The Hazard Mapping and Vulnerability Monitoring (HaMMon) Post-Event Analysis Pipeline

Leonardo Pelonero[1,*,†], Mauro Imbrosciano[1,*,†], Eva Sciacca[1,†], Fabio Vitello[1,†], Alessandra Casale[2,‡], Stefano Stalio[2,‡] and Sandra Parlati[2,‡]

[1]*INAF Astrophysical Observatory of Catania, Catania, Italy*
[2]*INFN Laboratori Nazionali del Gran Sasso, Assergi L'Aquila, Italy*

Abstract

In recent years, the monitoring and study of natural hazards have gained significant attention, particularly due to climate change, which exacerbates incidents like floods, droughts, storm surges, and landslides. Together with the constant risk of earthquakes, these climate-induced events highlight the critical necessity for enhanced risk assessment and mitigation strategies in susceptible areas such as Italy.

In this work, we present a portable and fully automated pipeline for post-event analysis on High-Performance Computing (HPC) infrastructures. Our methodology integrates Photogrammetry techniques, Data Visualization and Artificial Intelligence technologies to analyze final high-resolution 3D models of areas affected by natural disasters. This process enables the fusion and association of heterogeneous information directly onto the geometric data, creating a semantically enriched model useful to assess extreme natural events and evaluate their impact on risk-exposed assets.

Keywords

Post-Event Analysis, Hazard Mapping, Machine Learning, Photogrammetry, Mesh segmentation, Texture 3D models

1. Introduction

The monitoring and analysis of natural hazards have gained increasing relevance in recent years, as their frequency and intensity continue to rise, particularly under the influence of climate change, which intensifies the frequency and severity of events such as floods, droughts, storm surges, and landslides. These climate-driven phenomena highlight the urgent need for more effective risk assessment and mitigation strategies, especially in highly vulnerable areas like Italy.

Hazard mapping plays a central role in this context, as it enables the identification and spatial representation of risk-prone areas. Having a ready-to-use and easily deployable analysis pipeline is of paramount importance when responding to disastrous events. To address this need, automated pipelines that orchestrate photogrammetric workflows and AI-driven image preprocessing are essential to support timely decision-making.

The National Institute for Astrophysics (INAF), together with other national institutions, is partner in the HaMMon (Hazard Mapping and vulnerability Monitoring) project — an initiative launched within the Italian National Research Centre for High Performance Computing, Big Data and Quantum Computing (ICSC)[1] and coordinated by UnipolSai[2]. The project aims to develop advanced tools and methodologies for the management of natural hazards, addressing all aspects from risk assessment to post-event analysis, including intervention planning and damage estimation.

In this paper, we present a portable and reusable post-event analysis framework based on Photogrammetry, AI and Data Visualization into a single orchestrated pipeline within an HPC platform, to assess

ITADATA-WS 2025: The 4th Italian Conference on Big Data and Data Science – Workshops, September 9–11, 2025, Turin, Italy
leonardo.pelonero@inaf.it (L. Pelonero); mauro.imbrosciano@inaf.it (M. Imbrosciano); eva.sciacca@inaf.it (E. Sciacca); fabio.vitello@inaf.it (F. Vitello); alessandra.casale@lngs.infn.it (A. Casale); stefano.stalio@lngs.infn.it (S. Stalio); sandra.parlati@lngs.infn.it (S. Parlati)
0000-0002-0002-3128 (L. Pelonero); 0009-0009-7427-6900 (M. Imbrosciano); 0000-0002-5574-2787 (E. Sciacca); 0000-0003-2203-3797 (F. Vitello)

[1]ICSC: https://www.supercomputing-icsc.it/
[2]UnipolSai: https://www.unipolsai.it/

extreme natural events and analyze their effects on assets at risk.

The next Section 2 details the HaMMon computing platform, outlining its dual architecture composed of a Kubernetes cluster for scalable, containerized workloads and Slurm-based HPC cluster for batch processing. Section 3 outlines the overall post-event analysis pipeline of our proposed approach, from UAV images acquisition to web application. Section 4 shows our preliminary results. Section 5 draws our conclusions and outlines future works.

2. HaMMon HPC Infrastructure

The HaMMon computing platform is hosted on INFN resources at the Laboratori Nazionali del Gran Sasso (LNGS) and managed by the LNGS Computing and Network Service. The LNGS HPC infrastructure comprises two primary clusters: a Kubernetes[3] cluster dedicated to the HaMMon project, augmented by central services developed by the INFN Data Cloud working group, and a general purpose SLURM-based[4] HPC cluster, available for HaMMon research groups. The LNGS HPC infrastructure is connected at 10Gb/s to the national research network GARR[5]; this connection will upgrade at 100Gb/s in the next few months.

The following sections provide a technical description of the two clusters depicted in Figure 1.

