutional layers followed by three fully-connected layers. The input shape of the convolutional layers is 224 by 224 matching the dimensions of the synthetic images. Max Pooling layers have been placed after the two convolutional layers in order to reduce sensitivity of the estimation to the position in the image of low-level features as points or lines. The pooling operation aims for invariance to slight displacements in the feature maps as most of the pooled outputs do not change for small displacements of the input features [89]. The kernel size and the strides of each convolution have been fine tuned as part of an iterative process so to optimize estimation accuracy. Before each fully-connected layer, a flatten layer has been placed to reshape the output tensor dimensions matching the expected input for the fully-connected layers. In addition, the dropout technique [90] has been applied during training of the fully-connected layers. This regularization technique selects random neurons for which its contribution is ignored by downstream neurons during the forward pass, so no weights update is applied to the selected neurons during back-propagation. A dropout rate (fraction of input units dropped) of 0.25 has been selected. For every layer, except for the last fully-connected, the ReLU (Rectified Linear) [91] activation function has been used. The function is shown in (3.1). This piecewise linear activation function returns the input value if it is positive, otherwise, it will output zero. This nearly-linear behavior retains many of the properties that makes linear

Table 2.2: Description of the High Level Multi-class Classification CNNs trained for this work

HL Network	Random Illumination Direction [-]	Random Boresight Rotation [deg]	Random Offset [pixels]	Random Brightness [%]	ZMGN std [-]	Training Dataset
sr_model	[(-1.0, -1.0, -1.0), (1.0, 1.0, 1.0)]	0	0	0	0	simTrain224CG_sr
sr_rr_model	[(-1.0, -1.0, -1.0), (1.0, 1.0, 1.0)]	[0, 360]	0	0	0	simTrain224CG_sr
sr_rr_br_model	[(-1.0, -1.0, -1.0), (1.0, 1.0, 1.0)]	[0, 360]	0	[-98, 260]	0	simTrain224CG_sr_br
sr_rr_br_o10_model	[(-1.0, -1.0, -1.0), (1.0, 1.0, 1.0)]	[0, 360]	[-10, 10]	[-98, 260]	0	simTrain224CG_sr_br
sr_rr_br_o70_model	[(-1.0, -1.0, -1.0), (1.0, 1.0, 1.0)]	[0, 360]	[-70, 70]	[-98, 260]	0	simTrain224CG_sr_br
sr_rr_br_o10_ng_model	[(-1.0, -1.0, -1.0), (1.0, 1.0, 1.0)]	[0, 360]	[-10, 10]	[-98, 260]	0.1	simTrain224CG_sr_br
sr_rr_br_o50_ng_model	[(-1.0, -1.0, -1.0), (1.0, 1.0, 1.0)]	[0, 360]	[-50, 50]	[-98, 260]	0.1	simTrain224CG_sr_br
sr_rr_br_o70_ng_model	[(-1.0, -1.0, -1.0), (1.0, 1.0, 1.0)]	[0, 360]	[-70, 70]	[-98, 260]	0.1	simTrain224CG_sr_br

models easy to optimize [89]. Moreover, with the output for negative inputs being always zero, ReLU remains a nonlinear function, more suitable for learning complex nonlinear relations existing in the data sets. ReLU has been chosen over other activation functions, as the sigmoid or hyperbolic tangent, as it solves the vanishing gradient problem. The vanishing gradient refers to the dramatic decrease of the back-propagated error in deep neural networks given by the derivative of the activation function of each additional layer, preventing the proper update of the network parameters [89]. The derivative for the sigmoid and hyperbolic tangent functions is close to zero when the input is very positive or negative, leading to the vanishing gradient problem, however, the derivative for the ReLU function is constant and equal to 1 across its active region (positive values).

$$\begin{cases} y = x & x > 0 \\ y = 0 & x \leq 0 \end{cases} \tag{2.1}$$

As per the last fully-connected layer, its dimension and activation function depends on the CNN being analysed. In this work, two types of CNNs have been produced differing on the last layer. The first type are Multiclass-classification CNNs, for which the last fully-connected layer has dimension 32 (same dimension as the labels vector for the camera position space sectors, each sector covering 45 degrees longitude and latitude). The Softmax [92] activation function is applied to the last layer as it normalizes the last layer output vector into a probability distribution over sector labels, meaning the element in the function output with the highest probability represents the estimated sector. In (4.1), z_i refers to the i-th element of the vector provided as output by the last fully-connected layer, and L_i are the elements of the resulting probability distribution.

$$L_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.2}$$

The second type are Regression CNNs, for which the last fully-connected layer implements the linear activation function yielding directly as output the values to be estimated, having dimension 3 for estimating camera position vector in Cartesian coordinates, 2 for

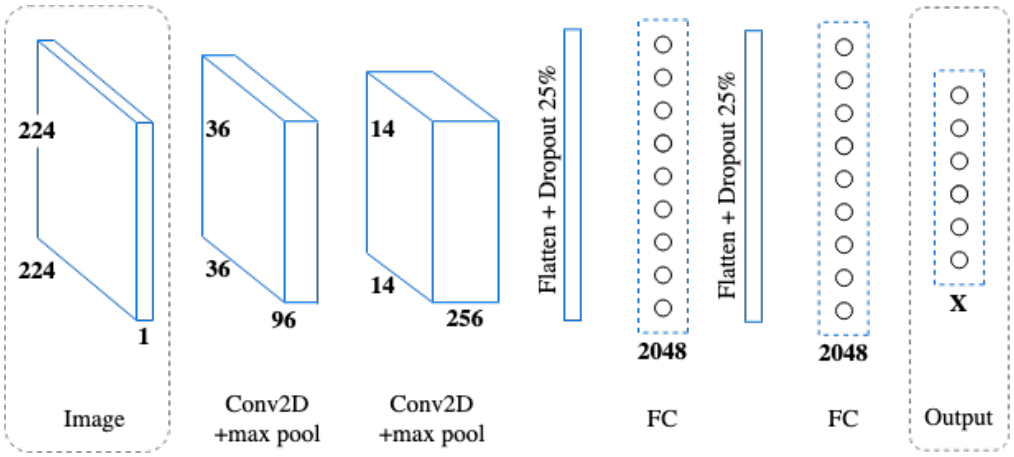


Figure 2.6: Deep Convolutional Neural Network Architecture used in the CNN blocks of Churinet

Yaw and Pitch angles, and 1 for Roll (around boresight direction) angle. The reasoning behind having three different Regression CNNs for estimating position vector, pitch and yaw angles, and roll angle is that the estimation accuracy is notably impacted by the difference in scale between the outputs. A spread range of values in the target variables causes weights to abruptly change, introducing instabilities in the training process [93]. Keep in mind that the roll angle ranges from 0 to 360 degrees while pitch and yaw will approximately range, in this case, from -2 to 2 degrees (camera field-of-view aperture). This difference of two orders of magnitude yields an estimation error for the Pitch and Yaw angles higher than the angles maximum value, while having an independent CNN for Pitch and Yaw substantially reduces the estimation absolute error.

In order to achieve the best possible performance for the designed architecture and to reduce the training times, some parameters which will configure the training process have to be defined. The most relevant are the Loss function, the optimization algorithm, and the training epochs. The error or loss function, is used to estimate the loss of the model at the current iteration of the optimization algorithm, such that the weights are updated accordingly to reduce the defined loss at the next iteration. The chosen loss function depends on the neural network to be trained and the predictive problem for which it will be applied. For Multiclass-classification CNNs, the Sparse Categorical Cross-Entropy loss function has been selected as it has been proven competitive in most domains and the preferred default option [94]. Categorical Cross-Entropy function computes the average difference between the predicted and actual probability distributions for all labels, which should be minimized. The term sparse [95] means that instead of one-hot encoding the target variable before training, the more suitable integer encoding (where each label or sector is assigned an integer value) is applied. The expression for the Categorical Cross-Entropy is shown in (4.2), where y_i represents the target value, and y'_i represents the i-th element in the model output.

$$L = -\sum y_i \log(y'_i) \tag{2.3}$$

For the Regression CNNs, the chosen loss function is the Mean Squared Error (MSE). Mean Absolute Error (MAE) loss was also tested providing similar performance in the net training, but MSE was preferred as the distribution of the target variables, either position or camera angles, can be considered as zero mean Gaussian distributions. Considering that outliers are not expected in the target variables, and as the MSE penalizes larger mistakes in the estimation, it is more likely that the weights are updated so to avoid producing outliers as output [96]. The equations for MAE and MSE are shown in (4.3) and (4.4) for n samples, where y_i is the ground truth and y'_i the predicted value.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y'_i| \tag{2.4}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y'_i)^2 \tag{2.5}$$

For the optimizer, Adam (Adaptive Moment) [55] has been selected as optimization algorithm in charge of minimizing the loss function. By using Adam optimizer, the learning rate is automatically adapted during the training process. The training epochs (number of passes through the whole train set) have been set to ensure that validation loss does not change if epochs are increased further.

2.6. Hybrid Neural Network Solution

The main drawback of using Multiclass-classification CNNs for position estimation is that the resolution of the estimated output directly depends on the number of sectors or labels in which the 3D space has been discretized. This implies that in order to achieve an accuracy of about 1 degree in longitude and latitude estimation, the 3D space should be divided in 64800 sectors. But the greater the dimension of the label vector is, the poorer the sector estimation performance. The rationale for this is that Classification problems take the different possible outputs as independent discrete values without accounting for ordering or continuous relations between them. In general, continuous variables such as the camera position or the Euler angles are better estimated by regression CNNs. Nevertheless, when trying to train a regression CNN for global position estimation, specially for a complex target as C-G/67P, it was not feasible to train a global model capable of accurately estimating the position in Cartesian coordinates. The optimizer was not able to successfully minimize the loss function. Moreover, the accuracy achieved by the global regression CNN was far below than the one achieved by a Classification CNN with a moderate sector discretization.

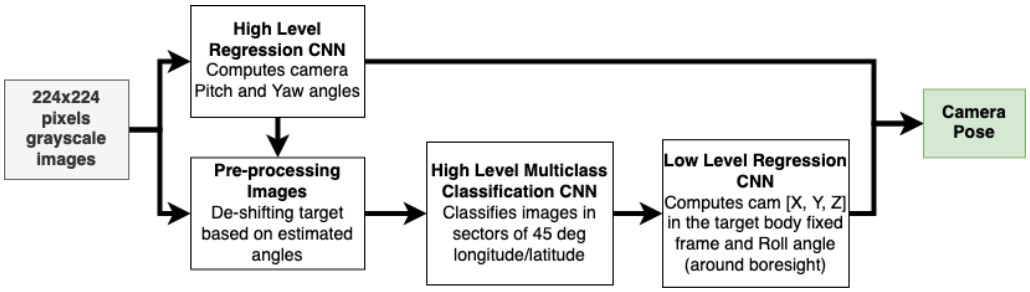


Figure 2.7: Churinet two levels Neural Network flowchart

This finding leads to the main contribution of this work, the hybrid two-level architecture, consisting of a High-level Multiclass-classification CNN in charge of estimating the local region or sector of the 3D space, and a set of Low-level Regression CNNs, each trained for one specific sector and capable of accurately computing camera position in Cartesian coordinates and camera angles. In principle, the estimation of camera position and attitude could be approached as two decoupled problems. The different scale in position vector coordinates, and pitch and yaw angles also supports the utilisation of two independent CNNs, one for position estimation and another for attitude. However, as it will be discussed in next section, the deviation from nadir pointing, i.e. target not centered in the image, has a substantial impact in the accuracy of the High-level Classification CNN. On the other hand, the estimation of pitch and yaw angles, associated to vertical and horizontal shift of the target from the center of the image, could be accurately estimated by a single regression CNN trained from a global image set. Considering the latter, an extra step has been introduced before the High-level Classification, consisting on a High-level Regression CNN estimating pitch and yaw in terms of horizontal and vertical shift expressed in pixels. This first result is used to pre-process the image set by de-shifting the target in the images, therein improving the accuracy of the High-level Classification CNN. Fig. 2.7 shows a flowchart depicting the Churinet working concept. First, the input image is de-shifted before ingestion into the High-level classification CNN, providing sector estimation. This output is used to select the Low-level regression CNN which will process the image in order to accurately estimate camera position and roll angle.

It is worth noting that the benefits of using independent CNNs could also be achieved by implementing a multi-branch model [97]. Even though this type of networks are more difficult to train, its implementation would alleviate the total size of the CNNs having branches with common upstream convolutional layers. But for the sake of this work, it was decided to implement independent sequential networks, seeking for modularity and allowing to investigate different image effects independently on each target variable.

2.7. Results

The experiments carried out for evaluating the training and performance of the three types of CNN composing Churinet are described in this section. For each type of CNN the

following aspects of the training process have been investigated:

- Optimal training epochs for achieving loss function convergence.
- Monitoring of training metrics, i.e. accuracy and loss function for the Multi-class Classification network and loss function for the regression networks.
- Train and validation estimation behavior and regularization.
- Impact of image effects and data augmentation on the network training and estimation performance.

2.7.1. High-Level Regression for De-shifting

The camera pitch and yaw angles are estimated by the High-Level CNN in terms of horizontal and vertical target displacement in pixels. This target shift could then be used to artificially centering the target in the image, which is then provided as input to the High-level classification CNN. Therefore, the parameter to be estimated by this CNN should be a 2-components vector containing vertical and horizontal shift in pixels. Because of this, the last fully-connected layer of this CNN has dimension 2. The data set simTrain224CG_sr_rr_br_o50_ng has been used for training. This data set is composed of 50000 images, 80% devoted for training and the other 20% for testing. A random target shift up to 50 pixels has been introduced in order to produce the un-centered images, enough for the comet to be at the border or to some extent outside of the field-of-view. Data augmentation techniques were used when producing this image set in order to improve the generalization of the estimation. These are, random image rotation, random illumination intensity, and Gaussian noise. Although illumination intensity and Gaussian noise are relevant for estimating the shift in the presence of image noise, ejected dust, or saturated images, the random illumination direction has the greater impact on shift estimation. As it was observed in Fig. 2.5, the difference between having the illumination direction parallel or perpendicular to the boresight direction, introduces long shadows and causes the illumination center to be notably displaced. The advantage of using a CNN for estimating the shift, is that it can be taught to distinguish the limb of the target, therein estimating the actual geometrical center of the target in the image instead of the illumination center. Note that the horizontal and vertical shift values for each image are recorded in the corresponding label file. The values of shift for every image are then stored in the label vector initialized at the beginning of the training process. This label vector represents the estimation target which the optimizer uses as true value to obtain the loss. For regression CNNs, Mean Squared Error (MSE) has been used as the loss function to be minimized. Nevertheless, for visualization purposes the Mean Absolute Error (MAE) has been plotted in Fig. 2.8 for easily monitoring the evolution of train and validation loss during training. Note that because the output of the CNN has dimension 2 (horizontal and vertical shift), the loss is computed as the average of the MAE for the two output

variables. It can be observed that while the train loss (dashed line) keeps decreasing with the number of epochs, the validation loss converges with just 10 epochs to a MAE of 4 pixels. For a 224x224 pixels image with a field-of-view of 2.2 degrees, this pixel error is equivalent to an approximate pitch/yaw angle error of 0.04 degrees.

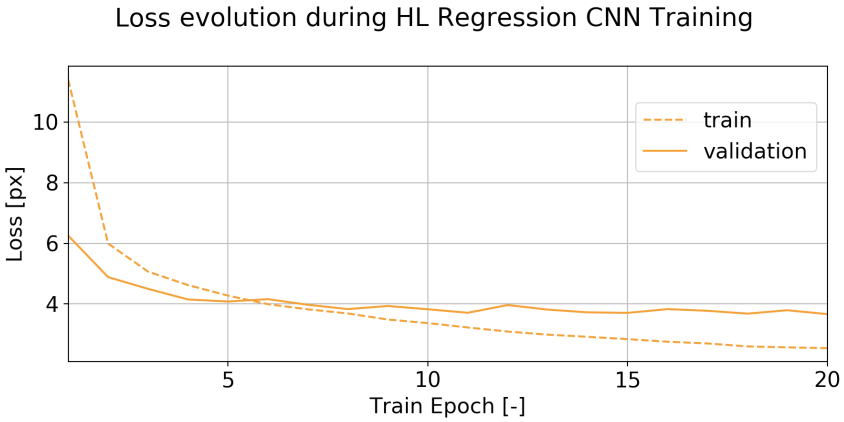


Figure 2.8: High-level (HL) Target Shift Regression Loss evolution during training

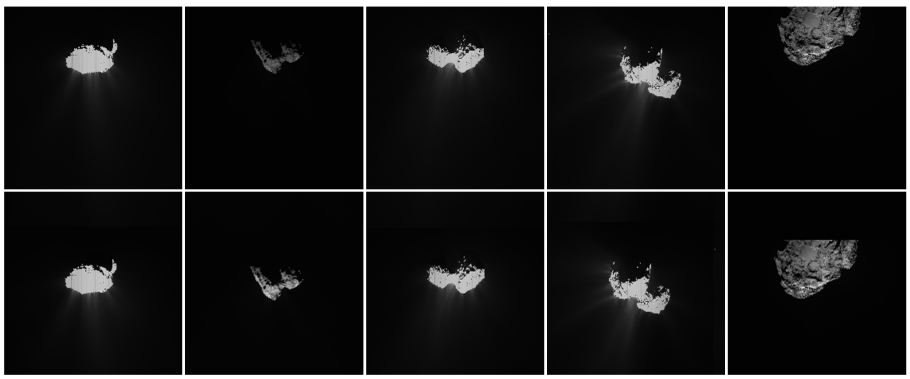


Figure 2.9: HL Target De-shifting applying estimated Roll and Pitch

The trained high-level regression CNN has been tested with real images from OSIRIC NAC captured between October 2015 and November 2015. For most of these images, the camera boresight was not aligned with the nadir direction, so the comet is not centered in the image. This sequence includes over-exposed images which increase the contrast for the target in the image, but also makes the ejected dust to be clearly visible. Fig. 2.9 shows some examples of real OSIRIC NAC images obtained from the Planetary Science Archive [74] for the specified time period, compared with the corresponding de-shifted images based on the CNN estimated shift. It can be appreciated that the network learnt during training to distinguish the comet body from the ejected dust thanks to the modeling of Gaussian noise. In this way, the net ignores the dust when estimating the shift of the illuminated pixels associated only to the target body.

2.7.2. High-Level Multi-class classification

The core component of Churinet is the high-level multiclass-classification CNN in charge of estimating the current sector of the 3D space. Differently from the high-level shift estimation CNN, image effects as target shift or around boresight rotation have a huge impact on the classification CNN accuracy. For the classification problem, the performance of the CNN is measured in terms of accuracy of predictions and the F1-score of classifications. The accuracy is defined as the percentage of correct sector estimations over the size of the image set. The F1-score is the harmonic mean of two quantities, the precision and recall. These are related to the number of true positives (TP), false positives (FP), and false negatives (FN) over the image set, providing a measure of performance even if the image set is affected by class imbalance. Because these metrics are devoted to binary classification problems, a positive sample is defined when it belongs to the correct sector, and negative for all the other sectors.

$$precision = \frac{TP}{TP + FP} \tag{2.6}$$

$$recall = \frac{TP}{TP + FN} \tag{2.7}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.8}$$

The different image sets have been used to train multiple classification CNNs in order to asses the impact of each image effect on the networks performance. The sector associated to every image is stored in the label files loaded before training to construct the labels vector. In this case, the last fully-connected layer of the CNN has dimension 32, same as labels or sectors in which the 3D space has been discretized. Furthermore, the estimated sector corresponds to the greatest element of the probability distribution output by the softmax function applied to the last fully-connected layer. The resulting precision, recall, and F1-score for each trained model are listed in Table 4.2. In Fig. 2.10, the train (dashed lines) and validation (solid lines) loss and accuracy evolution during training for the different image sets are shown. The first training case, denoted by "sr", is based on images generated only with random illumination direction, maintaining nadir pointing, fixed illumination intensity and without noise. For this reference case, the validation accuracy reaches 80% and there is some over-fitting to the training set, with train accuracy close to 100%. Note that train loss keeps decreasing but validation loss increases from epoch 5. For the next case, denoted by "rr", random rotation around the boresight direction (roll angle in the range 0-360 degrees) has been added to the input images. While over-fitting has completely disappeared, both validation and train accuracy slightly decreased to approximately 75%. The next case denoted by "br", introduces random illumination intensity, yielding from over-exposed images to low contrast images. As small scale surface features are not appreciable anymore for over or under-exposed images, the

Table 2.3: Precision, recall, and F1-score for the High Level Multi-class Classification CNNs trained for this work

HL Network	**Precision [-]**	**Recall [-]**	**F1-score [-]**
sr_model	0.82	0.81	0.81
sr_rr_model	0.81	0.74	0.76
sr_rr_br_model	0.66	0.58	0.59
sr_rr_br_o10_model	0.66	0.58	0.60
sr_rr_br_o10_ng_model	0.65	0.58	0.59
sr_rr_br_o50_ng_model	0.46	0.36	0.37
sr_rr_br_o70_ng_model	0.30	0.29	0.28

validation accuracy is substantially reduced and some over-fitting has appeared. From a practical point of view, it may be appropriate to remove over-exposed images for training if these are not expected in the real application, or at least, reduce the illumination intensity range to improve performance. The next result denoted by "o10" has been obtained adding random pitch and yaw angles equivalent to a maximum vertical and horizontal image shift of 10 pixels. It can be observed that the effect of introducing some off-nadir angle with respect to the previous case, barely reduces the validation accuracy although more epochs are required for it to converge. For the last three cases denoted by "ng_o10", "ng_o50", and "ng_o70", Gaussian noise has been added to the images and the horizontal and vertical shift has been kept at 10 pixels and increased to 50 and 70 pixels respectively. The Gaussian noise slightly reduced the validation accuracy by approximately a 5%, but as mentioned previously is fundamental to model different image perturbations. On the other hand, the increase of image shift to 50 and 70 pixels had the largest impact on the network performance, reducing validation accuracy to 35% and 25% respectively. This last result was the main driver on the implementation of the high-level de-shifting CNN for reducing the target shift, boosting accuracy on sector estimation by more than 20%.

It is important to notice that the network training is not always able to converge at the same epoch, moreover, the optimizer might not be able to get to a minimum for the loss function. The stochastic character of the algorithm implies that multiple training runs are sometimes required in order to successfully train the networks. While Fig. 2.10 shows network metrics for 50 training epochs, the final results for the ng_o10 case could be slightly improved to reach 62% accuracy by increasing the training epochs further and applying the l2-regularization technique. The l2-regularization consists on penalising very large weights, typically associated to over-fitting, in order to improve validation accuracy. The sector estimation error obtained when analysing a sequence of images corresponding to one orbit around the target is shown in Fig. 2.11. It can be appreciated that while the estimations are correct close to the center of each sector, the CNN fails to estimate the sector the closer the camera gets to a sector boundary. Depending on the illumination conditions and geometry of the target, it is sometimes difficult for the network to distinguish between the current and the neighbor sector at sector changes. To

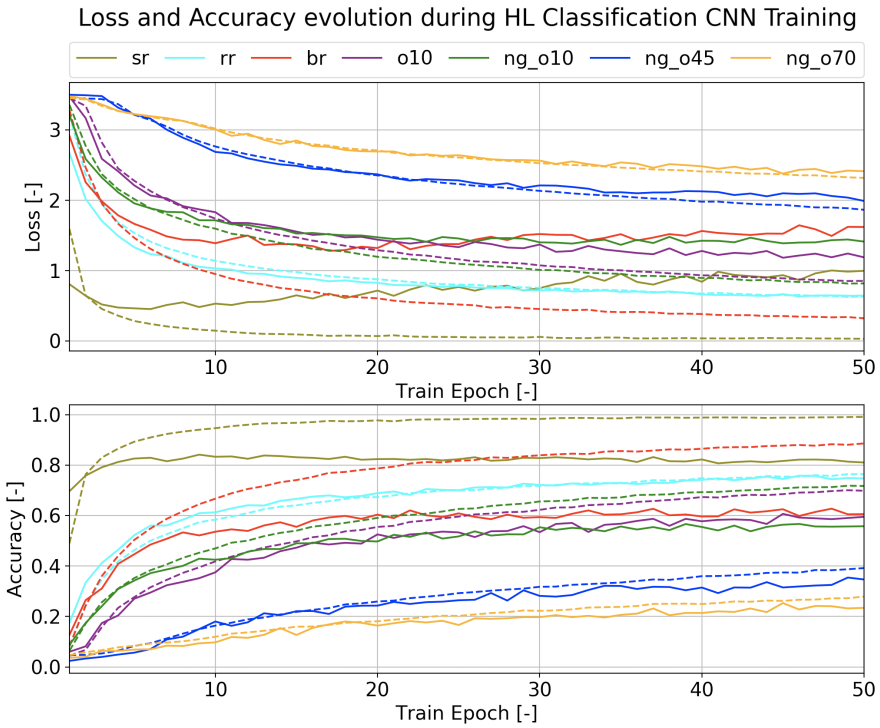


Figure 2.10: High-level (HL) Multi-class Classification Loss and Accuracy evolution during training

mitigate this, the sectors which have been used for training the low-level regression CNNs have been extended in all directions by a 20%, in such a way that they are overlapping. With this approach, if the high-level classification fails close to a boundary and estimates the neighbor sector, the low-level regression receiving the image is still able to estimate position as it was trained with images which also cover that region. Note that increasing further the sector extension margin would cause the same problems that global regression showed, i.e., it is harder for the optimizer to find minimums of the loss function and the network is less accurate.

