AI-Based 3D Vision System for Inspection and Monitoring

Nazanin Padkan[1,2], Samuele Facenda [1], Luca Morelli[1], Ahmad Elalailyi[1,3], Fabio Remondino[1]

[1] 3D Optical Metrology (3DOM) unit, Bruno Kessler Foundation (FBK), Trento, Italy
Web: http://3dom.fbk.eu - Email: <npadkan><sfacenda><lmorelli><aelalailyi><remondino>@fbk.eu

[2] Dept. Mathematics, Computer Science and Physics, University of Udine, Italy

[3] Dept. of Architecture, Built environment and Construction engineering (DABC), Politecnico di Milano, Italy

Commission II

KEY WORDS: Crack Detection, Monocular Depth Estimation, Stereo Matching, Visual-SLAM, YOLO, Inspection, Monitoring

ABSTRACT

Simultaneous Localization and Mapping (SLAM) has become a fundamental technology in various applications, including robotics, autonomous navigation, geographic information systems (GIS), and infrastructure inspection. This paper presents a new version of GuPho, a low-cost, lightweight, and portable visual SLAM-based system equipped with AI-driven capabilities for real-time mapping, object detection, and defect analysis. The system integrates stereo vision and deep learning (DL) methods to enhance spatial understanding and enable accurate real-time scene interpretation. In particular, we explore DL-based semantic segmentation, monocular depth estimation (MDE), and stereo depth estimation to improve 3D reconstruction and size measurement of cracks for infrastructure monitoring. We implement state-of-the-art neural networks, including RF-DETR and YOLO for real-time crack and windows segmentation and Depth Anything V2, Depth Pro, and Unimatch for depth estimation. Our results demonstrate the potential of GuPho as an affordable and efficient system for real-time mobile mapping and defect assessment. The real-time and AI capabilities of our in-house solution are showcased here: https://youtu.be/ATIwn4zOSFw

1. INTRODUCTION

Mobile mapping systems (MMS) are increasingly utilized in the field of surveying and mostly rely on Simultaneous Localization and Mapping (SLAM) techniques. SLAM is widely applied in various domains, including robotics (Yousif et al., 2015), autonomous vehicles (Takleh et al., 2018), geographic information systems (GIS) (Ghosh et al., 2020), augmented reality (AR) (Liu et al., 2016), tunnel inspections (Loupos et al., 2018), and more. SLAM-based systems simultaneously construct a map of the environment while determining the position of the device within that map (Alsadik et al., 2021).

Integrating artificial intelligence (AI) capabilities, such as object detection and semantic segmentation, into MMS enhances the richness of automatically extracted information, reducing the need for manual intervention. This advancement improves efficiency for both autonomous systems and human operators.

Beyond the accuracy of object detection and segmentation, real-time performance is also essential for various applications, such as road pavement inspection based on Convolutional Neural Networks (CNNs) (Zhang et al., 2023), automatic bridge crack detection based on STDC-Net (Li et al., 2022), crack segmentation with multi-type features of different depths (Yong et al., 2022), etc.. For such tasks, the detection model must be computationally efficient to ensure real-time processing speed while maintaining accuracy.

Recent studies have extended mobile mapping capabilities to handheld platforms, leveraging AI techniques to enhance spatial awareness and contextual understanding in real-time. Li et al. (2024) introduced YoD-SLAM, an indoor VSLAM system based on ORB-SLAM3, YOLOv8 and depth information using a RGB-D camera. YG-SLAM (Chu et al., 2023) presents a method based on YOLOv8 and geometric constraints within the ORB-SLAM2 framework to adapt effectively to dynamic scenarios.

This work proposes the integration of DL algorithms within the SLAM capabilities of GuPho (Di Stefano et al., 2021; Torresani et al., 2021; Menna et al., 2022), a low-cost and portable modular prototype system based on stereo vision and Visual-SLAM (V-SLAM), with an emphasis on real-time processing. This research primarily explores:

- Real-time semantic segmentation and object detection, enabling online identification and monitoring, especially for pavement and structural defects;
- Measurements of segmented objects using multiple approaches, including MDE and DL-based stereo depth estimation, comparing results with traditional stereo triangulation.

This study builds upon the work presented in Padkan et al., 2023a, which explored the integration of deep learning techniques into the GuPho system for semantic segmentation, object detection, and monocular depth estimation. In contrast to the prior work, the present research emphasizes real-time processing capabilities and investigates multiple depth measurement approaches—including stereo triangulation, monocular depth estimation, and deep learning-based stereo depth estimation—for improved metric analysis. Additionally, the application focus is extended to include the monitoring of structural and pavement defects under real-world conditions.

2. GUPHO SYSTEM

GuPho (Guided Photogrammetry System) is an ARM-based low-cost, lightweight, and handheld mobile mapping system based on V-SLAM algorithms. The initial version of the system is built on a Raspberry Pi 4 Model B, featuring a Broadcom BCM2711 chip, a quad-core Cortex-A72 processor (ARM v8) running at 1.8 GHz, and 8 GB of RAM.

