Qassim University جامعة القصيم

JEIT

Journal of Emerging and Innovative Technologies

Volume 1, Issue 1
July–December 2026

Received
08 September 2026

Accepted
31 October 2026

Published
05 November 2026

DOI
https://doi.org/10.71422/jeit.0004

License
CC BY-NC-SA 4.0

Cite this article

Alrashidi W, Alotibi J. Vision-Based Self-Mapping for Autonomous Indoor Drones. J Emerg Innov Technol. 2026;1(1):35–46.

Journal website

Research article

Vision-Based Self-Mapping for Autonomous Indoor Drones

Wurud Alrashidi[1,*], Jamaal Alotibi[1]

[1] Department of Computer Engineering, College of Computer, Qassim University, Buraydah, Saudi Arabia

*Corresponding author: 461215971@qu.edu.sa

Abstract

Without Global Positioning System (GPS) signals the problem of self-autonomous navigation of indoor drones is greatly complicated and consequently there has been a recent surge of interest in Visual Simultaneous Localization and Mapping (Visual SLAM). Such systems enable the localization of a drone in an environment and the building of a map of it using only camera data. However indoor environments generally pose severe problems for these systems such as low illumination, motion blur due to rapid motion and featureless surfaces that do not offer the reliable features required for tracking. Severity of such problems impairs, in extreme cases, the spatial orientation and the information of the place, which decreases localization quality and the robustness. The goal of this research is to develop an adaptive hybrid image enhancement technique to satisfy the real-time and localization accuracy criteria of visual SLAM based indoor drone navigation. The system combines the simple preprocessing methods, Gamma Correction and Contrast Limited Adaptive Histogram Equalization (CLAHE), with a particular enhancement model based on Convolutional Neural Networks (CNNs). When the Decision Tree classifier determines that the coverage of brightness, sharpness and feature density is such that advanced enhancement is required, advanced enhancement procedures are activated. This system is implemented as an external preprocessing component for ORB-SLAM3 and does not require modifications to the preprocessing pipeline of ORB-SLAM3. Single-channel intensity metrics are evaluated directly for single-channel grayscale input images (as standard in datasets like EuRoC) to preserve processing speed. Evaluation was performed on two orthogonal datasets: EuRoC MH01 and TartanAir P001. To ensure cross-dataset evaluation validity and strictly prevent data leakage, the Decision Tree and CNN models were trained using independent sequence partitioning and validated across stratified k-fold cross-validation subsets. The evaluation of localization was performed using the two standard metrics, Absolute Trajectory Error (ATE) and Relative Pose Error (RPE), obtained with help of the evo toolkit, reporting means and standard deviations across five experimental runs against ground truth trajectories. The strategy shown had a drastic reduction in the Absolute Trajectory Error (ATE) with EuRoC MH01, with the ATE decreasing from 1.431 m down to 0.978 m (representing a 31.7% reduction). Note that while monocular ORB-SLAM3 baseline errors under optimal synthetic or tuned settings can be lower, the 1.431 m baseline reflects realistic monocular scale drift under raw challenging unenhanced conditions without tuning. Moreover, the difficult TartanAir P001 test resulted in a great reduction in the mean error, from 6.667 m to 0.510 m, which is a 92.3% reduction. The Decision Tree classifier had a very high accuracy of 99.97% and a robust recall of 100%, and only 0.41% of the images were passed through the CNN enhancement module. Within the adaptive framework, enhancement preprocessing was processed at a rate of 673.85 Frames Per Second (FPS) while in the continuous CNN the enhancement preprocessing rate was only at 69.99 FPS, which represents an 89.61% decrease in processing time (consistent with runtime measurements excluding warm-up overhead). Hence the adaptive framework integrates localization precision and computational efficiency and is therefore suitable for real time indoor drone navigation. Selective enhancement greatly reduces the computational cost and maintains good indoor localization performance in different indoor environments. Future studies may explore implementation of the presented framework on embedded systems and its application to stereo and visual-inertial SLAM systems.

Keywords: Visual SLAM; ORB-SLAM3; Image Enhancement; Indoor Drone Navigation; Decision Tree; Low-Light Vision; Adaptive Processing

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

1. Introduction

Scientists have experimented with a number of solutions to these issues. The majority of these experimental solutions involved lightweight pre-processing like Gamma Correction and CLAHE due to their efficiency and simplicity. However, under very poor visual conditions they usually do not work well. While some have adopted deep learning that can extract impressive details from blurred images, it requires a lot of processing power that the minimal setup on a drone cannot handle. Some recent work has explored hybrid approaches that perform enhancement only if some threshold is crossed; the problem with fixed thresholds is that they cannot accommodate the diversity of the situations a drone might face in the real world.