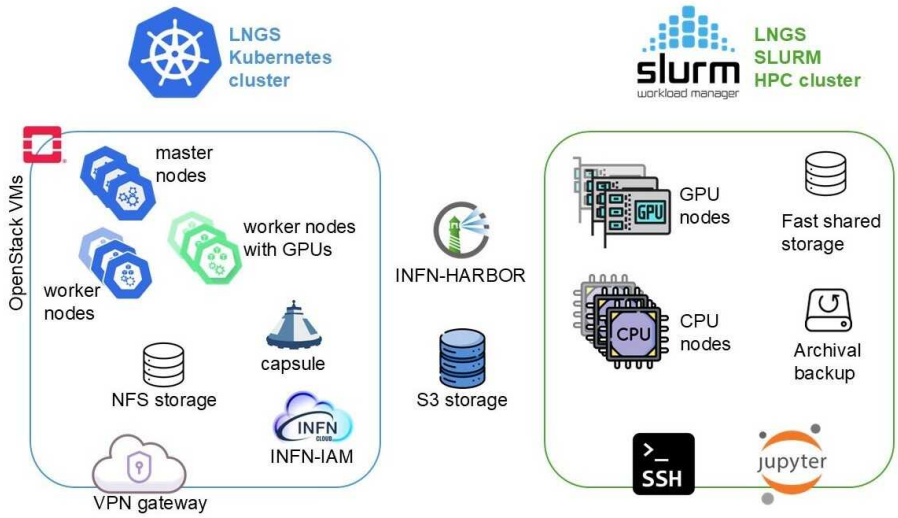


Figure 1: HaMMon HPC Infrastructure hosted at INFN-LNGS

2.1. Kubernetes Cluster on OpenStack

The cluster is built on OpenStack[6] virtual machines and orchestrated with Rancher's RKE2[7]. High availability is ensured by three master nodes, while worker nodes can be scaled dynamically to match workload demands. Deployment and lifecycle management are fully automated via a Puppet module developed by INFN. One hypervisor equipped with four Nvidia H100 GPUs is reserved for the HaMMon workloads: each Nvidia H100 GPU is accessible through a Kubernetes worker node.

User authentication is handled by a custom INFN webhook built upon the official Kubernetes OIDC plugin (kubelogin [8]) and backed by the INFN Identity and Access Management system (INFN-IAM).

[3] https://kubernetes.io/
[4] https://slurm.schedmd.com/
[5] https://www.garr.it/it/
[6] https://www.openstack.org/
[7] https://docs.rke2.io/
[8] https://github.com/int128/kubelogin

Fine-grained, multi-tenant access control is enforced through Capsule[9], so that each institute operates in an isolated workspace (called capsule tenant) with its own namespaces, Network File System (NFS) storage, and applications.

The storage services comprise two tiers. For short-term needs, an NFS CSI provisioner delivers more than 20 TB of storage; the migration from NFS to a CephFS[10] backend is planned in the next months. For long-term retention, an S3-compatible object store on INFN-LNGS resources supports large-scale data retention across all HaMMon partners.

A Harbor[11] registry, deployed and managed by INFN Data Cloud and secured by INFN credentials, hosts Docker[12] images for all users.

To safeguard user-deployed services, at the moment a VPN gateway (integrated with INFN-IAM) restricts access so that no user service is directly exposed to the public Internet.

2.2. SLURM-Based HPC Cluster

The SLURM batch-mode HPC cluster consists of 72 CPU nodes (Lenovo NeXtScale nx360 M5) plus one node hosting four NVIDIA A100 GPUs. A significant increase in computing power is planned in a few months with the addition of 300 CPU nodes and five nodes hosting four Nvidia H100 GPUs each. Interconnects include a 100 Gb/s Intel Omni-Path network, a 100 Gb/s Ethernet fast interconnection network, and, in the next few months, a 400 Gb/s Infiniband network.

The storage architecture features a high-performance shared filesystem of about 350 TB accessible by all nodes. A significant increase of the order of 5 PB in available storage and performance improvement is planned in a few months. Long-term data retention is ensured by an archival backup system providing 5 PB of LTO9 tape.

The software stack is based on Linux Rocky[13] 8.6 and includes compiler suites such as OpenMPI[14] and Intel MPI[15], together with core libraries like MPI and CUDA and a broad set of scientific packages managed via Spack[16]. User environments are configured through environment modules, while SLURM handles job scheduling, priority management, and accounting.

Two public SSH login nodes and two Jupyter Notebook[17] gateway nodes provide command-line and web-based access, respectively. Authentication uses INFN-LNGS credentials; authorization leverages UNIX group membership with a shared project filesystem for group members.

The LNGS HPC infrastructure health and performance are monitored via Checkmk[18] at the cluster level, Prometheus[19] and Grafana[20] collect and visualize in-cluster metrics, while accounting tools track resource usage. Together, these two complementary clusters deliver to the HaMMon project a flexible, scalable, and secure HPC environment.

3. Post-event analysis pipeline

The unpredictable nature of damage from natural disasters to assets and infrastructure makes post-event assessment a critical yet complex task for stakeholders like insurance companies and public administrations.

[9] https://projectcapsule.dev/
[10] https://docs.ceph.com/
[11] https://goharbor.io/
[12] https://www.docker.com/
[13] https://rockylinux.org/
[14] https://www.openmp.org/
[15] https://www.intel.com/content/www/us/en/developer/tools/oneapi/mpi-library.html
[16] https://spack.io/
[17] https://jupyter.org/
[18] https://checkmk.com/it
[19] https://prometheus.io/
[20] https://grafana.com/

A significant obstacle in this process is the lack of standardized methodologies capable of addressing the diverse impacts of different catastrophic events. To overcome this limitation, this work introduces a versatile pipeline designed to detect and to evaluate results that are applicable across various types of natural hazards. The primary objective is to significantly assist damage assessment, reducing the time-consuming and costly on-site inspections.