Designed for both indoor and outdoor applications, GuPho supports fish-eye and rectangular lenses. OpenVSLAM (Sumikura et al., 2019), which builds upon ORB-SLAM2 (Mur-Artal and Tardós, 2017), is used to provide real-time feedback to the user in terms of reconstruction (triangulated 3D tie points) and user trajectory (Figure 1).

To enhance real-time DL capabilities, the Raspberry Pi was replaced with a more powerful Jetson Nano with 4GB of RAM,

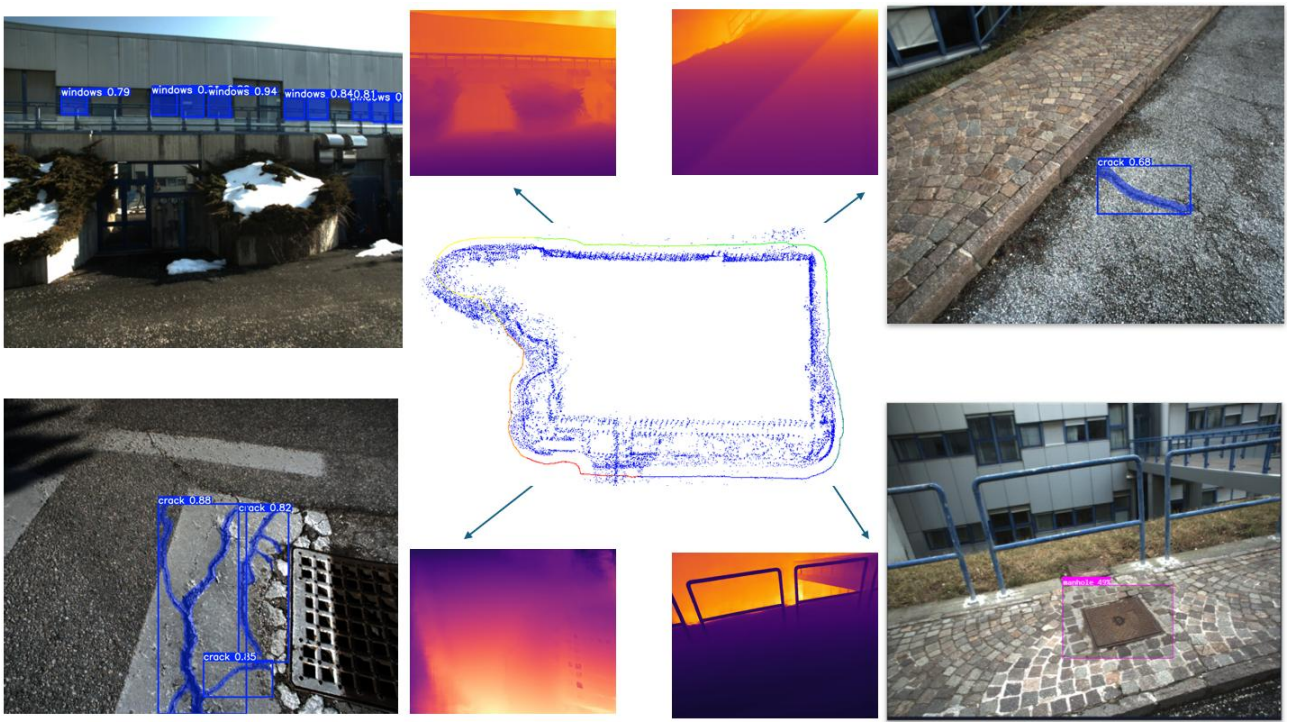

Figure 1: Results of GuPho's SLAM process in an outdoor scenario with segmentation (cracks and windows) and detection (manhole) results based on YOLO and estimated depth using Depth Anything V2.

significantly improving processing speed and allowing GuPho to recognize objects in real-time. Compared to the Raspberry Pi, Jetson Nano allows DL algorithms to be run using a NVIDIA Maxwell architecture GPU with 128 CUDA cores. Despite this upgrade, the core components of the V-SLAM system remain unchanged. Beyond its SLAM capabilities, GuPho has been extended to incorporate object detection and semantic segmentation (specifically cracks and windows in this paper), allowing GuPho to enrich its 3D reconstructions with a semantic layer of information to the geometric and colorimetric data.

3. INTEGRATION OF DL-BASED CAPABILITIES IN GUPHO

In our development, deep learning (DL)-based algorithms have been integrated into GuPho to support multiple tasks. This includes object detection and instance segmentation (Section 3.1) and depth estimation (Section 3.2) which run in parallel with the visual simultaneous localization and mapping (V-SLAM) module.