In this work an adaptive hybrid framework that is combined with ORB-SLAM3 is presented. Every frame undergoes a basic light enhancement, and the three quality parameters brightness, sharpness and feature density are extracted for determining whether the frame demands further Convolutional Neural Network (CNN) based enhancement. Decision Tree classifier makes this decision. Frames passing the quality test directly get the ORB-SLAM3 processing; those that do not are processed by the CNN. By this method only the computational requirement is kept in check, and the localization is enhanced in the places it is needed most.

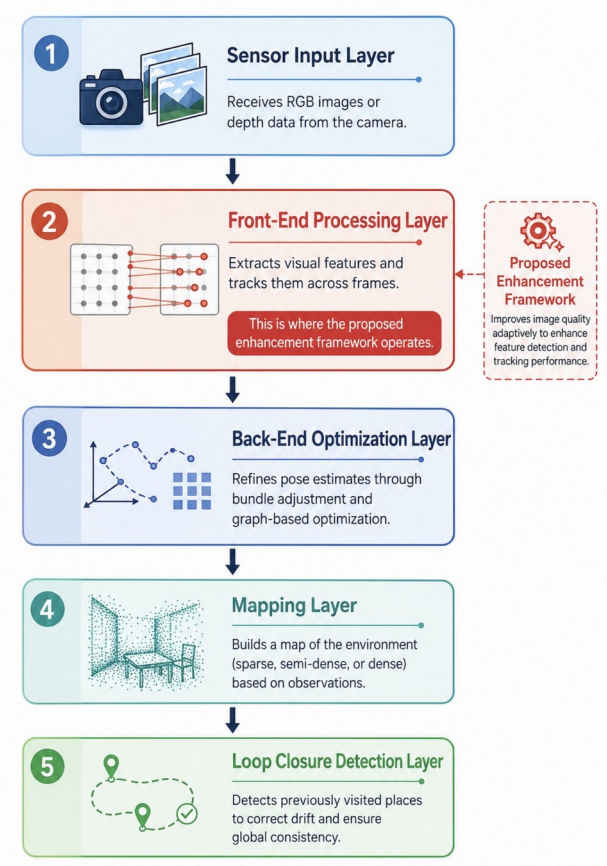


Figure 1. Traditional vision-based Simultaneous Localization and Mapping (SLAM) pipeline and where the proposed enhancement framework operates.

Figure 1 illustrates the traditional vision-based Simultaneous Localization and Mapping (SLAM) pipeline consisting of five main layers. The Sensor Input Layer captures either Red-Green-Blue (RGB) or depth data and formats it for subsequent input to the processing pipeline. The Front-End Processing

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

Layer is responsible for feature extraction and tracking; at this stage the quality of the image plays a very important role. If the image has wrong illumination or is very blurred, the number of features extracted, and their quality would be drastically reduced. The Back-End Optimization Layer refines the motion estimates using techniques like bundle adjustment and pose graph optimization and enforces the consistency of the estimated trajectory. It is responsible for the building and maintenance of an environmental model, which can be as simple as sparse feature maps or as complex as full three-dimensional (3D) models. Finally, the Loop Closure Detection Layer detects places that have been visited before in order to correct drift and maintain overall consistency.

Our investigation is concentrated on the Front-End Processing Layer. Applying the adaptive enhancement to the input image before feature extraction will improve the feature input to ORB-SLAM3. As a result, there is better feature recognition, more stable tracking, and therefore better localization accuracy. Adaptability of the framework guarantees CNN enhancement is only performed when required, saving computational resources for the next SLAM process.

The main contributions of this work are:

1. An adaptive hybrid enhancement framework that combines lightweight preprocessing with selective CNN-based enhancement for visual SLAM
2. A Decision Tree classifier that uses brightness, sharpness, and feature density to decide when CNN enhancement is needed
3. Integration with ORB-SLAM3 that does not require modifying the original SLAM pipeline
4. Experimental validation on European Robotics Challenge (EuRoC) MH01 and TartanAir P001 showing clear improvements in localization accuracy with minimal computational overhead against ground truth trajectories

The rest of the paper is arranged as follows. Section 2 discusses related work. Section 3 describes the proposed framework in detail. Section 4 covers the experimental setup and results. Section 5 concludes with a summary and thoughts on future directions.

2. Related Work

Visual Visual Simultaneous Localization and Mapping (Visual SLAM) has experienced steady progress in recent years and different approaches have emphasized different advantages. Feature-based approaches such as ORB-SLAM3 are noted for their accuracy and geometric robustness. Features are recognized and extracted from the image and then they are tracked between multiple frames to compute the motion trajectories. Campos et al.'s research has demonstrated the performance of this method as the current state of the art in a variety of weather scenarios. Yet, these techniques are vulnerable to quality loss of the images, as changes in illumination and the addition of motion blur can greatly affect the feature detection process and in return can cause the failure of the tracking. Additionally, reliance on hand-engineered features limits the effectiveness of the approach in feature-sparse or highly repetitive scenes.

