Politecnico di Torino

DIPARTIMENTO DI INGEGNERIA MECCANICA E AEROSPAZIALE

Master's Degree in Aerospace Engineering

Visual-based navigation with AI for satellite pose estimation

Supervisors:

Fabrizio STESINA
Maurizio FANTINO
Stefano BERGIA

Candidate:

Irene CARACCIOLO

ACADEMIC YEAR 2025–2026

"Don't let anyone rob you of your imagination, creativity, or curiosity.
It's your place in the world; it's your life.
Go on and do all you can with it, and make it the life you want to live."
– Mae Jemison, first African American woman astronaut in space

Dedicato alle donne che hanno plasmato la donna che sono
e che mi hanno guidata in questo percorso delicato.
Alle mie nonne Agnese e Irene,
a mia mamma Valeria,
a mia sorella Agnese,
alle mie zie Caterina, Gabriella e Tiziana,
alle mie cugine, Lucia, Rachele, Giuliana e Cristiana.

Abstract

The rapid growth of the space market is introducing many technical challenges in ensuring the long-term sustainability of the orbital environment and mission safety. Many solutions are being investigated, ranging from on-orbit servicing to active space debris removal, increasing the need for reliable and robust relative navigation systems. In particular, accurate monocular 6-DoF pose estimation of spacecraft represents a key enabling capability for future proximity operations.

This thesis investigates learning-based monocular vision approaches for spacecraft pose estimation under realistic illumination conditions and domain shifts. The study is conducted using the SPEED+ benchmark dataset, which combines large-scale synthetic imagery with Hardware-in-the-Loop (HIL) acquisitions, enabling systematic evaluation across synthetic-to-real transitions.

The work focuses on keypoint-based indirect pipelines, where neural networks are employed to localise 2D projections of known 3D spacecraft landmarks, and the full relative pose is recovered through geometric Perspective-n-Point (PnP) solvers. Particular emphasis is placed on high-resolution convolutional architectures for dense keypoint localisation, analysing the impact of heatmap-based representations, architectural depth, loss formulations, and augmentation strategies on pose accuracy and robustness.

A comparative analysis between different network paradigms is conducted, including coordinate-regression and heatmap-based models, highlighting the trade-offs in terms of accuracy, generalisation capability, and computational efficiency. Robustness to illumination variability and domain gap is systematically evaluated using the official SPEED+ metrics, including translation error, orientation error, and the ESA composite score.

The results provide a comprehensive experimental assessment of high-resolution keypoint localisation strategies for spacecraft pose estimation, examining their geometric consistency, robustness, and operational reliability. The proposed framework contributes to the development of reliable and flight-oriented monocular navigation solutions for future autonomous proximity operations.

Contents

Abstract **II**

List of Acronyms **XI**

1 Introduction **1**
1.1 Problem Definition 4
1.1.1 Relative pose as a rigid-body transformation 4
1.2 Cooperative vs. Uncooperative Targets 6
1.2.1 Cooperative Targets 6
1.2.2 Uncooperative Targets 7
1.3 Sensors for Spacecraft Pose Estimation 8
1.4 Scope and Contributions of This Work 9

2 State of the Art **11**
2.1 Overview 11
2.2 Datasets for CNN-based Spacecraft Pose Estimation 12
2.2.1 Benchmark datasets for instance-level pose estimation 12
2.2.2 Generalisation-oriented datasets 14
2.3 SPEED+ dataset 16
2.3.1 Evaluation Metrics 18
2.3.2 Positioning of SPEED+ 20
2.4 Vision-based Pose Estimation Pipelines 20
2.4.1 Direct, Indirect, and Hybrid Methods 20
2.4.2 Pipeline Architectures 21
2.4.3 Performance Comparison 23
2.5 Positioning of This Work 24

3 YOLOv11-based Pipeline **26**
3.1 YOLOv11-Pose Framework 26
3.1.1 `Ultralytics` and the YOLO Environment 26
3.1.2 YOLO Architecture Overview 27

3.1.3 YOLOv11-Pose Architecture 27
3.1.4 Model Variants and Scaling Strategy 28
3.2 Label Generation from SPEED+ Annotations 29
3.2.1 SPEED+ annotation format and keypoint geometry 29
3.2.2 Pinhole projection model 29
3.2.3 Bounding box generation from projected keypoints 31
3.2.4 Ray-casting visibility model for the keypoint visibility flag computation 31
3.2.5 Label format 35
3.3 YOLOv11 Training Setup 35
3.3.1 Training script and hyperparameters 35
3.3.2 Augmentation techniques 35
3.3.3 Loss Functions 37
3.3.4 Mean Average Precision 40
3.4 6-DOF Pose Recovery via PnP solver 42
3.4.1 Pose estimation pipeline overview 42
3.4.2 Evaluation metrics 43
3.4.3 Inference and Confidence Threshold Selection 43
3.4.4 Comparison between different PnP solvers 48
3.4.5 Final pose evaluation with SQPnP 51
3.5 Refinement Process 52
3.5.1 Refinement Strategy A: Top-K selection + edge-based alignment 52
3.5.2 Refinement Strategy B: robust reprojection refinement (LM on keypoints) 54
3.5.3 Refinement Process: Results Overview 56
3.6 Results Overview 61
3.6.1 Runtime Evaluation 63
3.6.2 Error Distribution vs Relative Distance 64
3.6.3 Final Considerations 66

4 HRNet-based Pipeline 67
4.1 Input Preprocessing and Coordinate Systems 67
4.1.1 Network Input Standardisation 67
4.1.2 Heatmap Supervision Grid and Coordinate Mapping 68
4.2 HRNet Architecture 69
4.2.1 Backbone Configuration 69
4.2.2 Keypoint Prediction Head 70
4.2.3 Keypoint Coordinate Extraction 70

4.3 Loss Function Definition . . . 72
4.3.1 Strategy 1: Heatmap loss only . . . 73
4.3.2 Strategy 2: Heatmap loss + Coordinate loss . . . 73
4.3.3 Strategy 3: Heatmap Loss + Coordinate Loss Scheduling . . . 75
4.3.4 Results and Formulation Comparison Overview . . . 75
4.4 Data augmentation . . . 78
4.5 Bounding Box Cropping . . . 81
4.6 Post processing refinements . . . 83
4.6.1 Translation Threshold . . . 83
4.6.2 Dynamic Evaluation . . . 86
4.6.3 LM refinement for the HRNet pipeline . . . 86
4.7 Results . . . 87
4.7.1 Comparison between Training Strategies . . . 87
4.7.2 Error distribution vs relative distance . . . 90
4.7.3 HRNet vs YOLO . . . 91
4.8 On-board Deployment Assessment on NVIDIA Jetson Orin . . . 93
4.8.1 Model Export and Precision Reduction . . . 93
4.8.2 Runtime Evaluation on Embedded Hardware . . . 94
4.9 Final Considerations . . . 95

5 Conclusions and Future Work **96**
5.1 Summary of the Work . . . 96
5.2 Main Findings . . . 97
5.3 Comparison with SOTA . . . 98
5.4 Limitations of the Proposed Approach . . . 99
5.5 Future Work . . . 100

List of Figures

1.1 Illustration of the increasing density of space debris in LEO, with classification based on the debris dimensions. Image credit: ESA. . . 1
1.2 Evolution of the Fengyun-1C debris orbit planes. Credit: [3] 2
1.3 Examples of sensors used for relative navigation purposes. 8

2.1 Visualisation of the TRON facility at SLAB 17
2.2 Representative samples from the three image domains of the SPEED+ dataset. 18

3.1 Visual representation of the YOLOv11 architecture layout. 28
3.2 Conceptual illustration of the ray–triangle intersection test used for keypoint visibility evaluation[51]. A ray is cast from the camera centre toward the 3D keypoint. If an intersection with the object surface occurs at a distance smaller than the keypoint depth, the keypoint is classified as occluded. 32
3.3 Examples of ground-truth keypoint projections on SPEED+ images. Visible keypoints are shown in green, while self-occluded keypoints are highlighted in yellow. 33
3.4 Interactive 3D visualisation of the Tango spacecraft in the camera reference frame. The spacecraft geometry, characteristic keypoints, camera origin, and field of view are shown. The scene illustrates the ray casting and triangle–intersection model used to determine keypoint visibility by evaluating line-of-sight occlusions between the camera and the spacecraft surface. The bottom picture shows the camera's field of view. 34
3.5 Examples of training batches generated during YOLOv11 training on the SPEED+ dataset. Images illustrate bounding boxes, keypoints, and the effect of data augmentation techniques applied during training. 36
3.6 Pose loss per keypoint evolution: comparison between YOLOv11-n and YOLOv11-l models on training and validation sets. 39

3.7 Comparison of training and validation loss components for YOLOv11-n and YOLOv11-l during optimization. 40
3.8 Comparison of $mAP_{50–95}$ evolution during training for YOLOv11-n and YOLOv11-l models. 42
3.9 Percentage of the discarded samples from the inference with varying confidence value. The plots refer to both the YOLOv11n and the YOLOv11l models. Each plot shows a curve for each subset. 46
3.10 The figures illustrate the trade-off between confidence threshold and pose estimation accuracy in terms of translation error E_t, orientation error e_q, SLAB score and RMSE. The plots refer to both the YOLOv11n model (left) and the YOLOv11l model (right). Each plot presents a curve for each subset. 47
3.11 Boxplots of rotation error e_q, translation error E_t and SLAB score for each dataset (`synthetic`, `lightbox` and `sunlamp`) evaluated with different PnP using the YOLOv11-nano model predictions. 49
3.12 Boxplots of rotation error e_q, translation error E_t and SLAB score for each dataset (`synthetic`, `lightbox` and `sunlamp`) evaluated with different PnP using the YOLOv11-large model predictions. 50
3.13 Pareto analysis of the evaluated PnP solvers across the `synthetic`, `lightbox` and `sunlamp` domains. The horizontal axis represents the median SLAB score (accuracy), the vertical axis the failure rate (robustness), while marker size and colour represent the mean execution time (computational cost). Solvers located closer to the bottom-left region provide a more favourable trade-off between accuracy, robustness and speed. 51
3.14 The figures show the improvement obtained with the edge-refinement process for `lightbox` (top) and `sunlamp` (bottom), for both the YOLOv11n model (left) and the YOLOv11l model (right). The green mask represents the initial pose (R_0, t_0), while the blue mask represents the refined pose (R_{ref}, t_{ref}). 58
3.15 Visual examples of best and worst predictions and LM refinements on the `lightbox` domain, using the YOLOv11-l model. The pictures show ground-truth keypoint (green), predicted correspondences (blue) and refined pose (red). 59
3.16 Visual examples of best and worst predictions and LM refinements on the `sunlamp` domain, using the YOLOv11-l model. The pictures show ground-truth keypoint (green), predicted correspondences (blue) and refined pose (red). 60

3.17 Runtime overview for each dataset split (`synthetic`, `lightbox` and `sunlamp` 64
3.18 Mean and normalised translation errors and mean rotation error as a function of the relative target distance. 65

4.1 Representation of the spatial transformations used in the HRNet-based keypoint detection pipeline. The original SPEED+ image is resized to 512×512, the network predicts keypoint heatmaps on a 128×128 grid, and the final keypoint coordinates are obtained via argmax and remapped to the resized image coordinate system. 68
4.2 Example of keypoint heatmaps. From left to right: input image, ground-truth heatmaps, and predicted heatmaps. High-intensity regions correspond to the estimated keypoint locations. 71
4.3 HRNet-based CNN architecture 72
4.4 Loss evolution during training for the three investigated loss formulations. For each training configuration, the total loss is shown in linear scale (left), while the individual loss components are reported in logarithmic scale (right) to highlight their relative magnitudes and convergence behaviour. 77
4.5 Examples of augmented samples. 80
4.6 Pipeline of the crop-based preprocessing and coordinate reprojection used in the HRNet pose estimation framework. 82
4.7 Examples of bounding-box cropping applied to SPEED+ images. The first column shows the original frame, the second column shows the resized image with the detected bounding box, and the third column shows the cropped patch used as network input. 83
4.8 Diagnostic analysis of the peak bin in the `sunlamp` dataset, highlighting the presence of a small number of extreme outliers responsible for the spike in the mean error. 84
4.9 Histogram of the ground–truth distance distribution across the three datasets. 86
4.10 Mean and normalised translation errors and mean rotation error as a function of the relative target distance, evaluated on `lightbox` and `sunlamp`. 91
4.11 Distribution of the normalised translational error $E_{T,norm}$ as a function of the relative distance for the YOLO-based and HRNet-based pipelines. 92

List of Tables

2.1 Summary of publicly available datasets for spacecraft pose estimation and tracking. . . . 13
2.2 Dataset composition and splits for SPEED+. . . . 16
2.3 Pose estimation results on the SPEED+ dataset reported in the literature under different dataset configurations and evaluation protocols. The best results for each metric on the `lightbox` and `sunlamp` split are highlighted in red and blue, respectively. . . . 23

3.1 Fixed 3D coordinates of the 11 spacecraft keypoints expressed in the target body frame. . . . 30
3.2 Summary of the `Ultralytics` YOLO augmentation parameters used during training. Identical settings were adopted both for YOLOv11n-pose and YOLOv11l-pose. . . . 37
3.3 Pose estimation mean results for the first YOLOv11 training (no refinement), evaluated on the SPEED+ `validation` set and HIL domains, namely `lightbox` and `sunlamp`. . . . 52
3.4 Refinement mean results across all of the SPEED+ datasets. . . . 58
3.5 Pose estimation metrics for the YOLOv11 model with LM refinement, evaluated on the SPEED+ `validation` set and HIL domains, namely `lightbox` and `sunlamp`. . . . 62
3.6 YOLOv11 runtime performance comparison across datasets. . . . 63

4.1 SPEED+ metric results obtained on the intersection samples (in the `validation` dataset split) resulting from the comparison of all three loss formulation studies. The common valid subset accounts for 11801 samples (98.39% of intersection). . . . 78
4.2 Domain-shift augmentation strategy overview. The table also shows the activation probability value for each type of augmentation applied. 79
4.3 Pose estimation performance on the `Synthetic` split for different training strategies. . . . 88

4.4 Pose estimation performance on the `Lightbox` dataset for the different training strategies. 89
4.5 Pose estimation performance on the `Sunlamp` dataset for the different training strategies. 90
4.6 Performance assessment based on the metrics comparison between different platforms. 94
4.7 Runtime and latency evaluation on the NVIDIA Jetson Orin Nano platform, for both *Training 5* and *Training 6*. 95

5.1 Selected SOTA results for comparison with the proposed pipelines. . . 99

List of Acronyms

ASAT Anti-SATellite. 2

AVGS Advanced Video Guidance Sensor. 6

CAD Computer-Aided Design. 7

CNN Convolutional Neural Network. 9, 12, 13, 20, 67, 69, 97

DARPA Defense Advanced Research Projects Agency. 6

DART Double Asteroid Redirection Test. 8

DLR Deutsches Zentrum für Luft- und Raumfahrt. 8

DOF Degrees of Freedom. 4, 9, 12, 15, 16, 20, 24, 26, 28, 42, 43, 66, 67, 71, 96, 97, IV

EKF Extended Kalman Filter. 21, 100

ENVISAT ENVIronmental SATellite. 15

EOL End-Of-Life. 3

ESA European Space Agency. 1, 3, 8, 18, VI

G-SPET Generalizable Spacecraft Pose Estimation and Tracking. 15

GNC Guidance, Navigation and Control. 4, 66

GPS Global Positioning System. 6

HIL Hardware-in-the-Loop. 11–13, 15–19, 21, 22, 42, 51, 52, 56, 57, 61–63, 66, 78, 86, 96, 98–101, IX

HRSC High Resolution Stereo Camera. 8

HSV Hue, Saturation, Value. 35

IADC Inter-Agency Space Debris Coordination Committee. 3

ICP Iterative Closest Point. 7

IOS In-Orbit Servicing. 3

IoU Intersection-over-Union. 37, 38, 40, 41

LED Light Emitting Diodes. 6

LEO Low Earth Orbit. 1–3, 13, VI

LiDAR Light Detection and Ranging. 7, 8

NASA National Aeronautics and Space Administration. 6–8

PnP Perspective-n-Point. 9, 21, 24, 26, 28, 42–45, 48–51, 57, 62–64, 66, 67, 70, 71, 73, 76, 83–87, 89, 91, 93–97, IV, VII

PRISMA Prototype for Rendezvous and Autonomous Mission Operations. 6, 12, 16

RF Radio Frequency. 6

RMSE Reprojection Mean Squared Error. 43, 47, 56, 57, 61, VII

RSO Resident Space Object. 1

SHIRT Satellite Hardware-In-the-loop Rendezvous Trajectories. 13

SLAB Stanford Space Rendezvous Laboratory. 16, 17, 47–51, 56, 57, 61, 66, VI, VII

SLAM Simultaneous Localization and Mapping. 7

SOTA State Of The Art. 20, 21, 23, 26, 43, 66, 98, 99, V, X

SPARK SPAcecraft Recognition leveraging Knowledge of Space Environment. 14

SPE3R Satellite Pose Estimation and 3D Reconstruction. 14

SPEC Satellite Pose Estimation Competition. 11, 12, 18–20, 97

SPEED Spacecraft PosE Estimation Dataset. 10–16, 18, 20, 21, 23, 24, 26, 29, 33, 35, 36, 43, 52, 58, 62, 66, 67, 69, 71, 73, 78, 83, 86, 94, 96–98, III, IV, VI, VIII, IX

SWaP-C Size–Weight–and–Power–Cost. 8

ToF Time-of-Flight. 8

TRON Testbed for Rendezvous and Optical Navigation. 13, 16, 17, 19, VI

UKF Unscented Kalman Filter. 21

URSO Unreal Rendered Spacecraft On-orbit. 14

YOLO You Only Look Once. 9, 22, 24, 26–28, 31, 35–43, 45–47, 49–52, 57–66, 86, 87, 91, 92, 95–97, III–IX

Chapter 1

Introduction

As the space sector experiences an unprecedented expansion, maintaining a safe and sustainable orbital space has become a fundamental requirement for the success of future missions. The growing population of Resident Space Objects (RSOs), including active satellites, defunct spacecraft, launcher stages, and fragments of space debris, significantly increases the complexity and risk of operations in Low Earth Orbit (LEO).

According to the European Space Agency (ESA), more than 30,000 debris objects are currently tracked, while millions of smaller fragments remain undetected. Despite their small size, these fragments still pose a serious threat to operational spacecraft and human spaceflight activities due to the extremely high orbital velocities involved [1]. A visual representation of the current situation is illustrated in Figure 1.1, where debris are quantified and classified according to their dimensions.

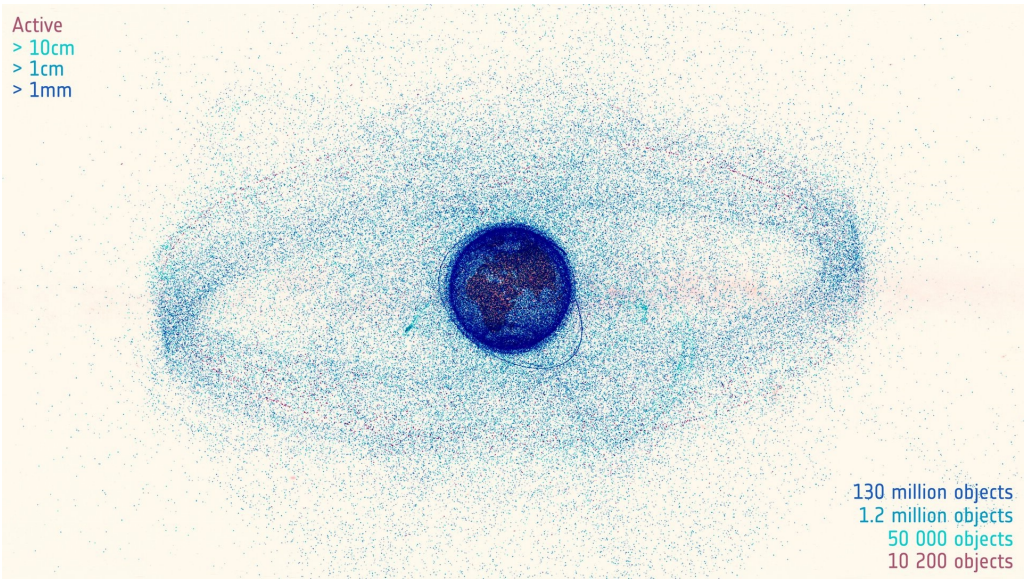


Figure 1.1: Illustration of the increasing density of space debris in LEO, with classification based on the debris dimensions. Image credit: ESA.

A well-known theory describing the long-term evolution of the orbital debris environment is the *Kessler Syndrome*. Originally proposed by Kessler and Cour-Palais, this theory describes a critical scenario in which collisions between orbiting objects generate debris faster than it can naturally decay from orbit. In such a regime, collisions produce fragments that trigger additional collisions, potentially leading to a self-sustaining cascade of debris generation [2].

Unfortunately, this scenario is no longer purely theoretical. Several major fragmentation events have already significantly altered the debris population in Earth orbit. One of the most severe incidents occurred in 2007, when the Chinese weather satellite Fengyun–1C was intentionally destroyed during an Anti-SATellite (ASAT) test. The explosion generated more than 3 000 trackable debris objects and tens of thousands of smaller fragments, increasing the catalogued debris population in LEO by roughly 30% [1], [3]. Figure 1.2 shows the expected evolution of the debris population generated by the aforementioned explosion over a period of 12 months.

A similar event occurred in 2009, when the operational Iridium-33 satellite collided with the defunct Russian spacecraft Cosmos-2251. This accidental collision produced more than 2 000 trackable debris fragments and further contributed to the congestion of critical orbital regions [4].

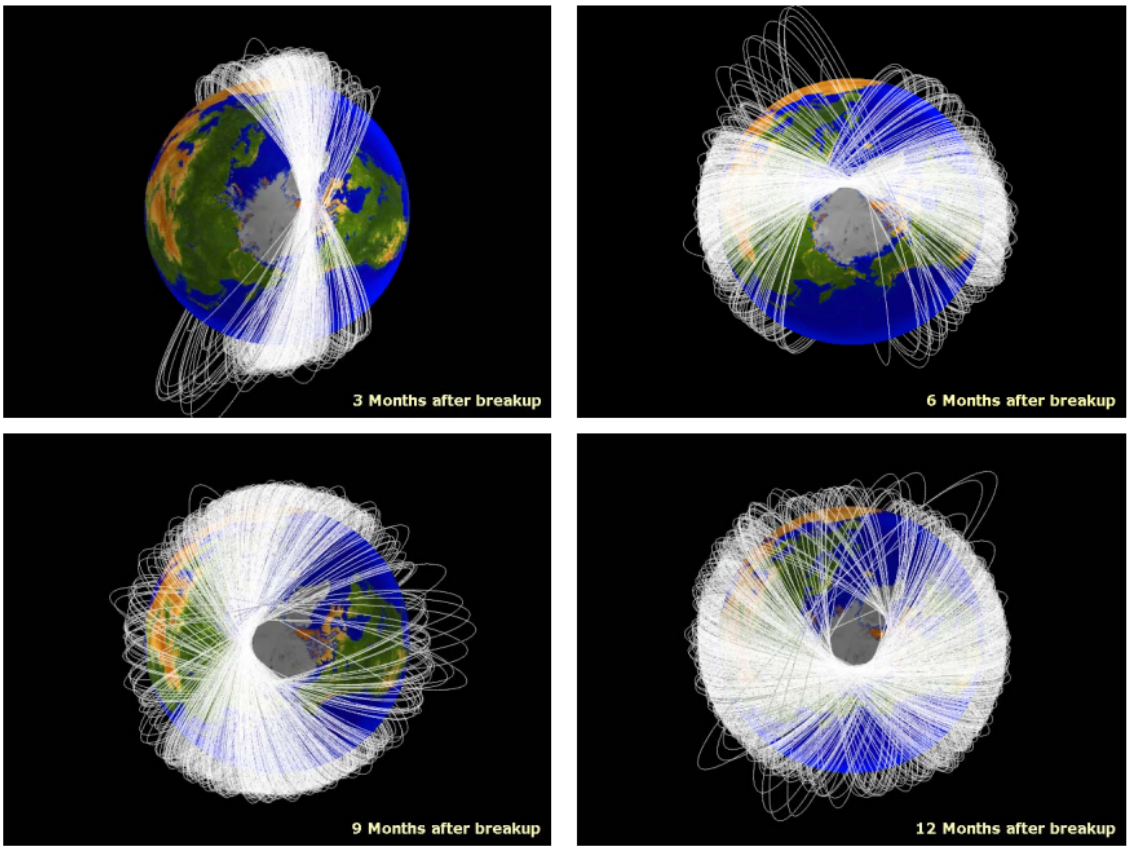


Figure 1.2: Evolution of the Fengyun-1C debris orbit planes. Credit: [3]

Beyond the technical risks, the mitigation and removal of space debris also raises complex legal and institutional challenges. Under the current international space law framework, orbital objects remain under the jurisdiction and ownership of the launching State, even after the mission has ended. This principle is mainly established by the *Outer Space Treaty* of 1967, which defines State responsibility for national space activities, and by the *Convention on International Liability for Damage Caused by Space Objects* of 1972, which regulates liability for damage caused by space objects on Earth, in the atmosphere, or in space [5], [6].

While these treaties clearly establish responsibility, they do not impose binding obligations for the active removal of defunct satellites or debris. They also do not provide a fully defined legal framework for third-party debris removal. As a result, active debris removal and on-orbit servicing operations remain legally ambiguous, particularly when dealing with non-cooperative or abandoned targets.

Because of this lack of binding regulations, most debris mitigation strategies currently rely on voluntary guidelines issued by international organizations such as the Inter-Agency Space Debris Coordination Committee (IADC) and national space agencies.

In response to these challenges, several international initiatives have recently emerged to promote sustainability of the orbital environment. Among them, the ESA *Zero Debris Approach* represents a long-term commitment to reduce debris generation and improving End-Of-Life (EOL) disposal strategies [7].

According to IADC and ESA mitigation guidelines, spacecraft operating in LEO should be removed from orbit within 25 years after the end of their operational life. This can be achieved through controlled atmospheric re-entry, natural orbital decay, or transfer to a graveyard orbit. At the same time, public–private partnerships, such as the collaboration between ESA and the Swiss start-up ClearSpace [8], are promoting the development of active debris removal missions and new in-orbit servicing technologies.

Beyond debris mitigation, another emerging strategy involves extending the operational lifetime of satellites through In-Orbit Servicing (IOS) activities. These include operations such as refuelling, component replacement, inspection, anomaly recovery, and debris removal. Such capabilities could significantly enhance mission resilience, reduce costs, and support long-duration space operations.

In all of these scenarios, accurate knowledge of the target spacecraft's relative position and attitude is essential to guarantee safe proximity operations, autonomous rendezvous, and capture manoeuvres.

1.1 Problem Definition

In spacecraft proximity operations, pose estimation refers to the determination of the full relative state between two rigid bodies: a servicing spacecraft, hereafter referred to as the *chaser*, and a cooperative or non-cooperative *target*. This state is commonly described as a 6-Degrees of Freedom (DOF) pose, comprising the relative position and relative attitude of the target with respect to a reference frame fixed to the chaser.

1.1.1 Relative pose as a rigid-body transformation

In spacecraft proximity operations, *pose estimation* refers to the determination of the relative configuration of a target spacecraft with respect to a chaser, modelled as a rigid body in three-dimensional space. From a physical and mechanical standpoint, this configuration is fully described by the relative position and relative attitude between two reference frames rigidly attached to the spacecraft.

Let $\mathcal{F}_C$ denote a reference frame fixed to the chaser spacecraft, typically aligned with the onboard camera or body frame, and let $\mathcal{F}_T$ denote a body-fixed frame attached to the target spacecraft. The relative pose of the target with respect to the chaser can be represented as a rigid-body transformation belonging to the Special Euclidean group SE(3),

$$\mathbf{T}_{C\leftarrow T} = \begin{bmatrix} \mathbf{R}_{C\leftarrow T} & \mathbf{t}_{C\leftarrow T} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{1.1}$$

where $\mathbf{t}_{C\leftarrow T} \in \mathbb{R}^3$ is the position vector of the origin of $\mathcal{F}_T$ expressed in $\mathcal{F}_C$, and $\mathbf{R}_{C\leftarrow T} \in \mathrm{SO}(3)$ is the rotation matrix that maps vectors from the target frame to the chaser frame.

The translational component $\mathbf{t}_{C\leftarrow T}$ represents the relative displacement between the two spacecrafts and evolves according to their relative orbital motion. In proximity operations, this motion is governed by the underlying orbital dynamics and relative kinematics, commonly described using linearised or nonlinear relative motion models depending on the operational regime [9]. Although the present work focuses on estimation rather than propagation, it is important to recognise that the estimated relative position constitutes a physically meaningful state variable required by guidance and control laws.

The rotational component $\mathbf{R}_{C\leftarrow T}$ describes the relative attitude of the target with respect to the chaser. Several parametrisations can be adopted to represent spacecraft attitude, including Euler angles, rotation matrices, axis–angle representations, and unit quaternions. In spacecraft Guidance, Navigation and Control (GNC)

systems, unit quaternions are widely preferred due to their global, non-singular representation and numerically stable kinematic equations [10].

Given the unit quaternion $\mathbf{q}_{C\leftarrow T} \in \mathbb{H}$

$$\mathbf{q}_{C\leftarrow T} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^\top, \qquad \|\mathbf{q}_{C\leftarrow T}\| = 1, \tag{1.2}$$

the corresponding rotation matrix $\mathbf{R}_{C\leftarrow T}$ can be expressed as

$$\mathbf{R}_{C\leftarrow T} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix}. \tag{1.3}$$

The time evolution of the relative attitude is governed by the quaternion kinematic equation

$$\dot{\mathbf{q}}_{C\leftarrow T} = \frac{1}{2}\,\boldsymbol{\Omega}(\boldsymbol{\omega}_{C\leftarrow T})\;\mathbf{q}_{C\leftarrow T}, \tag{1.4}$$

where $\boldsymbol{\omega}_{C\leftarrow T} \in \mathbb{R}^3$ is the relative angular velocity of the target with respect to the chaser, expressed in the chosen reference frame, and $\boldsymbol{\Omega}(\cdot)$ is the standard quaternion rate matrix [10]. In particular, the matrix $\boldsymbol{\Omega}(\boldsymbol{\omega})$ associated with the angular velocity vector $\boldsymbol{\omega}_{C\leftarrow T} = [\omega_x, \omega_y, \omega_z]^\top$ is defined as

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}, \tag{1.5}$$

which ensures a linear relation between the quaternion rate and the relative angular velocity [10]. Equation 1.4 highlights that the relative pose is not a static geometric quantity, but a dynamical state evolving under attitude dynamics and control torques.

In the context of proximity manoeuvres involving non-cooperative targets, the relative pose plays a central role in both navigation and control. As discussed by Zheng *et al.* [11], accurate estimation of the relative position and attitude is a prerequisite for pose-constrained guidance and tracking, collision avoidance, and observation tasks, especially in the presence of unknown target accelerations and rotational dynamics.

Within a vision-based navigation framework, pose estimation aims at recovering the instantaneous relative configuration $(\mathbf{t}_{C\leftarrow T}, \mathbf{q}_{C\leftarrow T})$. While the estimation process relies on image-derived observations, the underlying state being estimated retains a clear physical and mechanical interpretation as the relative rigid-body configuration

between two spacecrafts. This distinction is fundamental to ensure that learning-based methods can be consistently integrated within physically grounded navigation and control architectures.

1.2 Cooperative vs. Uncooperative Targets

The selection of a suitable pose-determination technique and sensor suite is strongly influenced by the specific mission scenario and by the cooperative or uncooperative nature of the target, as extensively discussed in the review by *Opromolla et al.* [12].

A target is considered *cooperative* if it is designed to aid pose estimation, for example by providing tracking signals or fiducial markers, whereas an *uncooperative* target provides no such support.

1.2.1 Cooperative Targets

For cooperative targets, the chaser can exploit additional information or hardware on the target to simplify pose estimation. Cooperation can be *active* or *passive* [12]. In active cooperation, the target actively communicates or provides navigation signals, for example carrying Radio Frequency (RF) transceivers or Global Positioning System (GPS) receivers to share relative navigation data. The Prototype for Rendezvous and Autonomous Mission Operations (PRISMA) formation-flying mission demonstrated this by using differential GPS between two satellites to achieve centimetre-level relative positioning [13]. In passive cooperation, the target is equipped with known visual markers or retro-reflectors that the chaser's vision sensors can easily detect. Examples include Light Emitting Diodes (LED) beacons or corner-cube reflectors arranged in a specific pattern on the target [14], [15].

Several in-orbit demonstrations have validated cooperative pose determination. Notably, Japan's ETS-VII mission (1997–1999) achieved autonomous rendezvous and docking using a prepared target satellite with a set of passive cooperative markers [16]. Similarly, the Defense Advanced Research Projects Agency DARPA/NASA Orbital Express mission (2007) successfully performed autonomous rendezvous, docking, and even orbital servicing operations between two spacecraft designed to be cooperative. The Orbital Express target module carried retro-reflective markers for the chaser's Advanced Video Guidance Sensor (AVGS), enabling precise range and attitude measurements for docking [15]. The PRISMA mission also tested vision-based and RF-based relative navigation in a cooperative two-satellite formation, demonstrating reliable pose estimation when the target actively or passively cooperates [13]. Thanks to such cooperative aids, pose determination algorithms can be

simpler and more efficient, and their performance has reached operational maturity in these scenarios [12].

1.2.2 Uncooperative Targets

Uncooperative targets provide no helpful telemetry or markers to the chaser, making pose determination significantly more challenging [12]. There are two broad cases: *known* vs. *unknown* uncooperative targets. A *known* uncooperative target is one whose physical characteristics (shape, size, etc.) are at least partially known in advance (for instance, a defunct satellite for which a CAD model or imagery exists). An *unknown* target has no reliable prior model (e.g., an arbitrary piece of space debris or a small asteroid). In both cases, pose estimation results to be critical since the target does not aid the relative navigation [12]. For this reason, non-cooperative targets' pose estimation must rely solely on passive sensors without any fiducial markers or transmitted position data [17], due to the fact that no inter-satellite link or cooperative signal is available.

For uncooperative *known* targets, a common strategy is *model-based pose estimation*. In this approach, onboard algorithms are provided with a 3D model or shape of the target, and they compute the relative pose by matching real sensor data (images or point clouds) to the known model [18]. Various techniques, such as edge matching, template matching, or Iterative Closest Point (ICP) on Light Detection and Ranging (LiDAR) data, are used to match the observed features to the target model. The advantage of model-based methods is accuracy and observability, as all six pose parameters can be recovered if the target geometry is known. However, their performance can degrade if the actual appearance of the target deviates from the model (e.g. due to damage or deployed appendages not in the model). Real-world tests and simulations (e.g. using known satellite images) have shown model-based techniques achieving degree-level attitude accuracy and a few percent of range in position error under favourable conditions. Continued research is addressing improvements in speed and robustness for on-board use, since matching against a complex model can be computationally intensive.

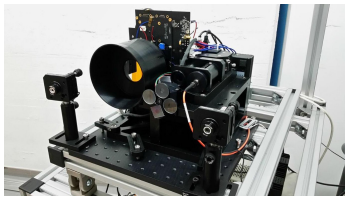
For uncooperative *unknown* targets, pose determination must be accompanied by some form of simultaneous target mapping or model update. One approach is to employ Simultaneous Localization and Mapping (SLAM) techniques [19]. Here, the chaser uses its sensor data over time to progressively build a 3D model (a point cloud or mesh of the target) while simultaneously estimating its relative pose. This was the case for NASA's OSIRIS-REx mission to asteroid Bennu, where the spacecraft first conducted extensive surveys to build a shape and feature map of the target, which was then exploited for optical navigation and guidance [20]. Such SLAM or model-

free approaches are critical when confronting a completely unknown object, but they face challenges with reliability and drift. Researchers are therefore exploring hybrid methods that first perform a coarse model construction and then switch to model-based tracking on the generated model.

Recent and ongoing missions underscore the need for uncooperative pose determination capabilities. The Double Asteroid Redirection Test DART mission targeted the Didymos asteroid moon Dimorphos, which provided no navigation aids. The DART spacecraft successfully navigated to impact in 2022 by using onboard cameras and autonomous guidance, essentially homing in on the image of the target [21]. Its success demonstrated reliable real-time image-based relative navigation toward a small, unknown natural object. Likewise, missions like ESA's planned active debris removal or NASA's Restore-L servicing will require vision-based pose estimation of defunct satellites that were not originally designed to be approached [12].