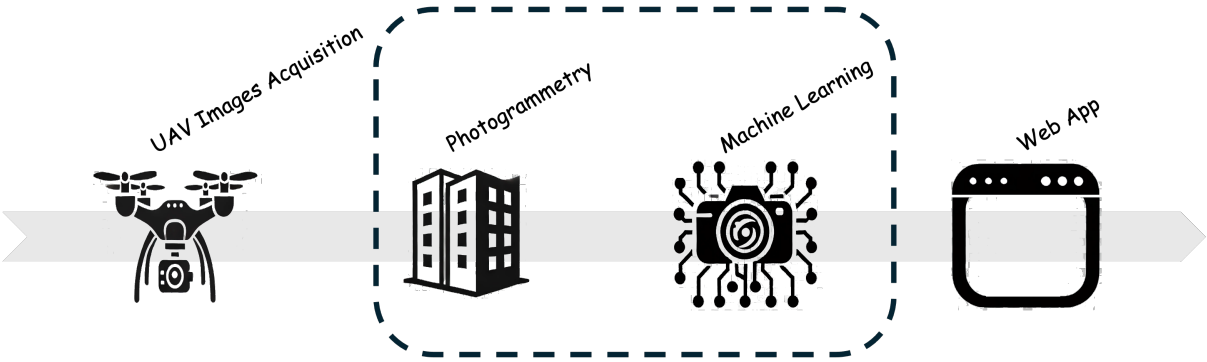


Figure 2: Post-event analysis pipeline

The pipeline involves the following steps depicted in Figure 2 (see also [1] for more details):

1. **UAV Images Acquisition** To ensure both effectiveness and precision in the final results, the initial phase involves planning and execution of drone surveys over the affected area. This process includes delineating the area of interest and setting drone flight specifics—such as altitude, flight path, and coverage area.
2. **Photogrammetry** The collected imagery is used to generate high-resolution (centimeter-scale) 3D models using the "Aerial Structure-from-Motion" (ASfM) techniques [2, 3]. This photogrammetric approach autonomously determines the geometry of the area, as well as the position and orientation of the cameras. By leveraging overlapping images captured from multiple viewpoints, it enables the reconstruction of detailed 3D models and produces georeferenced maps of the examined area (see [4] for more details).
3. **Machine Learning** The process is complemented by Artificial Intelligence (AI) algorithms that extract features from aerial images and integrate them into the model, enriching it with semantic information.
4. **Web App** These steps aim to detect and classify damages in 3D models, offering a scalable framework for post-disaster analysis. The dataset of augmented digital twins models provides stakeholders and claims adjusters with detailed visual references for remote damage assessment and is accessible through a user-friendly web application. The platform allows for the immediate 3D visualization of the analysis results, powered by Cesium[21], facilitating a prompt understanding of the damage location and extent. Furthermore, the resulting data are downloadable, allowing further offline analysis and integration into other systems.

3.1. Machine Learning: implementation

The HaMMon project employs machine learning algorithms to enrich 3D photogrammetric models with semantic information extracted from aerial imagery (see [5] for more details). Deep learning models generate black and white segmentation masks that highlight features such as buildings, roads, and damaged or flooded areas.

At this stage, we rely on publicly available datasets for semantic segmentation of post-disaster scenarios, such as *RescueNet* [6] and *FloodNet* [7] , to train our models. We selected the *Tiramisu* [8] and

[21] CesiumJS: https://cesium.com/platform/cesiumjs/

Attention-UNet [9] Convolutional Neural Networks (CNN) for their good balance between performance and computational requirements, with the latter showing superior results.

Neural network training is performed on the SLURM cluster, leveraging high-capacity GPUs, such as the NVIDIA A100 which count 80 GB of memory each. The ability to use high-capacity GPUs is critical for enabling effective batch normalization, a crucial technique to stabilize training and mitigate overfitting. This is particularly important in the context of high-resolution UAV imagery, where images cannot be significantly downscaled or cropped without losing critical details, and CNNs, which are inherently memory-intensive. The use of libraries like PyTorch[22]Distributed Data Parallel (DDP) further enhances efficiency by enabling training distribution across multiple processes, effectively combining the memory of different GPUs.

Model predictions are performed on the Kubernetes cluster to be integrated into the photogrammetry pipeline. Since inference is executed each time on the entire dataset, it is crucial to run it in parallel across multiple images, leveraging vectorization and GPU acceleration. Different steps—such as image downscaling, patch division, and mask upscaling—are carried out on separate pods, thereby freeing GPU resources for photogrammetry tasks.

3.2. SfM: parallel implementation

This section outlines the architectural design and implementation for deploying the photogrammetry workflow based on Agisoft Metashape within a Kubernetes framework. The work focused on three key engineering objectives for workflow orchestration:

- Creating a compatible and portable execution environment for Agisoft Metashape. This involves packaging Metashape and all its dependencies into Docker images. This approach guarantees that the workflow executes reliably and predictably, regardless of the specific hardware configuration of a node, while also simplifying deployment.
- Developing a Python-based workflow to efficiently manage the parallel execution of Metashape. This project involves designing and implementing a master-worker architecture in Python, following Agisoft's official documentation[23] for distributed processing that adapts to any number of available worker nodes.
- Prioritizing resource management and performance optimization across the diverse Kubernetes cluster. This includes analyzing the resource consumption (CPU, RAM, GPU) of the Metashape workflow on Kubernetes infrastructure. The goal is to devise strategies that maximize throughput while navigating the cluster's heterogeneous nature and the Agisoft Metashape floating license limitation.

The application software is packaged through a modular Docker containerization strategy, ensuring flexibility and maintainability. License management is addressed with a pragmatic and transparent approach based on manual scaling of worker deployments, giving the operator full control over active computational resources. The complete details of the implementation can be found in [10].