In Fig. 2.12, the largest component of the probability distribution produced by the softmax function of the last fully-connected layer of the CNN is shown (blue line) for a subset of validation images. In addition, the error in sector estimation is plotted together (green line), when equal to 0, the sector is correct, when error is equal to 1 the estimation is wrong. It is worth noting that the larger is the maximum probability associated to the estimated sector, the most likely is that the estimation will be correct. This can be observed by discarding the estimations with probability lower than a defined probability threshold. For the high confidence solution obtained by increasing the probability threshold to a 75%, only 1/3 of the images fulfilled this condition but the mean accuracy for this case is increased to 83%.

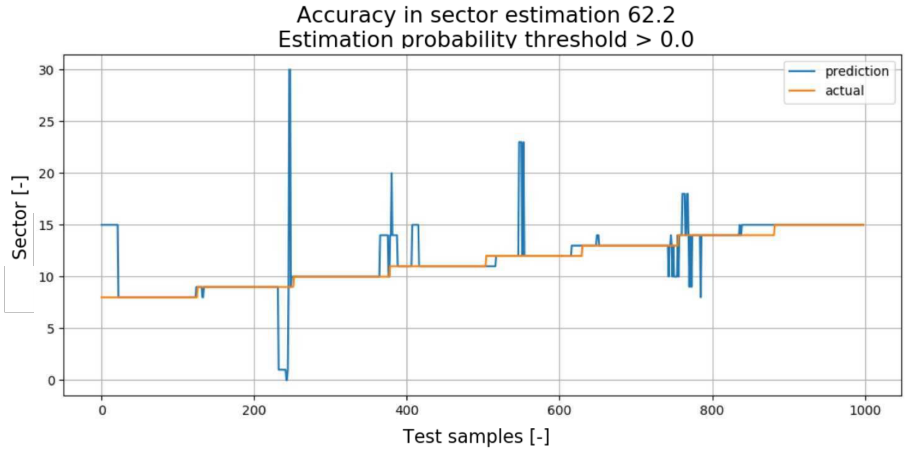


Figure 2.11: High-level (HL) Multiclass classification CNN sector estimation for one orbit

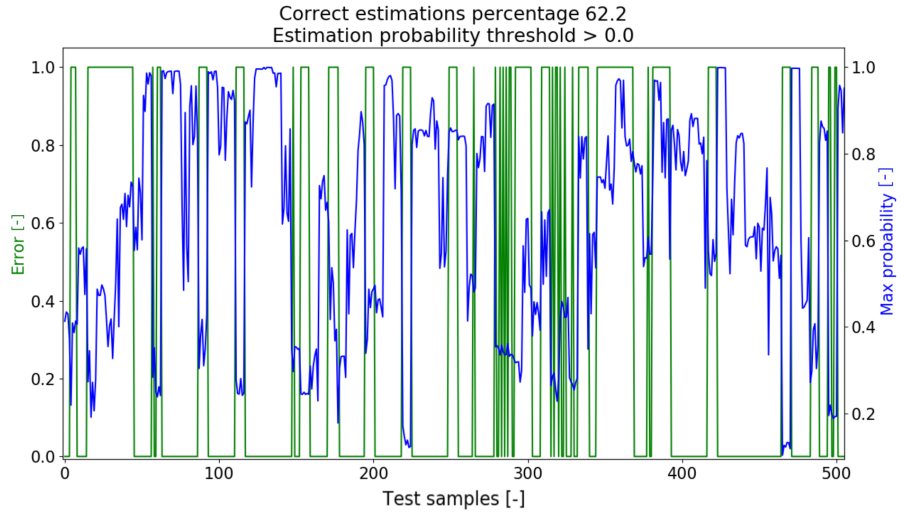


Figure 2.12: High-level (HL) Multiclass classification CNN largest component of output probability distribution (blue line) and binary classification error (green line) i.e. 0 for correct sector, 1 otherwise

2.7.3. Low-Level Regression

The last component of Churinet is a set of low-level regression CNNs, each trained for one specific sector of the 3D space and capable of estimating camera position vector in Cartesian coordinates and roll angle (around camera boresight rotation). Note that two independent networks are used, one for position estimation having last fully-connected layer of dimension 3, and other for roll angle estimation with dimension 1. In order to train these CNNs, one image set has been produced per sector. For all these sector specific image sets, the same image effects as those used for the image set simTrain224CG_sr_rr_br_o10_ng have been applied. Moreover, images with an offset of 50 pixels could be ingested by Churinet, but thanks to the high-level de-shifting CNN, the images processed by the low-level regression CNNs have a maximum shift of 10 pixels.

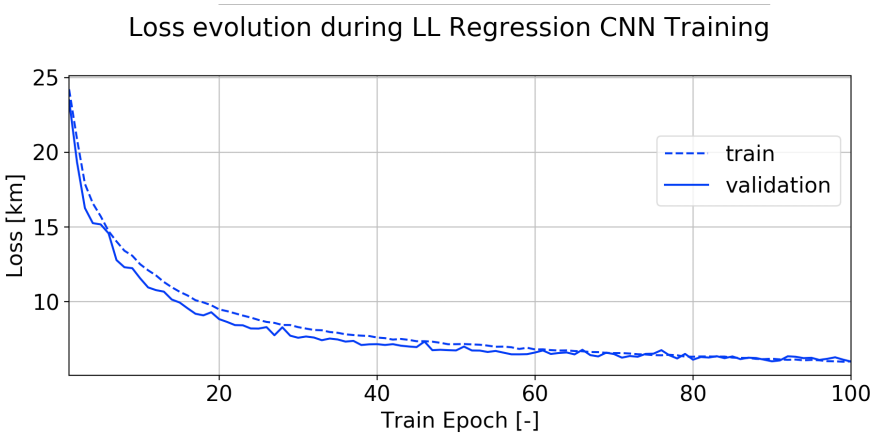


Figure 2.13: Low-level (LL) Position Regression loss evolution during training

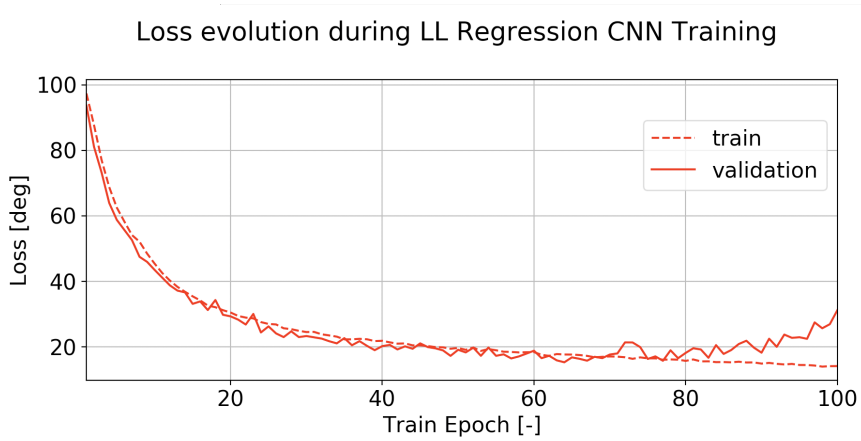


Figure 2.14: Low-level (LL) Roll angle Regression loss evolution during training

Therein, 10 pixels is the maximum random image shift applied to train the low-level regression CNNs. Camera position and roll angle are extracted from the label files of the image set to construct the label vector. As for the high-level regression CNN, the loss function used for training in this case is the MSE, but MAE is used in the plots for visualization. Furthermore, for the position regression, because the output of the CNN has dimension 3, the model loss is computed as the average of the MAE for each component of the position vector. The accuracy of the estimated relative position could be measured as well by the mean of the translation error E_T [98], between the true relative position $\vec{r}_i$ and the predicted relative position $\vec{r'}_i$ as defined in (2.9).

$$E_T = \frac{1}{n}\sum_{i=1}^{n} |\vec{r}_i - \vec{r'}_i| \tag{2.9}$$

Fig. 2.13 and Fig. 2.14 show the loss evolution during training for the position and roll angle CNNs respectively. These plots are specific to sector 8, covering the 0 to 45 degrees longitude and latitude range, but are representative for the training of other sectors. For the position loss, after 100 epochs a MAE of 6 kilometers is achieved. In addition, the

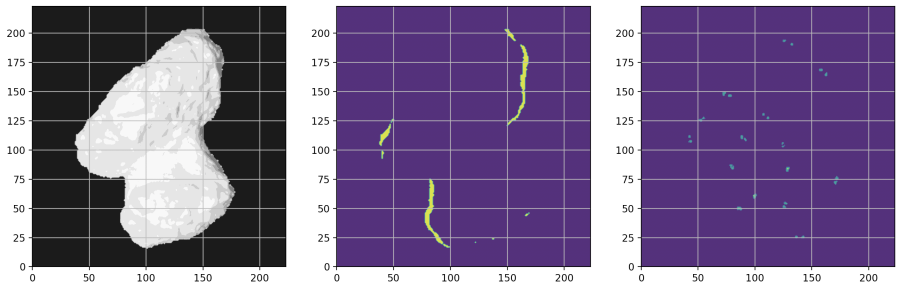


Figure 2.15: Actual vs predicted comet landmarks difference due to Churinet position error. Correct image (left), subtracted images (center), landmarks displacement (right)

mean translation error E_T reached 8.5 kilometers. For an altitude ranging between 70 and 150 kilometers, this means an average relative error of 8%. On the other hand for the roll angle, validation and train losses decrease and are close to each other up to epoch 60, moment at which they start diverging such that the network overfits the train set and validation loss starts increasing. In this case, the trained model obtained is not the one at the last epoch but the one providing the best validation loss value during training, equal to 16 degrees. As appreciated in Fig. 2.15, the average landmarks deviation due to the error in Churinet pose estimation is 4 pixels, equivalent to 0.04 degrees. This landmark deviation is of the same order as for featured based methods involving optical navigation team operator actions, used in the Rosetta mission, and which were successfully provided as input to the navigation algorithm for orbit determination [99].

2.8. Conclusion

In this work, a successful training and testing of CNNs applied to monocular vision navigation has been proven, setting up the basis for developing feasible deep learning navigation algorithms for orbiting minor bodies. SPyRender, an automatic pipeline for the generation of large photo-realistic synthetic image sets using GPU accelerated methods has been presented. This pipeline allows for the generation of image sets implementing user-defined geometric and illumination conditions, enabling the training and testing of CNNs. This configurability, enables the production of conditions which cover the whole range of situations in an hypothetical mission.

Furthermore, multiple CNNs have been trained and tested using the produced image sets, each of them implementing different image effects and restricted to specific geometric conditions. In this way, the impact on estimation accuracy of: target horizontal and vertical shift, target rotation, illumination conditions, and image noise, could be analysed. The selected CNN architecture has been proven to be adequate for the pose estimation problem, although wider layers were required for maintaining accuracy level when introducing more image effects. Similarly, more training epochs were required to compensate for the network loss convergence. Dropout and l2-regularization techniques were applied

for preventing overfitting, however it could be further improved by using larger training image sets. Both Regression and Multiclass-classification sequential CNNs have been trained and tested in various cases, investigating the advantages and limitations of each of them, resulting in the proposed two-levels global network, Churinet. The high-level consisting of a multiclass-classification CNN in charge of labeling camera position to one sector among a predefined set of the discretized global space. Based on the estimated sector, it chooses the low-level regression CNN which solves a local position estimation problem and estimates the camera position and Euler angles.

There are still some aspects which require further development. Due to the expected evolution of objective targets, shape model variations should be accounted for when training, introducing random small-scale model variations when generating the image set. Furthermore, the global estimation accuracy achieved by Churinet should be assessed not only for few real images but for the complete set of OSIRIS NAC images, evaluating the performance during a real mission timeline. Finally, the network model could be optimized by testing other deeper architectures or state-of-the-art computationally efficient-oriented architectures. Similarly, optimization techniques as quantization could be applied to reduce the total size of the model, enabling its use for on-board orbit and attitude determination in low-resources systems.

2.9. Acknowledgment

The authors acknowledge the Principal Investigator(s) H. Sierks (MPS, Goettingen, Germany) of the OSIRIS instrument onboard the Rosetta mission for providing datasets in the archive. Datasets of the OSIRIS instrument have been downloaded from the ESA Planetary Science Archive (http://archives.esac.esa.int/psa).

3. EarthNet - Applying Machine Learning Techniques for Optical Relative Navigation in Planetary Missions

The content of the current chapter coincides with the following journal publication:

A. Escalante, P. Ghiglino and M. Sanjurjo-Rivo, "Applying Machine Learning Techniques for Optical Relative Navigation in Planetary Missions," in IEEE Transactions on Geoscience and Remote Sensing, Volume 62, Pages 1-11, 2024, Art no. 4702811, doi: doi.org/10.1109/TGRS.2024.3374454 (**Paper II**).

3.1. Paper content and author contribution

This article presents the implementation of high fidely models of celestial bodies in the rendering pipeline to achieve objective **O.1.1**. Suitable for minor bodies such as asteroids and comets as well as for large moons and planets, the combination of topography, albedo and atmospheric data, bridges the gap between synthetic and scarce real images as identified by research gap **G.1**.

By using target models at different scales and resolutions, the synthetic images training sets are enhanced with respect to contribution **C.1**, better representing variable operational regimes. Directly tackling the lack of generalized navigation solutions indicated in gap **G.3**, these improvements provide the trained neural networks with the operational invariance outlined in objective **O.2.1**. The previous is also supported by the addition of new data augmentation techniques, that help make the trained CNNs more robust to optical effects and atmospheric perturbations.

The author crafted the digital twin of the target celestial bodies by combining multiple topography, albedo and atmospheric models at different spatial and time scales. Gathering and adapting data from different missions and sources, the Ph.D. candidate produced a very high resolution tiled model of the Earth achieving high fidelity synthetic images. In addition, the author implemented state-of-the-art efficiency oriented and more complex CNN architectures, moving away from the baseline CNN tested in **C.1**, evaluating the improvements in accuracy, robustness and inference times. These developments are included in the manuscript elaborated by the Ph.D. candidate, submitted to the indexed journal *IEEE Transactions on Geoscience and Remote Sensing*.

3.2. Abstract

Artificial Intelligence (AI) algorithms are playing an increasingly crucial role in onboard data processing to improve spacecraft autonomy and enhance the quality of scientific observations. While AI has found prominent use in Earth Observation missions, it is also gaining importance in planetary missions. These missions require precise spacecraft navigation, especially in scenarios where GPS positioning is unavailable, such as current and future missions to the Moon or Mars. This paper presents two novel elements: first, the utilisation of on-board Convolutional Neural Networks (CNNs) to provide real-time in-orbit navigation for planetary orbiters, and second, a new method for generating synthetic images to train the CNNs. The proposed solution consists on a set of regression CNNs, each responsible of estimating a different part of the pose solution. Furthermore, multiple data augmentation techniques have been implemented in the training process to extend during runtime the training set and successfully fill the gap between synthetic and real images. To illustrate the proposed solution, OPS-SAT, an Earth-orbiting spacecraft in a Sun-synchronous orbit has been selected as a use case to test and validate the CNNs, improving the current Two-Line Element (TLE) derived geo-localisation tag of its images. Although extensive datasets of Earth imaging exist, including a wide range of illumination conditions, weather conditions or seasonal changes, a method for systematically generating synthetic image datasets for generic conditions is proposed. Earth albedo and terrain elevation datasets have been combined into a very high definition Earth 3D model, enabling photo-realistic synthetic image generation.

3.3. Introduction

Planetary Missions play a pivotal role in unraveling the origin and evolution of the Solar System and the search for extraterrestrial life. Since the inception of the Space Age, satellites have been instrumental in capturing planets topography, surface and atmospheric composition, magnetic field, climate patterns, and diverse data streams for a multitude of applications. The past years have witnessed an exponential surge in Earth Observation (EO) and Solar System exploration missions, with nearly a thousand satellites launched by 2021, attributed in part to the emergence of CubeSats. These compact satellites capitalize on miniaturization trends and Commercial-Off-The-Shelf components, fostering cost-effective and streamlined spacecraft development [34], [35] not only for EO missions but also for exploring the Moon [100] and Mars [101]. CubeSats have evolved into standardized platforms equipped with scientific instruments, competing with traditional large-scale science satellites. Navigation precision is paramount for the success of these missions, with the demand for sub-meter accuracy in imaging and remote sensing data delivery. The absence of Global Navigation Satellites (GNSS) infrastructure on celestial bodies like the Moon or Mars necessitates alternative position determination methods for exploration missions. Currently, these missions heavily rely on radio signals transmitted

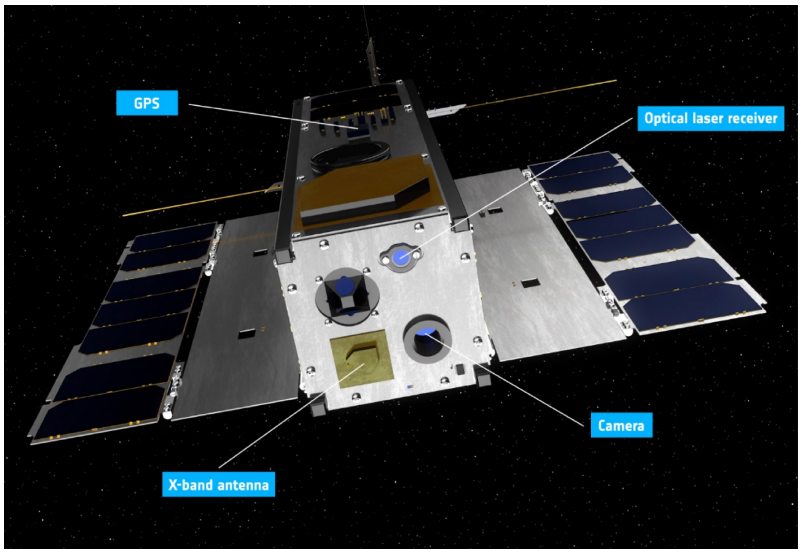


Figure 3.1: OPS-SAT spacecraft and its instruments.

between the spacecraft and Earth-based ground stations, along with the use of Doppler range measurements. Nevertheless, the availability of these methods can be limited due to factors such as ground station usage constraints or budget limitations, specially for low-cost missions. As a result, it becomes imperative to investigate and develop alternative navigation solutions for enhanced space exploration.

The reason for testing in an EO mission is threefold; the large amount of high-detailed datasets of the Earth compared to other celestial bodies enables the testing and validation of a wider range of operational scenarios; on the second place, GNSS positioning may not be available in some cases due to operational or functional constraints, requiring for a back-up solution; finally, some currently flying missions could be used to test in-orbit the trained models. In our investigation, we have selected the OPS-SAT spacecraft as a primary use-case to illustrate our approach for position determination in planetary missions and for testing and validating the proposed solutions. OPS-SAT [102] is an Earth-Orbiting 3U CubeSat launched and controlled by the European Space Agency (ESA) in 2019. The spacecraft is a flying platform easily accessible to European industry and institutions. With an uplink rate four times higher than any ESA spacecraft, it enables the rapid prototyping, testing, and validation of software and firmware experiments in orbit. The spacecraft is equipped with a full set of sensors and actuators including a camera, GNSS receiver, star tracker, reaction wheels, high speed X-band and S-band communication, laser receiver, software defined radio receiver, and a processor with a reconfigurable FPGA at its heart. An artistic representation of OPS-SAT and its instruments is shown in Fig. 3.1. Delving into scenarios where GNSS signal processing is absent, necessitating exploration of alternative navigation approaches, we investigate the utilization of OPS-SAT's onboard camera for position estimation. The OPS-SAT camera has a spatial resolution of 53 meters at a 600-kilometer altitude and a field-of-view spanning 6.5x5 degrees, equivalent to 135x105 kilometers [103]. This imaging capacity enables the iden-

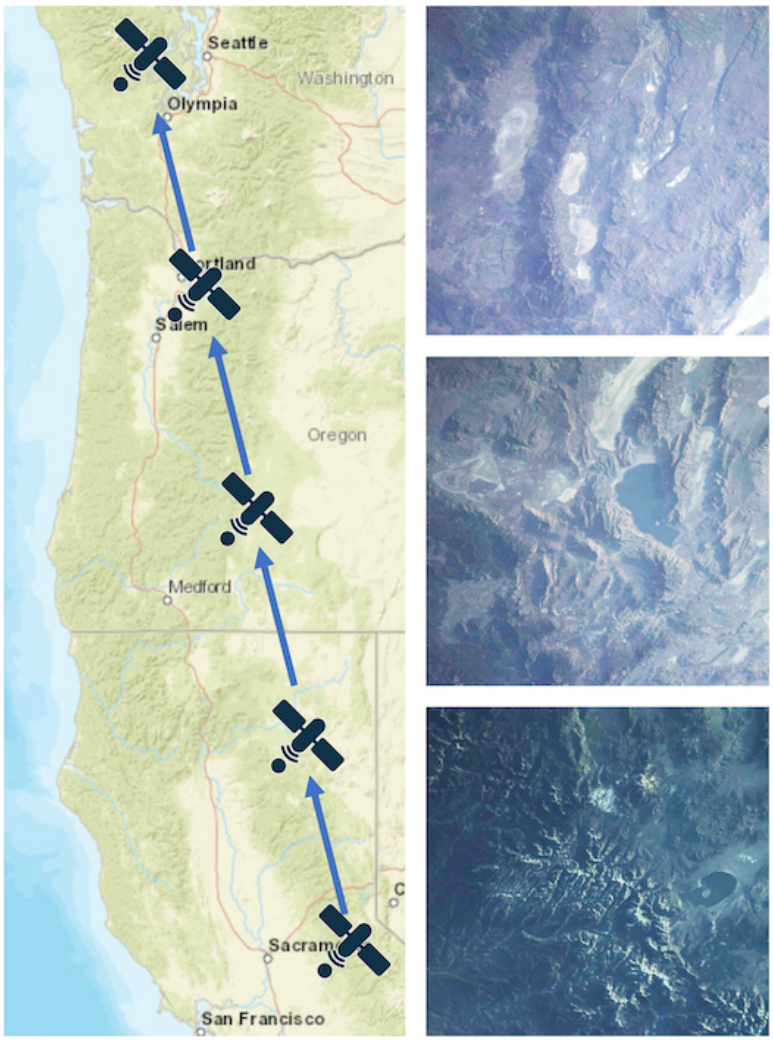


Figure 3.2: Depiction of OPS-SAT consecutive images captured riding the Sun terminator. Source: OPS-SAT Smart Cam Map - ops-sat.io.esa.int/smartcam-map

tification of surface features, contributing to relative position determination.

The spacecraft's approximate orbit is derived from Two Line Elements (TLEs) provided by NORAD [104], describing its dawn-dusk orbit geometry, tracing the Sun terminator and ensuring consistent illumination conditions for successive orbits (see Fig. 3.2), facilitating the work of identifying surface features. Nevertheless, the position tags for a validation subset of OPS-SAT products have been fine tuned by manual inspection in order to correct the orbit uncertainty associated to the TLE. Optical Relative Navigation has proven effective in deep-space missions like Rosetta [30], [43], Hayabusa-1 and -2 [45], [46], and OSIRIS-REx [40], as well as in autonomous Unmanned Aerial Vehicles (UAVs) navigation [105], [106]. Traditional optical-based pose estimation methods rely on classical image processing algorithms that identify pre-defined visible target features or landmarks, consequently they are computationally expensive and its accuracy is strongly dependant on illumination conditions. The main contribution of this paper consists on a novel approach employing Convolutional Neural Networks (CNNs), lever-

aging their capacity for efficient, robust and precise computer vision tasks. Deep learning models, particularly CNNs, have emerged as powerful tools for various geoscience and remote sensing applications in recent years. Their strengths in pattern recognition and feature extraction have led to significant advancements in tasks like: terrain classification of high-resolution satellite images on Earth [51] and other planets [52]; water body mapping with Synthetic-Aperture Radar (SAR) images [107] and despeckling of SAR images [53]; on-board image processing for coverage estimation and detection of clouds [54]; and others. Some recent work has been done on applying deep learning for feature extraction for terrain relative optical navigation in celestial bodies [49], [50], however these contributions still rely on classical feature matching algorithms to estimate the relative position. Unlike traditional feature-based techniques, CNNs can learn nonlinear mappings from the 2-D input gray-scale image (or 3-D RGB image) space to the output space; e.g. 2-D for latitude and longitude; 3-D for the position vector; and 3-D for the Euler angles. This approach, while powerful, presents challenges due to the black box nature of CNNs, necessitating thorough testing and validation to identify potential pitfalls. This paper presents a regression-based method that directly maps input images to continuous 6-dimensional output pose spaces [58], [59]. To ensure the CNNs generalization capabilities, data augmentation techniques are employed, accounting for target shifts and rotations, optical distortions, image exposure, and cloud occultation, while maintaining a reasonable accuracy in the pose estimation. There exist many accessible datasets of Earth Imagery, Spectroscopy and Topography missions such as the Landsat [108] or Shuttle Radar Topography Mission (SRTM) [109] by NASA, Sentinel satellites [110], [111], [112] by ESA, or Pléiades by CNES [113]. However, those sets are restricted by the spacecraft orbit and illumination conditions, limiting the cases for which a Neural Network could be successfully trained.