1.3 Sensors for Spacecraft Pose Estimation

Depending on mission constraints and operational range, spacecraft pose estimation can rely on active or passive sensors, each characterised by specific advantages and limitations [12]. Some examples, discussed below, are shown in Figure 1.3.



(a) LiDAR sensor
Image credit: ESA

(b) Stereo vision camera
Image credit: DLR (High Resolution Stereo Camera (HRSC), Mars Express)

(c) Monocular camera
Image credit: Dragonfly Aerospace

Figure 1.3: Examples of sensors used for relative navigation purposes.

Active sensors, such as LiDAR, radar, or Time-of-Flight (ToF) cameras, directly measure range and can provide dense 3D point clouds of the target [12]. These sensors offer high accuracy and observability even under challenging illumination. However, their adoption is often limited by stringent SWaP-C requirements, restricted operational range, and increased system complexity. As a results, active sensors are typically reserved for short-range operations or larger spacecraft platforms.

Stereo vision systems represent an intermediate solution, exploiting geometric triangulation between two cameras to recover depth [22]. While stereo systems can

achieve accurate relative pose estimation, they require precise calibration, synchronised acquisition, and sufficiently large baselines, which may be incompatible with specific mission scenarios.

Monocular vision systems, on the other hand, are lightweight, low-power, and mechanically simple, making them well suited for small satellites and deep-space missions [12]. For these reasons, this solution has emerged as a particularly attractive sensing modality for proximity operations.

However, monocular pose estimation is inherently poorly posed due to the lack of direct depth measurements, requiring advanced estimation algorithms to infer the 6-DOF relative state from a single image.

Recent advances in computer vision and machine learning have significantly improved the feasibility of monocular pose estimation, enabling accurate relative navigation even in the absence of cooperative markers or active ranging sensors. As a result, monocular vision has become a cornerstone of modern research on autonomous rendezvous and proximity operations.

1.4 Scope and Contributions of This Work

Within the context outlined above, this thesis focuses on monocular, vision-based pose estimation of an uncooperative target for proximity operations between satellites.

The main objectives of this work are the following.

First, a comprehensive analysis of the state of the art in spacecraft pose estimation is presented. This includes a structured comparison between direct, indirect, hybrid, and model-free approaches, as well as an overview of the most relevant public datasets and benchmarking works.

Second, a modern keypoint-based pose estimation pipeline based on YOLOv11 is developed and trained. The pipeline is designed to operate on the dataset images and to infer 2D keypoint locations, which are subsequently used by a geometry-based Perspective-n-Point (PnP) solver to recover the full 6-DOF relative pose. Particular attention is devoted to robustness across domain shifts between synthetic and laboratory-based imagery, confidence-threshold selection, and runtime trade-off between accuracy and reliability.

Third, a dedicated CNN architecture for dense 2D keypoint detection is designed, implemented, and analysed. The proposed model is based on a high-resolution backbone network and predicts spatially consistent keypoint heatmaps rather than direct coordinate regressions. Different architectural configurations are investigated through a systematic ablation study, including alternative loss formulations, aug-

mentation strategies, and pre- and post-processing refinements.

The performance of the proposed pipelines is quantitatively evaluated on the Spacecraft Pose Estimation Dataset (SPEED+) under multiple illumination conditions and operating regimes. Evaluation is conducted using the official SPEED+ metrics, enabling direct comparison with existing approaches and highlighting the trade-offs between detection reliability, pose accuracy and computational efficiency.

By systematically addressing both methodological and practical aspects of monocular spacecraft pose estimation, this work aims to contribute to the development of reliable, interpretable and flight-ready navigation solutions for future proximity operations between satellites.

Chapter 2

State of the Art

2.1 Overview

This chapter aims to review the state of the art in vision-based spacecraft pose estimation, with particular focus on monocular approaches developed for proximity operations involving cooperative and non-cooperative targets. The objective of this study is to provide a structured overview of existing methodologies, highlighting their underlying assumptions, architectural choices, and performance characteristics under realistic spaceborne conditions.

The literature on spacecraft pose estimation spans over a wide range of techniques, from model-based image processing pipelines to modern deep learning approaches. In recent years, the increasing availability of synthetic datasets and Hardware-in-the-Loop (HIL) benchmarks has accelerated the adoption of deep learning methods, particularly for feature extraction. At the same time, classical geometric solvers and recursive estimation frameworks remain central components for enforcing physical consistency and temporal coherence.

To organise this diverse body of work, this chapter aims to provide a methodological classification between different methods. Specifically, vision-based approaches are grouped into *direct*, *indirect*, and *hybrid* pipelines, depending on whether the neural network directly regresses the pose, provides intermediate geometric measurements, or is integrated within a dynamical filtering framework.

Following this classification, this chapter will review representative methods from the literature, discussing their architectural design, training strategies, and robustness to illumination variability, partial occlusions, and domain shifts between synthetic and real imagery. Commonly used dataset will also be analysed in this chapter, with particular attention to the SPEED+ dataset and the metrics defined by the associated Satellite Pose Estimation Competition (SPEC).

This structured overview will provide the right context in which the contribution of this thesis is located and helps motivating the design choices adopted in the proposed pipelines.

2.2 Datasets for CNN-based Spacecraft Pose Estimation

The performance, robustness, and generalisation capability of CNN-based monocular pose estimation methods are essentially linked to the availability and characteristics of suitable training and evaluation datasets. Due to the limited accessibility of labelled spaceborne imagery, existing datasets are predominantly constructed using synthetic rendering pipelines, HIL facilities, or a combination of both.

Beyond the data generation process, an important distinction can be made in terms of the role played by each dataset in the literature.

In this section, datasets are therefore classified according to their primary scientific purpose.

On the one hand, *benchmark datasets* are designed for instance-level pose estimation on a fixed target spacecraft, enabling systematic and reproducible comparisons between different methods.

On the other hand, a growing class of *generalisation-oriented datasets* and simulation frameworks aims at evaluating robustness across varying illumination conditions and unknown target geometries, addressing the limitations related to instance-level benchmarks.

Table 2.1 provides a comparative overview of the most relevant datasets for spacecraft pose estimation discussed in this section.

2.2.1 Benchmark datasets for instance-level pose estimation

Benchmark datasets constitute the backbone of quantitative evaluation for CNN-based spacecraft pose estimation. They are typically characterised by a single target spacecraft with known geometry, accurate pose annotations, and well-established evaluation protocols.

The SPEED dataset represents the first large-scale benchmark for monocular spacecraft pose estimation [23]. It focuses on a single target spacecraft, Tango, from the PRISMA mission, and consists primarily of synthetic images annotated with full 6-DOF pose labels. SPEED served as the reference dataset for the SPEC2019 challenge, enabling a systematic comparison of CNN-based regression and keypoint-driven approaches under a unified evaluation framework.

Dataset	Year	#Targets	Type	Scale	Generation	Annotations
SPEED [23]	2019	1	Individual	∼15k	Synthetic	6D pose, 3D model
SPEED+ [24]	2021	1	Individual	∼70k	Synthetic + HIL	6D pose, 3D model
URSO [25]	2019	2	Individual	∼15k	Synthetic	6D pose, 3D model
Cygnus [26]	2021	1	Individual	∼20k	Synthetic	6D pose, 3D model
SwissCube [27]	2021	1	Individual	∼50k	Synthetic	6D pose, 3D model
SPARK-T [28]	2021	3	Sequential	7.5k	Synthetic	6D pose, 3D model, 3D trajectories
SPARK 2022 [29]	2022	1	Sequential	30k	Synthetic	6D pose, 3D model, 3D trajectories
SHIRT [30]	2023	1	Sequential	9.6k	Synthetic + HIL	6D pose, 3D model, 3D trajectories
SPE3R [31]	2024	64	Individual	∼64k	Synthetic (Unreal Engine)	6D pose, 3D model, binary mask
G-SPET [32]	2025	174	Sequential	∼14M	Blender simulation	6D pose, 3D model, 3D trajectories, mask, depth
ENVISAT [33], [34]	2025	1	Individual	20k	Synthetic (Blender) + HIL	6D pose, 3D model

Table 2.1: Summary of publicly available datasets for spacecraft pose estimation and tracking.

As highlighted in [35], models trained exclusively on synthetic imagery are affected by a pronounced domain gap, as neural networks tend to overfit visual patterns that are specific to synthetic training. This mismatch in data distributions leads to a systematic degradation in performance when the models are deployed on real images or domains that differ significantly from the training set [30], [36].

Building on this foundation, SPEED+ extends the original dataset in terms of scale and realism while preserving the fixed-target assumption [24]. In addition to a substantially larger synthetic training set, SPEED+ introduces two independent HIL test domains acquired in the Testbed for Rendezvous and Optical Navigation (TRON) facility under different illumination conditions. This design aims at assessing robustness across the synthetic-to-real domain gap while maintaining a controlled and well-defined target geometry. A detailed description of the SPEED+ dataset and its experimental design is provided in the Section 2.3.

While SPEED and SPEED+ focus on single image pose estimation, the SHIRT dataset extends the benchmark paradigm to the temporal domain [30].

SHIRT consists of sequential image data acquired along representative rendezvous trajectories in LEO, generated using both an OpenGL-based rendering pipeline and the TRON HIL facility. By organising samples as temporally ordered sequences rather than independent images, Satellite Hardware-In-the-loop Rendezvous Trajectories (SHIRT) enables the evaluation of the target pose based on CNN when embedded within recursive navigation filters, supporting the analysis of temporal consistency, convergence behaviour, and robustness in closed-loop tracking scenarios.

Additional datasets, such as SwissCube [27], Cygnus[26], and SPAcecraft Recognition leveraging Knowledge of Space Environment (SPARK) [28], have been proposed to address specific targets or sensing conditions. However, their adoption as standard benchmarks remains limited compared to SPEED and SPEED+.

Overall, benchmark datasets provide accurate annotations, realistic imaging artefacts, and well-defined evaluation protocols. However, their applicability is inherently limited by the small number of target spacecraft and the implicit assumption that the target geometry is known and fixed during both training and inference.

2.2.2 Generalisation-oriented datasets

To overcome the limitations of instance-level benchmarks, a complementary class of datasets and simulation frameworks has been developed to evaluate robustness, scalability, and generalisation across different targets and observation conditions. Rather than serving as fixed benchmarks, these resources are mainly intended as experimental platforms to investigate domain gap effects, cross-object generalisation, and model independence.

A representative example of this paradigm is Unreal Rendered Spacecraft On-orbit (URSO), which is best interpreted as a configurable visual simulation framework rather than a static dataset [25]. Built on Unreal Engine 4, URSO enables the generation of photorealistic synthetic images and depth masks by varying spacecraft geometry, relative pose, camera viewpoint, and illumination conditions. In the reference experimental setup introduced by Proença and Gao [25], several spacecraft-specific datasets were configured, including *Soyuz-easy*, *Soyuz-hard*, and *Dragon-hard*, each comprising 5,000 labelled images with controlled levels of geometric and photometric complexity. While URSO also includes spaceborne imagery of the same spacecraft classes, the absence of pose annotations limits their use to qualitative analysis and domain gap assessment. As such, URSO serves primarily as a dataset generator and experimentation platform rather than a universal benchmark.

A more explicit focus on cross-object generalisation is adopted by the SPE3R dataset [31]. Satellite Pose Estimation and 3D Reconstruction (SPE3R) comprises 3D models and corresponding rendered images of 64 different satellites and spacecraft, covering a wide range of structural configurations. For each target, synthetic RGB images and associated segmentation masks are generated using a high-fidelity Unreal Engine rendering pipeline, with precise control over viewpoint, illumination, and background appearance. Crucially, the dataset partitioning strategy enforces generalisation by evaluating models on images depicting spacecraft geometries that are not observed during training, enabling a systematic assessment of cross-target performance.

Recently, a significant step towards large-scale generalisable evaluation has been introduced with the Generalizable Spacecraft Pose Estimation and Tracking (G-SPET) dataset [32]. G-SPET is the first dataset explicitly designed to support training and evaluation of generalisable spacecraft pose estimation and tracking methods. It includes 174 distinct spacecraft models, increasing the number of available targets by nearly two orders of magnitude with respect to existing benchmarks. The dataset consists of sequential image data rendered using Blender under a wide range of conditions, including varying illumination, background (Earth and star fields), imaging distance, and target attitude dynamics. Annotations are provided for each frame, including a 6-DOF pose, 3D models, depth maps, segmentation masks, and 3D trajectories. By combining scale, target diversity, and temporal structure, G-SPET enables a systematic evaluation of both instance-level and generalisable pose estimation methods within a unified framework.

A complementary line of work focuses on mission-driven datasets centred on the Environmental Satellite (ENVISAT), which has been extensively adopted as a reference target for active debris removal missions and proximity operations. In [33], large-scale synthetic datasets of ENVISAT are generated using Blender. The rendering pipeline explicitly models the spacecraft geometry and surface textures, realistic Earth backgrounds derived from the Blue Marble collection, star fields, and physically consistent solar illumination. The main training and validation dataset comprises 20,000 synthetic images, densely sampling relative position and attitude spaces, while additional test datasets are generated to evaluate acquisition, tracking, and navigation performance along realistic orbital trajectories. Beyond synthetic imagery, ENVISAT has also been employed in HIL validation campaigns using scaled mock-ups and real imaging hardware [34], providing physically realistic test data primarily intended for robustness assessment rather than for large-scale supervised training.

In summary, generalisation-oriented datasets and simulation frameworks provide unmatched scalability and flexibility, enabling accurate studies on robustness, domain gaps, and multi-object detection performance. However, their reliance on rendered imagery and the absence of standardised evaluation metrics often limit their direct use as benchmarks.

For these reasons, the SPEED+ benchmark dataset was adopted in the current work, as it allows a direct and fair comparison with existing methods in the literature while explicitly targeting the domain shift problem.

2.3 SPEED+ dataset

Introduced by Park *et al.* [24], SPEED+ extends the previous SPEED dataset [35] by essentially increasing the number and diversity of HIL images, while maintaining a large synthetic set labelled for supervised training. Therefore, the SPEED+ dataset is explicitly designed to support robustness studies across the synthetic-to-real domain gap.

SPEED+ depicts the Tango spacecraft from the PRISMA mission and targets relative 6-DOF pose estimation from a single monocular image. Each sample is associated with the camera intrinsics and a relative pose label (translation and attitude) defined with respect to the camera frame and the known spacecraft model.

A distinctive feature of SPEED+ is the presence of three image domains originating from two fundamentally different generation pipelines: a labelled `synthetic` domain intended for training and validation, and two unlabelled HIL domains, `lightbox` and `sunlamp`, intended for testing under domain shift. The split follows the assumption that the labelled images are not available prior to in-orbit deployment.

Split	**Synthetic**	**Lightbox**	**Sunlamp**
Train	47,966	–	–
Validation	11,994	–	–
Test	–	6,740	2,791
Total	59,960	6,740	2,791

Table 2.2: Dataset composition and splits for SPEED+.

The `synthetic` domain comprises 59,960 images rendered using Stanford Space Rendezvous Laboratory SLAB's OpenGL-based Optical Stimulator (OS) camera emulator within the multi-satellite software simulator (S^3). These images are provided with pose labels and serve as the primary source for supervised learning. The `synthetic` split is organised as an 80:20 train/validation partition.

To create representative non-synthetic test domains, SPEED+ leverages the TRON robotic facility in SLAB. TRON is an $8 \times 3 \times 3$ m simulation room equipped with two 6-DOF KUKA robot arms: one holds a reduced-scale mock-up of the target spacecraft while the other carries the camera along a ceiling-mounted rail.

The reduced-scale mock-up of the Tango spacecraft is manufactured with two mounting interfaces located on opposite sides of the structure, allowing the model to be mounted in alternative configurations to ensure that the supporting robot arm does not occlude the target during image acquisition. As a result, the dataset avoids

artificial occlusions that could otherwise bias the visual appearance of the spacecraft and compromise the validity of the pose estimation task.

All HIL images are captured using a Point Grey Grasshopper 3 camera paired with a Xenoplan 1.4/17 mm lens. The camera is calibrated prior to acquisition, and the dataset release includes the corresponding intrinsic parameters and distortion coefficients in the provided `camera.json` file.

The true pose label is recovered through a calibrated fusion of robot telemetry and an external motion-capture system (12 Vicon Vero cameras) tracking infrared markers, enabling millimetre-level position and sub-degree attitude accuracy at close range.

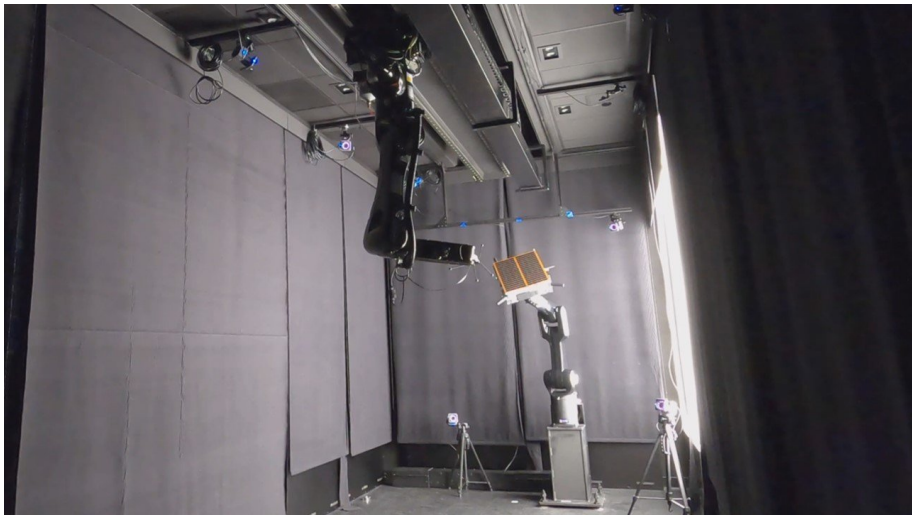


Figure 2.1: Visualisation of the TRON facility at SLAB

To emulate realistic spaceborne illumination conditions, the TRON facility is equipped with a dedicated lighting system designed to reproduce both diffuse Earth albedo and direct solar illumination. The setup includes ten Earth-albedo lightboxes, which generate diffuse environmental lighting representative of typical low-Earth orbit conditions, as well as a high-intensity sunlamp system to simulate direct sunlight.

The albedo lightboxes are carefully designed and calibrated to provide a uniform maximum radiance of approximately 14 $\mathrm{W/m^2sr}$, corresponding to the mean radiance reflected by the Earth assuming an albedo coefficient of 0.3 and a solar irradiance of 1366 $\mathrm{W/m^2}$. The sunlamp consists of a metal-halide arc lamp coupled with a paraboloidal mirror, producing a collimated beam with an intensity equivalent to one solar constant in space (nominally 1357 $\mathrm{W/m^2}$) and a spectral distribution close to a blackbody temperature of 6000 K.

During data acquisition, the illumination conditions are deliberately varied across samples. In the `lightbox` domain, the intensity of each activated albedo light-

box is randomly drawn from a distribution representative of different orbital altitudes, thereby introducing realistic variability in diffuse lighting. Conversely, in the `sunlamp` domain, the sunlamp operates at a fixed intensity and orientation, facing the spacecraft mock-up to generate strong directional illumination, pronounced shadows, and saturation effects. The results are shown in Figure 2.2.

Compared to purely computer-generated imagery, the resulting HIL images naturally retain complex optical artefacts such as stray light, surface glow, and self-shadowing effects, which are difficult to reproduce accurately through rendering alone. These characteristics make the HIL domains particularly valuable for evaluating the generalisation capability of pose estimation algorithms across the synthetic-to-real domain gap.

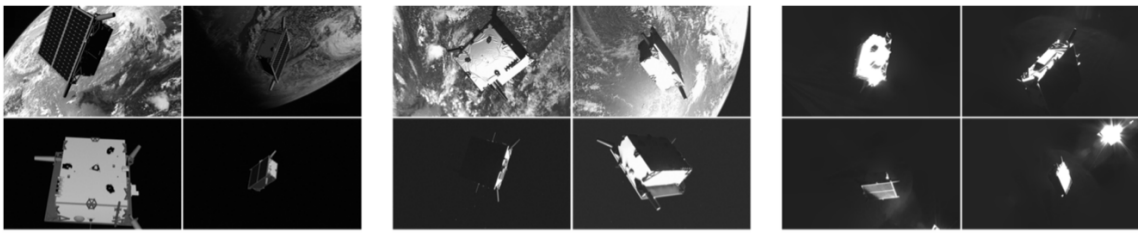


Figure 2.2: Representative samples from the three image domains of the SPEED+ dataset.

2.3.1 Evaluation Metrics

The performance of the proposed monocular pose estimation pipeline is evaluated using the standard metrics defined for SPEED+ and adopted in the ESA SPEC. These metrics have been specifically designed to reflect the requirements of non-cooperative spacecraft rendezvous and proximity operations, where both translational and rotational accuracy are critical for mission safety and guidance performance [24].

Let $\mathbf{t}_{gt} \in \mathbb{R}^3$ and $\mathbf{t}_{pred} \in \mathbb{R}^3$ denote the ground-truth and the predicted relative position vectors of the target spacecraft expressed in the camera frame, respectively. Similarly, let $\mathbf{q}_{gt}$ and $\mathbf{q}_{pred}$ denote the corresponding unit quaternions representing the relative orientation, expressed in the (w, x, y, z) convention. The following metrics are employed.

Absolute Translation Error

The absolute translation error is defined as the Euclidean distance between the estimated and ground-truth relative positions:

$$E_t = \|\mathbf{t}_{gt} - \mathbf{t}_{pred}\|_2 \tag{2.1}$$

This metric, expressed in metres, provides a direct measure of the positional accuracy of the estimated pose.

Normalized Translation Error

To account for the wide range of relative distances encountered in rendezvous scenarios, a normalised translation error is also considered:

$$e_t = \frac{\|\mathbf{t}_{gt} - \mathbf{t}_{pred}\|}{\|\mathbf{t}_{gt}\|} \tag{2.2}$$

This dimensionless quantity enables a scale-invariant comparison of translational errors across different relative ranges.

Orientation Error

The rotational accuracy is quantified using the angular distance between the estimated and ground-truth quaternions:

$$e_q = 2 \arccos\left(|\langle \mathbf{q}_{gt}, \mathbf{q}_{pred} \rangle|\right) \tag{2.3}$$

where $\langle \cdot, \cdot \rangle$ denotes the quaternion inner product. This formulation accounts for the double-cover property of quaternions by considering the absolute value of the dot product. The resulting orientation error e_q is expressed in radians.

ESA Composite Score

Following the evaluation protocol of the SPEC2019 challenge, a composite mean score is computed by combining the normalised translation and orientation errors.

$$E_{pose} = \frac{1}{N} \sum_{i=1}^{N} \{e_q^i + e_t^i\} \tag{2.4}$$

In the SPEC2021 challenge, the calibration process of the TRON facility was revised to account for errors arising in the determination of pose labels for HIL images. Specifically, threshold values $\tau_t = 0.002173$ for the normalized translation error and $\tau_q = 0.169°$ for the orientation error are defined. The corresponding penalised components are given by:

$$s_t = \begin{cases} 0, & e_t < \tau_t, \\ e_t, & \text{otherwise} \end{cases} \qquad s_q = \begin{cases} 0, & e_q < \tau_q, \\ e_q, & \text{otherwise} \end{cases} \tag{2.5}$$

The final score, in this thesis referred as SLAB, is computed as in 2.4:

$$E^*_{\text{pose}} = \text{SLAB} = \frac{1}{N}\sum_{i=1}^{N}\{s_t^i + s_q^i\} \tag{2.6}$$

Lower values of the ESA score indicate better overall pose estimation performance. This composite metric emphasises both translational and rotational accuracy and reflects the operational constraints of autonomous rendezvous missions.

Note. SPEC2019 introduced the benchmark, but its evaluation protocol was not yet fully standardised. SPEC2021 provides a clearer and more consistent definition of pose error aggregation, which is the reason for its adoption as the primary reference for quantitative comparison in this thesis. All quantitative comparisons in this thesis follow the official SPEC2021 evaluation protocol to ensure consistency and reproducibility.

2.3.2 Positioning of SPEED+

Within this context, the SPEED+ dataset plays a central role, as it effectively combines purely synthetic datasets and laboratory-based test setups, offering a benchmark that is both controlled and representative of real in-orbit scenarios. This makes it suitable for the evaluation of CNN-based pose estimation methods.

For this reason, the following State Of The Art (SOTA) analysis focuses mainly on the comparison based on the results of the performance metrics achieved on the SPEED+ dataset. Moreover, SPEED+ dataset is also adopted in this work for training, validation, and testing, exploiting its structured synthetic-to-real setting to develop and assess a reliable CNN-based spacecraft pose estimation pipeline.

2.4 Vision-based Pose Estimation Pipelines

2.4.1 Direct, Indirect, and Hybrid Methods

Monocular spacecraft pose estimation approaches can generally be divided into three main categories: *direct*, *indirect*, and *hybrid* pipelines. This classification is widely adopted in the literature on vision-based navigation [37], [38].

Direct approaches formulate pose estimation as an end-to-end learning problem, in which a neural network maps an input image directly to a 6-DOF relative pose of the target [37]. In this formulation, the network implicitly learns both the visual information and the geometric relationship between the image and the relative pose. While this approach leads to conceptually simple pipelines and fast inference, several

studies report limited robustness to illumination changes, partial occlusions, and domain shift. In addition, the absence of an explicit geometric reconstruction stage may reduce interpretability and physical consistency of the estimated pose [33], [39].

Indirect approaches instead decouple visual perception from pose recovery. In these pipelines the neural network estimates intermediate geometric features, most commonly the 2D projections of known 3D landmarks on the spacecraft. The pose is then recovered by solving a geometric inverse problem, typically through a PnP formulation [38], [39]. By explicitly enforcing projective geometry, these methods combine the robustness of learned feature extraction with the reliability of analytical pose solvers. For this reason, keypoint-based pipelines currently represent the dominant paradigm in spacecraft pose estimation benchmarks such as SPEED and SPEED+ [24].

Finally, *hybrid* approaches integrate vision-based measurements within recursive estimation frameworks, such as Extended (EKFs) or Unscented Kalman Filters (UKFs). In these architectures the neural network provides pose estimates or geometric measurements, while the filtering stage exploits temporal information to enforce dynamical consistency and mitigate noise or outliers [30]. Recent works further explore uncertainty-aware formulations, where the confidence associated with the visual measurements is used to adaptively weight the observations within the filtering process [17].

Across all methodological families, the key challenge for operational deployment remains the domain gap between synthetic training images and realistic spaceborne observations. Addressing this issue typically requires dedicated strategies such as data augmentation, HIL validation, or domain adaptation techniques, which are discussed in detail in the following sections.

2.4.2 Pipeline Architectures

The majority of SOTA approaches adopt an indirect formulation based on keypoint detection. In these pipelines the pose estimation process is decomposed into three main stages: spacecraft detection and Region of Interest (ROI) extraction, 2D keypoint localisation, and geometric pose recovery through a PnP solver. This modular structure allows the learning process to focus on robust feature extraction, while the final pose estimate is obtained through a geometric formulation.

Several works enhance the robustness of this pipeline by introducing mechanisms such as outlier rejection, confidence-based keypoint selection, or non-linear pose refinement to reduce the sensitivity of PnP solvers to incorrect correspondences [39], [40]. In addition, some methods interpret the spatial distribution of predicted keypoint heatmaps as a measure of uncertainty, allowing this information to be

included in the pose estimation process [17], [41].

Within this framework, different architectural solutions have been proposed. High-resolution convolutional networks are widely employed for keypoint localisation because of their ability to preserve fine spatial information during feature extraction. Architectures such as HRNet maintain high-resolution feature maps throughout the network while processing multiple resolutions in parallel. This design enables continuous information exchange across scales and has proven particularly effective for landmark detection tasks [37], [39], [42], [43], [44].

For direct pose regression, architectures emphasising global feature aggregation and stable gradient flow are typically preferred. DenseNet-based encoders have been proposed to regress translation and attitude directly from monocular images, leveraging dense connectivity to improve information propagation and parameter efficiency [45].

Lightweight detectors inspired by human pose estimation frameworks have gained attention for real-time applications. In these models, target detection and keypoint regression are performed jointly within a single network, reducing preprocessing and post-processing stages. YOLO-based pose models are representative examples of this class of architectures and are particularly attractive for embedded or onboard implementations, where the pose estimation stage must be performed under limited computational resources [38], [40].

More recently, transformer-based architectures have been investigated to improve robustness under domain shift. Vision Transformer backbones have shown competitive performance on HIL datasets while maintaining favourable computational efficiency for onboard deployment [41].

Finally, alternative paradigms address scenarios in which a geometric model of the target spacecraft is not available. In such cases Neural Radiance Fields (NeRF) can be used to learn an implicit representation of the target from a limited set of images and subsequently generate synthetic datasets with known pose labels, enabling the training of conventional pose estimation networks even in the absence of a CAD model [46].

Overall, the different pipeline designs reflect a trade-off between geometric interpretability, robustness to realistic operational conditions, and computational complexity. Among these paradigms, keypoint-based pipelines provide a favourable balance between geometric consistency and estimation accuracy, which explains their common adoption in pose estimation benchmarks and motivates the architectural choices adopted in this work.

2.4.3 Performance Comparison

Table 2.3 summarises the quantitative performance reported in the literature for representative vision-based spacecraft pose estimation methods evaluated on the SPEED+ dataset. The objective of this comparison is to provide a reference for the accuracy levels currently achievable by SOTA pipelines under different evaluation settings.

Reference	Backbone	Dataset type		E_q [°]	E_t [m]	SLAB [-]	#Param [M]
SPEC evaluation metrics							
[43]	HRNet + BiFPN	Lightbox		4.32	**0.09**	0.09	–
		Sunlamp		6.94	0.14	0.14	
	Lite-HRNet	Lightbox		7.42	0.14	0.15	–
		Sunlamp		15.12	0.20	0.30	
[39]	HRNet	Lightbox		**3.187**	0.105	**0.073**	-
		Sunlamp		4.299	0.086	0.090	
[44]	HRNet-w48	Lightbox		6.665	0.302	0.165	7.8
		Sunlamp		**2.728**	**0.065**	**0.059**	
[47]	SPNv2	Lightbox		5.577	0.15	0.122	22.7
		Sunlamp		9.788	0.161	0.197	
[46]	SPNv2	HIL	Lightbox	9.5	0.25	0.203	–
			Sunlamp	15.7	0.30	0.322	
		NeRF	Lightbox	7.2	0.20	0.158	
			Sunlamp	16.9	0.29	0.341	
[40]	YOLOv8-pose	Synthetic		1.4	0.095	0.033	11.5
		Lightbox		18.0	0.7583	0.4316	
		Sunlamp		19.8	0.5177	0.4318	
Non-uniform metrics							
[37]	HRNet	Synthetic		0.438	0.026	-	30.7
		Lightbox		2.95	0.014*	0.065	
		Sunlamp		2.74	0.011*	0.059	
[41]	SPNv3-B	Lightbox		2.033	0.012*	0.047	86.3
		Sunlamp		3.400	0.015*	0.074	
[45]	DenseNet	Synthetic		2.79	0.096	–	13.25
		Lightbox		14.41	0.39	–	
[42]	HRNet	Lightbox		< 10	< 0.1*	–	–
		Sunlamp		< 13	< 0.1*	–	

* These quantities refer to the normalised translation error e_t.

Table 2.3: Pose estimation results on the SPEED+ dataset reported in the literature under different dataset configurations and evaluation protocols. The best results for each metric on the `lightbox` and `sunlamp` split are highlighted in red and blue, respectively.

It is important to note that, although all the listed works rely on the same underlying dataset, the reported results are not always directly comparable. In

particular, some approaches follow the official evaluation protocol defined by the SPEED+ benchmark, while others report alternative error definitions. For this reason, the absolute performance values reported in Table 2.3 should be interpreted primarily as an indication of general performance trends rather than as a strict ranking between methods.

Among the methods evaluated on the SPEED+ dataset, approaches relying on high-resolution keypoint-based pipelines consistently achieve lower median rotation errors with respect to fully end-to-end regression models. In particular, architectures based on HRNet backbones exhibit strong performance across multiple evaluation settings, both in terms of rotational and translational accuracy and stability across different domains.

These results suggest that preserving high-resolution spatial representations during feature extraction is a key factor in achieving accurate and stable keypoint localisation. Since the final pose is recovered through a geometric solver, small localisation errors in the image plane can propagate into larger pose estimation errors. High-resolution feature representations therefore play a critical role in reducing this error amplification effect.

The quantitative evidence reported in the literature therefore supports the adoption of indirect pipelines that decouple visual perception from pose recovery while maintaining high-resolution spatial features. These considerations motivate the architectural choices adopted in this work.

2.5 Positioning of This Work

Within the methodological landscape reviewed in this chapter, the present thesis positions itself within the class of *indirect* vision-based pose estimation approaches.

The first part of the work investigates a keypoint-based indirect pipeline built upon a YOLOv11 backbone, in which target detection and 2D keypoint regression are performed jointly within a single convolutional network. The estimated keypoints are subsequently used in a model-based reconstruction stage to recover the full 6-DOF relative pose through a PnP solver. This pipeline is not intended as the final solution proposed in the thesis, but rather as an exploratory framework for developing a comprehensive understanding of learning-based pose estimation. In addition, the YOLOv11-based pipeline is particularly appealing due to its low computational requirements and real-time performance.

Building on the knowledge gained from this initial investigation, the core contribution of the thesis is a second, more advanced pipeline based on HRNet. Unlike YOLOv11, HRNet estimates dense heatmaps encoding the spatial probability

distribution of each keypoint. This fundamental difference enables more accurate localisation of target features. Moreover, it provides a richer representation of pixel-level prediction uncertainty, which is particularly advantageous under challenging illumination conditions and partial occlusions.

As in the previous pipeline, the final relative pose is recovered through an explicit model-based reconstruction stage, thereby ensuring physical consistency with the target spacecraft's rigid-body model. The adoption of a heatmap-based keypoint estimation strategy, combined with the preservation of high-resolution spatial features, aligns with the best-performing approaches reported in the literature and supports the selection of HRNet as the primary backbone for the proposed vision-based navigation framework.

By explicitly decoupling visual perception from pose reconstruction and learning-based feature extraction within a consistent geometric estimation framework, this work aims to develop a robust and experimentally validated vision-based pose estimation pipeline for spacecraft proximity operations.

Chapter 3

YOLOv11-based Pipeline

This chapter describes the pipeline adopted to train YOLOv11 pose models (*nano* and *large* variants) on the SPEED+ dataset, from label generation to network training, and outlines the subsequent 6-DOF pose recovery stage based on a PnP solver.

The performance of the proposed pipeline is further assessed using the SPEED+ benchmark metrics presented in Section 2.3.1, allowing a direct comparison with the SOTA methods previously discussed.

3.1 YOLOv11-Pose Framework

The aim of this section is to describe the main architectural principles behind YOLOv11-pose and to clarify how the network is structured to address both object detection and keypoint regression.

3.1.1 `Ultralytics` and the YOLO Environment

`Ultralytics` is an open-source software project dedicated to creating modern computer vision models, especially within the You Only Look Once (YOLO) family. Over time, it has expanded to cover different visual tasks, including object detection, instance segmentation, image classification, and pose estimation.

A key strength of the `Ultralytics` environment is its high level of abstraction, as many implementation details are handled internally while still giving users sufficient control over training, inference, and evaluation. Additionally, dataset management, data augmentations, loss functions, and evaluation metrics are consistently implemented across different model variants, making it easier to compare performance between various architectures.

At the time of this work, YOLOv11 represents one of the most recent evolutions of the YOLO architecture, preceded only by YOLO26. All experiments presented in

this thesis are based on the official `Ultralytics` implementation of YOLOv11 [48], which is therefore adopted as the reference framework.

3.1.2 YOLO Architecture Overview

The YOLO family follows a single-stage detection model, where object localisation and classification are performed simultaneously by the network, whereas in two-stage approaches they are treated as separate steps.

From an architectural point of view, YOLO networks are typically organised into three main components:

- the **backbone**, which extracts hierarchical feature representations from the input image;
- the **neck**, which combines features at different spatial scales to capture both semantic and geometric information;
- the **head**, which produces the final predictions, such as bounding boxes, class scores, and keypoints.