3.1 Instance Segmentation and Object Detection

The primary objective of 2D object detection is to identify the precise 2D location of objects in a given image and assign relevant labels to their bounding boxes. In contrast, instance segmentation goes beyond object detection by labelling individual objects at the pixel level, providing a more detailed understanding of object boundaries.

This unified approach enables simultaneous environment mapping and semantic understanding of the scene in real time. Applications include defect and obstacle detection (e.g., cracks and stones) (Padkan et al., 2023a), agricultural monitoring (e.g., grape detection in vineyards) (Facenda et al., 2025), and building inspection (e.g., window detection).

One of the groundbreaking state-of-the-art neural networks for image object detection is the YOLO (You Only Look Once) algorithm (Redmon et al., 2016), which achieves both high precision and speed. Due to its efficiency and robustness, we adopted YOLO-based models for crack detection and instance segmentation within our framework. Specifically, we used YOLOv8 (J. Glenn, 2023), YOLOv11 (Jocher et al., 2024), and RF-DETR (Robinson et al., 2025) for training on our dataset, leveraging their improved accuracy, speed, and adaptability to diverse environments.

However, since GuPho runs on the Jetson Nano platform, which has limited RAM and computational resources, we were constrained in deploying heavier models in real time. This limitation necessitated a focus on resource-efficient models optimized for edge computing, ensuring efficient inference without compromising critical performance.

3.2 Depth Estimation

Beyond geometric SLAM capabilities, monocular depth estimation (MDE) is explored as a complementary approach to perform measurements in areas with weak texture. Additionally, a DL-based stereo depth estimation model is considered.

3.2.1 Monocular Depth Estimation

Although GuPho is equipped with stereo camera pair, MDE i.e. the process of estimating depth from a single RGB image is added (Ming et al., 2021). Unlike stereo vision, which requires two synchronized cameras to compute a depth based on disparity, MDE algorithms primarily rely on DL models to infer depth from a single image (Padkan et al., 2023b; Yan et al., 2025b). MDE has numerous applications in robotics (Dong et al., 2022), cultural heritage preservation (Yao et al., 2024), smart agriculture (Cui et al., 2022) and autonomous driving (Yan et al., 2025a). MDE performs well in scenarios where traditional photogrammetry struggles, such as textureless environments.

Depth Anything V2 (Yang et al., 2024b): it is a state-of-the-art MDE model which outperforms its predecessor, Depth Anything V1 (Yang et al., 2024a). This model improves accuracy

Figure 2: Segmented and detected objects on GuPho images, using rectangular and fish-eye lenses.

estimation leveraging training on synthetic images instead of real ones, scaling up the teacher model, and leveraging large-scale pseudo-labeled real images for student models training. This model is based on training a reliable teacher model based on DINOv2-G purely on high-quality synthetic images, producing precise pseudo depth on large-scale unlabeled real images, and training final student models on pseudo-labeled real images for robust generalization.

Depth Pro (Bochkovskii et al., 2024): it is a zero shot MDE model developed by Apple that estimates depth from a single image. This model estimates the metric depth map without needing any extra information such as camera parameters. By combining real-world and synthetic data during training, it excels at capturing fine details, especially around edges, making it ideal for applications like 3D modeling and augmented reality.Depth Pro network processes an input image by downsampling it across multiple scales. At each scale, the image is divided into patches, which are encoded using a ViT-based (Dosovitskiy, A., 2020) patch encoder with shared weights across all scales. These patches are then merged to form feature maps, which are subsequently upsampled and fused using a DPT decoder (Ranftl et al., 2021).Additionally, predictions are grounded by a separate image encoder that provides global context, ensuring that the output benefits from both local and holistic information. The input image is upsampled or downsampled to 1536x1536 px resolution and the estimated depth is resized to the original image resolution.

3.2.2 Stereo Approaches

GuPho's stereo vision capability allows us to include also a DL-based stereo estimation approach which could support e.g. crack size measurement in areas with poor texture.
We utilize UniMatch (Xu et al., 2023), a unified formulation for three tasks: optical flow, rectified stereo matching and unrectified stereo depth estimation based on transformer (cross-attention mechanism). These tasks are framed as a dense correspondence matching problem, where feature similarities are directly compared to establish correspondences. The framework utilizes a transformer-based architecture with the cross-attention mechanism, which facilitates cross-view interactions, enabling the model to extract highly discriminative features. First, dense features are extracted from the input images, followed by a parameter-free matching layer that find correspondences task-specific constraints. A self-attention layer is then employed to propagate high-quality predictions to regions that were previously unmatched, based on feature self-similarity. Notably, the same set of learnable parameters is shared across all three tasks, making cross-task transfer possible and improving the generalization of the model in motion and depth estimation.
Additionally, GuPho includes also a SIFT-based stereo matching (Lowe, 2004) method, where keypoints are extracted from YOLOv8-segmented crack masks in both left and right images to measure crack sizes efficiently.