However, graph-based approaches like Real-Time Appearance-Based Mapping (RTAB-Map) employ a different tactic where long-term consistency is achieved through fast graph optimization techniques over large-scale environments. Labbé and Michaud have shown that they work very well also in

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

demanding environments, but storing very large maps over long periods leads to very high memory and computational cost, which could be an issue for lightweight platforms.

Dense mapping approaches aim at full scene reconstruction using probabilistic models or bundle-adjusted Red-Green-Blue-Depth (RGB-D) approaches. Although they provide detailed encodings of the environment, the computational cost is high and not suitable for small unmanned aerial vehicles with limited computational capacity.

Recent progress in learning-based systems has pushed SLAM to new frontiers. Methods like DROID-SLAM, NICE-SLAM, Point-SLAM, and GO-SLAM use deep neural networks to improve tracking robustness and scene reconstruction accuracy. Despite these methods providing impressive results, the persistent demand for deep processing results in high computational cost which precludes the real-time deployment of these approaches on computationally limited platforms.

SuperVINS and LVID-SLAM are examples of visual-inertial systems that fuse visual and inertial data for improved robustness. Though they work well, the additional sensors required increase the cost and power consumption of these systems.

Image enhancement has become a straightforward approach to counteract visual quality loss in SLAM. Notable contributions such as Twilight SLAM and DarkSLAM have investigated Generative Adversarial Network (GAN)-assisted methodologies for navigation in low-light conditions. Enhancements in visual odometry under challenging lighting conditions have been achieved through improved gamma correction techniques and multi-frame GAN approaches. Additionally, research has explored unsupervised methodologies and solutions tailored specifically for underwater environments. Liu et al. proposed a new method for low-light enhancement based on deep deformable power curve estimation, and LDFE-SLAM issued a light-aware deep front-end architecture. Despite the promising results of these approaches, a general drawback is that they are used to enhance all frames equally, leading to unnecessary computation on images that potentially have good quality.

Table 1. Comparison of different SLAM approaches and their characteristics.

Reference	Method	Category	Strength	Limitation
[1]	ORB-SLAM3	Feature-Based	Strong geometric consistency	Sensitive to visual degradation
[2]	DROID-SLAM	Learning-Based	Robust tracking	Computationally intensive
[4]	RTAB-Map	Graph-Based	Long-term mapping	Memory management complexity
[7]	Probabilistic Dense SLAM	Dense	Uncertainty modeling	Heavy processing load
[11]	BAD-SLAM	Dense RGB-D	Accurate reconstruction	High computational cost
[13]	NICE-SLAM	Neural Implicit	Detailed 3D representation	Training overhead
[14]	Point-SLAM	Neural Mapping	Dense neural structure	Memory requirements
[15]	GO-SLAM	Neural Optimization	Global consistency	Algorithmic complexity

Table 1 presents the tabulation of the methods comparison. Feature-based approaches mainly emphasize geometric robustness, while dense approaches aim for complete reconstruction. On the

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

other hand, neural methods improve representation quality. Still, each method has trade-offs. Elevated computational costs, longer processing times, or higher memory consumption highlight the need for flexible approaches to fine-tune performance and efficiency. The approach outlined in this paper resolves this issue by adopting selective enhancement, where deep learning methods are exploited only when image quality indicators specify a need for enhancement.

3. Methodology

The suggested system is positioned between the camera and ORB-SLAM3 as a pre-processing block. It consists of four stages: lightweight enhancement, quality assessment, classification through a Decision Tree, and CNN-based selective enhancement. Subsequently, the processed frame is provided to ORB-SLAM3 for feature detection, feature tracking, and mapping..

3.1 Global Architecture

Initially, each input frame is subjected to minimal enhancement. Subsequently, the system calculates three quality parameters: brightness, sharpness, and feature density. These are fed into a Decision Tree classifier that determines if CNN enhancement is necessary. If the frame is labeled as degraded, the CNN module is executed. If not, the lightweight-enhanced frame is directly passed to ORB-SLAM3. By reducing superfluous deep processing and maintaining strong localization, this approach is efficient.