YOLOv11 follows this general structure and adopts a fully convolutional design, as illustrated in Figure 3.1 [49].

3.1.3 YOLOv11-Pose Architecture

YOLOv11-pose extends the standard YOLO detection architecture by adding an additional branch for keypoint regression. As a result, the network is trained to simultaneously predict bounding boxes, class labels, and a predefined set of semantic keypoints for each detected instance.

Starting from the input image, the backbone extracts visual features at different spatial resolutions. These features are then combined by the neck, which merges local spatial details and more global information. The resulting feature maps are used by the detection head to predict bounding boxes, object confidence scores, and class labels.

In parallel, YOLOv11-pose includes a dedicated head that estimates the two-dimensional image coordinates of the predefined keypoints. Each keypoint prediction is also associated with a confidence or visibility score, which provides an indication of prediction reliability and helps the model handle partially occluded or visually ambiguous landmarks.

In the framework adopted in this work, object detection and keypoint localisation are performed jointly within the same neural network. However, the estimation of

the full 6-DOF pose is not carried out directly by the network. Instead, the predicted 2D keypoints are used as input to a subsequent geometric pose estimation stage, where the object pose is recovered using a PnP solver.

Overall, the proposed pipeline follows a two-stage approach: a learning-based stage for robust keypoint detection, followed by a model-based stage for geometric pose estimation.

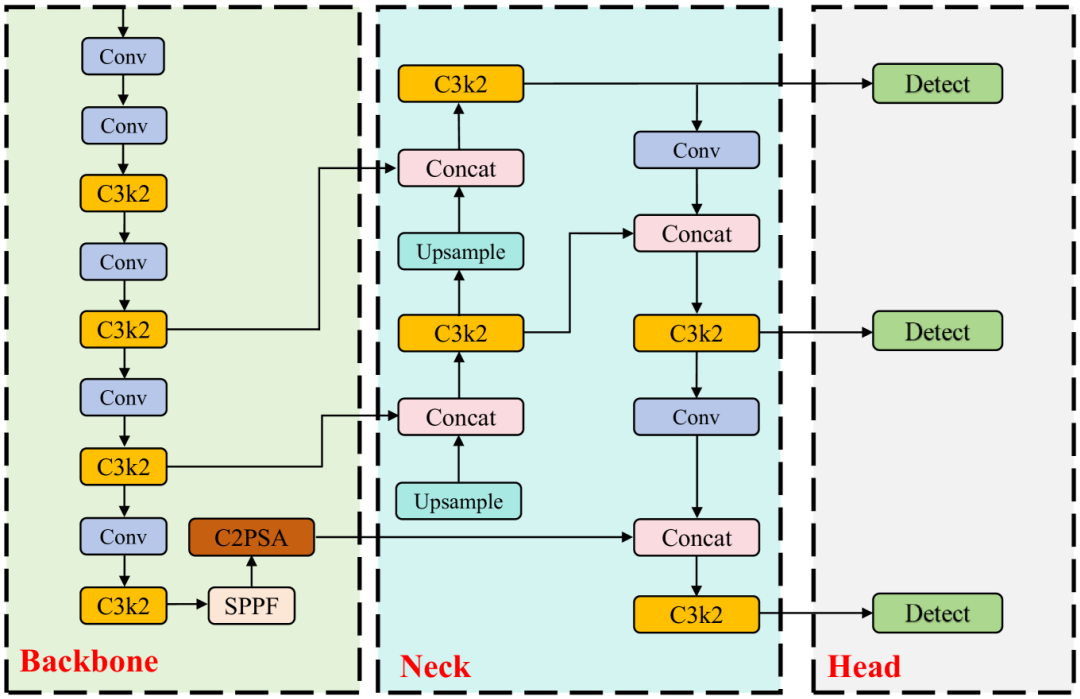


Figure 3.1: Visual representation of the YOLOv11 architecture layout.

3.1.4 Model Variants and Scaling Strategy

YOLOv11 features a scalable architecture and comes in multiple variants with varying capacities. Although all models share the same network structure, they differ in depth, width, and parameter count. This design allows users to balance computational load and accuracy, making the framework adaptable to diverse application needs.

In this work, two representative model scales are considered:

- **YOLOv11-nano (YOLOv11n-pose):** a lightweight configuration optimised for reduced parameter count and fast inference, comprising approximately 2.76 million trainable parameters.
- **YOLOv11-large (YOLOv11l-pose):** a higher-capacity model designed to improve accuracy, with approximately 26.17 million trainable parameters.

The substantial difference in model capacity between these two variants directly affects several aspects, including inference latency, memory footprint, and overall computational load. These considerations are particularly relevant in the context of on-board deployment for space applications, where processing resources, power availability, and hardware constraints are inherently limited.

To ensure a fair and meaningful comparison, both models are trained using the same dataset splits, training pipeline, and hyperparameters configuration. In addition, both networks are initialised from pre-trained pose checkpoints and subsequently fine-tuned on identical SPEED+ data splits. This controlled setup allows differences in convergence behaviour, loss dynamics, and pose estimation accuracy to be attributed primarily to model capacity rather than to variations in data or training strategy. A quantitative comparison between the two variants is presented in Section 3.6.

3.2 Label Generation from SPEED+ Annotations

3.2.1 SPEED+ annotation format and keypoint geometry

SPEED+ dataset provides ground-truth relative pose annotations between the camera and the target spacecraft, including a unit quaternion and a translation vector, together with a predefined set of 3D keypoints expressed in the target body frame [50].

$$\begin{cases} \mathbf{q} = \{q_w, q_x, q_y, q_z\}, & \text{quaternion} \\ \mathbf{t} = \{x, y, z\}, & \text{translation vector} \end{cases}$$

In addition, a fixed set of 3D keypoints $\{\mathbf{P}_i\}_{i=1}^{N}$ is defined in the target body frame (Table 3.1). In this work $N = 11$ keypoints are used, consistent with the Tango geometry adopted in SPEED+.

3.2.2 Pinhole projection model

Given the image size (W, H) and the camera intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

where $\{f_x, f_y\}$ are the focal lengths expressed in pixels and $\{c_x, c_y\}$ are the image optical centre coordinates, each 3D keypoint $\mathbf{P}_i$ expressed in the target body frame

Keypoint ID	**X** [m]	**Y** [m]	**Z** [m]
1	-0.3700	-0.3850	0.3215
2	-0.3700	0.3850	0.3215
3	0.3700	0.3850	0.3215
4	0.3700	-0.3850	0.3215
5	-0.3700	-0.2640	0.0000
6	-0.3700	0.3040	0.0000
7	0.3700	0.3040	0.0000
8	0.3700	-0.2640	0.0000
9	-0.5427	0.4877	0.2535
10	0.5427	0.4877	0.2591
11	0.3050	-0.5790	0.2515

Table 3.1: Fixed 3D coordinates of the 11 spacecraft keypoints expressed in the target body frame.

is transformed into the camera reference frame according to

$$\mathbf{P}_i^c = \mathbf{R}_{\text{cam}\leftarrow\text{obj}}\,\mathbf{P}_i + \mathbf{t}, \tag{3.2}$$

where $\mathbf{t} \in \mathbb{R}^3$ denotes the translation vector of the target with respect to the camera, and $\mathbf{R}_{\text{cam}\leftarrow\text{obj}} \in SO(3)$ [1] is the rotation matrix mapping vectors from the object frame to the camera frame. The rotation matrix $\mathbf{R}_{\text{cam}\leftarrow\text{obj}}$ (previously defined 1.3), is obtained from the ground-truth unit quaternion and is explicitly given by

$$\mathbf{R}_{\text{cam}\leftarrow\text{obj}} = \begin{bmatrix} 1-2(q_y^2+q_z^2) & 2(q_xq_y - q_zq_w) & 2(q_xq_z + q_yq_w) \\ 2(q_xq_y + q_zq_w) & 1-2(q_x^2+q_z^2) & 2(q_yq_z - q_xq_w) \\ 2(q_xq_z - q_yq_w) & 2(q_yq_w + q_xq_w) & 1-2(q_x^2+q_y^2) \end{bmatrix}. \tag{3.3}$$

Let $\mathbf{P}_i^c = [X_i\ Y_i\ Z_i]^\top$ denote the coordinates of the i-th keypoint in the camera frame. Under the pinhole camera model, normalised coordinates are

$$x_i = \frac{X_i}{Z_i}, \qquad y_i = \frac{Y_i}{Z_i}, \tag{3.4}$$

and pixel coordinates follow as

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \tag{3.5}$$

To avoid numerical issues for very small depths, a small ϵ is used when $|Z_i|$ approaches zero in the implementation.

[1] Special Orthogonal Group

3.2.3 Bounding box generation from projected keypoints

`Ultralytics` pose labels follow a YOLO-like format, consisting of specifics related to the bounding box and to keypoints 2D coordinates (normalised to $[0, 1]$). After projection, only keypoints satisfying basic validity checks are retained:

- in front of the camera: $Z_i > 0$,
- inside the image domain: $0 \leq u_i < W$ and $0 \leq v_i < H$.

Keypoints failing these checks are assigned visibility flag $v_i = 0$ (see Sec. 3.2.4).

The ground-truth bounding box is computed as the tight axis-aligned rectangle containing all keypoints with $v_i > 0$. Denoting the set of valid projected keypoints as $\mathcal{U} = \{(u_i, v_i) \,|\, v_i > 0\}$, the bounding box corners are

$$\begin{cases} x_1 = \min_{(u,v)\in\mathcal{U}} u \\ y_1 = \min_{(u,v)\in\mathcal{U}} v \\ x_2 = \max_{(u,v)\in\mathcal{U}} u \\ y_2 = \max_{(u,v)\in\mathcal{U}} v \end{cases}$$

with width, height and centre's 2D coordinates

$$\begin{cases} w_b = x_2 - x_1 \\ h_b = y_2 - y_1 \\ c_x = x_1 + \frac{w_b}{2} \\ c_y = y_1 + \frac{h_b}{2} \end{cases}$$

The YOLO normalised representation is then

$$\hat{c}_x = \frac{c_x}{W},\ \hat{c}_y = \frac{c_y}{H},\ \hat{w}_b = \frac{w_b}{W},\ \hat{h}_b = \frac{h_b}{H} \tag{3.6}$$

Similarly, keypoint coordinates are normalized as $\hat{u}_i = u_i/W$ and $\hat{v}_i = v_i/H$ and clipped to $[0, 1]$ in the implementation .

3.2.4 Ray-casting visibility model for the keypoint visibility flag computation

For pose datasets with strong self-occlusions (i.e., situations in which portions of the spacecraft obscure other parts of the same structure due to complex geometry and viewing angle, such as solar panels or antennas blocking keypoints), providing

a visibility indicator per keypoint is essential to distinguish (i) points outside the image or behind the camera from (ii) points projected inside the image but occluded by the spacecraft itself.

A ray-casting model was adopted to assign a visibility flag $v_i \in \{0, 1, 2\}$ for each keypoint :

- $v_i = 0$: keypoint not valid for supervision (behind camera or outside image);
- $v_i = 1$: keypoint projects inside the image but is *self-occluded*;
- $v_i = 2$: keypoint projects inside the image and is *visible*.

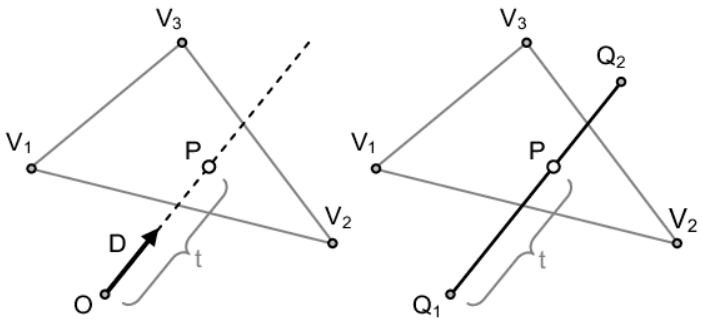


Figure 3.2: Conceptual illustration of the ray–triangle intersection test used for keypoint visibility evaluation[51]. A ray is cast from the camera centre toward the 3D keypoint. If an intersection with the object surface occurs at a distance smaller than the keypoint depth, the keypoint is classified as occluded.

Let the camera centre be $\mathbf{o} = [0\ 0\ 0]^\top$ and the 3D keypoint in camera coordinates be $\mathbf{P}_i^c$. As illustrated in Figure 3.2, a ray is cast from the camera centre toward the keypoint location. The ray direction is defined as

$$\mathbf{d}_i = \frac{\mathbf{P}_i^c - \mathbf{o}}{\|\mathbf{P}_i^c - \mathbf{o}\|}. \tag{3.7}$$

where $D_i = \|\mathbf{P}_i^c - \mathbf{o}\|$ is the Euclidean distance to the keypoint.

The spacecraft geometry is represented by a triangular mesh; in this work, a coarse mesh consistent with the main Tango body faces is adopted for visibility evaluation. For each triangle $\triangle(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, the algorithm tests whether the ray $(\mathbf{o}, \mathbf{d}_i)$ intersects the triangle at a distance t satisfying

$$0 < t < D_i - \delta, \tag{3.8}$$

with a small margin δ to avoid classifying the keypoint's own surface as an occluder. If such an intersection exists, the keypoint is marked as occluded ($v_i = 1$), otherwise it is marked as visible ($v_i = 2$) (see Figure 3.3).

The ray–triangle intersection problem, as suggested in [40], is solved using the *Möller–Trumbore algorithm* [51], a widely adopted method in computer graphics and geometric visibility analysis due to its computational efficiency and minimal storage requirements. The approach described is shown schematically in Figure 3.4, which illustrates the geometric setup behind the keypoint visibility test.

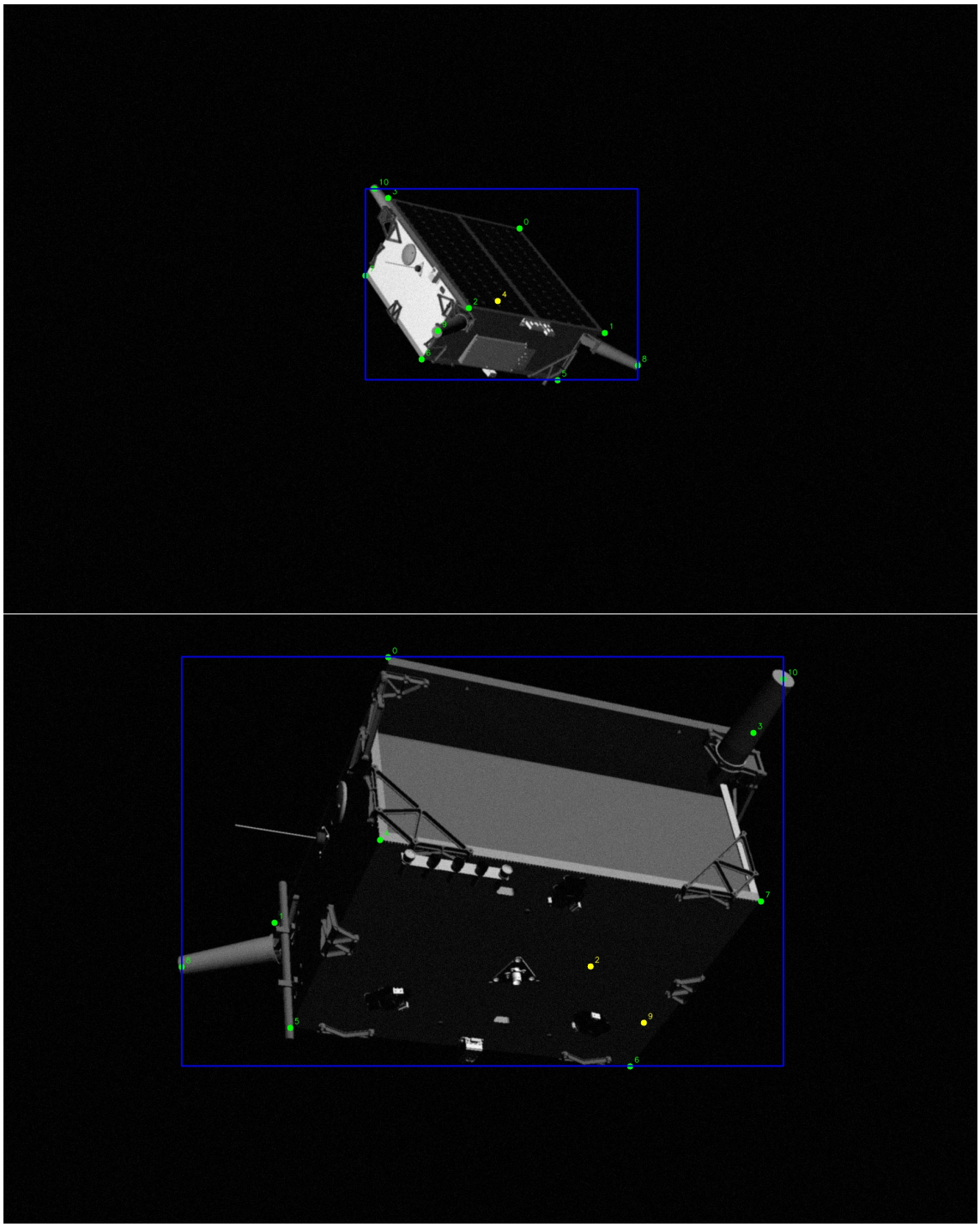


Figure 3.3: Examples of ground-truth keypoint projections on SPEED+ images. Visible keypoints are shown in green, while self-occluded keypoints are highlighted in yellow.

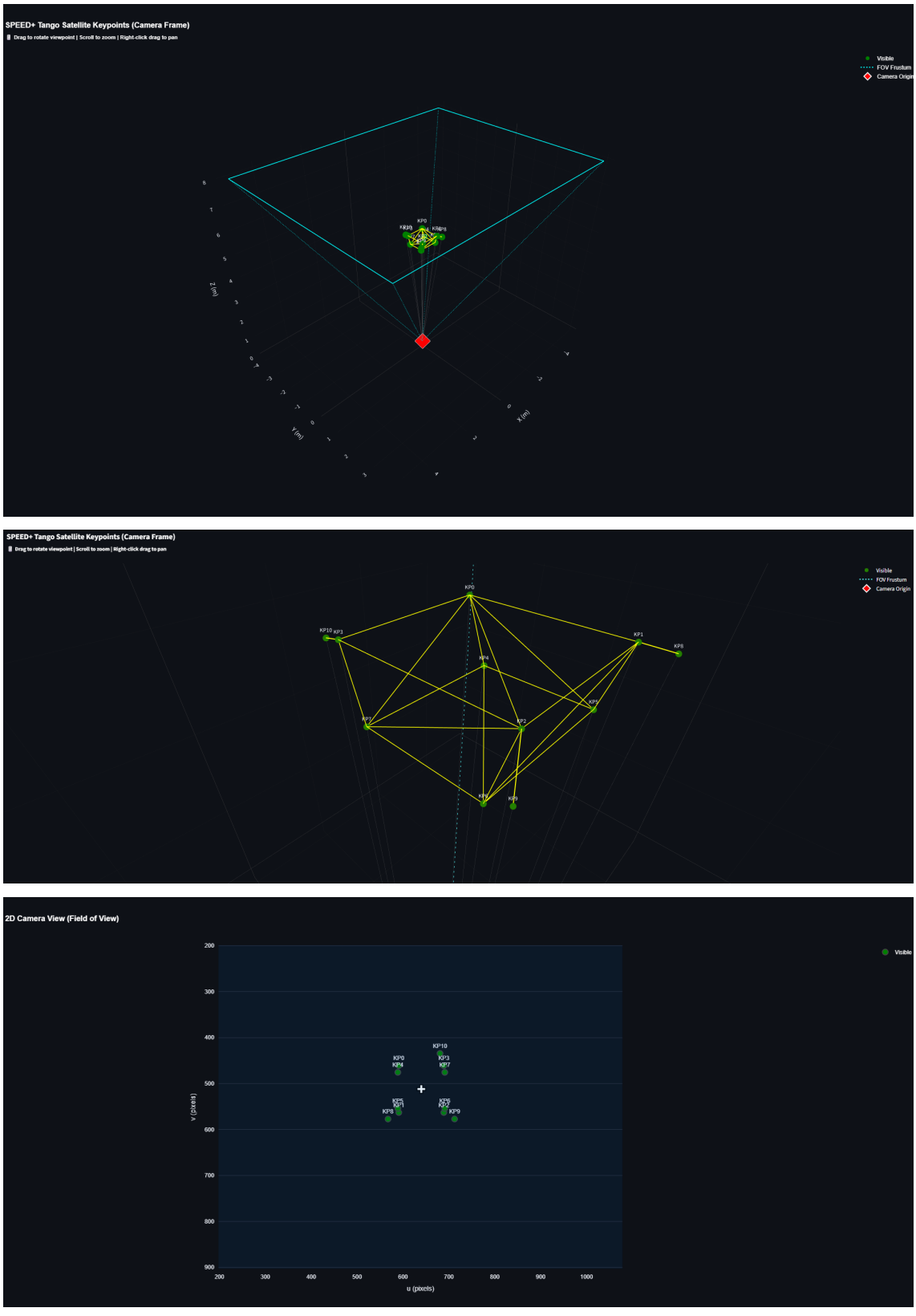


Figure 3.4: Interactive 3D visualisation of the Tango spacecraft in the camera reference frame. The spacecraft geometry, characteristic keypoints, camera origin, and field of view are shown. The scene illustrates the ray casting and triangle–intersection model used to determine keypoint visibility by evaluating line-of-sight occlusions between the camera and the spacecraft surface. The bottom picture shows the camera's field of view.

3.2.5 Label format

Each image produces one text file containing:

$$\underbrace{c}_{\text{class}} \; \underbrace{\hat{c}_x \; \hat{c}_y \; \hat{w}_b \; \hat{h}_b}_{\text{bbox}} \; \underbrace{\hat{u}_1 \; \hat{v}_1 \; v_1 \; \cdots \; \hat{u}_N \; \hat{v}_N \; v_N}_{\text{keypoints}}, \tag{3.9}$$

where $c = 0$ for the single class "Tango" and $N = 11$ in this work.

3.3 YOLOv11 Training Setup

3.3.1 Training script and hyperparameters

The training procedure adopted in this work is based on the exploitation of the YOLOv11 pose model. This model, pre-trained on large-scale image datasets, is enhanced on the SPEED+ dataset in order to adapt the network to a specific visual characteristic of the TANGO spacecraft. The training setup comprises a set of hyperparameters, including input resolution, batch size, and number of epochs, aiming at guaranteeing a compromise between computational cost and training effectiveness.

In this work the following hyperparameters were chosen:

- image size: 640×640
- batch size: 16
- epochs: 100

3.3.2 Augmentation techniques

In order to improve robustness to illumination variations, scale effects, and domain shifts, the default `Ultralytics` augmentation pipeline was enabled during training. Representative examples of augmented training batches are shown in Fig. 3.5, while the complete list of augmentation techniques adopted during training are summarised in Table 3.2.

YOLO adopts a mixed augmentation strategy combining both *photometric* and *geometric* transformations. Photometric augmentations are applied directly to the image features. In particular, they include random perturbations in Hue Saturation Value (HSV) colour space: hue, saturation and brightness are randomly modified through the parameters `hsv_h`, `hsv_s`, and `hsv_v`, improving robustness to illumination changes, reflections, and exposure variability.

In addition, YOLO employs compositional augmentations. One example is the *Mosaic* augmentation, which combines four images into a single training sample, increasing contextual diversity and exposing the model to a various object scales and partial views during each iteration. Mosaic is enabled by default and is gradually disabled towards the end of training through the `close_mosaic` mechanism. This allows the network to improve localisation accuracy during the final optimisation stage.

Finally, a *Random Erasing* strategy is applied with a fixed probability. This augmentation randomly removes rectangular areas from the input image, simulating partial occlusions and missing visual information, thereby improving robustness to these kind of challenges.

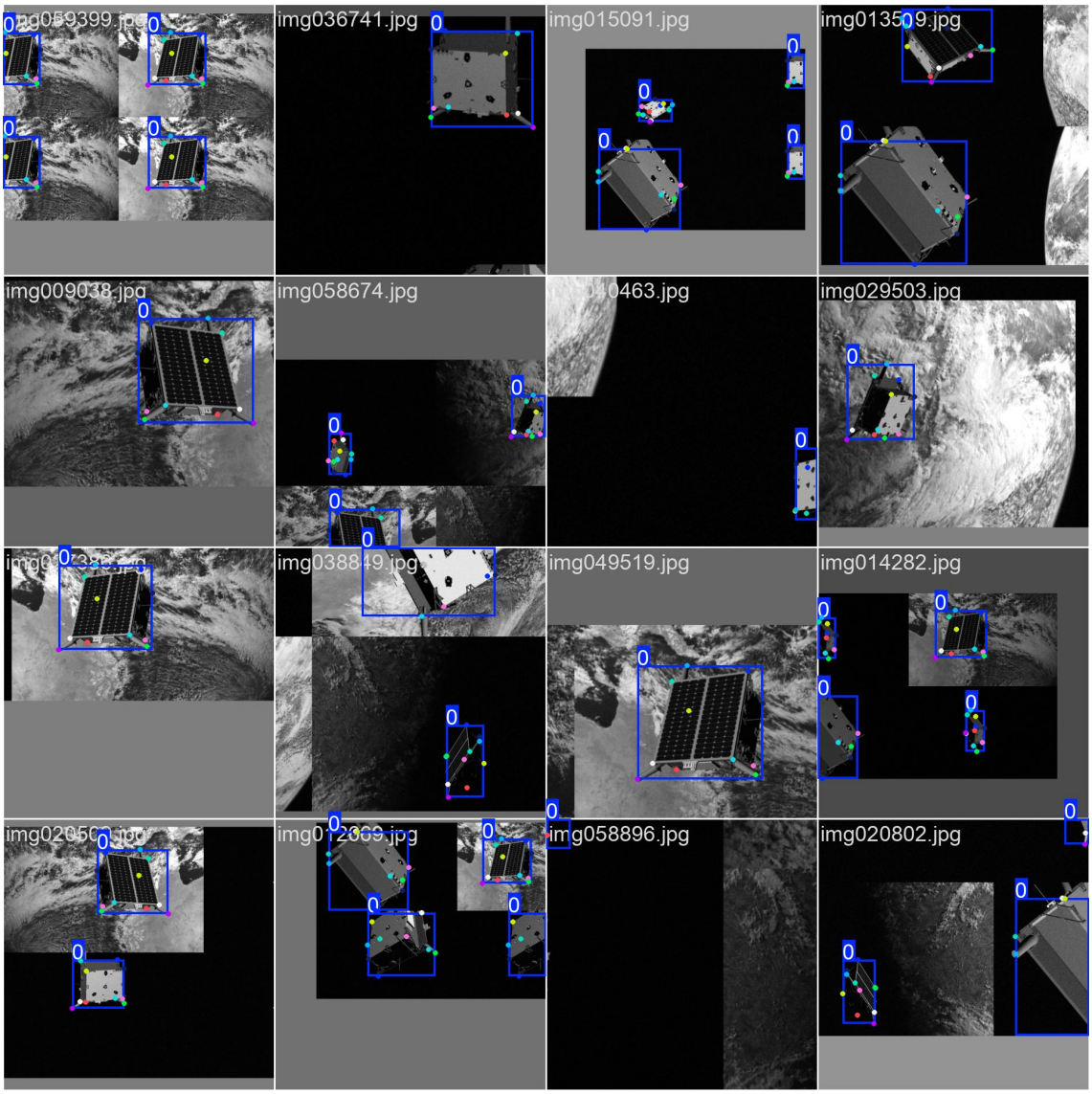


Figure 3.5: Examples of training batches generated during YOLOv11 training on the SPEED+ dataset. Images illustrate bounding boxes, keypoints, and the effect of data augmentation techniques applied during training.

Augmentation	Default value	Description
HSV hue (`hsv_h`)	0.015	Random hue shift in HSV space to improve robustness to color variations.
HSV saturation (`hsv_s`)	0.70	Random saturation change to account for sensor and lighting variability.
HSV brightness (`hsv_v`)	0.40	Random brightness perturbation to handle exposure changes.
Translation (`translate`)	0.10	Random translation as a fraction of image size.
Scale (`scale`)	0.50	Random isotropic scaling of the image and annotations.
Horizontal flip (`fliplr`)	0.50	Horizontal mirroring with fixed probability.
Mosaic (`mosaic`)	1.0	Four-image mosaic composition to increase scale and context diversity.
Random erasing (`erasing`)	0.40	Random removal of rectangular image regions to simulate occlusions.

Table 3.2: Summary of the `Ultralytics` YOLO augmentation parameters used during training. Identical settings were adopted both for YOLOv11n-pose and YOLOv11l-pose.

3.3.3 Loss Functions

In this section a comprehensive analyses of the training behaviour of both YOLOv11-n and YOLOv11-l models is presented, examining the evolution of the loss functions during the training process. Since both models are trained on the same dataset and training setup, the comparison highlights the effect of model capacity on convergence dynamics and pose estimation accuracy.

YOLOv11 employs multiple loss components simultaneously, each associated with a specific prediction task. Following the `Ultralytics` YOLO training framework [52], [53], the total loss can be expressed as a weighted sum of different loss terms:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{dfl}}\mathcal{L}_{\text{dfl}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{kobj}}\mathcal{L}_{\text{kobj}} \tag{3.10}$$

where the weighting coefficients λ_i are fixed during training and adapted to balance the contribution of each component.

The `bounding box regression loss` $\mathcal{L}_{\text{box}}$ supervises object localisation and is based on a metric related to the Intersection-over-Union (IoU), which corresponds to

$$\text{IoU} = \frac{\text{area of overlap}}{\text{area of union}} \tag{3.11}$$

This loss favours maximal overlap between predicted and ground truth bounding

boxes. Denoting by $\mathbf{b}$ and $\mathbf{b}^*$ the predicted and ground-truth bounding boxes, respectively, the `bounding box regression loss` is defined as

$$\mathcal{L}_{\text{box}} = 1 - \text{CIoU}(\mathbf{b}, \mathbf{b}^*) \tag{3.12}$$

where CIoU (Complete IoU) extends the standard IoU by including penalties on the distance between box centres and on the aspect ratio consistency. This formulation encourages not only maximal overlap but also geometric coherence between predicted and target boxes.

The `classification loss` $\mathcal{L}_{\text{cls}}$ measures the difference between predicted class probabilities and ground truth labels, typically using a binary cross-entropy formulation. Denoting with p_c the predicted probability for class c and with $y_c \in \{0, 1\}$, the corresponding ground-truth label, the `classification loss` is defined as

$$\mathcal{L}_{\text{cls}} = \sum_c [y_c \log p_c + (1 - y_c) \log(1 - p_c)] \tag{3.13}$$

To further refine bounding box localisation, YOLOv11 employs the `Distribution Focal Loss` (DFL). Instead of predicting each box coordinate as a single scalar value, DFL models it as a discrete probability distribution and computes the final coordinate as its expected value. This formulation improves localisation precision and stabilises training, especially during the early optimisation stages.

For each box side $d \in \{l, t, r, b\}$, the network predicts a discrete probability distribution $p_d(k)$ over K bins. Given the ground-truth value d^*, the `Distribution Focal Loss` is defined as a weighted cross-entropy between the predicted distribution and the two neighbouring bins surrounding d^*:

$$\mathcal{L}_{\text{DFL}}(d) = -(1 - \delta) \log p_d(k) - \delta \log p_d(k + 1) \tag{3.14}$$

where $k = \lfloor d^* \rfloor$ and $\delta = d^* - k$. The total DFL is obtained by summing over the four box sides.

In addition to terms related to detection, YOLOv11-pose introduces a dedicated `pose loss` $\mathcal{L}_{\text{pose}}$, which directly supervises keypoint regression. Let $\hat{k}_i$ and $\hat{k}_i^*$ denote the predicted and ground-truth image coordinates of the i-th keypoint. The `pose loss` penalises the error between predicted and ground-truth keypoint coordinates, typically using an L2-based formulation, and it is modulated by a visibility mask m_i:

$$\mathcal{L}\text{pose} = \sum i = 1^{N_k} m_i \left| \hat{\mathbf{k}}_i - \mathbf{k}_i^* \right|_2^2 \tag{3.15}$$

This formulation ensures that only visible keypoints contribute to the optimisa-

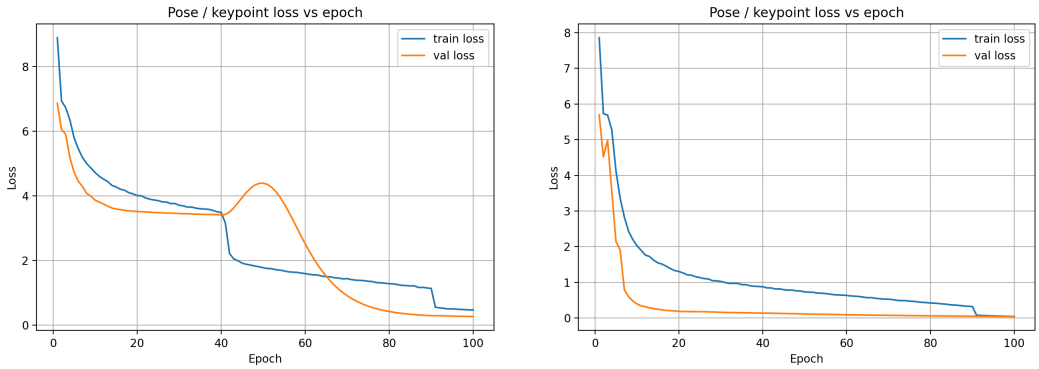
tion, while occluded or undefined landmarks are ignored. Finally, the `keypoint objectness loss` $\mathcal{L}_{\text{kobj}}$ modulates the contribution of individual keypoints based on their visibility. Let $\hat{v}_i$ be the predicted visibility logit for the i-th keypoint. The corresponding loss term is defined as a binary cross-entropy:

$$\mathcal{L}_{\text{kobj}} = \sum_{i=1}^{N_k} [m_i \log \sigma(\hat{v}_i) + (1 - m_i) \log(1 - \sigma(\hat{v}_i))] \tag{3.16}$$

where $\sigma(\cdot)$ denotes the sigmoid function. This term reduces the influence of occluded or uncertain landmarks, improving robustness in difficult scenarios.

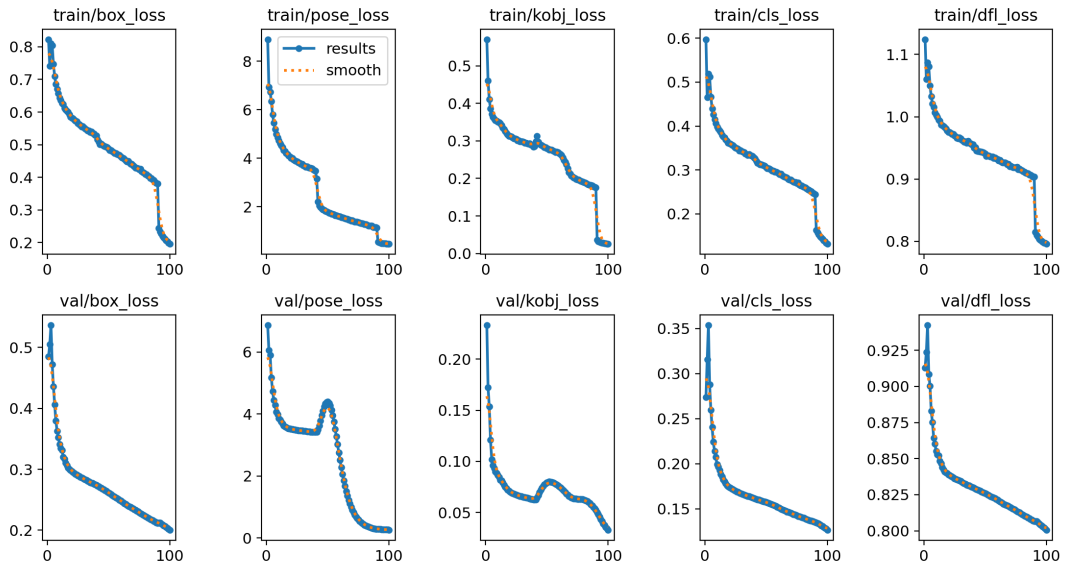
Figure 3.7 reports the evolution of all training and validation loss components for the nano and large models, respectively. In both cases, all loss terms exhibit a consistent decreasing trend, indicating stable optimisation. Compared to the nano configuration, the large model generally shows smoother loss trends and lower final loss values, reflecting its increased representational capacity.