The secondary contribution of this paper introduces a method to generate high-resolution 3D models of a planet surface combining topography and albedo data at multiple scales and resolutions coming from different sources. These models are then integrated with Blender [114] GPU-accelerated cycles render engine aimed at extending the amount of available training data creating photo-realistic images. The method systematically generates synthetic images with varying camera poses, camera intrinsic parameters (such as focal length, aperture, field-of-view, resolution) and illumination conditions. The illumination source, camera model, and target shape can be configured and modified during runtime when rendering the scene, allowing for the efficient production of different combinations of geometric conditions. Furthermore, Physically-Based Rendering (PBR) [69] materials can be utilized with this method based on the Blender rendering engines, enabling the use of multiple texture maps for achieving increased photo-realism. The fact of counting with synthetically generated sets covering any possible mission scenario, enables the proper training and testing of CNNs capable of providing a robust and efficient pose estimation solution.

The rest of this paper is organized as follows: Section II describes the methodology

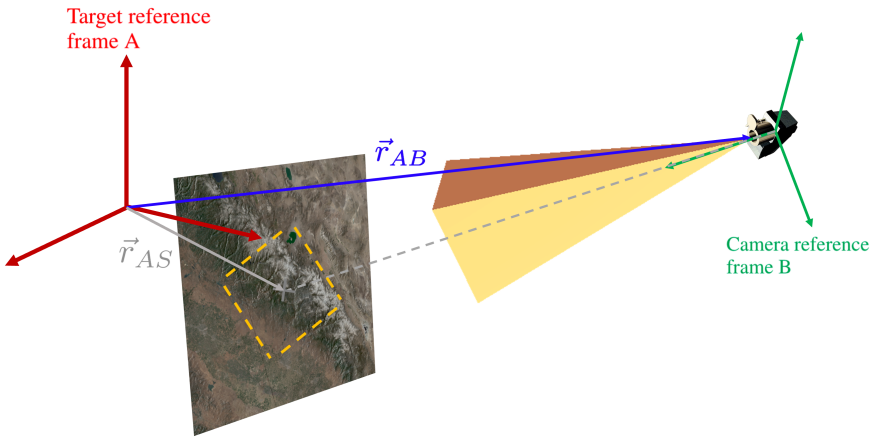


Figure 3.3: Depiction of camera pose estimation with respect to target

used for the synthetic image sets generation; Section III explains the CNNs architecture and training methods; Section IV presents the results of the trained networks and its application to pose estimation; and Section V summarises the conclusions from the current study and the basis for further work and developments.

3.4. Data Generation Methods

The primary objective of this study is to train Convolutional Neural Networks (CNNs) capable of accurately estimating the relative position and attitude of a camera with respect to the Earth, expressed in the Earth's body-fixed frame. The scenario to be reproduced for generating the needed training synthetic image sets is illustrated in Fig. 3.3, where the camera's focal point's position vector, denoted as $\vec{r}_{AB}$, with respect to the Earth-centered body-fixed frame, is the target to be estimated through the CNNs. Furthermore, the CNNs also strive to determine the rotation transformation represented by Euler Angles, corresponding to the conversion from the Earth body-fixed frame (ref_A) to the camera reference frame (ref_B). Alternatively, the CNNs can infer the position of the terrain that the camera observes in relation to the Earth body-fixed frame, i.e. the point corresponding to the intersection of the camera boresight with the surface (denoted by $\vec{r}_{AS}$), in addition to the distance from this point to the camera. When combined with the attitude of the camera, this information can facilitate the computation of the spacecraft position with respect to the Earth. Estimating the observed terrain position can also serve to provide on-board geo-localization data for the acquired images.

To comprehensively assess the CNNs accuracy under various geometric configurations within the depicted scenario, multiple synthetic image datasets have been produced covering the geometric and illumination conditions not available from existing real image sets. The initial phase of this study centers around the development of tools tailored for producing extensive sets of labeled synthetic images, essential for the training of CNNs. These datasets offer configurability, enabling the introduction of diverse image effects

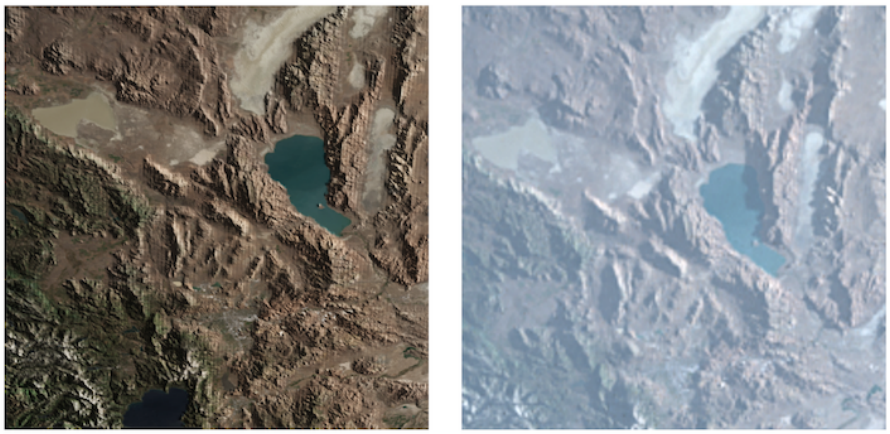


Figure 3.4: Effect of projected ellipsoid with normal map compared to 3D model with projected shadows and atmosphere scattering.

such as the direction and intensity of the illumination source, camera position and orientation (including off-nadir and around boresight rotation), camera field-of-view aperture angles, and image resolution.

To render synthetic images, multiple data sources are necessary to construct the scene accurately. Leveraging the capabilities of Google Earth Engine [60], a public platform facilitating access and manipulation of diverse datasets related to Earth Observation Missions, users can define parameters such as the area of interest, spatial resolution, temporal range, cloud percentage and output format. A wide range of Earth observation imagery datasets, including Landsat, Sentinel, MODIS, and higher-resolution mapping datasets, are publicly accessible through Google Earth Engine. Notably, Sentinel-2 [115], a cornerstone of the European Space Agency's Copernicus program, stands out as a wide swath, high-resolution, multispectral imaging mission with a 5-day global revisit cycle. This mission offers visible band images at spatial resolutions up to 10 meters per pixel, making Sentinel data perfect for evaluating multiple spatial resolution images. Moreover, the mission's high-frequency groundtrack repeatability facilitates the assessment of CNN performance across seasonal variations.

While these datasets serve as a foundation for producing synthetic images, they often come with limitations, such as low phase angles (the angle between the direction of illumination and the position vector of the observer). As a result, they may not be suitable for scenarios beyond their original geometry, such as Dawn-dusk orbits with high phase angles or geosynchronous orbits with variable phase angles. To address this shortcoming, the incorporation of terrain elevation data is imperative to properly model surface slopes and shadowing, producing an accurate representation of real images. Notably, topography data is available from sources such as the Shuttle Radar Topography Mission (SRTM) [116], offering global coverage at a resolution up to 90 meters per pixel, or even finer resolutions, ranging from a few meters to less than one meter per pixel for specific regions.

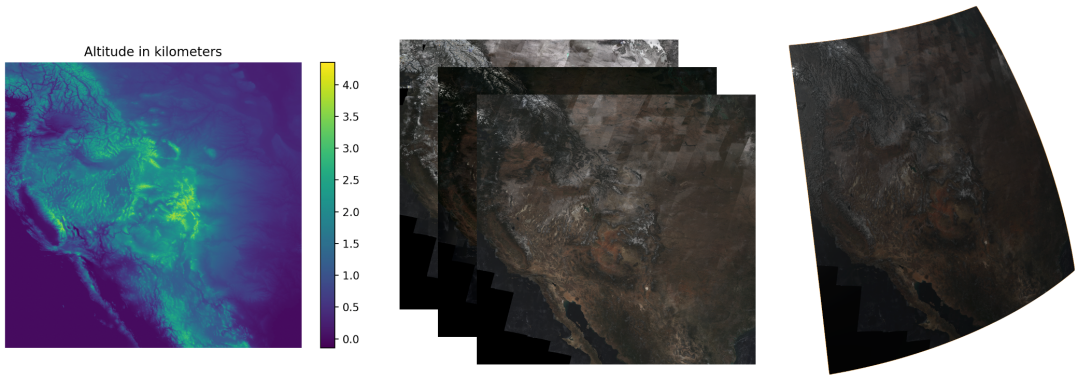


Figure 3.5: From left to right; DEM based on SRTM data; multiple albedo maps based on Sentinel-2 data; resulting textured Earth 3D model.

Elevation data can be seamlessly transformed into a normal map, containing vector normals for each point on the surface. Typically used in rendering software, this map models the angle of incidence – the angle between the illumination direction and the vector normal to the illuminated surface. It is employed to capture how the surface appears illuminated from various angles. By merging surface albedo with the normal map as textures on an ellipsoid, synthetic Earth images can be generated. However, this method is better suited for images with lower phase angles, as the relevance of terrain-cast shadows increases with higher phase angles. To address scenarios with pronounced cast shadows, a high-resolution 3D model of the Earth has been constructed. Beginning with an ellipsoidal mesh, vertex positions have been adjusted based on SRTM elevation data. This high-resolution 3D model empowers the accurate simulation of shadows cast by surface features upon themselves. It is important to note that this approach – using a shape model – incurs significantly higher computational costs compared to the normal map technique. For instance, a mesh with 300 meters per pixel spatial resolution, covering a 10-degree latitude and longitude area, can swiftly occupy several gigabytes of memory. However, the dawn-dusk orbit, characterized by elevated incidence angles and projected shadows that encompass substantial nearby regions, necessitates this heightened level of detail. Fig. 3.4 illustrates the disparity in effects between projected shadows generated using the shape model versus the normal map rendering approach. Despite atmospheric light scattering that prevents the complete obscuration of areas beneath projected shadows, these shadows remain crucial elements in images and their accurate representation is paramount for generating images suitable for training CNNs for precise pose estimation.

For the purpose of this study, the open-source software Blender has been selected to incorporate the generated Earth shape model and produce synthetic image datasets. Fig. 3.5 shows for the area of interest: the Digital Elevation Model (DEM) derived from SRTM data; multiple albedo maps derived from Sentinel-2 data obtained between January 2021 and December 2022; and the resulting 3D model with combined topography and surface albedo. The general workflow, illustrating the combination of Earth Engine data into a 3D

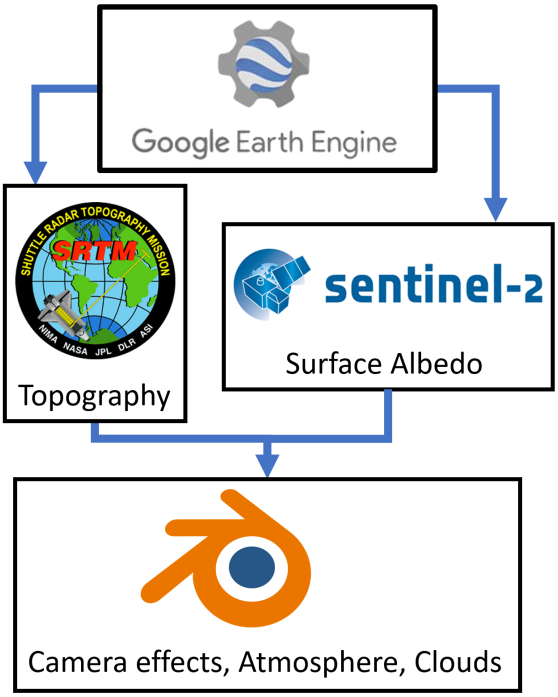


Figure 3.6: Workflow for extracting and combining datasets from Earth Engine for generating the 3D model ingested in Blender for rendering.

model within Blender, is outlined in Fig. 3.6. Both the elevation and albedo maps have been implemented with a resolution of 200 meters per pixel. Through the programmatic interaction between Blender and Python, the process of scene creation and updating is configured and automated for each specific geometric scenario under investigation. Notably, the extensive array of adjustable parameters within its rendering engine facilitates the attainment of the required level of photorealism and fidelity to real OPS-SAT images. Regarding the image resolution, the publicly available OPS-SAT images are trimmed to a squared shape with 583 by 583 pixels, however for training the networks, smaller resolution images of 224 by 224 pixels are produced resulting in reduced memory requirements.

To train and test the CNNs, the generated image datasets cover a region spanning 30 degrees in both latitude and longitude. Specifically, the central and west regions of the United States, slightly extending over Canada and Mexico, have been selected for this study, given the greater number of OPS-SAT images available at the time from these regions for which model validation can be performed. Multiple camera effects have been randomized to ensure diverse training conditions. The camera coordinates are randomized within predetermined latitude and longitude ranges. The camera altitude relative to Earth surface varies between 450 and 550 kilometers, with a distinct random value for each synthetic image. This approach aids in training the CNNs to accommodate the variable altitude of the elliptical orbit. For attitude, the starting point is nadir pointing, with a random off-nadir angle of up to ± 40 degrees (both along-track and cross-track). It is worth noting the effect of the off-nadir in this scenario, where the surface of the target covers the full image and the limb is not observed (compared to instruments with a very

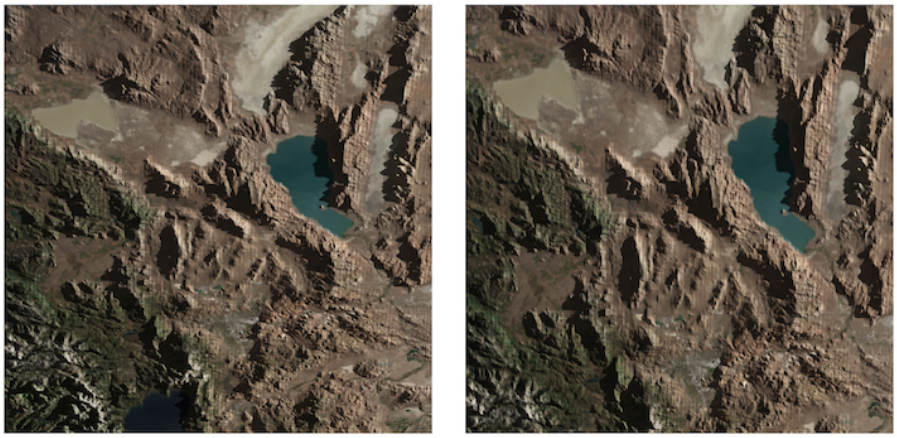


Figure 3.7: Effect of variable off-nadir angle on the appearance of the same region.

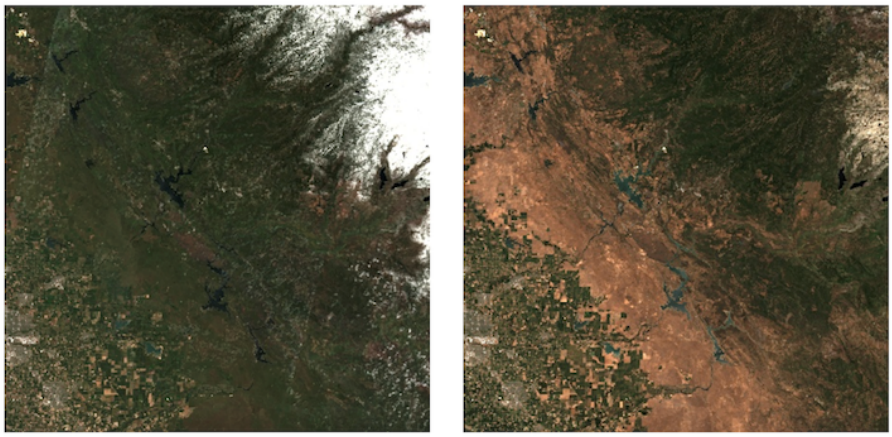


Figure 3.8: Effect of seasonal variation between Spring and Summer of the same region.

wide field-of-view or deep space missions to minor bodies, when the complete target is visible in the image). A large off-nadir angle results in distorted shapes, with surface features appearing shrunken along the off-nadir direction compared with the same area observed under a pure-nadir perspective as illustrated in Fig. 3.7. Furthermore, a random rotation around the camera boresight (roll angle) ranging from 0 to 180 degrees is applied, enhancing the network's capacity for rotational invariance. The parameters encompassing camera field-of-view, focal length, and output image resolution are configured in the Blender render engine. Regarding the illumination source, a directional light is defined, initially adopting its orientation from SPICE [71] in reference to the Earth-fixed coordinate system for a specific epoch of interest. Subsequently, variations in intensity and direction of up to 20 degrees are introduced, reflecting the season-wise variability in Sun position. This adaptability of the illumination source is crucial for training the CNN to exhibit illumination invariance.

To represent Earth's 3D surface aspect within the scene, a suite of albedo maps has been generated. Each map is constructed from Sentinel images corresponding to different seasons. It is worth noting that Earth's appearance transforms drastically over the course

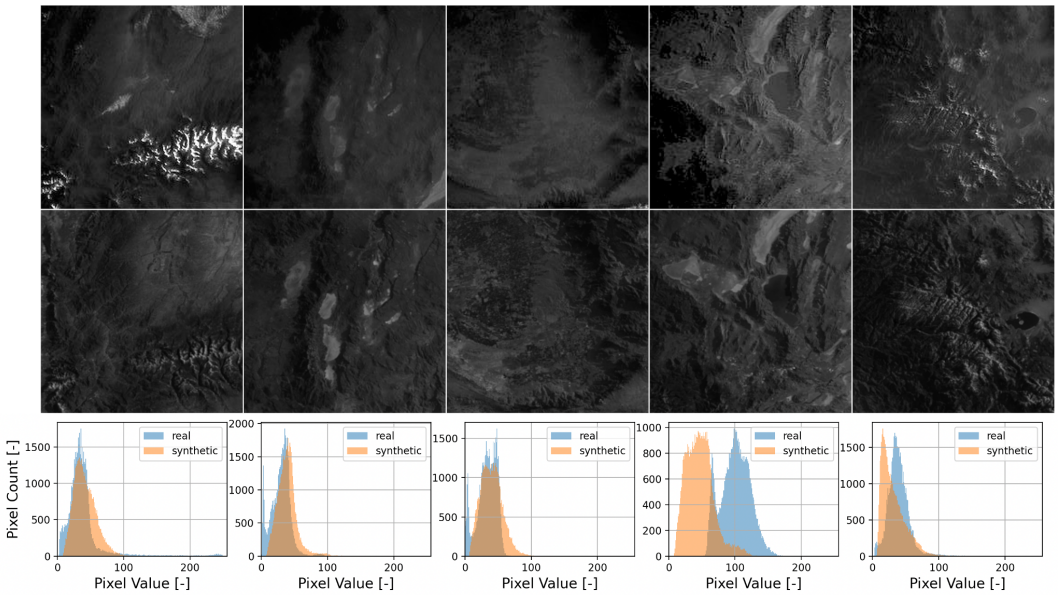


Figure 3.9: Comparison of real OPS-SAT images (top row) vs synthetic ones (second row). The grayscale intensity histogram for both images is displayed below

of a year, for instance, areas may appear white due to snow in winter, green during spring, and yellow or brown post-harvest, with even water bodies varying in size or visibility between seasons. This phenomenon is illustrated in Fig. 3.8, showcasing the transformation of California's mountainous regions from green and white in January to a palette of brownish hues between June and August. By incorporating these dynamic albedo maps for the observable surface, the network is trained to accommodate these changes and factor them into its position estimation. Numerous sets of synthetic images have been produced, each implementing distinct combinations of the aforementioned effects as summarised in Table 3.1. This strategy facilitates a comprehensive evaluation of the influence of each effect on CNN performance. Initial training attempts utilizing smaller datasets of 20,000 images encountered overfitting issues. This phenomenon occurs when a model memorizes specific training data points rather than learning generalizable patterns. To mitigate this issue, we opted for a significantly larger dataset composed of 50,000 images (split in 90% for training and 10% for testing), ensuring a more homogeneous distribution of geometric conditions across the image set. This broader data representation provides the model with a wider range of examples, fostering its ability to generalize to unseen data and reduce overfitting.

3.5. Deep Convolutional Neural Network Architecture and Training Methods

The baseline CNNs developed for this study, utilizing TensorFlow [117], are constructed based on a simplified version of the AlexNet architecture [80]. The *KAMnet* baseline architecture is displayed in Fig. 3.10. This streamlined architecture facilitates rapid training and convergence, enabling the evaluation of the influence of various image and geometric

Table 3.1: Description of the synthetic image datasets generated for this work

Dataset	Description	Images
sim224US_sr	No Gaussian noise, Nadir-pointed, No boresight rotation	50000
sim224US_ng_sr	Nadir-pointed, no boresight rotation	50000
sim224US_ng_sr_rr45	Nadir-pointed, boresight rotation [0, 45]	50000
sim224US_ng_sr_rr180	Nadir-pointed, boresight rotation [0, 180]	50000
sim224US_ng_sr_rr45_o40	Off-nadir [-40, 40], boresight rotation [0, 45]	50000
sim224US_ng_sr_rr180_o40	Off-nadir [-40, 40], boresight rotation [0, 180]	50000
sim224US_ng_sr_rr180_o40_s30	Off-nadir [-40, 40], boresight rotation [0, 180], shear and zoom [-30, 30]	50000
sim224US_ng_sr_rr180_o40_s30_cs50	Off-nadir [-40, 40], boresight rotation [0, 180], shear and zoom [-30, 30] channel shift [-50%, 50%]	50000
sim224US_ng_sr_rr180_o40_s30_cs50_co	Off-nadir [-40, 40], boresight rotation [0, 180], shear and zoom [-30, 30], channel shift [-50%, 50%], cutout erase	50000
sim224US_ng_sr_rr180_o40_s30_cs50_co_alb	Off-nadir [-40, 40], boresight rotation [0, 180], shear and zoom [-30, 30], channel shift [-50%, 50%], cutout erase, variable albedo	50000

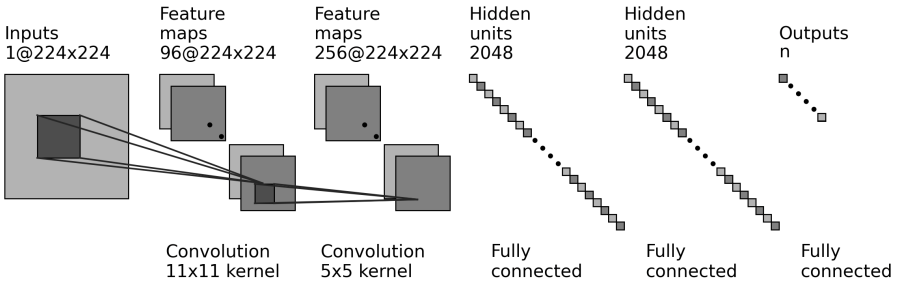


Figure 3.10: KAMnet baseline Convolutional Neural Network Architecture

effects on the training process. After analyzing these trends, more complex architectures, characterized by deeper and wider layers, such as VGG [118] or ResNet [84], have been employed, resulting in enhanced levels of accuracy. The network architecture consists of two convolutional layers followed by three fully-connected layers. The input shape of the convolutional layers is set at 224 by 224, mirroring the dimensions of the produced synthetic images. Max Pooling layers have been inserted after both convolutional layers, aiming to confer invariance to minor displacements in the feature maps. This approach capitalizes on the observation that most pooled outputs remain stable in the presence of slight shifts in input features [89].