To validate the efficiency and scalability of the parallel and distributed task management implementation, an experimental validation was conducted at the cloud-HPC infrastructure based on OpenStack and Kubernetes, implemented in the green HPC4AI@UNITO data center at the University of Turin [1]. The entire photogrammetric workflow, from image import to final model export, was executed on a cluster comprising three worker nodes equipped with Tesla T4 GPUs with 16 GB of memory. This evaluation utilized a dataset consisting of approximately 2,099 UAV images, representative of Tredozio's Square, obtained following a survey specifically conducted to document and analyze the impacts of landslides, seismic activity, and flooding caused by the Tramazzo river.

[22] https://pytorch.org/

[23] How to configure the network processing (Agisoft Helpdesk Portal): https://agisoft.freshdesk.com/support/solutions/articles/31000145918-how-to-configure-the-network-processing

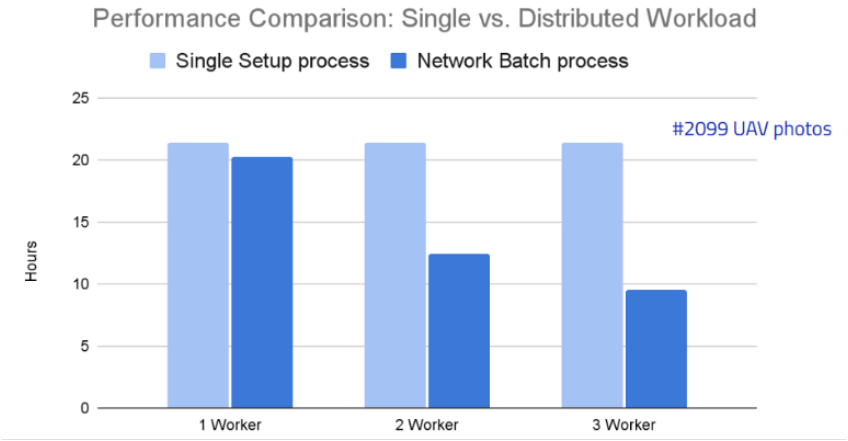


Figure 3: Comparative performance of the SfM workflow across one, two, and three worker nodes on 2099 UAV photos

The results demonstrated a significant acceleration in processing times compared to single-node executions. Figure 3 presents a bar chart that directly compares the total processing times and performance of the Structure from Motion (SfM) workflow processing the same dataset. The chart compares the total processing time of the entire workflow executed with two different methods: a standard standalone execution on a single node (baseline, shown in light blue) and our distributed network process using a master-worker architecture (shown in dark blue). The standalone execution completed the task in 21 hours and 40 minutes, a result that is comparable to the 20 hours taken by our distributed architecture using only a single worker node. The key benefits and strong scalability of our implementation become evident when multiple workers participate in the process. As shown in the Figure, by introducing a second worker the execution time is nearly halved to approximately 12 hours, and with a third worker it is further reduced to under 10 hours, confirming the efficiency of our parallelization strategy.

This visual comparison clearly illustrates how the processing times are effectively halved as additional computational nodes are introduced, highlighting the direct benefits of parallelization.

It should be noted that the experimental validation on this initial infrastructure was limited to three workers due to the practical constraints imposed by the commercial Agisoft Metashape floating license.

The application has since been migrated to the HaMMon HPC Infrastructure reported in Section 2.

In the next Section 4, we present the quantitative results of the training process for the ML module and a comparative analysis of two alternative approaches for 3D semantic classification. The two methods were evaluated on the same use case: the first performs segmentation on 3D model, while the second uses classification primitives on tiled models for the CesiumJS platform.