Particular attention is devoted to the `pose loss` per keypoint, as it directly reflects the network's ability to regress accurate keypoint positions. Figure 3.6 illustrates the evolution of the `pose loss` for both training and validation sets. Both models show rapid initial convergence followed by a gradual refinement phase. The validation loss remains close to the training loss throughout optimisation, suggesting good generalisation and absence of significant overfitting. As expected, the large model achieves lower pose loss values and faster stabilisation compared to the nano variant.

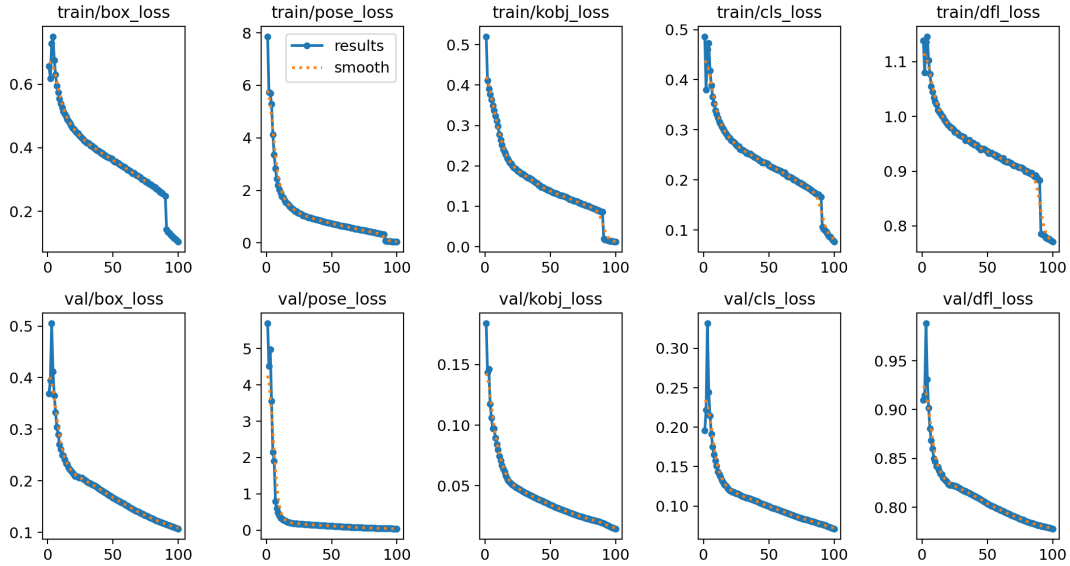


(a) `YOLOv11n-pose` (b) `YOLOv11l-pose`

Figure 3.6: Pose loss per keypoint evolution: comparison between YOLOv11-n and YOLOv11-l models on training and validation sets.



(a) `YOLOv11n-pose`



(b) `YOLOv11l-pose`

Figure 3.7: Comparison of training and validation loss components for YOLOv11-n and YOLOv11-l during optimization.

3.3.4 Mean Average Precision

To further assess the effectiveness and stability of the training process, the evolution of the *mean Average Precision* computed over multiple IoU thresholds, denoted as $\text{mAP}_{50\text{–}95}$, is reported in Figure 3.8 . This metric represents the primary performance indicator adopted in multiple pose estimation frameworks, including the YOLO family, and provides a comprehensive measure of localisation accuracy across different levels of tolerance.

In the YOLO evaluation pipeline, predictions are first ranked according to their confidence score and then classified as *true positives* (TP), *false positives* (FP), *true negatives* (TN) and *false negatives* (FN) by comparing the predicted bounding boxes with the ground truth using the IoU criterion. For a given IoU threshold τ, *precision* and *recall* are defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \ , \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.17}$$

The *Average Precision* at a given IoU threshold τ is defined as the area under the corresponding precision-recall curve:

$$AP_r = \int_0^1 p_\tau(r) \ dr \tag{3.18}$$

where $p_\tau(r)$ denotes precision as a function of recall. A detection is considered correct if the predicted bounding box (and the associated keypoints) achieves an IoU grater than τ with respect to the ground truth annotation. As discussed in [54], this formulation allows AP to jointly account for both localisation accuracy and confidence calibration.

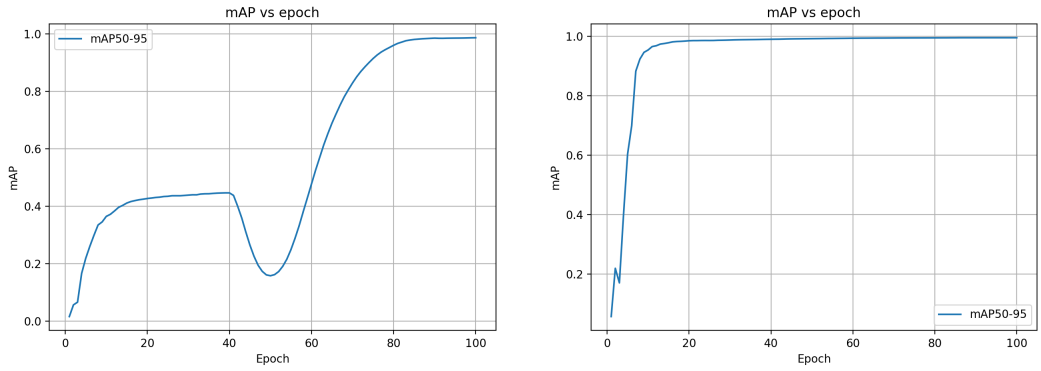
YOLO defines the $\text{mAP}_{50\text{–}95}$ averaged over a set of IoU thresholds, ranging from 0.50 to 0.95 with a step of 0.05 (for instance, $\tau \in \{0.5, 0.55, ..., 0.95\}$):

$$mAP_{50\text{–}95} = \frac{1}{10} \sum_{\tau=0.5}^{0.95} AP_\tau \tag{3.19}$$

This metric provides a stricter and more informative evaluation than mAP_{50} alone, as higher IoU thresholds increasingly penalise small localisation errors. In the context of keypoint-based pose estimation, this implicitly reflects the quality of the predicted keypoint configuration, since inaccurate keypoint localisation typically leads to reduced bounding box overlap and lower detection confidence.

As shown in Figure 3.8, both the nano and large YOLOv11 models show a monotonous increase in $\text{mAP}_{50\text{–}95}$ as training progresses, confirming a progressive refinement of the learnt feature representations and a continuous improvement in keypoint localisation accuracy. A temporary degradation in performance is observed during intermediate epochs for the nano model.

Following this phase, the nano model performance steadily recovers and converges to a stable plateau, indicating successful adaptation to the augmented training. In contrast, the large model reaches higher mAP values at earlier epochs and maintains a more stable performance throughout the entire training. This behaviour reflects its increased capacity and robustness.



(a) `YOLOv11n-pose` (b) `YOLOv11l-pose`

Figure 3.8: Comparison of $mAP_{50–95}$ evolution during training for YOLOv11-n and YOLOv11-l models.

3.4 6-DOF Pose Recovery via PnP solver

This section describes the procedure adopted to recover the full 6-DOF relative pose of the target spacecraft, namely the translation vector and the quaternion. To solve the pose estimation problem, the `solvePnP` function is designed to minimise the reprojection error between the projected 3D points and the 2D keypoint predictions generated by the YOLOv11 network.

A trade-off analysis was performed to study the influence of the confidence threshold on the overall robustness and accuracy of the proposed method, particularly in the HIL domain evaluation process.

Furthermore, a comprehensive study on various PnP algorithms is also provided to explain the decision process that led to the selection of the `SQPnP` solver for the pose estimation problem [55].

3.4.1 Pose estimation pipeline overview

The complete pose recovery pipeline comprises four main stages.

First, keypoint inference is performed using a YOLOv11 pose estimation model. For each input image, the network predicts one or more spacecraft instances, from which a single detection is selected based on a confidence-driven criterion. The corresponding predictions are then stored in a label format aligned with YOLO, allowing a clear separation between inference and evaluation.

Second, a filtering step is applied to the predicted labels based on keypoint confidence and visibility. Only keypoints classified as visible are considered valid for pose estimation, improving robustness against occlusions and inaccurate localisations.

Third, the filtered 2D keypoints are used to estimate the relative pose through a PnP solver, leveraging the known 3D coordinates of the spacecraft keypoints and the camera intrinsic parameters. An initial pose estimate is obtained using an EPnP-based RANSAC procedure, which identifies a consistent set of inlier correspondences. The pose is then refined using an `SQPnP` solver, ensuring a more accurate and deterministic pose estimate.

Finally, the estimated pose is quantitatively evaluated using the SPEED+ benchmark metrics.

This pipeline enables a direct analysis of how 2D keypoint localisation accuracy impacts the final 6-DOF pose estimation performance.

3.4.2 Evaluation metrics

All three dataset splits (e.g, `synthetic`, `lightbox` and `sunlamp`) are evaluated using the metrics outlined in Section 2.3.1, enabling a fair comparison with the other SOTA works.

In addition to those metrics, the geometric consistency of the estimated pose is evaluated through the Reprojection Mean Squared Error (RMSE) in the image plane. Given a set of N known 3D keypoints $\{\mathbf{X}_i\}$ on the target spacecraft and their corresponding ground-truth 2D image projections $\{\mathbf{x}_i^{gt}\}$, the RMSE is defined as:

$$\text{RMSE}_{px} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left\|\mathbf{x}_i^{gt} - \pi(\mathbf{X}_i; \mathbf{R}_{est}, \mathbf{t}_{est})\right\|^2}, \tag{3.20}$$

where $\pi(\cdot)$ denotes the camera projection function using the estimated rotation $\mathbf{R}_{est}$ and translation $\mathbf{t}_{est}$. This metric, expressed in pixels, provides insight into the quality of the 2D–3D alignment achieved by the pose estimate.

3.4.3 Inference and Confidence Threshold Selection

Pose recovery experiments are conducted on all three subsets of the SPEED+ dataset. The `synthetic` validation split is used to characterise nominal operating conditions, while the `lightbox` and `sunlamp` test splits introduce progressively more challenging illumination scenarios, enabling a systematic assessment of the robustness of the complete detection and pose estimation pipeline.

Inference Process

During inference, each input image is processed by the YOLOv11 pose model, which predicts one or more candidate bounding boxes together with the corresponding 2D

keypoint locations and associated confidence scores. The detection stage operates as follows:

1. The network produces a set of candidate bounding boxes, each associated with an object confidence score.
2. Detections with confidence lower than a predefined threshold τ_{conf} are discarded.
3. Non-Maximum Suppression (NMS) is applied to remove redundant overlapping detections.
4. If at least one bounding box survives this filtering stage, the corresponding keypoints are extracted and stored as a prediction file.

Therefore, a prediction file is generated for a given image only if at least a 2D-pose is computed.

More formally, let $D = \{d_i\}_{i=1}^{N}$ denote the set of detections predicted by the network for a given image, each associated with a confidence score $s_i \in [0, 1]$. Given a confidence threshold τ_{conf}, the subset of retained detections is defined as

$$D_\tau = \{d_i \in D \mid s_i \geq \tau_{\text{conf}}\}. \tag{3.21}$$

A prediction file is generated if and only if

$$|D_\tau| \geq 1. \tag{3.22}$$

If $|D_\tau| = 0$, no keypoints are exported and the image is excluded from the subsequent PnP stage.

Effect of the Confidence Threshold

The confidence threshold τ_{conf} controls the trade-off between evaluation coverage and prediction reliability.

Let N_{tot} denote the total number of images in a dataset split, and $N_{\text{valid}}(\tau_{\text{conf}})$ the number of images for which at least one detection satisfies the confidence constraint. The evaluation coverage can therefore be defined as

$$C(\tau_{\text{conf}}) = \frac{N_{\text{valid}}(\tau_{\text{conf}})}{N_{\text{tot}}}. \tag{3.23}$$

This quantity directly corresponds to the complement of the discarded sample percentage reported in Figure 3.9.

Lowering τ_{conf} increases the number of images for which a prediction file is produced, as detections with lower scores are admitted. However, such detections are typically characterised by a higher uncertainty in the localisation of keypoints. Since the PnP solver is highly sensitive to 2D keypoint noise, this increased uncertainty propagates into larger translational and rotational errors, degrading the overall pose accuracy (Figure 3.10).

In contrast, increasing τ_{conf} restricts inference to high-confidence detections. Although this improves pose accuracy on the retained subset, it drastically reduces the number of evaluated samples. For instance, setting a very high thresholds such as $\tau_{\text{conf}} = 0.9$, results in evaluating only $\approx 20\%$ of the dataset, which significantly compromises the statistical representativeness and operational robustness of the pipeline.

The `synthetic` split exhibits minimal sensitivity to variations in τ_{conf}. Across the analysed range, nearly all images return at least one detection, resulting in stable coverage and nearly constant pose estimation metrics. This behaviour reflects the controlled illumination and reduced domain variability of the synthetic data.

Conversely, the `lightbox` and `sunlamp` splits display a considerably stronger dependence on the confidence threshold. Under challenging lighting conditions, the distribution of detection confidence scores changes, and a significant fraction of candidate detections fall below moderate thresholds.

Based on the trade-off analysis, a confidence threshold of $\tau_{\text{conf}} = 0.25$ is selected for all subsequent experiments. This value provides a consistent compromise between coverage and pose estimation accuracy, maintaining sufficient statistical representativeness while avoiding excessive degradation of translational and rotational performance under challenging illumination conditions.

Analysis of Coverage Differences Across Dataset Splits

By referring to Figure 3.9, it is possible to understand that with $\tau_{\text{conf}} = 0.25$, the percentage of images for which a prediction file is generated is the following:

- 84.39% for YOLOv11n and 45.93% for YOLOv11l in the `lightbox` split;
- 78.40% for YOLOv11n and 91.22% for YOLOv11l in the `sunlamp` split.

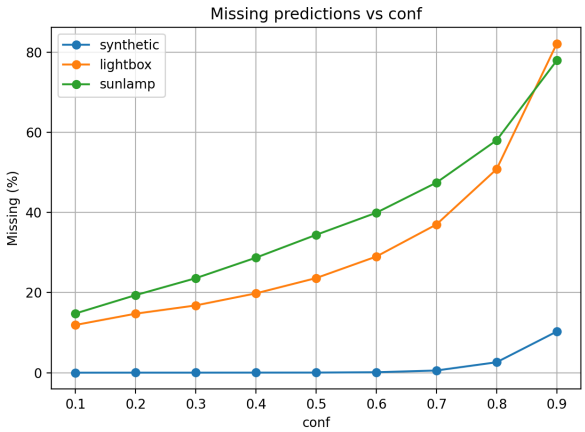
The asymmetric behaviour between the two models across the `lightbox` and `sunlamp` splits can be attributed to differences in confidence score calibration and domain sensitivity.

Moreover, in the `lightbox` split, the large model produces significantly fewer detections above threshold than the nano model. This suggests a more conservative

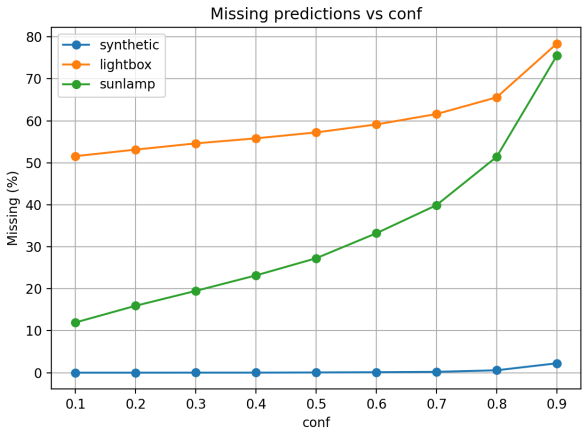
confidence distribution under these illumination conditions. The larger network, characterised by higher representational capacity, may exhibit stronger domain sensitivity, resulting in lower confidence scores. As a consequence, a substantial fraction of detections falls below τ_{conf}, reducing coverage.

In contrast, in the `sunlamp` split, the large model produces the highest coverage (91.22%), indicating that its confidence distribution remains well above the threshold under these specific illumination characteristics. This suggests that the large architecture is capable of extracting more robust features in scenarios where the lighting configuration remains geometrically consistent, even if challenging.

The nano model, due to its reduced capacity, may exhibit a less selective behaviour, producing detections with moderately high confidence even in ambiguous conditions. This results in higher coverage in some cases (e.g., `lightbox`), but may also lead to increased keypoint noise and reduced pose precision.



(a) `YOLOv11-n`



(b) `YOLOv11-l`

Figure 3.9: Percentage of the discarded samples from the inference with varying confidence value. The plots refer to both the YOLOv11n and the YOLOv11l models. Each plot shows a curve for each subset.

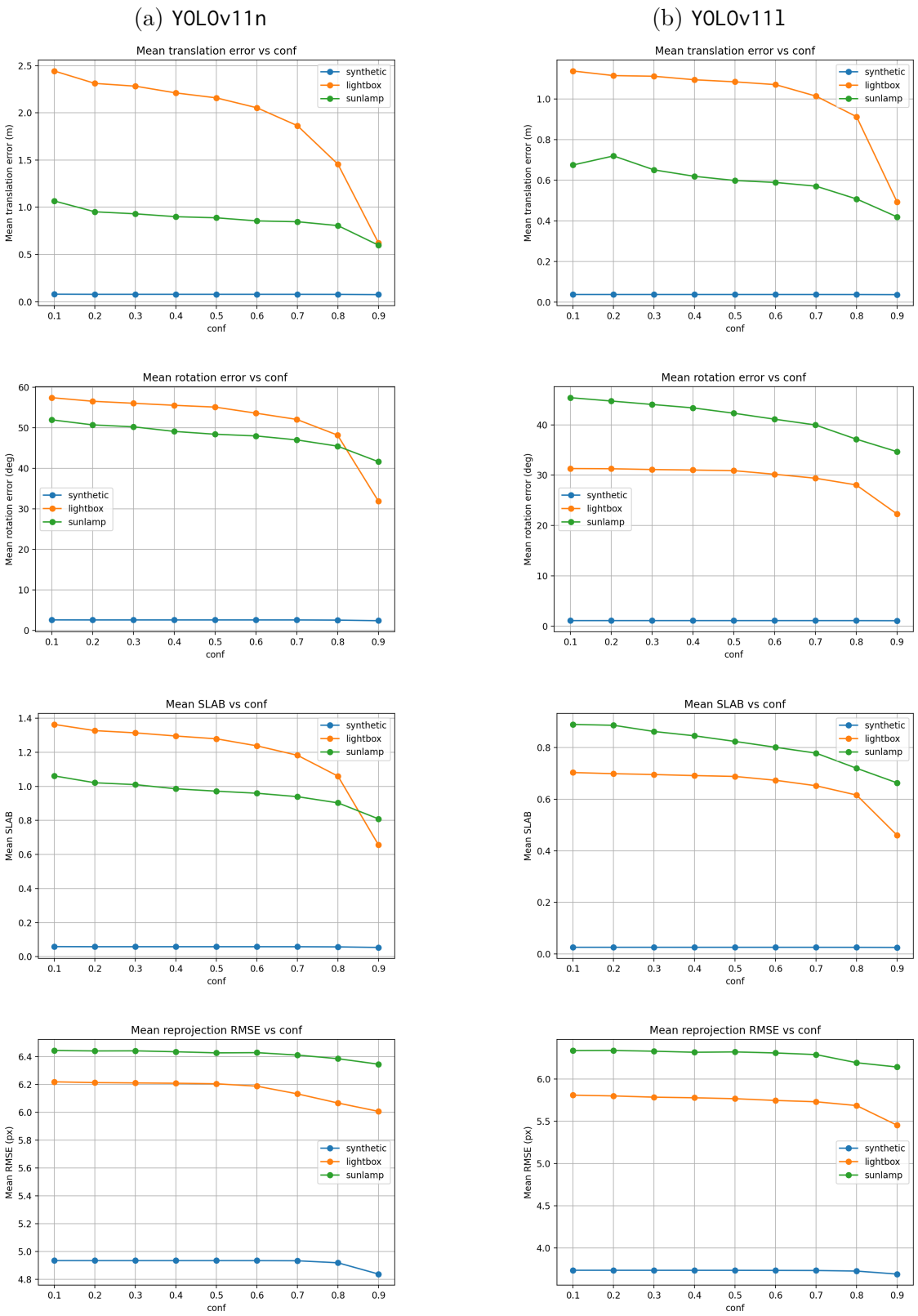


Figure 3.10: The figures illustrate the trade-off between confidence threshold and pose estimation accuracy in terms of translation error E_t, orientation error e_q, SLAB score and RMSE. The plots refer to both the YOLOv11n model (left) and the YOLOv11l model (right). Each plot presents a curve for each subset.

3.4.4 Comparison between different PnP solvers

Given the 2D–3D correspondences and camera intrinsics, the PnP solver determines the quaternion and translation vector that best align known 3D model points with their predicted 2D projections.

A comparative analysis of several PnP solvers was conducted to evaluate their impact on final pose estimation performance for the considered satellite pose estimation problem. The following OpenCV implementations were evaluated: `AP3P`, `DLS`, `EPnP`, `ITERATIVE`, `P3P`, `SQPnP` and `UPnP` [55].

In the `synthetic` domain, all solvers exhibit very similar performance. Under controlled illumination and low keypoint noise, solver-specific differences are almost negligible, as reflected by the extremely low metric values in Figures 3.11 and 3.12.

In contrast, in the `lightbox` and `sunlamp` splits, the dispersion of the error metrics increases for all methods, consistently with the higher uncertainty in the predicted correspondences.

Among the evaluated approaches, `ITERATIVE` shows the largest dispersion in translation error and SLAB and significant failure rate, particularly in the realistic domains. Minimal solvers (`P3P`, `AP3P`) exhibit slightly higher variability compared to the remaining methods, although their central tendency remains comparable. Linear formulations (`EPnP`, `DLS`, `UPnP`) display stable and homogeneous behaviour across the different dataset splits.

Overall, `SQPnP` maintains low median errors together with limited dispersion across all domains, without additional computational cost.

To complement the boxplot analysis, a Pareto study was performed (Figure 3.13), considering accuracy (SLAB), robustness (failure rate) and runtime simultaneously. Although the differences are not significantly pronounced, the Pareto representation confirms that `SQPnP` provides the most balanced compromise between accuracy, robustness and computational cost.

Based on these observations, `SQPnP` is selected as the reference solver for the subsequent experiments.

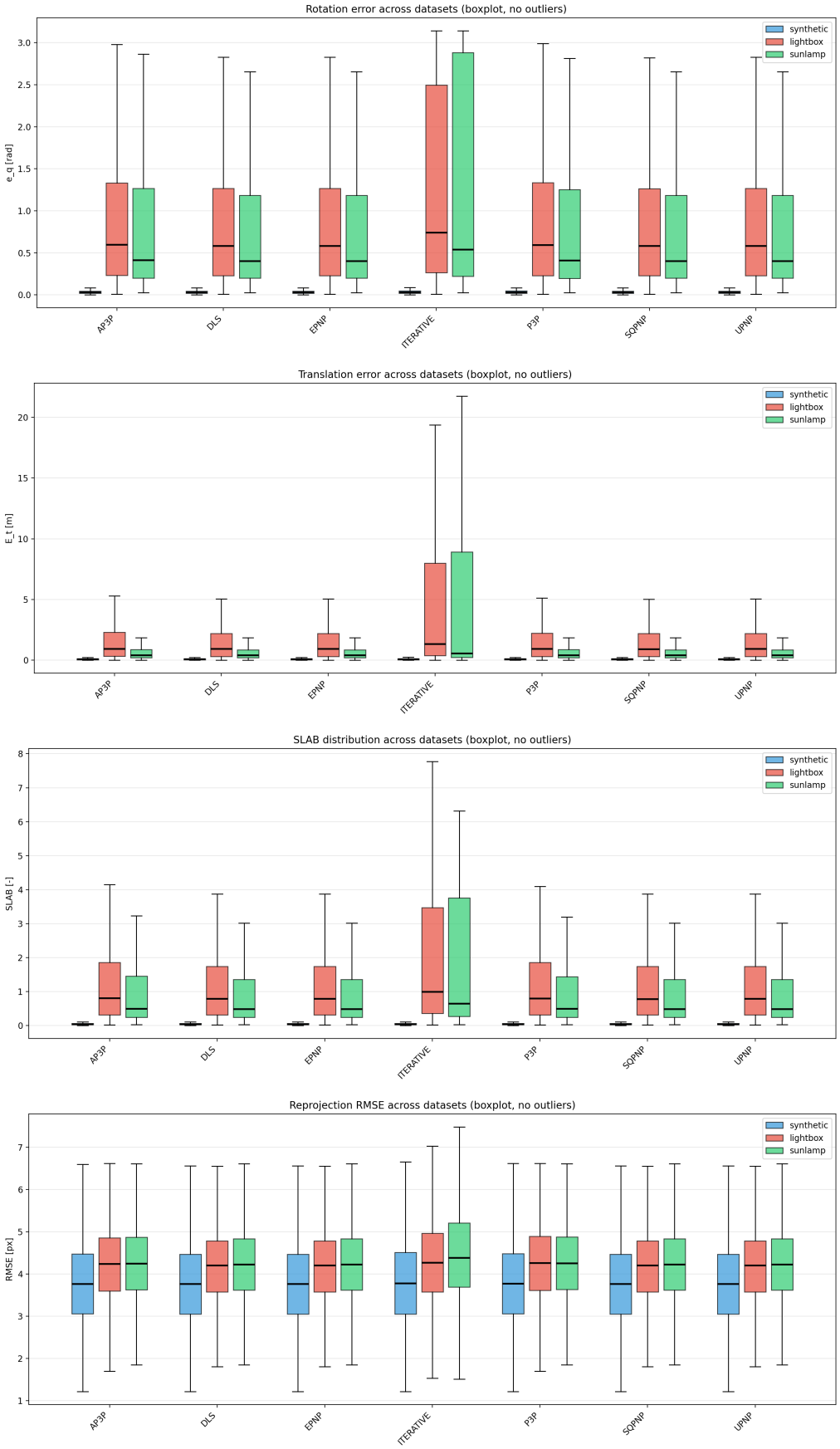


Figure 3.11: Boxplots of rotation error e_q, translation error E_t and SLAB score for each dataset (`synthetic`, `lightbox` and `sunlamp`) evaluated with different PnP using the YOLOv11-nano model predictions.

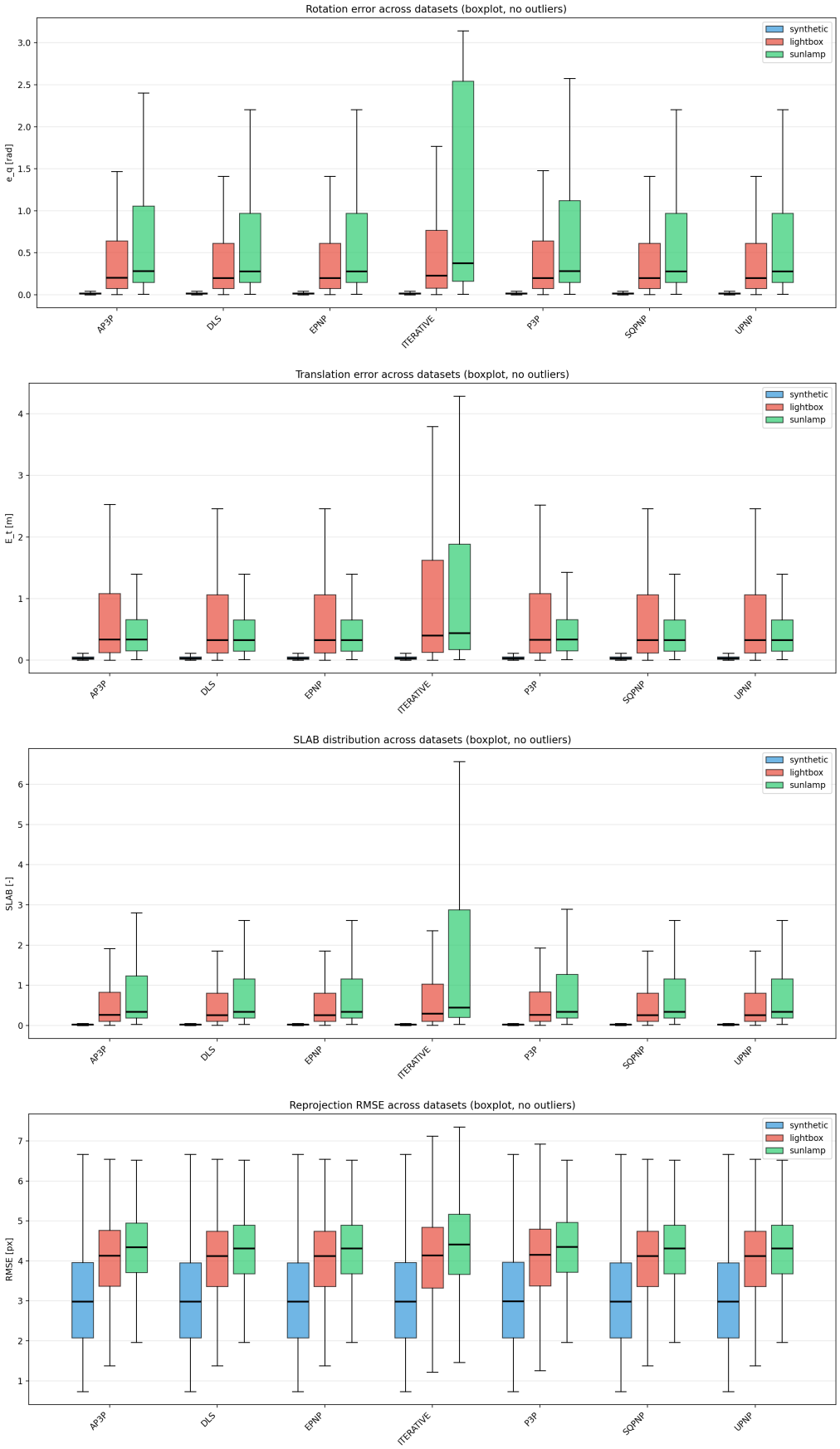
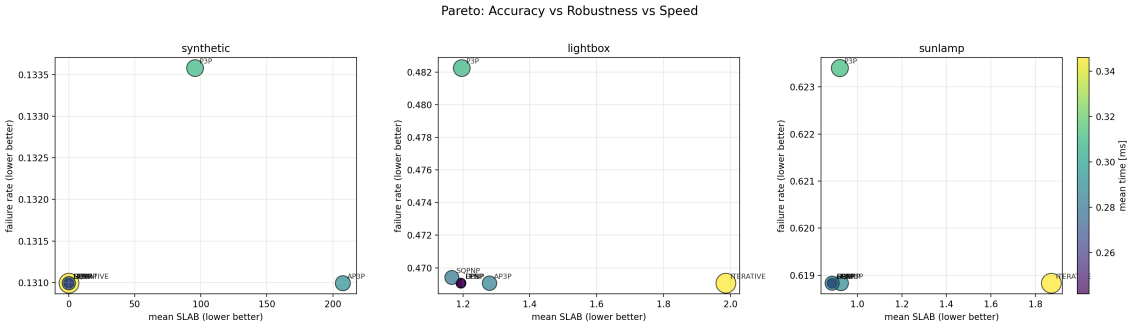
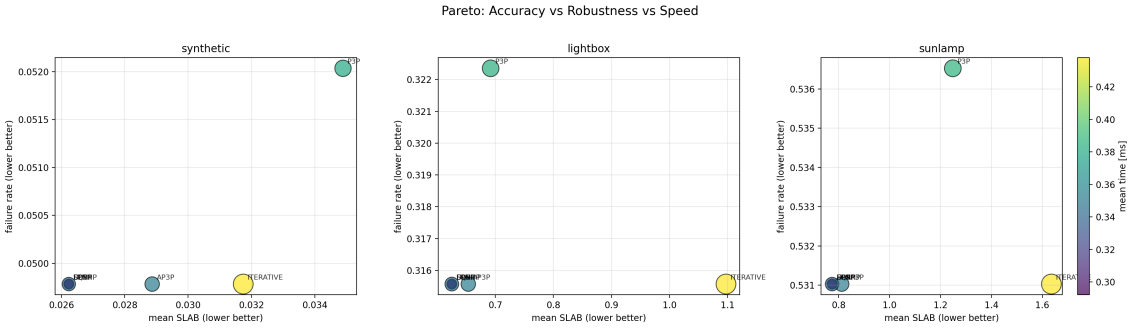


Figure 3.12: Boxplots of rotation error e_q, translation error E_t and SLAB score for each dataset (`synthetic`, `lightbox` and `sunlamp`) evaluated with different PnP using the YOLOv11-large model predictions.



(a) `YOLOv11n-pose`



(b) `YOLOv11l-pose`

Figure 3.13: Pareto analysis of the evaluated PnP solvers across the `synthetic`, `lightbox` and `sunlamp` domains. The horizontal axis represents the median SLAB score (accuracy), the vertical axis the failure rate (robustness), while marker size and colour represent the mean execution time (computational cost). Solvers located closer to the bottom-left region provide a more favourable trade-off between accuracy, robustness and speed.

3.4.5 Final pose evaluation with SQPnP

The final pose estimation results are obtained by combining the YOLOv11 keypoint predictions with the `SQPnP` solver. On the `synthetic` validation set, the proposed pipeline achieves accurate and stable pose estimates, confirming the effectiveness of the overall approach under nominal conditions.

On the `lightbox` and `sunlamp` test sets, performance degradation is observed, particularly in terms of rotational accuracy. This behaviour is primarily attributed to the reduced reliability of 2D keypoint localisation under extreme illumination conditions, rather than to intrinsic limitations of the PnP solver itself. These results highlight the critical role of robust keypoint detection in monocular pose estimation pipelines and motivate further investigation into HIL domains.

Metrics	VALIDATION		LIGHTBOX		SUNLAMP	
	YOLOv11-n	YOLOv11-l	YOLOv11-n	YOLOv11-l	YOLOv11-n	YOLOv11-l
E_t [m]	0.0965	0.0542	1.5978	1.0902	0.9989	0.6856
e_t [−]	0.0182	0.0125	0.2699	0.1686	0.1732	0.1186
e_q [rad]	0.0532	0.0259	1.0784	0.6408	1.1779	0.9911
e_q [°]	3.048	1.4840	61.7878	36.7151	67.4887	56.7858
SLAB [−]	0.0714	0.0383	1.3483	0.8094	1.3511	1.1097
RMSE [px]	12.8729	9.3605	42.0579	28.3207	51.9410	39.3016
PnP timing [fps]	20671.34	16315.55	10498.02	13063.80	8583.77	10007.76
Success rate [%]	99.99	99.98	84.39	45.93	78.39	82.23

Table 3.3: Pose estimation mean results for the first YOLOv11 training (no refinement), evaluated on the SPEED+ `validation` set and HIL domains, namely `lightbox` and `sunlamp`.

3.5 Refinement Process

The evaluation on the SPEED+ HIL domains (`lightbox` and `sunlamp`) highlighted a pronounced degradation in pose accuracy compared to the `synthetic` validation set. This section discusses two refinement strategies investigated with the objective of improving robustness and pose estimates' precision: a model-to-image alignment based on the Canny edge detection algorithm and a refinement method based on the Levenberg-Marquardt algorithm.

3.5.1 Refinement Strategy A: Top-K selection + edge-based alignment

The refinement procedure implemented in the evaluation script consists of two sequential steps: stabilising the pose estimate by removing inconsistent keypoints and refining the pose using image edges.

1. **Initial pose estimation and Top-K selection**

   Given the predicted 2D–3D correspondences $\{(\mathbf{X}_i, \mathbf{u}_i)\}_{i=1}^{N}$, an initial pose $(R_0, \mathbf{t}_0)$ is computed using `SQPnP`. Each 3D point is projected into the image according to

   $$\hat{\mathbf{u}}_i = \pi\left(K(R_0\mathbf{X}_i + \mathbf{t}_0)\right) \tag{3.24}$$

   where $\pi(\cdot)$ denotes perspective projection.

   For each correspondence, the reprojection error is computed as

   $$e_i = \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|_2 \tag{3.25}$$

The keypoints are then sorted according to e_i, and only the best K points (with the smallest reprojection error) are retained. This removes correspondences that are geometrically inconsistent with the majority.