4. EXPERIMENTAL EVALUATION AND RESULTS

4.1. Segmentation Results

The dataset used to train and evaluate the DL-based crack detection model has been acquired with GuPho and includes images 1280x1024 px featuring cracks on surfaces such as asphalt, cement, sidewalks, and building walls. After data augmentation, the dataset comprised 633 images, which were split into 70% for training, 20% for validation, and 10% for testing.
We employed models from the YOLO family, which are known for their efficiency and accuracy in object detection tasks. Specifically, YOLOv8 (Jocher et al., 2023) and YOLOv11 (Jocher et al., 2024) were trained on our dataset. After 100 training epochs, YOLOv8x - the largest variant - achieved a recall of 73% and a precision of 68%, while YOLOv8l reached a recall of 62% and a precision of 70% (Table 1, Figure 3). YOLOv8s, the smallest model, did not show improvement beyond 180 epochs, stabilizing at 36% recall and 65% precision. For comparative analysis, we also trained YOLOv11 and RF-DETR (Robinson et al., 2025). YOLOv11, after 200 epochs, achieved a recall of 41% and a precision of 80%, while RF-DETR attained 52% recall and 61% precision. Although performance metrics were similar across models, YOLOv8 was selected for

real-time deployment due to its superior trade-off between detection performance and computational efficiency.

Model	Epochs	Recall	Precision	mAP
YOLOv8S	180	0.36	0.65	0.33
YOLOv8X	100	0.73	0.68	0.69
YOLOv8L	100	0.62	0.70	0.62
YOLOv11x	150	0.33	0.76	0.33
YOLOv11x	200	0.41	0.80	0.42
RF-DETR	200	0.52	0.61	0.38

Table 1: Evaluation of Crack Segmentation and detection using different models.

4.2. Stereo

A traditional approach for crack size estimation is implemented using SIFT features (Lowe, 2004) in combination with semantic segmentation. In this method, we first segment the crack using YOLOv8 in both the left and right images. After segmentation, we extract keypoints and descriptors from the crack masks (Figure 3), to find 2D tie points with classical nearest neighbor approach. Since the GuPho system is pre-calibrated, tie points are triangulated to reconstruct the crack in 3D.
After reconstructing the crack in 3D (Figure 3) using the extracted keypoints, the length measured was 144 cm, compared to 143 cm measurement of the ground truth (GT). This discrepancy of 1 cm can be attributed to the annotated segmentation mask, which slightly overestimates the crack's actual dimensions. Such variations typically introduce an error margin of 1–3 cm, considered acceptable for the application sector.

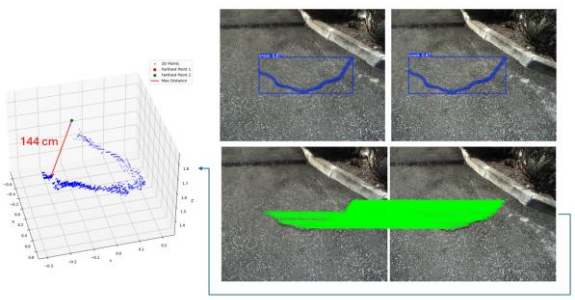

Figure 3: SIFT-based matching of Crack 1, where keypoints are selected based on the mask values from YOLOv8-segmented crack and its 3D visualization.

Stereo matching in real-world scenarios, particularly for fine-scale features such as cracks, presents significant challenges due to inconsistencies in object segmentation across stereo pairs. Variations in illumination, focus, occlusions, and viewing angles often lead to different segmentation results between the left and right images. For example, a crack may be detected in one image but missed or fragmented in the other, resulting in an unequal number of detected instances (Figure 4). These discrepancies complicate the establishment of reliable feature correspondences using traditional keypoint-based methods such as SIFT, which rely on consistent and well-aligned visual cues. Consequently, incorrect matches may lead to inaccurate object matching, unreliable 3D reconstruction and size measurements.

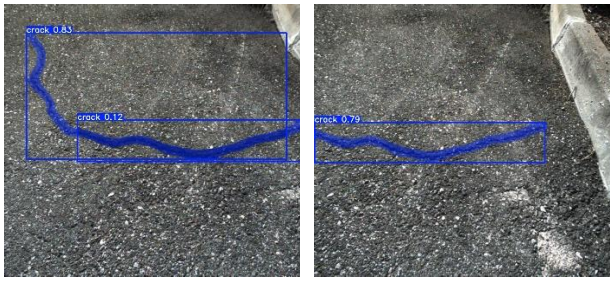

Figure 4: A pair of images with different number of segmented cracks.