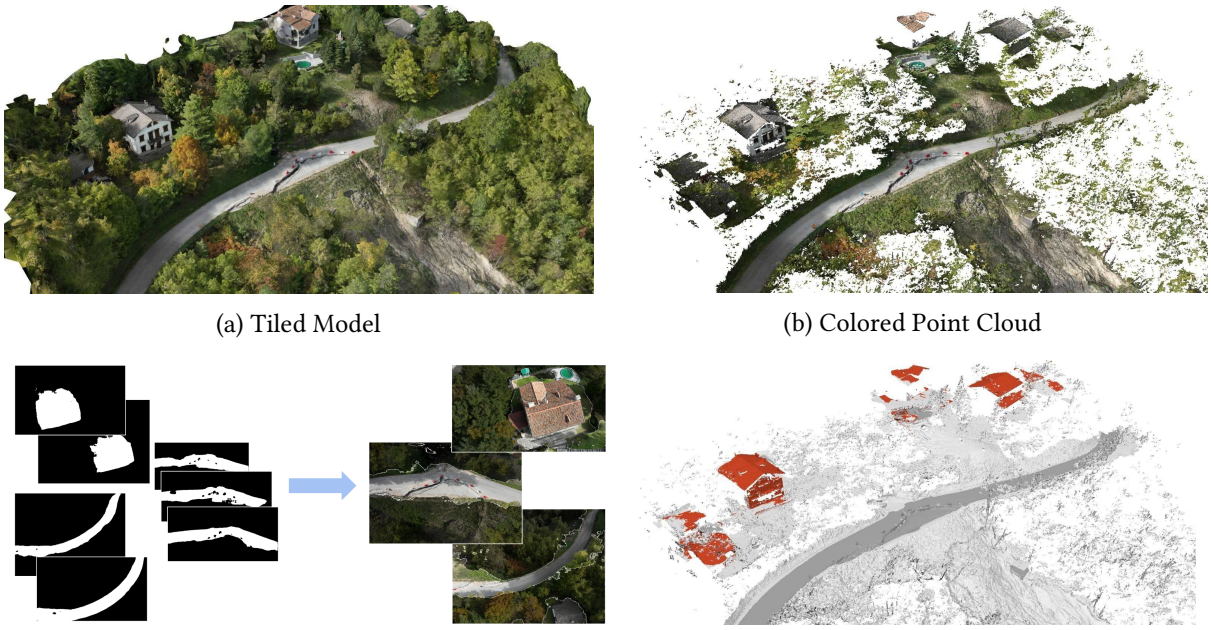
4. Results

4.1. Digital Twin products

The dataset for this study was acquired during a survey conducted in the municipality of Tredozio (province of Forlì-Cesena, Italy) in October 2024. This survey was carried out as part of the HaMMon project, coordinated by INAF-OACT, UNIMIB and ENEA[24].

Data acquisition was performed using Unmanned Aerial Vehicles (UAVs) equipped with a camera and Real-Time Kinematic (RTK) modules to produce directly georeferenced imagery. The overlapped images obtained are fundamental for UAV-based photogrammetry, emerging as a particularly effective solution for disaster management in remote sensing [11, 12]. This capability is fundamental for post-event analysis, timely risk assessment, and effective mitigation measures planning.

[24] *Università Degli Studi Milano-Bicocca* (UniMiB) and *Agenzia nazionale per le Nuove Tecnologie, l'Energia e lo Sviluppo Economico Sostenibile* (ENEA) are project partners.



(a) Tiled Model

(b) Colored Point Cloud

(c) ML-generated segmentation mask for building and road detection

(d) Point cloud classification of buildings (red) and road (grey)

Figure 4: Visual outputs generated by the workflow on the survey carried out in Tredozio (Monte Busca area)

Data acquired on-site in Monte Busca area were processed using the HaMMoN platform. Figure 4 summarizes the primary outputs generated through our workflow. The top-left quadrant shows the high-detail tiled 3D model. The model clearly reveals the impact of the landslide. The affected area was estimated to be approximately 30 meters in width and 110 meters in depth. The top-right quadrant displays the corresponding colorized dense point cloud. The result lacks fine detail due to the presence of dense vegetation, which introduces irregularities and noise in the photogrammetric reconstruction process.

The Figure 4 also presents the preliminary outputs from the integrate Artificial Intelligence classification module. In the bottom-left quadrant, represents how the binary segmentation masks generated by the module fit precisely on the original UAV images. The bottom-right quadrant shows the classified point cloud, where features such as buildings and roads are identified in red and grey, obtained by projecting all computed masks from the input aligned cameras onto the 3D model. The results are noticeably affected by the fact that the input images differ significantly from the training domain, both in terms of represented objects and image characteristics such as angle and distance. The development is still at an early stage, and more robust models with improved generalization capabilities will be employed in future iterations. Nevertheless, these outcomes offer a solid starting point for future development and serve as a valuable test for assessing the integration and coherence of the automated workflow.

4.2. Machine Learning Outcomes

The main challenges encountered in training deep learning models for UAV-based post-disaster analysis were the limited number of available images and the strong imbalance among semantic classes. To mitigate the scarcity of training examples, we explored transfer learning [13] by pre-training networks on the larger RescueNet dataset and then fine-tuning them on FloodNet. We tested three fine-tuning strategies—TL0, TL1, and TL2—differing in the number of frozen layers: TL0 retrains the entire network, TL2 updates only the final classification layer, while TL1 represents an intermediate strategy, defined specifically for each architecture.

Quantitative results are reported (see Tab. 1) in terms of accuracy and mean Intersection over Union

RescueNet			
Model	Setup	Accuracy (%)	mIoU
Tiramisù	Baseline	82.4	52.8
Attention U-Net	Baseline	84.7	57.3
FloodNet			
Model	Setup	Accuracy (%)	mIoU
Tiramisù	Baseline	83.9	53.2
Tiramisù	TL-0	87.7	63.6
Tiramisù	TL-1	85.6	59.4
Tiramisù	TL-2	83.9	55.5
Attention U-Net	Baseline	85.1	59.9
Attention U-Net	TL-0	86.9	63.9
Attention U-Net	TL-1	85.1	60.2
Attention U-Net	TL-2	61.1	22.0

Table 1
Accuracy and mIoU for Tiramisù and Attention U-Net on RescueNet and FloodNet, for baseline models and for three different transfer learning setups (TL0, TL1, TL2).

(mIoU) [25]. The best performances were consistently achieved with TL0, where the entire network was fine-tuned on FloodNet, both for Tiramisù and Attention U-Net. This confirms that re-adapting all convolutional filters is beneficial, as the two datasets, though similar, have noticeable domain differences. Overall, transfer learning experiments outperformed the baselines, underlining the impact of limited training data.

As shown in the confusion matrices (see, as an example, Fig. 5) the networks generally perform well, with the most significant misclassifications occurring in categories that involve a degree of subjectivity—such as the damage scale of buildings—or that depend on the observer's viewpoint, as in the case of roads that may appear either "blocked" or "clear" depending on the angle of the image.

Despite the differences in acquisition setups—RescueNet and FloodNet provide strictly nadiral UAV views, whereas HaMMon relies on multi-angle imagery for photogrammetry—the training phase yielded encouraging results and provides a solid baseline for validating the workflow and demonstrate the feasibility of integrating machine learning within the HaMMon pipeline.