A second `SQPnP` estimation is performed using only these Top-K pairs, yielding a more stable pose $(R_1, \mathbf{t}_1)$. This Top-K solution serves as the baseline pose for evaluation and as the initialisation for the edge-based refinement.

2. **Edge-based model-to-image refinement**

   Starting from $(R_1, \mathbf{t}_1)$, an additional refinement step aligns the projected 3D wireframe model with image contours.

   First, Canny edge detection [2] is applied to the image, producing a binary edge map. A distance transform $D(\mathbf{p})$ is then computed, assigning to each pixel $\mathbf{p}$ the distance to the nearest detected edge.

   On the model side, each 3D edge of the spacecraft wireframe is uniformly sampled, producing a set of 3D points $\{\mathbf{Y}_j\}_{j=1}^{M}$ lying along structural lines.

   For a candidate pose $(R, \mathbf{t})$, each sampled point is projected into the image:

$$\hat{\mathbf{v}}_j = \pi\left(K(R\mathbf{Y}_j + \mathbf{t})\right) \tag{3.26}$$

   The main residual for each projected point is defined as the distance transform value:

$$r_j^{edge} = D(\hat{\mathbf{v}}_j) \tag{3.27}$$

   The optimisation aims at minimising the sum of these distances,

$$E_{edge}(R, \mathbf{t}) = \sum_{j=1}^{M} D\big(\hat{\mathbf{v}}_j\big) \tag{3.28}$$

   encouraging projected model edges to lie closer to strong image contours. To prevent large and unstable corrections, two regularisation terms are introduced:

$$E_{rot} = \lambda_R \|\Delta R\|^2 \tag{3.29}$$

$$E_{trans} = \lambda_t \|\mathbf{t} - \mathbf{t}_1\|^2 \tag{3.30}$$

   where ΔR represents the deviation from the initial Top-K rotation, and λ_R, λ_t are weighting factors.

[2] The Canny edge detector is a classical multi-stage algorithm that extracts image contours by computing intensity gradients, applying non-maximum suppression, and performing double-threshold hysteresis filtering to produce a binary edge map. [56],[57]

The total objective function becomes:

$$E(R, \mathbf{t}) = E_{edge}(R, \mathbf{t}) + E_{rot} + E_{trans} \tag{3.31}$$

The optimisation is solved using non-linear least squares with a Huber loss, which reduces the influence of large residuals.

3. **Fallback rule**

   If optimisation converges, the refined pose $(R_{ref}, \mathbf{t}_{ref})$ is used. Otherwise, the algorithm falls back to the Top-K pose $(R_1, \mathbf{t}_1)$, guaranteeing a certain stability and robustness.

This two-stage strategy therefore operates at two different levels: the Top-K selection increases robustness at the correspondence level, while the edge-based optimisation attempts a controlled geometric correction at the image level without allowing large deviations from the initial estimate. Some visual examples of the edge-refinement process are shown in Figure 3.14.

3.5.2 Refinement Strategy B: robust reprojection refinement (LM on keypoints)

A second refinement strategy was investigated to directly improve the pose estimate by minimising the reprojection inconsistency of the 2D–3D correspondences. In contrast to Strategy A, which operates at the contour level, this approach refines the pose exclusively at the correspondence level through a local non-linear optimisation.

Figure 3.15 and 3.16 show some example of refined poses obtained using the refinement procedure.

In particular, the refinement process adopted in this case consists of the following steps.

1. **Initial pose estimation and residual computation**

   Starting from the predicted correspondences $\{(\mathbf{X}_i, \mathbf{u}_i)\}_{i=1}^{N}$, a robust initial pose $(R_0, \mathbf{t}_0)$ is obtained in two stages. First, an `EPnP` solver is included in a RANSAC loop in order to identify a geometrically consistent subset of inlier correspondences. Subsequently, a `SQPNP` solution is computed using the RANSAC inliers, yielding the initial pose estimate.

   Each 3D point is projected using 3.24, and the reprojection error is defined consistently with 3.25. These residuals quantify the geometric inconsistency between predicted keypoints and the current pose hypothesis.

2. **Weighted robust LM refinement**

The pose parameters are refined by minimising a robust reprojection objective over the inlier subset $\mathcal{S}$:

$$E_{repr}(R, \mathbf{t}) = \sum_{i \in \mathcal{S}} w_i \, \rho_\delta\big(e_i(R, \mathbf{t})\big), \tag{3.32}$$

where $\rho_\delta(\cdot)$ is a Huber loss defined as:

$$\rho_\delta(e) = \begin{cases} \frac{1}{2}e^2, & e \le \delta, \\ \delta\left(e - \frac{1}{2}\delta\right), & e > \delta. \end{cases} \tag{3.33}$$

The parameter δ controls the transition between quadratic and linear behaviour: for residuals $|e| < \delta$, the loss behaves as an ℓ_2 norm, while for $|e| > \delta$ it transitions to an ℓ_1-like regime. Smaller values of δ increase robustness to outliers, whereas larger values make the loss closer to a pure ℓ_2 formulation. Hence, this loss behaves quadratically for small residuals and linearly for large residuals.

The correspondence weights w_i combine two factors: the first factor is related to visibility and accounts for the annotation reliability, while the second is a pseudo-confidence term is derived from the reprojection residual under the initial pose:

$$c_i = \exp\left(-\left(\frac{e_i(R_0, \mathbf{t}_0)}{\sigma}\right)^2\right), \tag{3.34}$$

which reduces the influence of correspondences exhibiting large initial inconsistencies.

The final weight is defined as

$$w_i = w_i^{\text{vis}} \, c_i^2, \tag{3.35}$$

where w_i^{vis} encodes the visibility class of the keypoint.

Inspired by [33], [38], [40], [58], the optimisation is solved using the Levenberg–Marquardt[3] algorithm via a non-linear least-squares formulation. Let $\boldsymbol{\xi} \in \mathbb{R}^6$ denote a minimal pose parametrisation. Linearising the residual vector around

[3]The Levenberg–Marquardt algorithm is a damped non-linear least-squares optimisation method that iteratively combines Gauss–Newton updates with gradient descent to improve convergence and numerical stability [59].

the current estimate yields

$$\mathbf{r}(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}) \approx \mathbf{r}(\boldsymbol{\xi}) + \mathbf{J}\Delta\boldsymbol{\xi}, \tag{3.36}$$

where $\mathbf{J}$ is the Jacobian of the stacked reprojection residuals. The LM update solves the damped normal equations

$$\left(\mathbf{J}^\top \mathbf{W}\mathbf{J} + \lambda\mathbf{I}\right) \Delta\boldsymbol{\xi} = \mathbf{J}^\top \mathbf{W}\mathbf{r}, \tag{3.37}$$

with damping parameter $\lambda > 0$ and diagonal weight matrix $\mathbf{W}$.

3. **Acceptance criterion**

   The refined pose $(R_{\text{ref}}, \mathbf{t}_{\text{ref}})$ is accepted only if it decreases the reprojection RMSE, computed over the inlier subset $\mathcal{S}$, as defined in Equation 3.20:

   $$\text{RMSE}(R_{ref}\,; t_{ref}\,; \mathcal{S}) < \text{RMSE}(R_0\,; t_0\,; \mathcal{S})$$

   If no improvement is achieved, the algorithm falls back to the initial `SQPNP` estimate $(R_0, \mathbf{t}_0)$. This acceptance criterion ensures that the refinement process does not compromise the geometric consistency of the estimated pose.

This strategy therefore performs a controlled geometric refinement directly at the correspondence level. By combining RANSAC-based outlier rejection, visibility-aware weighting, pseudo-confidence modulation and robust loss minimisation, the method achieves a stable improvement of pose accuracy while limiting the impact of residual mismatches.

3.5.3 Refinement Process: Results Overview

Table 3.4 summarises the mean performance obtained on the `synthetic` validation set and on the two HIL test domains (`lightbox` and `sunlamp`). Two main trends can be highlighted.

The edge-based refinement provides only marginal benefits and, in several cases, it does not translate into measurable improvements of the final pose metrics. On the `synthetic` split, the method can occasionally correct small misalignments (Fig. 3.14), but the average improvement remain limited and unsteady across models and splits. On HIL domains, the refinement is even less effective: the rotation error e_q and the composite SLAB score remain essentially unchanged with respect to the `SQPnP` baseline, and in some configurations the RMSE increases, indicating that the optimisation often converges to locally consistent edge matches that do not correspond to the true spacecraft pose.

This behaviour can be explained by several considerations. First, the objective optimised by Strategy A relies on the assumption that dominant visible contours in the image correspond to the projected model wireframe. In HIL domains this assumption is frequently violated: illumination gradients, glare, blooming, and partial saturation create high-contrast edges that are not generated by the spacecraft geometry, while true geometric edges may be attenuated or fragmented. Second, Tango geometry exhibits repeated structures (solar panel edges, symmetric faces, and long straight segments) that lead to ambiguous edge alignments. When the initial pose is already affected by a significant rotational error, the distance-transform becomes highly non-convex. This means that multiple incorrect poses achieve similarly low edge cost by aligning different subsets of projected segments with spurious or repeated contours. In this setting, the refinement has a limited convergence region and tends to be driven by local minima rather than by the global geometric configuration.

In contrast, Strategy B consistently improves the results obtained with the `SQPnP` baseline. In the `synthetic` validation set, the method produces a clear reduction in the RMSE, which results in improved pose accuracy (lower E_t, e_t, and e_q) and a better SLAB score for both YOLOv11-n and YOLOv11-l. More importantly, on both HIL domains the method reduces the rotation error e_q and improves the SLAB score, particularly for the higher-capacity model. These improvements are coherent with the design goal of Strategy B: rather than relying on image gradients, it directly minimises the geometric inconsistency between predicted keypoints and the projected 3D model, while minimising large residuals through robust weighting and the Huber loss.

A relevant observation is that Strategy B may reduce the PnP success rate in some configurations (see Table 3.5). The LM refinement enforces a stronger notion of validity and improvement, prioritising reliability of accepted solutions over keeping a larger number of potentially poor estimates.

As a result of all the considerations outlined, Strategy B is selected as the reference refinement method and will be adopted for subsequent evaluations and trade-off analyses presented in the following sections.

Dataset	Metric	SQPnP		Model-to-Image		LM on keypoints	
		YOLOv11-n	YOLOv11-l	YOLOv11-n	YOLOv11-l	YOLOv11-n	YOLOv11-l
Validation	E_t [m]	0.0965	0.0542	0.0983	0.0473	0.0781	**0.0370**
	e_t [–]	0.0182	0.0125	0.0173	0.0097	0.0131	**0.0066**
	e_q [rad]	0.0532	0.0259	0.0525	0.0236	0.0461	**0.0192**
	SLAB [–]	0.0714	0.0383	0.0698	0.0333	0.0591	**0.0258**
	RMSE [px]	12.8729	9.3605	14.6329	10.8209	4.9323	**3.6819**
Lightbox	E_t [m]	1.5978	1.0902	1.4797	**0.9934**	2.3492	1.0841
	e_t [–]	0.2699	0.1686	0.2522	0.1545	0.3447	**0.1499**
	e_q [rad]	1.0784	0.6408	1.0815	0.6415	0.9789	**0.5391**
	SLAB [–]	1.3483	0.8094	1.3337	0.7959	1.3236	**0.6890**
	RMSE [px]	42.0579	28.3207	48.1550	33.1218	6.2982	**5.8105**
Sunlamp	E_t [m]	0.9989	**0.6856**	0.9642	0.7561	0.9270	0.7210
	e_t [–]	0.1732	0.1186	0.1688	0.1328	0.1330	**0.1061**
	e_q [rad]	1.1779	0.9911	1.1853	1.0677	0.8842	**0.8076**
	SLAB [–]	1.3511	1.1097	1.3541	1.2005	1.0173	**0.9138**
	RMSE [px]	51.9410	39.3016	59.3307	51.4781	6.4362	**6.3807**

Table 3.4: Refinement mean results across all of the SPEED+ datasets.

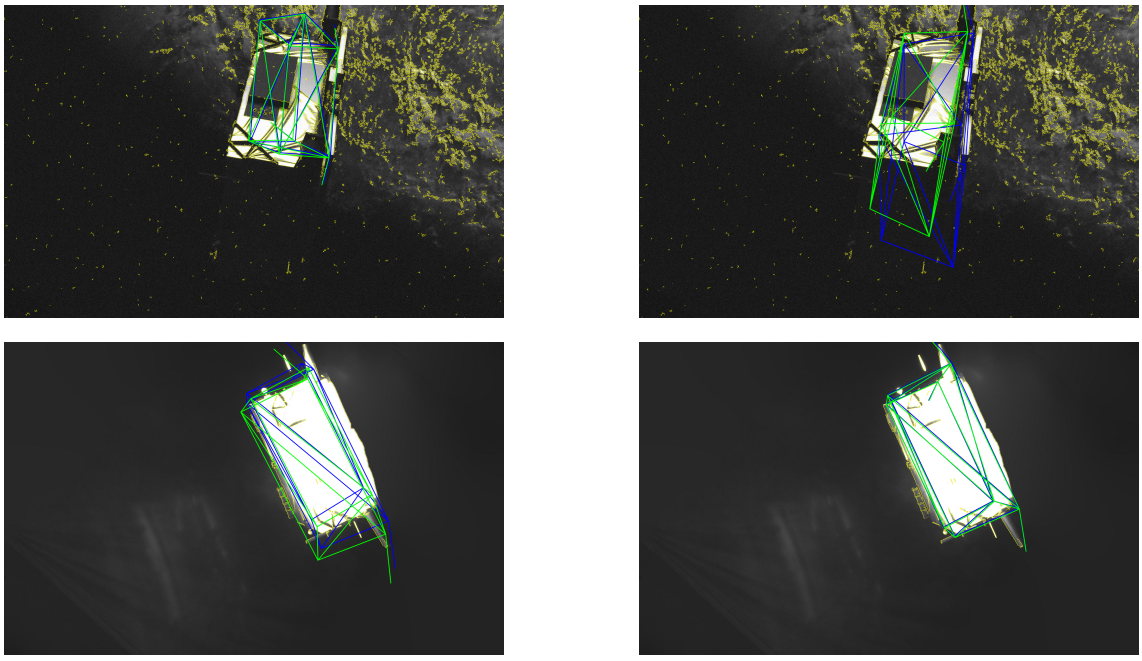
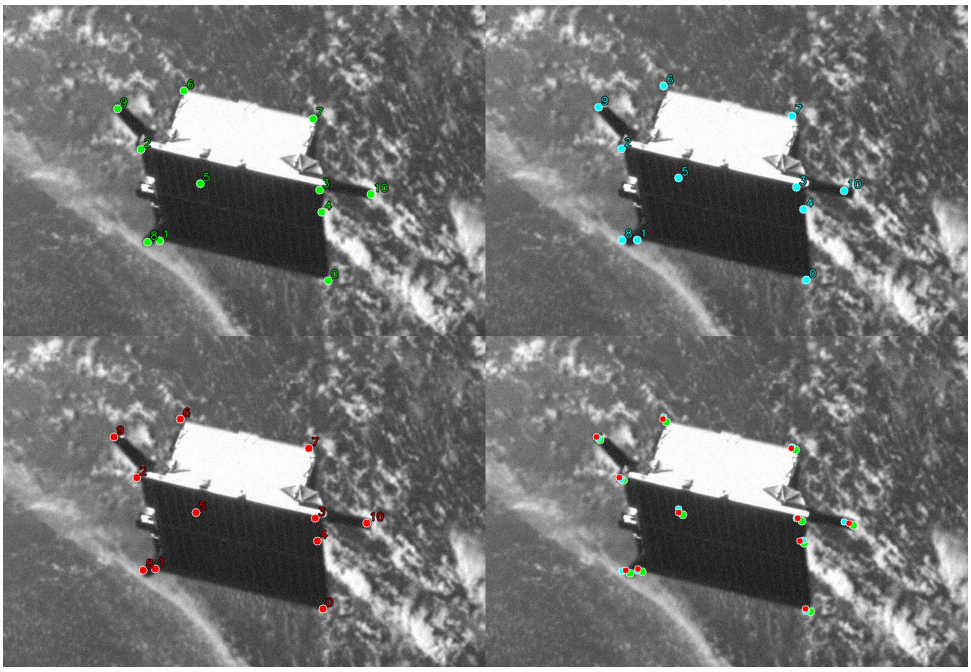
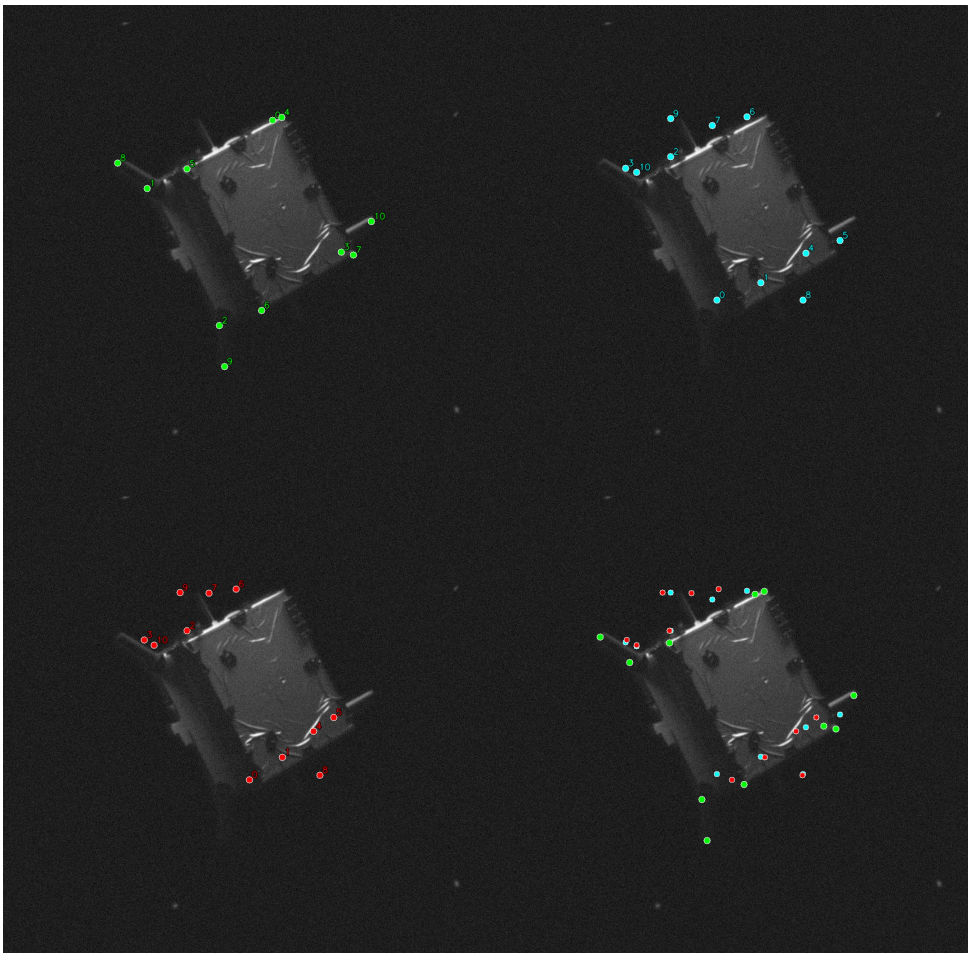


Figure 3.14: The figures show the improvement obtained with the edge-refinement process for `lightbox` (top) and `sunlamp` (bottom), for both the YOLOv11n model (left) and the YOLOv11l model (right). The green mask represents the initial pose (R_0, t_0), while the blue mask represents the refined pose (R_{ref}, t_{ref}).

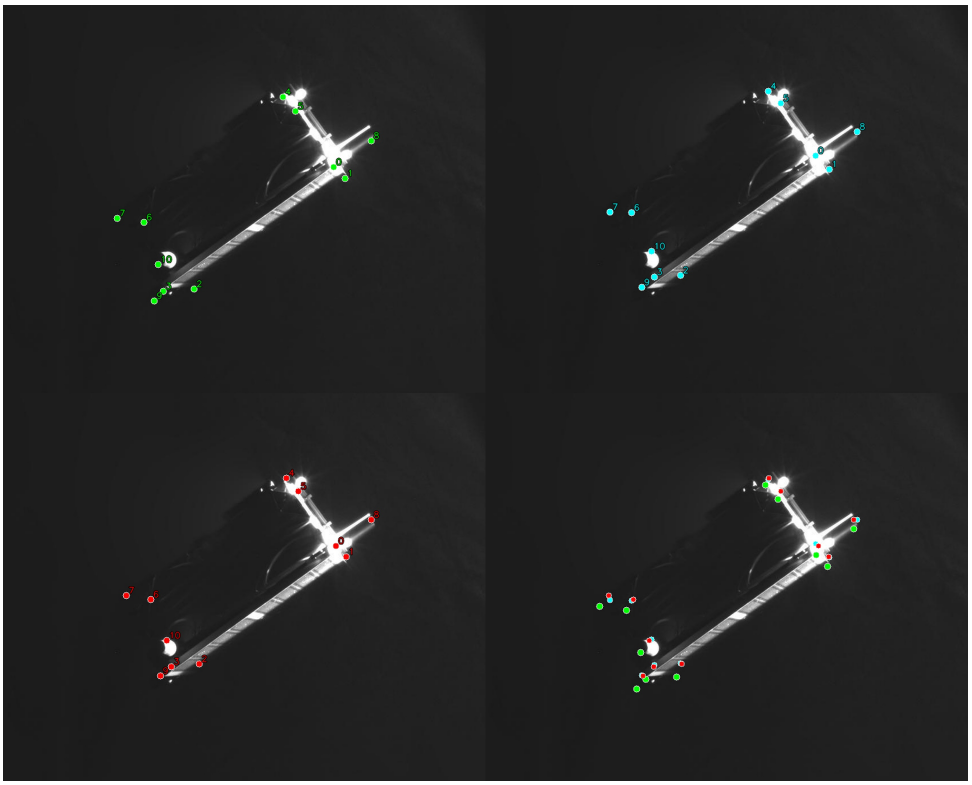


(a) `lightbox best`

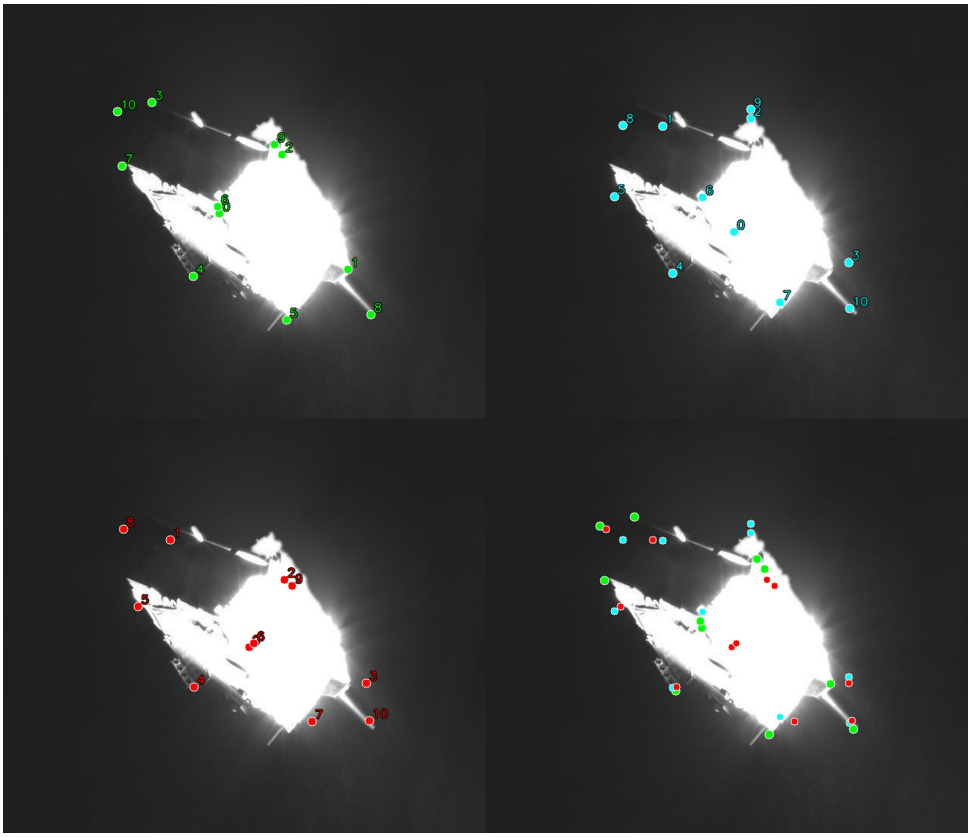


(b) `lightbox worst`

Figure 3.15: Visual examples of best and worst predictions and LM refinements on the `lightbox` domain, using the YOLOv11-l model. The pictures show ground-truth keypoint (green), predicted correspondences (blue) and refined pose (red).



(a) `sunlamp best`



(b) `sunlamp worst`

Figure 3.16: Visual examples of best and worst predictions and LM refinements on the `sunlamp` domain, using the YOLOv11-l model. The pictures show ground-truth keypoint (green), predicted correspondences (blue) and refined pose (red).

3.6 Results Overview

Table 3.5 summarises the overall pose estimation performance of the proposed pipeline, defined accordingly to Section 2.3.1. In addition to the geometric accuracy metrics, the table includes a set of indicators assessing the robustness of the detection, inference, and refinement stages. These indicators are defined as follows.

The *Detection Rate* is computed as:

$$\text{Detection Rate} = \frac{N_{\text{YOLO detections}}}{N_{\text{total images}}} \times 100 \tag{3.38}$$

where $N_{\text{YOLO detections}}$ denotes the number of images for which the detector produced a valid prediction file.

The *Success Rate* represents the percentage of samples for which a valid pose estimate is produced and it is computed as:

$$\text{Success Rate} = \frac{N_{\text{valid poses}}}{N_{\text{YOLO detections}}} \times 100 \tag{3.39}$$

The *LM Convergence Rate* is defined as:

$$\text{LM Convergence} = \frac{N_{\text{LM converged}}}{N_{\text{valid poses}}} \times 100 \tag{3.40}$$

while the *LM Improvement Rate* quantifies the percentage of cases in which the refinement stage effectively reduces the reprojection error:

$$\text{LM Improvement Rate} = \frac{N_{\text{SLAB improved}}}{N_{\text{valid poses}}} \times 100 \tag{3.41}$$

On the `synthetic` set, both models achieve high geometric accuracy. The large variant consistently outperforms the nano model, with a reduction of the mean rotational error from 2.63° to 1.10° and a halving of the translation error (0.078 m to 0.037 m). The improvement is also reflected in the SLAB metric and in the reprojection error RMSE.

Significant performance degradation is observed when moving to the HIL domains: rotational errors increase up to 56° in `lightbox` for YOLOv11-n, and both SLAB and translation metrics worsen by more than one order of magnitude compared to the `synthetic` set. This behaviour highlights the impact of domain shift, as discussed in depth previously.

The Detection Rate further clarifies the influence of domain conditions. While both models reach 100% on the `synthetic` set, a marked decrease is observed in the HIL domains. In `lightbox`, YOLOv11-l exhibits a significantly lower Detection Rate

(45.93%) compared to YOLOv11-n (84.39%). This suggests that the larger model, although more accurate when a detection is produced, may be more sensitive to domain mismatch and confidence thresholds, leading to a reduced number of valid prediction files.

In `sunlamp`, both models show comparable Detection Rates, indicating that illumination variability affects them in a more similar manner in this domain.

The LM Convergence Rate closely follows the Success Rate across all domains, indicating that the optimisation process converges in nearly all samples. This confirms the numerical stability of the LM refinement once a geometrically consistent pose has been provided by the PnP stage.

A more informative indicator is the LM improvement rate. On the `synthetic` dataset, the refinement stage improves the SLAB metric in approximately 55%–64% of the evaluated samples. In the `lightbox` domain, the improvement rate decreases to 27%–37%, while in `sunlamp` it drops further to around 20%–28%.

This behaviour suggests that the refinement stage is most effective when the initial pose estimate is already reasonably close to the correct solution. Under more challenging HIL conditions, where the initial rotational error can be large, the LM optimisation often converges but does not necessarily lead to an improvement in the final pose estimate.

Metrics	VALIDATION		LIGHTBOX		SUNLAMP	
	YOLOv11-n	YOLOv11-l	YOLOv11-n	YOLOv11-l	YOLOv11-n	YOLOv11-l
E_t [m]	0.078	0.037	2.349	1.084	0.927	0.684
e_t [−]	0.013	0.007	0.345	0.150	0.133	0.100
e_q [rad]	0.046	0.019	0.979	0.539	0.884	0.778
e_q [°]	2.630	1.101	56.087	30.888	50.662	44.558
SLAB [−]	0.059	0.026	1.324	0.689	1.017	0.878
RMSE [px]	4.932	3.738	6.298	5.811	6.436	6.358
Detection Rate [%]	100	100	84.39	45.93	78.39	82.23
Success Rate [%]	93.75	98.27	71.69	80.26	55.53	67.15
LM Convergence Rate [%]	93.70	98.26	71.68	80.26	55.53	69.85
LM Improvement Rate [%]	55.08	64.32	27.62	37.37	19.70	27.76

Table 3.5: Pose estimation metrics for the YOLOv11 model with LM refinement, evaluated on the SPEED+ `validation` set and HIL domains, namely `lightbox` and `sunlamp`.

In conclusion, these results highlight three key aspects. First, model capacity significantly affects geometric accuracy and robustness to domain shift, with YOLOv11-l consistently outperforming the nano variant. Second, the detection stage remains the primary bottleneck in HIL domains, as reflected by the reduced Detection Rate. Third, the implementation of the LM refinement improves the overall pose estimation accuracy, as deeply discussed in Section 3.5.3.

3.6.1 Runtime Evaluation

A dedicated runtime analysis was conducted in order to provide a comprehensive assessment of the operational feasibility of the proposed pipeline under realistic deployment conditions.

It is essential to distinguish between offline and online stages of the system. The training phase was executed offline and therefore does not impact real-time operational performance. On the other hand, inference and pose estimation constitute the online processing stages and directly determine the achievable performance in a practical implementation. Specifically:

- the network inference was executed on a NVIDIA GeForce RTX 2080 Ti GPU;
- the geometric pose estimation (PnP and LM refinement) was executed on an Intel(R) Xeon(R) Silver 4216 CPU @ 2.10 GHz.

Table 3.6 and Figure 3.17 illustrate the runtime breakdown, expressed in FPS, accounting for the inference and evaluation stages for each dataset split.

		VALIDATION		**LIGHTBOX**		**SUNLAMP**	
		YOLOv11-n	**YOLOv11-l**	**YOLOv11-n**	**YOLOv11-l**	**YOLOv11-n**	**YOLOv11-l**
Inference	FPS	72.499	46.52	70.988	47.257	72.537	47.841
	s	165.437	257.823	94.946	142.625	38.477	58.339
EVAL	FPS	74.622	96.734	23.862	27.691	9.984	17.049
	s	150.667	121.829	170.903	89.741	121.689	90.385
TOTAL	FPS	21.278	20.251	10.589	6.762	5.935	8.416
	s	528.378	581.955	385.111	367.468	204.711	183.102

Table 3.6: YOLOv11 runtime performance comparison across datasets.

As expected, inference speed remains nearly constant across domains, since it depends primarily on the network architecture and hardware configuration rather than on dataset characteristics.

In contrast, the evaluation stage exhibits a marked variability across domains. The `synthetic` split achieves significantly higher total runtime values, whereas the HIL domains show a substantial reduction in total FPS. Quantitatively, the `synthetic` split is approximately 2–4 times faster than the HIL domains, depending on the model configuration.

This degradation can be attributed to the increased computational load during the evaluation in synthetic-to-real shifting conditions, where a higher number of challenging cases, refinement attempts, and failure handling procedures contribute to longer CPU processing times.

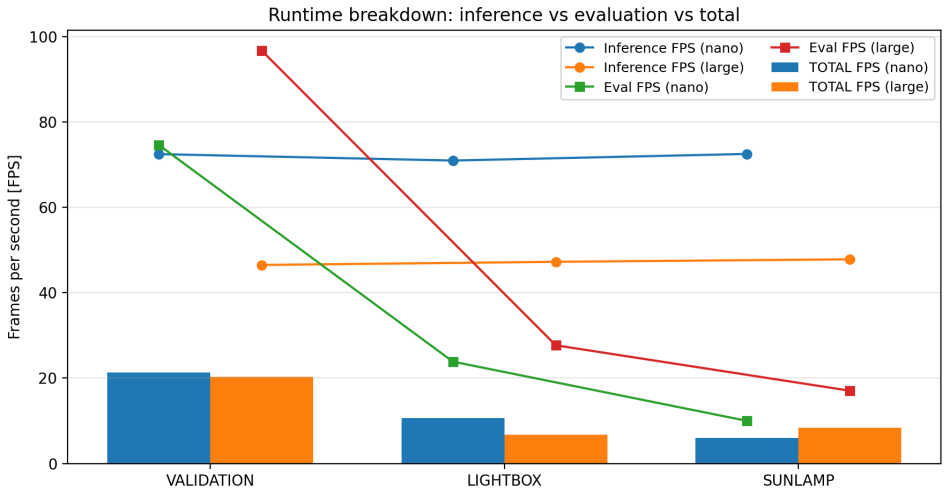


Figure 3.17: Runtime overview for each dataset split (`synthetic`, `lightbox` and `sunlamp`

3.6.2 Error Distribution vs Relative Distance

Figure 3.18 shows the mean translation and rotation errors (for the YOLOv11-n and the YOLOv11-l models, respectively) as a function of the relative target distance, for both `lightbox` and `sunlamp` domains.

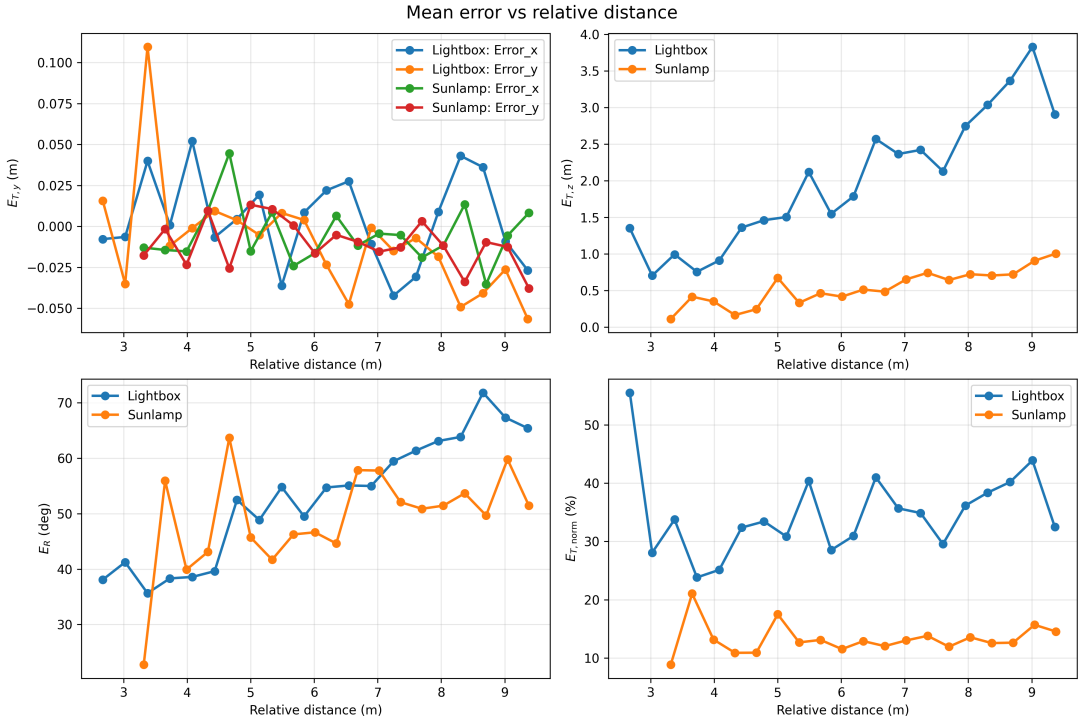
A clear distance-dependent trend can be observed across all configurations. While the lateral translation components E_x and E_y remain centred around zero with limited systematic bias, the depth component E_z exhibits a pronounced growth as the relative distance increases. This behaviour is consistent with perspective projection geometry: as the target moves farther from the camera, its apparent size decreases, reducing the geometric constraints available to the PnP solver. Consequently, small keypoint localisation errors are amplified along the depth direction.