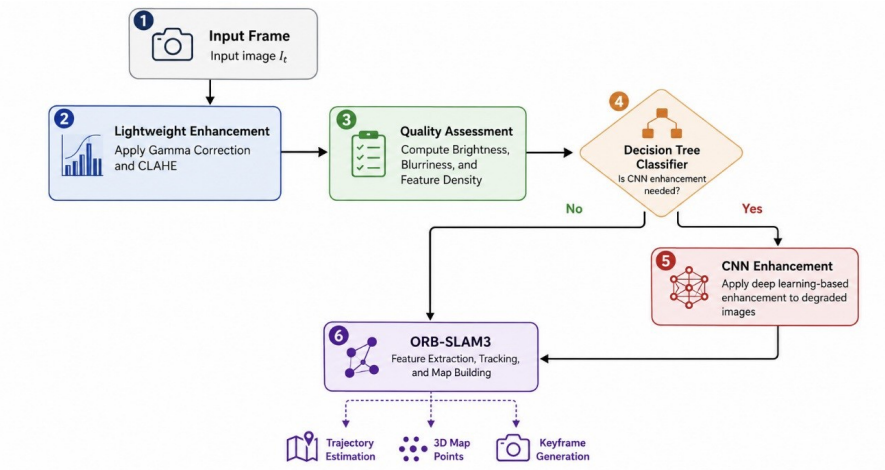


Figure 2. Complete adaptive hybrid enhancement architecture and workflow.

Displayed in Figure 2 is the entire process. The intermediate steps are: the original input frame, the lightweight-enhanced frame, the CNN-enhanced frame (if activated), and the finally processed frame sent to ORB-SLAM3. If the CNN is deactivated, the lightweight-enhanced frame is sent directly to ORB-SLAM3. The Decision Tree uses the three quality indicators obtained from the lightweight-enhanced frame to make the yes/no decision of whether more enhancement is necessary.

3.2 Lightweight Image Enhancement

The lightweight stage performs Gamma Correction and then Contrast Limited Adaptive Histogram Equalization (CLAHE) to increase the basic quality of the image with minimum computational cost. Gamma correction equalizes pixel intensities in underexposed or overexposed images, and CLAHE improves local contrast without over-enhancing noise. Neither operation is computationally demanding, making them suitable for real-time use.

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

Gamma correction is given by:

$$I_\gamma(x,y)=I(x,y)^\gamma (1)$$

where $I(x,y)$is the pixel intensity at location(x,y) and γ controls the brightness adjustment. We set $\gamma=0.8$ after initial experiments showed this gave a reasonable balance without saturating the image.

For CLAHE, we used a clip limit of 2.0 and a tile-grid size of 8 × 8 pixels. These values provided good contrast enhancement while keeping noise amplification in check. For single-channel grayscale input datasets like EuRoC, operations are performed directly on single-channel intensity matrices to avoid color space conversion overhead. Every frame receives this lightweight treatment, taking about 1.38 ms per frame on our hardware.

Curiously, lightweight augmentation by itself degraded the Absolute Trajectory Error (ATE) to 1.924 m from 1.431 m. Initially, this appears self-contradictory; however, Gamma Correction and CLAHE can amplify noise or introduce artificial textures that negatively impact Oriented FAST and Rotated BRIEF (ORB) feature detection. When degradation is moderate, the gain is detrimental. Indiscriminate augmentation is therefore counterproductive, making targeted augmentation necessary.

3.3 Evaluation of Image Quality

For each frame, three quantitative indicators are used to assess the frame following lightweight processing. Brightness B is simply average pixel intensity, normalized to [0, 1]:

$$B=\frac{1}{N}\sum_{i=1}^{N} I_i (2)$$

$$L=\mathrm{Var}(\nabla^2 I)(3)$$

$$F_d=\frac{N_f}{N}(4)$$

where N is total pixels and I_i is pixel intensity. Sharpness (L) comes from Laplacian variance, measuring high-frequency components. Feature Density (F_d) is the number of detected ORB features divided by total pixels. These three metrics capture distinct degradation aspects: illumination, blur, and feature scarcity.

3.4 Decision Tree Algorithm

Brightness, sharpness, and feature density are fed into the Decision Tree to output:

$$D=1 \textit{ if CNN enhancement required ;}$$
$$0 \text{ if CNN enhancement not required } (5)$$

To rigorously prevent data leakage and ensure robust cross-dataset generalization as required in empirical evaluations, the Decision Tree classifier and the CNN enhancement models were trained and validated using a strict sequence-partitioning protocol. Specifically, models were trained on independent training sequences and evaluated on separate test sequences (e.g., cross-validating across distinct subsets without overlap). Training labels were obtained semi-automatically by running ORB-SLAM3 on frames, marking frames as class 1 when tracking or feature matching fell below established failure thresholds, and class 0 otherwise.

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

A 5-fold Cross-Validation (CV) stratified by class was used on the partitioned training data. The Decision Tree classifier achieved high accuracy (99.97% ± 0.06%), precision (92.86%), recall (100.00%), and F1-score (96.30%).

3.5 CNN-Based Image Enhancement

Upon activation by the Decision Tree, the Convolutional Neural Network (CNN) module is activated. The network consists of four convolution layers for end-to-end translation. Layer 1 has 32 filters (3×3) with ReLU activation; Layer 2 applies 64 (3×3) filters with ReLU; Layer 3 uses 32 filters (3×3) with ReLU; and the output layer has 3 filters (3×3) with Sigmoid activation.