4.3. YOLO + MDE

To overcome the above-mentioned limitations, GuPho couples a monocular depth estimation model with a segmentation neural network. Specifically, we employed metric MDE algorithms to estimate the depth from a master camera (left) and YOLOv8 to detect and localize objects through bounding boxes and masks. By combining the estimated depth with the spatial segmentation data, size and 3D position of an object can be directly estimated from a single view. This strategy eliminates the dependency on consistent segmentation across stereo pairs and provides a more robust alternative in scenarios where stereo matching is unreliable. One aspect of our investigation involves assessing the accuracy of the generated point clouds based on the estimated depth from different methods. For this, we used the left image as the only input for MDE algorithms, and both the left and right images for the stereo approaches.
Using Open3D (Zhou et al., 2018) and known camera intrinsic parameters, we generated point clouds from RGB images and the corresponding estimated depth maps.
In this analysis, we choose different cracks. These cracks are gathered in different environments and from different viewpoints. For each example, we estimated the depth using two MDE algorithms (Depth AnythingV2 and Depth Pro) as well as DL-based stereo depth estimation (UniMatch) and SIFT-based crack matching. Since GT data is unavailable for this dataset, we manually measured the distance between two points at the beginning and end of the crack to serve as the GT.
We then compared the distance of the same points in the generated point clouds from the mentioned algorithms. For the Crack 1, the size of the measured part of the crack is 143 cm. Based on the estimated depth using Depth Anything V2 and Depth Pro, the measured sizes of the crack in CloudCompare software were 200 cm and 141 cm, respectively, as shown in Figure 5.
It is important to note that for some cracks, such as Crack 4 and Crack 5 (Table 2), MDE algorithms struggle to estimate the real depth values. This is because the images contain few distinguishable features and objects, making it difficult even for humans to accurately estimate the real depth values.
Additionally, using the UniMatch algorithm, both left and right images were used as input. Based on the estimated stereo depth, the measured size of the crack in the generated point cloud was 137 cm, which is 6cm smaller than the GT size of 143cm.
To improve computational efficiency, we optimize the SIFT matching process by selecting a limited number of keypoints per row, rather than considering all pixels within the segmentation mask. This significantly enhances processing speed while maintaining reliable feature matching.

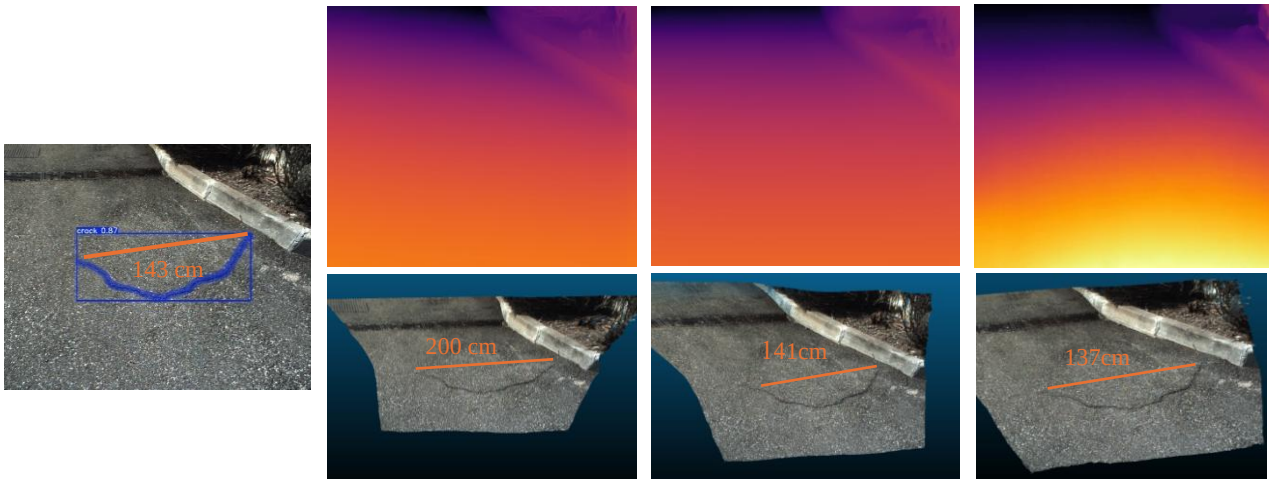

Figure 5: The estimated depth using MDE algorithms (Depth Anything V2 and Depth Pro) and DL-based stereo depth (UniMatch) with their point clouds and the measured sizes in CloudCompare.

	GT size (cm)	Depth Pro (cm)	Anything V2 (cm)	UniMatch (cm)	SIFT (cm)
Crack 1	143	141	200	137	144
Crack 2	77	61	100	70	72
Crack 3	148	58	163	102	143
Crack 4	117	135	234	101	115
Crack 5	76	89	156	165	74
Crack 6	40	59	73	49	41
Crack 7	75	55	79	100	80
Crack 8	35	16	50	40	38

Table 2: Comparison of crack size measurement performance of the used methods.