A similar monotonic increase is observed in the rotation error E_R. At larger distances, the projected keypoints occupy a smaller image region, increasing rotational uncertainty. This confirms that the dominant degradation mechanism at long range is geometric conditioning rather than numerical instability of the solver.

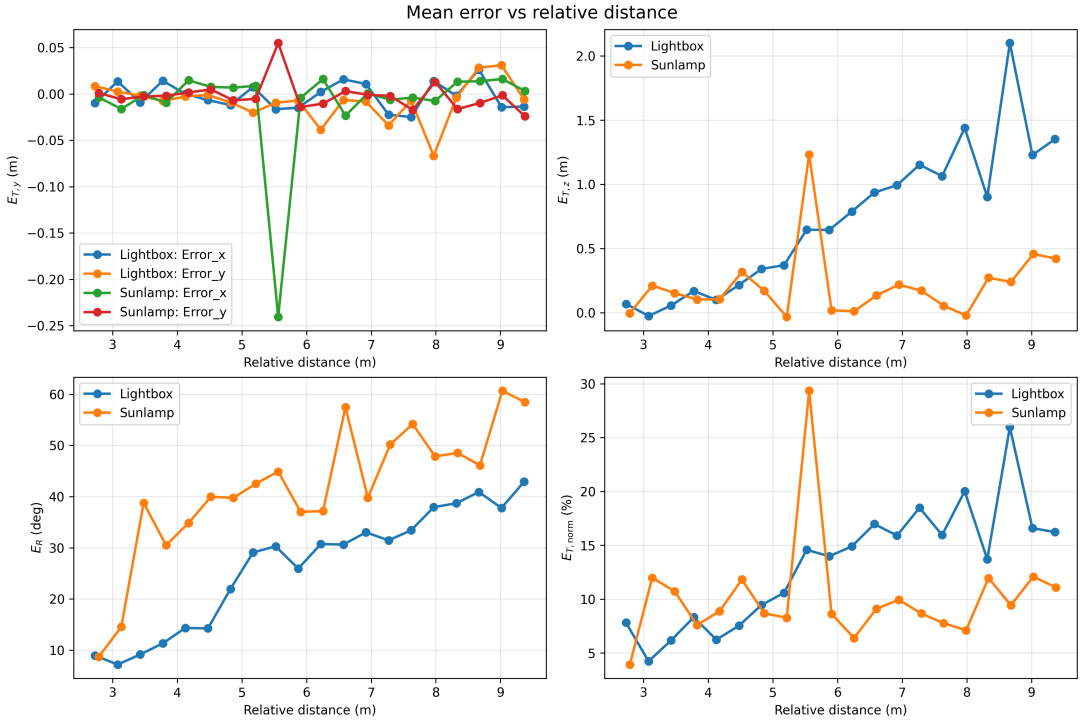
When comparing the two domains, a clear performance gap emerges. The `lightbox` configuration generally shows larger rotational and depth errors than `sunlamp`, particularly at mid-to-large distances. This difference can be attributed to harsher illumination challenges, which affect keypoint detection reliability and introduce structured localisation noise.

The normalised translation error $E_{T,\text{norm}}$ (expressed as a percentage of the ground-truth distance) provides further clarification of this behaviour. Although normalisation compensates for the geometric scaling effect, the relative error still increases

with distance. This indicates that performance degradation at larger ranges is due not only to perspective scaling but also to the progressive amplification of keypoint localisation noise and reduced geometric constraints in the projection.



(a) YOLOv11-n



(b) YOLOv11-l

Figure 3.18: Mean and normalised translation errors and mean rotation error as a function of the relative target distance.

3.6.3 Final Considerations

The experimental analysis presented in this chapter highlights both the strengths and limitations of the proposed YOLOv11-pose pipeline for 6-DOF pose recovery on the SPEED+ dataset.

From an accuracy perspective, the model achieves competitive performance in terms of rotational error e_q, normalised translation error e_t, and SLAB score, particularly in the `synthetic` domain. However, as observed in the `lightbox` and `sunlamp` subsets, the degradation in performance under domain shifts indicates a sensitivity to illumination variations. This behaviour is consistent with the challenges already reported in the literature for keypoint-based monocular pose estimation.

When compared with SOTA methods summarised in Table 2.3, it can be noted that the lowest absolute errors are typically achieved by heavy high-resolution architectures such as HRNet-w48 or SPNv3 variants, which involve significantly larger backbone networks and higher computational complexity. In contrast, YOLOv11-pose provides a substantially lighter architecture.

A key advantage of the proposed approach lies in its computational efficiency. The inference stage, executed on a GPU, achieves real-time performances, while the subsequent pose recovery stage is executed on CPU with a limited computational budget. This enables the complete pipeline to operate at frame rates compatible with on-board GNC requirements, where low latency and deterministic execution times are critical.

Therefore, although the proposed method does not reach the absolute lowest errors compared to SOTA, it offers a favourable trade-off between accuracy, robustness, and runtime performance. This balance is particularly relevant for autonomous proximity operations, where real-time processing capability can be as important as marginal improvements in pose accuracy.

The limitations observed in keypoint localisation precision and robustness under HIL conditions motivate the investigation of alternative high-resolution architectures in the following chapter. In particular, HRNet-based models are explored as a means to enhance spatial localisation accuracy while preserving geometric consistency in the PnP stage.

Chapter 4

HRNet-based Pipeline

This chapter presents the high-resolution convolutional pipeline developed for monocular spacecraft pose estimation on the SPEED+ dataset.

Instead of directly regressing the 6-DOF pose parameters, the proposed approach follows an indirect formulation, decomposing the task into two sequential stages. In the first stage, an HRNet-based backbone predicts K dense heatmaps, each representing the spatial likelihood distribution for a predefined semantic keypoint on the spacecraft. In the second stage, 2D keypoints extracted from the heatmaps are fed to a PnP solver, which recovers the full relative pose.

The quality of the predicted heatmaps directly impacts the numerical conditioning and robustness of the pose estimation stage. For this reason, the keypoint detector is designed and trained to ensure accurate localisation, sharp and unambiguous peaks, and consistent geometry under domain shifts.

4.1 Input Preprocessing and Coordinate Systems

This section describes the full preprocessing chain applied to SPEED+ images before they are provided to the HRNet network. A precise formulation is necessary to correctly perform the coordinate mappings between the different spatial grids used for supervision and pose recovery.

4.1.1 Network Input Standardisation

SPEED+ images are provided with original resolution $H_{\text{orig}} \times W_{\text{orig}} = 1920 \times 1200$, which is significantly higher than the spatial resolution used for the CNN training and inference. As a matter of fact, to enable stable batching and computationally feasible training, each frame is resized to a fixed square resolution of 512×512:

$$\mathbf{I}_{\text{orig}} \in \mathbb{R}^{1920\times1200\times3} \quad \longrightarrow \quad \mathbf{I} \in \mathbb{R}^{512\times512\times3}$$

After resizing, images are normalised and converted to tensor format

$$\mathbf{x} \in \mathbb{R}^{B\times3\times512\times512}$$

4.1.2 Heatmap Supervision Grid and Coordinate Mapping

Keypoint supervision is performed through Gaussian heatmaps defined on a fixed grid of resolution $H_m \times W_m = 128 \times 128$. Therefore, annotated keypoints in resized-image coordinates $(x, y) \in [0, 511]^2$ are mapped to the heatmap grid (u, v) through a scale transformation:

$$u = s_x\, x, \qquad v = s_y\, y, \qquad s_x = \frac{W_m}{512}, \;\; s_y = \frac{H_m}{512}$$

In the adopted setting $s_x = s_y = \frac{1}{4}$.

During inference, predicted heatmaps are converted to 2D keypoint coordinates. Let $(\hat{u}, \hat{v})$ denote the coordinates extracted from the predicted heatmap. These are mapped back to the resized-image pixel grid as:

$$\hat{x} = \frac{\hat{u}}{s_x}, \qquad \hat{y} = \frac{\hat{v}}{s_y} \quad \Rightarrow \quad \hat{x} = 4\hat{u}, \;\; \hat{y} = 4\hat{v} \tag{4.1}$$

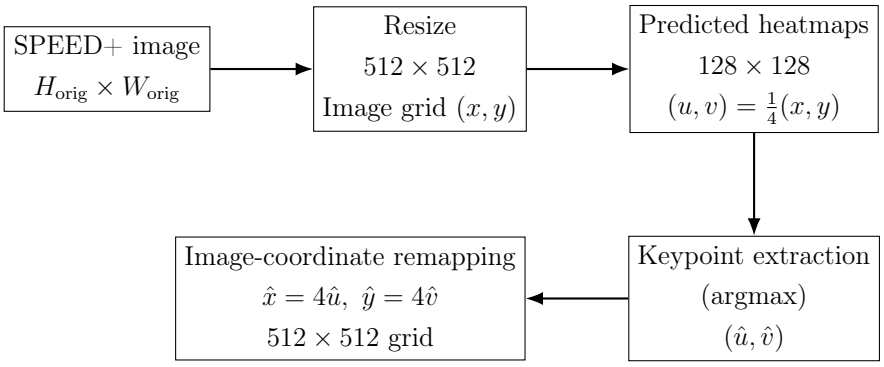


Figure 4.1: Representation of the spatial transformations used in the HRNet-based keypoint detection pipeline. The original SPEED+ image is resized to 512×512, the network predicts keypoint heatmaps on a 128×128 grid, and the final keypoint coordinates are obtained via argmax and remapped to the resized image coordinate system.

4.2 HRNet Architecture

The keypoint detector is based on HRNet, a convolutional architecture designed to preserve high-resolution representations throughout the entire feature extraction process. Unlike conventional encoder-style CNNs that progressively downsample spatial resolution, HRNet maintains parallel multi-resolution branches and repeatedly exchanges information across them.

Starting from a single high-resolution stream, additional lower-resolution branches are introduced in subsequent stages. At each stage, cross-scale fusion units integrate features from all resolutions, enabling the network to combine spatially precise representations with semantically richer low-resolution features.

This architectural design is particularly well suited for dense prediction tasks such as keypoint localisation, where spatial accuracy is critical.

In this work, the HRNet-w32 variant, initialised with ImageNet pre-trained weights, is employed as backbone. For a resized input of 512×512, the selected high-resolution branch produces feature maps of size 128×128, corresponding to an effective stride of 4 pixels. This feature tensor is subsequently processed by a shared convolutional head to generate K keypoint heatmap logits.

4.2.1 Backbone Configuration

The backbone is instantiated through the `timm` implementation in `features_only` mode, which exposes intermediate feature tensors:

$$\{\mathbf{F}^{(i)}\}_{i=0}^{4}, \qquad \mathbf{F}^{(i)} \in \mathbb{R}^{B \times C_i \times H_i \times W_i}. \tag{4.2}$$

where B denotes the batch size, C the number of feature channels (in this case $C = 1$, as SPEED+ images are in grey scale), and $H \times W$ the spatial resolution of the feature grid.

In the adopted configuration, the feature tensor used for dense keypoint regression has spatial resolution 128×128, matching the supervision grid (Sec. 4.1.2). With a 512×512 resized input, this corresponds to an effective spatial stride with respect to the resized input of 4 pixels.

[0]In dense prediction, the term *stride* commonly denotes the ratio between the input spatial resolution and the feature-map resolution. A $512 \rightarrow 128$ mapping corresponds to a stride of 4, meaning that adjacent feature locations sample neighbouring 4×4 pixel regions.

4.2.2 Keypoint Prediction Head

Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, the network predicts a set of K heatmaps, one for each keypoint

$$\hat{\mathbf{H}} = \{\hat{H}_k\}_{k=1}^{K}, \qquad \hat{H}_k \in \mathbb{R}^{H_m \times W_m} \tag{4.3}$$

On top of the selected backbone feature tensor, a shared convolutional head is employed to regress K keypoint heatmap logits:

$$\hat{\mathbf{H}} = \mathrm{Head}(\mathbf{F}), \qquad \hat{\mathbf{H}} \in \mathbb{R}^{B \times K \times 128 \times 128}. \tag{4.4}$$

Rather than using a single 1×1 projection, the adopted architecture consists of a module of 3×3 convolutional blocks, each defined as:

$$\mathbf{F}_{j+1} = \sigma(\mathrm{BN}(\mathrm{Conv}_{3\times3}(\mathbf{F}_j))), \tag{4.5}$$

where $\mathrm{Conv}_{3\times3}$ denotes a convolution with kernel size 3×3 and unit stride (with padding to preserve spatial resolution), $\mathrm{BN}(\cdot)$ denotes batch normalisation, which stabilises training by normalising feature statistics across the mini-batch, and $\sigma(\cdot)$ is a rectified linear activation function (ReLU), introducing non-linearity into the representation.

After the refinement stage, a final 1×1 convolution projects the refined features to K heatmap logits:

$$\hat{\mathbf{H}} = \mathrm{Conv}_{1\times1}(\mathbf{F}_3) \qquad \hat{\mathbf{H}} \in \mathbb{R}^{B \times K \times 128 \times 128}$$

This design ensures architectural simplicity and parameter efficiency while still allowing sufficient representational capacity to model fine spatial structures.

Figure 4.2 shows qualitative examples of the predicted heatmaps compared to the ground-truth ones.

4.2.3 Keypoint Coordinate Extraction

Once the K heatmaps are predicted, the pipeline requires a single 2D location per keypoint to build 2D–3D correspondences for PnP.

During inference and pose evaluation, keypoints are extracted using a discrete $\arg\max$ operation[1]:

$$(\hat{u}_k, \hat{v}_k) = \arg\max_{(u,v)} \hat{H}_k(u, v) \tag{4.6}$$

[1]The $\arg\max$ operator returns the argument of a function at which its maximum value is attained. In this context, it identifies the spatial grid location (u, v) corresponding to the highest activation of the predicted heatmap.

followed by the heatmap-to-image mapping in Eq. 4.1. The resulting coordinates $(\hat{x}_k, \hat{y}_k)$ are therefore expressed in the resized image reference frame and can be directly used to establish 2D–3D correspondences with the known spacecraft keypoints defined in the object model.

Once the full set of predicted keypoints is obtained, the 6-DOF pose of the target spacecraft is recovered through a PnP formulation. Finally, the estimated pose is evaluated using the official SPEED+ metrics defined in Section 2.3.1.

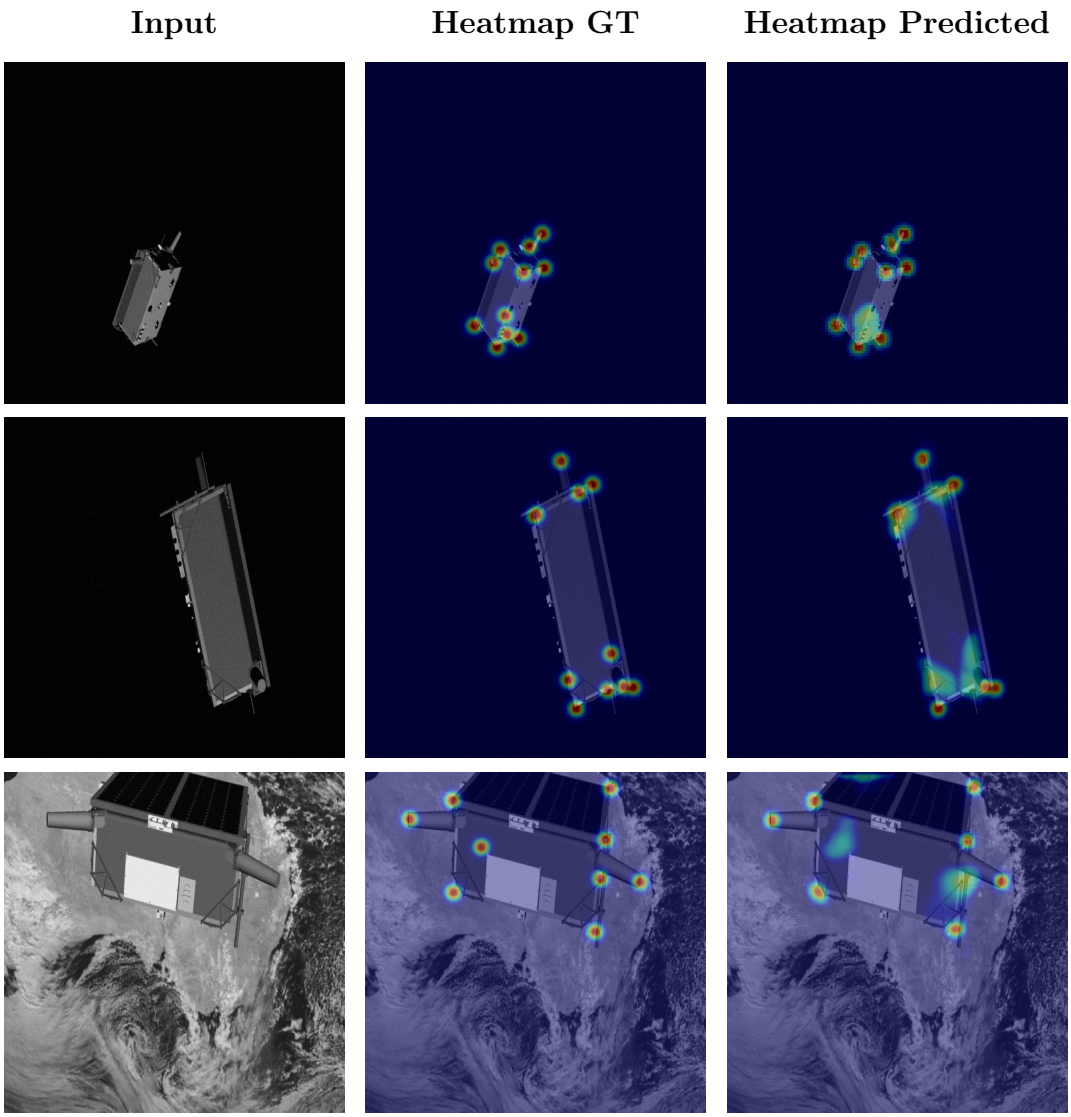


Figure 4.2: Example of keypoint heatmaps. From left to right: input image, ground-truth heatmaps, and predicted heatmaps. High-intensity regions correspond to the estimated keypoint locations.

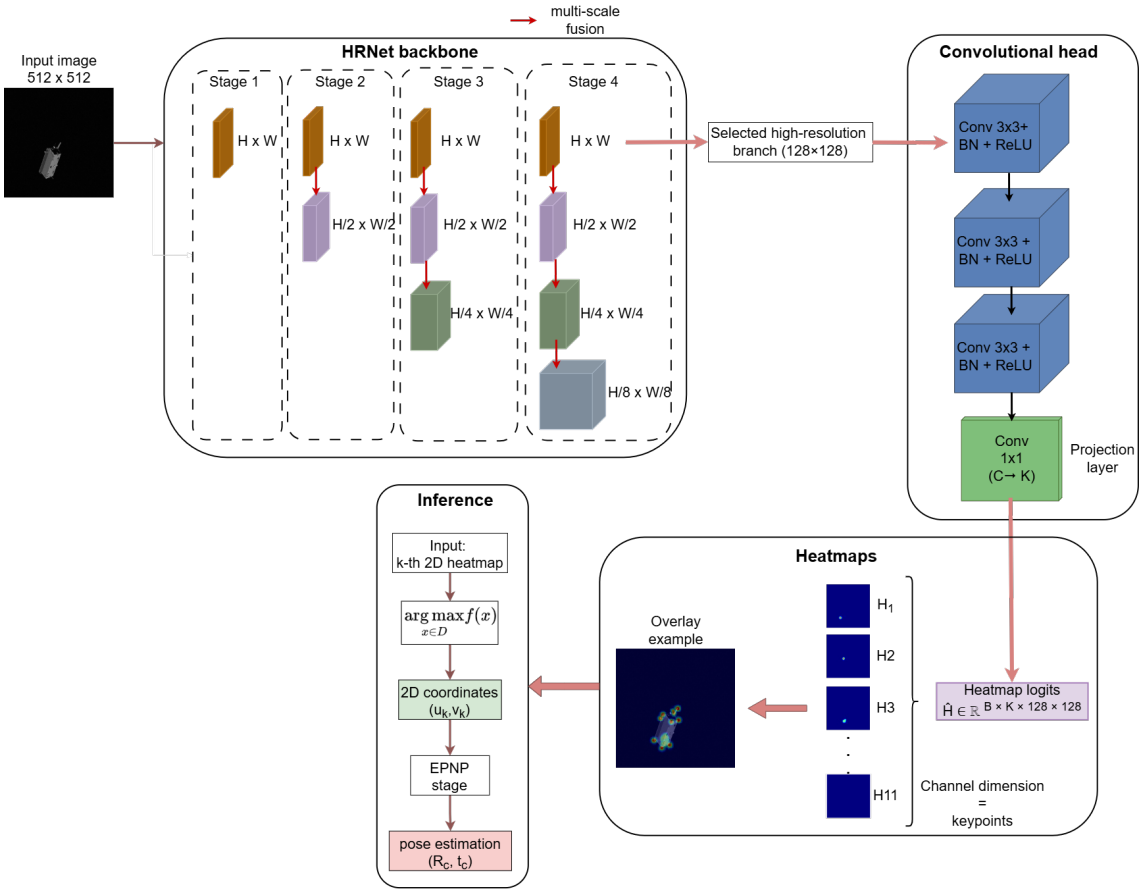


Figure 4.3: HRNet-based CNN architecture

4.3 Loss Function Definition

The definition of the loss function plays a fundamental role in the overall robustness of the pose estimation pipeline, as it controls how localisation accuracy and geometric consistency are enforced during the training process.

In heatmap-based keypoint detection, supervision can be applied at different levels: directly on the dense heatmap representation, on the extracted 2D coordinates, or through a combination of both.

However, an excessively strong coordinate constraint may destabilise heatmap learning, whereas heatmap-only supervision may produce visually sharp responses that are not optimally aligned for the pose estimation process.

For this reason, in this work, three progressively refined training strategies were investigated:

1. Heatmap loss only
2. Heatmap loss + Coordinate loss
3. Heatmap loss + Coordinate loss scheduling

Each configuration was evaluated not only in terms of keypoint localisation behaviour, but also considering and comparing the final pose estimation accuracy, evaluated through the official SPEED+ metrics.

4.3.1 Strategy 1: Heatmap loss only

The first training strategy relies exclusively on heatmap regression. Each keypoint is supervised through a Gaussian heatmap, and the network is trained to minimise the discrepancy between predicted and ground-truth heatmaps.

Initial experiments revealed a degenerate behaviour: predicted heatmaps tended to become nearly constant, with no clear spatial peaks. To address this issue, a *Weighted Mean Squared Error* (weighted MSE) formulation was adopted.

Let $\mathbf{H}_k$ denote the ground-truth heatmap and $\hat{\mathbf{H}}_k$ the predicted heatmap for keypoint k. The heatmap loss is defined as:

$$\mathcal{L}_{\text{HM}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{H_m W_m} \sum_{u,v} w_k(u,v) \left(\hat{H}_k(u,v) - H_k(u,v) \right)^2 \tag{4.7}$$

where $w_k(u,v)$ assigns larger weights to pixels close to the Gaussian peak and smaller weights to background regions.

Under this formulation, heatmaps transitioned from nearly uniform representations to well-localised peaks, confirming that the network had begun discriminating individual keypoints.

This configuration therefore constitutes a strong baseline, particularly in terms of localisation precision on samples successfully processed by the PnP stage.

4.3.2 Strategy 2: Heatmap loss + Coordinate loss

While heatmap regression ensures spatially structured predictions, it does not explicitly constrain the final 2D keypoint coordinates extracted from the heatmaps. Given these considerations, a coordinate-level loss was introduced.

Let $(\hat{x}_k, \hat{y}_k)$ denote the predicted keypoint coordinates extracted via soft-argmax during training, and (x_k, y_k) the ground-truth coordinates. The Huber loss introduced in Section 3.5.2 is used with $\delta = 1$, corresponding to the standard Smooth-ℓ_1 formulation. In reality, this value combines the stability of an ℓ_2 loss near zero with the robustness of an ℓ_1 loss for larger errors.

The coordinate loss is therefore formulated as follows:

$$\mathcal{L}_{\text{coord}} = \frac{1}{K} \sum_{k=1}^{K} \text{SmoothL1}\left((\hat{x}_k, \hat{y}_k), (x_k, y_k) \right) \tag{4.8}$$

The total loss can be expressed as the following linear combination:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{HM}} + \lambda_{\text{coord}}\mathcal{L}_{\text{coord}}$$

where scalar coefficient λ_{coord} controls the relative influence of coordinate supervision with respect to dense heatmap regression.

Since coordinate errors are expressed in pixel units of the resized image (512×512), their natural scale is significantly larger than that of the heatmap regression term. To prevent the coordinate term from dominating the optimisation process, coordinate errors are normalised with respect to the image dimension, which is equivalent to imposing $\lambda_{\text{coord}} = 1/512$.

Soft-argmax temperature

To apply coordinate-level supervision using gradient-based optimisation, the mapping from heatmaps to keypoint coordinates must be differentiable. The discrete $\arg\max$ operator used during inference is not differentiable, since small variations in the heatmap values may produce discontinuous changes in the selected peak location.

For this reason, during training a differentiable approximation known as *soft-argmax* is adopted.

Given the predicted heatmap logits $\hat{H}_k(u, v)$, a spatial softmax defines a discrete probability distribution over the heatmap grid:

$$\pi_k(u, v) = \frac{\exp\left(\beta\, \hat{H}_k(u, v)\right)}{\sum_{u',v'} \exp\left(\beta\, \hat{H}_k(u', v')\right)} \tag{4.9}$$

where $\beta > 0$ is a temperature parameter controlling the sharpness of the distribution.

The predicted keypoint location in heatmap coordinates is then computed as the expectation of this spatial distribution:

$$\hat{\mathbf{p}}_k^{hm} = \sum_{u,v} \pi_k(u, v)\, [u, v]^\top = [\hat{u}_k, \hat{v}_k]^\top \tag{4.10}$$

The resulting coordinates are subsequently mapped back to image pixel coordinates using the heatmap-to-image mapping defined in Sec. 4.1.2.

4.3.3 Strategy 3: Heatmap Loss + Coordinate Loss Scheduling

Directly introducing coordinate supervision from the first training epoch proved ineffective. At early stages of training, heatmaps are poorly structured and often multimodal, leading to the production of unreliable coordinates. Applying the coordinate loss at this stage introduces noisy gradients into the optimisation process.

To mitigate this issue, a scheduling strategy was introduced:

$$\mathcal{L}_{\text{total}}(t) = \mathcal{L}_{\text{HM}} + \lambda(t)\mathcal{L}_{\text{coord}}$$

where $\lambda(t)$ evolves over training epochs. This weighting factor dependent on time is defined as:

$$\lambda(t) = \lambda_{max} \cdot \min\left(1, \max\left(0, \frac{t - E_w}{E_r}\right)\right) \tag{4.11}$$

where E_w is the number of epochs during which only heatmap supervision is active, E_r is the number of ramp epochs, and λ_{max} is the final value for the coordinate loss weight.

Specifically:

- For the first E_w epochs, $\lambda(t) = 0$ (heatmap loss only);
- Over the subsequent ramp period, $\lambda(t)$ gradually increases up to a maximum value λ_{max};
- At the end of the ramp period, $\lambda(t) = \lambda_{max}$.

This progressive activation ensures that coordinate refinement is applied only after the network has learnt meaningful spatial heatmap structures.

An additional reliability mechanism was introduced: coordinate loss is applied only to keypoints whose predicted heatmap peak exceeds a confidence threshold. Let $p_k = \max_{u,v} \hat{H}_k(u, v)$. The coordinate loss is applied only if:

$$p_k > \tau_{peak}$$

This prevents poorly formed or flat heatmaps from corrupting the coordinate supervision outcome.

4.3.4 Results and Formulation Comparison Overview

In order to exploit a fair comparison between all three strategies, the evaluation of the results obtained was performed only on the common samples for which a pose estimate was generated and metrics were computed.

Specifically, given a set of methods $\{M_i\}$ and denoting by $\mathcal{S}_i$ the set of samples for which method M_i outputs valid pose metrics, the common subset is defined as the intersection

$$\mathcal{S}_{\text{common}} = \bigcap_i \mathcal{S}_i$$

Table 4.1 summarises the results obtained from this intersection analysis.

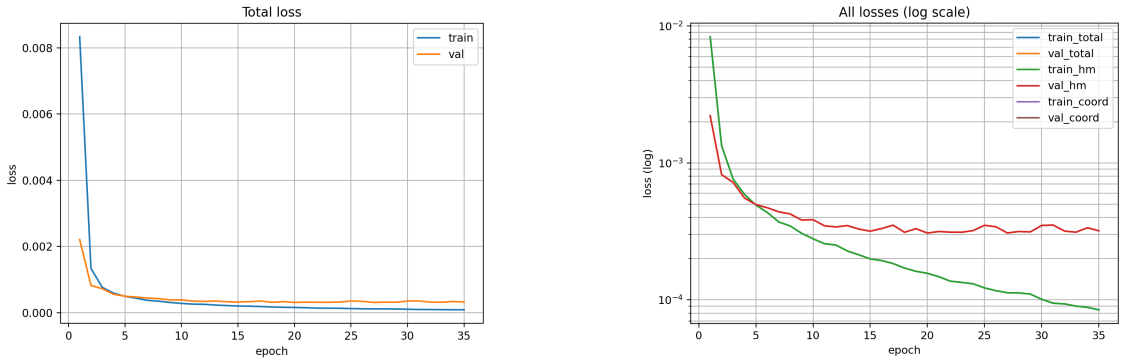
The comparative evaluation highlights an important and counter-intuitive result: the heatmap-only formulation yields the most robust overall performance. This outcome contradicts the intuitive expectation that direct coordinate supervision should necessarily improve metric pose accuracy.

In fact, although coordinate supervision theoretically enforces direct metric accuracy, it introduces a point-wise constraint that depends on soft-argmax extraction. This may increase the sensitivity of the optimisation process to small spatial biases and peak confidence variations.

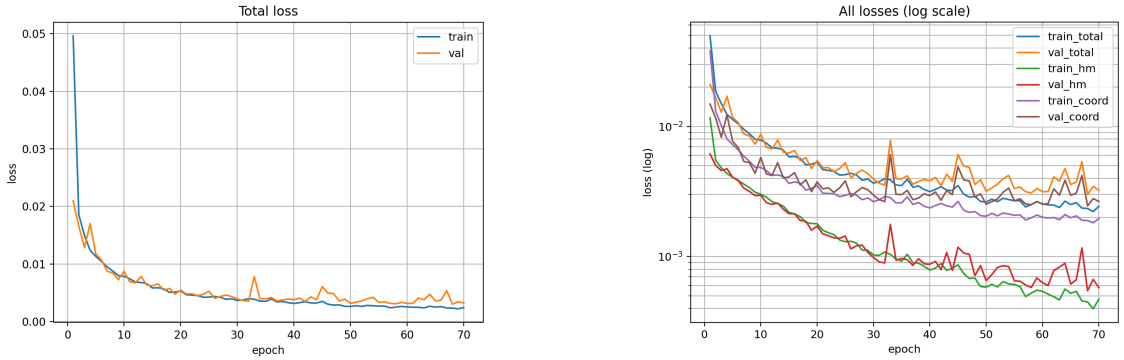
In contrast, heatmap regression provides dense spatial supervision, preserving spatial coherence more effectively, and reducing the occurrence of unstable or biased configurations.

With regard to coordinate loss strategies specifically, compared to the fixed-weight formulation, the scheduling strategy improved the overall robustness, resulting in a more stable PnP behaviour. The scheduling strategy, in fact, successfully mitigates the instability of the fixed coordinate loss, confirming that early coordinate supervision can corrupt heatmap formation. However, even with scheduling, coordinate-level constraints do not produce a consistent improvement over the heatmap-only baseline.

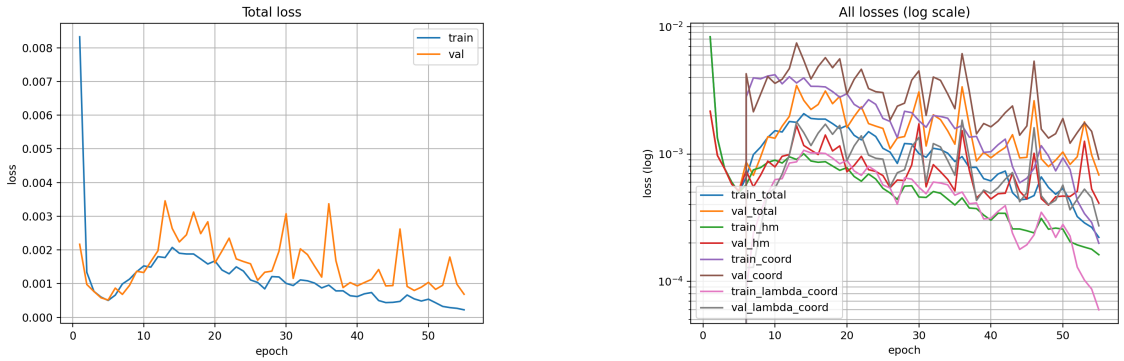
This suggests that, for the considered pose estimation pipeline, dense heatmap supervision already provides sufficient localisation accuracy, while maintaining superior robustness. For this reason, the heatmap loss formulation was adopted for the following ablation studies.



Training 1 – Heatmap loss only



Training 2 – Heatmap loss + coordinate loss



Training 3 – Heatmap loss + scheduled coordinate loss

Figure 4.4: Loss evolution during training for the three investigated loss formulations. For each training configuration, the total loss is shown in linear scale (left), while the individual loss components are reported in logarithmic scale (right) to highlight their relative magnitudes and convergence behaviour.

Strategy	E_T **[m]**	e_t **[-]**	e_q **[rad]**	RMSE **[px]**	**SLAB [-]**	**Coverage [%]**
HM-only	**0.04447**	**0.00687**	**0.01958**	**1.45178**	**0.02615**	99.15
Coord-fixed	0.06249	0.00989	0.02560	3.11467	0.03541	99.14
Coord-scheduled	0.05349	0.00826	0.02593	1.88862	0.03394	98.85

Table 4.1: SPEED+ metric results obtained on the intersection samples (in the `validation` dataset split) resulting from the comparison of all three loss formulation studies. The common valid subset accounts for 11801 samples (98.39% of intersection).

4.4 Data augmentation

To reduce the synthetic-to-real gap observed between the `synthetic` training split and the HIL domains (`lightbox` and `sunlamp`), a dedicated data augmentation strategy was applied during training.

The augmentation pipeline is implemented with `Albumentations` and it is organised into independent transformation blocks, each activated with a predefined probability. During training the blocks are evaluated sequentially. As a result, multiple perturbations may be applied on the same image. This results in combined variations affecting geometry, illumination, colour statistics and sensor characteristics. Such a design enables controlled diversification of the training distribution while preserving label consistency.

Geometric perturbations (probability p_{geom}) include small affine transformations (translations, in-plane rotations, isotropic scaling and shear) and mild perspective distortions. These operations simulate residual misalignment between the camera optical axis and the target, as well as minor calibration imperfections. Importantly, geometric transforms are applied consistently to both the image and the associated 2D keypoints. After transformation, ground-truth heatmaps are regenerated from the augmented keypoints to ensure supervision consistency.

Illumination variability is handled through a photometric augmentation block (probability p_{photo}). This introduces random brightness and contrast scaling, gamma correction, tone-curve perturbations and histogram-based contrast enhancement (e.g., CLAHE and equalisation). In practice, these transformations approximate exposure variations and non-linear intensity mappings observed in real sensors.

A complementary colour perturbation block further modifies hue, saturation and RGB channel balance. This is particularly relevant in HIL setups where illumination spectra and surface reflectance properties differ from idealised conditions.

To reproduce acquisition artifacts typical of physical imaging systems, a dedicated block (probability p_{noise}) introduces additive Gaussian noise, ISO-like noise

models and JPEG compression artifacts. These perturbations approximate electronic sensor noise and low-resolution effects.

Gaussian and motion blur simulate optical defocus and relative motion between camera and target, whereas sharpening counterbalances excessive smoothing.