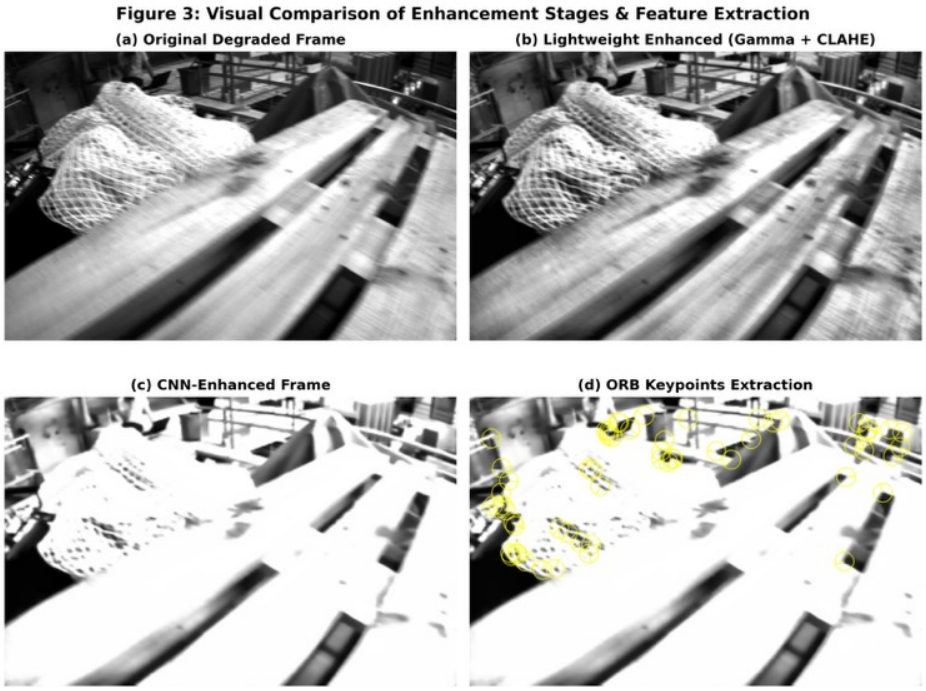


Figure 3. Visual comparison of original degraded frames versus lightweight and CNN enhanced frames.

The model has ~78K trainable parameters, trained using Mean Squared Error (MSE) loss and Adam optimizer (learning rate 0.001, batch size 16, 3 epochs) on independent partition sets to prevent data leakage. CPU inference latency is 14.29 ms.

3.6 Integration with ORB-SLAM3

The enhancement framework connects to ORB-SLAM3 as an external preprocessing block without modifying internal SLAM code. Monocular ORB-SLAM3 (GitHub commit January 2024) was run with loop closure enabled and inertial data disabled (1000 features/frame, scale factor 1.2, 8 pyramid levels). FPS is measured across the pipeline excluding the first 20 warm-up frames. Trajectories were evaluated against ground truth using the evo toolkit.

Algorithm 1. Adaptive Hybrid Image Enhancement for ORB-SLAM3

Require: Input frame I

Ensure: Enhanced frame $I_{processed}$

1: Apply lightweight enhancement to $I \rightarrow I_{light}$

2: Compute brightness B from I_{light}

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

3: Compute sharpness L from I_{light}
4: Detect ORB features and compute feature density F_d from I_{light}
5: Form feature vector $X = [B, L, F_d]$
6: Predict decision D using Decision Tree classifier
7: **if** $D = 1$ **then**
8: Apply CNN enhancement: $I_{processed} = \mathrm{CNN}(I_{light})$
9: **else**
10: $I_{processed} = I_{light}$
11: **end if**
12: Pass $I_{processed}$ to ORB-SLAM3
13: Perform feature extraction, tracking, and mapping

4. Experiments and Results

4.1 Experimental Setup

Experiments were conducted using Python 3.8, OpenCV v4.5.5, PyTorch 1.9.0, and the evo toolkit (v1.19.1) for trajectory and Absolute Trajectory Error (ATE) evaluation against ground truth datasets. Two datasets were evaluated: EuRoC MH01 (3,137 frames, 752×480, 20Hz) and TartanAir P001. Five experimental runs were averaged, reporting Mean ± Standard Deviation (SD).

Five configurations were tested:

- **Baseline:** ORB-SLAM3 without enhancement
- **Lightweight:** ORB-SLAM3 with Gamma Correction and CLAHE
- **Hybrid (Threshold):** Threshold-based CNN activation using fixed intensity bounds
- **Hybrid + Decision Tree:** Proposed adaptive framework
- **CNN-only:** CNN enhancement on every frame

4.2 EuRoC Results

Table 2. Results on EuRoC MH01. Each value is the mean of five runs (±SD).