4.3. Semantic Visualization of 3D Models

The true potential of this workflow is realized by enriching the geometric models from Structure from Motion (SfM) with semantic context from Machine Learning (ML). This integration process transforms bidimentional data segmentation labels into an intuitive visual 3D representation, highlighting specific features identified during the analysis. The two distinct methodologies detailed in this section are extensions of our previously validated, Python-based photogrammetric workflow [1], designed to bridge the gap between semantic data and its final 3D representation. This section details two distinct methodologies that use the classified point cloud as input to achieve this integration[26]: (1) direct mesh recolorization for static, high-fidelity models, and (2) dynamic styling of tiled 3D models for scalable web-based applications.

4.3.1. 3D Wireframe from dense cloud classification

This first approach embeds the segmentation data directly into the 3D model texture by altering the vertex colors of the source dense point cloud within the photogrammetric pipeline. This method is particularly suited for generating self-contained mesh where classified features must be permanently

[25] Accuracy measures the fraction of correctly classified pixels, while mIoU is the mean ratio of overlap between predicted and ground truth regions across classes.

[26] The source code and implementation details for both methodologies are publicly available in our GitHub repository: https://github.com/Fliki1/Cesium-3D-Tile-ClassificationPrimitive.

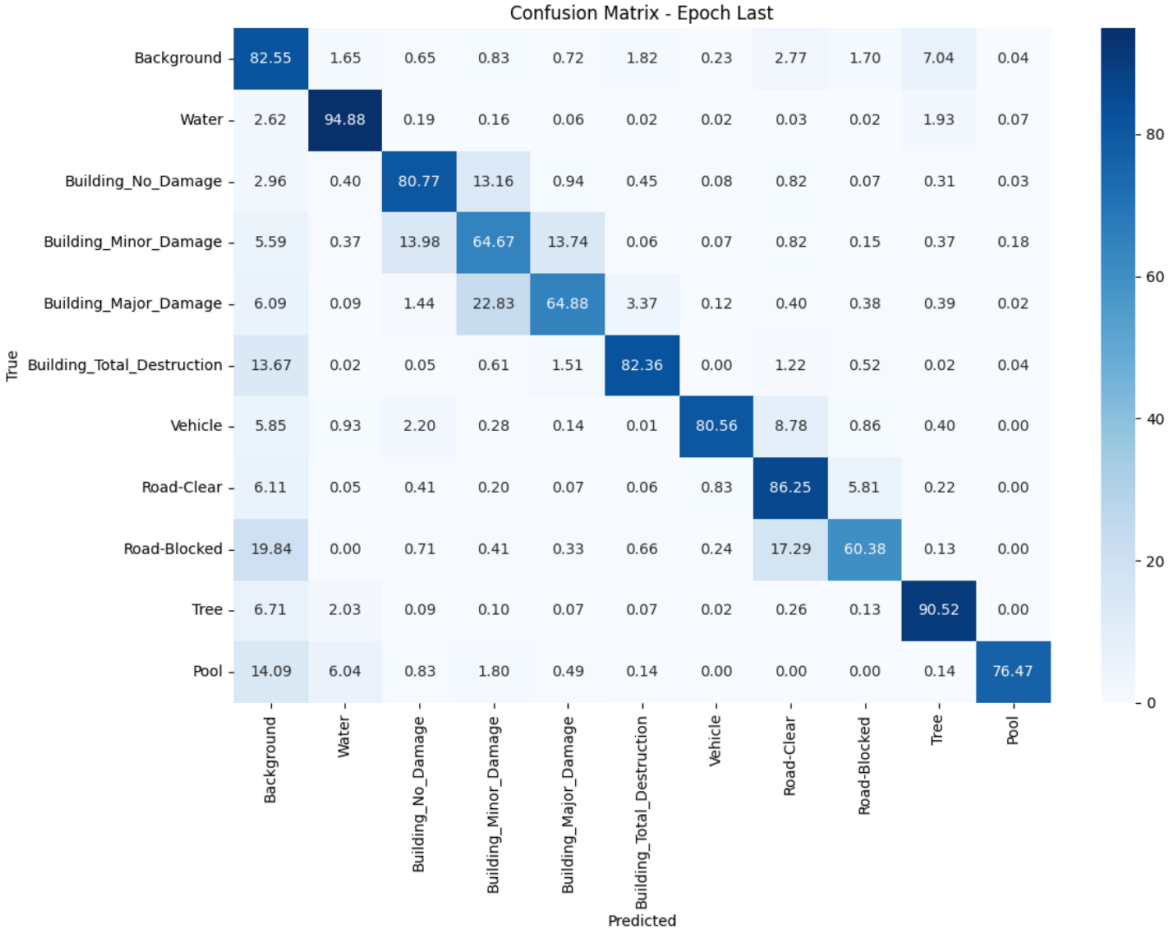


Figure 5: Confusion matrix of the Attention U-Net on RescueNet. Each row corresponds to a true class and shows the fraction of predictions for each category. The matrix is row-normalized, so the entries in each row add up to one.

and clearly delineated. The workflow was implemented using Agisoft Metashape Python module and consists of the following steps.

Initially, a subset of the dense point cloud, containing only the points belonging to a desired class (e.g., “building”), is exported. The color attributes of these points are then automatically overwritten. A Python script was developed to modify the RGB values of the points based on their classification ID. The mapping between class IDs and their new colors is defined in an external user-editable JSON configuration file, allowing for flexible color assignments.

Once recolored, this dense point cloud subset is re-imported into the Metashape project. The 3D model generation process is then executed, with the point cloud set as the data source for the mesh. The algorithm thus generates the new mesh surface by interpolating the points from the modified cloud, projecting the color attributes of the points onto the corresponding areas of the model’s texture.