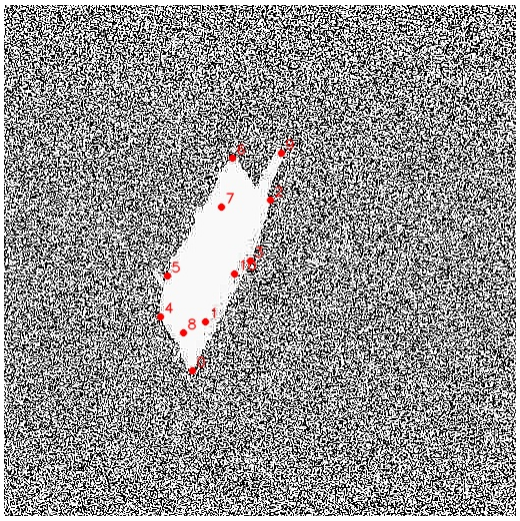
Robustness to partial visibility is promoted through coarse dropout, which introduces randomly positioned black patches of variable size. In practice, this mechanism encourages the network to rely on distributed geometric cues rather than individual keypoints.

In addition, a solar flare augmentation is employed to approximate glare patterns observed in the `sunlamp` domain. This is particularly relevant for spaceborne vision systems, where strong directional light sources can induce lens artifacts and local saturation.

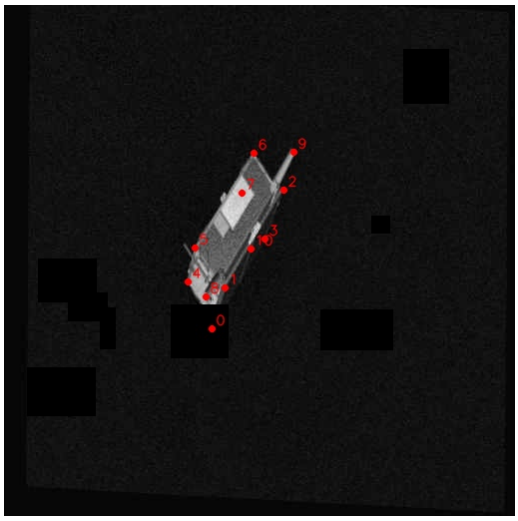
All augmentations are applied exclusively to the training split. Validation and test sets remain unaltered to ensure a fair and reproducible evaluation of pose estimation performance. This separation avoids contamination of the evaluation protocol and enables an objective assessment of generalisation capability.

Block	**Activation Probability**	**Main Operations and Intensity Range**
Geometry	$p_{\text{geom}} = 0.30$	Affine jitter (translation ±4%, scale 0.92–1.08, rotation ±8°, shear ±4°) and mild perspective distortion.
Photometric Core	$p_{\text{photo}} = 0.95$	Brightness/contrast scaling (±0.6), gamma correction ($\gamma \in [70, 160]$), tone curve perturbations, CLAHE and histogram equalisation.
Colour Perturbations	0.55	Hue (±10), saturation (±20), value (±15) shifts, RGB channel shift (±15), colour jitter.
Solar Flare	$p_{\text{flare}} = 0.20$	Simulated glare patterns within upper image region (6–10 flare circles).
Noise & Compression	$p_{\text{noise}} = 0.85$	Additive Gaussian noise (std 0.02–0.12), ISO-like noise, JPEG compression (quality 25–85).
Blur / Sharpen	0.75	Gaussian blur (kernel 3–9), motion blur (kernel 7), median blur (kernel 5), sharpening (controlled alpha and lightness).
Occlusion (Coarse Dropout)	$p_{\text{occl}} = 0.30$	3–9 random black patches, size 14–90 px.
Style-like	$p_{\text{style}} = 0.20$	Greyscale conversion and posterisation (4–6 bits).

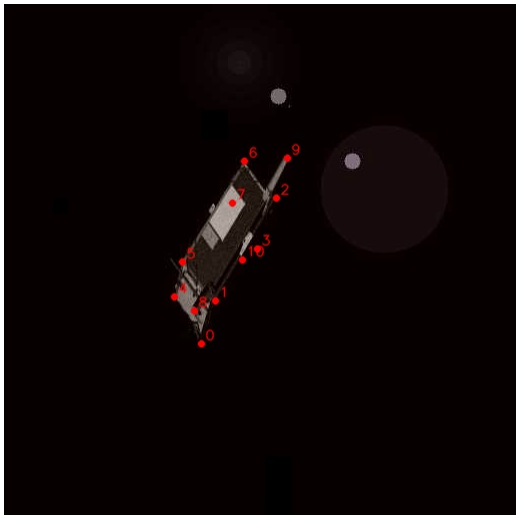
Table 4.2: Domain-shift augmentation strategy overview. The table also shows the activation probability value for each type of augmentation applied.



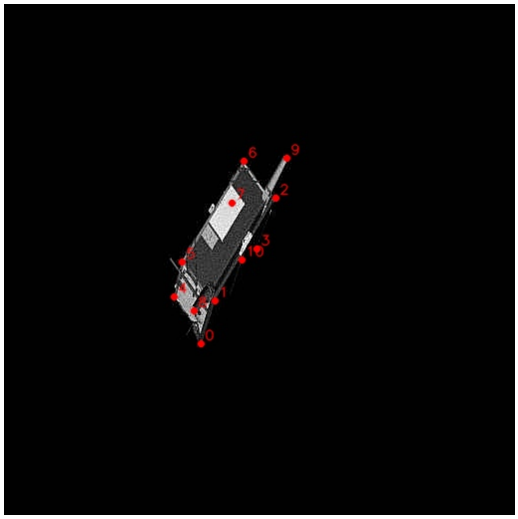
(a) *Perspective + Equalise + Colour Jitter*



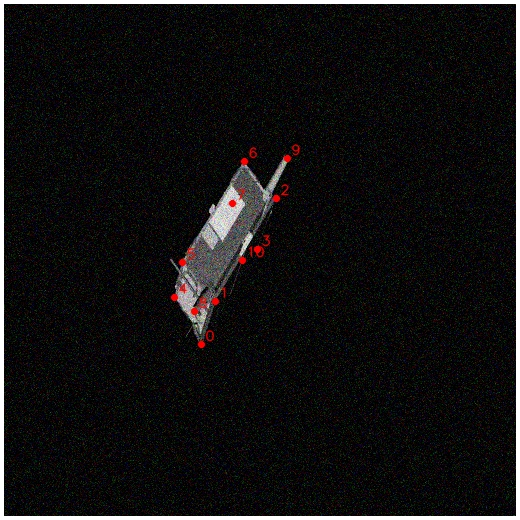
(b) *Affine + CLAHE + JPEG compression + Occlusion*



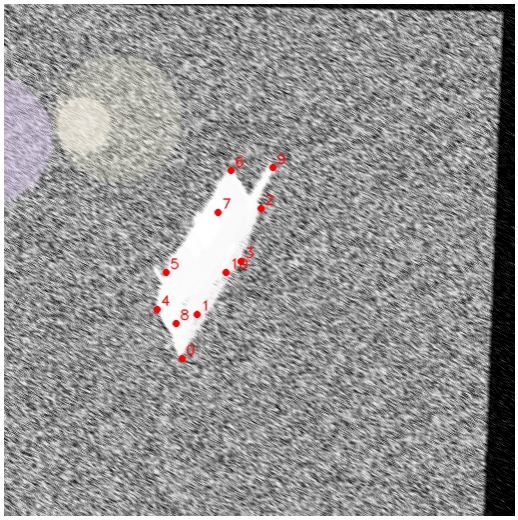
(c) *Brightness/Contrast + RGB shift + Sun Flares*



(d) *Tone Curve + JPEG compression + Sharpen*



(e) *Tone Curve + HSV Shift + Gaussian Noise*



(f) *Affine + Equalise + Sun Flares + ISO Noise*

Figure 4.5: Examples of augmented samples.

4.5 Bounding Box Cropping

In order to increase the effective spatial resolution of the target object and reduce background clutter, an additional cropping operation is optionally introduced in the preprocessing pipeline. The crop is defined by a bounding box enclosing the spacecraft keypoints and is applied *after* the global image resizing step. The geometric transformations introduced by the crop-based preprocessing are summarised in Figure 4.6.

Let the bounding box be defined on the resized image grid as

$$\mathbf{b} = [x_{\min},\, y_{\min},\, x_{\max},\, y_{\max}],$$

where $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ denote the top-left and bottom-right coordinates respectively. Given a point (x, y) expressed in resized image pixel coordinates, cropping introduces a translation of the coordinate origin:

$$x^c = x - x_{\min}, \qquad y^c = y - y_{\min},$$

where (x^c, y^c) represents the point coordinates in the cropped reference frame.

After cropping, the extracted image patch of size (w_c, h_c) [2] is resized back to the network input resolution $(W, H) = (512, 512)$. This introduces an additional anisotropic scaling transformation

$$x^{c'} = s_x\, x^c, \qquad y^{c'} = s_y\, y^c,$$

with scale factors

$$s_x = \frac{W}{w_c}, \qquad s_y = \frac{H}{h_c}.$$

Therefore, the overall transformation between the resized image coordinates (x, y) and the cropped image coordinates $(x^{c'}, y^{c'})$ can be expressed as

$$\begin{bmatrix} x^{c'} \\ y^{c'} \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{\min} \\ y_{\min} \end{bmatrix} \right).$$

This transformation is applied consistently to both the input image and the associated keypoint annotations during training and inference. The cropping procedure effectively enlarges the spacecraft footprint in the input image, increasing the spatial separation between keypoints and improving the conditioning of the subsequent

[2] The crop width and height are defined as $w_c = x_{\max} - x_{\min}$ and $h_c = y_{\max} - y_{\min}$, corresponding to the spatial extent of the bounding box in the resized 512×512 image grid.

pose estimation problem.

During inference, the network predicts keypoints in the cropped image coordinate system. In order to perform geometric evaluation and pose estimation in a common reference frame, these coordinates must be mapped back to the resized image domain.

The inverse transformation is therefore applied:

$$x = \frac{x^{c'}}{s_x} + x_{\min}, \qquad y = \frac{y^{c'}}{s_y} + y_{\min}.$$

In addition, the intrinsic camera matrix must be adapted to the cropped image geometry. The cropping step modifies the principal point and the effective focal length according to

$$f'_x = s_x f_x, \qquad f'_y = s_y f_y,$$

$$c'_x = (c_x - x_{\min}) s_x, \qquad c'_y = (c_y - y_{\min}) s_y$$

This adjusted intrinsic matrix is used when solving the PnP problem in the cropped image domain. The pose estimated by the PnP solver is expressed in the camera reference frame and is therefore independent of the image coordinate system. Only the 2D keypoint coordinates are remapped to the resized image domain for visualisation and evaluation purposes.

Overall, the crop-based pipeline acts as a deterministic geometric transformation that preserves the projective structure of the problem while improving the effective spatial resolution of the target spacecraft.

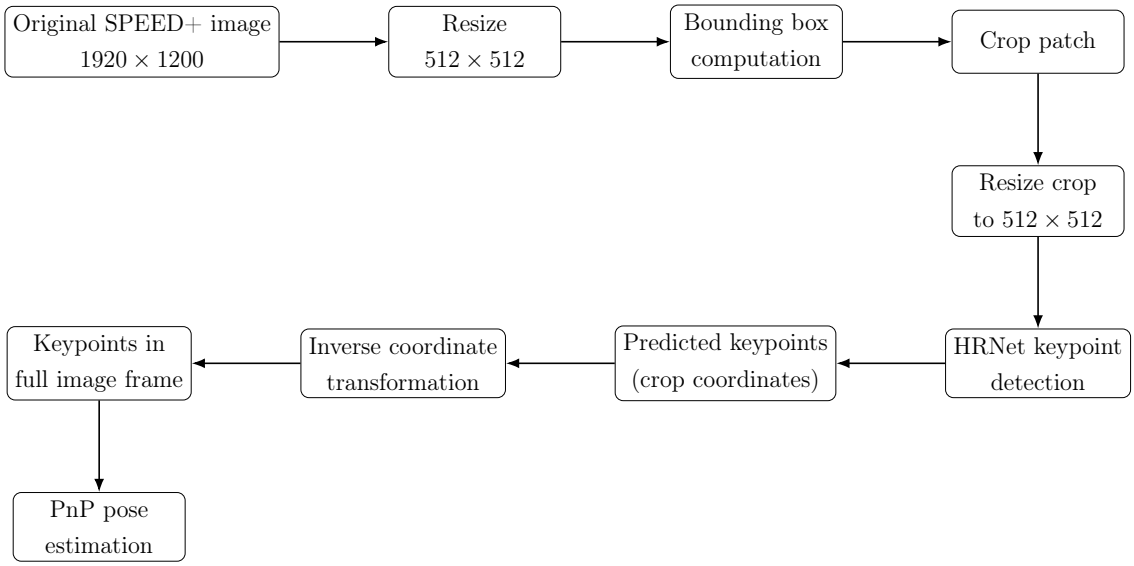


Figure 4.6: Pipeline of the crop-based preprocessing and coordinate reprojection used in the HRNet pose estimation framework.

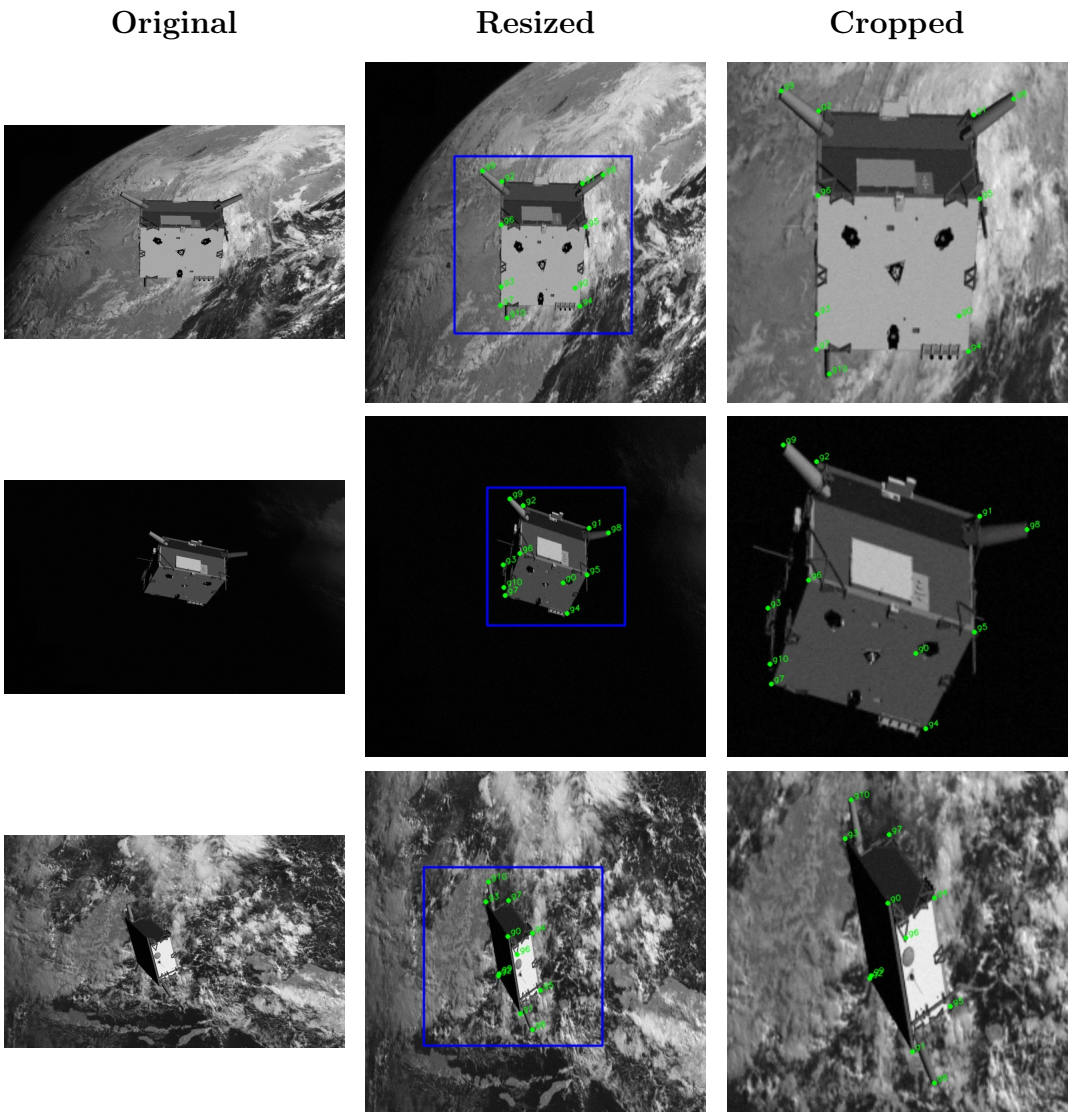


Figure 4.7: Examples of bounding-box cropping applied to SPEED+ images. The first column shows the original frame, the second column shows the resized image with the detected bounding box, and the third column shows the cropped patch used as network input.

4.6 Post processing refinements

4.6.1 Translation Threshold

During the initial evaluation of the pose estimation pipeline, some samples produced unusually large metric values, which appeared inconsistent with the general behaviour of the model. A preliminary inspection of the results revealed that these anomalies were associated with extremely large translation errors generated by the PnP solver.

In order to understand the origin of these values, a dedicated analysis of the affected samples was performed. The objective was to determine whether these failures were caused by a systematic issue in the dataset or by isolated instability cases in the pose reconstruction process. As a result, a statistical analysis of the normalised translation error was performed. Such failures typically occur when the geometric configuration of the detected keypoints becomes nearly degenerate. In these situations, small localisation errors in image space can produce large variations in the recovered translation when solving the PnP problem.

The analysis reported in Fig. 4.8 highlights the presence of a spike in the mean translation error. While the binned mean reaches approximately 8.8%, the corresponding median remains close to 1% and the 90^{th} percentile stays below 3.5%. A detailed percentile inspection shows that 95% of the samples exhibit $E_{T,norm} < 5.6\%$, whereas a very small number of extreme outliers generate extremely large errors, with a maximum value of approximately 743%.

The discrepancy between mean and median therefore reveals a heavy–tailed error distribution, where rare catastrophic failures dominate the arithmetic mean. Consequently, the observed spike does not correspond to a general degradation of the pose estimation pipeline, but rather to isolated cases in which the PnP reconstruction becomes unstable.

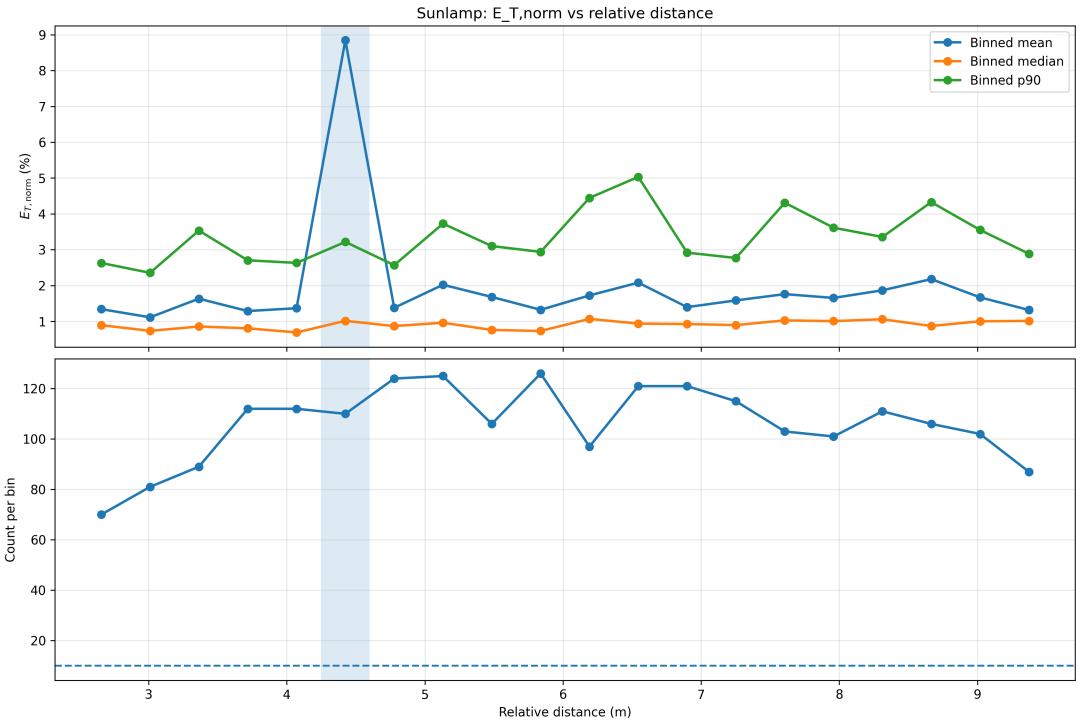


Figure 4.8: Diagnostic analysis of the peak bin in the `sunlamp` dataset, highlighting the presence of a small number of extreme outliers responsible for the spike in the mean error.

For this reason, introducing an upper bound on the norm of the predicted translation vector $||\mathbf{t}_{pred}||$ was necessary.

This bound does not aim to improve the accuracy of the pose estimation itself, but rather to discard physically implausible solutions produced by degenerate PnP configurations.

Empirical Distance Distribution

To define a meaningful threshold, an empirical analysis of the target distance distribution was performed across all available datasets. For each sample, the Euclidean norm of the ground–truth translation vector was computed as

$$d = \|\mathbf{t}_{GT}\| \tag{4.12}$$

The analysis was carried out for the `synthetic`, `lightbox`, and `sunlamp` domains. The distributions are very similar across the three domains, confirming that the experimental setups operate within a comparable range of distances. In particular, the maximum observed distances are approximately:

$$\text{validation} \approx 10.0\,\text{m} \quad \text{lightbox} \approx 9.55\,\text{m} \quad \text{sunlamp} \approx 9.55\,\text{m}$$

The corresponding histograms (Fig. 4.9) further show that the datasets cover the entire interval between roughly 2.3 m and 10 m with an approximately uniform distribution.

Threshold Definition

Based on this empirical analysis, the operational range of the dataset can be considered bounded by approximately 10 m. Consequently, an upper threshold was introduced on the predicted translation magnitude:

$$\|\mathbf{t}_{pred}\| \leq 10\,\text{m}$$

This value represents the maximum operational range observed in the datasets and therefore provides a conservative physical bound for plausible pose estimates. Pose estimates whose predicted distance exceeds this value are therefore considered physically implausible and are discarded as degenerate solutions.

Importantly, this filter acts only as a sanity check on the output of the PnP solver and does not influence the estimation process itself. Its role is solely to remove extreme outliers that would otherwise degrade the statistical evaluation of the pose estimation pipeline.

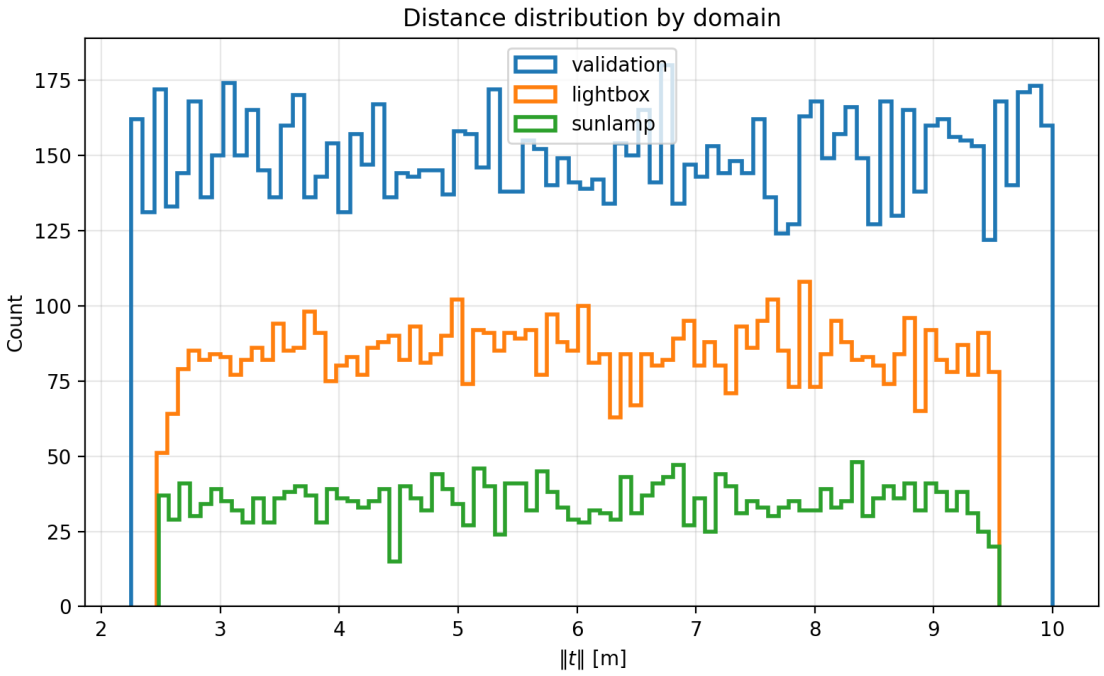


Figure 4.9: Histogram of the ground–truth distance distribution across the three datasets.

4.6.2 Dynamic Evaluation

The evaluation stage of the proposed pipeline was designed to remain consistent with the evaluation philosophy adopted in the SPEED+ benchmarks. In these benchmarks, performance on the HIL datasets (`lightbox` and `sunlamp`) is assessed over the entire dataset, without explicitly introducing the concept of *discarded samples*.

In practice, this requirement motivated the adoption of a dynamic acceptance strategy for the PnP stage. Instead of enforcing a fixed minimum number of inliers, the evaluation employs a progressively relaxed inlier schedule:

```
min_inliers_schedule = [11, 9, 8, 6, 4]
```

This mechanism allows pose solutions supported by a decreasing number of inliers to be considered, ensuring that a valid pose estimate can still be produced in challenging scenarios where the keypoint configuration becomes partially degraded. The objective of this strategy is to improve the robustness of the evaluation stage and to avoid rejecting samples that can still produce meaningful pose estimates.

4.6.3 LM refinement for the HRNet pipeline

Similarly to the YOLO-based pipeline described in Section 3.5.2, a geometric refinement stage based on the Levenberg–Marquardt algorithm was also integrated in the

HRNet-based pose estimation framework.

Starting from the initial pose estimate obtained through the PnP procedure, the pose parameters are locally refined by minimising the reprojection error of the inlier 2D–3D correspondences. The optimisation aims to improve the geometric consistency of the predicted keypoints with respect to the estimated pose.

While the conceptual objective remains the same with respect to the YOLO pipeline, the numerical implementation differs slightly. In the HRNet evaluation framework the refinement is performed using the `solvePnPRefineLM` function available in OpenCV, which directly applies a LM optimisation to the pose parameters by minimising the reprojection residuals of the selected correspondences.

In contrast to the weighted formulation used in the YOLO-based pipeline, the HRNet pipeline employs a standard unweighted LM refinement. This choice reflects the different nature of keypoint predictions: while YOLO outputs discrete keypoints with explicit visibility annotations, HRNet produces heatmap-based predictions whose uncertainty is already partially captured by the spatial distribution of the predicted heatmaps.

Specifically, robustness is improved through a two step refinement strategy: after an initial LM optimisation, the reprojection errors are evaluated and only the most consistent correspondences are retained. A second LM optimisation is then performed on this reduced subset, limiting the influence of residual outliers.

Finally, a consistency check is applied to ensure that the refined pose remains physically consistent. If the refined solution produces an invalid configuration, the algorithm falls back to the original PnP estimate.

4.7 Results

4.7.1 Comparison between Training Strategies

Tables 4.3, 4.4, and 4.5 report the pose estimation performance obtained on all three dataset splits considered. For clarity, the different training configurations evaluated throughout the experiments are identified using the following labels:

- **Training 1–3**: baseline strategies used to investigate different loss formulations. Training 1 uses heatmap supervision only, Training 2 introduces a coordinate loss from the beginning of training, and Training 3 applies the coordinate loss using a scheduling strategy.
- **Training 4**: augmented training pipeline aimed at reducing the synthetic-to-real domain gap.

- **Training 5**: configuration introducing the bounding-box crop preprocessing step. Additional pose refinement strategies are evaluated for this setup, including dynamic evaluation filtering and Levenberg–Marquardt (LM) optimisation.
- **Training 6**: final configuration using the larger HRNet-w64 model.

`Synthetic` validation

On the `synthetic` validation split, all models achieve very high success rates (> 98%), confirming that the network learns the spacecraft geometry reliably when training and evaluation occur within the same domain.

As discussed in Section 4.3.4, the coordinate-based loss formulations investigated in *Training 2* and *Training 3* do not provide a clear advantage over the heatmap-only supervision used in *Training 1*.

The augmented pipeline (*Training 4*) shows a slight degradation of the evaluated metrics, which is expected since stronger augmentations introduce variability not present in the validation distribution.

The bounding-box crop introduced in *Training 5* shows significantly improved metrics, further strengthened by the application of the LM refinement process.

Finally, the HRNet-w64 configuration achieves the best overall accuracy, confirming that a backbone with higher capacity improves keypoint localisation precision.

	Synthetic Validation						
Training	E_t [m]	e_t [-]	e_q [rad]	e_q [°]	SLAB	RMSE [px]	Success Rate [%]
Training 1	0.04371	0.00699	0.01990	1.14019	0.02658	**1.48658**	98.7160
Training 2	0.05377	0.00899	0.02349	1.34588	0.03237	2.98579	98.8244
Training 3	0.05212	0.00829	0.02598	1.48854	0.03402	1.92093	98.5243
Training 4	0.07315	0.01208	0.03588	2.05577	0.04787	3.39141	98.7160
Training 5	0.02136	0.00372	0.01113	0.63770	0.01424	6.18745	99.1746
+ LM refinement	0.02051	0.00348	0.01075	0.61593	0.01362	7.36878	99.14957
Training 6	**0.01949**	**0.00336**	**0.01022**	**0.58556**	**0.01295**	6.87964	**99.3664**

Table 4.3: Pose estimation performance on the `Synthetic` split for different training strategies.

`Lightbox` domain

The `lightbox` dataset, as expected, introduces a moderate domain shift due to differences in illumination and imaging conditions. Consequently, the baseline models (*Training 1–3*) show a drastic drop in success rate, falling between 17% and 42%.

The augmented pipeline (*Training 4*) substantially improves robustness, increasing the success rate to over 89%, while still achieving better results. This highlights

the importance of augmentation in reducing the synthetic-to-real gap.

The bounding-box crop in *Training 5* introduces significant improvements in pose accuracy and robustness, further enhanced by the LM refinement stage.

The HRNet-w64 model achieves the best overall trade-off between accuracy and robustness, reaching a success rate of 91.9%.

	Lightbox						
Training	E_t [m]	e_t [-]	e_q [rad]	e_q [°]	SLAB	RMSE [px]	Success Rate [%]
Training 1	0.32291	0.05231	0.47264	27.08028	0.52489	28.09397	19.59941
Training 2	0.56538	0.09970	0.53668	30.74950	0.63637	33.60168	42.75964
Training 3	0.37192	0.06547	0.45875	26.28444	0.52416	26.99673	17.71513
Training 4	0.29043	0.04845	0.25378	14.54052	0.30221	15.99727	89.09496
Training 5	0.07524	0.01253	0.06615	3.79012	0.07848	16.73063	90.86053
+ LM refinement	0.06676	0.01096	0.0627	3.59256	0.07344	17.66912	90.89021
Training 6	**0.06212**	**0.01011**	**0.05532**	**3.16960**	**0.06518**	**16.07657**	**91.95846**

Table 4.4: Pose estimation performance on the `Lightbox` dataset for the different training strategies.

`Sunlamp` domain

The `sunlamp` split, as previously discussed, represents the most challenging scenario due to strong illumination variations and specular reflections that significantly perturb keypoint localisation.

In this domain, the baseline models perform poorly, with particularly low success rates. The augmentation strategy improves robustness and accuracy, confirming its efficiency in addressing the domain gap problem.

Introducing the bounding-box crop significantly improves the geometric conditioning of the pose estimation problem, reducing both translational and rotational errors. Nevertheless, the success rate remains limited, probably due to occasional PnP failures caused by highly distorted keypoint configurations.

The HRNet-w64 configuration again provides the most stable performance, achieving the highest success rate ($\approx 80.4\%$).

	Sunlamp						
Training	E_t [m]	e_t [-]	e_q [rad]	e_q [°]	SLAB	RMSE [px]	Success Rate [%]
Training 1	0.94854	0.15673	1.24838	71.52691	1.40510	64.38698	7.88248
Training 2	2.01262	0.44709	2.07735	119.02339	2.52444	159.40292	1.79147
Training 3	0.68601	0.10414	1.12463	64.43655	1.22875	58.71589	18.0222
Training 4	0.32273	0.05152	0.40670	23.30219	0.45820	**23.55178**	80.4013
Training 5	0.10631	0.01713	0.09011	5.16292	0.10715	28.98408	75.9226
LM refinement	0.10223	0.01625	0.08364	4.79222	0.09978	30.06959	75.85095
Training 6	**0.09952**	**0.01569**	**0.06599**	**3.78095**	**0.08156**	26.68674	**80.43712**

Table 4.5: Pose estimation performance on the `Sunlamp` dataset for the different training strategies.

General observations

Two main trends emerge from the results. First, increasing the backbone capacity consistently improves both pose accuracy and robustness, highlighting the importance of high-resolution spatial representations for keypoint localisation. Second, the combination of the cropping pre-processing operation and the post-processing LM refinement process substantially increases the stability of the pose recovery stage, particularly under difficult illumination conditions.

Although *Training 3* on `lightbox` and `sunlamp` appears to achieve the best performance, the selection of the heatmap-only loss formulation as a baseline for the following trainings was based on the accuracy obtained on the subset of common samples evaluated on the `synthetic` split (see Section 4.3.4).

Overall, while the baseline pipeline performs well on synthetic data, robust performance in real scenarios requires both stronger architectures and geometric refinements in the pose estimation stage.

4.7.2 Error distribution vs relative distance

Figure 4.10 illustrates the behaviour of the main pose estimation errors as a function of the relative distance between the camera and the target spacecraft.

The translational error shows a clear increasing trend with distance for both domains. This behaviour is expected in monocular pose estimation problems, as the apparent size of the target decreases with range, reducing the effective spatial resolution of the predicted keypoints. Consequently, small localisation errors in image space translate into larger metric errors at larger distances.

The comparison between the two datasets highlights a consistent performance gap between the `lightbox` and `sunlamp` domains. While `lightbox` exhibits rela-

tively stable errors across the entire distance range, the `sunlamp` results show larger variability and higher overall errors. This behaviour can be attributed to the well known challenging illumination conditions of this setup.

Despite this domain gap, the normalised translational error remains approximately constant for the `lightbox` domain and increases only moderately for `sunlamp`. This indicates that the pose estimation pipeline maintains a stable scaling behaviour with distance and that the main performance degradation is primarily associated with the more realistic illumination conditions rather than with the geometric reconstruction process itself.

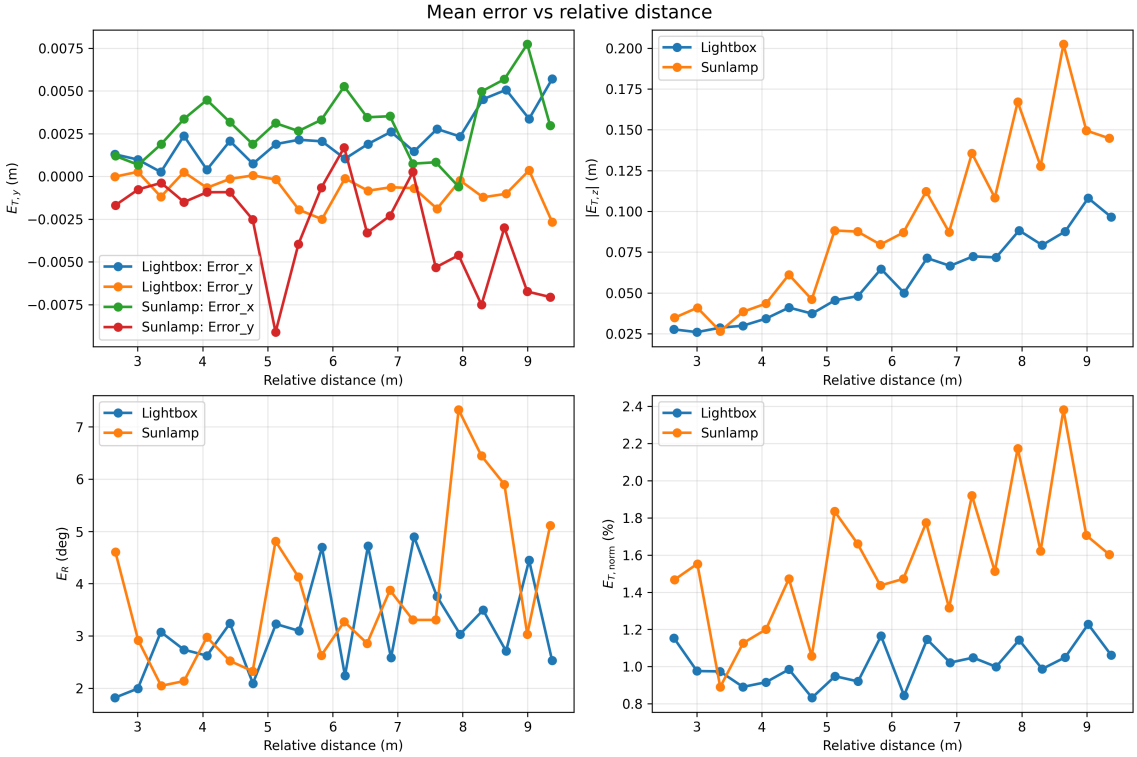


Figure 4.10: Mean and normalised translation errors and mean rotation error as a function of the relative target distance, evaluated on `lightbox` and `sunlamp`.