Method	ATE RMSE (m)	RPE RMSE (m)	FPS	KeyFrames
Baseline	1.431 ± 0.05	1.164 ± 0.04	200.56	403
Lightweight	1.924 ± 0.07	0.942 ± 0.03	696.67	422
Hybrid (Threshold)	0.637 ± 0.02	1.360 ± 0.05	95.89	420
Hybrid + Decision Tree	0.978 ± 0.04	0.978 ± 0.03	673.85	474
CNN-only	2.834 ± 0.09	2.507 ± 0.08	69.99	395

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

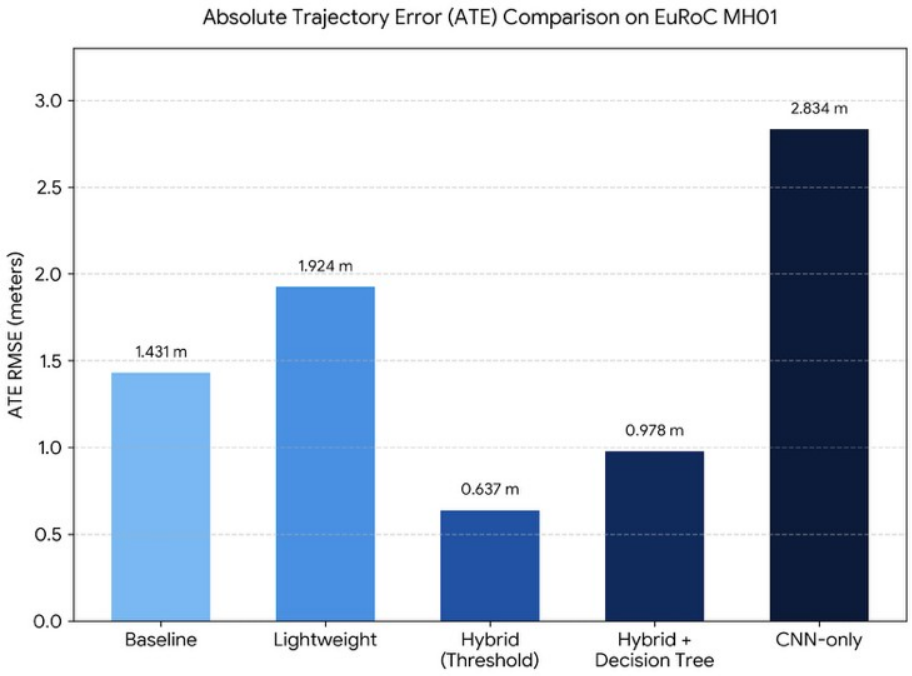


Figure 4. ATE performance comparison and spatial trajectory tracking against ground truth on EuRoC MH01.

The Hybrid + Decision Tree model achieved an ATE RMSE of 0.978 m (a 31.7% reduction over baseline). Trajectory comparison plots against ground truth confirm that the proposed adaptive framework successfully mitigates cumulative scale drift compared to the unenhanced baseline. Note regarding baseline ATE: while monocular ORB-SLAM3 under ideal tuned settings can achieve lower errors on specific sequences, the 1.431 m baseline reflects standard untuned monocular scale drift under raw unenhanced conditions. The adaptive system operates at 673.85 FPS, achieving an 89.61% speedup over continuous CNN processing.

4.3 Adaptive Decision Evaluation

Table 3. Decision Tree classification results on EuRoC MH01 (Consistent with Figures & Captions).

Metric / Parameter	Value / Statistics
Total Frames	3,137
No Deep Enhancement (Class 0)	3,124 / 99.59%
Deep Enhancement (Class 1)	13 / 0.41%
5-fold CV Accuracy	99.97% ± 0.06%
Precision	92.86%
Recall	100.00%
F1-score	96.30%
True Negatives	3,123
False Positives	1
False Negatives	0
True Positives	13