The kernel size and strides of the convolution layers have been fine-tuned to optimize estimation accuracy. Preceding each fully connected layer, a flatten layer reshapes the output tensor dimensions to match the anticipated input format for the subsequent fully-connected layers. To mitigate overfitting, a dropout rate of 0.25 has been adopted, indicating the proportion of input units that are dropped during training. For all layers except the final fully-connected layer, the ReLU (Rectified Linear Unit) activation function [91] has been employed. This piecewise-linear function maintains the input value if it is positive; otherwise, it outputs zero as indicated in (3.1). The nearly-linear nature of ReLU

retains many characteristics that simplify optimization of linear models [89]. Moreover, the output being zero for negative inputs maintains the nonlinearity essential for capturing complex relationships within datasets. ReLU's selection over other activation functions such as sigmoid or hyperbolic tangent is attributed to its ability to counteract the vanishing gradient problem [89].

$$\begin{cases} y = x & x > 0 \\ y = 0 & x \leq 0 \end{cases} \tag{3.1}$$

Regression-type CNNs have been used for the estimation, with the last fully-connected layer implementing the linear activation function. Consequently, this layer directly yields as output the values to be estimated. Specifically, the output dimensions are configured as follows: two dimensions for estimating the latitude and longitude of the observed area center within an image, essentially denoting the intersection point of the camera boresight with Earth's surface; one dimension for the distance from the camera to this surface point; two dimensions for Yaw and Pitch angles (off-nadir); and one dimension for Roll angle (rotation around the boresight direction). The rationale behind employing multiple distinct Regression CNNs to estimate position solution, off-nadir angles, and roll angle stems from the notable disparity in scale among these outputs. When the target variables exhibit a wide range of values, the resultant weights can experience abrupt changes, thereby introducing instabilities during the training process [93], ultimately reducing the accuracy of the estimation. It is essential to highlight that the roll angle spans a full 360 degrees, whereas the off-nadir angles are bounded to approximately 40 degrees – beyond this threshold, atmospheric scattering makes the surface underneath barely appreciable in the captured images. Regarding latitude and longitude training for localized datasets, a 30-degree range is adopted as the starting point. It is worth noting that a multi-branch model [97] could be used to address the scale disparity between the different target variables. A single architecture which has multiple branches with common upstream convolutional layers would alleviate the total size of the CNN and improve performance. However, it was decided to implement independent sequential networks, seeking for modularity and faster training, allowing to investigate different image effects independently on each target variable.

To attain optimal performance for the designed architecture and to streamline training times, certain parameters governing the training process must be carefully defined. The most relevant among these are the Loss function, the optimization algorithm, and the training epochs. The Loss function quantifies the model's error or loss at each iteration of the optimization algorithm. This guides the subsequent weight updates in a manner that aims to diminish the defined loss in forthcoming iterations. For the Regression CNNs, suitable loss functions include the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). Given that roll and off-nadir angles can be treated as zero-mean Gaussian distributions, MSE is the chosen loss function for CNNs estimating these angles. The selection of MSE is driven by its capability to penalize larger estimation errors,

consequently encouraging the network to update weights in a way that avoids generating outliers in the output [96]. In addition to the aforementioned standard loss functions, a customized loss function has been formulated to encompass the Mean Translation Error (MTE) between the ground truth and the estimated position for a specific image. While employing MAE or MSE for position estimation could lead to minimal error in some of the coordinates of the position vector, leaving the remaining coordinates with substantial deviations, the utilization of translation error focuses on minimizing the overall magnitude of the position error vector. Equations for MAE, MSE, and MTE are shown in (4.3), (4.4), and (4.5) for n samples, where y_i represents the ground truth and y_i' denotes the predicted value.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i'| \tag{3.2}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2 \tag{3.3}$$

$$\text{MTE} = \frac{1}{n}\sum_{i=1}^{n}|\vec{y}_i - \vec{y'}_i| \tag{3.4}$$

For the optimization algorithm, Adam (Adaptive Moment) [55] has been selected. This algorithm, devoted to the minimization of the loss function, autonomously adjusts the learning rate during the training epochs. The choice of Adam is based on its capacity to adapt to varying learning rates, thereby optimizing convergence speed. The training epochs (number of passes through the whole train set) have been set to ensure convergence of the validation loss.

Finally, on top of the different image effects introduced during the generation of the training sets, additional data augmentation effects are applied during runtime to randomly extend the training sets producing random variations of the nominal set during each training epoch. A Shear transformation is applied by fixing one axis of the image while stretching the other by a random shear angle. In a similar way the Zoom transformation scales the image horizontally or vertically by a random fraction. These two effects combined help modelling distortions in the images due to different perspectives but also lens optical distortions. On top of these effects, random Gaussian noise and exposure Gamma correction are applied to model light variations and noise. These also help dealing with image distortions derived from atmospheric effects such as temperature and pressure variations or turbulence. Less appreciable to the human eye, is the random channel shift transformation which has been applied to the images. Looking at the grayscale pixel histograms of a real OPS-SAT image and its synthetic counterpart, illustrated in Fig. 3.9, it is noticeable that the range and occurrence of pixel values are very similar between the two images. However, the histogram of the real image is subtly shifted towards higher pixel values. This histogram shift phenomenon is captured by the introduced channel shift ran-

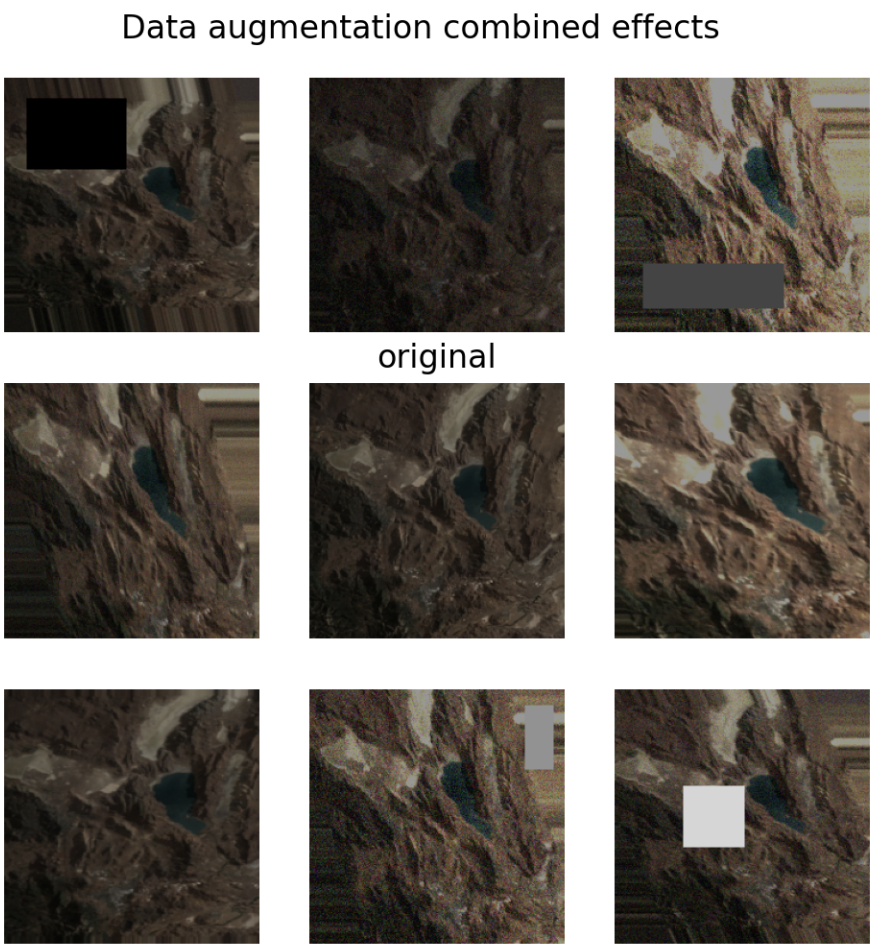


Figure 3.11: Example of combined data augmentation effects (random shear, random zoom, random exposure, random channel shift, random Gaussian noise, random cutout erase)

dom transformation. Moreover, a cutout erase effect has been added randomly removing a rectangular section of the image with variable size and color. This helps adapting the model to work in a situation in which the surface features of a section of the image are occult by elements such as clouds. Fig. 3.11 depicts some examples of applying the previous effects all-together to the same original image (in the middle).

3.6. Results

In this section, we present the experiments undertaken to assess the training and performance of the formulated neural networks. The principal metric analyzed for each trained CNN pertains to the evolution of the loss function throughout the training process. This encompasses the number of training epochs required for convergence, the regularization and potential overfitting, and the influence of diverse image effects and geometric conditions on the training progress and estimation accuracy. The first evaluated CNN is devoted to estimating the longitude and latitude of the input image center. For this specific net-

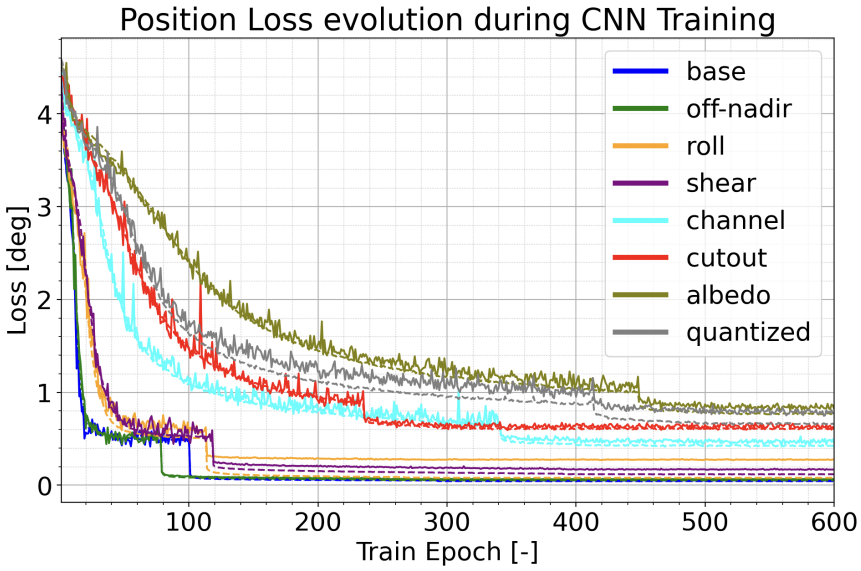


Figure 3.12: Longitude and latitude loss function evolution during training.

Table 3.2: CNN architectures achieved loss and size comparison

Architecture	**Loss [deg]**	**Size [MB]**
MobileNetV2	1.553	9.24
EfficientNetB0	1.750	16.07
DenseNet121	1.263	27.32
VGG16	0.910	56.38
VGG19	0.775	76.63
Baseline KAMnet	0.797	116.42
ResNet152V2	1.181	223.65

work, factors such as off-nadir angles and rotational deviations around the boresight wield substantial influence over estimation accuracy. The different produced image sets, summarised in Table 3.1, have been used to train multiple regression CNNs in order to asses the impact of each image effect and geometric condition on the networks performance. The coordinates of the image's center, together with complementary geometric information, are encoded within the label files that are loaded prior to training, thus configuring the label vector. The last fully-connected layer of this CNN has dimension 2, corresponding to the 2-D vector of planetocentric latitude and longitude on the Earth fixed reference frame expressed in degrees.

Fig. 3.12 illustrates the evolution of both training (dashed lines) and validation (solid lines) loss across the training epochs, covering a range of diverse image datasets. Commencing with a baseline case encompassing images solely generated under nadir pointing and zero roll conditions – where each image pixel columns align with the projection of

the Earth's North pole onto the camera detector plane – the validation loss reaches 0.05 degrees (equivalent to a 5-kilometer error on Earth's surface). It is worth noting that the validation loss follows the same evolution as training loss, experiencing a fast decrease up to epoch 40 followed by a quasi-linear slow-rate decrease until epoch 100, at which after a further decrease corresponding to the adjustment of the learning rate, the progress halts. Subsequently, when introducing an off-nadir angle up to 40 degrees, the converged loss is very similar to the baseline, however, the off-nadir acted as a regularizing agent, reducing the difference between training and validation loss and slightly improving the later. When adding a random roll rotation of up to 45 degrees around the boresight direction, the validation loss exhibits some instabilities during the fast decrease phase, eventually converging to a comparable level to the previous cases but with a larger gap between training and validation loss. After adding the zoom and shear random effects, the loss exhibited larger instabilities during most phases of the training, however these effects again acted as regularizers, slightly increasing the training loss but reducing overfitting and consequently the validation loss to 0.17 degrees. The random channel shift of the pixel values leads to a substantial increase of the loss which could be reduced with a fine tuning of the maximum and minimum shift values but as anticipated in Fig. 3.9, it is critical for overcoming the gap between synthetic and real images. On top of this, when introducing random cutout erase effect and variable albedo maps in the training set, the training dynamics unveil a more gradual loss decrease, necessitating an increased number of epochs to achieve convergence. Consequently, the converged loss increased to 0.8 degrees, equivalent to 90 kilometers on Earth's surface. Finally, it is worth noting the loss evolution resulting from applying quantization-aware training involving simulating the effects of quantization during the training process. This introduces some level of noise into the gradients when limiting the weights precision, acting as a regularizer and helping to escape local minima, therein slightly improving validation loss.

After testing the baseline KAMnet architecture with the different combinations of geometric and image effects, more complex architectures have been tested for the most demanding scenario implementing all-together the previously described effects. The results are summarised in Table 4.3 ordered by increasing model size. It can be appreciated that while performing well in terms of achieved loss, the baseline KAMnet architecture is the second-largest tested architecture. Using a deeper architecture like VGG19 (in green), the CNN is capable of capturing the complex patterns of the synthetic image sets while reducing the number of parameters, consequently decreasing the model size by roughly a 35% while achieving the smallest loss. However, moving to more complex architectures like DenseNet121 or EfficientNetB0 the model size is further reduced, but with a higher impact on the CNN loss. Depending on the required accuracy and memory requirements, it could be suitable to trade-off model size for accuracy by using an extremely light architecture from the MobileNet family, which having less than 10MB supposes a decrease of model size of 90% with an impact on loss of just 0.8 degrees.

For the training of the CNN estimating the distance from the camera focal point to the

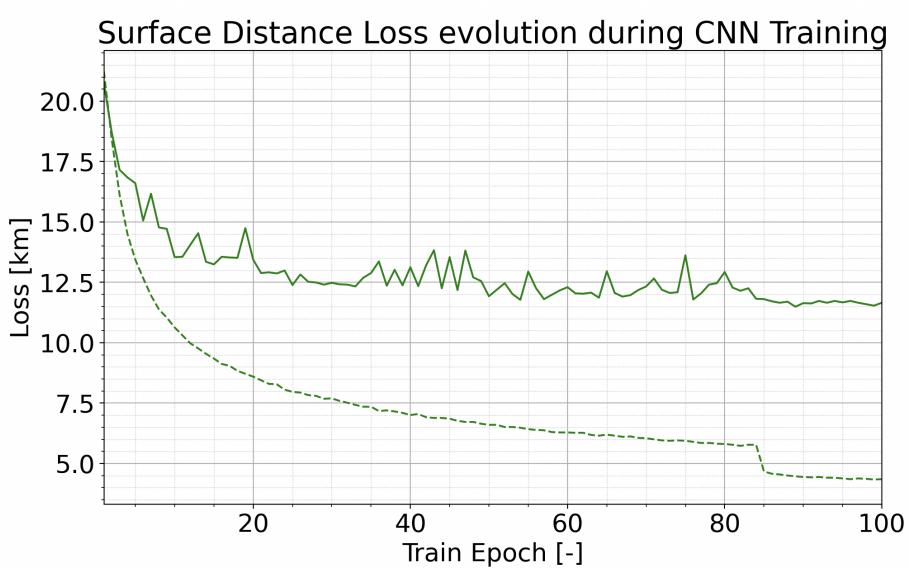


Figure 3.13: Surface distance loss function evolution during training.

boresight-surface intersection point, it was not possible to train the network from scratch as the optimizer did not manage to converge to a solution. Because of this, we applied the transfer learning technique using the VGG19 architecture pre-trained with weights for the latitude and longitude estimation so the initial model would already count with some insight on the input images, therein facilitating the search for local minimums. Keeping the pre-trained feature extraction layers, the optimizer now effectively finds a minima for the loss function on the camera to surface distance, reaching a validation loss value of 12 kilometers as shown in Fig 3.13. Nevertheless, the loss evolution shows a large overfitting since the first epochs of the training, with a training loss notably smaller than the validation one.

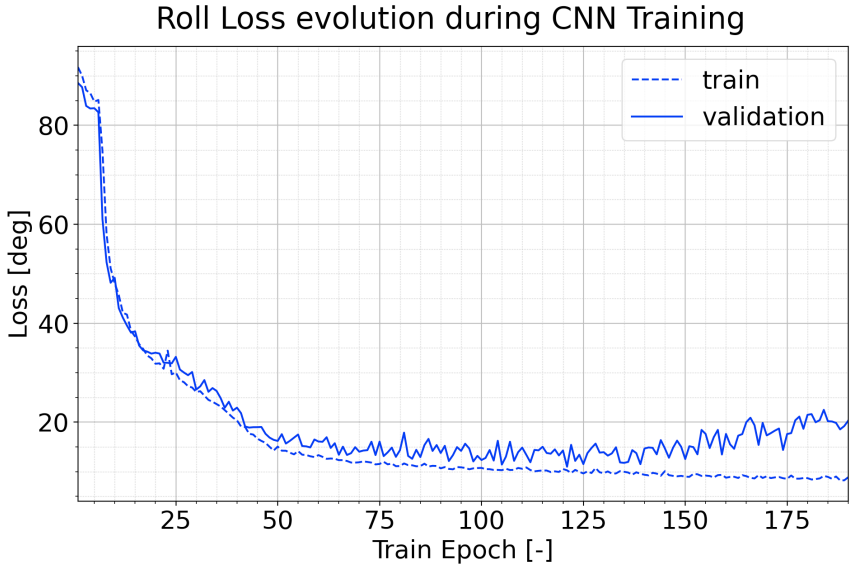


Figure 3.14: Roll angle loss function evolution during training.

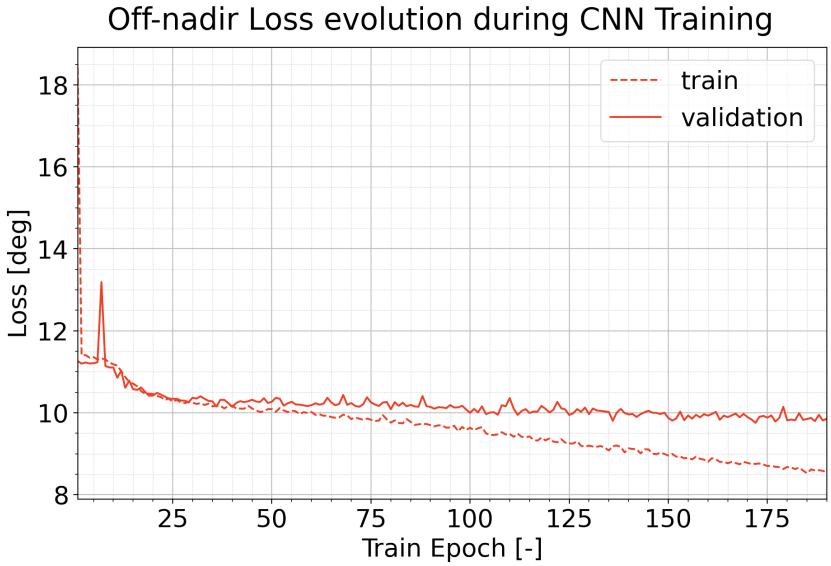
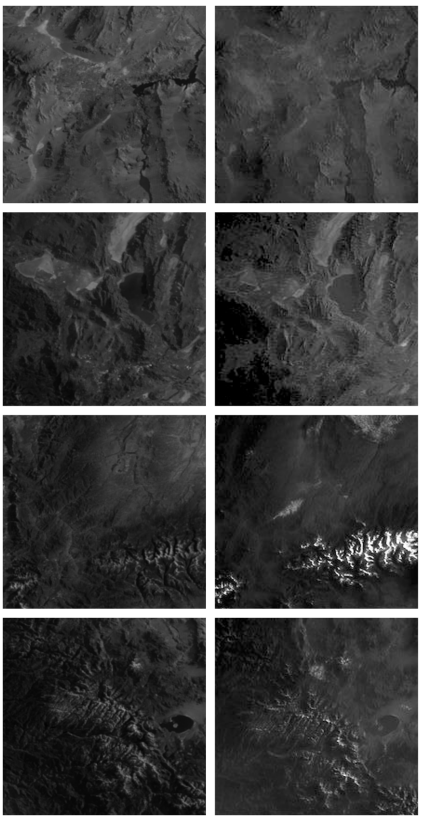


Figure 3.15: Off-nadir loss function evolution during training.

The loss evolution for the roll angle and off-nadir angles CNNs are illustrated in Fig. 3.14 and Fig. 3.15, respectively. Both networks were trained employing the image set characterized by random roll angles spanning up to 180 degrees and random off-nadir angles encompassing up to 40 degrees. Concerning the roll angle CNN, validation and training losses exhibit a closely-matched downward trend until epoch 125. Beyond this point, loss function diverges as the network overfits the training set. In light of this, the trained model – in this particular instance – corresponds not to the last epoch but rather the iteration yielding the most optimal validation loss value during training, equal to 10 degrees. Regarding the off-nadir angles CNN, convergence of validation loss is swiftly attained during the initial stages of training, at epoch 25, however this network also overfits and train loss keeps decreasing. For these two CNNs, other smaller network architectures or a larger training set would probably help reducing the overfitting and improving validation loss.

3.6.1. Optimization and Quantization

Because the trained CNNs are devoted to be uplinked and used on-board an already flying spacecraft such as OPS-SAT, the size of the network should have been optimized to keep the same architecture but at the same time, fulfill the link budget limitations. Keep in mind that the neural network coefficients have to be uploaded to the spacecraft for it to run the CNN as it is trained on ground. With this purpose, efficiency oriented state-of-the-art architectures as MobileNet [119] or EfficientNet [88] families have been trained with the same image sets and training configuration. Something that has to be taken into account and properly controlled if needed, is that these architectures are much deeper and complex than the baseline architecture, therefore requiring much more memory for storing the gradients and updating the weights at each epoch with the same batch size. In



	Lon [deg]	Lat [deg]
True	-114.877	35.921
Synth	-114.868	35.781
Real	-114.643	36.460

	Lon [deg]	Lat [deg]
True	-119.778	40.068
Synth	-119.576	39.207
Real	-119.644	38.735

	Lon [deg]	Lat [deg]
True	-110.964	41.231
Synth	-111.140	40.717
Real	-110.648	40.725

	Lon [deg]	Lat [deg]
True	-119.561	38.186
Synth	-119.484	37.479
Real	-119.643	37.551

Figure 3.16: Longitude and latitude estimation for some real and corresponding synthetic images.

addition, exploding or vanishing gradients should be carefully looked after and controlled for a successful training. Moreover, to further reduce the size of the trained networks, the Quantization [120] technique has been applied. Post-training quantization is a conversion technique that reduces the model size and can help improving CPU usage and hardware accelerator latency, all while having a very small impact on model accuracy. Full integer quantization has been applied, differing from other quantization techniques like dynamic range quantization, in the fact that not only model weights are converted from floating point to integer, but all the network math becomes integer quantized. This technique reduces the model size by a factor of 4 and improves CPU speedup by a factor of more than 3. The full integer conversion has been performed with the TensorFlow Lite Converter. The quantized model has been tested in a ARM64 Dual Core with 8GB RAM getting a processing rate of 12 frames per second and a CPU usage of 62%. Testing with the validation set showed no degradation in accuracy with respect to the source model, even for models which were not quantization-aware trained.

3.6.2. Validation with Real Images

While the introduction of the variable albedo may seem to introduce unnecessarily large errors in the overall estimation, it is required for the trained network to deal with variations

on the real images. Fig. 3.16 shows some examples of the position estimation with the network trained with variable albedo for real and its corresponding synthetic images. This demonstrates that introducing the previously mentioned image and geometric effects results in a negligible degradation of the estimation accuracy when ingesting real images never seen before by the network trained with synthetic images. Of special interest is the third case in Fig. 3.16, showing the Kings Peak mountain in Utah covered in snow in the real image (right) compared to the synthetic image (left), but with comparable levels of accuracy for both images.

3.7. Conclusion

In this work, the successful training and testing of CNNs applied to relative navigation on an Earth Observation mission has been introduced, setting up the basis for developing feasible deep learning navigation algorithms, not only for EO missions but for planetary and deep space missions as well. A method for the generation of high resolution planetary surface models and large photo-realistic synthetic image sets using GPU accelerated methods has been presented. This pipeline allows for the generation of image sets implementing user-defined geometric condition, illumination conditions, and image effects, enabling the training and testing of CNNs. This configurability enables the creation of an scenario with multiple variations covering the whole range of situations in an hypothetical Earth Observation or Planetary mission. Furthermore, multiple CNNs have been trained and tested using the produced image sets, each of them implementing different image effects and restricted to specific geometric conditions. Thanks to the applied data augmentation effects during runtime, the gap between real and synthetic images has been successfully overcome, enabling the estimation of the position on input real images with CNNs trained only with synthetic images derived from former datasets. Multiple state-of-the-art CNN architectures have been tested for the most complex scenario, identifying a trend between model size and accuracy. For this use case, a balance between size, shape and depth, results in VGG19 outperforming other architectures. The implementation of Time Distributed CNNs or Recurrent CNNs is of special interest for future works, as these type of Neural Network allows the evaluation of a sequence of images instead of a single image, taking advantage of the time-driven patterns which are present in the images captured by an orbiter spacecraft.