4.7.3 HRNet vs YOLO

A comparison between the YOLO-based and HRNet-based pipelines provides further insight into the behaviour of the pose estimation errors as a function of the relative distance. The corresponding mean statistics of the normalised translational and rotational error are shown in Figure 4.11. After removing degenerate PnP solutions using the filtering strategy described in Section 4.6.1, a clear difference in performance emerges between the two architectures.

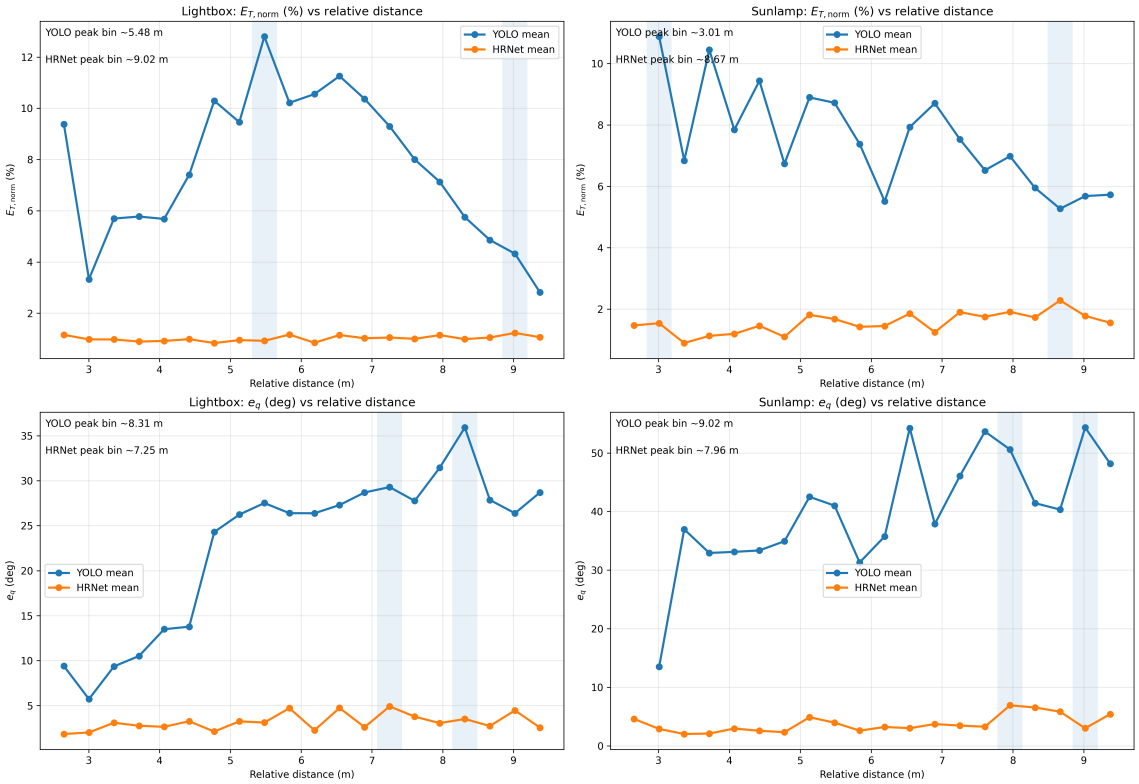


Figure 4.11: Distribution of the normalised translational error $E_{T,norm}$ as a function of the relative distance for the YOLO-based and HRNet-based pipelines.

In the `lightbox` domain, the HRNet-based pipeline consistently achieves significantly lower translational errors across the entire distance range. The mean normalised translation error remains mostly close to 1%. In contrast, the YOLO-based pipeline exhibits noticeably larger errors and a stronger dependence on the relative distance. In particular, the mean error increases towards the mid-range distances, where a peak is observed around 5.5 m, before gradually decreasing again for larger distances.

A similar behaviour is observed in the `sunlamp` domain. Although the more challenging illumination conditions introduce additional variability, HRNet maintains a relatively stable performance with mean translational errors generally below 2%. Conversely, the YOLO-based approach shows consistently larger errors across the entire distance range, typically between 6% and 10%, indicating a stronger sensitivity to image conditions and keypoint localisation noise.

The rotational error statistics further confirm this trend. In both domains, HRNet achieves mean angular errors of only a few degrees, remaining below approximately 5° across the considered distances. In contrast, the YOLO-based pipeline exhibits substantially larger rotational errors, which increase with the target distance and reach values above 50° in the most challenging cases on the `sunlamp` domain.

These differences can be attributed to the distinct keypoint localisation strategies adopted by the two architectures. Overall, these results indicate that the HRNet-based pipeline provides more accurate and stable keypoint estimates, resulting in a more reliable input for the subsequent PnP reconstruction, particularly when the apparent size of the spacecraft decreases and the geometric conditioning of the problem becomes less favourable.

4.8 On-board Deployment Assessment on NVIDIA Jetson Orin

To assess the feasibility of performing target pose estimation in an operational on-board scenario, the proposed pipeline was evaluated on an NVIDIA Jetson Orin platform. The objective of this test was to verify whether the complete pose estimation chain could be executed on hardware representative of the computational resources available on the chaser spacecraft.

In the considered mission scenario, images acquired by the navigation camera must be processed locally in order to estimate the relative pose of the target spacecraft. Evaluating the pipeline on an embedded GPU platform therefore provides an indication of its compatibility with the computational and memory constraints of an on-board vision-based navigation system.

4.8.1 Model Export and Precision Reduction

The HRNet-based network was originally trained in PyTorch using single-precision floating-point parameters (`FP32`). For deployment on the embedded platform, the trained model was exported to the NVIDIA TensorRT framework, which generates an optimised inference engine by applying several optimisations, such as layer fusion, kernel selection, memory reuse, and scheduling improvements.

In addition, the inference engine was configured to operate using half-precision floating-point arithmetic (`FP16`). This conversion reduces the numerical representation of weights and activations from 32 bits to 16 bits, decreasing memory footprint, and improving computational efficiency on hardware architectures supporting `FP16` acceleration, such as the Jetson Orin GPU.

While reduced-precision inference introduces a minor loss of numerical fidelity, it significantly improves execution efficiency and is widely adopted in embedded AI deployment.

Platform	Dataset split	E_t [m]	e_t [-]	e_q [rad]	e_q [°]	SLAB	Success Rate [%]
NVIDIA GeForce RTX 2080 Ti GPU	validation	0.01949	0.00336	0.01022	0.58556	0.01295	99.3664
	lightbox	0.06212	0.01011	0.05532	3.16960	0.06518	91.9585
	sunlamp	0.09952	0.01569	0.06599	3.78095	0.08156	80.43712
NVIDIA Jetson Orin Nano, 8 GB VRAM	validation	0.0188	0.0028	0.0104	0.6029	0.0132	99.91
	lightbox	0.0545	0.0089	0.0438	2.5113	0.0527	95.19
	sunlamp	0.1475	0.0252	0.0703	4.281	0.0955	81.51

Table 4.6: Performance assessment based on the metrics comparison between different platforms.

The results reported in Table 4.6 show that the performance obtained on the embedded Jetson platform remains comparable to that achieved on the desktop GPU. The largest discrepancy appears in the `sunlamp` domain, where the translational error increases moderately on the embedded platform. This effect is expected, as the most challenging illumination conditions amplify the sensitivity of the pose estimation pipeline to small perturbations introduced by reduced-precision inference.

Overall, the results confirm that the proposed pipeline maintains a comparable level of accuracy across platforms, supporting the feasibility of deploying the method on embedded hardware.

4.8.2 Runtime Evaluation on Embedded Hardware

The TensorRT inference engine was executed directly on the Jetson Orin Nano platform to evaluate the runtime performance of the complete pose estimation pipeline. The measured execution time includes both the neural network inference and the subsequent post-processing stages, namely the PnP pose recovery and the computation of the SPEED+ evaluation metrics.

The results reported in Table 4.7 show that real-time inference is achievable on the Jetson Orin platform. In particular, the fastest configuration reaches approximately 36.5 FPS with an average latency of 27.4 ms, while the most computationally demanding configuration still maintains a throughput above 18 FPS.

These results demonstrate that the proposed vision-based pose estimation pipeline can be executed efficiently on embedded GPU hardware representative of the computational resources available on-board spacecraft. Such performance levels are compatible with the update rates typically required by vision-based relative navigation systems during proximity operations.

	Runtime (FPS)	**Latency** (ms)
Training 5	36.5	27.38 ± 0.09
Training 6	18.5	54.09 ± 0.09

Table 4.7: Runtime and latency evaluation on the NVIDIA Jetson Orin Nano platform, for both *Training 5* and *Training 6*.

4.9 Final Considerations

The analyses presented in this chapter show that the proposed HRNet-based pipeline provides a significant improvement over the initial YOLO-based formulation in terms of both pose accuracy and robustness under domain shift. This result confirms the importance of preserving high-resolution spatial information for keypoint localisation.

Among the investigated design choices, the heatmap-only supervision strategy proved to be the most reliable baseline, while the introduction of domain-shift augmentation, bounding-box cropping, and geometric post-processing refinements substantially improved the overall behaviour of the pipeline on the `lightbox` and `sunlamp` domains. In particular, the crop-based preprocessing and the subsequent refinement stages contributed to improving the conditioning of the pose recovery step, reducing the impact of localisation noise and degenerate PnP configurations.

The final HRNet-w64 configuration achieved the best overall trade-off between accuracy, robustness, and success rate across all evaluated domains. In addition, the deployment assessment on the NVIDIA Jetson Orin platform demonstrated that the complete pipeline can be executed on embedded GPU hardware with real-time performance, supporting the practical feasibility of the proposed approach for on-board vision-based navigation during spacecraft proximity operations.

Chapter 5

Conclusions and Future Work

5.1 Summary of the Work

This thesis investigated the problem of monocular spacecraft pose estimation for vision-based navigation during proximity operations. Accurate and robust estimation of the relative pose between a chaser and a target satellite is a fundamental capability for several space missions, including rendezvous, inspection, servicing, and debris removal. In such scenarios, vision-based approaches are particularly attractive due to their passive nature, low mass, and compatibility with small spacecraft platforms.

The work focused on the development and evaluation of a learning-based pose estimation pipelines using the SPEED+ dataset, which provides both `synthetic` and HIL imagery of the Tango spacecraft. Two different keypoint-detection pipelines were investigated. The first is based on a YOLO pose estimation architecture, while the second relies on an HRNet-based keypoint detector. In both cases, the predicted 2D keypoints are combined with a geometry-based PnP solver in order to recover the full 6-DOF pose of the target spacecraft.

The main objective of the two evaluated pipelines consists in reaching robust and reliable pose accuracy in HIL domains. These are, in fact, characterised by significantly challenging illumination conditions and, more importantly, are excluded from the training process. For these reasons, the `lightbox` and `sunlamp` dataset splits are specifically designed to evaluate the robustness of the pose estimation pipeline, and therefore represent a critical benchmark for assessing the reliability of the approach in realistic space scenarios.

The core component of this thesis is the network based on the HRNet architecture, which maintains high-resolution representations throughout the network while combining information from multiple resolutions. This characteristic makes the ar-

chitecture particularly suitable for tasks requiring accurate spatial localisation, such as keypoint detection.

In the implemented framework, input images from the SPEED+ dataset are first resized to a fixed resolution and processed by the HRNet backbone to produce a set of keypoint heatmaps. Each heatmap represents the spatial likelihood distribution of a specific keypoint on the image plane. The final 2D keypoint locations are extracted from the predicted heatmaps using a deterministic arg max operation and subsequently mapped back to the image coordinate system. These 2D detections are then associated with the corresponding 3D keypoints of the spacecraft model to build the 2D–3D correspondences required for pose estimation.

The 6-DOF pose of the target spacecraft is recovered through a PnP-based estimation stage. To improve robustness, several evaluation strategies and acceptance criteria were investigated, including a translation-consistency gate designed to mitigate degenerate solutions. The resulting pipeline was evaluated following the evaluation protocol adopted in the SPEED+ based SPEC challenge.

Finally, the feasibility of deploying the proposed pipeline on embedded hardware was investigated through inference experiments on the NVIDIA Jetson Orin platform. The trained PyTorch model was converted to a TensorRT-optimised format to enable efficient execution in an onboard environment, where resources are limited, providing insights into the practical applicability of the approach for real space missions.

5.2 Main Findings

The experimental analysis carried out in this work provides several insights into the behaviour and robustness of learning-based spacecraft pose estimation pipelines.

First, the results confirm the effectiveness of combining a learning-based keypoint detector with a geometric pose estimation stage. In the proposed pipeline, the neural network is responsible for predicting the 2D image locations of the spacecraft keypoints, while the final 6-DOF pose is recovered through a PnP solver using the known 3D model of the target. This separation between visual detection and geometric reconstruction allows the pose estimation stage to remain physically consistent and interpretable, while leveraging the robustness of the CNN for feature localisation.

Second, the comparative analysis between the YOLO-based and HRNet-based pipelines highlights the impact of the keypoint detection strategy on the overall pose estimation performance. The YOLO-based approach provides a lightweight and fast inference pipeline, making it attractive for real-time applications. However, the

HRNet architecture generally produces improved spatial consistency in the predicted heatmaps, which leads to more stable 2D–3D correspondences.

Another important outcome of the study concerns the strategies adopted to mitigate the domain gap between `synthetic` training data and the `lightbox` and `sunlamp` datasets. Several augmentation techniques were investigated to improve the robustness of the keypoint detector under different illumination conditions. In particular, a significant improvement in performance was observed for the HRNet-based pipeline when the training process included an additional cropping operation performed on the bounding boxes. By increasing the effective spatial resolution of the spacecraft within the image, this preprocessing step allowed the network to focus more effectively on the relevant visual features, resulting in more reliable keypoint predictions in the HIL domains.

In addition to the previous findings, from a distance-constrain point of view, the analysis of the error distribution as a function of the relative distance between the chaser and the target satellite reveals that for close-range missions the proposed pipelines ensure very high fidelity and precision.

Finally, the deployment experiments performed on the NVIDIA Jetson Orin platform demonstrate that the proposed pipeline can be executed on embedded hardware after appropriate model optimisation. This result provides preliminary evidence of the feasibility of performing vision-based pose estimation directly onboard a spacecraft platform.

5.3 Comparison with SOTA

Table 5.1 summarises a selection of the best results reported in the literature for spacecraft pose estimation on the SPEED+ dataset. These methods adopt different architectures and training strategies, ranging from HRNet-based keypoint detectors to dedicated pose regression networks.

The results obtained in this work show that the proposed pipeline achieves competitive performance with respect to several existing approaches, particularly considering the simplicity of the overall framework and the absence of explicit domain adaptation techniques.

In contrast to many recent works, the models developed in this thesis were trained exclusively on synthetic data and evaluated directly on the `lightbox` and `sunlamp` domains without additional fine-tuning. This setting represents a more challenging scenario, as the models must generalise across significantly different illumination conditions and visual characteristics.

Despite this limitation, the obtained results demonstrate that robust keypoint

detection combined with a carefully designed geometric pose estimation stage can provide reliable pose estimates even in challenging conditions.

Moreover, contrary to other works that show high accuracy only on one of the two HIL domains, the proposed work proved reliable in both dataset splits. In particular, on the `lightbox` split the best metrics are achieved compared to SOTA results. Although the same performance was not achieved on the `sunlamp` split, it still shows competitive orientation error and high translational accuracy.

Furthermore, in the proposed work a robustness assessment in terms of success rate was also addressed, differently from most of the SOTA works.

Method	**Backbone**	E_q **[°]**	E_t **[m]**	$E_{\mathbf{SLAB}}$
		Lightbox		
[39]	HRNet	3.187	0.105	0.073
[44]	HRNet-w48	6.665	0.302	0.165
[43]	HRNet + BiFPN	4.32	0.09	0.09
[47]	SPNv2	5.577	0.15	0.122
proposed	HRNet-w64	**3.170**	**0.062**	**0.065**
		Sunlamp		
[39]	HRNet	4.299	0.086	0.090
[44]	HRNet-w48	**2.728**	**0.065**	**0.059**
[43]	HRNet + BiFPN	6.94	0.14	0.14
[47]	SPNv2	9.788	0.161	0.197
proposed	HRNet-w64	3.781	0.100	0.082

Table 5.1: Selected SOTA results for comparison with the proposed pipelines.

5.4 Limitations of the Proposed Approach

The main limitation of the proposed approach emerges from the evaluation results obtained on the HIL domains. In several cases, the pose estimation process fails to produce a valid solution and the corresponding samples are discarded during the evaluation stage.

This behaviour is largely related to the fact that no explicit domain adaptation or fine-tuning on real imagery was performed in this work. While this choice was intentional in order to assess the true generalisation capability of the networks under previously unseen conditions, it leads to a higher number of rejected samples compared to approaches that explicitly adapt the model to the domain shift problem.

From an experimental perspective, this setting provides a more transparent assessment of the robustness of the pose estimation pipeline when confronted with challenging illumination conditions and visual discrepancies between synthetic and HIL imagery. However, from an operational standpoint, this behaviour represents a clear limitation. In realistic onboard applications for spacecraft navigation, pose estimates must be produced continuously and reliably, and discarding a large fraction of frames would not be acceptable during real-time operations.

These considerations naturally motivate the exploration of strategies aimed at improving robustness across domains, which are discussed in the following section.

5.5 Future Work

Several research directions can be explored to further improve the robustness and operational applicability of the proposed pose estimation pipeline.

A first natural extension of this work concerns the introduction of domain adaptation strategies aimed at reducing the performance gap between synthetic training data and HIL imagery. Techniques such as fine-tuning on a limited set of real images, domain randomisation, or image-to-image translation methods could help the network better handle the illumination variations and visual discrepancies observed in the `lightbox` and `sunlamp` domains. Reducing the number of rejected samples during the evaluation stage would be fundamental for practical applications.

Another promising direction concerns improvements in the keypoint detection stage. While the HRNet-based architecture demonstrated good spatial consistency in the predicted heatmaps, further refinements could be investigated to increase robustness under challenging viewing conditions. For instance, different heatmap supervision strategies or alternative loss function formulations could potentially provide more reliable 2D–3D correspondences for the subsequent pose estimation stage.

Future work could also explore the integration of temporal information across image sequences. In realistic proximity operations, the pose estimation process is typically embedded within a navigation framework where consecutive frames are available. Filtering techniques (e.g.EKFs), could be employed to improve the stability of the estimated trajectory and mitigate occasional pose estimation failures.

Finally, further investigations could be conducted on the deployment of the proposed approach on embedded onboard hardware. Although the preliminary experiments on the NVIDIA Jetson Orin platform demonstrated the feasibility of running the inference pipeline in an environment where resources are limited, additional optimisation and robustness analysis would be required to meet the reliability standards of autonomous spacecraft navigation systems.

A further relevant direction concerns the in-orbit demonstration and validation of the proposed pose estimation pipeline. While the current work evaluates performance using synthetic and HIL datasets, a definitive assessment of the approach would require its integration within an operational space mission. Future technology demonstration missions could provide the opportunity to validate the robustness and reliability of the pipeline under real orbital conditions, including realistic illumination, background clutter, and sensor noise.

In this context, the proposed method could be embedded within a vision-based navigation system for proximity operations, enabling on-board pose estimation of a target satellite during rendezvous or inspection scenarios. Such in-orbit validation would not only confirm the practical applicability of the approach, but also provide valuable insights into failure modes, real-time constraints, and system-level integration aspects that cannot be fully captured through algorithmic validation.

Bibliography

[1] ESA Space Debris Office, *Space debris by the numbers*, https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers, 2023.

[2] D. J. Kessler and B. G. Cour-Palais, "Collision frequency of artificial satellites: The creation of a debris belt," *Journal of Geophysical Research*, vol. 83, pp. 2637–2646, Jun. 1978. DOI: 10.1029/ja083ia06p02637.

[3] N. L. Johnson, E. Stansbery, J. .-. Liou, M. Horstman, C. Stokely, and D. Whitlock, "The characteristics and consequences of the break-up of the fengyun-1c spacecraft," *Acta Astronautica*, vol. 63, pp. 128–135, Jul. 2008. DOI: 10.1016/j.actaastro.2007.12.044. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0094576507003281.

[4] J. Nicholas, "The collision of iridium 33 and cosmos 2251: The shape of things to come," in *Proceedings of the 60th International Astronautical Congress*, NASA Technical Report JSC-CN-18971, Daejeon, Republic of Korea, 2009. [Online]. Available: https://ntrs.nasa.gov/citations/20100002023.

[5] U. N. O. for Outer Space Affairs (UNOOSA), *Treaty on principles governing the activities of states in the exploration and use of outer space, including the moon and other celestial bodies*, 1967. [Online]. Available: https://www.unoosa.org/oosa/en/ourwork/spacelaw/treaties/outerspacetreaty.html.

[6] United Nations Office for Outer Space Affairs (UNOOSA), *Convention on international liability for damage caused by space objects*, 1972. [Online]. Available: https://www.unoosa.org/oosa/en/ourwork/spacelaw/treaties/introliability-convention.html.

[7] European Space Agency, *Esa zero debris approach*, https://www.esa.int/Zero_Debris, 2023.

[8] European Space Agency, *Esa purchases world-first debris removal mission from start-up*, https://www.esa.int/Space_Safety/ESA_purchases_world-

first_debris_removal_mission_from_start-up, Accessed: 2024-XX-XX, 2021.

[9] H. D. Curtis, *Orbital mechanics for engineering students*. Elsevier, 2014. [Online]. Available: https://www.elsevier.com/books/orbital-mechanics-for-engineering-students/curtis/978-0-08-097747-8.

[10] F. Ankersen, *Guidance, Navigation, Control and Relative Dynamics for Spacecraft Proximity Maneuvers: Ph. D. Thesis*. Institut for Elektroniske Systemer, ISBN: 9788792328724. [Online]. Available: https://books.google.it/books?id=aKsxMwEACAAJ.

[11] M. Zheng, Y. Zhang, X. Wang, and L. Fan, "Pose-constrained control of proximity maneuvering for tracking and observing noncooperative targets with unknown acceleration," *Aerospace*, vol. 11, p. 828, Oct. 2024. DOI: 10.3390/aerospace11100828. [Online]. Available: https://www.mdpi.com/2226-4310/11/10/828.

[12] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations," *Progress in Aerospace Sciences*, vol. 93, pp. 53–72, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:116231588.

[13] S. D'Amico, J.-S. Ardaens, and R. Larsson, "Spaceborne autonomous formation flying experiment on the PRISMA mission," *Journal of Guidance Control and Dynamics*, vol. 35, pp. 834–850, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:110297982.

[14] J. E. LeCroy, R. T. Howard, and D. S. Hallmark, "Effects of optical artifacts in a laser-based spacecraft navigation sensor," in *AIAA Aerospace Sciences Meeting*, NASA Tech. Rep. no. 20070013727, Reno, NV, USA: American Institute of Aeronautics and Astronautics, Jan. 2007. [Online]. Available: https://ntrs.nasa.gov/api/citations/20070013727/downloads/20070013727.pdf.

[15] R. Howard, A. F. Heaton, R. M. Pinson, C. L. Carrington, J. E. Lee, T. C. Bryan, B. A. Robertson, S. H. Spencer, and J. E. Johnson, "The advanced video guidance sensor: Orbital express and the next generation," 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:108801615.

[16] I. Kawano, T. Kasai, M. Mokuno, and T. Suzuki, "Result of autonomous rendezvous docking experiment of engineering test satellite-vii," *Journal of Spacecraft and Rockets - J SPACECRAFT ROCKET*, vol. 38, pp. 105–111, Jan. 2001. DOI: 10.2514/2.3661.

[17] L. P. Cassinis, T. H. Park, N. Stacey, S. D'Amico, A. Menicucci, E. Gill, I. Ahrns, and M. Sanchez-Gestido, "Leveraging neural network uncertainty in adaptive unscented kalman filter for spacecraft pose estimation," *Advances in Space Research*, vol. 71, no. 12, pp. 5061–5082, 2023, ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2023.02.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117723001400.

[18] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Pose estimation for spacecraft relative navigation using model-based algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 431–447, 2017. DOI: 10.1109/TAES.2017.2650785.

[19] T. Lei, X.-F. Liu, G.-P. Cai, Y.-M. Liu, and P. Liu, "Pose estimation of a non-cooperative target based on monocular visual SLAM," *International Journal of Aerospace Engineering*, vol. 2019, no. 1, p. 9 086 891, 2019. DOI: https://doi.org/10.1155/2019/9086891. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2019/9086891. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2019/9086891.

[20] D. S. Lauretta, S. S. Balram-Knutson, E. Beshore, W. V. Boynton, C. Drouet d'Aubigny, D. N. DellaGiustina, H. L. Enos, D. R. Golish, C. W. Hergenrother, E. S. Howell, C. A. Bennett, E. T. Morton, M. C. Nolan, B. Rizk, H. L. Roper, A. E. Bartels, B. J. Bos, J. P. Dworkin, D. E. Highsmith, D. A. Lorenz, L. F. Lim, R. Mink, M. C. Moreau, J. A. Nuth, D. C. Reuter, A. A. Simon, E. B. Bierhaus, B. H. Bryan, R. Ballouz, O. S. Barnouin, R. P. Binzel, W. F. Bottke, V. E. Hamilton, K. J. Walsh, S. R. Chesley, P. R. Christensen, B. E. Clark, H. C. Connolly, M. K. Crombie, M. G. Daly, J. P. Emery, T. J. McCoy, J. W. McMahon, D. J. Scheeres, S. Messenger, K. Nakamura-Messenger, K. Righter, and S. A. Sandford, "OSIRIS-REx: Sample return from asteroid (101955) bennu," *Space Science Reviews*, vol. 212, no. 1–2, pp. 925–984, Aug. 2017, ISSN: 1572-9672. DOI: 10.1007/s11214-017-0405-1. [Online]. Available: http://dx.doi.org/10.1007/s11214-017-0405-1.

[21] A. F. Cheng, A. S. Rivkin, P. Michel, J. Atchison, O. S. Barnouin, L. A. M. Benner, N. L. Chabot, C. M. Ernst, E. G. Fahnestock, M. Kueppers, P. Pravec, E. S. G. Rainey, D. C. Richardson, A. M. Stickle, and C. A. Thomas, "AIDA DART asteroid deflection test: Planetary defense and science objectives," *Planetary and Space Science*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:52034619.

[22] V. Pesce, M. Lavagna, and R. Bevilacqua, "Stereovision-based pose and inertia estimation of unknown and uncooperative space objects," *Advances in Space Research*, vol. 59, no. 1, pp. 236–251, 2017. DOI: 10.1016/j.asr.2016.10.002.

[23] S. Sharma and S. D'Amico, *Pose estimation for non-cooperative rendezvous using neural networks*, 2019. arXiv: 1906.09868 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1906.09868.

[24] T. H. Park, M. Märtens, G. Lecuyer, D. Izzo, and S. D'Amico, "Speed+: Next-generation dataset for spacecraft pose estimation across domain gap," in *2022 IEEE Aerospace Conference (AERO)*, 2022, pp. 1–15. DOI: 10.1109/AERO53065.2022.9843439.

[25] P. F. Proenca and Y. Gao, *Deep learning for spacecraft pose estimation from photorealistic rendering*, 2019. arXiv: 1907.04298 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1907.04298.

[26] K. Black, S. Shankar, D. Fonseka, J. Deutsch, A. Dhir, and M. R. Akella, *Real-time, flight-ready, non-cooperative spacecraft pose estimation using monocular imagery*, 2021. arXiv: 2101.09553 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2101.09553.

[27] Y. Hu, S. Speierer, W. Jakob, P. Fua, and M. Salzmann, *Wide-depth-range 6d object pose estimation in space*, 2021. arXiv: 2104.00337 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2104.00337.

[28] M. A. Musallam, K. A. Ismaeil, O. Oyedotun, M. D. Perez, M. Poucet, and D. Aouada, *Spark: Spacecraft recognition leveraging knowledge of space environment*, 2021. arXiv: 2104.05978 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2104.05978.

[29] A. Rathinam, V. Gaudilliere, M. A. Mohamed Ali, M. Ortiz Del Castillo, L. Pauly, and D. Aouada, *Spark 2022 dataset : Spacecraft detection and trajectory estimation*, Zenodo, Jun. 2022. DOI: 10.5281/zenodo.6599762. [Online]. Available: https://doi.org/10.5281/zenodo.6599762.

[30] T. H. Park and S. D'Amico, "Adaptive neural-network-based unscented kalman filter for robust pose tracking of noncooperative spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 9, pp. 1671–1688, Sep. 2023, ISSN: 1533-3884. DOI: 10.2514/1.g007387. [Online]. Available: http://dx.doi.org/10.2514/1.G007387.

[31] T. Park and S. D'Amico, "Rapid abstraction of spacecraft 3d structure from single 2d image," Jan. 2024. DOI: 10.2514/6.2024-2768.

[32] Y. Liu, Z. Qi, P. Chen, H. Fan, Y. Zhu, J. Yang, and Y. Zhang, "Tap-track: Generalizable spacecraft pose tracking by tracking any points," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–13, 2025. DOI: 10.1109/TGRS.2025.3584925.

[33] G. Napolano, A. Nocerino, G. Fasano, M. Grassi, and R. Opromolla, "A cnn-based architecture for relative state and target motion parameters estimation in active debris removal missions," *Acta Astronautica*, vol. 235, pp. 485–511, 2025, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2025.06.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576525003649.

[34] L. Pasqualetto Cassinis, A. Menicucci, E. Gill, I. Ahrns, and M. Sanchez-Gestido, "On-ground validation of a cnn-based monocular pose estimation system for uncooperative spacecraft: Bridging domain shift in rendezvous scenarios," *Acta Astronautica*, vol. 196, pp. 123–138, 2022, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2022.04.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576522001515.

[35] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Martens, and S. D'Amico, "Satellite pose estimation challenge: Dataset, competition design, and results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 4083–4098, Oct. 2020, ISSN: 2371-9877. DOI: 10.1109/taes.2020.2989063. [Online]. Available: http://dx.doi.org/10.1109/TAES.2020.2989063.

[36] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, pp. 151–175, May 2010. DOI: 10.1007/s10994-009-5152-4.

[37] H. Huang, B. Song, G. Zhao, and Y. Bo, "End-to-end monocular pose estimation for uncooperative spacecraft based on direct regression network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 5, pp. 5378–5389, 2023. DOI: 10.1109/TAES.2023.3256971.

[38] M. Bechini, G. Gu, P. Lunghi, and M. Lavagna, "Robust spacecraft relative pose estimation via cnn-aided line segments detection in monocular images," *Acta Astronautica*, vol. 215, pp. 20–43, 2024, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2023.11.049. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576523006185.

[39] B. Chen, J. Cao, A. Parra, and T.-J. Chin, "Satellite pose estimation with deep landmark regression and nonlinear pose refinement," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2816–2824. DOI: 10.1109/ICCVW.2019.00343.

[40] M. Bechini and M. Lavagna, "Robust and efficient single-cnn-based spacecraft relative pose estimation from monocular images," *Acta Astronautica*, vol. 233, pp. 198–217, 2025, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2025.04.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576525002176.

[41] T. H. Park and S. D'Amico, "Bridging the domain gap for flight-ready spaceborne vision," *Journal of Spacecraft and Rockets*, pp. 1–19, Dec. 2025, ISSN: 1533-6794. DOI: 10.2514/1.a36130. [Online]. Available: http://dx.doi.org/10.2514/1.A36130.

[42] L. P. Cassinis, T. H. Park, N. Stacey, S. D'Amico, A. Menicucci, E. Gill, I. Ahrns, and M. Sanchez-Gestido, "Leveraging neural network uncertainty in adaptive unscented kalman filter for spacecraft pose estimation," *Advances in Space Research*, vol. 71, no. 12, pp. 5061–5082, 2023, ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2023.02.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117723001400.

[43] A. Legrand, R. Detry, and C. De Vleeschouwer, "Domain generalization for in-orbit 6d pose estimation," *Journal of Aerospace Information Systems*, vol. 22, no. 11, pp. 938–947, Nov. 2025, ISSN: 2327-3097. DOI: 10.2514/1.i011498. [Online]. Available: http://dx.doi.org/10.2514/1.I011498.

[44] Z. Wang, M. Chen, Y. Guo, Z. Li, and Q. Yu, *Bridging the domain gap in satellite pose estimation: A self-training approach based on geometrical constraints*, 2022. arXiv: 2212.12103 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2212.12103.

[45] S. Shukla, R. Srivastava, R. Lima, and T. Bera, "Satellite-model-free deep learning based pose estimation of non-cooperative satellite and tracking using navigation filter," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 4548–4555. DOI: 10.1109/IROS58592.2024.10801805.

[46] A. Legrand, R. Detry, and C. D. Vleeschouwer, *Leveraging neural radiance fields for pose estimation of an unknown space object during proximity operations*, 2024. arXiv: 2405.12728 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2405.12728.

[47] T. H. Park and S. D'Amico, "Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap," *Advances in Space Research*, vol. 73, no. 11, pp. 5726–5740, Jun. 2024, ISSN: 0273-1177. DOI: 10.1016/j.asr.2023.03.036. [Online]. Available: http://dx.doi.org/10.1016/j.asr.2023.03.036.

[48] G. Jocher and J. Qiu, *Ultralytics yolo11*, version 11.0.0, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics.

[49] Z. He, K. Wang, T. Fang, L. Su, R. Chen, and X. Fei, "Comprehensive performance evaluation of yolov11, yolov10, yolov9, yolov8 and yolov5 on object detection of power equipment," in *2025 37th Chinese Control and Decision Conference (CCDC)*, 2025, pp. 1281–1286. DOI: 10.1109/CCDC65474.2025.11090973.

[50] *Speed+: Next-generation spacecraft pose estimation dataset*, Stanford Digital Repository, Accessed: 2025-12-16. [Online]. Available: https://purl.stanford.edu/wv398fc4383.

[51] T. Möller and B. Trumbore, "Fast, minimum storage ray/triangle intersection," 1997.

[52] Ultralytics, *Model training with ultralytics yolo*, Ultralytics.com, 2023. Accessed: Feb. 6, 2026. [Online]. Available: https://docs.ultralytics.com/modes/train/#loss-functions.

[53] G. Jocher, J. Qiu, and A. Chaurasia, *Ultralytics yolo*, version 8.0.0, Available at https://ultralytics.com, Jan. 10, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[54] D. Shah, *Mean average precision (map) explained: Everything you need to know*, www.v7labs.com, Mar. 2022. [Online]. Available: https://www.v7labs.com/blog/mean-average-precision.

[55] OpenCV, *Opencv: Perspective-n-point pose computation*, https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html, Accessed: 2025-12-22.

[56] J. F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, 1986. [Online]. Available: https://api.semanticscholar.org/CorpusID:13284142.

[57] S. D'Amico, M. Benn, and J. L. Jørgensen, "Pose estimation of an uncooperative spacecraft from actual space imagery," 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:1537971.

[58] Y. Huo, Z. Li, and F. Zhang, "Fast and accurate spacecraft pose estimation from single shot space imagery using box reliability and keypoints existence judgments," *IEEE Access*, vol. 8, pp. 216 283–216 297, 2020. DOI: 10.1109/ACCESS.2020.3041415.

[59] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963, ISSN: 03684245. Accessed: Feb. 16, 2026. [Online]. Available: http://www.jstor.org/stable/2098941.