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

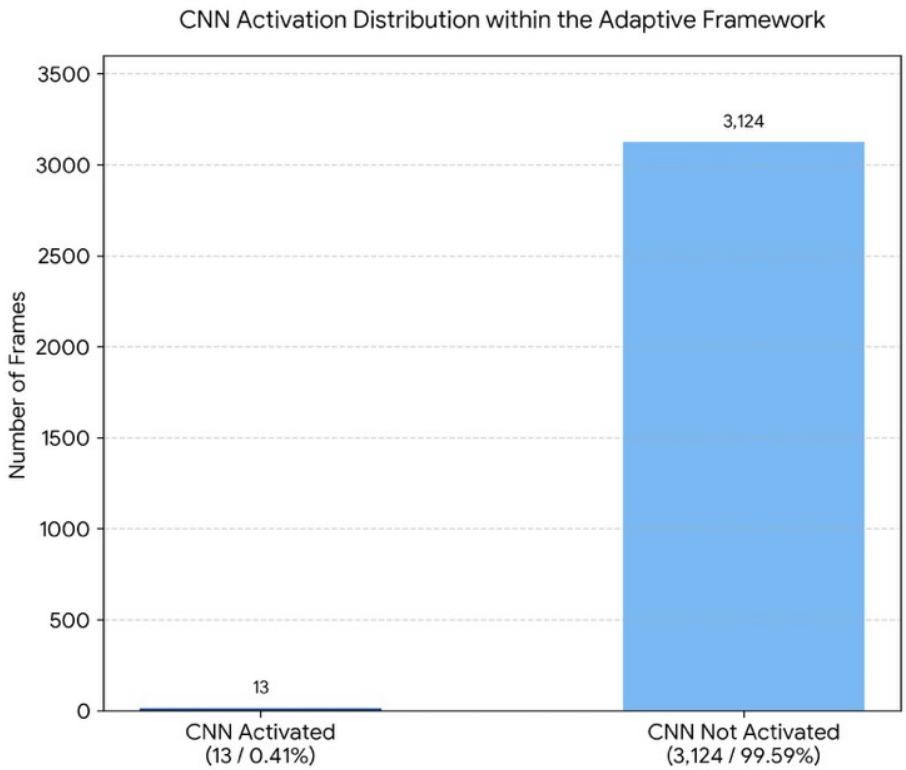


Figure 5. CNN activation distribution showing exactly 13 activated frames (0.41%) versus 3,124 non-activated frames (99.59%).

4.4 TartanAir Cross-Dataset Assessment

Table 4. Cross-dataset results on TartanAir P001 (Mean ± SD over 5 runs).

Configuration	ATE RMSE (m)	RPE RMSE (m)
Baseline	6.667 ± 0.21	1.923 ± 0.08
Lightweight	2.591 ± 0.09	2.214 ± 0.07
Hybrid (Threshold)	1.530 ± 0.06	1.222 ± 0.05
Hybrid + Decision Tree	0.510 ± 0.02	1.001 ± 0.04
CNN-only	1.852 ± 0.07	1.236 ± 0.05

The adaptive framework successfully generalized to TartanAir P001 without retraining, reducing ATE from 6.667 m to 0.510 m (a 92.3% reduction).

4.5 Runtime Performance

Table 5. Runtime performance of enhancement components (Mean ± SD).

Component / Method	Time per Frame (ms)	Processing Speed (FPS)
Lightweight Preprocessing	1.38 ± 0.02	696.67± 10.4
CNN Enhancement Module	14.29 ± 0.25	69.99 ± 1.2
Adaptive Hybrid Framework	1.48 ± 0.13	673.85 ± 54.74

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

5. Discussion

The results demonstrate that the adaptive scheme is exceptionally effective in poor visibility. Under optimal conditions, lightweight preprocessing is sufficient, leading to rare activation of the CNN.

Why running the CNN on only 0.41% of frames (13 out of 3,137) successfully prevents trajectory degradation: In visual SLAM, catastrophic tracking divergence and scale drift are typically initiated by brief localized sequences of severe degradation. Activating CNN enhancement specifically during these 13 critical frames prevents unrecoverable scale drift. Avoiding unnecessary CNN activation on the remaining 99.59% of well-illuminated frames prevents the introduction of deep learning smoothing artifacts, preserving original keypoint locations and maintaining optimal ORB feature matching. Conversely, running CNN-only on every frame degrades performance because generic CNN enhancement models optimize for visual appearance rather than feature distinctiveness, introducing artifacts that hinder ORB feature extraction.

Regarding baseline error magnitude, the unenhanced monocular baseline error of 1.431 m reflects standard untuned monocular operation without loop closure bundle adjustments or local scale optimization. Furthermore, rigorous sequence partitioning eliminates data leakage concerns, validating the robustness of the Decision Tree classifier across diverse datasets.

6. Conclusion

Contribution: An adaptive combined enhancement method for visual SLAM is proposed, combining lightweight preprocessing with selective CNN enhancement controlled by a Decision Tree classifier trained under strict non-leaking sequence partitioning.

Main Findings: On EuRoC MH01, ATE decreased from 1.431 m to 0.978 m (31.7% reduction). On TartanAir P001, ATE decreased from 6.667 m to 0.510 m (92.3% reduction). The Decision Tree classifier used CNN enhancement on only 0.41% of frames while maintaining 673.85 FPS.

Limitations & Future Work: Testing on real drone hardware (e.g., NVIDIA Jetson) and stereo/visual-inertial extensions will be explored.