4. BennuNet – An Update on Applying Deep Learning for Minor Bodies Optical Navigation

The content of the current chapter coincides with the following journal publication:

A. Escalante, P. Ghiglino and M. Sanjurjo-Rivo, "Bennunet - An Update on Applying Deep Learning for Minor Bodies Optical Navigation," in IEEE Transactions on Aerospace and Electronic Systems, Volume 61, Issue 3, Pages 7125-7139, June 2025, doi: doi.org/10.1109/TAES.2025.3533471 (**Paper III**).

4.1. Paper content and author contribution

This article delves into the implementation of custom loss functions and specialized target variables tailored for different blocks of the CNN-based pose estimation framework. In particular, the estimation of the camera reference frame rotation is decomposed into multiple networks, each with distinct target variables, effectively mitigating the training instabilities observed in previous works. This refinement completes the full CNN-based solution outlined in objective **O.2.2**.

To enable deployment on resource-constrained small platforms, as highlighted in gap **G.5**, existing state-of-the-art architectures have been adapted to meet the hardware limitations of real missions. By optimizing network size, inference time, and accuracy, novel CNN models have been successfully designed and trained for pose estimation and autonomous navigation. In alignment with objective **O.2.3**, the trained navigation models have been deployed and validated on flight-ready hardware intended for small platforms.

For this contribution, the Ph.D. candidate has been responsible for designing and validating new CNN architectures that substantially reduce the weight of state-of-the-art models while retaining operational accuracy levels. The author also implemented new training target variables and loss functions that further improved the CNN pose estimation accuracy. The Ph.D. candidate prepared the manuscript submitted to the indexed journal *IEEE Transactions on Aerospace and Electronic Systems*, including the previous analysis together with the results of the testing on flight-ready hardware.

4.2. Abstract

This paper presents Bennunet, a hybrid neural network-based method, devoted to on-board spacecraft relative position and attitude estimation in the vicinity of minor bodies using monocular vision. It is a follow-up investigation of Churinet, which set up the basis for using neural networks for pose estimation, offering a lightweight and robust alternative to the computationally expensive traditional methods which may fail under adverse illumination conditions. In this case, the asteroid Bennu has been chosen as the target of the investigation given the extensive data derived from the OSIRIS-REx mission. Multiple shape models of Bennu have been used to produce synthetic image training sets covering the whole range of camera position, attitude, illumination conditions, camera field-of-view, image resolution, and target albedo map variation, allowing to study the impact of different geometries and image effects in the network performance and making it more robust. Moreover, modified state-of-the-art architectures have been implemented for Bennunet, substantially improving its performance compared to the baseline Convolutional Neural Network (CNN) used in previous works. In addition, the implementation of a Time Distributed Convolutional Neural Network (TdCNN), taking as input a sequence of images, has further improved the model accuracy. Multiple data augmentation techniques have been implemented to further extend the image sets during training. Finally, the trained networks have been validated with real images of Bennu. The obtained results show that the network is able to maintain the same accuracy achieved with synthetic images without any degradation.

4.3. Introduction

For decades, small Solar System bodies have been the target of space missions. The low temperature and low gravity environment existing on comets and asteroids preserve its high volatile content, allowing scientists to investigate the origins of the Solar System. The need of taking in-situ measurements, has driven the exploration of various comets and asteroids in the last decades, the Vega 1 and 2 spacecrafts intercepted Comet Halley in March 1986 [18]; Rosetta International Mission which was the first to rendezvous with a comet, 67P/Churyumov-Gerasimenko, and to land on it [20] after performing flybys to asteroids Steins and Lutetia [21]; missions Hayabusa 1 and 2 explored asteroids Itokawa and Ryugu respectively, the later bringing samples from Ryugu back to the Earth in 2020 [23], [24]; and more recently, OSIRIS-REx mission [25] visited asteroid Bennu and collected samples that returned to the Earth in September 2023 before continuing its extended mission across the Solar System, set to visit asteroid Apophis in 2029 [26]. Beyond the scientific interest of asteroids, the threat of a catastrophic impact with the Earth has recently led Planetary Defense efforts to investigate some of these bodies. DART was the first mission devoted to investigating and demonstrating the deflection of asteroids through kinetic impact [63], successfully impacting Dimorphos, the satellite of the binary

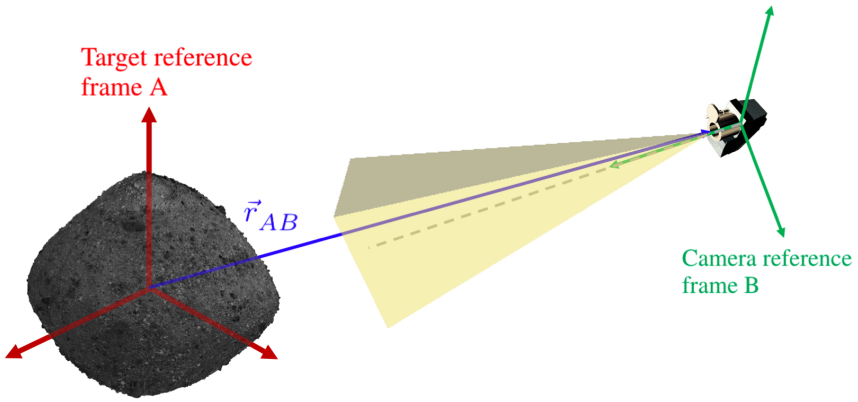


Figure 4.1: Depiction of camera pose estimation with respect to target body fixed frame.

asteroid system Didymos. Later on, the Hera mission to be launched in 2024 carrying two cubesats and a whole set of instruments, will rendezvous Didymos system and navigate about it to investigate the results of the DART impact [31].

In order to operate the spacecraft in the vicinity of such small bodies, a combination of radio navigation ground support and autonomous relative optical navigation involving the use of navigation cameras is generally used [29], [30], [42]. The main challenge of navigating around bodies with very small size and mass is that the ephemeris and physical properties of the target are typically not known with enough accuracy for using standard orbit determination techniques [22]. The small size of these minor bodies means that non-conservative and perturbing forces acting on the spacecraft, including solar radiation pressure (SRP) and spacecraft thermal re-radiation, contribute significantly to the dynamics, being of an equivalent order of magnitude as the target gravitational acceleration [38]. Therefore, navigation autonomy and robustness are paramount for these missions. Due to the lack of accuracy derived from ground observations of these objects, the only solution then is to rely on in-situ measurements for determining with the onboard computer the relative position of the spacecraft with respect to the target. Although other sensors as thermal cameras [39] or LiDARs [40] have also been used for navigation, monocular vision cameras are usually the main sensor used for performing optical relative navigation. These cameras can be used to estimate the relative position of the spacecraft with lower hardware complexity, mass, size, and power requirements.

The state-of-the-art monocular pose determination methods for optical relative navigation in deep space missions rely on Stereophotoclinometry (SPC) algorithms [41] that combine stereo and photoclinometry techniques to form the backbone of the terrain-modeling and landmark-navigation software [42]. A synthetic image is produced using the shape model of the target, and it is cross-correlated with the ground-truth image for obtaining the target shift between both, allowing the individual landmark matching. For the Rosetta mission, automatic landmark tracking was applied using the image database,

the landmark coordinates and the shape model as input parameters, with the major disadvantage that the image had to be downlinked to Earth for manual visual inspection and orbit determination [43]. Moreover, the robustness of this method is strongly dependant on the illumination conditions and its accuracy degrades at very low and very high phase angles (angle between the illumination source vector and spacecraft position vector) [44]. Other missions complemented the navigation cameras with additional payloads, as the Hayabusa and OSIRIS-REx spacecrafts which also counted with LiDAR for spacecraft-to-target range determination and accurate shape modelling [45], [42], or the Hayabusa 2, which utilized retroreflective artificial landmarks carried by the spacecraft and deployed to the surface of asteroid Ryugu, such that they could be tracked by the on-board autonomous navigation system [46]. Recent studies also propose the application of Simultaneous Localization and Mapping (SLAM) techniques using optical sensors to build a map of the environment while navigating with respect to it, enabling the exploration of previously uncharacterized minor bodies [47] or feature-based autonomous approach [48]. However, SLAM methods usually need to be complemented with sensor fusion and incorporate measurements from Accelerometers, Gyroscopes or Star Trackers [121]. In addition, the algorithm requires to keep in memory the landmark estimates, which results in increased time-varying computational complexity.

The usage of Convolutional Neural Networks (CNNs) is spreading in many industries as the main computer vision solution due to its lightweight, precision, robustness, and efficient performance in changing scenarios. Their strengths in pattern recognition and feature extraction have led to significant advancements in tasks like: terrain classification of high-resolution satellite images on Earth [51] and other planets [52]; despeckling of SAR images [53]; on-board image processing for coverage estimation and detection of clouds [54]; and others. Recent work has been done on applying deep learning for feature extraction for terrain relative optical navigation in celestial bodies [49], [50], however these contributions still rely on classical feature matching algorithms to estimate the relative position. Unlike traditional landmark-based techniques, machine learning algorithms could be trained to learn the nonlinear transformation from the 2-D input image space (for grayscale) to the 6-D pose vector space (3 position coordinates and 3 Euler angles). A significant challenge of employing this direct nonlinear transformation lies in the lack of human-interpretability of Neural Networks. Consequently, extensive testing and validation become imperative to identify potential failure scenarios. The estimation of the pose vector can be approached through two methods: discrete or continuous variable estimation. In the discrete variable estimation method, known as multiclass classification, the pose space is discretized and labeled, and a classification problem is solved accordingly [56], [57]. However, it is important to note that the maximum achievable accuracy of this method depends on the resolution of the discretization. Alternatively, the continuous method involves regression, where the Neural Network directly outputs the coordinates of the pose vector [58], [59].

Combining the described discrete and continuous estimation neural networks, a novel

lightweight and robust solution is described in this paper. With higher computational efficiency and exploiting the feature extraction capabilities of CNNs, the proposed method is suitable for autonomous navigation in adverse illumination conditions, when traditional navigation methods accuracy would be degraded due to extended shadows (at high phase angles) or washed-out surface features (at low phase angles). A set of Convolutional Neural Network (CNN) organised in two levels: high-level multiclass-classification and low-level regression, is presented, capable of estimating the relative pose of a camera with respect to a target body. The pose estimation problem is shown in Fig. 4.1, where $\vec{r}_{AB}$ represents the position vector of the camera focal point with respect to the asteroid centered body-fixed frame to be estimated by the CNNs. This approach was first used for the case of comet 67P/Churyumov-Gerasimenko with promising results [122]. State-of-the-art CNN architectures have been investigated for both the classification and regression tasks, testing also modified versions with larger input image dimensions, convolution kernel size and top layers. Moreover, Time Distributed Convolutional Neural Networks (TdCNN) have been introduced to accept as input a sequence of images instead of a single frame at each inference step. These kind of neural networks take advantage of the dynamics of the underlying problem to get contextual temporal information, in this way improving the estimation accuracy. Data augmentation techniques applied during training have been employed in order to generalise the CNNs estimating capabilities as much as possible, accounting for image shift, rotation, and distortions, while maintaining a reasonable accuracy in the pose estimation. In this case, Bennu has been selected as the main target to investigate the applicability of CNNs for pose estimation using data derived from the OSIRIS-REx mission. The OCAMS instrument onboard OSIRIS-REx counted with three cameras with different field-of-views [68], which allows to investigate the performance of the CNN at different altitudes and with different camera configurations. In addition, the multiple spatial resolution shape models and albedo maps available for Bennu can be used to evaluate the impact of the model resolution used to generate the training sets.

As it is well known, training CNNs requires a large amount of data, and even the extensive archive of the OSIRIS-REx mission is not enough to directly train the neural networks for all possible geometric configurations. For this purpose, a Python package named SPyRender, developed in previous works, has been extended to generate large sets of synthetic images suitable for training the CNNs. The geometry of the target and observer, illumination source, camera model, and target shape and albedo map can be configured and modified during runtime when rendering the scene, allowing for the efficient production of different combinations of geometric and target conditions. These sets of random combinations of the elements defining the scene together with additional data augmentation methods, are the key to maintain the CNN accuracy when providing for inference, input real images which are not part of the training sets, composed only by synthetic images.

The rest of this paper is organized as follows: Section 2 describes the methodology

used for the synthetic image sets generation; Section 3 explains the CNNs architecture and training methods; Section 4 presents the results of the trained networks and its application to pose estimation; and Section 5 summarises the conclusions from the current study and the basis for further work and developments.

4.4. Methods for Generating Synthetic Images

In order to have suitable training sets and aiming to study the impact of the different geometric configurations of the scenario on the CNN accuracy, multiple synthetic image data sets have been generated. For this purpose, the Python package SPyRender [122] has been extended to accept variable albedo maps and camera field-of-views. SPyRender was developed in previous works and is devoted to the systematic generation of synthetic images focused on the analysis of space-borne instrument observations. The graphic capabilities of SPyRender are based on Pyrender [70], a pure Python library for physically-based rendering and visualization implementing a GPU-accelerated offscreen renderer. Different image effects variations, either user-defined or random, are implemented via configuration, including: illumination source position, type and intensity, camera and target position and rotation, camera field-of-view aperture angles, image resolution, and textures of the target model.

For the study case of asteroid Bennu, multiple 3D models are available at the OSIRIS-REx SPICE archive PDS4 collection according to model coverage, production technique and spatial resolution [123]. In this work, two shape models have been used to evaluate the effect of the spatial resolution of the input model and scale of the surface features in the CNN accuracy. Fig. 4.2 shows the difference between different 3D models used for generating the synthetic images. From left to right, the preliminary model based on PolyCam images during approach phase at a spatial resolution of 6 meters, the more detailed model after close proximity operations with a spatial resolution of 880mm, and the same detailed model but adding albedo map to the shape model. The texture used for the albedo map is based on the normalized mosaic derived from the PolyCam images at low phase angles (less than 8 degrees) with a spatial resolution of approximately 60mm and available at the USGS [124]. Because this normalized map does not have coverage out of the ±55 degrees latitude range, the texture map has been completed with the normalized global mosaic produced with images at phase angles up to 30 degrees. The resulting merged albedo map is shown in Fig. 4.3. The comparison between the synthetic and real images is illustrated in Fig. 4.4, displaying some examples of real MapCam and PolyCam images (top row) compared to the corresponding synthetic ones generated with SPyRender (second row). The third row depicts the grayscale pixel intensity histogram for each pair of real and synthetic images, showing the similarity between both. The very rocky surface of Bennu, populated by an unexpected large density of boulders [125] results on a quite variable albedo distribution, with numerous light gray and black spots. This implies that the topography is not enough to accurately simulate the target, to complement

Figure 4.2: Comparison of different Bennu models at different spatial resolutions.

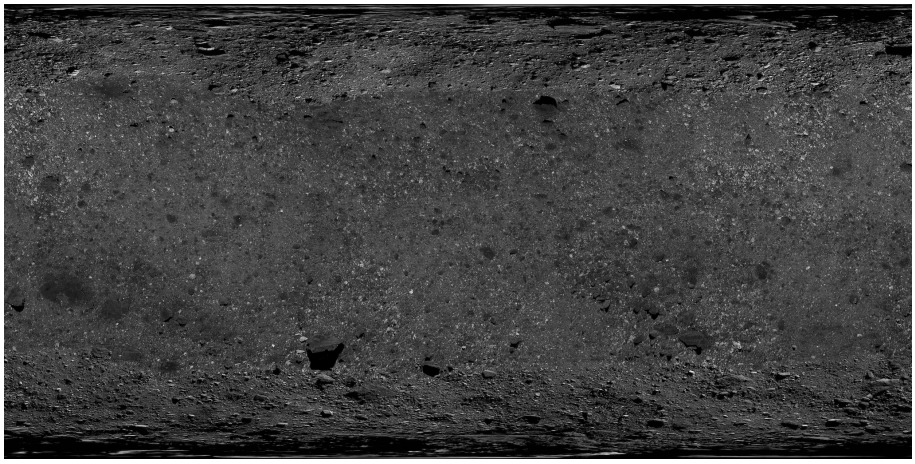


Figure 4.3: Albedo map for the surface of asteroid Bennu.

it, the albedo map is required. This can be appreciated with the last image in Fig. 4.4. At very low phase angles, shadows cast by the boulders disappear and the image is mainly dominated by the surface albedo.

For rendering the images, the aperture of the field-of-view (FoV) of the camera is defined in terms of reference and cross angles for a rectangular shape following the SPICE format. These angles have been taken from the OCAMS Instrument Kernel [126]. For MapCam, an angle of 3.97 degrees is used for a squared FoV, while for SamCam, a larger angle of 20.44 degrees is used. This difference in the FoV aperture allows to produce images with the full asteroid in the camera FoV when being far from the asteroid (using MapCam) or when closer to its surface (using SamCam), enabling to train and evaluate the capability of the CNN to achieve field-of-view invariance. Such a large angular difference in the field-of-view angles results on distorted or deformed shapes when comparing images with the same point of view captured by each camera. Therefore, training with multiple FoV sizes is critical for the CNN to interpret these variations associated to the camera intrinsic rather than the pose to be estimated. Regarding the resolution in terms of pixel lines and pixels samples of png images output by the offscreen renderer, 224x224 and 480x480 have been used. 224x224 is the standard resolution used in computer-vision CNN but in recent works, the usage of larger resolutions has shown to improve the accuracy of the CNN depending on the use case [127]. Because very high resolution target

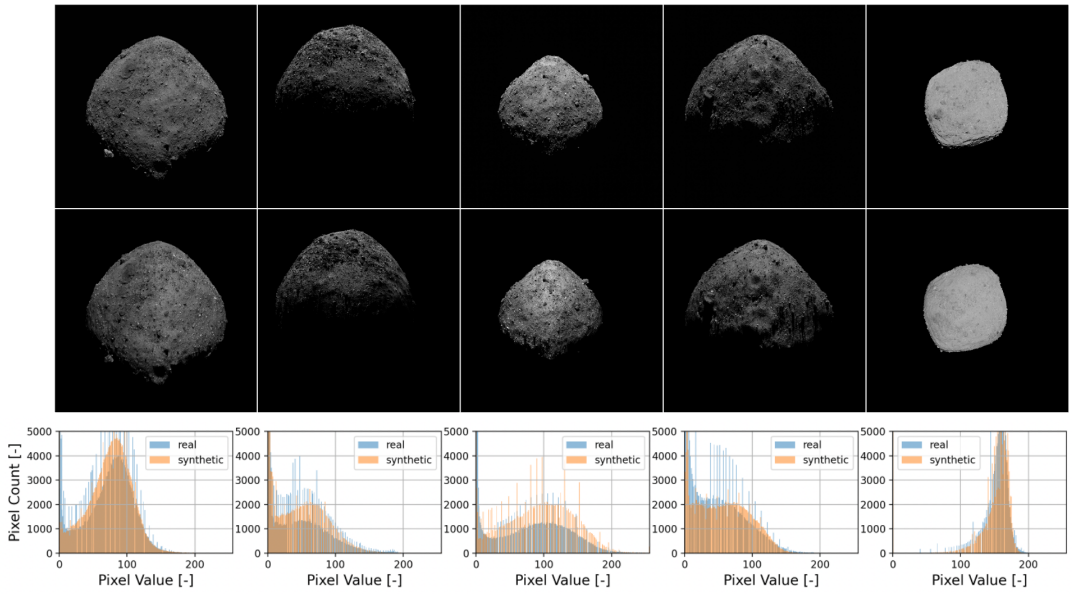


Figure 4.4: Comparison of real OCAMS images (top row) vs synthetic ones (second row). The grayscale intensity histogram for both images is displayed below.

models are available in this case, it is expected that increasing the image resolution could indeed improve the CNN accuracy.

Two primary types of image datasets have been produced to assess the performance of CNNs in classification and regression tasks. The first set comprises images captured from various camera positions encircling the target body. These datasets primarily focus on training CNNs for global position estimation. Conversely, the second set consists of regional datasets containing images from different local sectors of the target body. In order to evaluate the impact on the sector size for training the CNNs, two levels of discretization of the surrounding space have been applied: 16 sectors each spanning 90 degrees longitude and 45 degrees latitude, and 32 sectors each spanning 45 degrees longitude and 45 degrees latitude. For each sector, dedicated training and testing datasets have been prepared to analyse the training of CNNs capable of estimating camera position and orientation with increased precision.

Data augmentation techniques are applied to the data sets, either during rendering or directly during the training process, in order to extend the features of the images used for training, seeking rotational invariance, translational invariance and noise invariance. In addition to these basic modifications, a type of cutout erase technique has been applied. This method randomly removes up to half of an image from the side to the center of the image for either one or two continuous sides. This allows the CNN to interpret images in which a large part of the target, up to a 75%, was outside of the FoV. The effect of applying these data augmentation techniques altogether on a single image can be appreciated in Fig. 4.5. Variable camera roll, pitch and yaw angles can be achieved during training by rotating and applying a shift to the images (as deviating from nadir pointing), however

Table 4.1: Description of the global synthetic image datasets generated for this work for each field-of-view and image resolution configuration

Dataset	Description	Samples
simTrainBE_sr	Centered, No boresight rotation, Fixed brightness	200000
simTrainBE_sr_rr	Centered, boresight rotation [0, 360], Fixed brightness	200000
simTrainBE_sr_br	Centered, No boresight rotation, brightness [-98%, 260%]	200000
simTrainBE_sr_rr_br	Centered, boresight rotation [0, 360], brightness [-98%, 260%]	200000
simTrainBE_sr_rr_br_o10	Shift, boresight rotation [0, 360], brightness [-98%, 260%]	200000
simTrainBE_sr_rr_br_o10_ng	Shift, boresight rotation [0, 360], brightness [-98%, 260%], Gaussian noise	200000
simTrainBE_sr_rr_br_o10_se_ng	Shift, boresight rotation [0, 360], brightness [-98%, 260%], shift eraser 75%, Gaussian noise	200000
simTrainBE_seq4_sr_rr_br_o10_se_ng	Sequences of 4 frames on a trajectory arc, same geometric configuration as previous dataset	200000

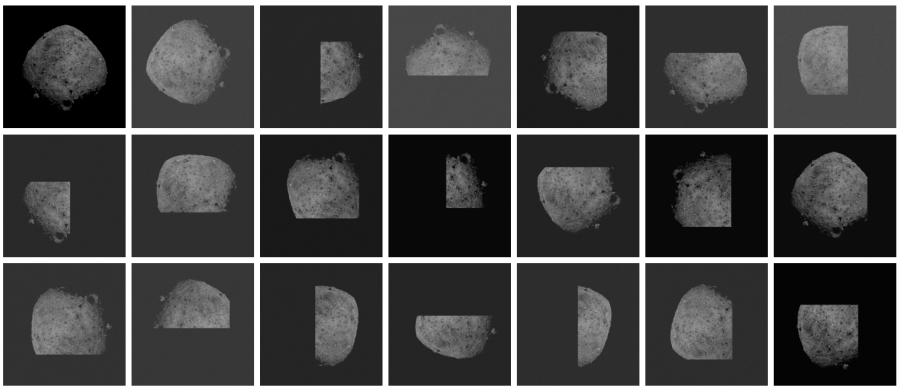


Figure 4.5: Example of data augmentation combined effects on a single image. The original image is shown in the top-left corner.

illumination conditions have to be generalised during rendering when generating the data sets. Illumination conditions play a key role in computer vision and optical navigation, so it should be properly configured to achieve illumination invariance in terms of intensity and direction. It is worth noting that depending on the mission, the illumination conditions could be constrained to a given range, therein simplifying the range of illumination variation. For instance, the spin axis of Bennu as well as its heliocentric orbital angular momentum are quasi-perpendicular to the ecliptic plane, as a consequence the Sun declination with respect to Bennu body-fixed frame XY plane is always close to zero. However, for the sake of assessing a generic solution, this fact is not taken into account in this work.