	GT size (cm)	Depth Pro (cm)	AnythingV 2 (cm)	UniMatch (cm)	SIFT (cm)
Window 1	95	90	107	100	92
Window 2	103	100	123	110	105
Window 3	97	92	105	102	95
Window 4	108	105	125	115	110
Window 5	97	93	89	100	96

Table 3: Comparison of window size measurement performance of the used methods.

Table 2 presents the measured crack sizes obtained using the different methods described. According to the measured size of cracks based on each method, SIFT has lower error compared to GT values.

However, despite its relative accuracy, this method has certain limitations. Its performance is highly dependent on the quality of crack segmentation and annotation. In some cases, the matched keypoints may lie slightly outside the actual crack boundaries, leading to an overestimation of the crack width. Overall, the SIFT-based stereo matching method outperformed the other approaches in our study. While the point clouds generated by the monocular depth estimation (MDE) algorithms were visually clean and free of significant noise, they lacked accuracy in terms of metric measurements for crack size estimation. Similarly, UniMatch did not perform well for this task, the estimated depths exhibited noticeable noise, resulting in deformed point clouds and inaccurate measurements. Since both the MDE models and UniMatch are DL-based, their performance is closely tied to the quality and relevance of the training data. Fine-tuning these models on domain-specific datasets may improve their metric accuracy and overall robustness in future applications.

To evaluate the quality of the generated point clouds, we performed both size measurements and a plane-fitting approach in CloudCompare. This method allowed us to assess whether the generated data maintain consistent elevation values in areas where cracks are present (Figure 6). Table 4 presents the RMSE values calculated using CloudCompare by comparing the generated point cloud with an ideal plane to assess their accuracy. computed for the cracks based on this comparison. Based on the results, Depth Pro shows lower RMSE values compared to the other two methods.

Also we did the same experiment for segmenting the windows. Based on our results (Table 3), SIFT outperformed other methods in terms of lower difference compared to GT size.

	Depth Pro	Depth Anything V2	UniMatch
Crack 1	1.734	4.835	6.870
Crack 2	2.708	3.193	10
Crack 3	3.148	2.965	5
Crack 4	3.830	4.181	3.947
Crack 5	1.859	2.480	14
Crack 6	1.456	2.324	1.733
Crack 7	1.280	2.671	8.754
Crack 8	1.573	3.629	4.134

Table 4: RMSE for evaluating the generated point clouds of DL-based algorithms in mm.

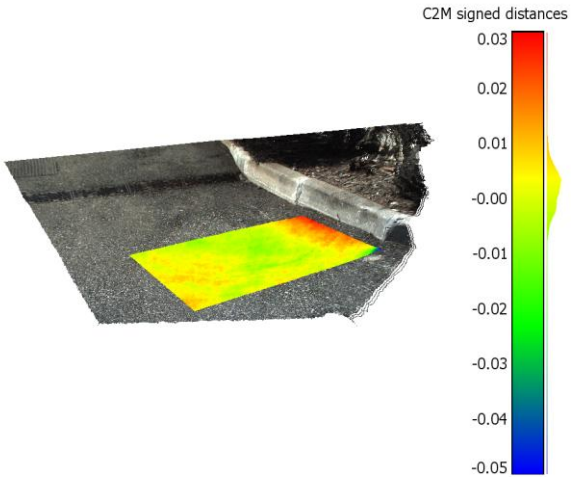

Figure 6: Fitting a plan to the generated point clouds of Crack 1 by Depth Anything V2.

Among the evaluated methods, SIFT-based crack matching yields the most accurate results, producing measurements closest

to the ground truth (GT) values. However, despite its relatively high accuracy, this approach has notable limitations. Its effectiveness heavily depends on the quality of crack segmentation and annotation. In some instances, matched keypoints may fall just outside the actual crack boundaries, leading to slight overestimations—particularly in crack width. Furthermore, the method relies on consistent segmentation of the same crack in both the left and right images. Variations in camera angle, lighting conditions, and image clarity can result in the crack being missed in one image or segmented as multiple fragments in the other. These inconsistencies can impair keypoint matching and reduce measurement reliability. Nevertheless, the accuracy of this method can improve in subsequent frames, as better camera positioning may lead to more consistent segmentation and more reliable matching.

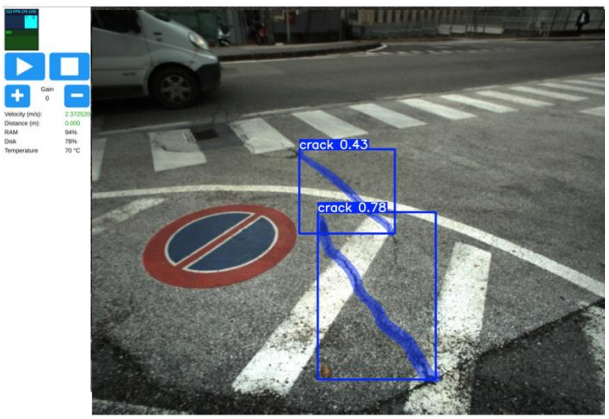

Figure 7: The real-time interface of crack segmentation on GuPho.