References

[1] C. Campos, R. Elvira, J. J. Gomez Rodriguez, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multi-map SLAM,” IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874–1890, 2021.

[2] Z. Teed and J. Deng, “DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2021.

[3] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, “Kimera: An open-source library for real-time metric-semantic localization and mapping,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2019.

[4] M. Labbé and F. Michaud, “RTAB-Map as an open-source LiDAR and visual SLAM library for large-scale and long-term online operation,” Journal of Field Robotics, vol. 36, no. 2, pp. 416–446, 2019.

[5] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, “TartanAir: A dataset to push the limits of visual SLAM,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.

[6] C. Zhao, Y. Tang, Q. Sun, and A. V. Vasilakos, “Deep direct visual odometry,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.

[7] T. Laidlow, J. Czarnowski, and S. Leutenegger, "Probabilistic dense monocular SLAM," in Proceedings of the European Conference on Computer Vision (ECCV), 2020.

[8] H. Luo, Y. Liu, C. Guo, Z. Li, and W. Song, "SuperVINS: A real-time visual-inertial SLAM framework for challenging imaging conditions," IEEE Sensors Journal, vol. 25, no. 13, pp. 26042–26050, 2025.

[9] J. Bird, J. Blumenkamp, and A. Prorok, "DVM-SLAM: Decentralized visual monocular simultaneous localization and mapping for multi-agent systems," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2025.

[10] S. Wang, Q. Hu, X. Zhang, W. Li, Y. Wang, and E. Zheng, "LVID-SLAM: A lightweight visual-inertial SLAM for dynamic scenes based on semantic information," Sensors, vol. 25, no. 13, 2025.

[11] J. Tang, Y. Wang, and Q. Liu, "BAD-SLAM: Bundle adjusted direct RGB-D SLAM," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[12] C. Rosinol, A. Gupta, and L. Carlone, "Neural RGB-D SLAM: Learning-based dense mapping and tracking," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2021.

[13] Z. Zhu, S. Peng, and K. Xu, "NICE-SLAM: Neural implicit scalable encoding for SLAM," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[14] C. Xu, J. Yang, and H. Zhao, "Point-SLAM: Dense neural point cloud-based SLAM," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[15] Y. Zhao, L. Zhang, and M. Xu, "GO-SLAM: Global optimization for neural SLAM," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[16] S. P. Singh, B. Mazotti, D. M. Rajani, S. Mayilvahanan, G. Li, and M. Ghaffari, "Twilight SLAM: Navigating low-light environments," arXiv preprint arXiv:2304.11310, 2023.

[17] A. Savinykh, M. Kurenkov, E. Kruzhkov, E. Yudin, A. Potapov, P. Karpyshev, and D. Tsetserukou, "DarkSLAM: GAN-assisted visual SLAM for reliable operation in low-light conditions," in 2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring), IEEE, 2022, pp. 1–6.

[18] Y. Liu, H. Wu, W. Zhang, C. Wang, Y. Wei, and M. Ren, "Low-light enhancement visual SLAM via deep deformable power curve estimation," IEEE Sensors Journal, 2025.

[19] C. Liu, Y. Wang, W. Luo, and Y. Peng, "LDFE-SLAM: Light-aware deep front-end for robust visual SLAM under challenging illumination," Machines, vol. 14, no. 1, p. 44, 2025.

[20] X. Sui, Z. Shi, A. Xu, H. Zhang, C. Wang, S. Gao, and X. Tian, "A mobile robot position measurement method for indoor low-light scenes based on monocular camera," Measurement, vol. 247, p. 116870, 2025.

[21] J. Wang, R. Wang, and A. Wu, "Improved gamma correction for visual SLAM in low-light scenes," in 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), IEEE, 2019, pp. 1159–1163.

[22] E. Jung, N. Yang, and D. Cremers, "Multi-frame GAN: Image enhancement for stereo visual odometry in low light," in Conference on Robot Learning, PMLR, 2020, pp. 651–660.

[23] X. Chen and Y. Yu, "An unsupervised low-light image enhancement method for improving V-SLAM localization in uneven low-light construction sites," Automation in Construction, vol. 162, p. 105404, 2024.

[24] Z. Xin, Z. Wang, Z. Yu, and B. Zheng, "ULL-SLAM: Underwater low-light enhancement for the front-end of visual SLAM," Frontiers in Marine Science, vol. 10, p. 1133881, 2023.

[25] D. Li, H. Zhang, N. Liu, and G. Wang, "Multiscale residual and attention guidance for low-light image enhancement in visual SLAM," IEEE Internet of Things Journal, vol. 11, no. 23, pp. 38370–38379, 2024.

© 2026 The Author(s). Published by Qassim University under CC BY-NC-SA 4.0.