The produced image data sets intended for global pose estimation are listed in Table 4.1, including the main features of the data sets. Each data set is composed by 200000

images, and its corresponding label files, for which 80% correspond to train set and 20% to test set. A sensitivity analysis has been carried out evaluating the impact of the training set size on the CNN training process and performance. First results suggested that the initial 40000 images used in previous works were not enough and the training process was over-fitting. Therefore, the number of images has been progressively increased to obtain the optimal size of the training set. The first and most simple set "simTrain224BE_sr" covers the whole range of camera positions around the target, with attitude fixed to Nadir pointing, meaning the target is centered in the images and there is no image rotation around the boresight direction (pixel lines aligned with asteroid North). Also, random illumination source direction has been introduced during image rendering. For the rest of the image sets, the different combinations of random around boresight rotation (roll angle), target off-nadir shift, illumination intensity, Gaussian noise, and shift cutout erase, have been applied. The same geometric conditions have been used to create the equivalent sets but using variable field-of-views associated to MapCam, SamCam and 2xSamCam (an artificial FoV double the size of SamCam), and altitude ranges: 6 to 20 kilometres for MapCam, 1 to 3 kilometres for SamCam, and 500 to 1000 metres for 2xSamCam. Similarly, two sets have been created for each of the previous combinations, at 224x224 and 480x480 input image resolution. In addition, a set of real MapCam and SamCam images obtained from the OSIRIS-REx archive has been compiled to validate the trained models and compare the performance when testing real images instead of synthetic ones.

4.5. Convolutional Neural Network Architectures and Training

The main objective of this work is to train with synthetic images, CNNs which are capable of estimating the pose of the camera in the vicinity of the asteroid. Moreover, the trained models have to retain its accuracy when real images are provided as input, being robust to variable illumination conditions and geometric scenarios. Therefor, the adequacy of multiple architectures and training configurations will be evaluated for this use case. As a starting point to analyse the impact of the different geometric and image combinations on the CNN performance, a simple architecture based on AlexNet was selected [80]. This baseline architecture has been proven to perform adequately in similar neural-network based applications like noncooperative spacecraft rendezvous [81] or asteroid centroiding for autonomous attitude navigation [82]. The baseline architecture consists of just two convolutional layers followed by three fully-connected layers, the last one being the output layer. With such a simple architecture, it is easier to achieve convergence during training and evaluate the relation between the architecture hyperparameters and the performance for each study case. Once the baseline architecture CNN has been trained against the multiple image sets presented in Table 4.1 and the trends in the training process have been identified, it was decided to test modern state-of-the-art architectures for the most relevant cases. These architectures often require longer training times, more memory for the same batch sizes due to the larger number of coefficients, and in some

cases convergence may not be reached due to gradient instabilities, however, they shown a great improvement in accuracy for other computer vision tasks. In this work, VGG-19 [128], ResNet50 [84], and DenseNet121 [87] architectures have been tested for some of the training sets seeking improved performance in the position estimation compared to the baseline architecture. At the expense of a larger size, the VGG family is characterised by a deeper architecture, 19 layers for VGG-19 (16 convolution layers plus 3 fully connected layers) compared to the 8 layers of AlexNet. With a homogeneous architecture and smaller kernel size, by increasing the depth, the network can capture better the non-linearities in the underlying problem. ResNets and DenseNets exploited even further the depth of the CNNs reaching 50 layers for ResNet50, and 121 layers for DenseNet121. By adding residual blocks from the preceding layer (ResNet) and dense connections from all preceding layers (DenseNet), the vanishing gradient problem is tackled and a much deeper architecture can be successfully trained. On the other hand, seeking for a lightweight oriented approach, MobileNet [119] family of CNNs has also been tested. In addition, three modified versions of VGG-19 have been analysed: first keeping just a single reduced fully connected layer before the output layer, second adding an extra fully connected layer before the output layer, and third increasing the kernel size of all convolutions from 3x3 to 5x5, hence increasing the number of coefficients and size of the CNN. These modifications are foreseen to substantially increase training times due to the large amount of parameters to tune, but specially a larger kernel size, could potentially improve the feature extraction capabilities essential for the pose estimation.

In order to investigate its effectiveness solving the global and local pose estimation problems, two types of CNNs (independent of the core architecture) have been produced differing on the last layer. The first type are Multiclass-classification CNNs, for which the last fully-connected layer has the same dimension as sector labels in which the space has been discretized. Two levels of discretization for the classification problem have been evaluated: 16 sectors each spanning 90 degrees longitude and 45 degrees latitude, and 32 sectors each spanning 45 degrees longitude and 45 degrees latitude. The Softmax [92] activation function is applied to the last layer as it normalizes the last layer output vector into a probability distribution over sector labels, meaning the element in the function output with the highest probability represents the estimated sector. In (4.1), z_i refers to the i-th element of the vector provided as output by the last fully-connected layer, and L_i are the elements of the resulting probability distribution.

$$L_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{4.1}$$

The second type are Regression CNNs, for which the last fully-connected layer implements the linear activation function providing directly as output the target variable values to be estimated. Therefore, the output dimensions are configured as follows: three dimensions for estimating camera position vector in Cartesian coordinates; two dimensions for yaw and pitch angles (off-nadir) as shift in pixel lines and pixel samples; and two

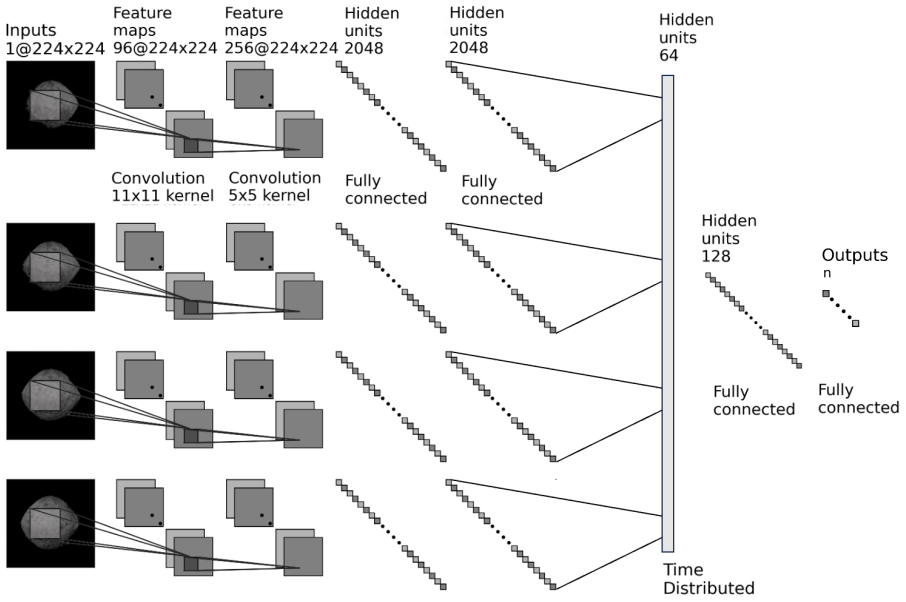


Figure 4.6: Time Distributed Convolutional Neural Network (TdCNN) architecture diagram.

dimensions for the roll angle around the camera boresight. The roll angle is estimated by decomposing into the sine and cosine (therein the two dimensions) instead of the direct angle (one dimension). While the roll angle variable by itself suffers of a discontinuity between 0 and 360 degrees causing large instabilities in the error back-propagation, the corresponding sine and cosine variables are continuous within the range -1 to +1. This yields better accuracy estimating the roll angle as the 2-argument arctangent of the network output. The decision to have three different Regression CNNs for estimating position vector, pitch and yaw angles, and roll angle is supported by the difference in scale between the output variables and the impact this has on the weights back-propagation and estimation accuracy. A spread range of values in the target variables causes weights to abruptly change, introducing instabilities in the training process [93]. It is worth noting that a multi-branch model [97] could be used to address the scale disparity between the different target variables. A single architecture with common convolution layers followed by multiple downstream branches, each providing a different variable as output and with different loss functions, would alleviate the total size of the CNN and improve inference efficiency. However, it was decided to implement independent sequential networks, seeking for modularity and faster training, allowing to investigate different image effects independently on each target variable.

4.5.1. Time-Distributed Neural Networks

Recent computer vision algorithms for autonomous navigation implementing Neural Networks use as input sequences of frames instead of a single image. This approach enables

the CNN to learn the dynamic behaviour of the specific use case, improving the accuracy of the estimation by using accumulated input data, and even opening the door to directly estimating time derived quantities like linear and angular velocities. Time-Distributed Neural Networks or TdCNNs are one type of CNNs which allows the ingestion of sequences of images by adding a Time-distributed layer like Gate Recurrent Units (GRU) in the architecture. In this model, each frame of the input image sequence is provided to the base CNN (note that in this case the same weights are used for the CNN applied to each image of the sequence), the output of the CNN for each image is then combined in the GRU layer. Finally, a decision network consisting on several fully-connected layers is stacked on top of the GRU to provide the final output of the neural network, either space sector for multiclass classification or pose vector for regression. The global architecture for the TdCNN is depicted in Fig. 4.6, using the baseline CNN (removing the former output layer) before the GRU (Time Distributed) layer. In order to alleviate the model size, optimized CNN architectures like MobileNet can be plugged-in as well for the CNN section before the GRU layer. With the objective of testing this approach for the relative pose estimation, a dedicated training set consisting of sequences of 4 images has been produced. For each sequence, the geometry of the first frame is computed as for the base training sets in Table 4.1, and then a random perturbation of the latitude, longitude, altitude and Euler angles is iterated to produce the remaining images in the sequence. This means that in order to get training sets of the same size as for the base CNNs, the training sets for the TdCNN are four times larger, making the training process substantially more computationally intensive.

4.5.2. Training Hyperparameters Selection

To obtain optimal performance for the designed architecture and to streamline training times, certain hyperparameters governing the training process must be carefully defined. The most relevant are: the Loss function, the optimization algorithm, the training epochs, and the learning rate. The error or loss function, is used to estimate the loss of the model at the current iteration of the optimization algorithm, such that the weights are updated accordingly to reduce the defined loss at the next iteration. The chosen loss function depends on the neural network to be trained and the predictive problem for which it will be applied. For Multiclass-classification CNNs, the Sparse Categorical Cross-Entropy loss function has been selected as it has been proven competitive in most domains and the preferred default option [94]. The expression for the Categorical Cross-Entropy is shown in (4.2), where y_i represents the target value, and y'_i represents the i-th element in the model output.

$$L = -\sum y_i \log(y'_i) \tag{4.2}$$

For the Regression CNNs, commonly used loss functions include the Mean Squared

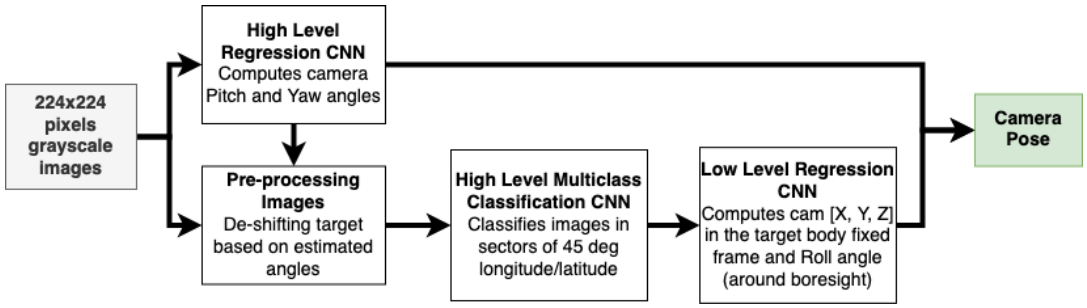


Figure 4.7: Two levels neural network flowchart.

Error (MSE) and the Mean Absolute Error (MAE), both suitable for variables represented by zero mean Gaussian distributions. Nevertheless, a customized loss function has been formulated to encompass the Mean Translation Error (MTE) between the ground truth and the estimated vector for a specific image. While employing MAE or MSE for position estimation could lead to minimal error in some of the coordinates of the output vector, leaving the remaining coordinates with substantial deviations, the utilization of translation error focuses on minimizing the overall magnitude of the position error vector. Since for the Euler angles both CNN outputs, off-nadir vertical and horizontal pixel shifts (equivalent to pitch and yaw angles), and roll angle decomposed into sine and cosine, are 2-dimensional vectors, the MTE will also be used as the loss function to minimise the magnitude of the error vector. Equations for MAE, MSE, and MTE are shown in (4.3), (4.4), and (4.5) for n samples, where y_i represents the ground truth and y'_i denotes the predicted value. Note that while y_i and y'_i are scalar variables, $\vec{y}_i$ and $\vec{y'}_i$ are vector variables, meaning that for a batch of n samples of the output vector variable, the MTE will be the mean of the magnitude of the error vector between the ground truth $\vec{y}_i$ and the predicted value $\vec{y'}_i$ for each sample.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - y'_i| \tag{4.3}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y'_i)^2 \tag{4.4}$$

$$\text{MTE} = \frac{1}{n}\sum_{i=1}^{n}|\vec{y}_i - \vec{y'}_i| \tag{4.5}$$

For the optimizer, SGD (Stochastic Gradient Descent with momentum) has been selected as optimization algorithm in charge of minimizing the loss function. While it may have slower convergence compared to Adam (Adaptive Moment), a better converged result has been achieved with SGD. The only drawback of SGD compared to Adam is that before training, both the input (images) and output (pose vector) variables have to be

scaled to the 0 to 1 range. The training epochs (number of passes through the whole train set) have been set to ensure that validation loss does not change if epochs are increased further. A scheduler has also been applied to reduce the learning rate by a factor of 10 when there is no improvement in the validation loss for the last 20 epochs. The learning rate controls how much the weights are updated at each training epoch based on the loss at that epoch. A large learning rate could result in training instabilities while a value too small could require excessively long training times. By starting with a learning rate of 0.01 that is reduced when loss converges about a minimum, the optimizer fine tunes the weights to get closer to the minimum, achieving better accuracy.

4.5.3. Hybrid Neural Network Solution

The main drawback of using Multiclass-classification CNNs for position estimation is that the maximum accuracy of the estimated output directly depends on the number of sectors or labels in which the 3D space has been discretized. Nevertheless, as the number of sectors increases, so does the instabilities in the training process and the accuracy in the sector estimation. The main reason for this is that Classification problems take the different possible values of the output variable as independent discrete values without accounting for ordering or underlying continuous relations between them. In general, continuous variables such as the camera position or the Euler angles are better estimated by regression CNNs. However, after trying to train for a global position regression solution, the optimizer was not able to successfully minimize the loss function, most likely due to the large non-linearities in the underlying transformation for the input image to the 6D pose vector space. This lead to the hybrid two-levels architecture, consisting of a High-level Multiclass-classification CNN in charge of estimating the local region or sector of the 3D space, and a set of Low-level Regression CNNs, each trained for one specific sector and capable of accurately computing camera position in Cartesian coordinates and camera angles. This approach took advantage of the strengths of both methods interconnecting both types of CNNs. Fig. 4.7 depicts the flow of this two-levels global approach for the pose estimation. In addition, a pre-processing step was added to perform a de-shifting operation to the input image centering the target in the image. This step is applied to improve the accuracy of the pose estimation, which in previous works was found to be strongly impacted by the position of the target centroid with respect to the center of the image [122].

4.6. Results

This section describes the experiments conducted to evaluate the training and performance of the investigated CNNs conforming the multiple blocks of the two levels approach described previously: High-Level Regression; High-Level Multiclass classification; and Low-Level Regression. For each type of CNN, the following aspects of the

training process were assessed: the loss function selection and optimal number of training epochs required to achieve convergence, the monitoring of training metrics (including accuracy and F1-score for the multi-class classification network and loss function for the regression networks), the behavior of training and validation estimates and the effects of regularization, the impact of image effects and data augmentation on the CNN training and estimation performance, and finally the validation of the CNNs with real OCAMS images.

4.6.1. High-Level Regression

The pitch and yaw angles of the camera reference frame are computed by the High-level Shift Regression CNN by estimating the target centroid displacement in pixels with respect to the center of the image. Knowing the model of the camera field-of-view, the pixels are directly converted into angles. This pixel shift is also used to add a de-shifting preprocessing and center the target in the image that will be provided to the pose estimation CNNs. Therefore, the parameter to be estimated by this CNN results in a 2-components vector containing vertical and horizontal shift in pixels, meaning the last fully-connected layer of this CNN has dimension 2. In order to train this CNN, starting from nadir pointing images, a random target shift up to 120 pixels (slightly more than half of the 224x224 pixels image) has been introduced in order to produce the un-centered images. Data augmentation techniques were used when producing this image set in order to improve the generalization of the shift estimation. These are, random image rotation, random illumination intensity, Gaussian noise, variable field-of-view, and variable asteroid shape models. The advantage of using a CNN for estimating the shift, is that it can be trained to rely on asteroid features independent of shadow length (when illumination direction changes) and independent to small scale changes between multiple asteroid shape models (or due to activity of the surface), therein estimating the actual geometrical center of the target in the image instead of the illumination center.

For this de-shifting CNN, the Mean Translation Error (MTE) defined in (4.5) has been used as the loss function to be minimized during training. In Fig. 4.8 the test (dashed lines) and validation (solid lines) MTE evolution during training is shown for: the baseline CNN using as loss function MAE (black line), the baseline CNN using MTE (blue line), and for MobileNetV2 using MTE as loss function (green line). It can be observed that using MTE instead of MAE as loss function, not only the magnitude of the error in pixels is reduced by more than half, but also the number of epochs for convergence is substantially reduced. Moreover, when using MobileNetV2 architecture, although the training is more unstable at the first epochs, the converged loss after the learning rate reduction (around epoch 80) is just 2 pixels (half the one achieved with the baseline CNN). For a 224x224 pixels image with a field-of-view of 20.44 degrees as SamCam, this pixel error is equivalent to an approximate pitch/yaw angle error of 0.1825 degrees, while for MapCam with 3.97 degrees it corresponds to an error of just 0.035 degrees. The high-

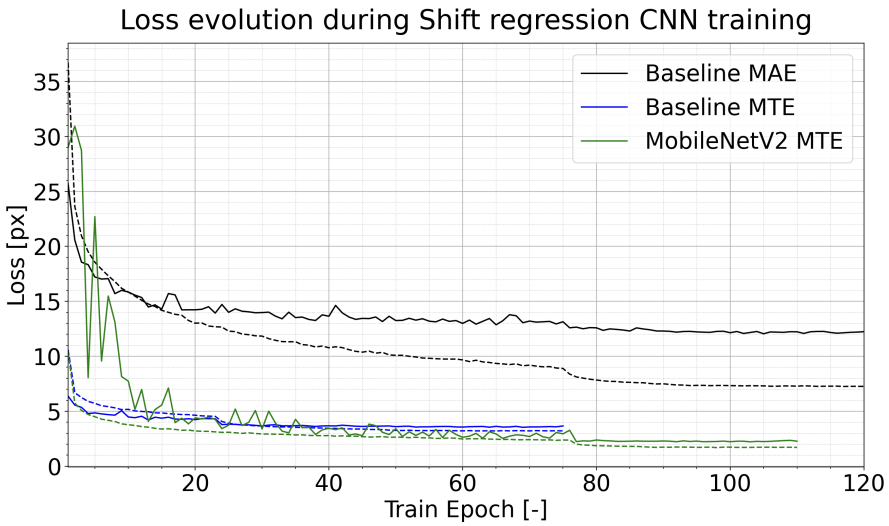


Figure 4.8: Train and Loss evolution for pixel shift regression.

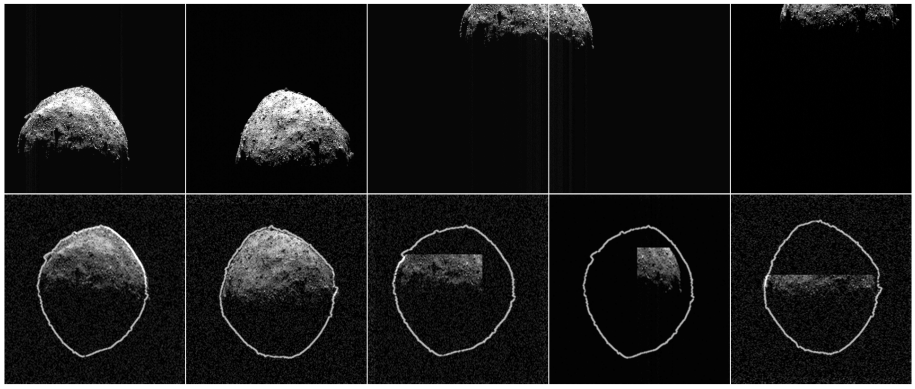


Figure 4.9: Results of de-shifting real OCAMS MapCam images. The asteroid outline for zero shift from image center is overlaid.

level regression CNN was trained with synthetic SamCam images, but in order to evaluate the field-of-view invariance, it has been tested with real images from MapCam captured during the preliminary survey and consecutive orbit phases. For most of these images, the camera boresight was not aligned with the nadir direction, so the asteroid is in general not centered in the image. Moreover, the combined effect of illumination conditions and nadir off-pointing results in extreme cases with just a small illuminated part of the asteroid being visible but for which the CNN can still accurately estimate the centroid shift. Fig. 4.9 shows some examples of real images obtained from the OCAMS PDS Bundle for the specified time period, compared with the corresponding de-shifted images based on the CNN estimated shift. The outline for the ground-truth of the asteroid centroid de-shift is overlaid for a clear comparison.

4.6.2. High-Level Multiclass classification

The first block of the pose estimation consists of the high-level multiclass-classification CNN in charge of estimating the current sector of the discretized 3D space. In this case, geometric variations such as target shift or around boresight rotation have a substantial impact on the classification CNN accuracy. For the classification problem, the performance of the CNN is measured in terms of accuracy of predictions and the F1-score of classifications. The accuracy is defined as the percentage of correct sector estimations over the size of the image set. The F1-score is the harmonic mean of two quantities, the precision and recall. These are related to the number of true positives (TP), false positives (FP), and false negatives (FN) over the image set. Because these metrics are devoted to binary classification problems, a positive sample is defined when it belongs to the correct sector, and negative for all the other sectors.

$$precision = \frac{TP}{TP + FP} \tag{4.6}$$

$$recall = \frac{TP}{TP + FN} \tag{4.7}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.8}$$

Training with the image set simTrainBE_sr_rr_br_o10_se_ng the achievable accuracy with multiple CNN architectures, level of sector discretization, and size of the training set has been evaluated. Fig. 4.10 shows the evolution of the estimation accuracy during training for the baseline CNN presented in previous section (blue line), the MobileNetV2 architecture (red line), and the Time Distributed CNN (orange line) using MobileNetV2 for the CNN block before the GRU layer. Using a state-of-the-art architecture like MobileNet results in a large boost in the performance compared to the baseline architecture, increasing accuracy from 50% to 88%, yet having an extremely lightweight and efficiency oriented model. Moreover, when implementing the Time Distributed CNN, the accuracy is further improved up to 95%, showcasing the benefits of this type of architecture capable of analysing sequences of images. This is specially relevant for extreme illumination conditions, for example when one image of the sequence has a very large phase angle and therefore, not enough landmarks are visible for the pose estimation. However, other image of the sequence could have more suitable illumination conditions, leading to a correct sector estimation for the overall sequence. It is worth noting that to avoid instabilities in the training of the TdCNN, transfer learning is applied to initialize the weights of the CNN block using the previously trained MobileNetV2 architecture. Since the feature extraction of the convolutions is the same for the TdCNN, applying transfer learning alleviates the optimization of most layers in the network and reduces convergence time. To summarise

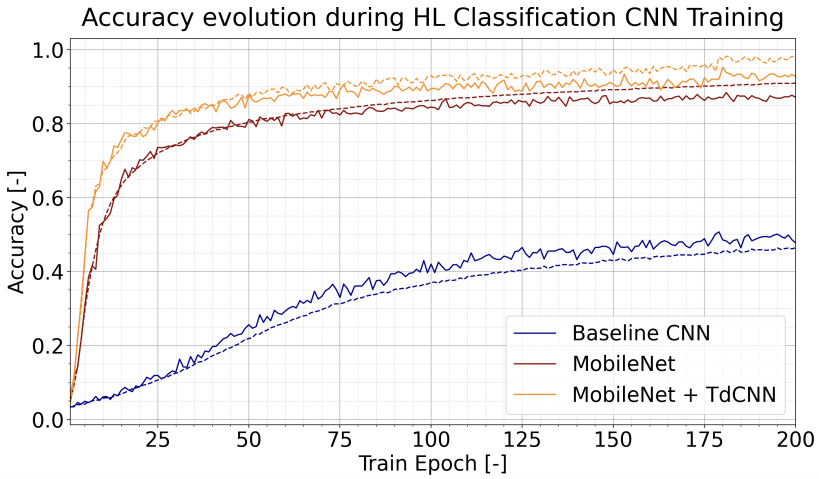


Figure 4.10: Train and Test Accuracy evolution for simple CNN and TdCNN for Multiclass-classification.