As shown in Figure 6, the final output is a static 3D model where segmented areas are immediately distinguishable (buildings colored red and the road surface dark gray) making it for visual reports and detailed offline analysis. It should be noted that the resulting mesh is not perfectly optimal. It presents holes and extraneous artifacts, which are primarily attributable to the known challenges of photogrammetric reconstruction in areas with high vegetation density [14]. Foliage creates occlusions and complex, non-static surfaces, leading to a point cloud with inherent gaps and inaccuracies that are inevitably inherited by the generated 3D mesh.

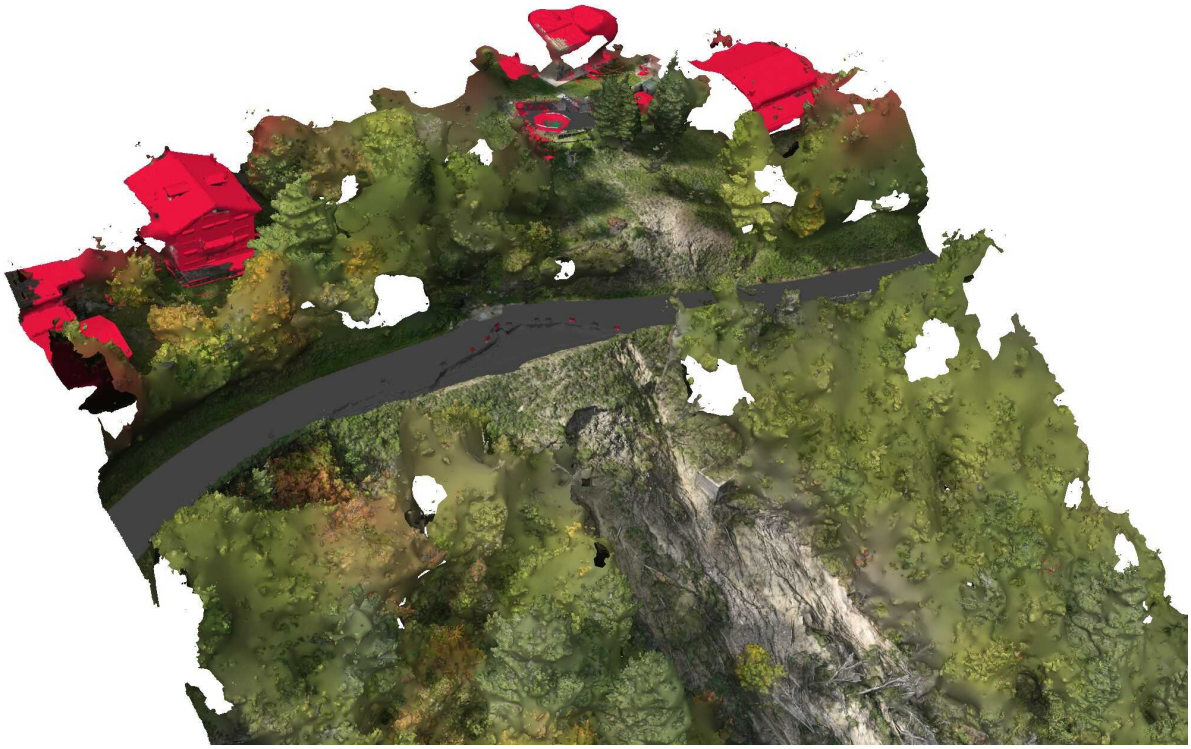


Figure 6: 3D model generated from classified dense point cloud

4.3.2. PrimitiveClassification with CesiumJS

This second methodology takes a different approach, shifting the challenge of data integration from a pre-processing step to the server-side web environment. This addresses the need for scalable and interactive visualization, especially in web applications handling large-scale 3D datasets, such as urban areas or complex infrastructure. This approach leverages CesiumJS, a robust open-source JavaScript framework for 3D maps, to apply dynamic styling to tiled 3D models generated from the SfM pipeline. Tiled models are crucial for performance, as they employ a Level of Detail (LOD) mechanism that optimizes rendering by loading progressively detailed versions based on viewing distance. They enable efficient data streaming, as only the visible and relevant tiles are requested from the server, significantly reducing network bandwidth and client-side memory consumption.

This method, operating on cloud computing, overlays styling information directly at runtime, inherently without altering the original 3D tiled model. The process begins by exporting the coordinates of the classified points of interest (from a .las file format) into a standard GeoJSON format. The process consists of mapping each coordinate from the GeoJSON file to a geometric primitive supported by Cesium.

Within the web application, the base 3D tileset is loaded first. Subsequently, the GeoJSON file is fetched, and its coordinates are used to create visual markers. For each feature, the script instantiates a Cesium.GeometryInstance in the form of a small ellipsoid (1 cm radius). This small volume acts as a 3D spatial marker precisely located at the position of the classified point. Each of these instances is assigned specific visual attributes, such as a semi-transparent color, which can be customized to represent different classes or types of features.

As shown in Figure 7, the final result is a smooth integration where classified areas are highlighted directly on the original 3D model. This approach is exceptionally powerful as it does not alter the source data, allows for toggling different classification layers on and off, and leverages the performance of the tiled model's Level of Detail (LOD) system, ensuring a fluid and interactive user experience even with massive datasets.

Figure 7: CesiumJS Classification Primitive on building area marked with a transparent yellow overlay

5. Conclusions and future work

In conclusion, this work presented a structured post-event analysis pipeline, integrating photogrammetric processing and machine learning semantic segmentation.