4.4. Real-time Evaluations

In order to evaluate the real-time performance of the system on GuPho, several YOLOv8 models were tested in different environments using GuPho. Table 5 shows the average inference times for both segmentation (crack detection) and detection (window detection) tasks. For crack detection in urban settings, YOLOv8s model performed well, with average inference times of 637.55 ms and 568.23 ms, suitable for near real-time use. The larger YOLOv8x model took 1254.57 ms, which is significantly slower.

For window detection on buildings, the YOLOv8x detection model showed higher inference times of 1848.67 ms and 1999.1 ms, making it less practical for real-time applications without further optimization. However, when using the smaller YOLOv8s model in segmentation mode for the same task, inference time dropped to 645.25 ms, showing much better real-time potential.

Table 5 highlights that smaller models like YOLOv8s offer better performance for real-time tasks on GuPho, especially in structured environments such as roads and buildings.

5. CONCLUSIONS

This paper presented the enhanced version of GuPho, a portable, low-cost, and AI-integrated visual SLAM-based system designed for real-time 3D mapping, object detection, and scene analysis. By combining stereo vision with DL methods - including semantic segmentation, object detection, and both monocular and stereo depth estimation - we demonstrated the system's capability to generate semantically enriched and geometrically accurate 3D reconstructions.

We evaluated different DL models, such as YOLOv8, YOLOv11, RF-DETR, Depth Anything V2, Depth Pro, and UniMatch, benchmarking their performance for object detection and size measurement. YOLOv8 was successfully integrated and tested for real-time object detection and crack analysis, exploiting the Jetson Nano capabilities and demonstrating a strong balance between detection accuracy and computational efficiency on resource-constrained hardware. Additionally, the stereo-based and monocular depth estimation approaches were compared against manually measured ground truth, with SIFT-based stereo matching and Depth Pro yielding the most consistent and accurate size estimations.

Despite the promising results, challenges remain in stereo segmentation consistency and monocular depth accuracy, especially under varying environmental conditions. Future work will aim to improve real-time inference efficiency, enhance model robustness across diverse scenes, and further integrate advanced AI models into the GuPho framework to enable further autonomous, reliable infrastructure inspection and monitoring in real-world applications.

Environment	Model	Model Type	Avg. Inference time (ms)	FPS	Notes-
Urban (Crack)	YOLOv8s	Segmentation	637.55	5	Street
Urban (Crack)	YOLOv8s	Segmentation	568.23	5	Street
Urban (Crack)	YOLOv8x	Segmentation	1254.57	5	Street
Infrastructure (Windows)	YOLOv8x	Segmentation	1848.67	5	Building
Infrastructure (Windows)	YOLOv8s	Segmentation	645.25	5	Building
Infrastructure (Windows)	YOLOv8x	Segmentation	1999.1	5	Building

Table 5: Average inference time of YOLOv8 models on GuPho in different environments.

ACKNOWLEDGMENTS

This study was partially carried out within the Interconnected Nord-Est Innovation Ecosystem (iNEST) and received funding from the European Union Next-GenerationEU (Piano Nazionale Di Ripresa e Resilienza (PNRR) – Missione 4, Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS00000043).

REFERENCES

Alsadik, B. and Karam, S., 2021. The simultaneous localization and mapping (SLAM): An overview. *Surveying and geospatial engineering journal*, 1(2), pp.1-12.

Bochkovskii, A., Delaunoy, A., Germain, H., Santos, M., Zhou, Y., Richter, S.R., Koltun, V., 2024. Depth pro: Sharp monocular metric depth in less than a second. Proc. *International Conference on Learning Representations.*

Cui, X.Z., Feng, Q., Wang, S.Z. and Zhang, J.H., 2022. Monocular depth estimation with self-supervised learning for vineyard unmanned agricultural vehicle. *Sensors*, 22(3), p.721.

Chu, G., Peng, Y., Luo, X. and Gong, J., 2023. YG-SLAM: Enhancing visual SLAM in dynamic environments with YOLOv8 and geometric constraints. *IEEE Access*, 11, pp.141421-141434.

Dong, X., Garratt, M.A., Anavatti, S.G. and Abbass, H.A., 2022. Towards real-time monocular depth estimation for robotics: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(10), pp.16940-16961.

Facenda, S., Trybała, P., Morelli, L., Padkan, N., Remondino, F., 2025. 3D robotics and LMM for vineyard inspection. *ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, in press.

Ghosh, K. and Musti, K.S., 2020, September. Integration of SLAM with GIS to model sustainable urban transportation system: A smart city perspective. *12th International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 261-267). IEEE.

Jocher G. Yolov8, 2023. https://github.com/ultralytics/ultralytics/tree/main (accessed 15 May 2025).