Table 4.2: Precision, recall, and F1-score for the High Level Multi-class Classification CNNs trained for this work

Case	**Precision [-]**	**Recall [-]**	**F1-score [-]**
Baseline	0.5650	0.4987	0.5023
MobileNet	0.8861	0.8844	0.8842
TdCNN	0.9527	0.9524	0.9522

the performance of the three tested CNN architectures, the precission, recall, and F1-score for each of them is shown in Table 4.2.

Regarding the size of the training sets and the discretization of the 3D space for classification, as it can be observed in Fig. 4.11, as the number of images for training increases, the difference in achievable accuracy for 16 or 32 sectors is reduced. For both levels of discretization, as expected, the overfitting is substantially reduced when increasing the number of images, as the CNN is exposed to more realisations of the generic scenario. For 16 sectors discretization, the improvement in validation accuracy with an enlarged training set is smaller since there are less classes (labels) to be homogeneously represented in the training set. Hence, as the number of sectors is doubled, having a sufficiently large training set has a strong impact on accuracy, suggesting that a further increase of the training set could still slightly improve the validation accuracy beyond 95%.

Finally, the relation between the CNN performance and the illumination conditions has been investigated. In Fig. 4.12, the histogram of percentage of correct estimations for each value of the Sun phase angle is shown. For intermediate values of the phase angle, the accuracy is close to 100% but for adverse conditions, which for OSIRIS-REx mission were considered to be below 20 degrees (images fully dominated by albedo) or above 70 degrees (images dominated by very long shadows) [40], the accuracy of the CNN is

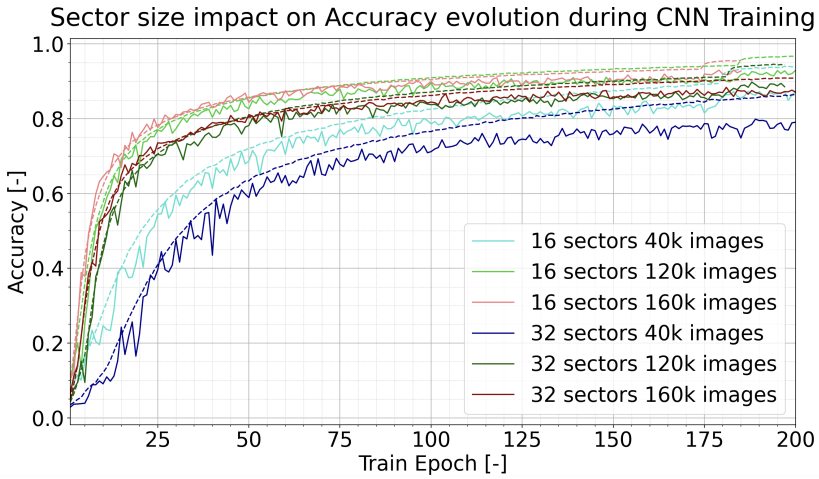


Figure 4.11: Train and Test Accuracy evolution for MobileNetV2 trained with different sector discretization and size of training set for Multiclass-classification.

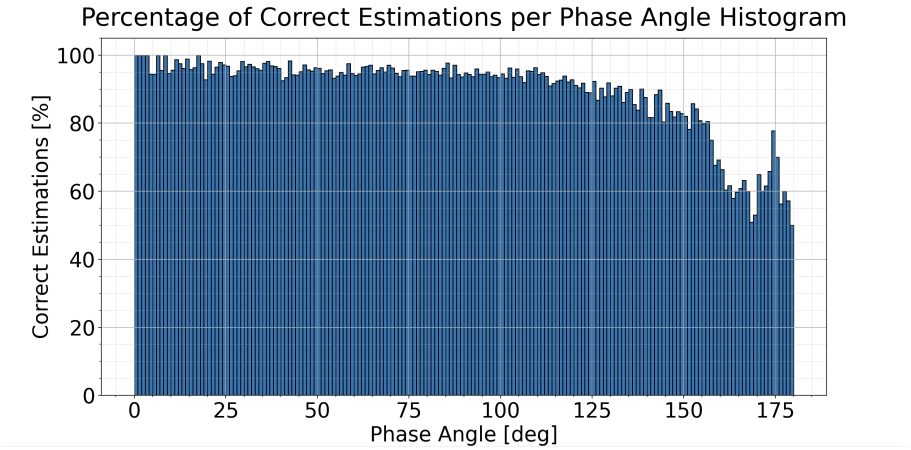


Figure 4.12: Histogram of percentage of correct estimations per Sun Phase Angle.

also maintained. For very low phase angles, the addition of the target albedo maps in the generation of the training sets has enabled the CNN to learn the surface appearance in the absence of shadows, achieving almost 100% accuracy when traditional landmark-based methods could fail. On the other hand, at high phase angles, the accuracy of the CNN degrades only for values above 125 degrees, when most of the target is already in shadow.

4.6.3. Low-Level Regression

Each low-level regression CNN has been trained for one specific sector of the 3D space, slightly extending the sector regions to overlap each other by a 20%. This is done to handle wrong estimations of the high level classification block, most common at the boundaries between sectors [122]. These CNNs are capable of estimating camera position vector in Cartesian coordinates, and roll angle (around camera boresight rotation) in terms of its

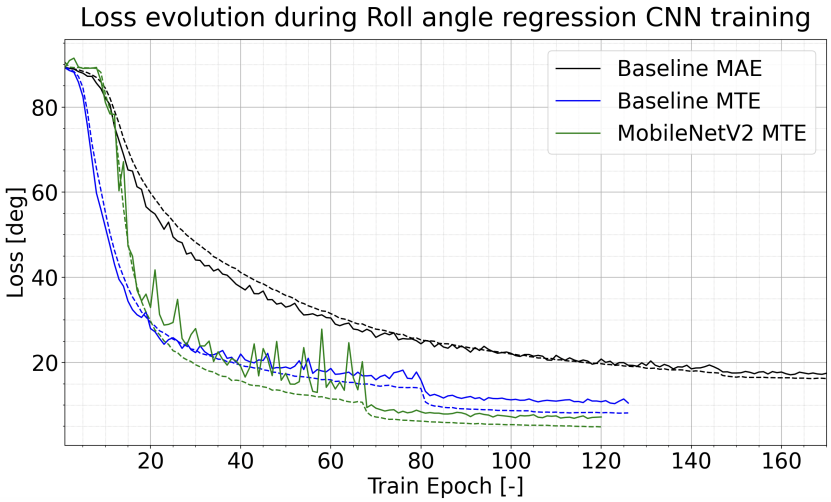


Figure 4.13: Train and Loss evolution for Boresight roll angle regression.

sine and cosine. Note that for each sector, two independent sequential networks have been trained, one for position estimation having last fully-connected layer of dimension 3, and other for roll angle sine and cosine estimation with dimension 2. Same as for the High-level shift, the MTE (4.5) has been used as the loss function. For the position estimation, the MTE minimises the magnitude of the error vector. Moreover, for the sine and cosine of the roll angle, the CNN provides a unit vector of 2 dimensions as output, therefore in this case, the MTE also minimises the magnitude of the error vector, which consequently minimises the error in the 2-argument arctangent of the CNN output. For the roll angle estimation, the same effects used for the High-level multiclass classification have been used, adding up all possible combinations of random shift up to 10 pixels, boresight rotation up to 360 degrees, brightness variations from -98% to 260%, Gaussian noise, and cutout eraser. Combining altogether these effects allows the CNN to estimate the roll angle for images which have been previously de-shifted by the high-level block, even when a large part of the target was outside of the camera field-of-view. For easier visualization, instead of the MTE, Fig. 4.13 shows the evolution of the roll angle error computed as the 2-argument arctangent of the CNN output at each training epoch for different architectures and loss functions. Starting with the baseline CNN architecture and using MAE as the loss function, the training converges in approximately 160 epochs to a roll angle error of 17 degrees. When using the MTE as loss function to minimise the magnitude of the output error vector, the training converges earlier around epoch 120. There is now some overfitting, but the validation roll angle error has been effectively reduced to just 10 degrees. Finally, using MobileNetV2 architecture results in a further decrease of the converged roll angle error to 7 degrees. It can be observed that when using MTE instead of MAE, the loss evolution is noisier at the first stages of the training. However, thanks to the scheduling of the learning rate reduction, the optimizer get closer to the loss minima, stabilising the training and notably reducing the achieved loss.

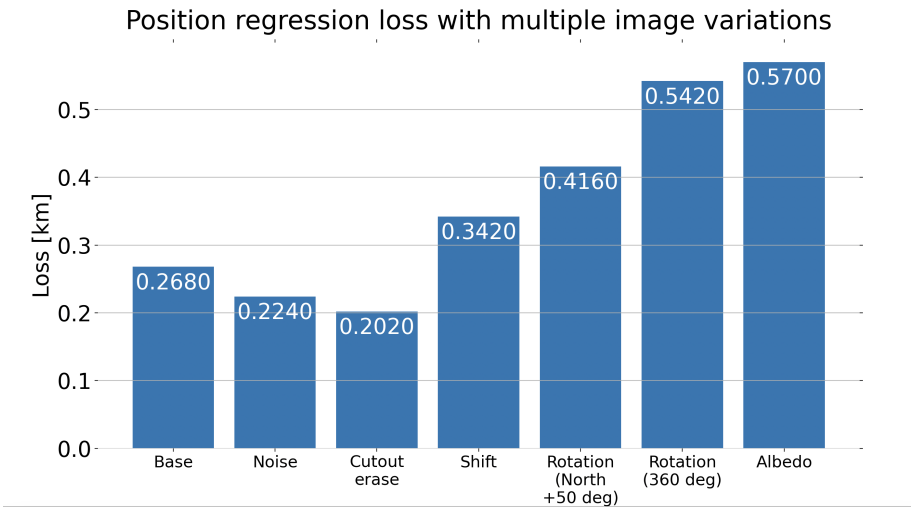


Figure 4.14: Effects on position estimation loss of multiple image and geometric conditions.

Table 4.3: CNN architectures achieved loss and size comparison

Architecture	**Loss [km]**	**Size [MB]**
DenseNet121	0.627	67
Baseline CNN	1.023	120
VGG-19 (reduced)	0.570	160
ResNet50	0.595	196
VGG-19 (base)	0.443	1065
VGG-19 (kernel 5x5)	0.389	1331

For the position estimation, multiple CNN architectures have been tested as summarised in Table 4.3, with VGG-19 outperforming the others. Compared to the high level classification, the low level position regression relies on learning all the minor scale surface features distributions visible in the images. Hence, a larger number of trainable coefficients, results in this case in VGG-19 achieving better accuracy than lightweight architectures with less parameters. In Fig. 4.14 the final value for the loss after training with the different image and geometric effects are summarised for the altitude range 5 to 20 kilometers with the MapCam field-of-view. As it can be observed, superimposing multiple geometric effects as noise and cutout erase have similar resulting loss, being slightly improved due to the regularization introduced by the data augmentation. The off-nadir shift yields a substantial increase of the loss, however with the high-level de-shifting CNN, this effect is avoided in the low level regression. The introduction of the roll angle around the boresight direction has the largest impact on the loss, being close to 0.54 kilometers for a fully random roll angle. Therefore, if the attitude of the spacecraft could

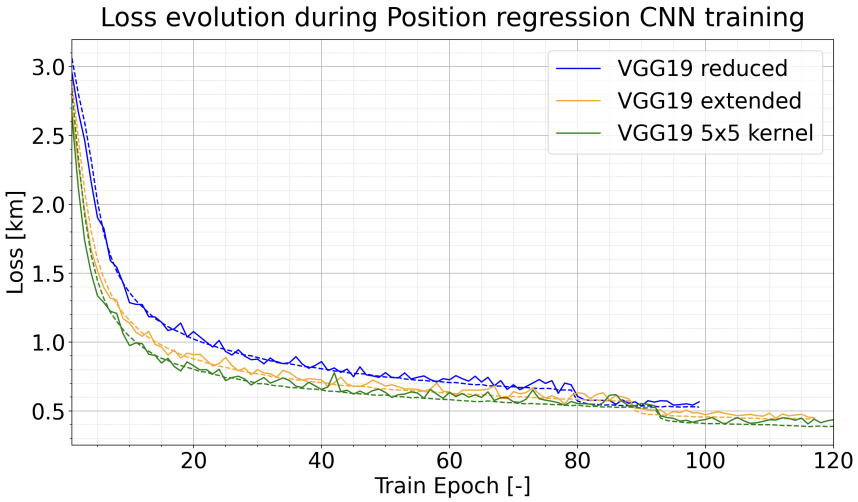


Figure 4.15: Train and Loss evolution for Position regression.

be estimated from other sources like star trackers and the dynamics of the target are also known to some degree, the accuracy can be substantially improved by constraining the roll angle range. For example, an angular separation of the camera frame vertical direction cap to 50 degrees from the North direction, reduced approximately a 20 percent the loss with respect to the 360 degrees random rotation, and for a large amount of the asteroid population, the spin axis direction has been characterized. Knowing the approximate asteroid spin axis direction and the spacecraft attitude with respect to the International Celestial Reference System, could potentially improve the accuracy of the CNN estimation. Regarding the introduction of an albedo map compared to the plain color shape model, the impact on training loss is small, however, it is required to avoid degraded accuracy when providing real images for estimation, specially at low phase angles when images are dominated by the albedo over the topography. For this last case with all image effects combined, two modified versions of VGG-19 have been compared with the reduced VGG-19 form used to analyse the different image effects. The results are shown in Fig. 4.15. First, the extended version adding two extra fully connected layers on top (orange line) results in a loss reduction of 0.127 kilometers with some extra training epochs. Then, when increasing the kernel size of all convolution layers from 3x3 to 5x5 (green line), the loss is further decreased 0.054 kilometers to a value of just 0.389 kilometers. However, as a consequence of the 5x5 kernel size extension, the number of parameters goes up to 175 millions compared to the 20 millions of the reduced version. Therefore, it might not be suitable for cases with limited disk budget or when the model has to be uplinked to the spacecraft.

The altitude-normalized loss resulting from training with multiple shape models and with multiple image resolution has been compared for three altitude ranges: 500 to 1000 metres, 1.4 to 3 kilometres, and 5 to 20 kilometres. When assessing the spatial resolution of the asteroid shape model, the higher resolution model results in reduced loss compared

to the low resolution model at any altitude. The more detailed surface features make easier for the CNN to locate and estimate the position of the camera. Moreover, when using the higher resolution model, the normalized loss is reduced when moving farther from the asteroid as more features on the asteroid limb become visible on the image. Regarding the image resolution, the higher 480x480 resolution yields a substantial decrease of the loss compared to the standard 224x224 for the low resolution model, however for the high resolution model the loss improvement is negligible. This suggest that while an increased image size might perform better for other targets with a smoother surface, for targets like Bennu with a surface full of features that the CNN can learn for pose estimation, the smaller 224x224 image size can provide optimal performance. Moreover, the reduced image size translates into smaller input layers and faster inference times.

4.6.4. Optimization and Quantization

An important aspect of neural networks is its lightweight and efficiency, characteristics that make them suitable to be executed onboard for autonomous navigation or even to be trained on ground and uplinked to the spacecraft after launch. Regarding the disk usage of the neural network, it is directly related to the size or number of coefficients composing the network. In general, the number of coefficients increases with the wider layers and deeper architectures. The baseline CNN architecture consisting on two convolution and two fully-connected layers has 15 million parameters, resulting on a disk space of 120MB. On the other hand, the deeper VGG-19 architecture on its base form has 139 million parameters, resulting on a substantial increase of 1065MB of disk space. By reducing the fully connected layers after the convolution blocks, the VGG-19 model can be reduced to 20 million parameters and a more adequate 160MB of disk usage. For the efficiency-oriented MobileNet architecture, the size is substantially optimised with just 2 million parameters and 19MB.

Independent of the chosen architecture, other techniques can be applied to reduce the size of the neural network, the most common one being quantization. Full integer quantization has been applied, differing from other quantization techniques like dynamic range quantization, in the fact that not only model weights are transformed from floating point to integer, but all the network math becomes integer quantized. This technique reduces the model size by a factor of 4 and improves CPU speedup by a factor of more than 3. The full integer conversion has been performed with the TensorFlow Lite Converter. Quantization-aware training has been applied to directly train the models applying quantization, and although it slightly improved generalization, substantial differences with models which were standard trained and quantized afterwards were not observed. With quantization, the size of the reduced VGG-19 architecture turns out to 40MB, while for MobileNet becomes just about 5MB. Considering multiple CNNs are required for a global pose estimation solution (one per sector of the discretized space), the total size required would be approximately 160MB for 32 sectors discretization and MobileNet architecture.

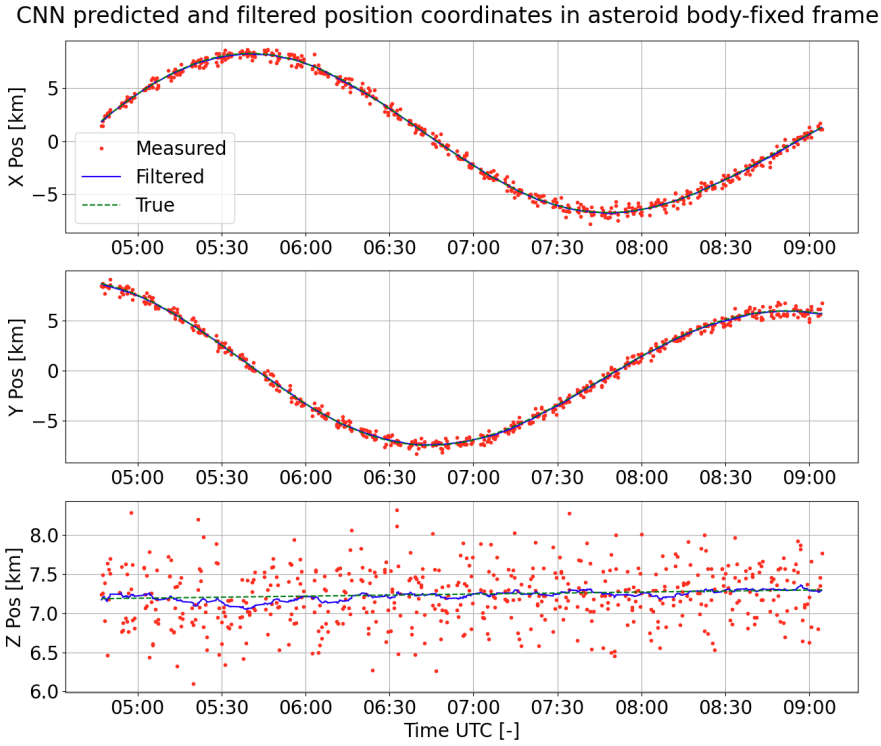


Figure 4.16: Comparison of CNN output position coordinates estimation and Kalman filtered for real images during hyperbolic flybys.

If VGG-19 is used instead for the CNNs, the total size would increase to 1280MB. The global hybrid model implementing the reduced VGG-19 architecture has been tested in a ARM64 Dual Core with 8GB RAM getting a data rate of 12 frames per second and a CPU usage of 62%.

4.6.5. Validation with Real Images

The trained networks have been validated with real OCAMS images at different phases of the OSIRIS-REx mission. To do so, the input images follow the full workflow described in Fig. 4.7. First, the images are de-shifted, resulting in the asteroid being centered in the images but missing all regions that were outside of the camera field-of-view. Thanks to the introduced shift cutout erase data augmentation in the training process, the classification network successfully estimates the sector and provides the de-shifted image to the low-level regression CNN trained for that sector. Finally, the low-level regression CNN estimates the position and roll angle of the camera associated with the input image. These estimations have been compared with the position and attitude computed using the OSIRIS-REx SPICE Data Set, showing there is no degradation in accuracy when providing as input real images which have never been seen before by the CNNs, trained only with synthetic images. Moreover, the error of the regression CNNs output has been observed to follow a normal distribution with zero mean. This can be appreciated in Fig.

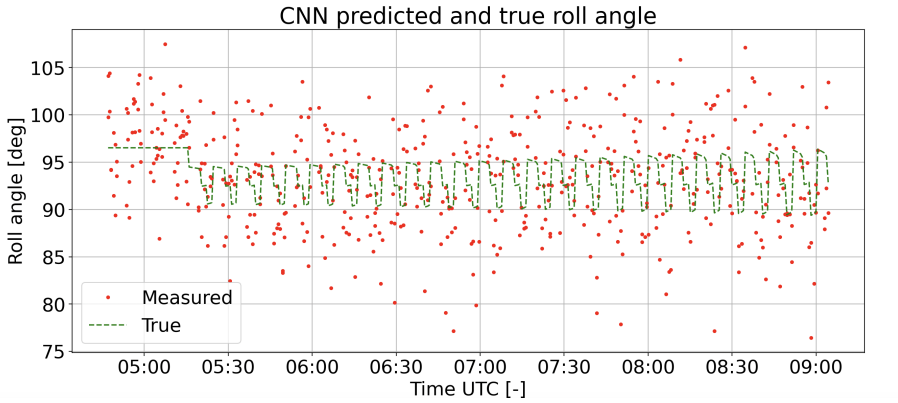


Figure 4.17: CNN output roll angle estimation for real images during hyperbolic flybys.

4.16 for the position vector coordinates with respect to the asteroid body fixed frame during one hyperbolic arc (resulting in a full revolution around the asteroid fixed frame) and in Fig. 4.17 for the roll angle.

Considering the CNN output as the signal of any other sensor, the filtering of the position estimation using a Kalman filter has been tested. A simple Kalman filter with a six-dimensional state space model representing position and velocity components has been implemented. The state space model is based on Keplerian dynamics, accounting for the asteroid point mass gravity as the only acceleration and applying numerical integration to propagate the states. The observation model includes only the position components (provided as output by the regression CNN). A value of 1 km is used for the measurement noise standard deviation based on CNN performances, and an estimate of 0.001 km/s^2 is chosen for the process noise standard deviation to account for unmodeled perturbations. As shown in Fig. 4.18 for the tested orbit arcs the results are promising, reducing the mean translation error from around 500 metres as directly output by the CNN (red dots) to less than 40 metres after filtering (blue line).

4.7. Conclusion

In this article, the training of CNNs applied to monocular vision navigation has been extended, assessing variations in target model and camera parameters which were not accounted in previous works. The main result is the successful validation with real images without any accuracy degradation, using CNNs trained only with synthetic images. This continues setting up the basis for developing feasible deep learning navigation algorithms for orbiting minor bodies. The usage of multiple shape models with variable spatial resolution together with the addition of a detailed albedo map was key to filling the gap between synthetic and real images. It is true that high-detail shape models of future missions targets required for generating the synthetic training sets, may not be available ahead of the spacecraft arrival and in-situ measurements have taken place. Nonetheless, there are

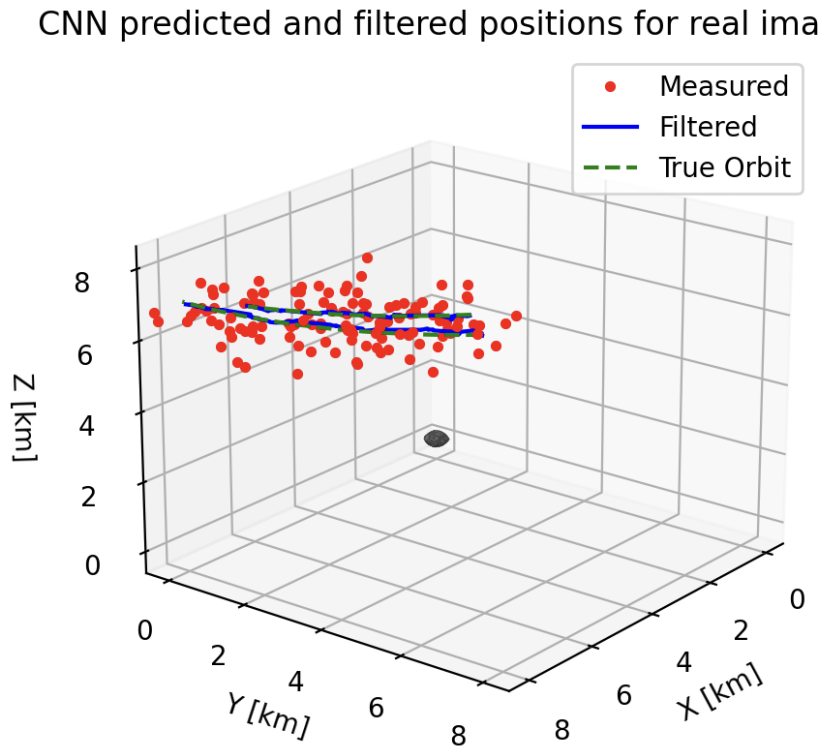


Figure 4.18: Comparison of CNN output position estimation and Kalman filtered for real images during hyperbolic flybys.

some interesting cases of shape models based on punctual observations of past and current missions; like for Phobos and Deimos based on Mars Express HRSC observations [76]; for the binary asteroid system composed by Didymos and Dimorphos visited by DART and LICIAcube [78] which will be later on visited by Hera spacecraft; and even asteroid Apophis that will be visited by Osiris-Apex and possibly a European counterpart mission, RAMSES [32]. In addition, the implementation of optimized and custom architectures like the Time Distributed CNN and modified versions of state-of-the-art CNNs like VGG or MobileNet have been proven to boost the achieved accuracy, which shall be further improved for more constrained scenarios. Of special interest for future developments is the TdCNN, that could be trained to learn time-derived quantities like velocity or target rotation.