The proposed framework was validated on real-world case study documenting landslide impacts in Tredozio, Italy, successfully generating classified 3D digital twins from UAV imagery.

Building upon the initial POCI infrastructure of Turin, the application has been transitioned to the HaMMon HPC Infrastructure, setting the stage for the next phase of our research. Future efforts will be dedicated to conducting a comprehensive performance analysis in this new environment, benchmarking the entire workflow, from data ingestion to model generation, to quantify the performance gains provided by the new hardware.

A fundamental contribution of this work is the development of two distinct approaches and methodologies to link 2D semantic labels and 3D visualization: a direct mesh building from dense point cloud and an interactive web-base application using CesiumJS classification primitive. We acknowledge that the current ML results are at an early stage. They are affected by the input images that differ significantly from the training domain, both in terms of represented objects and image characteristics such as angle and distance. To address this, future efforts will be dedicated to improving the model's robustness by expanding the training dataset to include a wider variety of scenes, objects, and imaging conditions.

Acknowledgments

This work is supported by the Spoke 1 "FutureHPC & BigData" and the Spoke 3 "Astrophysics and Cosmos Observations" of the ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by NextGenerationEU.

The authors are grateful to the team of Computing and Networking System Service of the Laboratori Nazionali del Gran Sasso (LNGS - National Institute of Nuclear Physics) for sharing the HPC cluster resources within the project ICSC - Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing (Codice Progetto CN00000013 - PNRR Missione 4, Componente 2, Investimento 1.4- ICSC - CUP I53C21000340006).

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4 in order to: Grammar and spelling check. After using these tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. Imbrosciano, E. Sciacca, F. Vitello, L. Pelonero, F. Franchina, U. Becciani, I. Colonnelli, D. Medić, The cloud-hpc infrastructure for hazard mapping and vulnerability monitoring (hammon), in: 2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), IEEE, 2025, pp. 309–316.

[2] F. Agüera-Vega, F. Carvajal-Ramírez, P. Martínez-Carricondo, Accuracy of digital surface models and orthophotos derived from unmanned aerial vehicle photogrammetry, Journal of Surveying Engineering 143 (2017) 04016025.

[3] F. Knuth, D. Shean, S. Bhushan, E. Schwat, O. Alexandrov, C. McNeil, A. Dehecq, C. Florentine, S. O'Neel, Historical structure from motion (hsfm): Automated processing of historical aerial photographs for long-term topographic change analysis, Remote Sensing of Environment 285 (2023) 113379.

[4] F. R. Vitello, L. Pelonero, M. Imbrosciano, UAV-Based Digital Twin Creation: Algorithms for Advanced Processing, Classification, and 3D Model Enrichment, 2025. URL: https://doi.org/10.15161/oar.it/eb6em-v6t60. doi:10.15161/oar.it/eb6em-v6t60.

[5] M. Imbrosciano, E. Sciacca, L. Pelonero, F. Vitello, Semantic Segmentation of UAV Imagery for Post- Disaster Damage Assessment, 2025. URL: https://doi.org/10.15161/oar.it/de5rk-qp426. doi:10.15161/oar.it/de5rk-qp426.

[6] M. Rahnemoonfar, T. Chowdhury, R. R. Murphy, Rescuenet: A high resolution uav semantic segmentation dataset for natural disaster damage assessment, Scientific Data 10 (2023) 913. URL: https://doi.org/10.1038/s41597-023-02799-4. doi:10.1038/s41597-023-02799-4.

[7] M. Rahnemoonfar, T. Chowdhury, A. Sarkar, D. Varshney, M. Yari, R. R. Murphy, Floodnet: A high resolution aerial imagery dataset for post flood scene understanding, IEEE Access 9 (2021) 89644–89652. URL: https://doi.org/10.1109/ACCESS.2021.3090981. doi:10.1109/ACCESS.2021.3090981.

[8] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, Y. Bengio, The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1175–1183. URL: https://arxiv.org/abs/1611.09326. doi:10.1109/CVPRW.2017.156.

[9] O. Oktay, J. Schlemper, L. Le Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, D. Rueckert, Attention u-net: Learning where to look for the pancreas, arXiv preprint arXiv:1804.03999 (2018). doi:10.48550/arXiv.1804.03999.

[10] L. Pelonero, M. Imbrosciano, E. Sciacca, F. R. Vitello, Parallel Photogrammetry with Metashape on Kubernetes, 2025. URL: https://doi.org/10.15161/oar.it/5prtw-4ca62. doi:10.15161/oar.it/5prtw-4ca62.

[11] N. Casagli, W. Frodella, S. Morelli, V. Tofani, A. Ciampalini, E. Intrieri, F. Raspini, G. Rossi, L. Tanteri, P. Lu, Spaceborne, uav and ground-based remote sensing techniques for landslide mapping, monitoring and early warning, Geoenvironmental Disasters 4 (2017) 9.

[12] M. Kucharczyk, C. H. Hugenholtz, Remote sensing of natural hazard-related disasters with small drones: Global trends, biases, and research opportunities, Remote Sensing of Environment 264 (2021) 112577.

[13] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proceedings of the IEEE 109 (2020) 43–76.

[14] M. Chen, A. Feng, K. McCullough, P. B. Prasad, R. McAlinden, L. Soibelman, Semantic segmentation and data fusion of microsoft bing 3d cities and small uav-based photogrammetric data, arXiv preprint arXiv:2008.09648 (2020).