Jocher, G., Qiu, J. and Chaurasia, A., 2024. Ultralytics YOLO11, Version 11.0. 0.

Loupos, K., Doulamis, A.D., Stentoumis, C., Protopapadakis, E., Makantasis, K., Doulamis, N.D., Amditis, A., Chrobocinski, P., Victores, J., Montero, R. and Menendez, E., 2018. Autonomous robotic system for tunnel structural inspection and assessment. *International Journal of Intelligent Robotics and Applications*, 2, pp.43-66.

Li, G., Liu, T., Fang, Z., Shen, Q. and Ali, J., 2022. Automatic bridge crack detection using boundary refinement based on real-time segmentation network. *Structural Control and Health Monitoring*, 29(9), p.e2991.

Li, Y., Wang, Y., Lu, L. and An, Q., 2024. YOD-SLAM: An indoor dynamic VSLAM algorithm based on the YOLOv8 model and depth information. *Electronics*, 13(18), p.3633.

Liu, H., Zhang, G. and Bao, H., 2016, September. Robust keyframe-based monocular SLAM for augmented reality. *In 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (pp. 1-10). IEEE.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, pp.91-110.

Ming, Y., Meng, X., Fan, C., Yu, H., 2021. Deep learning for monocular depth estimation: A review. *Neurocomputing*, Vol. 438: 14-33.

Mur-Artal, R. and Tardós, J.D., 2017. ORB-SLAM2: An opensource slam system for monocular, stereo, and RGB-D cameras. IEEE *Transactions on Robotics, 33(5), pp. 1255-1262.*

Menna, F., Torresani, A., Battisti, R., Nocerino, E., Remondino, F., 2022. A modular and low-cost portable VSLAM system for real-time 3D mapping: from indoor and outdoor spaces to underwater environments. *ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLVIII-2/W1-2022, pp. 153-162

Padkan, N., Battisti, R., Menna, F. and Remondino, F., 2023a. Deep learning to support 3d mapping capabilities of a portable vslam-based system. *ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, 48(1), pp.363-370.

Padkan, N., Trybala, P., Battisti, R., Remondino, F. and Bergeret, C., 2023b. Evaluating monocular depth estimation methods. *ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, 48(1), pp.137-144.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. Proc. *CVPR*, pp. 779-788.

Robinson, I., Robicheaux, P., Popov, M., 2025. RF-DETR: SOTA real-time object detection model. Available at: https://github.com/roboflow/rf-detr (accessed 15 May 2025).

Sumikura, S., Shibuya, M. and Sakurada, K., 2019. OpenVSLAM: A versatile visual SLAM framework. In Proc. *27th ACM International Conference on Multimedia*, pp. 2292-2295.

Takleh, T.T.O., Bakar, N.A., Rahman, S.A., Hamzah, R. and Aziz, Z., 2018. A brief survey on SLAM methods in autonomous vehicle. *Int. J. Eng. Technol*, 7(4), pp.38-43.

Torresani, A., Menna, F., Battisti, R., Remondino, F., 2021. A V-SLAM guided and portable system for photogrammetric applications. *Remote Sensing*, Vol.13(12), 2351.

Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., Geiger, A., 2023. Unifying flow, stereo and depth estimation. IEEE *TPAMI*, 45(11), pp.13941-13958.

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J. and Zhao, H., 2024a. Depth anything: Unleashing the power of large-scale unlabeled data. Proc. *CVPR*, pp. 10371-10381.

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J. and Zhao, H., 2024b. Depth anything v2. *Advances in Neural Information Processing Systems*, 37, pp.21875-21911.

Yao, X., Hu, L. and Zhang, J., 2024. Reconstructing the local structures of Chinese ancient architecture using unsupervised depth estimation. Heritage Science, 12(1), p.318.

Yan, Z., Dong, W., Shao, Y., Lu, Y., Haiyang, L., Liu, J., Wang, H., Wang, Z., Wang, Y., Remondino, F. and Ma, Y. 2025a. Renderworld: World model with self-supervised 3d label. *Proc. IEEE ICRA*, in press.

Yan, Z., Padkan, N., Trybała, P., Farella, E.M. and Remondino, F., 2025b. Learning-based 3D reconstruction methods for non-collaborative surfaces—A metrological evaluation. *Metrology*, 5(2), p.20.

Yong, P. and Wang, N., 2022. RIIAnet: A real-time segmentation network integrated with multi-type features of different depths for pavement cracks. *Applied Sciences*, 12(14), p.7066.

Yousif, K., Bab-Hadiashar, A. and Hoseinnezhad, R., 2015. An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intelligent Industrial Systems*, 1(4), pp.289-311.

Zhang, T., Wang, D. and Lu, Y., 2023. ECSNet: An accelerated real-time image segmentation CNN architecture for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems.*

Zhou, Q.Y., Park, J. and Koltun, V., 2018. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv*:1801.09847.