4.8. Acknowledgment

This research is part of the R+D+i project TED2021-132099B-C31 funded by MCIN/AEI (10.13039 / 501100011033) and by the European Union NextGenerationEU PRTR. The authors also acknowledge the Principal Investigator(s) Rizk, B. (University of Arizona) of the OCAMS instrument onboard the OSIRIS-REx mission for providing data sets in the data archive.

5. Conclusions and Future Work

5.1. Conclusions

This thesis presents a complete framework for the training with synthetic data of neural networks devoted to optical navigation in deep space missions. The proposed approach was first implemented targeting missions to small bodies like comets, asteroids and minor moons, being then extended to large bodies such as planets. This wide operational range is paramount for future missions, now in development, with extremely challenging mission profiles. Moving away from single fly-by or orbiter missions studying a single target, there is currently a trend to have multitarget missions which can include fly-bys, orbital phases, touch-and-go, sample-return and extended mission to yet another target, all performed by a single spacecraft. Moreover, the shift towards private companies and small satellites into science and solar system exploration missions, turned traditional methods for optical navigation cost-inefficient and unsuitable due to infrastructure limitations and operational requirements. These would consequently limit the scientific outcome of such missions.

Considering the previous, the neural network approach for optical navigation presented in this dissertation is a key enabler to support the autonomous navigation in the new space era. As indicated in previous sections, the lack of enough generic and curated data of previous space missions has been one of the main obstacles towards the integration of a full CNN solution for autonomous navigation. To tackle this, the software package SPyRender has been developed as part of this thesis with the purpose of generating high quality synthetic data that can be used to train neural networks. Starting with a proof of concept, a GPU-accelerated rendering pipeline was put in place to generate synthetic images of small inactive bodies, showcasing the capabilities of using synthetic data to fit the needs of the hungry machine learning models. Throughout the three contributions included in this thesis, Chapters 2, 3, and 4, SPyRender has been developed into a complete framework for rendering synthetic images across any type of operational regime and combining multiple types of geospatial data, including topography, albedo, spectroscopy, atmospheric models, and more. The capabilities of SPyRender have been successfully tested in use cases as different as active comets ejecting dust clearly visible in the images, Earth observation missions affected by atmospheric perturbations, clouds and even long-term seasonal variations of the terrain. Specially focusing on adverse illumination conditions, the main weakness of state-of-the-art optical navigation methods, the synthetic images produced by SPyRender have been compared with real images at extreme low and high phase angles for multiple missions including Rosetta, OSIRIS-REx and OPS-SAT. The work invested on producing digital twins of celestial bodies and achieving a realistic rendering has been the basis enabling to successfully train CNNs for optical navigation.

The extensive training sets produced in the contributions included in this thesis have been of great value in comprehending the impact on the characteristics of the training data on the achieved performance of the trained networks, specially identifying the main correlations between the image variables and the CNN training behavior. Regarding the intrinsic characteristics of the images, the impact of off-nadir angle, target rotation, illumination conditions, and image intensity to noise ratio, have been quantified properly determining the limitations of the trained models. On the other hand, collective properties of the training sets such as total number of images, batch sizes, distribution and homogeneity of the target variables in the data, and data augmentation techniques have also been proven to be of equal relevance on the achieved accuracy, specially coping with generalization and training stability. For instance, the implementation of on-runtime data augmentation techniques, like the cut-out erase or pixel histogram shift, combined with the variations introduced by the rendering pipeline, has been key to successfully bridge the gap between synthetic and real data during training. These were critical in the most challenging scenarios such as cloud-covered images, overexposed frames and most important, images at low and high phase angles up to 150 degrees, situations in which state-of-the-art methods like SPC would not work.

Regression and Multiclass-classification sequential CNNs have been trained and tested for multiple missions, investigating the advantages and limitations in each case, resulting in the main outcome of this dissertation, the proposed two-levels sequential global network. The high-level consisting of a multiclass-classification CNN in charge of labeling camera position to a sector among a predefined set of the discretized global space. Based on the estimated sector, it feeds the de-shifted input image to the corresponding low-level regression CNN which solves a local position estimation problem and estimates the camera position and Euler angles. In addition, the implementation of optimized and custom architectures like the Time Distributed CNN and modified versions of state-of-the-art CNNs like VGG or MobileNet have been proven to boost the network accuracy, while building on extremely light architectures. The designed models have been tested with flight-ready hardware, specially oriented for small satellites with low-resources, successfully testing the integration of the CNNs with standard navigation filters and validating its operational readiness.

5.2. Future Work

The proposed CNN methodology has been tested for multiple bodies as reported in this dissertation: regular shape asteroid, irregular shape comet, and planet with an atmosphere. The results for these cases serve as a proof of concept, however, a more in depth benchmark should be performed by applying this methodology to a broader list of known bodies. There is available topography and spectral data for a number of minor bodies such as: asteroids Itokawa and Ryugu, comets 9P/Tempel 1 and 103P/Hartley, or minor moons like Phobos and Deimos of Mars, Amalthea of Jupiter, and Hyperion of Saturn. For large

bodies, the extensive data of Mars would be a great comparison with the performance obtained for the Earth, but it could also be applied for Mercury, dwarf planets like Ceres, giant asteroids like Vesta and large moons like Ganymede, Enceladus or the Moon itself. In fact, the presented approach has already been tested in Lunar missions (not included in this dissertation) using data from LRO and SELENE missions creating extremely high resolution Lunar DTMs. The rendering pipeline has been extended to integrate very large models while handling very long cast shadows. This is of special interest to future Lunar missions targeting the polar regions of the Moon, with shadows extending hundreds of kilometers.

Regarding the architecture of the global CNN approach, although it serves its purpose for estimating the spacecraft pose, the total size of the network could be further reduced by applying a multi-branch model. This implies that the different blocks of the CNN solution share the upstream convolution layers, but then multiple branches follow, each of them with different output layers estimating a different component of the pose solution. As the upstream layers are shared across the blocks, the weights associated with these layers are the same, and consequently the weight is reduced. However, the training of a multi-output network suffers from instabilities and convergence issues that have to be carefully addressed. On the input side, real missions count with other sensors which can directly provide a part of the pose solution. Even small spacecrafts like cubesats count with star trackers providing absolute attitude estimation and sometimes LiDAR, capable of resolving the distance to the target. Although these measurements could be integrated with the CNN output pose at the navigation filter stage applying traditional sensor fusion methods, it should be investigated the possibility of providing these measurements as an auxiliary input to the CNN. This multi-input approach could potentially help the trained CNN learning faster and more accurate pose estimations by counting already with part of the actual geometry associated with an input image.

In recent years, Transformers have experienced an incredible surge with the development of tools like ChatGPT or Dall-E. Mainly used for Large Language Models, Transformers can also be applied to Computer Vision tasks. An image is split into smaller fixed-sized patches which are treated as a sequence of tokens. Compared to CNNs, Visual Transformers require less resources to pretrain and its performance on large datasets can be transferred to smaller downstream tasks. Machine learning models are evolving at an astonishing pace, and further developments of the proposed machine learning navigation solution should also explore other architectures apart from CNNs.

Bibliography

[1] H. E. Stauss et al., "Scientific findings from explorer vi," *NASA Scientific and Technical Information Division*, vol. 1, pp. 1–7, 1965.

[2] W. R. Shelton, *Soviet Space Exploration: the First Decade*. MW Books, 1969.

[3] J. A. Dunne, "Mariner 10 mercury encounter," *Science*, vol. 185, pp. 141–142, 1974.

[4] T. G. Northrop et al., "Pioneer 11 saturn encounter," *J. Geophys. Res.*, vol. 85, pp. 5651–5652, 1980.

[5] B. Evans, *NASA's Voyager Missions*. Springer, 2022.

[6] E. Levin et al., "Lunar orbiter missions to the moon," *Scientific American*, vol. 218, pp. 58–66, 1968.

[7] P. Rathsman et al., "Smart-1: Development and lessons learnt," *Acta Astronautica*, vol. 57, pp. 455–468, 2005.

[8] M. Kato et al., "The kaguya mission overview," *Space Science Review*, vol. 154, pp. 3–19, 2010.

[9] N. Bhandari, "Chandrayaan-1: Science goals," *Journal of Earth System Science*, vol. 6, pp. 701–709, 2005.

[10] W. G. Breckenridge and C. H. Acton, "A detailed analysis of mariner nine tv navigation data," *American Institute of Aeronautics and Astronautics, Guidance and Control Conference*, 1972.

[11] S. Lauro et al., "Multiple subglacial water bodies below the south pole of mars unveiled by new marsis data," *Nature Astronomy*, vol. 5, pp. 63–70, 2021.

[12] D. L. Clements, *Venus phosphine: Updates and lessons learned*, 2024. arXiv: 2409.13438 [astro-ph.EP].

[13] R. L. McNutt et al., "Messenger at mercury: Early orbital operations," *Acta Astronautica*, vol. 93, pp. 509–515, 2014.

[14] L. J. Spilker, "Cassini's final year at saturn: Science highlights and discoveries," *Geophysical Research Letters*, vol. 46, no. 11, pp. 5754–5758, 2019.

[15] M. R. Combi et al., "The study of comets," *The Future of Solar System Exploration*, vol. 272, pp. 323–336, 2002.

[16] V. Shevchenko and R. Mohamed, "Spacecraft exploration of asteroids," *Solar System Research*, vol. 39, pp. 73–81, 2005.

[17] T. Rosenvinge et al., "The international cometary explorer (ice) mission to comet giacobini-zinner," *Science*, vol. 232, pp. 353–356, 1986.

[18] R. Sagdeev et al., "Vega spacecraft encounters with comet halley," *Nature*, vol. 321, pp. 259–262, 1986.

[19] S. Sandford et al., "The stardust sample return mission," *Sample Return Missions*, pp. 79–104, 2021.

[20] A. Accomazzo et al., "The final year of the rosetta mission," *Acta Astronautica*, vol. 136, pp. 354–359, 2017.

[21] M. Barucci et al., "Rosetta asteroid targets: 2867 steins and 21 lutetia," *Space Science Reviews*, vol. 128, pp. 67–78, 2007.

[22] B. Williams, "Technical challenges and results for navigation of near shoemaker," *Johns Hopkins APL Tech. Dig.*, vol. 23, 2002.

[23] J. Kawaguchi et al., "Hayabusa—its technology and science accomplishment summary and hayabusa-2," *Acta Astronautica*, vol. 62, pp. 639–647, 2008.

[24] Y. Mimasu et al., "Mission results and extended mission of hayabusa2," *44th COSPAR Scientific Assembly*, vol. 44, p. 161, 2022.

[25] D. Lauretta et al., "Osiris-rex: Sample return from asteroid (101955) bennu," *Space Science Reviews*, vol. 212, pp. 925–984, 2017.

[26] D. DellaGiustina et al., "Osiris-apex: An osiris-rex extended mission to asteroid apophis," *The Planetary Science Journal*, vol. 4, p. 198, 2023.

[27] E. Kulu, *Cubesats nanosatellites - 2024 statistics, forecast and reliability*, 2024.

[28] ESA, *Esa's in orbit demonstration fleet*, 2024.

[29] S. Bhaskaran et al., "Autonomous nucleus tracking for comet/asteroid encounters: The stardust example," *1998 IEEE Aerospace Conference Proceedings (Cat. No.98TH8339)*, pp. 353–365, 1998.

[30] T. Morley et al., "Rosetta navigation from reactivation until arrival at comet 67p/churyumov-gerasimenko," *Proceedings of the 25th International Symposium on Space Flight Dynamics (ISSFD)*, 2015.

[31] P. Michel et al., *The esa hera mission to the binary asteroid didymos: Planetary defense and bonus science*, 2020.

[32] M. Kueppers et al., "Ramses – esa's study for a small mission to apophis," *Bulletin of the AAS*, vol. 55, p. 8, 2023.

[33] V. Franzese et al., "Deep-space optical navigation for m-argo mission," *The Journal of the Astronautical Sciences*, vol. 68, pp. 1034–1055, 2021.

[34] R. Sandau, "Small satellites for earth observation," *Springer*, 2008.

[35] P. Mhangara, "The emerging role of cubesats for earth observation applications in south africa," *Photogrammetric Engineering and Remote Sensing*, 2020.

[36] W. Owen, *Spacecraft Optical Navigation (JPL Deep-Space Communications and Navigation Series)*. Wiley, 2024.

[37] T. C. Duxbury and J. D. Callahan, "Phobos and deimos astrometric observations from mariner 9," *Astronomy and Astrophysics*, vol. 216, pp. 284–293, 1989.

[38] K. Getzandanner, "Small-body proximity operations & tag: Navigation experiences & lessons learned from the osiris-rex mission," *AIAA SciTech Forum*, 2022.

[39] T. Okada, "Thermography of asteroid and future applications in space missions," *Applied Sciences*, vol. 10, 2020.

[40] D. A. Lorenz et al., "Lessons learned from osiris-rex autonomous navigation using natural feature tracking," *2017 IEEE Aerospace Conference*, pp. 1–12, 2017.

[41] R. Gaskell, "Comet 67p/cg: Preliminary shape and topography from spc," in *46th American Astronomical Society, DPS meeting*, 2014.

[42] C. Adam et al., "Stereophotoclinometry for osiris-rex spacecraft navigation," *Planetary Science Journal*, vol. 4, p. 167, 2023.

[43] R. Pardo de Santayana and M. Lauer, *Optical measurements for rosetta mission near the comet*, 2015.

[44] D. Wibben et al., "Osiris-rex post-tag observation trajectory design and navigation performance," *44th Annual AAS Guidance, Navigation and Control (GNC) Conference*, 2022.

[45] T. Hashimoto et al., "Vision-based guidance, navigation, and control of hayabusa spacecraft. lessons learned from real operation," *IFAC Proceedings Volumes*, vol. 43, pp. 259–264, 2010.

[46] O. Naoko et al., *Image-based autonomous navigation of hayabusa2 using artificial landmarks: Design and in-flight results in landing operations on asteroid ryugu*, 2020.

[47] S. Chiodini et al., "Robust visual localization for hopping rovers on small bodies," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[48] B. Morrell et al., "Automatic feature tracking on small bodies for autonomous approach," *AIAA ASCEND 2020*, 2020.

[49] R. Del Prete, "A novel visual-based terrain relative navigation system for planetary applications based on mask r-cnn and projective invariants," *Aerotec. Missili Spaz*, vol. 101, pp. 335–349, 2022.

[50] P. Mancini, "Deep learning for asteroids autonomous terrain relative navigation," *Advances in Space Research*, vol. 71, 2023.

[51] A. Adegun, "Review of deep learning methods for remote sensing satellite images classification: Experimental survey and comparative analysis," *Journal of Big Data*, vol. 10, 2023.

[52] A. Barret, "Noah-h, a deep-learning, terrain classification system for mars: Results for the exomars rover candidate landing sites," *Icarus*, vol. 371, 2022.

[53] E. Dalsasso, "Sar image despeckling by deep neural networks: From a pre-trained model to an end-to-end training strategy," *Remote Sensing*, vol. 12, p. 2636, 16 2020.

[54] F. Feresin, "In space image processing using ai embedded on system on module: Example of ops-sat cloud segmentation," *On-Board Payload Data Compression (OBPDC) Workshop*, 2020.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv e-prints*, 2017.

[56] S. Sharma et al., "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," *2018 IEEE Aerospace Conference*, pp. 1–12, 2018.

[57] R. Linares et al., "A deep learning approach for optical autonomous planetary relative terrain navigation," *Spaceflight Mechanics 2017*, pp. 3293–3302, 2017.

[58] A. Kendall et al., "Posenet: A convolutional network for real-time 6-dof camera relocalization," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2938–2946, 2015.

[59] A. Garcia et al., *Lspnet: A 2d localization-oriented spacecraft pose estimation neural network*, 2021.

[60] N. Gorelick, "Google earth engine: Planetary-scale geospatial analysis for everyone," *Remote Sensing of Environment*, 2017.

[61] JAXA, *Martian moons exploration (mmx) mission supplemental design document*, 2017.

[62] G. Jones et al., *Comet interceptor: A proposed esa mission to a dynamically new comet*, 2019.

[63] J. Atchison et al., *Double asteroid redirection test (dart) mission design and navigation for low energy escape*, 2018.

[64] N. Mastrodemos et al., "Optical navigation for the dawn mission at vesta," *23rd International Symposium on Space Flight Dynamics*, 2012.

[65] B. Geiger et al., *Rosetta-navcam to planetary science archive interface control document*, 2016.

[66] J. Saito et al., *Hayabusa asteroid multi-band imaging camera (amica) data archive*, 2008.

[67] R. Honda et al., *Plans of hayabusa2's onc image archiving and public release*, 2020.

[68] B. Rizk et al., *Origins, spectral interpretation, resource identification, security, regolith explorer (osiris-rex): Osiris-rex camera suite (ocams) bundle*, 2019.

[69] M. Pharr et al., *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2016.

[70] M. Matl, *Pyrender*, GitHub, 2019.

[71] C. Acton, "Ancillary data services of nasa's navigation and ancillary information facility," *Planetary and Space Science*, vol. 44, pp. 65–70, 1996.

[72] A. Annex et al., "Spiceypy: A pythonic wrapper for the spice toolkit," *Journal of Open Source Software*, vol. 5, no. 46, p. 2050, 2020.

[73] D. Haggerty et al., *Trimesh*, GitHub, 2019.

[74] S. Besse et al., "Esa's planetary science archive: Preserve and present reliable scientific data sets," *Planetary and Space Science*, vol. 150, pp. 131–140, 2018.

[75] R. Gaskell et al., *Spc shap5 cartesian plate model for comet 67p/c-g 3m plates*, Planetary Science Archive, 2017.

[76] K. Willner et al., "Pds release of phobos data from hrsc on mars express: Shape model, orthoimages and maps," in *European Planetary Science Congress*, 2015.

[77] P. Stooke, *Stooke small body shape models v2.0. ear-a-5-ddr-stooke-shape-models-v2.0.* NASA Planetary Data System, 2016.

[78] A. Capannolo et al., "Challenges in licia cubesat trajectory design to support dart mission science," *Acta Astronautica*, vol. 182, pp. 208–218, 2021.

[79] E. S. Service, *Rosetta archived spice kernel dataset*, 2017.

[80] A. Krizhevsky et al., "Imagenet classification with deep convolutional neural networks," *Advances In Neural Information Processing Systems*, vol. 25, pp. 1–9, 2012.

[81] S. Sharma and S. D'Amico, "Neural network-based pose estimation for noncooperative spacecraft rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4638–4658, 2020.

[82] M. Pugliatti et al., "Data-driven image processing for onboard optical navigation around a binary asteroid," *Journal of Spacecraft and Rockets*, vol. 59, pp. 1–17, 2022.

[83] C. Szegedy et al., "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[84] K. He et al., "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[85] A. Canziani et al., "An analysis of deep neural network models for practical applications," *CoRR*, 2016.

[86] M. Brahimi et al., "Deep learning for plant diseases: Detection and saliency map visualisation," *Human and Machine Learning*, 2018.

[87] H. Gao et al., "Densely connected convolutional networks," *arXiv e-prints*, 2016.

[88] A. Wongpanich et al., "Training efficientnets at supercomputer scale: 83% imagenet top-1 accuracy in one hour," *arXiv e-prints*, 2020.

[89] I. Goodfellow et al., *Deep Learning. Adaptive Computation and Machine Learning series*. MIT Press, 2016.

[90] G. Hinton et al., "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv e-prints*, 2012.

[91] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, vol. 21, 2010, pp. 807–814.

[92] C. Bishop, *Neural networks for pattern recognition*. 1995, p. 238.

[93] C. Bishop et al., *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[94] M. Lapin et al., "Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1533–1554, 2018.

[95] F. Chollet et al., *Keras probabilistic losses*, 2015.

[96] J. Qi et al., "On mean absolute error for deep neural network based vector-to-vector regression," *IEEE Signal Processing Letters*, vol. 27, pp. 1485–1489, 2020.

[97] X. Zhu and M. Bain, "B-cnn: Branch convolutional neural network for hierarchical classification," *arXiv e-prints*, 2017.

[98] S. Sharma and S. D'Amico, "Pose estimation for non-cooperative rendezvous using neural networks," *arXiv e-prints*, 2019.

[99] P. Muñoz et al., "Rosetta navigation during the end of mission phase," in *Proceedings of the 26th International Symposium on Space Flight Dynamics (ISSFD)*, 2017.

[100] B. Cheetham, "Capstone: A unique cubesat platform for a navigation demonstration in cislunar space," *ASCEND 2022*, 2022.

[101] S. Asmar and S. Matousek, "Mars cube one (marco) shifting the paradigm in relay deep space operation," *SpaceOps 2016 Conference*, 2016.

[102] D. Evans, "Ops-sat: Preparing for the operations of esa's first nanosat," *14th International Conference on Space Operations*, 2016.

[103] D. Evans, "The esa ops-sat mission: Don't just say there is a better way, fly it and prove it," *15th International Conference on Space Operations*, 2018.

[104] T. Mladenov, "Ops-sat status update and tle," *ESOC, opssat1.esoc.esa.int/news/18*, 2020.

[105] N. Ammann, "Using an uav for testing an autonomous terrain-based optical navigation system for lunar landing," *2018 IEEE Aerospace Conference*, 2018.

[106] X. Wan, "Terrain aided planetary uav localization based on geo-referencing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2022.

[107] F. Pech-May, "Sentinel-1 sar images and deep learning for water body mapping," *Remote Sensing*, vol. 15, p. 3009, 12 2023.

[108] M. Wulder and J. Masek, "Landsat—continuity and change: Improved observations over 45 years," *Remote Sensing of Environment*, vol. 225, pp. 127–147, 2019.

[109] T. Farr and M. Kobrick, "Shuttle radar topography mission produces a wealth of data," *Eos, Transactions American Geophysical Union*, vol. 81, pp. 583–585, 48 2000.

[110] R. Torres et al., "Gmes sentinel-1 mission," *Remote Sensing of Environment*, vol. 120, pp. 9–24, 2012.

[111] M. Drusch, U. Del Bello, and S. Carlier, "Sentinel-2: Esa's optical high-resolution mission for gmes operational services," *Remote Sensing of Environment*, vol. 120, pp. 25–36, 2012.

[112] C. Donlon et al., "The global monitoring for environment and security (gmes) sentinel-3 mission," *Remote Sensing of Environment*, vol. 120, pp. 37–57, 2012.

[113] R. Perko et al., "Very high resolution mapping with the pléiades satellite constellation," *Am. J. Remote Sens.*, vol. 6, pp. 89–99, 2018.

[114] B. O. Community, "Blender - a 3d modeling and rendering package," *Blender Foundation, Stichting Blender Foundation, Amsterdam*, 2018, URL: http://www.blender.org.

[115] ESA, "Sentinel-2 user handbook," *ESA*, 2015.

[116] NASA, "The shuttle radar topography mission (srtm) collection user guide," *NASA*, 2015.

[117] M. Abadi, "Tensorflow: Large-scale machine learning on heterogeneous systems," *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, 2015.

[118] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICLR)*, 2015.

[119] A. Howard, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[120] P. Novac, "Quantization and deployment of deep neural networks on microcontrollers," *CoRR*, 2021.

[121] M. Dor et al., "Visual slam for asteroid relative navigation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2066–2075, 2021.

[122] A. Escalante et al., "Churinet—applying deep learning for minor bodies optical navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, pp. 3566–3578, 2023.

[123] P. Navigation and A. I. F. (NAIF), *Osiris-rex spice data archive*, https://naif.jpl.nasa.gov, 2023.

[124] Lunar and U. o. A. Planetary Laboratory, *Bennu osiris-rex ocams global albedo mosaic 6.25cm v6*, 2021.

[125] K. Walsh et al., "Craters, boulders and regolith of (101955) bennu indicative of an old and dynamic surface," *Nature Geoscience*, vol. 12, p. 1, 2021.

[126] B. Semenov, *Osiris-rex archived spice kernel dataset*, 2021.

[127] J. Jerubbal et al., "Impact of image size on accuracy and generalization of convolutional neural networks," *International Journal of Research and Analytical Reviews*, vol. 6, pp. 70–80, 2019.

[128] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv e-prints*, 2015.